

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

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Синтез та моделювання операційного автомату на основі автомату Мілі
Назва теми

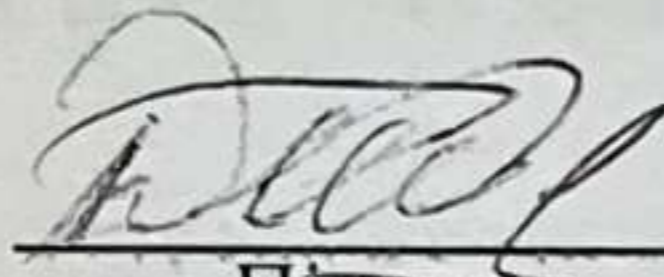
КвРКІ.0170159.17.01.01 ПЗ
Шифр

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 КІ-17-1


Підпис

Д.О. Барчук
Ініціали, прізвище

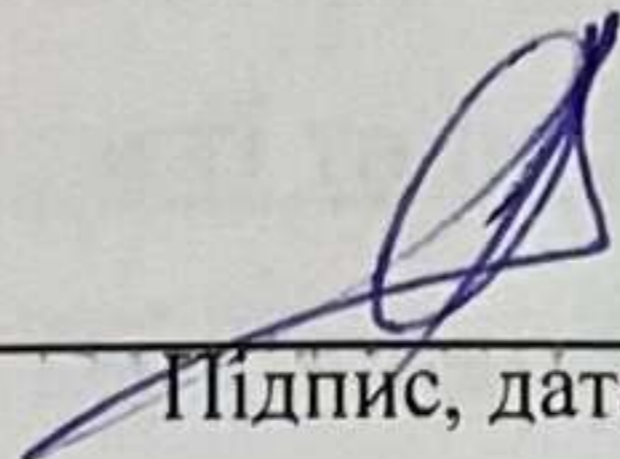
Керівник



Підпис, дата

О.М. Березький
Ініціали, прізвище

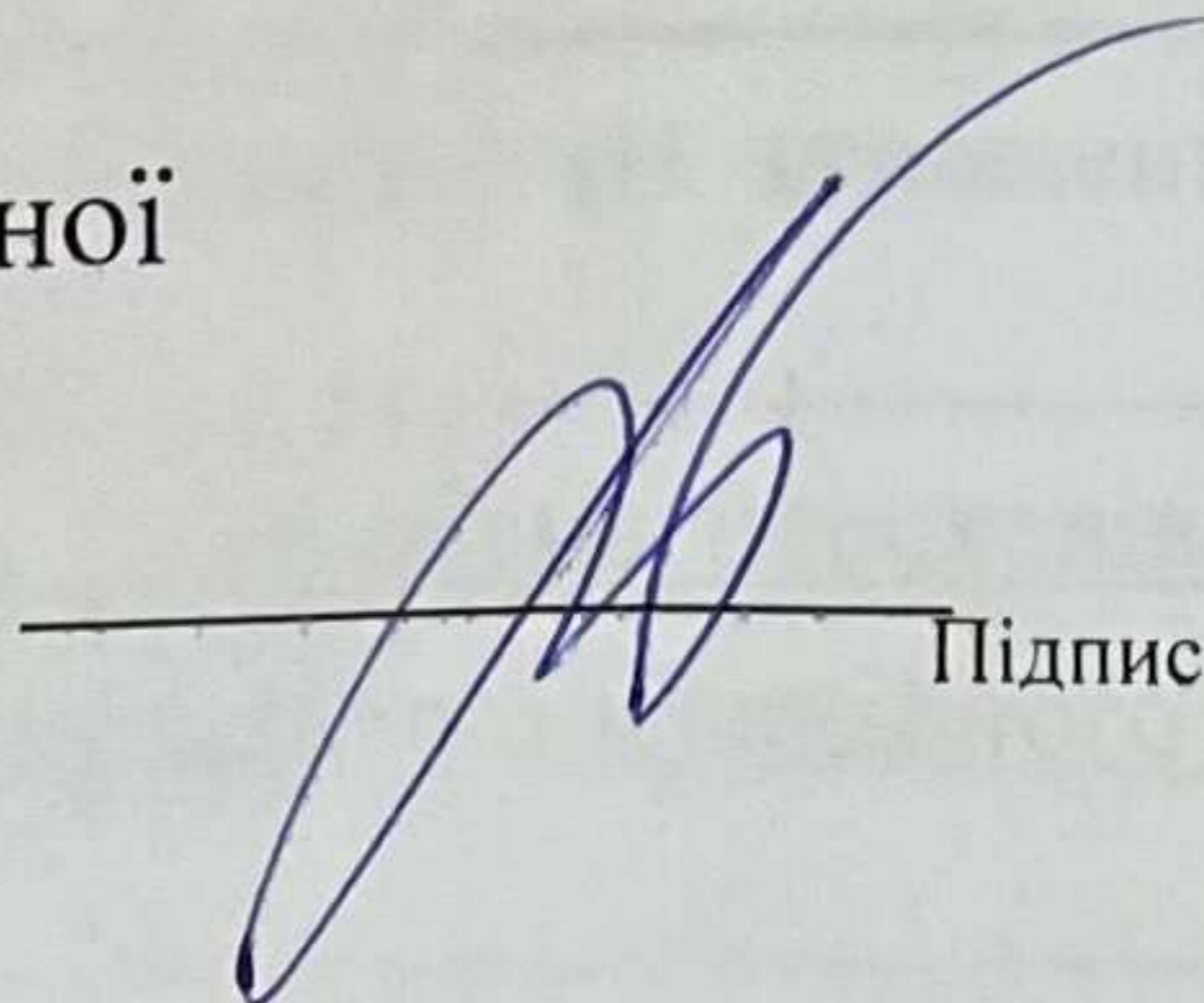
Нормоконтролер



Підпис, дата

С.М. Лисенко
Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
Інженерії та системного
Програмування



Підпис

Т.О. Говорущенко

Ініціали, прізвище

«29» червня 2021 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ПРОГРАМУВАННЯ ТА КОМП'ЮТЕРНИХ І ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА СИСТЕМНОГО ПРОГРАМУВАННЯ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЯ ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

“ 11 ” 01 2021 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Барчуку Денису Олександровичу

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 05.02.2021 р. № 11

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

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

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

Дослідження предметної області та постановка задачі

Моделювання та проектування операційного автомату на основі автомату Мілі

Апаратна реалізація операційного автомату на основі автомату Мілі

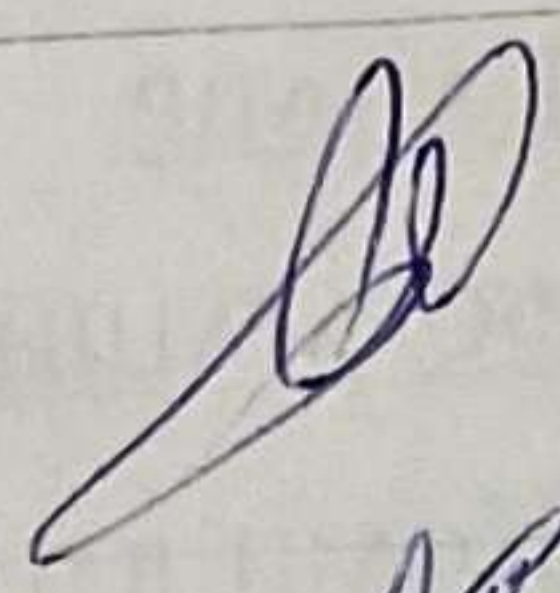
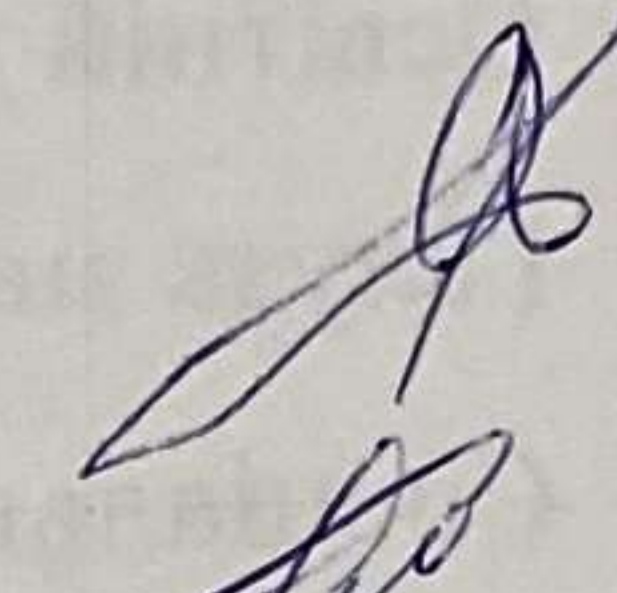
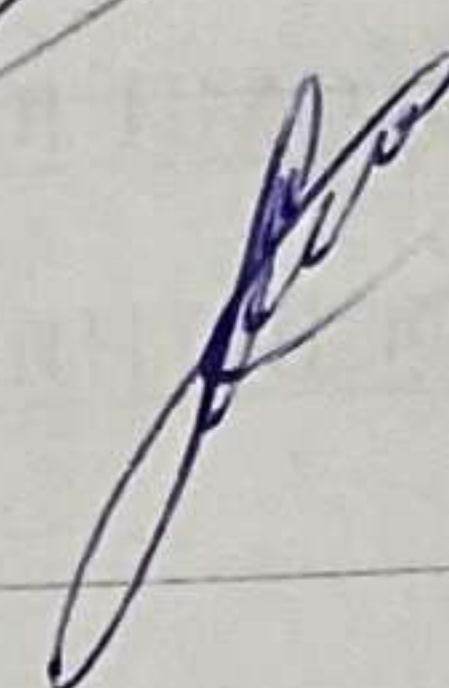
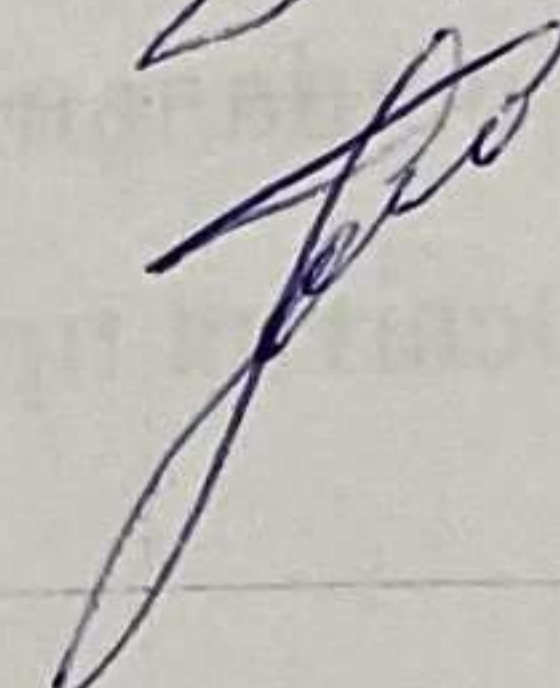
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

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 при
Нормоконтроль	Лисенко С.М., професор кафедри КІСП		
Антиплагіат	Нічепорук А.О., доцент кафедри КІСП		

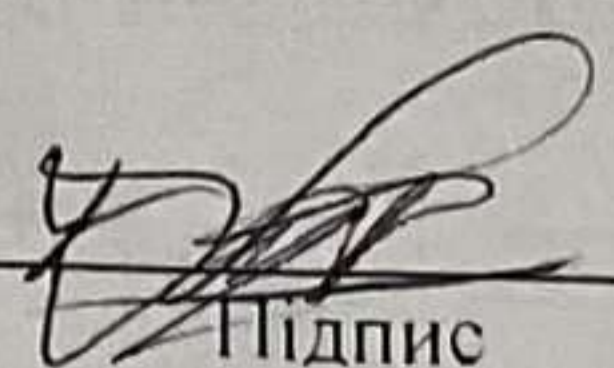
7. Дата видачі завдання « 11 » 01 2021 р.


КАЛЕНДАРНИЙ ПЛАН

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

Студент

Керівник проекту (роботи)


Підпис


Підпис

Д.О. Барчук
Ініціали, прізвище

О. М. Березький
Ініціали, прізвище

Ф о р м а т	Позначення	Найменування	К і л · л и с т і в	№ ек з	П р и м і т к а
		Текстові документи			
	КВРКІ 170159.17.01.01 ПЗ	Пояснювальна записка	78		
		Графічні матеріали			
	КВРКІ 170159.17.01.01 Е8	Логічні схеми алгоритмів	1		
	КВРКІ 170159.17.01.01 Е8	Мікрокомандна схема	1		
		та граф-схема переходів			
		автомату Мура			
	КВРКІ 170159.17.01.01 Е2	Схема електрична	1		
		функціональна			
		операційного автомату			

КВРКІ 170159.17.01.01 ВП				
Ар к	№ докум	Підп ис	Дат а	
Зробив	Барчук Д.О.			Відомість проекту
Перевір.	Березький О.М.			
Н. онтр.	Лисенко С.М.			
Затв.	Говорущенко Т.О.		29.06	
				Літера
				У
				Арку ш
				1
				Аркуш ів
				1
				ХНУ, КІ-17-1

АНОТАЦІЯ

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

Автор роботи: Барчук Денис Олександрович.

Керівник роботи: Березький О.М.

Пояснювальна записка: 78 с., 19 рис., 5 табл., 4 дод., 47 джерел.

Графічна частина: 10 презентаційних слайдів.

ПРОГРАМОВАНА ЛОГІЧНА ІНТЕГРАЛЬНА СХЕМА, ЦИФРОВА ІНТЕГРАЛЬНА СХЕМА, СИСТЕМА АВТОМАТИЗАЦІЇ, VHDL.

Метою роботи є розробка ЦА для обчислення добутку непарних елементів.

У цій роботі розроблено операційний пристрій для розв'язання наступної задачі: обчислення добутку непарних елементів двовимірного масиву $A[n, m]$, що належать інтервалу $[0; 100]$. Складено мікропрограму операційного пристрою, до складу якого входить керуючий автомат, у вигляді змістовної та кодованої ГСА. Виконано абстрактний та структурний синтез керуючого автомату як автомату Мілі.

Підпис студента

Дата 29.06.21

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	4
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	6
1.1 Аналіз задачі, обґрунтування вибору моделі життєвого циклу для реалізації проекту	6
1.2 Обґрунтування вибору апаратних ресурсів, мови програмування	6
1.3 Кінцевий автомат	15
1.4 Автомат Мілі	21
1.5 Автомат Мура	23
2 ПРОЄКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ	26
2.1 Аналіз предметної області і виявлення наявних проблем і завдань	26
2.2 Порівняльний аналіз переваг та недоліків існуючих рішень	26
2.3 Методологічні підходи до вирішення задачі за темою дослідження	27
2.4 Постановка задачі	31
2.5 Абстрактний синтез автомату Мілі	36
2.6 Висновки	44
3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ	45
3.1 Схема автомату Мілі	45
3.2 Опис апаратної реалізації керуючого автомату на основі автомату Мілі	47
3.3 Моделювання керуючого автомату на основі автомату Мілі в середовищі Quartus II	52
3.4 Висновки	59
ВИСНОВКИ	60
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	61

КвРКІ. 170159.17.01.01 ПЗ

Зм	Арк.	Недокум.	Підпис	Дата		Літера	Аркуш	Аркушів
Викона		Барчук Д.О.			Тема: «Синтез та моделювання керуючого автомату на основі автомату Мілі».			
Перевір.		Березький О.М.						
Н.контр		Лисенко С.М.						
Затвер		Горбухін		29.06				
						ХНУ, КІ-17-1		

Додаток А Логічні схеми алгоритмів.....	60
Додаток Б Мікрокомандна схема та граф-схема переходів автомату Мілі	61
Додаток В Схема електрична функціональна операційного автомату.....	62
Додаток Г Перелік елементів.....	63
Додаток Д Результати роботи програми.....	67

					КвРКІ. 170159.17.01.01 ПЗ	Ар
						3
Зм..	Арк.	Нодокум.	Підпис	Дата		

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ЦА - цифровий автомат

ПЛІС - Програмована логічна інтегральна схема

КА – кінцевий автомат

ТПВ - таблиці переходів і виходів.

ГСА – граф-схема алгоритма

ЕОМ – електронно - обчислювальна машина

ГА – граф автомата

ПЗ - програмне забезпечення

ОС - операційна система

					КвРКІ. 170159.17.01.01 ПЗ	Ар
						4
Зм..	Арк.	№докум.	Підпис	Дата		

ВСТУП

Електронні пристрої роблять значний вплив на життя сучасної людини. З різними електронними пристроями людина так чи інакше взаємодіє кожен день. Без деяких з них, неможлива повсякденне життя людей. Прикладом застосування цифрових автоматів в повсякденному житті: світлофори, термінали, обчислювальна техніка, побутова техніка. Очевидним є факт швидкого розвитку галузі і напрямком розробки автоматів різного призначення є перспективним.

Мета даної роботи розробка ЦА для обчислення добутку непарних елементів двовимірного масиву.

Актуальність теми роботи полягає у необхідності детального розгляду синтезу та моделювання автомату Мілі, що дозволить більш глибоко зрозуміти основні принципи роботи даного винаходу.

Завданням роботи є:

- 1) розглянути існуючі рішення;
- 2) описати математичну модель і змодельовати автомат Мілі;
- 3) розробити мікропрограму операційного пристрою провести абстрактний та структурний синтез.

					КвРКІ. 170159.17.01.01 ПЗ	Ар
						5
Зм..	Арк.	Нодокум.	Підпис	Дата		

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз задачі, обґрунтування вибору моделі життєвого циклу для реалізації проекту

Цифровий автомат - це логічний пристрій, здатний перебувати в одному з декількох стійких станів, здійснювати обробку, зберігання та отримання дискретної інформації відповідно до деякого алгоритму

Такі пристрої вже давно широко використовуються не тільки в різних галузях промисловості, а й в побуті та уявити життя сучасної людини без таких засобів допомоги дуже важко. За допомогою цифрових автоматів працюють: автомати з продажу квитків(жетонів), світлофори, станки та роботизовані механізми на виробництвах, різноманітна обчислювальна техніка і тд. Зважаючи на тенденцію розвитку різноманітних допоміжних пристроїв до автоматизованих виникає попит на розробку засобів керування цими пристроями. Високим попитом користуються автомати для обчислення масивів чисел. ЦА мають обмежену кількість можливостей та кожен проектується для вирішення конкретної задачі тому в цій галузі є безліч не вирішених завдань. Відповідно до завдання проаналізуємо наявність готових рішень для знаходження добутку непарних елементів двовимірного масиву. Пошук готових рішень даної проблеми завершився безрезультатно, тобто засобів для розв'язку такої задачі не виявлено.

Завданням роботи є:

- 1) розглянути існуючі рішення;
- 2) описати математичну модель і змодельювати автомат Мілі;
- 3) провести абстрактний та структурний синтез ;
- 4) розробити мікропрограму операційного пристрою.

					КВРКІ. 170159.17.01.01 ПЗ	Ар
						6
Зм..	Арк.	№докум.	Підпис	Дата		

1.2 Обґрунтування вибору апаратних ресурсів, мови програмування

MPLAB v5.70.40 – це розширювана, конфігурувана програмна програма, яка включає потужні інструменти, що дозволяють дизайнерам виявляти, налаштовувати, розробляти, налагоджувати та кваліфікувати вбудовані конструкції для більшості мікроконтролерів Microchip (MCU) та цифрових контролерів сигналів (DSC). MPLAB X IDE безперерійно працює з екосистемою розробки MPLAB програмного забезпечення та інструментів, багато з яких абсолютно безкоштовні. Гнучкий та настроюваний інтерфейс MPLAB X IDE дозволяє одночасно підключати до комп'ютера кілька інструментів налагодження. Ви можете вибрати будь-який інструмент, який ви бажаєте для конкретного проекту або конфігурації в рамках проекту. Завдяки повному управлінню проектами, візуальним графікам дзвінків, настроюваному віконцю перегляду та багатофункціональному редактору, який включає в себе завершення коду та навігацію гіперпосиланнями, MPLAB X IDE повністю обладнаний для задоволення потреб досвідчених користувачів, залишаючись гнучким та зручним для користувачів навіть ті, хто новачок в IDE.

