

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ  
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**Навчальний посібник  
з дисципліни  
“Мікропрограмування”**

Тернопіль - 2011

Р.П. Шевчук // Навчальний посібник з дисципліни „Мікропрограмування”, для студентів напрямку „Програмна інженерія”. – Тернопіль, 2011. - 121 с.

**Анотація.** Особливо важливим аспектом підготовки спеціалістів комп’ютерних наук є успішне засвоєння ними дисципліни “Мікропрограмування”. У навчальному посібнику наведено теоретичні та практичні основи роботи з однокристальними мікроконтролерами, досліджено можливості та сфери застосування мікропроцесорної техніки; архітектурні, операційні та програмні особливості сучасних мікроконтролерів; основи створення програмних продуктів для мікроконтролерів. Викладений матеріал повинен сприяти формуванню висококваліфікованих фахівців у галузі програмного забезпечення автоматизованих систем.

**Укладач:** Шевчук Руслан Петрович, к.т.н., доцент кафедри комп’ютерних наук ТНЕУ

**Відповідальний за випуск:** Дивак Микола Петрович, д.т.н., професор., завідувач кафедри комп’ютерних наук ТНЕУ

**Рецензенти:** професор кафедри комп’ютерної інженерії ТНЕУ, д.т.н., професор Карпінський М.П.

доцент кафедри комп’ютерних технологій і систем управління Івано-Франківського національного технічного університету нафти і газу, к.т.н., доцент Малько О.Г.

Затверджено на засіданні кафедри комп’ютерних наук ТНЕУ.  
Протокол № 17 від «14» червня 2011 р.

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ .....	4
ТЕМА 1. ОСНОВНІ ХАРАКТЕРИСТИКИ ТА ОСОБЛИВОСТІ РОБОТИ МІКРОПРОЦЕСОРНИХ СИСТЕМ.....	5
1.1. Структура мікропроцесорної системи .....	5
1.2. Ядро мікропроцесорної системи .....	8
1.3. Організація внутрішніх шин мікропроцесорних систем .....	10
1.4. Режими роботи мікропроцесорної системи .....	14
1.5. Типи мікропроцесорних систем .....	18
Тестові питання до теми 1 .....	20
ТЕМА 2 – ФУНКЦІЇ ПРИСТРОЇВ МІКРОПРОЦЕСОРНОЇ СИСТЕМИ .	29
2.1. Функції процесора.....	29
2.2. Функції пам'яті.....	34
2.3. Функції пристроїв введення/виведення.....	40
Тестові питання до теми 2.....	43
ТЕМА 3. ОСОБЛИВОСТІ ПРОЕКТУВАННЯ МІКРОПРОЦЕСОРНИХ СИСТЕМ І ПРИСТРОЇВ НА МІКРОКОНТРОЛЕРАХ.....	53
3.1. Формалізація проектування МПС і пристроїв .....	53
3.2. Схема та методика процесу проектування МПС та пристроїв .....	56
3.3. Типові структури МК-систем і пристроїв.....	59
3.4. Використання жорсткої і програмованої логіки .....	60
3.5. Дуалізм "програмні засоби – апаратура" .....	61
Тестові питання до теми 3 .....	62
ТЕМА 4. РОЗРОБКА МК-СИСТЕМ НА ОДНОКРИСТАЛЬНИХ МІКРОКОНТРОЛЕРАХ.....	74
4.1. Основні характеристики і класифікація однокристальних мікроконтролерів.....	74
4.2. Обґрунтування застосування та вибору сімейства ОМК.....	77
4.3. Етапи розробки МК-систем на базі ОМК.....	79
4.4. Особливості розробки прикладного програмного забезпечення ОМК.....	83
Тестові питання до теми 4.....	86
ТЕМА 5. СТРУКТУРНА ОРГАНІЗАЦІЯ ТА РЕЖИМИ РОБОТИ ОМК РІС.....	94
5.1. Загальні відомості про ОМК РІС та їх класифікація.....	94
5.2. Однокристальні мікроконтролери базового сімейства РІС16С5Х .	97
СПИСОК ЛІТЕРАТУРИ.....	121

## ПЕРЕЛІК СКОРОЧЕНЬ

МПС – мікропроцесорна система  
МКП – мікроконтролерний пристрій  
МП – мікропроцесор  
МК - мікроконтролер  
ОЗУ – оперативно запам'ятовуючий пристрій  
ПЗП – постійно запам'ятовуючий пристрій  
АЦП – аналогово-цифровий пристрій  
ЦАП – цифрово-аналоговий пристрій  
МА – магістраль адрес  
ША – шина адрес  
МД – магістраль даних  
ШД – шина даних  
МУ – магістраль управління  
ШУ – шина управління  
ОУ – об'єкт управління  
ВМ – виконавчі механізми  
ІК – інформаційні контролери  
БСД – блок спряження з датчиками  
БСІК – блок спряження з інформаційними контролерами  
ОП – основна пам'ять  
ДП – додаткова пам'ять  
РЗЗ – регістр загального значення  
ПДП – прямий доступ до пам'яті  
АЛП – арифметико-логічний пристрій  
ТЗ – технічне завдання  
ОМК – однокристальний мікроконтролера  
ТЕЗ – типовий елемент заміни  
АВЗ – апаратура взаємного зв'язку  
ВІС – велика інтегральна мікросхема  
НВІС – надвелика інтегральна мікросхема  
ОМК – однокристальні мікроконтролери  
ТТЛ – транзисторно-транзисторна логіка  
КМОП - комплементарна логіка на транзисторах метал-оксид-напівпровідник

# ТЕМА 1. ОСНОВНІ ХАРАКТЕРИСТИКИ ТА ОСОБЛИВОСТІ РОБОТИ МІКРОПРОЦЕСОРНИХ СИСТЕМ

## 1.1. Структура мікропроцесорної системи

Мікропроцесорною системою (МПС) називається система цифрового керування та оброблення інформації. Розглянемо МПС як окремий випадок електронної системи, призначеної для оброблення вхідних сигналів і видачі вихідних сигналів (рисунок 1.1).

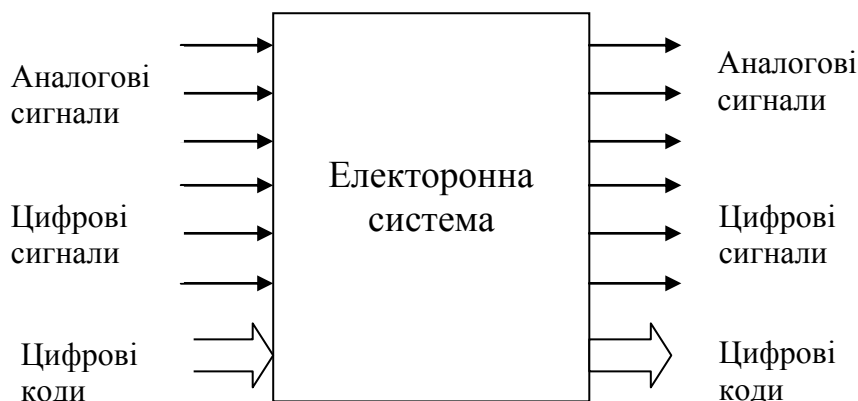


Рисунок 1.1 - Електронна система

Вхідними та вихідними сигналами в електронній системі можуть бути аналогові сигнали, одиночні цифрові сигнали, цифрові коди, послідовності цифрових код. Всередині системи проводиться зберігання або накопичення сигналів. Якщо система цифрова (а МПС відносяться до розряду цифрових), то вхідні аналогові сигнали перетворюються в послідовності кодових вибірок за допомогою АЦП, а вихідні аналогові сигнали формуються з послідовності кодових вибірок з допомогою ЦАП. Оброблення і зберігання інформації виконується у цифровому вигляді.

Характерна особливість традиційної цифрової системи полягає в тому, що алгоритми оброблення і зберігання інформації в ній жорстко пов'язані з схемотехнікою системи. Тобто зміна цих алгоритмів можлива тільки шляхом зміни структури системи, заміни електронних вузлів, що входять в систему, і/або зв'язків між ними. Наприклад, якщо нам потрібна додаткова операція підсумовування, то необхідно додати в структуру системи суматор. Якщо потрібна додаткова функція зберігання коду протягом одного такту, то ми повинні додати в структуру ще один регістр. Такі зміни в структурі системи практично неможливо зробити в процесі експлуатації, обов'язково потрібен новий виробничий цикл проектування, виготовлення, відлагодження всієї системи. Саме тому традиційна цифрова система часто називається системою на "жорсткій логіці".

Будь-яка система на "жорсткій логіці" обов'язково є спеціалізованою системою, налаштованою виключно на одне завдання або (рідше) на декілька близьких, заздалегідь відомих завдань. Це має свої безперечні переваги.

По-перше, спеціалізована система (на відміну від універсальної) ніколи не має апаратної надмірності, тобто кожен її елемент обов'язково працює в з повною віддачею.

По-друге, саме спеціалізована система може забезпечити максимально високу швидкодію, оскільки швидкість виконання алгоритмів оброблення інформації визначається в ній тільки швидкістю окремих логічних елементів і вибраною схемою шляхів проходження інформації. Самі логічні елементи при цьому завжди володіють максимальною швидкістю.

В той же час великим недоліком цифрової системи на "жорсткій логіці" є те, що для кожного нового завдання її потрібно проектувати і виготовляти заново. Цей процес тривалий, дорогий і вимагає високої кваліфікації виконавців.

Шлях подолання цього недоліку досить очевидний: треба побудувати таку систему, яка могла б легко адаптуватися під будь-яке завдання, перебудовуватися з одного алгоритму роботи на інший без зміни апаратури. Це питання вирішується шляхом введення в структуру системи необхідної програми (рисунок 1.2).

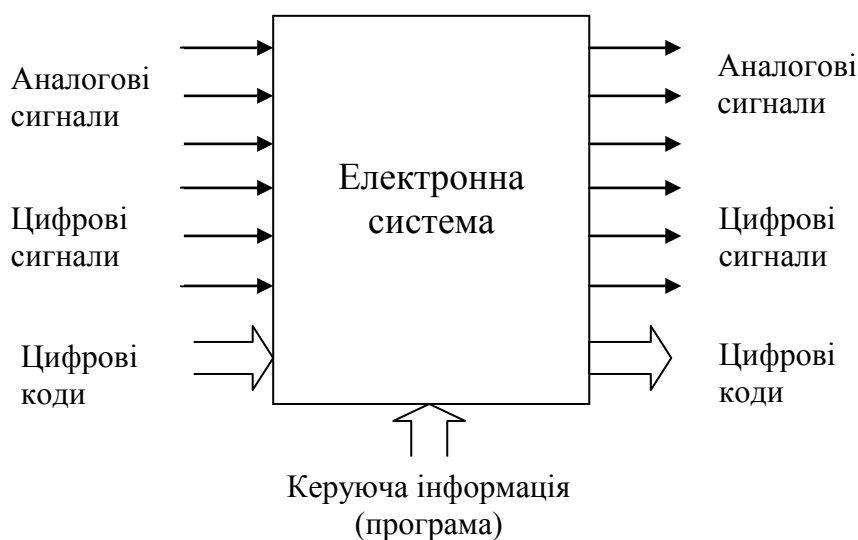


Рисунок 1.2 - Програмована електронна система

Тоді система стає універсальною, або програмованою, не жорсткою, а гнучкою. Саме це і забезпечує МПС. Будь-яка універсальність обов'язково приводить до надмірності. Адже рішення максимально важкого завдання вимагає значно більше засобів, чим рішення максимально простого завдання. Тому складність універсальної системи повинна бути такою, щоб забезпечувати рішення найважчого завдання, а при рішенні простого завдання система повинна працювати не на повну потужність. Чим простіше, тим більше надмірність, і тим менш виправданою стає універсальність. Надмірність веде до збільшення вартості системи, зниження її надійності, збільшення споживаної потужності і так далі.

Крім того, універсальність, як правило, призводить до істотного зниження швидкодії. Оптимізувати універсальну систему так, щоб кожне нове завдання

вирішувалося максимально швидко, просто неможливо. Загальне правило таке: чим більша універсальність, гнучкість - тим менша швидкодія. Більше того, для універсальних систем не існує таких завдань (нехай навіть і найпростіших), які б вони вирішували з максимально можливою швидкістю.

Таким чином, можна зробити наступний висновок. Системи на "жорсткій логіці" хороші там де:

- завдання не міняється тривалий час;
- потрібна найвища швидкодія;
- алгоритми оброблення інформації гранично прості.

А універсальні, програмовані системи хороші там де:

- часто міняються завдання;
- висока швидкодія не дуже важлива;
- алгоритми обробки інформації складні.

За останні десятиліття швидкодія універсальних (мікропроцесорних) систем сильно виросла (на декілька порядків). До того ж великий об'єм випуску мікросхем для цих систем привів до різкого зниження їх вартості. В результаті область застосування систем на "жорсткій логіці" різко звузилася. Більш того, високими темпами розвиваються програмовані системи, призначені для вирішення одного завдання або декількох близьких за змістом завдань. Вони вдало суміщають в собі як переваги систем на "жорсткій логіці", так і програмованих систем, забезпечуючи поєднання достатньо високої швидкодії і необхідної гнучкості.

У складі МПС є, принаймні, один мікропроцесор (МП), один або декілька модулів основної (ОЗУ і ПЗУ) і додаткової пам'яті, пристрої введення і виведення, блоки спряження (контролери) з пристроями введення і виведення, які зв'язані один з одним за допомогою системної магістралі, що складається з магістралей (шин) адрес (МА, ША), магістралей (шин) даних (МД, ШД) і магістралей (шин) управління (МУ, ШУ).

Логічна структура МПС приведена на рисунок 1.3, де ОУ – об'єкт управління, Д – датчики, ВМ – виконавчі механізми, ІК – інформаційні контролери, БСД – блок спряження з датчиками, БСІК – блок спряження з інформаційними контролерами, ОП – основна пам'ять, ДП – додаткова пам'ять.

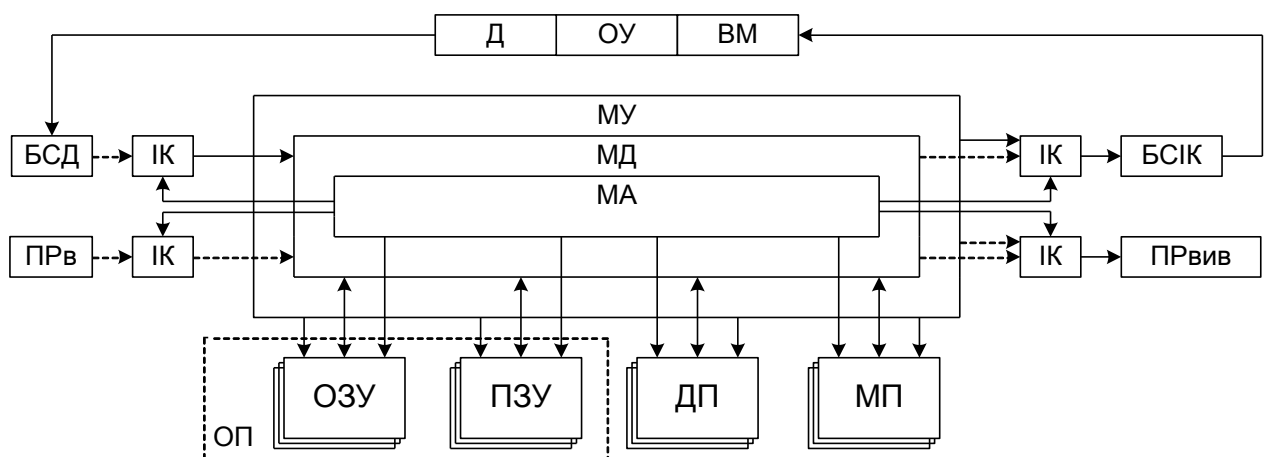


Рисунок 1.3 - Логічна структура МПС

ОЗУ МПС забезпечує читання і запис інформації і реалізується як енергозалежна пам'ять, вміст якої стирається при виключенні МПС. ПЗП забезпечує тільки читання інформації і реалізується у вигляді незалежної пам'яті. Контроллерами є пристрої сполучення апаратури введення-виведення з системною магістраллю і реалізують певний інтерфейс. Магістраль забезпечує комунікацію апаратних засобів МПС і є набором провідників і підсилювачів сигналів.

## 1.2. Ядро мікропроцесорної системи

Ядром будь-якої МПС є мікропроцесор або просто процесор. МП - це той вузол, який виконує все оброблення інформації всередині МПС. Процесор замінює практично всю "жорстку логіку", яка потрібна будь-якій традиційній цифровій системі. Він виконує арифметичні функції, логічні функції, тимчасове зберігання коду, пересилку коду між вузлами МПС і багато інших функцій. Кількість елементарних операцій, що виконуються процесором, може досягати декількох сотень. Процесор можна порівняти з мозком системи.

МП характеризується дуже великим числом параметрів і якостей, оскільки він, з одного боку, є функціонально складним програмно-керуючим цифровим процесором, тобто пристроєм ЕОМ, а з іншого боку – інтегральною схемою з високим ступенем інтеграції елементів, тобто електронним приладом.

МП класифікуються по наступних ознаках.

По числу ВІС в мікропроцесорному комплекті – однокристалні і багатокристалні МП. Однокристалні МП виходять при реалізації всіх апаратних засобів процесора у вигляді однієї ВІС або НВІС. Для отримання багатокристалного МП необхідно провести розбиття його логічної структури на функціонально закінчені частини і реалізувати їх у вигляді ВІС.

За призначенням розрізняють універсальні і спеціалізовані МП. По виду оброблювальних вхідних сигналів МП ділять на цифрові і аналогові. По характеру тимчасовій організації роботи – синхронні і асинхронні.

Окрім цього МП, як правило, класифікуються: за технологією виготовлення (р-МОП, п-МОП, к-МОП і т.д); по числу шин; по розрядності; за способом управління (схемне, мікропрограмне); по числу акумуляторів, рівнів переривання і програмних лічильників; за типом і ємкістю стека; по числу і довжині команд і по видах адресації.

Всі свої операції МП виконує послідовно, тобто одну за іншою. Звичайно, існують МП з паралельним виконанням деяких операцій, зустрічаються також МПС, в яких декілька МП працюють над одним завданням паралельно. З одного боку, послідовне виконання операцій - безперечна перевага, оскільки один процесор виконує будь-які алгоритми оброблення інформації. З іншого боку, послідовне виконання операцій приводить до того, що час виконання алгоритму залежить від його складності. Прості алгоритми виконуються швидше складних. Тобто МПС здатна зробити все, але працює вона не дуже швидко, адже всі інформаційні потоки доводиться пропускати через один-



єдиний вузол - мікропроцесор (рисунок 1.3). У традиційній цифровій системі можна легко організувати паралельне оброблення всіх потоків інформації.

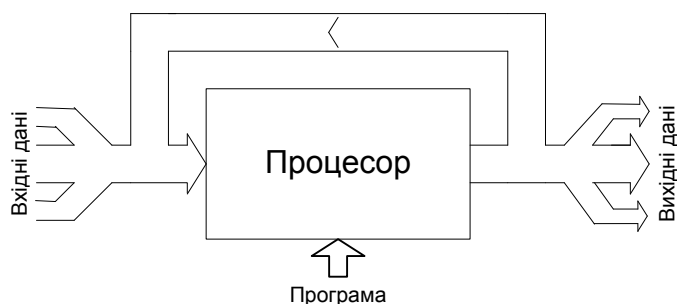


Рисунок 1.4 - Інформаційні потоки в мікропроцесорній системі

Отже, мікропроцесор здатний виконувати безліч операцій. Але звідки він дізнається, яку операцію йому треба виконувати в даний момент? Саме це визначається інформацією, яка керує програмою. Програма є набором команд (інструкцій), розшифрувавши які, процесор дізнається, що йому треба робити. Програма від початку і до кінця складається людиною, програмістом, а процесор виступає в ролі виконавця цієї програми.

Всі команди, що виконуються МП, утворюють систему команд процесора. Структура і об'єм системи команд МП визначають його швидкодію, гнучкість, зручність використання. Всього команд у МП може бути від декількох десятків до декількох сотень. Система команд може бути розрахована на вузький круг вирішуваних завдань (у спеціалізованих процесорів) або на максимально широкий круг завдань (в універсальних процесорів). Коди команд можуть мати різну кількість розрядів (займати від одного до декількох байт). Кожна команда має свій час виконання, тому час виконання всієї програми залежить не тільки від кількості команд в програмі, але і від того, які саме команди використовуються.

Для виконання команд в структуру МП входять внутрішні регістри, АЛП, мультиплектори, буфери, регістри і інші вузли. Робота всіх вузлів синхронізується загальним зовнішнім тактовим сигналом МП. Тобто процесор є досить складний цифровий пристрій (рисунок 1.5).

Втім, для розробника МПС інформація про внутрішню структуру процесора не дуже важлива. Розробник повинен розглядати процесор як "чорний ящик", який у відповідь на вхідні коди, проводить ту або іншу операцію і видає вихідні сигнали. Розробникові необхідно знати систему команд, режими роботи процесора, а також правила взаємодії процесора із зовнішнім світом (протоколи обміну інформацією). Про внутрішню структуру процесора треба знати тільки те, що необхідне для вибору тієї або іншої команди або режиму роботи.



Рисунок 1.5 - Приклад структури простого процесора

### 1.3. Організація внутрішніх шин мікропроцесорних систем

Для досягнення максимальної універсальності і спрощення протоколів обміну інформацією в МПС використовується шинна структура зв'язків між окремими пристроями, що входять в МПС. Класична структура шинних зв'язків наведена на рисунку 1.6.

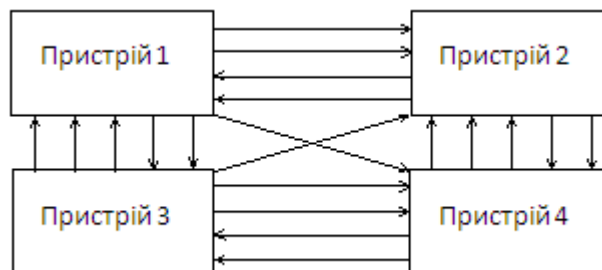


Рисунок 1.6 – Класична структура шинних зв'язків

При класичній структурі зв'язків (рис. 1.6) всі сигнали і коди між пристроями передаються по окремих лініях зв'язку. Кожен пристрій, що входить в систему, передає свої сигнали і коди незалежно від інших пристроїв. При цьому в системі виходить дуже багато ліній зв'язку і різних протоколів обміну інформацією.

При шинній структурі зв'язків (рисунок 1.7) всі сигнали між пристроями передаються по одних і тих же лініях зв'язку, але в різний час (мультиплексова передача). Причому передача по всіх лініях зв'язку може здійснюватися в обох напрямках (так звана двонаправлена передача). В результаті кількість ліній зв'язку істотно скорочується, а правила обміну (протоколи) спрощуються. Група ліній зв'язку, по яких передаються сигнали або коди називається шиною.

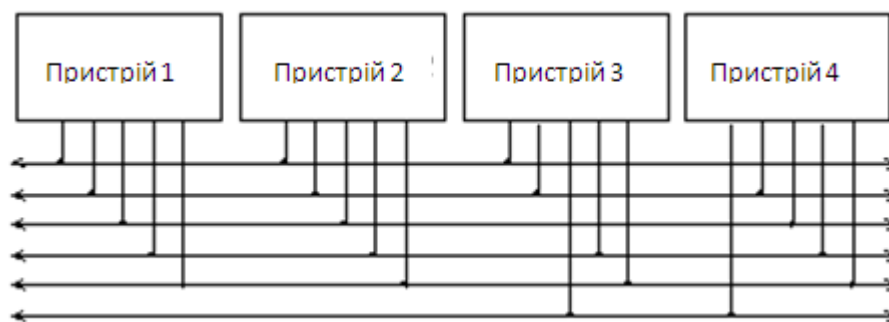


Рисунок 1.7 – Шинна структура зв'язків

Зрозуміло, що при шинній структурі зв'язків легко здійснюється пересилка всіх інформаційних потоків в потрібному напрямі, наприклад, їх можна пропустити через один МП, що дуже важливе для МПС. Проте при шинній структурі зв'язків вся інформація передається по лініях зв'язку послідовно в часі, по черзі, що знижує швидкодію системи в порівнянні з класичною структурою зв'язків.

Перевага шинної структури зв'язків полягає в тому, що всі пристрої, підключені до шини, повинні приймати і передавати інформацію по одних і тих же протоколам. Відповідно, всі вузли, що відповідають за обмін з шиною в цих пристроях, повинні бути одноманітні та уніфіковані. Недолік шинної структури пов'язаний з тим, що всі пристрої підключаються до кожної лінії зв'язку паралельно. Тому несправність будь-якого пристрою може вивести з ладу всю МПС, якщо вона пошкоджує лінію зв'язку. Саме тому відлагодження системи з шинною структурою зв'язків досить складне і зазвичай вимагає спеціальних засобів.

Всі пристрої МПС об'єднуються загальною системною шиною (системна магістраль або канал). Системна магістраль включає чотири основні шини нижнього рівня:

- шина адреси (Address Bus);
- шина даних (Data Bus);
- шина управління (Control Bus);
- шина живлення (Power Bus).

ША служить для визначення адреси (номера) пристрою, з яким МП обмінюється інформацією в даний момент. Кожному пристрою (окрім процесора), кожному елементу пам'яті в МПС присвоюється унікальна адреса. Коли код якоїсь адреси подається процесором на ША, пристрій, якому ця адреса приписана, розуміє, що його чекає обмін інформацією. Шина адреси може бути однонаправленою або двонаправленою.

ШД - це основна шина, яка використовується для передачі інформаційних кодів між всіма пристроями МПС. Зазвичай в пересилці інформації бере участь МП, який передає код даних в якийсь пристрій або в елемент пам'яті або ж приймає відповідний код. Можлива також і передача інформації між пристроями без участі МП. Шина даних завжди двонаправлена.

ШУ на відміну від ША і ШД складається з окремих сигналів, що керують. Кожен з цих сигналів під час обміну інформацією має свою функцію. Деякі

сигнали служать для визначення процесу передаваних чи приймання даних. Інші керуючі сигнали можуть використовуватися для підтвердження прийому даних, для скидання всіх пристроїв в початковий стан і так далі. Лінії ШУ можуть бути однонаправленими або двонаправленими.

Шина живлення призначена для живлення МПС. Вона складається з ліній живлення і загального дроту. У МПС може бути одне джерело живлення (частіше +5 В) або декілька джерел живлення (зазвичай ще -5 В +12 В і -12 В). Кожній напрузі живлення відповідає своя лінія зв'язку. Всі пристрої підключені до цих ліній паралельно.

Якщо в МПС треба ввести вхідний код (або вхідний сигнал), то МП по шині адреси звертається до потрібного пристрою введення/виведення та приймає по шині даних вхідну інформацію. Якщо з МПС треба вивести вихідний код (або вихідний сигнал), то МП звертається по шині адреси до потрібного пристрою введення/виведення та передає йому по шині даних вихідну інформацію.

Якщо інформація повинна пройти складну багатоступінчасту обробку, то процесор може зберігати проміжні результати в системній оперативній пам'яті. Для звернення до будь-якої комірки пам'яті МП подає її адресу на ША і передає в неї інформаційний код по ШД або ж приймає з неї інформаційний код по ШД. У пам'яті (оперативній і постійній) знаходяться також коди (команди виконуваної МП програми), що управляють, які МП також читає по ШД з адресацією по ША. Постійна пам'ять використовується в основному для зберігання програми початкового пуску МПС, яка виконується кожного разу після включення живлення. Інформація в неї заноситься розробником.

Таким чином, в МПС всі інформаційні коди і коди команд передаються по шинах послідовно, по черзі. Це визначає порівняно невисоку швидкість МПС.

Важливо враховувати, що пристрої введення/виведення найчастіше є пристроями на "жорсткій логіці". На них може бути покладена частина функцій, що виконуються МПС. Тому у розробника завжди є можливість перерозподіляти функції системи між апаратною і програмною реалізаціями оптимальним чином. Апаратна реалізація прискорює виконання функції, але має недостатню гнучкість. Програмна реалізація значно повільніша, але забезпечує високу гнучкість. Апаратна реалізація функцій збільшує вартість системи і її енергоспоживання, програмна - не збільшує. Найчастіше застосовується комбінування апаратних і програмних функцій.

Іноді пристрої введення/виведення мають в своєму складі процесор, тобто є невеликою спеціалізованою МПС. Це дозволяє перекласти частину програмних функцій на пристрої введення/виведення, розвантаживши центральний процесор системи.

При проектуванні МПС слід враховувати, що їх продуктивність і функціональні можливості безпосередньо залежать від організації внутрішніх шин МП – їх число істотно впливає на структуру і характеристики МПС в цілому.

При визначенні оптимального числа шин слід враховувати, що зменшення числа шин приводить до зменшення швидкодії МП і супроводжується введенням додаткових буферних регістрів, збільшує площу на кристалі, що відводиться під функціональні елементи, і тим самим збільшує функціональні можливості МП і МПС.

У трьохшинному МП при певній внутрішній організації регістру загального значення (РЗЗ) можливе виконання операцій за один такт, включаючи вибірку операндів з РЗЗ і запис результату в один з регістрів. Переваги: висока швидкодія і відсутність буферних регістрів, недолік – велика площа шин на кристалі.

Двохшинна організація при меншій площі шин вимагає введення одного-двох буферних регістрів і операції виконуються за два такти.

Організація МП на основі однієї шини дозволяє максимально ускладнити архітектуру МП, проте вимагає введення двох-трьох буферних регістрів і трьох тактів для виконання операцій.

При використанні магистральної організації МПС виникає складність в підключенні виходів декількох елементів до однієї шини (до одного провідника загальної шини). Відомо три наступні способи вирішення цієї задачі.

Логічне об'єднання (рисунок 1.8, а) – виконується за допомогою схеми АБО, на входи якої поступають сигнали від різних джерел інформації, заздалегідь оброблені сигналами управління на входах схем І.

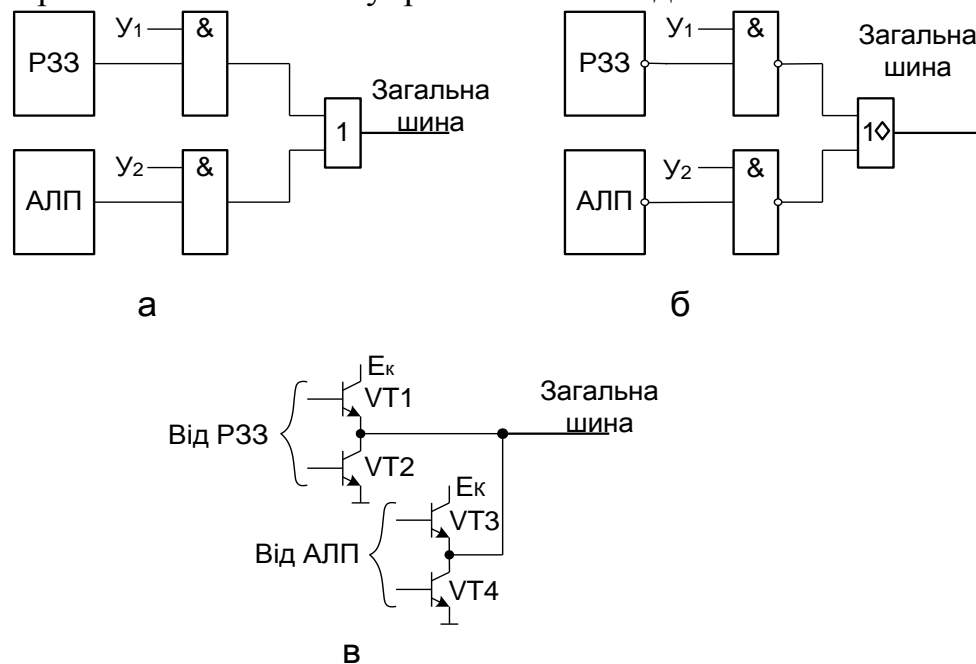


Рисунок 1.8 - Способи підключення пристроїв до загальної шини

Об'єднання за допомогою схем з відкритим колектором (рисунок 1.4, б) характеризується електричним з'єднанням виходів декількох логічних елементів. Часто цей спосіб називають «монтажним АБО» або «монтажним І».

Об'єднання з використанням схем з трьома станами (рисунок 1.4, в) відрізняється саме таким характером навантаження. На відміну від звичайних ключових схем тут можливий третій режим, при якому обидва транзистори

одного каскаду (VT1 і VT2 або VT3 і VT4) закриті. В цьому випадку з боку виходу каскад володіє високим опором і практично не впливає на стан загальної шини. Якщо в стані високого опору знаходяться обидва каскади, то загальна шина може використовуватися довільно будь-якими зовнішніми по відношенню до МП пристроями. Цей спосіб широко використовується при організації прямого доступу до пам'яті і при побудові мультипроцесорних систем.

#### 1.4. Режими роботи мікропроцесорної системи

МПС забезпечує велику гнучкість роботи, вона здатна настроюватися на будь-яке завдання. Гнучкість ця обумовлена перш за все тим, що функції, які виконуються системою, визначаються програмою, яку виконує МП. Апаратура залишається незмінною при будь-якому завданні. Записуючи в пам'ять системи програму, можна змусити МПС виконувати будь-яке завдання, що підтримується апаратурою. До того ж шинна організація зв'язків МПС дозволяє досить легко замінювати апаратні модулі, наприклад, замінювати пам'ять на нову більшого об'єму або вищої швидкодії, додавати або модернізувати пристрої введення/виведення, замінювати процесор. Це дозволяє збільшити гнучкість системи, продовжити її життя при будь-якій зміні вимог до неї.

Гнучкість МПС визначається не тільки цими фактами. Налаштування на завдання допомагає ще і вибір режиму роботи системи, тобто режиму обміну інформацією по системній магістралі.

Практично будь-яка сучасна МПС (у тому числі і комп'ютер) підтримує три основні режими обміну по магістралі:

- програмний обмін інформацією;
- обмін з використанням переривань (Interrupts);
- обмін з використанням прямого доступу до пам'яті (ПДП, DMA — Direct Memory Access).

Програмний обмін інформацією є основним в будь-якій МПС. Він передбачений завжди, оскільки без нього неможливі інші режими обміну. У цьому режимі МП є одноосібним господарем (Master) системної магістралі. Всі операції обміну інформацією в даному випадку ініціюються тільки МП та виконуються строго в порядку вказаному у програмі.

МП читає (вибирає) з пам'яті коди команд і виконує їх, читаючи дані з пам'яті або з пристрою введення/виведення, обробляючи їх, записуючи дані в пам'ять або передаючи їх в пристрій введення/виведення. Виконання програми МП може бути лінійним, циклічним, може містити переходи (стрибки), але цей процес завжди безперервний і повністю знаходиться під контролем процесора. Процесор не реагує ні на які зовнішні події, не пов'язані з програмою (рис. 1.9). Всі сигнали на магістралі в даному випадку контролюються процесором.



Рисунок 1.9 - Програмний обмін інформацією

Обмін по перериваннях використовується тоді, коли необхідна реакція МПС на якусь зовнішню подію, на прихід зовнішнього сигналу. У комп'ютері зовнішньою подією може бути натиснення на клавішу клавіатури або прихід по локальній мережі пакету даних. Комп'ютер повинен реагувати на це, відповідно, виведенням символу на екран або ж читанням і обробкою прийнятого по мережі пакету.

У загальному випадку організувати реакцію на зовнішню подію можна трьома різними шляхами:

- за допомогою постійного програмного контролю факту настання події (так званий метод опиту прапорця або polling);
- за допомогою переривання, тобто примусового передавання ресурсів процесора з виконання поточної програми на виконання екстрено необхідної програми;
- за допомогою прямого доступу до пам'яті, тобто без участі процесора при його відключенні від системної магістралі.

Проілюструвати ці три способи можна наступним простим прикладом. Допустимо, ви готуєте собі сніданок, поставивши на плиту кип'ятитися молоко. Природно, на закипання молока треба реагувати, причому терміново. Як це організувати? Перший шлях - постійно стежити за молоком, але тоді ви нічого іншого не зможете робити. Правильніше регулярно поглядатиме на молоко, роблячи одночасно щось інше. Це програмний режим з опитом прапорця. Другий шлях - встановити на каструлю з молоком датчик, який подасть звуковий сигнал при закипанні молока, і спокійно займатися іншими справами. Почувши сигнал, ви вимкнете молоко. Правда, можливо, вам доведеться спочатку закінчити те, що ви почали робити, так що ваша реакція буде повільніша, ніж в першому випадку. Нарешті, третій шлях полягає в тому, щоб з'єднати датчик на каструлі з управлінням плитою так, щоб при закипанні молока пальник був вимкнений без вашої участі (правда, аналогія з ПДП тут не дуже точна, оскільки в даному випадку на момент виконання дії вас не відволікають від роботи).

