

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Програмно-технічний пристрій розпізнавання типу електронних компонентів та вимірювання їх основних параметрів
Назва теми

КВРКІ. 190246.19.22.48 ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

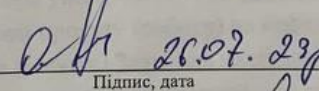
Назва

Виконав: студент IV курсу, група KI2-19-2


Підпис

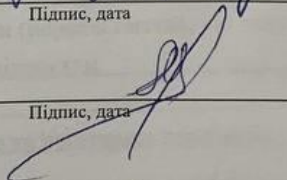
А.А. Стойко
Ініціали, прізвище

Керівник


Підпис, дата

В.М. Стецюк
Ініціали, прізвище

Нормоконтролер


Підпис, дата

С.М. Лисенко
Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
інженерії та інформаційних
систем


Підпис

Т.О. Говорущенко
Ініціали, прізвище

«26» червня 2023 р.

Хмельницький 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Г.О.Говорушенко

“ 11 ” 01 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Стойко Антону Андрійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-технічний пристрій розпізнавання типу електронних компонентів та вимірювання їх основних параметрів

Керівник проекту (роботи) Стежок В.М., д.т.н., проф.

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

Затверджена наказом ректора університету від 01.03.2023 р. № 5

2. Строк подання студентом проекту (роботи) на кафедру 23.06.2023 р.

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

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

Огляд приладів тестування мікросхем

Теоретичний опис проекту

Принципова схема пристрою та програмна реалізація

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

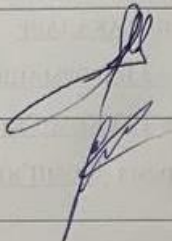
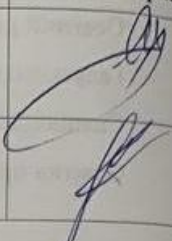
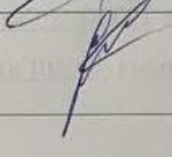

Фрагмент програмного коду для пристрою автоматичного розпізнавання типу електронних компонентів та вимірювання їх основних параметрів

Копія креслення схеми функціонування пристрою

Копія креслення схеми функційної пристрою

Копія креслення схеми принципової пристрою

6. Консультанти розділів дипломного проекту (роботи)

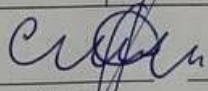
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КПС		
Антиплагиат	Нічепорук А.О., доцент кафедри КПС		

7. Дата видачі завдання « 01 » 03 2023 р.


КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	20.02.2023	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.03.2023	виконано
3	Робота над розділом 1 – огляд приладів тестування мікросхем	10.03.2023	виконано
4	Робота над розділом 2 – теоретичний опіс проекту	20.04.2023	виконано
5	Робота над розділом 3 – принципова схема пристрою та програмна реалізація	30.04.2023	виконано
6	Оформлення пояснювальної записки згідно вимог	31.05.2023	виконано
7	Попередній захист ВКР	02.06.2023	виконано
8	Захист ВКР на засіданні ЕК	Червень 2023 року	

Студент


Підпис А.А. Стойко
Ініціали, прізвище

Керівник роботи


Підпис В. М. Стецюк
Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-технічний пристрій розпізнавання типу електронних компонентів та вимірювання їх основних параметрів».

Автор роботи: Стойко Антон Андрійович.

Керівник роботи: Стецюк Василь Миколайович.

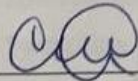
Пояснювальна записка: 71 с., 35 рис., 9 табл., 4 дод., 40 джерел.

Графічна частина: 3 креслення, фрагмент коду.

Метою роботи є розробка програмно-технічного пристрою розпізнавання типу електронних компонентів та вимірювання їх основних параметрів.

Об'єктом дослідження є програмно-технічні пристрої.

Предметом дослідження є сукупність необхідних умов, що забезпечують найкращий підхід до створення пристрою розпізнавання типу електронних компонентів (тестер мікросхем).



Підпис студента

26.06.2023

Дата

ЗМІСТ

ВСТУП.....	4
1 ОГЛЯД ПРИЛАДІВ ТЕСТУВАННЯ МІКРОСХЕМ	7
1.1. Аналіз стану справ в розглядуваній області (цифрових вимірювальних пристроїв).....	7
1.2. Основні характеристики приладів тестування	23
1.3. Висновок тестувань мікросхем	27
2 ТЕОРЕТИЧНИЙ ОПИС ПРОЕКТУ	29
2.1. Розробка структурної схеми пристрою контролю електронних компонентів.....	29
2.2. Розробка функціональної схеми	38
2.3. Висновок розробки схем.....	54
3 ПРИНЦИПОВА СХЕМА ПРИСТРОЮ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ... 56	
3.1. Допоміжне джерело живлення, перетворювач рівнів.....	56
3.2. Система управління живлення.....	58
3.3. Дистанційно керований модуль тестування мікросхем пам'яті.....	59
3.4. Розробка принципової схеми пристрою.....	61
3.5. Розробка алгоритма його роботи	62
3.6. Розробка монтажної схеми пристрою	68
3.7. Програмне забезпечення для програмування та управління мікроконтролером.....	70
3.8. Опис контрольованих об'єктів (мікросхеми серії 7400)	73
3.9. Аналіз та призначення будови мікроконтролера ATMEGA 128 Header board.....	75
3.10. Висновок програмної реалізації.....	77
ВИСНОВКИ	79
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	80

КвРКІ 190246.19.22.48 ПЗ								
Зм.	Арк.	№докум.	Підпис	Дата	Програмно-технічний пристрій розпізнавання типу електронних компонентів та вимірювання їх основних параметрів. Пояснювальна записка	Літера	Аркуш	Аркушів
Виконав		Стойко А.А.	<i>[Signature]</i>	26.08.2023		у		71
Перевір.		Стецюк В.М.	<i>[Signature]</i>	26.08.2023				
Н.контр.		Лисенко С.М.	<i>[Signature]</i>	16.09				
Затвер.		Говорухенко Т.О.	<i>[Signature]</i>					
					ХНУ КІ2-19-2			

ДОДАТОК А	81
Фрагмент програмного коду для пристрою автоматичного розпізнавання типу електронних компонентів та вимірювання їх основних параметрів ..	81
ДОДАТОК Б	86
Копія креслення схеми функціонування пристрою.....	86
ДОДАТОК В	87
Копія креслення схеми функційної пристрою	87
ДОДАТОК Г	88
Копія креслення схеми принципової пристрою.....	88

ВСТУП

В даний час, на практиці дуже поширюється виробництво радіоелектронних мікросхем і компонентів. Однак існує цілий ряд питань стосовно працездатності окремих елементів при їх виробництві. Відмова в роботі одного компонента може призвести до катастрофічних наслідків або поломок комп'ютерних систем.

Виробник не завжди гарантує якості поставляються компонентів і не може служити критерієм оцінки працездатності того чи іншого елемента. Не маючи в своєму розпорядженні вимірювального і випробувального обладнання, споживач часто обмежується лише перевіркою зовнішнього вигляду компонента. Однак для кожного елемента повинен бути проведений комплекс випробувань на відповідність необхідним характеристикам і працездатності. У зв'язку з цим, завдання розробки пристрою тестування цифрових мікросхем є вельми актуальною [5].

Тому після виготовлення необхідно проводити випробування, за результатами якого робити висновки про працездатність елемента. Існуюче на сьогоднішній день тестове обладнання для контролю якості цифрових мікросхем не зручні для приватного користування, так як мають великі габарити і складність в експлуатації, або ж навпаки, малі розміри, але можливість тестування тільки певної серії цифрових інтегральних мікросхем.

Під приладами контролю мікросхем, або називають тестерами розуміються пристрої, які проводять перевірку правильності функціонування. Для перевірки працездатності мікросхем доцільно застосовуються універсальні тестери, здатні випробувати кілька типів мікросхем. Представлене дослідження присвячене розробці програмно-технічного пристрою розпізнавання типу електронних компонентів та вимірювання їх основних параметрів.

Завдання дослідження можна сформулювати так:

– провести аналіз стану справ в розглядуваній області (цифрових вимірювальних пристроїв);

					КВРКІ 190246.19.22.48 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

- розглянути будову та призначення мікроконтролера ATMEGA 128 Header board;
- назвати такі основні характеристики приладів тестування, як тестер мікросхем-YBD-868, цифровий інтегральний тестер на базі AT89C52, тестер інтегральних схем SIMI100;
- дати опис контрольованих об'єктів (мікросхеми серії 7400);
- провести розробку структурної схеми пристрою контролю електронних компонентів;
- навести алгоритм його роботи, функціональну, принципovu та монтажну схеми пристрою;
- перелічити обране обладнання для розробки дистанційно керованого модуля тестування мікросхем пам'яті;
- розглянути будову допоміжного джерела живлення та перетворювача рівнів;
- розробити програмне забезпечення для програмування та управління мікроконтролером.

Методи дослідження: теоретичний аналіз наукової літератури; аналіз та узагальнення. В основу дослідження, лягли методи порівняльного аналізу та класифікації. Вирішення поставлених у роботі завдань здійснювалося з використанням системного підходу в доборі матеріалу, методів індуктивного і логічного аналізу, спостереження та статистичні методи аналізу літературних даних, програмування.

При проведенні наукових досліджень, залежно від поставлених завдань було використано наступні наукові методи: ситуаційний підхід, методи статистичного аналізу, порівняння, прогнозування, графічний та табличний.

Теоретична та практична цінність роботи полягає в наявності теоретичного матеріалу по дослідженню, відсіяного з-поміж іншого в процесі пошуку інформації по темі, та в систематизації матеріалу напрямку дослідження. Проведене дослідження має більш глибокий ступінь розробки напрямку дослідження,

відносно попередніх досліджень вчених, дисертантів та дослідників напрямку дослідження.

Новизна дослідження. Зроблено широкий літературний пошук з детальним аналізом наукової інформації. Проведено систематизацію та адаптацію отриманих літературних результатів.

Відповідно до мети і завдань дослідження структура дипломна роботи складається із вступу, трьох розділів, висновків, списку використаних джерел та додатків.

Джерелами інформації для вирішення перерахованих вище завдань є збірники наукових праць, монографії, періодична література, підручники та довідники, періодичні фахові журнали.

					КвРКІ 190246.19.22.48 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

1 ОГЛЯД ПРИЛАДІВ ТЕСТУВАННЯ МІКРОСХЕМ

1.1. Аналіз стану справ в розглядуваній області (цифрових вимірювальних пристроїв)

Тестер мікросхем є спеціальним інструментом для тестування мікросхем. Тестер є комбінованим електровимірювальним приладом для перевірки працездатності, визначення параметрів і налагодження радіоелектронного та електротехнічного пристрою. Тестування мікросхем є одним із ключових інструментів для забезпечення працездатності мікросхем. Технологія тестування ІС-це одна з трьох основних технологій підтримки розробки інтегральних схем.

На практиці існує багато різних типів тестерів. За класифікацією ІС виділяють цифровий інтегральний тестер і Аналоговий інтегральний тестер. За функціями тестера виділяють прилад для тестування функції мікросхем і прилад для тестування параметрів мікросхем. За формою тестера виділяють переносний тестер і настільний тестер [11].

Хоча буває такі різні тестери, але у них має кілька однакових основних особливостей. По-перше, можна випробувати аналогові, цифрові та аналого-цифрові гібридні інтегральні схеми. По-друге, можна випробувати динамічні параметри динамічні, параметри по постійному і параметри по змінному.

У Китаї тестери широко використовуються на практиці. На початку 1970-х років в Китаї почалися дослідження і розробка тестера інтегральних схем. Наприкінці 1980-х років рівень вітчизняного інтегрального тестера, особливо здатність самостійно проектувати, був значно покращений. Хоча вітчизняний тестер інтегральної схеми був певним розвитком, але в порівнянні з Міжнародним рівнем ще існує великий розрив.

Різні типи тестерів на внутрішньому ринку, малоємні і середні 80%, лише кілька з них використовують комп'ютерне тестування. Широкомасштабна система тестування ІС в основному покладається на імпорт для вирішення вітчизняних наукових досліджень, виробництва та тестування додатків.

					КВРКІ 190246.19.22.48 ПЗ	Арк. 7
Зм.	Арк.	№ докум.	Підпис	Дата		

Мікроконтролер. Спеціальна мікросхема, яка використовується для управління різними електронними приладами, вляється мікроконтролером.

Мікроконтролери – це одночипові мікрокомп’ютери, які об’єднують основну частину мікрокомп’ютера на одному чіпі.

Мікроконтролер можна класифікувати за різними аспектами: відповідно до ширини шини даних можна розділити на 8-розрядну, 16-розрядну і 32-бітну машину; відповідно до структури пам’яті можна розділити на структуру Гарварду і структуру фон Неймана; відповідно до типу вбудованої пам’яті програми можна розділити на OTP, маска, EPROM / EEPROM і флеш-пам’ять; відповідно до структури інструкцій можна розділити на мікроконтролери CISC і RISC [7].

Відрізняється від мікропроцесора інтегрованими в мікросхему пристроями введення-виведення, таймерами та іншими периферійними пристроями.

Крім того, ще є багато периферійні пристрої, використані в мікроконтролерах. Наприклад: контролери дисплеїв і клавіатур; аналого-цифрові і цифро-аналогові перетворювачі; контролери безколекторних двигунів, в тому числі крокових; широтно-імпульсні модулятори; радіочастотні приймачі і передавачі і т.д.

Мікроконтролер створився в 20 столітті, в середині 70-х років. Після 20 років розробки, вартість стає все нижче і нижче, а більш потужна продуктивність, що робить її застосування повсюдним у всіх областях. Такі, як управління двигуном, Побутова Електроніка, Ігрові пристрої, телефони, HVAC, захист будівлі та контроль доступу, промисловий контроль та автоматизація та побутова техніка (Пральні машини, мікрохвильові печі).

Проаналізуємо теоретичні підходи до обґрунтування сутності процесу тестування цифрових вимірювальних пристроїв. Функція НІ (логічне заперечення, інверсія, доповнення) означає обернення аргументу x_1 , тобто зміну його логічного значення на протилежне. Інверсія позначається у формулах рисою над змінними:

$y_1 = \overline{x_1}$ (читається: не x_1), а на умовних графічних позначеннях інвертора (елемента НІ) та інших елементів – маленьким колом.

Функція АБО (логічне додавання, логічна сума, диз'юнкція, об'єднання) – це операція збирання інформації від кількох джерел до одного каналу: на виході y_0 з'являється рівень лог. 1, якщо він надходить хоча б з одного з входів. Логічне додавання відрізняється від арифметичного останнім рядком таблиці: $1+1=1$, позначається $y_0 = x_1 + x_2 = x_1 \vee x_2$ (читається: x_1 або x_2), реалізується елементом АБО (елементом збирання, диз'юнктором). З інверсією на виході утворюється функція АБО-НІ (логічне додавання із запереченням, стрілка Пірса, заперечення диз'юнкції, доповнення об'єднання): $y_1 = x_1 \vee x_2 = x_1 + x_2$ (читається: ні x_1 , ані x_2), виконується елементом АБО-НІ (елементом Пірса) [10].

Функція І (логічне множення, добуток, кон'юнкція, перетин) – операція визначення збігу інформації на всіх входах; вона не відрізняється від двійкового множення одноцифрових чисел: набуває значення лог. 1, якщо високий рівень збігається на всіх входах. У виразах позначається $y_0 = x_1 x_2 = x_1 \wedge x_2$ (читається: x_1 і x_2), а на умовних позначеннях – знаком & (амперсанд, and). Виконується операція елементом І (елементом збігу, кон'юнктором). Симетрична їй функція І-НІ (логічне множення із запереченням, риска Шеффера, заперечення кон'юнкції, несумісність): $y_1 = \overline{x_1 x_2} = \overline{x_1} \wedge \overline{x_2}$ (читається: не x_1 або не x_2), реалізується елементом І-НІ (елементом Шеффера). Логічні функції АБО, І та їх заперечення можуть виконуватися над довільною кількістю змінних. Функція виключне АБО (додавання або сума за модулем 2, нерівнозначність, антиеквівалентність) відрізняється від функції АБО тим, що під час додавання кожної пари одиниць утворюється нуль; найчастіше операцію над багатьма змінними називають додаванням за модулем 2, а над двома аргументами – виключним АБО. Позначається функція за допомогою символу псевдо-плюс: $y_0 = x_1 \oplus x_2 = x_1 \overline{x_2} + \overline{x_1} x_2$ (читається: або x_1 , або x_2 ; x_1 або x_2 виключно), а на графічних зображеннях – символом M^2 та (для двовходових елементів) позначкою $\neq 1$. Елемент, що її реалізує, з довільною кількістю входів звичайно називають суматором за модулем 2, а двовходовий, найчастіше, – елементом виключне АБО (елементом нерівнозначності). Суміжна функція виключне АБО-НІ (рівнозначність,

					КВРКІ 190246.19.22.48 ПЗ	Арк. 9
Зм.	Арк.	№ докум.	Підпис	Дата		

еквівалентність): $y_1 = x_1 \oplus x_2 = x_1 \sim x_2$ (читається: x_1 як x_2) виконується цифровим пристроєм, який теж називається елементом виключне АБО-НІ (елементом рівнозначності, еквівалентором).

Функція заборона (збіг із заборною, заперечення імплікації, різниця) означає, що вхідна змінна x_1 надходить до виходу лише за відсутності заборони x_2 (якщо $x_2=0$), інакше на виході є рівень лог. 0. Найчастіше під цією функцією розуміють логічне множення з інверсією по одному з двох входів: $y_0 = x_1/x_2 == x_1x_2$ (читається: x_1 , але не x_2 ; x_1 і не x_2). Проте інколи її зручно тлумачити як своєрідне логічне віднімання $x_1 / x_2 = x_1 - x_2$: з колонки x_1 вилучаються одиниці (якщо вони є) стовпця x_2 . Реалізується вона елементом заборони (елементом І з інверсним входом) [2].

Функція імплікація (розділення із заборною, селекція) отримала назву з логіки висловлень через зв'язку «якщо $x_1=1$, то $y_1=x_2$ » (інакше $y_1=1$), проте часто має тлумачення логічного додавання з інверсією одного з двох входів (або функції заборони із запереченням по виходу): $y_1 = x_1 \rightarrow x_2 = \overline{x_1} + x_2$ (читається: якщо x_1 , то x_2 ; не x_1 або x_2). Елемент, що реалізує таку функцію, називається імплікатор (селектор).

Не всі перелічені логічні функції є незалежні. Такий їх набір, за допомогою якого можна утворити логічну функцію будь-якої складності, становить функціонально повну систему або базис. Система функцій НІ, АБО, І утворює булів базис.

Проте й ця система є надлишкова: вона лишається повною й у тому разі, якщо вилучити з неї один з елементів, крім НІ. Мінімально повною є система, з якої не можна вилучити жодної функції без утрати її повноти. Можна утворити функціонально повні системи, у тому числі й мінімальні, з різних наборів функцій, але обов'язково в них має бути інверсія [8].

Практичного значення крім булевого набули базиси з двох елементів: 1) І, НІ та 2) АБО, НІ, а частіше навіть з одного елемента: 1) І-НІ та 2) АБО-НІ; через це елементи І-НІ та АБО-НІ називають універсальними. Проте для зменшення

					КВРКІ 190246.19.22.48 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		

кількості елементів, необхідних для побудови пристрою, забезпечення гнучкості проектування використовують і складніші функції.

Отже, алгебра логіки оперує з логічними або двійковими змінними й функціями, які можуть набувати двох значень: 0 або 1. Перетворення в алгебрі логіки з метою аналізу, синтезу, доведення положень тощо здійснюють, користуючись, головним чином, основними логічними функціями НІ, АБО, І, але під час практичної реалізації цифрових пристроїв широко застосовують елементи, що виконують комбіновані функції І-НІ, АБО-НІ, ІАБО-НІ, виключне АБО тощо.

У цифровій електроніці відомі основні логічні функції І, АБО, НІ, І-НІ, АБО-НІ, додавання по модулю 2 (виняток або), які знаходять найбільш широке застосування при реалізації цифрових пристроїв різного призначення. Зазначені логічні функції для випадку двох аргументів підкоряються таблиці істинності.

При поданні логічної функції математичним виразом використовують два види її подання. Диз'юнктивної нормальною формою (ДНФ) називається логічна сума елементарних логічних творів, в кожне з яких аргумент або його заперечення входять 1 раз [1].

ДНФ може бути отримана з таблиці 1.1 наступним чином: для кожного набору аргументів, на якому функція дорівнює «1», записують елементарні твори змінних, причому змінні, значення яких дорівнює нулю, записують з інверсією. Отримані твори, які носять назву конституант одиниці або мінтермів, підсумовують [3].

Таблиця 1.1 - Істинності для логічних функцій

Комбінації аргументів x_2x_1	Функція І $(y = x_2x_1)$	Функція або $(y = x_2 + x_1)$	Функція І-НІ $(y = \overline{x_2 \cdot x_1})$	Функція Або-НІ $(y = \overline{x_2 + x_1})$	Функція Виключення або $(y = x_2 \oplus x_1)$
00	0	0	1	1	0
01	0	1	1	0	1
10	0	1	1	0	1
11	1	1	0	0	0

Наприклад, нехай задана логічна функція трьох змінних, яка дорівнює одиниці в разі, якщо хоча б дві з вхідних змінних рівні «1». Потрібно записати ДНФ цієї функції.

Для даної логічної функції ДНФ має вигляд:

$$y(x_2, x_1, x_0) = \bar{x}_2 x_1 x_0 + x_2 \bar{x}_1 x_0 + x_2 x_1 \bar{x}_0 + x_2 x_1 x_0 . \quad (1.1)$$

Диз'юнктивна нормальна форма (ДНФ), отримана підсумовуванням конститuent одиниці, називається досконалою (ДДНФ).

Кон'юнктивною нормальною формою (КНФ) називається логічний добуток елементарних сум, в кожному з яких аргумент або його заперечення входять один раз.

КНФ може бути отримана з таблиці 1.2: для кожного набору аргументів, на якому функція дорівнює 0, складають елементарну суму, причому змінні, значення яких дорівнює 1, записуються з запереченням.

Отримані суми, які носять назву конститuent нуля або макстермів, об'єднують операцією логічного множення. Наприклад, кон'юнктивна нормальна форма (КНФ) для функції з попереднього прикладу має вигляд:

$$y(x_2, x_1, x_0) = (x_2 + x_1 + x_0) + (x_2 + x_1 + x_0) + (x_2 + x_1 + x_0) + (x_2 + x_1 + x_0) . \quad (1.2)$$

Таблиця 1.2 - Логічна функцію у вигляді таблиці істинності

x ₁ 0	x ₂	X ₁	x ₀	У
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

КНФ також називається досконалою, тому що кожна елементарна сума містить всі змінні. Іноді зручніше користуватися не самою логічною функцією, а її інверсією.

В цьому випадку при використанні вищеописаних методик для запису ДДНФ треба використовувати нульові, а для запису досконалою кон'юнктивною нормальною формою (ДКНФ) одиничні значення функції.

Наприклад, для логічної функції попереднього прикладу ДДНФ і ДКНФ інверсної функції мають вигляд:

$$\text{ДДНФ: } \bar{y}(x) = \overline{x_2 x_1 x_0} + \overline{x_2 x_1 x_0} + \overline{x_2 x_1 x_0} + \overline{x_2 x_1 x_0} \quad (1.3)$$

$$\text{ДКНФ: } \bar{y}(x) = (x_2 + \bar{x}_1 + \bar{x}_0) + (\bar{x}_2 + x_1 + \bar{x}_0) + (\bar{x}_2 + \bar{x}_1 + x_0) + (\bar{x}_2 + \bar{x}_1 + \bar{x}_0) . \quad (1.4)$$

Іноді для скорочення запису логічну функцію представляють послідовністю десяткових чисел.

Для представлення логічної функції послідовністю чисел задають десяткові значення конститuant одиниці або нуля.

Наприклад, запис логічна функція з попереднього прикладу у вигляді послідовності чисел має вигляд [12]:

$$\text{ДДНФ: } y(x_2, x_1, x_0) = \Sigma(3,5,6,7) = V(3,5,6,7) , \quad (1.5)$$

$$\text{ДКНФ: } y(x_2, x_1, x_0) = \Pi(0,1,2,4) = \Lambda(0,1,2,4) = \&(0,1,2,4) . \quad (1.6)$$

Перехід від логічної функції до логічної схеми. Для побудови логічної схеми, які призначені для виконання логічних операцій, зазначених у логічній функції, розташовувати, починаючи від входу в порядку, зазначеному в булевому вираженні.

Наприклад, логічна схема пристрою, що реалізує логічну функцію:

$$F = \bar{x}_2 x_1 x_0 + x_2 \bar{x}_1 x_0 + x_2 x_1 \bar{x}_0 + x_2 x_1 x_0 \quad (1.7)$$

має вигляд, представлений на рисунку. 1.1.

Синтез логічних пристроїв в заданому базисі логічних елементів. До сих пір для побудови структури логічних пристроїв користувалися функціонально повною системою логічних елементів (ЛЕ), що реалізують три основні логічні операції І, АБО, НЕ.