Altium Designer - комплексна система автоматизованого проектування (САПР) радіоелектронних засобів, розроблена австралійською компанією Altium. Раніше ця ж фірма розробляла САПР P-CAD, яка приобріла неабічайну популярність серед російських розробників електроніків. У 2008 році фірма Altium заявила про прекрасні вставки програмних пакетів P-CAD та запропонувала розробникам використовувати програму Altium Designer, яка з'явилася в 2000 році та спочатку назвала назву Protel. У 2006 році був розроблений ребрендинг програмного продукту і він отримав поточне назва, остання версія якого називається Altium Designer 20.

Редактор плат FreePCB розрахований на спільну роботу з редактором схеми TinyCAD. Крім цього, можлива пряма трансляція списку з'єднань із симулятора LTspice в програмі FreePCB. Появляється можливість на базі безкоштовних програм зібрати повноцінну систему сквозного проектування, що містить симулятор, редактор електричних схем та розвідник друкованих плат.

					КВРКІ. 170159.17.01.01 ПЗ	Ар
						7
Зм..	Арк.	Нодокум.	Підпис	Дата		

Програма FreePCB повністю написана англійською мовою та не має русифікатора. Однак радіолюбителем-ентузіастом Валентином Володимиром був укладений дуже прикладний переклад керівництва користувача, який можна знайти на цій сторінці. Там же зберігається опис редактора корпусів програмного пакету FreePCB на російській мові.

Весь програмний пакет FreePCB від першої до останньої строчки був написаний одним єдиним людським іменем від імені Алан Райт (Allan Wright). Талантливий програмист живе у США. Помімо електроніки, не маючи на власному передньому віці, Алан обожає подорожувати, кататися на гірських місцях і навіть самостійно пілотувати власний самолет RV-4. Щоб отримати цікаву інформацію про програму, досить задати питання на форумі сайту.

FreePCB працює у всіх операційних системах сімейства Microsoft Windows, включаючи останню - Windows 7. Можна запустити в Linux та Macintosh за допомогою програм Wine.

PonyProg2000 v.2.07c Програма для прошивання контролерів PIC, AVR і мікросхем EEPROM. Розроблена для спільного використання спеціальних технічних пристроїв (программатором), призначеною для перепрошивання різних ПЗУ та мікроконтролерів з подальшим інтерфейсом програмування. Даний устрій підключається до комп'ютера за допомогою паралельного порту LPT або послідовного - COM. Завдяки комп'ютеру запускається одне з додатків PonyProg, програматор встановлює нульовий мікроконтролер або мікросхему ЕСПЗУ (програма підтримує наступні апаратні інтерфейси: JDM / Ludipiro, AVR ISP (STK200 / 300), DT-006 AVR і EasyI2C). Далі, після попередньої настройки, ви можете записати у пам'ять потрібної мікросхеми нового ПО. Помімо цього, PonyProg може просто прослуховувати дані з ЕСПЗУ або навіть через усі мікропрограми з пам'яттю мікроконтролера. Завдяки підтримці програмних пристроїв, що передаються, передача більшості найпоширеніших мікроконтролерів PIC та AVR, а також різні види мікрошпими пам'яті. PonyProg підтримує наступний інтерфейс зв'язку з програмними мікроконтроллерами: Microchip PIC micro, I2C, SPI eeprom, Microwire та Atmel AVR.

					КВРКІ. 170159.17.01.01 ПЗ	Ар
						8
Зм..	Арк.	Нодокум.	Підпис	Дата		

MPLAB X IDE пропонує безліч функцій, які допомагають налагоджувати проекти та мінімізувати час розробки. Особливості включають:

Візуалізатор даних: Не потрібно купувати додаткові інструменти візуалізації, оскільки потокові дані в режимі реального часу можна переглядати у Візуалізаторі даних.

Вид вводу-виводу: Стани контактів можна перевіряти та маніпулювати ними за допомогою перегляду вводу-виводу для швидкої перевірки обладнання

Корисні ресурси для проектування: заощаджуйте час за допомогою корисних посилань на бібліотеки програмного забезпечення, таблиці даних та посібники користувача, які надаються автоматично.

Простота у використанні: Реєстрація та визначення бітів тепер доступні лише за один клік.

Програма прекрасно підходить для розробки та виготовлення друкованих плат за допомогою технологій ЛУТ (Лазерно-Утюжна-Технологія). Внутрішню програму малюнок друкованої плати можна розвернути, відобразити зеркально, відмасштабувати і надіслати на принтер для друку.

Якщо вам потрібно спроектувати достатньо складного електронного пристрою з більшим числом і щільністю електронних компонентів, то програма Sprint Layout - явно не найкращий варіант, тут ми вже найкраще дивимось на сторону DipTrace та аналогічні CAD-програми з більш широким та професійним функціоналом.

У 6-й версії програми Sprint Layout, яка представлена тут, отечественними ентузіастами були включені деякі доповнення, а також весь інтерфейс, переведений з англійської та німецької мов на російську мову, додано велику кількість електронних компонентів. Внимание !!! За суті це Sprint Layout 5-ї версії з деякими змінами.

Програма протестирована і завжди працює під ОС: Win98, WinNT, Win2000, WinXP, Win7, Win8. Також можливий випуск програм під операційною системою Linux за допомогою програмного пакету Wine або же у VirtualBox (Qemu), де встановлена любов із вищеперелічених ОС сімейства Microsoft

					КВРКІ. 170159.17.01.01 ПЗ	Ар
						9
Зм..	Арк.	Нодокум.	Підпис	Дата		

Windows. Популярне середовище розробки для PIC контролерів, пряме посилання для скачування з офіційного сайту.

DipTrace 2.0.07 -Програмне забезпечення EDA / CAD для створення основних схем і друкованих плат. Розробники пропонують багатоязичний інтерфейс та навчальні програми (зараз доступні англійською та 21 іншою мовою). DipTrace має 4 модулі: редактор схеми, редактор компоновки друкованих плат із вбудованим автотрасировщиком на основі форм, а також попередній перегляд та експорт у 3D, редактор компонентів та редактор робочих місць. Відмінна програма для проектування друкованих плат (безкоштовна версія), на офіційному сайті також можна завантажити бібліотеки 3D-моделей.

Sprint Layout 6.0 - відмінно підтримує формат файлів * .lay з попередньої версії Sprint Layout 5. Примітити до передачі, що файли, створені у версії 6, не можна відкрити в програмі 5-ї версії.

Програма прекрасно підходить для розробки та виготовлення друкованих плат за допомогою технологій ЛУТ (Лазерно-Утюжна-Технологія). Внутрішню програму малюнок друкованої плати можна розвернути, відобразити зеркально, відмасштабувати і надіслати на принтер для друку.

Якщо вам потрібно спроектувати достатньо складного електронного пристрою з більшим числом і щільністю електронних компонентів, то програма Sprint Layout - явно не найкращий варіант, тут ми вже найкраще дивимось на сторону DipTrace та аналогічні CAD-програми з більш широким та професійним функціоналом.

У 6-й версії програми Sprint Layout, яка представлена тут, отечественними ентузіастами були включені деякі доповнення, а також весь інтерфейс, переведений з англійської та німецької мов на російську мову, додано велику кількість електронних компонентів. Внимание !!! За суті це Sprint Layout 5-ї версії з деякими змінами.

Програма протестирована і завжди працює під ОС: Win98, WinNT, Win2000, WinXP, Win7, Win8. Також можливий випуск програм під операційною системою Linux за допомогою програмного пакету Wine або же у VirtualBox (Qemu), де встановлена любов із вищеперелічених ОС сімейства Microsoft

					КВРКІ. 170159.17.01.01 ПЗ	Ар
						10
Зм..	Арк.	№докум.	Підпис	Дата		

Windows. Популярне середовище розробки для PIC контролерів, пряме посилання для скачування з офіційного сайту

PonyProg2000 v.2.07c Програма для прошивання контролерів PIC, AVR і мікросхем EEPROM. Розроблена для спільного використання спеціальних технічних пристроїв (программатором), призначеною для перепрошивання різних ПЗУ та мікроконтролерів з подальшим інтерфейсом програмування. Даний устрій підключається до комп'ютера за допомогою паралельного порту LPT або послідовного - COM. Завдяки комп'ютеру запускається одне з додатків PonyProg, програматор встановлює нульовий мікроконтролер або мікросхему ЄСПЗУ (програма підтримує наступні апаратні інтерфейси: JDM / Ludipiro, AVR ISP (STK200 / 300), DT-006 AVR і EasyI2C). Далі, після попередньої настройки, ви можете записати у пам'ять потрібної мікросхеми нового ПО. Помимо цього, PonyProg може просто прослуховувати дані з ЄСПЗУ або навіть через усі мікропрограми з пам'яттю мікроконтролера. Завдяки підтримці програмних пристроїв, що передаються, передача більшості найпоширеніших мікроконтролерів PIC та AVR, а також різні види мікрошпими пам'яті. PonyProg підтримує наступний інтерфейс зв'язку з програмними мікроконтроллерами: Microchip PIC micro, I2C, SPI eeprom, Microwire та Atmel AVR.

Altera Quartus - це середовище компанії Altera для програмування і розробки під ПЛИС. Quartus дозволяє аналізувати і синтезувати HDL конструкції, що дозволяє розробник складати свої проекти, виконувати годин аналіз, тестувати RTL діаграми, імітує реакцію дизайну на різні подразники, і налаштувати цільової Пристрій на програміста. Quartus Включає в себе реалізацію VHDL і Verilog для апаратного забезпечення, візуального Редагування логічних схем і моделювання векторних сигналу.

Основними можливості САПР Quartus II є:

- 1) різні способи введення поведінкових и структурних описів пристроїв;
- 2) інтегровані засоби допомоги для створення складних проектів;
- 3) MegaWizard® & SOPC;
- 4) Підсистема синтезу;
- 5) Підсистема розміщення внутрішніх ресурсів і розводка ПЛИС;

					КВРКІ. 170159.17.01.01 ПЗ	Ар
						11
Зм..	Арк.	Нодокум.	Підпис	Дата		

- 6) Підсистема моделювання;
- 7) Підсистема годинного АНАЛІЗУ і АНАЛІЗУ споживаної ЕНЕРГІЇ;
- 8) Підсистема програмування НВІС;
- 9) Підсистема оптимізації швидкодії проекту - LogicLock™;
- 10) Підсистема ПІДТРИМКИ інтеграції з іншими засобами автоматизації проектування - NativeLink®;
- 11) система проектування блоків цифрової обробки сигналів - DSP;
- 12) Builder;
- 13) інтегровані засоби розробки програмного забезпечення для
- 14) вбудованих мікро ЕОМ;
- 15) підтримка використання ІР-модулів;
- 16) вбудовані засоби налагодження НВІС в складі системи SignalTap®;
- 17) SignalProbe™;
- 18) підтримка операційних систем Windows, Solaris і Linux;
- 19) підтримка різних схем ліцензування (nodelocked, network).

Verilog, Verilog HDL (англ. Verilog Hardware Description Language) - це мова опису апаратури, що використовується для опису і моделювання електронних систем. Verilog HDL, не слід плутати з VHDL (конкуруючий мова), найбільш часто використовується в проектуванні, верифікації та реалізації (наприклад, у вигляді НВІС) аналогових, цифрових та змішаних електронних систем на різних рівнях абстракції.

Розробники Verilog зробили його синтаксис дуже схожим на синтаксис мови С, що спрощує його освоєння. Verilog має препроцесор, дуже схожий на препроцесор мови С, і основні керуючі конструкції «if», «while» також подібні однойменною конструкціям мови С. Угоди щодо форматування виводу також дуже схожі (див. Printf).

Слід зазначити, що опис апаратури, написаний на мові Verilog (як і на інших HDL-мовах) прийнято називати програмами, але на відміну від загальноприйнятого поняття програми як послідовності інструкцій, тут програма задає структуру системи. Також для мови Verilog не застосовують

					КВРКІ. 170159.17.01.01 ПЗ	Ар
Зм..	Арк.	№докум.	Підпис	Дата		12

Verilog, стандартизований як IEEE 1364, є рухом опису апаратного забезпечення (HDL), що використовується для моделювання електронних систем. Найкраще використовувати розробку та верифікацію цифрових схем на рівнях передачі регістрів абстракції. Він також застосовується при верифікації аналогових схем та схем змінного сигналу, а також при розробці генетичних схем. [1] У 2009 році стандарт Verilog (IEEE 1364-2005) був об'єднаний із стандартом SystemVerilog, створивши стандарт IEEE 1800-2009. З тих пір Verilog офіційно є частиною мови SystemVerilog. Поточна версія - стандарт IEEE 1800-2017.

Мови програмування мікроконтролерів за своєю структурою мало відрізняються від класичних мов для комп'ютерів. Єдиною відмінністю стає орієнтованість на роботу з вбудованими периферійними пристроями. Архітектура мікроконтролерів вимагає, наприклад, наявності бітово-орієнтованих команд. Останні дозволяють виконувати роботу з окремими лініями портів введення / виводу або прапорами регістрів. Подібні команди відсутні в більшості великих архітектур. Навіть ядро ARM, активно застосовується в мікроконтролерах, не містить бітових команд, внаслідок чого розробникам довелося створювати спеціальні методи бітового доступу.

Асемблер є мовою найнижчого рівня. При цьому він дозволяє найбільш повно розкрити всі можливості мікроконтролерів і отримати максимальну швидкодію і компактний код. У деяких випадках альтернативи асемблеру немає, але тим не менше він має безліч недоліків. Незважаючи на одержувану компактність машинного коду, програма, написана на мові Асемблер, громізка і труднопонімаема. Для її створення потрібно відмінне знання архітектури та системи команд мікроконтролерів.

Асемблер відмінно підходить для програмування мікроконтролерів, які мають обмежені ресурси, наприклад 8-ми бітних моделей з малим об'ємом пам'яті. Для великих програм і тим більше 32-розрядних контролерів, краще використовувати інші мови, що відрізняються більш високим рівнем. Це дозволить створювати більш складні і при цьому зрозумілі програми.

Мова програмування C / C ++, відноситься до мов більш високого рівня, в порівнянні з Асемблером. Програма на цій мові краще зрозуміла людині.

					КВРКІ. 170159.17.01.01 ПЗ	Ар
						13
Зм..	Арк.	Нодокум.	Підпис	Дата		

Перевагою C / C ++ є величезна кількість програмних засобів і бібліотек, що дозволяють просто створювати необхідний код. Фактично, C / C ++ сьогодні стала основною мовою розробки керуючих програм. Компілятори даної мови реалізовані практично для всіх моделей мікроконтролерів. Стандартна мова дає можливість перенесення програм з однієї платформи на іншу. Теоретично, використовуючи різні компілятори, можна перетворити будь-яку програму в команди мікроконтролера потрібного типу. На практиці додатково потрібно враховувати архітектуру мікроконтролера кожного типу.

Мова C / C ++ має досить складну для вивчення структуру. Одержуваний програмний код конкретного завдання, має більший обсяг, ніж код тієї ж завдання, реалізованої на Асемблері. Проте мова C / C ++ слід визнати єдиним правильним вибором для професійного програмування мікроконтролерів.

Мова Pascal ще більш зручний для сприйняття і вивчення. Проте, він не має такого поширення як C / C ++, особливо при програмуванні мікроконтролерів. Деякі окремі фірми підтримують дану мову, з метою спрощення переходу на контролери з великих ПК. Зокрема варіант мови під назвою MicroPASCAL входить до складу поставки налагоджувальних засобів фірми Mikroelektronika.

На відміну від класичних мов програмування, візуальні мови дозволяють розробляти програми у вигляді зображень. Серед таких мов можна виділити FlowCODE або Scratoh. Перевагою візуальних мов є добре сприймається структура алгоритму. Це дозволяє просто розібратися в його функціонуванні будь-якій людині, що знає основні символи мови. Переклад структурних схем в команди мікроконтролера, як правило, виконується не відразу. Спочатку алгоритм транслюється в команди асемблера або будь-якої мови високого рівня. Тільки потім, все перетворюється в машинний код. Така схема, незважаючи на свою складність, дозволяє використовувати найбільш зручні компілятори різних розробників.

Ще однією перевагою візуального програмування стає простота вивчення, тому подібні мови часто використовуються для навчання дітей. Недоліком візуального підходу є громіздкість вихідних матеріалів. Проте, подібні мови

					КВРКІ. 170159.17.01.01 ПЗ	Ар
Зм..	Арк.	№докум.	Підпис	Дата		14

програмування знайшли дуже велике поширення для вирішення спеціальних завдань.

Arduino C ++ Хоча його часто називають «мовою програмування Arduino», Arduino фактично програмується на старому доброму C ++ і використовує підмножина бібліотек, специфічних для Arduino.

Відмінності Arduino C ++ від звичайного C ++ полягають в підпрограма для мікроконтролерів, оскільки стандартні функції бібліотеки C ++ не можуть працювати на обмежених потужностях і пам'яті.

1.3 Кінцевий автомат

Цифровий автомат - це логічне пристрій, при якому помимо логічні елементи мають елементи пам'яті. Значення вихідних сигналів такого пристрою залежить не тільки від аргументів входу в даний момент часу, але і від попереднього стану автомата, який фіксує елементи пам'яті. У якості елементів пам'яті можна використовувати триггери. Каждое внутрішнє стан цифрового автомата визначає вихідний стан тригерів та послідовність вхідних сигналів, що діють на вході в даний момент часу, тому такі пристрої називаються послідовними схемами. До послідовності схем можна віднести: триггери, рахунки, реєстри. У загальному випадку структурна схема цифрового автомата може бути представлена у вигляді набору трьох узлів - комбінованої схеми формування вихідних сигналів, комбінованої схеми формування сигналів управління триггерами і, власне, пам'яті.

У будь-якому пристрої обробки цифрової інформації можна виділити два основних блоки - операційний автомат та керуючий автомат.

Операційний автомат служить для зберігання інформації, що виконує набори мікрооперацій та вчислення значень логічних умов, тобто операційний автомат є структурою, організованою для виконання дій над інформацією.

Управляючий автомат генерує послідовність керуючих сигналів, призначену мікропрограму та відповідні значення логічним умовам. Інаше

					КВРКІ. 170159.17.01.01 ПЗ	Ар
						15
Зм..	Арк.	Нодокум.	Підпис	Дата		

говорячи, керуючий автомат задає порядок виконання дій в операційному автоматі, витягуючи з алгоритмів виконаних операцій. Управляючий автомат може бути представлений у двох видах: автомат з жорсткою логікою (з схемною логікою) та автомат з гнучкою логікою (з програміруємою логікою). Розрізнення між автоматами з жорсткою логікою та автоматами з гнучкою логікою в затратах обладнання, необхідних для реалізації одних та тих же функцій, т. е. в вартість автоматів. Кількість обладнання в автоматі з жорсткою логікою використовується майже пропорційно складності мікропрограм. Для автоматів із гнучкою логікою типових більших удельних затрат обладнання при реалізації щодо нескладних мікропрограм. Автомати з жорсткою логікою мають більш високе швидке співробітництво, ніж автомати з гнучкою логікою.

Таким чином, любое пристрій - це композиція операційного та керуючого автоматів. Операційний автомат, реалізація дій над словами інформації, є виконавчою частиною пристроїв, роботою якого керує керуючий автомат, генеруючи необхідні послідовності управління сигналами.

З точки зору сигналів цифрового автомата полезно визначається як система, яка може приймати вхідні сигнали, під їх зворотнім переходом з одного стану в іншому, зберігати його до прибуття наступного вхідного сигналу та видавати вихідні сигнали.

Цифровий автомат вважається кінечним, якщо скінченна множинність вхідних сигналів x , стан i вихідних сигналів y .

