

**Міністерство освіти і науки України  
Київський міський педагогічний університет імені  
Б.Д. Грінченка**

**Кафедра інформатики  
та фізико-математичних дисциплін**

**АРХІТЕКТУРА  
ЕЛЕКТРОННО-  
ОБЧИСЛЮВАЛЬНИХ  
СИСТЕМ**

**Навчальний посібник**

**Київ  
2007**

**Рецензенти:**

**Кравець П.І.** – кандидат технічних наук, доцент (Національний технічний університет України «Київський політехнічний інститут»)

**Шекунов А.В.** – кандидат технічних наук, доцент (Київський міський педагогічний університет ім. Б.Д. Грінченка)

**Абрамов В.О. Архітектура електронно-обчислювальних машин:** навчальний посібник. – К.: КМПУ імені Б.Д. Грінченка, 2007. – 84 с.

Розглянуто структуру апаратних засобів електронно-обчислювальних машин, програмне керування комп'ютерними пристроями, підвищення ефективності та надійності комп'ютерних систем. Приділено увагу принципам взаємодії пристроїв персонального комп'ютера, побудові процесора та використанню системи його команд, багатопроцесорним комплексам. В посібнику наведені практичні питання для закріплення теоретичного матеріалу.

Матеріал посібника подається за модульною структурою відповідно до вимог кредитно-модульної системи організації навчального процесу.

Призначений для студентів педагогічних навчальних закладів, учнів і вчителів шкіл з поглибленим вивченням дисципліни «Інформатика».

*Рекомендовано науково-методичною радою  
Київського міського педагогічного університету імені  
Б.Д. Грінченка  
протокол № 6 від 19 березня 2007 року*

© В.О. Абрамов, 2007

© Київський міський педагогічний університет імені Б.Д.  
Грінченка, 2007

## АРХІТЕКТУРА ОБЧИСЛЮВАЛЬНИХ СИСТЕМ

## Зміст

**МОДУЛЬ 1 Апаратні засоби****1.1 Загальні поняття і принципи**

- 1.1.1 Принципи створення обчислювальних систем
- 1.1.2 Поняття архітектури комп'ютера
- 1.1.3 Апаратні і програмні засоби
- 1.1.4 Принципи взаємодії блоків у комп'ютері
- 1.1.5 Базова функціональна схема комп'ютера.

**1.2 Процесор**

- 1.2.1 Типова структура процесора
- 1.2.2 Мікропроцесорна пам'ять
- 1.2.3 Кеш-пам'ять процесора.
- 1.2.4 Сучасні процесори

**1.3 Пам'ять комп'ютера**

- 1.3.1 Типи пам'яті
- 1.3.2 Основна пам'ять
- 1.3.3 Принципи організації основної пам'яті
- 1.3.4 Сегментація пам'яті
- 1.3.5 Структура основної пам'яті

**1.4 Система введення-виведення**

- 1.4.1 Принцип роботи пристрою введення-виведення
- 1.4.2 Паралельний порт
- 1.4.3 Послідовний
- 1.4.4 Система переривань порт
- 1.4.5 Канали прямого доступу до пам'яті
- 1.4.6 Відеоадаптер
- 1.4.7 Звуковий пристрій

**1.5 Системна плата**

- 1.5.1 Структура системної плати
- 1.5.2 Конструкція ATX
- 1.5.3 Системна логіка
- 1.5.4 Шинні інтерфейси
- 1.5.5 Послідовні Інтерфейси
- 1.5.6 Загальні відомості про профілі устаткування

**МОДУЛЬ 2 Програмне управління пристроями****2.1 Поняття про машинну команду**

- 2.1.1 Представлення інформації
- 2.1.2 Структура машинної команди
- 2.1.3 Типи адресації
- 2.1.4 Система машинних команд процесора
- 2.1.5 Час виконання команд

**2.2 Машинна програма**

4

2.2.1 Структура машинної програми

2.2.2 Обробка переривань

2.2.3 Час виконання програми

### **2.3 Взаємодія програмних і апаратних засобів**

2.3.1 Ієрархія взаємодії

2.3.2 Базова система введення-виведення

2.3.3 Розподіл системних ресурсів

2.3.4 Адаптери, драйвери, контроллери

### **2.4 Програмне управління об'єктами**

2.4.1 Структура системи управління

2.4.2 Реалізація програмного управління устаткуванням

2.4.3 Розповсюджені системи управління

## **МОДУЛЬ 3 Комп'ютерні системи**

### **3.1 Класифікація комп'ютерних систем**

### **3.2 Проектування і системна інтеграція**

### **3.3 Якість комп'ютерних систем**

3.3.1 Характеристики якості

3.3.2 Основні параметри комп'ютера.