Перший випадок з опитом прапора реалізується в МПС постійним читанням інформації МП з пристрою введення/виведення, пов'язаного з тим зовнішнім пристроєм, на поведінку якого необхідно терміново реагувати.

У другому випадку в режимі переривання МП, отримавши запит переривання від зовнішнього пристрою (IRQ — Interrupt ReQuest), закінчує виконання поточної команди і переходить до програми обробки переривання. Закінчивши виконання програми обробки переривання, він повертається до перерваної програми з тієї точки, де його перервали (рисунк 1.10).

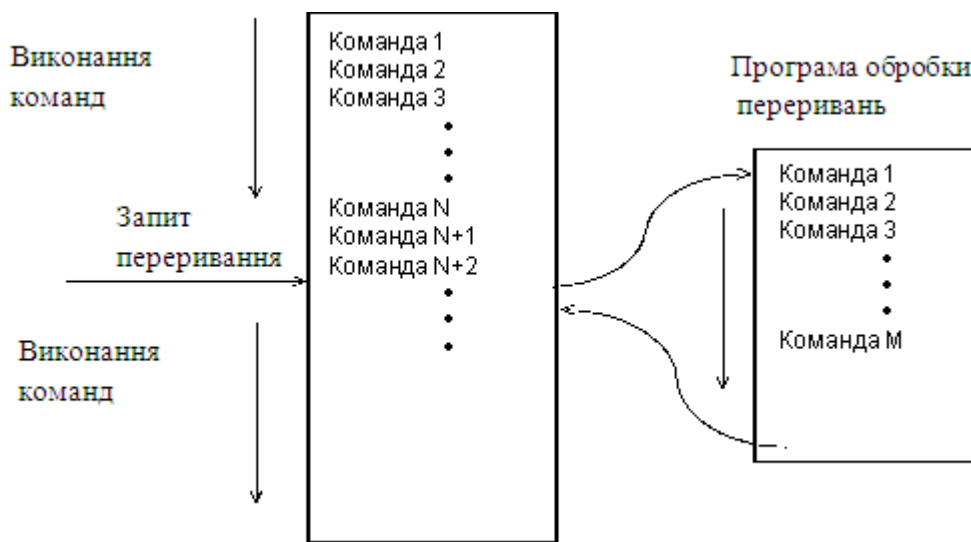


Рисунок 1.10 - Обслуговування переривання

Тут важливе те, що вся робота виконується самим МП, зовнішня подія просто тимчасово відволікає його. Реакція на зовнішню подію по перериванню в загальному випадку повільніша, ніж при програмному режимі. Як і у програмному обміні, всі сигнали на магістралі виставляються процесором, тобто він повністю контролює магістраль. Для обслуговування переривань в МПС іноді вводиться спеціальний модуль контролера переривань, але він в обміні інформацією не бере участь. Його завдання полягає в тому, щоб спростити роботу МП із зовнішніми запитами переривань. Цей контроллер зазвичай програмно управляється процесором по системній магістралі.

Зрозуміло, що ніякого прискорення роботи МПС переривання не дає. Його використання дозволяє тільки відмовитися від постійного опитування прапора зовнішньої події і тимчасово, до настання зовнішньої події, зайняти МП виконанням якихось інших завдань.

Прямий доступ до пам'яті (ПДП, DMA) - це режим, що принципово відрізняється від двох раніше розглянутих режимів тим, що обмін по системній шині відбувається без участі МП. Зовнішній пристрій, що вимагає обслуговування, сигналізує МП, що режим ПДП необхідний, у відповідь на це процесор закінчує виконання поточної команди і відключається від всіх шин, сигналізуючи зовнішньому пристрою, що обмін в режимі ПДП можна починати.



Операція ПДП зводиться до пересилки інформації з пристрою введення/виведення в пам'ять або ж з пам'яті в пристрій введення/виведення. Коли пересилка інформації буде закінчена, процесор знов повертається до перерваної програми, продовжуючи її з тієї точки, де його перервали (рис. 1.11). Це схоже на режим обслуговування переривань, але в даному випадку процесор не бере участь в обміні. Як і у випадку переривань, реакція на зовнішню подію при ПДП істотно повільніша, ніж при програмному режимі.

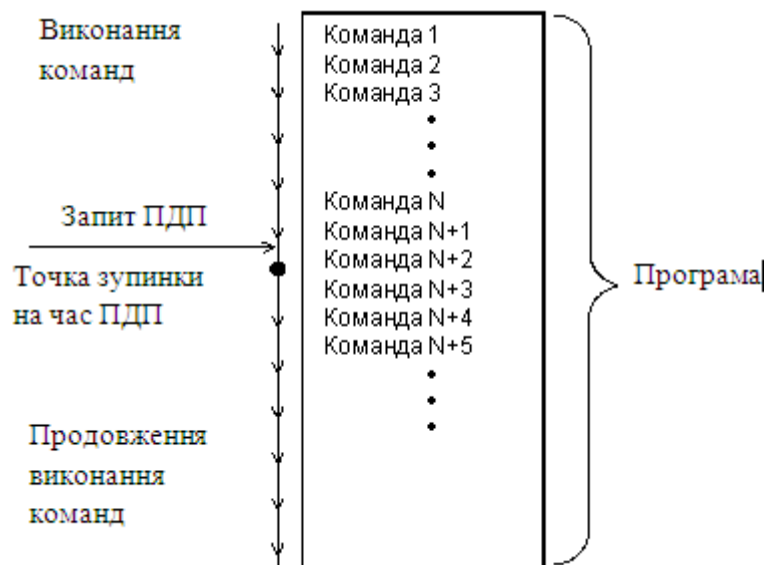


Рисунок 1.11 - Обслуговування ПДП

Зрозуміло, що в цьому випадку потрібне введення в систему додаткового пристрою (контролера ПДП), який здійснюватиме повноцінний обмін по системній магістралі без жодної участі МП. Причому процесор заздалегідь повинен повідомити контролер ПДП, звідки йому слід брати інформацію і/або куди її слід поміщати. Контролер ПДП може вважатися спеціалізованим процесором, який відрізняється тим, що сам не бере участь в обміні, не приймає інформацію і не видає її (рисунок 1.12).

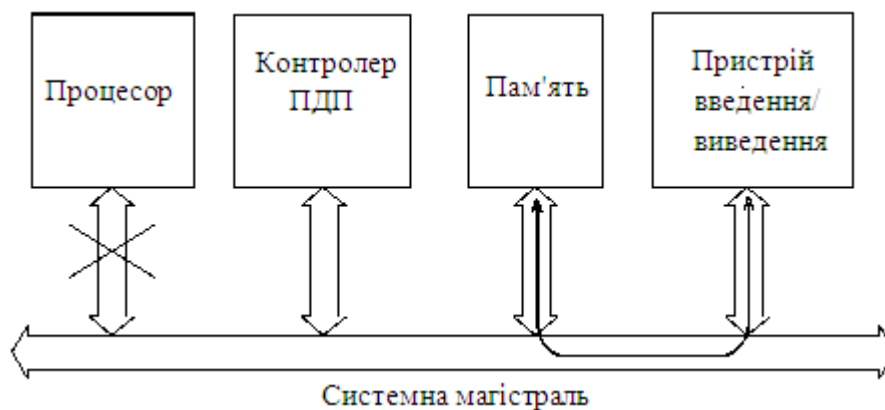


Рисунок 1.12 - Інформаційні потоки в режимі ПДП

В принципі контролер ПДП може входити до складу пристрою введення/виведення, якому необхідний режим ПДП або навіть до складу

декількох пристроїв введення/виведення. Теоретично обмін за допомогою ПДП може забезпечити вищу швидкість передачі інформації, чим програмний обмін, оскільки процесор передає дані повільніше, ніж спеціалізований контролер ПДП. Проте на практиці ця перевага реалізується далеко не завжди. Швидкість обміну в режимі ПДП зазвичай обмежена можливостями магістралі. До того ж необхідність програмного задання режимів контролера ПДП може звести нанівець вигоду від вищої швидкості пересилки даних в режимі ПДП. Тому режим ПДП застосовується рідко.

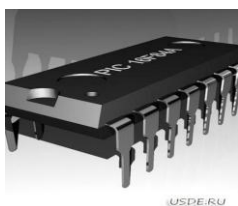
Якщо в системі вже є самостійний контролер ПДП, то це може у ряді випадків істотно спростити апаратуру пристроїв введення/виведення, що працюють в режимі ПДП. У цьому полягає єдина безперечна перевага режиму ПДП.

### 1.5. Типи мікропроцесорних систем

Діапазон застосування мікропроцесорної техніки дуже широкий з різними вимогами до МПС. Тому сформувалося декілька типів МПС, що розрізняються потужністю, універсальністю, швидкодією і структурними відмінностями. Основні типи наступні:

- мікроконтролери - найбільш простий тип МПС, в яких всі, або більшість вузлів системи виконані у вигляді однієї мікросхеми;
- контролери - мікропроцесорні системи, що управляють, виконані у вигляді окремих модулів;
- мікрокомп'ютери – потужні МПС з розвиненими засобами спряження із зовнішніми пристроями.
- комп'ютери (зокрема персональні) - найпотужніші і найбільш універсальні МПС.

Чітку межу між цими типами іноді провести досить складно. Швидкодія всіх типів мікропроцесорів постійно росте, і часто виникають ситуації, коли новий мікроконтролер виявляється швидшим за старішого персонального комп'ютера. Але деякі принципові відмінності все-таки є.



Мікроконтролерами є універсальні пристрої, які практично завжди використовуються не самі по собі, а у складі складніших пристроїв, в тому числі і контролерів. Системна шина мікроконтролера прихована від користувача всередині мікросхеми. Можливості підключення зовнішніх пристроїв до мікроконтролера обмежені. Пристрої на мікроконтролерах зазвичай призначені для вирішення одного завдання.



Контролери створюються для вирішення якогось окремого завдання або групи близьких завдань. Вони зазвичай не мають можливостей підключення додаткових вузлів і пристроїв, наприклад, великої пам'яті, засобів введення/виведення. Їх системна шина найчастіше недоступна користувачеві. Структура контролера

проста і оптимізована під максимальну швидкодію. В більшості випадків виконувані програми зберігаються в постійній пам'яті і не міняються. Конструктивно контроллери випускаються в одноплатному варіанті.



Мікрокомп'ютери відрізняються від контролерів відкритішою структурою, вони допускають підключення до системної шини декількох додаткових пристроїв. Мікрокомп'ютери можуть мати засоби зберігання інформації на магнітних носіях (магнітні диски) і досить потужні засоби зв'язку з користувачем (відеомонітор, клавіатура). Мікрокомп'ютери розраховані на широкий круг завдань, але на відміну від контролерів, до кожного нового завдання його треба пристосовувати наново. Програми, що виконуються мікрокомп'ютером можна легко міняти.

Нарешті, комп'ютери і найпоширеніші з них - персональні комп'ютери - це самі універсальні з МПС. Вони обов'язково передбачають можливість модернізації, а також широкі можливості підключення нових пристроїв. Їх системна шина, звичайно, доступна користувачеві. Крім того, зовнішні пристрої можуть підключатися до комп'ютера через декілька вбудованих портів



зв'язку (кількість портів доходить іноді до 10). Комп'ютер завжди має засоби зв'язку з користувачем, засоби тривалого зберігання інформації великого об'єму, засоби зв'язку з іншими комп'ютерами по інформаційних мережах. Області застосування комп'ютерів можуть бути самими різними: математичні розрахунки, обслуговування доступу до баз даних, управління роботою складних електронних систем, комп'ютерні ігри, підготовка документів і так далі.

Будь-яке завдання в принципі можна виконати за допомогою кожного з перерахованих типів МПС. Але при виборі типу треба по можливості уникати надмірності і передбачати необхідну для даного завдання гнучкість системи.

В даний час при розробці нових МПС найчастіше вибирають шлях використання мікроконтролерів (приблизно у 80% випадків). При цьому мікроконтролери застосовуються або самостійно, з мінімальною додатковою апаратурою, або у складі складніших контролерів з засобами введення/виведення.

Класичні МПС на базі мікросхем процесорів і мікропроцесорних комплектів випускаються зараз досить рідко, в першу чергу, із-за складності процесу розробки і відлагодження цих систем. Даний тип мікропроцесорних систем вибирають в основному тоді, коли мікроконтролери не можуть забезпечити необхідних характеристик.

Нарешті, помітне місце займають зараз МПС на основі персонального комп'ютера. Розробникові в цьому випадку потрібно тільки оснастити персональний комп'ютер додатковими пристроями спряження, а ядро МПС вже готове. Персональний комп'ютер має потужні засоби програмування, що

істотно спрощує завдання розробника. До того ж він може виконувати найскладніші алгоритми обробки інформації. Основні недоліки персонального комп'ютера - великі розміри корпусу і апаратурна надмірність для простих завдань. Недоліком є непристосованість більшості персональних комп'ютерів до роботи в складних умовах (запиленість, висока вологість, вібрації, високі температури і так далі). Проте випускаються і спеціальні персональні комп'ютери, пристосовані до різних умов експлуатації.

## Тестові питання до теми 1

### 1. Що таке мікропроцесорна система?

- А) система аналогового керування та оброблення інформації;
- Б) система цифрового керування та оброблення інформації;
- В) комунікаційна система для оброблення вхідних сигналів та видачі вихідних сигналів;
- Г) система управління сигналами.

### 2. Всередині мікропроцесорної системи відбувається :

- А) видача результатів обробки сигналів;
- Б) перетворення аналогових сигналів на цифрові;
- В) зберігання та накопичення сигналів;
- Г) перетворення цифрових сигналів на аналогові.

### 3. Характерною особливістю традиційної цифрової системи є:

- А) кожний компонент системи працює в автономному режимі;
- Б) алгоритми оброблення і зберігання інформації не пов'язані з схемотехнікою системи;
- В) швидке оброблення сигналів усіма компонентами системи;
- Г) алгоритми оброблення і зберігання інформації жорстко пов'язані з схемотехнікою системи.

### 4. Будь-яка система на «жорсткій логіці» є:

- А) спеціалізованою системою, налаштованою виключно на одне задалегідь відоме завдання;
- Б) універсальною системою, придатною для виконання багатьох завдань;
- В) автономною системою, призначеною для виконання багатьох завдань;
- Г) універсальною системою, налаштованою виключно на конкретне завдання.

### 5. Виберіть правильний варіант, який має стояти на місці «?»:



- А) текстові дані;
- Б) звукові сигнали;
- В) команди;
- Г) цифрові коди.

**6. Переваги МПС на «жорсткій логіці»:**

- А) універсальність програми та використання під абсолютно будь-яке завдання;
- Б) немає апаратної надмірності;
- В) низька швидкодія;
- Г) максимально висока швидкодія.

**7. Недоліки МПС на «жорсткій логіці» усуваються шляхом:**

- А) введення в структуру системи необхідної програми;
- Б) заміни архітектури системи;
- В) зменшення кількості розв'язуваних завдань;
- Г) всі відповіді вірні.

**8. Результатом універсальності є:**

- А) надмірність системи;
- Б) збільшення швидкодії;
- В) зниження швидкодії;
- Г) збільшення надійності системи.

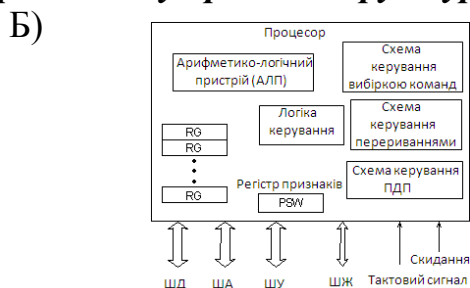
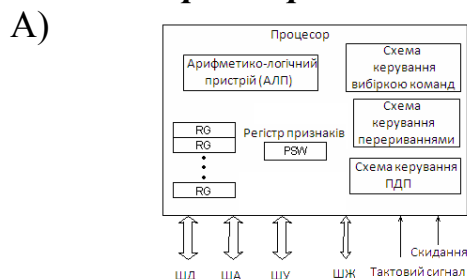
**9. МПС на «жорсткій логіці» використовуються там де:**

- А) завдання змінюється дуже часто;
- Б) потрібна незначна швидкодія;
- В) алгоритми оброблення інформації гранично прості;
- Г) потрібна висока швидкодія.

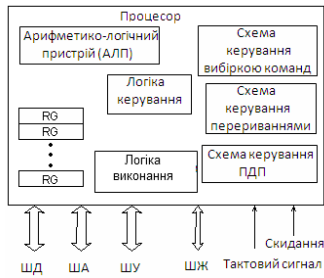
**10. Універсальні МПС використовуються там де:**

- А) завдання змінюється дуже часто;
- Б) потрібна висока швидкодія;
- В) алгоритми обробки інформації складні;
- Г) зміна завдань незначна.

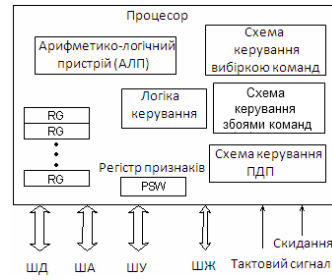
**11. Виберіть правильний варіант. Внутрішня структура мікропроцесора.**



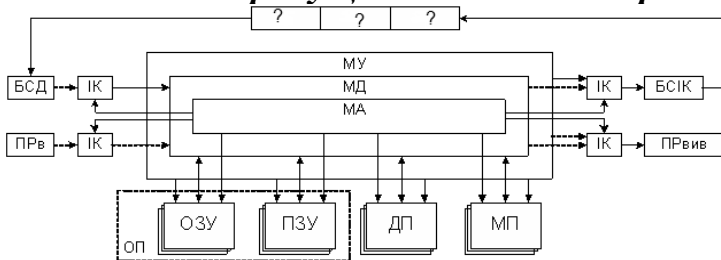
В)



Г)



12. Вставте пропущені елементи зображені на логічній структурі МПС «?».



- А) об'єкт управління, виконавчі механізми, реєстри керування;
- Б) об'єкт управління, реєстри читання, реєстри запису;
- В) датчики, об'єкт управління, виконавчі механізми;
- Г) керуючі блоки, датчики, об'єкт управління, .

13. Пристрій сполучення апаратури введення-виведення з системною магистраллю, виконаний у вигляді окремих модулів називається:

- А) мікроконтролер;
- Б) ПЛМ;
- В) процесор;
- Г) контролер.

14. Чим відрізняються різні типи МПС?

- А) потужністю;
- Б) універсальністю;
- В) швидкодією і структурними відмінностями;
- Г) всі вище перелічені відповіді.

15. Скільки існує основних типів МПС?

- А) 5;
- Б) 4;
- В) 2;
- Г) 8.

16. Мікроконтролери це - ...

- А) найбільш простий тип МПС, в яких всі, або більшість вузлів системи виконані у вигляді однієї мікросхеми;
- Б) найбільш простий тип МПС, в яких всі, або більшість вузлів системи виконані у вигляді окремих модулів;

- В) однокристална мікроЕОМ виконана у вигляді мікросхеми;
- Г) потужні МПС з розвиненими засобами спряження із зовнішніми пристроями.

**17. Контролери це - ...**

- А) потужні МПС з розвиненими засобами спряження із зовнішніми пристроями;
- Б) найбільш простий тип МПС;
- В) мікропроцесорні системи, що управляють, виконані у вигляді окремих модулів;
- Г) найбільш простий тип МПС, в яких всі, або більшість вузлів системи виконані у вигляді однієї мікросхеми.

**18. Що з переліченого не являється основним типом МПС?**

- А) мікроконтролери;
- Б) мікродатчики;
- В) мікрокомп'ютери;
- Г) ВІС.

**19. Контролери створюються для...**

- А) вирішення конкретних підзадач у складі потужних систем;
- Б) вирішення якогось окремого завдання або групи близьких завдань;
- В) для контролю над групою процесорів які вирішують окремі задачі;
- Г) правильної відповіді немає.

**20. Мікрокомп'ютери відрізняються від контролерів ...**

- А) наявністю вбудованого датчика управління;
- Б) відкритішою структурою
- В) наявністю засобів зв'язку з користувачем
- Г) всі відповіді правильні

**21. Основні недоліки персонального комп'ютера:**

- А) великі розміри корпусу;
- Б) апаратна надмірність для простих завдань;
- В) висока швидкодія;
- Г) пристосованість до складних умов роботи.

**22. Пристрої на мікроконтролерах ...**

- А) зазвичай призначені для вирішення групи завдань;
- Б) відсутні;
- В) призначені для роботи з паралельними обчисленнями;
- Г) зазвичай призначені для вирішення одного завдання.

**23. Шинна організація зв'язків МПС...**

- А) дозволяє досить легко замінювати апаратні модулі;
- Б) не допускає зміни конфігурації;

- В) дозволяє модернізацію;
- Г) робить неможливою заміну апаратних модулів.

**24. Скільки основних режимів обміну по магістралі підтримує сучасна МПС**

- А) 5;
- Б) 1;
- В) 6;
- Г) 3.

**25. Які існують основні режими обміну по магістралі?**

- А) програмний обмін інформацією;
- Б) обмін з використанням переривань;
- В) обмін з використанням прямого доступу до пам'яті;
- Г) всі перелічені.

**26. Скількома шляхами можна організувати реакцію на зовнішню подію?**

- А) 3;
- Б) 1;
- В) 6;
- Г) 2.

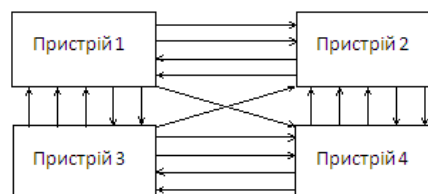
**27. Організувати реакцію на зовнішню подію можна такими шляхами:**

- А) за допомогою постійного апаратного таймера;
- Б) за допомогою переривання;
- В) за допомогою прямого доступу до пам'яті;
- Г) за допомогою не прямого доступу до пам'яті.

**28. IRQ — Interrupt ReQuest це —**

- А) переривання від зовнішнього пристрою;
- Б) сигнальне переривання від RTC;
- В) періодичне переривання від Interrupt ;
- Г) відсутня вірна відповідь.

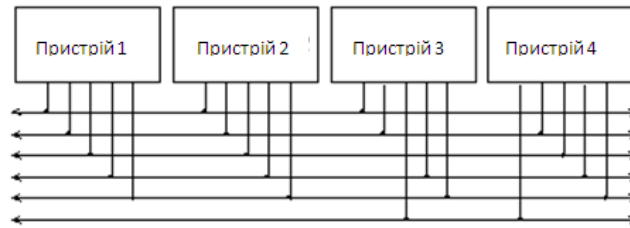
**29. При класичній структурі зв'язків всі сигнали і коди між пристроями передаються по:**



- А) шині;
- Б) окремих лініях зв'язку;
- В) двох основних ліній зв'язку;
- Г) магістралі.



**30. При шинній структурі зв'язків всі сигнали між пристроями передаються по:**



- А) шині адреси;
- Б) окремих лініях зв'язку;
- В) одних і тих же лініях зв'язку, але в різний час;
- Г) шині даних.

**31. Перевагами шинної структури зв'язків є:**

- А) підключення всіх пристроїв до одної шини;
- Б) приймання і передавання інформацію по одних і тих же протоколах;
- В) підключення всіх пристроїв до кожної лінії зв'язку паралельно;
- Г) приймання і передавання інформацію по різних протоколах.

**32. Скільки основних шин нижнього рівня включає системна магістраль:**

- А) 5;
- Б) 4;
- В) 2;
- Г) 8.

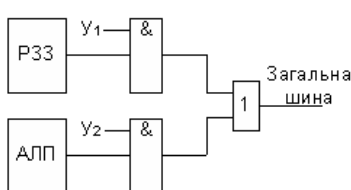
**33. Якої з основних шин нижнього рівня не існує?**

- А) шини адреси;
- Б) шини даних;
- В) шини управління;
- Г) шини переривань.

**34. Шина живлення процесора має:**

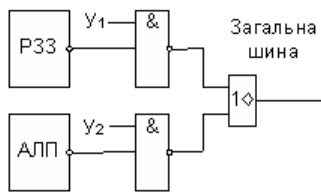
- А) напругу живлення 4 В;
- Б) напругу живлення 5 В або 3,3 В;
- В) напругу живлення 7 В;
- Г) напругу живлення 3 В.

**35. Який спосіб підключення пристроїв до системної магістралі зображений на малюнку?**



- А) об'єднання за допомогою схем з відкритим колектором;
- Б) об'єднання за допомогою схем з закритим колектором;
- В) об'єднання за допомогою схем з трьома станами;
- Г) логічне об'єднання.

**36. Який спосіб підключення пристроїв до системної магістралі зображений на малюнку?**

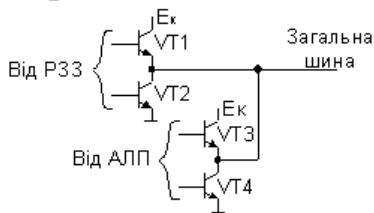


- А) об'єднання за допомогою схем з відкритим колектором;
- Б) об'єднання за допомогою схем з закритим колектором;
- В) об'єднання за допомогою схем з трьома станами;
- Г) логічне об'єднання.

**37. Шина адрес:**

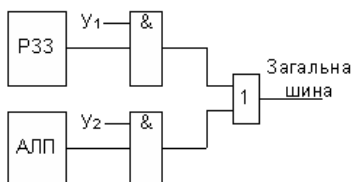
- А) служить для визначення адреси (номера) пристрою, з яким МП обмінюється інформацією в даний момент;
- Б) визначає максимально можливу складність мікропроцесорної системи;
- В) визначає швидкість оброблення адрес мікропроцесорною системою;
- Г) правильна відповідь відсутня.

**38. Який спосіб підключення пристроїв до системної магістралі зображений на малюнку?**



- А) об'єднання за допомогою схем з відкритим колектором;
- Б) об'єднання за допомогою схем з закритим колектором;
- В) об'єднання за допомогою схем з трьома станами;
- Г) логічне об'єднання.

**39. Об'єднання зображене на рисунку використовує в собі схеми:**



- А) АБО
- Б) І
- В) АБО-НЕ
- Г) І-НЕ

**40. Шина управління це...**

- А) допоміжна шина, керуючі сигнали на якій визначають тип адрес команд управління;
- Б) шина по якій передаються дані про адреси команд управління;
- В) шина по якій передаються станини об'єктів керування;
- Г) правильна відповідь відсутня.

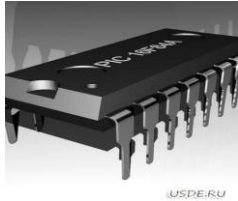
**41. Обмін по перериваннях використовується:**

- А) коли необхідна реакція МПС на якусь зовнішню подію;
- Б) при подачі внутрішнього сигналу;
- В) при збої виконання операції;
- Г) коли це зручно програмісту.

**42. Організація МП на основі однієї шини вимагає введення ...**

- А) двох-трьох буферних регістрів;
- Б) трьох тактів для виконання операцій;
- В) додаткових лічильників;
- Г) всі відповіді вірні.

**43. Який тип МПС зображений на малюнку?**



- А) комп'ютер;
- Б) контролер;
- В) мікроконтролер;
- Г) мікрокомп'ютер.

**44. Яку напругу живлення використовують у МПС ?**

- А) +5 В; +12В;
- Б) +3,5; - 5В;
- В) +5; -5 В ;
- Г) +12 В; -12 В.

**45. Мікропроцесор це:**

- А) функціонально-складний програмно - керуючий цифровий процесор;
- Б) інтегральна схема з високим ступенем інтеграції елементів;
- В) електронний прилад;
- Г) всі відповіді вірні.

**46. МП класифікують по наступних ознаках:**

- А) технологія виготовлення, за призначенням, числом шин;
- Б) об'ємом пам'яті програм та даних;
- В) кількості акумуляторних НВІС;
- Г) характером роботи алгоритмів введення/виведення.

**47. За характером роботи мікропроцесори поділяються на:**

- А) універсальні;
- Б) синхронні;
- В) асинхронні;
- Г) спеціалізовані.

**48. За способом управління мікропроцесори поділяють на:**

- А) контролюючі;
- Б) командно-керуючі;
- В) системні;
- Г) мікропрограмні.

**49. Структура та об'єм системи команд МП визначають його:**

- А) надійність;

- Б) гнучкість;
- В) швидкодію;
- Г) апаратні вимоги.

**50. Розробник МПС повинен знати:**

- А) всю внутрішню структуру та наповнення процесора;
- Б) протоколи обміну інформацією;
- В) систему команд, режими роботи процесора.
- Г) всі відповіді вірні.

## ТЕМА 2 – ФУНКЦІЇ ПРИСТРОЇВ МІКРОПРОЦЕСОРНОЇ СИСТЕМИ

### 2.1. Функції процесора

Процесор - пристрій управління, який генерує послідовність синхронізуючих і логічних сигналів, що визначають послідовності спрацьовування всіх логічних пристроїв МПС. Окрім цього мікропроцесор задає і послідовно здійснює мікрооперації витягання команд програми з пам'яті системи, їх розшифровку і виконання, тим самим виконуючи арифметичні, логічні або інші операції над числами.

Процесор (рисунок 2.1) являє собою окрему мікросхему або ж частину мікросхеми (в випадку мікроконтролера). У попередні роки процесор реалізовувався на комплектах з декількох мікросхем, але зараз від такого підходу вже практично відмовилися. Мікросхема процесора обов'язково має виводи трьох шин: шини адреси, шини даних і шини управління. Іноді деякі сигнали і шини мультиплекуються, щоб зменшити кількість ліній спряження мікросхеми процесора.

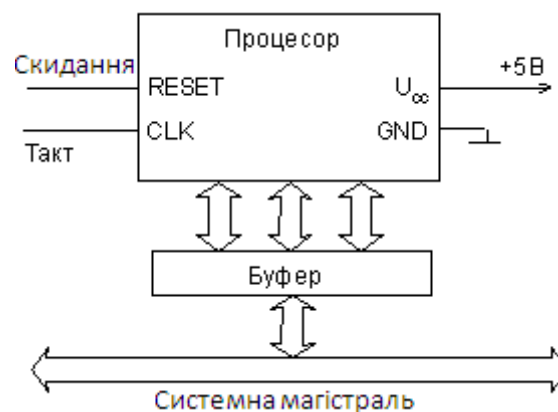


Рисунок 2.1 – Схема включення мікропроцесора до магістралі

Найважливішими характеристиками процесора - кількість розрядів його шини даних, кількість розрядів його шини адреси і кількість керуючих сигналів у шині управління. Розрядність шини даних визначає швидкість роботи системи. Розрядність шини адреси визначає допустиму складність системи. Кількість ліній управління визначає різноманітність режимів обміну та ефективність обміну процесора з іншими пристроями системи.

Крім ліній спряження з трьома основними шинами процесор завжди має лінію виведення (або дві лінії виведення) для підключення зовнішнього тактового сигналу або кварцового резонатора (CLK), тому що процесор завжди являє собою тактуючий пристрій. Чим більше тактова частота процесора, тим швидше він працює і тим швидше виконуються команди. Втім, швидкодія процесора визначається не тільки тактовою частотою, але й особливостями його структури. Сучасні процесори виконують більшість команд за один такт і мають засоби для паралельного виконання декількох команд. Тактова частота

процесора не пов'язана зі швидкістю обміну по магістралі, оскільки швидкість обміну по магістралі обмежена затримками поширення сигналів і спотвореннями сигналів на магістралі. Тобто тактова частота процесора визначає лише його внутрішню швидкодію. Іноді тактова частота процесора має нижню та верхню межу. При перевищенні верхньої межі частоти можливо перегрівання процесора, а також збої у його роботі.

Ще один важливий сигнал, який є в кожному процесорі, - це сигнал початкового скидання RESET. При включенні живлення, під час аварійної ситуації або зависанні процесора подача цього сигналу приводить до ініціалізації процесора, змушує його приступити до виконання програми початкового запуску. Аварійна ситуація може бути викликана перешкодами по ланцюгах живлення і "землі", збоями в роботі пам'яті, зовнішніми іонізуючими випромінюваннями та іншими причинами. У результаті процесор може втратити контроль над виконуваною програмою і зупинитися на якійсь адресі. Для виходу з цього стану використовується сигнал початкового скидання. Цей же вхід початкового скидання може використовуватися для оповіщення процесора про те, що напруга живлення стала нижчою за встановлену межу. У такому випадку процесор переходить до виконання програми збереження важливих даних.

Іноді у мікросхеми процесора є ще один-два входи радіальних переривань для обробки особливих ситуацій (наприклад, для переривання від зовнішнього таймера). Шина живлення сучасного процесора зазвичай має одну напругу живлення (5 В або 3,3 В) і загальний провід ("землю"). У деяких процесорах передбачений режим зниженого енергоспоживання. Взагалі, сучасні мікросхеми процесорів, особливо з високими тактовими частотами, споживають досить велику потужність. В результаті для підтримки нормальної робочої температури корпусу на них нерідко доводиться встановлювати радіатори, вентилятори або навіть спеціальні мікрохолодильники. Для підключення процесора до магістралі використовуються буферні мікросхеми, що забезпечують, якщо необхідно, демультимплексування сигналів та електричну буферизацію сигналів магістралі. Іноді протоколи обміну системної магістралі та шин процесора не збігаються між собою, тоді буферні мікросхеми узгоджують ці протоколи. Іноді в МПС використовується кілька магістралей (системних та локальних), тоді для кожної з магістралей застосовується свій буферний вузол. Така структура характерна, наприклад, для персональних комп'ютерів.

Після включення живлення процесор переходить в першу адресу програми початкового пуску і виконує цю програму. Дана програма попередньо записана в постійну (енергонезалежну) пам'ять. Після завершення програми початкового пуску процесор починає виконувати основну програму, що знаходиться в постійній або оперативній пам'яті, для чого вибирає по черзі всі команди. Від цієї програми процесор можуть відволікати зовнішні переривання або запити на ПДП. Команди з пам'яті процесор вибирає за допомогою циклів читання по магістралі. При необхідності процесор записує дані в пам'ять або в

пристрої введення-виведення за допомогою циклів запису або ж читає дані з пам'яті або з пристроїв введення-виведення за допомогою циклів читання.

Таким чином, основні функції будь-якого процесора наступні:

- вибірка (читання) виконуваних команд;
- введення (читання) даних з пам'яті або пристрою введення-виведення;
- виведення (запис) даних в пам'ять або в пристрої введення-виведення;
- обробка даних (операндів), в тому числі арифметичні операції над ними;
- адресація пам'яті, тобто задання адреси пам'яті, з яким буде проводитися обмін;
- обробка переривань і режиму прямого доступу.

Спрощено структуру МП можна представити в наступному вигляді (рисунок 2.2).

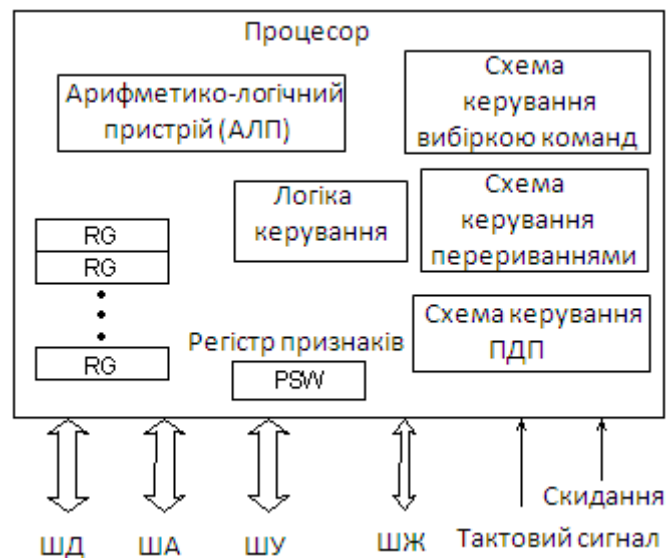


Рисунок 2.2 - Внутрішня структура МП

Основні функції показаних на рисунку 2.2 вузлів МП наступні.