Цифровий автомат називає дискретним перетворенням інформації, що вміє приймати різні умови, переходити під зворотний зв'язок вхідних сигналів або командних програм з одного стану в іншому та видавати вихідні сигнали.

У даній роботі модель працює на принципі кінцевого автомата. Кінцевий автомат - це деяка абстрактна модель, що містить кінцеве число станів чогонебудь. Використовується для подання і управління потоком виконання будь-яких команд. Кінцевий автомат являє собою алгоритмічну частину програми в чистому вигляді, тобто кінцевий автомат - це система, яка реагує тільки на зміни зовнішнього середовища, але не здатна до запам'ятовування. Така система не має змінною пам'яті, сама керуюча інформація про структуру автомата зберігається в

					КВРКІ. 170159.17.01.01 ПЗ	Ар
						16
Зм..	Арк.	№докум.	Підпис	Дата		

постійній пам'яті. Єдино збереженим параметром автомата є його поточний стан, що характеризує поточний крок виконання алгоритму.

Кінцевий автомат моделює «інстинктивне» поведінка, заснована на відтворенні програми дій, змінюється в залежності від послідовності зовнішніх подій, але не запам'ятовує інформації із зовнішнього середовища;

Комп'ютерна програма моделює «рефлекторне» поведінку, принципово пов'язане із запам'ятовуванням і обробкою інформації про зовнішні події і налаштуванням (адаптацією) до них. Кінцевий автомат (КА) - алгоритмічна компонента програми «без даних», що моделює «інстинктивне» поведінку, неадаптіруемое до послідовності дій зовнішнього середовища. Розглянемо формальне визначення кінцевого автомата в термінах дискретної математики і теорії множин. Кінцевий автомат визначається шістьма компонентами(табл. 1.1).

Таблиц 1.1 – Компоненти кінцевого автомата.

Умовне позначення	Опис
$A = \{a_1, a_2, \dots, a_m\}$	множина станів (внутрішній алфавіт)
$Z = \{z_1, z_2, \dots, z_r\}$	множина вхідних сигналів (вхідний алфавіт)
$W = \{w_1, w_2, \dots, w_g\}$	множина вихідних сигналів (вихідний алфавіт)
$\delta: A \cdot Z \rightarrow A$	функція переходів
$\lambda: A \cdot z \rightarrow W$	функція виходів
$a_1 \in A$	початковий стан автомата

Опишемо роботу даної системи:

- 1) КА є дискретною системою, що працює по тактам (кроків), в кожному з яких він отримує на вхід один із символів вхідного алфавіту;

2) послідовність символів на вході КА утворює вхідні ланцюжок; • на кожному кроці КА знаходиться в одному з можливих станів, яке називається поточним. Поточний стан - єдина характеристика внутрішнього опису КА, яка змінюється при його роботі (змінна пам'ять). Всі інші елементи його опису є статичними (тобто в будь-якій реалізації представляють собою постійну пам'ять);

3) крок роботи КА складається в переході з поточного стану в новий стан при отриманні на вхід чергового символу вхідний ланцюжка. Цей перехід визначається функцією переходів;

4) результат роботи КА полягає в запису в вихідну ланцюжок символів, що генеруються функцією виходів КА. Тобто паралельно з переходом в новий стан формується вихідний символ, який визначається парою «поточний стан - вхідний символ». Вихідний символ може бути і порожнім; Робота КА полягає в перетворенні вхідного ланцюжка символів в вихідну. Закон цього перетворення визначає поведінку КА. Завдання проектування КА складається в створенні КА, що забезпечує задані правила перетворення ланцюжків. Очевидно, що відсутність у КА внутрішньої пам'яті обмежує його можливості по перетворенню ланцюжків (загальноприйнятий термін - моделює здатність). З іншого боку, ці ж обмеження дозволяють вирішувати в загальному випадку багато проблем (термін - алгоритмічна розв'язність, тобто принципова можливість розробити відповідний алгоритм). Варто зауважити, що найбільш наочна форма подання кінцевих автоматів - це графічна форма подання.

Робота КА полягає в перетворенні вхідного ланцюжка символів в вихідну. Закон цього перетворення визначає поведінку КА. Завдання проектування КА складається в створенні КА, що забезпечує задані правила перетворення ланцюжків. Очевидно, що відсутність у КА внутрішньої пам'яті обмежує його можливості по перетворенню ланцюжків (загальноприйнятий термін - моделює здатність). З іншого боку, ці ж обмеження дозволяють вирішувати в загальному випадку багато проблем (термін - алгоритмічна розв'язність, тобто принципова можливість розробити відповідний алгоритм). Варто зауважити, що найбільш наочна форма подання кінцевих автоматів - це графічна форма подання.

					КВРКІ. 170159.17.01.01 ПЗ	Ар
						18
Зм..	Арк.	№докум.	Підпис	Дата		

Кінцевий автомат моделює «інстинктивне» поведінка, заснована на відтворенні програми дій, змінюється в залежності від послідовності зовнішніх подій, але не запам'ятовує інформації із зовнішнього середовища;

Найчастіше пристрої складаються з двох частин: виконавчої та керуючої. Виконавча частина пристрою зазвичай складна і незмінна. Керуюча частина значно простіша, може змінюватися, і вся гнучкість пристрою полягає саме в гнучкості керуючої частини. В обчислювальній машині одна зі складних і незмінних частин - арифметичний пристрій. Однак на ньому можна виконати як додавання, так і обчислення синуса кута, або рішення диференціального рівняння, або щось інше. Що саме буде отримано, залежить від керуючої частини. Взаємодія керуючої і виконавчої частин показано на (рис. 1.1).

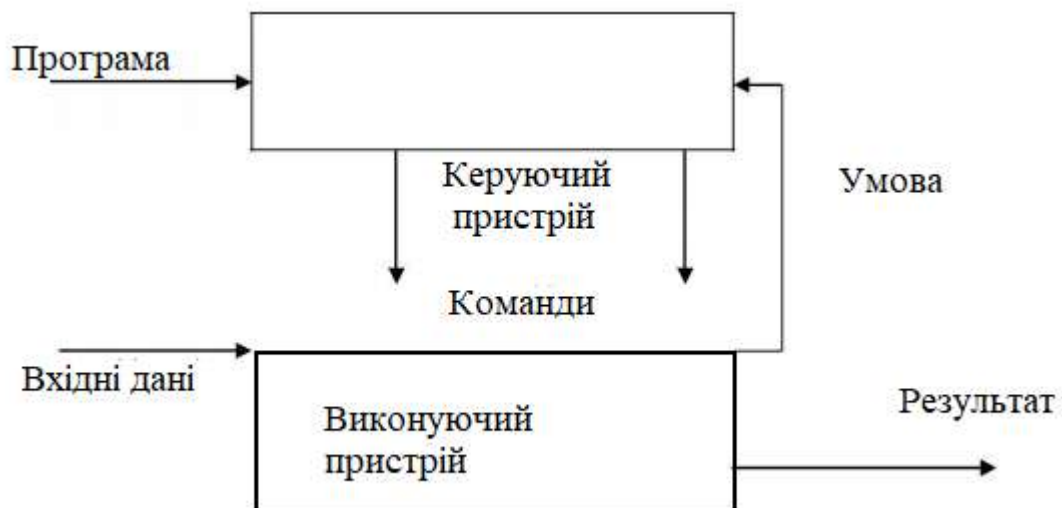


Рисунок 1.1 - Взаємодія керуючого і виконавчого пристроїв

Керуючий пристрій подає на виконавчий пристрій команди. Керуючий пристрій виконує команди і, крім того, має набір булевих функцій, які виробляють «умови», що свідчать про тих або інших події в виконавчому пристрої. основний потік вхідних даних надходить тільки в виконавчий пристрій. У пристрій управління надходить програма. Результати також надходять на вихід, минаючи пристрій управління. Програма завантажується в керуюче пристрій до початку обробки вхідних даних. Керуючий пристрій з завантаженою програмою є

операційним автоматом, який налаштований на вирішення завдання з обробки даних

Зміст виконання умов такої: програма повинна працювати з різними вхідними даними (тобто досить запустити її один раз). Хід обчислень може змінюватися в залежності від вхідних даних. Розпізнавання ситуації, критичної для виконання програми, може бути складною завданням, тому воно здійснюється в виконавчому пристрої. Повідомлення про те, що виникла критична ситуація чи ні, має тільки два значення: «так» або «ні». Повідомлення надходить в управляючий пристрій з виходу «Умови» виконавчий пристрій.

Таким чином, оскільки виконавчий пристрій не змінюється, то і функції, які виробляють умови, не змінюються. В роботі керуючого пристрою можуть використовуватися тільки такі умови, виготовлення яких передбачено в виконавчому пристрої. Розглянемо випадок, коли потрібно виконувати одну і ту ж програму для різних вхідних даних. Що являє собою пристрій, що управляє? Як його можна реалізувати? Виявляється, відповідь можна дати мовою кінцевих автоматів. Пристрій управління:

- 1) це кінцевий автомат. вхідні сигнали для нього;
- 2) умови і синхроімпульсів;
- 3) а вихідні - команди.

Створення керуючого автомата починається зі схеми алгоритму рішення задачі. Вона складається з таких блоків: «Початок», «Кінець», «Операційна вершина», «Умовна вершина».

Початок означає початок роботи алгоритму. Кінець - завершення його роботи. Операційна вершина означає роботу, яка виконується виконавчим пристроєм в фрагменті алгоритму. Вона повинна виконуватися однаково для всіх допустимих вхідних даних, тобто не містити розгалужень.

Умовна вершина містить булеву змінну, значення якої забезпечує вибір одного з двох напрямків подальшої роботи алгоритму.

Стрілки на лініях відображають порядок, в якому повинні виконуватися блоки в схемі.

					КВРКІ. 170159.17.01.01 ПЗ	Ар
						20
Зм..	Арк.	№докум.	Підпис	Дата		

Вміст операційних блоків може повторюватися так само, як і умови в умовних блоках.

При створенні автомата потрібно заздалегідь передбачити сукупність букв вхідного алфавіту для умовних блоків. Він повинні складатися з вхідних букв X_u , де x_0 - синхроімпульс. Остаточний вхідний алфавіт буде сформований в результаті розробки таблиці переходів.

Вихідний алфавіт - це команди, які формує керуючий автомат. Кожна команда запускає виконання деякого операційного блоку або викликає перехід на умовний блок (без виконання операційного блоку). «Порожню команду», що працює програмне забезпечення ніякого блоку, позначимо як y_0 (можна використовувати позначення у вигляді прочерку - «-»). Інші команди позначимо літерою Y_j з тим же номером, який і операційний блок, який ця команда запускає. В результаті отримуємо безліч виходів.

Кількість внутрішніх станів S_k залежить від того, якого типу автомат створюється - Мілі або Мура.

Побудова мікропрограмного автомата за схемою алгоритму (його іноді називають автоматом Уїлкса-Стрінджера) зводиться до таких дій:

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

1.4 Автомат Мілі

Автомат Мілі - це кінцевий автомат, вихідні сигнали якого залежить від стану автомата і сигналів, що подаються на вхід. Тобто, в графі станів кожному ребру відповідає певне значення (вихідний символ). У вершини графа автомата Мілі записуються вихідні сигнали, дугам графа приписують умова переходу із одного стану в інший та вхідні сигнали. Отже автомат типу Мілі продукує

					КВРКІ. 170159.17.01.01 ПЗ	Ар
Зм..	Арк.	№докум.	Підпис	Дата		21

вихідний сигнал, коли на вхідний сигнал змінюється, в залежності від його попереднього стану. При цьому протяжність вихідного сигналу не залежить від протяжності вхідного, лише від його присутності. Розглянемо граф даного кінцевого автомата (рис. 1.2)

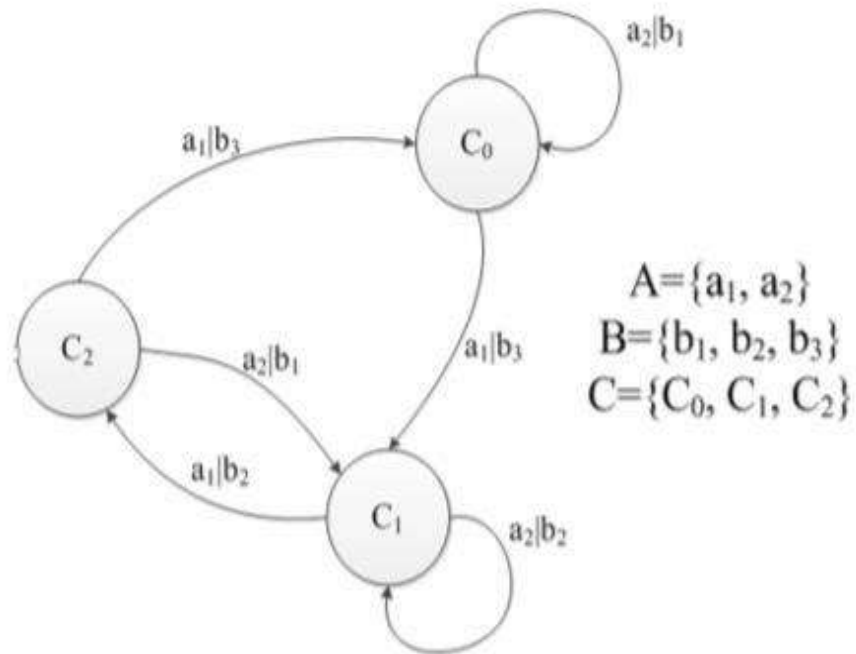


Рисунок 1.2 – граф автомата Милі

Граф автомата - це орієнтований зв'язний граф, вершини якого символізують внутрішні стану автомата, а дуги - переходи з одного стану в інший. Для графа Милі на дугах вказуються подібні і вихідні літери. Вихідні букви пишуться над дугами, символізуючи те, що вихідна стан залежить від стану автомата в попередній момент часу. Будь-автомат можна представити у вигляді таблиці переходів і виходів (ТПВ). В ТПВ рядками є внутрішні стану автомата, а стовпцями - вхідні літери. Побудуємо ТПВ для графа Милі. Якщо не визначена якась вхідні або вихідна буква, то замість неї ставиться прочерк. Якщо не визначено стан, то діє це ж просте правило. Таблиця переходів і виходів графа Милі

В ТПВ Милі в кожній клітині записані переходи і виходи. Наприклад, якщо автомат знаходиться в стані c_0 і на вхід приходять буква a_1 , то він перейде в стан c_1 і на виході з'явиться буква b_3 на рисунку 1.2.

	a_1	a_2
c_0	c_1/b_3	c_0/b_1
c_1	c_2/b_3	c_1/b_2
c_2	c_0/b_3	c_1/b_1

Рисунок 1.2 - Зображення таблиці переходів і виходів графа Милі

1.5 Автомат Мура

Автомат Мура - це кінцевий автомат, вихідне значення сигналу, в якому залежить лише від поточного стану даного автомата, і не залежить безпосередньо, на відміну від автомата Милі, від вхідних значень. Вихідний сигнал автомата Мура визначається парою значень: вхідний сигнал і стану кінцевого автомата в даний момент на рисунку 1.3.

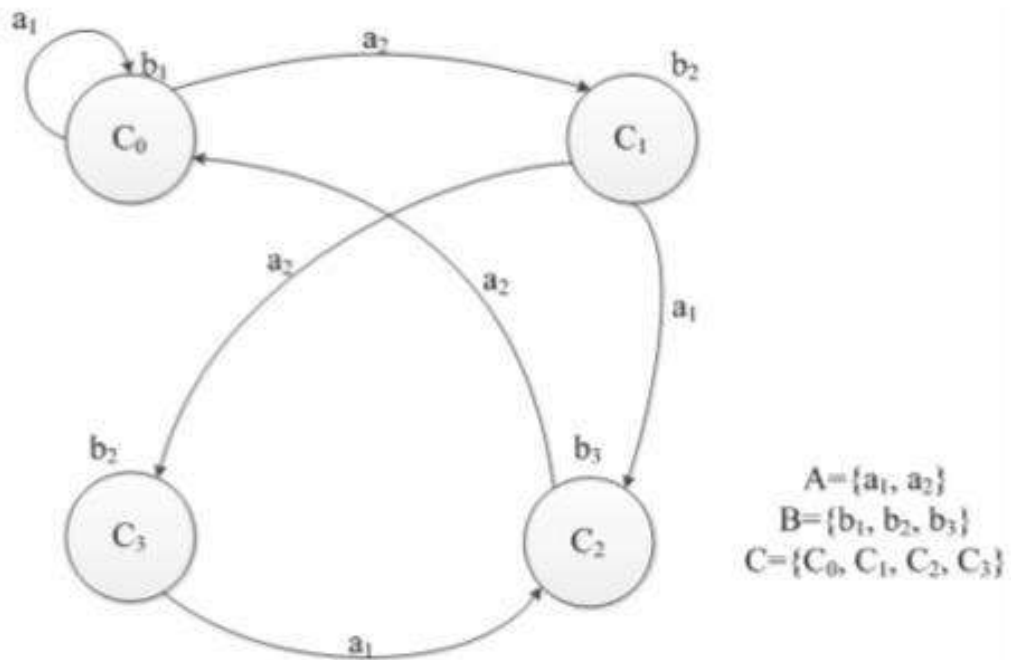


Рисунок 1.3 – Граф автомата Мура

Для графа автомата Мура на дугах записуються тільки вхідні літери, вихідні ж вказуються близько вершин. Варто зауважити, що якщо з кожної вершини виходить стільки дуг, скільки є вхідних букв, то автомат називається повним. Тобто якщо з кожної вершини визначені переходи для кожної вхідний букви. Розглянутий автомат Мілі є повним, а автомат Мура - частковим. Якщо з однієї вершини виходить дуг більше, ніж вхідних букв (тобто 2 і більше дуг з однаковими вхідними буквами), то такий автомат називається недетермінованим.

Побудуємо таблицю переходів і станів для зазначеного графа Мура. Для графа Мура будують зазначену таблицю переходів. Виділяється додатковий стовпець для вихідних букв. У клітці під вхідний буквою пишеться в який стан автомат переходить, в крайній правій клітині - яку вихідну букву повертає на рисунку 1.4.

	a_1	a_2	
C_0	C_0	C_1	b_1
C_1	C_2	C_3	b_2
C_2	-	C_0	b_3
C_3	C_2	-	b_1

Рисунок 1.4. Зображення таблиці переходів і виходів графа Мура

Головною метою синтезу цифрового автомата з пам'яттю є визначення всіх його можливих станів та переходів, відповідно заданому алгоритму функціонування, та отримання функцій збудження всіх входів тригерів, з яких складається автомат. Цього достатньо для складання логічної схеми цифрового автомата.

ЕОМ переробляє інформацію, виконуючи над нею якісь операції. Для виконання операцій над інформацією використовуються операційні пристрої – процесори, канали вводу-виводу, вузли керування зовнішніми пристроями і т.д. Функцією операційного пристрою є виконання заданої множини операцій $F=\{f_1, \dots, f_G\}$ над вхідними словами $D=\{d_1, \dots, d_n\}$ з метою обчислення слів $R=\{r_1, \dots, r_Q\}$, що представляють результати операцій $R=f_g(D)$, де $g=1, 2, \dots, G$.

Функціональна і структурна організація операційних пристроїв базується на принципі мікропрограмного керування, що полягає в наступному:

Будь-яка операція $f_g(g=1, \dots, G)$, реалізована пристроєм, розглядається як складна дія, що розділяється на послідовність елементарних дій над словами інформації. Ці елементарні дії називаються мікроопераціями (передача інформації з одного регістра в іншій, взяття оберненого коду, зсув і т.і).

Для керування порядком проходження мікрооперацій використовуються логічні умови, що у залежності від значень слів, перетворених мікроопераціями, приймають значення "неправда" або "істина" (0 або 1).