3.3.3 Конфігурація комп'ютера

### **3.4 Режими і технології роботи процесора**

3.4.1 Режими роботи

3.4.2 Нові технології обробки інформації

3.4.3 Конвейєрний обмін з пам'яттю.

### **3.5 Багатопроцесорні системи**

3.5.1 Однорідні багатопроцесорні системи

3.5.2 Розподілені функції управління

3.5.3 Розподілені процесори

### **3.6 Надійність системи і достовірність інформації**

3.7 Дослідження і тестування комп'ютера

3.7.1 Диспетчер пристроїв

3.7.2 Тестові програми

3.7.3 Програма Debug.exe

### **3.8 Профілактика і обслуговування комп'ютера**

### **Лабораторні роботи**

## МОДУЛЬ 1

### Апаратні засоби

Сучасний комп'ютер є складним комплексом технічних засобів, які функціонують як одне ціле і виконують загальне завдання. Головне завдання комп'ютера проводити обробку інформації шляхом реалізації різних інформаційних процесів відповідно до заданого алгоритму. Основними процесами у комп'ютері є обчислювальні (арифметичні, логічні операції) тому його ще називають обчислювальною машиною або електронно-обчислювальною машиною (ЕОМ). Цифрова обчислювальна машина оперує дискретними цифровими сигналами, а аналогова – аналоговими.

Комп'ютерна система це комплекс програмного забезпечення, обчислювального і телекомунікаційного устаткування, засобів введення і виведення даних, які створюють єдине середовище для вирішення завдань користувачів. Сучасні комп'ютери це, як правило, обчислювальна (або комп'ютерна) система.

У комп'ютерну систему можуть входити як одиночні або локальні комп'ютери, так і складні системи, що складаються з декількох процесорів або комп'ютерів, розташованих автономно або сполучених в обчислювальну мережу. Комп'ютерні системи можуть мати декілька однорідних процесорів з однаковими параметрами (наприклад, багатопроцесорні системи, багатоядерні процесори або декілька логічних процесорів в одному фізичному) або багато різномірних процесорів з різними параметрами (наприклад, співпроцесори або контролери, що беруть на себе частину функцій основного процесора).

Комп'ютерні системи призначені для вирішення наукових, учбових, інженерних завдань, управління об'єктами і підприємствами, створення інформаційно-довідкових систем. Комп'ютерна система слугує для реалізації і підтримка всіх інформаційних процесів, які здійснює чоловік в своїй діяльності.

### 1.1 Загальні поняття і принципи

#### 1.1.1 Принципи створення обчислювальних систем

*Комп'ютерна система або електронна обчислювальна машина (ЕОМ)* – це пристрій для обробки інформації, а саме виконання операцій введення інформації, перетворення її за певною програмою і виведення одержаних результатів у формі, придатній для сприймання людиною.

У цілому архітектуру ЕОМ визначають три принципи, які вперше запропонував американський вчений Дж. фон Нейман (1903 – 1957). Це такі принципи:

1. *Принцип програмного керування* – забезпечує автоматизацію процесу обчислень.

2. *Принцип програми, що зберігається в пам'яті ЕОМ* – команди, програми і дані подаються у вигляді чисел, також смороду зберігаються та обробляються таке саме, як і числа, що спрощує та прискорює обробку даних.

3. *Універсальність алгоритмів при розв'язанні завдань на ЕОМ* – набір операцій, які виконує універсальна ЕОМ є достатнім для записування будь-якого алгоритму, що реалізує чисельні методи розв'язання математичних завдань.

На підставі цих принципів можна стверджувати, що сучасна *ЕОМ* – це універсальний перетворювач інформації, технічний пристрій, який після введення в пам'ять початкових даних у вигляді цифрових кодів і програми їх оброблення, також у вигляді цифрових кодів, здатний автоматично виконувати обчислювальний алгоритм, який заданий програмою обчислень, і видати готові результати розв'язання задачі. Керування здійснюється тільки програмними засобами без змінення апаратної частини.