Схема керування вибіркою команд виконує читання команд з пам'яті і їх дешифрування. У перших МП було неможливо одночасне виконання попередньої команди та вибірка наступної команди, тому що МП не міг поєднувати ці операції. Але вже в 16-розрядних процесорах з'являється так званий конвеєр (черга) команд, що дозволяє вибирати кілька наступних команд, поки виконується попередня. Два процеси йдуть паралельно, що прискорює роботу процесора. Конвеєр являє собою невелику внутрішню пам'ять процесора, в яку при найменшій можливості (при звільненні зовнішньої шини) записується декілька команд. Читаються ці команди процесором в тому ж порядку, що і записуються в конвеєр (це пам'ять типу FIFO, First In - First Out, першим увійшов - першим вийшов). Правда, якщо виконується команда передбачає перехід не на наступну комірку пам'яті, а на віддалену (з меншим

або більшим адресом), конвеєр не допомагає, і його доводиться скидати. Але такі команди зустрічаються в програмах порівняно рідко.

Розвитком ідеї конвеєра стало використання внутрішньої кеш-пам'яті процесора, яка заповнюється командами, поки процесор зайнятий виконанням попередніх команд. Чим більший об'єм кеш-пам'яті, тим менша ймовірність того, що її вміст доведеться скинути при команді переходу. Зрозуміло, що обробляти команди, що знаходяться у внутрішній пам'яті, процесор може набагато швидше, ніж ті, що розташовані у зовнішній пам'яті. У кеш-пам'яті можуть зберігатися і дані, які обробляються в даний момент, це також прискорює роботу. Для більшого прискорення вибірки команд у сучасних процесорах застосовують поєднання вибірки та дешифрування, одночасне дешифрування кількох команд, кілька паралельних конвеєрів команд, передбачення команд переходів і деякі інші методи.

Арифметико-логічний пристрій (АЛП) призначений для обробки інформації відповідно до отриманої процесором командою. Прикладами обробки можуть бути логічні операції (типу логічного "І", "АБО" і т.д.) тобто побітні операції над операндами, а також арифметичні операції (типу додавання, віднімання, множення, ділення і т. д.). Над якими кодами проводиться операція, куди поміщається її результат - визначається виконуваною командою. Якщо команда зводиться лише до пересилання даних без їх обробки, то АЛП не бере участі в її виконанні. Швидкодія АЛП багато в чому визначає продуктивність процесора. Причому важлива не тільки частота тактового сигналу, яким тактується АЛП, але і кількість тактів, необхідна для виконання тієї чи іншої команди. Для підвищення продуктивності розробники прагнуть довести час виконання команди до одного такту, а також забезпечити роботу АЛП на якомога більш високій частоті. Один із шляхів вирішення цього завдання полягає у зменшенні кількості виконуваних команд АЛП, створення процесорів зі зменшеним набором команд (так звані RISC-процесори). Інший шлях підвищення продуктивності процесора - використання декількох паралельно працюючих АЛП.

Що стосується операцій над числами з плаваючою точкою і інших спеціальних складних операцій, то в системах на базі перших процесорів їх реалізовували послідовністю простіших команд, спеціальними підпрограмами, проте потім були розроблені спеціальні обчислювачі — математичні сопроцесори, які заміняли основний процесор на час виконання таких команд. Всі сучасні МП містять математичні сопроцесори.

Регістри процесора є по суті елементами дуже швидкої пам'яті і служать для тимчасового зберігання різних кодів: даних, адрес, службових кодів. Операції з цими кодами виконуються досить швидко, тому чим більше внутрішніх регістрів, тим краще. Крім того, на швидкодію процесора сильно впливає розрядність регістрів. Саме розрядність регістрів і АЛП називається внутрішньою розрядністю процесора, яка може не співпадати із зовнішньою розрядністю.

По відношенню до призначення внутрішніх регістрів існує два основні підходи. Першого дотримується, наприклад, компанія Intel, яка кожному



регістру відводить строго певну функцію. З одного боку, це спрощує організацію процесора і зменшує час виконання команди, але з іншого - знижує гнучкість, а іноді і сповільнює роботу програми. Наприклад, деякі арифметичні операції і обмін з пристроями введення-виведення проводяться тільки через один регістр - акумулятор, внаслідок чого при виконанні деяких процедур потрібно декілька додаткових пересилок між регістрами. Другий підхід полягає в тому, щоб всі (або майже всі) регістри роблять рівноправними, як, наприклад, в 16-розрядних процесорах T-11 фірми DEC. При цьому досягається висока гнучкість, але необхідне ускладнення структури процесора. Існують і проміжні рішення, зокрема, в процесорі MC68000 фірми Motorola половина регістрів використовувалася для даних, і вони взаємозамінні, а інша половина - для адрес, і вони також взаємозамінні.

Регістр признаков (регістр стану) займає особливе місце, хоча він також є внутрішнім регістром процесора. Інформація, що міститься в ньому - слово стану процесора (CCP, PSW — Processor Status Word). Кожен біт цього слова (прапор) містить інформацію про результат попередньої команди. Наприклад, є біт нульового результату, який встановлюється у тому випадку, коли результат виконання попередньої команди - нуль, і очищається у тому випадку, коли результат виконання команди відмінний від нуля. Ці біти (прапори) використовуються командами умовних переходів, наприклад, командою переходу у разі нульового результату. У цьому ж регістрі іноді містяться прапори управління, що визначають режим виконання деяких команд.

Схема керування перериваннями обробляє запит переривання, що поступає на процесор, визначає адресу початку програми обробки переривання (адреса вектора переривання), забезпечує перехід до цієї програми після виконання поточної команди і збереження в пам'яті (у стеку) поточного стану регістрів процесора. Після закінчення програми обробки переривання процесор повертається до перерваної програми з відновленими з пам'яті (із стека) значеннями внутрішніх регістрів.

Схема управління прямим доступом до пам'яті служить для тимчасового відключення процесора від зовнішніх шин і припинення роботи процесора на час надання прямого доступу пристрою, що запитав його.

Логіка керування організовує взаємодію всіх вузлів процесора, перенаправляє дані, синхронізує роботу процесора із зовнішніми сигналами, а також реалізує процедури введення і виведення інформації.

Таким чином, в ході роботи процесора схема керування вибіркою команд вибирає послідовно команди з пам'яті, потім ці команди виконуються, причому у разі потреби обробки даних підключається АЛП. На входи АЛП можуть подаватися оброблювані дані з пам'яті або з внутрішніх регістрів. У внутрішніх регістрах зберігаються також коди адрес оброблюваних даних, розташованих в пам'яті. Результат обробки в АЛП змінює стан регістра ознак і записується у внутрішній регістр або в пам'ять (як джерело, так і приймач даних вказується у складі коду команди). При необхідності інформація може переписуватися з пам'яті (або з пристрою введення/виведення) у внутрішній регістр або з внутрішнього регістра в пам'ять (або в пристрій введення/виведення).

Внутрішні реєстри будь-якого мікропроцесора обов'язково виконують дві службові функції:

- визначають адресу в пам'яті, де знаходиться виконувана в даний момент команда (функція лічильника команд або покажчика команд);
- визначають поточну адресу стека (функція покажчика стека).

У різних процесорах для кожної з цих функцій може відводитися один або два внутрішні реєстри. Ці два покажчики відрізняються від інших не тільки своїм специфічним, службовим, системним призначенням, але і особливим способом зміни вмісту. Їх вміст програми можуть міняти тільки у разі крайньої необхідності, оскільки будь-яка помилка при цьому загрожує порушенням роботи комп'ютера, зависанням і псуванням вмісту пам'яті.

Вміст покажчика (лічильника) команд змінюється таким чином. На початку роботи системи (при включенні живлення) в нього заноситься раз і назавжди встановлене значення. Це перша адреса програми початкового запуску. Потім після вибірки з пам'яті кожної наступної команди значення покажчика команд автоматично збільшується (інкрементується) на одиницю (або на два залежно від формату команд і типу процесора). Тобто наступна команда вибиратиметься з наступної по порядку адреси пам'яті. При виконанні команд переходу, що порушують послідовний перебір адрес пам'яті, в покажчик команд примусово записується нове значення - нова адреса в пам'яті, починаючи з якої адреси команд знову ж таки перебиратимуться послідовно. Така ж зміна вмісту покажчика команд проводиться при виклику підпрограми і поверненні з неї, або при початку обробки переривання і після його закінчення.

## 2.2. Функції пам'яті

Пам'ять МПС виконує функцію тимчасового або постійного зберігання даних і команд. Об'єм пам'яті визначає допустиму складність виконуваних системою алгоритмів, а також в деякій мірі і швидкість роботи системи в цілому. Модулі пам'яті виконуються на мікросхемах пам'яті (оперативної або постійної). Все частіше у складі мікропроцесорних систем використовується флеш-пам'ять (англ. - flash memory), яка є незалежною пам'яттю з можливістю багатократного перезапису вмісту.

Інформація в пам'яті зберігається в комірках, кількість розрядів яких рівна кількості розрядів шини даних процесора. Зазвичай вони кратні вісьми (наприклад, 8, 16, 32, 64). Допустима кількість елементів пам'яті визначається кількістю розрядів шини адреси як  $2^N$ , де  $N$  — кількість розрядів шини адреси. Найчастіше об'єм пам'яті вимірюється в байтах незалежно від розрядності елементу пам'яті. Використовуються також наступні одиниці об'єму пам'яті: кілобайт —  $2^{10}$  або 1024 байти (позначається Кбайт), мегабайт —  $2^{20}$  або 1 048 576 байт (позначається Мбайт), гігабайт —  $2^{30}$  байт (позначається Гбайт), терабайт —  $2^{40}$  (позначається Тбайт) Наприклад, якщо пам'ять має 65 536 комірок, кожна з яких 16-розрядна, то говорять, що пам'ять має об'єм 128

Кбайт. Сукупність елементів пам'яті називається зазвичай простором пам'яті системи.

Для підключення модуля пам'яті до системної магістралі використовуються блоки спряження, які включають дешифратор (селектор) адреси, схему обробки керуючих сигналів магістралі і буфери даних (рисунок 2.3).

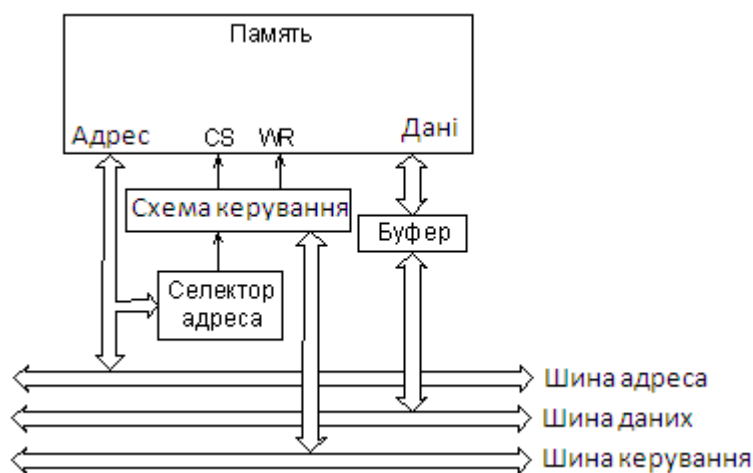


Рисунок 2.3 - Структура модуля пам'яті

Оперативна пам'ять взаємодіє з системною магістраллю в циклах читання і запису, постійна пам'ять - тільки в циклах читання. Зазвичай у складі системи є декілька модулів пам'яті, кожен з яких працює в своїй області простору пам'яті. Селектор адреси визначає, яка область адрес простору пам'яті відведена даному модулю пам'яті. Схема керування виробляє в потрібні моменти сигнали дозволу роботи пам'яті (CS) і сигнали дозволу запису в пам'ять (WR). Буфери даних передають дані від пам'яті до магістралі або від магістралі до пам'яті.

У просторі пам'яті МПС виділяються декілька особливих областей, які виконують спеціальні функції.

Пам'ять програми початкового запуску завжди виконується на ПЗП або флеш-пам'яті. Саме з цієї області процесор починає роботу після включення живлення і після скидання його за допомогою сигналу RESET.

Пам'ять для стека або стек (Stack) - це частина оперативної пам'яті, призначена для тимчасового зберігання даних в режимі LIFO (Last In — First Out).

Особливість стека в порівнянні з іншою оперативною пам'яттю - це заданий і незмінний спосіб адресації. При записі будь-якого числа (коду) в стек число записується за адресою, що визначається як вміст регістра покажчика стека, заздалегідь зменшене на одиницю (або на дві, якщо 16-розрядні слова розташовані в пам'яті по парних адресах). При читанні із стека число читається з адреси, яка визначається вмістом покажчика стека, після чого це вміст покажчика стека збільшується на одиницю (або на дві). В результаті виходить, що число, записане останнім, буде прочитано першим, а число, записане першим, буде прочитано останнім. Така пам'ять називається LIFO або пам'яттю

магазинного типу (наприклад, в магазині автомата патрон, встановлений останнім, витягуватиме першим).

Принцип дії стека показаний на рисунку 2.4 (адреси елементів пам'яті вибрані умовно).

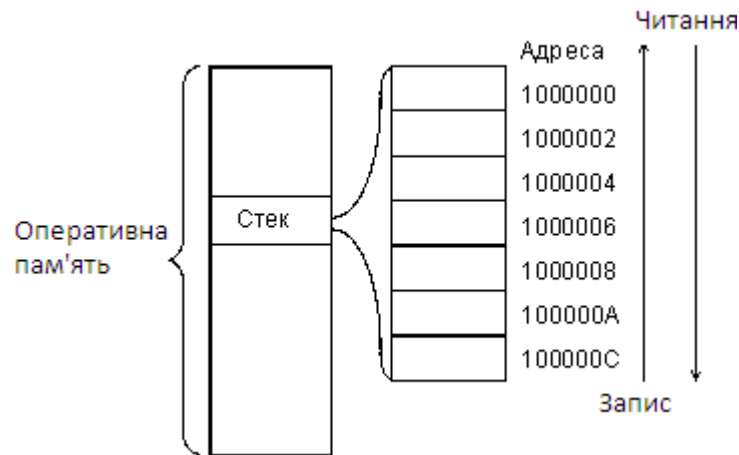


Рисунок 2.4 - Принцип роботи стека

Нехай, наприклад, поточний стан покажчика стека 1000008, і в нього треба записати два числа (слова). Перше слово буде записано за адресою 1000006 (перед записом покажчик стека зменшиться на два). Друге — за адресою 1000004. Після запису вміст покажчика стека — 1000004. Якщо потім прочитати із стека два слова, то першим буде прочитано слово з адреси 1000004, а після читання покажчик стека стане рівним 1000006. Другим буде прочитано слово з адреси 1000006, а покажчик стека стане рівним 1000008. Все повернулося до початкового стану. Перше записане слово читається другим, а друге — першим.

Необхідність такої адресації стає очевидною у разі багатократних вкладених підпрограм. Нехай, наприклад, виконується основна програма, і з неї викликається підпрограма 1. Якщо нам треба зберегти значення даних і внутрішніх реєстрів основної програми на час виконання підпрограми, ми перед викликом підпрограми збережемо їх в стеку (запишемо в стек), а після її закінчення прочитаємо їх із стека. Якщо ж з підпрограма 1 викликається підпрограмою 2, то ту ж саму операцію ми виконаємо з даними і вмістом внутрішніх реєстрів підпрограми 1. Зрозуміло, що всередині підпрограми 2 крайніми в стеку будуть дані з підпрограми 1, а дані з основної програми будуть глибші. При цьому у разі читання із стека автоматично дотримуватиметься потрібний порядок читання інформації. Те ж саме буде і у разі, коли таких рівнів вкладення підпрограм значно більше. Тобто те, що треба зберігати довше, ховається глибше, а те, що скоро знадобиться розміщується з краю.

У системі команд будь-якого процесора для обміну інформацією із стеком передбачені спеціальні команди запису в стек (PUSH) і читання із стека (POP). У стеку можна ховати не тільки вміст всіх внутрішніх реєстрів процесорів, але і вміст реєстра ознак (слово стану процесора, PSW). Це

дозволяє, наприклад, при поверненні з підпрограми контролювати результат останньої команди, виконаної безпосередньо перед викликом цієї підпрограми. Можна також зберігати в стеку і дані, для того, щоб зручніше було передавати їх між програмами і підпрограмами. У загальному випадку, чим більша область пам'яті, відведена під стек, тим більше свободи у програміста і тим складніші програми можуть виконуватися.

Наступна спеціальна область пам'яті — це таблиця векторів переривань.

Вектор – це повна адреса пам'яті (4 Байт) тієї процедури, що викликається з появою запиту на переривання. Таким чином, у таблиці векторів переривань у комірках пам'яті з 0 по 3 знаходиться адреса програми опрацювання переривання з номером 0, у комірках пам'яті з 4 по 7 – адреса програми опрацювання з номером 1 і т.д.

У IBM-сумісних комп'ютерах керування апаратними перериваннями здійснюється спеціальними мікросхемами – контролерами переривань. На сучасних материнських платах контролери переривань інтегровані в мікросхему контролера периферії Chipset.

Кожна лінія переривання має свій пріоритет. Найвищий пріоритет відповідає лінії IRQ0. Кожній лінії IRQ відповідає вектор переривань, що вказує місцезнаходження процедури обслуговування запиту на переривання в ROM BIOS. Лініям IRQ1-IRQ7 відповідають вектори переривань від 8H до 0FH, а лініям IRQ8-IRQ15 – вектори від 70H до 77H. Частина ліній переривань резервується системою, а частина може бути надана для використання додатковим устаткуванням, необхідним користувачеві. Так, наприклад, IRQ13 приділяється для математичного співпроцесора і воно не може бути використано іншим пристроєм, навіть якщо математичний співпроцесор не встановлений. Звичайно для IBM-сумісних комп'ютерів вільні IRQ3, IRQ5, IRQ7, IRQ10, IRQ11 і IRQ12. Стандартний розподіл апаратних переривань приведено в табл. 1.

Взагалі, поняття переривання досить багатозначне. Під перериванням в загальному випадку розуміється не тільки обслуговування запиту зовнішнього пристрою, але і будь-яке порушення послідовної роботи процесора. Наприклад, може бути передбачене переривання за фактом некоректного виконання арифметичної операції типу ділення на нуль. Або ж переривання може бути програмним, коли в програмі використовується команда переходу на якусь підпрограму, з якої потім слідує повернення в основну програму.

Будь-яке переривання обробляється через таблицю векторів (показчиків) переривань. У цій таблиці в простому випадку знаходяться адреси початку програм обробки переривань, які і називаються векторами. Довжина таблиці може бути досить великою (до декількох сотень елементів). Зазвичай таблиця векторів переривань розташовується на початку простору пам'яті (у елементах пам'яті з малими адресами). Адресою кожного вектора (або адреса початкового елемента кожного вектора) є номер переривання.

У разі апаратних переривань номер переривання або задається пристроєм, що запитав переривання (при векторних перериваннях), або ж задається номером лінії запиту переривань (при радіальних перериваннях). Процесор,

отримавши апаратне переривання, закінчує виконання поточної команди і звертається до пам'яті в область таблиці векторів переривань, в той її рядок, який визначається номером запитаного переривання. Потім процесор читає вміст цього рядка (код вектора переривання) і переходить в адресу пам'яті, що задається цим вектором. Починаючи з цієї адреси в пам'яті повинна розташовуватися програма обробки переривання з даним номером. В кінці програми обробки переривань обов'язково повинна розташовуватися команда виходу з переривання, виконавши яку, процесор повертається до виконання перерваної основної програми. Параметри процесора на час виконання програми обробки переривання зберігаються в стеку.

Розглянемо послідовність дій, викликаних сигналом переривання:

1. Перехід до підпрограми. Виконується апаратно. Він може реалізовуватись шляхом занесення в лічильник команд адреси спеціальної підпрограми (відповідно до причини переривань) або шляхом позачергового запису в регістр дешифратора команд спеціальної команди переривання, першими діями якої є організація переходу до цієї спеціальної програми.

2. Заборона інших переривань. Під час обробки переривання встановлюється апаратно або програмно декілька масок або ключів, які закривають деякі види переривань, що захищає виконувану підпрограму від переривань з нижчим пріоритетом.

3. Запам'ятовування інформації, необхідної для відновлення виконання перерваної програми. Оскільки метою переривання є виконання спеціальної підпрограми з наступним поверненням до перерваної програми, необхідно запам'ятати всю інформацію, зв'язану з виконанням перерваної програми для наступного відновлення її роботи.

4. Виконання під програми переривань. Ця частина містить фазу діагностики для виявлення точної причини переривання і наступне виконання необхідних у цьому випадку дій. Наприклад, переривання мало місце з тої причини, що прийшов сигнал про закінчення деякого обміну з зовнішнім пристроєм пам'яті. ЦП переводить програму, яка зажадала цього обміну і тому була заблокована, в стан готовності до подальшої роботи.

5. Відновлення інформації і поновлення виконання перерваної програми. Ця фаза є оберненою до фаз запам'ятовування інформації, заборони переривань та переходу до підпрограми. Відновлюються параметри перерваної програми. Заборони з переривань знімаються і виконується перехід до того місця, де програма була перервана.

Нехай, наприклад, процесор (рисунок 2.5) виконував основну програму і команду, що знаходиться в адресі пам'яті 5000 (умовно). У цей момент він отримав запит переривання з номером (адресою вектора) 4. Процесор закінчує виконання команди з адреси 5000. Потім він зберігає в стеку поточне значення лічильника команд (5001) і поточне значення PSW. Після цього процесор читає з адреси 4 пам'яті код вектора переривання. Нехай цей код рівний 6000. Процесор переходить в адресу пам'яті 6000 і приступає до виконання програми обробки переривання, що починається з цієї адреси. Нехай ця програма закінчується в адресі 6100. Дійшовши до цієї адреси, процесор повертається до

виконання перерваної програми. Для цього він витягує із стека значення адреси (5001), на якій його перервали, і значення PSW на той момент часу. Потім процесор читає команду з адреси 5001 і далі послідовно виконує команди основної програми.

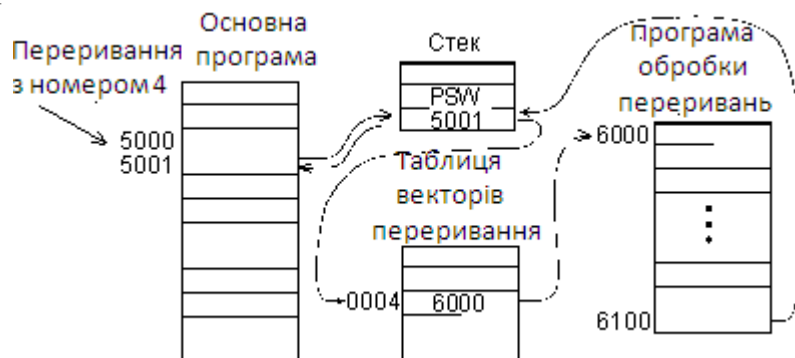


Рисунок 2.5 - Спрощений алгоритм обробки переривання

Переривання у разі аварійної ситуації обробляється так само, тільки адреса вектора переривання (номер рядка в таблиці векторів) жорстко прив'язана до типу аварійної ситуації.

Програмне переривання теж обслуговується через таблицю векторів переривань, але номер переривання вказується у складі команди, що викликає переривання.

Така складна, на перший погляд, організація переривань дозволяє програмістові легко міняти програми обробки переривань, розташовувати їх в будь-якій області пам'яті, робити їх будь-якого розміру і будь-якої складності.

Під час виконання програми обробки переривання може поступити новий запит на переривання. В цьому випадку він обробляється точно так, як і описано, але основною програмою вважається перервана програма обробки попереднього переривання. Це називається багатократним вкладенням переривань. Механізм стека дозволяє без проблем обслуговувати це багатократне вкладення, оскільки першим із стека витягується той код, який був збережений останнім, тобто повернення з обробки даного переривання відбувається в програму обробки попереднього переривання.

Відзначимо, що в складніших випадках в таблиці векторів переривань можуть знаходитися не адреси початку програм обробки переривань, а так звані дескриптори (описувачі) переривань. Але кінцевим результатом обробки цього дескриптора все одно буде адреса початку програми обробки переривань.

Нарешті, ще одна спеціальна область пам'яті МПС - це пам'ять пристроїв, підключених до системної шини. Таке рішення зустрічається нечасто, але іноді воно дуже зручне. Тобто процесор отримує можливість звертатися до внутрішньої пам'яті пристроїв введення/виведення як до своєї власної системної пам'яті. Звичайне вікно в просторі пам'яті, що виділяється для цього, не дуже велике.

Решта всіх частин простору пам'яті, як правило, має універсальне призначення. У них можуть розташовуватися як дані, так і програми (звичайно, у разі одношинної архітектури). Іноді простір пам'яті використовується як

єдине ціле, без всяких границь. А іноді простір пам'яті ділиться на сегменти з програмно змінною адресою початку сегменту і з встановленим розміром сегменту. Обидва підходи мають свої плюси і мінуси. Наприклад, використання сегментів дозволяє захистити область програм або даних, однак межі сегментів можуть погано впливати на розміщення великих програм і масивів даних.

Існує два основні підходи до вирішення проблеми розділення адрес пам'яті і адрес пристроїв введення-виведення:

- виділення в адресному просторі системи спеціальної області адрес для пристроїв введення/виведення;
- повне розділення адресних просторів пам'яті і пристроїв введення/виведення.

Перший підхід хороший тим, що при зверненні до пристроїв введення/виведення процесор може використовувати ті ж команди, які служать для взаємодії з пам'яттю. Але адресний простір пам'яті повинен бути зменшений на величину адресного простору пристроїв введення/виведення. Наприклад, при 16-розрядній шині адреси всього може бути 64К адрес. З них 56К адрес відводиться під адресний простір пам'яті, а 8К адрес — під адресний простір пристроїв введення/виведення.

Перевага другого підходу полягає в тому, що пам'ять займає весь адресний простір МПС. Для взаємодії з пристроями введення/виведення застосовуються спеціальні команди і спеціальні строби обміну на магістралі. Саме так зроблено в персональних комп'ютерах. Але можливості взаємодії з пристроями введення/виведення в даному випадку істотно обмежені в порівнянні з можливостями взаємодії з пам'яттю.

### 2.3. Функції пристроїв введення/виведення

Пристрої введення/виведення обмінюються інформацією з магістраллю по тих же принципах, що і пам'ять. Найбільш істотна відмінність з погляду організації обміну полягає в тому, що модуль пам'яті має в адресному просторі системи багато адрес (до декількох десятків мільйонів), а пристрій введення/виведення зазвичай має небагато адрес (зазвичай до десяти), а іноді і всього одна адреса.

Модулі пам'яті системи обмінюються інформацією тільки з магістраллю та процесором, а пристрої введення/виведення взаємодіють ще і із зовнішніми пристроями, цифровими або аналоговими. Тому різноманітність пристроїв введення/виведення набагато більша, ніж модулів пам'яті. Часто використовуються ще і інші назви для пристроїв введення/виведення: пристрої спряження, контролери, карти розширення, інтерфейсні модулі і так далі.

Об'єднують всі пристрої введення/виведення загальні принципи обміну з магістраллю і, відповідно, загальні принципи організації вузлів, які здійснюють сполучення з магістраллю. Спрощена структура пристрою введення/виведення (точніше, його інтерфейсна частина) приведена на рисунку 2.6. Як і у випадку модуля пам'яті, вона обов'язково містить схему селектора адреси, схему управління для обробки стробів обміну і буфери даних.



Найпростіші пристрої введення/виведення видають на зовнішній пристрій код даних в паралельному форматі і приймають із зовнішнього пристрою код даних в паралельному форматі. Такі пристрої введення/виведення часто називають паралельними портами введення/виведення. Вони найбільш універсальні, тобто задовольняють потреби сполучення з великим числом зовнішніх пристроїв, тому їх часто вводять до складу МПС як стандартні пристрої. Паралельні порти зазвичай є у складі мікроконтролерів. Саме через паралельні порти мікроконтролер зв'язується із зовнішнім світом.

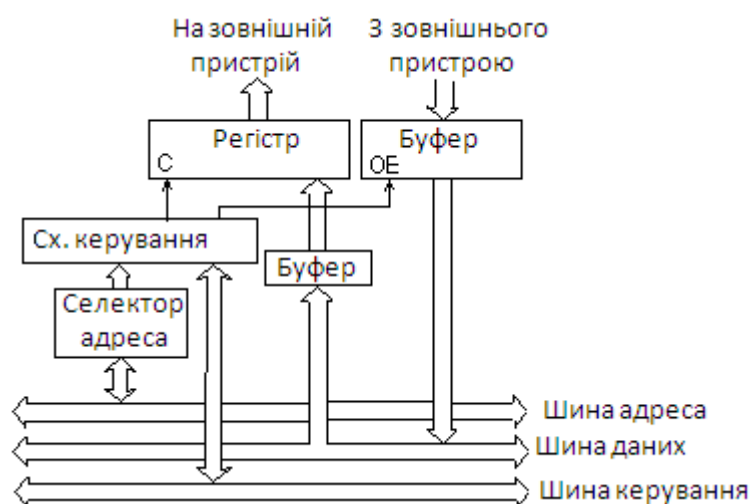


Рисунок 2.6 - Структура простого пристрою введення/виведення

Вхідний порт (порт введення) в простому випадку є паралельним регістром, в який процесор може записувати інформацію. Вихідний порт (порт виведення) зазвичай є просто однонаправленим буфером, через який процесор може читати інформацію від зовнішнього пристрою. Саме такі порти показані для прикладу на рисунку 2.6. Порт може бути і двонаправленим (вхідним/вихідним). В цьому випадку процесор пише інформацію в зовнішній пристрій і читає інформацію із зовнішнього пристрою поодиночі і у тій же адресі в адресному просторі системи. Вхідні і вихідні лінії для зв'язку із зовнішнім пристроєм при цьому можуть бути об'єднані порозрядно, утворюючи двонаправлені лінії.

При зверненні з боку магістралі селектор адреси розпізнає адреса, приписані даному пристрою введення/виведення. Схема керування видає внутрішні строби обміну у відповідь на магістральні строби обміну. Вхідний буфер даних забезпечує електричне узгодження шини даних з цим пристроєм (буфер може бути і відсутнім). Дані з шини даних записуються в регістр по сигналу С і видаються на зовнішній пристрій. Вихідний буфер даних передає вхідні дані із зовнішнього пристрою на шину даних магістралі в циклі читання з порту.

Складніші пристрої введення/виведення (пристрої спряження) мають в своєму складі внутрішню буферну оперативну пам'ять і навіть можуть мати мікроконтролер, на який покладено виконання функцій обміну із зовнішнім пристроєм.

Кожному пристрою введення/виведення відводиться своя адреса в адресному просторі МПС. Дублювання адрес повинне бути виключене, за цим повинні стежити розробник і користувач МПС.

Пристрої введення/виведення крім програмного обміну можуть також підтримувати режим обміну по перериваннях. В цьому випадку вони перетворюють сигнал запиту, що поступає від зовнішнього пристрою, на переривання в сигнал запиту переривання, необхідний для даної магістралі (або в послідовність сигналів при векторному перериванні). Якщо потрібно використовувати режим ПДП, пристрій введення/виведення повинен видати сигнал запиту ПДП на магістраль і забезпечити роботу в циклах ПДП, прийнятих для даної магістралі.

У складі МПС, як правило, виділяються три спеціальні групи пристроїв введення/виведення:

- пристрої інтерфейсу користувача (введення інформації користувачем і виведення інформації для користувача);
- пристрої введення/виведення для тривалого зберігання інформації;
- таймерні пристрої.

До пристроїв введення для інтерфейсу користувача відносяться контролери клавіатури, тумблерів, окремих кнопок, миші, трекбола, джойстика і так далі. До пристроїв виводу для інтерфейсу користувача відносяться контролери світлодіодних індикаторів, табло, рідкокристалічних, плазмових і електронно-променевих екранів і так далі. У простих випадках керуючих контролерів або мікроконтролерів ці засоби можуть бути відсутніми. У складних МПС вони обов'язково присутні. Роль зовнішнього пристрою в даному випадку відіграє людина.

Пристрої введення/виведення для тривалого зберігання інформації забезпечують сполучення МПС з дисководами (компакт-дисків або магнітних дисків), а також з накопичувачами на магнітній стрічці. Застосування таких пристроїв істотно збільшує можливості МПС відносно зберігання виконуваних програм і накопичення масивів даних. У простих контролерах ці пристрої відсутні.

Таймерні пристрої відрізняються від інших пристроїв введення/виведення тим, що вони можуть не мати зовнішніх виводів для підключення до зовнішніх пристроїв. Ці пристрої призначені для того, щоб МПС могла витримувати задані тимчасові інтервали, стежити за реальним часом, рахувати імпульси і так далі. В основі будь-якого таймера лежить кварцевий тактовий генератор і багаторозрядні двійкові лічильники, які можуть перезапускати один одного. Процесор може записувати в таймер коефіцієнти ділення тактової частоти, кількість відлічуваних імпульсів, задавати режим роботи лічильників таймера. В принципі виконати практично всі функції таймера можна і програмним шляхом, тому іноді таймери в системі відсутні. Включення в систему таймера дозволяє вирішувати складніші завдання і будувати ефективніші алгоритми.

Ще один важливий клас пристроїв введення/виведення - пристрої для підключення до інформаційних мереж (локальних і глобальних). Сьогодні

засоби зв'язку з інформаційними мережами вводяться іноді навіть в прості контролери.

Іноді пристрої введення/виведення забезпечують спряження із зовнішніми пристроями за допомогою аналогових сигналів.

## Тестові питання до теми 2

### **1. Процесор – це**

- А) пристрій управління, який генерує послідовність синхронізуючих і логічних сигналів, що визначають послідовності спрацьовування всіх логічних пристроїв МПС.
- Б) схема або пристрій, що розміщується між двома іншими пристроями для згладжування змін швидкості чи рівня або для забезпечення асинхронної роботи.
- В) частина оперативної пам'яті, призначена для тимчасового зберігання даних.
- Г) являє собою окрему мікросхему або ж частину мікросхеми (в випадку мікроконтролерА).

### **2. Найважливішими характеристиками процесора є:**

- А) кількість розрядів його шини даних, кількість розрядів його шини адреси і кількість керуючих сигналів у шині управління;
- Б) кількість розрядів його шини даних;
- В) кількість розрядів його шини даних та кількість розрядів його шини адреси;
- Г) немає жодної вірної відповіді.

### **3. Розрядність шини даних визначає:**

- А) різноманітність режимів обміну та ефективність обміну процесора з іншими пристроями системи;
- Б) допустиму складність системи;
- В) швидкість роботи системи;
- Г) різноманітність режимів обміну та допустиму складність системи.

