

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Синтез та моделювання операційного автомату на основі автомату Мілі
Назва теми

КВРКІ 180114.18.01.04 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-18-1



Підпис

М. С. Питляк
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 інформаційних
систем

Підпис

Т.О. Говоруценко
Ініціали, прізвище

« 26 » травня 2022 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

“ 10 ” 01 2022 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Питляку Максиму Сергійовичу

Прізвище, ім'я, по батькові студента

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

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

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

Затверджена наказом ректора університету від 01.03.2022 р. № 18

2. Строк подання студентом проекту (роботи) на кафедру 06.06.2022 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 Мілі

Апаратна реалізація операційного автомату на основі автомату Мілі



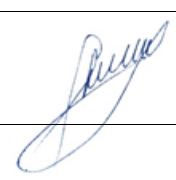
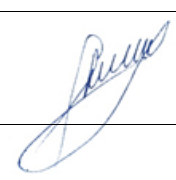
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Логічні схеми алгоритмів

Мікрокомандна схема та граф-схема переходів автомату Мілі

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 КПС		
Антиплагіат	Нічепорук А.О., доцент кафедри КПС		

7. Дата видачі завдання « 10 » 01 2022 р.

КАЛЕНДАРНИЙ ПЛАН

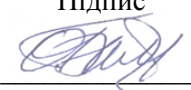
№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2022	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2022	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2022	виконано
4	Робота над розділом 2 – моделювання та проектування операційного автомату на основі автомату Мілі	01.04.2022	виконано
5	Робота над розділом 3 – реалізація та моделювання роботи операційного автомату на основі автомату Мілі	29.04.2022	виконано
6	Оформлення пояснювальної записки згідно вимог	20.05.2022	виконано
7	Попередній захист ВКР	24.05.2022	виконано
8	Захист ВКР на засіданні ЕК	Червень 2022 року	

Студент


Підпис

М. С. Питляк
Ініціали, прізвище

Керівник роботи


Підпис

О. М. Березький
Ініціали, прізвище

АНОТАЦІЯ

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

Автор роботи: Питляк Максим Сергійович.

Керівник роботи: Березький Олег Миколайович.

Пояснювальна записка: 65 с., 24 рис., 9 табл., 3 дод., 43 джерела.

Графічна частина: 8 презентаційних слайдів.

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

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

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

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

Практичне значення має змодельований, спроектований та реалізований операційний автомат на основі автомату Мілі, що використовується для обчислення кількості непарних елементів у двовимірному масиві $A[n, m]$, які належать інтервалу $[-5; 15]$.







Підпис студента

26.05.2022

Дата

ЗМІСТ

ВСТУП.....	3
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ..	5
1.1 Принципи дії операційного та керуючого автоматів	5
1.2 Особливості автоматів Мілі та Мура	13
1.3 Функціональна та принципова електричні схеми	17
1.4 Висновки. Постановка задачі.....	23
2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ОПЕРАЦІЙНОГО АВТОМАТУ НА ОСНОВІ АВТОМАТУ МІЛІ	25
2.1 Моделювання операційного автомату	25
2.2 Проєктування операційного автомату	32
2.3 Висновки	35
3 РЕАЛІЗАЦІЯ ТА МОДЕЛЮВАННЯ РОБОТИ ОПЕРАЦІЙНОГО АВТОМАТУ НА ОСНОВІ АВТОМАТУ МІЛІ	36
3.1 Застосування схеми Уїлкса-Стрінжера для реалізації операційного автомату на основі автомату Мілі	36
3.2 Реалізація операційного автомату на основі автомату Мілі.....	42
3.3 Моделювання роботи операційного автомату на основі автомату Мілі в середовищі Quartus II	48
3.4 Висновки	55
ВИСНОВКИ.....	56
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	59
ДОДАТОК А Копія креслення «Логічні схеми алгоритмів».....	63
ДОДАТОК Б Копія креслення «Мікрокомандна схема та граф-схема переходів автомату Мілі».....	64
ДОДАТОК В Копія креслення «Схема електрична функціональна операційного автомату».....	65

					КВРКІ 180114.18.01.04 ПЗ			
Зм.	Арк.	№докум.	Підпис	Дата	Синтез та моделювання операційного автомату на основі автомату Мілі. Пояснювальна записка	Літера	Арквщ	Арквщів
Виконав	Питляк М.С.					у	2	62
Перевір.	Березький О.М.					ХНУ КІ-18-1		
Н.контр.	Лисенко С.М.							
Затвер.	Говорущенко Т.О.							

ВСТУП

Важливість, відповідальність та складність задач, що покладаються на сучасні цифрові пристрої, потребують високої відмовостійкості, надійності, продуктивності та гарантоздатності цих пристроїв.

Кінцеві автомати наразі активно використовуються в комп'ютерних іграх, системах стиснення інформації, а також при реалізації мережевих протоколів. Кінцеві автомати можуть бути застосовані при розв'язанні задач, де потрібна складна логіка звернення та висока надійність, які програмісту важко реалізувати вручну.

Зручною абстракцією для таких задач є кінцеві автомати, особливо автомати Мілі та автомати Мура. Завдяки їх ґрунтуванню на постулатах булевої алгебри та теорії графів, завдяки чіткості їх графічного подання та завдяки детермінованості їх поведінки така абстракція має серйозні переваги з точки зору поставленої задачі.

Будь-який операційний пристрій має у своєму складі операційний автомат та керуючий автомат.

Операційний автомат – це пристрій, який безпосередньо виконує операції над інформацією з врахуванням керуючих сигналів і надає результати цих операцій. Його складовими вузлами є суматори, регістри, мультиплексори, дешифратори, арифметико-логічний пристрій та інші функціональні блоки, які беруть участь у процесі виконання команд (запис в регістр, зсув коду ліворуч або праворуч, дешифрування двійкового коду, інвертування вмісту регістра, додавання двох чисел, тощо).

Проектований операційний автомат повинен видавати певну продуктивність, тому синтезована структура не повинна вносити обмеження на сумісність мікрооперацій, тобто повинно бути можливим одночасне виконання мікрооперацій, що є функціонально сумісними.

					КВРКІ 180114.18.01.04 ПЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

Автомати Мілі (Mealy) – цифрові автомати, в яких існує залежність вихідного сигналу автомата від вхідного сигналу та його поточного внутрішнього стану.

Автомати Мура (Moore) – цифрові автомати, в яких існує залежність вихідного сигналу автомата тільки від його поточного внутрішнього стану.

Метою кваліфікаційної роботи є синтез та моделювання операційного автомату на основі автомату Мілі.

Поставлена мета досягається розв'язанням такої основної задачі: побудувати операційний автомат, що виконує обчислення кількості непарних елементів у двовимірному масиві $A[n, m]$, які належать інтервалу $[-5; 15]$. Слід використовувати схему Уілкса-Стрінжера для реалізації мікропроцесорного автомату як автомату Мілі. Слід використовувати логічні елементи, тригери для реалізації та синтезу функціональної схеми автомату.

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

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

Для досягнення поставленої мети використовуються такі методи дослідження, як методи синтезу, аналізу та моделювання процесів, принципи системного аналізу, теоретико-множинні підходи.

Практичне значення має спроектований, змодельований та реалізований операційний автомат на основі автомату Мілі, який використовується для обчислення кількості непарних елементів у двовимірному масиві $A[n, m]$, які належать інтервалу $[-5; 15]$.

					КВРКІ 180114.18.01.04 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Принципи дії операційного та керуючого автоматів

Цифрові пристрої складаються з двох частин: виконавчої та керуючої.

Пристрої обробки цифрової інформації, як-от: процесор, канал вводу-виводу, тощо, складається з наступних двох основних блоків – керуючого автомату (КА) та операційного автомату (ОА) [1-3].

Керуючий автомат – блок, пристрій, компонент апаратного забезпечення комп'ютерів. Являє собою кінцевий дискретний автомат. Структурно керуючий автомат складається з: дешифратора команд (операцій), регістру команд, вузла формування (обчислення) поточної виконавчої адреси, лічильника команд.

Пристрій керування сучасних процесорів, який є різновидом керуючого автомату, зазвичай реалізується у вигляді мікропрограмного автомата і в цьому випадку включає постійний запам'ятовуючий пристрій мікрокоманд.

Керуючий пристрій призначено для формування сигналів управління всім блокам машини [4, 5].

У архітектурі фон Неймана є невід'ємною частиною центрального процесора [6].

При виконанні програми керуючий пристрій на вході отримує чергову інструкцію або команду програми, яка надходить до регістру команд. Структурно регістр команд складається з двох частин: в одній частині зберігається код операції (КОП), в іншій частині розташовані адреси операндів, що беруть участь в управлінні. Код операції надходить у дешифратор операції, який запускає послідовність зміни станів кінцевого автомата керуючого пристрою, в сучасних машинах – запускає деяку мікропрограму, що зберігається в постійному запам'ятовуючому пристрої (ПЗП) мікропрограм. Суть роботи керуючого пристрою полягає у послідовному виставленні комбінацій сигналів керування на лініях керування [7-10].

					КВРКІ 180114.18.01.04 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

При переході з одного внутрішнього стану в наступний стан кінцевого автомата керуючим пристроєм також враховується стан регістру ознак результату виконання попередньої команди (регістра прапорців), стан керуючих входів (наприклад, сигналів готовності периферійних пристроїв, ліній апаратних переривань) і на вихід керуючий пристрій видає послідовності імпульсів управління (міжреєстровими пересиланнями, адресацією та сигналами пересилань даних практично між усіма блоками комп'ютера – арифметико-логічним пристроєм (АЛП), підсистемою введення/виведення, оперативною пам'яттю та інші).

Майже завжди для виконання команди/інструкції керуючий пристрій видає послідовність з кількох керуючих імпульсів або їх комбінацій у послідовності, що визначається кодом команди/інструкції. Після завершення видачі всіх імпульсів керування для виконання поточної команди керуючий пристрій зчитує код наступної команди і т.і.

Керуючий пристрій зазвичай синхронізується зовнішніми тактовими імпульсами, що формується генератором тактових імпульсів (ГТІ), тобто перехід зі стану в стан кінцевого автомата керуючого пристрою відбувається на кожному тактовому імпульсі, в деяких керуючих пристроях для збільшення швидкодії управляючі імпульси формуються як на фронті імпульсу, що наростає, так і на падаючому фронті імпульсу [7-10].

У ранніх комп'ютерах керуючий пристрій проектувався та виконувався у вигляді «жорсткої логіки» - комбінаційних логічних схем, тригерів, регістрів, дешифраторів.

У 1951 році М. В. Вілксом було запропоновано проектувати керуючий пристрій у вигляді мікропрограмного автомата, тобто формувати послідовності керуючих імпульсів за допомогою мікропрограмного автомата, керованого мікропрограмами, що складаються з мікроінструкцій, що зберігаються в постійній або перепрограмованій мікропрограмній пам'яті процесора. Такий підхід суттєво знижує трудовитрати на проектування кінцевого автомата керуючого пристрою та додатково дозволяє відносно легко його модифікувати. З розвитком

					КВРКІ 180114.18.01.04 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

напівпровідникової техніки та появою на ринку великої кількості програмованих контролерів під час створення керуючого пристрою цей підхід став домінуючим [7-10].

Проте в останні роки для деяких автоматизованих систем управління намітилася тенденція повернення (на новому технологічному рівні) до використання систем, заснованих на жорсткій логіці, зокрема тому, що вони мають високий рівень захищеності від комп'ютерних атак [4, 5].

Функціональну схему керуючого пристрою представлено на рис. 1.1.

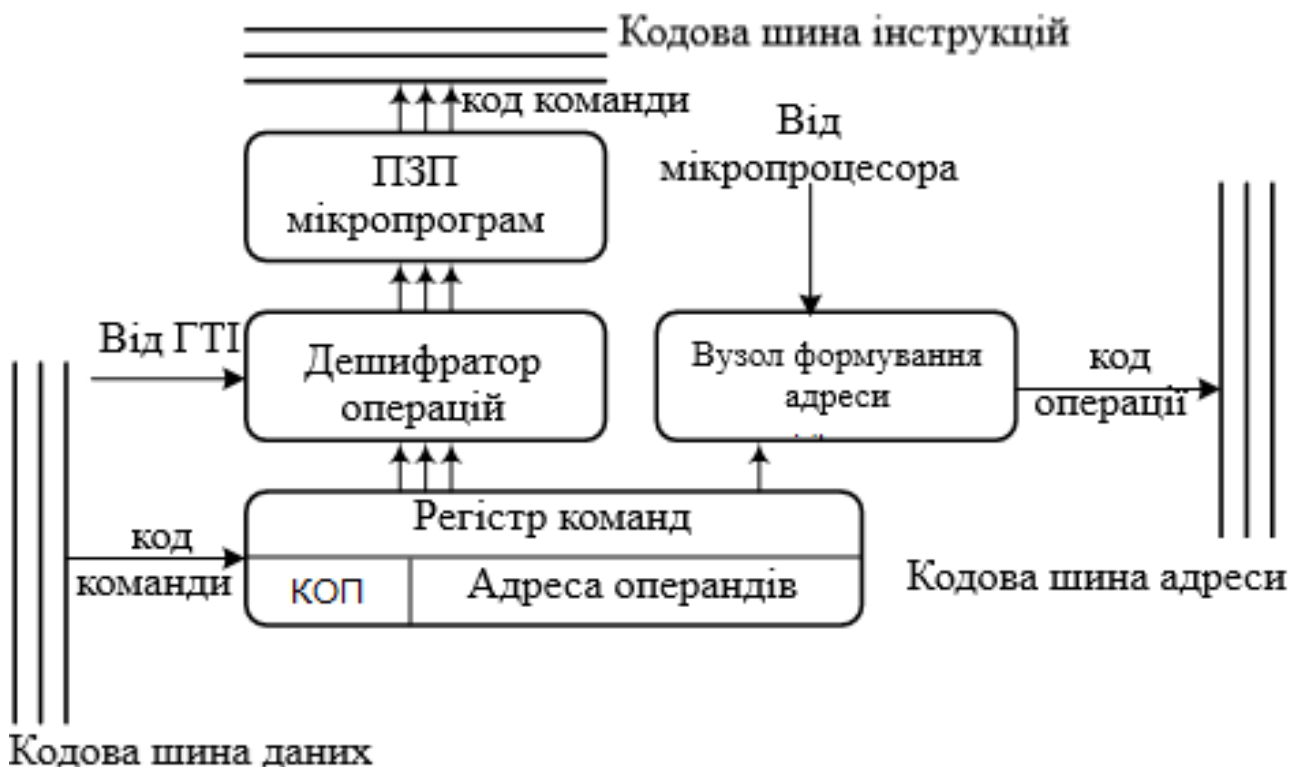


Рисунок 1.1 – Функціональна схема керуючого пристрою

Операційний автомат (ОА) – набір функціональних елементів, таких як арифметико-логічний пристрій (АЛП) та логічні операції [1-5], які виконують обробку даних. Операційні пристрої, такі як мікропроцесори, складаються з операційного автомата та керуючого автомата [1-10], більшу частину яких займає керуючий автомат, що регулює передачу даних між операційним автоматом та пам'яттю.

Зв'язок між операційним та керуючим автоматами представлено на рис. 1.2.

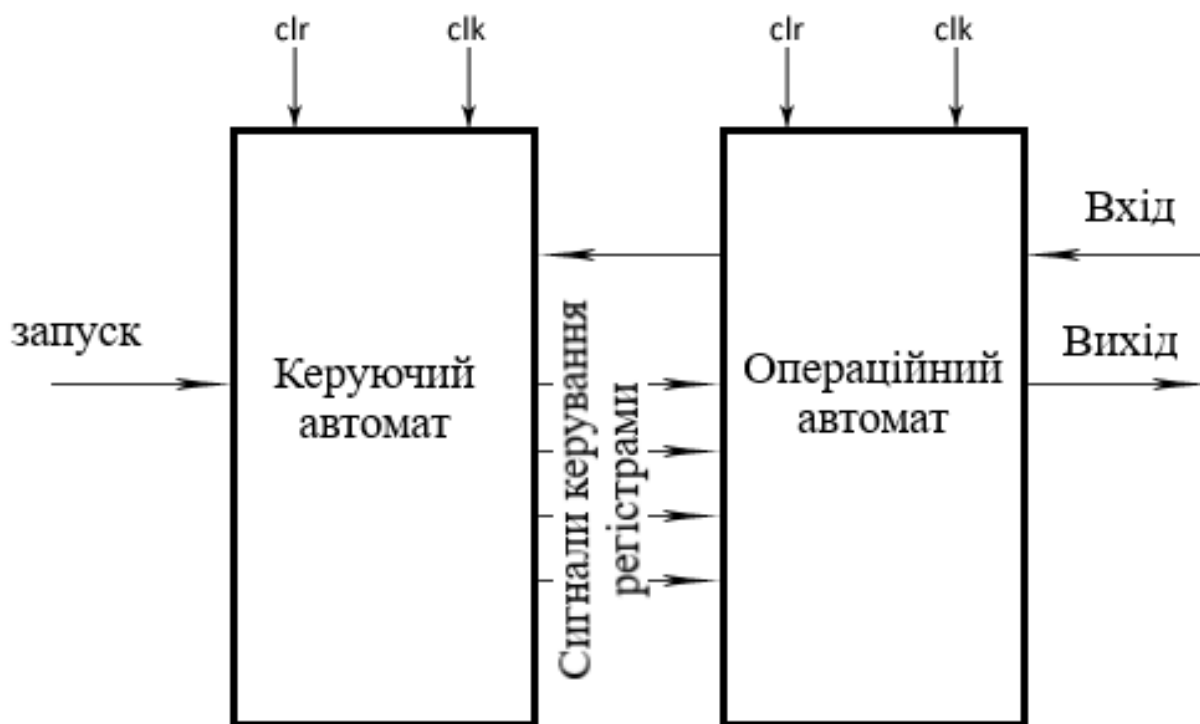


Рисунок 1.2 – Зв'язок між операційним та керуючим автоматами

Академік Глушков В.М. показав, що в будь-якому пристрої обробки цифрової інформації керуючий автомат (КА) та операційний автомат (ОА) взаємодіють наступним чином – рис. 1.3 [2, 3].