### **1.1.2 Поняття архітектури комп'ютера**

Як видно з попереднього визначення, в комп'ютері можна виділити дві групи засобів: апаратні (АЗ) і програмні (ПЗ), їх ще називають апаратне і програмне забезпечення. АС – це устаткування для введення, зберігання, перетворення і виведення даних (процесори, пам'ять, адаптери, канали зв'язку і т.і.). ПЗ – це сукупність програм для управління роботою АЗ і реалізації всіх функцій системи. Для розуміння роботи комп'ютерних систем потрібне розуміння як апаратних так і програмних засобів.

Склад, параметри та взаємозв'язки апаратних засобів (АЗ) і програмного забезпечення (ПЗ) ЕОМ, які складають комплекс і мають параметри та властивості єдиної системи, вважаються архітектурою цієї системи. Архітектура обчислювальних систем включає широкий круг питань пов'язаних з побудовою комплексу апаратних і програмних засобів. Архітектура системи визначає ту сукупність властивостей системи, які в даний момент часу важливі для користувача. Архітектура відображає структуру складових частин системи, способи їх взаємодії, звернення до її елементів, набір виконуваних операцій, формати даних, адресація даних, вхідні і вихідні зовнішні сигнали і реакція на них.

Архітектура - це організаційна структура комп'ютера, що має потоки та подання даних інтерфейси введення – виведення даних, набір команд, способи адресації, реєстри, апаратне і програмне забезпечення. Архітектура системи – це подання системи як сукупності її функціональних компонентів їхньої організації та взаємозв'язків (шин, сигналів, протоколів інтерфейсів тощо)

Одним з елементів архітектури є структура системи. Структура це порядок і принципи взаємодії елементів системи, що визначають основні її властивості. Архітектура сучасних комп'ютерів базується на певних стандартних і уніфікованих рішеннях, що дозволяє забезпечити їх сумісність і

взаємозамінюваність при можливості варіації технічних засобів для адаптації комп'ютера і настройки його на рішення специфічних задач.

Існують три базові функції, без яких не може бути створена жодна інформаційно-обчислювальна система (рис. 1). Це введення-виведення інформації, зберігання і перетворення інформації. Перша функція здійснює прив'язку до процесів навколишньої реальності, друга – до процесів, що тривають у часі і остання реалізує основну мету функціонування системи. Крім того, є додаткові функції по управлінню і інформаційній взаємодії між основними функціями.

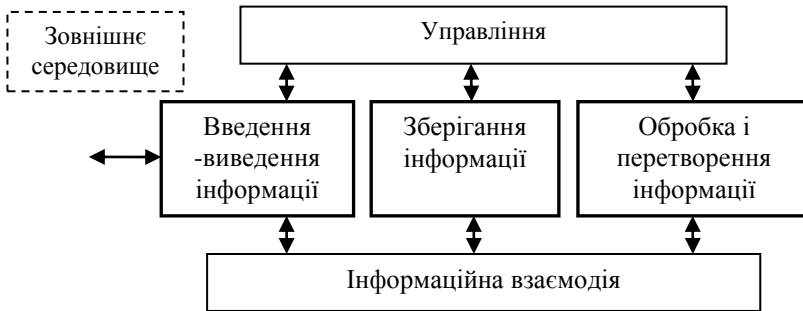


Рис. 1. Основні функції комп'ютера

Для реалізації цих функцій у будь-якій комп'ютерній системі мають бути технічні засоби і відбуватися відповідні інформаційні процеси (рис.2). Процеси – це послідовність цілеспрямованих дій з обробки даних, у яких задіяні певні ресурси. Ресурси обчислювальних систем – це можливості АЗ і ПЗ по обробці даних. Видів ресурсів багато, основні з них – об'єм всіх видів пам'яті, зайнятість програмних засобів і устаткування (процесора, пристроїв введення і виведення даних (ПВВ), периферійного устаткування) і т.і.

