

Хмельницький національний університет
Факультет програмування
та комп'ютерних і телекомунікаційних систем
Кафедра комп'ютерної інженерії та системного програмування

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Синтез та моделювання операційного автомату на основі автомату Мура
Назва теми

КвРКІ. 190362.17.03.09 ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія»

Назва

Виконав: студент IV курсу, група KI-17-3



Підпис

Д. А. Любовецький
Ініціали, прізвище

Керівник



Підпис, дата

О. М. Березький
Ініціали, прізвище

Нормоконтролер



Підпис, дата

С.М. Лисенко
Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
інженерії та системного
програмування



Підпис

Т.О. Говорущенко
Ініціали, прізвище

« 2 » червня 2021 р.

Хмельницький 2021

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ПРОГРАМУВАННЯ ТА КОМП'ЮТЕРНИХ І ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА СИСТЕМНОГО ПРОГРАМУВАННЯ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЯ ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

“ 11 ” 01 2021 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Любовецькому Денису Андрійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Синтез та моделювання операційного автомату на основі автомату Мура

Керівник проекту (роботи) Березький О.М., д.т.н., проф.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 05.02.2021 р. № 11

2. Строк подання студентом проекту (роботи) на кафедру 07.06.2021 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Дослідження предметної області та постановка задачі

Моделювання та проектування операційного автомату на основі автомату Мура

Апаратна реалізація операційного автомату на основі автомату Мура





5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Логічні схеми алгоритмів

Мікрокомандна схема та граф-схема переходів автомату Мура

Схема електрична функціональна операційного автомату

6. Консультанти розділів дипломного проекту (роботи)

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---------------|---|--|---|
| | | завдання видав | завдання прийняв |
| Нормоконтроль | Лисенко С.М., професор кафедри КІСП |  |  |
| Антиплагіат | Нічепорук А.О., доцент кафедри КІСП |  |  |

7. Дата видачі завдання « 11 » 01 2021 р.

КАЛЕНДАРНИЙ ПЛАН

| №з/п | Назва етапів (розділів) дипломного проекту (роботи) | Термін виконання етапів проекту (роботи) | Примітка |
|------|---|--|----------|
| 1 | Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником | 11.01.2021 | виконано |
| 2 | Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження | 01.02.2021 | виконано |
| 3 | Робота над розділом 1 – дослідження предметної області та постановка задачі | 01.03.2021 | виконано |
| 4 | Робота над розділом 2 – моделювання та проектування операційного автомату на основі автомату Мура | 01.04.2021 | виконано |
| 5 | Робота над розділом 3 – апаратна реалізація операційного автомату на основі автомату Мура | 30.04.2021 | виконано |
| 6 | Оформлення пояснювальної записки згідно вимог | 31.05.2021 | виконано |
| 7 | Попередній захист ВКР | 02.06.2021 | виконано |
| 8 | Захист ВКР на засіданні ЕК | Червень 2021 року | |

Студент

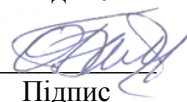


Підпис

Д. А. Любовецький

Ініціали, прізвище

Керівник роботи



Підпис

О. М. Березький

Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Синтез та моделювання операційного автомату на основі автомату Мура».

Автор роботи: Любовецький Денис Андрійович.

Керівник роботи: Березький Олег Миколайович.

Пояснювальна записка: 64 с., 24 рис., 10 табл., 3 дод., 42 джерела.

Графічна частина: 7 презентаційних слайдів.

ОПЕРАЦІЙНИЙ АВТОМАТ, АВТОМАТ МУРА, МІКРОПРОГРАМА, МІКРООПЕРАЦІЯ. МІКРОМАНДА, ГРАФ-СХЕМА ПЕРЕХОДІВ.

Метою роботи є синтез та моделювання операційного автомату на основі автомату Мура.

Об'єктом дослідження є програмно-технічний (апаратний) засіб – операційний автомат на основі автомату Мура.

Предметом дослідження є формалізований опис та схеми операційного автомату на основі автомату Мура.

Практичне значення має змодельований, спроектований та реалізований операційний автомат на основі автомату Мура, який використовується для обчислення кількості парних елементів, що належать інтервалу $[-3; 13]$ у двовимірному масиві $A[n, m]$.






Підпис студента

02.06.2021

Дата

ЗМІСТ

| | |
|--|----|
| СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ | 3 |
| ВСТУП..... | 4 |
| 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ .. | 7 |
| 1.1 Операційний та керуючий автомати | 7 |
| 1.2 Автомати Мілі та Мура | 15 |
| 1.3 Схеми електричні функціональна та принципова | 20 |
| 1.4 Висновки. Постановка задачі..... | 24 |
| 2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ОПЕРАЦІЙНОГО АВТОМАТУ НА ОСНОВІ АВТОМАТУ МУРА..... | 25 |
| 2.1 Моделювання операційного автомату | 25 |
| 2.2 Проєктування операційного автомату | 31 |
| 2.3 Висновки | 34 |
| 3 АПАРАТНА РЕАЛІЗАЦІЯ ОПЕРАЦІЙНОГО АВТОМАТУ НА ОСНОВІ АВТОМАТУ МУРА..... | 35 |
| 3.1 Схема Уілкса-Стрінжера | 35 |
| 3.2 Опис апаратної реалізації операційного автомату на основі автомату Мура..... | 43 |
| 3.3 Моделювання операційного автомату на основі автомату Мура в середовищі Quartus II | 53 |
| 3.4 Висновки | 56 |
| ВИСНОВКИ..... | 57 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ..... | 60 |
| ДОДАТОК А Копія креслення «Логічні схеми алгоритмів»..... | 64 |
| ДОДАТОК Б Копія креслення «Мікрокомандна схема та граф-схема переходів автомату Мура» | 65 |
| ДОДАТОК В Копія креслення «Схема електрична функціональна операційного автомату»..... | 66 |

| | | | | | | | | |
|----------|--------------|------------------|---|------|--|--------|-------|---------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | | | |
| Зм. | Арк. | Нодокм. | Підпис | Дата | Синтез та моделювання операційного автомату на основі автомату Мура. Пояснювальна записка | Літера | Аркуш | Аркущів |
| Виконав | Любовецький | Березький О.М. |  | | | у | 2 | 64 |
| Н.контр. | Лисенко С.М. | Говорущенко Т.О. |  | | ХНУ КІ-17-3 | | | |
| Затвер. | | |  | | | | | |

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ЕОМ – електронно-обчислювальна машина

ЦА – цифровий автомат

КА – керуючий автомат

ОА – операційний автомат

ПЛІС – програмована логічна інтегральна схема

АЛП – арифметико-логічний пристрій

ЦП – центральний процесор

ГСА – граф-схема алгоритму

ДДНФ – досконала диз'юнктивна нормальна форма

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 3 |

ВСТУП

Теорія формальних мов і методи побудови компілятора, що базуються на ній, а також кінцеві автомати застосовуються в комп'ютерних іграх, при реалізації мережевих протоколів, системах стискання інформації. Інакше кажучи, всюди, де потрібна висока надійність і складна логіка звернення, яку програмісту важко реалізувати.

З трьох головних конструкцій управління (проходження, цикл та розгалуження) останній є найскладнішим для програміста, оскільки з багатьма альтернативами перетворює лінійний алгоритм на дерево. У той же час складність послідовних програм також швидко зростає і іноді може перетворити дерево варіантів у модель, настільки важку для автоматичного аналізу, як традиційні шахи.

Поділ програми на процеси та об'єкти, виконуючи заміну багатоступеневого розгалуження обробкою повідомлень (подій), знімає проблему, натомість з'являється інша проблема – кількість компонентів, що взаємодіють, помітно збільшується. Логіка стає "розмитою", і в результаті з'являється слабо контрольована ситуація, оскільки через хаос "ручного" синтезу та неможливість побудови комплексного набору тестів відсутня впевненість у правильності побудованої системи.

Ключем до рішення є застосування формальних методів, створення зручної абстракції, яка може вибрати з алгоритму суть логіки роботи алгоритму і уможливити проведення всього необхідного аналізу.

Зручною абстракцією для цієї задачі є кінцеві автомати, серед яких варто виділити автомати Мілі та автомати Мура. Базованість на постулатах алгебри Буля та теорії графів, чіткість графічного подання та детермінованість поведінки є серйозними перевагами такої абстракції з точки зору поставленої проблеми.

Будь-який операційний пристрій є композицією операційного автомату та керуючого автомату.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. |
| | | | | | | 4 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Операційний автомат, який реалізує над словами інформації різноманітні операції, - це виконавчий блок пристрою. Керуючий автомат, що надає послідовності керуючих сигналів, необхідні для виконання обчислень, організовує роботу пристрою в цілому.

Операційний автомат – це пристрій, що виконує безпосередньо операції над інформацією під впливом керуючих сигналів і виробляє певні ознаки результатів цих операцій. Він включає в себе в якості складових вузлів реєстри, суматори, дешифратори, мультиплексори, АЛП та інші функціональні блоки, що реалізують елементарні дії, з яких складається процес виконання команд. Такими діями можуть бути запис в реєстр, інвертування вмісту реєстра, зсув коду ліворуч або праворуч, дешифрування двійкового коду, додавання двох чисел та інші. Кожна елементарна дія, що виконується в одному з вузлів операційного автомата протягом одного тактового періоду, називається мікрооперацією.

Автомати Мілі (Mealy) – це цифрові автомати, вихідні сигнали яких визначаються значеннями вхідних сигналів та попереднім станом автомату.

Автомати Мура (Moore) – це автомати, вихідні сигнали яких визначаються лише попереднім станом автомату і не залежать від вхідних сигналів.

Метою дипломної роботи є синтез та моделювання операційного автомату на основі автомату Мура.

Поставлена мета досягається розв'язанням такої основної задачі: побудувати операційний автомат, що виконує обчислення кількості парних елементів, що належать інтервалу $[-3; 13]$ у двовимірному масиві $A[n, m]$.

Для реалізації мікропроцесорного автомату як автомату Мура слід використовувати схему Уілкса-Стрінжера. Для реалізації та синтезу функціональної схеми автомату слід використовувати логічні елементи ТА, АБО, НІ, а також RS- та/або D-тригери із доповненням її (за необхідності) функціональними автоматами.

Об'єктом дослідження є програмно-технічний (апаратний) засіб – операційний автомат на основі автомату Мура.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|-----------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. 5 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Предметом дослідження є формалізований опис та схеми операційного автомату на основі автомату Мура.

Для досягнення поставленої мети використовувались наступні методи дослідження: принципи системного аналізу, методи синтезу, аналізу та моделювання процесів, теоретико-множинні підходи.

Практичне значення має змодельований, спроектований та реалізований операційний автомат на основі автомату Мура, який використовується для обчислення кількості парних елементів, що належать інтервалу $[-3; 13]$ у двовимірному масиві $A[n, m]$.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 6 |

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Операційний та керуючий автомати

Найчастіше пристрої складаються з двох частин: виконавчої та керуючої.

Виконавча частина пристрою зазвичай складна і незмінна.

Керуюча частина значно простіша, може змінюватися, і вся гнучкість пристрою полягає саме в гнучкості керуючої частини.

В обчислювальній машині одна зі складних і незмінних частин - арифметичний пристрій. Однак на ньому можна виконати як додавання, так і обчислення синуса кута, або розв'язання диференціального рівняння, або щось інше. Що саме буде отримано, залежить від керуючої частини.

Будь-який пристрій обробки цифрової інформації (процесор, канал введення-виведення, тощо) складається з двох основних блоків – операційного автомату (ОА) і керуючого автомату (КА).

Операційний автомат реалізує дії над словами інформації і є виконавчою частиною пристрою, роботою якого управляє керуючий автомат, який формує керуючі сигнали та їх послідовності. Операційний автомат зберігає слова інформації, виконує набір мікрооперацій і обчислює значення логічних умов, тобто це структура, організована для виконання дій над інформацією [1-3].

Керуючий автомат (КА) обирає з мікропрограми керуючі сигнали, враховуючи значення логічних умов. Інакше кажучи, КА задає порядок виконання операцій в операційному автоматі відповідно до використовуваних алгоритму та мікропрограми. Операція, яка повинна виконуватись, визначається операцією g , яка є зовнішньою по відношенню до КА. Сигнали g_1, \dots, g_h , які кодують операцію, та сигнали x_1, \dots, x_L , що формуються в ОА, впливають на порядок генерації керуючих сигналів Y .

Операційний автомат (ОА) зберігає слова інформації, виконує набір мікрооперацій і обчислює значення логічних умов, тобто ОА – це структура для виконання дій над інформацією.

Мікрооперації, які виконує операційний автомат, задаються множиною керуючих сигналів $Y = \{y_1, \dots, y_M\}$, кожен з яких визначає конкретну мікрооперацію. Значення логічних умов операційного автомату позначаються множиною інформаційних сигналів $X = \{x_1, \dots, x_L\}$, кожен з яких визначає конкретну логічну умову.

Принцип мікропрограмного керування, а також концепція операційних і керуючих автоматів лежать в основі проєктування операційних пристроїв різного призначення.

Мікропрограмування – це спосіб опису функцій операційних пристроїв, що не має відношення до використовуваних для їх реалізації технічних засобів. Таким чином, мікропрограмування дозволяє формалізувати структуру операційних пристроїв незалежно від способу керування роботою пристрою. Принципи і методи проєктування операційних і керуючих автоматів є основою теорії і практики проєктування більшої частини пристроїв ЕОМ [4-7].

ЕОМ опрацьовує інформацію, виконує над нею певні операції. Процесори, канали введення-виведення, вузли управління зовнішніми пристроями, які є операційними пристроями, використовуються для виконання операцій над словами інформації. Функцією операційного пристрою є виконання заданої множини операцій над вхідними словами з метою обчислення вихідних слів-результатів операцій.

Організація операційних пристроїв базується на принципі мікропрограмного керування [7-10]:

1) будь-яка операція – це складна дія, яка складається з послідовності елементарних дій (мікрооперацій) над словами інформації (передача інформації з одного регістра в інший, формування оберненого коду, зсув, тощо);

2) логічні умови використовуються для керування порядком проходження мікрооперацій;

3) процес виконання операцій описується у вигляді алгоритму в термінах мікрооперацій і логічних умов (мікропрограми), який формує порядок виконання мікрооперацій та перевірки логічних умов з метою одержання необхідних результатів;

4) мікропрограма – це функції пристрою, представлені за певною формою певною мовою, на основі яких синтезується структура пристрою і визначається його порядок функціонування в часі.

Складність і відповідальність задач, що розв’язуються сучасними цифровими пристроями, системами та ЕОМ, потребують високої надійності, відмовостійкості, гарантоздатності та продуктивності від пристроїв, систем та ЕОМ.

Операційний автомат визначають наступні множини [10-12]:

1) множина вхідних слів $D=\{d_1, \dots, d_H\}$ – операнди автомату;

2) множина вихідних слів $R=\{r_1, \dots, r_Q\}$ – результати операцій;

3) множина внутрішніх слів $S=\{s_1, \dots, s_N\}$ – інформація під час виконання операцій. Множини вхідних і вихідних слів є підмножинами множини внутрішніх слів: $D \subseteq S, R \subseteq S$;

4) множина мікрооперацій $Y=\{y_m\}$ – перетворення $S \rightarrow \varphi_m(s)$ слів інформації, де φ_m – обчислювана функція;

5) множина логічних умов $X=\{x_i\}$, де $x_i=\psi_i(s_i)$ і ψ_i – логічна функція.

Таким чином, операційний автомат та його функція визначені, якщо задані (визначені) множини D, R, S, Y, X .

Особливістю операційного автомату є той факт, що час не є аргументом його функції ОА [1, 12-15].

Вихідним для розробки структури операційного автомату є:

1) опис вхідних та вихідних слів ОА;

2) список множини операцій, які повинні виконуватися над словами.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. |
| | | | | | | 9 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Процес розробки операційного автомату, таким чином, слід починати з того, що необхідно визначити формати вхідних і вихідних слів, після чого перейти до розробки алгоритмів в термінах слів і дій над словами (додавання, копіювання, інверсія, зсуви, тощо).

Далі необхідно розробити структуру ОА, тобто визначити набір елементів, що входять в неї, і встановити зв'язки між цими елементами. Операційний автомат будується на базі операційних і логічних елементів. Структура операційного автомату синтезується згідно із алгоритмами, розробленими на попередньому етапі, з метою забезпечення реалізації всіх дій, передбачених в операційних вершинах.

Дії в структурі операційного автомату виконуються під управлінням мікрооперацій, тому при розробці ОА слід визначити повний список мікрооперацій, наявність яких забезпечить виконання в розробленій структурі всіх передбачених в алгоритмах перетворень слів.

Послідовність мікрооперацій в керуючому автоматі формується, враховуючи логічні умови, які обчислюються операційним автоматом. Відтак при розробленні ОА слід сформулювати список логічних умов, який визначається вмістом умовних вершин, і передбачити в структурі ОА (якщо це необхідно) спеціальні елементи для формування цих логічних умов.

Отже, процес розробки операційного автомату складається з наступних етапів:

- 1) підбір форматів слів – вхідних та вихідних даних;
- 2) розроблення виконуваних операцій;
- 3) розроблення структури ОА – вибір елементів і організація зв'язків;
- 4) визначення множини мікрооперацій, виконуваних в ОА.

