

МІНІСТЕРСТВО ОСВІТИ І НАУКИ КРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КОНОТОПСЬКИЙ ІНСТИТУТ
Факультет денної форми навчання

кафедра електронних
приладів і автоматики

**Розробка алгоритму функціонування мікропроцесорної
системи контролю**

Студентк гр. ЕП-61к

С.С. Овчарук

Перевірив:

Ю.В.Столярчук

РЕФЕРАТ

Об'єктом дослідження кваліфікаційної роботи є розробка алгоритму функціонування мікропроцесорної системи контролю.

Мета роботи полягає у розробці алгоритму функціонування мікропроцесорної системи контролю. При виконанні роботи було проаналізовано види, модифікації, переваги та недоліки мікропроцесорних систем контролю та розроблено алгоритму функціонування мікропроцесорної системи контролю.

У результаті проведення досліджень встановлено, алгоритми функціонування мікропроцесорної системи контролю призначені для автоматизованого управління паровими і водогрійними котлами, які працюють на газі або рідкому паливі, а також котельнями, ЦТП, теплогенераторами, полум'яними печами та іншими технологічними об'єктами в різних галузях промисловості

Робота викладена на 37 сторінці, у тому числі включає 12 рисунків, 2 таблиць, список цитованої літератури із 23 джерел.

КЛЮЧОВІ СЛОВА: МІКРОПРОЦЕСОРНА СИСТЕМА, МІКРОПРОЦЕСОР, АЛГОРИТМ РОБОТИ, НАЛАГОДЖЕННЯ

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1 МІКРОПРОЦЕСОРНА СИСТЕМА, СТРУКТУРА ТА ПРОГРАМНО – ТЕХНІЧНІ КОМПЛЕКСИ	5
1.1 Мікропроцесорна система	5
1.2. Методи і засоби налагодження мікропроцесорних систем	8
РОЗДІЛ 2 АЛГОРИТМ РОБОТИ МІКРОПРОЦЕСОРНОЇ СИСТЕМИ.	13
2.1 Принцип та порядок роботи мікропроцесора	13
2.2 Алгоритм функціонування мікропроцесорної системи контролю на прикладі клавіатури	14
2.3 Мікропроцесорна система та побудова алгоритму роботи охоронної сигналізації автомобіля	15
РОЗДІЛ 3 ЗАСТОСУВАННЯ МІКРОПРОЦЕСОРНОЇ СИСТЕМИ КОНТРОЛЮ	30
3.1. Промислові контролери СПЕКОН	30
3.2 Контролер управління системами вентиляції БіКуб-ВК02.....	31
ВИСНОВКИ	34
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	35

ВСТУП

На сьогоднішній день приділяється велика увага проектуванню і практичному застосуванню мікропроцесорних систем контролю. При цьому розв'язуються завдання збільшення числа і складності виконуваних функцій, підвищення ролі обробки первинної інформації і її відображення, з максимальним ступенем автоматизації процесів.

Зростає роль використання математичного і програмного забезпечення у мікропроцесорних системах. Важливого значення набувають такі властивості в мікропроцесорних систем, як адаптація до впливу дестабілізуючими факторами, точність результатів, надійність роботи при зміні станів функціонування.

Застосування мікропроцесорних систем контролю і управління в сучасній апаратурі відкриває виняткові можливості для її розробників, це і гнучкість в адаптації до конкретних умов роботи і широкі сервісні можливості, можливість здійснення різноманітних режимів самодіагностики, реалізація так званих smart - функцій і т.д.

Завдяки впровадженню «мислячої» мікросхеми - мікроконтролера розробляються комплекси що досить легко наділити модними нині функціями fuzzy logic. таким чином, раніше розроблені пристрої відразу виходять на якісно новий рівень [1]

Застосування мікропроцесорних систем практично у всіх електричних пристроях - найважливіша риса технічної інфраструктури сучасного суспільства.

Електроенергетика, промисловість, транспорт, системи зв'язку істотно залежать від комп'ютерних систем управління. Мікропроцесорні системи вбудовуються в вимірювальні прилади, електричні апарати, освітлювальні установки.

РОЗДІЛ 1

МІКРОПРОЦЕСОРНА СИСТЕМА, СТРУКТУРА ТА ПРОГРАМНО – ТЕХНІЧНІ КОМПЛЕКСИ

1.1 Мікропроцесорна система

Цифрова інтегральна схема (ІС) називається мікропроцесором. Розробивши мікропроцесор розробники відкрили зовсім нову еру для електроніки та систем управління. Ця революція сталася через те, що мікропроцесор дозволяє використовувати гнучкість програмного управління та збільшити обчислювальну потужність комп'ютера для вирішення будь-якої проблеми.

Автоматичні програми управління особливо добре підходять для використання цієї технології, а мікропроцесорні системи управління швидко замінюють багато старих систем управління на основі аналогових ланцюгів або електромеханічних реле. Одним з перших мікропроцесорних контролерів, створених спеціально для програм управління, був програмований логічний контролер (PLC).

На рис.1.1. зображено структура мікропроцесорної системи.

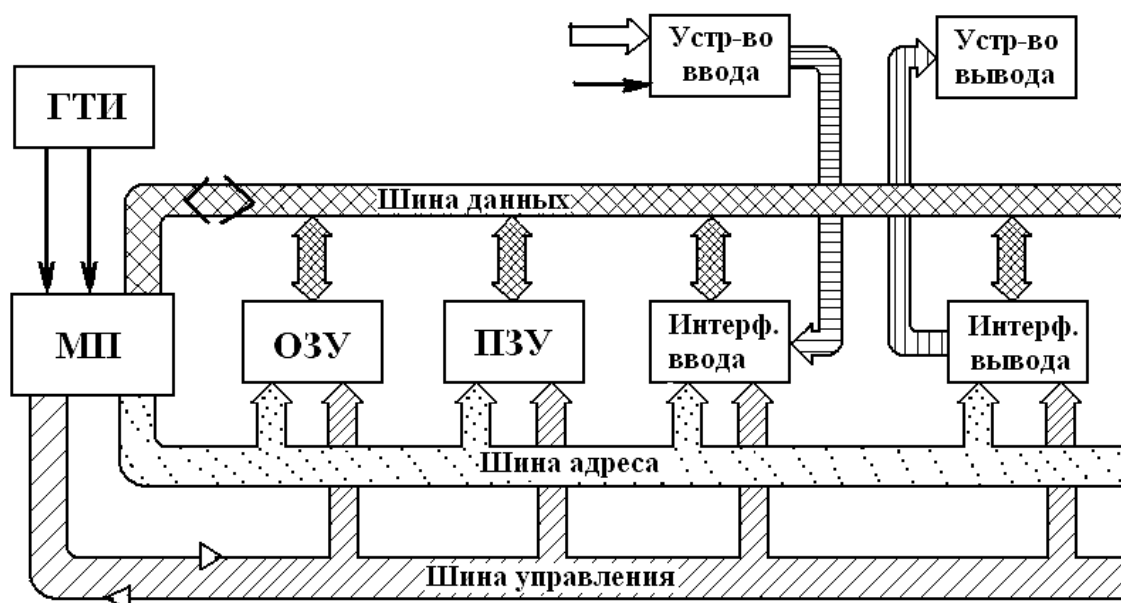


Рис.1.1. - Структура типової мікропроцесорної системи [2]

Мікропроцесор сам по собі не є комп'ютером. Для його роботи потрібні додаткові компоненти, такі як схеми пам'яті та введення / виведення. Однак мікроконтролер, який є близьким родичем мікропроцесора, містить усі комп'ютерні функції.

На рис.1.2. зображено зовнішній вигляд мікропроцесора.



Рис.1.2. - Зовнішній вигляд мікропроцесора intel pentium 4 [3]

Мікроконтролерам бракує деякої потужності та швидкості новіших мікропроцесорів, але їх компактність ідеальна для багатьох застосувань в управлінні, та в більшості так званих мікропроцесорних пристроїв.

Деякі конкретні причини використання цифрової мікропроцесорної конструкції в системах управління:

- сигнали низького рівня від датчиків, перетворені в цифрові. може передаватися на великі відстані практично без помилок;
- мікропроцесор може легко обробляти складні обчислення та стратегії управління;
- доступна довготривала пам'ять для відстеження параметрів у повільних системах;
- змінити стратегію управління легко, завантаживши нову програму, ніяких змін обладнання не потрібно;
- мікропроцесорні контролери легше підключаються до комп'ютерної мережі всередині організації.

Це дозволяє дизайнерам вводити зміни в програму і читати курси орендувати статус системи з їхніх терміналів.

Основні поняття системи на основі мікропроцесорів та мікроконтролерів, з особливим акцентом на застосуванні систем управління. Це аж ніяк не глибока обробка, але достатня, щоб зробити решту тексту більш змістовною. У перших розділах цієї глави вводяться основні поняття мікропроцесорного обладнання та функціонування (ці поняття також стосуються мікроконтролерів).

1.2 Методи і засоби налагодження мікропроцесорних систем

В робочому циклі мікропроцесорної системи виділяються три основних етапи:

- проектування;
- виготовлення;
- експлуатація.

Для підтримки мікропроцесорної системи в працездатному стані необхідно враховувати як специфіку кожного етапу її життєвого циклу, так і особливості МПС як об'єкта контролю.

Перш за все, розглянемо деякі основні поняття, пов'язані з контролем і діагностикою цифрових обчислювальних пристроїв.

Процедура перевірки правильності функціонування об'єкта називається контролем або тестуванням. В результаті контролю визначається, чи знаходиться об'єкт в справному стані. Якщо об'єкт знаходиться в несправному стані, то виникає задача пошуку несправності - діагностування. Процес усунення знайдених несправностей називається налагодженням.

Розглянемо в першу чергу особливості МП БІС і мікропроцесорної системи в цілому як об'єкта контролю, які ускладнюють процедуру контролю та пошуку несправностей.

Повна перевірка схеми повинна забезпечувати перевірку її працездатності при всіх можливих станах.

Число можливих станів визначається кількістю внутрішніх елементів пам'яті. Взявши для прикладу тільки реєстрову пам'ять однокристального мікроконтролера (32 регістра по 8 розрядів), отримуємо 2256 станів.

Для повного їх перебору з частотою 100 МГц буде потрібно час порядку 1067 років! Очевидно, що такий варіант неприйнятний і ніколи на практиці не реалізується. Отже, будь-яка МП БІС може містити несправності. Найнеприємніше в цій ситуації те, що ці несправності носять характер збоїв, тобто проявляються тільки при певному поєднанні даних всередині БІС, і тому дуже важко піддаються виявленню.