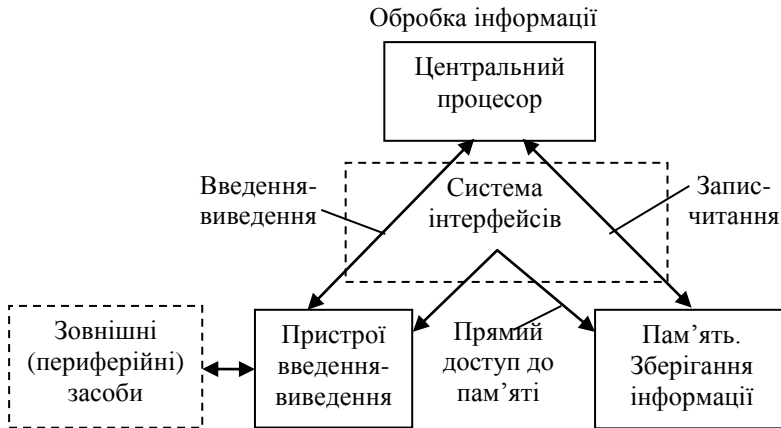


Рис. 2. Інформаційні процеси і потоки інформації у комп'ютері

Оскільки системи завжди багаторівневі і ієрархічні, то і архітектура комп'ютерних систем має багаторівневу структуру (рис.3). Зразком такої ієрархії може стати модель відкритих систем міжнародної організації стандартизації.

Завдання кожного рівня реалізуються всіма засобами нижчестоящих рівнів, які є платформою для цього рівня. Платформа це програмне, апаратне і мережеве середовище, у якому виконується подане завдання або функції.

Планування і виконання завдань кожного рівня не залежить від способу виконання завдань нижчестоящих рівнів. Тобто одні і ті ж завдання можуть виконуватися самими різними засобами і способами. Зворотне не вірно – засоби нижчестоящих рівнів (параметри і ресурси ) істотно визначаються завданнями вищестоящих рівнів.

Взаємодія між рівнями визначається міжрівневими інтерфейсами. Інтерфейси мають різний ступінь стандартизації (від міжнародних до корпоративних).



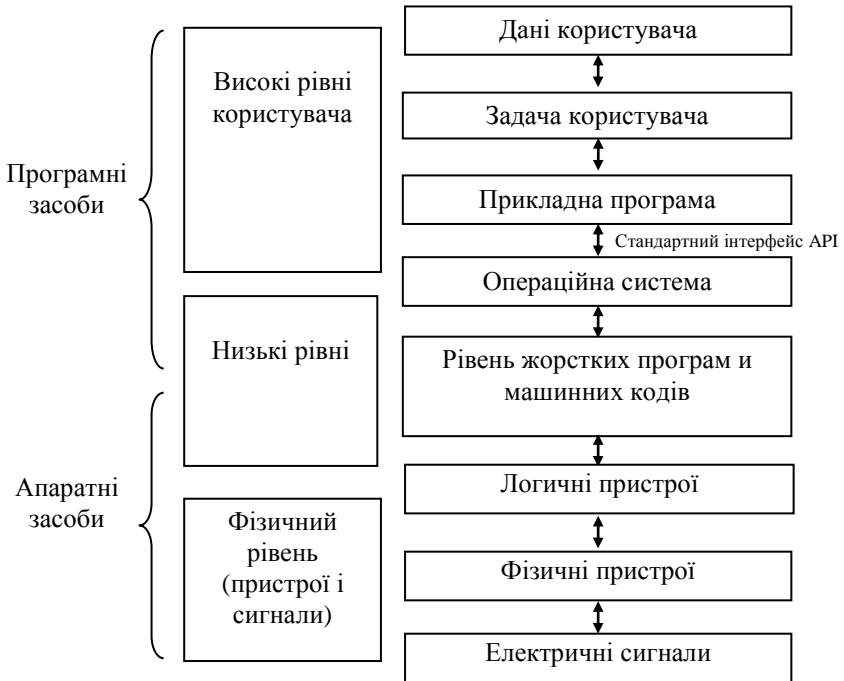


Рис. 3 Ієрархія рівнів архітектури комп'ютера

Архітектурні особливості комп'ютера визначаються наступними основними принципами:

**Магістральність** – всі блоки комп'ютера підключаються до загальної системи шин (магістралей) у вигляді уніфікованих каналів зв'язку, що передають дані і адреси пристроїв, яким ці дані призначені. Дані поступають на входи всіх пристроїв, але сприймають їх тільки ті, яким вони призначені. Цей принцип забезпечує високу гнучкість і легкість масштабування системи.