Процес виконання операцій у пристрої описується у формі алгоритму, що представляється в термінах мікрооперацій і логічних умов і називається

мікропрограмою. Мікропрограма визначає порядок перевірки значень логічних умов і проходження мікрооперацій, необхідний для одержання результатів, які вимагаються.

Мікропрограма використовується як форма представлення функції пристрою, на основі якої визначається структура і порядок функціонування пристрою в часі.

Абстрактний синтез мікропрограмного автомата за ГСА здійснюється в два етапи:

- 1) одержання позначеної ГСА;
- 2) побудова графа автомата або таблиць переходів і виходів.

Одержанням графа або таблиць переходів-виходів закінчується етап абстрактного синтезу мікропрограмного автомата. Як і для кінцевих автоматів, на етапі абстрактного синтезу можна виконати мінімізацію кількості внутрішніх станів автомата.

Структурний синтез мікропрограмних автоматів після одержання графу автомата або таблиці переходів-виходів в принципі аналогічний канонічному методів синтезу цифрових автоматів,. Однак існують і певні особливості, в першу чергу зв'язані з тим, що для реальних автоматів кількість елементів пам'яті і вхідних сигналів може досягати десяти і більше. Функції збудження і вихідних сигналів важко піддаються мінімізації та й практично мінімізація не дає істотного спрощення цих функцій при великій кількості змінних. Тому мінімізація практично не використовується при синтезі мікропрограмних автоматів.

					КВРКІ. 170159.17.01.01 ПЗ	Ар
Зм..	Арк.	Нодокум.	Підпис	Дата		26

2 ПРОЄКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ

2.1 Аналіз предметної області і виявлення наявних проблем і завдань

Цифровий автомат - це логічний пристрій, який здатний перебувати в одному з декількох стійких станів, здійснювати обробку, зберігання та отримання дискретної інформації відповідно до певного алгоритму.

Такі пристрої вже давно широко використовуються не тільки в різних галузях промисловості, а й в побуті та уявити життя сучасної людини без таких засобів допомоги дуже важко. За допомогою цифрових автоматів працюють: автомати з продажу квитків(жетонів), світлофори, станки та роботизовані механізми на виробництвах, різноманітна обчислювальна техніка і тд. Зважаючи на тенденцію розвитку різноманітних допоміжних пристроїв до автоматизованих виникає попит на розробку засобів керування цими пристроями. Високим попитом користуються автомати для обчислення масивів чисел. ЦА мають обмежену кількість можливостей і кожен проектується для вирішення конкретної задачі тому в цій галузі є безліч не вирішених завдань. Відповідно до завдання проаналізуємо наявність готових рішень для знаходження добутку непарних елементів двовимірного масиву. Пошук готових рішень даної проблеми завершився безрезультатно, тобто засобів для розв'язку такої задачі не виявлено.

Завданням роботи є:

- 1) розглянути існуючі рішення;
- 2) описати математичну модель і змодельювати автомат Мілі;
- 3) провести абстрактний та структурний синтез ;
- 4) розробити мікропрограму операційного пристрою.

2.2 Порівняльний аналіз переваг та недоліків існуючих рішень

Аналіз предметної області показав, що на ринку існує багато різних програмних засобів та пристроїв для обчислення та обробки масивів, але знайти приклад цифрового автомата, який би дозволив обробити масив за потрібним алгоритмом не вдалось.

					КВРКІ. 170159.17.01.01 ПЗ	Ар
Зм..	Арк.	№докум.	Підпис	Дата		27

Окрім автоматів, подібну задачу можна вирішити за допомогою різноманітних програмних засобів.

Використовуючи середовище Microsoft Excel можна виконати дане завдання, але для налаштування всіх параметрів потрібно вивчити всі команди та отримати навички з опрацювання всіх запитів.

Також можна написати спеціалізовану програму для вирішення даної проблеми. Перевагою такого методу є можливість використовувати наявний комп'ютерний пристрій, тобто не потрібно розробляти автомат, ще безумовною перевагою є можливість переносити програму на інші ком'ютери, та додати ще функції, що дозволить зробити програму більш універсальною і відповідно більш популярною.

Ще можна використати середовище Matlab, яким також можна обрахувати масив за потрібними параметрами, але також потрібно мати навички роботи в цьому середовищі.

Недоліками всіх вищеназваних програм є потреба в комп'ютерах для встановлення та роботи з програмою, що веде за собою додаткові витрати на покупку та утримання, також комп'ютери, відносно, цифрових автоматів є менш надійними, що є суттєвим недоліком.

2.3 Методологічні підходи до вирішення задачі за темою дослідження

Електронні цифрові схеми формально можна розкласти на 2 класи:

1. Комбінаційні схеми (КС) - не володіють пам'яттю. Вихідний сигнал формується в залежності від поєднання вхідних даних у фіксований момент часу (учитывающая задержка на перетворення сигналів). дешифратори та шифратори, перетворювачі кодів, комбіновані рахунки та сумматори та т. д.

2. Схеми з пам'яттю - алгоритм їх роботи залежить від стану входів і пам'яті (того, що було в попередні моменти часу). Ці схеми описуються із застосуванням теорії кінцевих автоматів. Мова про них піде далі..

					КВРКІ. 170159.17.01.01 ПЗ	Ар
						28
Зм..	Арк.	№докум.	Підпис	Дата		

Інші слова першого класу - логічні пристрої, обробляючі вхідний сигнал. Другі елементи володіють пам'яттю та реагують на сигнал у залежності від введених у них даних.

Автомат повинен реалізувати деякі функції, які задаються розробником. Він може бути простим сумматором, може реалізувати яку-небудь мікрокоманду процесора, вибирати слова з оперативної пам'яті або займатися синтаксичним аналітичним вираженням.

У загальному вигляді, не вдаючись у деталі, абстрактний автомат можна представити наступним чином:

Або, якщо перейти до видання до математичного опису:

$$A = \langle A, B, C, \delta, \lambda \rangle$$

Множина $\{A\}$ - представляє собою безліч значень на фізичних входах автомата. У наш час у нашому випадку буде забезпечена послідовність високих і низьких рівнів напруги, які будуть кодувати логічні нулі та одиниці.

Множина $\{B\}$ - представляє собою безліч значень на фізичних виходних автоматах.

Множина $\{C\}$ - безліч, що представляє внутрішню станцію автомата - пам'ять. На майбутнє C_0 будемо визначати початковий стан автомата.

$\delta = X \times Z \rightarrow Z$ - це функції переходів автомата, вони однозначно визначають стан a_i у будь-якому переході автомата із стану a_j .

$\lambda = X \times Z \rightarrow Y$ - функції виходу, які вони визначають, що знаходяться на вихідному автоматі в залежності від входів та внутрішнього стану.

δ і λ не показані на схемі для візуального збільшення.

Такий автомат функціонує дискретно на час, якщо значення значень доходів, відвідувань та внутрішньої станції автомата, що змінюються в дискових моментах.

Ітак ми в загальному бачили опис, що є Абстрактним автоматом. Примером такого автомата може бути триггер, реєстр ЕВМ або сумматор.

Виділяємо 2 типи автоматів:

					КВРКІ. 170159.17.01.01 ПЗ	Ар
						29
Зм..	Арк.	№докум.	Підпис	Дата		

Автомати Мили. Описується системою уравнень:

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

$$b(t) = \lambda(a(t), c(t-1)).$$

Автомати Мура. Описується уравненнями:

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

$$b(t) = \lambda(a(t), c(t)).$$

Який стан автомата $c(t)$ у поточний момент часу є функцією його стану в попередній момент часу та вхідного сигналу.

Відличаються автомати з вигляду функцій виходу. У автоматі Мілі вихідного сигналу визначається вхідний сигнал $a(t)$ та стан автомата в попередній момент часу $c(t-1)$. Вихідний сигнал автомата Мура визначає кількість вхідних сигналів $a(t)$ і стан в даний момент $c(t)$.

Так що можна відмітити, що з одного типу можна перейти до другого і наобороту, причому при переході від автомата Мілі к автомату Мура число внутрішніх станих автоматів залишається прежнім, а при обратному переході число внутрішніх стану може бути збільшено. На цьому залишається детально не бути, вважати, що синтезували (намалювали граф) автомат того типу, який надо.

Ітак, на цьому з матчастістю закінчено. Попробуємо описати автомати до цього моменту: Якщо з кожної вершини прийде стовп, сколько є вхідних букв, то автомат називається повним. Інші слова - якщо з кожної версії визначені переходи для кожної вхідної букви. У наших примірах автомат Мілі є повним, а автомат Мура - приватковим.

І ще: Якщо з однієї версії прийде довше, чим вхідні букви (якщо це 2 і більше дуг з однорідними вхідними буквами), цей автомат називається недетермінованими. Так може відбуватися при побудові формалізованого опису, а потім буде створено перехід до визначеного автомата, але його не завжди можна виконати. Опис цього процесу я важко впускаю, коли нарізував визначений автомат.

Методів для вирішення є два: автомат Мура, автомат Мили. Детальніше розглянемо кожний із них.

					КВРКІ. 170159.17.01.01 ПЗ	Ар
						30
Зм..	Арк.	№докум.	Підпис	Дата		

Для графа Милі на дугах вказуються подібні і вихідні літери. Вихідні букви пишуться над дугами, символізуючи те, що вихідна стан залежить від стану автомата в попередній момент часу. Будь- який автомат можна представити у вигляді таблиці переходів і виходів (ТПВ). В ТПВ рядками є внутрішні стану автомата, а стовпцями - вхідні літери. Побудуємо ТПВ для графа Милі. Якщо не визначена якась вхідні або вихідна буква, то замість неї ставиться прочерк. Якщо не визначено стан, то діє це ж просте правило. Таблиця переходів і виходів графа Милі

В ТПВ Милі в кожній клітині записані переходи і виходи. Наприклад, якщо автомат знаходиться в стані s_0 і на вхід приходять буква a_1 , то він перейде в стан s_1 і на виході з'явиться буква b_3 .

Таблиця 1. Таблиця переходів і виходів графа Милі

	a1	a2
s0	s1/b3	s0/b1
s1	s2/b3	s1/b2
s2	s0/b3	s1/b1

2.4 Постановка задачі

Завданням є розробка операційного пристрою для розв'язання наступної задачі: обчислення добутку непарних елементів двовимірного масиву $A[n, m]$, що належать інтервалу $[0; 100]$. Скласти мікропрограму операційного пристрою, до складу якого входить керуючий автомат, у вигляді змістовної та кодованої ГСА. Виконати абстрактний та структурний синтез керуючого автомату як автомату Милі.

На вхід автомата подаються значення комірок двовимірного масиву, якщо елемент масиву парний, перевіряємо чи його значення не дорівнює нулю, добуваємо добуток та переходимо до наступного елемента масиву поки не масив не закінчиться.

Послідовність виконання завдання:

- 1) розглянути існуючі рішення;
- 2) описати математичну модель і змодельювати автомат Мілі;
- 3) розробити мікропрограму операційного пристрою;
- 4) провести абстрактний та структурний синтез.

Згідно завдання роботи потрібно розробити операційний пристрій для розв'язання наступної задачі: обчислення добутку непарних елементів двовимірного масиву $A[n, m]$, що належать інтервалу $[0; 100]$. Скласти мікропрограму операційного пристрою, до складу якого входить керуючий автомат, у вигляді змістовної та кодової ГСА. Виконати абстрактний та структурний синтез керуючого автомату як автомату Мілі.

Цифровий автомат називає дискретним перетворенням інформації, що вміє приймати різні умови, переходити під зворотний зв'язок вхідних сигналів або командних програм з одного стану в іншому та видавати вихідні сигнали.

У даній роботі модель працює на принципі керуючого автомата. Кінцевий автомат - це деяка абстрактна модель, що містить кінцеве число станів чогонебудь. Використовується для подання і управління потоком виконання будь-яких команд. Кінцевий автомат являє собою алгоритмічну частину програми в чистому вигляді, тобто кінцевий автомат - це система, яка реагує тільки на зміни зовнішнього середовища, але не здатна до запам'ятовування. Така система не має змінною пам'яті, сама керуюча інформація про структуру автомата зберігається в постійній пам'яті. Єдино збереженим параметром автомата є його поточний стан, що характеризує поточний крок виконання алгоритму.

Згідно завдання роботи потрібно розробити операційний пристрій для розв'язання наступної задачі: обчислення добутку непарних елементів двовимірного масиву $A[n, m]$, що належать інтервалу $[0; 100]$. Скласти мікропрограму операційного пристрою, до складу якого входить керуючий автомат, у вигляді змістовної та кодової ГСА. Виконати абстрактний та структурний синтез керуючого автомату як автомату Мілі.

До складу змістовної схеми алгоритму входять операційні та умовні верхівки. Наш алгоритм виконує знаходження непарних, використовуючи при

					КВРКІ. 170159.17.01.01 ПЗ	Ар
Зм..	Арк.	№докум.	Підпис	Дата		33

цьому чотири умовні верхівки і вісім операційних верхівок. Позначення операційних верхівок показано на рис 2.2. Позначення умовних верхівок на рис 2.3. Будуємо блок-схему роботи алгоритму.

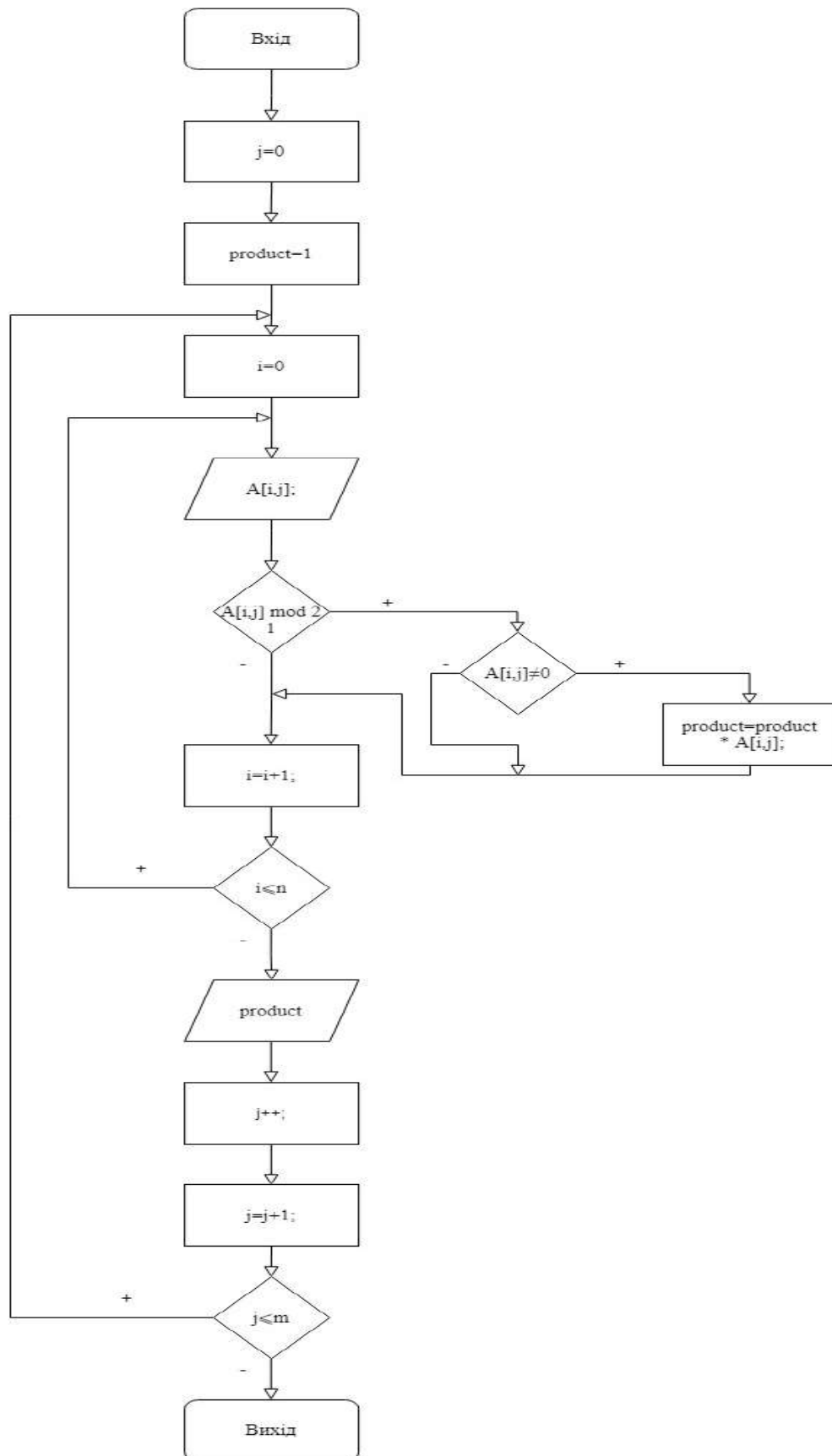


Рисунок 2.2 – Блок схема алгоритма

Кожна верхівка, чи то операційна чи то умовна, кодується. При чому, якщо, мікрооперації повторюються і умовні верхівки повторюються, вони кодуються однаково. В даному прикладі три мікрооперації повторюються двічі, тому, однакові верхівки ми можемо кодувати одним кодом. Таблиця кодування верхівок зображена на рисунку 2.3.

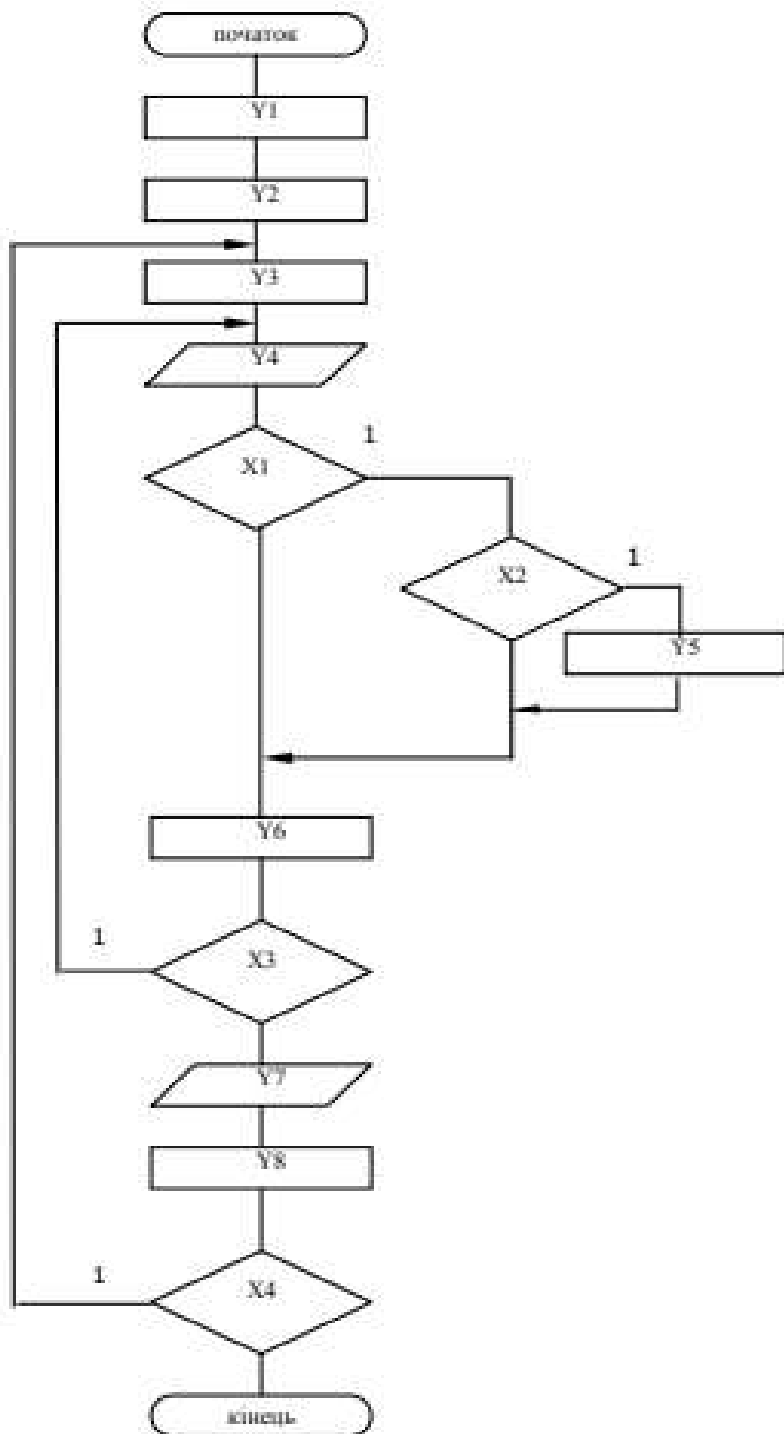


Рисунок 2.3 – Кодована блок схема

Складаємо таблицю кодування мікрокоманд. Кожна операція кодується своєю мікрокомандою. Мікрооперації, які виконуються одна за одною послідовно на протязі одного такта часу, об'єднуються в одну мікрокоманду. В даному прикладі двічі по дві мікрооперації виконуються послідовно. Тому ми об'єднуємо їх в одну мікрокоманду. Таблиця кодування зображена в таблиці.