### **4. CLK – це**

- А) зовнішній тактовий сигнал;
- Б) шина живлення;
- В) сигнал початкового скидання;
- Г) буфер;

### **5. RESET – це**

- А) зовнішній тактовий сигнал;
- Б) шина живлення;
- В) сигнал початкового скидання;
- Г) буфер;

**6.  $U, GND$  – це**

- А) зовнішній тактовий сигнал;
- Б) шина живлення;
- В) сигнал початкового скидання;
- Г) буфер;

**7. Функціями буферних мікросхем є:**

- А) демультіплексування сигналів;
- Б) вибірка виконуваних команд;
- В) затримка сигналів;
- Г) обробка даних, в т.ч. арифметичні операції над ними;

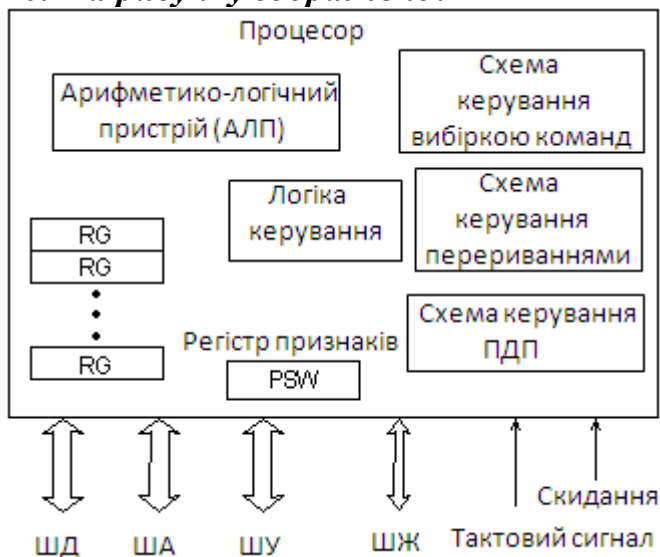
**8. Основними функціями процесора є:**

- А) демультіплексування сигналів;
- Б) вибірка виконуваних команд;
- В) електричне узгодження сигналів магістралі;
- Г) обробка даних;

**9. Чим більша тактова частота процесора...**

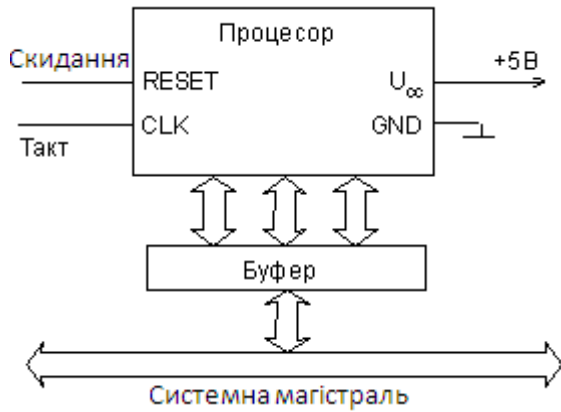
- А) тим швидше він працює;
- Б) тим повільніше він працює;
- В) тим повільніше виконуються команди;
- Г) тим швидше виконуються команди

**10. На рисунку зображено:**



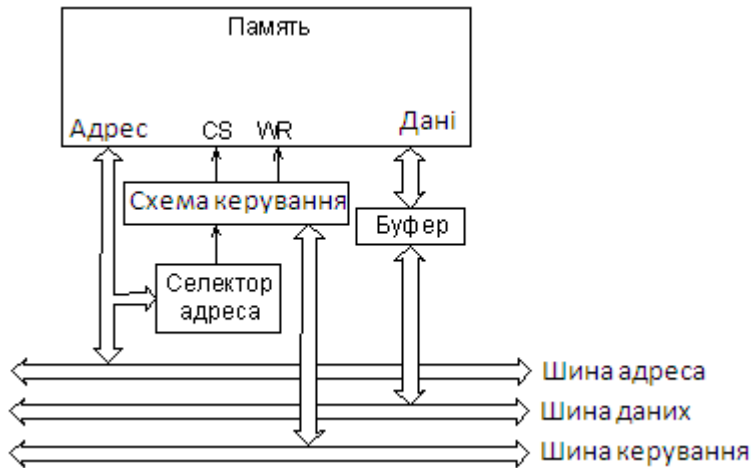
- А) внутрішня структура МП;
- Б) схема керування;
- В) схема включення мікропроцесора до магістралі;
- Г) схема обслуговування ПДП

11. На рисунку зображено:



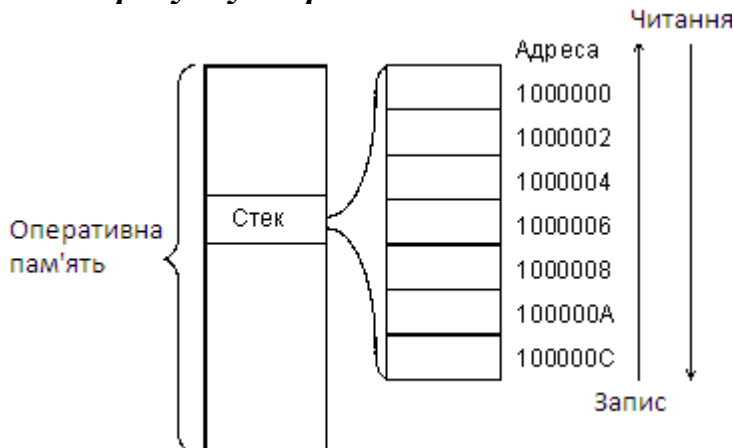
- А) внутрішня структура МП;
- Б) схема керування;
- В) схема включення мікропроцесора до магістралі;
- Г) схема обслуговування ПДП

12. На рисунку зображено:



- А) внутрішня структура МП;
- Б) схема керування;
- В) структура модуля пам'яті;
- Г) схема обслуговування ПДП

13. На рисунку зображено:



- А) принцип роботи ОП;
- Б) схема узгодження стеку;
- В) принцип роботи стека;
- Г) схема обслуговування ПДП

**14. Конвеєр являє собою:**

- А) невелику внутрішню пам'ять процесора, в яку при найменшій можливості записується декілька команд;
- Б) арифметико-логічний пристрій;
- В) слово стану процесора;
- Г) взаємодію всіх вузлів процесора;

**15. Арифметико-логічний пристрій (АЛП) призначений для:**

- А) зберігання даних, які обробляються в даний момент;
- Б) більшого прискорення вибірки команд у сучасних процесорах;
- В) обробки інформації відповідно до отриманої процесором командою;
- Г) переходу на наступну комірку пам'яті;

**16. Внутрішні реєстри будь-якого мікропроцесора обов'язково виконують такі функції:**

- А) функція лічильника команд або покажчика команд;
- Б) функція виведення даних з пам'яті;
- В) функція покажчика стека;
- Г) функція введення даних в пам'ять;

**17. Пам'ять МПС виконує функцію...**

- А) функція тимчасового або постійного зберігання даних і команд;
- Б) функція виведення даних;
- В) функція покажчика стека;
- Г) функція введення даних;

**18. Пам'ять для стека або стек – це**

- А) частина оперативної пам'яті, призначена для тимчасового зберігання даних в режимі LIFO;
- Б) пам'ять з можливістю багатократного перезапису вмісту;
- В) таблиця векторів переривань;
- Г) жодної вірної відповіді;

**19. Буфери даних передають дані...**

- А) від пам'яті до магістралі;
- Б) тільки до пам'яті;
- В) від магістралі до пам'яті;
- Г) тільки до магістралі;

**20. Вектор переривання – це**

- А) повна адреса пам'яті тієї процедури, що викликається з появою запиту на переривання;
- Б) повна адреса пам'яті тієї процедури, що не викликається з появою запиту на переривання;
- В) адреса пам'яті, що зберігає запити;
- Г) жодної вірної відповіді;

**21. Під перериванням розуміється...**

- А) виключення комп'ютера;
- Б) порушення послідовної роботи процесора;
- В) поява помилки;
- Г) обслуговування запиту зовнішнього пристрою;

**22. Будь-яке переривання обробляється...**

- А) через таблицю векторів (показчиків) переривань;
- Б) через графік переривань;
- В) через спеціальну мікросхему;
- Г) через BIOS;

**23. У разі апаратних переривань номер переривання задається...**

- А) пристроєм, що запитав переривання;
- Б) через графік переривань;
- В) номером лінії запиту переривань;
- Г) через таблицю векторів;

**24. Процесор, отримавши апаратне переривання...**

- А) закінчує виконання поточної команди;
- Б) звертається до пам'яті в область таблиці векторів переривань, в той її рядок, який визначається номером запитаного переривання;
- В) продовжує далі працювати з поточною командою;
- Г) сигналізує про помилку;

**25. Перехід до підпрограми під час переривання...**

- А) виконується апаратно, може реалізовуватись шляхом занесення в лічильник команд адреси спеціальної підпрограми;
- Б) захищає виконувану підпрограму від переривань з нижчим пріоритетом.;
- В) містить фазу діагностики для виявлення точної причини переривання і наступне виконання необхідних у цьому випадку дій;
- Г) виконує перехід до того місця, де програма була перервана;

**26. Основні підходи до вирішення проблеми розділення адрес пам'яті і адрес пристроїв введення-виведення:**

- А) виділення в адресному просторі системи спеціальної області адрес для пристроїв введення/виведення;

- Б) встановлення декількох масок або ключів, які закривають деякі види переривань, що захищає виконувану підпрограму від переривань з нижчим пріоритетом;
- В) повне розділення адресних просторів пам'яті і пристроїв введення/виведення;
- Г) заборони з переривань знімаються і виконується перехід до того місця, де програма була перервана;

**27. На рисунку зображено:**



- А) внутрішня структура обробника переривань;
- Б) схема керування перериваннями;
- В) спрощений алгоритм обробки переривання;
- Г) алгоритм обслуговування переривання під час ПДП

**28. Для взаємодії з пристроями введення/виведення застосовуються...**

- А) спеціальні команди;
- Б) схема керування;
- В) спрощений алгоритм;
- Г) спеціальні адреси магістралі;

**29. Модулі пам'яті системи обмінюються інформацією з...**

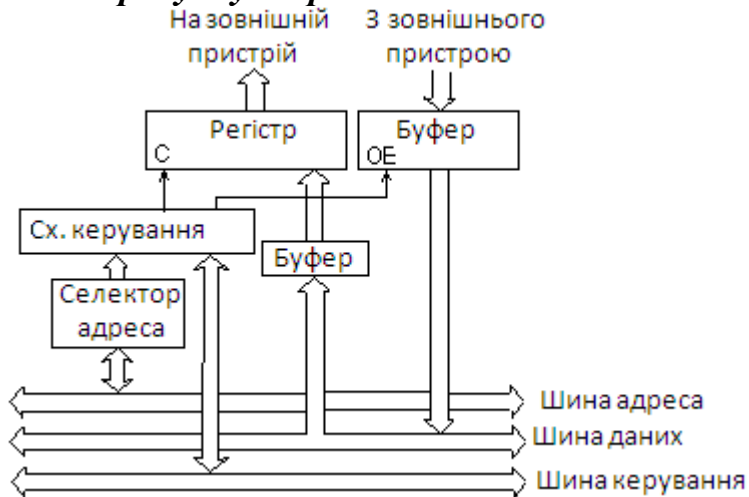
- А) магістраллю;
- Б) каналами контролера;
- В) людиною;
- Г) процесором;

**30. Групи пристроїв МПС:**

- А) пристрої інтерфейсу користувача;
- Б) таймерні пристрої;
- В) мікрокомп'ютери;
- Г) жодної вірної відповіді;

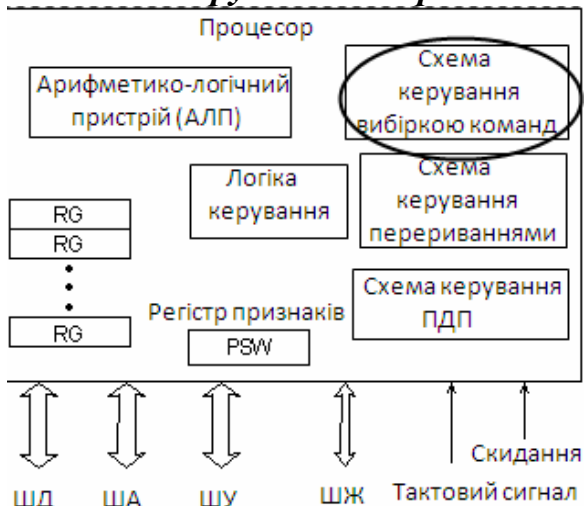


**31. На рисунку зображено:**



- А) внутрішня структура МП;
- Б) схема керування МП;
- В) структура простого пристрою введення/виведення;
- Г) схема обслуговування зовнішнього та внутрішнього пристроїв

**32. Схема керування вибіркою команд ...**



- А) виконує читання команд з пам'яті і їх дешифрування;
- Б) призначена для обробки інформації;
- В) містить слово стану процесора;
- Г) обробляє запит переривання;

**33. Читання команд з пам'яті та їх дешифрування виконує:**

- А) логіка керування;
- Б) схема керування ПДП;
- В) арифметико-логічний пристрій;
- Г) схема керування вибіркою команд.

**34. Для більшого прискорення вибірки команд у сучасних процесорах застосовується:**

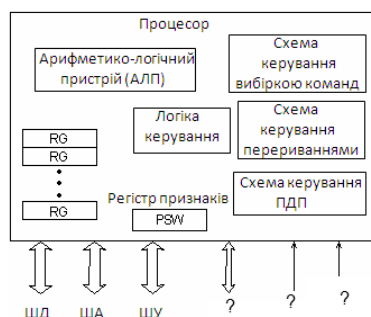
- А) паралельні конвеєра команд;
- Б) передбачення команд переходів;

- В) послідовні конвеєра команд;
- Г) шифрування команд та виокремлення вибірки.

**35. Який елемент у структурі МП містить слово процесора:**

- А) логіка керування;
- Б) схема керування перериваннями;
- В) регістр ознак;
- Г) схема керування вибіркою команд.

**36. Виберіть елементи, які відповідають пропущеним елементам на схемі позначеними знаками питання:**



- А) ШЖ, RES, CLK;
- Б) CLK, ШЖ, Скидання;
- В) RESET, CLK, Тактовий сигнал;
- Г) ШЖ, Тактовий сигнал, Скидання.

**37. Що забезпечує в роботі процесора виділений компонент:**



- А) перенаправлення даних та синхронізацію роботи процесора із зовнішніми сигналами;
- Б) тимчасове відключення процесора від зовнішніх шин;
- В) перехід до програми обробки переривання ;
- Г) організацію взаємодії всіх вузлів процесора.

**38. Щоб забезпечити роботу АЛП на більшій частоті використовують:**

- А) обмеження кількості виконуваних команд АЛП до найпростіших;
- Б) створення RISC – процесорів з зменшеним набором команд;
- В) доведення часу виконання команди до двох-трьох тактів;
- Г) використання декількох паралельно працюючих АЛП.

**39. Що заміняє процесор під час виконання операцій над числами з плаваючою крапкою та інших спеціальних складаних операцій:**

- А) арифметико-логічний пристрій;
- Б) логічний блок керування;
- В) математичний сопроцесор;
- Г) підпроцесор з вбудованим спеціальним математичним блоком.

**40. Тимчасове відключення процесора від зовнішніх шин і припинення його роботи на час надання прямого доступу пристрою, що запитав його, забезпечує:**

- А) логіка керування
- Б) схема управління прямим доступом до пам'яті;
- В) схема керування вибіркою команд;
- Г) регістр признаків.

**41. Яка функція виділеного на рисунку компонента МП?**



- А) перенаправлення даних та синхронізація роботи процесора із зовнішніми сигналами;
- Б) тимчасове відключення процесора від зовнішніх шин;
- В) обробка переривання та зберігання поточного стану регістрів;
- Г) припинення роботи процесора на час надання прямого доступу до пристрою

**42. Який елемент МП визначає область адрес простору пам'яті відведену певному модулю пам'яті**

- А) схема керування;
- Б) селектор адрес;
- В) буфер;
- Г) шина керування.

**43. Схема керування в структурі модуля пам'яті...**

- А) виробляє в потрібні моменти сигнали дозволу роботи пам'яті;
- Б) визначає, яка область адрес простору пам'яті відведена даному модулю пам'яті;
- В) виробляє сигнали дозволу запису в пам'ять;
- Г) передає дані від пам'яті до магістралі.

**44. До пристроїв введення/виведення в МПС належать...**

- А) пристрої для підключення до локальних мереж;
- Б) буфери;
- В) пристрої для підключення до глобальних мереж;
- Г) жодної вірної відповіді.

**45. До пристроїв введення для інтерфейсу користувача відносяться...**

- А) контролери клавіатури;
- Б) окремі кнопки, миші, трекболи, джойстики;
- В) контролери світлодіодних індикаторів, табло;
- Г) немає жодної вірної відповіді.

**46. До пристроїв виводу для інтерфейсу користувача відносяться...**

- А) контролери клавіатури;
- Б) окремі кнопки, миші, трекбола, джойстика;
- В) контролери світлодіодних індикаторів, табло;
- Г) немає жодної вірної відповіді.

**47. Контролери світлодіодних індикаторів, табло, рідкокристалічних, плазмових і електронно-променевих екранів відносяться до...**

- А) пристроїв виводу для інтерфейсу користувача;
- Б) пристроїв вводу для інтерфейсу користувача;
- В) пристроїв тривалого зберігання інформації;
- Г) немає жодної вірної відповіді.

**48. Контролери клавіатури, тумблерів, окремих кнопок, миші, трекбола, джойстика відносяться до...**

- А) пристроїв виводу для інтерфейсу користувача;
- Б) пристроїв вводу для інтерфейсу користувача;
- В) пристроїв тривалого зберігання інформації;
- Г) немає жодної вірної відповіді.

## ТЕМА 3. ОСОБЛИВОСТІ ПРОЕКТУВАННЯ МІКРОПРОЦЕСОРНИХ СИСТЕМ І ПРИСТРОЇВ НА МІКРОКОНТРОЛЕРАХ

### 3.1. Формалізація проектування МПС і пристроїв

При проектуванні мікроконтролерних пристроїв (МКП) або МПС часто використовують блоково-ієрархічний підхід, при якому уявлення про МКП або МПС, що проектується, розчленовуються на ієрархічні рівні. На вищому рівні використовується найменш деталізоване уявлення, що відображає лише тільки загальні риси і особливості системи, що проектується. На наступних рівнях ступінь подробиці розгляду зростає, при цьому система розглядається не загалом, а окремими блоками. Такий підхід дозволяє на кожному рівні формулювати і вирішувати задачі допустимої складності, що піддаються усвідомленню й розумінню людиною та рішення за допомогою доступних засобів проектування. Переваги такого підходу полягають в тому, що складна задача великої розмірності розбивається на групи задач малої розмірності, що послідовно вирішуються, причому всередині груп різні задачі можуть вирішуватися паралельно. Так загальна задача (ЗЗ) для МКП або МПС може бути розбита на ряд глобальних задач (ГЗ), що розташовуються на наступному (за загальною задачею) рівні ієрархії (рисунок 3.1) такому, наприклад, як:

- організація взаємодії мікроконтролера (МК) з об'єктом керування (ввід-вивід даних);
- обробка даних;
- організація зв'язку з оператором (задання режимів роботи, відображення результатів і т.п.);
- організація зв'язку між окремими МК в МКП або МПС;
- організація зв'язку з ЕОМ більш високого рівня, ніж МКП.

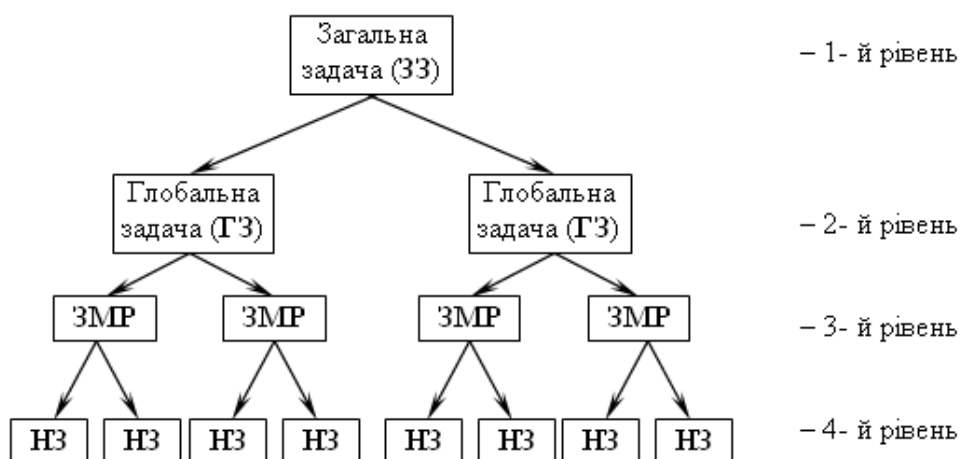


Рисунок 3.1 - Рівні ієрархії при блоково-ієрархічному підході

Кожну глобальну задачу можна розбити на ряд задач малої розмірності (ЗМР), які вирішуються на наступному рівні ієрархії. У свою чергу будь-яку задачу малої розмірності можна розбити на ряд найпростіших задач (НЗ), що

вирішуються на самому низькому рівні ієрархії. Таким чином, горизонтальні ієрархічні рівні проектування представляють собою рівні опису об'єктів, що відрізняються ступенем деталізації відображення властивостей об'єкта. На всіх горизонтальних рівнях є групи задач, які пов'язані з проектуванням схем, програм, конструкцій і технологій. Ці групи задач називаються аспектами або вертикальними рівнями проектування. Відрізняють такі аспекти, як: функціональний, алгоритмічний (програмний), конструкторський і технологічний (таблиця 3.1).

Таблиця 3.1

Рівні та аспекти проектування МПС

Рівні	Аспекти			
	Функціональний	Алгоритмічний	Конструкторський	Технологічний
1	Постановка задач	Розробка закону функціонування МКС. Проектування схем алгоритмів	–	–
2	Системний (структурний)	Архітектурний (машинний)	Стояк, панель	Розробка принципової схеми технологічного процесу
3	Функціонально-логічний	Програмування модулів	ТЕЗ, модуль	Розробка маршрутів технологічного процесу
4	Схемотехнічний Компонентний	Мікропрограмний	Кристал інтегральної мікросхеми	Проектування технологічних операцій

У функціональному і алгоритмічному аспектах можна виділити наступні рівні проектування:

1. Постановка задачі.
2. Системний (структурний) і архітектурний.
3. Функціонально-логічний і програмний.
4. Схемотехнічний і мікропрограмний.

Вихідною інформацією для системного і архітектурного рівнів проектування МКП і МПС є: закони їх функціонування, задані у вигляді аналітичних співвідношень перетворення вхідної інформації; набір схем алгоритмів виконання основних функцій МКП і МПС, що реалізуються апаратним і програмним шляхом; критерії оптимальності та набір обмежень на параметри МКП і МПС, що розробляються.

Розглянемо основні задачі, що вирішуються в функціональному і алгоритмічному аспектах на різних рівнях проектування МКП і МПС.

Основними задачами системного і архітектурного рівнів є:

- визначення принципів організації МКП і МПС;
- розробка структурної схеми, тобто визначення складу пристроїв і способів їх взаємодії в процесі функціонування;
- вибір однокристальних мікроконтролерів (ОМК) та інших апаратних засобів;
- уточнення функцій МКП і МПС, що реалізуються, а також поділ на функції, що реалізуються апаратним і програмним шляхами;
- розробка детальних блок-схем алгоритмів програм і визначення форматів даних;
- визначення вимог до параметрів пристроїв і формування технічних завдань (ТЗ) на розробку окремих блоків МКП і МПС.

Основні задачі функціонально-логічного і програмного рівнів:

- деталізація функцій, що виконуються кожним блоком і пристроєм;
- розробка функціональних і принципівих схем всіх пристроїв;
- алгоритмічна реалізація функцій, що виконуються програмним шляхом, і представлення алгоритмів на одній з прийнятих алгоритмічних мов або Асемблері;
- синтез контролюючих і діагностичних тестів.

На схемотехнічному і мікропрограмному рівнях розробляються принципові схеми та найпростіші програмні модулі, здійснюється запис робочих програм в постійний запам'ятовуючий пристрій (ПЗП) ОМК.

Конструкторський аспект включає в себе ієрархічні рівні проектування стояків, панелей, типових елементів заміни (ТЕЗів), модулів і кристалів (чипів) інтегральних мікросхем.

Технологічний аспект включає в себе ієрархічні рівні визначення принципової схеми технологічного процесу, тобто складу і послідовності етапів виготовлення МПС, розробки маршрутів технологічних процесів, визначення складу і послідовності операцій, вибір баз і групи технологічного обладнання, проектування технологічних операцій.

Блоково-ієрархічний підхід, розглянутий вище, дозволяє досить просто розподілити, у разі необхідності, інтелект пристроїв або систем між декількома мікроконтролерами. При цьому кожна глобальну задачу або навіть задачу меншої розмірності можна "доручити" вирішувати окремому мікроконтролеру. Очевидно, що в цьому випадку неминує виникає задача організації хороших інтерфейсних зв'язків як між окремими ОМК і МКП, так і між МКП і ЕОМ, що знаходяться на більш високому рівні ієрархії в МПС (наприклад, в локальних керуючих мережах на основі ОМК). Причому, для зв'язку ОМК в межах одного МКП доцільно використати більш прості (нестандартні) інтерфейси. Стандартні ж інтерфейси типу I2C або RS-232 краще використовувати для обміну даними між МКП або МКП і ЕОМ в межах однієї МПС.

На кожному із рівнів, що розглядаються, розробнику доводиться вирішувати задачі синтезу і аналізу. Метою задачі синтезу є отримання

конкретних варіантів апаратури, що проектується, а метою задачі аналізу – вивчення властивостей варіантів апаратури та їх оцінка.

Відрізняють структурний і параметричний синтез. Мета структурного синтезу – отримання структури пристрою, тобто складу елементів і способу зв'язку їх між собою. Задачу вибору оптимальної структури називають структурною оптимізацією. Мета параметричного синтезу – визначення числових значень параметрів елементів і пристроїв, причому задачу отримання оптимальних значень параметрів називають параметричною оптимізацією.

При вирішенні задач аналізу використовують моделі системи, що проектується. Відрізняють фізичні та математичні моделі. Фізичними моделями є різного роду макети, стенди. Математична модель – це сукупність математичних об'єктів (чисел, змінних, векторів, множин і т.п.) та відношення між ними, яка адекватно відображає властивості об'єкта, що проектується. Математичні моделі можуть бути функціональними, якщо вони відображають фізичні або інформаційні процеси, які протікають в об'єкті, що моделюється, і структурними, якщо вони відображають тільки структурні (в тому числі й геометричні) властивості об'єктів.

Функціональні моделі, як правило, задаються у вигляді аналітичних співвідношень, а структурні – у вигляді графів і матриць. При проектуванні МПС, розробці та відлагодженні програм функціонування МКП широко використовують функціональні моделі роботи МКП, що реалізуються на ЕОМ.

### 3.2. Схема та методика процесу проектування МПС та пристроїв

На рисунку 3.2 представлена типова для рівнів, що розглядаються, схема процесу проектування.

Розробка апаратури за технічним завданням, що пред'являється, починається з аналізу можливостей реалізації вимог ТЗ. У ТЗ на розробку окремих пристроїв МПС входять: перелік всіх функцій, що виконуються кожним пристроєм; умови працездатності пристрою; вимоги до його вхідних і вихідних параметрів; дані про зміст і форму інформації, якою даний пристрій обмінюється з іншими пристроями апаратури; елементна база для побудови пристрою. У разі позитивного результату аналізу розробляють моделі для структурного і параметричного синтезу апаратури. Відповідно до розроблених моделей генерується початковий варіант апаратури, параметри якого оцінюються з позицій задоволення умов працездатності. Якщо для варіанта структури, що синтезується, забезпечується виконання умов працездатності із заздалегідь обумовленим запасом (з урахуванням допустимих відхилень реально досягнутих значень параметрів апаратури), то задача синтезу вважається вирішеною. При цьому результати проектування оформляють у вигляді необхідної технічної документації. У разі невиконання умов працездатності проводять аналіз можливості поліпшення характеристик синтезованого варіанта об'єкта шляхом зміни структури і керованих параметрів



в рамках моделей, що застосовуються для структурного синтезу і параметричної оптимізації.



Рисунок 3.2 – Блок-схема процесу проектування МПС та пристроїв

У випадку неможливості виконання умов працездатності проводять коректування ТЗ на об'єкт, що проектується. Таким чином, процес проектування носить ітераційний характер, ітерації можуть включати в себе і більше одного рівня проектування, необхідність повернення до попереднього рівня може виявитися на будь-якому подальшому рівні проектування.

Проектування складних технічних систем проводиться на основі головних критеріїв:

- якості проектування;
- вартості проектування;
- строків розробки;
- кількості зайнятих фахівців-розробників.

Можливість формалізації в процесі проектування МПС та ітераційний характер проектування з урахуванням перелічених критеріїв робить переважним вибір автоматизованих методів проектування МПС. Розробка ж мікроконтролерної частини апаратури без використання автоматизованих методів проектування і відлагодження програм МПС в даний час практично неможлива.

Реалізація блоково-ієрархічного підходу до проектування складних систем може бути здійснена або у висхідній, або в низхідній послідовності рішення задач проектування різних рівнів. Низхідне проектування (проектування "зверху вниз") характеризується тим, що рішення задач проектування верхніх рівнів передують рішенням задач проектування нижніх рівнів. При цьому початкові дані для проектування представляються у вигляді ТЗ для вищого ієрархічного рівня. Висхідне проектування передбачає зворотню послідовність рішення задач. Об'єкти, що проектуються, на кожному рівні, є базовими для реалізації верхніх рівнів.

При розробці МПС знаходять застосування як низхідне, так і висхідне проектування. Уявлення про об'єкти, що проектуються, на різних рівнях розробки МПС задаються математичними моделями, що описують можливі структури об'єкта і процеси, що в них протікають.

При функціональному проектуванні апаратури використовуються наступні моделі та методи.

На системному рівні модель проектування повинна відображати необхідну сукупність функціональних елементів системи, взаємозв'язок між цими елементами і процеси перетворення інформації. Тому для рішення задач структурного синтезу використовується апарат теорії систем, теорії графів і мереж, теорії перетворення сигналів, теорії інформації, дискретного математичного програмування та інше. Задача аналізу на цьому рівні полягає у визначенні параметрів і характеристик, що описують процес обробки інформації в системі. При цьому треба оцінити продуктивність і відносно завантаження роботи всіх функціональних елементів системи, точність і достовірність перетворення інформації, основні параметри і характеристики спроектованих МПС. Математичним апаратом аналізу на системному рівні є теорія масового обслуговування, теорія обчислювальних систем.

На функціонально-логічному та схемотехнічному рівнях проектування МПС розробляються функціональні і принципіві схеми всіх цифрових й аналогових пристроїв апаратури. На цих рівнях в математичних моделях відображаються змінні та дії, які виконуються пристроєм, що моделюється, відповідно до алгоритму функціонування. При цьому інформаційні змінні ототожнюються з сигналами і кожному сигналу ставиться у відповідність деяка

фізична величина – напруга або струм на виході кожного елемента. Математичний апарат, що використовується для рішення задач аналізу і синтезу на цьому рівні: теорія електромагнітних ланцюгів, математична логіка і теорія цифрових автоматів, нелінійне програмування, положення теорії імовірностей та математичної статистики.

При рішенні задач алгоритмічного проектування застосовуються алгоритмічні мови різних рівнів, теорія алгоритмів, теорія обчислювальних систем і програмування.

### 3.3. Типові структури МК-систем і пристроїв

Типова структура МК-системи показана на рисунку 3.3 і складається з об'єкта керування, мікроконтролера та апаратури їх взаємного зв'язку (АВЗ).

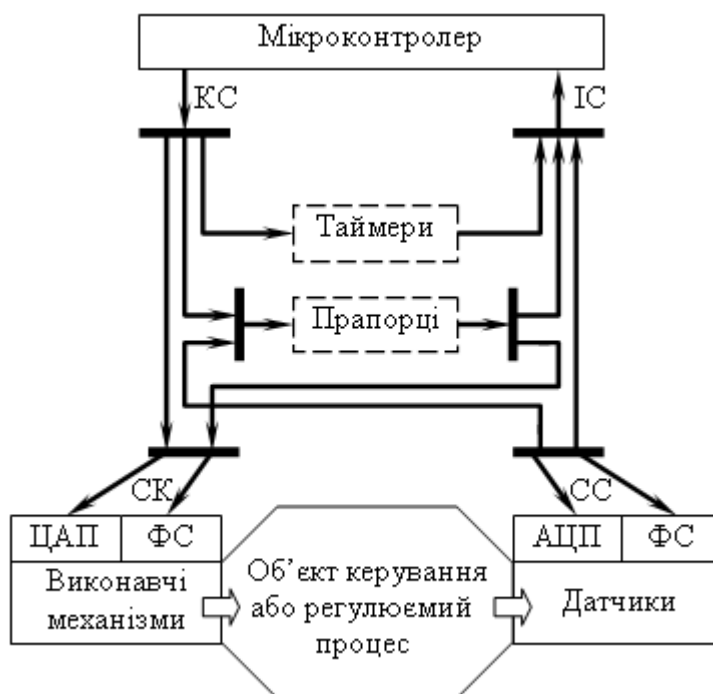


Рисунок 3.3. Типова структура МК-систем і пристроїв на основі ОМК

Мікроконтролер шляхом періодичного опиту інформативних слів (ІС) генерує відповідно до алгоритму керування послідовності керуючих слів (КС). Інформативні слова – це сигнали стану об'єкта (СС), сформовані датчиками об'єкта керування, і прапорці. Вихідні сигнали датчиків внаслідок їх різної фізичної природи можуть зажадати проміжного перетворення на аналого-цифрових перетворювачах (АЦП) або на схемах формувачів сигналів (ФС), які частіше за все виконують функції гальванічної розв'язки і формування рівнів двійкових сигналів стандарту ТТЛ.

Мікроконтролер з необхідною періодичністю оновлює керуючі слова на своїх вихідних портах. Деяка частина керуючого слова інтерпретується як сукупність прямих двійкових сигналів керування (СК), які через схеми формувачів сигналів (підсилювачі потужності, реле, оптрони і т.п.) надходять

на виконавчі механізми (ВМ) і пристрої індикації. Інша частина керуючого слова представляє собою упаковані двійкові коди, які через цифро-аналогові перетворювачі (ЦАП) впливають на виконавчі механізми аналогового типу.

Якщо об'єкт керування використовує цифрові датчики та цифрові виконавчі механізми, то наявність ЦАП і АЦП в системі необов'язкова.

До складу апаратури зв'язку, яка, як правило, побудована на інтегральних схемах серії ТТЛ, входить регістр прапорців, на якому фіксується деяка множина ознак, що специфікуються, як об'єкта керування, так і процесу роботи контролера.

Цей регістр прапорців використовується як апаратний засіб реалізації механізму взаємної синхронізації відносно повільних і ймовірних процесів в об'єкті керування і швидких процесів в контролері. Регістр прапорців доступний як контролеру, так і датчикам. Внаслідок цього він є зручним місцем фіксації сигналів ГОТОВИЙ/ОЧІКУВАННЯ при передачах сигналів ЗАПИТ НА ПЕРЕРИВАННЯ/ПІДТВЕРДЖЕННЯ при взаємодії контролера і об'єкта в режимі переривання. Якщо МК-система має багаторівневу систему переривань, то регістр прапорців містить схему впорядкування пріоритетів.

Для апаратної реалізації часових затримок, формування сигналів необхідної частоти і скважності до складу апаратури взаємного зв'язку включають інтервальні таймери, що програмуються в тому випадку, якщо вони не входять до складу МК або їх кількість недостатня.

Закони функціонування МК-системи керування зі структурою, показаною на рисунку 3.3, цілком визначаються прикладною програмою, що розміщується в резидентній пам'яті програм МК. Іншими словами, спеціалізація МКП або МПС типової структури на рішення задачі керування конкретним об'єктом здійснюється шляхом розробки прикладних програм МК і апаратури зв'язку МК з датчиками і виконавчими механізмами об'єкта.

### 3.4. Використання жорсткої і програмованої логіки

Існує два принципово різних підходи до проектування цифрових пристроїв: використання принципу схемної (жорсткої) логіки або використання принципу програмованої логіки.

У першому випадку в процесі проектування підбирається деякий набір цифрових мікросхем (звичайно малого і середнього ступеня інтеграції) та визначається така схема з'єднання їх виводів, яка забезпечує необхідне функціонування (тобто функціонування пристрою визначається тим, які вибрані мікросхеми і за якою схемою виконано з'єднання їх виводів). Пристрої, побудовані на такому принципі схемної логіки, здатні забезпечувати найвищу швидкодію при заданому типі технології елементів. Недолік цього принципу побудови МКП полягає в неможливості "перебудови" структури пристроїв і систем при необхідності зміни або розширення їх функціональних можливостей.

Ці обставини примушують звернутися до іншого підходу в проектуванні цифрових пристроїв, заснованого на використанні принципу програмованої

логіки. Цей підхід передбачає побудову з використанням однієї або декількох ВІС деякого універсального пристрою, в якому необхідне функціонування (тобто його спеціалізація) забезпечується занесенням в пам'ять пристрою певної програми (або мікропрограми). У залежності від введеної програми такий універсальний керуючий пристрій здатний забезпечувати необхідне керування операційним пристроєм при рішенні найрізноманітніших задач. У цьому випадку кількість типів ВІС, необхідних для побудови керуючого пристрою, невелика, а потреба в ВІС кожного типу висока. Це забезпечує доцільність їх випуску промисловістю.

Потрібно мати на увазі, що найвища швидкодія досягається в процесорах, в яких керуючий пристрій будується з використанням схемної логіки, а операційний пристрій виконується у вигляді пристрою, спеціалізованого для рішення конкретної задачі.

Якщо в пристрої, побудованому на принципі схемної логіки, будь-яка зміна або розширення функцій, що виконуються, спричиняє за собою демонтаж пристрою і його монтаж за новою схемою, то у випадку МКП, завдяки використанню принципу програмованої логіки, така зміна досягається заміною програми, що зберігається в пам'яті, новою програмою, яка відповідає новим функціям, що виконуються пристроєм. Така гнучкість застосувань разом з іншими пов'язаними із використанням ВІС перевагами (низькою вартістю, малими габаритами), а також висока точність і заводо захищеність, характерні для цифрових методів, обумовили бурхливе впровадження МКП в різні сфери виробництва, наукові дослідження та побутову техніку.