Взаємодія операційного та керуючого автоматів представлена на рис. 1.1.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. 10 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

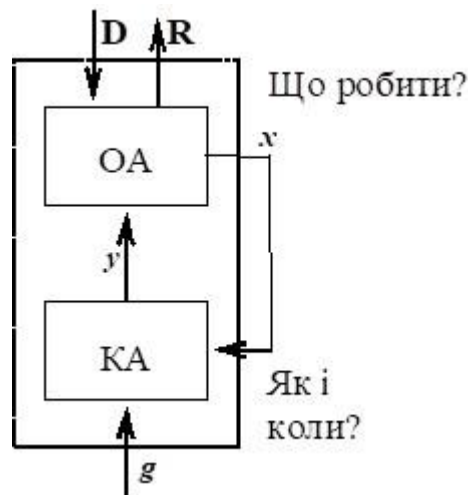


Рисунок 1.1 – Взаємодія операційного та керуючого автоматів

Функція операційного автомату визначає список дій (мікрооперацій і логічних умов), які автомат може і повинен виконувати, але порядок цих дій (у часі) жодним чином не визначає. Тоді функція операційного автомату характеризує засоби, які будуть використовуватись при обчисленнях, але не, власне, обчислювальний процес.

Порядок дій (в часі) визначають функції керуючого автомата, які задаються у вигляді операторних схем алгоритму (мікропрограми), де мікрооперації позначені символами y_1, \dots, y_m , а логічні умови позначені булевими змінними x_1, \dots, x_L .

Операторна схема алгоритму, як правило, має вигляд граф-схеми алгоритму (ГСА). ГСА демонструє перебіг обчислювального процесу в часі, встановлюючи порядок перевірки логічних умов і порядок виконання мікрооперацій. При побудові ГСА мікрооперації зображаються умовними графічними зображеннями, так званими вершинами. ГСА, як правило, містять вершини наступних типів [1-3, 14-18]:

- 1) вершина «початок» - початок мікропрограми;
- 2) вершина «кінець» - кінець мікропрограми;
- 3) операторна вершина - містить одну мікрокоманду як сукупність мікрооперацій з одночасним виконанням;

4) умовна вершина - містить логічний вираз, в залежності від значення якого (true або false) здійснюється вибір гілки подальшого виконання мікропрограми;

5) вершина очікування – дещо особлива умовна вершина, вихід з якої можливий тільки при виконанні умови (настання значення true).

Отже, операційний автомат – це сукупність комбінаційних схем, виконуючих мікрооперації, а також регістрів/оперативних запам'ятовуючих пристроїв для зберігання даних.

Алгоритм роботи операційного автомату [18-20]:

1) залежно від одержаних сигналів від керуючого автомату, операційний автомат виконує мікрооперації – різні дії над регістрами, операції введення і виведення;

2) після виконання мікрооперацій операційний автомат формує мікроумови, за якими керуючий автомат переходить у наступний стан.

Операційний автомат складається з:

1) вхідної частини, в якій розташовані тригери, логічні елементи, які приймають та опрацьовують вхідні сигнали, дешифратор та 2 шини (одна шина використовується для передачі сигналів з дешифратора, друга шина використовується для передачі сигналів з компаратора);

2) перехідної частини, яка проектується на основі системи рівнянь переходів та в якій виконується перетворення сигналу протягом одного такту часу, сигнал змінюється, і результат надходить на шину, звідки передається до програмованої логічної матриці;

3) вихідної частини, яка проектується на основі системи рівнянь виходів і в якій виконується процес, подібний до процесу перехідної частини, проте сигнали передаються на вихідну шину, з якої надходять до вихідної матриці.

Проектований операційний автомат має забезпечувати певну продуктивність. Задана продуктивність може забезпечуватись лише у випадку, якщо синтезована структура не вносить обмежень на сумісність мікрооперацій, тобто можливим є одночасне виконання всіх мікрооперацій, які є функціонально сумісними.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. 12 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Комбінаційні схеми, які обчислюють інформаційні сигнали та виконують мікрооперації, необхідно узагальнювати з метою мінімізації апаратних витрат.

До класу I-автоматів належать операційні автомати, які мають структуру, що уможливує одночасне виконання функціонально сумісних та еквівалентних мікрооперацій, використовуючи при цьому мінімально можливу кількість комбінаційних схем.

Структура I-автомату синтезується в такий спосіб [20-23]:

- 1) множина мікрооперацій Y розбивається на підмножини, відповідні внутрішнім регістрам;
- 2) на підмножинах множини Y виділяються класи еквівалентних мікрооперацій;
- 3) для кожного класу еквівалентних мікрооперацій, що містить не менше двох еквівалентних мікрооперацій, будується узагальнений оператор; якщо клас містить тільки одну мікрооперацію, то узагальненим оператором даного класу є сама мікрооперація;
- 4) структура схеми I-автомату синтезується з огляду на опис слів, логічних умов та узагальнених операторів.

Комбінаційні схеми, які використовуються як обслуговуючі для регістрів і якими виконуються еквівалентні операції, можуть мати місце в структурі I-автомату.

Апаратні витрати можуть бути мінімізовані для певного алгоритму, якщо узагальнити комбінаційні схеми відносно усіх регістрів.

Такі операційні автомати, які синтезуються з врахуванням принципу узагальнення, називають M-автоматами.

Швидкодія M-автомату незначно відрізняється від швидкодії I-автомату. Тривалість такту трохи збільшується за рахунок введення в схему мультиплексорів. Витрати обладнання M-автомату мінімальні, оскільки комбінаційна схема використовується для виконання всіх еквівалентних мікрооперацій з множини Y .

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. 13 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

М-автомат породжує специфічний набір керуючих сигналів. Кожен керуючий сигнал ініціює визначену дію, яку розглядають як мікрооперацію.

Операційний автомат – це набір функціональних елементів, таких як арифметико-логічний пристрій (АЛП) і логічні елементи, які виконують обробку даних. Разом з блоком управління він складає центральний процесор (ЦП).

Операційні пристрої, такі як мікропроцесори, складаються з операційного автомата і керуючого автомата, більшу частину яких займає керуючий автомат, який регулює передачу даних між операційним автоматом і пам'яттю.

У комп'ютерних процесорах операційний автомат часто складається з наступних функціональних блоків (або у вигляді будь-якої їх варіації):

- 1) реєстр команд – зберігає поточну інструкцію, яка буде виконана;
- 2) програмний лічильник – зберігає адресу наступної інструкції для видобутих даних;
- 3) адреса реєстра пам'яті – являє собою реєстр, який або зберігає адресу пам'яті, з якого будуть видобуватись дані на ЦП, або адреса, за якою дані будуть відправлені і збережені;
- 4) реєстр даних пам'яті – являє собою реєстр блоку управління комп'ютера, який містить дані, які будуть збережені в пам'яті ЕОМ (наприклад, ОЗП), або дані після вибірки з пам'яті ЕОМ;
- 5) два реєстри, що знаходяться в процесорі, які полегшують взаємодію процесора з пам'яттю.

Часто операційний пристрій може виконувати кілька різних операцій. Наприклад, АЛП виконуються всі 4 арифметичні операції і декілька логічних операцій. В такому разі команда, яка вказує на тип операції, що має виконуватись, подається на вхід керуючого автомату.

Враховуючи той факт, що різні операції над різними словами інформації виконуються за різний час, керуючий автомат повинен сформувати сигнал, що вказує на закінчення операції і сигналізує про готовність результуючих даних.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. 14 |
| Зм. | Арк. | № докum. | Підпис | Дата | | |

Тоді будь-який операційний пристрій (процесор, канал вводу-виводу, контролер зовнішнього пристрою, тощо) може бути представлений у вигляді сполучення операційного і керуючого автоматів.

Операційний автомат реалізує операції над інформацією, є виконавчою частиною пристрою. Роботу ж пристрою в цілому організовує керуючий автомат, який надає певні послідовності керуючих сигналів.

Такий підхід підвищує ефективність процедур синтезу операційного і керуючого автоматів, уможливорює формалізацію цих процедур і в деяких випадках забезпечує автоматизацію процесу синтезу цифрових пристроїв.

Отже, операційний автомат – це пристрій, що виконує безпосередньо операції над інформацією під впливом керуючих сигналів і виробляє певні ознаки результатів цих операцій. Він включає в себе в якості складових вузлів реєстри, суматори, дешифратори, мультиплексори, АЛП та інші функціональні блоки, що реалізують елементарні дії, з яких складається процес виконання команд. Такими діями можуть бути запис в реєстр, інвертування вмісту реєстра, зсув коду ліворуч або праворуч, дешифрування двійкового коду, додавання двох чисел та інші. Кожна елементарна дія, що виконується в одному з вузлів операційного автомата протягом одного тактового періоду, називається мікрооперацією.

1.2 Автомати Мілі та Мура

Цифровий автомат (ЦА) – це пристрій, який має певний набір внутрішніх станів, в які він потрапляє під впливом команд програми, яка в нього закладена. Перехід автомата зі стану в стан здійснюється у певний момент часу [20-22].

Цифровий автомат можна трактувати як пристрій, який приймає, зберігає і перетворює дискретну інформацію за певним алгоритмом. З певної точки зору до автоматів можна віднести як реальні пристрої (ЕОМ), так і абстрактні системи (математичні моделі).

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. 15 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Автоматом називається дискретний перетворювач інформації, здатний приймати різні стани, переходити під впливом вхідних сигналів з одного стану в інший і видавати вихідні сигнали.

Математична модель цифрового автомату представляє собою кортеж з 6 компонентів: $S=(A,Z,W, \delta, \lambda, a_1)$, де [22-24]:

- 1) $A = \{a_1, a_2, \dots, a_m\}$ - множина станів (внутрішній алфавіт);
2. $Z = \{z_1, z_2, \dots, z_f\}$ - множина вхідних сигналів (вхідний алфавіт);
3. $W = \{w_1, w_2, \dots, w_g\}$ - множина вихідних сигналів (вихідний алфавіт);
4. $\delta: A \cdot Z \rightarrow A$ - функція переходів, що реалізує відображення $D_\delta \subseteq A \cdot Z$ в A , тобто деяким парам «стан-вхідний сигнал» (a_m, z_f) ставить у відповідність стани автомату $a_s = \delta(a_m, z_f)$, $a_s \in A$;
5. $\lambda: A \cdot Z \rightarrow W$ - функція виходів, що реалізує відображення $D_\lambda \subseteq A \cdot Z$ на W , тобто деяким парам «стан-вхідний сигнал» (a_m, z_f) ставить у відповідність вихідні сигнали автомату $W_g = \lambda(a_m, z_f)$, $W_g \in W$;
6. $a_1 \in A$ - початковий стан автомату.

Функція переходів встановлює залежність внутрішнього стану автомата в наступний момент часу від стану входу і внутрішнього стану в даний момент часу.

Функція виходів встановлює залежність стану виходу автомата від стану входу і внутрішнього стану автомата.

Різний характер цих залежностей для різних автоматів дозволяє виділити окремі типи автоматів в класі синхронних кінцевих детермінованих автоматів.

Існують два класи цифрових автоматів.

Автомати Мілі (Mealy) – це цифрові автомати, в яких сигнали на виходах визначаються сигналами на входах та попереднім станом автомату.

Автомати Мура (Moore) – це цифрові автомати, в яких сигнали на виходах залежать лише від попереднього стану автомату та не залежать від значень сигналів на входах [24-28].

Закон функціонування автомату Мілі має вигляд:

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. 16 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

$$a(t+1) = \delta((a(t), z(t)));$$

$$w(t) = \lambda((a(t), z(t)), t = 0,1,2,\dots$$

Закон функціонування автомату Мура має вигляд:

$$a(t+1) = \delta((a(t), z(t)));$$

$$w(t) = \lambda(a(t)), t = 0,1,2,\dots$$

Порівняння законів функціонування демонструє, що сигнали на виходах автомату Мура залежать лише від станів автомату і не залежать від сигналів на входах.

Усі компоненти кортежу $S=(A, Z, W, \delta, \lambda, a_1)$ повинні бути описані для задання автомату. Множини A, Z, W задаються перерахунком своїх елементів. Серед різних способів завдань функцій δ і λ найбільш популярними є табличний і графічний способи.

При табличному способі задання автомат Мілі описується двома таблицями. Таблиця переходів задає функцію δ , таблиця виходів – функцію λ .

Автомат Мура задається одною відміченою таблицею переходів, в якій кожному стовпцю призначено не лише стан a_m , але й сигнал виходу w_g , який відповідає цьому стану, де $w_g = \lambda(a_m)$.

Для частково визначених автоматів Мілі та Мура ставиться прочерк на місці невизначених станів і невизначених вихідних сигналів. У таких автоматах вихідний сигнал на переході завжди невизначений, якщо невизначеним є сам стан переходу. Вихідний сигнал також може бути невизначеним і для деяких визначених переходів.

Поняття стану у визначенні автомату необхідне через те, що часто виникає необхідність описувати поведінку систем, виходи яких залежать не лише від сигналів входів у певний момент часу, але й від певної передісторії, тобто від сигналів, які надходили на входи системи раніше.

Стан автомату якраз і є певною пам'яттю, передісторією, яка дозволяє усунути час в явному вигляді та виразити сигнали виходів як функцію станів і входів в даний момент часу.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. 17 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Якщо автомат задається графічним способом, то він представляється у вигляді орієнтованого графа. Вершини цього орграфу відповідають станам автомату, а дуги – переходам між станами. Дуга, спрямована з однієї вершини графу в іншу, задає перехід в автоматі зі стану в стан. На початку цієї дуги записується вхідний сигнал, що викликає даний перехід.

Для графу автомата Мілі вихідний сигнал, який формується при певному переході, записується наприкінці дуги (біля стрілки), а для автомата Мура – поруч з вершиною, відзначеною станом, у якому він формується.

Якщо перехід в автоматі зі стану в стан генерується під дією декількох вхідних сигналів, то відповідній дузі графа приписуються всі ці вхідні і відповідні вихідні сигнали [28-30].

На рис.1.2 наведено граф автомату Мілі на 3 стани, який має 2 вхідних сигнали і 2 вихідних сигнали.

На рис.1.3 наведено граф автомату Мура на 5 станів, що має 2 вхідних сигнали і 2 вихідних сигнали.

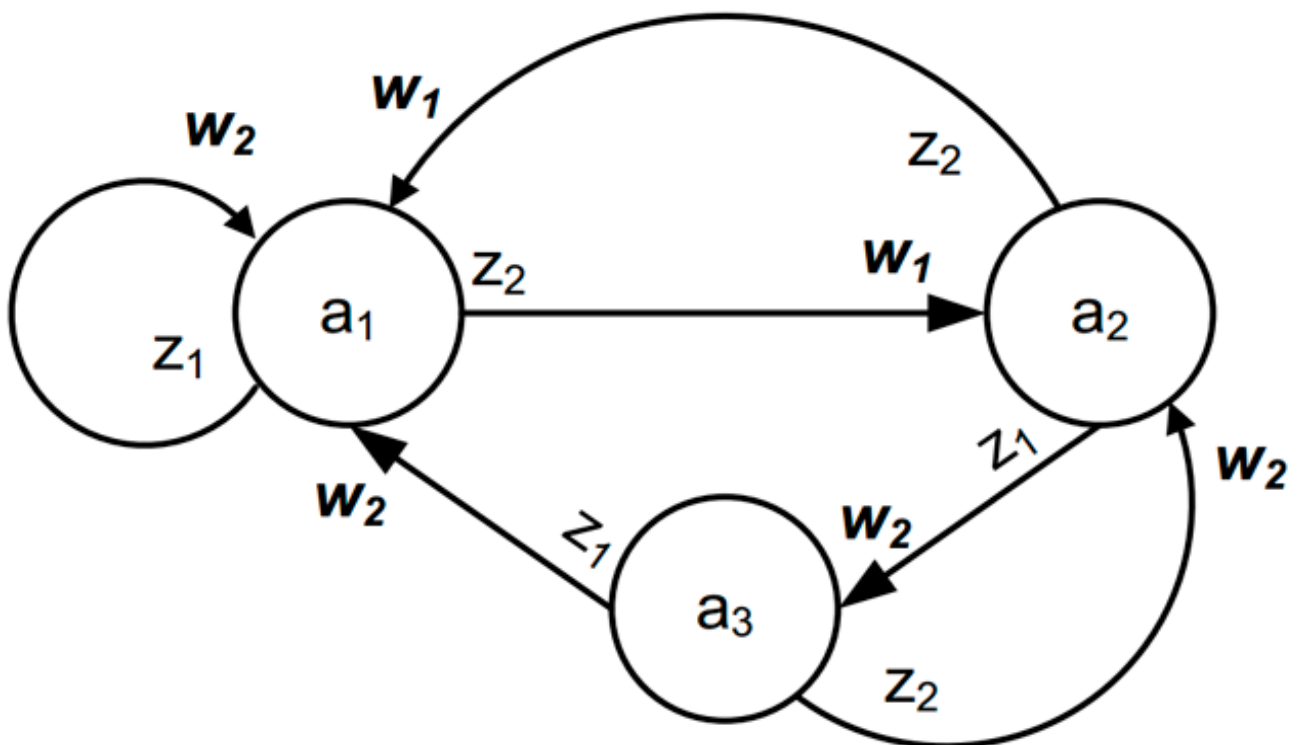


Рисунок1.2 – Приклад графу автомата Мілі

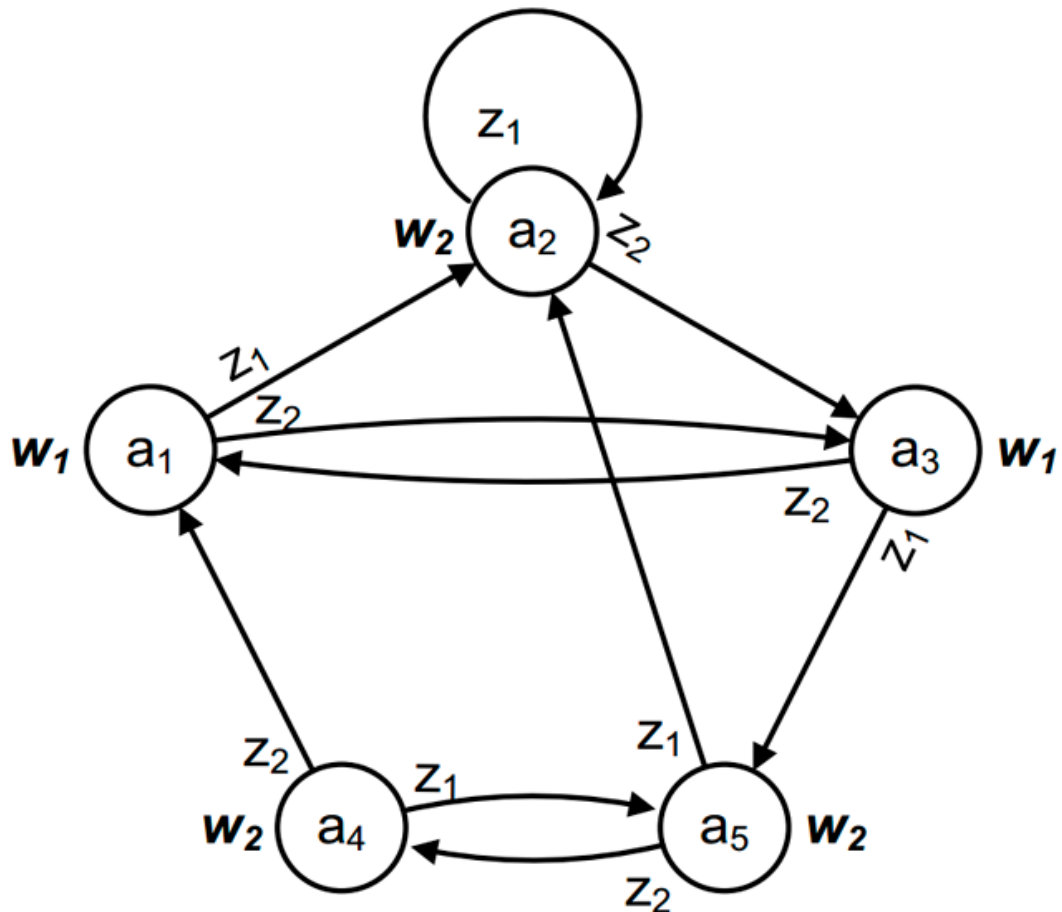


Рисунок 1.3 – Приклад графу автомата Мура

Функціонування цифрового автомату є однозначно визначеним, якщо встановлені зв'язки в часі між внутрішнім, вхідним та вихідним алфавітами.