**Модульність** – комп'ютерні системи будуються на основі уніфікованого ряду модулів, що мають конструктивну закінченість і функціональну автономність.

**Мікропрограмування** – послідовність виконуваних операцій гнучко змінюється залежно від поставлених завдань. Альтернативний спосіб – апаратний, коли послідовність виконуваних операцій жорстко задана схемою.

**Ефективність функціонування ЕОМ** залежить від її параметрів або показників якості, які постійно поліпшуються завдяки розробці нових технологій обробки інформації та виробництва комп'ютерних компонентів.

### 1.1.3 Апаратні і програмні засоби

Персональний комп'ютер є універсальною системою, конфігурація якої може бути налаштована на рішення широкого спектру завдань і на застосування в різних областях. Конкретні апаратні засоби вибираються з урахуванням задач і умов роботи системи. Після цього проводиться їх настройка шляхом вибору в базовій системі введення-виведення (BIOS) параметрів і режимів роботи для кожного апаратного засобу. Таким чином, апаратні засоби комп'ютера готові до рішення певних класів завдань. Далі вибирається операційна система, і теж настроюються її параметри. Остаточна настройка комп'ютера на рішення певної задачі проводиться програмою, що реалізує алгоритм рішення цієї задачі. Всі ці настройки повинні забезпечити максимальну ефективність комп'ютера в рішенні конкретних задач.

Апаратні і програмні засоби комп'ютера працюють у тісній і нерозривній взаємодії, а сполучною ланкою між ними є базова система введення-виведення (BIOS). Апаратна і програмна обробка даних це два принципово різних альтернативних способів обробки.

Прикладом апаратного управління є набір системної логіки. У цих мікросхемах проводиться обробка інформації по наперед заданим незмінним алгоритмам, які задані самою схемою. Принципово змінити ці алгоритми неможливо, завдяки цьому, швидкодія апаратних засобів на багато вище чим програмних. Апаратними засобами комп'ютера є процесор і зовнішні (по відношенню до процесора) пристрої. До зовнішніх відносяться всі види пам'яті і пристрої введення-виведення. Всі вузли комп'ютера, системної платні і самого процесора управляються апаратним чином, тому вони мають достатньо високу швидкодію.

При програмному управлінні на одну прикладну операцію виконується безліч допоміжних і службових операцій (читання і розшифровка команд, видача адрес і читання даних і т.і.). У рішенні будь-якої задачі беруть участь багато інших програм: прикладні програми виконують алгоритм завдання, підпрограми операційної системи контролюють роботу прикладних програм, розподіляють ресурси, драйвери управляють роботою апаратних пристроїв, антивіруси контролюють пам'ять і т.і. Крім того, у комп'ютері одночасно виконуються безліч програм і головна серед них операційна система, яка управляє роботою всієї решти програм і апаратних засобів.

Все це призводить до того, що швидкодія програмно-керованих пристроїв в порівнянні з апаратними засобами значно нижче. Але головною перевагою програмних засобів є їх універсальність, яка істотно знижує вартість рішення різних задач. Тому там, де потрібна висока швидкодія, застосовуються апаратні засоби, а там де головне універсальність використовуються програмні.

Таким чином, в комп'ютері є функції, частина з яких реалізована апаратними, а частина - програмними засобами. А їх взаємодію здійснює

базова система введення-виведення (BIOS – base input output system). BIOS є сполучною ланкою між програмними і апаратними засобами комп'ютера.

Апаратні і програмні засоби низького рівню складають системні ресурси комп'ютера. Їх кількість обмежена і тому робота на низькому рівні постійно пов'язана з розподілом і спільним використанням системних ресурсів прикладними програмами високого рівня.

#### 1.1.4 Принципи взаємодії блоків у комп'ютері

Всі сучасні електронні апарати, що мають багато вузлів і блоків, будуються за принципом загального інтерфейсу, зокрема використовується інтерфейс загальна шина. Всі пристрої через відповідні адаптери сполучені з інтерфейсом, по якому проводять обмін інформацією між собою (рис. 4).