Складаємо закодовану мікрокомандну схему алгоритма (Рис. 2.4). Проставляємо мітки внутрішніх станів автомата Мура таким чином:

- 1) мітки ставляться біля кожної мікрокоманди;
- 2) початок і кінець мікрокомандної схеми алгоритма відмічається міткою a_0 ;
- 3) перехід з одного стану в інший повинен проходити через умовну або операційну верхівку автомата;
- 4) біля кожної мікрокоманди мітки проставляються відповідно порядковому номеру;

$x_1 \quad x_2 \quad x_3 \quad x_4$

$a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6$

$y_0 \quad y_1 \quad y_2 \quad y_3 \quad y_4 \quad y_5 \quad y_6 \quad y_7 \quad y_8 \quad y_6$

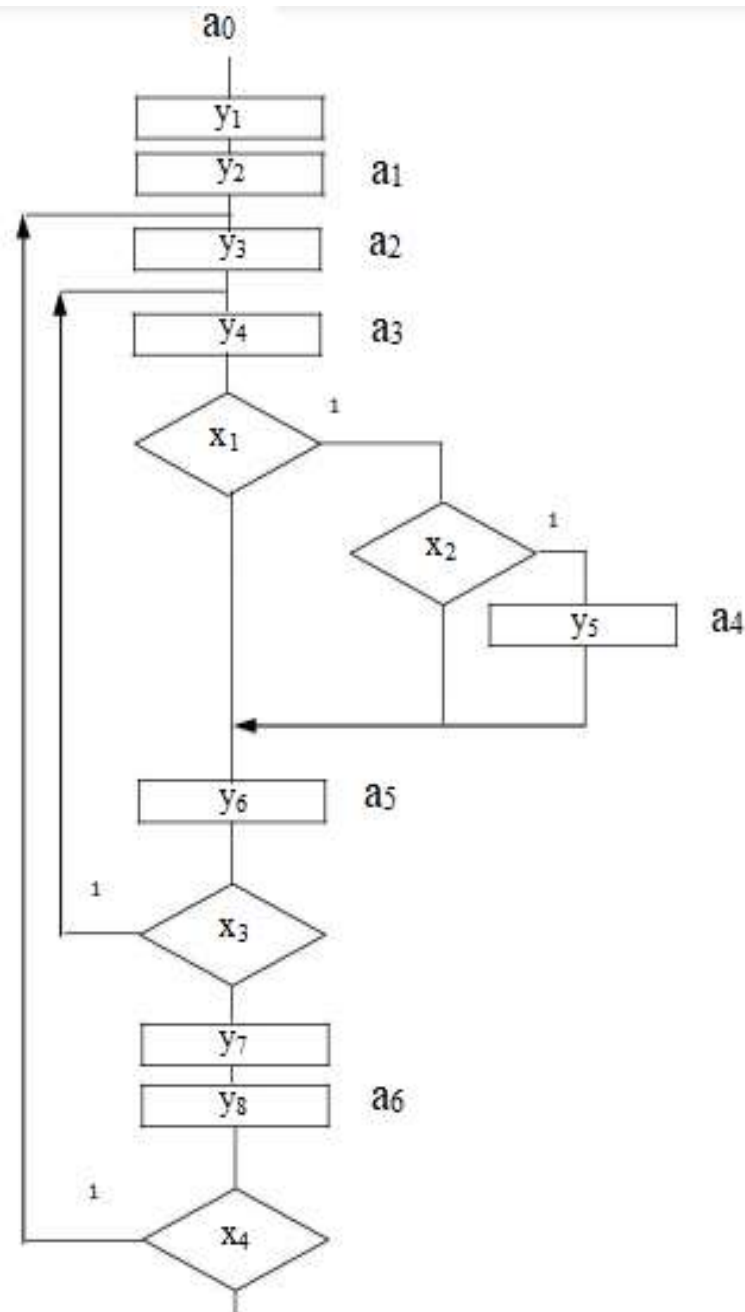


Рисунок 2.4 – Закодована мікрокомандна схема алгоритма

2.5 Абстрактний синтез автомату Мілі

Перший етап. Позначаємо вершини автомата. Для цього слід пам'ятати, що входи усіх вершин, які слідують за операторним, мають бути позначені; входи всіх вершин, окрім кінцевої, позначаються різними символами; всі входи вершини позначаються одним символом. Результат роботи проведеної на першому етапі є ГСА з позначеними вершинами(рис 2.5).

Зм..	Арк.	Нодокум.	Підпис	Дата

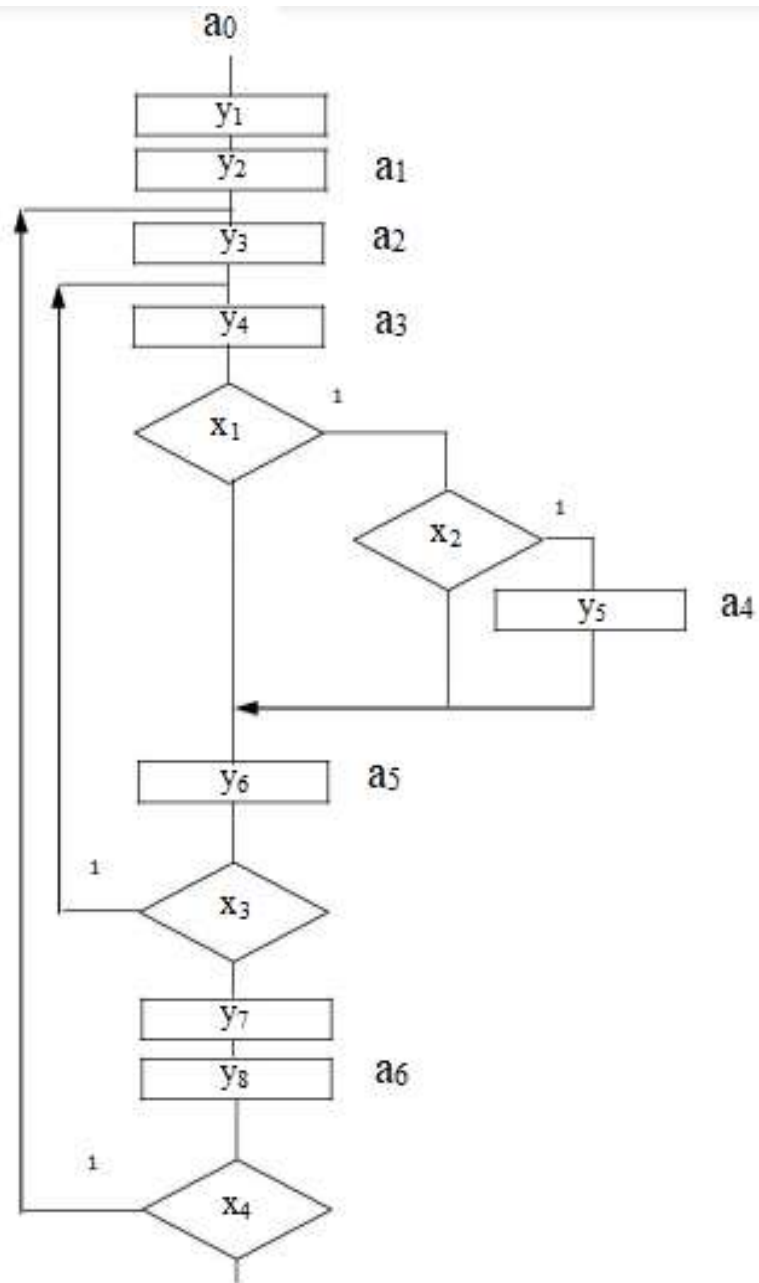


Рисунок 2.5 – ГСА, позначена для автомату Мілі

Стани автомата:

$a_0 = x_4$;

$a_1 = j_0$; $product = 0$;

$a_2 = i_0$; введення $product$;

$a_3 = a[i:j] \bmod 2 = 1$;

$a_4 = product + 1$; $a[i:y] \neq 0$;

$a_5 = i++$;

$a_6 = j \leq m$; $j++$, виведення $product$.

Зм..	Арк.	Нодокум.	Підпис	Дата

Другий етап. На основі позначеної ГСА будуюмо граф автомата та таблиці переходів. Для побудови графа, вважаємо, що автомат матимете стільки ж станів, як символів a_i позначених на ГСА.

Позначаємо на схемі всі стани автомата a_i та будуюмо всі переходи, які переводять автомат із одного стану в інший. Відмічаємо всі зазначені шляхи на графі у вигляді стрілок або дуг, приписуємо до кожної умову переходу та вихідний сигнал який утворюється в наслідок даного переходу. Результат роботи представлений на рисунку 2.6.

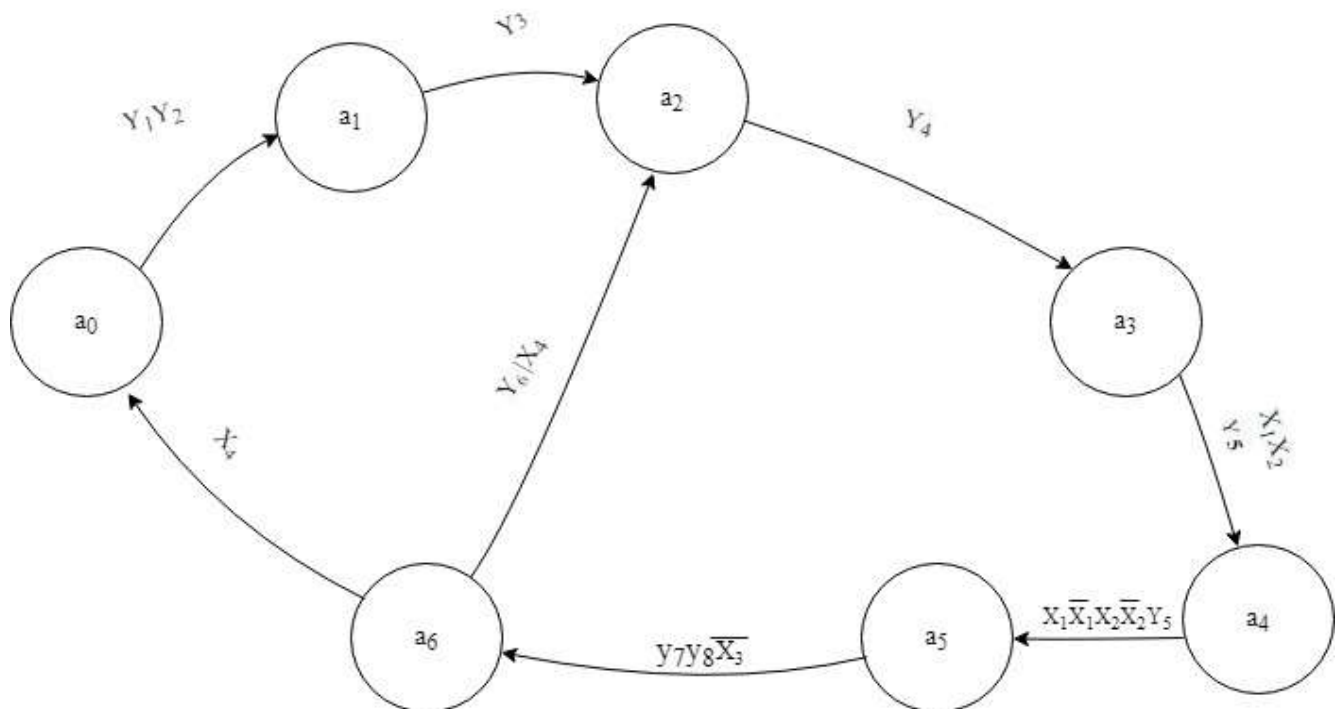


Рисунок 2.6 – Граф автомату Мілі

На основі графа будуюмо таблицю переходів – виходів. Прямая таблиця переходів відображена в таблиці 2.1. Зворотня таблиця переходів відображена в таблиці 2.1.

Зм..	Арк.	Нодокум.	Підпис	Дата

Таблиця 2.1 - Пряма таблиця переходів

a_m	a_s	X	Y
a_0	a_1	-	$y_1 y_2$
a_1	a_2	-	y_3
a_2	a_3	-	y_4
a_3	a_4	$x_1 x_2$	$y_5 y_6 y_4$
	a_5	$\overline{x_1 x_1 x_2 x_2 x_2}$	$y_4 y_6$
a_4	a_5	-	$y_5 y_6$
a_5	a_6	$\overline{x_3 x_3}$	$y_7 y_8$
a_6	a_0	x_4	-

Таблиця 2.2 - Зворотня таблиця переходів

a_m	a_s	X	Y
a_6	a_0	x_4	-
a_0	a_1	-	$y_1 y_2$
a_1	a_2	-	y_3
a_2	a_3	-	y_4
a_3	a_4	$x_1 x_2$	y_5
a_4	a_5	x_1	y_6
		$x_1 \overline{x_2 x_2}$	y_6
a_5	a_6	x_3	$y_7 y_8$

Умовні позначення:

a_m – вихідний стан

a_s – стан переходу

x – умова(вхідний сигнал)

y - вихідний сигнал, який виробляє автомат під час переходу з a_m в a_s .

Виконаємо структурний синтез автомата Мілі, заданий таблицею переходів-виходів.

Таблиця 2.3 - Пряма таблиця переходів

a_m	a_s	X	Y
a_0	a_1	-	$y_1 y_2$
a_1	a_2	-	y_3
a_2	a_3	-	y_4
a_3	a_4	$x_1 x_2$	$y_5 y_6 y_4$
	a_5	$\overline{x_1} \overline{x_1} x_2 \overline{x_2} x_2$	$y_4 y_6$
a_4	a_5	-	$y_5 y_6$
a_5	a_6	$\overline{x_3} x_3$	$y_7 y_8$
a_6	a_0	x_4	-

Кількість станів автомата $M=7$, відповідно кількість елементів пам'яті:

$$M = \lceil \log_2 M \rceil = \lceil \log_2 7 \rceil = 3.$$

Внутрішні стани автомата кодуємо за допомогою карт Карно (таблиця)

Таблиця 2.2 – Карта Карно для кодування станів автомата

$Q_1 Q_2 Q_3$	00	01	11	10
0	a_0	a_2	a_5	a_1
1		a_3	a_6	a_4

Зм..	Арк.	Нодокум.	Підпис	Дата
------	------	----------	--------	------

$$a_0 = 000$$

$$a_1 = 010$$

$$a_2 = 101$$

$$a_3 = 001$$

$$a_4 = 110$$

$$a_5 = 011$$

$$a_6 = 111$$

Далі побудуємо структурну таблицю переходів – виходів автомата Мілі. У стовпці $k(a_m)$ і $k(a_s)$ вказуємо код вихідного стану і код стану переходу. У стовці ФЗ (функції судження)

Таблиця 2.3 – Структурна таблиця переходів – виходів автомата Мілі

A_m початковий стан	$K(a_m)$ Кодовий початковий стан	a_s - стан переходу	$K(a_s)$ - кодовий стан переходу	X - умова(вхідний сигнал, що забезпечує перехід a_m - a_s)	Y - вихідний сигнал при переході a_m - a_s	ФЗ- функція збудження тригерів
a_0	000	a_1	010	-	$y_1 y_2$	
a_1	010	a_2	101	-	y_3	
a_2	101	a_3	001	-	y_4	
a_3	001	a_4	110	$x_1 x_2$	$y_5 y_6 y_4$	
		a_5	011	$\overline{x_1} \overline{x_1} x_2 \overline{x_2} x_2$	$y_4 y_6$	
a_4	110	a_5	011	-	$y_5 y_6$	
a_5	011	a_6	111	$\overline{x_3} \overline{x_3}$	$y_7 y_8$	
a_6	111	a_0	000	x_4	-	

Щоб отримати функції збудження виконаємо наступні дії. Вираз кожної функції збудження представлений у вигляді логічної суми добутків $a_i x$, де a_i -

початковий стан, X – умова переходу. Щоб спростити отримані вирази проведемо всі можливі операції поглинання та склеювання.

$$J_1 = a_5 \overline{x_3 x_3}$$

$$K_1 = a_1 + a_3$$

$$J_2 = a_3 + a_4 x_1 x_2$$

$$K_2 = a_3 \overline{x_1 x_1 x_2} \overline{x_2 x_2}$$

$$K_3 = a_4 x_1 x_2 + a_5 \overline{x_1 x_1 x_2} \overline{x_2 x_2}$$

$$J_3 = a_5 \overline{x_1 x_1 x_2} \overline{x_2 x_2}$$

Функції виходів:

$$y_1 = a_0 + a_6 \overline{x_4 x_4}$$

$$y_2 = a_0 + a_6 \overline{x_4 x_4}$$

$$y_3 = a_1 + a_6 x_4$$

$$y_4 = a_2 + a_3 x_3$$

$$y_5 = a_4 x_1 x_2 + a_3 \overline{x_1 x_1 x_2} \overline{x_2 x_2}$$

$$y_6 = a_4 x_1 x_2 + a_5 \overline{x_1 x_1 x_2} \overline{x_2 x_2}$$

$$y_7 = a_5 \overline{x_3 x_3}$$

$$y_7 = a_5 \overline{x_3 x_3}$$

На основі одержаних функцій збудження та функцій виході будемо функціональну схему автомату Мілі. Щоб побудувати функціональної схеми автомата на основі отриманих виразів потрібно замінити a_i значеннями через $Q_1 Q_2 Q_3$, або отримати сигнал, який відповідає a_i . Для одержання сигналу a_i також застосовують спосіб з використанням дешифратора станів. На вхід дешифратора подається сигнали з виходів елементів пам'яті $Q_1 Q_2 Q_3$.

$$A = a_0$$

$$B = a_6 \overline{x_4 x_4}$$

$$C = a_1$$

$$D = a_6 x_4$$

$$E = a_2$$

$$F = a_4 x_1 x_2$$

					КВРКІ. 170159.17.01.01 ПЗ	Ар
Зм..	Арк.	Нодокум.	Підпис	Дата		43

$$J_1 = D$$

$$K_1 = F + aZ$$

$$J_2 = F + E$$

$$K_2 = C$$

$$J_3 = B$$

На підставі отриманих рівнянь будуюмо функціональну схему автомата.

У загальному вигляді структура кінцевого автомата задається функціональним перетворенням (F) та блоком елементів пам'яті (ЕП). Як вже говорилося, потенційні виходи автомата (мікрокоманди) реалізуються як функції стану, запізаного в ЕП, а імпульсні виходи реалізуються у функціональному перетвореному Ф. Структурна схема управління автоматами, за таким чином, може бути представлена.