Мікроконтролерні пристрої в свою чергу забезпечили широке використання цифрових методів в різних технічних застосуваннях, і розмах впровадження цих нових методів розглядається як революція в техніці.

### 3.5. Дуалізм "програмні засоби – апаратура"

Проектування МПС проводиться на базі певних ОМК. Розробникам апаратури на основі ОМК недоступний рівень окремих транзисторів, зв'язків між ними. Мікроконтролер сприймається як щось суцільне, що має різні властивості, закладені в його архітектурі.

Архітектура мікроконтролера – це його логічна організація, яка визначається можливостями МК по апаратурній або програмній реалізації функцій, які покладаються на МКП, що проектуються. Архітектура відображає структуру МК, способи представлення і формати даних, набір операцій, формати керуючих слів, способи звертання до всіх доступних для користувача елементів структури, реакцію МК на зовнішні сигнали.

При розробці МПС необхідно визначити, яка частина функцій при створенні конкретних МПС повинна бути реалізована програмним способом, а яка – за допомогою додаткових апаратних засобів. Тому при проектуванні МПС необхідно:

- дати опис концептуальної моделі функціональної поведінки МПС і рекомендації по організації обчислювального процесу в мікроконтролерній апаратурі;
- визначити структуру і особливості побудови програмних засобів;
- описати характеристики внутрішньої організації потоків даних і керуючої інформації;
- провести аналіз функціональної структури та особливостей фізичної реалізації пристроїв МПС з позицій збалансованості програмних і апаратних засобів.

При розробці архітектури МКП визначають формати даних, обґрунтовують вимоги до інтерфейсів. Правильний вибір архітектури дає можливість оптимізувати обчислювальний процес реалізації алгоритмів функціонування МКП на вибраних апаратних засобах. У МКП процес оптимізації починається з рішення компромісу "програмні засоби – апаратура", який полягає в тому, що в МКП будь-яке функціональне перетворення може бути здійснено як програмним шляхом, так і на апаратних засобах. При цьому для реалізації будь-яких програмних засобів необхідна апаратна підтримка і, навпаки, жоден з апаратних засобів МКП не може функціонувати без програмного забезпечення.

Тестові питання до теми 3

**1. На найвищому рівні ієрархії в блоково-ієрархічному підході застосовується:**

- А) найменш деталізоване уявлення, що відображає тільки загальні риси і особливості системи;
- Б) найбільш деталізоване уявлення про систему;
- В) зв'язки між різними блоками системи;
- Г) немає вірної відповіді.

**2. До переваг блоково-ієрархічного підходу відносять:**

- А) можливість на кожному рівні формулювати і вирішувати задачі допустимої складності, які можна вирішити за допомогою доступних засобів проектування;
- Б) максимально висока швидкодія;
- В) складна задача великої розмірності розбивається на групи задач малої розмірності, що послідовно вирішуються;
- Г) всі відповіді вірні.

**3. Скільки існує рівнів блоково-ієрархічного підходу?**

- А) 6;
- Б) 4;
- В) 2;
- Г) 3.

**4. Аспектами або вертикальними рівнями проектування називають:**

- А) блоки, на які поділяється система;
- Б) зв'язки, які проходять від одних блоків до інших по вертикальних рівнях;
- В) групи задач, які пов'язані з проектуванням схем, програм, конструкцій і технологій;
- Г) групи задач, які безпосередньо не пов'язані з проектуванням схем, а відносяться до їх програмної реалізації.

**5. Які існують аспекти?**

- А) функціональний;
- Б) аспект проектування;
- В) алгоритмічний або програмний;
- Г) усі відповіді вірні.

**6. До якого аспекту відносять програмування модулів?**

- А) конструкторський;
- Б) технологічний;
- В) функціональний;
- Г) алгоритмічний.

**7. Технологічний аспект включає в себе:**

- А) розробка закону функціонування МКС;
- Б) розробка принципової схеми технологічного процесу;
- В) розробка маршрутів технологічного процесу;
- Г) всі відповіді вірні.

**8. Що не являється вихідною інформацією для системного і архітектурного рівнів проектування?**

- А) вимоги до параметрів пристроїв;
- Б) закони функціонування систем, задані у вигляді аналітичних співвідношень перетворення вхідної інформації;
- В) критерії оптимальності та набір обмежень на параметри МКП і МПС;
- Г) набір схем алгоритмів виконання основних функцій МКП і МПС.

**9. Що відноситься до задач функціонально-логічного і програмного рівнів?**

- А) деталізація функцій, що виконуються кожним блоком і пристроєм;
- Б) розробка функціональних і принципових схем всіх пристроїв;
- В) синтез контролюючих і діагностичних тестів;
- Г) всі відповіді вірні.

**10. До якого аспекту відноситься дане висловлювання «Він включає в себе ієрархічні рівні визначення принципової схеми технологічного процесу, тобто складу і послідовності етапів виготовлення МПС, розробки маршрутів технологічних процесів, визначення складу і послідовності**

*операцій, вибір бази і групи технологічного обладнання, проектування технологічних операцій»?*

- А) функціональний;
- Б) алгоритмічний;
- В) конструкторський;
- Г) технологічний.

**11. Метою структурного синтезу є:**

- А) отримання конкретних варіантів апаратури, що проектується;
- Б) визначення числових значень параметрів елементів і пристроїв;
- В) отримання структури пристрою, тобто складу елементів і способу зв'язку їх між собою;
- Г) вивчення властивостей варіантів апаратури та їх оцінка.

**12. Скільки існує основних класів моделей систем?**

- А) 2;
- Б) 4;
- В) 3;
- Г) немає вірної відповіді.

**13. Що зображено на даній блок-схемі?**



- А) процес структурного синтезу;
- Б) процес параметричного синтезу;
- В) процес проектування МПС та пристроїв;
- Г) жодної вірної відповіді.

**14. У технічне завдання на розробку окремих пристроїв МПС входять:**

- А) перелік всіх функцій, які виконуються кожним пристроєм;



- Б) дані про надійність та швидкодію пристрою;
- В) журнал тестувань даного пристрою;
- Г) умови працездатності пристрою; вимоги до його вхідних і вихідних даних.

**15. Що відбувається при проектуванні МПС у випадку неможливості виконання умов працездатності?**

- А) параметрична оптимізація;
- Б) структурний синтез;
- В) перевірка можливості зміни структури і параметрів;
- Г) аналіз технічного завдання.

**16. Скільки ітерацій може включати в себе процес проектування МПС?**

- А) лише одну;
- Б) не більше трьох;
- В) будь-яку кількість ітерацій;
- Г) тільки дві.

**17. На основі яких критеріїв проводиться проектування складних технічних систем?**

- А) термін окупності;
- Б) якість проектування;
- В) кількість зайнятих фахівців-розробників;
- Г) кількість операцій, яку здатна виконувати система за одиницю часу.

**18. До якого типу проектування відноситься дане визначення «Характеризується тим, що рішення задач проектування верхів рівнів передують рішенням задач проектування нижніх рівнів»?**

- А) низхідне проектування;
- Б) висхідне проектування;
- В) ітераційне проектування;
- Г) немає жодної вірної відповіді.

**19. Який з методів проектування використовують при розробці МПС?**

- А) висхідне проектування;
- Б) низхідне проектування;
- В) ітераційне проектування;
- Г) всі відповіді вірні.

**20. Модель проектування на системному рівні повинна відображати:**

- А) необхідну сукупність функціональних елементів системи, взаємозв'язок між цими елементами і процеси перетворення інформації;
- Б) змінні та дії, які виконуються пристроєм, що моделюється;
- В) передачу даних між окремими структурними блоками;
- Г) немає вірної відповіді.

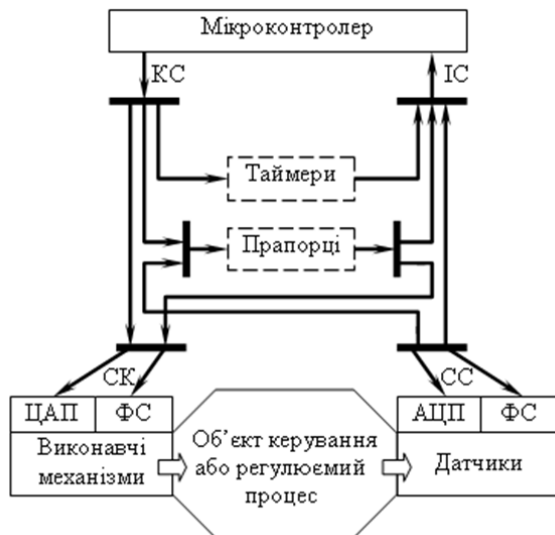
21. Що не застосовується при вирішенні задач алгоритмічного проектування:

- А) теорія алгоритмів;
- Б) алгоритмічні мови різних рівнів;
- В) теорія електромагнітних ланцюгів;
- Г) теорія графів і мереж.

22. До якого рівня відноситься даний математичний апарат, що використовується для задач аналізу і синтезу «апарат теорії систем, теорії графів і мереж, теорії перетворення сигналів, теорії інформації, дискретного математичного програмування»?

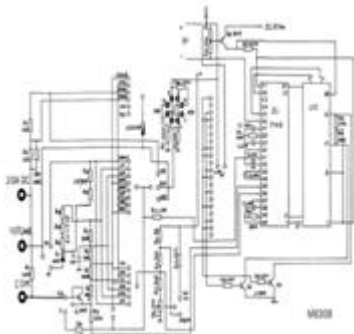
- А) функціонально-логічний рівень;
- Б) системний рівень;
- В) схмотехнічний рівень;
- Г) алгоритмічного проектування.

23. Що зображено на даному малюнку?



- А) структура центрального процесора;
- Б) структура блоку пам'яті;
- В) типова структура МК-систем і пристроїв;
- Г) схема передачі даних від мікроконтролера до зовнішніх пристроїв.

24. Що зображено на малюнку?



- А) приклад приладу з використанням програмованої логіки;
- Б) приклад приладу з використанням схемної логіки;
- В) приклад універсальної МПС;
- Г) немає вірної відповіді.

**25. Для чого застосовується виділений на схемі елемент?**



- А) для фіксації деякої множини ознак об'єкта керування та процесу роботи контролера;
- Б) для реалізації механізму взаємної синхронізації відносно повільних і ймовірних процесів в об'єкті керування і швидких процесів в контролері;
- В) для реалізації механізму взаємної синхронізації процесів в об'єкті керування;
- Г) для фіксації значення поточної операції, яку виконує мікроконтролер.

**26. Для чого застосовується виділений на схемі елемент?**



- А) для фіксації деякої множини ознак об'єкта керування та процесу роботи контролера;

- Б) для апаратної реалізації часових затримок, формування сигналів необхідної частоти і скважності;
- В) для реалізації механізму взаємної синхронізації процесів в об'єкті керування;
- Г) всі відповіді рівні.

**27. Які переваги пристроїв на «жорсткій логіці»?**

- А) найвища швидкодія при заданому типі технології елементів;
- Б) можливість перебудови структури пристроїв і систем при необхідності зміни або розширення їх функціональних можливостей;
- В) використовуються для вирішення широкого класу задач;
- Г) всі відповіді вірні.

**28. Підхід, який передбачає побудову з використанням однієї або декількох ВІС деякого універсального пристрою, в якому необхідне функціонування забезпечується занесенням в пам'ять пристрою певної програми (або мікропрограми), називається:**

- А) схемною логікою;
- Б) програмованою логікою;
- В) змішаною логікою;
- Г) немає вірної відповіді.

**29. Які переваги пристроїв з програмованою логікою?**

- А) найвища швидкодія при заданому типі технології елементів;
- Б) може використовуватись для рішення найрізноманітніших задач;
- В) невелика кількість ВІС, необхідних для побудови керуючого пристрою;
- Г) немає апаратної надмірності.

**30. Які етапи існують при проектуванні МПС:**

- А) визначення структури і особливостей побудови програмних засобів;
- Б) аналіз ефективності запропонованих апаратних засобів;
- В) визначення часової ефективності роботи МПС;
- Г) аналіз функціональної структури та особливостей фізичної реалізації пристроїв МПС з позицій збалансованості програмних і апаратних засобів.

**31. Що нам дає правильний вибір архітектури?**

- А) можливість зменшити час виконання задачі;
- Б) можливість оптимізувати обчислювальний процес реалізації алгоритмів функціонування МКП на вибраних апаратних засобах;
- В) можливість підвищити стійкість системи;
- Г) всі відповіді вірні.

**31. В чому полягає дуалізм «програмні засоби – апаратура»?**

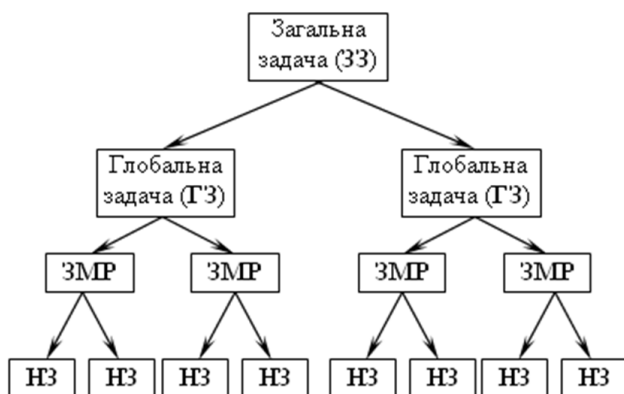
- А) в МКП будь-яке функціональне перетворення може бути здійснено як програмним шляхом, так і на апаратних засобах;

- Б) в МКП будь-яке функціональне перетворення може бути здійснено лише програмним шляхом;
- В) в МКП будь-яке функціональне перетворення може бути здійснено лише на апаратних засобах;
- Г) жодної вірної відповіді.

**32. В чому полягає блоково-ієрархічний підхід при проектуванні МКП?**

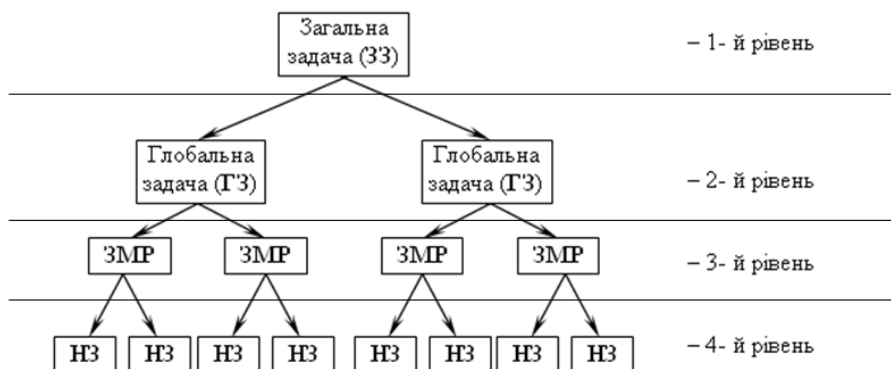
- А) у використанні блок-схем;
- Б) у розділенні МКП на незалежні частини;
- В) у розчленовуванні уявлення про МКП або МКС, що проектується, на ієрархічні рівні;
- Г) жодної вірної відповіді.

**33. Що зображено на даному малюнку?**



- А) етапи проектування МПС;
- Б) рівні ієрархії блоково-ієрархічного підходу;
- В) канали передачі інформації між рівнями;
- Г) горизонтальні рівні блоково-ієрархічного підходу.

**34. Що зображено на даному малюнку?**



- А) горизонтальні рівні блоково-ієрархічного підходу;
- Б) етапи проектування МПС;
- В) рівні ієрархії блоково-ієрархічного підходу;
- Г) канали передачі інформації між рівнями.

**35. Які рівні проектування відносяться до функціонального аспекту?**

- А) постановка задачі;
- Б) розробка принципової схеми технологічного процесу;
- В) програмування модулів;
- Г) системний (структурний).

**36. Який з аспектів включає в себе рівні проектування стояків, панелей, типових елементів заміни, модулів і кристалів інтегральних мікросхем?**

- А) функціональний;
- Б) алгоритмічний;
- В) конструкторський;
- Г) технологічний.

**37. Які рівні включає в себе алгоритмічний аспект?**

- А) функціонально-логічний;
- Б) програмування модулів;
- В) мікропрограмний;
- Г) компонентний.

**38. На яких рівнях здійснюється запис робочих програм в постійний запам'ятовуючий пристрій ОМК?**

- А) схемотехнічний;
- Б) програмування модулів;
- В) мікропрограмний;
- Г) системний.

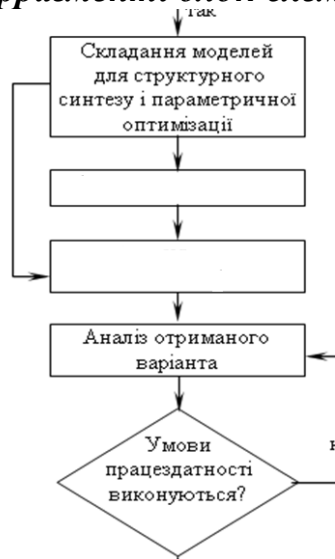
**39. Що відноситься до задач функціонально-логічного і програмного рівнів?**

- А) визначення принципів організації МКП;
- Б) деталізація функцій, що виконуються кожним блоком і пристроєм;
- В) розробка детальних блок-схем алгоритмів програм і визначення форматів даних;
- Г) синтез контролюючих і діагностичних тестів.

**40. Сукупність математичних об'єктів та відношення між ними, яка адекватно відображає властивості об'єкта, що проектується – це:**

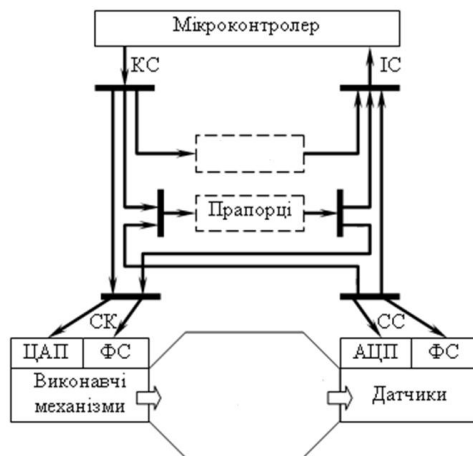
- А) фізична модель;
- Б) математична модель;
- В) функціональна модель;
- Г) структурна модель.

41. Які з етапів проектування МПС та пристроїв пропущені на даному фрагменті блок-схеми?



- А) структурний синтез;
- Б) аналіз ТЗ;
- В) зміна структури і параметрів;
- Г) параметрична оптимізація.

42. Які з елементів пропущені на даній схемі?

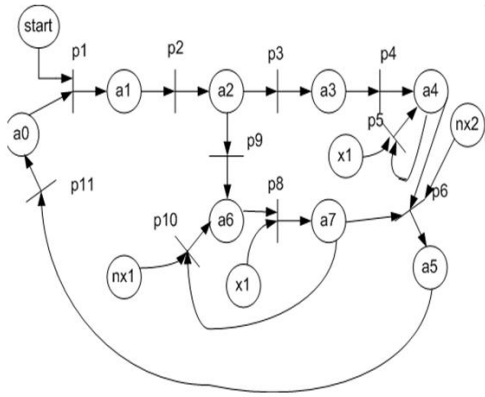


- А) таймери;
- Б) об'єкт керування або регульований процес;
- В) буфер;
- Г) модуль пам'яті.

43. Що характеризує даний вислів «будь-яка зміна або розширення функцій, що виконуються, спричиняє за собою демонтаж пристрою і його монтаж за новою схемою»?

- А) пристрій, побудований по принципу схемної логіки;
- Б) пристрій, побудований по принципу програмованої логіки;
- В) обидві відповіді вірні;
- Г) немає вірної відповіді.

**44. Що зображено на даному малюнку?**



- А) цифровий пристрій на жорсткій логіці;
- Б) цифровий пристрій з програмованою логікою;
- В) реалізація алгоритму проектування МПС;
- Г) жодної вірної відповіді.

**45. Що потрібно зробити при проектуванні архітектури МКП?**

- А) розробити детальні блок-схеми алгоритмів програм;
- Б) визначити формати даних;
- В) обґрунтувати вимоги до інтерфейсів;
- Г) визначити канали зв'язку між модулями.

**46. Чим визначаються закони функціонування МК-системи керування?**

- А) прикладною програмою, що розміщується в резидентській пам'яті програм МК;
- Б) мікроконтролером;
- В) виконавчими механізмами;
- Г) датчиками.

**47. Які з аспектів не передбачають запис робочих програм в постійний запам'ятовуючий пристрій?**

- А) функціональний;
- Б) алгоритмічний;
- В) конструкторський;
- Г) технологічний.

**48. Які виділяють типи синтезу?**

- А) структурний;
- Б) функціональний;
- В) параметричний;
- Г) програмний.

**49. Які виділяють типи математичних моделей?**

- А) матричні;
- Б) функціональні;



- В) структурні;
- Г) моделі даних.

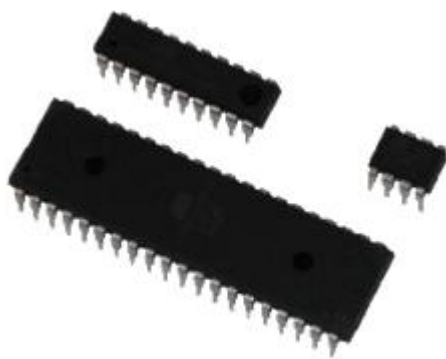
**50. В якому вигляді задаються структурні моделі?**

- А) графів;
- Б) таблиць;
- В) матриць;
- Г) блок-схем.

## ТЕМА 4. РОЗРОБКА МК-СИСТЕМ НА ОДНОКРИСТАЛЬНИХ МІКРОКОНТРОЛЕРАХ

### 4.1. Основні характеристики і класифікація однокристальних мікроконтролерів

Мікроконтролер є складним програмно-керованим цифровим пристроєм в мікроелектронному виконанні, виконаним у вигляді великої інтегральної мікросхеми (ВІС) або надвеликої інтегральної мікросхеми (НВІС) (рисунок 4.1). Тому він описується множиною параметрів, властивих як електронним приладам (швидкодія, споживана потужність, габарити і маса, кількість рівнів живлення, надійність, вартість, тип корпусу, температурний діапазон тощо), так і обчислювальним засобам (архітектура процесорного ядра, розрядність команд і даних, цикл виконання команд, число внутрішніх регістрів, тип пам'яті, склад резидентного і програмного забезпечення і т. д.). Мікроконтролери класифікують за найбільш істотними із перерахованих характеристик з метою вибору ефективної сфери застосування.



Зовнішній вигляд МК

Однокристальні мікроконтролери одержуються при реалізації всіх апаратних засобів контролера у вигляді однієї ВІС або НВІС. Всі операції, що виконуються ним, визначаються набором команд МК.

На сьогоднішній день найбільше поширення отримали однокристальні мікроконтролери, наведені в таблиці 4.1.

Таблиця 4.1 - Типи, класифікація і порівняльні характеристики ОМК

Сімейство МК	Фірма - виробник	Архітектура процесорного ядра	Розрядність даних (біт)	Частота синхронізації (МГц)
1	2	3	4	5
ST 62 M3820	SGT-Thomson Mitsubishi Electric	CISC CISC	8 8	8 8
K1816 BE51		CISC	8	12
Z86CXX	Zilog	CISC	8	12
AT 89	Atmel	CISC	8	12
COP800	National	CISC	8	20
MCS-51	Intel	CISC	8	12/ 24
MC68 HC05/08/11	Motorola	CISC	8	4/ 12/ 32

Продовження таблиці 4.1

M68HC16	Motorola	CISC	16	16
MCS-196/296	Intel	CISC	16	16/50
M683xx	Motorola	CISC	32	33
HT 4811/49100	Holtek	RISC	8	4
KP1878	Ангстрем	RISC	8	8
BE1/2	Microchip	RISC	8	4/20/33/40
PIC 12/16/17/18				
AVR 90	Atmel	RISC	8	20
SX 18/28AC/	Scenix	RISC	8	50/200
MPC50x	Motorola	RISC	32	25
MPC8xx	Motorola	RISC	32	50

В цій таблиці по суті наведена загальна класифікація МК за такими основними критеріями, як: архітектура процесорного ядра, розрядність команд, максимальна тактова частота.

З цієї таблиці видно, що більшість ОМК має традиційну (фон-нейманівську або Принстонську) CISC архітектуру, в якій команди і дані передаються по одній шині. Особливий клас представляють собою мікроконтролери, архітектура яких заснована на концепції роздільних шин і областей пам'яті для даних і команд (Гарвардська архітектура). Дані мікроконтролери мають RISC архітектуру, що забезпечує просту, але потужну систему команд, які виконуються за один цикл. До таких мікроконтролерів відносяться, зокрема, ОМК фірми Microchip сімейства PIC (12CXX, 16CXX, 17CXX, 18CXX).

У структуру ОМК сімейства PIC закладено багато різних функціональних особливостей, що роблять їх на сьогоднішній день одними із найбільш високопродуктивних, мікроспоживаючих, заводо захищених програмованих користувачем 8-бітних мікроконтролерів. Завдяки цим особливостям ОМК сімейства PIC можуть обробляти апаратно-програмним способом як дискретні, так і аналогові сигнали, формувати різного роду керуючі сигнали, а також здійснювати зв'язок між собою і з ЕОМ, що знаходиться на більш високому ієрархічному рівні в системі.

Фірмою Microchip здійснюється потужна програмна, апаратна та інформаційна підтримка своїх виробів через мережу Internet і широко розгалужену у всьому світі дилерську мережу.

Основні порівняльні характеристики ОМК серії K1816 (MCS-51), що стали по суті промисловим стандартом, і мікроконтролерів PIC приведені в таблиці 4.2.

Таблиця 4.2 - Порівняльні характеристики ОМК сімейств K1816 (MCS-51) і PIC

Основні характеристики		Тип мікроконтролера	
		K1816	PIC
Технологія		n-МОП	КМОП
Резидентна пам'ять програм	Тип	ПЗП/ППЗП	
	Об'єм	1К...4К	512...4К/8К
Ємність резидентної пам'яті даних, байт		64/128	25...192/454
Частота синхронізації, МГц		6/12	4/20/33/40
Тривалість циклу, мкс		2,5/1	0,2
Напруга живлення, В		+ 5	+ (2...6)
Енергоспоживання		300 мА (5 В, 6 МГц) < 300 мА (в режимі SLEEP)	< 2 мА (5 В, 4 МГц) 15 мкА (3 В, 32 кГц) < 1 мкА (3В, в режимі SLEEP)
Розрядність	команд, біт	8	12/14/16
	даних, біт	8	8

Мікроконтролери серії K1816 (MCS-51) мають традиційну Принстонську архітектуру, в якій команди і дані передаються по одній і тій же шині, вимагають одного джерела електроживлення напругою + 5 В, розсіюють потужність близько 1,5 Вт і працюють в діапазоні температур від 0 до 70 °С. По входах і виходах МК серії 1816 електрично сумісні з інтегральними мікросхемами ТТЛ. Мікроконтролер МК48 може працювати в діапазоні частот синхронізації від 1 до 6 МГц, а мінімальний час виконання команди становить 2,5 мкс. Мікроконтролер МК51 може працювати в діапазоні частот від 1,2 до 12 МГц, при цьому мінімальний цикл виконання команди становить 1 мкс, а швидкодія дорівнює одному мільйону коротких операцій за секунду.

Мікроконтролери серії PIC забезпечують більш ніж в 5 – 10 разів кращу продуктивність у порівнянні з більшістю найбільш поширених 8-бітових мікроконтролерів аналогічного класу, таких, наприклад, як ST62 фірми SGS-Thomson, MC68HC05...HC11 фірми Motorola, MCS-51 фірми Intel, K1816 фірми Ангстрем, Z86CXX фірми Zilog і COP800 фірми National.

Така продуктивність дозволяє реалізувати різні пристрої, працюючі в реальному масштабі часу, наприклад: дискові накопичувачі, різноманітні термінали, сканери, високошвидкісні системи керування автомобільними і електричними двигунами, зв'язні процесори і різноманітні пристрої вводу-виводу (рис. 4.2).

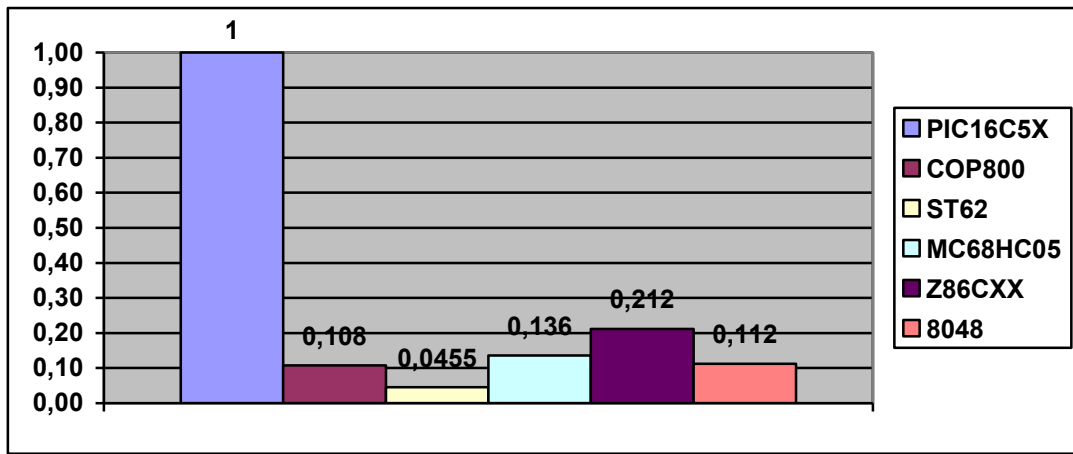


Рисунок 4.2 - Порівняння продуктивності мікроконтролерів

#### 4.2. Обґрунтування застосування та вибору сімейства ОМК

На системному і архітектурному рівнях проектування МК-систем і МКП завжди необхідно вирішувати задачу вибору ОМК. У цей час випускається велика кількість різних типів ОМК такими фірмами як Intel, Motorola, Zilog, Microchip та рядом інших. Однокристалні мікроконтролери стали сьогодні одним із самих найпоширеніших елементів програмованої логіки. Як же з "моря" пропозицій вибрати найбільш відповідний ОМК для реалізації систем і пристроїв, що проектуються?

Вибір ОМК для конкретного застосування є найменш вирішеною з численних проблем проектування мікроконтролерних систем і пристроїв. Це визначається постійним зростанням кількості ОМК, розширенням області їх застосування, а також відсутністю чіткої методики, що дозволяє зробити однозначний вибір ОМК.

ОМК є функціонально складним програмно керованим пристроєм, виконаним у вигляді ВІС, і характеризується великою кількістю параметрів. Тому задача вибору оптимального з технічної і економічної точок зору ОМК для конкретної області застосування є багатокритеріальною.

При виборі ОМК важливим є формування основних вимог, які пред'являються до пристроїв, що створюються. Пристрої з вбудованим ОМК, як правило, повинні задовольняти наступним вимогам:

- робота в реальному часі;
- підвищена надійність і заводо захищеність;
- простота обслуговування;
- наявність фіксованого набору задач, що багато разів вирішується протягом всього терміну служби пристрою;
- необхідна продуктивність;
- умови застосування.

Вибір ОМК звичайно проводиться з чотирьох основних позицій:

1) з точки зору системного проектування треба аналізувати наступні характеристики ОМК: тип архітектури, швидкодія, можливість переривання, типи і кількість портів вводу/виводу, ємність вбудованих ОЗП і ПЗП (ППЗП), наявність системи автоматизованого проектування тощо;

2) з точки зору розробки апаратних засобів МКП необхідно враховувати: технологію виготовлення ВІС, електричну сумісність з іншими ІМС, число джерел живлення і енергоспоживання, габарити, тип корпусу та число виводів, діапазон робочих температур тощо;

3) з точки зору розробки математичного забезпечення потрібно аналізувати: розрядність даних і команд, набір команд і способи адресації, наявність і організацію стека, час відлагодження робочих програм тощо;

4) з економічної точки зору визначальним параметром є вартість ОМК і МКП загалом.

Однією з основних характеристик, що відображають функціональні можливості ОМК, може служити його розрядність (розрядність даних і команд). Діапазон необхідної розрядності даних в МКП досить широкий і залежить від їх функціонального призначення. Для переважної більшості застосувань використовуються 8-розрядні ОМК, однак, в ряді випадків необхідно використовувати 16-розрядні (прецизійні контрольно-вимірювальні системи, системи збору і обробки даних) або навіть 32-розрядні ОМК (цифрові фільтри і спектральні аналізатори). У останніх випадках спільно з ОМК можуть використовуватися однокристальні мікроЕОМ і мікропроцесори з відповідною розрядністю, на які й покладаються задачі з обробки даних. ОМК в цьому випадку виконують в основному задачі з керування і вводу-виводу.

Після визначення допустимого класу ОМК, що задовольняють поставленим вимогам, проводиться оцінка програмних можливостей ОМК. При цьому потрібно враховувати наявність засобів автоматизованого програмування і відлагодження програмного забезпечення (редактори і компілятори, симулятори/відлагоджувачі, внутрішньосхемні емулятори, інтегровані середовища, програматори).

Системи команд ОМК відрізняються не тільки їх кількістю і розрядністю (8, 16 і 32 біти), але й можливостями організації компактних ланцюжків команд при програмуванні різних алгоритмів задач, що вирішуються. Тому порівняння лише кількості команд або їх розрядності не дає достатньої підстави для правильного вибору ОМК.

Для оцінки МКП часто використовується еталонне (пробне) програмування. Воно виконується для заздалегідь певного набору еталонних задач, що відображають специфіку області застосування. Знання області застосування ОМК дозволяє виділити найбільш специфічні та принципові частини алгоритмів і скласти програми для всіх допустимих типів ОМК з метою отримання експлуатаційних характеристик МКП. Із них найбільш важливі такі, як загальний час виконання програми, необхідна ємність ПЗП(ППЗП) і ОЗП, час реакції МКП на зовнішні сигнали і т.д.

Внаслідок еталонного програмування (тобто тестування) одержуються реальні експлуатаційні характеристики застосування конкретного ОМК в конкретній системі. Разом з інформацією фірми-розробника отримані дані, як правило, достатні для оцінки ОМК і МКП.

Найважливішим фактором, що впливає на вибір ОМК, є наявність програмного забезпечення і спеціальних технічних засобів для відлагодження

та автоматизації розробки робочих програм а також бібліотек стандартних (типових) програм і прикладів застосувань. Це визначається тим, що без програмного забезпечення ОМК не може використовуватися, а вартість його розробки може істотно перевищувати вартість апаратних засобів МКП або системи. При оцінці програмного забезпечення як критерія вибору ОМК потрібно враховувати альтернативу апаратного рішення задачі. Відомо, що пристрої і системи, що використовують апаратні засоби рішення задач, є більш швидкодіючими в порівнянні із МКП з програмною реалізацією їх рішення, але останні більш гнучкі (тобто можуть бути перепрограмовані). Крім того, вони мають різні складові загальних затрат на створення МКП:

- апаратні засоби рішення задач – низькі витрати на розробку і високі – на виробництво;
- при програмному рішенні задачі спостерігається протилежна картина.

При виборі ОМК важливо враховувати складність програмування і відлагодження (час відлагодження), які визначаються системою команд та архітектурою ОМК.

Мікроконтролери дуже складні пристрої, щоб їх можна було однозначно охарактеризувати переліком всіляких технічних характеристик, тим більше, що ряд характеристик не може бути оцінено кількісно (наприклад, складність програмування).

Основна мета вибору ОМК - обрати ОМК з мінімальною ціною (щоб знизити загальну вартість системи), але в той же час задовольняє системної специфікації, тобто вимогам по продуктивності, надійності, умовам застосування і т.д. Загальна вартість системи включає все: інженерне дослідження та розробку, виробництво (комплектуючі і праця), гарантійний ремонт, оновлення, обслуговування, сумісність, простоту в обігу і т.д.

Вибір ОМК є одним з найбільш важливих рішень, від яких залежить успіх або провал усього проекту. При виборі мікроконтролера існують численні критерії.

У багатьох випадках як критерії для порівняння і вибору класу (сімейства) ОМК досить використати наступні їх характеристики:

- 1) швидкодія;
- 2) час програмування і відлагодження;
- 3) енергоспоживання;
- 4) габарити;
- 5) вартість.