Рисунок 1.3 – Взаємодія операційного та керуючого автоматів

Операція g , зовнішня щодо КА, визначає операцію, яка повинна виконуватись. Сигнали g_1, \dots, g_h кодують операцію, сигнали x_1, \dots, x_L ОА впливають на порядок формування керуючих сигналів Y .

Множиною керуючих сигналів $Y = \{y_1, \dots, y_M\}$, де кожен елемент визначає конкретну мікрооперацію, задаються мікрооперації, що виконуються операційним автоматом. Множиною інформаційних сигналів $X = \{x_1, \dots, x_L\}$, де кожен елемент визначає деяку логічну умову, задаються значення логічних умов операційного автомату.

Операційний автомат є структурою, організованою для виконання дій над словами інформації. Операційний автомат – це виконавча частина пристрою. Роботою пристрою керує керуючий автомат, що формує керуючі сигнали, а також їх послідовності. Операційний автомат призначений для зберігання слів інформації, виконання набору мікрооперацій і обчислення значень логічних умов, тобто дії над інформацією [1-3].

Керуючий автомат з врахуванням обчислених операційним автоматом значень логічних умов обирає керуючі сигнали з мікропрограми. Отже, керуючий автомат визначає відповідно до використовуваних алгоритму та мікропрограми порядок виконання операцій в операційному автоматі.

Операційний автомат визначається наступними множинами [1-10]:

- 1) операнди автомату – множина вхідних слів $D = \{d_1, \dots, d_F\}$;
- 2) результати операцій – множина вихідних слів $R = \{r_1, \dots, r_T\}$;
- 3) інформація під час виконання операцій – множина внутрішніх слів $S = \{s_1, \dots, s_M\}$, причому $D \subseteq S$, $R \subseteq S$;
- 4) перетворення $S = \varphi_n(s)$ слів інформації (φ_n – обчислювана функція) – множина мікрооперацій $Y = \{y_n\}$;
- 5) множина логічних умов $X = \{x_j\}$, де $x_j = \psi_j(s_j)$, а ψ_j – логічна функція.

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

Час не є аргументом функції операційного автомату [1-10].

					КВРКІ 180114.18.01.04 ПЗ	Арк. 9
Зм.	Арк.	№ докум.	Підпис	Дата		

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

- 1) реєстр команд – містить поточну інструкцію, яка буде виконуватись;
- 2) програмний лічильник (PC) – містить адресу наступної інструкції для даних;
- 3) адреса реєстру пам'яті (MAR) – реєстр, який зберігає або адресу пам'яті, з якої будуть отримуватись дані на центральний процесор, або адресу, за якою дані будуть відправлені і збережені;
- 4) реєстр даних пам'яті (MDR) – реєстр блоку управління комп'ютера, що містить дані, які будуть зберігатись у пам'яті ЕОМ (наприклад, у оперативному запам'ятовуючому пристрої), або дані з пам'яті ЕОМ;
- 5) два реєстри, які містяться у процесорі, що полегшують взаємодію процесора та пам'яті.

Структура операційного автомату, отримана шляхом доповнення кожного елемента функції (слова, мікрооперації, логічної умови) відповідними елементами структурного базису (шинами, реєстрами, комбінаційними схемами), є основною для синтезу інших типів ОА і називається канонічною. Канонічну структуру операційного автомату представлено на рис. 1.4 [3].

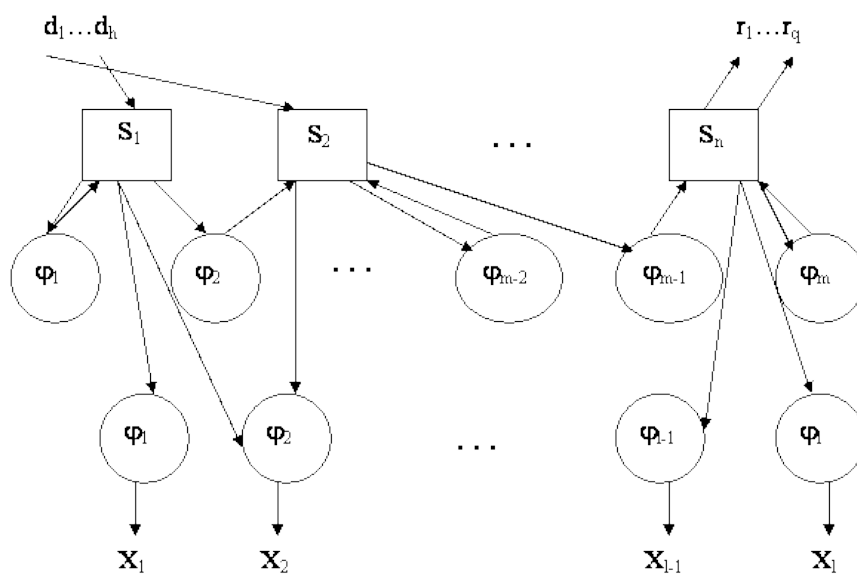


Рисунок 1.4 – Канонічна структура операційного автомату

Регістри S складають пам'ять автомату, в якій зберігаються початкові, проміжні та кінцеві результати. Початкові дані завантажуються в реєстр через входи d . Кінцеві результати виводяться через виходи $г$.

Отже, для розробки операційного автомату потрібні:

- 1) вхідні та вихідні слова ОА;
- 2) множини операцій, що будуть виконуватися над словами.

Тоді процес розробки ОА слід починати з того, щоб визначити формати вхідних і вихідних слів, а далі перейти до розроблення алгоритмів щодо дій над словами (інверсія, зсуви, додавання, копіювання, тощо). Після цього слід розробити структуру операційного автомату, а саме: визначити набір елементів, з яких вона складається, і з'ясувати зв'язки між такими елементами.

Операційний автомат складається з операційних і логічних елементів. Структура ОА формується на основі розроблених алгоритмів для забезпечення всіх дій, передбачених операційними вершинами.

Дії в ОА виконуються під керуванням мікрооперацій, тому при розробці операційного автомату варто мати повний список мікрооперацій, що забезпечить виконання всіх передбачених алгоритмами перетворень слів в розробленій структурі.

З врахуванням логічних умов, що обчислюються операційним автоматом, формується послідовність мікрооперацій керуючого автомату. При розробленні операційного автомату має бути сформований список логічних умов, що визначається умовними вершинами, і в структурі ОА мають бути передбачені спеціальні елементи для формування таких логічних умов.

Таким чином, процес розроблення операційного автомату складається з наступних кроків: 1) підбір форматів слів – вхідних та вихідних даних; 2) розробка виконуваних операцій; 3) розробка структури операційного автомату – вибір елементів і організація зв'язків; 4) визначення множини мікрооперацій, виконуваних в операційному автоматі.

					КВРКІ 180114.18.01.04 ПЗ	Арк. 11
Зм.	Арк.	№ докум.	Підпис	Дата		

Тоді, операційний автомат є сукупністю комбінаційних схем, які виконують мікрооперації, а також регістрів (або оперативних запам'ятовуючих пристроїв) для зберігання даних.

Алгоритм роботи операційного автомату [9, 10]:

1) в залежності від сигналів, що надходять від керуючого автомату, операційний автомат виконує різні мікрооперації – операції введення і виведення, дії над регістрами;

2) в результаті виконання мікрооперацій ОА формує мікроумови, згідно із якими керуючий автомат переходить у той чи інший стан.

Операторна схема представляється у вигляді граф-схеми алгоритму (ГСА), яка відображає перебіг в часі обчислювального процесу, встановлює порядок виконання мікрооперацій та порядок перевірки логічних умов.

Мікрооперації в ГСА зображаються умовними графічними зображеннями – вершинами наступних типів:

- 1) «початок» - початок мікропрограми;
- 2) «кінець» - кінець мікропрограми;
- 3) операторна вершина – одна мікрокоманда (сукупність мікрооперацій з одночасним виконанням);
- 4) умовна вершина – логічний вираз, в залежності від істинності якого обирається гілка подальшого виконання мікропрограми;
- 5) вершина очікування – умовна вершина, вихід з якої настає лише при істинності умови.

Схема операційного автомату містить:

1) вхідну частину – тригери, логічні елементи для прийняття та опрацювання вхідних сигналів, дешифратор та дві шини (для передачі сигналів з дешифратора, а також для передачі сигналів з компаратора);

2) перехідну частину – проектується на базі системи рівнянь переходів, виконує перетворення сигналу за один такт часу, результат зміни сигналу надходить на шину, після чого передається до програмованої логічної матриці;

3) вихідну частину – проєктується на базі системи рівнянь виходів, виконує процес, аналогічно до перехідної частини, але сигнали надходять на вихідну шину, з якої передаються до вихідної матриці.

Таким чином, операційний автомат – це пристрій, який безпосередньо виконує операції над інформацією з врахуванням керуючих сигналів і надає результати цих операцій. Його складовими вузлами є суматори, регістри, мультиплексори, дешифратори, арифметико-логічний пристрій та інші функціональні блоки, які беруть участь у процесі виконання команд (запис в регістр, зсув коду ліворуч або праворуч, дешифрування двійкового коду, інвертування вмісту регістра, додавання двох чисел, тощо).

Проєктований операційний автомат повинен видавати певну продуктивність, тому синтезована структура не повинна вносити обмеження на сумісність мікрооперацій, тобто повинно бути можливим одночасне виконання мікрооперацій, що є функціонально сумісними.

1.2 Особливості автоматів Мілі та Мура

Автомат – дискретний перетворювач інформації, який приймає різні стани, переходить з одного стану в інший під впливом різних вхідних сигналів і видає вихідні сигнали [11].

Цифровий автомат (ЦА) – пристрій з певним набором внутрішніх станів, в які він може переходити під впливом різних команд закладеної програми. Перехід автомату з одного стану в інший здійснюється у визначений програмою момент часу [12].

Цифровий автомат – пристрій, що приймає, зберігає і перетворює інформацію, використовуючи певний алгоритм. До автоматів належать і реальні пристрої (ЕОМ), і абстрактні системи (математичні моделі) [13].

Автомат повинен реалізовувати деякі функції, задані розробником. Він може бути простим суматором, може реалізовувати якусь мікрокоманду

					КВРКІ 180114.18.01.04 ПЗ	Арк. 13
Зм.	Арк.	№ докум.	Підпис	Дата		

процесора, вибрати слова з оперативної пам'яті або займатися синтаксичним аналізом виразу.

У загальному вигляді, не вдаючись до подробиць, абстрактний автомат можна представити так – рис. 1.5 [14].

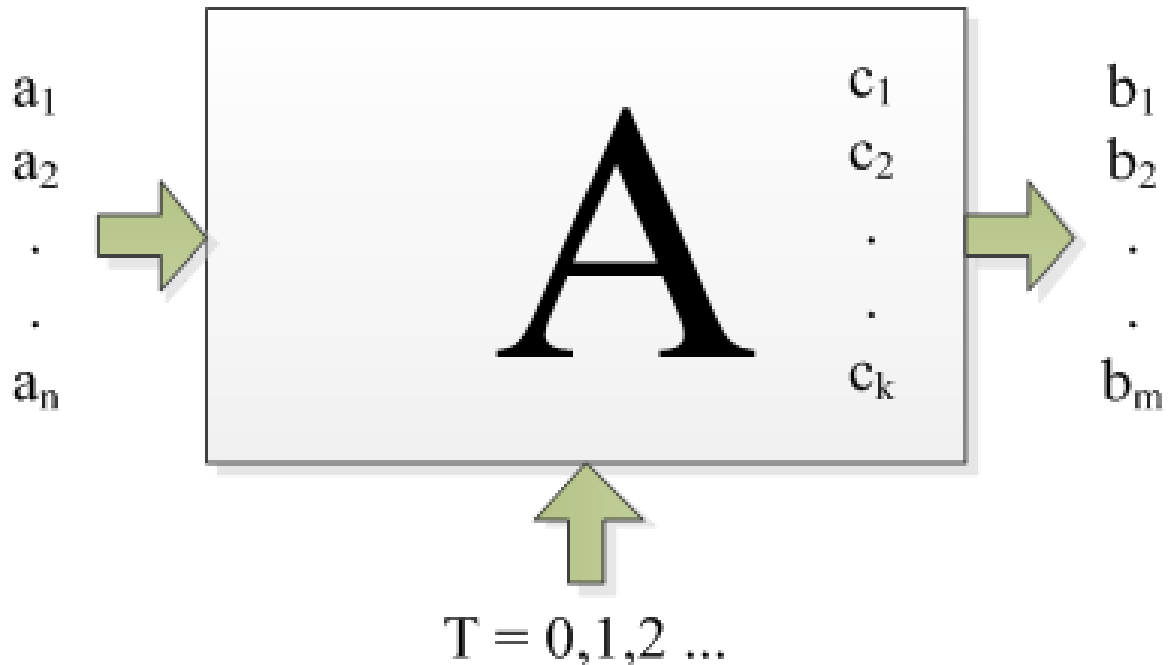


Рисунок 1.5 – Структурна схема абстрактного автомату

Тоді математична модель цифрового автомату представляє собою кортеж з 6 компонентів: $S=(A, B, C, \delta, \lambda)$, де [15-17]:

- 1) A – множина значень на фізичних входах автомата (вхідний алфавіт); на вході буде якась послідовність високих та низьких рівнів напруги, які кодуватимуть логічні нулі та одиниці;
- 2) B – множина значень на фізичних виходах автомата (вихідний алфавіт);
- 3) C – множина внутрішніх станів автомата (внутрішній алфавіт); стан автомату якраз і є певною пам'яттю, передісторією;
- 4) $\delta: C \cdot A \rightarrow C$ – функція переходів, що реалізує відображення $D_\delta \subseteq C \cdot A$ в C ;
- 5) $\lambda: C \cdot A \rightarrow B$ – функція виходів, що реалізує відображення $D_\lambda \subseteq C \cdot A$ у B .

Функція переходів відображає залежність наступного стану автомата від вхідного сигналу та поточного внутрішнього стану.

Функція виходів відображає залежність вихідного сигналу автомата від вхідного сигналу та його поточного внутрішнього стану.

Саме такі залежності дають можливість виділити два класи цифрових автоматів [18-20].

Автомати Мілі (Mealy) – цифрові автомати, в яких існує залежність вихідного сигналу автомата від вхідного сигналу та його поточного внутрішнього стану [21, 22].

Тоді закони функціонування автомату Мілі:

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

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

Автомати Мура (Moore) – цифрові автомати, в яких існує залежність вихідного сигналу автомата тільки від його поточного внутрішнього стану [23, 24].

Тоді закони функціонування автомату Мура:

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

$$b(t) = \lambda(c(t)), t = 0, 1, 2, \dots$$

Множини А, В, С задаються набором своїх елементів. Функції переходів та виходів δ і λ задаються табличним (для автомату Мілі таблиця переходів задає функцію δ , таблиця виходів – функцію λ ; автомат Мура задається лише відміченою таблицею переходів) і графічним (у вигляді орієнтованого графа, вершини якого відтворюють стани автомату, а дуги – переходи зі стану в стан із зазначенням вхідного сигналу, що викликає даний перехід) способами.

Для графу автомата Мура вихідний сигнал(и), який(і) формується внаслідок певного переходу записується поруч з вершиною, позначеною станом, у якому він(вони) формується; для графу автомата Мілі вихідний сигнал(и), який(і) формується внаслідок певного переходу, записується наприкінці відповідної дуги [25-27].

На рис.1.6 наведено приклад графу автомата Мура. На рис.1.7 наведено приклад графу автомата Мілі.