Зміни станів цифрового автомату викликають сигналами входів, що виникають поза автоматом та по вхідних каналах передаються у автомат. Результатом роботи цифрового автомату є видача сигналів виходів, що передаються з автомату по вихідних каналах назовні.

Логічні схеми цифрового автомату представляються у вигляді комбінаційних схем логічних елементів, які виконують певні операції, а також запам'ятовуючих елементів.

Автомат Мура переходить в автомат Мілі, якщо всім переходам в стан поставити вихідні впливи цього стану. Після таких перетворень отримаємо еквівалентний автомат Мілі [31-34].

Однак, щоб перетворити автомат Мілі в автомат Мура такий алгоритм не підходить, тому що в один стан можуть вести різні переходи. Але можна просто додати нові стани, встановлюючи необхідні відповідності.

Автомати Мура і Мілі широко застосовуються при проектуванні цифрових пристроїв з використанням програмованих логічних інтегральних схем (ПЛІС).

Основна перевага використання автомату Мілі полягає в можливості реакції автомату протягом поточного такту, що обумовлено залежністю поточної вихідної комбінації від поточної вхідної комбінації.

Наявність мінімальної затримки, відсутність стабільності перехідного процесу на виході автомата, відсутність наскрізного поширення сигналу через комбінаційну схему від входу до виходу автомата, простота опису на мовах опису апаратури HDL робить автомат Мура практично незамінним.

Також автомати Мура і взаємодіючі автомати Мілі використовуються в генетичному програмуванні.

Методи взаємотранспозиції автоматів Мілі і Мура демонструють, що при переході від автомата Мілі до автомата Мура кількість станів принципово не змінюється. У той час як при зворотному переході в автомат Мура кількість станів, як правило, збільшується [35-41].

1.3 Схеми електричні функціональна та принципова

Схема електрична функціональна відображає процеси, що проходять в окремих функціональних колах пристрою. Такі схеми призначені для демонстрування принципів роботи пристрою, а також використовуються при його налагодженні, моніторингу, ремонті.

Схема електрична функціональна докладніше розкриває функції окремих елементів та вузлів, ніж схема електрична структурна.

В той же час, при розробленні схеми електричної функціональної слід уникати надмірної деталізації, хоча на схемі електричній функціональній окремі

блоки розроблюваного пристрою можуть наводитись більш детально, ніж на схемі електричній структурній, з метою демонстрації особливостей формування пристрою та перетворення сигналів під час його роботи.

Схема синтезується на базі існуючих елементів – інтегральних мікросхем різних ступенів інтеграції та дискретних елементів.

Функціональні частини позначають прямокутниками, а також за допомогою умовних графічних позначень, визначених ДСТУ та/або ГОСТ.

При виконанні схем дозволено повертати умовні графічні позначення на 90°, зазначати розрядність функціональних частин, поєднувати графічні позначення за більшою стороною, якщо виходи однієї з функціональних частин цілком відповідають входам іншої функціональної частини.

Лінії зв'язку на функціональних схемах можуть бути інформаційними та керуючими.

Інформаційні лінії підходять до більшої сторони умовного графічного позначення, а виходять від сторони навпроти. Керуючі лінії підходять до меншої сторони умовного графічного позначення.

Горизонтальні лінії з обмежувачами над умовними графічними позначеннями застосовують для однозначного позначення входів і виходів частин пристрою. На них зазначаються розрядність функціональної частини та її складових.

Якщо багато керуючих сигналів входять до умовного графічного позначення, можна збільшити сторони умовного графічного позначення або обмежувачі ліній, до яких вони підходять.

Для функціональних частин всередині умовного графічного позначення вказують їх найменування або умовні позначення, можна також вказати символ функції.

Найменування складових можуть наводитись скорочено, як на електричній структурній схемі, але скорочення обов'язково мають бути розшифровані на тлі схеми.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. 21 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Опис основних процесів автомату, взаємодія його складових частин, перелік та послідовність сигналів, які беруть участь у зазначених процесах, надається у розділі пояснювальної записки “Опис схеми електричної функціональної”.

Необхідні для демонстрації роботи пристрою часові діаграми наводяться також в цьому розділі для більшої наочності.

На схемотехнічному етапі проектування виконується розроблення схеми. Це більш високий рівень синтезу схем, ніж синтез структурної схеми. Функціональна схема – це сукупність формальних моделей функціональних складових пристрою. Принципова схема – це сукупність електричних моделей функціональних складових пристрою.

Формальна модель дільника частоти, наприклад, має вигляд прямокутника з написом в ньому “дільник частоти f/n ”. Електрична модель цього дільника має вигляд електричного кола, здатного виконувати функцію дільника. Отже, принципова схема – менш абстрактна модель пристрою.

Принципова схема – це основа для розробки інших конструкторських документів (схем з’єднань і креслень).

Схема електрична принципів – це найбільш повна електрична схема пристрою, на якій зображено електричні елементи і прилади, необхідні для здійснення і контролю певних електричних процесів у виробі, всі зв’язки між ними та елементи підключення.

Всі елементи представлені в довідниках із зазначенням їх функціонального призначення та параметрів. Елементи необхідно підбирати згідно із характеристиками проєктованого пристрою, тобто згідно із розрядністю шин адреси і даних, швидкодії, габаритів. Підбір елементів слід виконувати таким чином, щоб вони найбільш підходили до всім вимог пристрою і по максимуму використовувались.

Обрані елементи об’єднуються у функціональні вузли (наприклад, блок оперативного запам’ятовуючого пристрою, шинний буфер, тощо), після чого ці вузли з’єднуються між собою, утворюючи цілий пристрій (прилад).

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. |
| | | | | | | 22 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Повний склад елементів та зв'язків між ними визначається саме принциповою схемою, яка деталізує принципи роботи пристрою. У перелік елементів, який має бути розташований на аркуші зі схемою електричною принциповою або виконується у вигляді самостійного документу, вносять всі відомості про елементи, що входять до складу пристрою і зображаються на схемі. Перелік елементів виконують у вигляді таблиці.

Елементи в переліку та їх умовні графічні позначення на схемі пов'язуються завдяки позиційному позначенню елементів.

Перелік елементів, який розміщують на аркуші зі схемою, розташовують, в основному, над основним написом.

Електричні елементи на схемі позначають умовними графічними позначеннями. Розміри умовних графічних позначень встановлені стандартом ЄСКД. Якщо елементи використовуються у пристрої неповністю, то зображають лише їх використовувані частини (повністю можна не зображати). До складу схеми, крім зображення, обов'язково входять написи, перелік елементів, позиційні позначення елементів.

Умовні літерно-цифрові позиційні позначення (порядкові номери) призначаються всім зображеним на схемі елементам. Порядкові номери елементам в схемі слід призначати, починаючи з 1 в межах кожного виду елементів, яким на схемі надано однакове літерне позначення.

Порядкові номери надають в межах кожного виду елементів згідно із послідовністю розташування елементів на схемі згори донизу в напрямку з лівого боку, рухаючись праворуч.

Послідовність присвоєння порядкових номерів може бути змінена залежно від розміщення елементів пристрою, враховуючи напрямок проходження сигналів чи функціональну послідовність процесу, а також при внесенні в схему змін.

Позиційні позначення підписують на схемі поряд з умовними графічними позначеннями елементів з правого боку або над ними.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. |
| | | | | | | 23 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

1.4 Висновки. Постановка задачі

Будь-який операційний пристрій може бути представлений у вигляді сполучення операційного і керуючого автоматів. Операційний автомат реалізує операції над інформацією, є виконавчим компонентом пристрою. Його роботу організовує керуючий автомат, що надає певні послідовності керуючих сигналів.

Операційний автомат – це пристрій, що виконує безпосередньо операції над інформацією під впливом керуючих сигналів і виробляє певні ознаки результатів цих операцій. Він включає в себе в якості складових вузлів регістри, суматори, дешифратори, мультиплексори, АЛП та інші функціональні блоки, що реалізують елементарні дії, з яких складається процес виконання команд. Такими діями можуть бути запис в регістр, інвертування вмісту регістра, зсув коду ліворуч або праворуч, дешифрування двійкового коду, додавання двох чисел та інші. Кожна елементарна дія, що виконується в одному з вузлів операційного автомата протягом одного тактового періоду, називається мікрооперацією.

Автомати Мілі (Mealy) – це цифрові автомати, в яких сигнали виходів визначаються сигналами входів та станами автомату в попередній момент часу.

Автомати Мура (Moore) – це автомати, для яких сигнали виходів залежать лише від попереднього стану автомату та не залежать від значень сигналів входів.

В кваліфікаційній роботі слід розв'язати наступну задачу: побудувати операційний автомат, що виконує обчислення кількості парних елементів, що належать інтервалу $[-3; 13]$ у двовимірному масиві $A[n, m]$.

Для реалізації мікропроцесорного автомату як автомату Мура слід використовувати схему Уілкса-Стрінжера.

Для реалізації та синтезу функціональної схеми автомату слід використовувати логічні елементи ТА, АБО, НІ, а також RS- та/або D-тригери із доповненням її (за необхідності) функціональними автоматами.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. 24 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ОПЕРАЦІЙНОГО АВТОМАТУ НА ОСНОВІ АВТОМАТУ МУРА

2.1 Моделювання операційного автомату

Побудова мікропрограмного автомату за схемою Уілкса-Стрінжера зводиться до таких дій:

- 1) кодування операційних та умовних верхівок;
- 2) побудова основної таблиці і граф-схеми переходів;
- 3) побудова системи рівнянь для функцій переходів і системи рівнянь для функцій виходів (для автоматів Мілі);
- 4) кодування внутрішніх станів автомату;
- 5) побудова функціональної схеми автомату.

До складу змістовної схеми алгоритму (рис. 2.1) входять операційні та умовні верхівки. Наш алгоритм, згідно поставленого завдання виконує знаходження обчислення кількості парних елементів, що належать інтервалу $[-3; 13]$ у двовимірному масиві $A[n, m]$, використовуючи при цьому чотири умовні верхівки і вісім операційних верхівок.

Кожна верхівка (як операційна, так і умовна) кодується. При цьому повторювані мікрооперації та повторювані умовні верхівки однаково кодуються. В даному алгоритмі жодна з мікрооперацій не повторюється, тому однакові верхівки, які ми можемо кодувати одним кодом, відсутні. Таблиця кодування верхівок представлена у таблиці 2.1.

На основі змістовної схеми алгоритму (рис. 2.1) і на основі таблиці кодування операційних та умовних верхівок (таблиця 2.1) шляхом заміни відповідних блоків будується закодована мікроопераційна схема алгоритму (рис. 2.2).

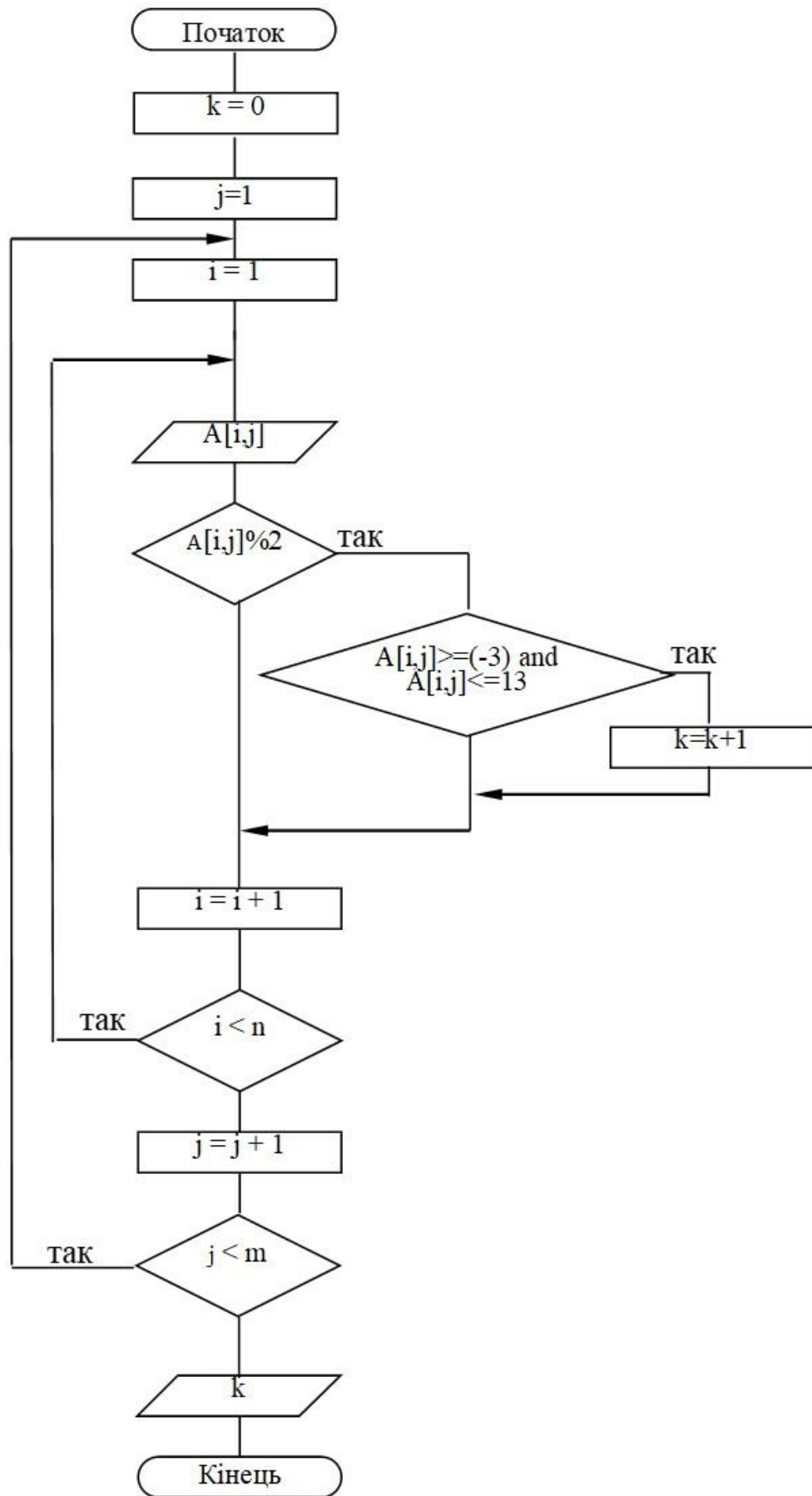


Рисунок 2.1 – Змістовна схема алгоритму

Для складання таблиці кодування мікрокоманд кожна операція кодується мікрокомандою. Мікрооперації, які послідовно виконуються одна за одною протягом одного такту часу, об'єднуються в одну мікрокоманду. В даному завданні один раз дві мікрооперації ($mY1$, $mY2$) послідовно виконуються. Тому об'єднуємо їх в одну мікрокоманду. Таблиця 2.2 містить таблицю кодування мікрокоманд.

Будуємо закодовану мікрокомандну схему алгоритма (рис. 2.3). Робимо розмітку внутрішніх станів автомату Мура таким чином: мітки ставимо біля кожної мікрокоманди (відповідно порядковому номеру); початок і кінець мікрокомандної схеми алгоритму позначається міткою $a0$. Проте слід врахувати, що перехід з одного стану в інший обов'язково повинен проходити через умовну або операційну верхівку автомату.