Однак на практиці, з метою зменшення номенклатури використовуваних МС, часто користуються функціонально повною системою ЛЕ в складі двох, що виконують операцій І-НІ, АБО-НІ. Будь-яку логічну функцію можна записати в заданому базисі ЛЕ.

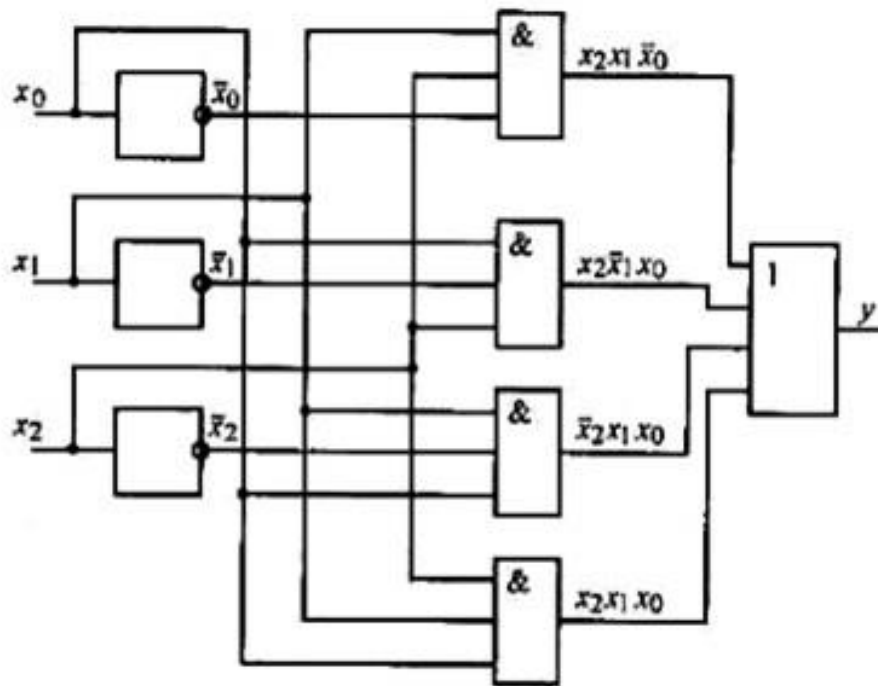


Рисунок 1.1- Схема логічна програмно технічного комплексу розпізнання електроних компонентів

Якщо заданий базис і-ні, то шляхом подвійного інвертування вихідного виразу або його частини і застосування теорем де Моргана логічна функція

приводиться до виду, що містить тільки операції логічного множення і інвертування.

Якщо ж заданий базис АБО-НІ, вихідну логічну функцію тими ж прийомами призводять до виду, що містить тільки операції логічного додавання і інверсії.

Далі логічний вираз записується через умовні позначення обраних операцій [18].

Наприклад, вихідна ДНФ в базисі І – не має вигляд:

$$y = x1x3 + \overline{x0}x2\overline{x3} + x0\overline{x1}x2 = \overline{\overline{x1x3 + \overline{x0}x2\overline{x3} + x0\overline{x1}x2}}$$

$$= \overline{(x1x3)(\overline{x0}x2\overline{x3})(x0\overline{x1}x2)} = (x1|x3)(\overline{x0}|x2|\overline{x3})(x0|\overline{x1}|\overline{x2}). \quad (1.8)$$

Аналогічно, КНФ в базисі або-НЕ має вигляд:

$$y = \overline{(x0 + x3)(\overline{x0} + x1 + x2)(\overline{x1} + \overline{x2} + \overline{x3})} =$$

$$= \overline{(x0 + x3) + (\overline{x0} + x1 + x2) + (\overline{x1} + \overline{x2} + \overline{x3})} =$$

$$= (x0 \downarrow x3) \downarrow (\overline{x0} \downarrow x1 \downarrow x2) \downarrow (\overline{x1} \downarrow \overline{x2} \downarrow \overline{x3}). \quad (1.9)$$

Розглянемо приклад розв'язку: логічна функція задана виразом:

$$y = x0x3 + \overline{(\overline{x0} + x2 + \overline{x3})}(x1 + \overline{x2}) \quad (1.10)$$

Таблиця 1.3 - Істинності логічного виразу

a	b	c	d	f
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0

Кінець таблиці 1.3 - Істинності логічного виразу

0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Привести логічну формулу до базису І-НІ, АБО-НІ.

Рішення:

а) наводимо функцію до базису І-НЕ:

$$y = y_1 + \overline{y_2}y_3 = y_1 + \overline{\overline{y_2}y_3} = y_1 + \overline{(y_2|y_3)} = y_1 + \overline{\overline{(y_2|y_3)}} = \overline{\overline{\overline{y_1}(\overline{y_2|y_3})}} = \overline{\overline{y_1|(\overline{y_2|y_3})}} \quad (1.11) \quad \overline{y_1} = \overline{x_0x_3} = x_0|x_3$$

$$y_2 = \overline{x_0} + x_2 + \overline{x_3} = x_0\overline{x_2}x_3 = \overline{x_0\overline{x_2}x_3} \quad (1.12)$$

$$y_3 = x_1 + \overline{x_2} = \overline{\overline{x_1 + \overline{x_2}}} = \overline{\overline{x_1}x_2} = \overline{x_1|x_2} \quad (1.13)$$

$$y = (x_0|x_3)|(\overline{(x_0|\overline{x_2}|x_3)}|(\overline{x_1|x_2})) \quad (1.14)$$

б) наводимо функцію до базису або-НЕ:

$$y = y_1 + \overline{y_2}y_3 = y_1 + \overline{\overline{y_2} + y_3} = y_1 + \overline{(y_2 + \overline{y_3})} = y_1 + \overline{\overline{(y_2 + \overline{y_3})}} =$$

$$= \overline{y1 \downarrow (y2 \downarrow \overline{y3})} \quad (1.15)$$

$$y1 = x0x3 = \overline{\overline{x0x3}} = \overline{\overline{x0} + \overline{x3}} = \overline{x0} \downarrow \overline{x3} \quad (1.16)$$

$$y2 = \overline{x0} + x2 + \overline{x3} = \overline{\overline{\overline{\overline{\overline{x0} + x2 + \overline{x3}}}}} = \overline{\overline{x0} \downarrow x2 \downarrow \overline{x3}} \quad (1.17)$$

$$y3 = x1 + \overline{x2} = \overline{\overline{x1 + \overline{x2}}} = \overline{x1 \downarrow \overline{x2}} \Rightarrow \overline{y3} = x1 \downarrow \overline{x2}. \quad (1.18)$$

В якості прикладу приведемо ще один приклад з обраного напрямку дослідження. Відповідно до варіанту написати канонічну суму мінтермів.

Варіант 0.

$$f = ab\bar{c} + ab + bc + \bar{a}cd + abcd + \bar{a}b\bar{c}$$

Заповнюється таблиця істинності для даного логічного виразу (таблиця 1.3)

Проведемо визначення мінімального вираження логічної функції. Для визначення мінімального вираження логічної функції кожен контур виписується і мінімізується

$$1. f1 = \bar{a}bc\bar{d} + \bar{a}bcd + \bar{a}b\bar{c}d + \bar{a}bc\bar{d} + abc\bar{d} + abcd + ab\bar{c}d + abc\bar{d}$$

$$f1 = b(\bar{a}c\bar{d} + \bar{a}cd + \bar{a}c\bar{d} + \bar{a}c\bar{d} + ac\bar{d} + acd + a\bar{c}d + a\bar{c}d);$$

$$f1 = b(1) = b$$

$$2. f2 = \bar{a}bcd + \bar{a}b\bar{c}d$$

$$f2 = \bar{a}cd(\bar{b} + b)$$

$$f2 = \bar{a}cd(1) =$$

$$= \bar{a}cd$$

Мінімальний логічний вираз буде:

$$f = f1 + f2 = b + \bar{a}cd.$$

В якості прикладу змодельємо в середовищі Proteus дешифратор чисел (від 0 до 9 у десятковому представленні) з двійкового коду в символи семисегментного індикатора (стилізовані арабські цифри), (рис. 1.2).

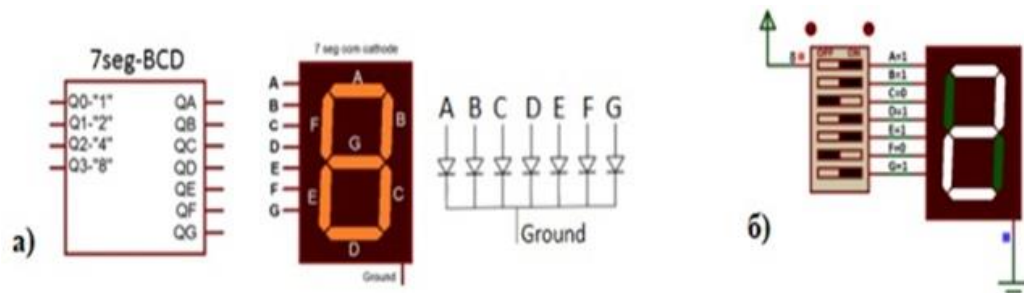


Рисунок 1.2- Позначення дешифратора кодів, розташування сегментів індикатора та схема його внутрішньої будови а) та приклад формування символу «2» вручну б)

Для цього будемо таблицю відповідності чисел у двійковому коді і кодів сегментів індикатора (табл. 1.4): у чотирьох стовпчиках Q0, Q1, Q2, Q3 за кількістю адресних входів сформуємо сигнали; у наступних восьми стовпчиках прописуємо «зображення» арабської цифри (наприклад, для «1» мають бути підсвічені сегменти «В» та «С», тому у відповідних комірках ставимо 1, у інших – нулі). Оскільки обрана нами схема індикатора має спільний катод, то, приклавши до деякого сегменту напругу логічної одиниці (+4В), ми змусимо цей сегмент світитись.

Таблиця 1.4 - Відповідності арабських цифр їх двійковим кодам та кодам керування семисегментним індикатором

Арабська цифра	Q3	Q2	Q1	Q0	QA	QB	QC	QD	QE	QF	QG
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0

Кінець таблиці 1.4 - Відповідності арабських цифр їх двійковим кодам та кодам керування семисегментним індикатором

8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

Для формування кодів керування семисегментним дешифратором у середовищі ISIS Proteus складаємо допоміжну схему із елементів DIPSWC_7 та 7SEG-COM-CAT показаного на рисунку. 1.3. Ця схема надає можливість в інтерактивному режимі ввімкнути / вимкнути довільний сегмент. Формуємо коди арабських цифр від нуля до дев'яти. Відповідні коди заносимо в таблицю (табл. 1.5), яка стане основою для формування принципіальної електричної схеми дешифратора. Аналіз апаратної реалізації вузла керування сегментом показує, що для побудови логічної функції більш економно обирати «нулі», якщо у стовпці менше нулів, і «одиниці» – в іншому випадку (для сегменту «А» наводимо таблицю таких функцій – табл. 1.5).

Таблиця 1.5 - Логічні функції для сегменту «А»

Аргументи				Функція	Реалізація функції	
03	02	01	Q0	0А	Нулі	Одиниці
0	0	0	0	1		$nx0 \cdot nx1 \cdot nx2 \cdot nx3$
0	0	0	1	0	$nx0+nx1+nx2+nx3$	
0	0	1	0	1		$x0 \cdot nx1 \cdot nx2 \cdot nx3$
0	0	1	1	1		$x0 \cdot x1 \cdot nx2 \cdot nx3$
0	1	0	0	0	$x0+nx1+nx2+nx3$	
0	1	0	1	1		$x0 \cdot nx1 \cdot x2 \cdot nx3$
0	1	1	0	1		$nx0 \cdot x1 \cdot x2 \cdot nx3$
0	1	1	1	1		$x0 \cdot x1 \cdot x2 \cdot nx3$
1	0	0	0	1		$nx0 \cdot nx1 \cdot nx2 \cdot x3$
1	0	0	1	1		$x0 \cdot nx1 \cdot nx2 \cdot x3$

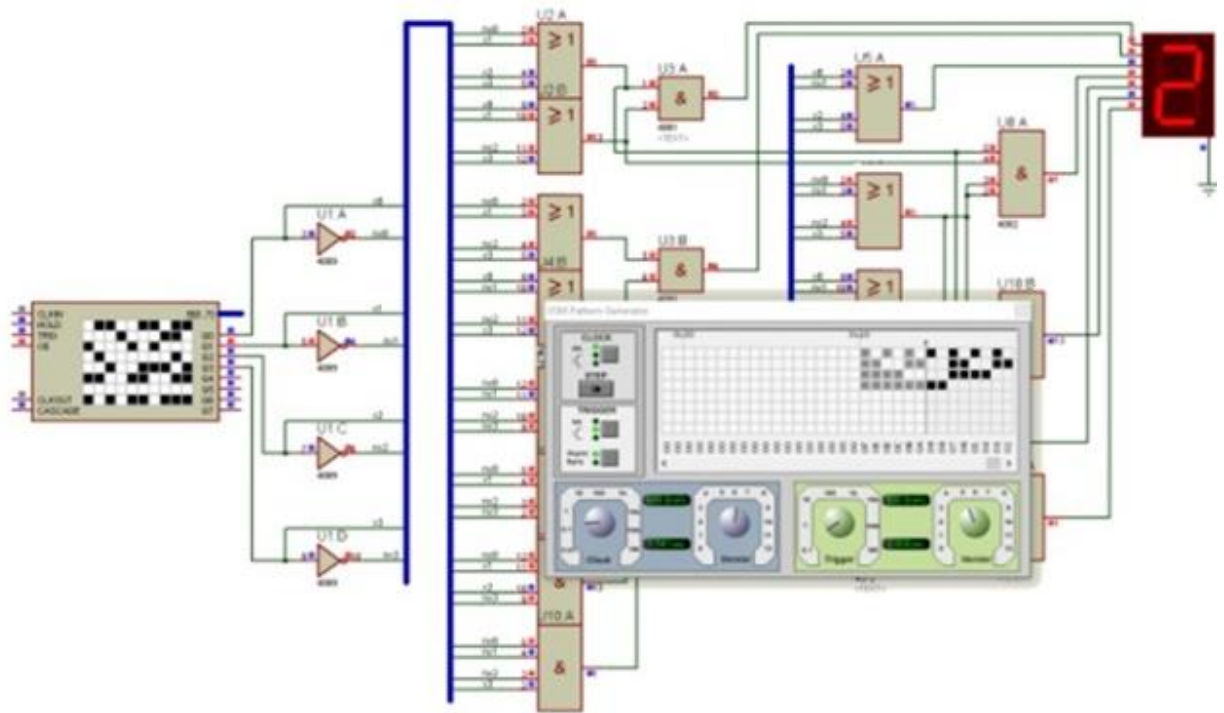


Рисунок 1.5- Візуалізація роботи семисегментного дешифратора (код на вході – 0010 відповідає арабській цифрі «2» на дешифраторі)

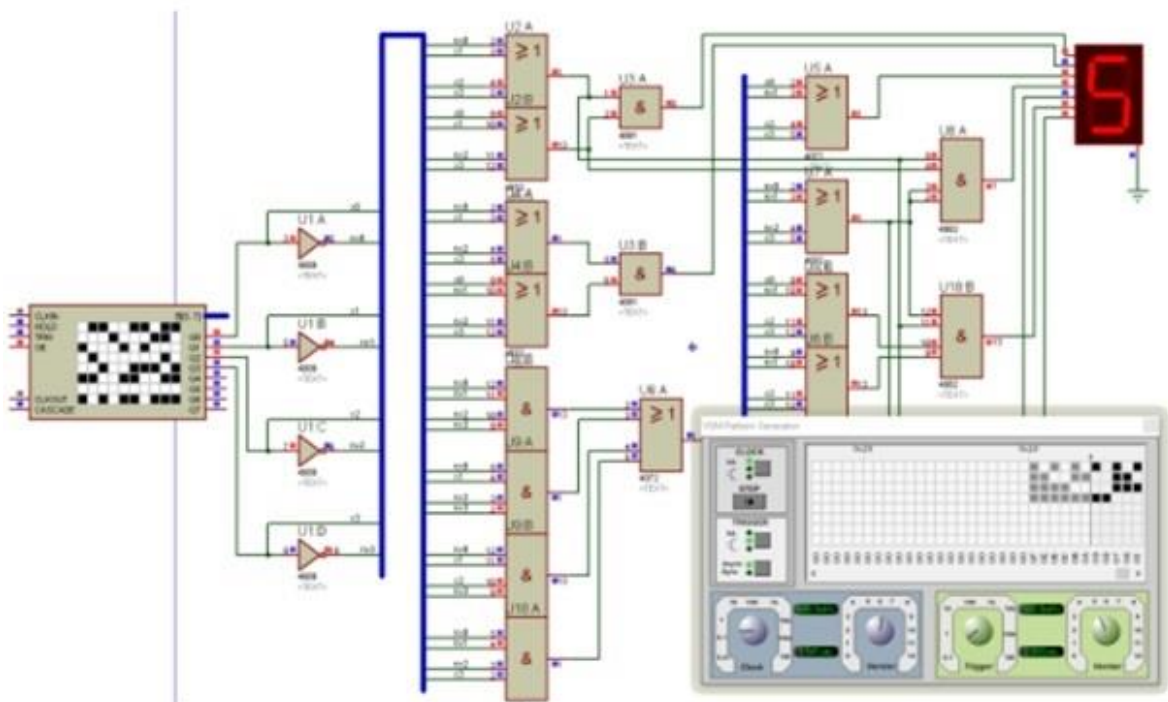


Рисунок 1.6 - Візуалізація роботи семисегментного дешифратора (код на вході – 0101 відповідає арабській цифрі «5» на дешифраторі)

мікросхем YBD868 – точний і надійний, і простий в експлуатації. Тестер мікросхем YBD868 і його принципова схема показані на рисунку. 1.8 а).

Основні функції цього тестера включаються в себе: Перевірка функції пристрою; ідентифікація типу пристрою; динамічне старіння пристрою; запит на заміну пристрою.

Тестер мікросхем YBD868 широко використовується в техобслуговуванні, тестуванні різних типів комп'ютерів, обладнання для промислової автоматизації, великого медичного обладнання, верстатів з ЧПУ, комп'ютерної периферії, програмно-керованих вимикачів, електронної апаратури, обладнання цифрового зв'язку, електронних релейних пристроїв і різних видів електронних продуктів. За допомогою тестера мікросхем YBD868 можна випробувати такі серії мікросхеми: 1. Серія ПЪ54; 2. Серія TTL55; 3. Серія TTL74; 4. Серія TTL 75; 5. Серія CMOS 14; 6. Серія CMOS40; 7. Серія CMOS 45; 8. Серія оптопари; 9. Серія світлодіодних індикаторів.

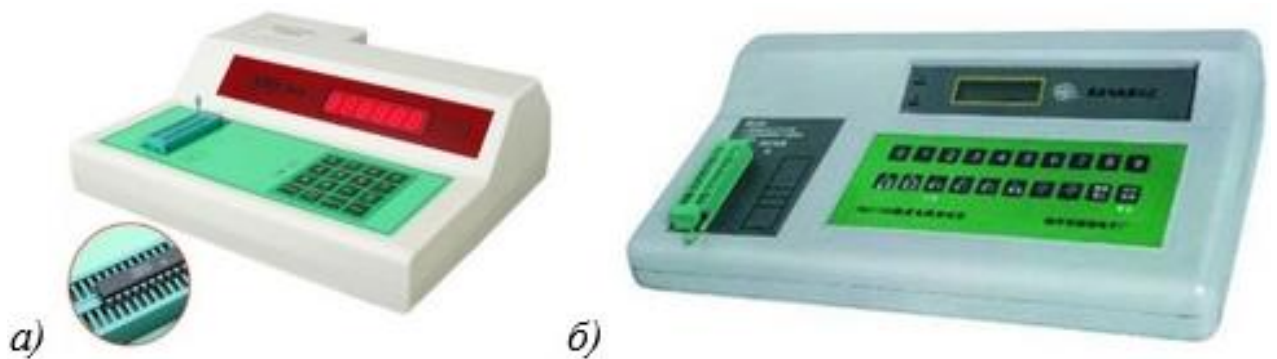


Рисунок 1.8 - Тестер мікросхем YBD868 а) та цифровий інтегральний тестер на базі AT89C51 б)

Цифровий інтегральний тестер на базі AT89C52. Тестер інтегральної схеми за допомогою контрольних і тестових ланцюгів, частина зв'язку з ПК. Контрольна і детективна частина основного завершення завдань виявлення мікросхем, результати, відправлені на головний комп'ютер через послідовний порт, хост-комп'ютер буде отримувати дані через підготовлену програму обробки

					КВРКІ 190246.19.22.48 ПЗ	Арк. 24
Зм.	Арк.	№ докум.	Підпис	Дата		

комунікацій Vb, результати одночіпового тесту за допомогою самокомпільованого програмного забезпечення можуть бути продемонстровані на комп'ютері [6].

Виберіть правильний номер послідовного порту, клацніть з'єднання, яке можна перевірити, якщо немає прямого підключення до серійного тесту, спливаючий програмний коментар такого змісту – «будь ласка, підключіть послідовний порт, а потім перевірте!». Якщо тестовий чіп є нормальним, програмне забезпечення виведе діалогове вікно «чіп не пошкоджений, він може бути використаний нормально». Чіп знаходиться в хорошому стані. Якщо тестовий чіп пошкоджений, програма виведе діалогове вікно «чіп пошкоджений, він не може бути використаний!». Тестер використовує джерело живлення 5 В. цифровий інтегральний тестер на базі AT89C51 і його принципова схема показані на рис. 1.8 б.

Контрольна і випробувальна частина схеми, показаної на рис. 1.9. Головний комп'ютер для надсилання даних на мікроконтролер AT89C51 [4], щоб визначити тип перевіряється мікросхеми, Мікроконтролер отримує дані та порівнюється з даним значенням для визначення типу перевіряється мікросхеми, контролює тестову схему для тестування.

Цільова тестова схема вводить дані в попередню настройку через одночіповий комп'ютер, потім зчитує дані за допомогою одночіпового комп'ютера і порівнюється зі звичайним результатом і виводить результат тесту і передає результат на верхній комп'ютер через мікросхему для перекладу рівня MAX232. Система для досягнення загальної цифрової функції інтегральної схеми, простої роботи, надійних результатів, низької вартості. Після фактичного тестування система може тестувати і оцінювати функцію в цілому 20 видів чіпів, крім того, тестер легко розширює цю функцію, тільки на основі існуючого оновлення програмного забезпечення для досягнення більшої кількості таких чіпів можна протестувати за допомогою тестера мікросхем-УВБ–868 можна випробувати такі серії мікросхеми: 1. Серія CMOS 14; 2. Серія CMOS40; 3. Серія CMOS 45; 4. Серія TTL 54; 5. Серія TTL55; 6. Серія TTL74; 7. Серія TTL 75; 8. Серія оптопар; 9. Серія світлодіодних індикаторів.

					КВРКІ 190246.19.22.48 ПЗ	Арк. 25
Зм.	Арк.	№ докум.	Підпис	Дата		

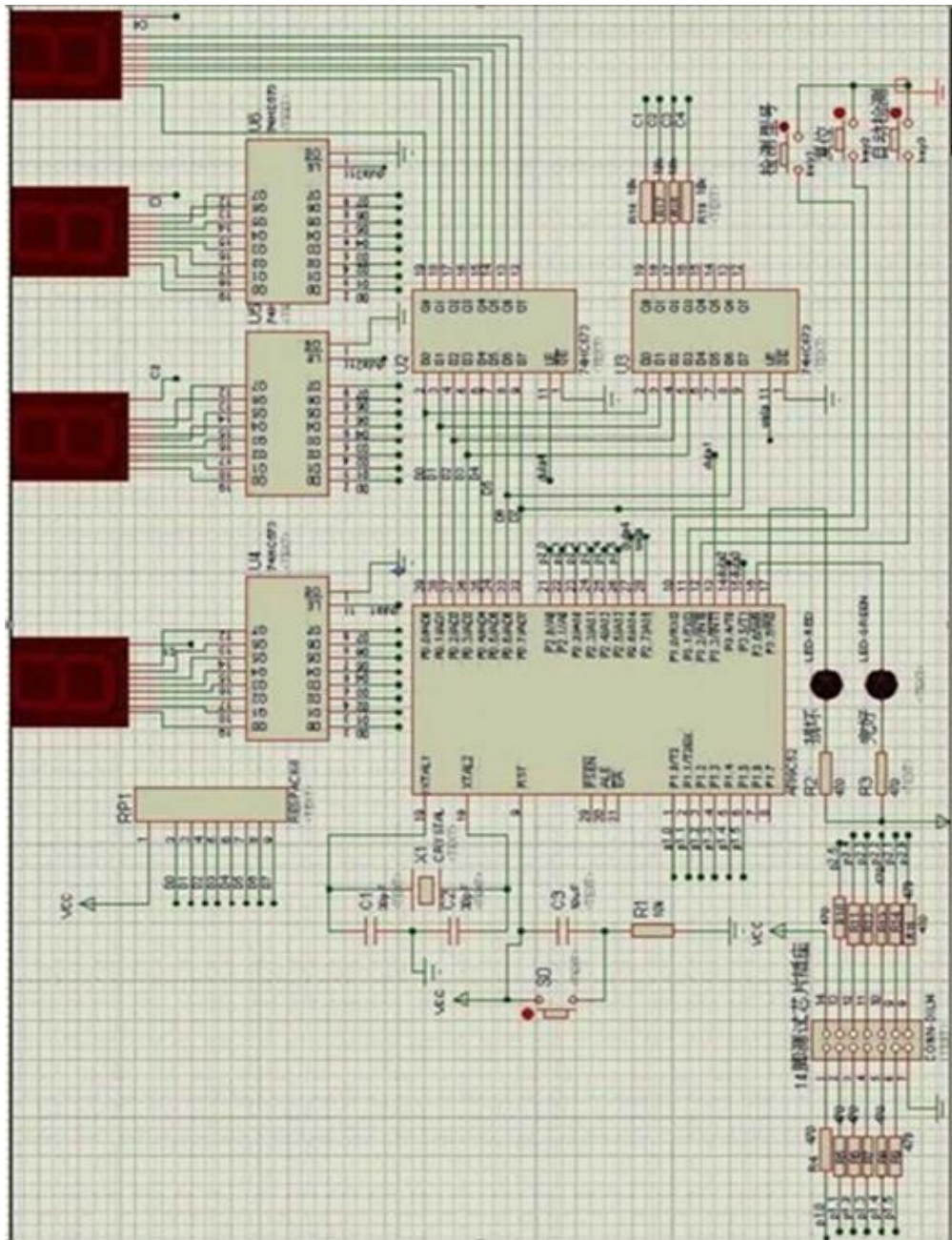


Рисунок 1.9 - Схема електрична принципова програмно-технічного комплексу розпізнання електронних компонентів

Тестер інтегральних схем SIMI-100. Тестер мікросхем – SIMI-100 широко використовується на практиці. За допомогою тестера мікросхем YBD868 можна випробувати такі серії мікросхеми: (1) 54 серії; (2) 74 серії; (3) Серія 4500; (4) серія 4000; ((5) Серія Ед 0.