Найвідомішою помилкою цього роду є, безумовно, помилка в процесорі обробки чисел з плаваючою точкою мікропроцесора Pentium, випущеного в 1993 році. Вона проявлялася в неточному виконанні операції ділення (похибка в 8-м десятковому знаку після коми) для деяких операндів.

Помилка виявлялася, тільки якщо дільник містив шість послідовних біт, з 5-го по 10-й, установлених в одиницю. Вона була викликана дефектами в мікрокоді реалізації алгоритму розподілу. Ймовірність виявлення помилки оцінювалася як один до дев'яти мільйонам, а ймовірність зустрічі з нею - 1 раз в 27 000 років. Однак незважаючи на це, фірма Intel пішла на те, щоб замінити на вимогу всі дефектні процесори, зазнавши при цьому, природно, значні збитки.

Аналогічна помилка була виявлена в кінці 2007 року в чотириядерних серверних процесорах Barcelona, а також в чіпах Phenom для настільних комп'ютерів компанії AMD. Вона була пов'язана з особливостями реалізації буфера швидкого перетворення адреси (TLB) кеш-пам'яті третього рівня. Помилка в окремих ситуаціях приводила до зависання системи.

Іноді окремі несправності виявляють себе тільки на високих частотах роботи БІС. Тому деякі виробники схем, визначивши в ході тестування, що окремі БІС з деякою партії схем дають збої на початковій, найвищою частоті, перевіряють її працездатність на більш низьких частотах, де схеми цієї партії

можуть виявитися цілком працездатними. Саме із зазначенням такої частоти роботи ці БІС і надходять на ринок. Такий підхід є однією з основ "розгону" схем.

Мала кількість контрольних точок схем. Незважаючи на те що корпуса сучасних мікропроцесорних БІС мають до 1,5 тис. Висновків, складність мікропроцесорів така, що безпосередньо подати тестове вплив на будь-яку точку схеми неможливо.

Також неможливо безпосередньо перевірити стан будь-якого вузла БІС. Це вимагає подачі складних послідовностей сигналів на доступні для впливу точки схеми і синхронного аналізу довгих послідовностей станів в точках, доступних для прямого контролю. Тому впливу і представлення результатів носять непрямий, опосередкований характер, що може привести до невірної інтерпретації одержуваних при тестуванні даних.

Неподільність апаратури і програмного забезпечення. Апаратура і програмне забезпечення МПС є єдиним комплексом. Свої функції мікропроцесорна система виконує під впливом програм, частина з яких до того ж зберігається в постійній пам'яті БІС і системи. Помилки в роботі системи можуть бути обумовлені як несправностями апаратної частини, так і помилками, допущеними при розробці ПЗ.

Складність і неподільність апаратури, що становить мікропроцесорну систему. МПС, особливо на етапі її експлуатації, неможливо розділити на функціональні вузли для їх автономної перевірки. Це пояснюється конструктивними і схемотехнічними особливостями мікропроцесорних засобів.

По-перше, чисто конструктивно підключити тестуючу апаратуру до окремих БІС, що становить систему, вельми складно. По-друге, поділ мікропроцесорної системи на окремі функціональні вузли при сучасному рівні інтеграції елементів на кристалі неможливо. сам мікропроцмістить широкий набір функціонально різних пристроїв від цілочисельного АЛП до запам'ятовуючих пристроїв різного виду і призначення.

У той же час одні і ті ж функції розподіляються по різним БІС. Наприклад, обробкою переривань займається як сам мікропроцесор, так і контролер пріоритетних переривань, що входить до складу схем чіпсета, які, в свою чергу, виконують безліч інших функцій.

Тому перевірка виконання тієї чи іншої функції в мікропроцесорній системі зачіпає роботу всіх складових її схем і пред'являє високі вимоги до кваліфікації обслуговується персоналу. Необхідність одночасного контролю стану великого числа багаторозрядних шин.

Загальна розрядність шин, за якими відбувається взаємодія між компонентами мікропроцесорної системи, досить велика. Але головну проблему становить не тільки кількість точок, в яких потрібно аналізувати сигнали. Проста реєстрація їх поведінки мало що дасть для визначення місця і характеру несправності. [4]

Зазвичай потрібна реєстрація і аналіз сигналів протягом значного часового інтервалу і в зв'язку з рідкісними і однократними подіями. До числа таких подій, ідентифікація яких необхідна в процесі контролю, відноситься поява заданих комбінацій сигналів на шинах МПС і навіть поява заданій послідовності таких комбінацій: звернення за певною адресою на запис певних даних, поява сигналу запиту переривання певного типу або навіть сигнал перешкоди.

При настанні такої ситуації слід проаналізувати, а отже, попередньо запам'ятати стан шин протягом як декількох попередніх тактів, так і декількох тактів, які слідують за вказаними подією. У той же час для МПС на основі однокристальних мікроконтролерів і ЦСП ця проблема не настільки гостра, так як реалізація більшості функцій системи здійснюється внутрішніми засобами цих мікропроцесорів.

Тому такі системи мають мале число периферійних схем, а обмін між ними і мікропроцесором часто виконується по послідовним інтерфейсів. Висока швидкодія схем, що утворюють МПС, яке вимагає надзвичайно швидкодіючої контрольної апаратури. Частоти, на яких працюють сучасні БІС, доходять до

декількох гігагерц. У той же час щоб визначити, наприклад, факт перемикання будь-якого сигналу, частота контролюючої апаратури повинна бути як мінімум в два рази більше.

Шинна організація мікропроцесорної системи. Магістрально-модульний принцип побудови МПС ускладнює визначення джерела спотворення інформації в магістралі. Неправильне зчитування інформації з ОЗУ в мікропроцесор може бути викликано не поломкою будь-якого з цих двох пристроїв, а несправністю абсолютно сторонньої для даного процесу схеми, при якій її вихід постійно заземляє одну з ліній загальної магістралі.

Поряд з наявними труднощами в процесі контролю і налагодження мікропроцесорна система має ряд переваг, що полегшують цей процес.

Здатність до самоконтролю, тобто здатність МП під дією програми порівняно невеликого обсягу генерувати дуже великі тестові послідовності. У цьому випадку використовується метод розкрутки. [5]

Спочатку налагоджували синхронізація МПС і створюється деякий ядро з мікропроцесора і пам'яті невеликої ємності. На згадку записується програма, яка під управлінням мікропроцесора генерує тести для контролю всього ОЗУ, а потім, у міру підключення нових блоків пам'яті, - і для інших вузлів мікропроцесорної системи.

Здатність мікропроцесора до логічної обробки інформації, що дозволяє порівнювати відгуки перевіряються вузлів з еталонними без застосування додаткової апаратури.

При цьому значно зменшується обсяг інформації, яку повинна зафіксувати контрольноїспитательная апаратура, а весь аналіз одержуваної інформації (наприклад, порівняння одержуваних реакцій з еталонними) виконує сам МП, який може не тільки визначити характер несправності, але іноді і точно вказати її місце.

Програмна доступність всіх вузлів МПС дозволяє мікропроцесору видавати на них тестові впливи, отримувати і аналізувати реакції на ці дії.

При складанні тестових і діагностичних програм розробник має справу не з новою мовою програмування контрольної апаратури, а зі звичними засобами програмування мікропроцесора.

Важливою особливістю МПС, як і інших цифрових систем, є стандартне уявлення електричних сигналів. Це дозволяє спростити контроль стану точок і звести його лише до визначення приналежності даного стану до зони сигналу 0 або 1. Вдаватися до вимірювань аналогових величин - тривалості фронту імпульсу, амплітуди сигналу і т. П. - доводиться тільки при підозрі на вихід з ладу одного з електронних компонентів схеми.

Спрощена внутрішня архітектура типового 8-розрядного мікропроцесора показана на рис.1. 3.

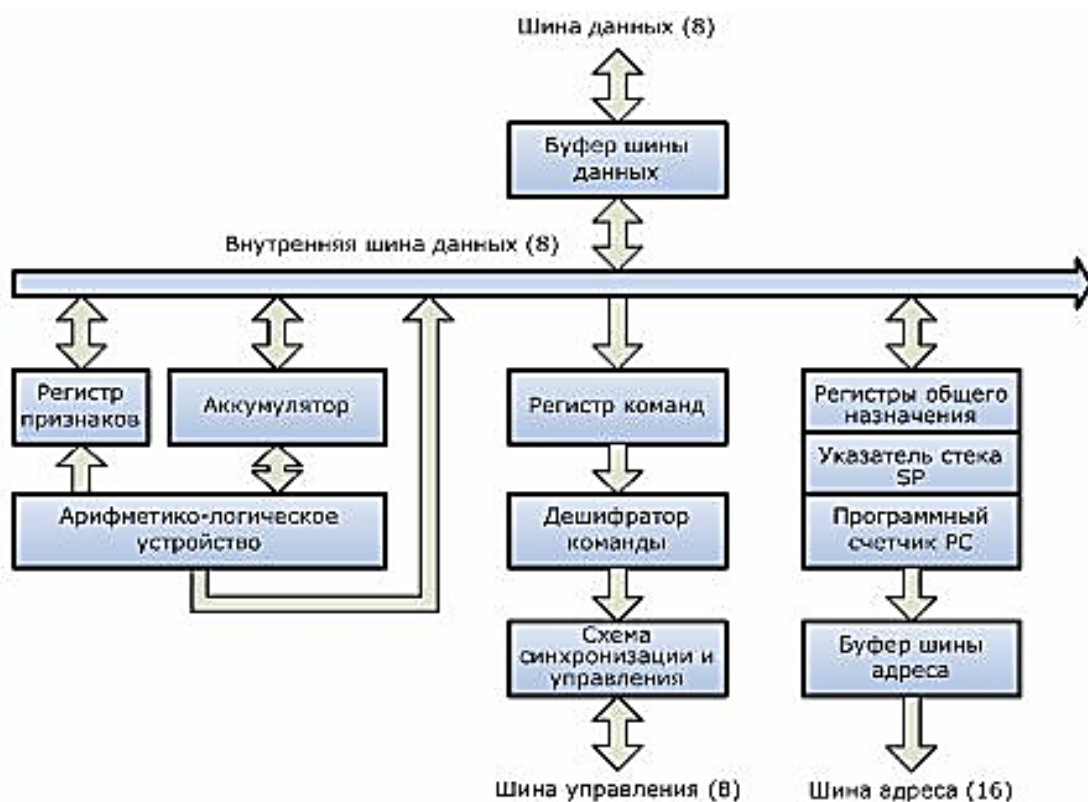


Рис.1.3. - Архітектура типового 8-розрядного мікропроцесора [6]

РОЗДІЛ 2

АЛГОРИТМ РОБОТИ МІКРОПРОЦЕСОРНОЇ СИСТЕМИ

2.1 Принцип та порядок роботи мікропроцесора

Алгоритм - точне розпорядження, однозначно задає процес перетворення вихідної інформації в послідовність операцій, що дозволяють вирішувати сукупність завдань певного класу і отримувати шуканий результат.