Прежде чем приступить к реализации автомата, замість того, що його граф переходів не повністю визначений; переходи під зворотним сигналом "а" не мають вихідних сигналів, а з стану перехід під зворотним сигналом "b" не повідомляється. Неопределенность першого типу, очевидно, означає наступне: в результаті переходу під зворотним сигналом "а" неявно передбачається, що на виході виникає нейтральний сигнал, котрому не відповідає жодна мікрокоманда. Очевидний метод вивчення цієї невизначеності - явне введення такого сигналу. Второй вид неопределенности - відсутність переходу - також повинен бути устранин, оскільки в цьому стані зовнішній сигнал "b" може бути поданий. Тому слід явно визначити реакцію автомата на цей сигнал. Примем наступне рішення: у використаному стані автомат на входному сигналі "b" не реагує, тобто залишається в тому же стані, видається на виході нейтральний сигнал (обозначаючи його z_3). Полностью определенного графа переходов управляющего автомата електронних годин, представлених на рисунку 2.7.

					КВРКІ. 170159.17.01.01 ПЗ	Ар
						44
Зм..	Арк.	№докум.	Підпис	Дата		

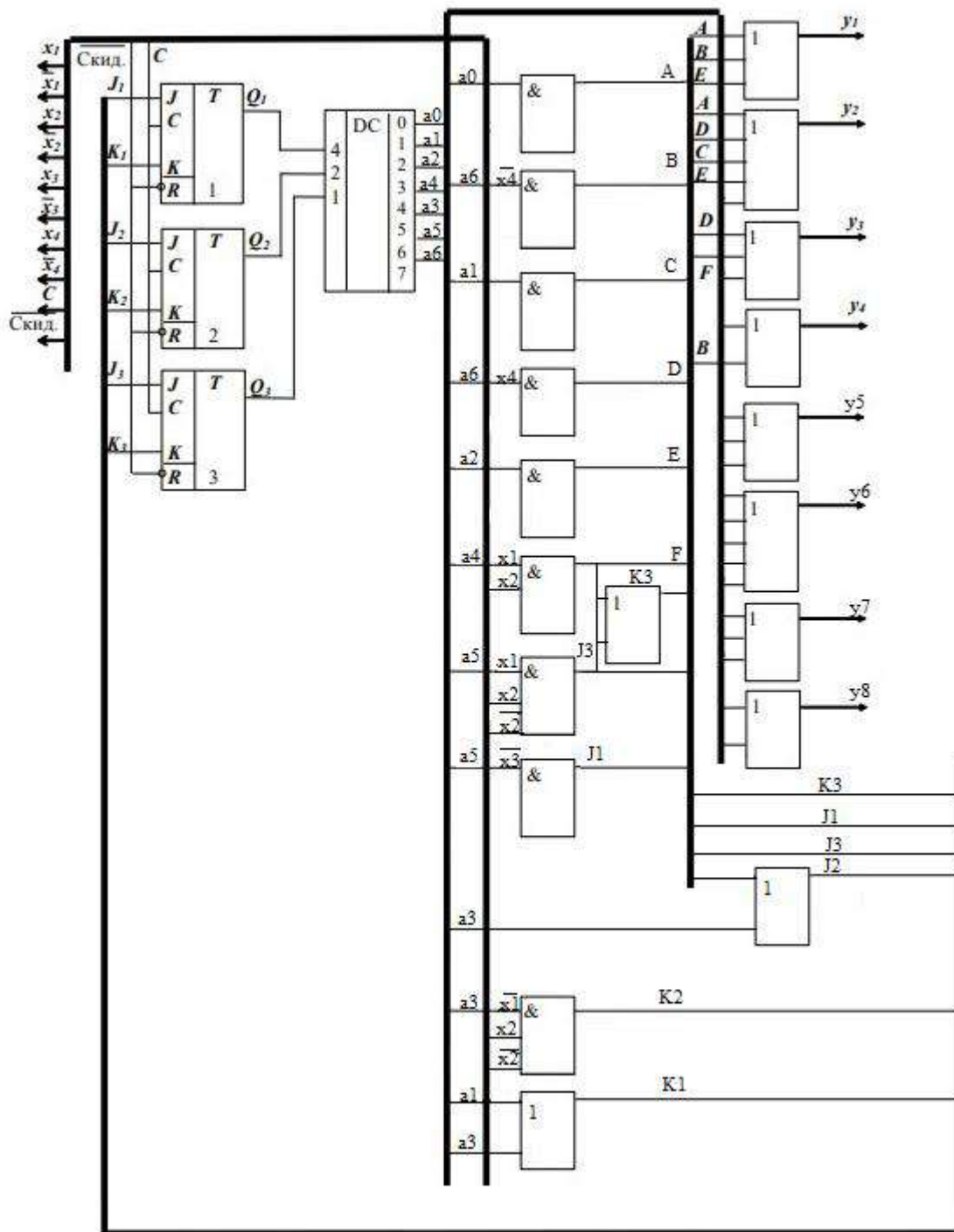


Рисунок 2.7 – Функціональна схема автомата Мілі

У вхідній частині розташовані три D-тригери, які є трьома логічними елементами АБО, до яких подається вхідний сигнал, декодер та дві шини, одна шина необхідна для передачі сигналів, що надходять з декодера, а друга - для сигналів з виходів компаратора. Перетворення сигналу протягом одного такту часу виконується у перехідній частині автомата. Після проходження через логічні

Зм..	Арк.	Но́докум.	Підпис	Дата

елементи ТА та (або) АБО, чи не роблячи того, сигнал трансформується і результат передається на шину (а), з якої він продовжує передаватися до програмованої логічної матриці. На основі системи рівнянь переходів будується перехідна частина. Вихідна частина будується на основі системи рівнянь виходів.

На цьому етапі виконується аналогічний процес, що відбувався в перехідній частині, окрім того, що сигнали подаються до вихідної шини у, з вихідної шини сигнал передається до вихідної матриці.

Безпосередня установка або скидання тригера здійснюється подачею напруги низького рівня на входи S (set-установка) або R (reset - скидання) відповідно. Подавати логічний нуль на обидва входи відразу неприпустимо. Якщо низький рівень присутній на R або S, то сигнали на входах D і C ніяк не впливають на стан тригера. Однак у функціональному плані більш важливими є саме входи. Тригер працює так, що сигнал від входу D передається на вихід Q в момент позитивного перепаду напруги на C. На виході Q устанавлюється відповідно рівень напруги, протилежний рівню на Q. При цьому під час постійного рівня напруги або в момент негативного перепаду напруги на C значення Q не може змінитися. Щоб тригер перемикався правильно, рівень на вході D слід зафіксувати заздалегідь, перед приходом тактового перепаду на C. Мінімальний час між появою сигналу на D і на C називається захисним інтервалом. При роботі в такому режимі на входах R і S повинен бути присутнім рівень логічної одиниці.

Використання елементів пам'яті при реалізації кінцевого автомата має деякі особливості. Справа в тому, що в момент перемикавання автомата необхідно одночасно подавати на вхід F код старого стану і запам'ятовувати код нового, одержуваній з виходу F. Для цього можна в кожному розряді коду стану використовувати два тригера. Ці тригери з'єднуються послідовно, тому що інформація про номер стану на виході F при одному перемиканні автомата буде використовуватися на вході F наступного разу, коли.

Розглянемо задачу впливу на входи C тригерів. Сигнали хіс'являються практично одночасно. На виході F після появи на вході встановлюється код наступного стану, який необхідно запам'ятати в тригерах T1-T2.

					КВРКІ. 170159.17.01.01 ПЗ	Ар
						46
Зм..	Арк.	Нодокум.	Підпис	Дата		

+ Необхідно пам'ятати, що в той момент, коли на тактовий вхід Z надходить позитивний перепад напруги, сигнал на інформаційному вході D повинен бути встановлений заздалегідь. Тому на С входи тригерів T1 і T2 ми будемо подавати синхроімпульс SC $t_0 < t_1 < t_s$ - затримкою. Причому, t_0 - це сума часу затримки в блоці F і захисного інтервалу тригера, t_s - тривалість синхроімпульса. Позитивний перепад напруги на тактові входи С тригер T3 треба подати тільки після закінчення дії синхроімпульса. Це можна зробити інверсією сигналу, який ми подаємо на С входи T1-T2.

2.6 Висновки

За час роботи над першим та другим розділом дипломної роботи більш поглиблено ознайомились з теорією побудови КА, проведено аналіз існуючих рішень, сформувавши задачу для дослідження та вирішення виявлених недоліків.

В першу чергу відбулось ознайомлення з абстрактними автоматами, усіма принципами, етапами і структурою їх побудови. Основний акцент був спрямований на вивчення автомата Мілі. Досліджено аспекти структурного та абстрактного синтезу автомата.

Наступним етапом роботи є проектування автомата Мілі середовищі Intel Quartus Prime II та написання мікропрограми.

					КВРКІ. 170159.17.01.01 ПЗ	Ар
						47
Зм..	Арк.	Нодокум.	Підпис	Дата		

3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ

3.1 Схема автомату Мілі

Процес проектування мікропрограмного автомату Мілі зводиться до такого переліку дій:

- 1) кодування умовних та операційних верхівок;
- 2) опис таблиці та граф-схеми переходів;
- 3) структурний та абстрактний синтез автомата;
- 4) кодування станів автомату;
- 5) будуємо функціональну схему спроектованого автомату.

Керуючий автомат завжди представляє об'єднання функціональних блоків, кожен з яких опрацьовує певну мікрооперацію, описану в мікропрограмі. Частина блоків є блоками умовних переходів, які формують сигнали істинності мікроумов. До складу керуючого автомату входить блок пам'яті, що складається з блоків оперативного запам'ятовуючого пристрою, який використовують для збереження проміжних і кінцевих результатів обчислення.

Операції пересилання, складання та інші мікрооперації, що не мають впливу на порядок виконання інструкцій прошивки реалізують операційні функціональні блоки. На виходи всіх операційних функціональних блоків встановлені тристабільні буфери, для пропуску результатів мікрооперацій на блок пам'яті, але тільки якщо такий сигнал встановлено. В другому випадку на вихід тристабільного буферу встановлюють Z-стан. Це означає, що результати обробки даних операційним функціональним блоком переносяться в пам'ять, лише при встановленому сигналі.

Слід розглянути базову ЕВМ із зазначеними позиціями, щоб визначити, що мова машинного команди не є мовою самого нижнього рівня. На нижньому рівні ЕВМ виконуються основні дії (мікрооперації) над словами даних. Управління порядком слідування мікрооперацій здійснюється за допомогою пристроїв управління (УУ) базової ЕВМ. Якщо мікро-ЕВМ побудовано на однокристальному МП, то УУ, як правило, входить до складу БІС МП.

					КВРКІ. 170159.17.01.01 ПЗ	Ар
Зм..	Арк.	Нодокум.	Підпис	Дата		48

На етапі циклу вибору команд УП інтерпретує код операції команд, вибраної з програмної пам'яті. На етапі виконання команд УУ формує необхідний набір мікрооперацій. Одною з найважливіших характеристик УУ є можливість зміни послідовності управління сигналами (УС). За цим критерієм УУ розділяються на УУ з жорсткою логікою та на універсальних або мікропрограмних УУ.

Спеціалізовані УУ формують неизмінну послідовність УС. Мікропрограма зберігається для зчеплення системи зв'язку між елементами УУ. Після подальшої послідовності вироблення керуючих сигналів залежить від визнаних операцій, сформованих АЛУ, до схемних рішень спеціалізованих УУ отримують складні. Відсутність запоминаючих пристроїв робить спеціалізовані УУ схеми технічних складних. Введення нової операції або зміна старого вимагає змін схем. Невимінюваність і складність структури цього типу управління є основним його недостатком, а висока швидкість - головним перевагою.

Мікропрограмне управління - вид ієрархічного управління роботою ЕОМ, при якому кожна машинна команда є зверненням до послідовності мікрокоманд нижчого рівня, ніж сама команда. Набір мікрокоманд називається мікропрограмою і зберігається в постійній пам'яті пристрою управління УУ.

Будь-яка машинна команда представляє послідовність мікрокоманд (МК), яка утворює мікропрограму. Записані в пам'яті УУ мікрокоманд визначають роботу всіх пристроїв ЕОМ..

Існують мікропроцесорні набори, що мають єдиний мову - мову мікрокоманд. Наприклад, мікропроцесорний комплект К589. Мікропрограмне пристрій управління (МПУУ) за своєю структурою є дуже простий ЕОМ. Для такої ЕОМ реєстри і вентиляльні схеми базової ЕОМ служать як би пристроями введення-виведення. Структурна схема МПУУ показана нижче. Принцип мікропрограмного управління зумовлює можливість зберігання мікропрограм системи команд ЕОМ в ЗУ того чи іншого типу, що скорочує апаратний склад МП.

					КВРКІ. 170159.17.01.01 ПЗ	Ар
						49
Зм..	Арк.	№докум.	Підпис	Дата		

Мікропрограми зберігаються в пам'яті мікрокоманд ПМК. СФАМК- схема формування адреси мікрокоманд. Рамки - регістр адреси мікрокоманд. РМК - регістр мікрокоманд.

3.2 Опис апаратної реалізації керуючого автомату на основі автомату Мілі

Суть нанесення станів на ГСА для автомата Милі впливає з структурної моделі автомата (рисунок 3.1) і закону його функціонування для вихідних сигналів.

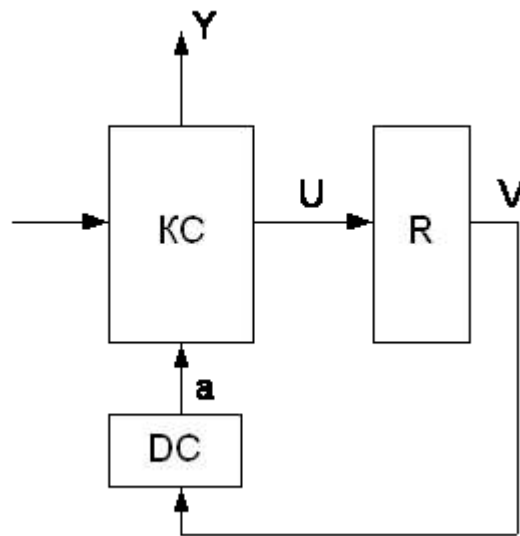


Рисунок 3.1 - Структурна схема

Тут вихідні сигнали y_t залежать не тільки від станів автоматів в даний момент часу a_t (в даному такті), але і від того які значення мають осведомітельних сигнали x_t . Це означає, що в одному і тому ж стані a_t автомат може виробляти різні керуючі сигнали u залежно від того, які значення отримують осведомітельних сигнали, аналізований в даному стані.

Нехай в стані a_t автомат аналізує сигнал x_3 . Якщо $x_3 = 0$, то автомат вироблятиме керуючий сигнал u_2 , а якщо $x_3 = 1$, то - керуючі сигнали Y_5 і Y_6 (рисунок 3.1).

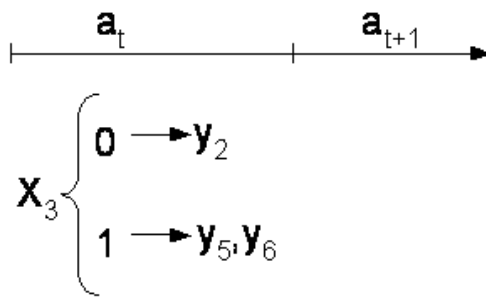


Рисунок 3.2 – Діаграма

Це відповідає ділянці ГСА, який представлений на малюнку 3.2.

Таким чином стан автомата a_t має бути зазначено перед умовної вершиною x_3 . Цей стан відзначається зірочкою (*) на ГСА (рисунок 3.3).

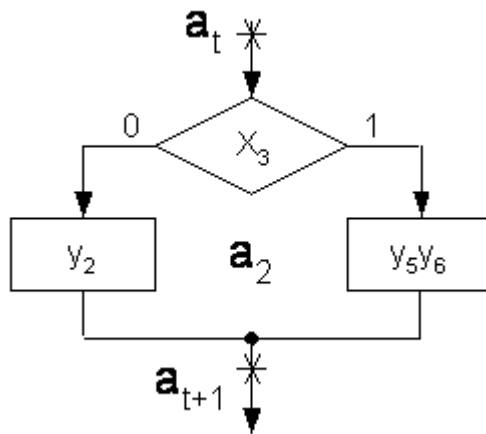


Рисунок 3.3 – Зміна станів автомата

Наступне стан a_{t+1} , буде відзначено після операторних вершин. На виконання мікрооперацій, записаних в одній операторної вершині витратиться один такт роботи процесора, тому наступний стан a_{t+1} , буде відзначатися після операторної, але перед першою умовної або операторної вершинами.

Покажемо, яким чином наносяться стану для автомата милі на прикладі ГСА.

Початковий стан a_0 буде відзначено перед умовної вершиною x_0 . У цьому стані автомат не виробляє керуючих сигналів, якщо $x_0=0$, тобто автомат знаходиться в режимі очікування. Однак, якщо x_0 стане рівним одиниці, то

автомат в стані a_0 буде виробляти сигнали y_0 і v_1 і в наступному такті перейде в стан a_1 , яке зазначено на ГСА перед вершиною x_1 .

У стані a_1 автомат при $x_1 = 1$ буде виробляти сигнал y_2 і в наступному такті перейде в стан a_2 . Якщо x_1 буде дорівнює 0, то автомат вироблятиме керуючі сигнали y_3 і y_4 і в наступному такті перейде в стан a_3 .

Зі стану A_2 автомат завжди буде переходити в стан a_3 з виробленням сигналів y_3 і y_4 .

У стані a_3 при $x_2 = 0$ і $x_1 = 1$ автомат витратить 1 такт на вироблення сигналу y_2 і перейде в стан a_2 . Якщо $x_2 = 0$ і $x_1 = 0$, то автомат витратить 1 такт на вироблення сигналів y_3 і y_4 і знову перейде в стан a_3 .

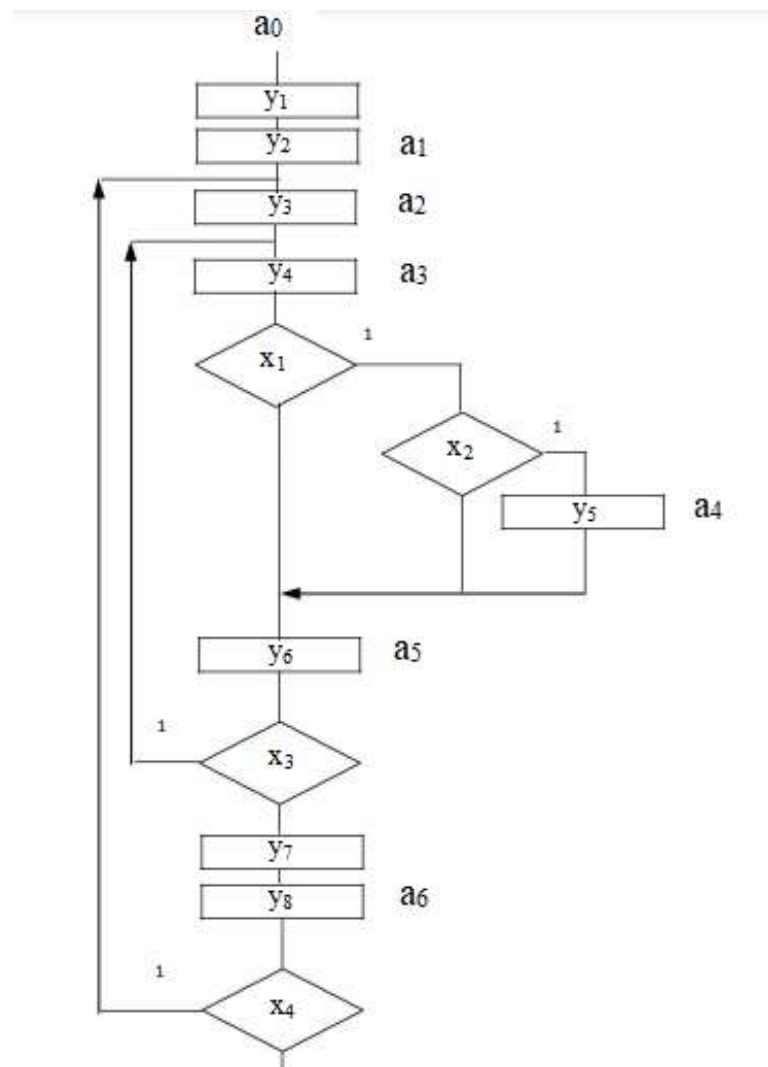


Рисунок 3.4 – ГСА, позначена для автомату Мілі

Опис станів:

$a_0 = x_4$;

$a_1 = j_0$; $product = 0$;

$a_2 = i_0$; введення $product$;

$a_3 = a[i:j] \bmod 2 = 1$;

$a_4 = product + 1$; $a[i;y] \neq 0$;

$a_5 = i++$;

$a_6 = j \leq m$; $j++$, виведення $product$.