Крім цього, можна тестувати вхідний струм, споживану потужність, вхідний струм витоку, струм навантаження пристрою; знайти Невідомий тип чіпа і

тестувати вихід пристрою: «три-стан» і «ОС» при використанні тестера мікросхем SIMI–100, який зображений на рисунку 1.10.



Рисунок 1.10 - Тестер інтегральних схем SIMI–100.

1.3. Висновок тестувань мікросхем

Проведено дослідження поточного рівня розвитку цифрових вимірювальних пристроїв, конкретніше тестерів мікросхем, з метою визначення їх стану.

В рамках цього аналізу було надано огляд найбільш поширених пристроїв для тестування мікросхем, зокрема тестера мікросхем YBD–868, цифрового інтегрального тестера на базі AT89C52 та тестера інтегральних схем SIMI100.

Крім того, були розглянуті теоретичні підходи до обґрунтування сутності процесу тестування цифрових вимірювальних пристроїв.

У результаті аналізу стану розвитку цифрових вимірювальних пристроїв, зокрема тестерів мікросхем, виявлено, що пристрої, такі як YBD–868, цифровий інтегральний тестер на базі AT89C52 та тестер інтегральних схем SIMI100, є одними з найбільш поширених і використовуваних у сучасній промисловості.

Тестер мікросхем YBD–868 заслуговує на увагу завдяки своїм передовим можливостям і високій точності вимірювань. Цифровий інтегральний тестер на базі AT89C52, у свою чергу, пропонує широкий спектр функціональності та підтримує різні типи мікросхем.

Тестер інтегральних схем SIMI100 є ще одним ефективним рішенням, забезпечуючи швидке тестування та вимірювання різних параметрів.

					КВРКІ 190246.19.22.48 ПЗ	Арк. 27
Зм.	Арк.	№ докум.	Підпис	Дата		

У контексті аналізу було розглянуті не лише конкретні пристрої, але й теоретичні підходи до обґрунтування сутності процесу тестування цифрових вимірювальних пристроїв.

Це включає в себе розгляд питань, пов'язаних з методами тестування, вимогами до точності вимірювань, забезпеченням надійності та ефективності процесу тестування.

					КВРКІ 190246.19.22.48 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

2 ТЕОРЕТИЧНИЙ ОПИС ПРОЕКТУ

2.1. Розробка структурної схеми пристрою контролю електронних компонентів

Для пристрою автоматичного тестування мікросхем розроблено програмне забезпечення (ПЗ), яке здійснює наступні дії:

- 1) встановлення зв'язку мікроконтролера з персональний комп'ютер;
- 2) здійснення вибору мікросхеми і запуск тесту для даної мікросхеми;
- 3) відображення результатів проведеного тесту;
- 4) зберігання даних про мікросхеми і алгоритмів тестування;
- 5) додавання нових мікросхем в базу даних з можливістю автоматичного складання тесту для даної мікросхеми.

ПЗ розроблялося не тільки для управління пристроєм, але також і для зберігання великої кількості даних необхідних для проведення тестів. Це дозволяє розширювати базу тестованих мікросхем і позбавляє від необхідності використання внутрішньої пам'яті пристрою [17].

Структурна схема приладу контролю мікросхем представлена на рисунку. 2.1.

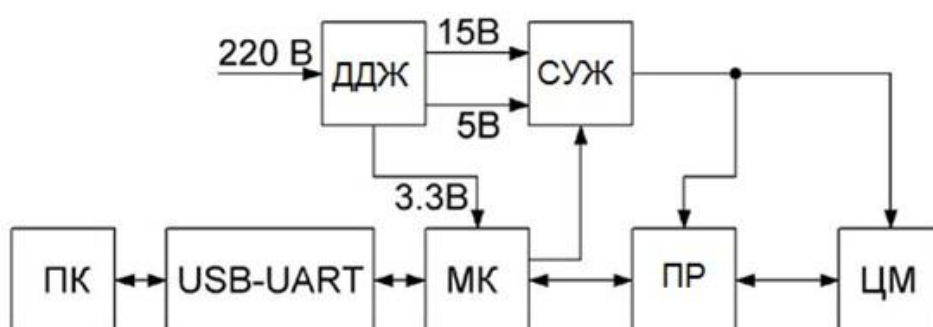


Рисунок 2.1- Схема структурна приладу контролю мікросхем

1. МК – мікроконтролер, є основною частин даної автоматичної керуючої системи.

2. ПК – персональний комп’ютер, призначений для відправлення команди в МК і відображення результати на екрані.
3. USB-UART-перетворювач USB-to-UART, призначений для з’єднання МК і ПК, передачі інформації між МК і ПК.
4. ПР – перетворювач рівнів, призначений для узгодження вхідного і вихідного сигналу по напругах і струмах при застосуванні в одному пристрої інтегральної мікросхеми з різного сімейства і з різними напругами живлення.
5. ЦМ – цифрова мікросхема, яка є контрольованим об’єктом.
6. ДДЖ – допоміжне джерело живлення, призначений для перетворення змінний напрямок 220В, 50 гц в постійне, яке може застосовуватися в цифровій мікросхемі, перетворювачі рівнів і мікроконтролері.
7. СУЖ – система управління живлення, призначена для управління підключення живлення до цифрової мікросхеми і перетворювача рівнів (рис 2.2).

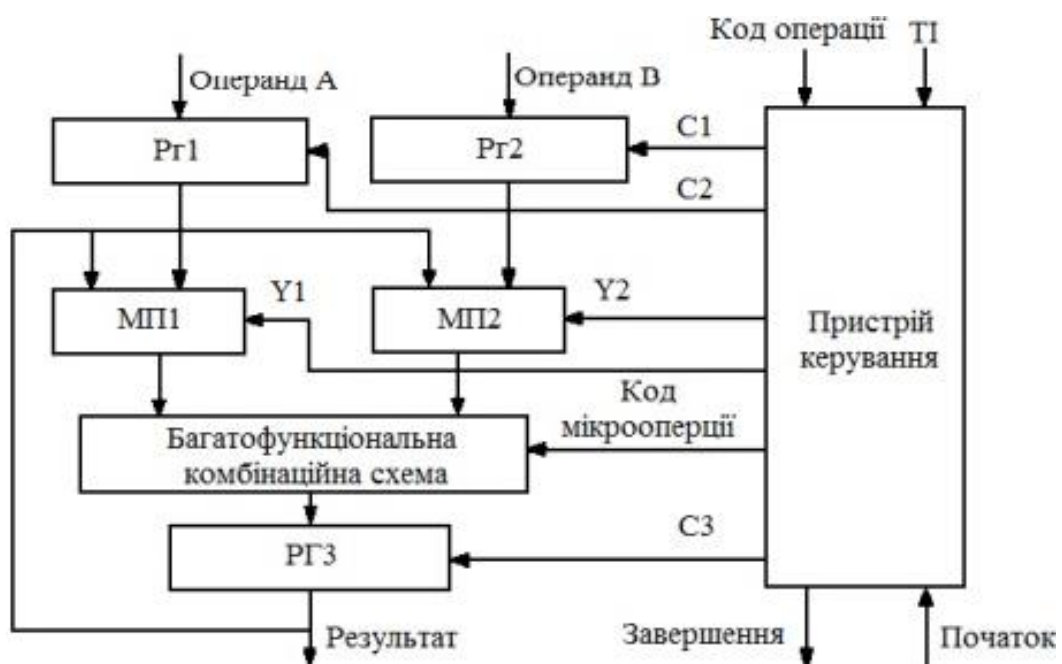


Рисунок 2.2. – Схема структури багатотактового операційного пристрою

Принцип роботи пристрою заключається в наступному. Спочатку на консоль персонального комп’ютера введені команди для підключення приладу. Через перетворювач USB-to-UART інформації передаються в мікроконтролер, і

мікроконтролер передається сигнал в перетворювач рівнів, і перетворювач рівнів може перетворювати рівень напруги сигналу з МК в рівень напруги, який для цифрової мікросхеми потрібен. Вихідний сигнал, який виходить з мікросхеми, через перетворювач рівнів подається в мікроконтролер. Після цього, мікроконтролер аналізується і зробить висновок по працездатності мікросхеми.

Результат відображений на екрані персонального комп'ютера. Коли мікроконтролер передається сигнал в перетворювача рівнів, одночасно управляє систему управління живлення, щоб підключили живлення 5В або 15В до перетворювача рівнів і цифровий мікросхеми (це залежить від типу цифрової мікросхеми). Коли вони не використовуються, живлення вимкнено за допомогою системи управління живлення [19].

USB – універсальний послідовний інтерфейс, що використовується для створення зв'язку між периферійними пристроями і персональним комп'ютером. Даний інтерфейс має широке застосування завдяки високій швидкості передачі даних і простоті підключення різних пристроїв. USB, зображений на рисунку. 2.3 а), бувають трьох версій:

- 1) USB 1.0 (максимальна швидкість передачі 1.5 Мбіт/сек);
- 2) USB 2.0 (Максимальна швидкість передачі 12 Мбіт/сек);
- 3) USB 3.0 (максимальна швидкість передачі 480 Мбіт/сек).

USB інтерфейс складається з чотирьох сигнальних проводів:

- 1) загальний провід-ground (gnd);
- 2) сигнальний провід «+ d»;
- 3) сигнальний провід «- d»;

Основні переваги USB інтерфейсу у складі пристрою контролю електронних компонентів такі:

- 1) наявність практично у всіх сучасних комп'ютерах і пристроях;
- 2) висока швидкість передачі даних;
- 3) простота підключення.

Недоліки USB інтерфейсу [4]:

- 1) максимальна довжина кабелю три метри;

Можливість роботи тільки з одним зовнішнім пристроєм на порт.

COM порт. Даний інтерфейс є послідовним портом з подвійною спрямованістю (рис 2.3 б)). Цей тип передачі даних використовується досить рідко внаслідок широкого застосування USB інтерфейс, тому в сучасних комп'ютерах COM порт найчастіше відсутня.

До позитивних якостей COM порту у складі пристрою контролю електронних компонентів відносяться:

- 1) простота підключення обладнання;
- 2) велика відстань для передачі даних (до двадцяти метрів).

Недоліки COM порту:

- 1) низька швидкість передачі даних [13].

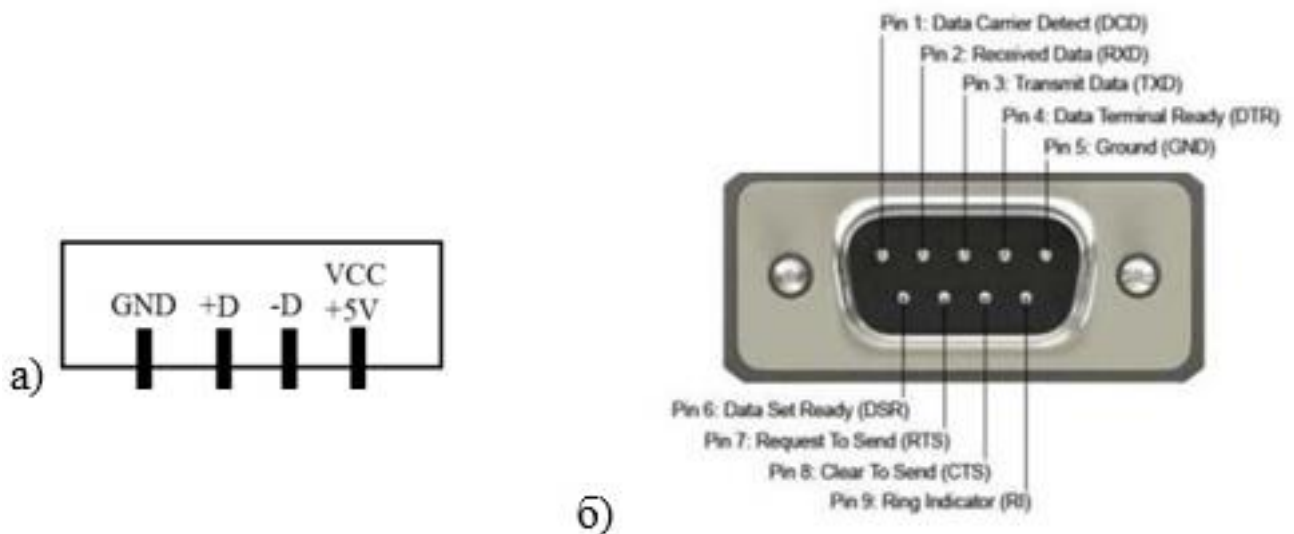


Рисунок 2.3- Схема USB інтерфейсу а) та COM порт б)

FireWire. Даний інтерфейс який зображений на рисунку. 2.4 є послідовною високошвидкісною шиною, призначеною для передачі даних між електронними пристроями і персональним комп'ютером. Порівняно з USB FireWire є більш

дорогим для реалізації, тому серед користувачів він не має широкого застосування, але, на відміну від USB, FireWire може передавати дані на відстані до п'яти метрів, і при цьому швидкість передачі даних досягає 800 Мбіт/с.



Рисунок 2.4- Інтерфейс FireWire

Перевагами FireWire у складі пристрою контролю електронних компонентів є таке:

- 1) можливість одночасного підключення до 17 пристроїв на порт;
- 2) довжина кабелю для передачі даних досягає п'яти метрів;
- 3) висока швидкість передачі даних серед послідовних інтерфейсів.

Недоліками FireWire у складі пристрою контролю електронних компонентів є:

- 1) висока вартість реалізації [16].

У своїй роботі ми вибрали COM порт за його простоту підключення.

UART – сукупність обчислювальних пристроїв, що використовуються для зв'язку з іншими цифровими пристроями. Така передача використовується в таких системах, де обмін даними відбувається час від часу і немає необхідності у високій швидкості передачі даних.

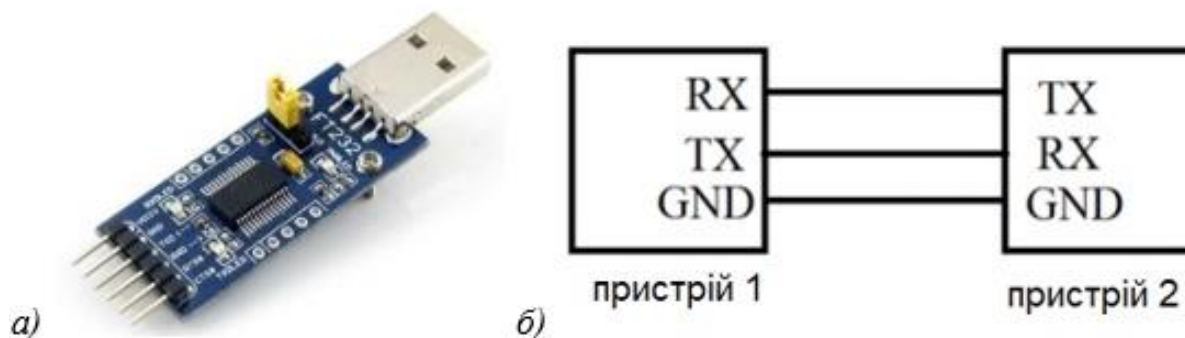


Рисунок 2.5- Плата UART а) та схема підключення UART б)

Принцип роботи:

1. Інтерфейс являє собою асинхронний послідовний протокол, в якому передавальна сторона по черзі видає в лінію код, а приймаюча відстежує його і заносить в пам'ять. Приймач і передавач заздалегідь встановлюють певну частоту, при якій будуть обмінюватися даними.
2. Підключення UART, яка показана на рисунку. 2.5а, відбувається за трьома лініями: RXD-прийом, txd – передача і GND-загальна лінія «земля» (рис. 2.5б).
3. При обміні даними можливі три методи передачі даних: – Односпрямована (ТУ, радіо); – Напівдуплексна передача, тобто почерговий прийом і передача даних; – Двонаправлена, при якому відбувається одночасний прийом і передача.

Перевагами UART у складі пристрою контролю електронних компонентів є таке:

- 1) проста і відпрацьована система;
- 2) мала вартість обладнання;

Недоліками UART у складі пристрою контролю електронних компонентів є:

- 1) частина пропускної здатності використовується для передачі службових бітів;
- 2) мала достовірність показань при численних помилках.

Проведемо розрахунки номіналів резисторів для структурної схеми пристрою контролю електронних компонентів [15]:

$$U_{\text{вих}} = 1.23 \cdot \left(1 + \frac{R_3}{R_4}\right) \rightarrow \frac{R_3}{R_4} = \frac{U_{\text{вих}}}{1.23\text{В}} - 1 \rightarrow R_3 = \left(\frac{U_{\text{вих}}}{1.23\text{В}} - 1\right) \cdot R_4 \quad (2.1)$$

Прийmemo $R_4 = 2 \text{ кОм} \rightarrow R_3 = \left(\frac{13\text{В}}{1.23\text{В}} - 1\right) \cdot 2 \text{ кОм} = 19.1 \text{ кОм}$

Оскільки у вимірюваному колі R_3 ; R_4 протікає струм $< 100 \text{ мкА}$, то використаємо метало- плівкові резистори MF 0.125 Вт.

$P_{R_3} = \Delta U \cdot I_{R_3} = (13 \text{ В} - 1.23 \text{ В}) \cdot 100 \text{ мкА} = 1 \text{ мВт}$ – що задовольняє вибраному типу;

$$P_{R_4} = 1.23 \text{ В} \cdot 100 \text{ мкА} = 0.1 \text{ мВт} \quad R_3 - \text{MF } 0.125 \text{ Вт}; 19.1 \text{ кОм}$$

$$R_4 - \text{MF } 0.125 \text{ Вт}; 2 \text{ кОм}$$

Для перевірки стійкості до навантаження по струму повинно виконуватись:

$$I_{\text{нав.мах}} = \frac{2.1 \text{ А} \cdot U_{\text{вх.мін}}}{U_{\text{вих}}} = \frac{2.1 \text{ А} \cdot 5 \text{ В}}{13 \text{ В}} = 0.8 \text{ А}$$

1,5 А – максимальний вхідний струм, наша схема споживає 0,8 А, що задовольняє нашим вимогам.

Розрахуємо параметри індуктивності L1 для пристрою контролю електронних компоненті:

Для цього розрахуємо робочий цикл D; який в межах ($0 \leq D \leq 0,9$).

$$D_{\text{мах}} = \frac{U_{\text{вих}} + U_f - U_{\text{вх(мін)}}}{U_{\text{вих}} + U_f - 0.6 \text{ В}} = \frac{13 \text{ В} + 0.5 \text{ В} - 5 \text{ В}}{13 \text{ В} + 0.5 \text{ В} - 0.6 \text{ В}} = \frac{3.5}{12.9} = 0.65$$

Де U_f – спад напруги на діоді VD1 у прямому напрямку. Для мінімізації витрат у перетворювачі використовуємо діод Шоткі, зі спадом напруги 0,5 В для пристрою контролю електронних компоненті.

Розрахуємо струм через індуктивність при певному навантаженні пристрою контролю електронних компоненті:

$$I_{\text{інд.}} = \frac{1.05 \cdot I_{\text{нав.мах}}}{1 - D_{\text{мах}}} = \frac{1.05 \cdot 0.8 \text{ А}}{1 - 0.65} = \frac{0.84}{0.35} = 2.4 \text{ А}$$

Індуктивність L1 для пристрою контролю електронних компоненті буде:

$$L_{\text{мін}} = \frac{6.4 \cdot (U_{\text{вх(мін)}} - 0.6 \text{ В}) \cdot (2 \cdot D_{\text{макс}} - 1)}{1 - D_{\text{макс}}} = \frac{6.4 \cdot (5 \text{ В} - 0.6 \text{ В}) \cdot (2 \cdot 0.65 - 1)}{1 - 0.65} = \frac{8.4}{0.35} = 24 \text{ мН}$$

З такими параметрами вибираємо індуктивність Renco, RL1954.

Елементи R2, R6 – утворюють компенсаційне коло:

Розрахуємо елементи:

$$R_2 = \frac{750 \cdot I_{\text{наб.макс}} \cdot U_{\text{вих}}^2}{U_{\text{вх.мін}}^2} = \frac{750 \cdot 0.8 \text{ А} \cdot 13^2 \text{ В}}{5^2 \text{ В}} = 4056 \text{ Ом} = 4 \text{ кОм}$$

R2 вибираємо типу MF 0,105 Вт, 4 кОм.

Ємність вихідного фільтра C7 визначається:

$$C_7 = \frac{0.19 \cdot L_i \cdot R_2 \cdot I_{\text{наб.}}}{U_{\text{вх.мін}} \cdot U_{\text{вих.}}} = \frac{0.19 \cdot 24 \text{ мН} \cdot 4 \text{ нОм} \cdot 0.8 \text{ А}}{5 \text{ В} \cdot 13 \text{ В}} = \frac{14.592}{65} = 2.24 \cdot 10^{-4} \text{ Ф} = 224 \text{ мФ}$$

Вибираємо тип C7, з урахуванням максимальної вихідної напруги, з 2-х кратним запасом: ESR 220 мФ × 25 В;

Компенсаційна ємність C6:

$$C_6 = \frac{58.5 \cdot U_{\text{вих}}^2 \cdot C_7}{R_2^2 \cdot U_{\text{вх(мін)}}} = \frac{58.5 \cdot 13^2 \text{ В} \cdot 220 \text{ мФ}}{4^2 \text{ нОм} \cdot 5 \text{ В}} = \frac{2175030}{8 \cdot 10^7} = 2.7 \cdot 10^{-8} \text{ Ф} = 27 \text{ нФ}$$

Вибираємо тип – керамічний конденсатор X7R 27 нФ × 50 В.

Розрахунок блокуючих ємностей C1, C5:

Виробник, згідно документації на мікросхему, рекомендує в безпосередній близькості до входу LM2577 установити керамічну ємність 0.1 мФ та електролітичну 47 мФ.

C1 – ESR 47 мФ × 16 В

C5 – X7R 0.1 мФ × 50 В

Діод Шоткі VD1 вибираємо виходячи з робочої частоти вихідного струму, та вихідної напруги для пристрою контролю електронних компоненті.

Вибираємо з двохкратним запасом: VD1 – IN5818.

Оптопара U1 використовується для комутації напруги програмування та гальванічної розв'язки між виходом та мікроконтролером. Для цього використовуємо оптопару PC817, яка дозволяє комутувати струми до 150 мА, та

напругу до 30 В. Обмеження струму світлодіоду оптопарі для пристрою контролю електронних компоненті здійснюється резистором R8 на значення 5мА:

$$R_8 = \frac{\Delta U}{I} = \frac{5\text{В} - 1.8\text{В}}{5\text{ мА}} = 640\text{ Ом}$$

Розсіювана потужність у складі пристрою контролю електронних компонентів буде:

$$R_8 - P_{R_8} = \frac{\Delta U^2}{R} = \frac{10.24\text{ В}}{640\text{ Ом}} = 16\text{ мВт}$$

Вибираємо тип R8 – MF 0.125 Вт; 640 Ом;

Аналогічно розраховуємо струмообмежуючі резистори R5, R9 для червоного світлодіода (індикатори режиму роботи програматора):

$$I_{\text{світлодіода}} = 7\text{ мА};$$

$$R_5 = \frac{5\text{ В} - 1.8\text{ В}}{7\text{ мА}} = 457\text{ Ом}; P_{R_5} = \frac{\Delta U^2}{R_5} = 22\text{ мВт} \rightarrow R_5 - MF0.125; 457\text{ Ом}$$

$$R_9 = \frac{13\text{ В} - 1.8\text{ В}}{7\text{ мА}} = 1.6\text{ кОм}; P_{R_9} = \frac{\Delta U^2}{R_9} = 78\text{ мВт} \rightarrow R_9 - MF0.125; 16\text{ кОм}$$

Резистор R1 – використовується для підтяжки виводу Reset. МК DD1: Вхідний струм 100 мА →

$$R_1 = \frac{5\text{ В}}{1000\text{ мА}} = 5\text{ нОм} \text{ тип } R1\text{ MF } 0.125\text{ Вт. } 5\text{ кОм.}$$

Конденсатори C3, C4 – обв'язка тактового кварцового генератора МК, для полегшення процесу запуску. Номінал їх використовується згідно рекомендацій виробника МК у межах 15–30 пФ.