#### 4.3. Етапи розробки МК-систем на базі ОМК

МК-системи на основі ОМК використовуються найчастіше як вбудовані системи для вирішення завдань управління об'єктами. Важливою особливістю даного застосування є робота в реальному часі, тобто забезпечення реакції на зовнішні події протягом певного тимчасового інтервалу.

Технологія проектування МКС на ОМК повністю відповідає принципу нерозривної проектування і налагодження апаратних і програмних засобів, прийнятому в мікропроцесорній техніці. Це означає, що перед розробником такого роду МКС стоїть завдання реалізації повного циклу проектування, починаючи від розробки алгоритму функціонування і закінчуючи комплексними випробуваннями виробу, а, можливо, і супроводом при виробництві. Методологія проектування ОМК представлена так на рисунку 4.3.



Рисунок 4.3 - Основні етапи розробки МКС на ОМК

У технічному завданні формулюються вимоги до контролера з точки зору реалізації визначеної функції управління. Технічне завдання включає в себе набір вимог, який визначає, що користувач хоче від контролера і що розроблюваний прилад повинен робити. Технічне завдання може мати вигляд текстового опису.

На підставі вимог користувача складається функціональна специфікація, яка визначає функції, виконувані контролером для користувача після



завершення проектування, уточнюючи тим самим, наскільки пристрій відповідає висунутим вимогам. Вона включає в себе опис форматів даних, як на вході, так і на виході, а також зовнішні умови, що керують діями ОМК.

Функціональна специфікація і вимоги користувача є критеріями оцінки функціонування ОМК після завершення проектування. Може знадобитися проведення декількох ітерацій, що включають обговорення вимог і функціональної специфікації з потенційними користувачами ОМК, і відповідну корекцію вимог і специфікації. Вимоги до типу використовуваного ОМК формулюються на даному етапі найчастіше в неявному вигляді.

Етап розробки алгоритму управління є найбільш відповідальним, оскільки помилки даного етапу зазвичай виявляються тільки при випробуваннях закінченого виробу і призводять до необхідності дорогої переробки всього пристрою. Розробка алгоритму зазвичай зводиться до вибору одного з декількох можливих варіантів алгоритмів, що відрізняються співвідношенням обсягу програмного забезпечення і апаратних засобів.

При цьому необхідно виходити з того, що максимальне використання апаратних засобів спрощує розробку і забезпечує високу швидкодію ОМК в цілому, але супроводжується, як правило, збільшенням вартості і споживаної потужності. Пов'язано це з тим, що збільшення частки апаратних засобів досягається або шляхом вибору більш складного ОМК, або шляхом використання спеціалізованих інтерфейсних схем. І те, і інше приводить до зростання вартості і енергоспоживання. Збільшення питомої ваги програмного забезпечення дозволяє скоротити число елементів ОМК і вартість апаратних засобів, але це призводить до зниження швидкодії, збільшення необхідного обсягу внутрішньої пам'яті ОМК, збільшення термінів розробки та налагодження програмного забезпечення. Критерієм вибору тут і далі є можливість максимальної реалізації заданих функцій програмними засобами при мінімальних апаратних затратах і за умови забезпечення заданих показників швидкодії і надійності в повному діапазоні умов експлуатації. Часто визначальними вимогами є можливість захисту інформації (програмного коду) контролера, необхідність забезпечення максимальної тривалості роботи в автономному режимі та інші. В результаті виконання цього етапу остаточно формулюються вимоги до параметрів використовуваного ОМК.

При виборі типу ОМК враховуються наступні основні характеристики:

- розрядність;
- швидкодія;
- набір команд і способів адресації;
- вимоги до джерела живлення і споживана потужність в різних режимах;
- обсяг ПЗУ програм і ОЗП даних;
- можливості розширення пам'яті програм і даних;
- наявність та можливості периферійних пристроїв, включаючи засоби підтримки роботи в реальному часі (таймери, процесори подій і т.п.);

- можливість перепрограмування у складі пристрої;
- наявність і надійність засобів захисту внутрішньої інформації;
- можливість постачання в різних варіантах конструктивного виконання;
- вартість у різних варіантах виконання;
- наявність повної документації;
- наявність і доступність ефективних засобів програмування й налагодження ОМК;
- кількість і доступність каналів постачання, можливість заміни виробами інших фірм.

Список цей не є вичерпним, оскільки специфіка проектного пристрою може перенести акцент вимог на інші параметри ОМК. Визначальними можуть виявитися, наприклад, вимоги до точності внутрішнього компаратора напруг або наявність великої кількості вихідних каналів.

Номенклатура що випускаються в даний час ОМК обчислюється тисячами типів виробів різних фірм. Сучасна стратегія модульного проектування забезпечує споживача різноманітністю моделей ОМК з одним і тим же процесорним ядром. Така структурна різноманітність відкриває перед розробником можливість вибору оптимального ОМК, що не має функціональної надмірності, що мінімізує вартість комплектуючих елементів.

Однак для реалізації на практиці можливості вибору оптимального ОМК необхідна досить глибока опрацювання алгоритму керування, оцінка обсягу виконуваної програми і числа ліній сполучення з об'єктом на етапі вибору ОМК. Допущені на даному етапі прорахунки можуть згодом призвести до необхідності зміни моделі ОМК і повторної розведення друкованої плати макета контролера. У таких умовах доцільно виконувати попереднє моделювання основних елементів прикладної програми з використанням програмно-логічної моделі обраного ОМК.

При відсутності ОМК, що забезпечує необхідні по ТЗ характеристики проектного контролера, необхідний повернення до етапу розробки алгоритму керування і перегляд вибраного співвідношення між обсягом програмного забезпечення і апаратних засобів. Відсутність відповідного МК частіше за все означає, що для реалізації необхідного обсягу обчислень (алгоритмів керування) за відведений час потрібна додаткова апаратна підтримка. Негативний результат пошуку ОМК з необхідними характеристиками може бути пов'язаний також з необхідністю обслуговування великої кількості об'єктів управління.

На етапі розробки структури контролера остаточно визначається склад наявних і підлягають розробці апаратних модулів, протоколи обміну між модулями, типи роз'ємів. Виконується попередня опрацювання конструкції контролера. У частині програмного забезпечення визначаються склад і зв'язку програмних модулів, мова програмування. На цьому ж етапі здійснюється вибір засобів проектування та налагодження.

Можливість перерозподілу функцій між апаратними та програмними засобами на даному етапі існує, але вона обмежена характеристиками вже обраного МК. При цьому необхідно мати на увазі, що сучасні МК випускаються, як правило, серіями (сімействами) контролерів, сумісних програмно і конструктивно, але розрізняються за своїми можливостями (обсяг пам'яті, набір периферійних пристроїв і т.д.). Це дає можливість вибору структури контролера з метою пошуку найбільш оптимального варіанту реалізації.

Не можна не згадати тут про нової ідеології розробки пристроїв на базі МК, запропонованої фірмою "Scenix". Вона базується на використанні високошвидкісних RISC-мікроконтролерів серії SX з тактовою частотою до 100 МГц. Ці МК мають мінімальний набір вбудованої периферії, а все більш складні периферійні модулі емулюються програмними засобами. Такі модулі програмного забезпечення називаються "віртуальними периферійними пристроями", вони забезпечують зменшення числа елементів контролера, часу розробки, збільшують гнучкість виконання. До теперішнього часу розроблені цілі бібліотеки віртуальних пристроїв, що містять налагоджені програмні модулі таких пристроїв як послідовні інтерфейси, генератори та вимірники частоти, контролери переривань і багато інших.

#### 4.4. Особливості розробки прикладного програмного забезпечення ОМК

Як вже відмічалось, при проектуванні МК-систем передусім виникає необхідність рішення задачі про оптимальний (за рядом критеріїв) розподіл функцій між апаратними засобами і програмним забезпеченням.

При цьому в самому загальному випадку необхідно виходити з того, що використання спеціалізованих інтерфейсних ВІС спрощує розробку і забезпечує високу швидкодію системи загалом, але зв'язано із збільшенням вартості, об'єму і споживаної потужності. Більша питома вага програмного забезпечення дозволяє скоротити кількість компонентів системи і вартість її апаратних засобів, але це призводить до зниження швидкодії, збільшення витрат і термінів розробки та відлагодження прикладних програм. При цьому ще може дещо збільшитися і кількість ВІС зовнішньої пам'яті МК-системи. Рішення про вибір того або іншого варіанта розподілу функцій між апаратними і програмними засобами системи приймається в залежності від тиражності виробу, обмежень за вартістю, об'ємом, споживаною потужністю та швидкодією виробу. Попутно зазначимо, що термін існування виробу, в якого більша частина функцій реалізована в програмному забезпеченні, зростає в багато разів за рахунок того, що термін "морального старіння" виробу може бути істотно відсунений. Програмна реалізація основних елементів алгоритму роботи контролера допускає модифікацію відносно простими засобами (шляхом перепрограмування), в той час як можливість зміни вже існуючої фіксації елементів алгоритму в апаратурі контролера практично відсутня.

Досить поширена практика роботи "тандемом", коли над розробкою прикладних програм для МК спільно працюють професійний програміст і

непрограмуючий професіонал, тобто фахівець, який володіє “таємницями ремесла” в конкретній предметній області, має серйозним недоліком те, що при спробах викласти програмісту значення прикладної задачі, це значення часто вислизає. Внаслідок такої практики формалізуються і програмуються найбільш очевидні, грубо кажучи – тривіальні, прикладні задачі, а найбільш професіонально цікаві залишаються поза межами досяжності. Видимо, це пояснюється тим, що час, необхідний на формалізацію професійних знань при роботі “тандемом”, нерідко складає до 70 % всього часу, що вимагається для отримання закінченого мікроконтролерного виробу.

Робота “тандемом” у величезній більшості випадків призводить до того, що кінцевий користувач МК-системи відмовляється від своїх раніше сформульованих вимог на програму і стверджує, що “малося на увазі щось схоже, але не це”. Таке положення пояснюється тим, що початок процесу програмування задач, які ставить кінцевий користувач, негайно змінює його власне уявлення про ці задачі. Відмітимо попутно, що до 60 % помилок прикладних програм для МКП і МКС викликані не помилками в машинних кодах, не логічними помилками в програмі, а помилковою формалізацією прикладної задачі. Трудомісткість усунення цих помилок, напрацьованих “тандемом” (професійний програміст – непрограмуючий професіонал), така велика, що часто змушує приступити до розробки прикладної програми МК-системи наново і з іншими засобами.

Ресурси, що затрачуються власне на програмування, тобто на отримання машинних кодів, незначні в порівнянні з ресурсами, що затрачуються на процес формалізації прикладної задачі та розробку алгоритму, тому потрібно говорити не про проблему розробки прикладного програмного забезпечення МК-систем, а про проблему формалізації професійних знань кінцевого користувача мікроконтролерних виробів.

Подібно тому, як поява мікропроцесорних і мікроконтролерних засобів призвела до процесу, що продовжується переміщенням основного об’єму витрат на проектування контролерів із сфери розробки апаратних засобів в сферу розробки програмного забезпечення, так і стрімке розширення можливих областей застосування МК призводить до переміщення центра тягаря зусиль з розробки прикладного програмного забезпечення з фази реалізації на фазу постановки і формалізації задачі.

Структура трудовитрат, що склалася до теперішнього часу в розробці МПС, дозволяє виділити три основні стадії проектування прикладного програмного забезпечення:

- аналіз предметної області з метою визначення задач, автоматизація рішення яких на основі МК обіцяє найбільший ефект;
- розробку алгоритму рішення поставленої задачі (або комплексу задач);
- власне програмування, або, точніше, супровід розробки прикладних програм системними засобами підтримки проектування.

Розподіл трудовитрат в процентах по цих трьох стадіях виглядає приблизно так: 40 – 50 – 10. Це означає, що якщо перша стадія роботи вже

виконана з участю фахівця з системного аналізу, тобто, якщо задача вже поставлена, то найбільш складною, слабо формалізуємою і трудомісткою стадією роботи, (через тісну зв'язаність із областю застосування даної програми) є стадія аналізу задачі, її інженерної інтерпретації і розробки “функціональної специфікації” програми для формування алгоритму рішення поставленої задачі. Вся подальша робота з перетворення алгоритму в машинні коди, тобто, створення прикладного програмного забезпечення, представляє собою просто сукупність процесів трансляції. Ці процеси добре формалізуються і їх реалізація спирається на вже існуючі системні засоби підтримки (транслятори, редактори, відлагоджувачі). Саме внаслідок цього власне програмування вимагає тільки близько 10 % загальних трудовитрат. Очевидно, що основне творче навантаження при розробці прикладних програм для МК-систем несе не професійний програміст, а непрограмуючий професіонал – фахівець в даній області знань. Якщо цей фахівець оволодіє основами програмування і стане програмуючим професіоналом, то можна чекати, що процес формалізації його професійних знань буде протікати більш результативніше, ніж при “грі в зіпсований телефон”, тобто, при алгоритмізації прикладної задачі “тандемом”.

Орієнтація на розробку прикладних програм для МПС силами програмуючих професіоналів набуває поширення ще і тому, що в умовах швидко дешевіючої пам'яті змінилися стиль і технологія розробки програм. Економлять тепер вже не пам'ять МПС, а час розробника програмного забезпечення, тобто, скорочують терміни розробки виробу. Внаслідок цього прикладні програми, створені програмуючим професіоналом, з точки зору професійного програміста, часто виглядають незграбними і неграціозними. Але зате вони мають одну загальну перевагу – вони дійсно працюють в контролерах, чого не можна сказати про дев'ять з кожних десяти витончених програм, створених професійним програмістом, який не може (по визначенню) бути професіоналом і в кожній конкретній предметній області знань.

З урахуванням масштабів випуску і перспектив застосування засобів мікроконтролерної техніки вихід із становища, що склалося, бачиться в тому, щоб спонукати фахівців, працюючих в своїх предметних областях знань, взяти справу розробки прикладного програмного забезпечення МК-систем в свої руки повністю (при деякій технічній підтримці професійних програмістів). Для цього потрібні не дуже значні зусилля і первинні затрати: треба передусім наважитися взяти всю повноту відповідальності за програмне забезпечення на свої плечі, треба оволодіти однією з мов програмування і освоїти "кухню" програмної реалізації, обмеженої множиною найбільш вживаних процедур і функцій об'єкта автоматизації. Однак ця задача відносно легко вирішується з використанням методу декомпозиції (розбиття складної функції на безліч простих взаємопов'язаних функцій). До подібної постановки питання організації розробки прикладного програмного забезпечення для МК-систем приводить і цілком очевидне міркування про те, що швидке зростання кількості МК і областей їх застосувань не може більше супроводжуватися відповідним зростанням кількості програмістів.

Тестові питання до теми 4

**1. Однокристальний мікроконтролер – це:**

- А) спеціалізована МПС, що включає мікропроцесор, блоки пам'яті для збереження коду програм і даних, порти вводу-виводу і блоки зі спеціальними;
- Б) схема або пристрій, що розміщується між двома іншими пристроями для згладжування змін швидкості чи рівня або для забезпечення асинхронної роботи;
- В) електронна система для оброблення вхідних сигналів та видачі вихідних сигналів;
- Г) система управління сигналами.

**2. Які параметри ОМК характерні для електронних приладів:**

- А) швидкодія;
- Б) споживана потужність;
- В) цикл виконання команд;
- Г) тип пам'яті.

**3. Які параметри ОМК характерні для обчислювальних засобів:**

- А) вартість;
- Б) число внутрішніх регістрів;
- В) розрядність команд і даних;
- Г) температурний діапазон.

**4. Особливості ОМК сімейства PIC:**

- А) високопродуктивні;
- Б) термостійкі;
- В) заводо захищені;
- Г) жодної вірної відповіді.

**5. Ємність резидентної пам'яті даних (в байтах) ОМК сімейства PIC :**

- А) 64/128;
- Б) 25...192/254;
- В) 128/512;
- Г) 8...16/32.

**6. Тривалість циклу(мкс) ОМК сімейства PIC :**

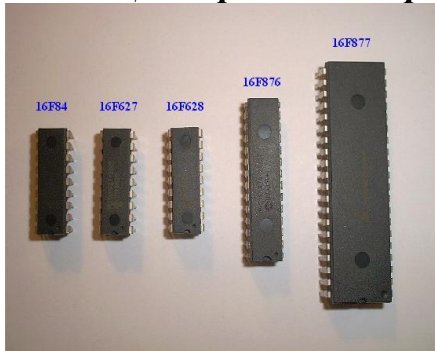
- А) 0,1;
- Б) 0,2;
- В) 0,3;
- Г) 0,4.

**7. Напруга живлення (Вольти) ОМК сімейства PIC :**

- А) + 5;
- Б) + 8;

- В) + (2..6);  
Г) + (5..8).

**8. Що зображено на рисунку:**



- А) ОМК сімейства PIC;  
Б) ОМК сімейства COP;  
В) ОМК сімейства ST62;  
Г) ОМК сімейства K1816.

**9. Ємність резидентної пам'яті даних (в байтах) ОМК сімейства K1816:**

- А) 64/128;  
Б) 25...192/254;  
В) 128/512;  
Г) 8...16/32.

**10. Тривалість циклу(мкс) ОМК сімейства K1816 :**

- А) 1,5/1;  
Б) 2,5/1;  
В) 3,5/6;  
Г) 4,5/5.

**11. Напруга живлення (Вольти) ОМК сімейства K1816:**

- А) + 5;  
Б) + 8;  
В) + (2..6);  
Г) + (5..8).

**12. Яка технологія використовується в мікро контролерах PIC:**

- А) n-GTR;  
Б) ASDE;  
В) n-МОП;  
Г) КМОП.

**13. Яка технологія використовується в мікро контролерах K1816:**

- А) КМОП ;  
Б) n-МОП;  
В) n-GTR;

Г) ASDE.

**14. Яким вимогам повинні задовольняти пристрої з вбудованим ОМК:**

- А) підвищена надійність і завадозахищеність;
- Б) робота в реальному часі;
- В) мала тепловіддача;
- Г) жодної вірної відповіді.

**15. Для чого використовується еталонне (пробне) програмування:**

- А) для оцінки МКП;
- Б) для оцінки тривалості роботи;
- В) для оцінки НК;
- Г) для оцінки МКС.

**16. Скількох кристалні мікроконтролери стали сьогодні одним із самих найпоширеніших елементів програмованої логіки:**

- А) однокристалні;
- Б) двокристалні;
- В) чотирьох кристалні;
- Г) вісьмикристалні.

**17. ОМК є функціонально складним програмно керованим пристроєм, виконаним у вигляді:**

- А) ВІС;
- Б) DІС;
- В) PІС;
- Г) МНІ.

**18. Де застосовуються 32-розрядні ОМК :**

- А) спектральні аналізатори;
- Б) цифрові фільтри ;
- В) комутаторні схеми;
- Г) аналогові резистори.

**19. Що потрібно враховувати при оцінці програмних можливостей ОМК:**

- А) наявність засобів автоматизованого програмування ;
- Б) відлагодження програмного забезпечення ;
- В) функціональні можливості;
- Г) тактову частоту.

**20. Які експлуатаційних характеристик МКП:**

- А) тактова частота;
- Б) необхідна ємність ПЗП(ППЗП) і ОЗП;
- В) загальний час виконання програми;
- Г) Всі варіанти правильні .



**21. Які критерії вибору ОМК:**

- А) обрати ОМК з мінімальною ціною;
- Б) обрати ОМК з найбільшою продуктивністю;
- В) обрати ОМК що задовольняє системні специфікації;
- Г) обрати ОМК з найменшою енергоємністю.

**22. Загальна вартість системи включає в себе:**

- А) інженерне дослідження та розробку;
- Б) гарантійний ремонт;
- В) оновлення;
- Г) всі варіанти вірні.

**23. З якої точки зору реалізації формується вимоги до контролера у технічному завданні:**

- А) найбільшій продуктивності;
- Б) функції розподілу;
- В) визначеної функції управління;
- Г) функціональної специфікації.

**24. На підставі чого складається функціональна специфікація:**

- А) на підставі наданої звітності;
- Б) на підставі вимог користувача;
- В) на підставі розрахунків продуктивності;
- Г) немає вірної відповіді.

**25. Що включає в себе функціональна специфікація:**

- А) опис форматів даних, як на вході, так і на виході;
- Б) зовнішні умови, що керують діями МК;
- В) внутрішні умови, що керують діями МК;
- Г) детальний опис вихідної інформації.

**26. Які критерії оцінки функціонування МК після завершення проектування:**

- А) функціональна специфікація;
- Б) внутрішні умови функціонування;
- В) зовнішні умови функціонування;
- Г) вимоги користувача.

**27. У якому вигляді найчастіше формуються вимоги до типу використовуваного МК на етапі розробки ТЗ:**

- А) у легкозрозумілому вигляді;
- Б) у неявному вигляді;
- В) у чітко структурованому вигляді;
- Г) у явному вигляді.

**28. *Етап розробки алгоритму управління є :***

- А) найбільш відповідальним;
- Б) найбільш часоємним;
- В) найбільш структуризованим;
- Г) усі відповіді вірні.

**29. *Чим відрізняються можливі варіанти алгоритмів при розробці алгоритму управління:***

- А) складністю;
- Б) співвідношенням обсягу затраченого часу на їх виконання;
- В) співвідношенням обсягу програмного забезпечення і апаратних засобів;
- Г) всі відповіді вірні.

**30. *Максимальне використання апаратних засобів спрощує розробку і забезпечує високу швидкодію МК в цілому, але супроводжується, як правило:***

- А) збільшенням вартості ;
- Б) збільшенням споживаної потужності;
- В) збільшенням часу виконання;
- Г) всі відповіді вірні.

**31. *До чого призводить збільшення питомої ваги програмного забезпечення:***

- А) до скорочення числа елементів МК;
- Б) до збільшення швидкодії;
- В) до зменшення необхідного обсягу внутрішньої пам'яті МК;
- Г) до зменшення вартості апаратних засобів.

**32. *В результаті виконання етапу розробки алгоритму управління остаточно формулюються:***

- А) алгоритм управління;
- Б) вимоги до параметрів використовуваного МК;
- В) опис функції управління;
- Г) чітко визначена межа між апаратними і програмними засобами.

**33. *На етапі виборі типу ОМК враховуються наступні основні характеристики:***

- А) розрядність;
- Б) швидкодія;
- В) тепловіддача;
- Г) всі відповіді вірні.

**34. *На етапі виборі типу ОМК враховуються наступні основні характеристики:***

- А) обсяг ПЗУ програм і ОЗП даних;
- Б) вимоги до джерела живлення і споживана потужність в різних режимах;

- В) можливості розширення пам'яті програм і даних;
- Г) всі відповіді вірні.

**35. На етапі виборі типу ОМК враховуються наступні основні характеристики:**

- А) наявність та можливості периферійних пристроїв, включаючи засоби підтримки роботи в реальному часі (таймери, процесори подій і т.п.);
- Б) можливість перепрограмування у складі пристрої;
- В) наявність і надійність засобів захисту внутрішньої інформації;
- Г) всі відповіді вірні.

**36. На етапі розробки структури контролера остаточно:**

- А) визначається склад наявних і підлягають розробці апаратних модулів;
- Б) протоколи обміну між модулями;
- В) обсяг ПЗУ програм і ОЗП даних;
- Г) набір команд і способів адресації.

**37. У частині програмного забезпечення на етапі розробки структури апаратних і програмних засобів визначається:**

- А) типи роз'ємів;
- Б) склад і зв'язка програмних модулів;
- В) мова програмування;
- Г) обсяг ПЗУ програм і ОЗП даних.

**38. Чим обмежена можливість перерозподілу функцій між апаратними та програмними засобами на етапі розробки структури апаратних і програмних засобів:**

- А) характеристиками обраного МК;
- Б) часом виконання програмних модулів;
- В) часом виконання апаратних модулів;
- Г) кількістю даних на вході.

**39. Як саме випускаються сучасні МК:**

- А) серіями контролерів;
- Б) сімействами контролерів;
- В) партіями контролерів;
- Г) асортиментом контролерів.

**40. Скільки людей працює при роботі "тандемом":**

- А) 1;
- Б) 2;
- В) 4;
- Г) 7;

**41. При роботі “тандемом” не формалізуються і програмуються такі задачі:**

- А) професійно цікаві;
- Б) тривіальні;
- В) найбільш очевидні;
- Г) прикладні.

**42. Який час, необхідний на формалізацію професійних знань при роботі “тандемом”:**

- А) до 50%;
- Б) до 60%;
- В) до 70%;
- Г) до 80%.

**43. До 60 % помилок прикладних програм для МКП і МКС викликані:**

- А) помилками в машинних кодах;
- Б) не логічними помилками в програмі;
- В) помилковою формалізацію прикладної задачі;
- Г) всі відповіді вірні.

**44. Куди сьогодні спостерігається переміщення центра тягаря зусиль з розробки прикладного програмного забезпечення:**

- А) з фази постановки на фазу формалізації;
- Б) з фази постановки на фазу реалізації;
- В) з фази формалізації і реалізації на фазу постановки;
- Г) з фази реалізації на фазу постановки і формалізації задачі.

**45. Який розподіл трудовитрат в процентах по цих трьох стадіях проектування прикладного програмного забезпечення:**

- А) 40 – 50 – 10;
- Б) 40 – 30 – 30;
- В) 10 – 50 – 40;
- Г) 30 – 30 – 20.

**46. З точки зору системного проектування треба аналізувати наступні характеристики ОМК:**

- А) тип архітектури;
- Б) можливість переривання;
- В) технологію виготовлення ВІС;
- Г) час відлагодження робочих програм.

**47. З точки зору розробки апаратних засобів МКП необхідно враховувати:**

- А) тип архітектури;
- Б) технологію виготовлення ВІС;
- В) тип корпусу та число виводів;

Г) наявність і організацію стека.

**48. Вибір ОМК з точки зору розробки математичного забезпечення потрібно аналізувати:**

- А) розрядність даних і команд;
- Б) число джерел живлення і енергоспоживання;
- В) габарити;
- Г) набір команд і способи адресації.

## ТЕМА 5. СТРУКТУРНА ОРГАНІЗАЦІЯ ТА РЕЖИМИ РОБОТИ ОМК РІС

### 5.1. Загальні відомості про ОМК РІС та їх класифікація

У 1975 році фірма GI розробила периферійний контролер (Peripheral Interface Controller або PIC), призначений для підтримки вводу/виводу 16-розрядного процесора. Від нього не вимагалася складна обробка інформації, тому його набір команд був сильно обмежений, але майже всі команди в ньому виконувалися за один машинний цикл. Цей контролер, що мав RISC-архітектуру, став прообразом сьогоденної архітектури мікроконтролерів РІС, які випускаються з кінця 80-х років компанією Arizona Microchip Technology Ltd., дочірньою компанією GI Microelectronics Inc.

Мікроконтролери сімейства РІС по'єднують всі передові технології ОМК: одноразово або багаторазово електрично перепрограмуємі користувачем ППЗП, мінімальне енергоспоживання, виключну продуктивність, потужну RISC-архітектуру і мінімальні розміри корпусу. Ці широкі можливості та низька вартість зробили серію мікроконтролерів РІС однією з кращих виборів для інженерних застосувань. Понад 250 мільйонів мікроконтролерів РІС використовується в декількох тисячах прикладень по всьому світу. Використовувати ці мікроконтролери рекомендується в усіх випадках, коли критичні енергозбереження, габарити і вартість пристрою.

Ось тільки деякі приклади застосування мікроконтролерів РІС:

- Комп'ютери і периферія: принтери, плотери, мережні картки, модеми, мишки, сканери, накопичувачі на гнучких і жорстких магнітних дисках, CD ROM, мультимедійні пристрої тощо.
- Радіотехніка (ТНП): CD програвачі, аудіосистеми, системи синтезу мовних повідомлень, блоки дистанційного керування, модулі телетексту, відеоігри.
- Техніка зв'язку: модеми, радіомодеми, мікро-АТС, автовідповідачі, мобільні телефони, пейджері, факс-апарати та ін.
- Промислові контролери: інтелектуальні датчики, пристрої попередньої обробки даних та керування (наприклад, електродвигунами), промислові роботи, регулятори температури, вологості, тиску та ін.
- Автомобільна електроніка: системи керування запалюванням, мікрокліматом і вприском палива, панелі приладів, радарні детектори, автомобільні сигналізації, комбіновані вимірювальні пристрої.
- Побутова техніка: системи сигналізації, вимірювальні пристрої, лічильники води, газу та електроенергії, детектори іонізуючого випромінювання, іграшки і т.д.

В залежності від розрядності команд, архітектурних особливостей і функціональних можливостей, мікроконтролери РІС поділяються на чотири основні групи (сімейства):

1. Сімейство найпростіших ОМК (12-розрядні команди) – РІС 12СХХ;
2. Базове сімейство (12-розрядні команди) – РІС 16 С5ХХ;

3. Розширене сімейство (14-розрядні команди) – PIC 16CXX;

4. Високопродуктивні сімейства (16-розрядні команди) – PIC 17CXX та PIC 18CXX.

Архітектура ОМК PIC основана на концепції роздільних шин та областей пам'яті для даних і команд (Гарвардська архітектура). Шина і пам'ять даних (ОЗП) має ширину 8 бітів, а програмна шина і пам'ять команд (ПЗП або ППЗП) має ширину 12, 14 або 16 бітів в залежності від сімейства ОМК. Така концепція забезпечує просту, але потужну систему команд, а двоступеневий конвеєр забезпечує їхню одночасну вибірку і виконання. Всі команди складаються з одного слова (шириною 12, 14 або 16 бітів) і виконуються за один цикл (200 нс при тактовій частоті 20 МГц), крім команд переходу, які виконуються за два цикли. Контрольні випробування показують, що застосування ОМК серії PIC дозволяють зменшити час налагодження в 1,5 – 2 рази у порівнянні із звичайними 8-розрядними мікроконтролерами.

Система команд ОМК PIC включає тільки 33/35/57 команд і може бути легко і швидко вивчена. В конструкцію PIC включено багато особливостей по енергозбереженню, що роблять їх на сьогоднішній день найбільш мікроспоживаючими (в режимі SLEEP струм, що споживається, менше 1 мкА), самими низьковольтними за напругою живлення (2В), що програмується користувачем, мікроконтролерами.

Найпростіші типи таких ОМК містять 8-бітний таймер-лічильник з 8-бітним програмуємим попереднім дільником (фактично 16-бітний таймер) і 6 – 20 ліній двонаправленого вводу-виводу. Корпус таких ОМК має 8 (18) виводів. Мікроконтролери розширеного і високопродуктивного сімейства містять окрім цього цілий ряд додаткових функціональних вузлів та блоків, таких, наприклад, як: багатоканальні аналого-цифрові перетворювачі, розгалужену систему переривань, блоки керування рідиннокристалічними індикаторами, компаратори, широтно-імпульсні модулятори, паралельні та послідовні інтерфейси типу I2C, RS-232 і т. д., цифрові помножувачі, додаткові таймери-лічильники, збільшену кількість портів вводу-виводу дискретних сигналів та інше.

Таким чином, ОМК PIC мають вагомі переваги у порівнянні з іншими типами мікроконтролерів того ж класу.

В нинішній час випускаються мікроконтролери з різною ємністю постійної і оперативної пам'яті, з різними типами тактових генераторів, з різною швидкодією і конструктивним виконанням, а також з різними функціональними можливостями. Конкретний тип мікроконтролера для вирішення певної задачі можна вибрати на підставі критеріїв, які були розглянуті у попередніх розділах.

В залежності від технології виготовлення ПЗП всі типи МК поділяються на п'ять груп:

1. Мікроконтролери, які можуть бути багаторазово запрограмовані користувачем, що, в свою чергу можуть бути поділені також на дві наступні групи:

- *Мікроконтролери з ультрафіолетовим стиранням.* Ці МК оптимальні

для експериментальних розробок і налагодження програм. Необхідну конфігурацію тактового генератора <RC>, <XT>, <HS> або <LP> встановлюють програмним шляхом. За замовчуванням встановлюється тип генератора <RS>. В залежності від вибраного типу генератора і частоти, напруга живлення повинна бути у тому ж діапазоні, що і для пристроїв OTP/QTP, розглянутих нижче.

- *Мікроконтролери, ПЗП програм і даних яких можуть бути багаторазово електрично перепрограмовані користувачем (EEPROM)*. Ці МК (такі наприклад, як PIC 16C84X чи 16F84X) дозволяють легко підстроювати програму і дані під конкретні вимоги навіть після завершення її відлагодження і тестування. Ця можливість може бути використана як для тиражування, так і для занесення калібровочних даних вже після остаточного тестування розробленого мікроконтролерного пристрою (МКП). Проте дані МК мають обмежену кількість циклів перепрограмування.

2. Мікроконтролери, які одноразово програмується (OTP). Ці МК можуть бути одноразово запрограмовані користувачем і застосовуються в тих випадках, коли немає потреби часто змінювати зміст програми або конфігурацію мікроконтролера в розробленому МКП. Для деяких типів МК (таких наприклад, як PIC 16C54, PIC 16C55, PIC 16C57 і т.д.) тип генератора задається на підприємстві, яке їх виготовляє, і мікроконтролер тестується тільки для заданого типу генератора. Для інших типів МК (таких, наприклад, як PIC16C52, PIC16C54A, PIC16C58A і т.д.) тип генератора програмується користувачем. Додатково можна при цьому підключити сторожовий таймер і (або) логіку захисту програм від зчитування. Також можна запрограмувати 16 спеціальних бітів для ідентифікації МК.

3. Мікроконтролери, які програмується на підприємстві, що їх виготовляє (QTP). Ці МК замовляються і повністю програмується на підприємстві, яке їх виготовляє, за заздалегідь наданою користувачем інформацією. Тобто, це також мікроконтролери, що програмується одноразово (OTP) з єдиною різницею, що програмування здійснюється не користувачем, а на підприємстві, яке їх виготовляє.

4. Мікроконтролери, які послідовно програмується на підприємстві, що їх виготовляє (SQTP). Це також замовні МК типу QTP, які одноразово програмується на підприємстві, що їх виготовляє, у яких декілька комірок, що задаються користувачем, в кожному мікроконтролері програмується різними серійними номерами. Причому, ці номери можуть бути випадковими, псевдовипадковими або послідовними. Послідовне програмування дозволяє кожному МК мати власний номер, що може бути використано в якості пароля, коду доступу чи ідентифікації.

5. Масочні мікроконтролери (ROM). Ці МК також являються замовними і забезпечують максимально низьку вартість при великих серійних замовленнях (наприклад, такими МК є PIC16CR54, PIC16CR56, PIC16CR57, PIC16CR58 і т.д.).



## 5.2. Однокристалні мікроконтролери базового сімейства PIC16C5X

Сімейство мікроконтролерів PIC16C5X почала випускати фірма Microchip першим і тому воно одержало назву “базового”. Структурна схема ОМК PIC16C5X показана на рисунку 5.1. Основу структури даного мікроконтролера складають дві внутрішні шини: двонаправлена 8-бітна шина даних і 12-бітна шина команд. Шина даних зв'язує між собою всі основні функціональні блоки МК: пам'ять даних (ОЗП); арифметичнологічний пристрій (АЛП); порти вводу-виводу (PORT A, B і C); регістри стану (STATUS), побічної (непрямої) адресації (FSR), таймера-лічильника (RTCC/TMRO), програмного лічильника (PCL).