Таблиця 2.1 – Таблиця кодування верхівок

| № | Код | Зміст | Примітки |
|----|-------|---|------------------------------------|
| 1 | $mY1$ | $k = 0$ | |
| 2 | $mY2$ | $j = 1$ | |
| 3 | $mY3$ | $i = 1$ | |
| 4 | $mY4$ | $A [i,j]$ | Введення елемента масиву $A [i,j]$ |
| 5 | $mY5$ | $k = k+1$ | |
| 6 | $mY6$ | $i = i + 1$ | |
| 7 | $mY7$ | $j = j + 1$ | |
| 8 | $mY8$ | k | Виведення k |
| 9 | X1 | $A [i,j] \% 2$ | так - 1, ні - 0 |
| 10 | X2 | $A[i,j] \geq (-3)$ and $A [i,j] \leq 13$ | так - 1, ні - 0 |
| 11 | X3 | $i < n$ | так - 1, ні - 0 |
| 12 | X4 | $j < m$ | так - 1, ні - 0 |

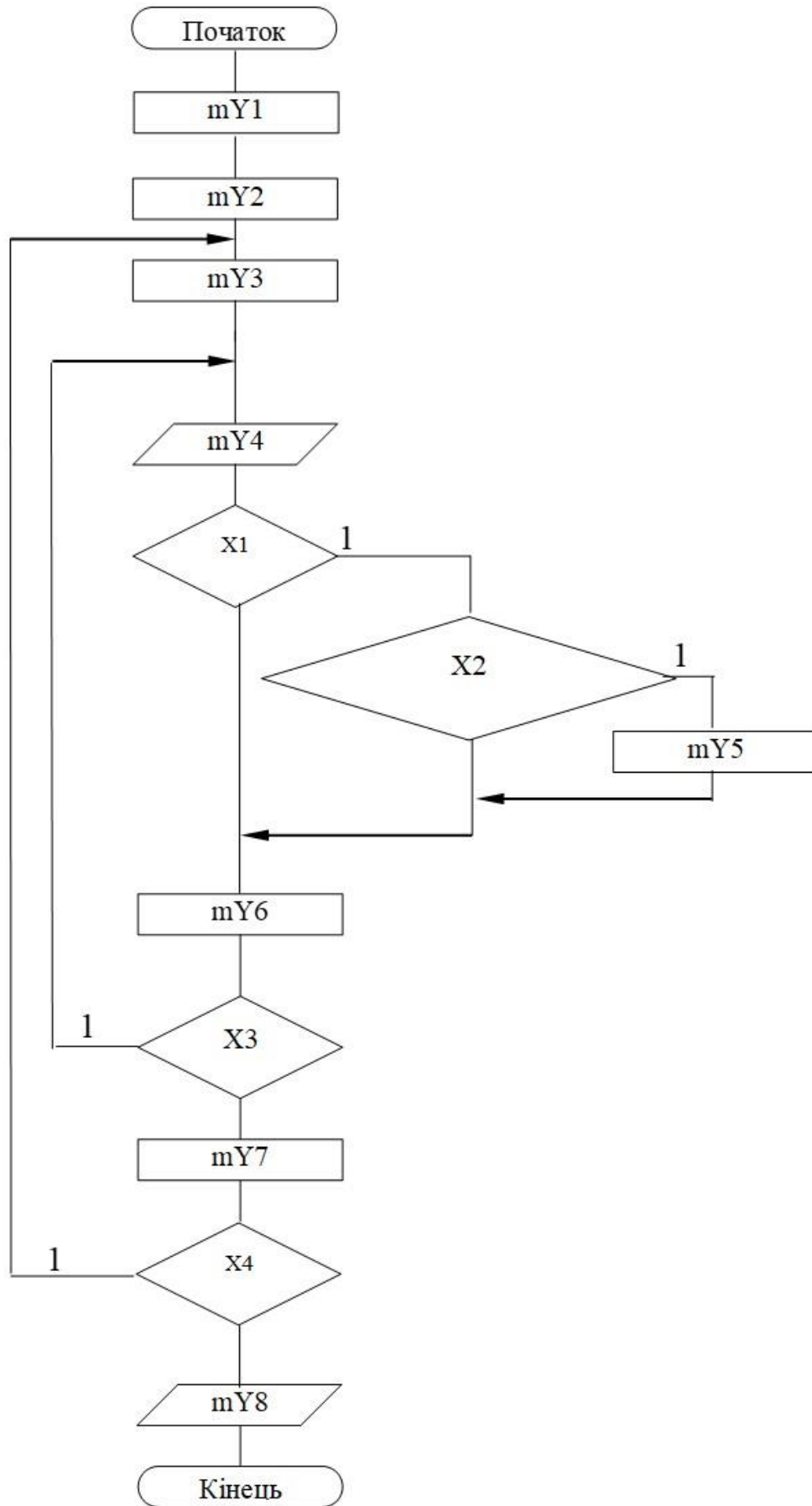


Рисунок 2.2 – Закодована мікроопераційна схема алгоритму

Таблиця 2.2 – Таблиця кодування верхівок

| № | Мікрокоманда | Мікрооперація |
|---|--------------|---------------|
| 1 | Y1 | mY1, mY2 |
| 2 | Y2 | mY3 |
| 3 | Y3 | mY4 |
| 4 | Y4 | mY5 |
| 5 | Y5 | mY6 |
| 6 | Y6 | mY7 |
| 7 | Y7 | mY8 |

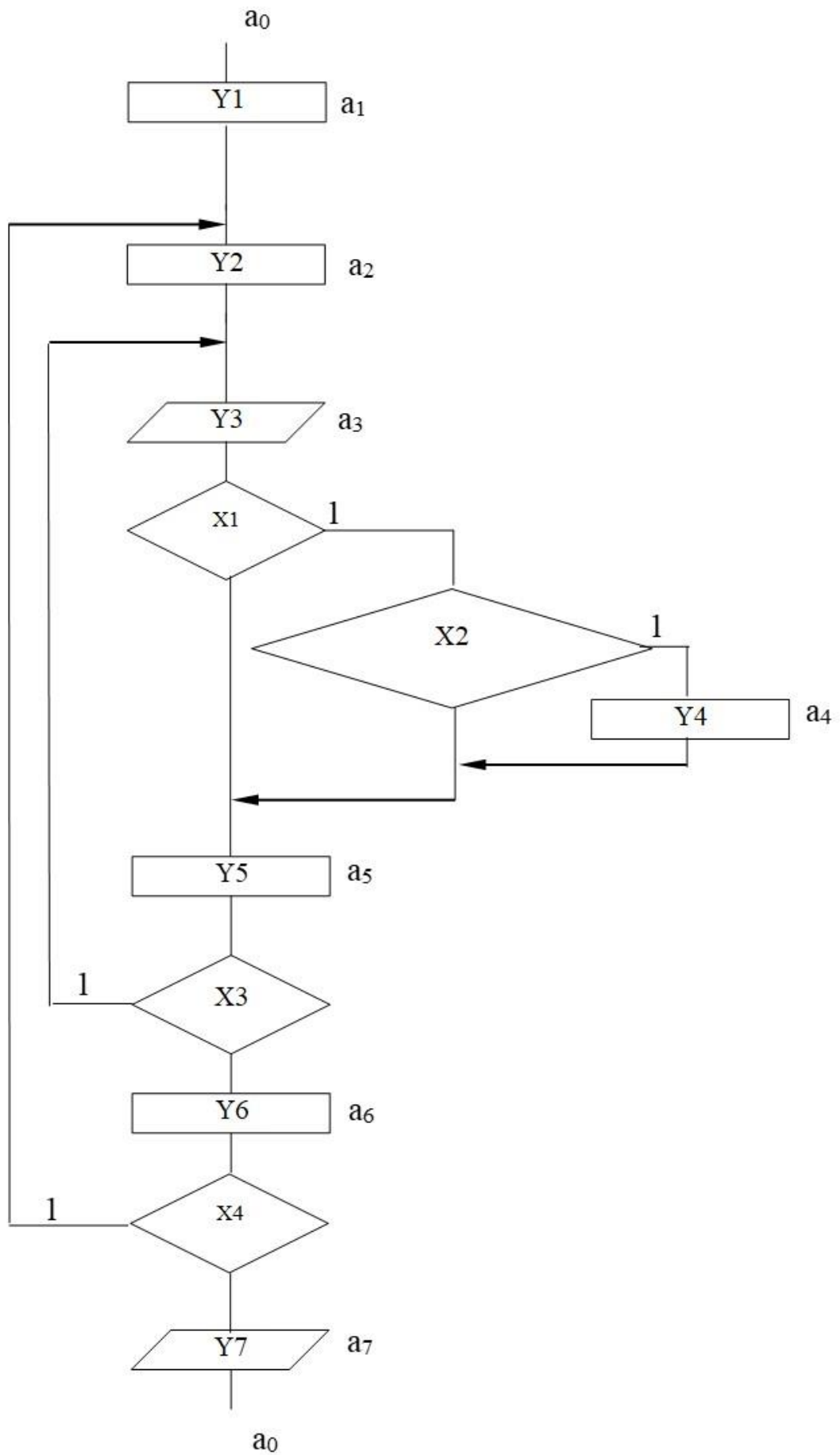


Рисунок 2.3 – Закодована мікрокомандна схема алгоритму

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Арк. | № докум. | Підпис | Дата |

2.2 Проектування операційного автомату

Основна таблиця автомату Мура (таблиця 2.3) складається на основі закодованої мікрокомандної схеми алгоритму (рис. 2.3). У першому стовпчику таблиці записуються стани, в яких може перебувати автомат. Способи переходу автомату зі стану в стан записуються в першому рядку таблиці. Тобто перший рядок таблиці визначає безумовний перехід СІ (перехід, під час якого на шляху немає жодної вершини) або умову переходу автомату. В комірках таблиці відображається, перехід в який стан виконується та яким буде вихідний сигнал. Наприклад, із стану a_0 автомат здійснює перехід у стан a_1 , в результаті якого на виході автомата буде сигнал Y_1 , тобто автомат виконає мікрооперації, що виконуються протягом одного такту часу (mY_1 , mY_2 , закодовані мікрокомандою Y_1), причому цей перехід буде безумовним (під дією синхроімпульсного сигналу).

Будуємо граф-схему переходів (рис. 2.4). Граф-схема будується на основі рис. 2.3. і таблиці 2.3. Колами позначаються можливі стани автомату. Дуги-стрілки показують переходи з одного стану до іншого стану. Над стрілкою зазначається, під дією якого вхідного сигналу станеться перехід, а також що при цьому буде на виході автомата.

Складемо систему рівнянь переходів на основі граф-схеми переходів (рис. 2.4) або основної таблиці абстрактного автомата (таблиця 2.3).

Для кодування внутрішніх станів автомату визначимо кількість необхідних для цього тригерів (n) із співвідношення: $\log_2 A \leq n$, де A – кількість міток a_i (для нашого автомата $a_0 - a_7$): $A = 8$; $\log_2 8 = 3$; $n=3$.

Оскільки для нашого автомату необхідно три тригери, то внутрішні стани автомата будуть закодовані трирозрядним двійковим кодом. Процес кодування зображено в таблиці 2.4.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. 31 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Таблиця 2.3 – Основна таблиця автомату Мура

| | $CI=1$ | $\overline{X_1}$ | $X_1 X_2$ | $X_1 \overline{X_2}$ | X_3 | $\overline{X_3}$ | X_4 | $\overline{X_4}$ |
|-------|-----------|------------------|-----------|----------------------|-----------|------------------|-----------|------------------|
| a_0 | a_1/Y_1 | | | | | | | |
| a_1 | a_2/Y_2 | | | | | | | |
| a_2 | a_3/Y_3 | | | | | | | |
| a_3 | | a_5/Y_5 | a_4/Y_4 | a_5/Y_5 | | | | |
| a_4 | a_5/Y_5 | | | | | | | |
| a_5 | | | | | a_3/Y_3 | a_6/Y_6 | | |
| a_6 | | | | | | | a_2/Y_2 | a_7/Y_7 |
| a_7 | $a_0/ _$ | | | | | | | |

Таблиця 2.4 – Таблиця кодування внутрішніх станів автомату Мура

| Внутрішній стан | S1 | S2 | S3 |
|-----------------|----|----|----|
| a_0 | 0 | 0 | 0 |
| a_1 | 0 | 0 | 1 |
| a_2 | 0 | 1 | 0 |
| a_3 | 0 | 1 | 1 |
| a_4 | 1 | 0 | 0 |
| a_5 | 1 | 0 | 1 |
| a_6 | 1 | 1 | 0 |
| a_7 | 1 | 1 | 1 |

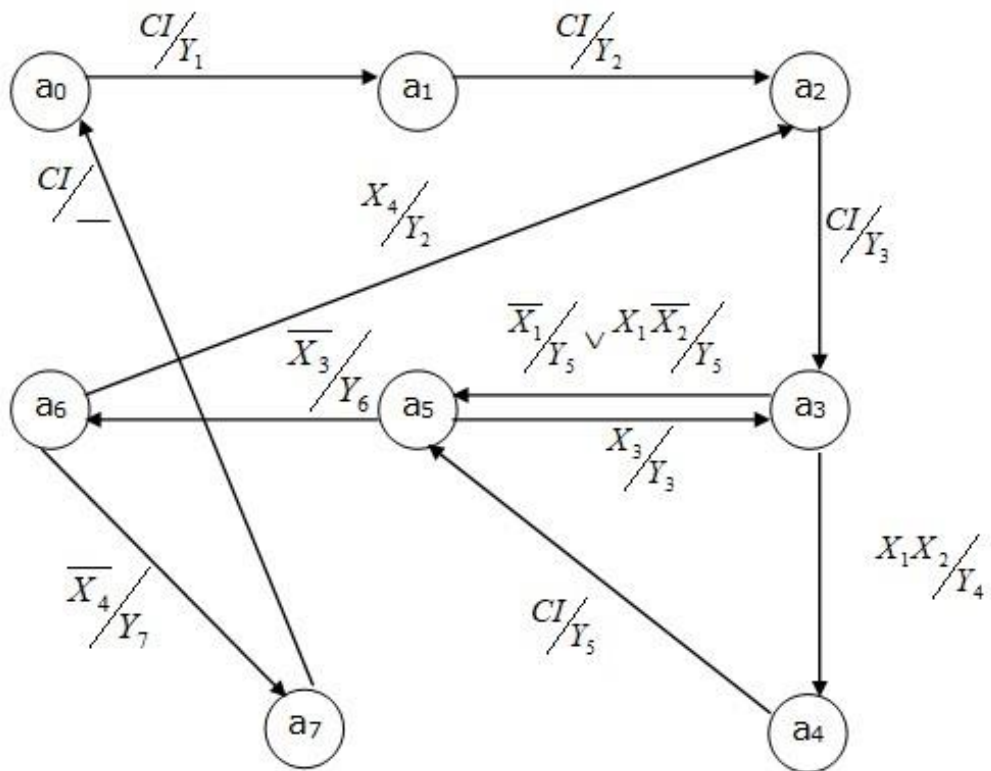


Рисунок 2.4 – Граф-схема переходів автомату Мура

Система рівнянь переходів автомату Мура:

$$\left\{ \begin{array}{l} a_{0t} = a_{7t-1} \cdot 1 \\ a_{1t} = a_{0t-1} \cdot 1 \\ a_{2t} = a_{1t-1} \cdot 1 \vee a_{6t-1} \cdot X_4 \\ a_{3t} = a_{2t-1} \cdot 1 \vee a_{5t-1} \cdot X_3 \\ a_{4t} = a_{3t-1} \cdot X_1 X_2 \\ a_{5t} = a_{4t-1} \cdot 1 \vee a_{3t-1} \cdot (\overline{X_1} \vee X_1 \overline{X_2}) \\ a_{6t} = a_{5t-1} \cdot \overline{X_3} \\ a_{7t} = a_{6t-1} \cdot \overline{X_4} \end{array} \right.$$

2.3 Висновки

В другому розділі за схемою Уілкса-Стрінжера спроектовано операційний автомат на основі автомату Мура, який виконує обчислення кількості парних елементів, що належать інтервалу $[-3; 13]$ у двовимірному масиві $A[n, m]$, а саме:

- 1) розроблено змістовну схему алгоритму;
- 2) закодовано верхівки змістовної схеми алгоритму;
- 3) розроблено закодовану мікроопераційну схему алгоритму;
- 4) закодовано мікрокоманди;
- 5) розроблено закодовану мікрокомандну схему алгоритму;
- 6) побудовано основну таблицю автомату Мура;
- 7) спроектовано граф-схему переходів автомату Мура;
- 8) складено систему рівнянь переходів;
- 9) визначено необхідну кількість елементів пам'яті (тригерів) для реалізації автомату Мура;
- 10) закодовано внутрішні стани автомату.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 34 |

3 АПАРАТНА РЕАЛІЗАЦІЯ ОПЕРАЦІЙНОГО АВТОМАТУ НА ОСНОВІ АВТОМАТУ МУРА

3.1 Схема Уїлкса-Стрінжера

Побудова мікропрограмного автомату за схемою Уїлкса-Стрінжера зводиться до таких дій:

- 1) кодування операційних та умовних верхівок;
- 2) побудова основної таблиці і граф-схеми переходів;
- 3) побудова системи рівнянь для функцій переходів і системи рівнянь для функцій виходів (для автоматів Мілі);
- 4) кодування внутрішніх станів автомату;
- 5) побудова функціональної схеми автомату.

Операційний автомат завжди представляє сукупність функціональних блоків, кожен з яких реалізує певну мікрооперацію, використану в мікропрограмі. Деякі блоки є блоками умовних переходів. Ці блоки формують сигнали істинності мікроумов. В операційному автоматі також розташовано блок пам'яті, який складається з регістрів і блоків оперативного запам'ятовуючого пристрою, які використовуються для зберігання проміжних і підсумкових результатів обчислень.