Вибираємо C4, C3 – керамічні X7R 15 пФ × 50 В.

Резистори R6, R7 – струмообмежуючі, на випадок короткого замикання. у сигнальній лінії. Дублюють струмове обмеження МК, так як МК обмежує вихідний струм на 20 мА на кожен вихід.

$$R_6 = \frac{5\text{ В}}{20\text{ мА}} = 250\text{ Ом} \rightarrow P_{R_6} = \frac{\Delta U^2}{R_6} = \frac{5^2}{250\text{ Ом}} = \frac{25}{250\text{ Ом}} = 0.1\text{ Вт} \rightarrow$$

З запасом по потужності вибираємо R6, R7 – MF 0.125 Вт 250 Ом для пристрою контролю електронних компоненті.

Кінець тааблиці 2.1 - Зразки контролерів для пристрою розпізнавання типу електронних компонентів

5	TeleSafe, SmartWire, ScadaPack	«Control Microsystems», Канада
6	Simatic S7–200	«Siemens», Німеччина
7	Simatic S7–300C	«Siemens», Німеччина
8	Simatic C7–620	«Siemens», Німеччина
9	Versamax Nano, VersaMax	«General Electric Fanuc», США-Японія
10	Unitronics M90 Micro	«Industrial Automation Systems», Ізраїль
11	Unitronics Visiologik	«Industrial Automation Systems», Ізраїль
12	Modicon TSX Zelio	«Schneider Electric», Франція
13	Modicon TSX Twido	«Schneider Electric», Франція
14	FXIS, FXIN	«Mitsubishi Electric», Японія
15	Alpha	«Mitsubishi Electric», Японія
16	Pico	«Rockwell Automation», США
17	MicroLogix	«Rockwell Automation», США
18	MC8	«МЗТА», РФ

Ряд модифікацій модуля ЦПУ (FPO-C10RS, – C14RS, – C16CP, – C32CP і- T32CP) на бічній стороні забезпечені сом-портом, що представляє інтерфейс RS232 загального призначення для зв'язку з комп'ютером або віддаленими модулями. Також є роз'єм живлення модуля постійною напругою 24 В, перемикач RUN / PROG з індикаторами режимів.

Дані входів контролера:

- 1) гальванічна розв'язка: оптронна;
- 2) вхідна напруга (номінал): 24 VDC;
- 3) допустима Вхідна напруга: від 21,6 до 26,4 VDC;
- 4) вхідний струм: 4,3 мА при 24 VDC;

5) вихідний опір: 5,6 кОм.

Дані релейних виходів контролеру:

1) тип контакту: нормально відкритий, 2а / 250VAC, 2а / 30VDC.

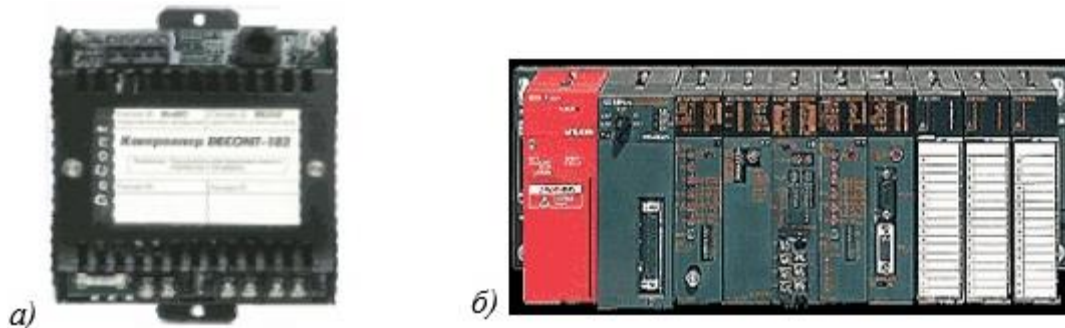


Рисунок 2.6- Зовнішній вигляд контролеру Decont – 182 а) та контролеру AnSH б)
Контролер Decont () – 182 («ДЕП», РФ).

Контролер Decont – 182 які зображені на рисунку. 2.8 для розпізнавання типу електронних компонентів становить основу апаратної частини програмно-технічного комплексу «Деконт». До складу комплексу також входять модулі введення-виведення, налагоджувальний пульт оператора (мініпульт), програмне забезпечення (DeSystem, WinDecont). Комплекс «Деконт» володіє підвищеною надійністю роботи в умовах електромагнітних перешкод і працює в широкому діапазоні температур від – 40 до +70 °С. Контролер Decont – 182 являє собою базовий процесорний модуль з операційною системою реального часу (ОС РВ) і забезпечує взаємодію з модулями вводу-виводу, формує алгоритми, архіви, підтримує зв'язок з іншими контролерами через два вбудованих інтерфейси (RS232 і RS–485) і два змінних інтерфейсу (мезоніну).

Основними характеристиками контролера Decont – 182 для пристрою розпізнавання типу електронних компонентів є:

1. Базовий процесор-Zilog 80182 (30 МГц).
2. Допоміжний процесор (співпроцесор) – PIC16C73A.
3. Ємність ПЗУ (на основі FLASH) – 512 кбайт.

4. Ємність енергонезалежного ОЗУ–512 кбайт.
5. Число портів послідовного зв'язку – 4 (порт «а» – RS232, порт «В» – RS–485, порти «С» і «D» – універсальні).
6. Годинник реального часу та сторожовий таймер.
7. Живлення: 24 В постійного струму (до V6.1 від 22 В до 26 В. з V7.1 від 9 до 30 В).
8. Споживаний струм (без інтерфейсних плат) – до 75 мА.
9. Встановлювані в контролері змінні інтерфейсні плати: Z-RS–485 (два гальванічно ізольованих канали RS–485 зі швидкістю передачі до 460.8 Кбод).
10. Z-RS232 (відповідає стандарту RS232 IBM PC).
11. Z-A'Т (два гальванічно ізольованих каналу «струмова петля» 20 мА зі швидкістю передачі до 38.4 Кбод).
12. Плати-модеми для зв'язку з лініями прийому-передачі даних (по телефонних, силових, двопровідних лініях і радіомодем).
13. Плата аналогового введення Z-AIN6–120, що встановлюється безпосередньо в гніздо «С» або «В» контролера. Плата має шість універсальних каналів введення-виведення, кожен з яких може працювати в одному з трьох режимів: введення сигналів постійного струму 0–20 мА або 4–20 мА (A1), дискретне введення (D1) і дискретний висновок (D0).

Модулі вводу-виводу для пристрою розпізнавання типу електронних компонентів призначені для підключення зовнішніх ланцюгів вимірювання, управління та сигналізації. Модулі, виконані в єдиному конструктиві, розраховані на безперервну роботу, мають єдине живлення і єдиний мережевий інтерфейс RS–485. Обмін даними (мережевими параметрами) здійснюється по мережі SYBUS. Основні параметри модуля зберігаються в EEPROM, за винятком таких параметрів, як «адреса модуля в локальній мережі «і» поточна швидкість обміну в мережі», які задаються програмно з міні-пульта. Модулі дискретного введення і виведення мають світлодіодні індикатори стану каналів. Розміри модуля – 106 x 120 (142-приєднувальний розмір) x50 мм.

					КВРКІ 190246.19.22.48 ПЗ	Арк. 41
Зм.	Арк.	№ докум.	Підпис	Дата		

Контролер PS-416 («Moeller», Німеччина). Програмовані контролери фірми Moeller працюють в промислових мережах Profibus DP, CANopen, DeviceNet, Interbus і ін. контролер PS-416 використовується для вирішення складних завдань управління процесами. Контролер PS-416 складається з крейта (кошика) BGT-400, блоку живлення, процесорного модуля CPU-400, мережевих карт NET-400 і карт входів-виходів INP-400 і OUT-400. Крейт контролера має кілька модифікацій в залежності від кількості посадочних місць (9, 13 або 19).

Процесорний модуль CPU-400 має обсяг основної пам'яті 1 мбайт, додаткової пам'яті (Flash-карта) – 512 кбайт, 1, 2 або 4 Мбайта. Процесор – 10-розрядний МП з RISC-архітектурою. Час обробки базової команди – 0,5 мкс. Інтерфейс для програмування – RS232 / RS-485. Кількість підключаємих станцій – 30. Для індикації стану модуля служать 5 світлодіодів: Run (робота), Ready (готовий), Not Ready (не готовий), Error (помилка), Change (Заміна батареї). Керуючі програми можуть зберігатися в ОЗУ або в EEPROM (в енергонезалежній карті Flash Memory Card).

Мережева карта NET-400 по інтерфейсу RS-485 забезпечує швидкість передачі 187,5 Кбіт/с на відстань 600 м. Кількість підключаємих станцій – 30.

Контролери серії КОНТРАСТ (ЗАТ «ВОЛМАГ», РФ). Контролери серії КОНТРАСТ побудовані на базі конструктивів і модулів УСО контролера Р-130, але володіють більшою надійністю, розширеними функціональними можливостями, високою швидкістю обробки інформації і передачі її по зовнішніх інтерфейсах, можливістю збільшення числа входів-виходів за рахунок підключення блоків розширення УСО. До складу серії КОНТРАСТ входять три типи контролерів: РК-131 / 300, КР-300-310. Контролери побудовані за єдиною ідеологією і розрізняються характеристиками (числом входів – виходів, об'ємом пам'яті, продуктивністю). Модель РК-131 / 300 відноситься до класу міні – контролерів і призначена для автоматизації невеликих об'єктів. У моделі використовуються модулі УСО контролера Р-130.

Контролер має до 48 гальванічно-ізольованих аналогових входів, до 12 аналогових виходів, до 96 дискретних входів-виходів.

					КВРКІ 190246.19.22.48 ПЗ	Арк. 42
Зм.	Арк.	№ докум.	Підпис	Дата		

Модель КР–300 для пристрою розпізнавання типу електронних компонентів відноситься до контролерів середнього класу з великою кількістю входів-виходів. Крім модулів УСО Р–130 до складу контролера входить вбудований пульт оператора. З урахуванням модулів УСО контролер має до 190 гальванічно-ізольованих аналогових входів-виходів, до 304 дискретних входів-виходів.

Контролер серії ЕК–2000 (ЗАТ «Емікон») для пристрою розпізнавання типу електронних компонентів. Контролери серії ЕК–2000 складаються з обчислювального і кросового блоків, з'єднаних між собою кабелями. До складу обчислювального блоку входять модуль живлення і процесорний модуль, що встановлюються в каркас разом з модулями УСО. До складу кросового блоку входять панелі монтажні і блок. Джерела живлення (2 канали 24 VDC, 2 А). Контролери мають безкорпусне, однокорпусне або двокорпусне виконання. В безкорпусному виконанні контролер розміщується в шафах Користувача, а при одно- або двокорпусному виконаннях контролер розміщується в захисних кожухах зі ступенем захисту IP54.

До складу контролера входить панель оператора, яка може встановлюватися в захисному кожусі або на двері електрошкафа. На панелі оператора розташовуються дисплей і технологічна клавіатура. Панель оператора підключається до контролера по послідовному інтерфейсу RS–232 або RS–485. Ступінь захисту панелі оператора IP65.

Контролери підтримують мережеву зіркоподібну або магістральну архітектуру. Фізичний рівень локальної мережі – RS–485, каналний рівень - Modbus або SDLC. Максимальна кількість вузлів в мережі – 256. Швидкість передачі від 19,2 Кбод до 2,5 Мбод (при використанні мережевих модулів С–02А і С–05а).

До складу модулів УСО входить більше 36 модулів, в тому числі процесорні модулі Срі-ОЗА і Срі – ОЗВ (процесор 20 МГц, ПЗУ–32 кбайт, Flash – 512 кбайт, ОЗУ–128 кбайт), модулі введення-виведення аналогових і дискретних сигналів, перетворювачі інтерфейсів, модулі лічильників, мережеві модулі, блоки живлення.

					КВРКІ 190246.19.22.48 ПЗ	Арк. 43
Зм.	Арк.	№ докум.	Підпис	Дата		

Всі модулі мають гальванічну розв'язку між вхідними і вихідними ланцюгами і системним живленням.

Програмування контролерів здійснюється за допомогою спеціальної мови СОКТ. Програми, написані на мові СОКТ, працюють в середовищі багатозадачної операційної системи реального часу ОС-188, що поставляється разом з контролером (рис. 2.7).

Особливістю мови є текстове написання команд і символічне представлення змінних. В якості змінних в мові СКТ використовуються розряди дискретного введення-виведення, прапори, регістри, лічильники і таймери. Мова СОКТ є відкритою, тобто у вихідний текст програми можна включати підпрограми, написані на асемблері процесора IBM PC.

Для написання і налагодження програм використовується інтегроване середовище «CONT-Designer», що працює в операційному середовищі MS-DOS на IBM-сумісному комп'ютері і включає в себе редактор вихідних текстів, транслятор і відладчик. При налагодженні програм контролер підключається до персонального комп'ютера по інтерфейсу RS232.

Контролери MELSEC AnSH, QnAS («Mitsubishi Electric», Японія). Модульна серія AnSH для пристрою розпізнавання типу електронних компонентів надає можливість оптимальної конфігурації системи для широкого спектру самих різних завдань. Існує вибір з 7 процесорних модулів, диференційованих за рівнями, і більш ніж 60 модулів різних додатків. Контролер AnSH займає дуже мало місця: наприклад, до 160 сигналів введення / виводу можна розмістити на площі в 32,5 · 13 см, гарантуючи цілком рентабельні рішення автоматизації.

Незважаючи на свої невеликі розміри, компактний AnSH володіє потужністю набагато більш габаритної контролерної системи: пам'ять на 30 до програмних кроків, час виконання логічної інструкції 0,25 мкс і 10–30 мкс для арифметичної інструкції і обробки слова. За допомогою контролера AnSH можна управляти до 96 осями крокових двигунів.

					КВРКІ 190246.19.22.48 ПЗ	Арк. 44
Зм.	Арк.	№ докум.	Підпис	Дата		

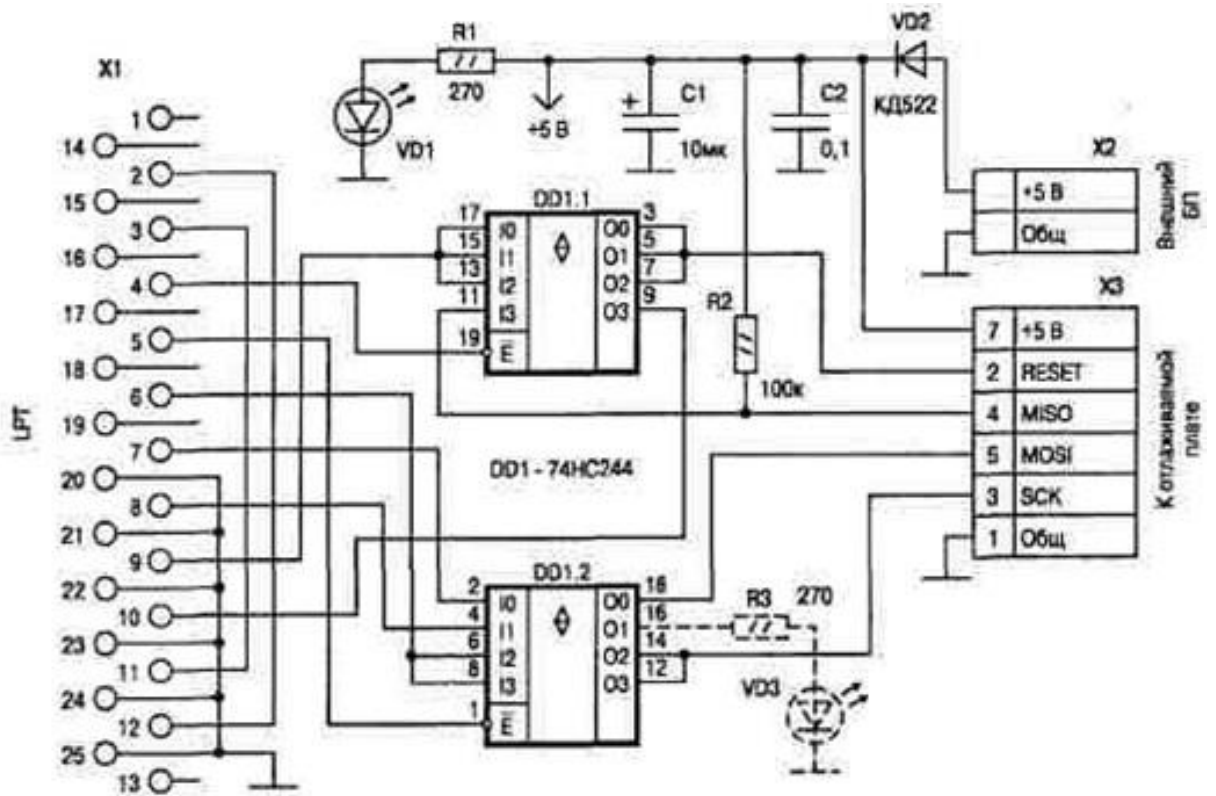


Рисунок 2.7- Схема програматора з підключенням до паралельного порту.

Контролер для пристрою розпізнавання типу електронних компонентів підтримує Profibus і Ethernet, а також власні варіанти шин фірми MITSUBISHI-MELSECNET і MELSECNET / 10. Серія MELSEC AnSH має прикладні модулі:

- 1) модулі введення / виведення дискретних сигналів;
- 2) модулі введення / виведення аналогових сигналів;
- 3) модулі спеціалізованих функцій для обробки переривань, введення температури, регулювання температури, позиціонування, а також модулі послідовних і мережевих інтерфейсів.

Контролер ломіконт ТМ (ВАТ «Електроприлад»).

Контролер ломіконт з вбудованою системою Trace Mode (ломіконт ТМ) в якості апаратної частини базового модуля містить одноплатний PC-сумісний комп'ютер на основі процесора i486, а також Flash-диск, порт RS-485, два порти RS232, порт Ethernet і слот розширення формату PC-104.

Контролер ломіконт з вбудованою системою Trace Mode (ломіконт ТМ) в якості апаратної частини базового модуля містить одноплатний PC-сумісний

комп'ютер на основі процесора i486, а також Flash-диск, порт RS-485, два порти RS232, порт Ethernet і слот розширення формату PC-104.

Новий базовий модуль забезпечує спадкоємність базових комплектів на контролерах «Ломіконт – 110» за умови перезапису програми Користувача з використанням засобів Trace Mode.

Для підготовки програм користувача в Trace Mode передбачені 2 мови технологічного програмування

«ТехноFBD» – мова функціональних блоків, що повторює техніку програмування контролерів «Реміконт – 110».

Мова заснована на з'єднанні в ланцюжки і деревовидні структури бібліотечних функцій і алгоритмів.

«ТехноFBD» – мова послідовних інструкцій. Дані мови є розширеннями мов FBD і IL, що входять в інструментальну систему технологічного програмування ISaGRAF за стандартом IEC 61131-3.

Система Trace Mode, крім основної функції відображення на екрані ходу технологічного процесу і трендів, дозволяє вирішувати наступні завдання:

1) об'єднання технологічних процесів, керованих декількома контролерами ломіконт ТМ, на одному екрані в одну мнемосхему (використовуючи технологію OPC);

2) складання та оформлення звітів;

3) архівування даних (3 типи архівів);

4) зв'язок з різними додатками Windows;

Захист інформації від несанкціонованого доступу (паролі).

В якості базових процесорів в процесорних модулі контролера для пристрою розпізнавання типу електронних компонентів широко застосовуються процесори фірм Intel (I80C186EC, i80188, i80386EX*X), i80486DX4, i80586, Pentium MMX), Motorola (MC68302, MC68360), Zilog (80182), AMD, NEC, Hitachi, Atmel.

Для збору і обробки даних використовуються допоміжні процесори (співпроцесори) типів i80C188 (8 МГц), i80051 (31), P1C16C73A, а також RISC-вбудовані співпроцесори.

					КвРКІ 190246.19.22.48 ПЗ	Арк. 46
Зм.	Арк.	№ докум.	Підпис	Дата		

Комунікаційні модулі для пристроїв розпізнавання типу електронних компонентів призначені для обміну даними між контролером і зовнішніми пристроями по обчислювальних мереж передачі даних.

Останнім часом набувають поширення універсальні процесорні модулі, виконані за мезонінною технологією.

В якості зовнішніх пристроїв виступають інтелектуальні термінали, промислові комп'ютери, контролери, модулі віддаленого введення-виведення.

Серед комунікаційних модулів знаходять широке застосування комунікаційні модулі локальних мереж GAN, Device Net, Profibus-DP, Ethernet, мережі GSM та ін., комунікаційні модулі інтерфейсів RS-232, RS-422 / 485.

Характеристиками комунікаційних модулів для пристрою розпізнавання типу електронних компонентів є допустима швидкість обміну (біт/с), число обслуговуваних пристроїв обміну даними, кількість портів введення-виведення, напруга гальванічної ізоляції, напруга живлення, діапазон робочих температур.

З метою недопущення зниження обчислювальних ресурсів базового процесору, що відповідає за введення-виведення, обробку інформації та роботу системи в цілому, доцільно використовувати автономний процесор для реалізації процедури послідовного обміну даними по мережі, так званий мережевий комунікаційний контролер.

В якості комунікаційних контролерів при автоматизованого розпізнавання типу електронних компонентів технології використовуються контролери фірми Motorola MC68302, MC68360, MPC8xx / 82xx. До складу комунікаційного контролера фірми Motorola входять 3 типових модулів: модуль центрального процесора CPU (Central Processor Unit), модуль системної інтеграції SIM (System Integration Module), модуль комунікаційного процесора (CPM-Communication Processor Module). Структура комутаційного контролера наведена на рис. 2.8.

Серія комунікаційних контролерів MC68302, куди входять контролери MC68LC302 / 302v, MC68EШ02, використовують 16-розрядний процесор 68E000 і володіють малим енергоспоживанням і невисокою вартістю, використовуються в телекомунікаційній апаратурі різного призначення.

					КВРКІ 190246.19.22.48 ПЗ	Арк. 47
Зм.	Арк.	№ докум.	Підпис	Дата		

Контролери серії MC68360 є подальшим розвитком серії MC68302 і володіють більш високою продуктивністю і розширеними функціональними можливостями. Контролер має 32-розрядну шину даних і 32-розрядну шину адреси.

До складу комунікаційного контролера серії MC68360 входять 32-розрядний процесор CR12+ (33 МГц). Модуль системної інтеграції 51M60 реалізує інтерфейс з системною шиною.

До складу модуля входять генератор тактових імпульсів, контролер динамічного ОЗП, блок тестування і налагодження та ін., а також сторожовий таймер (watch-dog timer) і таймер переривань. До складу модуля СРМ входять RISC-процесор, 2-портове ОЗУ і великий набір периферійних пристроїв.

До складу периферійних пристроїв входять 4 зв'язкових послідовних контролера (SCC), 2 керуючих послідовних контролера (SMC), послідовний інтерфейс SP1, паралельний порт Centronics, 3 паралельних порти, що забезпечують протоколи квитованого і неквитованого обміну, чотири 16-розрядних таймера.

Зв'язкові контролери SCC моделі MC68EN360 для пристрою розпізнавання типу електронних компонентів реалізують протокол обміну по локальних мережах Ethernet зі швидкістю обміну до 10 Мбіт/с, а також протокол Centronics. У комунікаційних контролерах MPC8xx / 82xx в якості процесора використовується 32-розрядний RISC-процесор Power PC з вбудованими блоками кеш-пам'яті команд і даних.

Комунікаційний процесор СРМ і блок SIM цих контролерів забезпечують підвищену швидкість обміну даними (до 100 Мбіт/с по протоколу Fast Ethernet) і більш високу продуктивність. Тактова частота процесора – 80 МГц.

Серія MPC82xx для пристрою розпізнавання типу електронних компонентів є наступним поколінням комунікаційних контролерів на базі RISC-процесора Power PC, тактова частота якого підвищена до 200 МГц, збільшено число послідовних каналів в блоці СРМ, введений контролер шини РС.