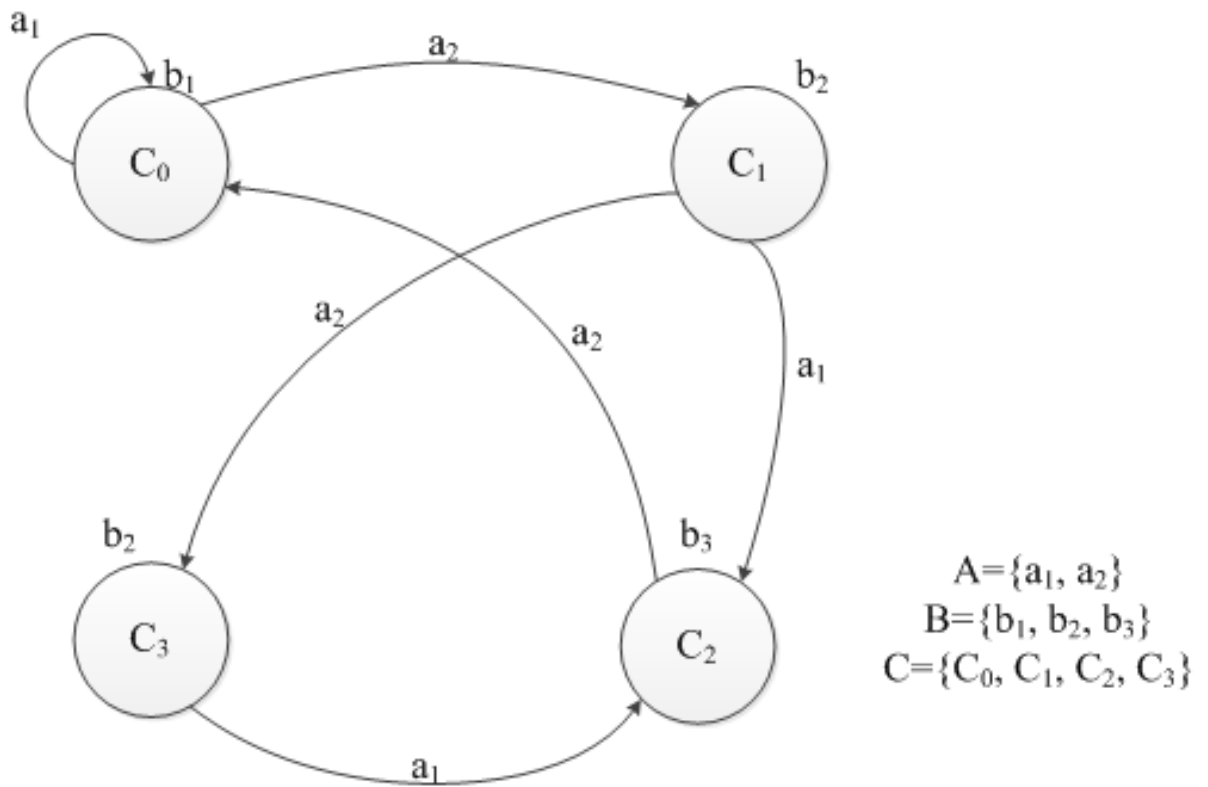


Рисунок 1.6 – Приклад графу автомата Мура

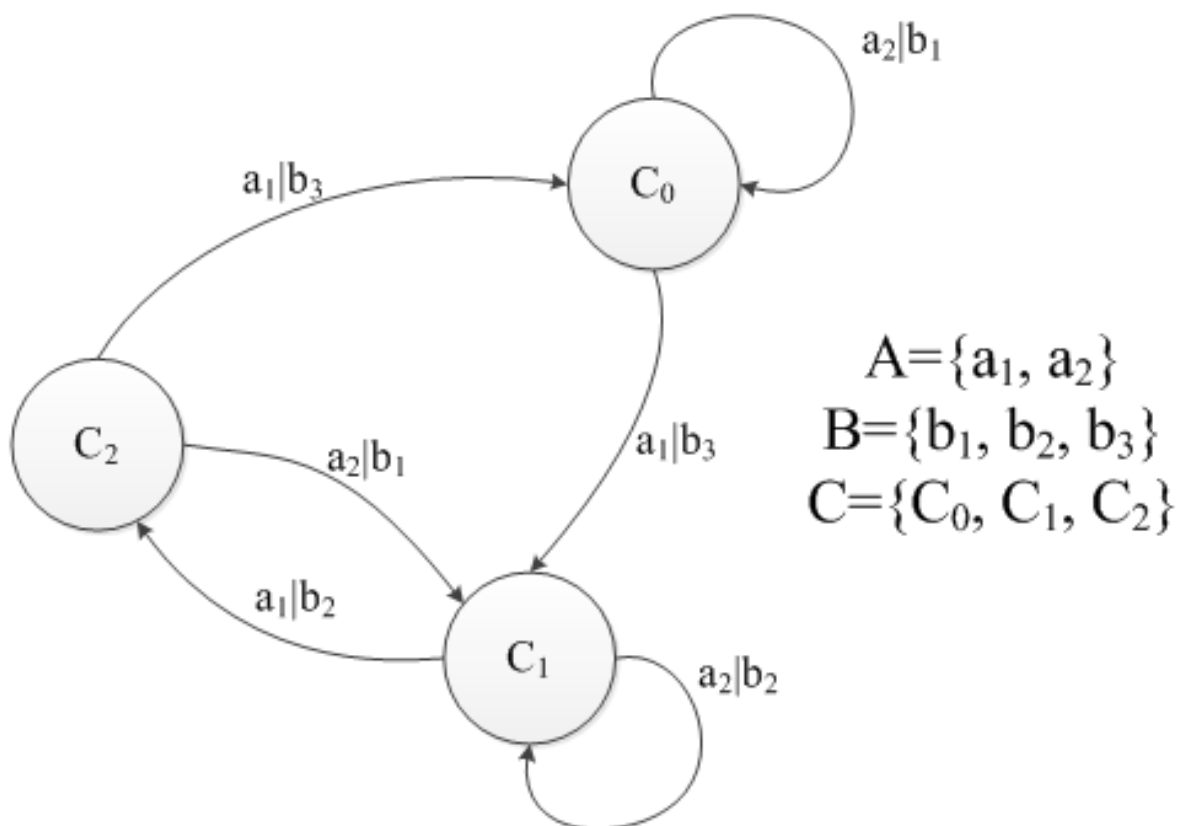


Рисунок 1.7 – Приклад графу автомата Мілі

Автомати Мура і Мілі застосовуються при проектуванні цифрових пристроїв на базі програмованих логічних інтегральних схем (ПЛІС).

Основна перевага автомату Мілі – це можливість реакції автомату на протязі поточного такту, що ґрунтується на залежності поточного вихідного сигналу від поточного вхідного сигналу.

Основна перевага автомату Мура – наявність мінімальної затримки, простота опису мовами HDL, відсутність наскрізного поширення сигналу від входу до виходу автомата.

Схеми цифрового автомату складаються з комбінаційних схем, які виконують певні операції, а також використовують запам'ятовуючі елементи.

1.3 Функціональна та принципова електричні схеми

Схема електрична – документ, що містить у вигляді умовних зображень або позначень складові частини виробу, що діють за допомогою електричної енергії, та їх взаємозв'язки [28].

Схема електрична структурна – документ, що визначає основні функціональні частини виробу, їх призначення та взаємозв'язки [28].

На структурній схемі зображують усі основні функціональні частини виробу (елементи, пристрої та функціональні групи) та основні взаємозв'язки між ними. Графічна побудова схеми має забезпечувати найкраще уявлення про послідовність взаємодії функціональних елементів у виробі. На лініях взаємозв'язків рекомендується стрілками позначати напрямок ходу процесів, що відбуваються у виробі.

Схема електрична функціональна – документ, який роз'яснює процеси, що протікають в окремих функціональних ланцюгах виробу (установки) або виробу (установки) в цілому [29].

Даними схемами користуються вивчення принципів роботи виробів, і навіть за її налагодженні, контролю та ремонті [30].

					КВРКІ 180114.18.01.04 ПЗ	Арк. 17
Зм.	Арк.	№ докум.	Підпис	Дата		

На функціональній схемі зображують функціональні частини виробу (елементи, пристрої та функціональні групи), що беруть участь у процесі, що ілюструється схемою, та зв'язки між цими частинами. Окремі функціональні частини допускається зображати у вигляді прямокутників. Графічна побудова схеми має давати найбільш наочне уявлення про послідовність процесів, що ілюструються схемою [31].

На електричній функціональній схемі мають бути вказані [32]:

1) для кожної функціональної групи – позначення, присвоєне їй на принциповій схемі, та(або) її найменування. Якщо функціональна група зображена як умовне графічне зображення, її найменування не вказують;

2) для кожного пристрою, зображеного у вигляді прямокутника, – позиційне позначення, присвоєне йому на принциповій схемі, його найменування та тип та(або) позначення документа, на підставі якого цей пристрій застосовано;

3) для кожного пристрою, зображеного у вигляді умовного графічного зображення, – позиційне позначення, присвоєне йому на принциповій схемі, та(або) його тип;

4) для кожного елемента – позиційне позначення, присвоєне йому на принциповій схемі, та(або) його тип. Найменування, позначення та типи рекомендується вписувати у прямокутники. Скорочені чи умовні найменування, якщо такі є на схемі, мають бути пояснені на полі схеми.

На схемі рекомендується [32]:

1) вказувати характеристики функціональних частин поруч із графічним позначенням чи на вільному полі схеми;

2) поміщати написи, діаграми або таблиці, що пояснюють послідовність процесів у часі;

3) вказувати параметри у характерних точках, наприклад, величини струмів, напруг, форми імпульсів тощо.

Функціональна схема відображає процеси, які відбуваються в окремих функціональних колах пристрою.

З метою підвищення зручності читання допускається функціональні ланцюги на одній схемі виконувати різними за товщиною лініями, застосовуючи не більше трьох розмірів.

Елементи та пристрої на схемі допускається зображати суміщеним або рознесеним способом, а схему виконувати у багатолінійному або однолінійному зображенні за правилами виконання принципів схем.

При рознесеному способі зображення допускається роздільно зображені частини елементів та пристроїв з'єднувати лінією механічного зв'язку (штрихова лінія).

Для виробу, до складу якого входять елементи різних видів, рекомендується випускати кілька схем відповідних видів одного типу або одну комбіновану схему, що містить елементи та зв'язки різних видів.

Схема електрична принципова – документ, що визначає повний склад елементів та взаємозв'язки між ними та, як правило, дає повне (детальне) уявлення про принципи роботи виробу (установки) [33].

На принциповій схемі зображують всі електричні елементи або пристрої, необхідні для здійснення та контролю у виробі встановлених електричних процесів, всі електричні взаємозв'язки між ними, а також електричні елементи (з'єднувачі, затискачі тощо), якими закінчуються вхідні та вихідні ланцюги. На схемі допускається зображати сполучні та монтажні елементи, що встановлюються у виробі з конструктивних міркувань [34].

Схеми виконують для виробів, що перебувають у відключеному положенні. В обґрунтованих випадках допускається окремі елементи схеми зображати у робочому положенні із зазначенням на полі схеми режиму, для якого зображені ці елементи [35].

Елементи або пристрої, що використовуються у виробі частково, допускається зображати неповністю, обмежуючись зображенням частин або елементів, що використовуються [36].

					КВРКІ 180114.18.01.04 ПЗ	Арк. 19
Зм.	Арк.	№ докум.	Підпис	Дата		

Елементи та пристрої зображують на схемах суміщеним або рознесеним способом. При суміщеному способі складові частини елементів або пристроїв зображують в безпосередній близькості один до одного. При рознесеному способі складові частини елементів та пристроїв зображують на схемах у різних місцях таким чином, щоб окремі ланцюги виробу були зображені найбільш наочно. Рознесеним способом допускається зображати всі окремі елементи або пристрої схеми [37].

При оформленні схем, з метою підвищення наочності, рекомендується використовувати рядковий спосіб зображення елементів (пристроїв), при якому умовне графічне зображення елементів або їх складових частин, що входять в один ланцюг, послідовно зображують один за одним по горизонтальній або вертикальній прямій, а окремі ланцюги – поруч, утворюючи паралельні (горизонтальні чи вертикальні) рядки. При оформленні схеми рядковим способом допускається нумерувати рядки арабськими цифрами [38].

При зображенні елементів (пристроїв) рознесеним способом допускається на вільному полі схеми поміщати умовне графічне зображення елементів (пристроїв), виконаних суміщеним способом. В даному випадку елементи (пристрої), що використовуються у виробі частково, зображують повністю із зазначенням як використаних, так і невикористаних частин (елементів). Висновки (контакти) невикористаних частин (елементів) зображають коротше, ніж виводи (контакти) невикористаних частин (елементів) [39].

Схеми виконують у багатолінійному чи однолінійному зображенні. При багатолінійному зображенні кожен ланцюг є окремою лінією, а елементи, що містяться в цих ланцюгах, - окремими умовними графічними зображеннями. При однолінійному зображенні ланцюга, що виконують ідентичні функції, є однією лінією, а однакові елементи цих ланцюгів – одним умовним графічним зображенням [40].

У разі зображення на схемі різних функціональних ланцюгів для підвищення зручності читання допускається ці ланцюги розрізняти за товщиною

					КвРКІ 180114.18.01.04 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

ліній. На одній схемі рекомендується застосовувати не більше трьох розмірів ліній за товщиною, при цьому на полі схеми, при необхідності, поміщають відповідні пояснення. Для спрощення схеми допускається кілька електрично не зв'язаних ліній зв'язку зливати до лінії групового зв'язку, але при підході до контактів (елементів) кожену лінію зв'язку зображують окремою лінією. При злитті ліній зв'язку кожену лінію позначають у місці злиття, а за необхідності, і на обох кінцях умовними позначеннями (цифрами, літерами чи їх поєднанням). Лінії зв'язку, які зливаються до лінії групового зв'язку, зазвичай, не повинні мати розгалужень, тобто кожен умовний номер повинен зустрічатися на лінії групового зв'язку двічі. При необхідності розгалужень їхня кількість вказується після порядкового номера лінії через дріб [40].

Кожен елемент та (або) пристрій, що має самостійну принципову схему та розглядається як елемент, що входять у виріб та зображені на схемі, повинні мати позиційне буквено-цифрове позначення. Пристроєм, які не мають самостійних важливих схем, і функціональним групам рекомендується також надавати позначення. Позиційні позначення елементам слід надавати в межах виробу. Порядкові номери елементам слід надавати, починаючи з одиниці, у межах групи елементів, яким на схемі присвоєно однакове літерне позиційне позначення, наприклад, С1, С2, С3 і т.д. Порядкові номери повинні бути присвоєні відповідно до послідовності розташування елементів на схемі зверху вниз у напрямку зліва направо. У технічно обґрунтованих випадках допускається змінювати послідовність присвоєння порядкових номерів залежно від розміщення елементів або функціональної послідовності передачі сигналів (інформації). При внесенні змін до схеми (корегування схеми) послідовність присвоєння порядкових номерів може бути порушена. Позиційні позначення проставляються на схемі поряд із умовним графічним зображенням елементів з правого боку або над ними. При зображенні на схемі елемента рознесеним способом позиційне позначення проставляють біля кожної складової частини. Якщо до складу виробу входять пристрої, що не мають самостійних принципових схем, то на схемах таких

					КВРКІ 180114.18.01.04 ПЗ	Арк. 21
Зм.	Арк.	№ докум.	Підпис	Дата		

виробів допускається позиційне позначення елементам пристроїв привласнювати в межах кожного пристрою. Якщо до складу виробу входить кілька однакових пристроїв, то позиційні позначення елементам пристроїв слід надавати в межах цих пристроїв. На схемі виробу, до складу якого входять функціональні групи, позиційні позначення елементам присвоюють наступним чином – спочатку надають позиційні позначення елементам, що не входять до функціональних груп, а потім елементів, що входять до функціональних груп. Якщо у виробі є кілька однакових функціональних груп, то позиційні позначення елементів, присвоєні в одній із цих груп, слід повторювати у всіх наступних групах. Позначення функціональної групи вказують біля зображення функціональної групи зверху або праворуч. Якщо поле схеми розбите на зони або схема виконана рядковим способом, то праворуч від позиційного позначення або під ним допускається вказувати у круглих дужках позначення зон та номери рядків, у яких зображені всі складові частини цього елемента або пристрою. Для підвищення зручності читання схеми допускається роздільно зображені частини елементів з'єднувати лінією механічного зв'язку, що вказує на належність їх до одного елемента. Позиційні позначення елементів у разі проставляють біля одного чи біля обох кінців лінії механічного зв'язку. При зображенні окремих елементів пристроїв у різних місцях у позиційні позначення цих елементів має бути включено позиційне позначення пристрою, яке вони входять за типом. При рознесеному способі зображення функціональної групи до складу позиційних позначень елементів, що входять до цієї групи, має бути включено позначення функціональної групи типу. При однолінійному зображенні біля одного умовного графічного зображення, що замінює кілька умовних графічних зображень однакових елементів (пристроїв), вказують позиційні позначення всіх цих елементів (пристроїв). Якщо однакові елементи (пристрої) перебувають у всіх ланцюгах, зображених однолінійно, то праворуч від позиційного позначення чи під ним у квадратних дужках вказують позначення ланцюгів, у яких перебувають ці елементи (пристрої) [41].

					КВРКІ 180114.18.01.04 ПЗ	Арк. 22
Зм.	Арк.	№ докум.	Підпис	Дата		

На принциповій схемі повинні бути однозначно визначені всі елементи та пристрої, що входять до складу виробу та показані на схемі. Дані про елементи та пристрої повинні бути записані до списку елементів. Зв'язок переліку елементів з умовними графічними зображеннями елементів та пристроїв має здійснюватись через позиційні позначення. У технічно обгрунтованих випадках допускається всі відомості про елементи та пристрої розміщувати біля умовних графічних позначень [42].