Головним керуючим елементом всієї мікропроцесорної системи є процесор. Саме він, за винятком кількох особливих випадків, управляє всіма іншими пристроями. Решта ж пристроїв, такі, як ОЗУ, ПЗУ і порти введення / виводу є веденими. [7]

Відразу після включення процесор починає читати цифрові коди з тієї області пам'яті, яка відведена для зберігання програм. Читання відбувається послідовно осередок за осередком, починаючи з найпершої. В осередку записані дані, адреси і команди. Команда - це одне з елементарних дій, яке здатний виконати мікропроцесор. Вся робота мікропроцесора зводиться до послідовного читання і виконання команд.

Послідовність дій мікропроцесор під час виконання команд програми:

- перед виконанням чергової команди мікропроцесор містить її адресу в програмному лічильнику РС;
- МП звертається до пам'яті за адресою, що міститься в РС, і зчитує з пам'яті перший байт чергової команди в регістр команд;
- дешифратор команд декодує (розшифровує) код команди;
- відповідно до отриманої від дешифратора інформацією пристрій управління виробляє впорядковану в часі послідовність мікрооперацій, що реалізують приписи команди, в тому числі, витягує операнди з регістрів і пам'яті, виконує над ними запропоновані кодом команди арифметичні, логічні або інші операції;

- в залежності від довжини команди модифікує вміст РС;
- передає управління чергової команді, адреса якої знову знаходиться в програмному лічильнику РС.

Сукупність команд мікропроцесора можна розділити на три групи:

- команди переміщення даних, переміщення відбувається між пам'яттю, процесором, портами введення / виведення (кожен порт має свою власну адресу), між регістрами процесора;
- команди перетворення даних;
- команда передачі управління

Дуже рідко програма складається з однієї послідовної команд. Переважна кількість алгоритмів вимагають розгалуження програми. Для того, щоб програма мала можливість змінювати алгоритм своєї роботи в залежності від будь-якої умови, і служать команди передачі управління. Дані команди забезпечують виконання програми за різними шляхами і організують цикли.

Зовнішні пристрої підключаються до мікропроцесорної системі фізично - за допомогою роз'ємів, і логічно - за допомогою портів (контролерів). Для взаємодії процесора та зовнішніх пристроїв застосовується система (механізм) переривань. Це спеціальний механізм, який дозволяє в будь-який момент, по зовнішньому сигналу змусити процесор призупинити виконання основної програми, виконати операції, пов'язані із зухвалим переривання подією, а потім повернутися до виконання основної програми. У будь-якого мікропроцесора є хоча б один вхід запиту на переривання INT (від слова Interrupt - переривання).

2.2 Алгоритм функціонування мікропроцесорної системи контролю на прикладі клавіатури

Клавіатура - пристрій для введення символної інформації і команд управління. Для підключення клавіатури в комп'ютері є спеціальний порт клавіатури (мікросхема). На рисунку 2.1 зображено взаємодії клавіатури та мікропроцесора.

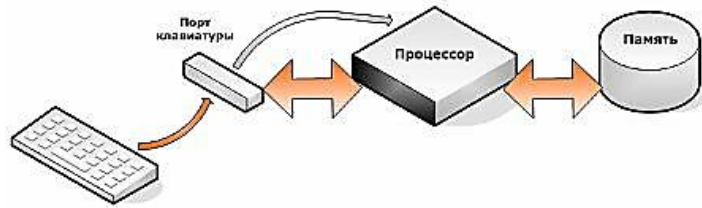


Рис.2.1.- Робота мікропроцесора з клавіатурою [8]

Алгоритм роботи:

- при натисканні клавіші контролер клавіатури формує цифровий код, цей сигнал надходить в мікросхему порту клавіатури;
- порт клавіатури посилає процесору сигнал переривання. Кожне зовнішнє пристрій має свій номер переривання, по якому процесор його і розпізнає;
- отримавши переривання від клавіатури, процесор перериває виконання програми (наприклад, редактор Microsoft Office Word) і завантажує з пам'яті програму обробки кодів з клавіатури. Така програма називає драйвер;
- ця програма направляє процесор до порту клавіатури, і цифровий код завантажується в реєстр процесора;
- цифровий код зберігається в пам'яті, і процесор переходить до виконання іншої задачі;
- завдяки високій швидкості роботи, процесор виконує одночасно велику кількість процесів.

Більш детально розглянемо далі побудову алгоритмів на прикладі охоронної системи автомобіля.

2.3 Мікропроцесорна система та побудова алгоритму роботи охоронної сигналізації автомобіля

Одним з етапів отримання тексту вихідної програми по якій буде працювати охоронна сигналізація є розробка загальної схеми алгоритму (СА) роботи мікроконтролера. Мовою алгоритмів треба описати метод, обраний для вирішення поставленого завдання. Спосіб вирішення завдання, обраний на

етапі її інженерної інтерпретації, на основі якого формується СА, визначає не тільки якість розробленої прикладної програми, а й якісні показники кінцевого виробу. В основу розробки СА покладена та ж сама процедура модульного проектування, яка традиційно використовується розробниками апаратних засобів.

СА роботи складається з наступних елементів: "Початок", "Ініціалізація", "Перевірка слова стану (СС)". Залежно від слова стану (СС) програма переходить в один з трьох режимів роботи. Так при $CC = 1$ активізується режим «Охорона», при $CC = 2$ - режим «Тривога», при $CC = 3$ - режим «Очікування». Схема алгоритму роботи представлена на рис.2.2.

Ініціалізація полягає у виборі режимів роботи окремих елементів структури РІС-контролера і в установці необхідних регістрів.

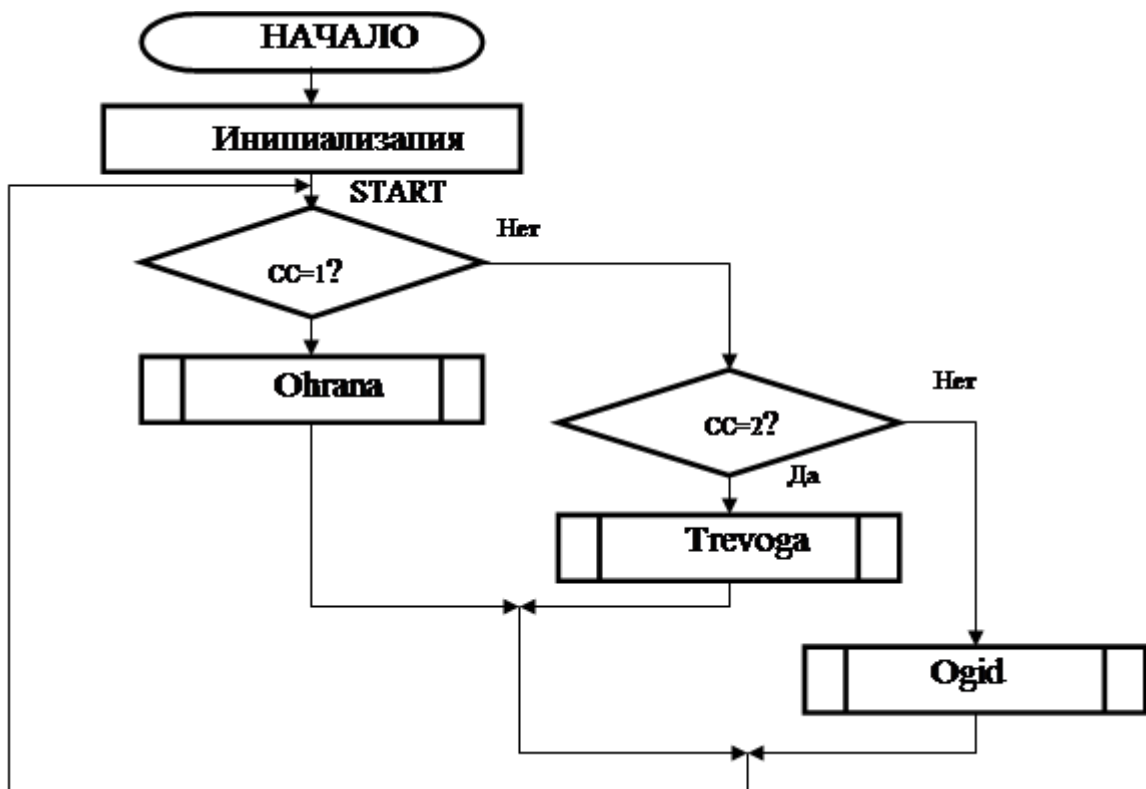


Рис.2.2 - Схема алгоритму роботи охоронної системи автомобіля [9]

Розглянемо докладніше алгоритми роботи системи в кожному з трьох режимів. Схема алгоритму підпрограми режиму «Охорона» представлена на рис.2.3

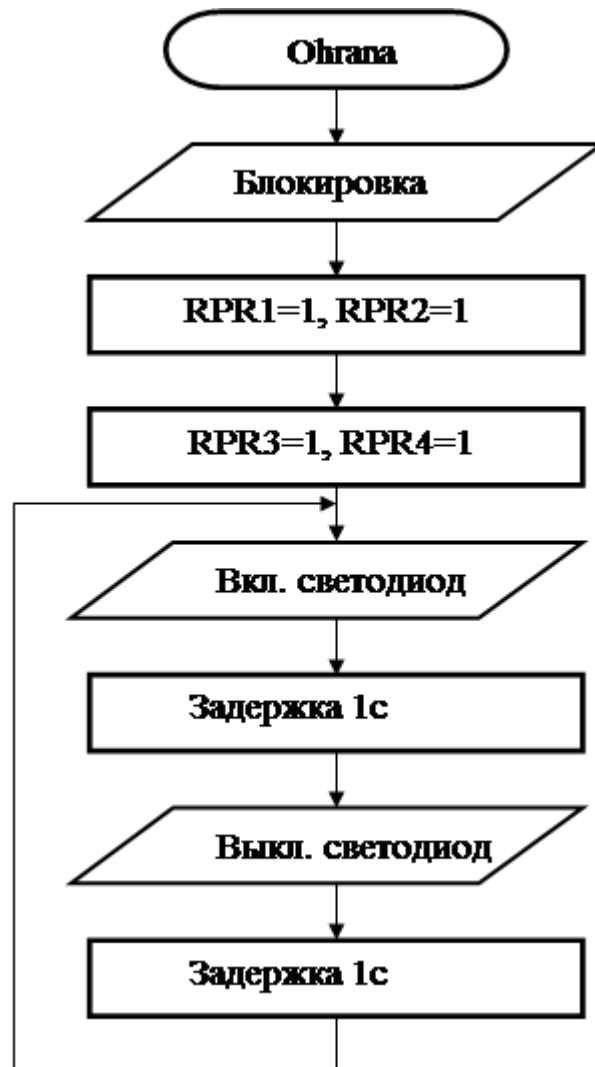


Рис.2.3 - Схема алгоритму підпрограми режиму «Охорона» [9]

Для зручності опису введемо такі логічні змінні RPR1, RPR2, RPR3, RPR4, RPRTMR1. Вони будуть показувати дозволені або заборонені чи переривання від датчиків дверей, УЛЗ датчика, датчика ударів, детектора правильного коду і від таймера відповідно. При наявності будь-якого з дозволених переривань, програма переходить до підпрограми визначення джерела і пріоритету переривань RZINT, структурна схема алгоритму якої показана на рис.2.4.