Операційні функціональні блоки реалізують мікрооперації, які не впливають на порядок виконання інструкцій прошивки, наприклад, операції пересилання, складання тощо. На виході кожного такого функціонального операційного блоку встановлюється тристабільний буфер, який пропускає результат мікрооперації на блок пам'яті, тільки якщо відповідний сигнал встановлено. В іншому випадку на виході тристабільного буферу встановлюється Z-стан. Тобто результат обробки даних операційним функціональним блоком записується в пам'ять тільки при встановленому сигналі.

Умовні функціональні блоки реалізують мікрооперації умовних переходів. Ці блоки представляють собою комбінаційні схеми, які видають сигнали істинності мікроумов. Кожен блок видає два сигнали X_{2n} і X_{2n+1} , де n – це номер

мікроумови. Якщо встановлено сигнал X_{2n} , то мікроумова істинна. Якщо встановлено X_{2n+1} , то мікроумова хибна. Якщо автомат перебуває в стані, в якому дана мікроумова не аналізується, то обидва сигнали скидаються. Не існує ситуації, при якій обидва сигнали могли б бути встановлені.

Наразі реалізовано два види автоматів: найбільш поширений у вигляді схеми Уїлкса-Стрінжера [9] і автомат типу «зсувний регістр». Кожен з цих варіантів має ряд переваг і недоліків. Найчастіше перевагу тому чи іншому варіанту можна віддати тільки після синтезу відповідного автомата. Як показала практика, схема Уїлкса-Стрінжера найбільш споживана при синтезі малих мікропрограм при відсутності жорстких вимог до швидкодії автомату. При синтезі пристроїв за великими мікропрограмами виявилось вигідніше використовувати схему «зсувний регістр», оскільки вона в цьому випадку забезпечує велику економію резервних під пристрій осередків ПЛІС. Якщо ж порівнювати ці варіанти з точки зору швидкодії, то в більшості випадків схема «зсувний регістр» дозволяє працювати автомату на більш високій частоті. Проте, вибір схеми автомату рекомендується проводити після синтезу конкретного мікропрограмного автомату і порівняння його реальних характеристик: кількості комірок, займаних в ПЛІС, і частоти, на якій може працювати схема.

Схема Уїлкса-Стрінжера складається з лічильника, дешифратора і комбінаційної схеми переходів. Поточний стан автомату зіставляється зі значенням, збереженим в лічильнику. Дешифратор служить для перетворення стану автомату в вихідні сигнали автомату.

Механізм роботи схеми Уїлкса-Стрінжера є наступним. У разі лінійного виконання прошивки просто інкрементується лічильник. У разі ж умовного або безумовного переходів в лічильник записується новий стан автомату, отриманий від схеми переходів. Він обчислюється за поточним станом автомату і мікроумовою, відповідною мікрооперації умовного переходу, що виконується в даний момент. У разі безумовного переходу, наступний стан автомату однозначно визначається за поточним.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. 36 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Схема Уїлкса-Стрінжера є найбільш поширеною структурою схемою мікропрограмного автомату. Спрощено схему Уїлкса-Стрінжера може бути представлена у наступному вигляді – рис. 3.1.

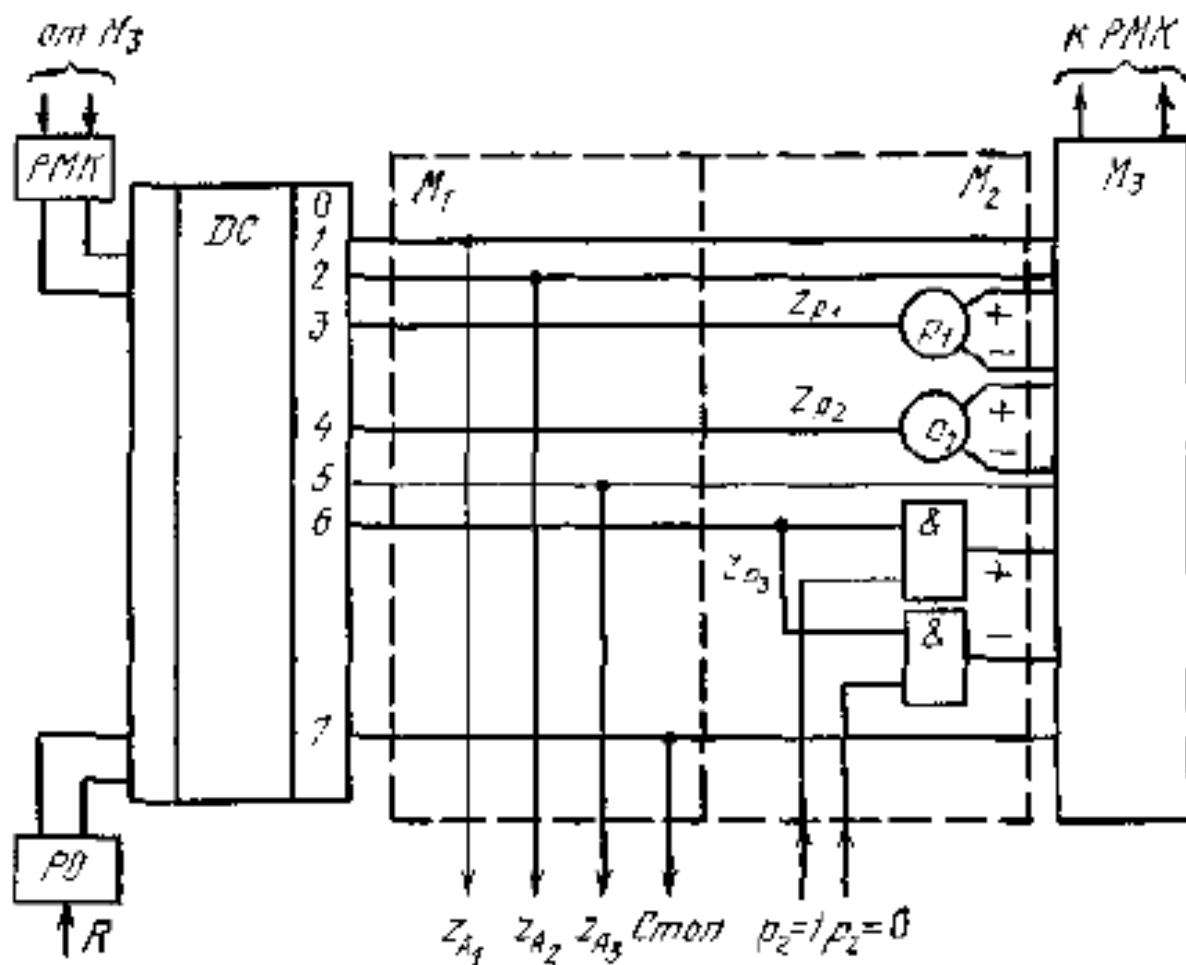


Рисунок 3.1 – Спрощене представлення схеми Уїлкса-Стрінжера [42]

Схема складається з регістру мікрокоманд РМК, дешифратора DC, матриці зовнішніх мікрооперацій M1, матриці внутрішніх мікрооперацій M2, матриці формування коду наступної мікрокоманди M3 і регістру операцій PO.

У разі застосування мікропрограмного автомата кожна зовнішня і внутрішня мікрооперація є керуючими сигналами.

Вихід дешифратора DC зіставляється з мікрокомандою, виконуваною за один такт роботи автомату. Код кожної мікрокоманд зберігається в РМК, який може складатися з тригерів. Після виконання даної мікрокоманди в M3

формується код наступної мікрокоманди, який передається в РМК. Послідовність мікрокоманд утворює мікропрограма, код якої зберігається в РО. При надходженні в РВ коду, автомат починає виробляти відповідно до мікропрограми, якій приписаний відповідний код, послідовність мікрокоманд. При цьому зазвичай передбачається, що код може змінитися лише після того, як закінчиться виконання відповідної мікропрограми. Тому часто РО поєднують з РМК, а код зіставляється з кодом першої мікрокоманди прошивки. Тоді мікрокоманди всіх мікропрограм утворюють одну загальну послідовність (рис. 3.2).



Рисунок 3.2 – Послідовність мікрокоманд всіх мікропрограм [42]

При другій реалізації мікропрограмного автомата кількість розрядів РО-РМК визначається загальною кількістю мікрокоманд, натомість при першій реалізації – найбільшою кількістю мікрокоманд в одній мікропрограмі (розрядність РМК) і кількістю різних мікропрограм (розрядність РО). Вхід R мікропрограмного автомата є зовнішнім входом, РМК утворює пам'ять мікропрограмного автомата. Якщо використовується синхронний мікропрограмний автомат, то імпульси від тактового генератора надходять на вхід дешифратора DC [42].

Мікропрограма, тобто послідовність виконання мікрооперацій, зручно описується мовою логічних схем алгоритмів, причому з оператором співставляється зовнішня мікрооперація, а з логічною умовою – внутрішня мікрооперація.

Відповідно до отриманої кількості виходів дешифратора обирається кількість розрядів РМК. При такій реалізації в кожен мікрокоманду входить тільки одна мікрооперація (зовнішня чи внутрішня). Кількість внутрішніх станів автомату повністю визначається кількістю членів логічної схеми алгоритму. Обсяг матриць M1 і M2 залежить від кількості операторів і логічних умов в логічній схемі алгоритму. В окремому випадку, коли в логічну схему алгоритму входять тільки оператори, матриця M2 відсутня. В матриці M3 при цьому кожен раз формується номер наступної за порядком мікрокоманди.

При наявності в логічній схемі алгоритму логічних умов необхідна матриця M2, в якій формуються сигнали на включення. При хибному (false) значенні логічної умови, яка перевіряється, природний порядок виконання членів логічної схеми алгоритму порушується. Тоді в M3 повинен бути сформований номер тієї мікрокоманди, який необхідний для правильного виконання логічної схеми алгоритму.

Очевидно, що якщо з кожним членом логічної схеми алгоритму співставляти окрему мікрокоманду, для реалізації навіть досить простих алгоритмів потрібна велика кількість внутрішніх станів автомату. Однак в

більшості практичних випадків немає необхідності відводити внутрішній стан для кожного члена логічної схеми алгоритму.

Деякі зовнішні мікрооперації можна виконувати не послідовно, а одночасно – за один мікротакт. Тоді з кожним внутрішнім станом автомату співставляється не один оператор, а група одночасно виконуваних операторів логічної схеми алгоритму. Природно, при цьому зменшується кількість виходів дешифратора, що, в свою чергу, може привести до скорочення числа розрядів РМК. Крім скорочення обсягу обладнання, суміщення в часі окремих мікрооперацій приводить до підвищення швидкодії, оскільки зменшується кількість мікротактів, необхідних для виконання алгоритму.

Подальше спрощення схеми автомату може бути досягнуто за рахунок одночасного виконання зовнішніх і внутрішніх мікрооперацій. Тоді кількість мікрокоманд МА буде визначатися не кількістю членів логічної схеми алгоритму, а кількістю груп одночасно виконуваних мікрооперацій.

При розгляді різних можливих способів побудови схеми автомату досі ми вважали, що автоматом реалізується тільки один алгоритм. Однак програмний спосіб управління використовується саме тоді, коли в автоматі необхідно реалізувати декілька різних алгоритмів. В такому разі автомат перестає бути автономним, оскільки є зовнішні входи. Залежно від набору значень цих змінних в автоматі повинен бути реалізований один з алгоритмів.

Зовнішні сигнали можуть надходити у автомат з реєстра операцій. Очевидно, розрядність цього реєстра визначається кількістю різних алгоритмів, що реалізуються автоматом. Ці алгоритми є частковими, на відміну від загального алгоритму, що описує роботу автомату в цілому.

Найбільш поширена схемна реалізація неавтономного мікропрограмного автомату представлена на рис. 3.1. В цій схемі, крім РМК, що входить до складу автономного автомату, є РО, з якого на дешифратор надходять вхідні сигнали. Розрядність РМК визначається найбільшою кількістю мікрокоманд, необхідних для реалізації часткової логічної схеми алгоритму. Кількість виходів з

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. 40 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

дешифратора дорівнює сумарній кількості мікрокоманд, необхідних для реалізації всіх часткових логічних схем алгоритмів. У найгіршому випадку, коли в мікрокоманду входить тільки один член логічної схеми алгоритму, кількість виходів з дешифратора буде дорівнювати сумарній кількості членів всіх часткових логічних схем алгоритмів. При такій реалізації функції, що описують вихідні сигнали автомату, і функції збудження елементів пам'яті, які формуються в матриці МЗ, залежать від всіх сигналів з РО і РМК, тобто від усіх вхідних сигналів автомату. Крім того, оскільки різні часткові логічні схеми алгоритму містять багато однакових операторів і логічних умов, їм будуть відповідати різні виходи дешифратора. Тому для спрощення схеми автомату бажано скоротити кількість повторюваних членів в логічній схемі алгоритму. Така операція здійснюється в процесі об'єднання часткових алгоритмів.

В схемі автомату, побудованій за об'єднаною логічною схемою автомату, функції збудження елементів пам'яті залежать від всіх вхідних сигналів автомата, а вихідні функції – тільки від сигналів з елементів пам'яті. При такій реалізації загальний обсяг обладнання автомату може скоротитися.

Однак для вибору того чи іншого способу реалізації потрібно оцінити для кожного конкретного випадку складність окремих блоків автомату. Очевидно, що складність матриці МЗ буде залежати від обраного варіанту кодування внутрішніх станів автомату. Від обраного коду буде залежати і стійкість автомата.

Таким чином, при кодуванні мікропрограмного автомату виникають ті ж завдання, що і при кодуванні кінцевих автоматів загального типу. Оскільки внутрішні стану автомату співставляються з мікрокомандами, то очевидно, що структура графа переходів мікропрограмного автомату значно залежить від вибору мікрокоманд.

При занадто великих мікрокомандах граф переходу МА буде сильнозв'язним, що ускладнить задачу усунення критичних змагань при кодуванні автомата. Ситуація тут повністю аналогічна тій, яка виникає при мінімізації кількості внутрішніх станів кінцевого автомата: в результаті етапу мінімізації

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. 41 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

може бути отримана така таблиця переходів, в якій не можна усунути критичні змагання, не ввівши додаткових елементів пам'яті, тобто зусилля, які були витрачені на отримання мінімального автомата, виявляться марними, оскільки потім будуть потрібні не менші зусилля, щоб розширити таблицю переходів на етапі кодування.

Таким чином, важливим завданням є розробка методів, що дозволяють одночасно здійснювати мінімізацію і кодування автомата з урахуванням різних вимог, тобто формування мікрокоманд, що дозволяє скоротити кількість внутрішніх станів мікропрограмного автомата. Така постановка задачі представляє інтерес, оскільки часто зустрічається у випадку, коли для усунення критичних змагань вже прийняті якісь заходи, наприклад, застосована подвійна пам'ять.

При використанні для побудови мікропрограмного автомата схеми Уілкса-Стрінжера кожна мікрокоманда співставляється з внутрішнім станом автомата. У зв'язку з цим завдання вибору кількості мікрокоманд виявляється вельми важливим, оскільки як від його розв'язання істотно залежить складність мікропрограмного автомата.

Будемо вважати, що робота мікропрограмного автомата задається однією логічною схемою алгоритму, в процесі виконання якої значення всіх її логічних умов змінюються тільки операторами цієї логічної схеми алгоритму або залишаються незмінними до закінчення виконання логічної схеми алгоритму.

Один і той же член логічної схеми алгоритму може входити в кілька різних гілок. Групу членів логічної схеми алгоритму називається сумісною, якщо в кожній гілці, утвореній членами цієї групи, не міститься суперечливих членів. Сумісну групу членів логічної схеми алгоритму є максимальною, якщо додавання в неї будь-якого іншого члена логічної схеми алгоритму перетворює її в несумісну.

Кожна максимальна сумісна група членів логічної схеми алгоритму може інтерпретуватися як окрема мікрокоманда. Така мікрокоманда складається з

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. 42 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

набору як внутрішніх, так і зовнішніх мікрооперацій, але при певному наборі значень логічних умов одночасно виконуються мікрооперації, що входять лише в одну гілку мікрокоманд.

Майже всі отримані мікрокоманд є пересічними, оскільки одні й ті ж члени логічної схеми алгоритму входять в різні мікрокоманди. Це є причиною того, що кількість входжень операторів і логічної схеми алгоритму в отриману систему мікрокоманд значно перевищує кількість їх входжень у початкову логічну схему алгоритму.

Складність схеми мікропрограмного автомату визначає саме отримана система мікрокоманд. Таким чином, кількість членів заданої логічної схеми алгоритму не може служити хорошою характеристикою складності схеми мікропрограмного автомату. Точніше визначає складність схеми мікропрограмного автомату інша логічна схема алгоритму, яка є розширеною.

Розширену логічну схему алгоритму складають за існуючою системою мікрокоманд, і кількість її членів дорівнює кількості входжень операторів і логічних умов в систему мікрокоманд.

Якщо у вихідній логічній схемі алгоритму немає повторюваних членів, то в розширеній логічній схемі алгоритму одні і ті ж члени можуть неодноразово повторюватися.

Перехід до складних мікрокоманд ускладнює матриці M1 і M2, але спрощує дешифратор і зменшує розрядність регістру мікрокоманд.

3.2 Опис апаратної реалізації операційного автомату на основі автомату Мура

Операційний автомат складається з трьох частин (схема електрична функціональна, додаток А).

У вхідній частині розташовані 3 D-тригери, 3 логічних елементи АБО, які приймають вхідний сигнал, дешифратор та 2 шини (одна з шин використовується

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КвРКІ 190362.17.03.09 ПЗ | Арк. 43 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

для передачі сигналів з дешифратора, а інша шина використовується для передачі сигналів з виходів компаратора).