При складному входженні, наприклад, коли у пристрій, що не має самостійної принципової схеми, входить один або кілька пристроїв, що мають самостійні принципові схеми, і (або) функціональних груп, або якщо в функціональну групу входить один або кілька пристроїв і т.д., то в переліку елементів у графі «Найменування» перед найменуванням пристроїв, що не мають самостійних принципових схем, і функціональних груп допускається проставляти порядкові номери (тобто подібно до позначення розділів, підрозділів тощо) текстового документа в межах всієї схеми виробу [43].

1.4 Висновки. Постановка задачі

Операційний автомат – це пристрій, який безпосередньо виконує операції над інформацією з врахуванням керуючих сигналів і надає результати цих операцій. Його складовими вузлами є суматори, регістри, мультиплексори, дешифратори, арифметико-логічний пристрій та інші функціональні блоки, які беруть участь у процесі виконання команд (запис в регістр, зсув коду ліворуч або праворуч, дешифрування двійкового коду, інвертування вмісту регістра, додавання двох чисел, тощо).

Проектований операційний автомат повинен видавати певну продуктивність, тому синтезована структура не повинна вносити обмеження на сумісність мікрооперацій, тобто повинно бути можливим одночасне виконання мікрооперацій, що є функціонально сумісними.

					КВРКІ 180114.18.01.04 ПЗ	Арк. 23
Зм.	Арк.	№ док.ум.	Підпис	Дата		

Автомати Мілі (Mealy) – цифрові автомати, в яких існує залежність вихідного сигналу автомата від вхідного сигналу та його поточного внутрішнього стану.

Автомати Мура (Moore) – цифрові автомати, в яких існує залежність вихідного сигналу автомата тільки від його поточного внутрішнього стану.

Схема електрична функціональна – документ, який роз'яснює процеси, що протікають в окремих функціональних ланцюгах виробу (установки) або виробу (установки) в цілому.

Схема електрична принципова – документ, що визначає повний склад елементів та взаємозв'язки між ними та, як правило, дає повне (детальне) уявлення про принципи роботи виробу (установки).

В кваліфікаційній роботі слід розв'язати наступну задачу: побудувати операційний автомат, що виконує обчислення кількості непарних елементів у двовимірному масиві $A[n, m]$, які належать інтервалу $[-5; 15]$. Слід використовувати схему Уілкса-Стрінжера для реалізації мікропроцесорного автомату як автомату Мілі. Слід використовувати логічні елементи, тригери для реалізації та синтезу функціональної схеми автомату.

					КВРКІ 180114.18.01.04 ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ОПЕРАЦІЙНОГО АВТОМАТУ НА ОСНОВІ АВТОМАТУ МІЛІ

2.1 Моделювання операційного автомату

Функція операційного автомату представляє список дій, які автомат може і повинен виконувати, але жодним чином не визначає порядок цих дій (у часі). Отже, функція операційного автомату характеризує засоби, що використовуватимуться при обчисленнях, але не сам обчислювальний процес.

Вищерозглянута концепція операційних і керуючих автоматів, а також принцип мікропрограмного керування лежать в основі проєктування та реалізації операційних пристроїв різного призначення.

Мікропрограмування – спосіб опису функцій операційних пристроїв, який не стосується використовуваних технічних засобів для їх реалізації. Так, мікропрограмування дозволяє виконати формалізацію структури операційних пристроїв і не залежить від способу керування роботою пристрою. Основою теорії і практики проєктування та реалізації пристроїв ЕОМ є принципи і методи проєктування керуючих та операційних автоматів.

Операційними пристроями є процесори, вузли управління зовнішніми пристроями, канали введення-виведення, які використовуються для виконання операцій над інформацією. Функція операційного пристрою – виконання конкретної множини операцій над вхідними словами для обчислення вихідних слів (результатів операцій).

Принцип мікропрограмного керування лежить в основі організації операційних пристроїв і полягає в наступному:

- 1) будь-яка операція є складною дією, до складу якої входить послідовність елементарних дій (мікрооперацій) над словами інформації;
- 2) логічні умови застосовуються для керування порядком проходження мікрооперацій;

3) процес виконання операцій визначається як алгоритм в термінах мікрооперацій і логічних умов, що формує порядок перевірки логічних умов та виконання мікрооперацій для отримання необхідних результатів;

4) мікропрограма складається з функцій пристрою, представлених певною мовою у певному вигляді, на базі яких визначається порядок функціонування пристрою в часі та синтезується його структура пристрою.

Проектування мікропрограмного автомату за схемою Уїлкса-Стрінжера складається з наступних кроків:

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

Змістовна схема алгоритму (рис. 2.1) складається з операційних та умовних верхівок. Алгоритм, згідно із поставленим завданням виконує обчислення кількості непарних елементів у двовимірному масиві $A[n, m]$, які належать інтервалу $[-5; 15]$. Алгоритм складається з чотирьох умовних верхівок і восьми операційних верхівок.

І операційні, і умовні верхівки кодуються. Якщо мікрооперації та умовні верхівки потворюються (виконують ту ж саму дію), то кодуються вони однаково. В алгоритмі, розробленому згідно із поставленим завданням, мікрооперації не повторюються, отже, верхівок, які могли б кодуватись одним кодом, немає. У таблиці 2.1 представлена таблиця кодування верхівок.

Закодована мікроопераційна схема алгоритму (рис. 2.2) розробляється на основі змістовної схеми алгоритму (рис. 2.1) і таблиці кодування верхівок (таблиця 2.1) шляхом кодування та заміни відповідних блоків.

Змістовна схема алгоритму та закодована мікроопераційна схема алгоритму представлені також у Додатку А.

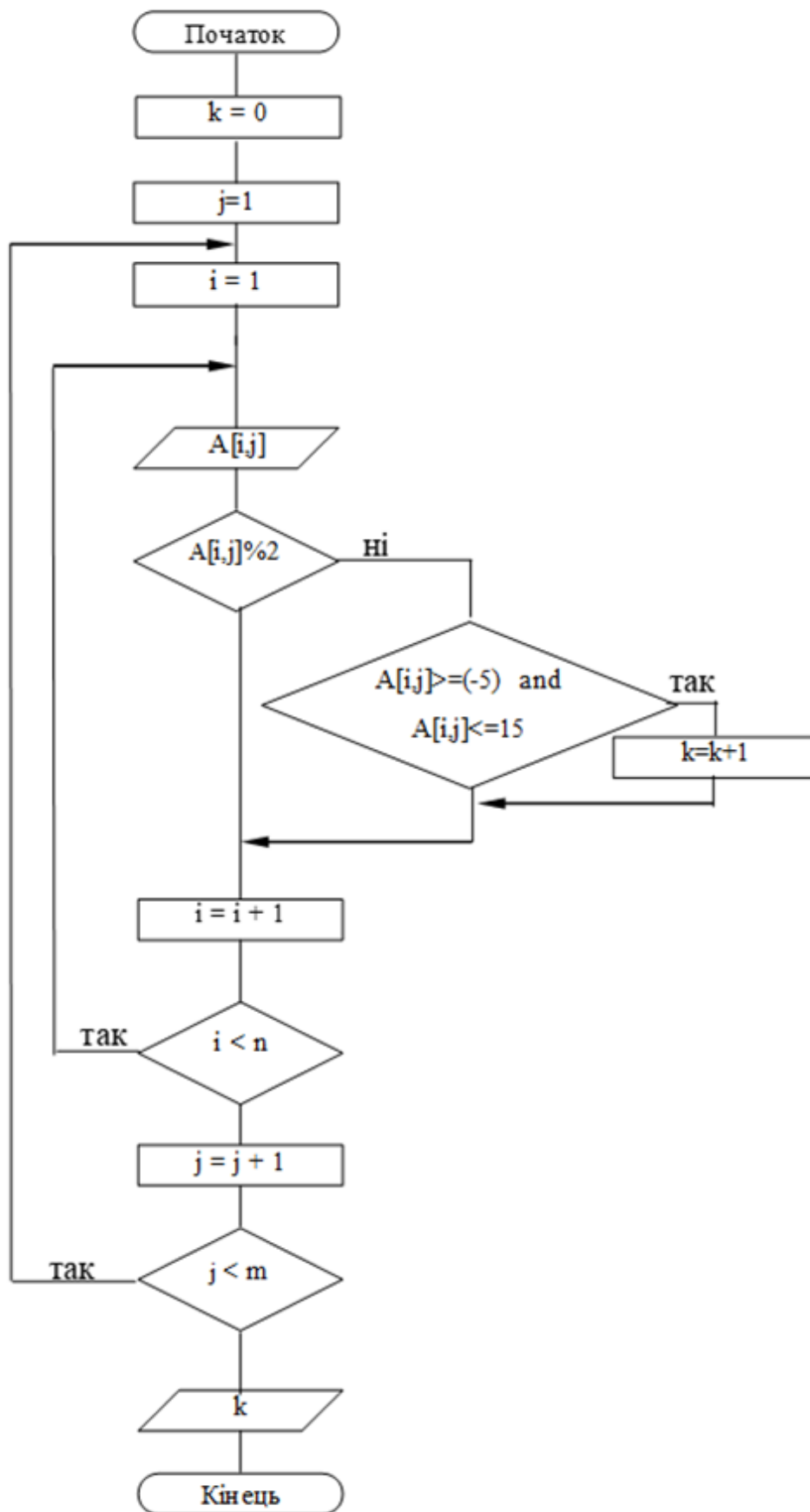


Рисунок 2.1 – Змістовна схема алгоритму

Кожна операція кодується мікрокомандою для створення таблиці кодування мікрокоманд. В одну мікрокоманду об'єднуються ті мікрооперації, які виконуються одна за одною послідовно протягом одного такту часу. В алгоритмі, побудованому згідно із поставленим завданням, дві мікрооперації (mY1, mY2) виконуються послідовно протягом одного такту часу. Отже, їх можна об'єднати в одну мікрокоманду. Таблиця кодування мікрокоманд представлена у таблиці 2.2.

Для побудови закодованої мікрокомандної схеми алгоритма (рис. 2.3) зробимо розмітку внутрішніх станів автомату Мілі наступним чином: мітки ставляться після кожної мікрокоманди (перед блоком після операційного блоку); початковий та кінцевий блоки мікрокомандної схеми алгоритму позначаються мітками a0; мітки нумеруються згідно із порядковим номером. Варто врахувати, що перехід зі стану в стан повинен проходити через верхівку автомату (умовну або операційну).

Таблиця 2.1 – Таблиця кодування верхівок

№	Код	Зміст	Примітки
1	mY1	$k = 0$	
2	mY2	$j = 1$	
3	mY3	$i = 1$	
4	mY4	$A [i,j]$	Ввести елемент масиву $A [i,j]$
5	mY5	$k = k+1$	
6	mY6	$i = i + 1$	
7	mY7	$j = j + 1$	
8	mY8	k	Вивести k
9	X1	$A [i,j] \% 2$	так - 1, ні - 0
10	X2	$A[i,j] \geq (-5)$ and $A [i,j] \leq 15$	так - 1, ні - 0
11	X3	$i < n$	так - 1, ні - 0
12	X4	$j < m$	так - 1, ні - 0