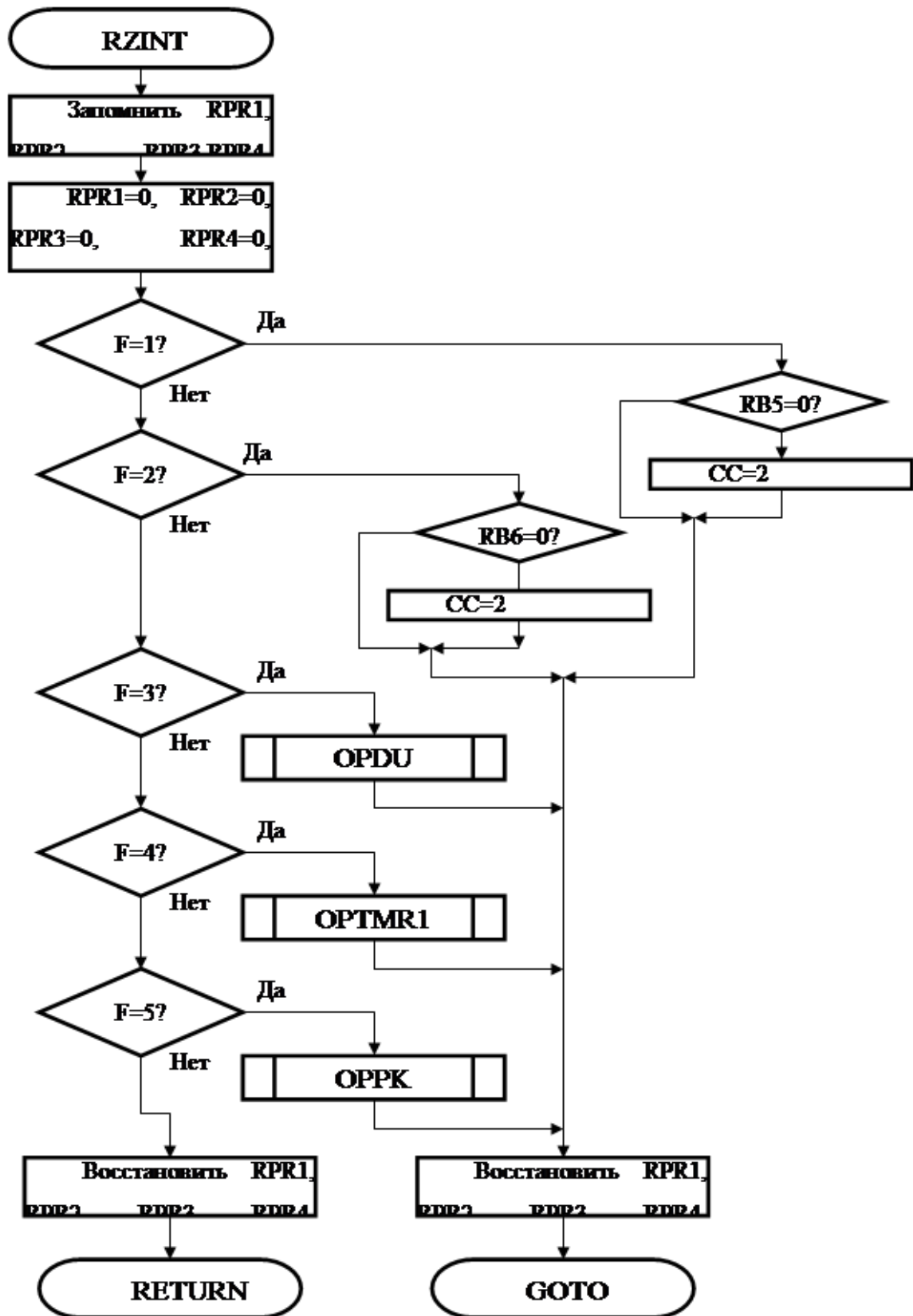


Рис.2.4.- Схема алгоритму RZINT [9]

Якщо вони будуть рівні «1», то переривання дозволені, якщо «0», то переривання заборонені. Спочатку режиму «Охорона» блокується запалювання

видачею на порт RA1 логічної одиниці. Далі вирішуються переривання від всіх датчиків і від детектора правильного коду і програма переходить в режим очікування переривань, видаючи з періодом 1с сигнали включення і виключення світлодіода.

Вводиться логічний прапорець F, який буде вказувати джерело переривання:

- F = 1, то переривання викликано зміною стану датчика відкриття дверей;
- F = 2, то зміною стану УЛЗ датчика;
- F = 3, то зміною стану датчика ударів;
- F = 4, то джерело преривання- таймер;
- F = 5, то джерело - детектор правильного коду.

Починається підпрограма RZINT з запам'ятовування змінних RPR1, RPR2, RPR3, RPR4, RPRTMR1. Потім забороняються всі переривання. Далі слід здійснити перевірку джерел переривання. Черговість перевірок визначає пріоритетність переривань. В такому випадку вона наступна: від датчика відкриття дверей, від УЛЗ датчика, від датчика ударів, від таймера і нарешті від детектора правильного коду [10]

Обробка переривання від датчика відкриття дверей відбувається наступним чином. Перевіряється стан порту RB5, до якого підключені відповідні датчики. Якщо на порті присутній логічний нуль, то приймається рішення про наявність посягання на автомобіль, при цьому слову стану присвоюється значення 2, що відповідає режиму «Тривога» і після відновлення змінних RPR1, RPR2, RPR3, RPR4, RPRTMR1 управління передається головній програмі і до мітки START .

Обробка переривання від УЛЗ датчика відбувається за таким же алгоритмом.

При виникненні переривання від датчика ударів видається короткий звуковий сигнал сиреною. Потім лічильник ударів SU зменшується на одиницю і проводиться перевірка: чи рівний лічильник ударів нулю. Спочатку лічильник

ударів SU дорівнює 2. Якщо SU НЕ дорівнює 0, то запускається підпрограма затримки на 3 секунди. Після цього SU знову присвоюється значення 2 і відбувається вихід з підпрограми. Якщо за ці 3 секунди знову відбулося переривання, то лічильник ударів буде дорівнювати нулю після чергового зменшення на одиницю. Це призведе до того, що система інтерпретує ці два спрацьовування датчика ударів протягом 5 секунд як посягання, слову стану присвоїти значення 2, і відбудеться вихід до мітки START. Схема алгоритму підпрограми обробки переривання від датчика ударів OPDU приведена на рис.2.5.

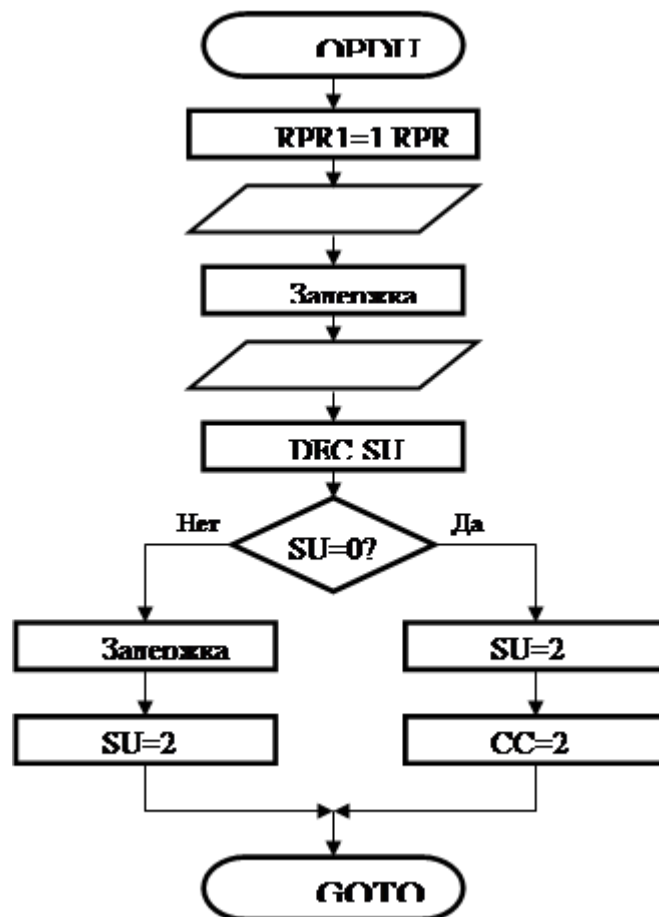


Рис.2.5. - Схема алгоритму обробки переривання від датчика ударів [11]

При появі напруги високого рівня на порт RB0, відбувається переривання і керування передається підпрограмі обробки переривання від детектора правильного коду ОРПК, схема алгоритму якій зображена на рис.2.6.