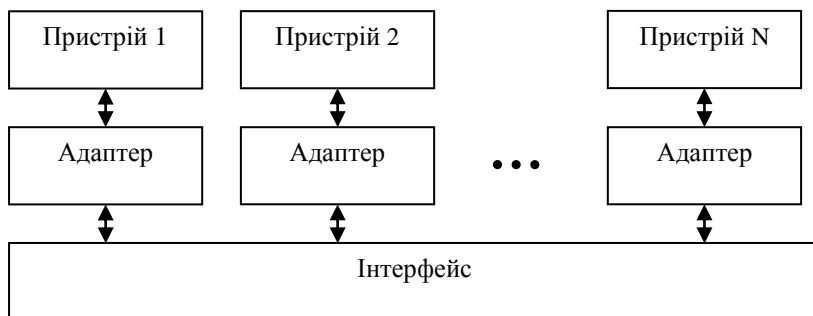


Рис. 4. Структура систем з загальним інтерфейсом

Інтерфейс – це засоби взаємодії різних блоків комп'ютера, що мають різну будову і принцип дії. Для взаєморозуміння блоків їх інтерфейси повинні бути зрозумілими один одному, тобто відповідати стандарту. Як правило, інтерфейси не тільки стандартизовані, але і забезпечуються типовими наборами мікросхем і програм. У комп'ютері є два типу інтерфейсів: внутрішньомашинні (для зв'язку блоків комп'ютера) і периферійні (для зв'язку комп'ютера з периферійними пристроями введення-виведення).

Інтерфейси мають два рівні: фізичний (дроти, роз'єми, контакти) і логічний (частота, полярність, амплітуда, тривалість сигналів, протоколи взаємодії).

Розрізняють інтерфейси двоточкові (для зв'язку двох пристроїв) і багатоточкові (для зв'язку між собою багатьох пристроїв). Існують також інтерфейси паралельні і послідовні. Багатоточкові паралельні інтерфейси мають назву загальна шина. Шина - це набір провідників, по яким передаються дані, адреса або сигнали управління і технічні засоби

забезпечення з'єднання. У паралельному інтерфейсі всі розряди  $M$ -розрядного коду передаються одночасно по  $M$  проводам за один такт. Водночас по шині адреси передається адреса комірки пам'яті, до якої проводиться звернення, тобто в яку або з якої пересилаються дані. Збільшення кількості провідників шини адреси дозволяє збільшити число елементів пам'яті (максимальний об'єм пам'яті), що адресується процесором.

До кожного шинного інтерфейсу підключаються декілька пристроїв, але передача здійснюється тільки між двома з них. Вибір і управління, пристроями виконується по шині адреси (ША) і шині управління (ШУ), а передача даних по шині даних (ШД). Дані і адреси в комп'ютері передаються у вигляді двійкових цифр через однакові проміжки часу (такти), задаються тактовою частотою. Кожен тип шини має свою тактову частоту, яка може мати значення від найбільшої тактової частоти процесора до найнижчої частоти низькошвидкісних пристроїв введення-виведення (клавіатура, мишка і т.д.).

Недоліком паралельних інтерфейсів є наявність великої кількості проводів і дуже високі вимоги до ідентичності їх фізичних параметрів (опір, ємність, індуктивність). І чим більше швидкості передачі, тим вище ці вимоги. Полепшити реальні фізичні параметри проводів дуже важко, тому швидкість передачі в паралельних інтерфейсах обмежена. Так при сучасних швидкостях паралельні інтерфейси вже працюють на межі своїх можливостей.

У послідовному інтерфейсі передаються всі  $M$  розрядів послідовно у часі по одному проводу (мультиплексування сигналів), тому швидкість передачі двійкових розрядів в  $M$  разів більше. Реальні фізичні параметри проводів уже не мають такого значення, тому максимальна допустима швидкість передачі по одному проводу послідовного інтерфейсу набагато вище, ніж у паралельних. Підвищувати швидкість сприяє також те, що сучасні мікросхеми забезпечують достатньо високу швидкодію. Тому зараз послідовні інтерфейси знаходять все більше застосування.

На рис.5 представлений приклад часових діаграм сигналів паралельного і послідовного інтерфейсів. На рисунку показано, що за будь-який часовий інтервал  $T_1$ ,  $T_2$  або  $T_3$  передається три розряди даних  $P_1$ ,  $P_2$ ,  $P_3$ . У паралельному інтерфейсі вони передаються одночасно по трьом проводам, а в послідовному - по черзі по одному дроту. Унаслідок різних параметрів проводів паралельного інтерфейсу виникає нерівномірна затримка сигналів ( $T_{z1}$ ,  $T_{z2}$ ,  $T_{z3}$ ) і на приймальну сторону сигнали приходять не одночасно, що створює проблему з їх ресстрацією в момент часу  $T_{пр}$ . У послідовному інтерфейсі затримка всіх сигналів однакова ( $T_z$ ) і проблем з нерівномірністю затримки сигналів при прийомі немає.