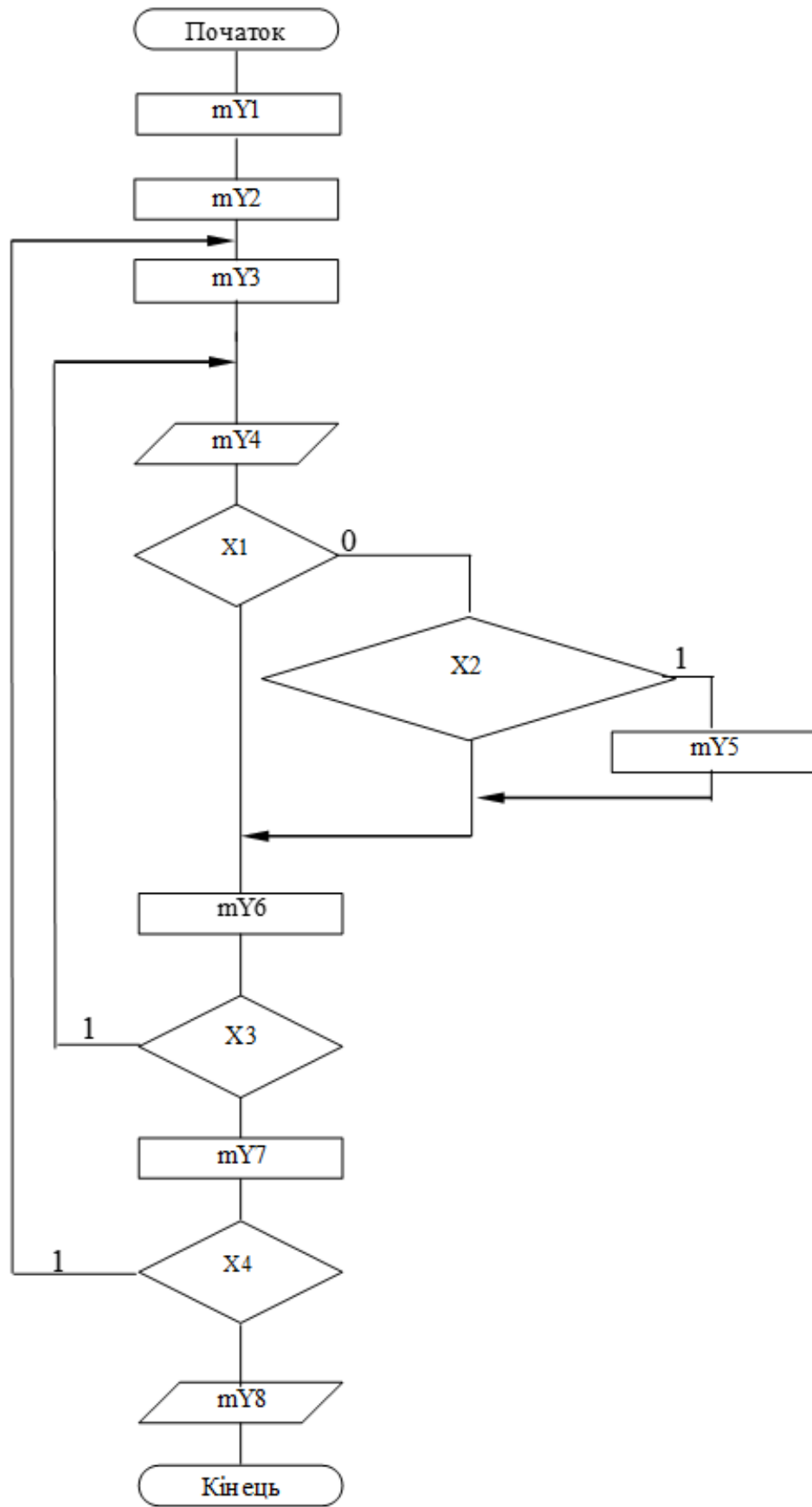


Рисунок 2.2 – Закодована (для автомату Мілі) мікроопераційна схема алгоритму

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

Зм.	Арк.	№ докум.	Підпис	Дата

КВРКІ 180114.18.01.04 ПЗ

Арк.
30

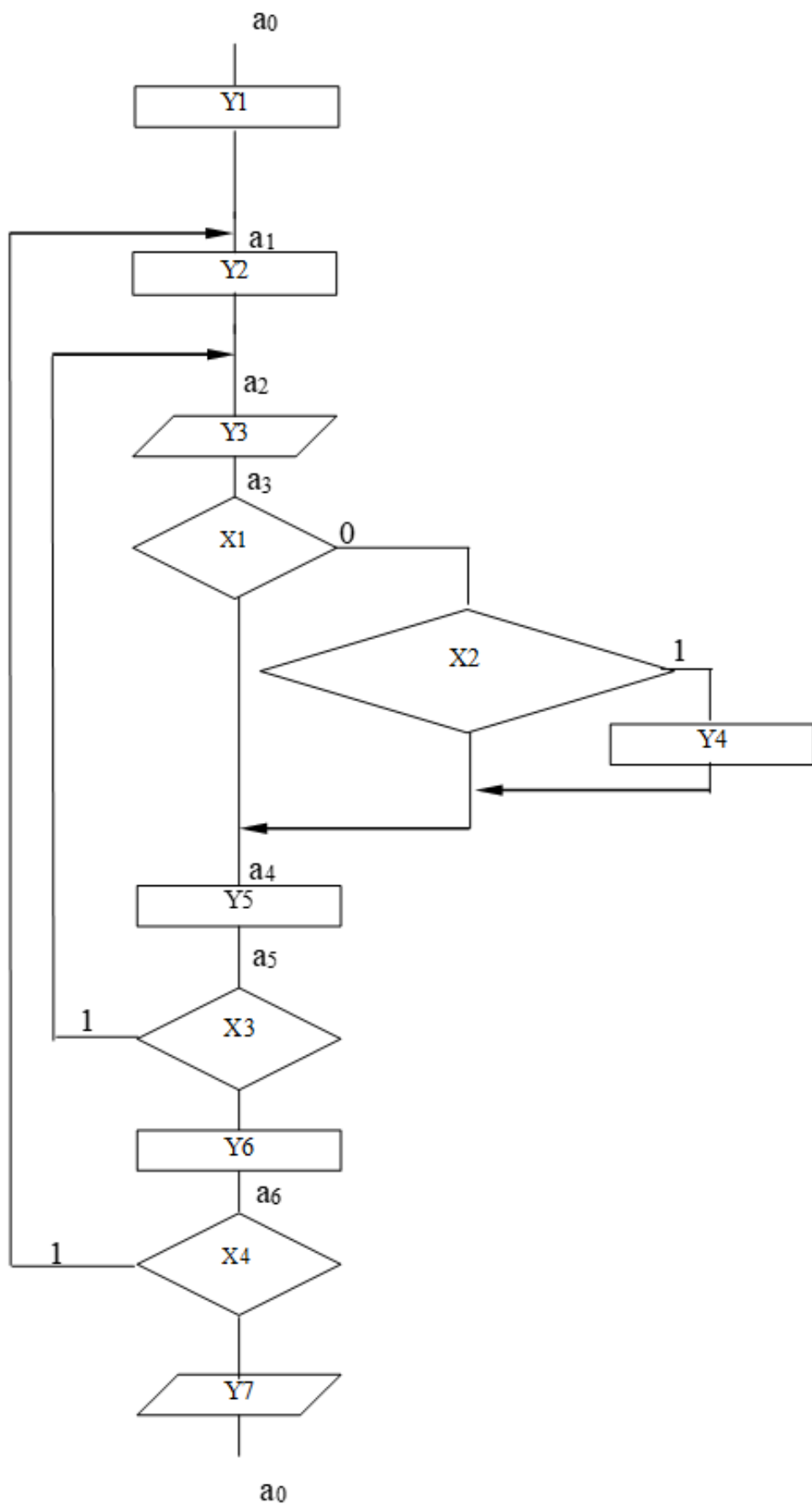


Рисунок 2.3 – Закодована (для автомату Мілі) мікрокомандна схема алгоритму

Зм.	Арк.	№ докум.	Підпис	Дата

2.2 Проектування операційного автомату

Основна таблиця автомату Мілі (таблиця 2.3) створюється на основі закодованої (для автомату Мілі) мікрокомандної схеми алгоритму (рис. 2.3). Перший ствець таблиці містить стани, в яких може знаходитись автомат. Умови та способи переходу автомату з одного стану в інший містить перший рядок таблиці. У другому стовпці таблиці визначено безумовний перехід СІ (під дією синхроімпульсного сигналу), у наступних стовпцях – умови переходу автомату. У відповідних комірках таблиці відображено, в який стан виконується перехід та який вихідний сигнал видається. Наприклад, із стану a_1 відбувається перехід автомату в стан a_2 , в результаті чого видається вихідний сигнал Y_2 , цей перехід є безумовним, отже, в рядку стану a_1 в другому стовпці таблиці СІ буде записано a_2/ Y_2 .

Після складання таблиці автомату Мілі побудуємо граф-схему переходів (рис. 2.4). Така граф-схема буде базуватись на закодованій мікрокомандній схемі алгоритму (рис. 2.3) і основній таблиці автомату Мілі (таблиця 2.3). Це буде орієнтований граф, в якому можливі стани автомату позначаються колами, а переходи зі стану в стан показують орієнтовані дуги (дуги-стрілки). Над стрілкою слід підписати умову переходу, а також який вихідний сигнал видається.

Закодована мікрокомандна схема алгоритму та граф-схема переходів автомату представлені також у Додатку Б.

На основі створених основної таблиці автомату Мілі (таблиця 2.3) і граф-схеми переходів автомату Мілі (рис. 2.4) побудуємо систему рівнянь переходів та систему рівнянь виходів автомату Мілі.

Для цього спершу визначимо кількість необхідних для кодування внутрішніх станів автомату тригерів (n) за формулою: $\log_2 A \leq n$, де A – кількість станів a_i (для розроблюваного автомату $a_0 - a_6$): $A = 7$; $\log_2 7 \approx 3$; отже, для кодування внутрішніх станів автомату знадобиться 3 тригери.

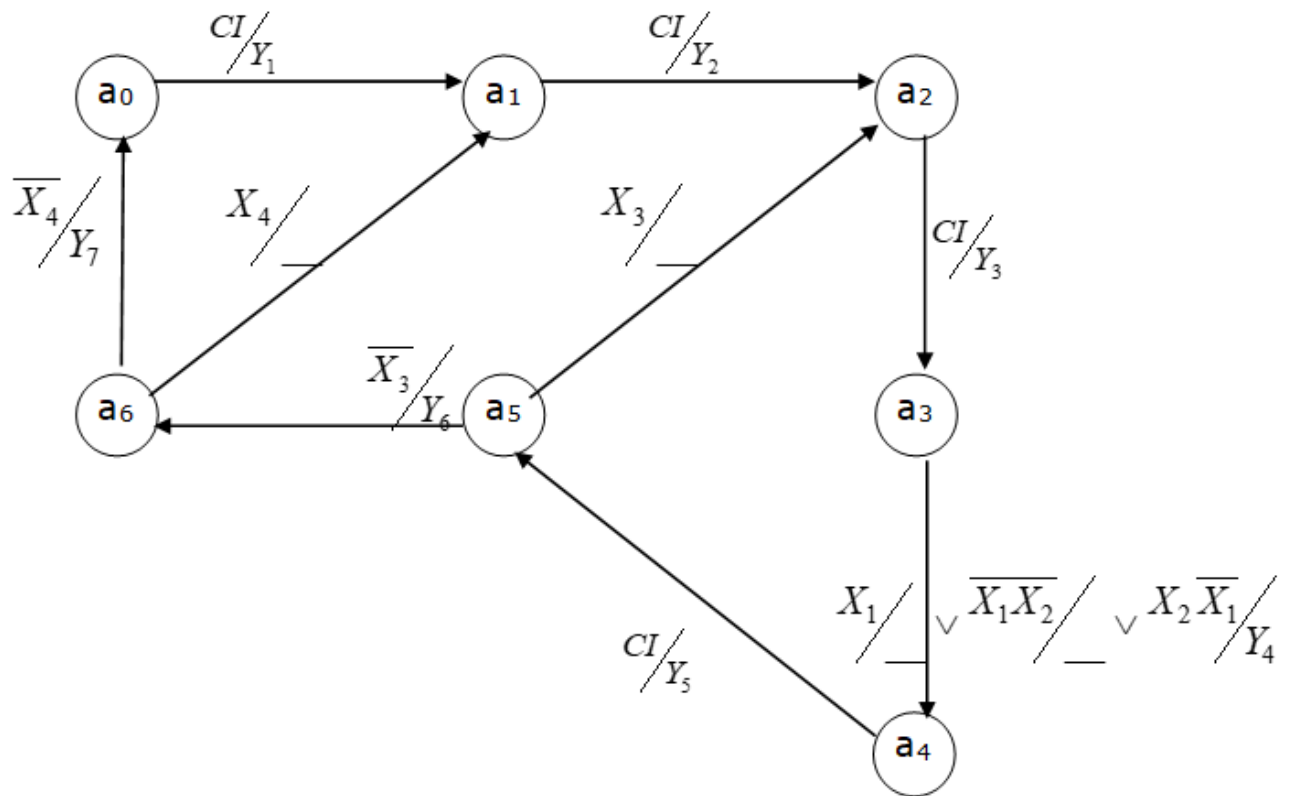


Рисунок 2.4 – Граф-схема переходів автомату Мілі

Система рівнянь переходів автомату Мілі:

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

Система рівнянь виходів автомату Мілі:

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

2.3 Висновки

В другому розділі операційний автомат на основі автомату Мілі спроектовано за схемою Уїлкса-Стрінжера. Спроекований автомат виконує обчислення кількості непарних елементів у двовимірному масиві $A[n, m]$, які належать інтервалу $[-5; 15]$. Отже, у другому розділі кваліфікаційної роботи розроблено:

- 1) змістовну схему алгоритму;
- 2) закодовану мікроопераційну схему алгоритму;
- 3) закодовану мікрокомандну схему алгоритму;
- 4) основну таблицю автомату Мілі;
- 5) граф-схему переходів автомату Мілі;
- 6) системи рівнянь переходів та виходів.

3 РЕАЛІЗАЦІЯ ТА МОДЕЛЮВАННЯ РОБОТИ ОПЕРАЦІЙНОГО АВТОМАТУ НА ОСНОВІ АВТОМАТУ МІЛІ

3.1 Застосування схеми Уїлкса-Стрінжера для реалізації операційного автомату на основі автомату Мілі

Проектування мікропрограмного автомату на основі автомату Мілі за схемою Уїлкса-Стрінжера складається з наступних кроків:

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

Мікропрограмним автоматом називається блок управління, у якого схемна логіка замінена постійним запам'ятовуючим пристроєм (ПЗП) мікропрограм. В принципі він може бути побудований за будь-якою моделлю – Мілі або Мура.

Логічну структуру мікропрограмного автомата можна описати, виводячи її зі структури керуючого автомата зі схемною логікою. Відмінною особливістю мікропрограмного автомата із програмованою логікою є наявність пам'яті мікропрограм. Кожній команді обчислювального пристрою цієї пам'яті відповідає мікропрограма.

Найпростіші розділи електронного комп'ютера є тими, які мають просту логічну структуру. Крім того, що ця структура може бути прийнята до уваги інженером з техобслуговування при пошуку відмови, вона також дозволяє використовувати визначальні місця відмови програми і протестувати обладнання без використання ретельно продуманого тестового механізму. Саме в розділі управління електронно-обчислювальних машин зазвичай виникає найбільша складність. Це важливо, тому що в машині розробляли всебічний код замовлення, щоб зробити його простим та швидким у роботі. В цілому для кожного різного порядку в коді має бути забезпечене деяке спеціальне обладнання, і складніша

					КВРКІ 180114.18.01.04 ПЗ	Арк. 36
Зм.	Арк.	№ докум.	Підпис	Дата		

функція порядку - це складніше обладнання. У минулому страх перед незаконним ускладненням схеми керування машин перешкодив тому, щоб розробники електронних машин надали такі послуги, як замовлення на операції з плаваючою комою, незважаючи на те, що досвід із релейними машинами та з підпрограмами, що інтерпретують, показав, наскільки цінні такі замовлення. Схема Уілкса-Стрінжера описує метод розробки схем управління машиною, яка є абсолютно логічною і яка дозволяє змінам або доповненням до коду замовлення бути зробленими без оперативних змін до схем. Ці елементарні операції будуть згадуватись як мікрооперації. Основні операції машини, такі як додавання, віднімання, множення і т.і., розглядаються як складені з мікропрограм мікрооперації, що вимагали мікропорядку.

Система описується щодо паралельної машини, розробляючи арифметичний модуль вздовж стандартних рядків. Вона містить ряд регістрів та суматор разом із системою комутації, яка дозволяє мікроопераціям машини виконуватись. Частина мікрооперацій будуть використовуватись для передачі числа від одного регістра до іншого, в той час як інші включають використання суматора. Якась конкретна мікрооперація може бути виконана, застосувавши імпульси одночасно до належних логічних елементів комутаційної системи. У певних випадках для двох або більше мікрооперацій може бути одночасне виконання.

Регістр необхідний, щоб утримувати адресу наступного порядку, який повинен виконуватись, та інший, щоб утримувати поточний порядок, у той час як він виконується. Деякі засоби підрахунку числа кроків або при множенні повинні також бути забезпечені. Метод відповідності цим вимогам повинен забезпечити групу регістрів та суматора разом із системою комутації, яка включає передачі чисел. Цю частину системи управління викликає модуль регістру команд. У будь-якому випадку операції повинні бути виконані на числах, що стоять у модулі регістру команд під час виконання порядку, як операції, що виконуються в арифметичному модулі, розціненому як послідовність мікрооперацій, кожна з яких виконується додатком імпульсів до належних логічних елементів.

					КВРКІ 180114.18.01.04 ПЗ	Арк. 37
Зм.	Арк.	№ докum.	Підпис	Дата		

Інша частина системи управління стосується управління послідовності мікрокоманд, необхідних виконати кожен порядок машини, і пов'язана з роботою логічних елементів, необхідних для виконання кожної мікрокоманди. Це викликає мікроблок керування; він складається з дерева декодування, двох матриць і двох регістрів (додаткових до регістрів з модуля регістра команд), який показує, як імпульси раніше керували логічними елементами в арифметичному модулі і як генерується модуль регістру команд. Серія імпульсів управління імпульсного генератора застосовується до введення дерева декодування. Кожен імпульс спрямований до одного з вихідних рядків дерева, згідно з числом, що стоїть у регістрі. Вихідні рядки передаються в матрицю випрямляча та виводи цієї матриці є імпульсами, які керують різними логічними елементами, пов'язаними з мікроопераціями. Таким чином, один вхідний рядок матриці відповідає одній мікрокоманді. Адреса мікрокоманди - число, яке має бути поміщене в регістр, щоб змусити імпульс управління бути спрямованим до відповідного рядка. Вихідні рядки від дерева також передаються до другої матриці. Безпосередньо перед тим, як наступний імпульс управління застосовується до введення дерева, зв'язок встановлюється між регістрами, і адреса мікрокоманди, яка повинна виконуватися, передається в регістр. Таким чином, дерево декодування готове направити наступний імпульс управління до коректного вихідного рядка. Таким чином, додаток імпульсів по черзі до введення дерева і до з'єднання логічного елемента реєструє причини наперед визначеної послідовності мікрокоманд, які будуть виконуватись.

Необхідно мати засоби, за допомогою чого курс мікропрограми може бути зроблений умовним виразом на тому, чи дана цифра в одному з регістрів арифметичного модуля або модуля регістра команд 1 або 0. Двосторонній перемикач, яким керує спеціальний тригер, викликає умовний тригер, вставлений між матрицями. Умовний тригер може бути встановлений ранішою мікрокомандою з будь-якою цифрою від будь-якого з регістрів. Дві окремі адреси, з'єднані у матрицю, стають адресою наступної мікрокоманди.

					КВРКІ 180114.18.01.04 ПЗ	Арк. 38
Зм.	Арк.	№ докум.	Підпис	Дата		

Умовні мікрокоманди грають ту саму роль у конструкції мікропрограм як умовна гра замовлень у конструкції стандартних програм; крім їх очевидного використання в мікропрограмах для таких операцій як множення і ділення, вони дозволяють циклам мікрокоманд, що повторюються, використовуватися.

При бажанні два розгалуження можуть бути вставлені у зв'язку між матрицями так, щоб будь-яка з чотирьох альтернативних адрес для наступної мікрокоманди могла бути обрана відповідно до налаштувань двох умовних тригерів. Інша можливість у тому, щоб зробити висновок з відгалуження дерева декодування, перш ніж це введе матрицю так, щоб природа мікрооперації, яка виконується, залежала від установки умовного тригера.

Мікропрограма містить розділи для виконання операцій, що вимагаються кожним порядком у основному коді команди машини. Щоб ініціювати роботу, тільки необхідно, щоб управління мікропрограмою було відправлено в коректну точку входу. Це може бути зроблено, помістивши функціональні цифри порядку найменш значної частини регістру, інші цифри у цьому обнуленому регістрі. Мікропрограма створена так, щоб коли це число передається в регістр, управління в мікропрограмі було відправлено в коректну точку входу.

Система комутації в арифметичному модулі може бути розроблена, щоб дозволити великій різноманітності мікрооперацій виконуватися, або це може бути обмежено, щоб дозволити тільки невелику кількість таких операцій. Машина з всебічним кодом замовлення має більш гнучку систему комутації, що дозволяє економії бути зробленою в числі мікрокоманд, необхідних в мікропрограмі.

Подібний коментар застосовується у зв'язку зі ступенем гнучкості, який буде забезпечений при розробці системи комутації для модуля регістру команд. Якщо специфікація машини дозволяє тому ж числу регістрів використовуватися в арифметичних розділах та розділах управління, конструкція цих двох розділів може бути ідентичною. Можна використовувати ідентичні модулі в арифметичному модулі і в модулі регістру команд.

Припущення, що ці мікрооперації займають той самий відрізок часу до виконання, малоімовірно, що підтверджено практично. Зокрема, в паралельній машині може бути неможливим розробити суматор, в якому час розповсюдження переносу досить короткий, щоб дозволити доповненню виконуватися в істотно той же відрізок часу, тому, генератор форми хвилі, що подає дерево декодування, повинен, коли відповідно стимульовано імпульсом від одного з виводів матриці А, надати трохи більший імпульс, ніж це зазвичай потрібно. Інші операції можуть брати багато разів так само довго, щоб виконуватись, як звичайна мікрокоманда; наприклад, доступ до та від сховища (особливо, якщо сховище затримки використовується), і робота пристроїв введення та виведення машини. Послідовність операцій у мікропрограмі тому має бути перервано. Спосіб зробити це полягає в тому, щоб запобігти імпульсам від генератора форми хвилі, що досягає дерева декодування протягом часу очікування. Цей метод, незважаючи на те, що здається досить можливим, включає просто вид складності, для уникнення якої існуюча система розроблена. Привабливіша система повинна змусити машину чекати на умовній мікрокоманді, яка повертає управління собі, якщо пов'язаний умовний тригер не встановлений. Установка цього тригера має місце, коли робота завершена, і управління переходить до наступної мікрокоманди в послідовності. Машина знаходиться, таким чином, в умові "динамічної зупинки", чекаючи роботи, яка буде завершена. Ця система має перевагу, що жодна складність не введена в модулі, що надають форми хвилі до дерева декодування, і що необхідні контрольно-вимірювальні прилади подібні до цього, вже передбачили інші цілі.