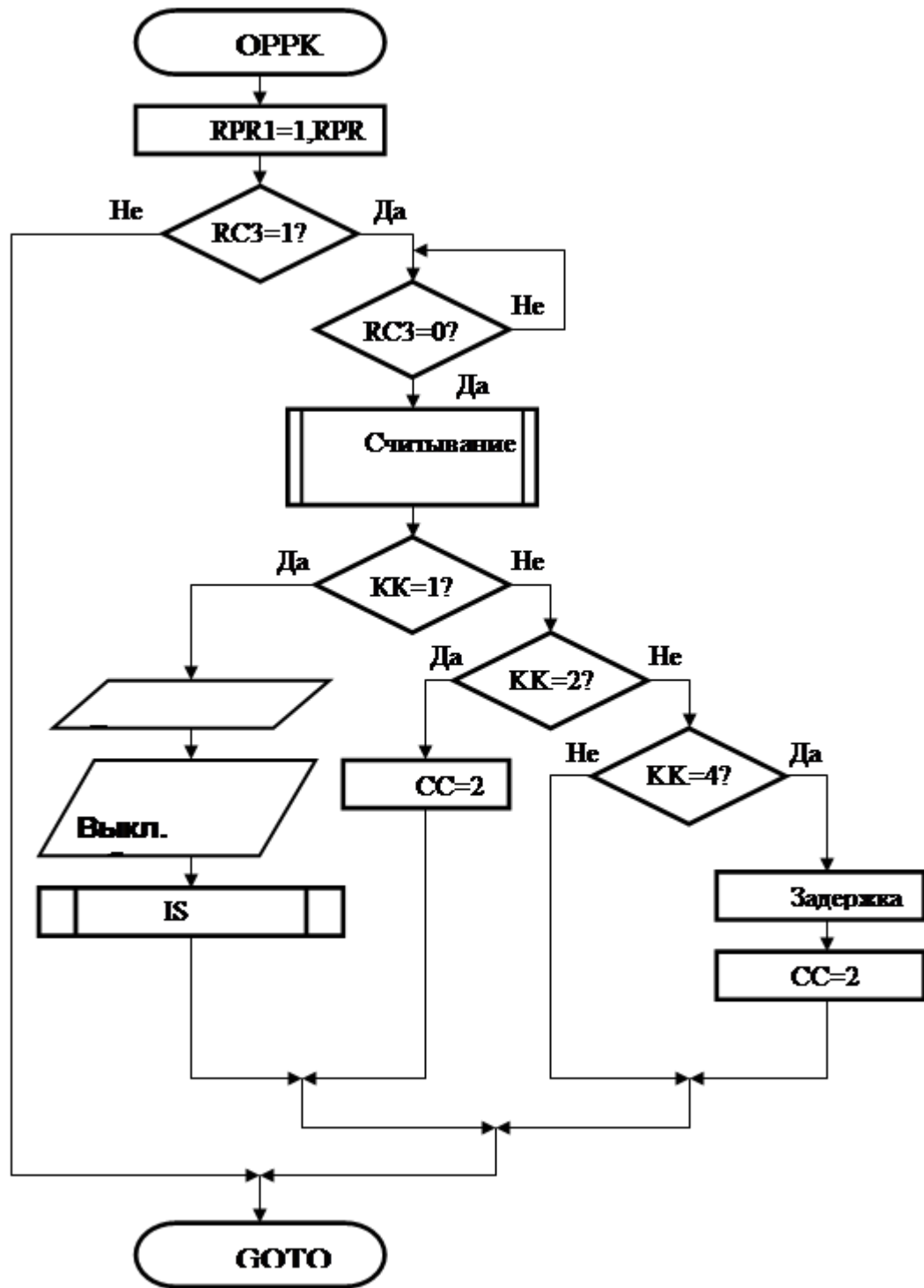


Рис.2.6. - Схема алгоритму обробки переривання від детектора правильного коду [11]

Підпрограма працює наступним чином. Після прийому правильної кодової послідовності, дешифратор посилає на порт RC3 протягом 500 мс логічну одиницю. Починається ОРПК з дозволу переривань від датчиків дверей і від УЛЗ датчика, потім проводиться перевірка стану порту RC3. Якщо на ньому

присутня напруга низького рівня, то відбувається вихід з підпрограми, якщо високого рівня - то приймається рішення про прийом дешифратором правильної кодової послідовності. Після цього відбувається очікування закінчення передачі дешифратором «одиниці» і можна перейти до зчитування коду клавіші. Код клавіші поміщається в регістр з ім'ям КК. Далі проводиться ідентифікація коду клавіші.

Якщо $КК = 1$, то вважається, що прийшов код від клавіші «Постановка / зняття». При цьому відбувається вимикання сирени і миготіння габаритних вогнів і перехід до підпрограми перевірки і зміни стану IS. Якщо $КК = 2$, то вважається, що прийшов код від клавіші «Паніка». При цьому слову стану присвоюється значення 2, яке відповідає режиму «Тривога» і відбувається вихід з підпрограми.

Якщо $КК = 4$, то вважається, що прийшов код від клавіші «АНІ». При цьому запускається підпрограма затримки на 20 секунд, потім слову стану присвоюється значення 2 і відбувається вихід з підпрограми до мітки START.

Підпрограма IS починається з перевірки слова стану. Якщо $СС = 3$, то це означає, що система перебувала в режимі «Очікування» і проводиться видача сигналу на закриття центрального замку (ЦЗ), подача двох коротких сигналів сиреною, відключення сигналу закриття ЦЗ, привласнення $СС = 1$ і вихід з підпрограми. Затримки під час видачі звукових сигналів досить для надійного замикання ЦЗ.

Якщо $СС = 2$, то це означає, що система перебувала в режимі «Тривога» і видавши короткий звуковий сигнал слову стану присвоюється 1 і відбувається вихід з підпрограми.

В іншому випадку ($СС$ не дорівнює ні 3, ні 1), це означає, що система перебувала в режимі «Охорона» і проводиться включення сигналу на відкриття ЦЗ, видача короткого звукового сигналу, запускається підпрограма затримки на 1 с, вимикається сигнал на відкриття ЦЗ, слову стану присвоюється значення 3 і відбувається вихід з підпрограми. Схема алгоритму підпрограми IS зображена на рис.2.7

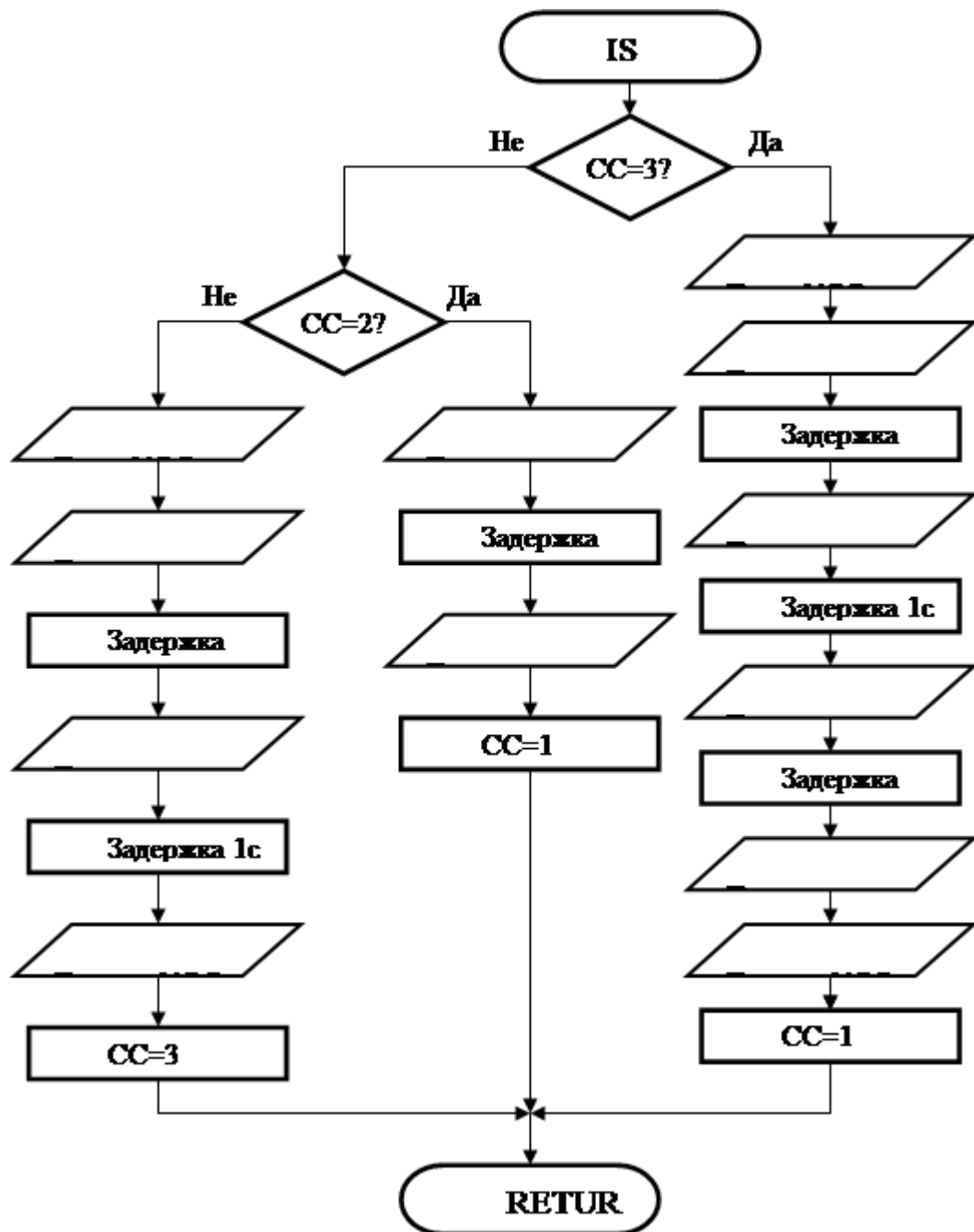


Рис.2.7. - Схема алгоритму підпрограми IS [11]

Підпрограма режиму «Тривога» починається з видачі сигналу на блокування запалювання. Далі включається сирена, підпрограма миготіння габаритних вогнів, забороняються переривання від датчиків, встановлюється лічильник передач SPER = 5, забороняються переривання від детектора правильного коду, встановлюється лічильник посилок SPOS = 40, включається передавач і видається на нього повідомлення через асинхронний передавач.