					КВРКІ 190246.19.22.48 ПЗ	Арк. 48
Зм.	Арк.	№ докум.	Підпис	Дата		

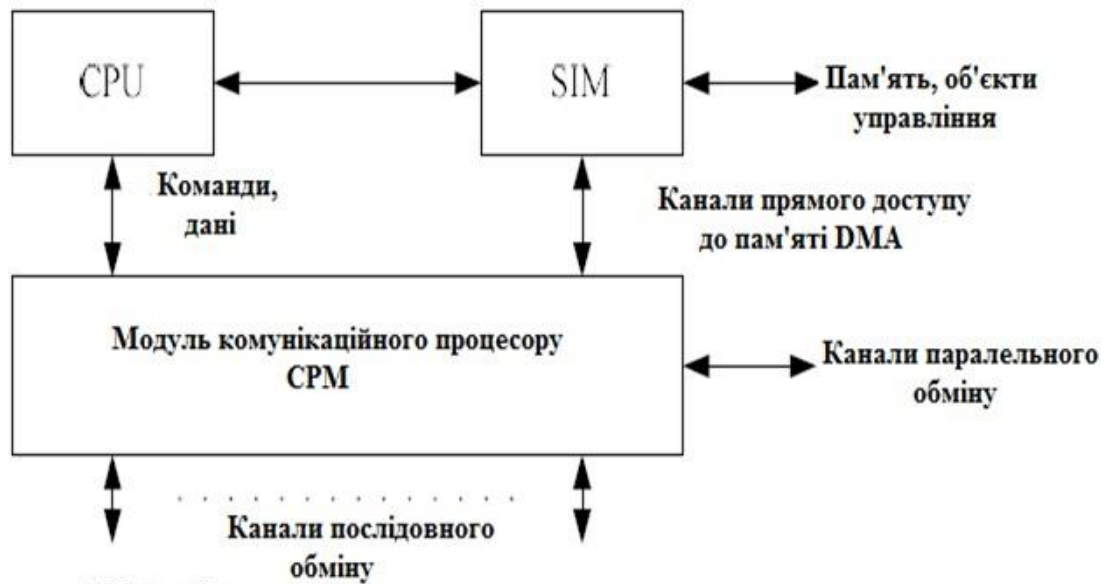


Рисунок 2.8- Схема функціональна комутаційного контролера для пристрою розпізнавання типу електронних компонентів

Комунікаційні контролери MC68EШ60 / ЕШ60У мають інтерфейси RS232, RS-485, Ethernet, Flash-пам'ять до 4 Мбайт, Sram. Комунікаційні процесори дозволяють організувати обмін даними по мережі між програмованими контролерами, станціями введення – виведення, робочими станціями та іншими інтелектуальними пристроями (рис. 2.9) [6].

Зв'язок здійснюється за способом «точка-точка» (PPI-Point-to-Point Interface) для під'єднання одного пристрою до іншого або зв'язок по мережі, що забезпечує обмін даними між абонентами мережі.

Корпус комунікаційного модуля виконаний з пластмаси або металу і кріпиться до стійки за допомогою гвинтових роз'ємів або спеціальної системи з'єднання, як правило, аналогічної системі кріплення інших модулів на стійку контролера. Модуль також має роз'єм для підключення на внутрішню шину контролера.

Комунікаційний контролер при автоматизації розпізнавання типу електронних компонентів виконує всі необхідні операції по формуванню переривань, обміну даними по мережі і по внутрішній шині контролера.

Обсягу пам'яті окремого модуля достатньо для зберігання декількох Web-сторінок підвищеної складності.

Таким чином, багато функцій стають доступними через Internet, в тому числі детальна діагностика системи управління, Перегляд даних реального часу, модифікація даних програми, відображення даних ПЛК у вигляді мнемосхем, графіків, таблиць на Web-сторінках.

Конфігурування та програмування комунікацій контролера здійснюється за допомогою спеціального програмного забезпечення, зазвичай поставляється фірмою-виробником разом з контролером.

В табл. 2.2 представлені дані основних комунікаційних модулів різних фірм.

Таблиця 2.2 - Дані комунікаційних модулів для пристрою розпізнавання типу електронних компонентів

Тип модуля (ПЛК)	Виробник	Інтерфейс	Швидкість обміну	«Шина»
140 NOE 771 00 (TSX Quantim)	Schneider Electric	1x Ethernet 10 / 100BaseT, 1x100BaseFX	10 / 100 Мбіт/с	
140 CRP 811 00 (TSX Quantim)	Schneider Electric	1xProfibus-DP, 1xRS232C, 1xRS-485	12 Мбіт/с; 19,2 Кбіт/с	
CP 343-5 (Simatic S7-300)	Siemens	1 x Profibus-DP	12 Мбіт/с	
1756-DNB (ControlLogix)	Allen- Bradley	1 x DeviceNet	125, 250 або 500 Кбіт/с	
IC697CMM742 (Series 90-70)	GE Fanuc	1x Ethernet 10 / 100BaseT	10 / 100 Мбіт/с	

Кінець таблиці 2.2 - Дані комунікаційних модулів для пристрою розпізнавання типу електронних компонентів

ADAM-5090	Advantech	4xRS-232	50 біт/с... 115,2 Кбіт/с	
ADAM-5090	Advantech	3xRS-232, 1xRS-485, 1x Ethernet 10/	115,2 Кбіт/с, 10 / 100 Мбіт/с	
ADAM 8208-1DP01	Advantech	Profibus-DP Master	9600 біт/с. 12 Мбіт/с	-
ADAM 8253-1DP00	Advantech	Profibus-DP Slave	9600 біт/с. 12 Мбіт/с	
ADAM-4581	Advantech	1xRS-232 / RS-485, GSM-модем	115,2 Кбіт/с	
750-815	WAGO	Modbus RS-485	115,2 Кбіт/с	-
FP2-MW	Matsushita	MEWNET-W	500 Кбіт/с	
PCM-3601	Advantech	Факс-модем	56 Кбіт/с (V. 90)	PC / 104
PA 7200	ADDI-DATA	1 x CAN 1.2, 1 x RS232 / RS-485	75 бітА; 115,2 Кбіт/с	ISA
EMERALD-MM / 8	Diamond Systems	4 / 8 x RS232 / RS-485	115,2 Кбіт/с	PC / 104

До спеціальних (функціональних) модулів при автоматизованому розпізнаванні типу електронних компонентів відносяться модулі позиціонування (рис. 2.10), лічильники, модулі зважування, модулі годин реального часу і пам'яті, обчислювальні, логічні та діагностичні модулі, модулі енкодера, технічного зору, одно- і двоканальні модулі бар'єрів іскробезпеки, модулі мезонінних плат, модулі живлення.

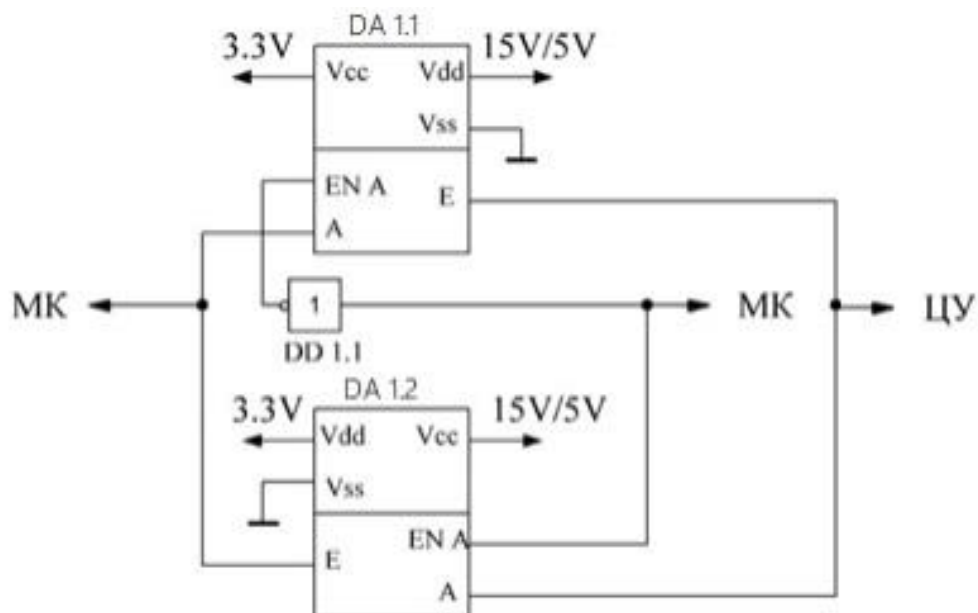


Рисунок 2.10- Схема підключення модулів програмно-технічного комплексу розпізнання електронних компонентів

2.3. Висновок розробки схем

Було проведено обґрунтування вибору необхідного обладнання для розробки дистанційно керованого модуля тестування мікросхем пам'яті. У цьому контексті, було вирішено використати програматор JTAG ICE XP II з метою забезпечення ефективного та надійного тестування мікросхем пам'яті.

Під час обґрунтування вибору обладнання були розглянуті різні фактори, включаючи технічні характеристики програматора, його сумісність з мікросхемами пам'яті, наявність необхідних інтерфейсів та здатність до дистанційного керування. JTAG ICE XP II відповідає всім цим вимогам, що робить його ідеальним вибором для розробки дистанційно керованого модуля тестування мікросхем пам'яті.

Крім обладнання, було розроблено та реалізовано програмну реалізацію для забезпечення розробки дистанційно керованого модуля тестування. Це означає, що були створені програмні компоненти, які співпрацюють з програматором JTAG ICE XP II для здійснення тестування мікросхем пам'яті. Ця програмна реалізація включає в себе функції для керування програматором, зчитування та запису даних

в мікросхеми пам'яті, аналізу результатів тестування та забезпечення зручного і ефективного інтерфейсу користувача.

Розроблена програмна реалізація дозволяє розширити можливості програматора JTAG ICE XP II та забезпечує зручний і гнучкий підхід до розробки дистанційно керованого модуля тестування мікросхем пам'яті. Цей модуль може бути використаний для ефективного та автоматизованого тестування мікросхем пам'яті, що сприяє підвищенню якості та продуктивності розробки в цій галузі.

					КВРКІ 190246.19.22.48 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

3 ПРИНЦИПОВА СХЕМА ПРИСТРОЮ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1. Допоміжне джерело живлення, перетворювач рівнів

Структурна схема допоміжного джерела живлення представлена на рис. 3.1.

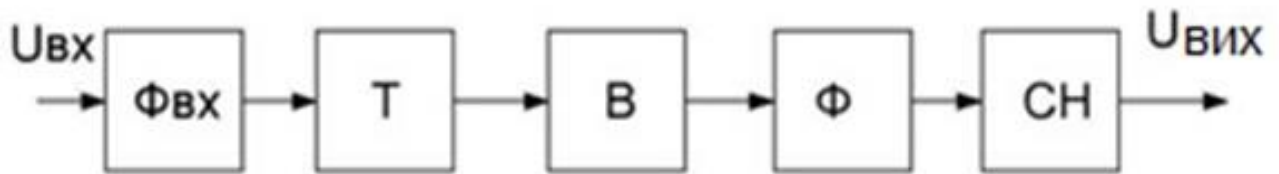


Рисунок 3.1- Схема структурна джерела живлення

1. Фвх – вхідний фільтр, призначений для захисту ланцюга і фільтрації вхідного сигналу. (тобто пульсація струму, виключення перешкод із землі та виключення перешкод між провідниками)

2. Т – трансформатор, призначений для зниження напруги.

3. В – випрямляч, призначений для перетворення негативної частини синусоїдальної напруги в позитивну.

4. Ф – фільтр, зазвичай буде стояти згладжує фільтр, конденсатор, який потрібен для зниження пульсацій нижніх частот сигналу. Цей фільтр складається з конденсатора і резисторів, завдяки яким він здатний запасати енергію при збільшенні вхідної напруги і віддавати її при її зменшенні.

5. СН – Стабілізатор напруги, призначений для стабілізації напруги, тобто отримання постійної напруги.

Для того, щоб перетворювати змінну напругу 220В, 50кГц в постійне, яке може застосовуватися в цифровий мікросхемі і перетворювачі рівнів, став допоміжний джерело живлення, який зображений на рисунку. 3.2, у виході якої можна отримати постійну напругу 5В і 15В,

Перелік основних елементів показаний у таблиці. 3.1

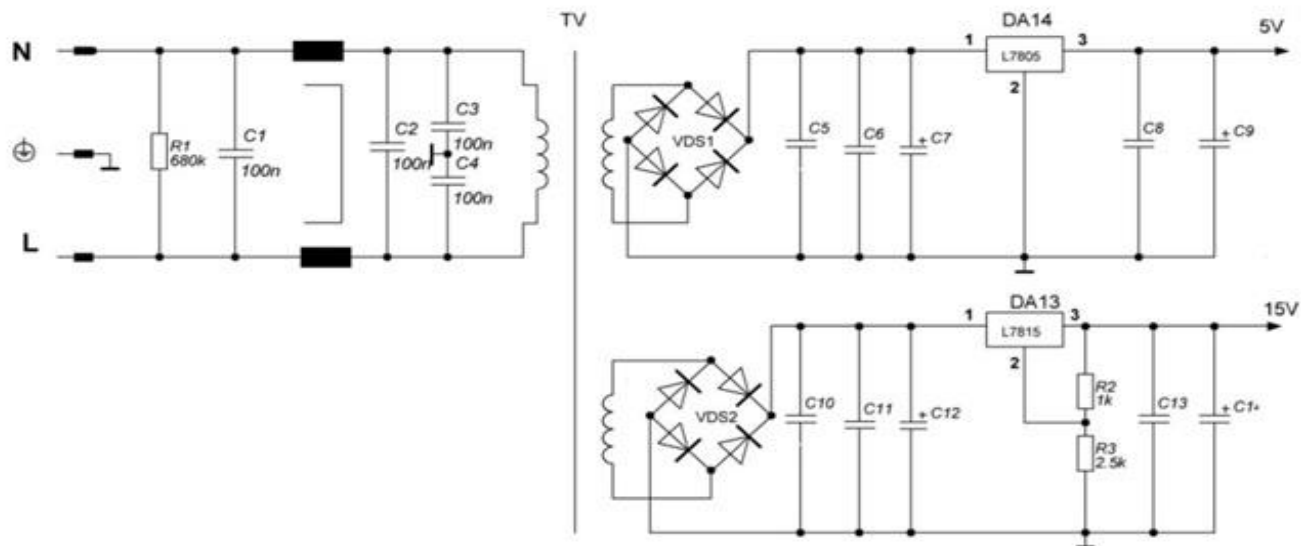


Рисунок 3.2- Схема джерела живлення

Таблиця 3.1 - Перелік основних елементів

Елемент	Кількість	Примітка
ЕВ1558	2	Діодний міст
Трансформатор ТПК-2Х18	1	Трансформатор
L7805ABD2T -TR	1	Стабілізатор напруги
L7815ABD2T-ТК	1	Стабілізатор напруги
В84111-А-А30	1	Фільтр

Перетворювач рівнів. До мікроконтролера підключено живлення 3.3 В, але до цифрової мікросхеми підключено живлення 5В або 15В. для того щоб перетворити 3.3 В В 5В / 15В або навпаки, став перетворювач рівнів на основі мікросхеми HCF40109В.

За структурною схемою HCF40109В видно що, у мікросхеми HCF40109В мають 4 входи інформації: А, В, С, D; 4 виходи: Е, F, Н, G; 4 входи: EN А, EN В, EN С, EN D.

Високий рівень напруги вхідного сигналу пов'язаний з живленням V_{cc} , а низький рівень напруги вхідного сигналу пов'язаний з живленням V_{ss} .

Високий рівень напруги вихідного сигналу пов'язаний з живленням V_{dd} , а низький рівень напруги вихідного сигналу пов'язаний з живленням V_{ss} . V_{ss} завжди підключено до землю.

Щоб перетворити 3.3 В у 5В / 5 В або навпаки, схема представлена на рис 3.3.

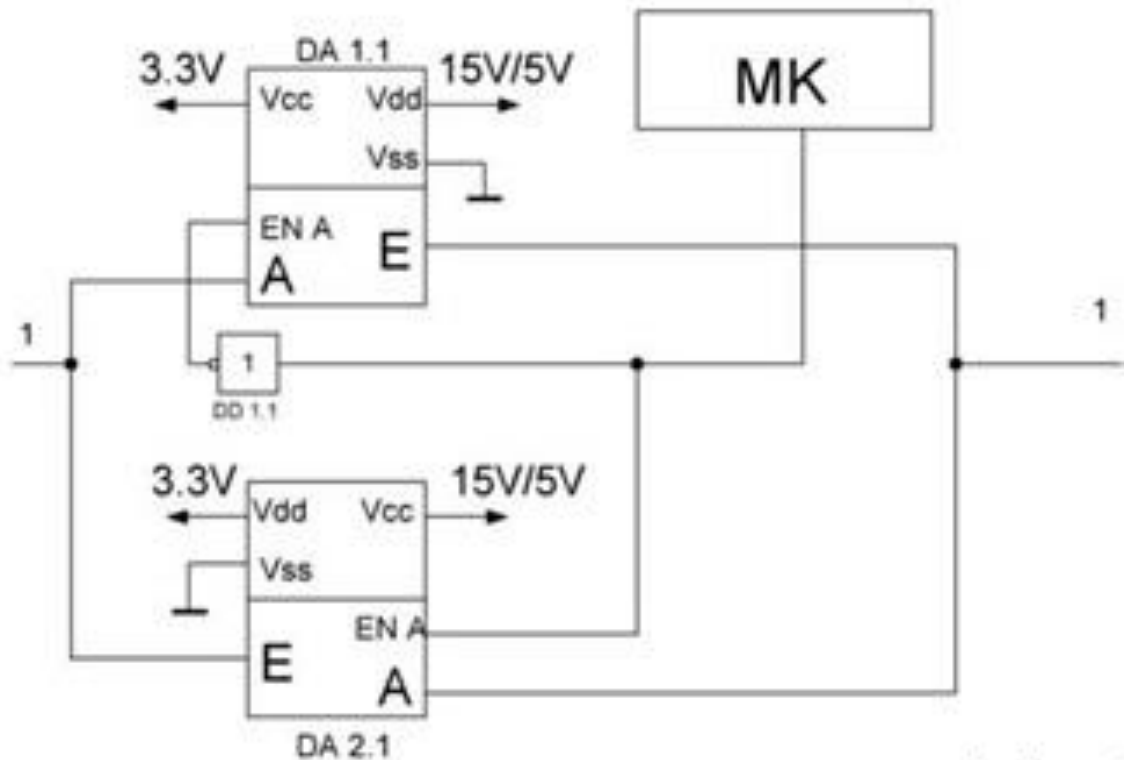


Рисунок 3.3- Перетворювач рівнів

3.2. Система управління живлення

Для того щоб управляти підключення живлення до цифрової мікросхеми і перетворювача рівнів, стала система управління живлення основі реле Т78–5VDC. Схема показані на рисунку. 3.4.

За структурною схемою Т78–5VDC видно, що реле Т78–5-VDC як двонаправлений перемикач з управлінням.

Якщо тип цифрової мікросхеми є ТТЛ, то 16-ий вив / хід підключений до живлення 5В. якщо тип цифрової мікросхеми є КМОП, то 16-ий вихід підключений

До складу даного модуля входять:

- 1) мікроконтролер atmega 128 header board (рис. 3.5);
- 2) програматор JTAG ICE XP II;
- 3) інтерфейси: RS-232, SPI, JTAG, ISP;
- 4) дроти і кабелі для з'єднання елементів між собою;

Програмне забезпечення для програмування та управління мікроконтролером.

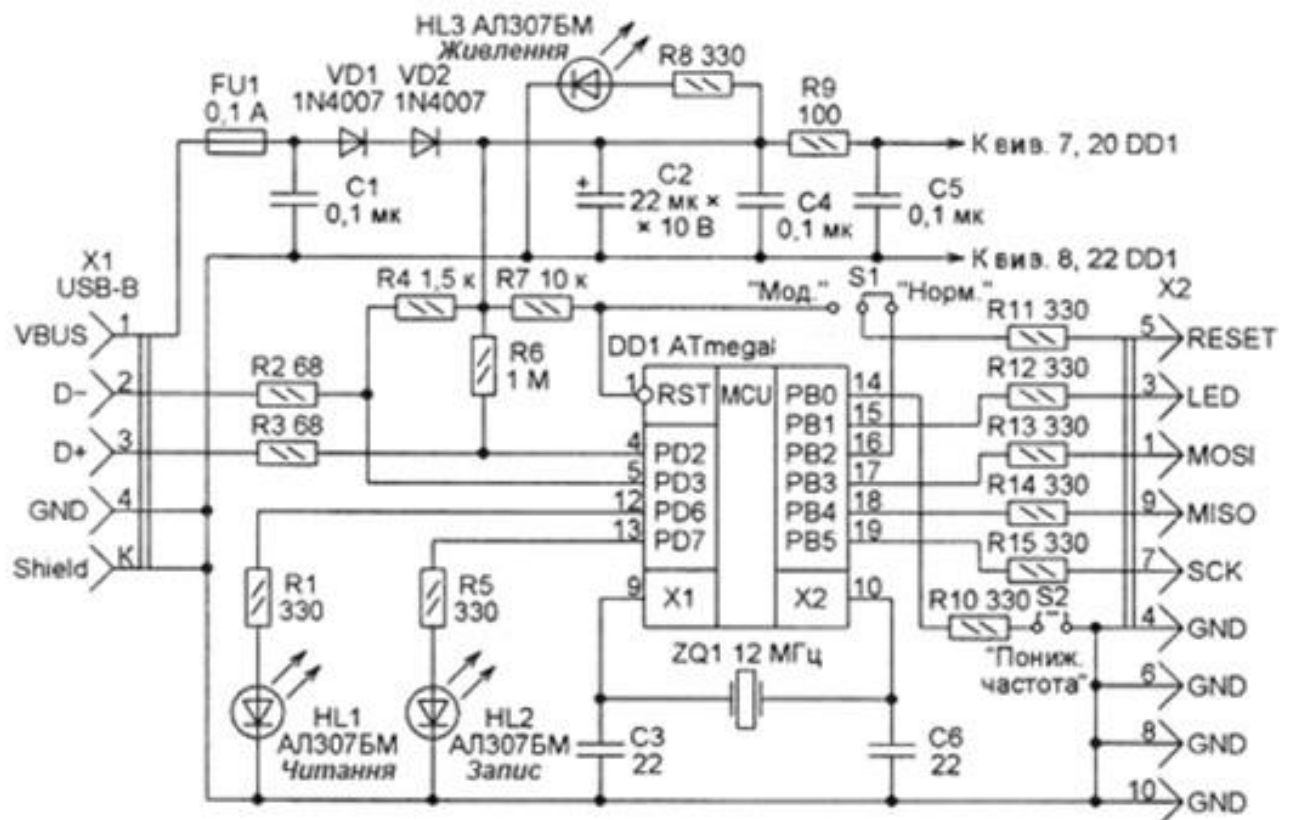


Рисунок 3.5- Схема програматора мікроконтролеру

Програматор JTAG ICE XP II показаний на рисунку. 3.6. Внутрішньосхемний програматор – відладчик підтримує всі AVR і AVR32 мікроконтролери з можливістю внутрішньосхемного налагодження. Можливості емулятора дозволяють переглядати внутрішній стан мікроконтролера, здійснювати швидкісну налагодження, програмувати пам'ять, програмувати через ISP або JTAG.

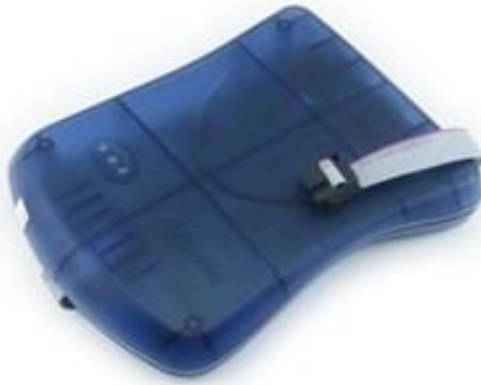


Рисунок 3.6- Програматор JTAG ICE XP II

Інтерфейси: RS–232, SPI, STAC, ISP, дроти кабелі.

Дані інтерфейси призначені для забезпечення зв'язку між пристроями модуля. У нашому випадку вони з'єднують:

- 1) персональний комп'ютер і програматор;
- 2) програматор і мікроконтролер;
- 3) мікроконтролер і мікросхему.

3.4. Розробка принципової схеми пристрою

Необхідно розробити та розрахувати осцилограф з наступними параметрами: розрядність 8 біт; частота дискретизації 1 МГц; діапазон вимірювання напруги 0... +50В; напруга живлення 7... 12В; похибка вимірювання $\pm 5\%$; максимальна частота вимірювання 200 кГц; вхідний опір не менше ніж 100 кОм; робота схеми в температурному діапазоні від – 10 до +40 °С.

Пристрій має включати а саме: функцію відображення сигналу у заданому діапазоні; точність відображення у заданому діапазоні; прийнятний час індикації; відображення результатів на екрані; можливість керування (рис. 3.7).

Для індикації використовуються світлодіоди KLS9-T0805UGC, які мають малий струм споживання та необхідну яскравість.

Діоди обрані відповідно до функцій, які вони виконують в схемі, а саме функцію захисту від перевищення допустимої напруги на вході осцилографа.

					КВРКІ 190246.19.22.48 ПЗ	Арк. 60
Зм.	Арк.	№ докум.	Підпис	Дата		

Основним критерієм при цьому були: напруга, швидкість відкриття каналу, ємність діоду, а також зворотна напруга пробою діода. Обрано діоди Шотткі ВАТ30, що забезпечують умови ТЗ та задані критерії.