Схема Уїлкса-Стрінжера – це найбільш поширена структура схеми мікропрограмного автомату. Спрощена схема Уїлкса-Стрінжера представлена на рис. 3.1.

					КВРКІ 180114.18.01.04 ПЗ	Арк. 40
Зм.	Арк.	№ докум.	Підпис	Дата		

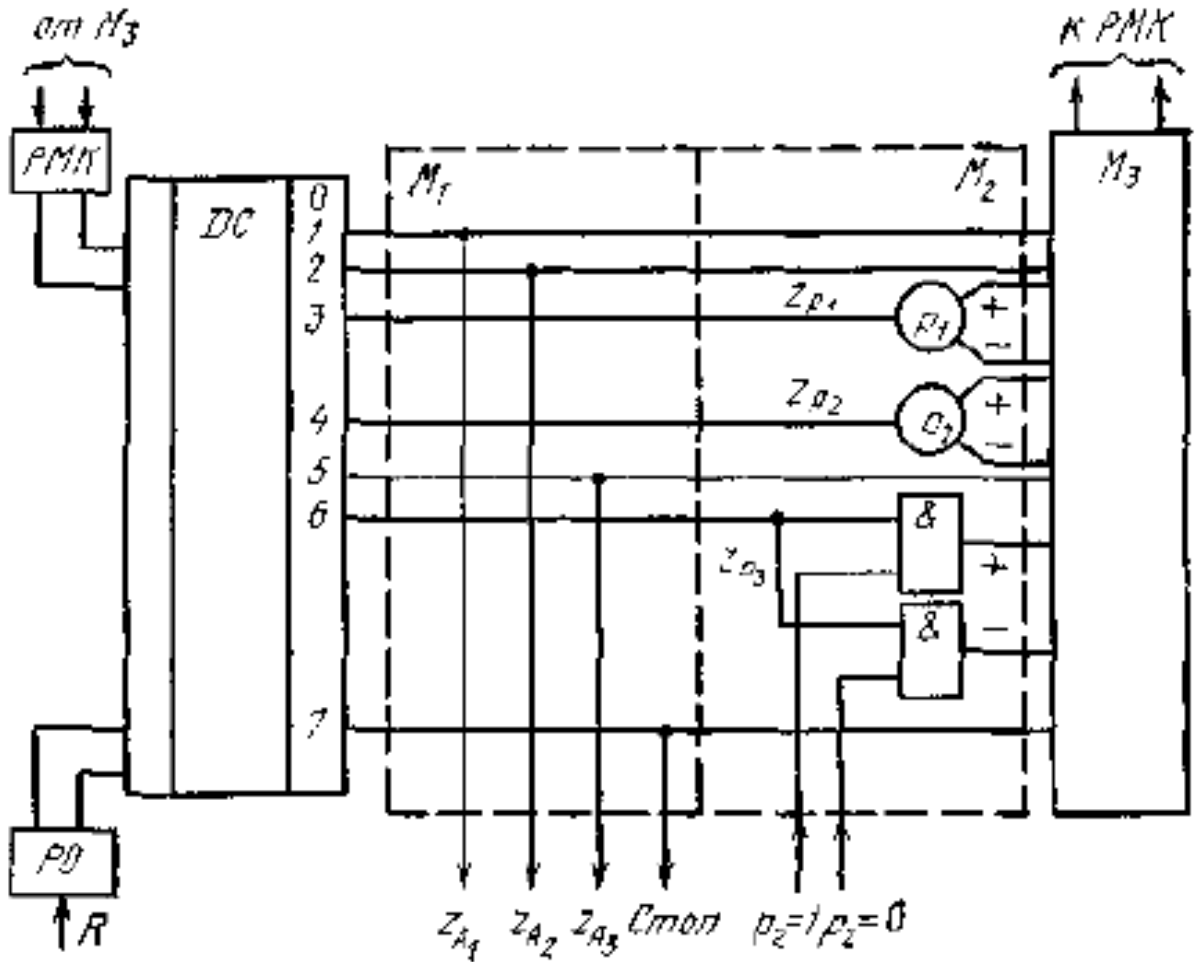


Рисунок 3.1 – Спрощена схема Уїлкса-Стрінжера [2, 3]

Схема містить регістр мікрокоманд РМК, дешифратор DC, матрицю зовнішніх мікрооперацій M1, матрицю внутрішніх мікрооперацій M2, матрицю формування коду наступної мікрокоманди M3 і регістр операцій PO.

Вихід дешифратора DC зіставляється з мікрокомандою, виконуваною за один такт роботи автомата. Код кожної мікрокоманд зберігається в РМК. Після виконання даної мікрокоманди в M3 формується код наступної мікрокоманди, який передається в РМК. Послідовність мікрокоманд утворює мікропрограма, код якої зберігається в PO. 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 PO поєднують з РМК, а код зіставляється з кодом першої мікрокоманди прошивки.

3.2 Реалізація операційного автомату на основі автомату Мілі

Операційний автомат складається з трьох частин – вхідної, перехідної та вихідної (схема електрична функціональна, додаток В).

Вхідна частина складається з чотирьох RS-тригерів, чотирьох логічних елементів АБО, на які подається вхідний сигнал, декодера та двох шин, одна з яких необхідна для передачі сигналів, які надходять з декодера, а інша - для сигналів з виходів компаратора.

Тригер є пристроєм з двома стійкими станами, у які він переходить під дією наперед визначених вхідних сигналів. В тригерів розрізняють два типи вхідних сигналів: інформаційні сигнали та синхросигнали.

Новий стан тригера визначається інформаційними сигналами, які є в будь-яких тригерах. Класифікація тригерів здійснюється якраз за типом інформаційних сигналів. Для фіксації переходу тригера в новий стан використовується синхросигнал, який не є обов'язковим.

RS-тригер – тригер з роздільними входами. Такий тригер має два вхідних канали (R і S) та один вихідний канал (Q). Вхід S (set) - вхід установки в одиницю, вхід R (reset) – вхід установки в нуль. Умовне позначення RS-тригера представлено на рис. 3.2.

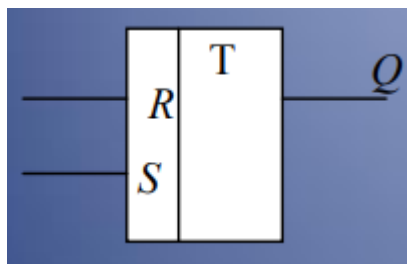


Рисунок 3.2 – Умовне позначення RS-тригера

Таблиця переходів RS-тригера представлена в таблиці 3.1.

Таблиця 3.1 – Таблиця переходів RS-тригера

R	S	Q^{t+1}
0	0	0
0	1	1
1	0	0
1	1	-

Таблиця функцій входів RS-тригера представлена в таблиці 3.2.

Таблиця 3.2 – Таблиця функцій входів RS-тригера

Q^t	Q^{t+1}	R^t	S
0	0	X	0
0	1	0	1
1	0	1	0
1	1	0	X

В таблиці переходів при подачі комбінації $S = R = 1$ стан переходу Q^{t+1} не визначений, і ця комбінація сигналів є забороненою для RS-тригера.

Аналізуючи таблиці 3.1 і 3.2, відзначимо що перехід тригера з 0 у 0 вимагає подачі комбінації $R=0, S=0$ або $R=1, S=0$, тобто цей перехід буде при комбінації $R=X, S=0$.

Диз'юнкція (функція АБО) – це функція, яка має істинне значення лише тоді, коли хоча б один з її аргументів є істинним. Умовне позначення логічного елемента АБО представлена на рис. 3.3.

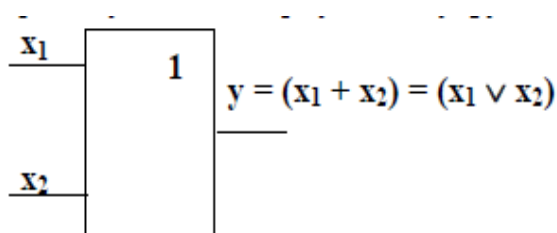


Рисунок 3.3 – Умовне позначення логічного елемента АБО

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

Декодер – елемент, який виконує перетворення n-розрядного двійкового коду в унітарний код «один з N». Дешифратор є повним, якщо $N=2^n$; дешифратор є частковим, якщо $N<2^n$.

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

Таблиця 3.4 – Таблиця істинності декодера

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

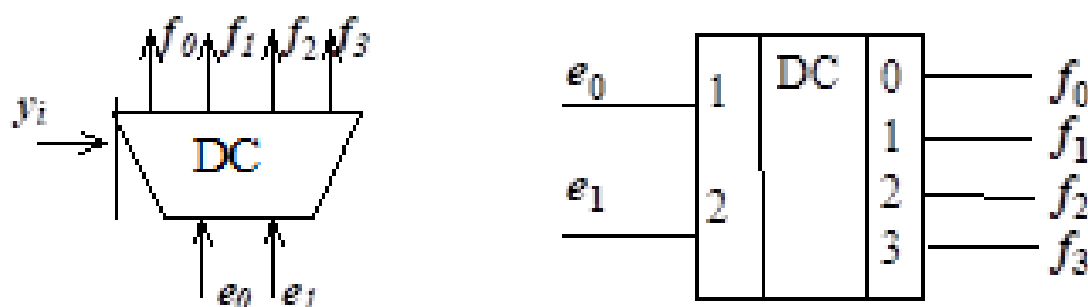


Рисунок 3.4 – Умовні позначення декодера

Шина – це сукупність провідників, яка використовується для передачі слів інформації.

Цифровий компаратор – цифровий пристрій, який використовується для порівняння двійкових або двійково-десяткових чисел. Компаратор діє за законами двійкової арифметики над арифметичними числами 0 і 1.

В такому разі можуть мати місце три випадки при порівнянні двох чисел: $A = B$; $A > B$ і $A < B$. Тобто компаратор може визначати не тільки рівність чисел A і B , але й їх нерівність. Компаратори нерівності мають 2 виходи: y_1 та y_2 . На виході y_1 при $A < B$ з'являється рівень логічної 1, а при $A > B$ рівень логічної 1 з'являється на виході y_2 .

Умовне позначення повного компаратора наведено на рис. 3.5.

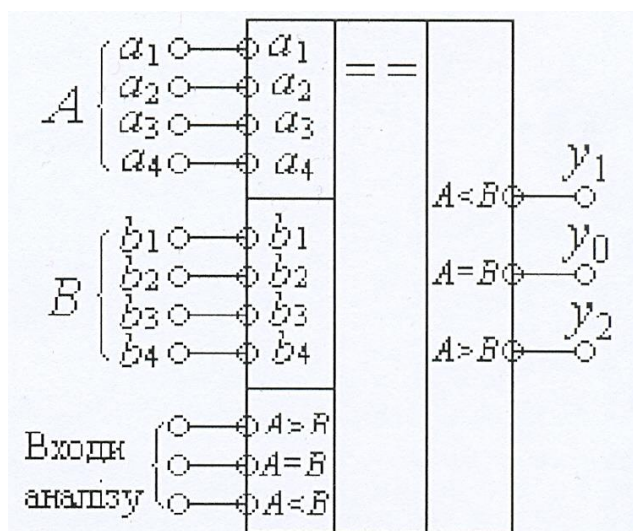


Рисунок 3.5 – Умовне позначення повного компаратора

У перехідній частині автомата відбувається перетворення сигналу протягом одного такту часу. Після логічних елементів ТА і/або АБО сигнал змінюється, і результат надходить на шину (a_t), звідки продовжує передаватися до програмованої логічної матриці (ПЛМ). Перехідна частина автомату будується на основі системи рівнянь переходів.

Кон'юнкція (функція ТА) – функція, яка має істинне значення тільки тоді, коли одночасно істинними є всі її аргументи. Умовне позначення логічного елемента ТА представлено на рис. 3.6.

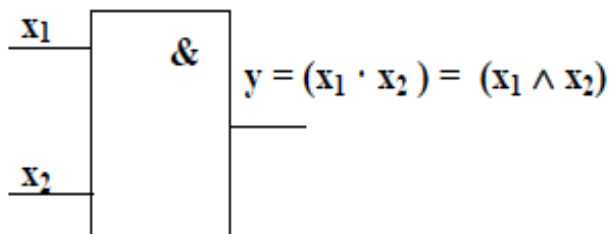


Рисунок 3.6 – Умовне позначення логічного елемента ТА

Таблиця істинності логічного елемента ТА представлена в таблиці 3.5.

Таблиця 3.5 – Таблиця істинності логічного елемента ТА

x1	x2	y
0	0	0
0	1	0
1	0	0
1	1	1

Узагальнена структура програмованої логічної матриці (ПЛМ) наведена на рис. 3.7.

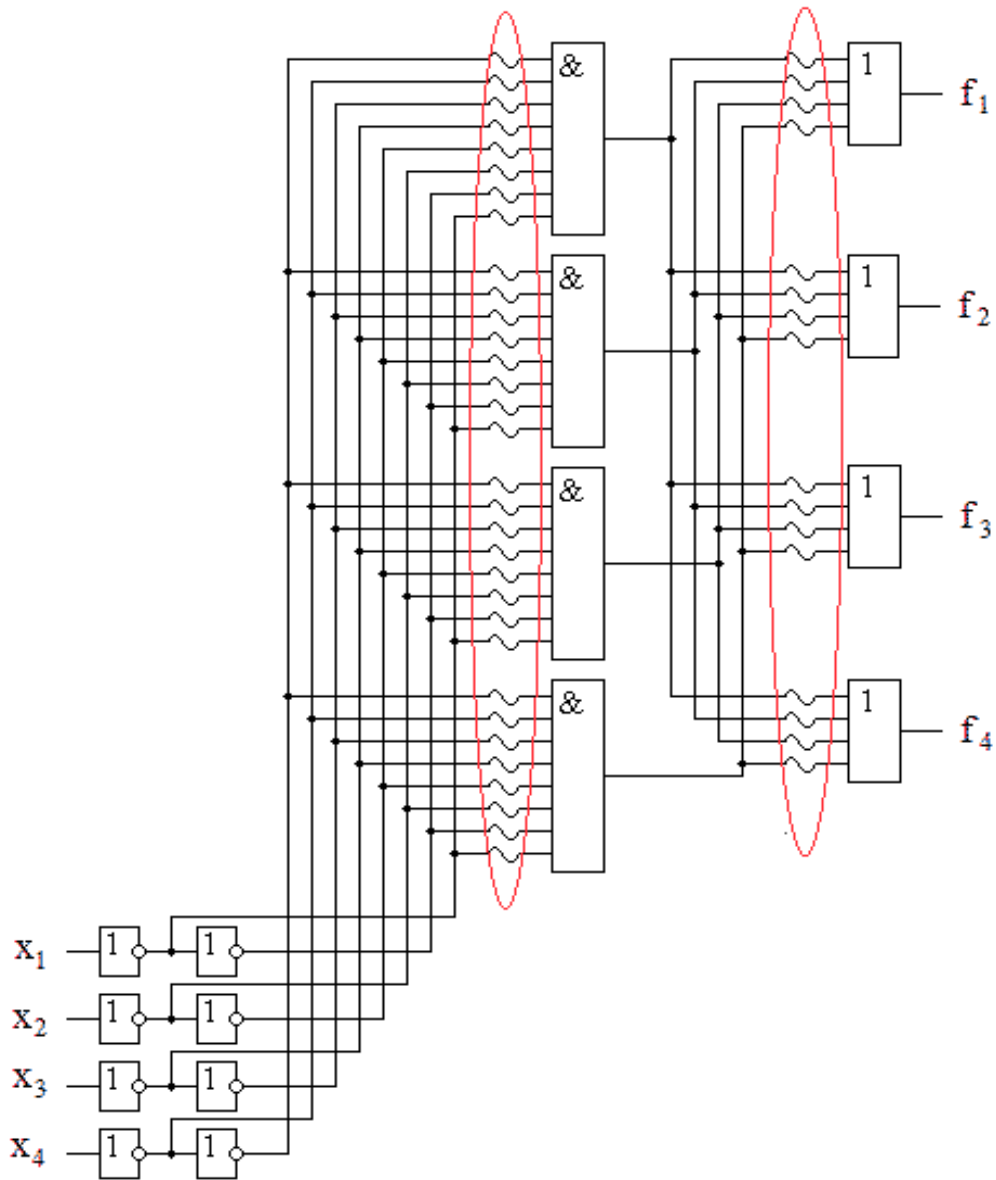


Рисунок 3.7 – Узагальнена структура ПЛМ

Вихідна частина схеми автомату базується на системі рівнянь виходів. Виконується процес, аналогічний до процесу у перехідній частині, але сигнали подаються на вихідну шину Y_t , після якої сигнал передається до вихідної програмованої логічної матриці.

На рис. 3.8 представлено функціональну схему операційного автомату як автомату Мілі, який обчислює кількість непарних елементів у двовимірному масиві $A[n, m]$, які належать інтервалу $[-5; 15]$. Функціональна схема також представлена і у Додатку В.

ПЛІС застосовується до створення певної заданої схеми цифрових інтегральних схем. Структура роботи ПЛІС визначається шляхом програмування за допомогою програматора та програмного забезпечення.