Схема алгоритму підпрограми режиму «Тривога» приведена на рис.2.8

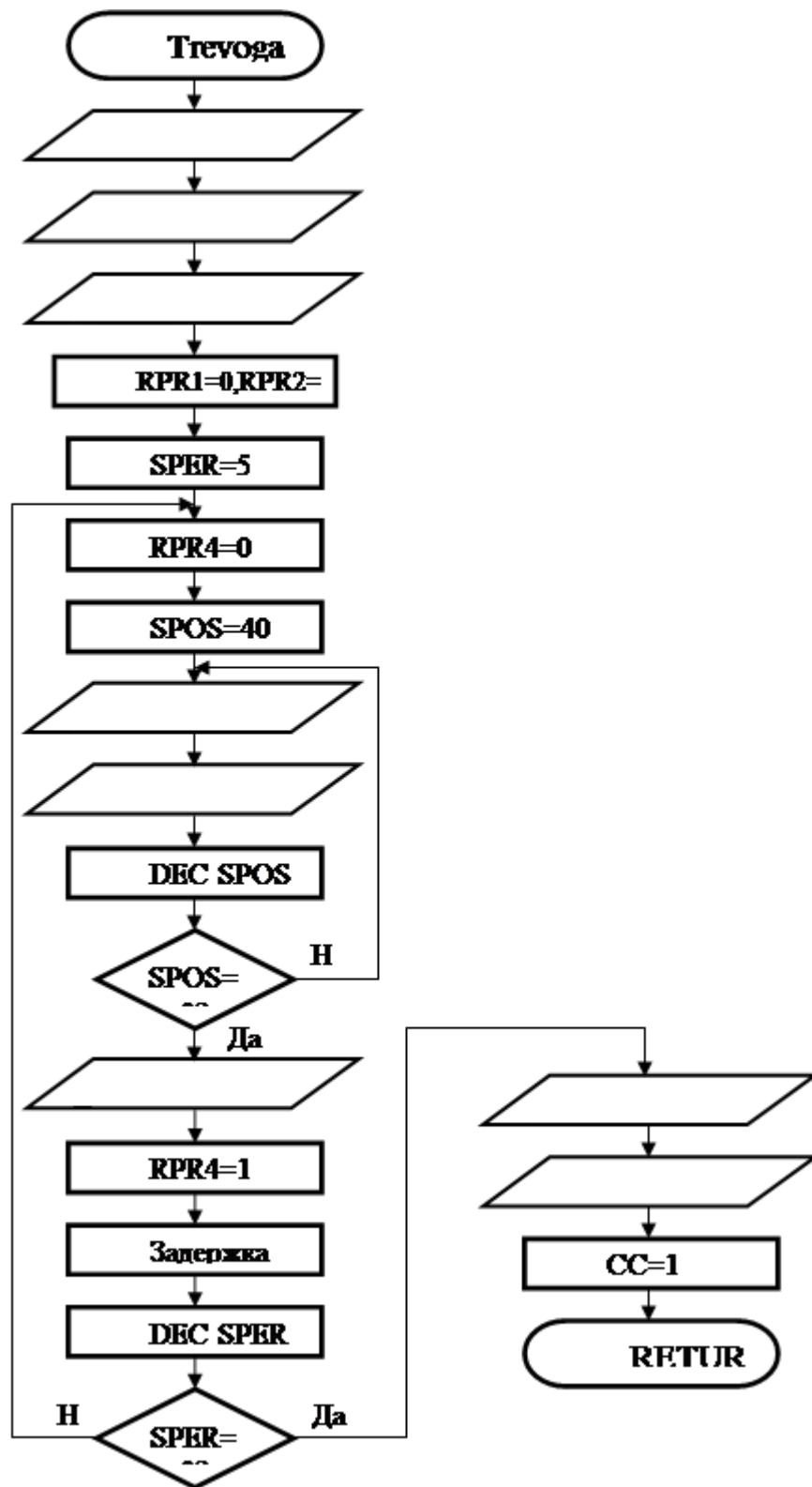


Рис.2.8. - Схем алгоритму підпрограми режиму «Тревога» [11]

Потім здійснюється зменшення лічильника посилок і перевірка його на нуль. Якщо SPOS НЕ дорівнює 0, то останні три дії повторюються, якщо дорівнює, то передавач вимикається, дозволяється переривання від детектора

правильного коду, включається підпрограма затримки на 10 с, зменшується на одиницю SPER і перевіряється на рівність нулю. Якщо SPER не дорівнює нулю, то знову забороняється переривання від детектора правильного коду і повторюються перераховані вище дії. Якщо $SPEP = 0$, то сирена і габаритні вогні вимикаються, слову стану присвоюється значення 1 і проводиться вихід з підпрограми. Загальний час роботи системи в режимі «Тривога» становить приблизно 1 хв.

Схема алгоритму підпрограми режиму «Очікування» представлена на рис.2.9. Спочатку забороняються переривання від всіх датчиків і детектора правильного коду. Завантажується лічильник спроб SPOP числом 3 і передається керування підпрограмі введення коду з клавіатури і його перевірки INKEY. Результатом роботи підпрограми INKEY є установка прапорця вірності KVER:

- $KVER = 0$, то код буде вважатися неправильним;
- $KVER = 1$ - то вірним;
- $KVER = 0$, то здійснюється зменшення лічильника спроб.

Поки SPOP не дорівнює нулю, то повторюється процедура INKEY з мітки MET2. Коли SPOP = 0, то завантажується лічильник сигналів SSIG числом 3. Далі включається на 0,5 с сирена, зменшується лічильник SSIG і це повторюється поки SSIG не стане рівним нулю. Після цього слова стану присвоюється значення 1 і відбувається вихід з підпрограми.

Якщо $KVER = 1$, то знімається блокування запалювання, дозволяється переривання від детектора правильного коду і в нескінченному циклі запускається підпрограма роботи з клавіатурою RUNKEY.

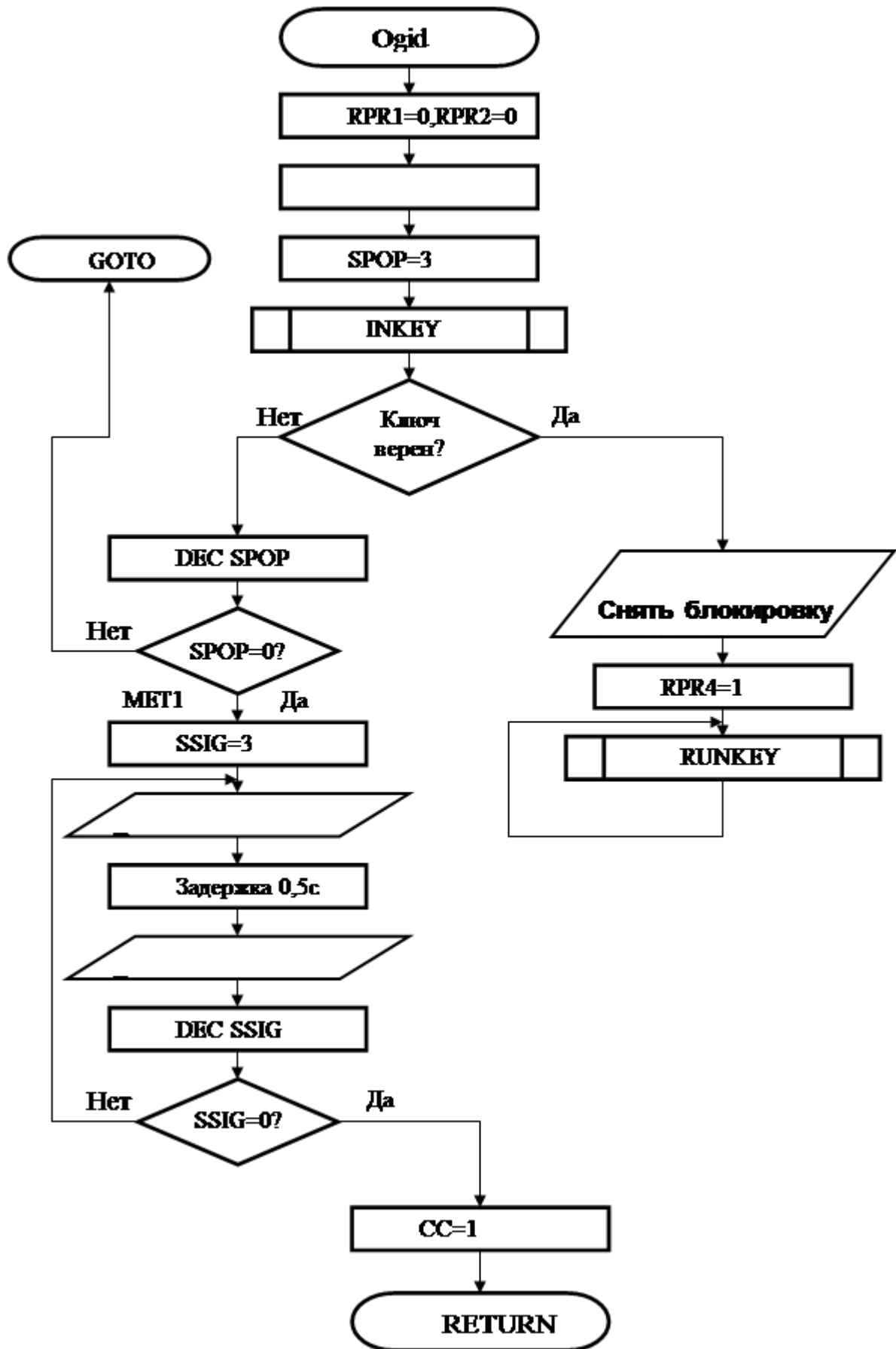


Рис.2.9. - Схема алгоритму підпрограми режиму «Очікування» [12]

Схема алгоритму процедури INKEY приведена на рис.2.10. В осередках пам'яті даних з символічними іменами NK0, NK1, NK2 і NK3 буде зберігатися 4 розряду ключа доступу. В осередках пам'яті даних з символічними іменами VK0, VK1, VK2 і VK3 буде зберігатися 4 розряду вводиться ключа доступу. Регістр з символічним ім'ям BUK буде буфером клавіатури і в ньому буде зберігатися набране на клавіатурі число.

Процедура INKEY починається з завантаження лічильника скидів SSBR числом 114. Потім дозволяється переривання від таймера і запускається сам таймер з попереднім діленням на 8. За допомогою нього задається час, відведений на набір коду. Воно дорівнює приблизно 1 хвилину. Далі перевіряються осередки пам'яті NK0, NK1, NK2 і NK3.

Якщо у всіх них містяться нульові значення (відразу після ініціалізації), то підпрограма переходить до початкового введення ключа. Робиться це так. Проводиться опитування клавіатури і очікується натискання клавіші з кодом 11 «Е». Як тільки це сталося, видається сигнал на включення світлодіода, завантажуються лічильник натискань клавіш SN числом 4. Після здійснюється введення чотирьох перших кодів і послідовний занос їх в осередки пам'яті NK0, NK1, NK2 і NK3. Після чотирьох натискань видається сигнал на вимикання світлодіода, встановлюється прапорець KVER = 1 і проводиться вихід з підпрограми.

Якщо в осередках пам'яті NK0, NK1, NK2 і NK3 міститься ненульовий ключ, то підпрограма переходить до введення ключа і його перевірки. Робиться це так.