Крім цього зі стану a_3 можливі переходи в стан a_0 при:

1) $x_2 = 1$ і $x_3 = 1$ з виробленням y_0 ;

2) $x_2 = 1$ і $x_3 = 0$ без вироблення керуючих сигналів.

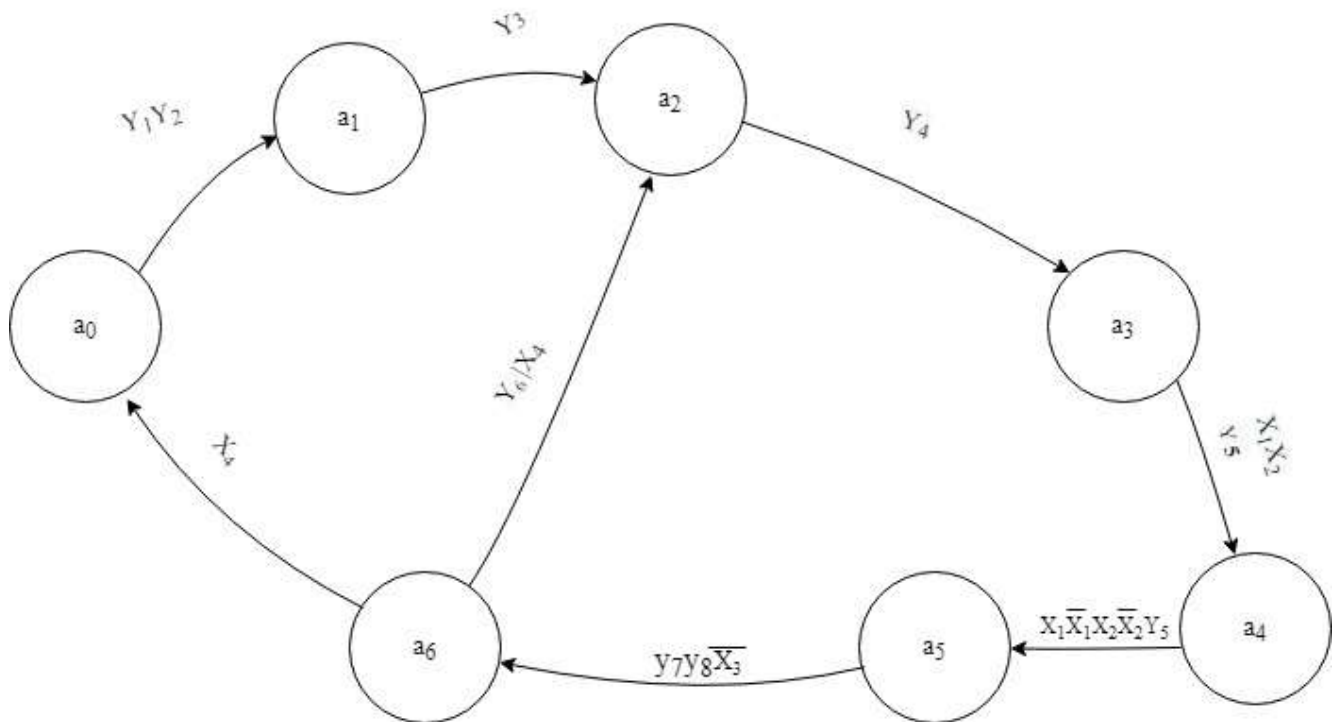


Рисунок 3.5 – Граф автомата Мілі

Кількісна різниця числа станів в автоматах Мура і Мілі призводить до різної складності регістра (R) і дешифратора (DC).

На дугах графа автомата зазначаються умови, при якому це перехід здійснюється, а через дріб - які сигнали виробляються в даному стані.

Зм..	Арк.	Нодокум.	Підпис	Дата

Цей автомат має всього 6 стану (a_0, a_1, a_2, a_3), для кодування яких досить 4 розряду:

$a_0 = 000$

$a_1 = 010$

$a_2 = 101$

$a_3 = 001$

$a_4 = 110$

$a_5 = 011$

$a_6 = 111$

Якщо в автоматі Мілі із зазначеного графа було отримано 5 станів, то для їх кодування використовуватиметься 3-х розрядний регістр (R) і дешифратор (DC) на 3 входи і 8 виходів .

В автоматі милі було визначено 4 стану, тому для кодування станів вводимо 2-х розрядний регістр R і дешифратор на 2 входи і 4 виходи на рисунку 3.6.

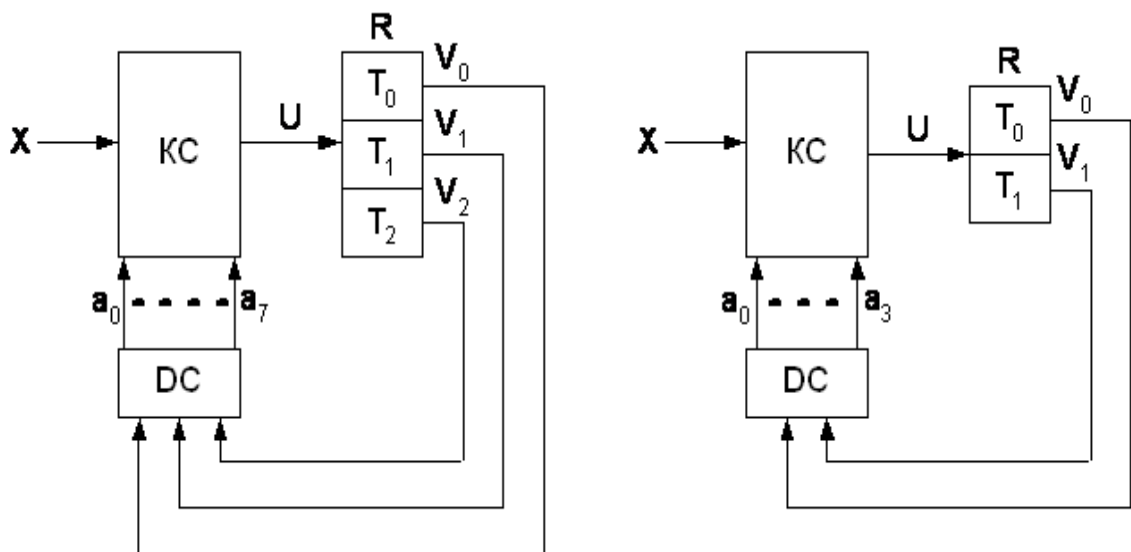


Рисунок 3.6 –Принципова схема

Таким чином, в структурах автомата Мілі визначені розрядністю регістрів R и складність дешифратора (DC).

Справжній синтез включає в себе всі наступні етапи:

- 1) вибір типу тригерів;

- 2) побудова таблиці переходів, виходів і збуджень;
- 3) запис логічних виразів для вихідних сигналів (Y) і сигналів збуджень (U);
- 4) побудова комбінаційної схеми.

Як і раніше виконаємо синтез КС спочатку для автомата МУРА, а, потім, для автомата милі.

Керуючий автомат зображений на електричній функціональній схемі(додаток А) і складається з трьох частин. У вхідній частині розташовані три D-тригери, які є трьома логічними елементами АБО, до яких подається вхідний сигнал, декодер та дві шини, одна шина необхідна для передачі сигналів, що надходять з декодера, а друга - для сигналів з виходів компаратора. Перетворення сигналу протягом одного такту часу виконується у перехідній частині автомата. Після проходження через логічні елементи ТА та (або) АБО, чи не роблячи того, сигнал трансформується і результат передається на шину (а), з якої він продовжує передаватися до програмованої логічної матриці. На основі системи рівнянь переходів будується перехідна частина. Вихідна частина будується на основі системи рівнянь виходів.

На цьому етапі виконується аналогічний процес, що відбувався в перехідній частині, окрім того, що сигнали подаються до вихідної шини u_i , з вихідної шини сигнал передається до вихідної матриці

3.3 Моделювання керуючого автомату на основі автомату Мілі в середовищі Quartus II

Моделюємо роботу моделі кінцевого автомату Мілі в середовищі Quartus II. Написана програма та модель успішно пройшли компіляцію. Для перевірки роботи автомата Мілі запустимо редактор Waveform Editor та відслідкуємо зміни та реакцію вихідних сигналів на різні вхідні сигнали та на різні внутрішні стани (рис. 3.7-3.9).

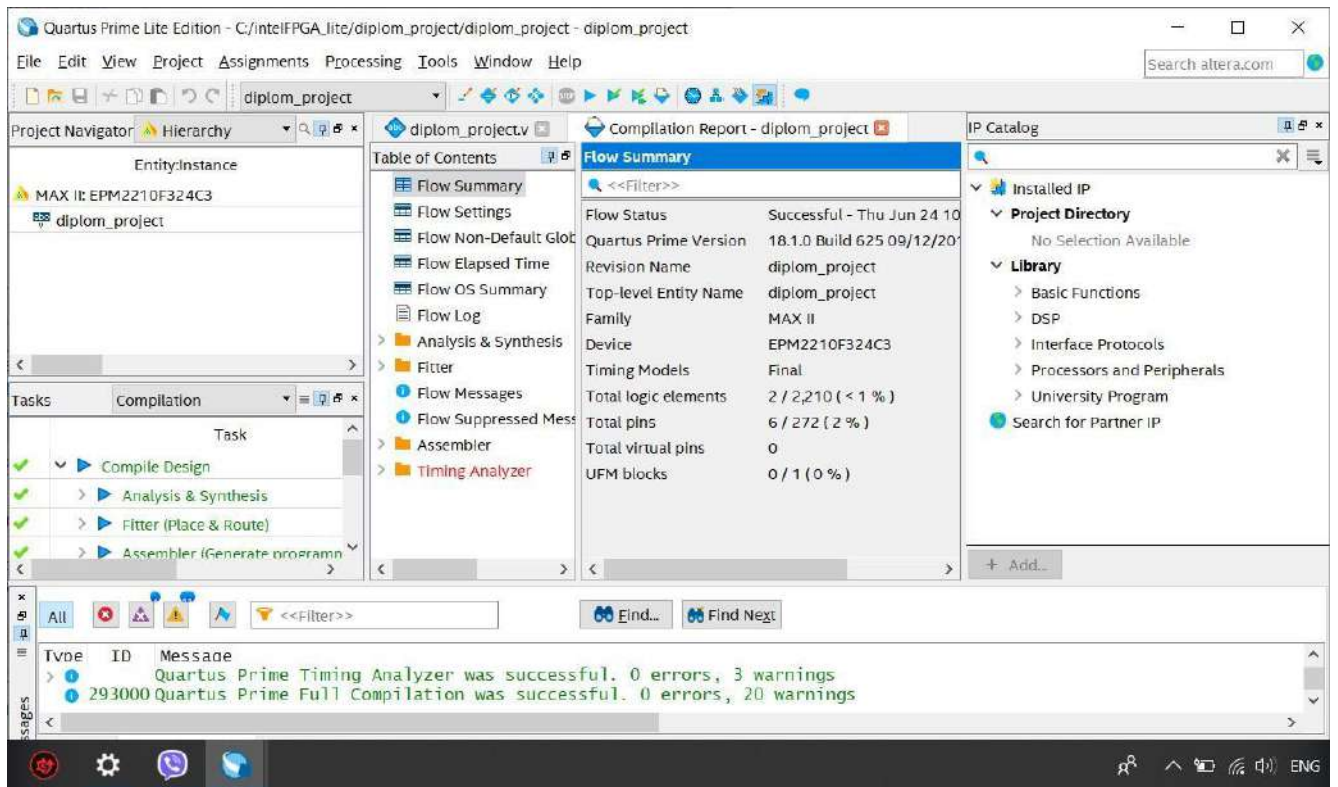


Рисунок 3.7 – Налаштування проекту

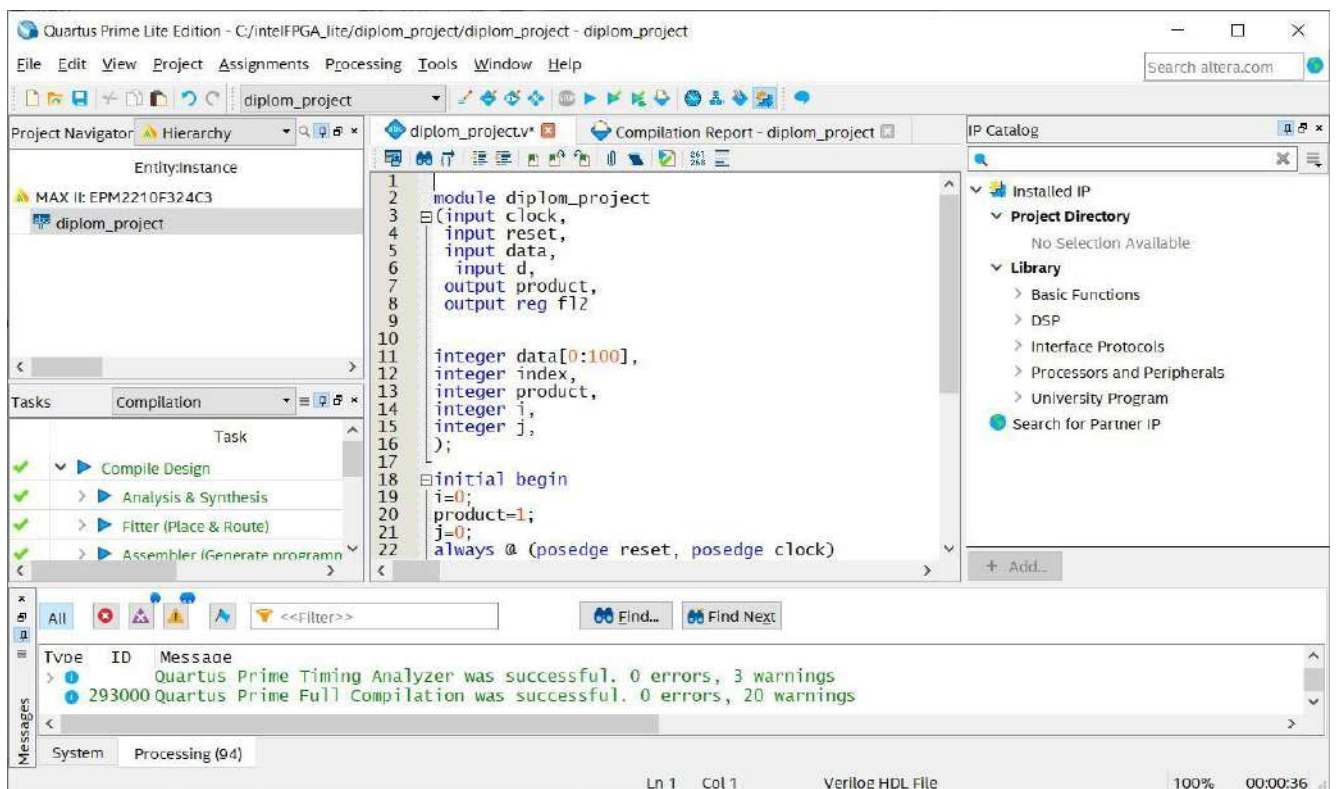


Рисунок 3.8 – Результат компіляції проекту

Зм..	Арк.	Нодокум.	Підпис	Дата
------	------	----------	--------	------

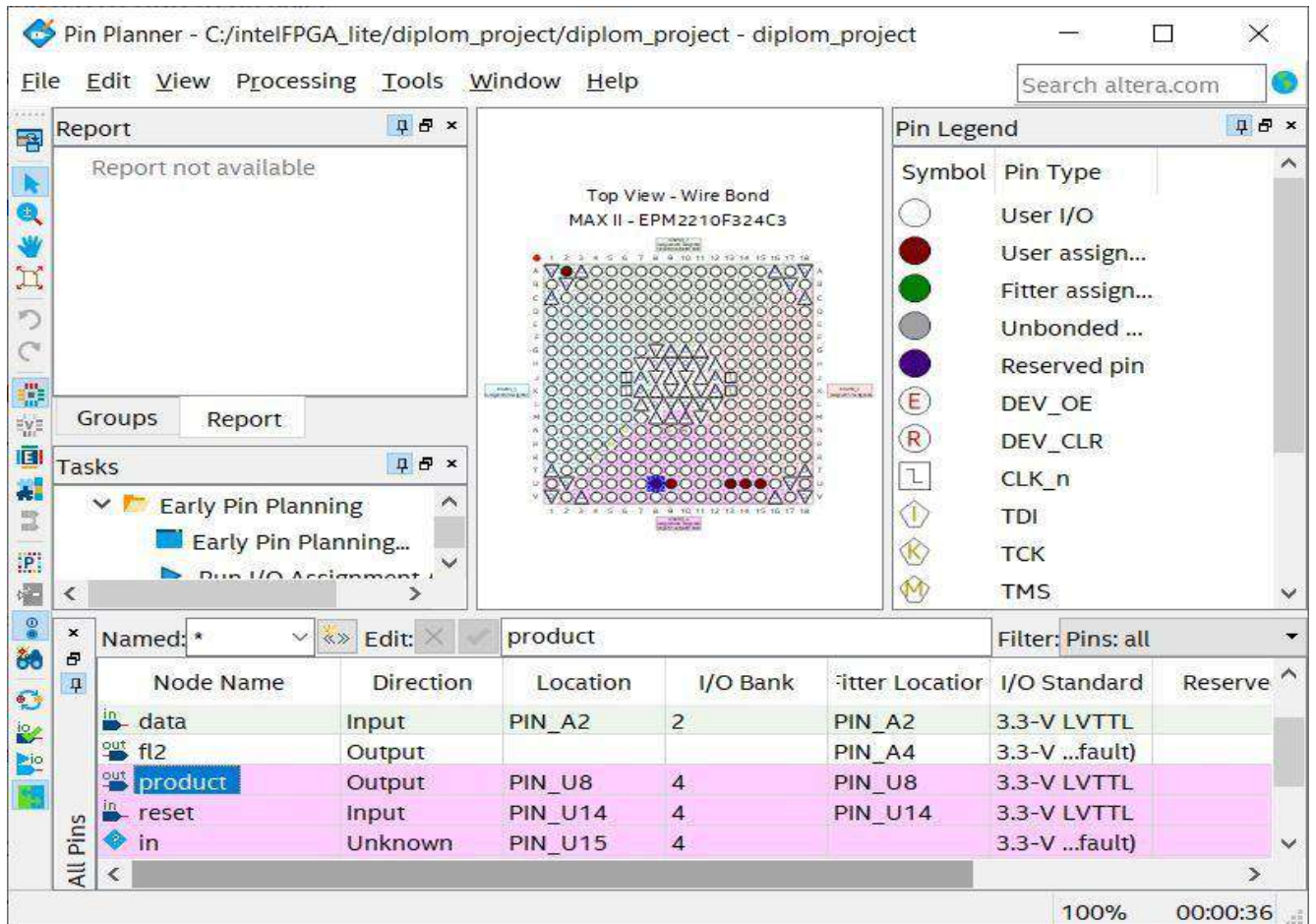


Рисунок 3.9 - Розподіл входів та виходів

В процесі моделювання розробляється мікропроцесорного пристрою будуються тимчасові діаграми роботи його функціональних блоків і вирішуються такі основні завдання:

- 1) перевірка правильності логічної структури розроблюваного устрою;
- 2) порівняння характеристик різних варіантів логічних схем;
- 3) виявлення явища змагань і ризику збою в логічних схемах.