Тригер є елементарним цифровим автоматом з пам'яттю. Це пристрій з двома стійкими станами, в які він переходить під дією певних вхідних сигналів. Тригери мають 2 типи вхідних сигналів – інформаційні та синхронізуючі. Інформаційні сигнали визначають новий стан тригера і є в будь-яких тригерах. Якраз саме за типом інформаційних сигналів здійснюється класифікація тригерів. Синхронізуючий сигнал не є обов'язковим, він вводиться і використовується в тригерах для фіксації моменту переходу тригера в новий стан, який задається інформаційними входами. D-тригер – це елемент затримки. Він має 1 інформаційний вхід D і 1 вихід Q, використовується для здійснення затримки на один такт сигналу, який надійшов на його вхід.

Умовне позначення D-тригера представлена на рис. 3.3.

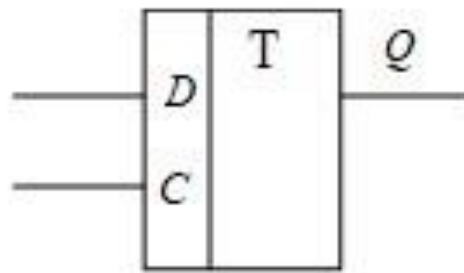


Рисунок 3.3 – Умовне позначення D-тригера

Таблиця переходів D-тригера ($Q^{t+1} = f(Q^t, D^t)$) представлена в таблиці 3.1.

Таблиця 3.1 – Таблиця переходів D-тригера

| D | Q^t | Q^{t+1} |
|---|-------|-----------|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Таблиця функцій збудження (входів) D-тригера ($D^t = \varphi(Q^t, Q^{t+1})$) отримується з таблиці переходів (з таблиці 3.1) і представлена в таблиці 3.2.

Таблиця 3.2 – Таблиця функцій збудження (входів) D-тригера

| Q^t | Q^{t+1} | D^t |
|-------|-----------|-------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Як видно з таблиці 3.2, стан, у який переходить тригер (Q^{t+1}), збігається з сигналом D^t , що надійшов на вхід тригера. Тоді таблиця функцій збудження D-тригерів буде цілком збігатися з кодованою таблицею переходів цього тригера.

Диз'юнкція (логічне додавання, функція АБО) – це функція $f(x_1, x_2)$, яка є істинною тоді, коли істинним є хоча б один з її аргументів. Умовне позначення цієї функції: $f(x_1, x_2) = x_1 + x_2 = x_1 \vee x_2$.

Умовне позначення логічного елемента АБО представлена на рис. 3.4.

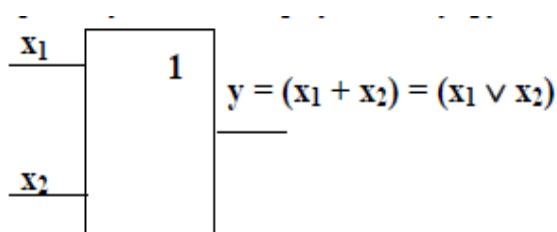


Рисунок 3.4 – Умовне позначення логічного елемента АБО

Таблиця істинності логічного елемента АБО представлена в таблиці 3.3.

Таблиця 3.3 – Таблиця істинності логічного елемента АБО

| x1 | x2 | y |
|----|----|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Дешифратор – операційний елемент, який виконує функцію перетворення певного n-розрядного двійкового коду в унітарний код «один з N». Якщо $N=2^n$, то дешифратор є повним. Якщо $N < 2^n$, то дешифратор є частковим.

Таблиця істинності найпростішого повного дешифратора і його умовні позначення наведені в таблиці 3.4 і на рис. 3.5 відповідно.

Таблиця 3.4 – Таблиця істинності дешифратора

| e ₁ | e ₂ | f ₁ | f ₂ | f ₃ | f ₄ |
|----------------|----------------|----------------|----------------|----------------|----------------|
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |

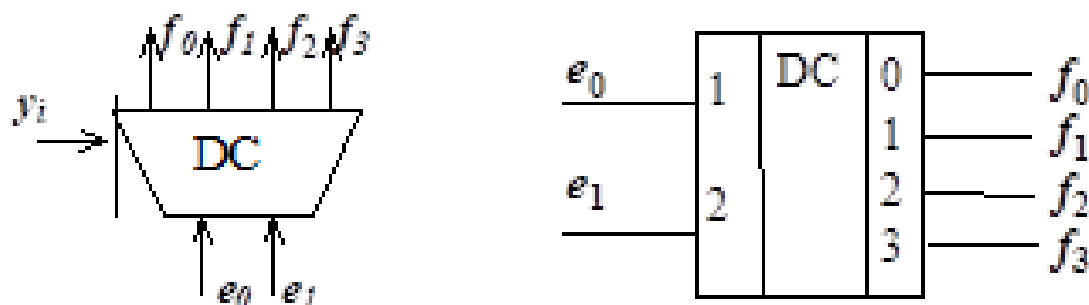


Рисунок 3.5 – Умовні позначення дешифратора

рівності або нерівності поданих на його входи чисел A і B і генерують на виході сигнал рівності (логічну 1) або нерівності (логічний 0). Числа A і B можуть мати певну кількість розрядів і подаватись на входи порозрядно. Компаратор рівності порівнює окремі розряди в такому випадку. Такий компаратор рівності реалізується за допомогою двох логічних елементів Виключаюче АБО (елементи $D1$ і $D2$) та логічного елементу ТА ($D3$) (рис. 3.7).

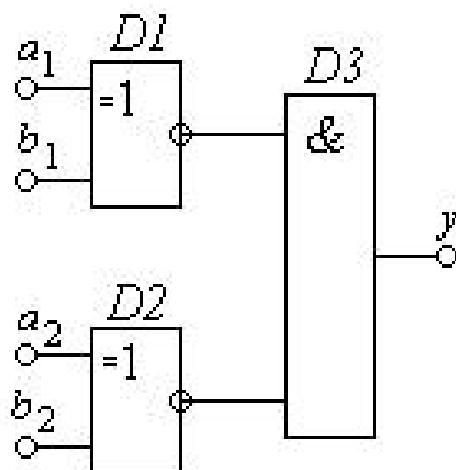


Рисунок 3.7 – Найпростіший компаратор рівності

На виходах елементів $D1$ і $D2$ рівень логічної 1 з'являється тоді, коли $a_1 = b_1$ та $a_2 = b_2$. Тому, на виході y компаратора логічна одиниця присутня лише тоді, коли рівними є обидва розряди двійкових чисел A і B .

У другому випадку (компаратор нерівності) компаратори можуть визначати не тільки рівність чисел A і B , але й їх нерівність, тобто $A > B$ або $A < B$. Компаратори нерівності будуються на основі більш складної схеми. Також існують компаратори нерівності, які мають 2 виходи: y_1 та y_2 . При $A < B$ на виході y_1 з'являється рівень логічної одиниці, а при $A > B$ – на виході y_2 . Порівняння також виконуються порозрядно. Такі компаратори є повними компараторами.

Умовне позначення повного компаратора наведено на рис. 3.8.

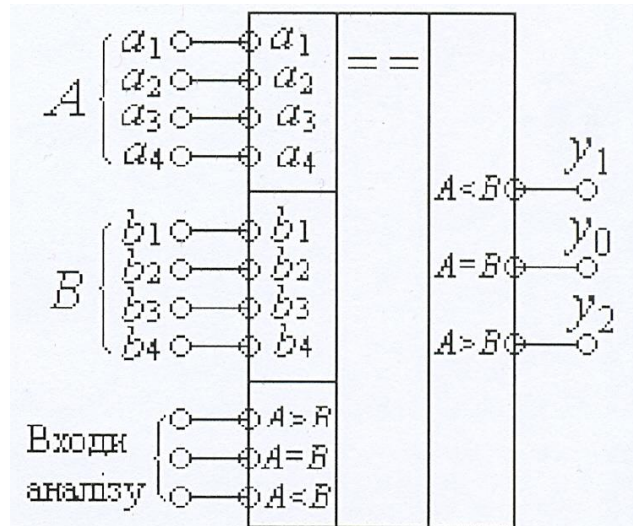


Рисунок 3.8 – Умовне позначення повного компаратора

У перехідній частині схеми електричної функціональної автомату виконується перетворення сигналу протягом одного такту часу.

Після проходження через логічні елементи ТА і АБО, сигнал змінюється, і результат надходить на шину, після чого передається до програмованої логічної матриці.

Перехідна частина базується на системі рівнянь переходів.

Кон'юнкція (логічне множення, функція ТА) – це функція $f_1(x_1, x_2)$, яка є істинною тоді і тільки тоді, коли всі її змінні є одночасно істинними. Умовно цю функцію позначають у такий спосіб:

$$f_1(x_1, x_2) = x_1 \wedge x_2 = x_1 \cdot x_2 = x_1 \& x_2.$$

Умовне позначення логічного елемента ТА представлена на рис. 3.9.

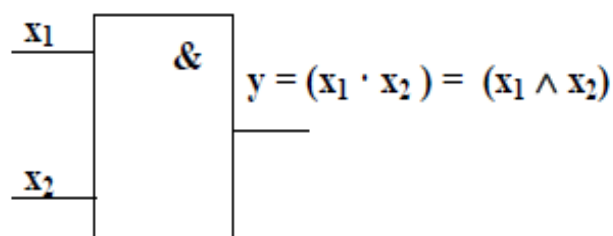


Рисунок 3.9 – Умовне позначення логічного елемента ТА

Таблиця істинності логічного елемента ТА представлена в таблиці 3.6.

Таблиця 3.6 – Таблиця істинності логічного елемента ТА

| x1 | x2 | y |
|----|----|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Першим представником великого класу програмованих логічних пристроїв (ПЛІС) стали програмовані логічні матриці (ПЛІМ).

У зарубіжній літературі вони називаються PLA (programmable logic array).

Узагальнена структура ПЛІМ наведена на рис. 3.8.

Основна ідея роботи ПЛІМ полягає в реалізації логічної функції, представленої в ДДНФ – досконалій диз'юнктивній нормальній формі.

На рис. 3.9 чітко простежуються логічні елементи ТА, здатні реалізувати будь-який мінтерм ДДНФ і логічні елементи АБО, які здійснюють додавання термів, що вимагаються згідно логічного виразу ДДНФ.

У схемі ПЛІМ, наведеної на рис. 3.10, ранг терму обмежений кількістю входів і дорівнює чотирьом, кількість термів теж дорівнює чотирьом.

Приклад внутрішньої структури ПЛІМ, прийнятий в зарубіжній літературі, представлено на рис. 3.11.

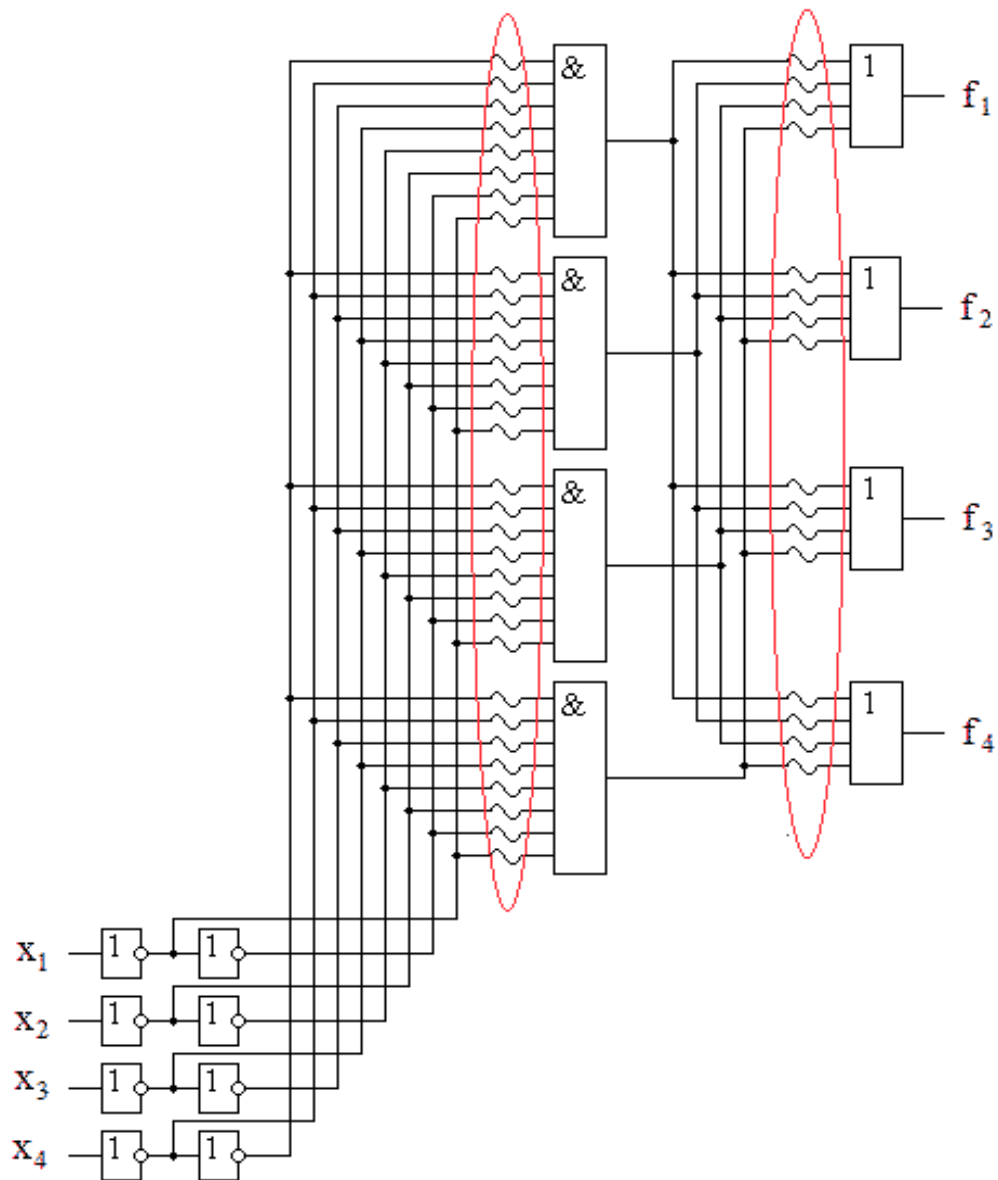


Рисунок 3.10 – Узагальнена структура ПЛМ

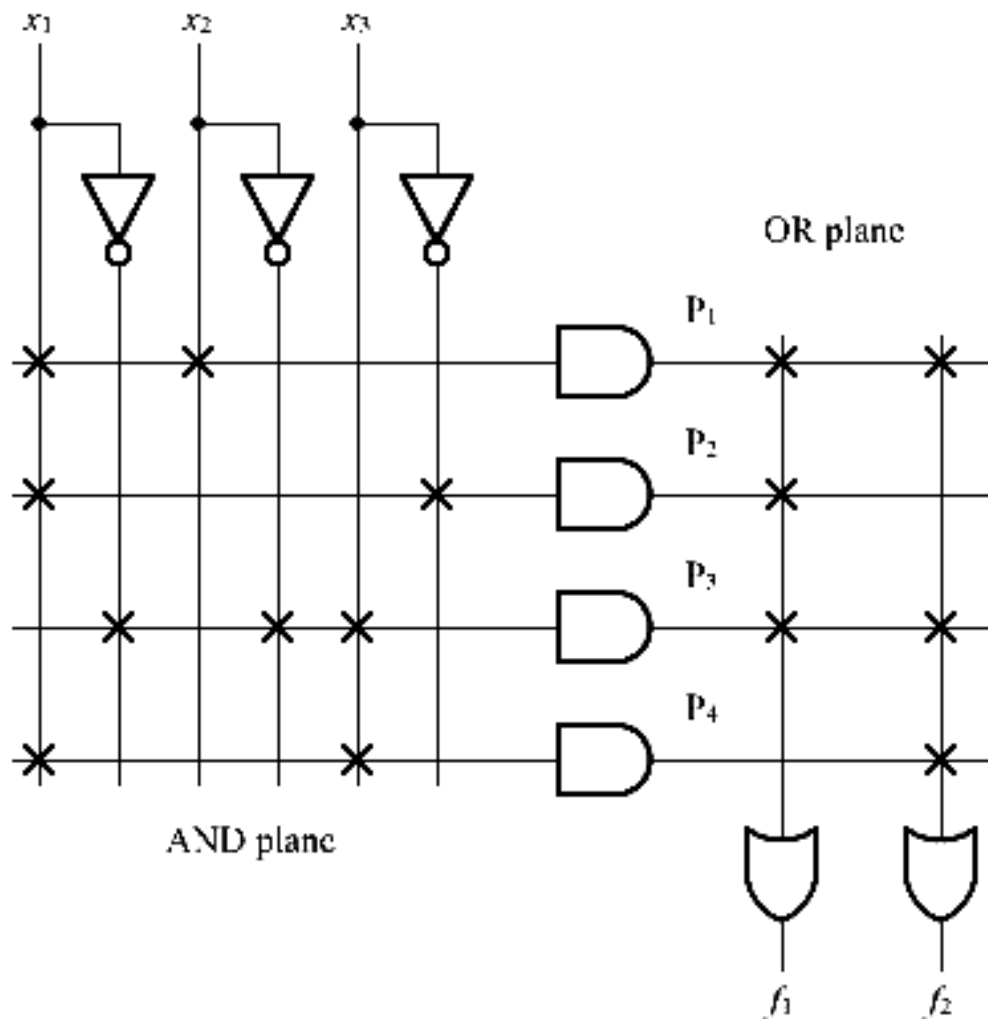


Рисунок 3.11 – Приклад внутрішньої структури ПЛМ, прийнятий в зарубіжній літературі

Вихідна частина схеми електричної функціональної автомату базується на системі рівнянь виходів. У вихідній частині виконується процес, подібний до перехідної частини, але сигнали подаються на вихідну шину Y_i , звідки сигнал передається до вихідної програмованої логічної матриці.