У більшості випадків мікросхема ПЛІС складається з набору логічних блоків, що перебудовуються, які реалізують необхідну логічну функцію; програмованих перемикачів, які здійснюють необхідне з'єднання логічних осередків; програмованих блоків введення/виводу, що забезпечують зв'язок зовнішнього виведення мікросхеми із внутрішньою структурою ПЛІС. У сучасних ПЛІС часто бувають вбудовані додатково блоки пам'яті, блоки DSP (Digital Signal Processor, цифровий сигнальний процесор) або помножувачі, PLL (Phase Locked Loop, фазове автопідстроювання частоти) та інші компоненти.

Якщо розробляється певний проект, його розробник може, зазвичай, не враховувати особливості елементів, з яких складається ПЛІС. При цьому він описує бажані функції проекрованої мікросхеми або записує текст у системі програмного забезпечення (наприклад, мовами опису апаратури Verilog або VHDL). Необхідну схему конструює компілятор на основі відомої внутрішньої структури ПЛІС, який сам розміщає елементи схеми по наявних логічних блоках, що конфігуруються, і з'єднує ці блоки за допомогою наявних програмованих ліній зв'язку.

На даний момент існує безліч виробників мікросхем ПЛІС, і кожен з них має свою систему автоматизованого проектування (САПР). САПР Quartus II є багатофункціональним середовищем проектування, яке містить у собі набори утиліт, що дозволяють розробити, верифікувати та запрограмувати проект для обраного сімейства мікросхем ПЛІС.

Логічна схема, розроблена у програмному забезпеченні Quartus II, називається проектом. Проект може мати ієрархічну структуру або складатися з багатьох файлів-модулів, коли головний модуль містить кілька додаткових модулів, а кожен додатковий модуль включає ще кілька файлів. І тут головний модуль називається об'єктом верхнього рівня ієрархії. САПР Quartus II, крім

ієрархічної структури, зберігає у одному каталозі всю технічну інформацію, що стосується обраної мікросхеми (наприклад: призначення виводів, попередні варіанти трасування логічної схеми, конфігурація проекту тощо). Ці файли створюються в єдиній папці проекту, що зручно при копіюванні та перенесенні проектів до інших каталогів або інших комп'ютерів.

Quartus II дозволяє легко реалізувати необхідну логічну схему на програмованій логічній інтегральній схемі (ПЛІС).

САПР для проектування ПЛІС, а саме компілятор (синтезатор логіки та фіттер та асемблер) – це, можливо, найскладніша частина всієї ПЛІС технології. Компілятор повинен проаналізувати проект користувача (схеми та текстові описи на Verilog HDL або VHDL) і згенерувати нетлист (netlist) – список всіх елементів схеми та зв'язку між ними. Netlist має бути оптимізований - логічні функції потрібно мінімізувати, можливі дубльовані регістри потрібно видалити. Потім компілятор повинен вмістити всю логіку з netlist в наявну архітектуру ПЛІС. Це робить фітер (fitter). Він розміщує логічні елементи та виконує трасування зв'язків між ними (процес place and route). Складність полягає в тому, що той самий проект може бути розміщений в ПЛІС у різний спосіб, і цих способів мільйони. Деякі розміщення та трасування виявляються кращими, інші – гірші. Головний критерій якості отриманої системи – максимальна частота, на якій зможе працювати проект при даному розміщенні елементів та при даному трасуванні зв'язків. Тут впливають довжина зв'язків між логічними блоками та кількість програмованих комутаторів між ними.

Компілятор, знаючи архітектуру ПЛІС, за результатами роботи додатково видає звіт про час проходження сигналів від регістру до регістру. Ця інформація часто є корисною для розробника високопродуктивних систем. Розробник для ПЛІС може давати деякі поради компілятору: де, в якому місці кристала краще розмістити той чи інший модуль проекту.

Вибираючи для свого проекту конкретну модель мікросхеми ПЛІС, розробник певною мірою потрапляє у залежність від виробника цієї ПЛІС, тому

					КВРКІ 180114.18.01.04 ПЗ	Арк. 50
Зм.	Арк.	№ докум.	Підпис	Дата		

що повинен у роботі користуватися програмним забезпеченням від цього виробника.

Програмне забезпечення компанії Альтера: Quartus II. ПЗ Xilinx для проектування для ПЛІС: ISE Suite, Vivaldo Design Suite. ПЗ компанії Microsemi: Libero IDE, Libero SoC.

Програмне забезпечення, компілятори для ПЛІС – це одна з найважливіших складових інтелектуальної власності компаній-виробників ПЛІС.

Проектування в середовищі автоматизованого проектування (САПР) Quartus II включає такі етапи:

1) введення проекту (Design Entry) – бажана схема проекту задається або графічним способом, або з використанням мов опису апаратних засобів, таких як Verilog HDL, VHDL та ін.;

2) синтез (Synthesis) – проект, що вводиться, синтезується в схему, яка складається з логічних елементів (ЛЕ) і логічних блоків (ЛБ) в мікросхемі ПЛІС;

3) функціональне моделювання (Functional Simulation) – схема, що синтезується, тестується на предмет коректності функціонування у вбудованому симуляторі, який моделює залежність стану (вибраних розробником) сигналів схеми від часу; це моделювання не враховує тимчасових затримок сигналів (званих логічними перегонами) між ЛЕ/ЛБ мікросхеми ПЛІС;

4) трасування (Fitting) – інструмент трасування САПР обчислює оптимальне розміщення та з'єднання ЛЕ/ЛБ, визначених у списку з'єднань реальної мікросхеми ПЛІС; трасувальник також обирає дроти або «маршрут руху» у чіпі для реалізації необхідних зв'язків між заданими ЛЕ/ЛБ;

5) тимчасовий аналіз (Timing Analysis) – аналізує затримки розповсюдження сигналів уздовж різних шляхів у трасованій логічній схемі для обчислення наявності/відсутності логічних перегонів, щоб сигнали з різних ЛБ приходили одночасно в той чи інший кінцевий вузол схеми;

6) тимчасове моделювання (Timing Simulation) – схема тестується для перевірок функціональності та тимчасових обмежень, але, на відміну від

функціонального моделювання, тут для виявлення наявності або відсутності логічних перегонів враховуються реальні затримки вибраних сигналів;

7) програмування та конфігурація (Programming and Configuration) - розроблена схема розміщується в мікросхемі ПЛІС шляхом програмування електронних зв'язків між конфігурованими ЛЕ/ЛБ, що реалізується шляхом передачі конфігураційного файлу з комп'ютера або мікросхему ПЛІС, або в додаткову (не вбудовану в ПЛІС) пам'ять, якщо така є в налагоджувальному комплекті.

За допомогою Block Diagram/Schematic File складемо схему до розроблюваного автомату Мілі, яка успішно пройшла етап компіляції. Модель операційного автомату на основі автомату Мілі в середовищі Quartus II представлена на рис. 3.9.

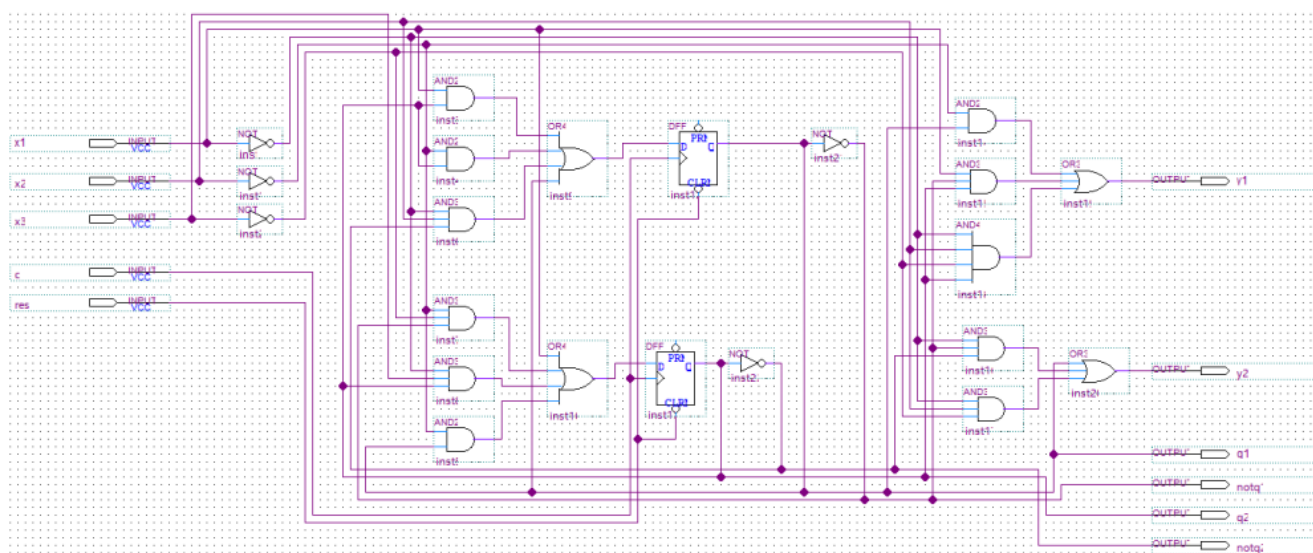


Рисунок 3.9 – Модель операційного автомату на основі автомату Мілі в середовищі Quartus II

Далі дослідимо роботу автомату з використанням засобу середовища Quartus II для перегляду станів і сигналів Simulation Waveform Editor – рис. 3.10.

3.4 Висновки

В третьому розділі кваліфікаційної роботи виконано реалізацію та моделювання роботи операційного автомату на основі автомату Мілі, а саме:

1) застосовано схему Уілкса-Стрінжера для реалізації операційного автомату на основі автомату Мілі;

2) реалізовано операційний автомат на основі автомату Мілі, який обчислює кількість непарних елементів у двовимірному масиві $A[n, m]$, які належать інтервалу $[-5; 15]$;

3) змодельовано роботу операційного автомату на основі автомату Мілі в середовищі Quartus II.

					КВРКІ 180114.18.01.04 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

ВИСНОВКИ

Кінцеві автомати наразі використовуються в комп'ютерних іграх та в системах стиснення інформації, а також при реалізації мережевих протоколів. Кінцеві автомати застосовуються при розв'язанні задач, де наявна складна логіка та необхідна висока надійність, які важко реалізуються програмістом.

Для таких задач зручно використовувати кінцеві автомати (автомати Мілі та автомати Мура). Завдяки використанню ними апарату булевої алгебри та теорії графів, завдяки можливості їх графічного подання та завдяки їх детермінованій поведінці автомати мають серйозні переваги.

Будь-який операційний пристрій має у своєму складі операційний автомат та керуючий автомат.

Операційний автомат виконує операції над інформацією і надає результати цих операцій. Він може мати у своєму складі суматори, регістри, мультиплексори, арифметико-логічний пристрій, дешифратори та інші функціональні блоки для виконання команд (додавання двох чисел, запис в регістр, зсув коду праворуч або ліворуч, дешифрування двійкового коду, інвертування вмісту регістра, тощо).

Синтезована структура операційного автомату не повинна містити обмежень на сумісність мікрооперацій, тобто одночасно потрібно виконувати функціонально сумісні мікрооперації.

Автомат Мілі (Mealy) – автомат, в якому існує залежність вихідного сигналу автомата як від вхідного сигналу, так і від його поточного внутрішнього стану.

Автомат Мура (Moore) – автомат, в якому існує залежність вихідного сигналу автомата лише від його поточного внутрішнього стану.

В результаті виконання кваліфікаційної роботи було спроектовано, реалізовано та змодельовано роботу операційного автомату, що виконує обчислення кількості непарних елементів у двовимірному масиві $A[n, m]$, які належать інтервалу $[-5; 15]$. Використано схему Уілкса-Стрінжера для реалізації

					КВРКІ 180114.18.01.04 ПЗ	Арк.
						56
Зм.	Арк.	№ докум.	Підпис	Дата		

мікропроцесорного автомату як автомату Мілі. Використано логічні елементи, тригери для реалізації та синтезу функціональної схеми автомату.

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

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

Для досягнення поставленої мети використано такі методи дослідження, як методи синтезу, аналізу та моделювання процесів, принципи системного аналізу, теоретико-множинні підходи.

В першому розділі кваліфікаційної роботи виконано дослідження предметної області (проаналізовано принципи дії операційних та керуючих автоматів, особливості автоматів Мілі та Мура, а також принципи розроблення функціональної та принципової електричних схем). Крім цього, в першому розділі виконано постановку задачі подальшого дослідження.

В другому розділі кваліфікаційної роботи проведено моделювання та проектування операційного автомату на основі автомату Мілі, а саме виконано: формалізований опис операційного автомату; розроблення змістовної схеми алгоритму; кодування верхівок змістовної схеми алгоритму; розроблення закодовану мікроопераційну схему алгоритму; кодування мікрокоманд; розроблення закодованої мікрокомандної схеми алгоритму; розроблення основної таблиці автомату Мілі; проектування граф-схеми автомату Мілі; розроблення системи рівнянь переходів; розроблення системи рівнянь виходів; визначення необхідної кількості елементів пам'яті (тригерів) для реалізації автомату Мілі; кодування внутрішніх станів автомату; розроблення структурної схеми операційного автомату; розроблення схему операційного автомату.

В третьому розділі кваліфікаційної роботи виконано реалізацію та моделювання роботи операційного автомату на основі автомату Мілі, а саме: застосовано схему Уілкса-Стрінжера для реалізації операційного автомату на основі автомату Мілі; реалізовано операційний автомат на основі автомату Мілі,

					КВРКІ 180114.18.01.04 ПЗ	Арк. 57
Зм.	Арк.	№ докум.	Підпис	Дата		

який обчислює кількість непарних елементів у двовимірному масиві $A[n, m]$, які належать інтервалу $[-5; 15]$; змодельовано роботу операційного автомату на основі автомату Мілі в середовищі Quartus II.

Практичне значення має спроектований, змодельований та реалізований операційний автомат на основі автомату Мілі, який використовується для обчислення кількості непарних елементів у двовимірному масиві $A[n, m]$, які належать інтервалу $[-5; 15]$.

					КВРКІ 180114.18.01.04 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Джон Хопкрофт, Раджив Мотвани, Джеффри Ульман. Введение в теорию автоматов, языков и вычислений. 2-е издание. Москва: Диалектика-Вильямс, 2016. 528 с.
2. Говорущенко Т. О. Комп'ютерна логіка: практикум: навчальний посібник. Хмельницький: Хмельницький національний університет, 2018. 294 с.
3. Матвієнко М.П. Комп'ютерна логіка: навчальний посібник. Київ: ТОВ "Центр навчальної літератури", 2012. 288 с.
4. Зацерковний В.І. Обчислювальна техніка: історія розвитку від найпростіших пристроїв для лічби до електромеханічних комп'ютерів: монографія. Ніжин: Аспект-Поліграф, 2012. 416 с.
5. 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.
6. Дж. Фон Нейман. Теория самовоспроизводящихся автоматов. Москва: ВАН, 2010. 673 с.
7. Матвієнко М.П. Архітектура комп'ютерів: навчальний посібник. Київ: ТОВ "Центр навчальної літератури", 2012. 264 с.
8. Матвієнко М.П. Комп'ютерна схемотехніка: навчальний посібник. Київ: ТОВ "Центр навчальної літератури", 2012. 190 с.
9. Salcido A. Cellular Automata - Simplicity Behind Complexity. InTech, 2011. 580 pp.
10. Кононюк А.Е. Дискретно-непрерывная математика. Книга 11. Автоматы. Часть 2. Детерминированные автоматы. Киев: Освіта України, 2017. 578 с.
11. 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.

					КВРКІ 180114.18.01.04 ПЗ	Арк.
						59
Зм.	Арк.	№ докум.	Підпис	Дата		

12. 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.

13. 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.

14. 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.

15. 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.

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

17. 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.

18. 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.

19. 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.

20. 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.

21. 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.

22. 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.

23. 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.
24. 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.
25. 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.
26. 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.
27. 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.
28. Paulu F., Hospodka J. Web-based Application for Analysis of Electrical Circuits and Systems. *Proceedings of the International Conference on New Trends in Signal Processing.* 2018. Pp. 147-150.
29. Lucero B., Adams M., Turner C. Introduction to quantitative engineering design methods via controls engineering. *AI Edam-Artificial Intelligence for Engineering Design Analysis and Manufacturing.* 2017. Vol. 31. Issue 4. Pp. 458-475.
30. Carranza G., Robles U., Valle C., Gutierrez J., Rumpf R. Design and Hybrid Additive Manufacturing of 3-D/Volumetric Electrical Circuits. *IEEE Transactions on Components Packaging and Manufacturing Technology.* 2019. Vol. 9. Issue 6. Pp. 1176-1183.
31. Kukul T., Mathur A., Ahuja J., Mohan S. Electrical Overstress Estimation for Printed Circuit Board Design. *Reliability and Maintainability Symposium.* 2020. Pp. 211-215.
32. Kaiser J., Bernasch J., Baron K., Ehlers J. Model based design and evaluation of E/E systems. *VDI Electronic Systems for Vehicles.* 2007. Vol. 2000. Pp. 245-263.
33. Glicksman H. White is green: New schematic diagrams. *Proceedings of the Society of Photo-Optical Instrumentation Engineers.* 2020. Vol. 4421. Pp. 508-511.