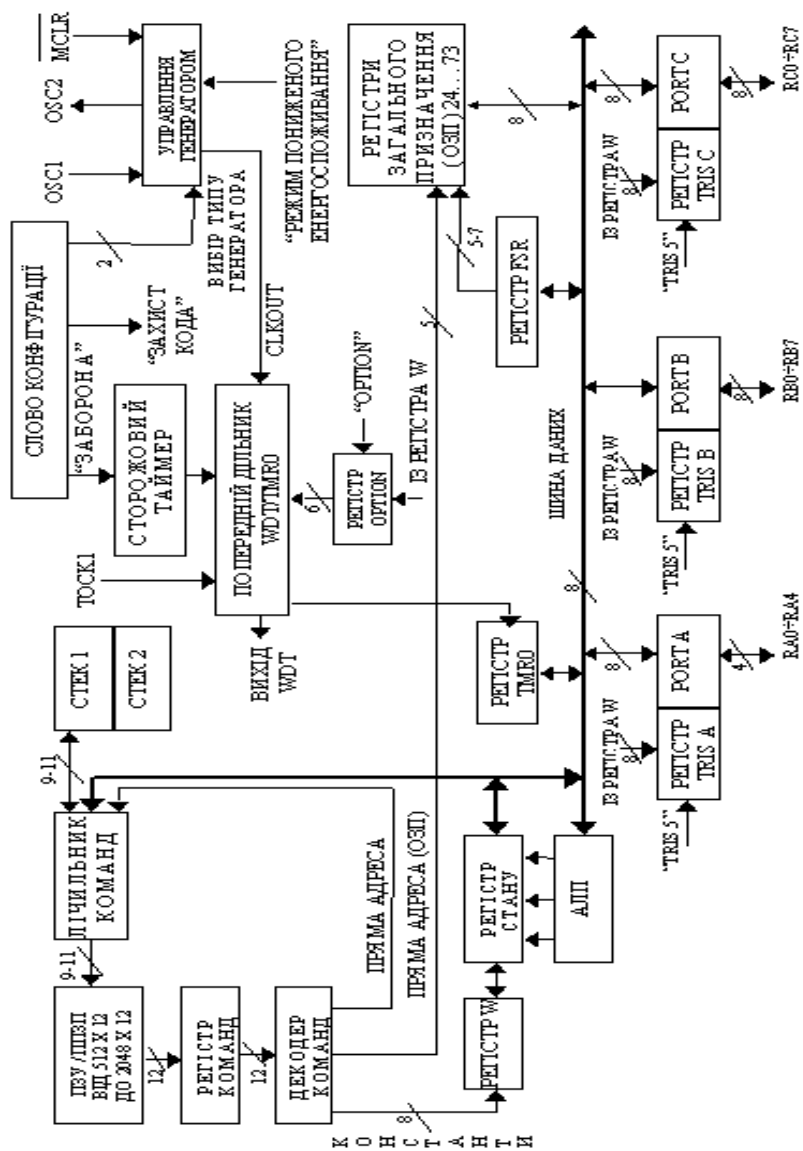


Рисунок 5.1 – Структурна схема мікроконтролера

Регістри загального призначення (Register File) створюють пам'ять даних і використовуються програмою для зберігання змінних за розсудом користувача.

Регістри поділяються на дві функціональні групи: спеціальні регістри і регістри загального призначення. Спеціальні регістри включають в себе регістр таймера/лічильника реального часу (TMRO/RTCC), лічильник команд (PC), регістр стану (STATUS), регістри вводу/виводу (PORT) і регістр непрямої адресації (FSR). Крім того, спеціальні регістри керують конфігурацією портів вводу-виводу (TRIS) і режимом попереднього дільника (OPTION).

В мікроконтролерах сімейства PIC 16C5X існує пряма та непряма адресація всіх регістрів і комірок пам'яті даних. Всі спеціальні регістри та лічильник команд також відображаються на пам'ять даних і розміщуються в молодших адресах 0-ї сторінки (0-го банку).

Мікроконтролери PIC16C5X мають ортогональну (симетричну) систему команд, що дозволяє виконувати будь-яку операцію з будь-яким регістром, використовуючи будь-який метод адресації. Це полегшує їх програмування і значно зменшує час, необхідний для навчання роботі з ними.

В мікроконтролерах PIC16C5X є 8-розрядний арифметичнологічний пристрій (АЛП) і робочий регістр W. Арифметичнологічний пристрій виконує додавання, віднімання, зсув, бітові та логічні операції. У командах, що мають два операнди, одним із операндів є робочий регістр W. Другий операнд може бути константою або вмістом будь-якого регістра ОЗП. У командах із одним операндом, операнд може бути вмістом робочого регістра або вмістом будь-якого регістра. Для виконання всіх операцій АЛП використовується робочий регістр W, що не може бути прямо адресований. В залежності від результату виконання операції, можуть змінитися значення бітів переносу C, десяткового переносу DC і нуля Z у регістрі стану STATUS. При відніманні біти C і DC працюють як біти позики і десяткової позики, відповідно. В описі команд SUBWF і ADDWF наведені необхідні приклади.

До складу мікроконтролерів PIC16C5X входить ще цілий ряд функціональних блоків і вузлів, таких, наприклад, як: ПЗП/ППЗП програм (ROM/EPROM), лічильник команд або програмний лічильник (PC), стек (Stack 1,2), регістри керування портами вводу-виводу (TRIS A, B, C), сторожовий таймер Watch Dog Timer (WDT), попередній дільник Prescaler (PRS) для WDT і TMRO/RTCC, слово конфігурації Configuration EPROM (CONE), яке топологічно розташоване в ПЗП програм, блок генераторів тактових послідовностей і сигналів керування Oscillator Timing & Control (OSTC).

Вхідна тактова частота, яка надходить з виводу OSC1/CLKIN, всередині МК ділиться на чотири і з неї формуються чотири циклічні тактові послідовності Q1, Q2, Q3 і Q4, що не перекриваються за часом (рисунок 5.2).

Вибірка команди та її виконання суміщені за часом таким чином, що вибірка команди займає один цикл, а її виконання наступний цикл. Ефективний час виконання команди складає один цикл. Якщо команда змінює лічильник

команд (наприклад, команда GOTO), то для виконання цієї команди необхідно два цикли.

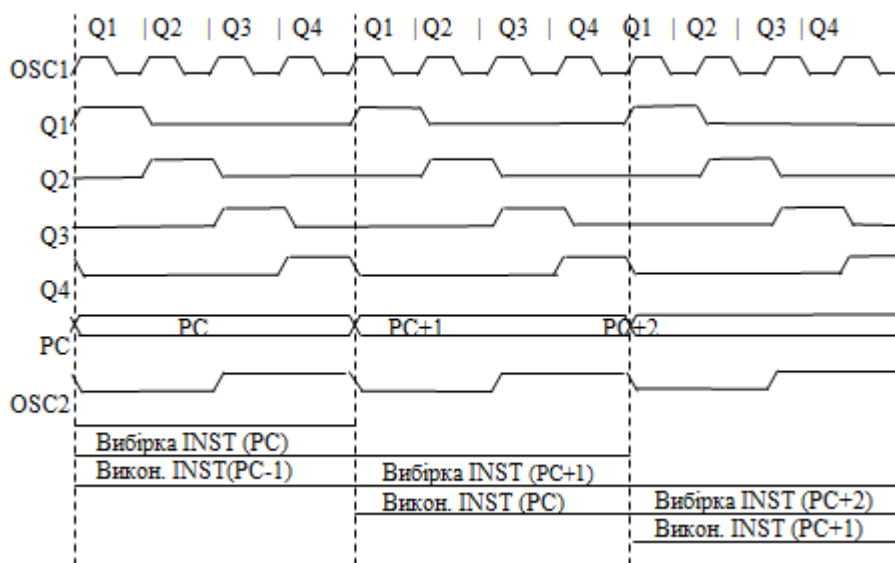


Рисунок 5.2. Цикл виконання команди

Цикл вибірки починається із збільшення лічильника команд у такті Q1. Команда зчитується з ПЗП пам'яті програм в тактах Q2 і Q3 та записується в регістр команд в такті Q4.

В циклі виконання команди вибрана команда зацепується в регістр команд в такті Q1. На протязі тактів Q2, Q3 і Q4 відбувається декодування і виконання команди. В такті Q3 зчитується пам'ять даних (читання операнда), а запис відбувається в такті Q4.

Таким чином, цикл виконання команди (рисунок 5.2) складається з 4-х тактів Q1 – Q4, в кожному з яких виконуються різні заздалегідь заплановані дії:

- Q1 – вибірка певної команди з пам'яті програм і її декодування або змушений NOP – “Вибірка INST (PC + 1)”;
- Q2 – вибірка даних або NOP;
- Q3 – виконання команди і обробка даних – “Виконання INST (PC)”;
- Q4 – Запис даних або NOP.

Розглянемо послідовно основні елементи структури, а також особливості організації і функціонування МК.

Позначення виводів PIC16C5X і їхнє функціональне призначення наведені на рисунку 5.3 а також в таблиці 5.1 і 5.2.

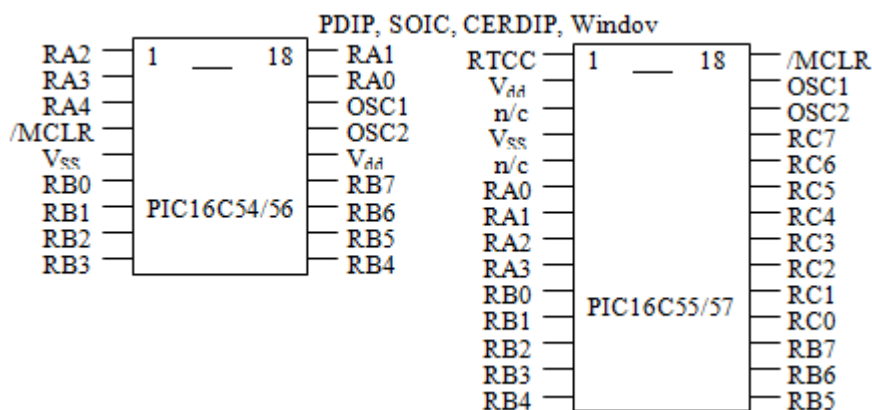


Рисунок 5.3 - Розташування виводів PIC16C5X

Таблиця 5.1 - Позначення і функціональне призначення виводів ОМК PIC 16C52/16C54/16C56/16C58

Наймен.	Тип корпусу		Тип	Буфер	Примітка
	DIP SOIC	SSOP			
RA0	17	19	I/O	TTL	Двонаправлений порт вводу-виводу
RA1	18	20	I/O	TTL	
RA2	1	1	I/O	TTL	
RA3	2	2	I/O	TTL	
	DIP SOIC	SSOP			
RB0	6	7	I/O	TTL	Двонаправлений порт вводу-виводу
RB1	7	8	I/O	TTL	
RB2	8	9	I/O	TTL	
RB3	9	10	I/O	TTL	
RB4	10	11	I/O	TTL	
RB5	11	12	I/O	TTL	
RB6	12	13	I/O	TTL	
RB7	13	14	I/O	TTL	
TOCKI	3	3	1	ST	Вхід таймера TMRO
$\overline{\text{MCLR}}/V_{PP}$	4	4	1	ST	Вхід скидання/напруги програмування. Скид низьким рівнем
OSC1/CLKI N	16	18	1	ST	Вхід генератора або зовнішня тактова частота
OSC2/CLKO UT	15	17	0	-	Вихід генератора. Підключається до резонатора. В режимі RC вихід ¼ тактової частоти OSC1
$V_{DD}$	14	15,16	P	-	Позитивна напруга живлення
$V_{SS}$	5	5,6	P	-	Загальний вихід

Таблиця 5.2 - Позначення і функціональне призначення виводів  
ОМКРІС 16С55/16С57

Наймен.	Тип корпусу		Тип	Буфер	Примітка
	DIP SOIC	SSOP			
RA0	6	5	I/O	TTL	Двонаправлений порт вводу-виводу
RA1	7	6	I/O	TTL	
RA2	8	7	I/O	TTL	Двонаправлений порт вводу-виводу
RA3	9	8	I/O	TTL	
RB0	10	9	I/O	TTL	Двонаправлений порт вводу-виводу
RB1	11	10	I/O	TTL	
RB2	12	11	I/O	TTL	
RB3	13	12	I/O	TTL	
RB4	14	13	I/O	TTL	
RB5	15	15	I/O	TTL	
RB6	16	16	I/O	TTL	
RB7	17	17	I/O	TTL	
RC0	18	18	I/O	TTL	Двонаправлений порт вводу-виводу
RC1	19	19	I/O	TTL	
RC2	20	20	I/O	TTL	
RC3	21	21	I/O	TTL	
RC4	22	22	I/O	TTL	
RC5	23	23	I/O	TTL	
RC6	24	24	I/O	TTL	
RC7	25	25	I/O	TTL	
TOCKI	1	2	I	ST	Вхід таймера TMR0
MCLR/Vpp	28	28	I	ST	Вхід скидання/напруги програмування. Скид низьким рівнем
OSC1/ CLKIN	27	27	I	ST	Вхід генератора/зовнішня тактова частота
OSC2/CLKOUT	26	26	O	—	Вихід генератора. Підключається до резона- тора. В режимі RC вихід 1/4 тактової частоти OSC1
VDD	2	3,4	P	—	Позитивна напруга живлення

Позначення: I – вхід, O – вихід, I/O – вхід/вихід, P – живлення, — – не використовується, TTL – вхід TTL, ST – вхід з тригером Шмітта.

Пам'ять програм (ПЗП/ППЗП) має сторінкову організацію (рисунок 5.4). Обсяг однієї сторінки 512 байтів, тому пряма адресація пам'яті програм в сімействі PIC16C5X можлива в межах 512 байтів. Мікроконтролери PIC 16C5X мають одну, дві або чотири сторінки пам'яті програм для зберігання 12-розрядних кодів команд (512 x 12, 1024 x 12, 2048 x 12). У відповідності з цим всі ОМК даного сімейства можна поділити на три групи. Доступ до пам'яті понад 512 байтів здійснюється після переключення на відповідну сторінку

пам'яті. МК PIC16C52/54/C54A/CR54/CR54A/ CR54B/C55 мають одну сторінку пам'яті програми, PIC16C56/CR56 мають 2 сторінки по 512 байтів, PIC16C57/CR57A/CR57B/C58A/CR58A/CR58B мають 4 сторінки по 512 байтів кожна.

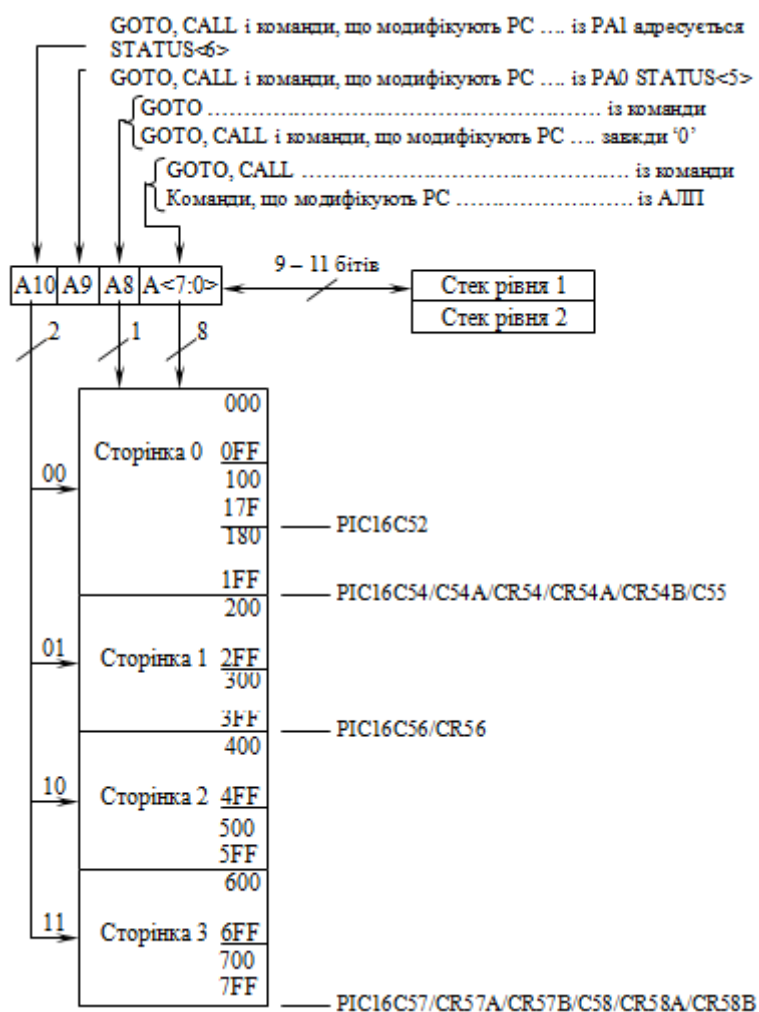


Рисунок 5. 4 - Організація пам'яті програм

Пам'ять даних (ОЗП) також має сторінкову організацію (рисунок 5.5). Вона складається максимум з 4-х банків (0... 3). Причому, банки в різних типах PIC можуть мати різні обсяги. Мінімальний обсяг банку 16 байтів, а максимальний – 32 байти. Шина даних розрядністю 8 бітів з'єднує групу регістрів ОЗП, порти вводу-виводу і 8-розрядний арифметичнологічний пристрій. Прямо адресуються 32 байти оперативної пам'яті, інша пам'ять адресується банками по 16 байтів кожний.

Мікроконтролери PIC16C52/C54/C54A/CR54/CR54A/CR54B/C56/CR56 також мають один банк оперативної пам'яті ємністю 32 байти, з яких 25 байт можуть бути використані в якості регістрів загального призначення. Для PIC16C55, що має PORTC, обсяг – 24 байт. PIC16C57/CR57A/CR57B і PIC16C58A/CR58A/CR58B мають 4 банки пам'яті даних. Ємність для них 72 і 73 байти відповідно.

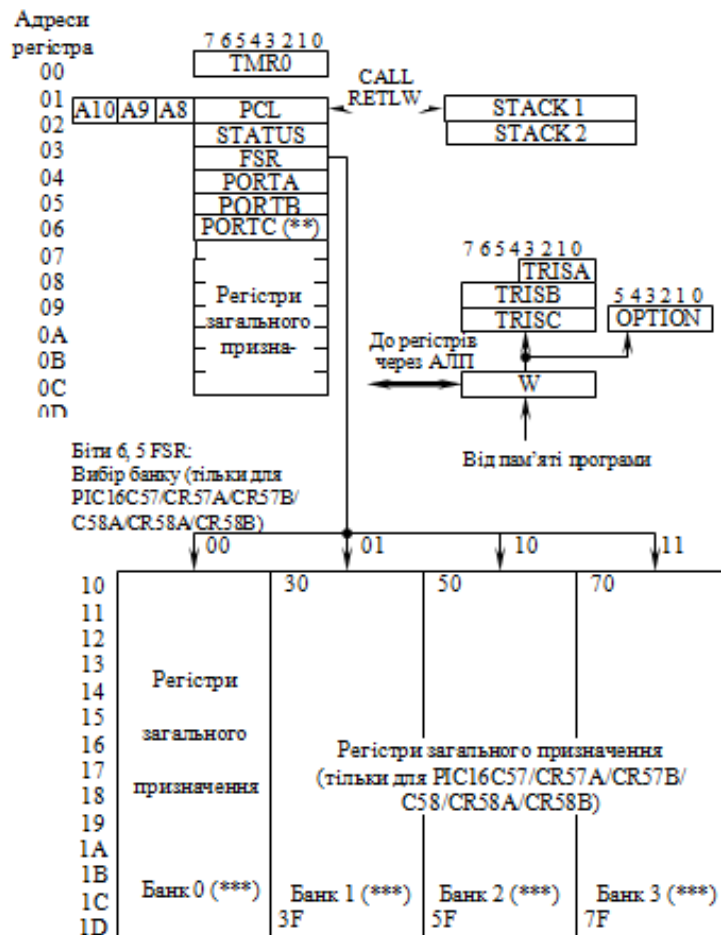


Рисунок 5.5 - Організація пам'яті даних

- (\*) – Регістр фізично не існує
- (\*\*) – Регістр 07h для PIC16C52/C54/C56 використовується як регістр загального призначення
- (\*\*\*) – Банк 0 доступний у всіх мікроконтролерах сімейства PIC16C5X. Банки 1, 2, 3 доступні тільки в PIC16C57/58.

Регістри можуть бути адресовані прямо або побічно, з використанням реєстра побічної (непрямої) адресації FSR. Безпосередня (пряма) адресація підтримується спеціальними командами, що завантажують дані з пам'яті програми в робочий регістр W. Регістри, як вже згадувалося, поділяються на дві функціональні групи: спеціальні реєстри та реєстри загального призначення.

Спеціальні реєстри (таблиця 5.3) мають у своєму складі реєстр таймера/лічильника реального часу (RTCC-TMR0), лічильник команд (PC), реєстр стану (STATUS), реєстри вводу-виводу (PORT) і реєстр побічної адресації (FSR).

Таблиця 5.3 - Опис спеціальних регістрів PIC16C5X

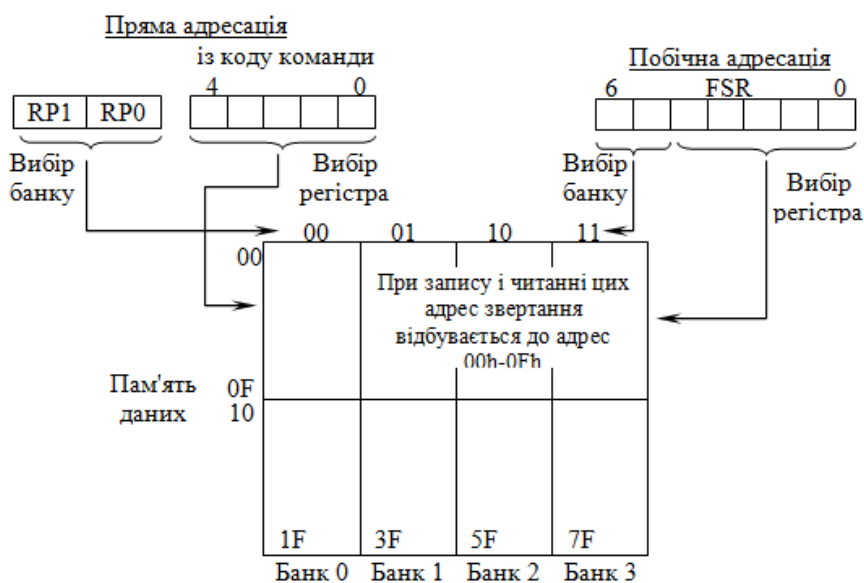
Адреса	Назва	Біт 7	Біт 6	Біт 5	Біт 4	Біт 3	Біт 2	Біт 1	Біт 0	Значення по включенню живлення	Значення по скиданню по MCLR і WDT
00h	INDF	Використовується значення FSR для доступу до пам'яті даних (не фізичний регістр)								-----	-----
01h	TMRO	8-розрядний лічильник/таймер								xxxx xxxx	uuuu uuuu
02h	PCL	Молодші 8 розрядів лічильника команд PC								1111 1111	1111 1111
03h	STATUS	PA2	PA1	PA0	TO	PD	Z	DC	C	0001 1xxx	000? ?uuu
04h	FSR	Регістр побічної адресації								xxxx xxxx	uuuu uuuu
05h	PORTA	-	-	-	-	PA3	RA2	RA1	RA0	---- xxxx	---- uuuu
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu

**Позначення:** x – не визначене, u – не змінюється, — – відсутня, читається як "0", ? – значення залежить від умов скидання.

**Примітки:**

1. Старші розряди лічильника команд безпосередньо не доступні. Звернення до старших бітів здійснюється через біти PA1, PA0 (STATUS<6:5>).
2. Для PIC16C52/54/C54A/CR54/CR54A/CR54B/C56/CR56/C58A/CR58A/CR58B регістр 07h є регістром загального призначення.
3. Заштриковані фізично відсутні біти.

При прямій адресації (рисунок 5.6) вибір банку здійснюється за допомогою 2-х бітів RP0 і RP1, що знаходяться в 5 і 6-му розрядах регістра побічної адресації (FSR/Pointer) відповідно. Вибір регістра в банку здійснюється за допомогою 5-розрядної адреси, що надходить в ОЗП прямо з коду команди по виділеній для цих цілей окремій шині адреси, що зв'язує регістр команд і пам'ять даних.



FSR – Регістр побічної адресації  
 RP1, RP0 – Біти 6 і 5 регістра FSR, відповідно

Рисунок 5.6 - Прямі і побічна адресація



При побічній адресації (рис. 5.6) використовується перемикач побічної адресації (регістр f0 у пам'яті даних, що фізично не існує) і покажчик Pointer (регістр f4 – FSR).

Існують деякі відмінності при здійсненні побічної адресації для різних типів МК. Так для PIC16C54/C55/C56: біти 0 – 4 регістра (f4) вибирають один з 32 регістрів в режимі побічної адресації, тобто коли у команді є звернення до регістра побічної адресації (f0). Біти 5 – 7 не використовуються і завжди читаються як одиниці. Якщо побічна адресація не використовується, регістр f4 може бути використаний як 5-бітовий регістр загального призначення. Для PIC16C57 перші 16 байтів кожного банку представляють собою ті ж самі фізичні регістри. Тільки тоді, коли біт 4 (f4) встановлений у одиницю (вказує на наступні 16 байтів), біти 5 і 6 вибирають один з чотирьох банків по 16 регістрів кожний. Біт 7 регістра (f4) не використовується і завжди читається як одиниця.

Регістр стану (f3) містить арифметичні прапорці АЛП, стан мікроконтролера при скиданні і біт вибору сторінок для програм з ємністю пам'яті, що перевищує 512 слів (PIC16C56/PIC16C57). Регістр STATUS доступний для будь-якої команди так само, як будь-який інший регістр. Проте біти TO і PD встановлюються апаратно і не можуть бути записані в регістр статусу програмно. Це слід мати на увазі при виконанні команди із використанням регістра статусу. Наприклад, команда CLRf3 обнулить всі біти, крім бітів TO і PD, а після цього встановить біт Z = 1. Після виконання цієї команди регістр статусу може і не мати нульового значення (через біти TO і PD) f3 = 000?? 100. Тому рекомендується для зміни регістра статусу використовувати тільки команди бітового встановлення BCF, BSF, MOVWF, що не змінюють інші біти статусу. Вплив всіх команд на біт статусу можна подивитися в описі команд.

Біти 5 – 6 регістра статусу визначаються як біти адреси сторінок PA0-PA1 програмної пам'яті (тільки для PIC16C56/PIC16C57 !!!). Коли виконуються команди GOTO, CALL і команди, коли програмний лічильник змінюється, наприклад MOVWF 2, біти адреси сторінок PA0-PA1 завантажуються в біти програмного лічильника A9-A10. Таким чином, пряма адреса, зазначена в слові команди, вказує на місце всередині певної сторінки пам'яті. Команда RETLW не змінює біти вибору сторінок. Після сигналу "Скид" біти регістра статусу PA0-PA1 обнуляються.

Розміщення прапорців в регістрі STATUS (Адреса: 03h. Значення по включенню живлення: 0001 1xxx) наступне:

b7	b6	b5	b4	b3	b2	b1	b0
PA2	PA1	PA0	TO	PD	Z	DC	C

C – Прапорць перенесення/позики.

Для команд ADDWF і SUBWF. Цей біт встановлюється, якщо в результаті операції з самого старшого розряду відбувається перенесення. Віднімання здійснюється шляхом додавання додаткового коду другого

операнда. При виконанні команд зсуву цей біт завжди завантажується з молодшого або старшого біта джерела, що зсувається.

```
;----- Приклад
; SUBWF Example #1
;
clrf      0x20      ; f(20h) = 0
movlw    1          ; wreg = 1
subwf    0x20      ; f(20h) = f(20h)-wreg = 0 = FFh
                ; Carry = 0: Результат від'ємний.

; SUBWF Example #2
;
movlw    0xFF
movwf    0x20      ; f(20h) = FFh
clrw
subwf    0x20      ; f(20h) = f(20h)-wreg = FFh-0 = FFh
                ; Carry = 1: Результат додатний.
```

DC – Прапорець десяткового перенесення/позики.

Для команд ADDWF і SUBWF. Цей біт встановлюється, якщо в результаті операції з четвертого розряду відбувається перенесення. Механізм встановлення десяткового біта перенесення "DC" такий же, відрізняється лише тим, що відстежується перенесення з четвертого біта.

Z – Прапорець нульового результату:

Встановлюється, якщо результатом арифметичної або логічної операції є нуль.

PD – Power Down (режим зберігання даних):

Встановлюється в "1" при включенні живлення або за командою CLRWDT. Скидається в "0" командою SLEEP.

TO – Time Out. Прапорець спрацьовування сторожового (Watchdog) таймера:

Встановлюється в "1" при включенні живлення і командами CLRWDT, SLEEP. Скидається в "0" по закінченню затримки часу таймера WDT.

PA0, PA1 – Біти вибору сторінки пам'яті програм.

Для кристалів PIC16C54/C55 – це біти загального призначення.

Для PIC16C56:

PA0 – Біт вибору сторінки  
0 = Сторінка 0 (000—FF),  
1 = Сторінка 1 (200—FF).

PA1 – Біт загального призначення.

Для PIC16C57:

PA1, PA0 – обидва біти вибору сторінки:

- 00 = Сторінка 0 (000—FF),
- 01 = Сторінка 1 (200—FF),
- 10 = Сторінка 2 (400—FF),
- 11 = Сторінка 3 (600—FF).

PA2 – Біт загального призначення (зарезервовано для майбутніх розробок).

Регістр OPTION - це регістр конфігурації попереднього дільника і RTCC (TMRO). Значеннями бітів в цьому регістрі (він не має адреси!) визначається підключення попереднього дільника до RTCC або WDT, коефіцієнт його ділення, джерело лічильних імпульсів, вибирається фронт сигналу для RTCC. Регістр OPTION призначений тільки для запису і має ширину 6 бітів. Під час виконання команди "OPTION" вміст регістра W завантажується в цей регістр. За сигналом "Скид" всі біти цього регістра встановлюються в одиниці.

Регістр OPTION (Адреса: немає. Значення по ввімкненню живлення: 111111) має наступний зміст:

b5	b4	3	2	1	0
RTS (TOSC)	RTE (TOSE)	SA	S2	S1	S0

PS2 ... PS0 – Настроювання попереднього дільника (див. табл. 5.4).

PSA – Біт, який вказує, що дільник працює з RTCC (TMRO) або WDT:  
0 – RTCC, 1 – WDT.

RTE (T0SE) – Фронт сигналу RTCC:

- 0 – інкремент по позитивному фронту на ніжці RTCC,
- 1 – інкремент по негативному фронту на ніжці RTCC.

RTS (T0CS) – Джерело сигналу для RTCC:

- 0 – сигнал від внутрішнього генератора,
- 1 – зовнішній сигнал на ніжці RTCC.

Таблиця 5.4 - Настроювання попереднього дільника

PS2...PS0	Коефіцієнти ділення	
	RTCC (TMRO)	WDT
0 0 0	1 : 2	1 : 2
0 0 1	1 : 4	1 : 4
0 1 0	1 : 8	1 : 8
0 1 1	1 : 16	1 : 16
1 0 0	1 : 32	1 : 32
1 0 1	1 : 64	1 : 64
1 1 0	1 : 128	1 : 128
1 1 1	1 : 256	1 : 256

Програмний лічильник забезпечує доступ до 12-бітних комірок вбудованої постійної пам'яті (EPROM), довжина якої може бути до 2048 комірок.

В залежності від типу PIC, програмний лічильник (PC) і відповідний дворівневий апаратний стек мають ширину від 9 до 11 бітів (див. табл. 5.5).

Таблиця 5.5 - Ширина програмного лічильника і стека

Тип МК	Ширина PC (біт)	Ширина стека (біт)
PIC16C54/PIC16C55	9	9
PIC16C56	10	10
PIC16C57	11	11

При скиді, всі біти програмного лічильника встановлюються в одиниці. У процесі виконання програми PC автоматично інкрементується при виконанні кожної команди, якщо команда сама не змінить його у наступних випадках:

1) при виконанні команди GOTO в програмний лічильник завантажуються молодші 9 бітів (PC<8:0>). У випадку використання PIC16C56 або PIC16C57, в старші два біти програмного лічильника (PC<10:9>) завантажуються біти вибору сторінок PA1 і PA0 (біти 6 і 5 регістра статусу). За допомогою команди GOTO можна переміститись в будь-яке місце будь-якої сторінки пам'яті програм;

2) при виконанні команди CALL у програмний лічильник завантажуються молодші 8 бітів, в той час як дев'ятий біт обнуляється. Значення PC, збільшене на одиницю, буде збережене в стеці. У випадку PIC16C56/PIC16C57, в старші два біти PC (PC <10:9>) завантажуються біти вибору сторінок PA1, PA0 (біти 6 і 5 регістра статусу);

3) команда RETLW вивантажує в програмний лічильник вміст з вершини стека;

4) якщо програмний лічильник задіяний в команді (наприклад, MOVWF 2, ADDWF 2, BSF 2,5), то 8 бітів результату виконання команди будуть завантажені в молодші 8 бітів програмного лічильника. Дев'ятий біт PC буде обнулятися. У випадку PIC16C56/PIC16C57, в старші біти PC<10,9> будуть завантажені біти вибору сторінок PA1, PA0 (біти 6 і 5 в регістрі статусу).

Так як дев'ятий біт програмного лічильника обнуляється при виконанні команди CALL, або команд, що записують свій результат в PC, то виклик підпрограм або здійснення переходу, що вираховується, обмежені першою половиною сторінки програмної пам'яті, тобто першими 256 адресами.

В мікроконтролерах PIC16C5X використовується дворівневий апаратний стек. Це означає, що він може зберігати водночас не більше двох адрес повернення.

Команда CALL інкрементує поточне значення програмного лічильника і записує його в стек на рівень 1. При запису на 1-й рівень стека його попереднє значення автоматично записується на 2-й рівень. Якщо викликається більше

двох вкладених підпрограм, то тільки дві останні адреси повернення будуть збережені.

При виконанні команди RETLW в програмний лічильник вивантажується вміст стека першого рівня. Водночас вміст стека другого рівня копіюється в стек першого рівня. Якщо виконуються більше двох команд RETLW одна за одною, в стеку обох рівнів буде записане значення другого рівня.

Робочий регістр W містить другий операнд в двооперандних командах і підтримує внутрішню передачу даних.

Регістри вводу-виводу можуть керуватися, як будь-які інші регістри. Проте, команда "читання" (наприклад MOVF 6, W) завжди зчитує фактичний рівень сигналу на ніжці порту, незалежно від того, запрограмований цей розряд порту на ввід чи на вивід. Після сигналу "Скид" всі порти вводу-виводу встановлюються на "ввід" (електрично еквівалентно третьому стану), а керуючі регістри TRISA, TRISB та TRISC встановлюються в одиниці (конфігурація на ввід).

Для того, щоб сконфігурувати деякі лінії порту на вивід, необхідно встановити відповідні біти у потрібному TRIS регістрі в "0". Це можна робити командою "TRIS f".

Порт A є 4-бітним портом вводу-виводу. Використовуються тільки молодші 4 біти (RA0 – RA3). Старші 4 біти – завжди читаються як нулі.

Порт B є 8-бітним портом вводу-виводу.

Порт C для PIC 16C55/C57: 8-бітний порт вводу-виводу. Для PIC16C54/C56: внутрішній регістр загального призначення пам'яті даних (не порт).

Схема одного розряду (лінії) порту вводу-виводу наведена на рис. 2.7.

При операціях вводу порти не защепуються. Вхідний сигнал повинен бути присутній поки іде процес читання (наприклад, MOVF 6, W). При операціях виводу порти защіпаються і зберігають значення до тих пір, поки не будуть перезаписані. На рис. 2.7 не показані діоди, що захищають ніжку порту від зовнішніх імпульсів великої напруги. Вони обмежують імпульсну напругу на ніжці значеннями від V<sub>SS</sub> (0,6 В) до V<sub>DD</sub> (+ 6 В).

Якщо статична напруга за якихось причин вийде за означені межі, то виникнуть великі статичні струми, які здатні вивести мікроконтролер з ладу.

Проблеми при організації двонаправлених портів. Деякі команди виконуються в режимі читання-модифікація-запис. Наприклад, команди BCF і BSF повністю читають порт, модифікують один біт і виводять результат назад. Тут необхідна обережність. Наприклад, команда BSF для біта 5 регістра f6 (порт B) спочатку зчитує всі 8 бітів. Після цього виконуються дії над бітом 5 і нове значення байта повністю записується у вихідні защіпки. Якщо другий біт регістра f6 використовується в якості двонаправленого вводу-виводу (скажімо біт 0) і в даний момент він визначений як вхідний, вхідний сигнал на цій ніжці буде зчитаний і записаний назад у вихідну защіпку цієї ж ніжки, стираючи її попередній стан. До тих пір, поки ця ніжка залишається в режимі вводу, жодних проблем не виникає. Проте, якщо лінія 0 переключиться пізніше в режим виводу, її стан буде невизначеним.

На вивід, що працює в режимі виходу, не повинні підключатися зовнішні навантаження за схемою "монтажне І" чи "монтажне АБО". Великі струми, що виникають при цьому, можуть пошкодити кристал.