На рис. 3.12 представлено функціональну схему операційного автомату як автомату Мура, який виконує обчислення кількості парних елементів, що належать інтервалу $[-3; 13]$ у двовимірному масиві $A[n, m]$.

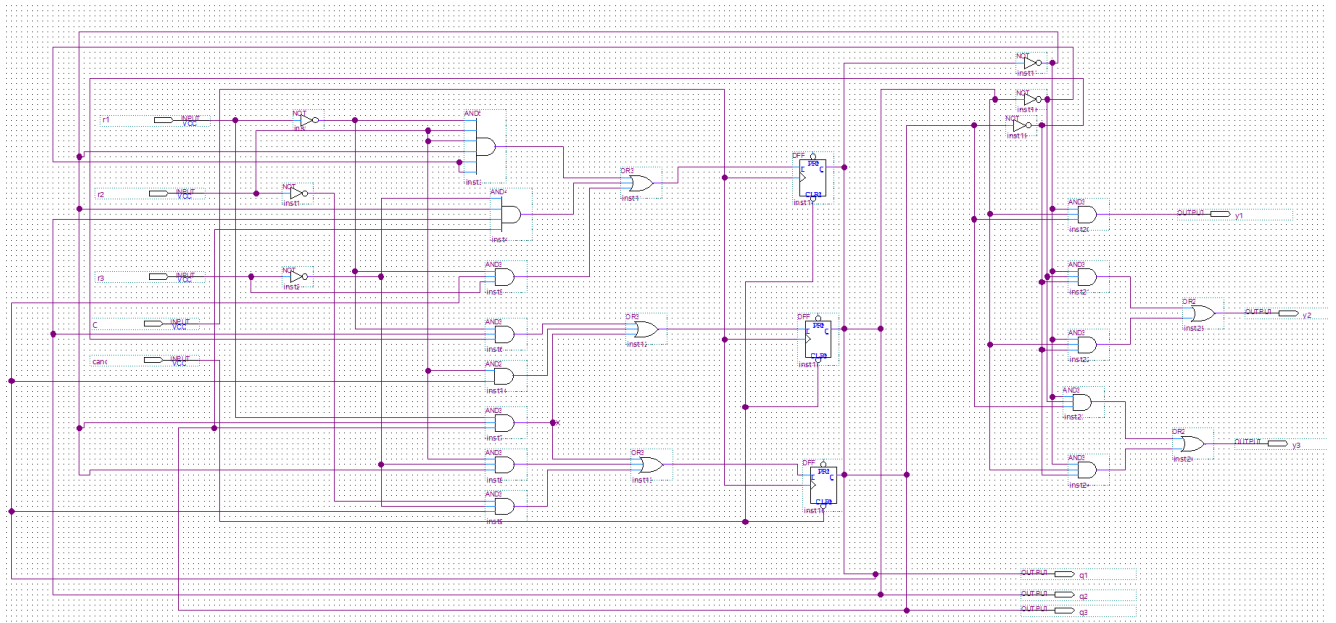


Рисунок 3.13 – Модель операційного автомату на основі автомату Мура в середовищі Quartus II

Розроблена модель була успішно скомпільована, тому далі відбулась перевірка станів і виходів автомату Мура на VWF – перевірка реакції вихідних сигналів на різні вхідні сигнали, а також на різні внутрішні стани.

Перевірка на VWF відбувалась на основі побудованої основної таблиці автомату Мура – рис. 3.14-3.17.

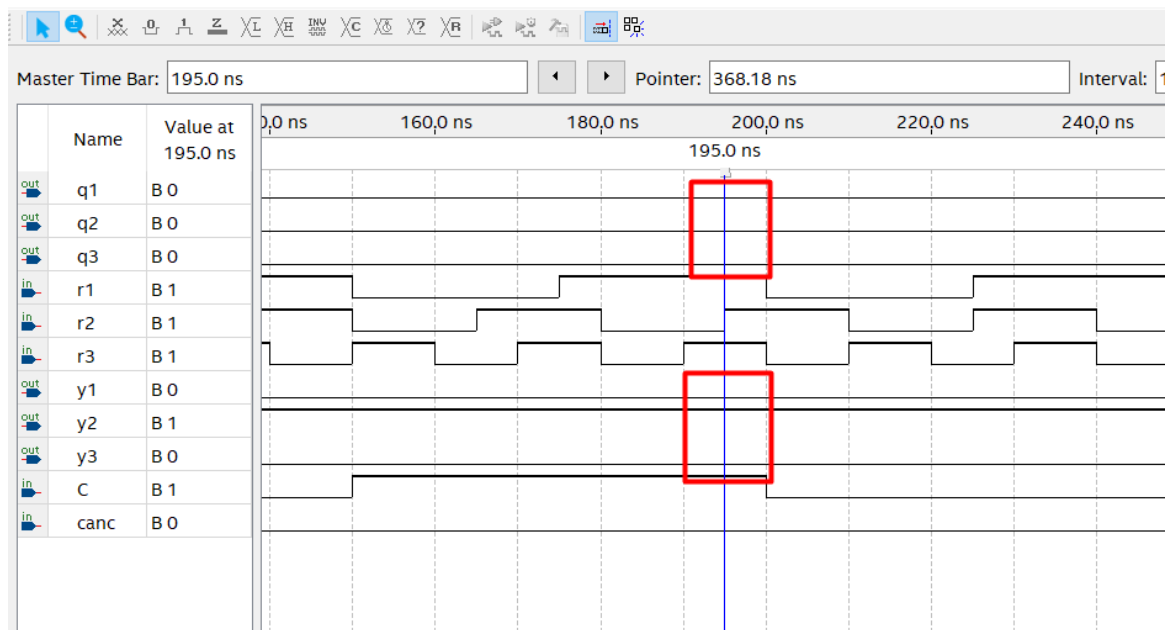


Рисунок 3.16 – Перевірка на VWF на основі основної таблиці автомату Мура на проміжку часу 750-770 наносекунд

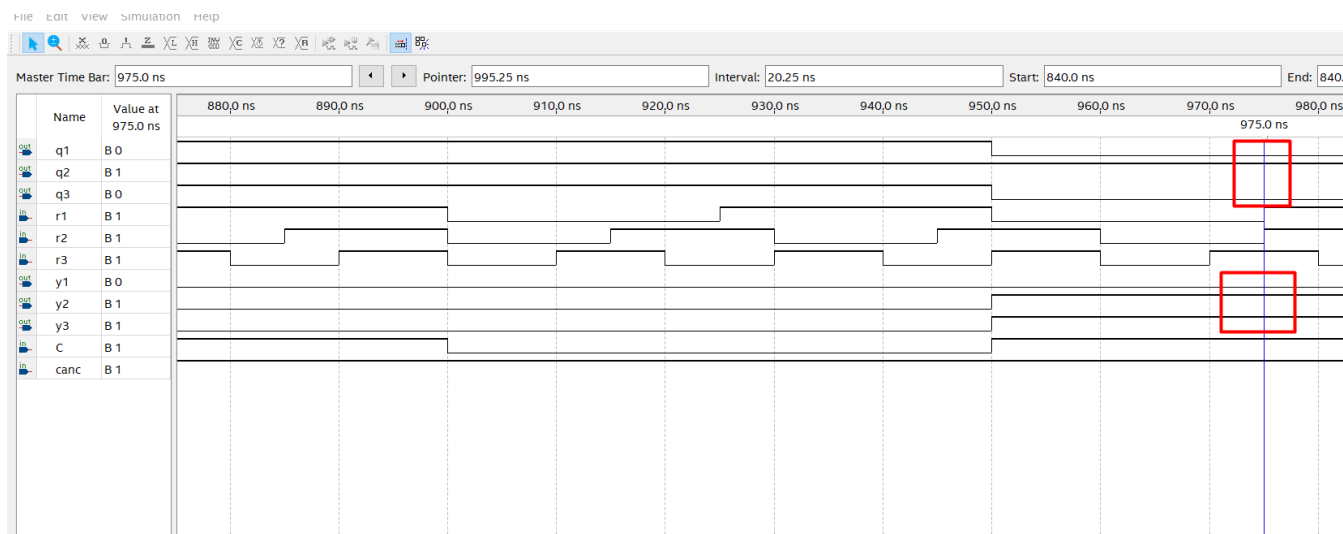


Рисунок 3.17– Перевірка на VWF на основі основної таблиці автомату Мура на проміжку часу 970-980 наносекунд

Порівняння отриманих при моделюванні результатів з основною таблицею автомату Мура (таблиця 2.3) та граф-схемою переходів автомату Мура (рис. 2.4) показало, що результати збігаються, отже, автомат працює коректно.

3.4 Висновки

В третьому розділі кваліфікаційної роботи виконано апаратну реалізацію операційного автомату на основі автомату Мура, а саме:

- 1) реалізовано схему електричну функціональну операційного автомату на основі автомату Мура;
- 2) змодельовано схему в середовищі Quartus II.

ВИСНОВКИ

Кінцеві автомати активно використовуються в комп'ютерних іграх, в реалізації мережевих протоколів, системах стискання інформації. Інакше кажучи, вони використовуються, де потрібна велика надійність і де є складна логіка звернення, яку програміст не може реалізувати на рівні здорового глузду.

Необхідна зручна абстракція, здатна обрати з алгоритму суть логіки його роботи і уможливити проведення всього необхідного аналізу.

Кінцеві автомати, різновидами яких є автомати Мілі і Мура, є однією із зручних абстракцій. Близькість до алгебри Буля і теорії графів, детермінованість поведінки та наочність графічного представлення є визначними перевагами цієї абстракції.

Автомати Мілі (Mealy) – це цифрові автомати, в яких вхідні сигнали та попередній стан автомату визначають сигнали виходів.

Автомати Мура (Moore) – це автомати, в яких попередній стан автомату визначає сигнали виходів, а от від значень вхідних сигналів сигнали виходів не залежать.

Будь-який операційний пристрій можна синтезувати у вигляді композиції операційного і керуючого автоматів.

Реалізуючи операції над словами інформації, операційний автомат представляє собою виконавчу частину пристрою. Роботу операційного автомату організовує керуючий автомат, який видає визначені послідовності керуючих сигналів.

В результаті виконання кваліфікаційної роботи було змодельовано, спроектовано та реалізовано (у вигляді схем) операційний автомат, що виконує обчислення кількості парних елементів, що належать інтервалу $[-3; 13]$ у двовимірному масиві $A[n, m]$. Мікропроцесорний автомат у вигляді автомата Мура реалізовано за схемою Уілкса-Стрінжера. Функціональну схему керуючих

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. |
| | | | | | | 57 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

частин автомата синтезовано на логічних елементах ТА, АБО, НІ з використанням D-тригерів в якості елементів пам'яті.

Об'єктом дослідження є програмно-технічний (апаратний) засіб – операційний автомат на основі автомату Мура.

Предметом дослідження є формалізований опис та схеми операційного автомату на основі автомату Мура.

Для досягнення поставленої мети використовувались принципи системного аналізу, методи синтезу, аналізу та моделювання процесів, теоретико-множинні підходи.

В першому розділі кваліфікаційної роботи проведено дослідження предметної області (проаналізовано теорію операційних та керуючих автоматів, автоматів Мілі та Мура, а також розроблення і опис схем електричних функціональної і принципової) та виконано постановку задачі дослідження.

В другому розділі кваліфікаційної роботи проведено моделювання та проектування операційного автомату на основі автомату Мура, а саме:

- 1) виконано формалізований опис операційного автомату;
- 2) розроблено змістовну схему алгоритму;
- 3) закодовано верхівки змістовної схеми алгоритму;
- 4) розроблено закодовану мікроопераційну схему алгоритму
- 5) закодовано мікрокоманди;
- 6) розроблено закодовану мікрокомандну схема алгоритму;
- 7) побудовано основну таблицю автомату Мура;
- 8) спроектовано граф-схему переходів автомату Мура;
- 9) складено систему рівнянь переходів;
- 10) визначено необхідну кількість елементів пам'яті (тригерів) для реалізації автомату Мура;
- 11) закодовано внутрішні стані автомату;
- 12) розроблено структурну схему операційного автомату;
- 13) побудовано схему операційного автомату.

В третьому розділі кваліфікаційної роботи виконано апаратну реалізацію операційного автомату на основі автомату Мура, а саме:

- 1) реалізовано схему електричну функціональну операційного автомату на основі автомату Мура;
- 2) змодельовано схему в середовищі Quartus II.

Практичне значення має змодельований, спроектований та реалізований операційний автомат на основі автомату Мура, який використовується для обчислення кількості парних елементів, що належать інтервалу $[-3; 13]$ у двовимірному масиві $A[n, m]$.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 59 |

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Матвієнко М.П. Комп'ютерна логіка: навчальний посібник. Київ: ТОВ "Центр навчальної літератури", 2012. 288 с.
2. Матвієнко М.П. Архітектура комп'ютерів: навчальний посібник. Київ: ТОВ "Центр навчальної літератури", 2012. 264 с.
3. Матвієнко М.П. Комп'ютерна схемотехніка: навчальний посібник. Київ: ТОВ "Центр навчальної літератури", 2012. 190 с.
4. Никитин В.А. Схемотехника интегральных схем ТТЛ, ТТЛШ и КМОП: учебное пособие. Москва: НИЯУ МИФИ, 2010. 64 с.
5. Микушин А. В. Сажнев А. М. Сединин В. И. Цифровые устройства и микропроцессоры. Санкт-Петербург: БХВ-Петербург, 2010. 640 с.
6. Говорущенко Т. О. Комп'ютерна логіка: практикум: навчальний посібник. Хмельницький: Хмельницький національний університет, 2018. 294 с.
7. Ожиганов А.А. Теория автоматов: учебное пособие. Санкт-Петербург: НИУ ИТМО, 2013. 84 с.
8. Джон Хопкрофт, Раджив Мотвани, Джеффри Ульман. Введение в теорию автоматов, языков и вычислений. 2-е издание. Москва: Диалектика-Вильямс, 2016. 528 с.
9. Кудрявцев В.Б. Теория автоматов: учебник для бакалавриата и магистратуры. Москва: ВАМ, 2018. 698 с.
10. Дж. Фон Нейман. Теория самовоспроизводящихся автоматов. Москва: ВАМ, 2010. 673 с.
11. Вельдер С.Э., Лукин М.А., Шалыто А.А., Яминов Б.Р. Верификация автоматных программ. Санкт-Петербург: Наука, 2011. 244 с.
12. Гуренко В.В. Введение в теорию автоматов: учебное пособие. Теория + Примеры. Москва: МГТУ имени Н.Э. Баумана, 2013. 63 с.
13. Иванов Н.М., Соловьева Т.Н. Конечные автоматы. Санкт-Петербург: ГУАП, 2015. 76 с.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. 60 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

14. Илюхин А.В., Колбасин А.М. Логические автоматы. Типовые последовательностные схемы. Учебное пособие. В 2 ч. Москва: МАДИ, 2013. 52 с.

15. Лобанов А.И. Модели клеточных автоматов. "Компьютерные исследования и моделирование". 2010. Т.2. №3. С. 273–293.

16. Поликарпова Н.И., Шалыто А.А. Автоматное программирование. Санкт-Петербург: Питер, 2010. 176 с.

17. Кузенков М.В. Проектирование автомата Мура. Микропрограммный автомат Мили. Красноярск: СФУ ИКИТ, 2010. 11 с.

18. Рощин А.Г. Теория автоматов. Часть I. Учебное пособие. Москва: МГТУ ГА, 2015. 28 с.

19. Сухинин Б.М. Разработка и исследование высокоскоростных генераторов псевдослучайных равномерно распределенных двоичных последовательностей на основе клеточных автоматов. Диссертация на соискание ученой степени кандидата технических наук. Москва, МГТУ им. Баумана, 2011. 224 с.

20. Титов И.И. Проектирование управляющих автоматов. Методические указания к выполнению курсовой работы для студентов направления 230100 - «Информатика и вычислительная техника» по профилю «Автоматизированные системы обработки информации и управления». Нижний Новгород: НГТУ, 2012. 23 с.

21. Salcido A. Cellular Automata - Simplicity Behind Complexity. InTech, 2011. 580 pp.

22. Зацерковний В.І. Обчислювальна техніка: історія розвитку від найпростіших пристроїв для лічби до електромеханічних комп'ютерів: монографія. Ніжин: Аспект-Поліграф, 2012. 416 с.

23. Информационная революция и становление информационного общества. Санкт-Петербург.: Невский институт языка и культуры, 2014. 22 с.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. 61 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

24. Klimovich A.S., Solov'ev V.V. Transformation of a Mealy finite-state machine into a Moore finite-state machine by splitting internal states. J. Comput. Syst. Sci. Int. 2010. Vol. 49. Pp. 900–908.

25. Solov'ev V.V. Implementation of finite-state machines based on programmable logic ICs with the help of the merged model of Mealy and Moore machines. J. Commun. Technol. Electron. 2013. Vol. 58. Pp.172–177.

26. Aarts F., Kuppens H., Tretmans J. et al. Improving active Mealy machine learning for protocol conformance testing. Mach Learn. 2014. Vol. 96. Pp. 189–224.

27. Dogra D., Ahmed A. & Bhaskar H. Smart video summarization using Mealy machine-based trajectory modelling for surveillance applications. Multimed Tools Appl. 2016. Vol. 75. Pp. 6373–6401.

28. Klimovich A.S., Solov'ev V.V. Minimization of Mealy finite-state machines by internal states gluing. J. Comput. Syst. Sci. Int. 2012. Vol. 51. Pp. 244–255.