Проводиться опитування клавіатури і очікується натискання клавіші з кодом 10 «Т». Як тільки це сталося, видається сигнал на включення світлодіода і проводиться послідовне введення ключа в осередку пам'яті VK0, VK1, VK2 і VK3, до натискання клавіші з кодом 11 «Е».

Після натискання цієї клавіші натискань. Видається сигнал на вимикання світлодіода, запускається процедура перевірки PROV і відбувається вихід з підпрограми.

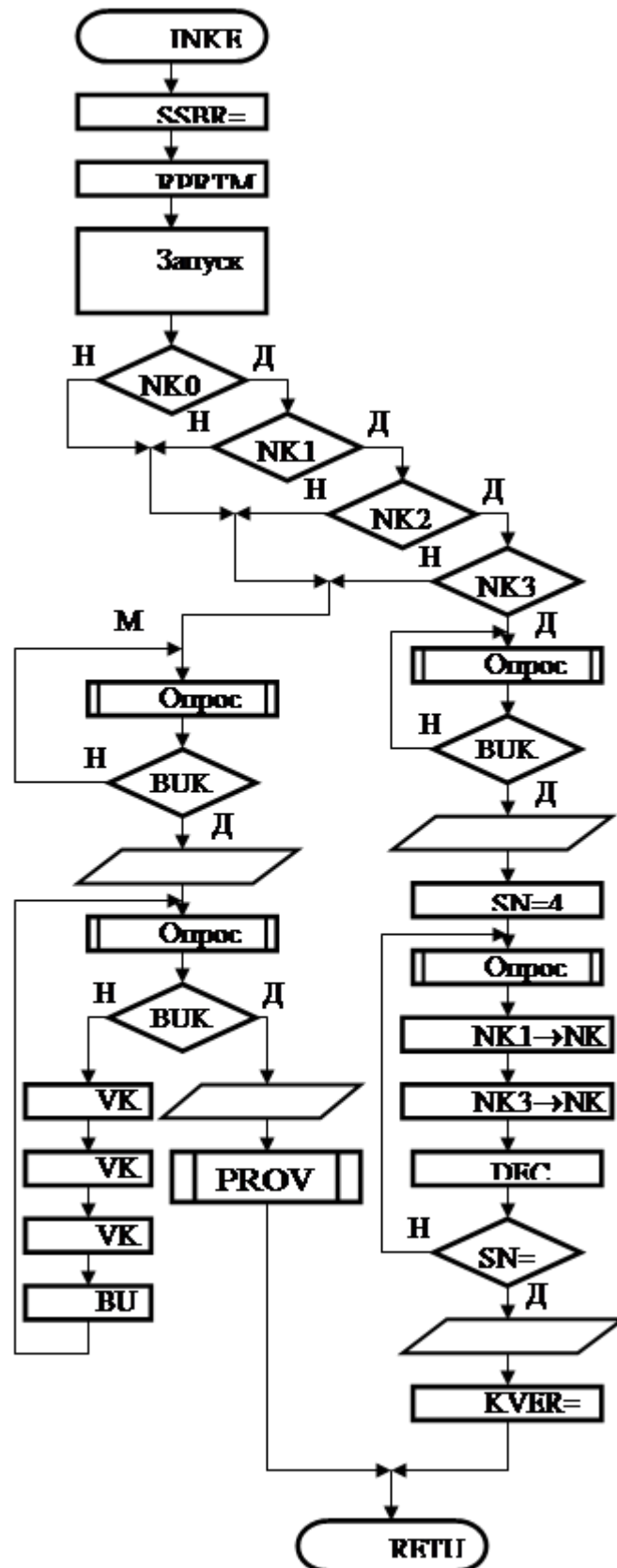


Рис.2.10. Схема алгоритму підпрограми INKEY [13]

Структурна схема алгоритму підпрограми PROV приведена на рис.2.11.

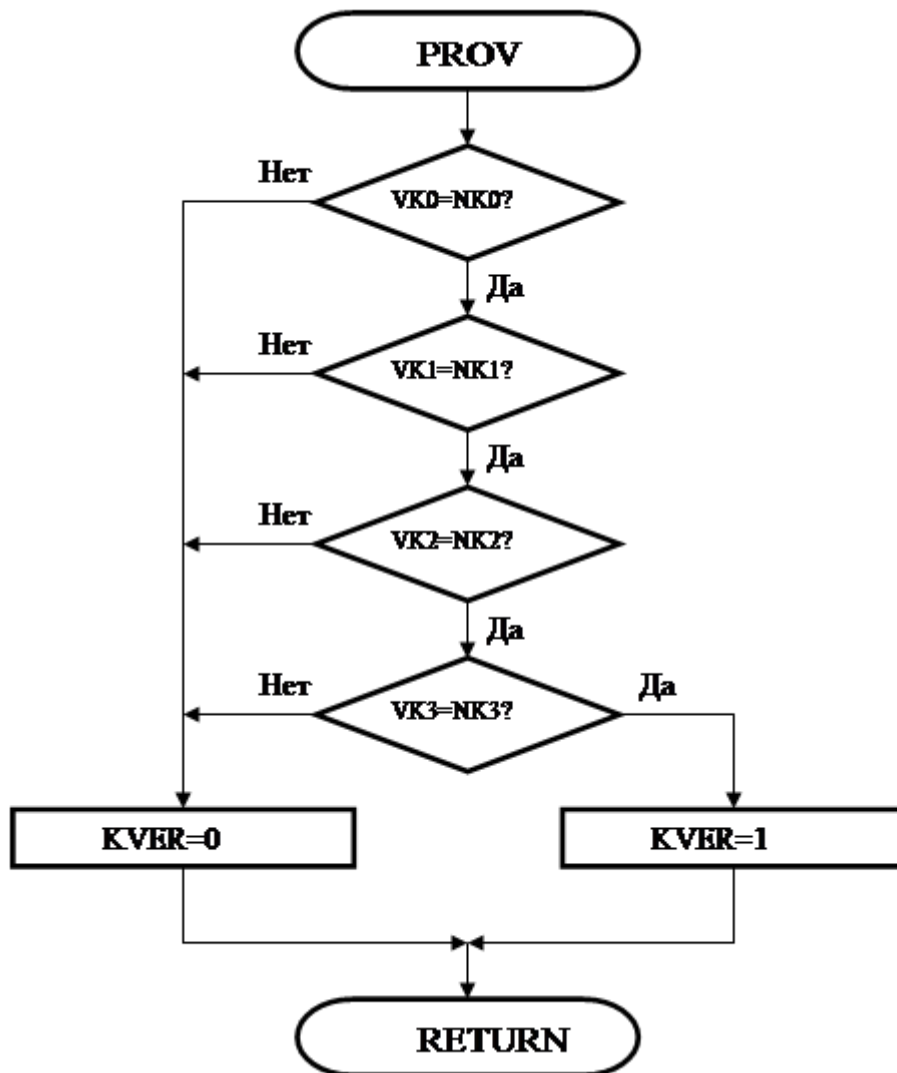


Рис.2.11. - Структурна схема алгоритму підпрограми PROV [14]

У ній виробляється послідовне порівняння вмісту комірок пам'яті VK_0 , VK_1 , VK_2 і VK_3 з вмістом осередків NK_0 , NK_1 , NK_2 і NK_3 відповідно. При збігу вмістів встановлюється прапорець $KVER = 1$, в іншому випадку $KVER = 0$ і відбувається вихід з підпрограми.

РОЗДІЛ 3

ЗАСТОСУВАННЯ МІКРОПРОЦЕСОРНОЇ СИСТЕМИ КОНТРОЛЮ

3.1 Промислові контролери СПЕКОН

Спеціалізовані промислові контролери СПЕКОН СК які зображені на рис. 3.1 призначені для автоматизованого управління паровими і водогрійними котлами, які працюють на газі або рідкому паливі, а також котельнями, ЦТП, теплогенераторами, полум'яними печами та іншими технологічними об'єктами в різних галузях промисловості.

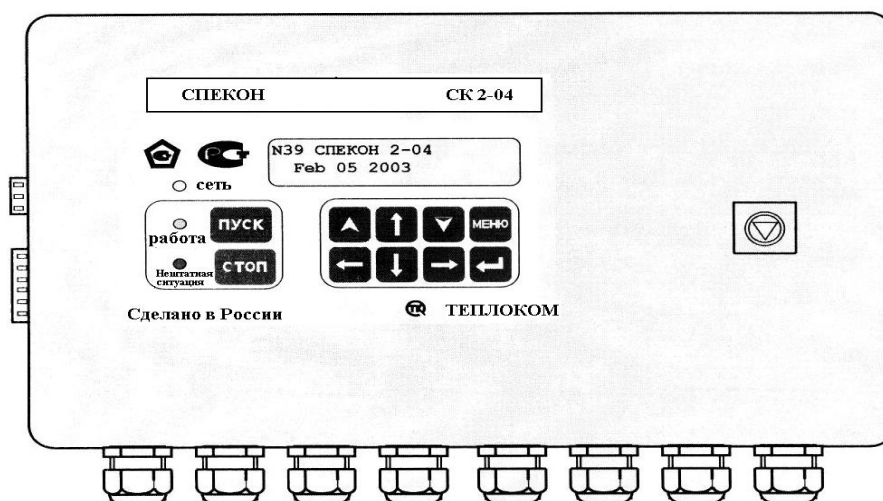


Рис. 3. 1. - Зовнішній вигляд контролера (вид спереду) [15]

Для подання інформації про хід технологічного процесу, значенні параметрів, складі системи на лицьовій панелі контролера розташовуються алфавітно-цифрове табло і світлові індикатори.

Алфавітно-цифрове табло рідкокристалічне, дворядкове, має по 16 знаків у кожному рядку. Табло має підсвічування «Мережа», «Робота», «Позаштатна ситуація».

Введення бази даних, вивід значень параметрів, управління техпроцесом, здійснюється з клавіатури лицьовій панелі.

3.2 Контролер управління системами вентиляції БіКуб-ВК02

Контролер являє собою регулюючий пристрій, виконане на базі мікроконтролера саморозміщуваних програмним забезпеченням, і призначений для регулювання температури припливного повітря в системах повітряного опалення. Контролер може бути конфігурований на роботу в різних модифікаціях систем припливної вентиляції.

Контролер може застосовуватися в автоматизованих системах контролю та управління. Прилад спільно з іншими виробами фірми ТОВ «НВП» Гірське Плюс» і виробами сторонніх фірм, що мають можливість підключення до інформаційних систем (електролічильники, теплотічильники) дозволяє організувати комплексне управління інженер інженерним обладнанням на рівні будівлі або комплексу будівель.