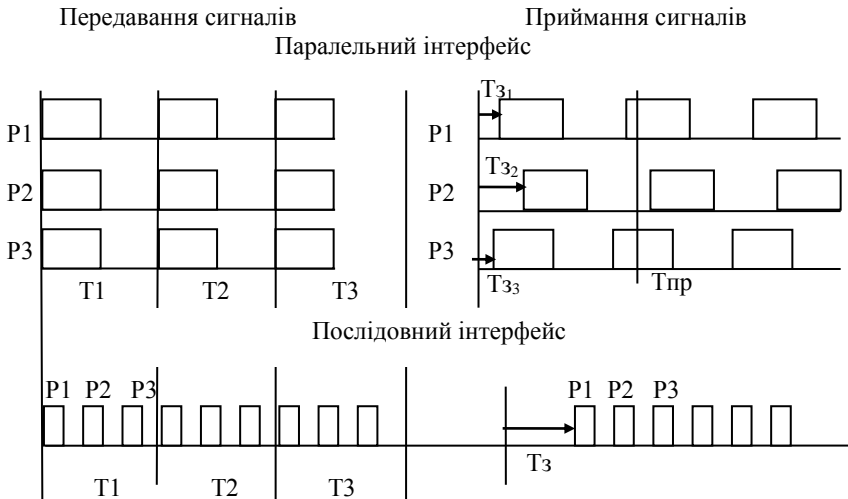


Рис. 5 Сигнали паралельного та послідовного інтерфейса

У даний час у якості внутрішнього інтерфейса в комп'ютерах використовується загальна шина. Шина характеризується пропускнуною спроможністю, яка залежить від розрядності і тактової частоти шини, а також числом пристроїв, що підключаються.

У комп'ютері є багато різних інтерфейсів з шинною організацією. Наприклад, системна шина – набір провідників, що з'єднує процесор з системною платою. Пристрої не заважають друг-другу, оскільки виходи неактивних пристроїв знаходяться в так званому третьому (високоомному) стані і не впливають на передачу сигналів по шині. При цьому передачу і прийом проводять тільки одна пара пристроїв, вибраних і керованих по шині адреси (ША) і управління (ШУ).

### 1.1.5 Базова функціональна схема комп'ютера.

Усі обчислювальні системи мають однакову базову структуру, яка включає один чи декілька центральних процесорів і сопроцесорів, що здійснюють обробку інформації, пристрої збереження інформації, а також пристрої введення та виведення інформації для спілкування із зовнішнім середовищем та іншими обчислювальними системами.

У ранніх поколіннях комп'ютерів процесори і зовнішні пристрої (пам'ять і пристрої введення-виведення) мали приблизно однакову і дуже невисоку тактову частоту (біля 10 МГц). Тому взаємодія процесора і зовнішніх пристроїв здійснювалася через єдиний інтерфейс – системну шину, яка працювала з тактовою частотою процесора. Швидкодія процесорів збільшувалася швидше за зовнішні пристрої. Тому тактова частота системної

шини зростала, а пам'ять і пристрої введення-виведення не могли працювати з такою частотою. Були розроблені спеціальні адаптери, які підключалися до системної шини і на знижених частотах взаємодіяли з зовнішніми пристроями. Так виникли шини пам'яті і шини розширення (ISA і ін.). При цьому, для збільшення швидкодії пам'яті була застосована концепція кеш пам'яті. Це пам'ять невеликого об'єму і високої швидкодії, яка працювала на високій частоті системної шини.

Із зростанням швидкодії окремих зовнішніх пристроїв довелося розробити локальні швидкодіючі шини (наприклад PCI, SCSI). Але надалі, всі зовнішні пристрої стали достатньо швидкодіючими і потреба в повільних шинах розширення відпала, їх місце зайняли локальні шини. Для обслуговування високошвидкісних пристроїв були розроблені спеціальні мікросхеми, які потім об'єдналися у великі інтегральні мікросхеми (ЧІП). Наприклад, «північний міст» і інші аналогічні.