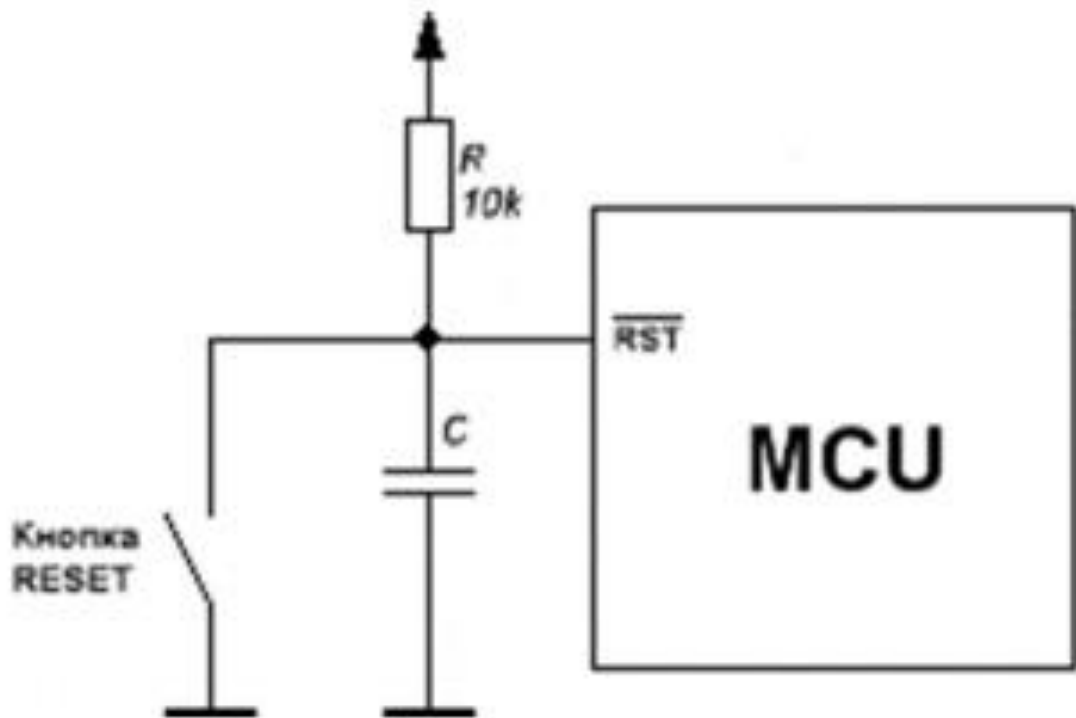


Рисунок 3.7- Схема скидання мікроконтролера

3.5. Розробка алгоритма його роботи

Використовуючи ПЗ, користувач вибирає тип логіки (ТТЛ, КМОП), тип корпусу (Б1 14, Б1Р16, 801 з 14, 801 з 16), тип мікросхеми (логічний елемент, комбінаційний логічний пристрій, послідовнісний логічний пристрій).

Далі посилає запит в базу даних звідки приходиться відповідь у вигляді назв мікросхем, що потрапляють під опис.

Користувач вибирає одну з наданих мікросхем, і посилає запит про надання даних про обрану

Користувачем мікросхеми в базу даних і отримує відповідь у вигляді команд настройки портів, а також команд для тестування даної мікросхеми.

За командою користувача по зраджує дані для налаштування портів тестер цифрових мікросхем (ВТЦМ) і запускає тестування мікросхеми.

Після завершення тестування ТЦМ зраджує дані в ПЗ, де відбувається аналіз і виведення результатів тестування.

Алгоритм «спілкування» ПК-МК.

Зв'язок мікроконтролера і програмного забезпечення відбувається за коштами інтерфейсу ІАТ [8].

Дані з ПЗ відправляються у вигляді пакетів. Пакет даних відправляються на мікроконтролер виглядає наступним чином:

мітка / наступні дані / % // . (3.1)

Мітка – це символ, службовець для позначення певної команди, і необхідний для визначення команди мікроконтролером. Мітками служать великі літери латинського алфавіту.

Мітки зазвичай складаються з великих літер латинського алфавіту, щоб вони були легко розпізнаваними та відрізнялися від інших елементів програми. Використання великих літер допомагає зробити мітки більш помітними та зрозумілими при перегляді програмного коду.

Наприклад, в програмному коді мікроконтролера можна мати мітку під назвою "START", яка позначає початок виконання програми. Коли мікроконтролер зустрічає цю мітку під час виконання коду, він знає, що повинен почати виконувати певні дії, які йдуть після мітки "START".

Використання міток дозволяє зробити програмний код більш структурованим, зрозумілим та підтримуваним. Вони є важливим елементом в програмуванні мікроконтролерів, який допомагає контролювати послідовність виконання команд та полегшує розробку та налагодження програмного забезпечення.

Наступні дані-це дані про висновки та режими роботи певних контактів необхідні мікроконтролери для правильного налаштування та взаємодії з

					КВРКІ 190246.19.22.48 ПЗ	Арк. 62
Зм.	Арк.	№ докум.	Підпис	Дата		

тестованою мікросхемою. Символ ' % ' означає закінчення пакета даних. Через наявність безлічі мікросхем, що відрізняються принципом роботи, була розроблена загальна система команд для управління пристроєм (Таблиця – 3.2).

Таблиця 3.2 - Команди, що посилаються на мікроконтролер

Назва команди	Позначення	Послідуючі дані за командою-;	Опис команди
Налаштування виводу	'P'	8 біт: номер виводу від 1-до – 16. 8 біт: позначення режиму out – o; in – i.	Команда, що інформує МК про те, який вивід як налаштувати
Установка логічної одиниці	'O'	8 біт: номер виводу від – 1 до 16.	Встановлюють логічну одиницю на заданий вивід
Установка логічного нуля	'Z'	8 біт: номер виводу від 1 до 16.	Встановлюємо логічний нуль на заданий вивід.
Зчитування	'R'	8 біт: номер виводу від 1 до 16.	Команда зчитування стану виводу. Після обробки та виконання даної команди відбувається відправка даних на ПК.
Імпульси	'I'	8 біт: номер виводу від 1 до 16. 8 біт: кількість імпульсів	Виводить на потрібний вивід необхідну кількість імпульсів
Підключення «землі»	'G'	8 біт: номер виводу 7 або 8.	Підключає мікросхему до «землі»
Підключення «живлення»	'U'	8 біт: символ '1' – підключення живлення, символ '0' – відключення живлення.	Підключення або відключення мікросхеми від живлення
Переключення «живлення»	'V'	Для +5В наступний символ: 'L'. Для +15В символ: 'H'.	Переключення значення Напруги живлення між +5 В та +15В і навпаки

Так як кожна команда виконується відразу ж після того як буде отримана, між командами в пакеті є тимчасові затримки 0,2 мс.

Вони потрібні для того щоб Мікроконтролер встиг обробити кожну прийняту команду.

Алгоритм тестування на прикладі мікросхеми КР1533ЛН1.

Для пояснення принципу тестування надано алгоритм тестування мікросхеми КР1533ЛН1. КР1533ЛН1-Цифрова інтегральна схема транзисторної логіки з діодами Шоттки серії ТТЛ. Являє собою шість логічних елементів НЕ.

Випускається в корпусі БІР-14. Внутрішнє розташування елементів і розташування висновків наведені на рисунку. 3.8. На початку тестування СУП підключає необхідний висновок тестованої мікросхеми до землі.

Далі відбувається підключення потрібного значення напруги живлення, 5В або 15В, яке залежить від логіки мікросхеми. І після цього подається напруга живлення на тестовану мікросхему.

Дані дії необхідні для безпечного підключення мікросхеми.

Далі відбувається послідовне тестування кожного інвертора окремо.

Перед подачею команд для тестування налаштовуються висновки мікроконтролера на певний режим.

Після цього подається логічний сигнал високого рівня і паралельно зчитується, і записується результат роботи інвертора до тих пір, поки не буде протестований кожен інвертор.

Далі таким же методом подається логічний сигнал низького рівня. Даний тест який зображений на рисунку 3.9 можна вважати універсальним і застосовувати до інших логічних елементів.

Після завершення тестування мікросхеми проводиться аналіз результатів і виведення їх у вікно програми.

					КВРКІ 190246.19.22.48 ПЗ	Арк. 64
Зм.	Арк.	№ докум.	Підпис	Дата		

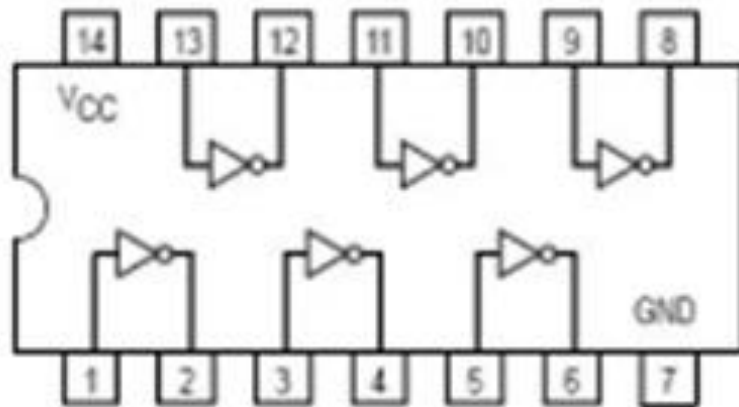


Рисунок 3.8- Розташування виводів мікросхеми КР1533ЛН1

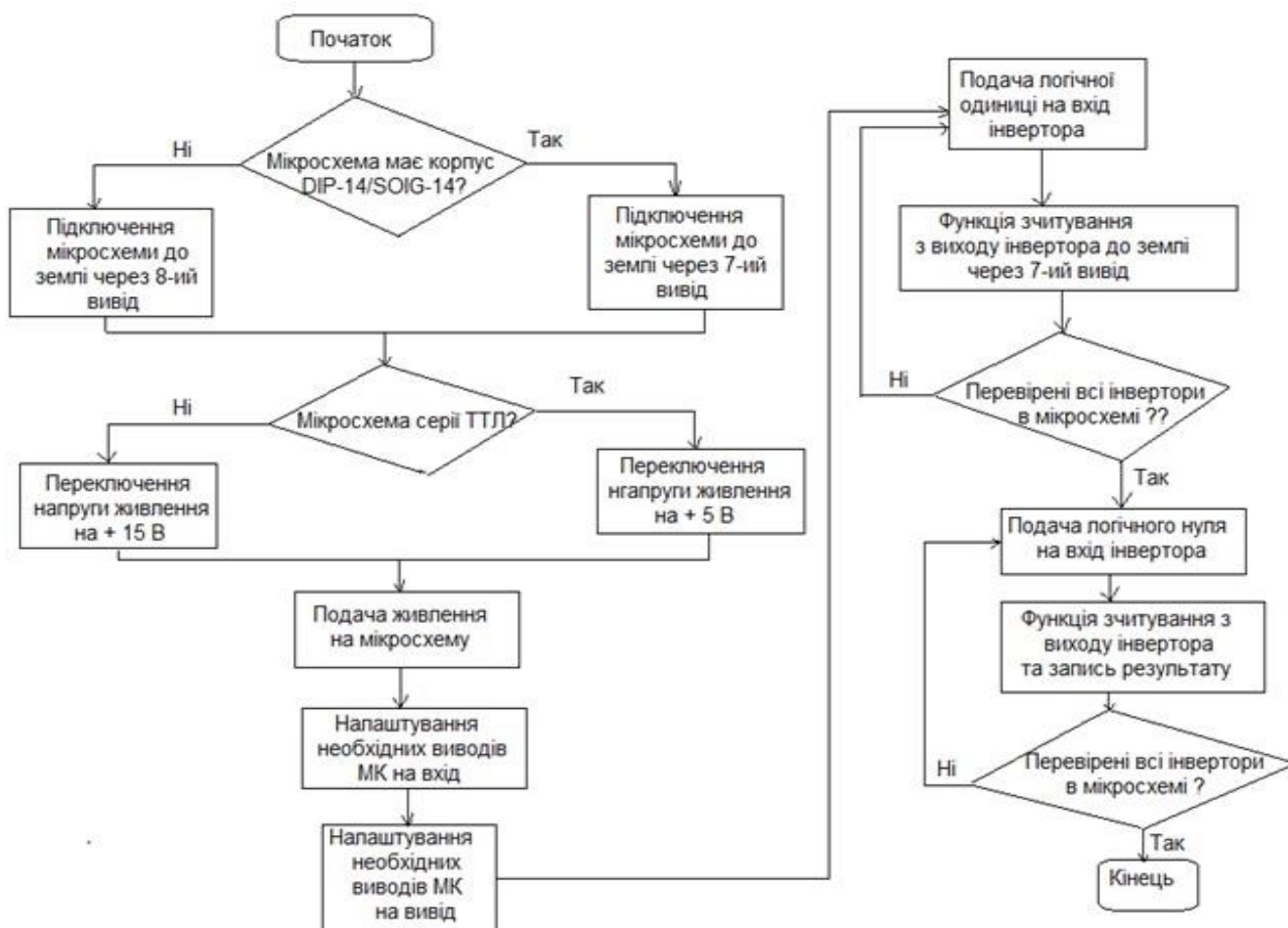


Рисунок 3.9- Алгоритм тестування на прикладі мікросхеми КР1533ЛН1, що складається з шести елементів «НЕ»

3.6. Розробка монтажної схеми пристрою

Для перевірки роботи пристрою, що зображений на рисунку 3.10 було вирішено провести експеримент, що складається з двох частин.

Перша частина – це тестування повністю робочої мікросхеми КР1533ЛН1.

Друга – тестування мікросхеми (рис. 3.11) з неробочим одним інвертором. Обидві частини експерименту проходять за алгоритмом, описаним в попередньому розділі.

Після підключення пристрою і установки мікросхеми в слот, був проведений тест повністю справної мікросхеми, показано на рисунку. 3.12.

Було отримано наступний звіт у програмі показаного на рисунку. 3.13 а). Програма показує, що мікросхема справна і тест пройдено.

Далі, в слот була поміщена мікросхема з свідомо несправним інвертором. Вхідний вивід інвертора (13 вивід) був не підключений. Звіт про даному експерименті наведено на рисунку. 3.13 б).



Рисунок 3.10. – Схема монтажна програмно-технічного комплексу розпізнання електроних компонентів



Рисунок 3.11- Зворотна сторона плати

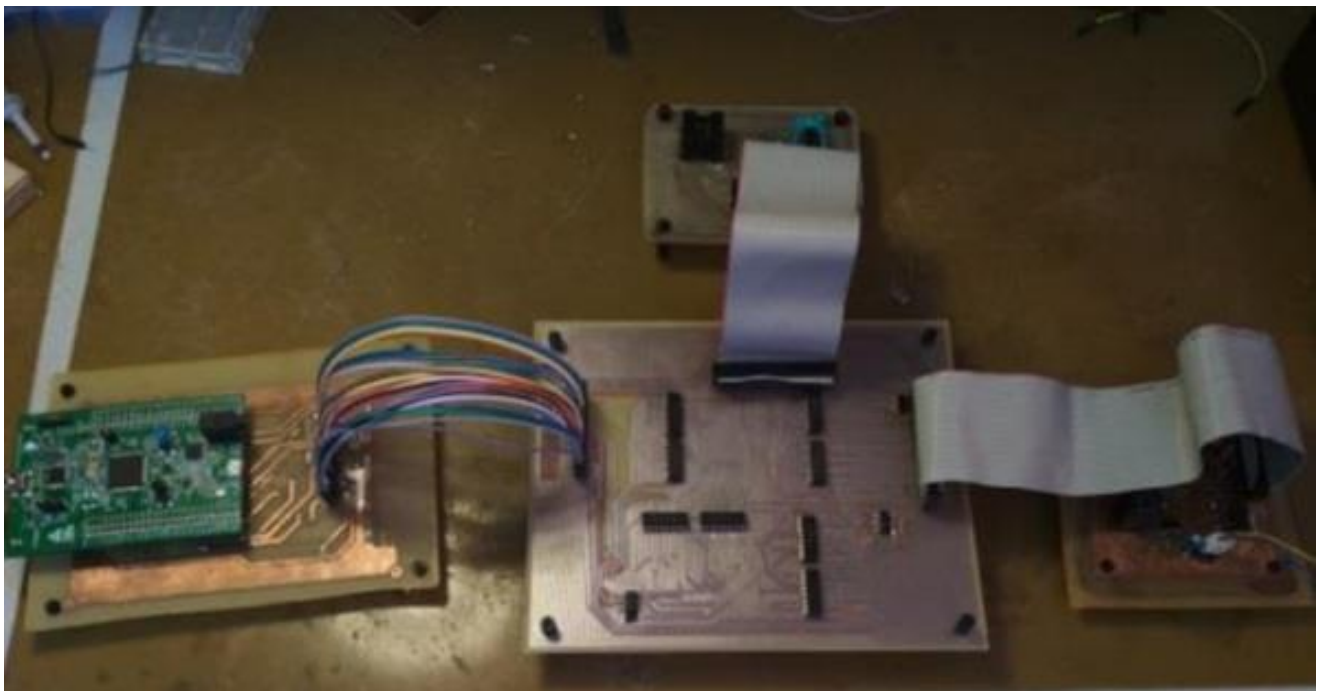


Рисунок 3.12- Макет програмно-технічного комплексу розпізнання електронних компонентів

Далі, в слот була поміщена мікросхема з свідомо несправним інвертором. Вхідний вивід інвертора (13 вивід) був не підключений. Звіт про даному експерименті наведено на рисунку. 3.13 б).

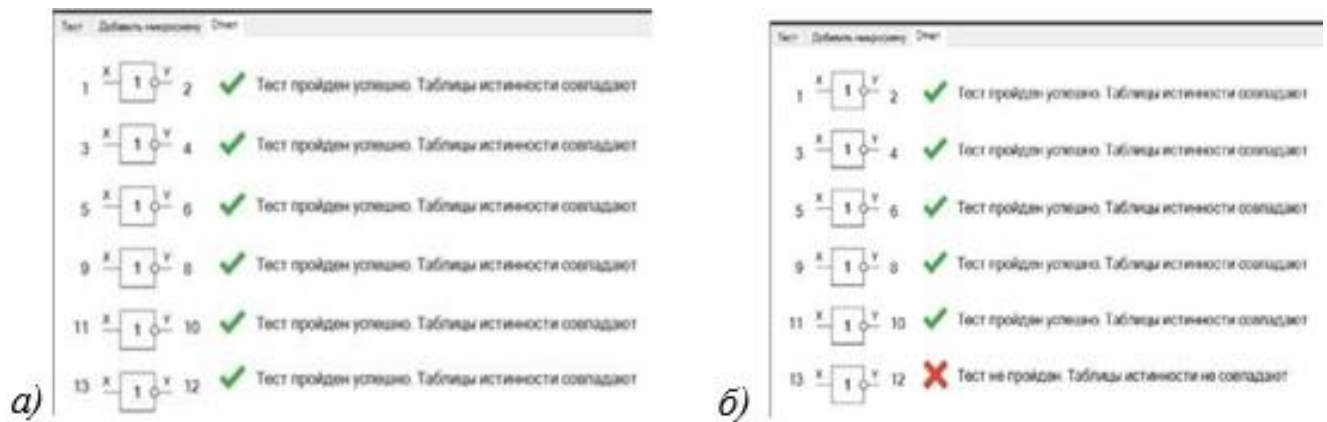


Рисунок 3.13- Успішно пройдений тест мікросхеми КР1533ЛН1 а) та не пройдений тест – б)

Як можна побачити в звіті програми тест виявився провальним. Програма вірно визначила несправний інвертор і сповістила про це користувача. Також можна зробити висновок, що несправність була визначена при подачі на вхід інвертора логічного нуля, при якому на виході повинна бути зчитана логічна одиниця. Це пояснюється тим, що при подачі логічної одиниці на виході був би логічний нуль, такий же результат буде і при несправному або не підключеному виведення.

Результати експерименту повністю відповідають очікуваному результату.

3.7. Програмне забезпечення для програмування та управління мікроконтролером

Для повного розкриття свого потенціалу, всім мікроконтролерам Atmel АУК необхідно надійне програмне забезпечення. Для його створення і налагодження використовується інтегроване середовище розробки (ГОЕ), наприклад Atmel Studio 7. Це середовище містить все необхідне для написання, компіляції і

налагодження коду і завантаження його прямо у флеш-пам'ять на чіпі мікроконтролера АУК без застосування додаткових програмних компонентів. Atmel Studio 7 підтримує такі мови програмування, як C++.

Atmel Studio 7 є потужним інструментом, який забезпечує зручне середовище розробки для програмування мікроконтролерів AVR. Воно підтримує різні мови програмування, включаючи C++, що дає розробникам можливість використовувати розширені можливості цієї мови для реалізації своїх проектів.

Atmel Studio 7 також надає широкий набір інструментів для налагодження, включаючи можливість встановлення точок зупину, стеження за змінними, аналізу стеку викликів та багато іншого. Це дозволяє розробникам ефективно відлагоджувати свій код і виявляти його помилки.

Завдяки інтегрованому підходу та вбудованим функціям, Atmel Studio 7 спрощує процес розробки програмного забезпечення для мікроконтролерів AVR, забезпечуючи швидкий цикл розробки та високу якість коду.

До основних особливостей цього середовища можна віднести підтримку більше 300 пристроїв Atmel АУЯ, наявність об'ємної бібліотеки, повне моделювання чіпа точної моделі мікроконтролера і периферійних пристроїв.

За допомогою програми можна відправляти і приймати дані про тестування через СОМ-порт комп'ютера по протоколу RS232.

Основні можливості такі:

- 1) лічильник переданих;
- 2) прийнятих байтів;
- 3) можливість відправляти файли; – крім стандартних швидкостей (baudrate)
- 4) можливість встановити свою нестандартну;
- 5) підтримує до 64 сом-портів;
- 6) можна весь хід роботи записувати в файл;
- 7) можна призначити до 24 макросів.

У розділі «Тест», реалізований механізм пошуку потрібної мікросхеми і запуску тесту для неї.

					КВРКІ 190246.19.22.48 ПЗ	Арк. 69
Зм.	Арк.	№ докум.	Підпис	Дата		

Для проведення тесту користувач вибирає тип логіки (ТТЛ, КМОП), тип корпусу (DIP14, DIP16, SOIC14, SOIC16) наведений на рисунку. 3.14, тип мікросхеми (логічний елемент, комбінаційний логічний пристрій, послідовнісний логічний пристрій).

Завдяки цьому механізму, користувач може швидко знайти потрібну мікросхему, встановити відповідні параметри для проведення тесту та запустити його. Це дозволяє ефективно виконувати процес тестування для різних типів мікросхем і забезпечує зручну інтерфейс для користувача.

Для обраної мікросхеми завантажується картинка з розташуванням висновків.

При натисканні кнопки «почати тест» відбувається з'єднання з мікроконтролером за коштами СОМ-порту, відкривається файл з набором команд для тестування обраної мікросхеми і послідовно з затримкою 1мс посилає їх мікроконтролеру, і періодично зчитує результати роботи мікросхеми.

Після закінчення тесту будується реальна таблиця істинності і порівнюється з очікуваною, результат виводиться у вікно зі звітом про тестування показаний на рисунку. 3.14 в).

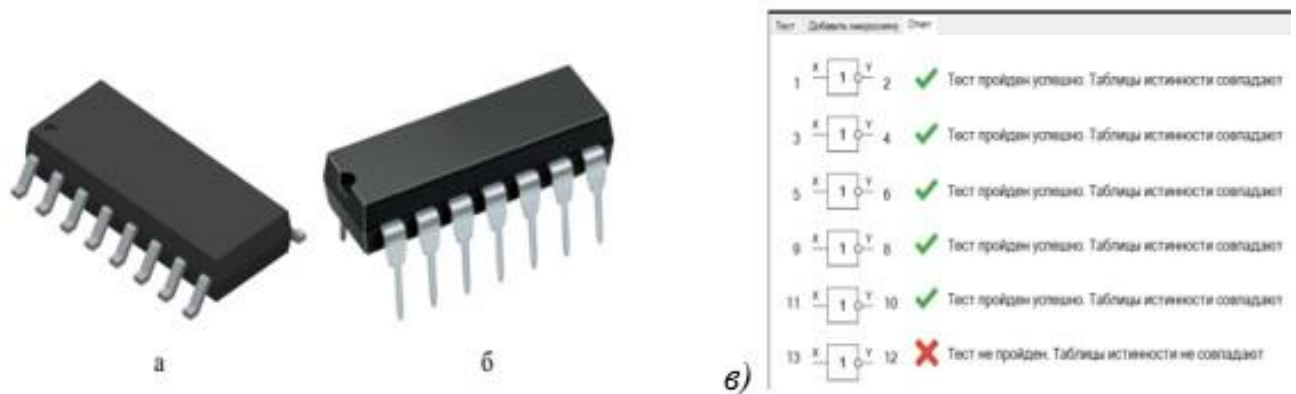


Рисунок 3.14- а – корпус SOIC16, б- корпус SOIC16 та вікно звіту про тестування в)

Далі ПЗ шукає в базі даних мікросхеми, які підходять під вибрані користувачем критерії. Користувач вибирає одну з представлених мікросхем.

У розділі «Додати мікросхему» реалізований механізм додавання мікросхеми в бібліотеку програми.