29. Solov'ev V.V. Minimization of Mealy finite state machines via internal state merging. J. Commun. Technol. Electron. 2011. Vol. 56. Pp. 207–213.

30. Solov'ev V.V. Minimization of Mealy finite-state machines by using the values of the output variables for state assignment. J. Comput. Syst. Sci. Int. 2017. Vol. 56. Pp. 96–104.

31. Klimowicz A.S., Solov'ev V.V. Minimization of incompletely specified Mealy finite-state machines by merging two internal states. J. Comput. Syst. Sci. Int. 2013. Vol. 52. Pp. 400–409.

32. Klimovich A.S., Solov'ev V.V. Transformation of a Mealy finite-state machine into a Moore finite-state machine by splitting internal states. J. Comput. Syst. Sci. Int. 2010. Vol. 49. Pp. 900–908.

33. Solov'ev V.V. Implementation of finite-state machines based on programmable logic ICs with the help of the merged model of Mealy and Moore machines. J. Commun. Technol. Electron. 2013. Vol. 58. Pp. 172–177.

34. Aarts F., Kuppens H., Tretmans J. et al. Improving active Mealy machine learning for protocol conformance testing. Mach Learn. 2014. Vol. 96. Pp. 189–224.

35. Klimovich A.S., Solov'ev V.V. Minimization of Mealy finite-state machines by internal states gluing. J. Comput. Syst. Sci. Int. 2012. Vol. 51. Pp. 244–255.
36. Paiva S.C., Simao A. Generation of complete test suites from Mealy input/output transition systems. Form Asp Comp. 2016. Vol. 28. Pp. 65–78.
37. Steffen B., Isberner M., Naujokat S. et al. Property-driven benchmark generation: synthesizing programs of realistic structure. Int J Softw Tools Technol Transfer. 2014. Vol. 16. Pp. 465–479.
38. Giantamidis G., Tripakis S., Basagiannis S. Learning Moore machines from input–output traces. Int J Softw Tools Technol Transfer. 2021. Vol. 23. Pp. 1–29.
39. Klimowicz A.S., Solov'ev V.V. Minimization of incompletely specified Mealy finite-state machines by merging two internal states. J. Comput. Syst. Sci. Int. 2013. Vol. 52. Pp. 400–409.
40. Solov'ev V.V. Minimization of mealy finite-state machines by using the values of the output variables for state assignment. J. Comput. Syst. Sci. Int. 2017. Vol. 56. Pp. 96–104.
41. Barkalov A.A., Titarenko L.A., Barkalov A.A. Structural decomposition as a tool for the optimization of an FPGA-based implementation of a mealy FSM. Cybern Syst Anal. 2012. Vol. 48. Pp. 313–322.
42. Кононюк А.Е. Дискретно-непрерывная математика. Книга 11. Автоматы. Часть 2. Детерминированные автоматы. Киев: Освіта України, 2017. 578 с.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 190362.17.03.09 ПЗ | Арк. 63 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Додаток А (обов'язковий)

Копія креслення «Логічні схеми алгоритмів»

КЕРКІ. 190362.17.03.09

Логічні схеми алгоритмів

КЕРКІ. 190362.17.03.09 Е8

```

      graph TD
        Start([Початок]) --> MY1[mY1]
        MY1 --> MY2[mY2]
        MY2 --> MY3[mY3]
        MY3 --> MY4[/mY4/]
        MY4 --> X1{X1}
        X1 -- 1 --> MY5[mY5]
        MY5 --> X2{X2}
        X2 -- 1 --> MY6[mY6]
        X2 --> X1
        X1 --> X3{X3}
        X3 -- 1 --> MY7[mY7]
        X3 --> X4{X4}
        X4 -- 1 --> MY8[/mY8/]
        MY8 --> End([Кінець])
      
```

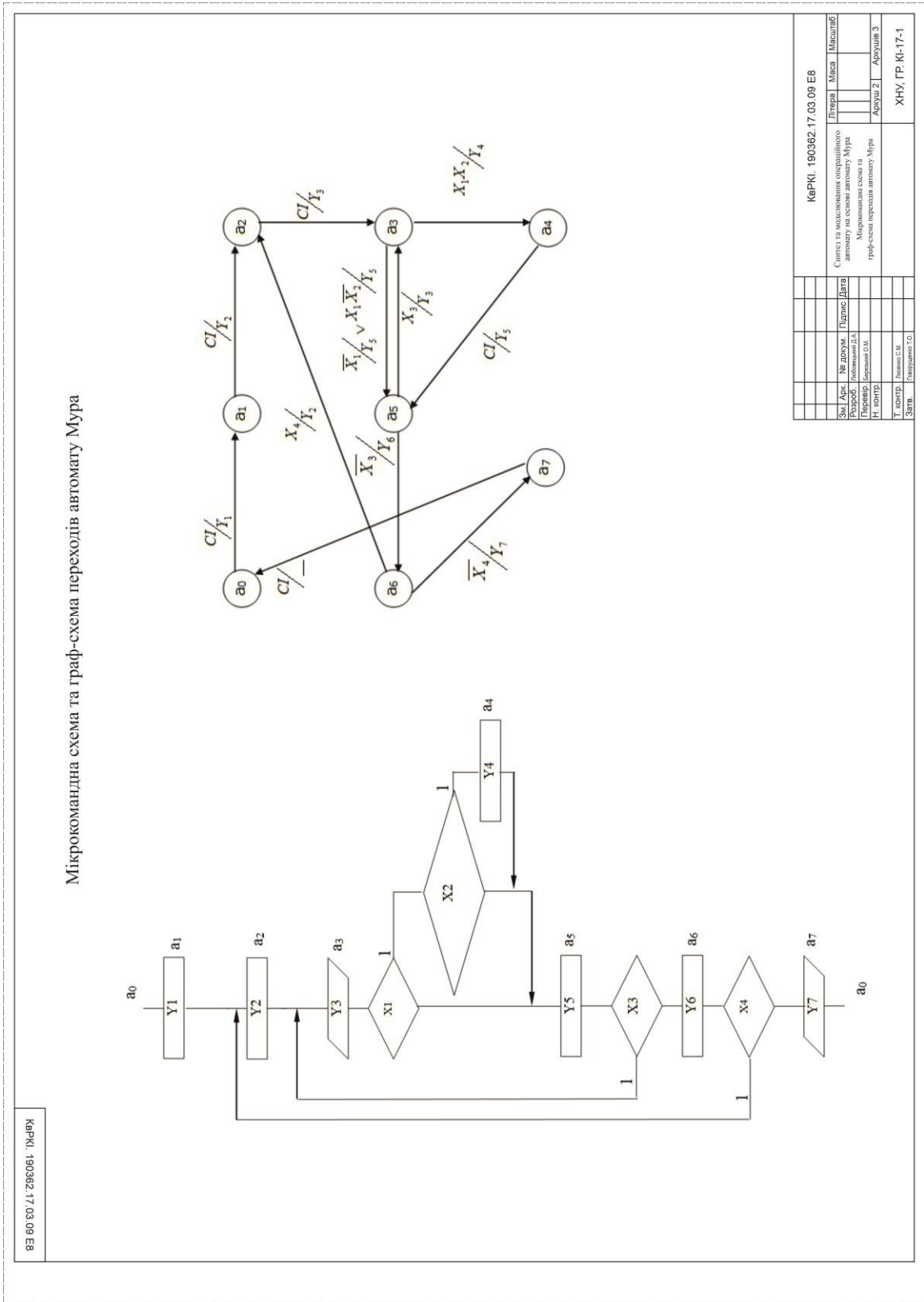
```

      graph TD
        Start([Початок]) --> K[k=0]
        K --> J[j=1]
        J --> I[i=1]
        I --> Aij[/A[i,j]/]
        Aij --> X1{A[i,j] % 2}
        X1 -- так --> X2{A[i,j] <= 3 and A[i,j] <= j}
        X2 -- так --> Kplus[k=k+1]
        Kplus --> X1
        X1 --> Iplus[i=i+1]
        Iplus --> X3{i < n}
        X3 -- так --> Jplus[j=j+1]
        Jplus --> X4{j < m}
        X4 -- так --> Aij
        X4 --> K([k])
        K --> End([Кінець])
      
```

| КЕРКІ. 190362.17.03.09 Е8 | | | |
|--|-------------|--------|---------------------|
| Літера | Маса | Місця | Місця |
| | | | |
| Синтез та моделювання спеціального автомату на основі алгоритму Мура | | | |
| Знак | № докум. | Підпис | Дата |
| Розроб. | Робочий | Д.А. | |
| Г. констр. | Варшавський | О.М. | |
| І. констр. | | | |
| Затв. | Делюк | С.М. | |
| | Государьова | Г.О. | |
| Листів сторінок | | | Аркушів з |
| | | | Аркуш 1 Аркушів 3 |
| | | | ХНУ, ГР. КІ-17-1 |

Додаток Б (обов'язковий)

Копія креслення «Мікрокомандна схема та граф-схема переходів автомату Мура»

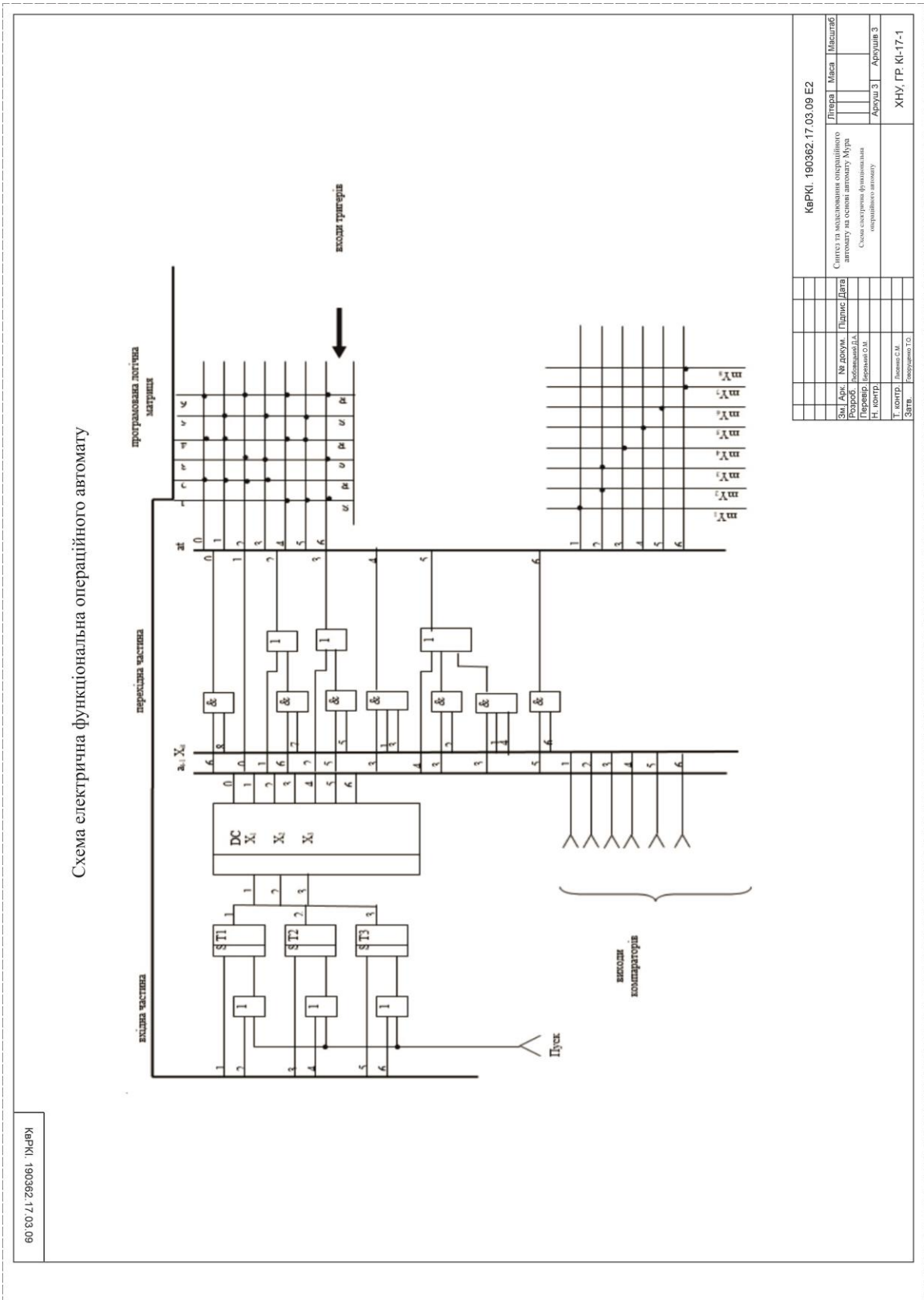


КерПКі, 190362, 17.03.09.Е8

| КерПКі, 190362, 17.03.09.Е8 | | | |
|-----------------------------|-----------------|---------|------------------|
| Лист | Листів | Масштаб | |
| 1 | 1 | | |
| Знак | № докум. | Підпис | Дата |
| Розроб. | Позивання П.А. | | |
| Перевір. | Березина О.М. | | |
| Т. констр. | Лавров С.М. | | |
| Заб.тв. | Посередине Т.О. | | |
| | | | ХНУ, ГР. КІ-17-1 |

Додаток В (обов'язковий)

Копія креслення «Схема електрична функціональна операційного автомату»



Ім'я користувача:
Кафедра КІ

ID перевірки:
1008133488

Дата перевірки:
02.06.2021 07:54:24 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
02.06.2021 07:56:48 EEST

ID користувача:
100005591

Назва документа: Любовецький_2_Синтез та моделювання операційного автомату на основі автомату Мура

Кількість сторінок: 68 Кількість слів: 10014 Кількість символів: 75898 Розмір файлу: 1.93 MB ID файлу: 1008214668

7.39% Схожість

Найбільша схожість: 1.53% з джерелом з Бібліотеки (ID файлу: 1008141803)

6.62% Джерела з Інтернету 601 Сторінка 70

2.11% Джерела з Бібліотеки 70 Сторінка 73

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

66.9% Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 8 слів та 0%)

Немає вилучених Інтернет-джерел

66.9% Вилученого тексту з Бібліотеки 1 Сторінка 73

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 8

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 2.0%

Словари проверки: en_US, ru_RU, ua_UA. Ошибок в документах: 6%

| | | | | |
|--|----------|---------|-------------------------------------|----------|
| ID: 91746 Название: Синтез та моделювання операційного автомату на основі автомату Мура Добавлено в БД: 2021-06-01 Авторы: Д. А. Любовецький Руководители: О. М. Березький Консультанты: Опоненты: | Документ | | Суммарное совпадение по Базе Данных | |
| | Символы | Лексемы | Символы | Лексемы |
| | 67243 | 521 | 5018 (7%) | 53 (10%) |

Источник плагиата

| ID | Описание | Наличие плагиата в документе | |
|----|----------|------------------------------|---------|
| | | Символы | Лексемы |

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Любовецький Денис Андрійович

Тема: Синтез та моделювання операційного автомату на основі автомату Мура

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 64

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є синтез та моделювання операційного автомату на основі автомату Мура
2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.
3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі кваліфікаційної роботи проведено дослідження предметної області (проаналізовано теорію операційних та керуючих автоматів, автоматів Мілі та Мура, а також розроблення і опис схем електричних функціональної і принципової) та виконано постановку задачі дослідження. В другому розділі кваліфікаційної роботи проведено моделювання та проектування операційного автомату на основі автомату Мура, а саме: виконано формалізований опис операційного автомату; розроблено змістовну схему алгоритму; закодовано верхівки змістовної схеми алгоритму; розроблено закодовану мікроопераційну схему алгоритму; закодовано мікрокоманди; розроблено закодовану мікрокомандну схему алгоритму; побудовано основну таблицю автомату Мура; спроектовано граф-схему переходів автомату Мура; складено систему рівнянь переходів; визначено необхідну кількість елементів пам'яті (тригерів) для реалізації автомату Мура; закодовано внутрішні стани автомату; розроблено структурну схему операційного автомату; побудовано схему операційного автомату. В третьому розділі кваліфікаційної роботи виконано апаратну реалізацію операційного автомату на основі автомату Мура, а саме:

реалізовано схему електричну функціональну операційного автомату на основі автомату Мура; змодельовано схему в середовищі Quartus II.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: недостатня увага моделюванню схеми автомату в середовищі Quartus II.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.


8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Федула Микола Васильович, кандидат технічних наук, доцент, доцент кафедри
АКІТІТК

“ 2 ” червня 2021 р.



(підпис)

Завідувачу кафедри КІСП
д-ру техн.наук, проф. Говорущенко Т. О.

Любовецького Д.А.

ІІБ здобувача вищої освіти

ФПКТС, 4 курсу, групи КІ-17-3

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність плагіату ознайомлений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unichack та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

31.05.2021

дата



**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА СИСТЕМОГО
ПРОГРАМУВАННЯ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Синтез та моделювання операційного автомату на основі автомату Мура

Автор: Любовецький Денис Андрійович

Спеціальність: 123 – Компютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Березький Олег Миколайович, д.т.н, професор

Після аналізу звіту подібності зроблено такий висновок:

| № | Висновок | Позначка про відповідність |
|---|---|----------------------------|
| 1 | Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту. | відповідає |
| 2 | Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи | |
| 3 | Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат. | |
| 4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту. | |

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення не описують безпосередньо авторське дослідження і не стосуються отриманих в роботі результатів;
- 2) усі запозичення фрагментарні, або мають належним чином оформлені посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами;
- 4) в якості запозичень в окремих місцях системою зафіксовано таблиці істинності деяких елементів, які є вхідними даними до великої кількості задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
- 5) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів із україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 7.39% і адресується до 671 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІСП

О. М. Березький

С. М. Лисенко

Т. О. Говорущенко