Данна модель представляє собою програмний продукт, який розроблений на універсальному мові програмування Verilog. Для здійснення моделювання приймаються наступні початкові параметри: період імпульсу синхронізації елементів пам'яті - 1 секунда еталонного машинного часу; час моделювання 10 секунд еталонного машинного часу для керування мікропрограмним автоматом Мілі по числу стану в графі відповідних автоматів; метод моделювання - дискретний з фіксованим шагом; розмір кроку моделювання - фіксований,

автоматичний; режим автоматичний; для імітації вхідних сигналів використовуються блоки одиничного переходу сигналів у заданий час; для відображення результатів моделювання використовуються діаграми; моделювання здійснюється при виконанні та невиконанні логічних умов X_1 (наявність напруги в електричній мережі). Зовнішній вид вікна програми Мілі пристрою представлений на рисунку 3.10.

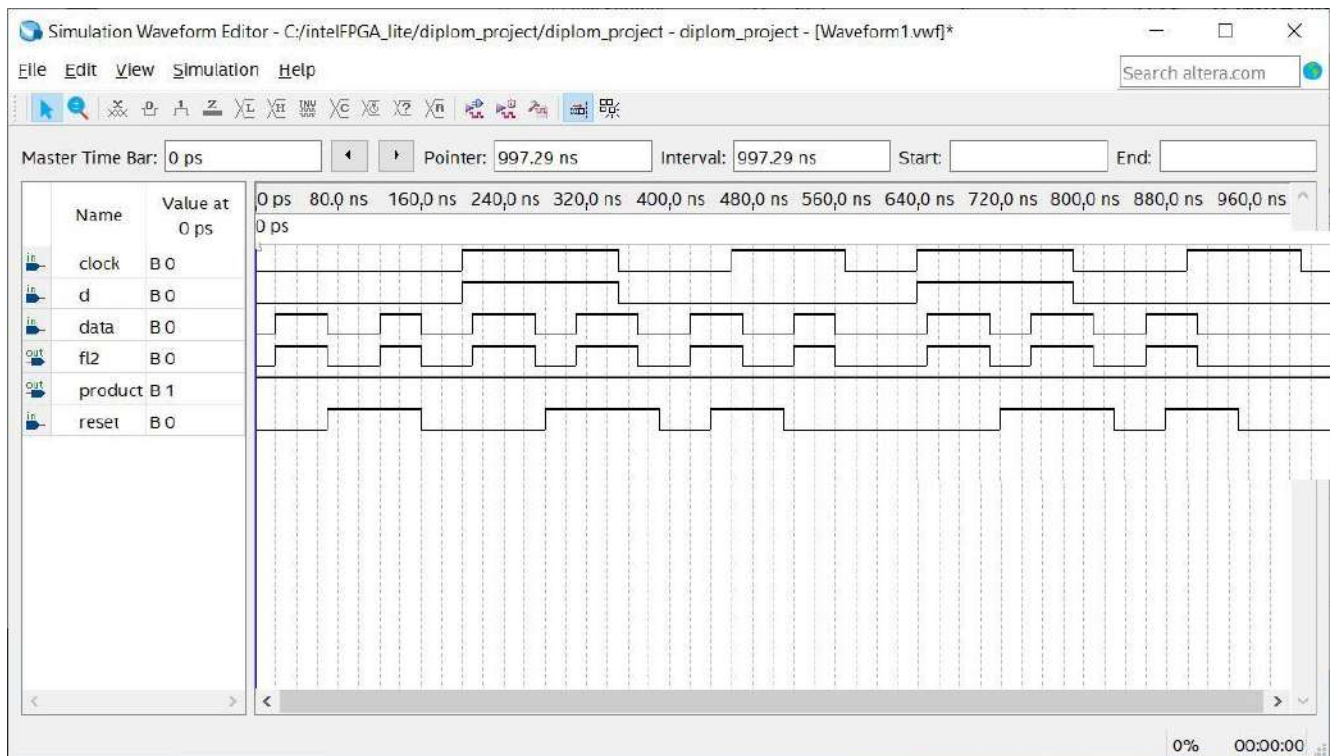


Рисунок 3.10 – Результат перевірки роботи програми в Waveform

При виконанні логічних умов x_1 (наявність значень елементів масиву) в результаті моделювання наведено дві діаграми (рис.3.4): діаграма вихідних сигналів (мікрокоманди); діаграма вхідних сигналів (логічних умов); діаграма состояний елементов памяти (D-триггеров). При невикористанні логічних умов x_1 в результаті моделювання наведено дві діаграми (рис.3.5): діаграма вихідних сигналів (мікрокоманда); діаграма вхідних сигналів (логічних умов); діаграма состояний элементов памяти (D-триггеров). Логічна схема та графічні моделі куруючого автомата пристрої представлені на (рис.3.5), ілюструє процес моделювання управління мікропрограмного автомата Мілі. При виконанні логічних умов x_1 у результаті моделювання наведено дві діаграми

(рис. 3.11): діаграма вихідних сигналів (мікрокоманда); діаграма вхідних сигналів (логічних умов); діаграма состояний елементів пам'яті (D-триггерів).

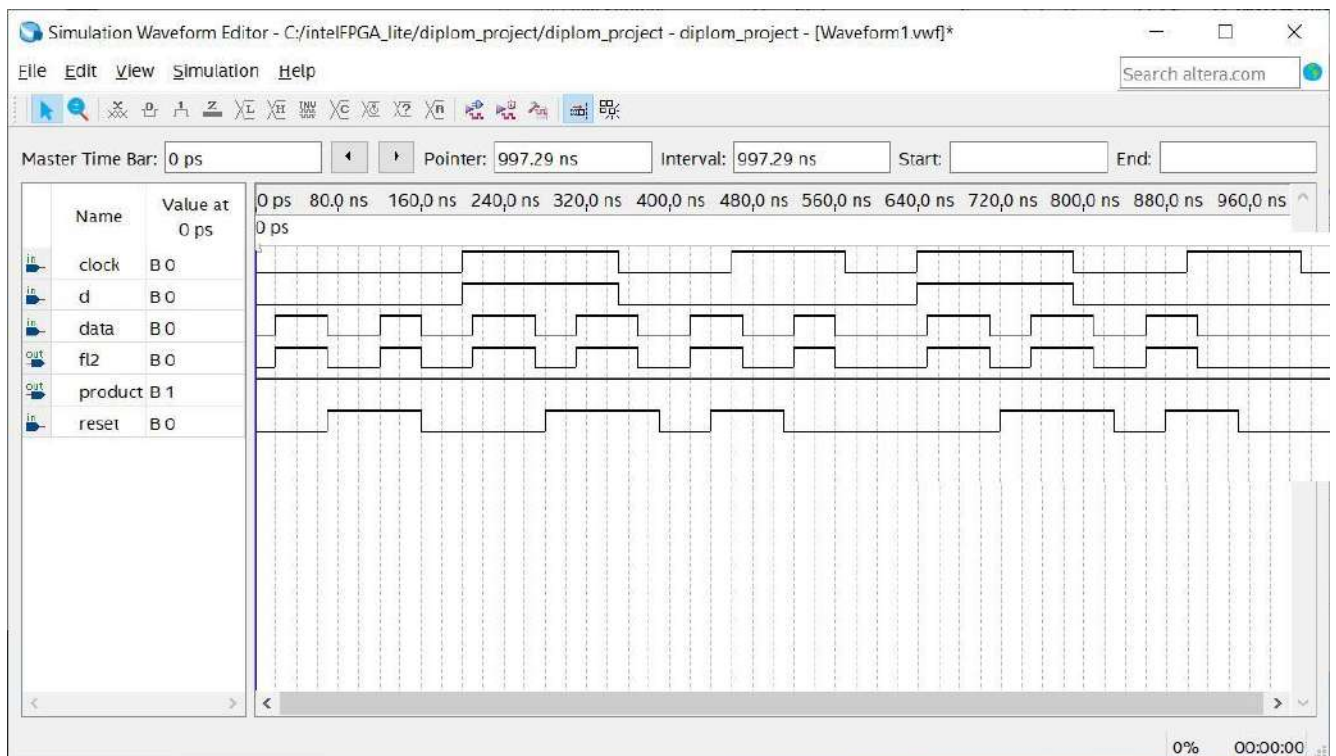


Рисунок 3.11 – Результат перевірки роботи програми в Waveform

При невикористанні логічних умов X_1 в результаті моделювання наступних діаграма вихідних сигналів (мікрокоманди); діаграма вхідних сигналів (логічних умов); діаграма состояний елементів пам'яті (D-триггерів). На діаграмі вихідних сигналів моделей управляючого мікропрограмного автомата Мілі показаний ступінчастий перехід від однієї мікрокоманди до подальшої відповідно до графіку автомата при виконанні логічних умов. За умови невикористання умов H_1 (відсутність напруги в електричній мережі) відбувається перехід до мікрокоманди Y_{10} (вихід заданого інтервалу вимірювань), що вказує на безошибочну роботу, розроблену моделями керуючих мікропрограмних автоматів Мілі та Мультиінформації. Діаграми стаціонарних елементів пам'яті моделей управляючого мікропрограмного автомата Мілі і Мура відповідають кодам стаціонарних даних автоматів.

З діаграми вихідних сигналів моделей керуючого автомата Мілі слід, що вихідні сигнали автомата виражається із затримкою на одному такті, що в

автоматі Мілі. Це повідомлення про різницю даних моделей, що замикається в тому, що в автоматі Мілі вихідний сигнал повертається одночасно з визиванням його вхідним сигналом. Таким чином, вихідний сигнал виявляється залежним від вхідного сигналу в попередній момент часу.

Наступним кроком після моделювання роботи проекту є створення працюючого макету, для перевірки працездатності розробленого методу. Щоб виготовити працюючий макет для цього нам знадобиться:

Перелік елементів, які можуть знадобитись при переносі проекту з середовища Quartus в фізичний розроблений нами спосіб вирішення завдання поставленого завдання(таблиця 3.1).

Таблиця 3.1 – Перелік основних елементів

Поз. Позначення	Найменування		Примітка
			Кільк.
	Мікросхеми		
DD1	FT232 FTDI	1	
DD2	ATMega16A Atmel	1	
DD3	Дисплей LCD WH1602A Winstar	1	

Кінець таблиці 3.1 – Перелік основних елементів

Поз. Позна- чення	Найменування	Кільк.	Примітка
	Світлодіоди		
HL1- HL5	АЛ307БМ ГОСТ 23448-79	5	
	Кварцеві генератори		
Q1	Генератор кварцевий ГК241-9Н5-6МГц		
	ТУ 307-182.004-96	1	
Q2	Генератор кварцевий ГК241-9Н5-16МГц		
	ТУ 307-182.004-96	1	
	Резистори		
R1	R-0603 4,7 kOhm 1 % 0,1 W TC100 50 V	1	
R2	R-0603 1 kOhm 5 % 0,063 W TC200 50 V	1	
R3, R4	R-0603 5,1 kOhm 5 % 0,1 W TC200 50 V	2	
R5-R8	R-0603 10 kOhm 5 % 0,1 W TC200 50 V	4	
R9, R10	R-0603 1000 kOhm 5 % 0,063 W TC200 50 V	2	
R11, R12	R-0603 5,1 kOhm 5 % 0,1 W TC200 50 V	2	
R13	R-0603 51 Ohm 5 % 0,063 W TC200 50 V	1	

Зм..	Арк.	Нодокум.	Підпис	Дата

КВРКІ. 170159.17.01.01 ПЗ

Ар

61

Повний перелік всіх потрібних елементів представлені у вигляді таблиці та розміщені в додатку В.

Далі нам потрібно об'єднати елементи на одній платі. Тому для цього необхідно правильно розмістити на друкованій платі, протравити її, припаяти всі елементи плати. та посадкове місце для мікросхеми.

Щоб автомат працював за нашим алгоритмом і виконував потрібні задачі потрібно запрограмувати мікросхему. Програмування мікросхеми виконується через спеціальний пристрій - програматор, на нього в посадкове місце встановлюється мікросхема і переноситься програма. Останнім кроком є підключення мікросхеми до плати автомата, подати живлення і плата готова до роботи.

3.4 Висновки

В ході виконання третього розділу кваліфікаційної було виконано апаратну реалізацію керуючого автомату на базі автомату МІЛІ. Для цього нам знадобились реалізувати схему електричну функціональну кінцевого автомату на базі автомабу Мілі. та змодельовано схему пристрою в середовищі Quartus II. Окрім цього підібрали всі потрібні елементи автомата для фізичної реалізації.

					КвРКІ. 170159.17.01.01 ПЗ	Ар
						62
Зм..	Арк.	№докум.	Підпис	Дата		

ВИСНОВКИ

В ході роботи над кваліфікаційною роботою було розроблено модель, спроектовано та реалізовано у вигляді схеми та підібрали необхідні елементи для подальшої роботи з керуючим автоматом, який виконує обчислення добутку непарних елементів двовимірного масиву $A[n, m]$, що належать інтервалу $[0; 100]$. керуючого автомату у вигляді автомату Мілі. Керуючу частину автомата синтезовано з допомогою елементів АБО, ТА НІ RS. Змодельований, спроектований та реалізований керуючий автомат на базі автомату Мілі на практиці використовується для обчислення добутку непарних елементів двовимірного масиву $A[n, m]$, що належать інтервалу $[0; 100]$.

					КвРКІ. 170159.17.01.01 ПЗ	Ар
						63
Зм..	Арк.	Нодокум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Илюхин А.В., Колбасин А.М. Логические автоматы. Типовые последовательностные схемы. Учебное пособие. В 2 ч. Москва: МАДИ, 2013. 52 с.
2. Информационная революция и становление информационного общества. Санкт-Петербург.: Невский институт языка и культуры, 2014. 22 с.
3. 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.
4. Лобанов А.И. Модели клеточных автоматов. "Компьютерные исследования и моделирование". 2010. Т.2. №3. С. 273–293.
5. Поликарпова Н.И., Шалыто А.А. Автоматное программирование. СанктПетербург: Питер, 2010. 176 с.
6. Кузенков М.В. Проектирование автомата Мура. Микропрограммный автомат Мили. Красноярск: СФУ ИКИТ, 2010. 11 с.
7. 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.
8. 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.
9. 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.
10. Микушин А. В. Сажнев А. М. Сединин В. И. Цифровые устройства и микропроцессоры. Санкт-Петербург: БХВ-Петербург, 2010. 640 с.
11. Говорущенко Т. О. Комп'ютерна логіка: практикум: навчальний посібник. Хмельницький: Хмельницький національний університет, 2018. 294 с.
12. Ожиганов А.А. Теория автоматов: учебное пособие. Санкт-Петербург: НИУ ИТМО, 2013. 84 с.

Зм..	Арк.	№докум.	Підпис	Дата

13. Джон Хопкрофт, Раджив Мотвани, Джеффри Ульман. Введение в теорию автоматов, языков и вычислений. 2-е издание. Москва: Диалектика-Вильямс, 2016. 528 с.

14. Кудрявцев В.Б. Теория автоматов: учебник для бакалавриата и магистратуры. Москва: ВАМ, 2018. 698 с.

15. Дж. Фон Нейман. Теория самовоспроизводящихся автоматов. Москва: ВАМ, 2010. 673 с.

16. Вельдер С.Э., Лукин М.А., Шалыто А.А., Яминов Б.Р. Верификация автоматных программ. Санкт-Петербург: Наука, 2011. 244 с.

17. Гуренко В.В. Введение в теорию автоматов: учебное пособие. Теория + Примеры. Москва: МГТУ имени Н.Э. Баумана, 2013. 63 с. 13. Иванов Н.М., Соловьева Т.Н. Конечные автоматы. Санкт-Петербург: ГУАП, 2015. 76 с.

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

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

16. Поликарпова Н.И., Шалыто А.А. Автоматное программирование. СанктПетербург: Питер, 2010. 176 с.

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

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

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

20. Титов И.И. Проектирование управляющих автоматов. Методические указания к выполнению курсовой работы для студентов направления 230100 - «Информатика и вычислительная техника» по профилю «Автоматизированные

					КВРКІ. 170159.17.01.01 ПЗ	Ар
Зм..	Арк.	№докум.	Підпис	Дата		65

системы обработки информации и управления». Нижний Новгород: НГТУ, 2012. 23 с.

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

23. Матвієнко М.П. Комп'ютерна логіка: навчальний посібник. Київ: ТОВ "Центр навчальної літератури", 2012. 288 с.

24. Матвієнко М.П. Архітектура комп'ютерів: навчальний посібник. Київ: ТОВ "Центр навчальної літератури", 2012. 264 с.

25. Матвієнко М.П. Комп'ютерна схемотехніка: навчальний посібник. Київ: ТОВ "Центр навчальної літератури", 2012. 190 с.

26. Никитин В.А. Схемотехника интегральных схем ТТЛ, ТТЛШ и КМОП: учебное пособие. Москва: НИЯУ МИФИ, 2010. 64 с.

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

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

29. 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.

30. 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.

31. 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.

32. 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.

33. 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.

Ім'я користувача:
Кафедра КІ

Дата перевірки:
29.06.2021 10:17:35 EEST

Дата звіту:
29.06.2021 10:18:04 EEST

ID перевірки:
1008363763

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005591

Назва документа: Барчук_Синтез та моделювання операційного автомату на основі автомату Мілі

Кількість сторінок: 65 Кількість слів: 10878 Кількість символів: 80823 Розмір файлу: 2.45 MB ID файлу: 1008433298

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

16.3%

Схожість

Найбільша схожість: 4.21% з Інтернет-джерелом (https://msn.khnu.km.ua/pluginfile.php/147614/mod_resource/content..)

12.4% Джерела з Інтернету

142

Сторінка 67

6.92% Джерела з Бібліотеки

86

Сторінка 68

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%

Вилучень

Немає вилучених джерел

Модифікації

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

Замінені символи

5353

Підозріле форматування

56

сторінок

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 0.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 52%**

ID: 95491 Название: Синтез та моделювання операційного автомату на основі автомату Мілі Добавлено в БД: 2021-06-29 Авторы: Д.О. Барчук Руководители: М.О. Березький Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	69250	568	329 (0%)	4 (1%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Барчук Денис Олександрович

Тема: Синтез та моделювання керуючого автомату на основі автомату Мілі

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень _____ Кількість сторінок записки _____

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є синтез та моделювання керуючого 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 та виконано постановку задачі дослідження. В другому розділі кваліфікаційної роботи проведено моделювання та проектування керуючого автомату на основі автомату Мілі. В третьому розділі кваліфікаційної роботи виконано апаратну реалізацію керуючого автомату на основі автомату Мілі та змодельовано схеми автомату в середовищі Quartus II.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: _____

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 високому науково-технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: відмінно

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

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

“ 2 ” червня 2021 р.

Фду (підпис)

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

Барчука Д.О.

ПІБ здобувача вищої освіти

ФПКТС, 4 курсу, групи КІ-17-1

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 наявність плагіату ознайомлений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (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.

24.06.21

дата



підпис

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

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

Назва: Синтез та моделювання операційного автомату на основі автомату Мілі

Автор: Барчук Денис Олександрович

Спеціальність: 123 – Компютерна інженерія

Освітня програма: освітньо-професійна

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

Після аналізу звіту подібності зроблено такий висновок:

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

Підтвердження:

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

- 1) запозичення 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) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;
- 4) в якості запозичень в окремих місцях системою зафіксовано послідовності чотирьохрозрядних двійкових кодів, які є вхідними даними до великої кількості задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 16.3% і адресується до 226 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

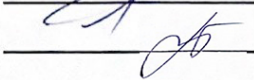
Керівник роботи

Гарант ОП

Завідувач кафедри КІСП


О. М. Березький


С. М. Лисенко


Т. О. Говорущенко