При виборі типу логічного пристрою з'являється панель з точним налаштуванням вхідних елементів.

На даний момент реалізований розділ логічні елементи при виборі користувачем даного розділу з'являється панель «вибір логічного елемента».

Далі користувач вибирає потрібний логічний елемент і задає його кількість в мікросхемі.

При збільшенні кількості логічних елементів з'являються спеціальні поля для налаштування ніг.

При натисканні користувачем на кнопку «Зберегти мікросхему» по здійсненій генерації набору команд і запис його в файл

3.8. Опис контрольованих об'єктів (мікросхеми серії 7400)

7400 серія інтегральних мікросхем на ТТЛ-логіці відома як перше широко поширене сімейство інтегральних мікросхем з ТТЛ-логікою. Вона призначена для побудови мейнфреймів і міні-комп'ютерів в 1970 і 1960 роках. Було кілька сумісних з розведення висновків поколінь оригінального сімейства, що став стандартом де факто.

Мікросхеми 7400 серії створені на різних технологіях, але Сумісність зберігалася з оригінальними рівнями логіки TTL і напругою живлення. Хоча елементи побудовані на КМОП-логіці, а не TTL, але зберігаються однакові номери, які визначають ідентичні логічні функції в різних підсеріях. Існує близько 40 підсерій, які використовують стандартні схеми нумерації.

Класифікації ІМС включаються в наступні:

- Бікмоп;
- ВСТ-Бікмоп, сумісні з вхідними рівнями перемикачів ТТЛ, використані в буфері;

– АВТ-поліпшені Бікмоп, з вхідними рівнями перемикання ТТЛ, швидше, ніж АСТ і ВСТ

1 біполярні ІМС:

– 74L – з малим споживанням (в порівнянні з оригінальним ТТЛ-сімейством), дуже повільні [9];

– 74 – «стандартне ТТЛ» сімейство, не має букв між «74» і номером пристрою;

– AS – покращені (Шотткі);

– F-швидкі (швидше звичайних Шотткі, аналогічні AS);

– LS-з малим споживанням (Шотткі);

– S-Шотткі (застаріли);

– ALS-поліпшені (Шотткі) з малим споживанням;

– H-високошвидкісні (до цих пір випускаються, але в основному замінені s-серіями, використовувалися в Ері комп'ютерів 1970-х);

2 КМОП

– C-КМОП, 4–15V, працюють як 4000 серія;

– НСТ-високошвидкісні, Сумісні за логічними рівнями з біполярними ІС;

– НС-високошвидкісні КМОП, аналогічні по швидкодії з Ls. 12 нс;

– АUC-низька напруга живлення – 0,8–2,7 В;

– АНС-поліпшені високошвидкісні КМОП, в три рази швидше НС;

– ALVC-низька напруга живлення – 1,65–3,3 В;

– АС-поліпшені КМОП, швидкодія в основному між S і F;

– FC-швидкі КМОП, швидкодія аналогічно з F;

– LVC-низька напруга живлення – 1,65–3,3 В і 5В входами;

– LCX-КМОП з 3В живильною напругою і 5В входами;

– LVQ-низька напруга живлення – 3,3 В;

– LVX-низька напруга живлення – 3,3 В і 5В входами;

– G-супер-високі швидкості (більше 1 ГГц), виробляються Potato Semiconductor;

					КВРКІ 190246.19.22.48 ПЗ	Арк. 72
Зм.	Арк.	№ докум.	Підпис	Дата		

– VHC – дуже високошвидкісні КМОП – «S» швидкодія з КМОП технологією і живлення;

Багато ІС в КМОП серіях HC, AC і FC також представлені в «Т» версіях, сумісних з Рівнями перемикачів і TTL, і 3,3 В КМОП. ІС без «Т» мають рівні перемикачів КМОП.

3.9. Аналіз та призначення будови мікроконтролера ATMEGA 128 Header board

AVR 128 Header board – макетна плата фірми OLIMEX для мікроконтролера ATmega128. ATmega128-малопотужний 8-розр. КМОП Мікроконтролер, за рахунок виконання більшості інструкцій за один машинний цикл він досягає продуктивності 1 млн операцій в секунду / МГц. Це дозволяє проектувальникам систем оптимізувати співвідношення енергоспоживання і швидкодії (рис 3.15) [12].

Основні параметри [13]:

- Високопродуктивний, малопотужний 8 – розрядний AVR-мікроконтролер.
- 133 потужних інструкцій, більшість з яких виконуються за один машинний цикл.
- 32 8-розр. регістрів загального призначення + регістри управління вбудованою периферією.
- повністю статична робота.
- продуктивність до 16 млн операцій в секунду при тактовій частоті 16 МГц.
- вбудований примножуючий пристрій виконує множення за 2 машинних циклу.
- енергонезалежність.
- 1000 циклів запис/стирання.
- внутрішньосистемне програмування вбудованої завантажувальної програмою.

					КВРКІ 190246.19.22.48 ПЗ	Арк. 73
Зм.	Арк.	№ докум.	Підпис	Дата		

- можливість читання під час запису.
- вбудоване статичне ОЗУ ємністю 4 кбайт.
- інтерфейс SPI для внутрішньосистемного програмування.
- інтерфейс JTAG (сумісність зі стандартом IEEE 1149.1).
- велика підтримка функцій вбудованої налагодження.
- програмування флеш-пам'яті.
- двопровідний послідовний інтерфейс, орієнтований на передачу даних в байтному форматі.
- послідовний інтерфейс SPI з підтримкою режимів ведучий / підлеглий.
- введення-виведення і корпусу: 53 – програмовані лінії введення-виведення.
- робочі напруги – 4.5–5.5 В.

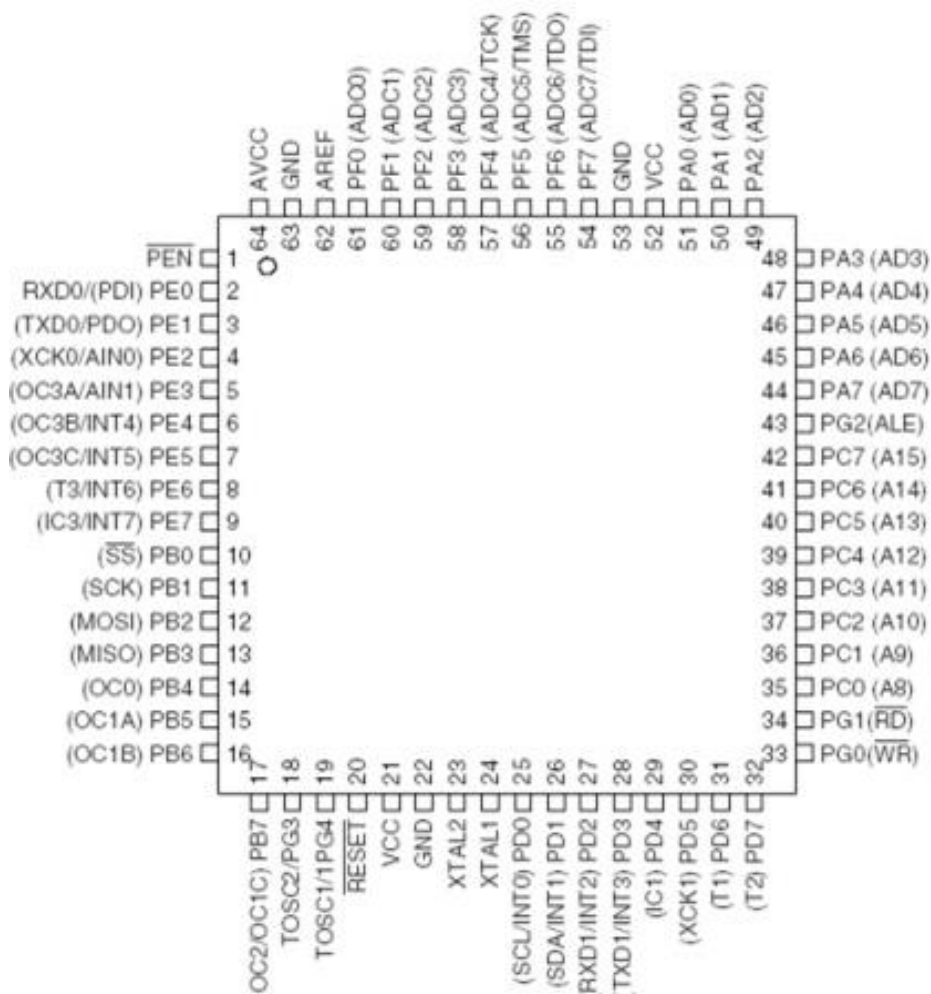


Рисунок 3.15- Розташування виводів

3.10. Висновок програмної реалізації

У середовищі Proteus було змодельовано дешифратор чисел від 0 до 9 у двійковому коді, який перетворює цей код на символи семисегментного індикатора.

Для моделювання використовувався мікроконтролер ATMEGA 128 Header board, який є прикладом мікроконтролера зі змодельованим дешифратором.

У аналітичній частині роботи була описана розробка структурної схеми пристрою контролю електронних компонентів.

Також були наведені розрахунки номіналів резисторів для структурної схеми цього пристрою. В роботі була описана черговість розробки алгоритму для створюваного пристрою, а також був наведений алгоритм тестування на прикладі мікросхеми КР1533ЛН1.

Для створення та моделювання пристрою було описано порядок розробки та представлено готову функціональну та принципову схему. Додатково, було проведено аналіз та реалізація програмної частини для цієї розробки.

Усі ці етапи роботи дозволили розробити функціональний пристрій, який виконує контроль електронних компонентів і має можливість тестування мікросхеми КР1533ЛН1.

Цей пристрій був розроблений з використанням моделювання в середовищі Proteus та програмної реалізації, що забезпечує його функціональність та ефективність.

					КвРКІ 190246.19.22.48 ПЗ	Арк. 75
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

На початку дослідження наголошено, що тестування мікросхем є важливим питанням дотримання якості та працездатності радіоелектронних мікросхем і компонентів, адже несвоєчасне виявлення таких вад може привести до поломок комп'ютерних систем на які вони будуть встановлені.

Проведено аналіз стану розвитку цифрових вимірювальних пристроїв (тестерів мікросхем). Дано опис таких найбільш поширених пристроїв для тестування. тестер мікросхем-YBD-868, цифровий інтегральний тестер на базі AT89C52, тестер інтегральних схем SIM100 та розглянуто теоретичні підходи до обґрунтування сутності процесу тестування цифрових вимірювальних пристроїв.

В якості прикладу змодельовано в середовищі Proteus дешифратор чисел (від 0 до 9 у десятковому представленні) з двійкового коду в символи семисегментного індикатора. Розглянуто будову та призначення мікроконтролера (ATMEGA 128 Header board).

В аналітичній частині роботи описана розробка структурної схеми пристрою контролю електронних компонентів. Наведено розрахунки номіналів резисторів для структурної схеми пристрою контролю електронних компонентів. Наведено черговість розробки алгоритму для створюваного пристрою та дано алгоритм тестування на прикладі мікросхеми KP1533ЛН1. В роботі описано порядок створення та приведено готову функціональну та принципову схему розробки. Зроблено опис монтажною схемою пристрою та наведено його макет.

Проведено обґрунтування необхідного обладнання для розробки дистанційно керованого модуля тестування мікросхем пам'яті з використанням програматора JTAG ICE XP II. Окремо проаналізована та реалізована програмна реалізація для розробки.

					КВРКІ 190246.19.22.48 ПЗ	Арк. 76
Зм.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Варфоломеев А. Ю. Функціонально-логічне проектування: Комбінаційні пристрої [Електронний ресурс] / КПІ ім. Ігоря Сікорського. К.: КПІ ім. Ігоря Сікорського, 2017 р. 135 с.
2. Головка М. В. Моделювання віртуального фізичного експерименту для систем дистанційного навчання у загальноосвітній та вищій педагогічній школах. *Інформаційні технології та засоби навчання*, 2015. Т. 47, №3. С. 36–48.
3. Джамал Х. М. Алгоритм і структура модуля для обчислення квадратного кореня в ПЛІС. *Праці міжнародної конференції «Безпека, відмовостійкість, інтелект»*, 2018. С. 74–77.
4. Дубровін В. С. Використання віртуальних лабораторних робіт – як елемент підвищення якості підготовки фахівців. *High technologies in Earth space research*, 2012. № 2. С. 11–13.
5. Казимир В. В. Проектування комп'ютерних систем на основі мікросхем програмованої логіки: монографія. Суми: СДУ. 2013. 313 с.
6. Козловський Є. А. Віртуальна лабораторія в структурі системи дистанційного навчання. *Інформаційні технології в освіті*, 2011. № 10. С. 102–109.
7. Колонтаєвський Ю. П. Мікропроцесорна техніка: конспект лекцій. Укл. Колонтаєвський Ю. П. Харків: ХНАМГ, 2009 89 с.
8. Николайчук Я. М. Проблеми реорганізації структури процесорів у різних теоретико-числових базисах. *Збірник матеріалів міжнародної наукової координаційної наради «Інформаційні проблеми комп'ютерних систем, юриспруденції, енергетики, економіки, моделювання та управління» (ICSM–2014)*. Тернопіль, 2014. С. 110–114.
9. Семеніхіна О. В. Теорія і практика формування професійної готовності майбутнього вчителя математики до використання засобів комп'ютерної візуалізації математичних знань: арэф. дис.... доктора пед. наук: 13.00.04 / ДВНЗ «Донбаський державний педагогічний університет». Слов'янськ, 2017.

					КвРКІ 190246.19.22.48 ПЗ	Арк. 77
Зм.	Арк.	№ докум.	Підпис	Дата		

10. Семеніхіна О. В. Віртуальні лабораторії як інструмент навчальної та наукової діяльності. *Педагогічні науки: теорія, історія, інноваційні технології*, 2011. №1 (11). С. 341–346.
11. Скорнякова А. В. Аналіз сформованості конкурентоспроможності майбутніх ІТ-фахівців в процесі вивчення професійних дисциплін. *Український психолого-педагогічний Науковий збірник*, 2016. № 9. С. 100–105.
12. Таненбаум Е. Архітектура комп'ютера. СПб.: Пітер, 2007. 844 с.
13. Трухін А. В. Використання віртуальних лабораторій в освіті. *Відкрита та Дистанційна освіта*, 2002. № 4 (8). С. 67–69.
14. Harris Dm, Harris SL цифрова схема та архітектура комп'ютера, Morgan Kaufman, 2013. 162 с.
15. Щедролосьєв Д. Є. Особливості підготовки ІТ-фахівців в українських вищих навчальних закладах. *Комп'ютер у школі та сім'ї*, 2010. №8. С. 12.
16. Яцків В. В. Контроль виконання арифметичних операцій на основі модулярних кодів. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2015. № 4 (53). С. 135–138.
17. Яшанов С М. Теоретичні та методичні проблеми застосування вільно розповсюдженого програмного забезпечення в інформативній підготовці майбутнього вчителя. *Освітній дискурс*, 2013. випуск 2. С. 18.
18. Lee J. Leitner, John W. Cane. A virtual laboratory environment for online IT education. *Proceeding SIGITE '05 Proceedings of the 6th conference on Information technology education*. (Newark, NJ, USA, October 20–22, 2005). 2005. P. 283–289.
19. Potkonjak V., Gardner M., Callaghan V., Mattila P., Guetl C., Petrović V. M., Jovanović K. Virtual laboratories for education in science, technology, and engineering: A review. *Computers and Education*, 2016. 95. P. 309–327.
20. ARRL Inc., Oscilloscopes for Radio Amateurs, 2015. 1. P. 150-155.
21. George A., Oscilloscope Survival Guide: For absolute beginners, 2020. P. 60-62.
22. ARRL Inc, Eric P. Nicolas (KL7AL). Digital Storage Oscilloscopes for Ham Radio. First Edition. 2016. P 50-53.

					КВРКІ 190246.19.22.48 ПЗ	Арк. 78
Зм.	Арк.	№ докум.	Підпис	Дата		

23. R. A. Penfold. How to Use Oscilloscopes and Other Test Equipment (BP), 1989. P. 100-110.
24. Robert Middleton. Troubleshooting With the Oscilloscope, 1975. P. 20-23.
25. James McClanahan. Oscilloscope Applications for the QRP Enthusiast, 2015. P. 34-37.
26. William A. Stolcklin. Understanding and Using Your Oscilloscope, 1968. P. 13-21.
27. Mr Robert J Davis II. Arduino Oscilloscope Projects, 2015. P. 88-92.
28. Homer B. Tilton. The 3-D Oscilloscope: A Practical Manual and Guide, 1987. P. 16-19.
29. Graham Stoakes. Automotive Oscilloscopes: Waveform Analysis, 2017. P. 156-157.
30. Robert L. Goodman. Troubleshooting With Your Triggered-Sweep Oscilloscope, First Edition. 1992. P. 155.
31. David Herres. Oscilloscopes: A Manual for Students, Engineers, and Scientists, First Edition. 2020. P. 199-204.
32. Laung-Terng Wang, Charles E. Stroud, Nur A. Touba. System-on-Chip Test Architectures: Nanometer Design for Testability, First Edition. P. 506.
33. Yervant Zorian. Multi-Chip Module Test Strategies, 1997 Edition. 1997. P. 99-120.
34. Erik Larsson. Introduction to Advanced System-on-Chip Test Design and Optimization, 2005 Edition. 2006. P. 400-403.
35. Vikram Iyengar, Anshuman Chandra. Test Resource Partitioning for System-on-a-Chip, 2002 Edition. 2012. P. 14-15.
36. Mark Burns, Gordon W. Roberts. An Introduction to Mixed-Signal IC Test and Measurement (The Oxford Series in Electrical and Computer Engineering), First Edition. 2000. P. 670-680.
37. Chip Davis, Daniel Chirillo, Daniel Gouveia, Fariz Saracevic, Jeffrey B. Bocarsley, Larry Quesada, Lee B. Thomas, Marc van Lint. Software Test Engineering

with IBM Rational Functional Tester: The Definitive Resource, First Edition. 2009. P. 888-889.

38. Charles Chien. Digital Radio Systems on a Chip: A Systems Approach, 2001 Edition. 2007. P. 500.

39. Windell Oskay, Eric Schlaepfer. Open Circuits: The Inner Beauty of Electronic Components, 2022. P. 23.

40. Ricardo Reis, Marcelo Soares Lubaszewski, Jochen A.G. Jess. Design of Systems on a Chip: Design and Test, 2007 Edition. 2007. P. 15.

					КВРКІ 190246.19.22.48 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		80

Додаток А

Фрагмент програмного коду для пристрою автоматичного розпізнавання типу електронних компонентів та вимірювання їх основних параметрів

```
using System;
using System. Data;
using System. IO;
using System. Windows. Forms;
using System. IO. Ports;
using System. Linq;
using System. Data. Entity;
using System. Collections. Generic;
using System. Threading;
using Tester. Models;
using System. Text;
using Tester. Resources;
using System. Collections;
namespace Tester
{
    public partial class Form1: Form
    {
        SerialPort port;
        Panel[] panels;
        ChipContext db;
        List<byte> result;
        List<string> logicOne;
        ArrayList LogicOne;
        List<string> logicZero;
        ArrayList LogicZero;
        ArrayList listCommands;
        PictureBox[] pictures;
        Label[] labels;
        Chip dataChip;
        ComboBox[] comboBoxes;
        //очистка массивов содержащих результирующих
        данных
        private void Clear(){
            if (logicOne. Count != 0)
                logicOne. Clear();
            if (logicZero. Count != 0)
                logicZero. Clear();
            if (LogicZero. Count != 0)
                LogicZero. Clear();
            if (LogicOne. Count != 0)
                LogicOne. Clear();
            if (result. Count != 0)
                result. Clear();
            textBox2. Clear();
        }
        //анлиз полученных результатов
        private void ResultAnalysis(){
36
            PictureBox[] pictureBoxes = new[] { pictureBox3,
            pictureBox4, pictureBox5,
            pictureBox6, pictureBox7, pictureBox8 };
            foreach (var p in pictureBoxes){
                p. Visible = true;
                p. ImageLocation = @». / Resources / Images / ne. gif»;
                p. Load();
            }
            int countOut;
            if (result. Count == 0)
                return;
            if (dataChip. TypeCase == «DIP14» || dataChip.
                TypeCase == «SOIC14»)
                countOut = 6;
            else countOut = 7;
            for (int l=0, i =0, j=1; l < countOut && j < result.
                Count / 2; l++, i=i + 2, j = j + 2){
                if((Convert. ToByte (LogicOne [l]) == result [i] && l
                ==
                result [j])&&(Convert. ToByte (LogicOne [l]) == result
                [result. Count / 2 + i] && 0 ==
                result [result. Count / 2 + j])){
                    pictures [l]. ImageLocation = @». / Resources / Images
                    / Yes. png»;
                    pictures [l]. Load();
                    labels [l]. Text = «Тест пройден успешно.»;
                }
                else{
                    pictures [l]. ImageLocation =
                    @». / Resources / Images / w128h1281338911337cross.
                    png»;
                    pictures [l]. Load();
                    labels [l]. Text = «Тест не пройден.»;
                }
            }
            //чтение данных из COM порта
            private void ReadPort(){
                Thread. Sleep (100);
                int c = port. BytesToRead;
                byte[] text = new byte [c];
                port. Read (text, 0, c);
                foreach (var s in text){
                    textBox2. Text += s;
                    result. Add (s);
                }
            }
            //генерация теста
            private void TestGen(){
                using (StreamWriter writer = new
                StreamWriter (File. Create($». // Resources //
                TestFiles/{nameChip. Text}.txt»))
                {
37
                    if (DIP14. Checked || SOIC14. Checked){
                        writer. WriteLine(@»G07%»);
                    }
                    else if (DIP16. Checked || SOIC16. Checked){
                        writer. WriteLine(@»G08%»);
                    }
                }
            }
        }
    }
}
```

```

}
if (TTL.Checked){
writer.WriteLine («VL%»);
}
else if (KMOP.Checked){
writer.WriteLine («VH%»);
}
writer.WriteLine (@»U01%»);
if (radioButton1.Checked){
if (LogElements.Checked && (DIP14.Checked ||
SOIC14.Checked)){
for (int i = 0; i < 12; i += 2){
writer.WriteLine ($»P{comboBoxes [i]. Text}o%»);
writer.WriteLine ($»P{comboBoxes [i + 1]. Text}i%»);
}
for (int i = 0; i < 12; i += 2){
writer.WriteLine ($»Z{comboBoxes [i]. Text}%»);
writer.WriteLine ($»R{comboBoxes [i + 1]. Text}%»);
}
for (int i = 0; i < 12; i += 2){
writer.WriteLine ($»O{comboBoxes [i]. Text}%»);
writer.WriteLine ($»R{comboBoxes [i + 1]. Text}%»);
}
}
if (LogElements.Checked && (DIP16.Checked ||
SOIC16.Checked)){
for (int i = 0; i < 14; i += 2){
writer.WriteLine ($»P{comboBoxes [i]. Text}o%»);
writer.WriteLine ($»P{comboBoxes [i + 1]. Text}i%»);
}
for (int i = 0; i < 14; i += 2){
writer.WriteLine ($»Z{comboBoxes [i]. Text}%»);
writer.WriteLine ($»R{comboBoxes [i + 1]. Text}%»);
}
for (int i = 0; i < 14; i += 2){
writer.WriteLine ($»O{comboBoxes [i]. Text}%»);
}
38
writer.WriteLine ($»R{comboBoxes [i + 1]. Text}%»);
}
}
}
if
(radioButton4.Checked||radioButton2.
Checked||radioButton3.Checked||radioButton6.
Checked){
if (LogElements.Checked && (DIP14.Checked ||
SOIC14.Checked|| DIP16.Checked
|| SOIC16.Checked)){
for (int i = 0, k=13; i < 8; i += 2, k++){
writer.WriteLine ($»P{comboBoxes [i]. Text}o%»);
writer.WriteLine ($»P{comboBoxes [k + 1]. Text}o%»);
writer.WriteLine ($»P{comboBoxes [i + 1]. Text}i%»);
}
for (int i = 0, k = 13; i < 8; i += 2, k++){
writer.WriteLine ($»Z{comboBoxes [i]. Text}%»);
writer.WriteLine ($»Z{comboBoxes [k + 1]. Text}%»);
writer.WriteLine ($»R{comboBoxes [i + 1]. Text}%»);
}
for (int i = 0, k = 13; i < 8; i += 2, k++){
writer.WriteLine ($»O{comboBoxes [i]. Text}%»);
writer.WriteLine ($»O{comboBoxes [k + 1]. Text}%»);
writer.WriteLine ($»R{comboBoxes [i + 1]. Text}%»);
}
}
}