					КВРКІ 180114.18.01.04 ПЗ	Арк.
						61
Зм.	Арк.	№ докум.	Підпис	Дата		

34. Matsubayashi M., Watanabe S. Generating two-dimensional schematic diagrams of mechanical, electrical, and plumbing systems from three-dimensional building information models. *International Journal of Architectural Computing*. 2016. Vol. 14. Issue 3. Pp. 219-232.

35. Matsubayashi M., Watanabe S. Generating Schematic Diagrams of Mep Systems from 3D Building Information Models for Use in Conservation. *Emerging Experiences in the Past, Present and Future of Digital Architecture*. 2015. Pp. 293-302.

36. Zhang J., Ji C. An easy way of generating operative schematic diagrams. *International Journal of Surgery*. 2019. Vol. 7. Issue 2. Pp. 171-172.

37. Chivers D., Rodgers P. Improving Search-Based Schematic Layout by Parameter Manipulation. *International Journal of Software Engineering and Knowledge Engineering*. 2016. Vol. 25. Issue 6. Pp. 961-991.

38. Zhao H., Huang S., Zhou Y., Wang R. Schematic Diagrams in Second Language Learning of English Prepositions a Behavioral and Event-Related Potential Study. *Studies in Second Language Acquisition*. 2020. Vol. 42. Issue 4. Pp. 721-748.

39. Ozogul G., Johnson A., Moreno R., Reisslein M. Technological Literacy Learning with Cumulative and Stepwise Integration of Equations into Electrical Circuit Diagrams. *IEEE Transactions on Education*. 2012. Vol. 55. Issue 4. Pp. 480-487.

40. Bhattacharya A., Roy S., Sarkar N., Malakar S., Sarkar R. Circuit Component Detection in Offline Hand-drawn Electrical/Electronic Circuit Diagram. *Proceedings of 2020 IEEE Calcutta Conference*. 2020. Pp. 80-84.

41. Johnson A., Butcher K., Ozogul G., Reisslein M. Introductory Circuit Analysis Learning from Abstract and Contextualized Circuit Representations: Effects of Diagram Labels. *IEEE Transactions on Education*. 2014. Vol. 57. Issue 3. Pp. 160-168.

42. Deskur J. Models of magnetic circuits and their equivalent electrical diagrams. *Compel - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*. 2019. Vol. 18. Issue 4. Pp. 600-610.

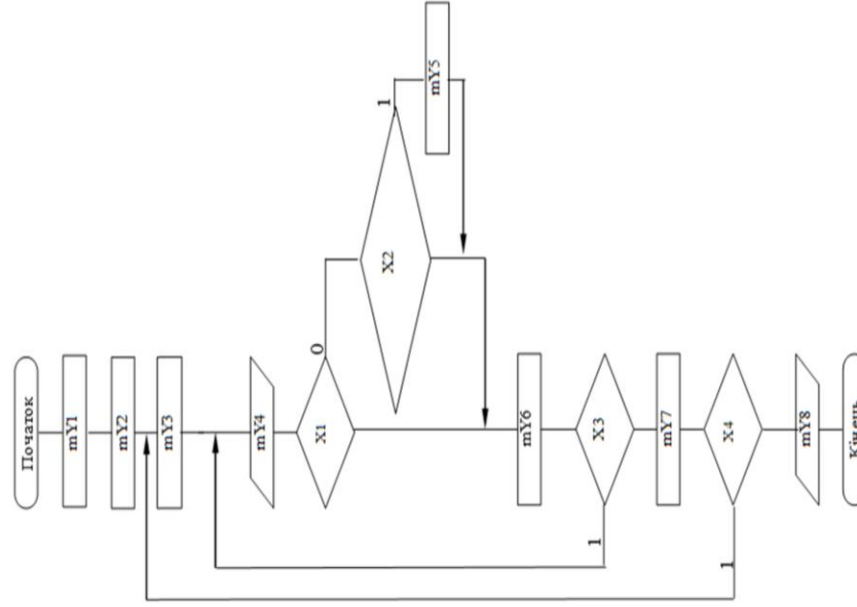
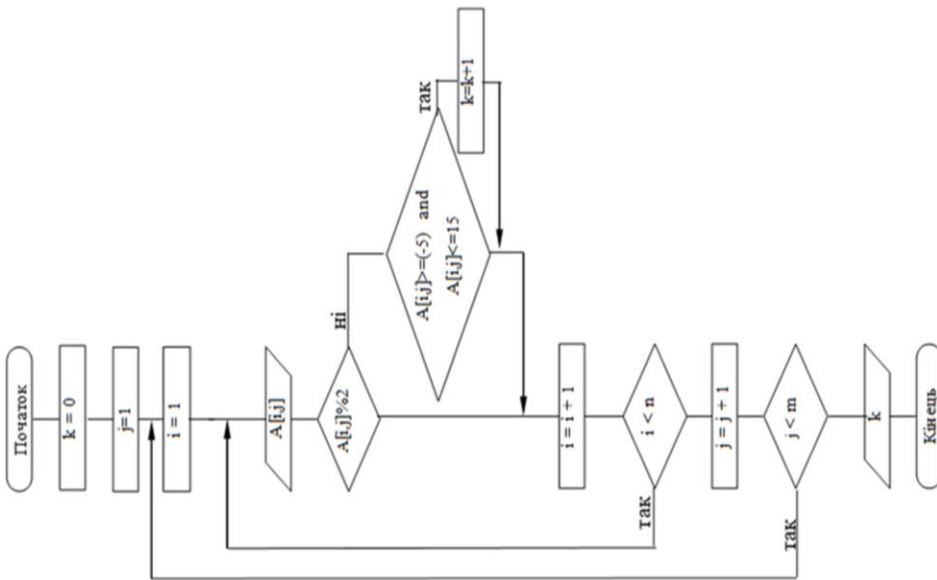
43. DesPortes K., Anupam A., Pathak N., DiSalvo B. Circuit Diagrams Vs. Physical Circuits: The Effect of Representational Forms During Assessment. *2016 IEEE Frontiers in Education Conference*. 2016. Pp. 103-109.

					КВРКІ 180114.18.01.04 ПЗ	Арк. 62
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток А (обов'язковий)

Копія креслення «Логічні схеми алгоритмів»

Логічні схеми алгоритмів



КвРКІ 180114.18.01.04

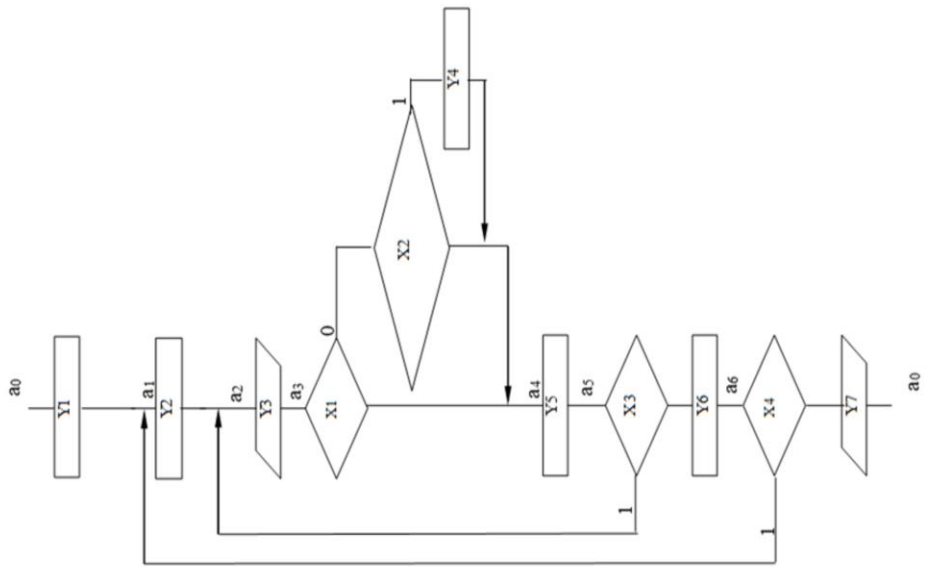
КвРКІ 180114.18.01.04			
Літера	Маса	Масштаб	
Список та модальності спеціального апарату на основі платформи МІА			
Знак	Аркуш	№ докум.	Підпис / Дата
Розроб.	Літка М.С.		
Перевір.	Бережний О.М.		
Н. контр.			
Т. контр.	Лисенко С.М.		
Дата.	Позв'язаний І.С.		
		ХНУ, гр. КІ-18-1	

Додаток Б
(обов'язковий)

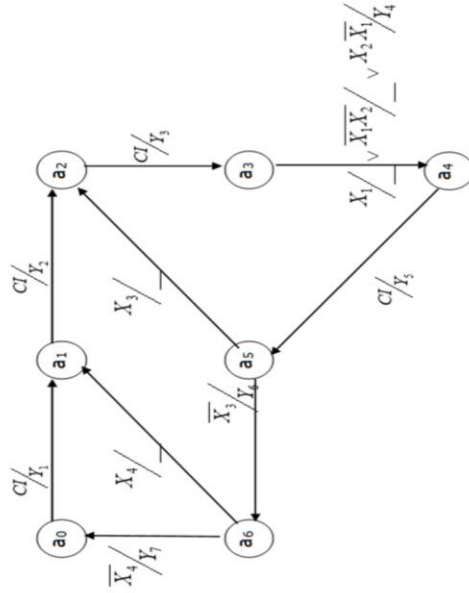
Копія креслення «Мікрокомандна схема та граф-схема переходів автомату Мілі»

КВРКІ 180114.18.01.04

Мікрокомандна схема алгоритму



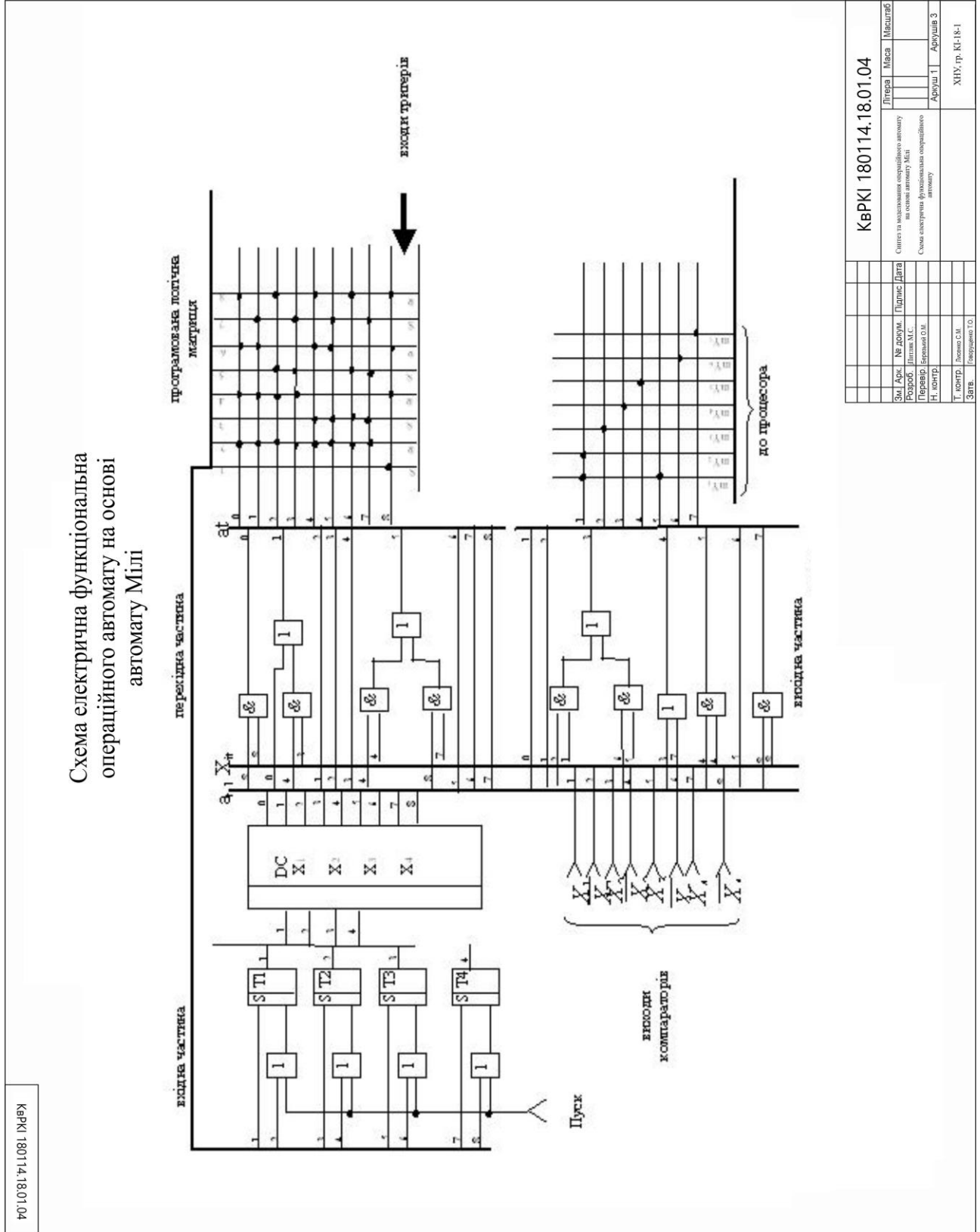
Граф-схема автомату Мілі



КВРКІ 180114.18.01.04		Підпис	Дата
Зм. Авт.	№ докум.	Підпис	Дата
Розроб.	Інформац.		
Н. контр.	Лекція С.М.		
Завт.	Спеціаліст Т.О.		
Сторінка		Місяць	Місяць
Мікрокомандна схема алгоритму та граф-схема автомату Мілі		Арсен 1	Арсен 3
		ХНУ, р. КІ-8-1	

Додаток В (обов'язковий)

Копія креслення «Схема електрична функціональна операційного автомату»



КВРКІ 180114.18.01.04		Літера	Масштаб
Зм.	Арх.	№ докум.	Підпис
Розроб.	Літвин М.С.	Синтез та моделювання операційного автомату на основі автомату Мілі	
Н. контр.	Перевір.	Березний О.М.	Схема електрична функціональна операційного автомату
Т. контр.	Лозинко С.М.	Армуш 1	Армуш 3
Затв.	Говорченко О.О.	ХНУ, гр. КІ-18-1	



Ім'я користувача:
Кафедра КІ

Дата перевірки:
25.05.2022 08:06:25 EEST

Дата звіту:
25.05.2022 08:11:20 EEST

ID перевірки:
1011328880

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005591

Назва документа: Питляк Синтез та моделювання операційного автомату на основі автомату Мілі

Кількість сторінок: 59 Кількість слів: 9312 Кількість символів: 70359 Розмір файлу: 1.44 MB ID файлу: 1011215016

16.3% Схожість

Найбільша схожість: 6.65% з джерелом з Бібліотеки (ID файлу: 1008214668)

7.79% Джерела з Інтернету

152

Сторінка 61

10.5% Джерела з Бібліотеки

85

Сторінка 63

0% Цитат

Не знайдено жодних цитат

Не знайдено жодних посилань

0% Вилучень

Немає вилучених джерел

Модифікації

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

Замінені символи

1

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 2.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 7%**

ID: 103919 Название: Синтез та моделювання операційного автомату на основі автомату Мілі Добавлено в БД: 2022-05-25 Авторы: М. С. Питляк Руководители: О. М. Березький Консультанты: Опоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	66774	483	2408 (4%)	29 (6%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Питляк Максим Сергійович

Тема: Синтез та моделювання операційного автомату на основі автомату Мілі

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 62

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є синтез та моделювання операційного автомату на основі автомату Мілі.
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 принципи дії операційних та керуючих автоматів, особливості автоматів Мілі та Мура, а також схем електричних функціональної і принципової) та виконано постановку задачі дослідження. В другому розділі кваліфікаційної роботи проведено моделювання та проектування операційного автомату на основі автомату Мілі, 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 Мілі; спроектовано граф-схему переходів автомату Мілі; складено систему рівнянь переходів; визначено необхідну кількість елементів пам'яті (тригерів) для реалізації автомату Мілі; закодовано внутрішні стани автомату; розроблено структурну схему операційного автомату; побудовано схему операційного автомату. В третьому розділі кваліфікаційної роботи виконано реалізацію та моделювання операційного автомату на основі автомату Мілі.

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 інженерно-технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре (4.50/В)

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

Марченко Валерій Володимирович,
зав. каф. АІІТ, д.т.н., проф.

"26" 05 2022 р.

 (підпис)

Завідувачу кафедри КІС
д-р.техн.наук, проф. Говорущенко Т. О.

Питляка Максима Сергійовича

ІІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ-18-1

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 плагіатоповіщений (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 (Unicheck 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.

24 травня 2022 року



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОМАЦІЙНИХ СИСТЕМ
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 Мілі

Автор: Питляк Максим Сергійович

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 укріття 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 розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) найбільшу схожість встановлено з одним документом і становить вона 6.65% в частині загальноприйнятої термінології.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 16.3% і адресується до 63 першоджерел, що, з урахуванням наведених обґрунтувань, відповідає характеру дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

О. М. Березький

Гарант ОПП

С. М. Лисенко

Завідувач кафедри КІІС

Т. О. Говорущенко