Принципова схема застосування контролера «БіКуб-ВК02» представлена на рис. 3.2.

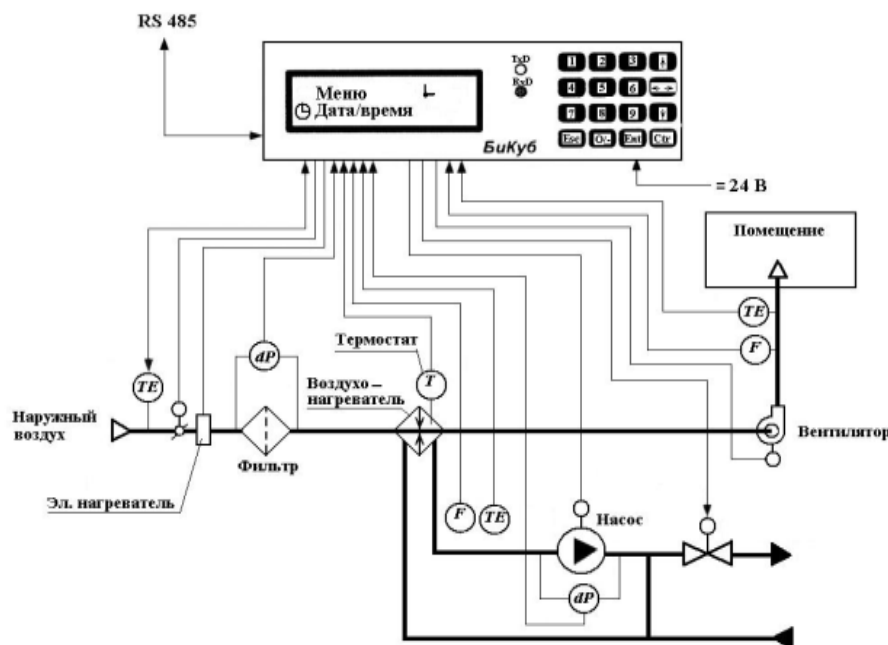


Рис. 3.2. - Приклад застосування контролера «БіКуб-ВК02» [16]

У розглянутому прикладі контролер управляє вентилятором, заслінкою з електронагрівачем, насосом і двох ходовим клапаном з електроприводом.

Сигнали з датчиків температури надходять на відповідні входи приладу і піддаються аналого-цифровому перетворенню.

Далі здійснюються перетворення відповідно до номінальних функціями перетворення з тим, щоб отримати в цифровій формі значення вимірюваних температур. Виміряні значення температур можна спостерігати на дисплеї або прочитати по мережі. [17]

У режимі «Контроль», прилад виконує операції, спрямовані на підтримання оптимальної температури теплоносія в зворотному трубопроводі, для запобігання заморожування системи і перевищення температури теплоносія в зворотному трубопроводі.

У режимі «Робота» контролер послідовно виконує функції запуску системи вентиляції, а потім функції пов'язані з підтриманням заданої температури припливного повітря.

У процесі роботи в цьому режимі контролер може переводити систему в різні стани.

Прогрів калорифера, перед початком роботи контролер здійснює прогрів калорифера, для чого при закритих жалюзі і вимкненому вентиляторі, здійснює відкриття регулюючого клапана, включення насоса і включення електронагрівача.

У цьому стані система знаходиться протягом часу заданого користувачем. У разі якщо температура зовнішнього повітря більше значення, що визначає «літній режим», то це система не переводиться в цей стан.

Управління системою припливної вентиляції. Після прогріву система переводиться в робочий стан. У цьому стані прилад підтримує значення температури припливного повітря відповідно до заданого.

Захист від заморожування. При падінні температури припливного повітря або температури теплоносія в зворотному трубопроводі нижче заданих користувачем значень, або виникненні несправностей контролер переводить систему в стан захисту від заморожування.

У цьому стані прилад закриває жалюзі, вимикає вентилятор і відкриває виконавчий механізм. Система буде перебувати в цьому режимі до тих пір, поки значення температур повітря та зворотної води не прийдуть в норму.

Черговий режим. Черговий режим передбачений для тих випадків, коли в роботі вентиляції немає необхідності. У цьому режимі прилад контролює тільки температуру зворотної води, жалюзі при цьому закриті, а вентилятор вимкнений.

Перехід в черговий режим здійснюється шляхом завдання тимчасового інтервалу відповідного цьому режиму. Якщо перехід в черговий режим здійснений з «літнього» режиму, то контроль зворотної води не виконується.

Літній режим. В цьому режимі управління температурою припливного повітря не здійснюється.

І циркуляція теплоносія через калорифер припинена. Контролер просто відкриває жалюзі і включає вентилятор. Блока схема підключення мотора, вентилятора зображено на рис.3.3.

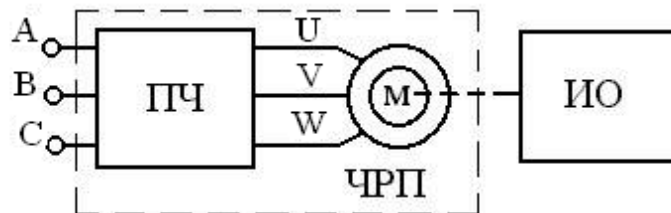


Рис.3.3. - Блока схема підключення мотора, вентилятора [18]

ВИСНОВОК

Мікропроцесор є функціонально завершеним універсальним програмно-керованим пристроєм цифрової обробки даних, виконане у вигляді однієї або декількох мікропроцесорних систем.

У структурі мікропроцесора можна виділити три основні частини:

- реєстри для тимчасового зберігання команд, даних і адрес;
- арифметико-логічний пристрій, який реалізує арифметичні і логічні операції;
- схема управління і синхронізації - забезпечує вибірку команд, організовує функціонування, забезпечує доступ до всіх реєстрах мікропроцесора, сприймає і генерує зовнішні керуючі сигнали.

Для взаємодії процесора та зовнішніх пристроїв застосовується система (механізм) переривань. Це спеціальний механізм, який дозволяє в будь-який момент, по зовнішньому сигналу змусити процесор призупинити виконання основної програми, виконати операції, пов'язані із зухвалим переривання подією, а потім повернутися до виконання основної програми.

Алгоритм - точне розпорядження, однозначно задає процес перетворення вихідної інформації в послідовність операцій, що дозволяють вирішувати сукупність завдань певного класу і отримувати шуканий результат.

Алгоритми функціонування мікропроцесорної системи контролю призначені для автоматизованого управління паровими і водогрійними котлами, які працюють на газі або рідкому паливі, а також котельнями, ЦТП, теплогенераторами, полум'яними печами та іншими технологічними об'єктами в різних галузях промисловості.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Архитектура микроконтроллеров семейства MCS-51: конспект лекций/ Бояринов А.Е., Дьяков И.А. Тамбов: Изд-во Тамб. гос. тех. ун-та, 2005. 64с.
2. Микропроцессоры в системах контроля: методические указания/ Бояринов А.Е. Тамбов: Изд-во Тамб. гос. тех. ун-та, 2005. 44с.
3. Денисов А.А., Колесников Д.Н. Теория больших систем управления: Учеб. пособие для вузов. Л.: Энергоиздат, 2005. 288 с.
4. Дружинин В.В., Конторов Д.С. Системотехника. М.: Радио и связь, 2007. 200 с.
5. Патент 2170423 РФ на изобретение. Термозонд для неразрушающего контроля теплофизических свойств материалов и готовых изделий / В.Н. Чернышов, З.М. Селиванова. Опубл. 10.07.2001 г., Бюл. № 19.
6. Методы определения теплопроводности и температуропроводности / Под. ред. А.В. Лыкова. М.: Энергия, 2010. 336 с.
7. Муромцев Ю.Л., Ляшков В.И., Селиванова З.М. и др. Микропроцессорная ИВС оперативного контроля // Гибкие автоматизированные производства и роботизация технологических процессов: Тез. докл. / ТГТУ. Тамбов, 2006. С. 91 – 92.
8. Цветков Э.И. Алгоритмические основы измерений. Л.: Энергоатомиздат, 2002. 254 с.
9. Селиванова З.М., Чернышов В.Н. Термозонд для неразрушающего контроля теплофизических свойств материалов и готовых изделий // Вестник ТГТУ. Тамбов, 2000. Т. 6, № 3. С. 402 – 407.
10. Преображенский В.П. Теплотехнические измерения и приборы. М.: Энергия, 2008. 704 с.
11. Селиванова З.М., Чернышов В.Н. Система неразрушающего контроля теплофизических свойств материалов со структурно-параметрической адаптацией на воздействие дестабилизирующих факторов // VI науч. конф. ТГТУ: Материалы конф. Тамбов, 2001. С. 218 – 219.

12. Селиванова З.М. Анализ дестабилизирующих факторов, воздействующих на адаптивную измерительно-вычислительную систему неразрушающего контроля теплофизических свойств материалов и готовых изделий // Информационные технологии в проектировании микропроцессорных систем: Тез. докл. Междунар. конф. / ТГТУ. Тамбов, 2000. С. 94 – 96.

13. Методы электрических измерений / Л.Г. Журавин, М.А. Мариненко, Е.И. Семенов, Э.И. Цветков; Под ред. Э.И. Цветкова. Л.: Энергоатомиздат, 2000. 288 с.

14. Варфоломеев Б.Г., Грошев В.Н., Селиванова З.М. Оценка погрешности определения теплофизических характеристик материалов, обусловленной контактным термосопротивлением // Теплофизические проблемы промышленного производства: Тез. докл. Всесоюзного теплофизического совещания. Тамбов, 2012. С. 53.

15. Григорьев А.С., Муромцев Ю.Л., Селиванова З.М. Комплексное определение теплофизических свойств и влажности твердых материалов // Теплофизика релаксирующих систем: Тез. докл. Всесоюзного теплофизического совещания. Тамбов, 2010. С. 106.

16. Селиванова З.М., Чернышов В.Н. Адаптивная микропроцессорная система для определения теплофизических свойств материалов в среде с переменной температурой // V науч. конф. ТГТУ: Тез. докл./ ТГТУ. Тамбов, 2000. С. 258.

17. Селиванова З.М., Чернышов В.Н. Метод и адаптивная измерительно-вычислительная система неразрушающего контроля теплофизических свойств материалов и изделий при воздействии дестабилизирующих факторов // Контроль, диагностика. 2001. С. 12 – 15.

18. Селиванова З.М., Казаков В.Н. Повышение точности и быстродействия теплофизических измерений полистирольных пенопластов Тез. докл. II Междунар. теплофизич. школы / ТГТУ. Тамбов, 2005. С. 246.