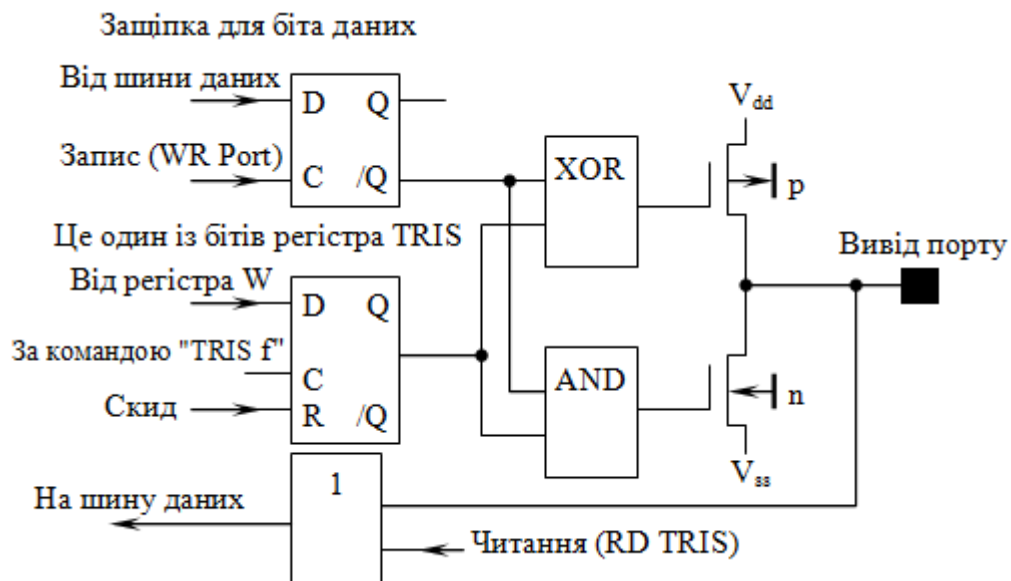


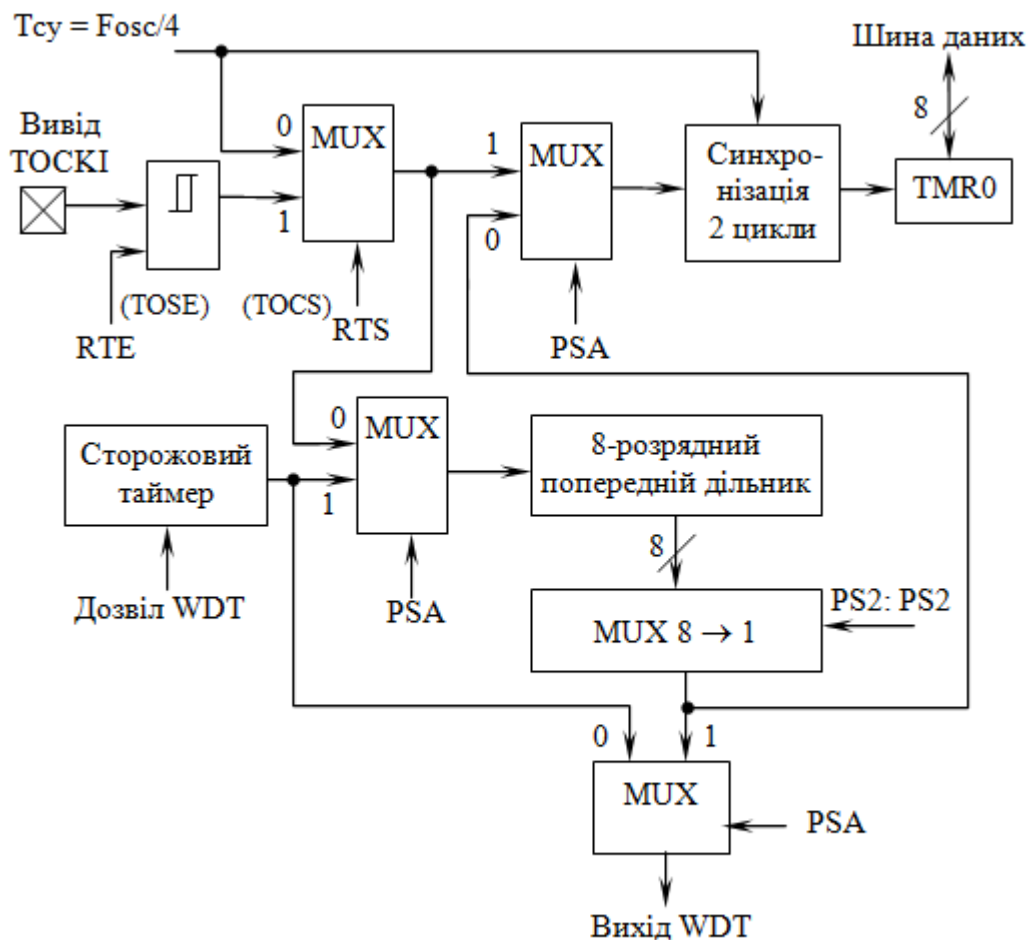
Рисунок 5.7 - Схема одного розряду порту вводу-виводу PIC 16C5X

Послідовне звертання до портів вводу-виводу. Запис в порт вводу відбувається наприкінці командного циклу. Але при читанні, дані повинні бути стабільні на початку командного циклу. Будьте уважні в операціях читання, що відбуваються відразу за записом в той же порт. Тут потрібно враховувати інерційність встановлення напруги на виводах. Може знадобитись програмна затримка, щоб напруга на ніжці (залежить від навантаження) встигла стабілізуватися до початку виконання наступної команди читання.

На рис. 5.8 наведена спрощена структурна схема модуля таймера. Він може працювати в одному з 2-х режимів: таймера або лічильника.

Режим таймера вибирається встановленням в "0" біта TOCS (RTS) в 5-му розряді регістра OPTION. В режимі таймера RTCC (TMRO) збільшується в кожному командному циклі (при відсутності попереднього дільника). Якщо відбувається запис в RTCC, то збільшення лічильника затримується на два наступних цикли виконання команди.

Запис в RTCC повинен вестися з урахуванням цієї затримки. При необхідності перевірки регістра TMRO на нуль без впливу на процес лічення рекомендується користуватися командою MOVF TMRO, W.



TOCS, TOSE, PSA, PS2: PS0 – Біти регістра OPTION <6:0>.

Рисунок 5.8 - Структурна схема модуля таймера

Режим лічильника вибирається встановленням в "1" біта TOCS (OPTION <5>). В цьому режимі вміст таймера збільшується по кожному перепаду 1/0 або 0/1 на виводі ТОСКІ. Перепад, що збільшує значення TMR0, вибирається бітом вибору фронту переключення TOSE (RTE) в регістрі OPTION (<4>). Встановлення цього біта в "0" викликає збільшення RTCC по перепаду 0/1.

В регістр RTCC можна завантажити дані або зчитати з нього, як з будь-якого іншого регістра. Вміст цього регістра може бути інкрементовано фронтом зовнішнього сигналу, що надходить на вхід RTCC кристала, або внутрішнім сигналом синхронізації (CLKOUT = Fosc/4).

Структурна схема містить елемент MUX – це електронний перемикач (мультиплексор).

Для того, щоб 8-бітний попередній дільник приєднати до таймера/лічильника, потрібно встановити певним чином біт PSA і біти PS в регістрі Option. Регістр Option - це спеціальний регістр (у нього немає адреси в пам'яті даних), але до нього можна звернутися за допомогою команди OPTION. Якщо попередній дільник підключений до таймера/лічильника (RTCC), то будь-які команди запису в самий RTCC (регістр f1, наприклад ClrF1 BSF1,5 ... і т.п.) очищають його.

Біт RTS (RTCC signal Source) в регістрі Option визначає джерело лічильних імпульсів (внутрішніх чи зовнішніх).

Коли RTS = 1:

Сигналом синхронізації для таймера/лічильника RTCC чи для попереднього дільника, якщо він підключений до RTCC, є сигнал, що надходить на вивід RTCC. Значення біта 4 (RTE) в регістрі Option визначає, чи відбувається лічення по задньому фронту (RTE = 1) або по передньому фронту (RTE = 0) сигналу на виводі RTCC.

Коли RTS = 0:

Регістр RTCC інкрементується за сигналом внутрішньої синхронізації CLKOUT ( $T_{cy} = F_{osc}/4$ ). В цьому випадку значення біта RTE в регістрі Option і сигналу на виводі RTCC не мають значення. Проте вивід RTCC слід підключити до Vdd або до Vss, як зручніше, щоб випадково не потрапити в режим тесту і забезпечити коректність в режимі пониженого споживання.

Доки на таймер/лічильник RTCC надходять синхроімпульси (не важливо зовнішні або внутрішні, підключений попередній дільник до RTCC чи ні), регістр f1 – RTCC інкрементується і при досягненні значення "FFH" скидається в 0. Після обнулення лічення продовжується.

Лічильні імпульси затримуються на два командних цикли. Наприклад, після запису інформації в RTCC, інкрементація його відбудеться через два командних цикли. Таке відбувається з усіма командами, що здійснюють запис або читання-модифікацію-запис f1 (наприклад, MOVF f1, CLRF f1). Якщо RTCC потрібно перевірити на рівність нулю без зупинки лічення, слід використати інструкцію MOVF f1,W.

Попередній дільник представляє собою 8-бітний лічильник, який використовується як дільник частоти перед RTCC або після Watchdog таймера. Якщо дільник з'єднаний з RTCC, то він не може бути підключений до Watchdog таймера і навпаки. Біти PSA і PS0 – PS2 в регістрі OPTION визначають місце підключення дільника і встановлюють його коефіцієнт ділення.

Попередній дільник може підключатись по-різному. Коли він приєднаний до RTCC, всі команди, що здійснюють запис в RTCC (наприклад, CLRF 1, MOVWF 1, BSF 1, x... і т.д.) будуть водночас обнуляти і дільник. Коли дільник підключений до WDT, команда CLRWDT також обнулить дільник разом з Watchdog таймером. Підключення дільника – програмно керуєме. Щоб уникнути несанкціонованого сигналу "Скид" під час переключення дільника з RTCC на WDT, слід виконати певну послідовність команд:

1. `MOVLW B`xx0x0xxx`` ; Вибрати внутрішню синхронізацію  
; і нове значення для дільника. Якщо  
; нове значення дільника рівне "000"  
; або "001", то треба тимчасово вибрати  
; інше значення дільника.
2. OPTION
3. `CLRF 1` ; Обнулити RTCC і дільник.
4. `MOVLW B`xxxx1xxx`` ; Вибрати WDT, не змінюючи



; значення дільника.

5. OPTION

6. CLRWD ; Обнулити WDT і дільник (Скид).

7. MOVLW B`xxxx1xxx` ; Вибрати нове значення для  
; дільника.

8. OPTION

Пункти 1 і 2 вимагаються тільки тоді, коли до RTCC було підключено зовнішнє джерело. Пункти 7 і 8 необхідні тоді, коли в дільник вимагається завантажити величину 000 або 001.

Переключення дільника з WDT на RTCC необхідно здійснювати за допомогою наступної послідовності команд:

1. CLRWDT ; Обнулити WDT і дільник.

2. MOVLW B`xxxx0xxx` ; Вибрати RTCC, нове значення для  
; дільника і джерело сигналу.

3. OPTION

Ця послідовність повинна бути виконана навіть у тому випадку, якщо сторожовий таймер WDT заборонений.

Сімейство мікроконтролерів PIC16C5X має набір спеціальних функцій, призначених для розширення можливостей системи, мінімізації вартості, виключення навісних компонентів, забезпечення мінімального енергоспоживання і захисту коду від зчитування. В PIC16C5X реалізовані наступні спеціальні функції:

- вибір типу генератора;
- скид;
- схема скидання по включенню живлення (POR);
- таймер скидання (DRT);
- сторожовий таймер (WDT), крім PIC16C52;
- режим пониженого енергоспоживання (SLEEP);
- захист коду від зчитування;
- біти ідентифікації.

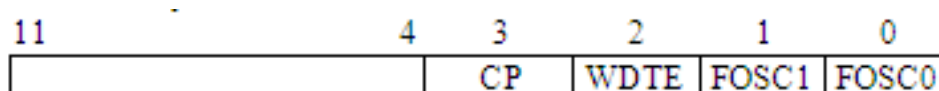
Мікроконтролери сімейства PIC16C5X, крім PIC16C52, мають вбудований сторожовий таймер WDT, що може бути вимкнений тільки через біт конфігурації WDTE. Для підвищення надійності він працює від власного RC-генератора. Таймер скидання DRT призначений для підтримки контролера в скинутому стані протягом 18 мс після включення живлення для стабілізації роботи генератора. Присутність цих таймерів дозволяє в багатьох застосуваннях відмовитися від схеми зовнішнього скидання.

Режим пониженого енергоспоживання SLEEP призначений для забезпечення дуже малого струму споживання в очікуванні (менш 1 мкА при вимкненому сторожовому таймері). Вихід з режиму SLEEP можливий за зовнішнім сигналом скиду або по закінченню витримки сторожового таймера.

Можливість вибору типу генератора дозволяє ефективно використовувати мікроконтролери сімейства в різних прикладеннях. Використання RC-генератора дозволяє зменшити вартість системи, а LP генератор зменшує енергоспоживання.

Мікроконтролери сімейства PIC16C5X мають чотири біти конфігурації, що встановлюються на етапі програмування кристала. Біти FOSC1 і FOSCO визначають тип генератора, біт WDTE дозволяє використання сторожового таймера WDT, а біт CP призначений для захисту програми від зчитування.

Слово конфігурації (CONFIG) розташовується за адресою 0FFh в пам'яті програм і має наступний зміст:



CP – захист від зчитування програми (Code Protect):

- 1 – захист включено,
- 0 – захист виключено.

WDTE – дозвіл сторожового таймера:

- 1 – сторожовий таймер дозволений,
- 0 – сторожовий таймер заборонений.

FOSC1, FOSCO – вибір типу генератора.

- FOSC1, FOSCO = 11: RC-генератор,
- 10:HS-генератор,
- 01:XT-генератор,
- 00:LP-генератор.

В мікроконтролерах сімейства PIC16C5X передбачено декілька типів генераторів. Для PIC16C54A/58A користувач може запрограмувати два конфігураційних біти (FOSC1 і FOSCO) для вибору одного із чотирьох режимів: RC, LP, XT, HS. Для PIC16C52 допустимі два режими: RC, XT. Для інших мікроконтролерів сімейства PIC16C5X тип генератора програмується на заводі-виготівнику. Для OTP мікроконтролерів PIC16C54/C55/C56 /C57 тип генератора вказується в маркуванні: XT – стандартний кварцовий генератор, HS – високочастотний кварцовий генератор, LP – низькочастотний генератор для економічних прикладень з керамічним резонатором.

Кварцовий генератор. В режимах XT, LP і HS до виводів OSC1/CLKIN і OSC2/CLKOUT підключається кварцовий або керамічний резонатор (рисунок 5.9). Схема генератора PIC 16C5X передбачає використання резонаторів з паралельним резонансом. Використання резонаторів з послідовним резонансом може призвести до збудження резонатора на частоті, що виходить за межі параметрів резонатора. Для резонаторів з АТ зрізом може використовуватися послідовний резистор Rs. В режимах XT, LP і HS генератор

PIC16C5X може також тактуватися від зовнішнього джерела, що підключається до виводу OSC1/CLKIN (рисунок 5.10).

RC-генератор. Коли не пред'являються високі вимоги до точності та стабільності частоти генератора, використання RC-генератора дозволяє додатково зменшити вартість системи.

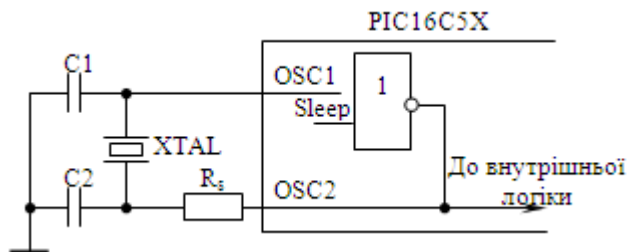


Рисунок 5.9 - Схема підключення кварцового і керамічного резонатора

Частота RC-генератора залежить від напруги живлення, значень резистора  $R_{ext}$ , конденсатора  $C_{ext}$ , робочої температури і незначно змінюється від розкидання характеристик кристалів. На частоту генерації при малих значеннях  $C_{ext}$  також впливає власна ємність корпусу кристала. Крім того, потрібно враховувати також температурний дрейф резистора  $R_{ext}$  і конденсатора  $C_{ext}$ . На рис. 2.11 наведена схема включення RC-генератора.

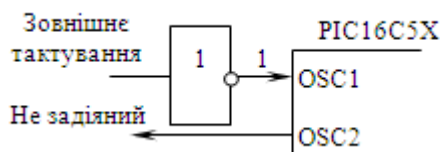


Рисунок 5.10 - Схема підключення зовнішнього генератора

Для значень  $R_{ext}$ , нижче 2.2 кОм, генератор може працювати нестабільно або не запускатися. При дуже великих значеннях  $R_{ext}$  (наприклад, 1 МОм), генератор стає чутливим до завад, витоків і вологості. Рекомендований діапазон значень  $R_{ext}$  від 3 до 100 кОм. Хоча генератор дієздатний і при відсутності зовнішнього конденсатора ( $C_{ext} = 0$ ), для збільшення стабільності роботи рекомендується використовувати конденсатор ємністю понад 20 пф. При малій ємності  $C_{ext}$ , чи взагалі без цього конденсатора, частота генератора сильно залежить від монтажних ємностей. Розкид буде тим більший, чим більше величина резистора  $R_{ext}$  (бо вплив струмів витоків на частоту RC генератора сильніший при більших значеннях  $R_{ext}$ ), і чим менша величина ємності  $C_{ext}$  (бо в цьому випадку сильніше проявляється вплив монтажних ємностей).

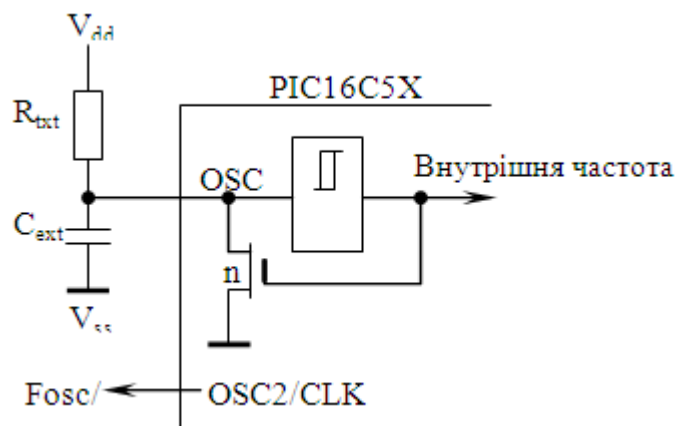


Рисунок 5.11 - Схема включення RC-генератора

На виводі OSC2/CLKOUT в режимі RC присутній сигнал з частотою генератора, поділеною на чотири ( $F_{osc}/4$ ), котрий може бути використаний для синхронізації інших схем.

Мікроконтролери PIC16C5X мають наступні способи скидання:

- скидання по включенню живлення (POR);
- скидання по входу MCLR при звичайній роботі;
- скидання по входу MCLR в режимі пониженого енергоспоживання SLEEP;
- скидання по сторожовому таймеру WDT.

На рис. 5.12 наведена структурна схема вузла скидання.

При скиданні деякі регістри не змінюють свій стан. При включенні живлення вони мають невизначене значення, в інших випадках їхній стан не змінюється (див. табл. 2.3 і більш докладно в [3, 5]). Більшість інших регістрів встановлюються в певний стан у випадках скидання при включенні живлення, за входом MCLR або за сторожовим таймером. PIC16C5X не розрізняє скидання по сторожовому таймеру при нормальній роботі та режимі SLEEP. Біти TO і PD (STATUS <4:3>) встановлюються в залежності від причини скидання.

Під час дії сигналу «Скидання» стан кристала наступний:

- Генератор працює або готовий до запуску (включення або вихід із SLEEP).
- Всі виводи портів вводу-виводу кристала встановлюються в третій стан, шляхом установки регістрів "TRIS" в "одиницю" (що відповідає режиму вводу).
- Всі біти програмного лічильника встановлюються в "одиниці" (у випадку PIC16C54/55 програмний лічильник дорівнює 1Fh, для PIC16C56 програмний лічильник дорівнює 3Fh, PIC16C57 – програмний лічильник дорівнює 7Fh).
- Біти регістра OPTION встановлюються в "одиниці".
- Watchdog таймер і його дільника обнуляються.
- Старші три біти статус-регістра (біти вибору сторінки) обнуляються.
- Сигнал CLKOUT на виводі OSC2 утримується в "0" (тільки для RC-генераторів).

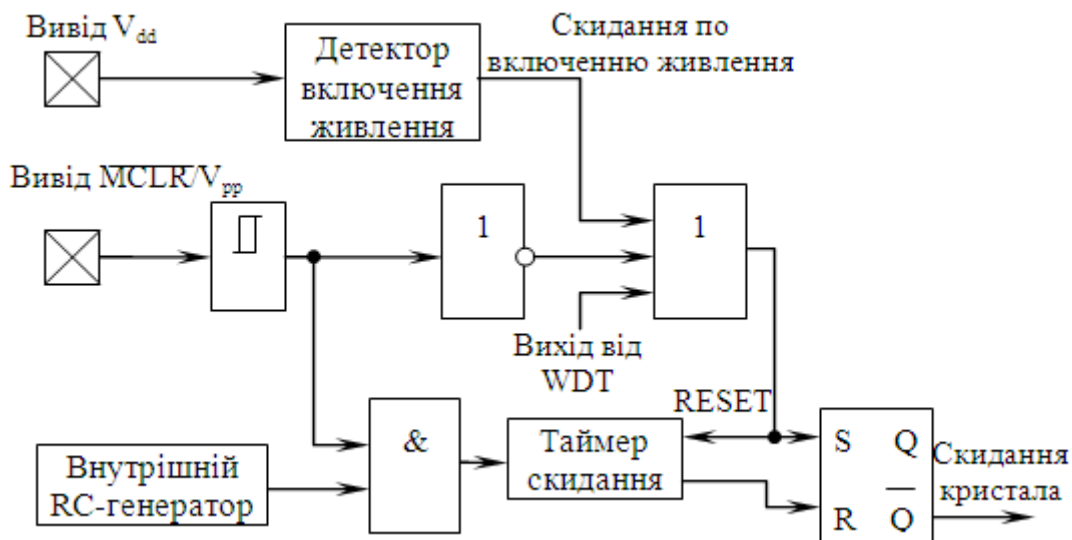


Рисунок 5.12 - Структурна схема вузла скидання

Таймер запуску. Генератор, побудований на кварцових або керамічних резонаторах, вимагає обов'язкової затримки після включення живлення для забезпечення стабільної генерації. Для цього вбудований таймер запуску генератора тримає пристрій в стані скидання приблизно 18 мс після того, як сигнал на /MCLR ніжки кристала досягне рівня логічної одиниці. Таким чином, зовнішній ланцюжок RC, зв'язаний з нішкою /MCLR, в багатьох випадках не вимагається.

Watchdog Timer також використовує таймер запуску. Це важливо для випадків, де WDT використовується для автоматичного виведення PIC16C5X із режиму SLEEP. В кристалах з низькою частотою, де вимагається понад

18 мс для забезпечення стабільної генерації, використання вбудованого таймера запуску недостатньо.

Зовнішнє скидання при включенні живлення. Кристали PIC16C5X мають вбудовану схему скидання при включенні живлення. Щоб використати цю схему, треба просто підключити вивід /MCLR до напруги живлення Vdd. Таймер запуску починає відлік затримки часу тільки якщо /MCLR досяг високого рівня. По закінченні цієї затримки (в середньому 18 мс) процес внутрішнього скидання завершується.

Тут існує проблема, коли Vdd зростає надто повільно і затримка на запуск генератора завершилася, а живлення ще не досягло свого мінімального значення Vdd (min) дієздатності. Вбудована схема скидання забезпечує роботу, якщо швидкість зростання Vdd при включенні не нижче 0.05 В/мс. Необхідно також, щоб початковий рівень живлення дорівнював 0 В. Вбудована схема скидання також не буде працювати з низькочастотними кристалами, що вимагають затримку на запуск, значно більшу ніж 18 мс. У таких випадках рекомендується використати зовнішні RC ланцюжки для скидання по /MCLR. Нижче на рис. 5.13 наведено такий ланцюжок.

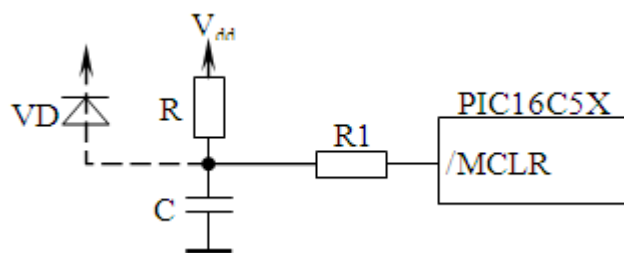


Рисунок 5.13 - Схема зовнішнього скидання

Тут можна застосувати діод для швидкого розряду конденсатора при виключенні живлення. На резисторі  $R < 40 \text{ кОм}$  не повинно падати більше 0.2 В. Резистор  $100 \text{ Ом} < R1 < 1 \text{ кОм}$  обмежить струм в ніжку /MCLR.

Сторожовий таймер представляє собою повністю готовий вбудований RC-генератор. Він буде працювати, навіть якщо основний генератор зупинений, як це буває при виконанні команди SLEEP. Таймер виробляє сигнал скидання. Вироблення таких скидань може бути заборонене шляхом запису нуля в спеціальний біт, що розміщений поза програмною пам'яттю EPROM. Цю операцію виконують на етапі пропалювання мікросхем.

Затримка часу WDT. Номінальна затримка WDT становить 18 мс (без використання дільника). Вона залежить від температури, напруги живлення, від особливостей типу мікросхеми. Якщо вимагаються великі затримки, то до WDT може бути підключений дільник з коефіцієнтом ділення до 1:128, що програмується шляхом запису в регістр OPTION. Тут можуть бути реалізовані затримки до 2.5 секунд.

Команди CLRWDT і SLEEP обнуляють WDT і дільник, якщо він підключений до WDT. Це запускає затримку часу спочатку і запобігає на деякий час вироблення сигналу скидання. Якщо сигнал скидання від WDT все ж відбувся, то водночас обнуляється біт TO в регістрі статусу (f3).

Підключення попереднього дільника до виходу WDT показано вище на рис. 5.8.

Режим пониженого енергоспоживання SLEEP призначений для забезпечення дуже малого струму споживання в очікуванні (менше 1 мкА при виключеному сторожовому таймері).

Вхід в режим SLEEP здійснюється командою SLEEP. За цією командою, якщо WDT дозволений, то він скидається і починає відлік часу, біт "TD" регістра статусу (f3) скидається, біт "TO" встановлюється, а вбудований генератор вимикається. Порти вводу-виводу зберігають стан, який вони мали до входу в режим SLEEP.

Для пониження струму, що споживається в цьому режимі, лінії на вивід повинні мати такі значення, щоб не протікав струм поміж кристалом і зовнішніми ланцюгами. Лінії на ввід повинні бути з'єднані зовнішніми резисторами з високим або низьким рівнем напруги живлення (V<sub>dd</sub> або V<sub>ss</sub>), щоб уникнути струмів переключення, що викликаються плаваючими високоомними входами.

Вихід із режиму SLEEP здійснює WDT (якщо він дозволений) або зовнішній нульовий імпульс на ніжці /MCLR – скидання. В обох випадках PIC16C5X буде знаходитись в режимі скидання протягом часу запуску генератора, а потім тільки почнеться виконання програми.

Біт "PD" в регістрі статусу (f3), котрий встановлюється при включенні, але обнуляється командою SLEEP, може бути використаний для визначення стану процесора до "прокидання": або процесор був в режимі SLEEP (гарячий старт), або було просто включено живлення (холодний старт).

Біт TO дозволяє визначити, чим був викликаний вихід із режиму SLEEP: або зовнішнім сигналом на ніжці /MCLR, або спрацьовуванням WDT.

Програма, записана в ППЗП або в ПЗП, може бути захищена від зчитування за допомогою установки в нуль біта захисту CP в слові конфігурації. В режимі захисту програми вміст пам'яті програми не може бути прочитаний у початковому вигляді, завдяки цьому неможливо реконструювати записану програму. Крім того, при встановленому біті захисту неможливо допрограмувати контролер. Регістр конфігурації CONFIG може бути зчитаний і запрограмований незалежно від стану біта захисту. Не рекомендується програмувати біт захисту в мікроконтролерах з ультрафіолетовим затиранням.

Мікроконтролери PIC16C5X мають чотири спеціальні адреси, які не являються частиною пам'яті програми. Вони призначені для зберігання ідентифікаційного коду (ID) користувача, контрольної суми або іншої інформації. Як і слово конфігурації, вони можуть бути прочитані або записані тільки за допомогою програматора. Програмно ці адреси недоступні.

Для забезпечення можливості читання однакової інформації як в звичайному режимі, так і у режимі із встановленим бітом захисту від зчитування, рекомендується використовувати тільки чотири молодших біти за кожною адресою для зберігання коду ID. Старші 8 бітів рекомендується встановлювати в "1".

Кожна команда PIC16C5X – це 12-бітне слово, що розділене по сенсу на наступні частини: код операції (OPCODE), поле для одного або більше операндів, що приймають або не приймають участь в цій команді. Формат команди наведений на рис. 5.14.

Для байт-орієнтованих команд "f" позначає собою регістр, з яким проводяться дії. Тобто "f" позначає один з 32-х регістрів ОЗП мікроконтролера, який буде використовувати команда з урахуванням поточного номера банку. Біт "d" визначає, куди покласти результат. Якщо  $d = 0$ , результат буде знаходитись в W регістрі. Якщо  $d = 1$ , результат буде знаходитись в регістрі "f", що згадується в команді.

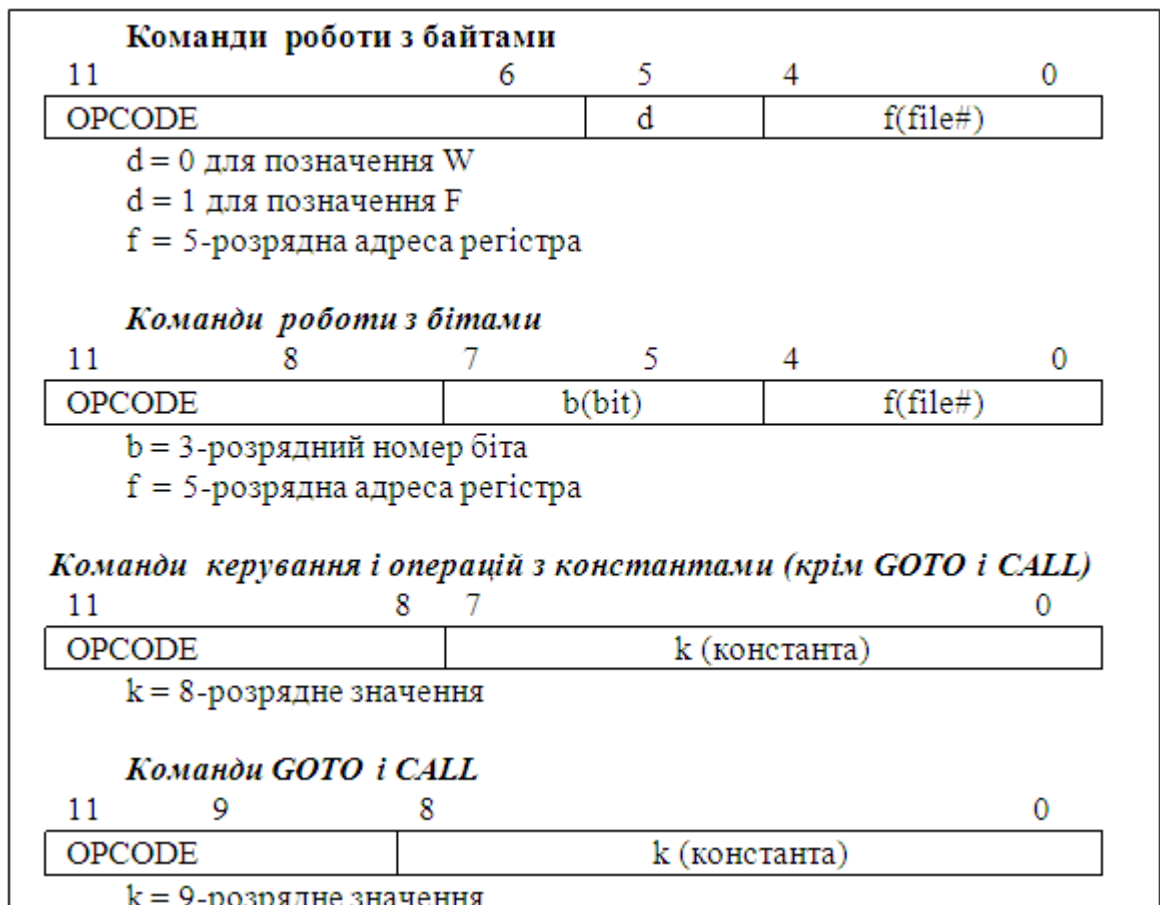


Рисунок 5.14 - Загальний формат команди

Для біт-орієнтованих команд "b" позначає номер біта, який бере участь в команді, а "f" – це регістр поточного банку, в якому цей біт розміщений.

Для команд передачі керування і операцій з константами "k" позначає восьми або дев'ятибітну константу.

Всі команди виконуються протягом одного командного циклу. В двох випадках виконання команди займає два командних цикли:

1. Виконання умовної команди.
2. Перевірка умови і перехід.

Один командний цикл складається з чотирьох періодів генератора. Таким чином, для генератора з частотою 4 Гц час виконання командного циклу буде 1 мкс.



## СПИСОК ЛІТЕРАТУРИ

1. Справочник по среднему семейству микроконтроллеров PICmicroTM. Техническая документация DS33023A Microchip Technology Incorporated, USA. [www.microchip.com](http://www.microchip.com).
2. Смит Б.Э., Джонсон М.Т. Архитектура и программирование микропроцессора Intel 80386. - М.: Конкорд, 2002. - 334 с.
3. В.В.Корнеев, А.В.Киселев. Современные микропроцессоры. – М.: НОЛИДЖ, 2008. – 240 с.
4. П. Хоровиц, У. Хилл. Искусство схемотехники. В 3-х томах. Т.3. Глава 11. – М.: Мир, 2003.
5. Цифровые процессоры обработки сигналов. Справочник под редакцией А.Г.Остапенко. - М.: Радио и связь, 2004. – 264 с.
6. Вашкевич Н.П. Синтез микропрограммных управляющих автоматов. Учеб. пособие. - Пенза: Пенз. политехн. ин-т, 1990. - 115 с.
7. Сергеев Н.П., Вашкевич Н.П. Основы вычислительной техники: Учеб. пособие. 2-е изд. перераб. и доп. - М.: Высш. шк., 2007. - 311 с.
8. Баранов С.И. Синтез микропрограммных автоматов. - 2-е изд. перераб. и доп. - Л.: Энергия, Ленингр. отд-ние, 2001. - 232 с.
9. Майоров С.А., Новиков Г.И. Структура электронных вычислительных машин. - Л.: Машиностроение. Ленингр. отд-ние, 1989. - 384 с.

Навчально-методичне видання

# Мікропрограмування

Навчальний посібник

**Відповідальний за випуск:**

*Дивак М. П., д. т. н., професор, декан факультету комп'ютерних  
інформаційних технологій*

Підписано до друку 12.06.2011 р.  
Формат 60x84/16. Папір офсетний.  
Друк офсетний. Зам. № 2-1390  
Умов.-друк. арк. 8,71.  
Тираж 100 прим.

Видавництво Тернопільського національного  
економічного університету "Економічна думка"  
46000, Тернопіль, вул. Львівська, 3, тел. 43-22-18  
E-mail: edition@tane.edu.ua

*Свідоцтво про внесення суб'єкта видавничої справи  
до державного реєстру видавців ДК № 3467 від 23.04.2009 р.*

Віддруковано ФО-П Шпак В. Б.  
Свідоцтво про державну реєстрацію В02 № 924434 від 11.12.2006 р.  
Свідоцтво платника податку: Серія Е № 897220  
м. Тернопіль, вул. Просвіти, 6.  
тел. 8 097 299 38 99  
E-mail: tooums@ukr.net