```

```

}
for (int i = 0, k = 13; i < 8; i += 2, k++){
writer.WriteLine ($»Z{comboBoxes [i]. Text}%»);
writer.WriteLine ($»O{comboBoxes [k + 1]. Text}%»);
writer.WriteLine ($»R{comboBoxes [i + 1]. Text}%»);
}
for (int i = 0, k = 13; i < 8; i += 2, k++){
writer.WriteLine ($»O{comboBoxes [i]. Text}%»);
writer.WriteLine ($»Z{comboBoxes [k + 1]. Text}%»);
writer.WriteLine ($»R{comboBoxes [i + 1]. Text}%»);
}
}
}
}
public Form1(){
InitializeComponent();
panels = new Panel[] {panel1, panel2, panel3, panel4,
panel5, panel6, panel7 };
db = new ChipContext();
db.Chips.Load();
port = new SerialPort();
39
result = new List<byte>();
logicOne = new List<string>();
logicZero = new List<string>();
LogicOne = new ArrayList();
LogicZero = new ArrayList();
pictures = new[] { pictureBox2, pictureBox10,
pictureBox11, pictureBox12,
pictureBox13, pictureBox14 };
labels = new[] { label19, label20, label21, label22,
label23, label24 };
comboBoxes = new[] { numberPin0, numberPin1,
numberPin2, numberPin3,
numberPin4, numberPin5, numberPin6, numberPin7,
numberPin8,
numberPin9,
numberPin10, numberPin11, numberPin12,
numberPin13, comboBox4, comboBox5, comboBox6, co
mboBox7, comboBox8, comboBox9, comboBox10 };
}
private void Form1_Load (object sender, EventArgs e){
#region Tecm
private void Button1_Click (object sender, EventArgs e){
COMports.Items.Clear();
COMports.Items.AddRange (SerialPort.
GetPortNames());
}
//поиск нужной микросхемы
private void Search_Click (object sender, EventArgs e){
selectChip.Items.Clear();
string typeLogic = comboBox1.Text;
string typeDevice = comboBox2.Text;
string typeCase = comboBox3.Text;
IEnumerable<Chip> chips;
if ((typeLogic == «») && (typeDevice == «») &&
(typeCase == «»)){
chips = from chip in db.Chips.Local select chip;
}
else if ((typeCase != «») && (typeDevice == «») &&
(typeLogic == «»)){
}
}
}
}

```

```

chips = from chip in db. Chips. Local where chip.
TypeCase == typeCase select chip;
}
else if ((typeCase == «» && (typeDevice != «») &&
(typeLogic == «»)){
chips = from chip in db. Chips. Local where chip.
TypeDevice == typeDevice select
chip;
}
else if ((typeCase == «») && (typeDevice == «») &&
(typeLogic != «»)){
chips = from chip in db. Chips. Local where chip.
TypeLogic == typeLogic select chip;
}
else if ((typeCase == «») && (typeDevice != «») &&
(typeLogic != «»)){
chips = from chip in db. Chips. Local where (chip.
TypeLogic == typeLogic) &&
(chip. TypeDevice == typeDevice) select chip;
}
else if ((typeCase != «») && (typeDevice == «») &&
(typeLogic != «»))
40
{
chips = from chip in db. Chips. Local where (chip.
TypeLogic == typeLogic) &&
(chip. TypeCase == typeCase) select chip;
}
else if ((typeCase != «») && (typeDevice != «») &&
(typeLogic == «»)){
chips = from chip in db. Chips. Local where (chip.
TypeCase == typeCase) &&
(chip. TypeDevice == typeDevice) select chip;
}
else {
chips = from chip in db. Chips. Local
where
(chip. TypeCase == typeCase) &&
(chip. TypeLogic == typeLogic) &&
(chip. TypeDevice == typeDevice)
select chip;
}
foreach (Chip chip in chips){
selectChip. Items. Add (chip. Name);
}
}
//заныцк mecma
private void StartTest_Click (object sender, EventArgs
e){
Clear();
StringBuilder str = new StringBuilder();
dataChip = (from chip
in db. Chips. Local
where chip. Name == selectChip. SelectedItem.
ToString()
select chip). First();
string locationFile = dataChip. File;
if (COMports. SelectedItem != null)
{
try
{
if (port. IsOpen)

```

```

port. Close();
}
catch (Exception ex)
{
ExceptionLable. Text = ex. Message;
}
port. PortName = COMports. SelectedItem. ToString();
port. BaudRate = 115200;
port. Parity = Parity. None;
port. DataBits = 8;
port. StopBits = StopBits. One;
try
{
port. Open();
41
}
catch (Exception ex){
ExceptionLable. Text = ex. Message;
}
ExceptionLable. Text = «Выбран: «+ COMports.
SelectedItem. ToString();
if (port. IsOpen){
using (TextReader sr = new StreamReader
(locationFile)){
List<string> testCommands = new List<string>();
try {
for (string command = sr. ReadLine(); command !=
null; command =
sr. ReadLine()){
testCommands. Add (command);
if (command. Contains («O»)) {
logicOne. Add (command);
}
else if (command. Contains («Z»)) {
logicZero. Add (command);
}
}
for (int l = 1; l < Parser. ParseCommands (logicOne).
Count; l++) {
LogicOne. Add (Convert. ToByte (Parser.
ParseCommands (logicOne) [l]));
}
for (int l = 1; l < Parser. ParseCommands (logicZero).
Count; l++) {
LogicZero. Add (Convert. ToByte (Parser.
ParseCommands (logicZero) [l]));
}
LogicOne. Sort();
LogicZero. Sort();
LogicOne. RemoveRange (6, LogicOne. Count - 6);
LogicZero. RemoveRange (6, LogicZero. Count - 6);
for (int i = 0; i < LogicOne. Count; i++){
if (Convert. ToByte (LogicOne [i]) < 8)
LogicOne [i] = Convert. ToByte (LogicOne [i]) + 1;
else LogicOne [i] = Convert. ToByte (LogicOne [i]) - 1;
}
for (int i = 0; i < LogicZero. Count; i++){
if (Convert. ToByte (LogicZero [i]) < 8)
LogicZero [i] = Convert. ToByte (LogicZero [i]) + 1;
else LogicZero [i] = Convert. ToByte (LogicZero [i]) -
1;
}
}
}

```

```

//запись в COM порт
42
int count = 0;
bool flag = false;
bool delay = true;
listCommands = Parser. ParseCommands
(testCommands);
foreach (var symbol in listCommands)
{
port. Write (new byte[] { Convert. ToByte (symbol) }, 0,
1);
if (Convert. ToChar (symbol) == '%') // задержка
между командами
Thread. Sleep (1);
if (delay && (Convert. ToChar (symbol) == 'Z' ||
Convert. ToChar (symbol) ==
'O')) //задержка между блоком команд настройки и
блоком команд подачи сигнала и
считывания
{
delay = false;
Thread. Sleep (2);
}
if (Convert. ToChar (symbol) == 'R')
flag = true;
if (flag == true)
count++;
if (count == 3){
count = 0;
flag = false;
ReadPort();
}
}
catch (Exception ex)
{
ExceptionLable. Text = ex. Message;
}
}
}
else {
ExceptionLable. Text = «COM-порт не был выбран»;
}
ResultAnalysis();
}
//выбор микросхемы
private void selectChip_SelectedIndexChanged (object
sender, EventArgs e){
string selectedChip = selectChip. SelectedItem.
ToString();
IEnumerable<string> image = from chip in db. Chips.
Local where chip. Name ==
selectedChip select chip. Image;
pictureBox1. ImageLocation = image. First();
if (pictureBox1. ImageLocation != «»)
pictureBox1. Load();
}
#endregion
43
#region Добавление микросхемы
private void LogElements_CheckedChanged (object

```

```

sender, EventArgs e){
SelectLogElem. Visible = true;
SelectLogElem. Enabled = true;
SelectedSequentialDevice. Visible = false;
SelectedSequentialDevice. Enabled = false;
}
private void CombDevice_CheckedChanged (object
sender, EventArgs e){
SelectLogElem. Visible = false;
SelectLogElem. Enabled = false;
SelectedSequentialDevice. Visible = false;
SelectedSequentialDevice. Enabled = false;
}
private void SequentialDevice_CheckedChanged (object
sender, EventArgs e){
SelectLogElem. Visible = false;
SelectLogElem. Enabled = false;
SelectedSequentialDevice. Visible = true;
SelectedSequentialDevice. Enabled = true;
}
private void numericUpDown15_ValueChanged (object
sender, EventArgs e){
if (radioButton1. Checked){
for (int i = 0; i < numericUpDown15. Value; i++){
panels [i]. Visible = true;
}
for (int i = Convert. ToInt32 (numericUpDown15.
Value); i < panels. Length; i++){
panels [i]. Visible = false;
}
}
if (radioButton2. Checked || radioButton3. Checked ||
radioButton4. Checked ||
radioButton6. Checked){
for (int i = 0; i < numericUpDown15. Value; i++){
panels [i]. Visible = true;
}
for (int i = Convert. ToInt32 (numericUpDown15.
Value); i < panels. Length; i++){
panels [i]. Visible = false;
}
}
}
private void DIP14_CheckedChanged (object sender,
EventArgs e){
if (radioButton1. Checked) numericUpDown15.
Maximum = 6;
44
else numericUpDown15. Maximum = 4;
foreach (var box in comboBoxes)
{
box. Items. Clear();
box. Items. AddRange (new string[] { «01», «02», «03»,
«04», «05», «06», «08», «09»,
«10», «11», «12», «13» });
}
}
private void DIP16_CheckedChanged (object sender,
EventArgs e){
if (radioButton1. Checked) numericUpDown15.
Maximum = 7;
else numericUpDown15. Maximum = 4;
}
}

```

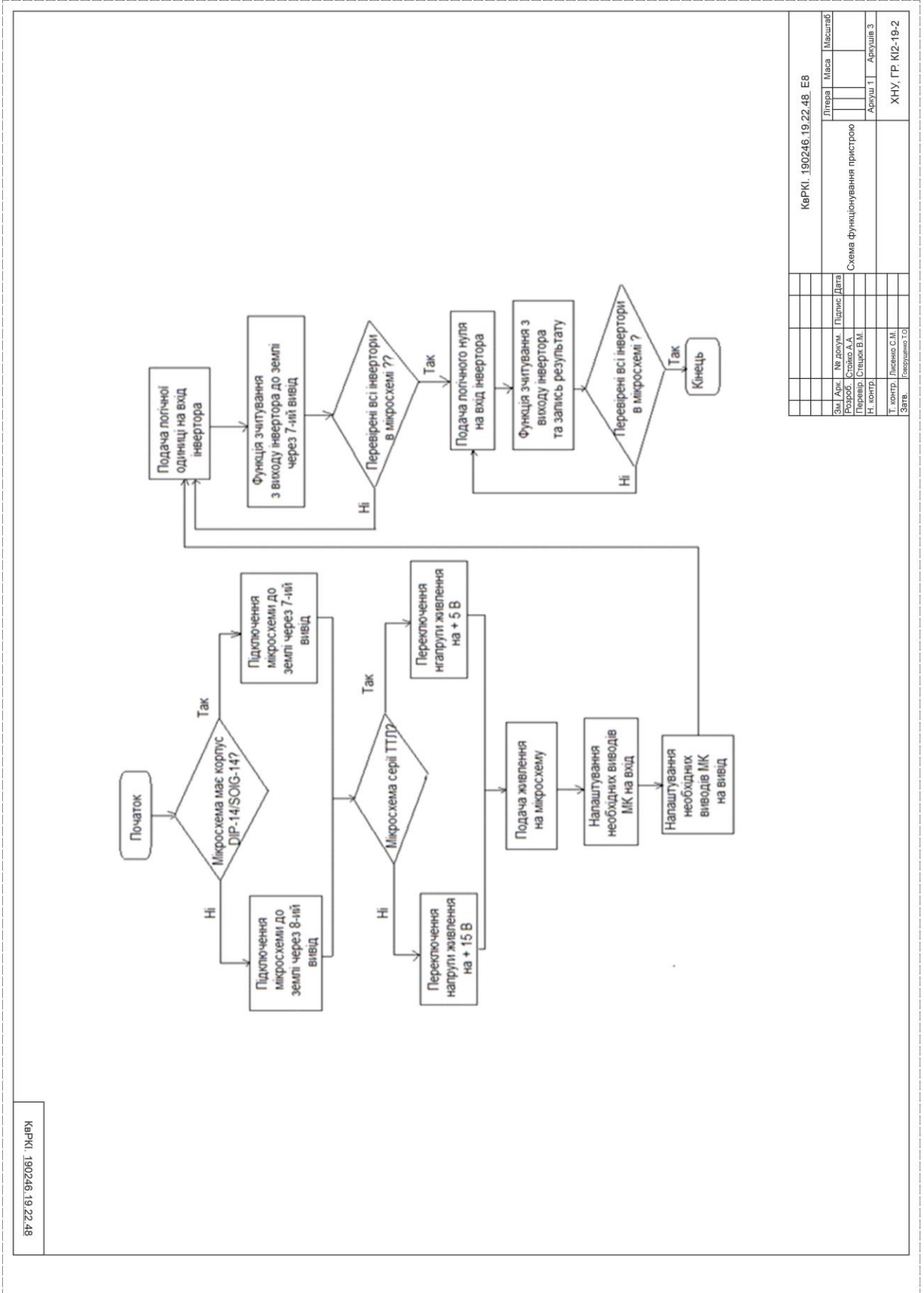
```

foreach (var box in comboBoxes){
    box.Items.Clear();
    box.Items.AddRange(new string[] { «01», «02», «03»,
«04», «05», «06», «07», «09»,
«10», «11», «12», «13», «14», «15» });
}
}
private void SOIC14_CheckedChanged (object sender,
EventArgs e){
    if (radioButton1.Checked) numericUpDown15.
Maximum = 6;
    else numericUpDown15.Maximum = 4;
    foreach (var box in comboBoxes){
        box.Items.Clear();
        box.Items.AddRange(new string[] { «01», «02», «03»,
«04», «05», «06», «08», «09»,
«10», «11», «12», «13» });
    }
}
private void SOIC16_CheckedChanged (object sender,
EventArgs e){
    if (radioButton1.Checked) numericUpDown15.
Maximum = 7;
    else numericUpDown15.Maximum = 4;
    foreach (var box in comboBoxes){
        box.Items.Clear();
        box.Items.AddRange(new string[] { «01», «02», «03»,
«04», «05», «06», «07», «09»,
«10», «11», «12», «13», «14», «15» });
    }
}
private void radioButton1_CheckedChanged (object
sender, EventArgs e){
    if (DIP14.Checked || SOIC14.Checked)
numericUpDown15.Maximum = 6;
    else if (DIP16.Checked || SOIC16.Checked)
numericUpDown15.Maximum = 7;
    for (int i = 0; i < comboBoxes.Length; i++)
    {
        if (i >= 14){
            comboBoxes [i].Enabled = false;
45
            comboBoxes [i].Visible = false;
        }
        else comboBoxes [i].Top = 15;
    }
    ImageElement1.ImageLocation = @». / Resources /
Images / ne.gif»;
    ImageElement2.ImageLocation = @». / Resources /
Images / ne.gif»;
    ImageElement3.ImageLocation = @». / Resources /
Images / ne.gif»;
    ImageElement4.ImageLocation = @». / Resources /
Images / ne.gif»;
    ImageElement5.ImageLocation = @». / Resources /
Images / ne.gif»;
    ImageElement6.ImageLocation = @». / Resources /
Images / ne.gif»;
    ImageElement7.ImageLocation = @». / Resources /
Images / ne.gif»;
    ImageElement1.Load();
    ImageElement2.Load();
    ImageElement3.Load();
    ImageElement4.Load();
    ImageElement5.Load();
    ImageElement6.Load();
    ImageElement7.Load();
}
private void radioButton2_CheckedChanged (object
sender, EventArgs e){
    numericUpDown15.Maximum = 4;
    for (int i = 0; i < comboBoxes.Length; i++){
        if (i >= 14){
            comboBoxes [i].Top = 30;
            comboBoxes [i].Enabled = true;
            comboBoxes [i].Visible = true;
        }
        else comboBoxes [i].Top = 0;
    }
    ImageElement1.ImageLocation = @». / Resources /
Images / and.gif»;
    ImageElement2.ImageLocation = @». / Resources /
Images / and.gif»;
    ImageElement3.ImageLocation = @». / Resources /
Images / and.gif»;
    ImageElement4.ImageLocation = @». / Resources /
Images / and.gif»;
    ImageElement5.ImageLocation = @». / Resources /
Images / and.gif»;
    ImageElement6.ImageLocation = @». / Resources /
Images / and.gif»;
    ImageElement7.ImageLocation = @». / Resources /
Images / and.gif»;
    ImageElement1.Load();
    ImageElement2.Load();
    ImageElement3.Load();
    ImageElement4.Load();
    ImageElement5.Load();
    ImageElement6.Load();
    ImageElement7.Load();
}
private void radioButton3_CheckedChanged (object
sender, EventArgs e){
    numericUpDown15.Maximum = 4;
    for (int i = 0; i < comboBoxes.Length; i++)
}

```

Додаток Б (обов'язковий)

Копія креслення схеми функціонування пристрою



Картка: 190246.19.22.48

Картка: 190246.19.22.48. Е8		Літера	Маса	Масштаб
Вид. Акт.	№ докум.	Підпис	Дата	
Розроб.	Стало А.А.			
Н. контр.	Перевір. Стецько В.М.			
	Т. контр.	Лисенко С.М.		
	Затв.	Богорядово Т.О.		
Схема функціонування пристрою				Аркуш 1 Аркуш 3
				ХНУ, ГР КІ2-19-2

Додаток В (обов'язковий)

Копія креслення схеми функційної пристрою

КвРКі. 190246.19.22.48

Схема програнатора з підключенням до паралельного порту

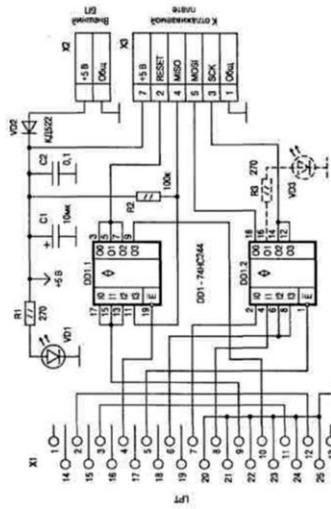


Схема комунаційного контролера

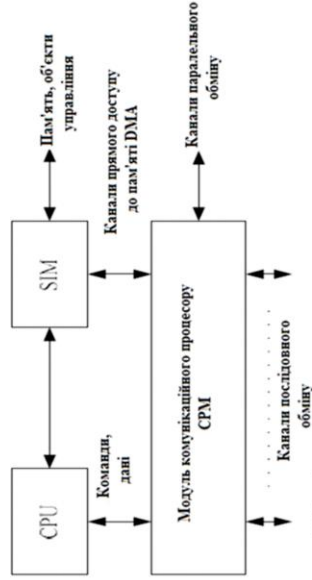
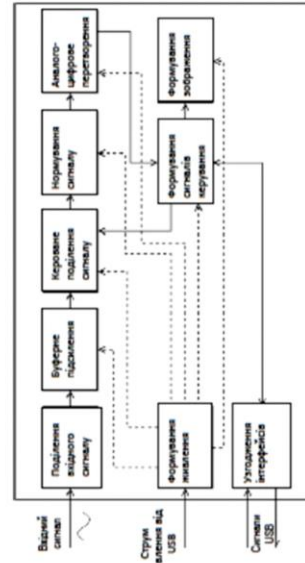


Схема розробленого пристрою



КвРКі. 190246.19.22.48 E8		Літера	Маса	Масштаб
Зм. Арк.	№ доум.	Підпис	Дата	
Розроб.	Сторо А.А.			
Перевір.	Стецюк В.М.			
Н. контр.				
Т. контр.	Лісєно С.М.			
Зав.	Варшавський			
Функційна схема пристрою				
ХНУ, ГР: Кі2-19-2				

Додаток Г (обов'язковий)

Копія креслення схеми принципової пристрою

КвРКІ. 190246.19.22.48

Схема джерела живлення

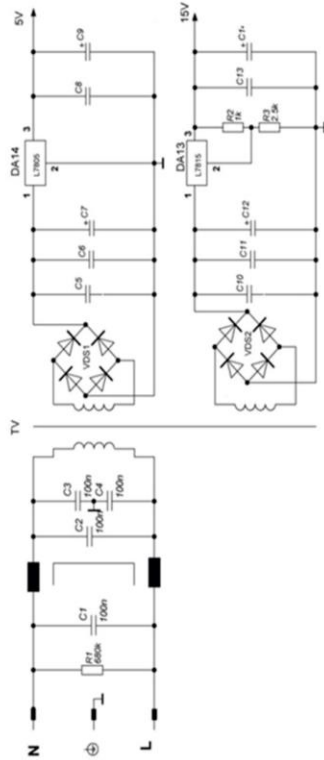
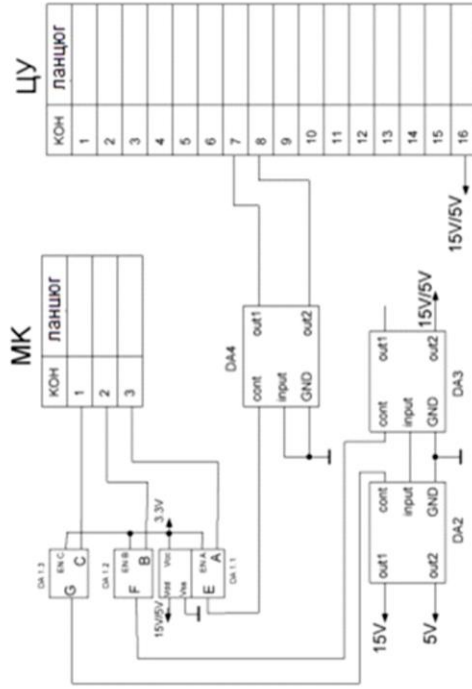


Схема системи управління живлення



КвРКІ. 190246.19.22.48.ЕВ		Літера	Маса	Масштаб
Зм. Акт.	№ розум.	Підпис	Дата	
Розроб.	Склад А.А.			
Перевір.	Стецькев В.М.			
Н. контр.				Архив 1 Архив 3
Т. контр.	Лисенко С.М.			ХНУ. ГР. К12-19-2
Зав. В.	Павловський І.О.			
Принципова схема пристрою				

Ім'я користувача: Кафедра КІ

ID перевірки: 1015687358

Дата перевірки:
24.06.2023 18:42:46 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
24.06.2023 18:43:42 EEST

ID користувача: 100005591

Назва документа: Стойко_Програмно-технічний пристрій розпізнавання типу електронних компонентів та ви...

Кількість сторінок: 83

Кількість слів: 14342 Кількість символів: 102127

Розмір файлу: 1.52 MB

ID файлу: 1015331351

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

22.6%

Схожість

Найбільша схожість: 7.05% з Інтернет-джерелом ([https://ela.kpi.ua/bitstream/123456789/26302/1/Pidruchnyk_Modeliuva](https://ela.kpi.ua/bitstream/123456789/26302/1/Pidруchnyk_Modeliuva))

22.4% Джерелаз Інтернету

813

Сторінка 85

2.32% Джерелаз Бібліотеки

66

Сторінка 93

2.01% Цитат

Цитати

3

Сторінка 94

Посилання

1

Сторінка 94

0% Вилучень

Немає вилучених джерел

Модифікації

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

Замінені символи

137

Підозріле форматування

13

сторінок

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 0.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 14%

ID: 118011 Назва: БКР Програмно-технічний пристрій розпізнавання типу електронних компонентів та вимірювання їх основних параметрів Додано в БД: 2023-06-24 Автора: А.А. Стойко Керівники: В.М. Стецюк Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	83852	749	2080 (2%)	26 (3%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Стойко Антон Андрійович

Тема: Програмно-технічний пристрій розпізнавання типу електронних компонентів та вимірювання їх основних параметрів

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 71

1. Короткий зміст роботи та прийнятих рішень: Метою роботи є розробка програмно-технічного пристрою розпізнавання типу електронних компонентів та вимірювання їх основних параметрів.

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 провів дослідження предметної області та постановка задачі. В другому розділі кваліфікаційної роботи виконано проведено аналіз стану розвитку цифрових вимірювальних пристроїв (тестерів мікросхем). В третьому розділі кваліфікаційної роботи проаналізована та реалізована програмна реалізація для розробки.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: висока складність контролю якості.

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 задовільному інженерно-технічному рівні.

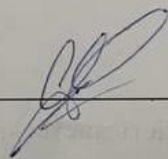
8. Інші зауваження: _____

9. Оцінка дипломної роботи: задовільно, 3.0

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

Степан Микола Валерійович доктер. філософії
старший викладач Київського

"26" травня 2023 р.

 (підпис)

Завідувачу кафедри КПС
д-р.техн.наук, проф. Говорушенко Т. О.

Стойко Антона Андрійовича

ІІІ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-19-2

ЗАЯВА

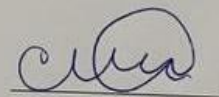
З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(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.

26.06.2023

дата



підпис

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

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

Назва: Програмно-технічний пристрій розпізнавання типу електронних компонентів та вимірювання їх основних параметрів

Автор: Стойко Антон Андрійович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Стецюк Василь Миколайович, ст. викладач

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:

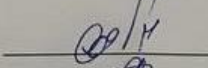
1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;

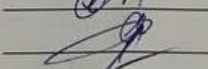
Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 22.6% і адресується до 879 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

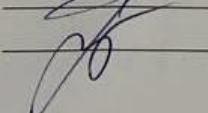
Керівник роботи

Гарант ОП

Завідувач кафедри КПС


В. М. Стецюк


С. М. Лисенко


Т. О. Говорушенко