Але все ще залишалися пристрої з принципово низькою швидкістю. Для їх обслуговування також створюються спеціальні мікросхеми, наприклад «південний міст». Для обслуговування дуже низькошвидкісних пристроїв, таких як модем, мишка, клавіатура, використовуються спеціальні мікросхеми (наприклад типу Super I/O).

Зараз паралельні шинні інтерфейси все частіше замінюються послідовними, а для різних по швидкодії груп пристроїв використовуються свої паралельні або послідовні інтерфейси.

На рис. 6 представлена спрощена функціональна базова схема, яка дозволяє одержати загальне уявлення про функції, пристрій і принцип дії комп'ютера. Основний елемент комп'ютера центральний процесор (ЦП) працює разом з математичним співпроцесором (сучасні процесори об'єднані із співпроцесорами в одному корпусі) і синхронізується тактовим генератором (Г). Із зовнішнім світом процесор взаємодіє через системну шину, яка через набір мікросхем системної логіки перетворюється у більш повільні інтерфейси. Загалом вони утворюють систему інтерфейсів, через яку з'єднується процесор і зовнішні пристрої.

Системна шина, до якої підключається процесор, ще називається процесорною шиною або передньою шиною ( Front Side Bus - FSB ) або внутрішньою шиною (Processor Side Bus - PSB). Існують ще шини розширення (ISA), локальні шини (PCI, SCSI і ін.). Блоки управління інтерфейсами (контроллери) знаходяться в спеціальних мікросхемах – Чіпах системної логіки. Процесор видає адреси і сигнали управління і обмінюється інформацією з пам'яттю, пристроями введення-виведення і зовнішніми пристроями.



Рис. 6. Узагальнена базова функціональна схема

Через шинний інтерфейс підключена основна пам'ять. Вона складається з швидкодіючої пам'яті для короткочасного оперативного зберігання даних ОЗП (оперативний запам'ятовуючий пристрій або RAM – random access memory) і енергонезалежної пам'яті (ЕНЗУ), зокрема постійної пам'яті (ПЗП-постійний запам'ятовуючий пристрій, або ROM – read only memory). Для довготривалого зберігання даних використовуються зовнішні накопичувачі. Наприклад, НЖМД (накопичувач на жорстких магнітних дисках) має низьку швидкість, але великий об'єм і зберігає дані при виключенні живлення. НГМД - накопичувач на гнучких магнітних дисках, місткість його невелика, але дискети зручні для перенесення інформації.

Безпосередньо до процесора підключена кеш-пам'ять, завдяки цьому реалізується її висока швидкість.

Зовнішні пристрої підключаються до інтерфейсів через спеціальні апаратні засоби адаптери і контроллери (по термінології ІВМ це синоніми). Введення-виведення даних на зовнішні пристрої проводиться через відповідні контроллери і порти. У операціях введення-виведення задіяний контроллер переривань, а також контроллер прямого доступу до пам'яті (ПДП - DMA).

При перериванні процесор тимчасово припиняє свою роботу і обслуговує пристрій введення-виведення, а при прямому доступі до пам'яті процесор

відключається від шини і не заважає прямому високошвидкісному потоку даних між зовнішніми пристроями і пам'яттю.

Електроживлення всіх пристроїв здійснюється через блок живлення (БЖ).

## 1.2 Процесор

Процесор (центральний процесор – ЦП, мікропроцесор – МП, CPU – Central Processing Unit) – програмно керований пристрій обробки інформації. ЦП призначений для управління роботою комп'ютера і виконання арифметичних і логічних операцій.

Процесор це одна або декілька великих інтегральних схем (ЧІП), що містять мільйони транзисторів.

### 1.2.1 Типова структура процесора

Існує велика кількість типів процесорів, які відрізняються своїми параметрами. На рис. 7 представлена логічна організація процесора с одним акумулятором. Двійкові числа, з якими виконуються дії, через інтерфейс поступають з системної шини і зберігаються в одному з буферних регістрів, акумуляторі або кеш-пам'яті. А арифметико-логічному пристрої (АЛП) виконуються арифметичні або логічні дії з цими числами, а результат зберігається у регістрах. Керує всіма діями пристрій керування.

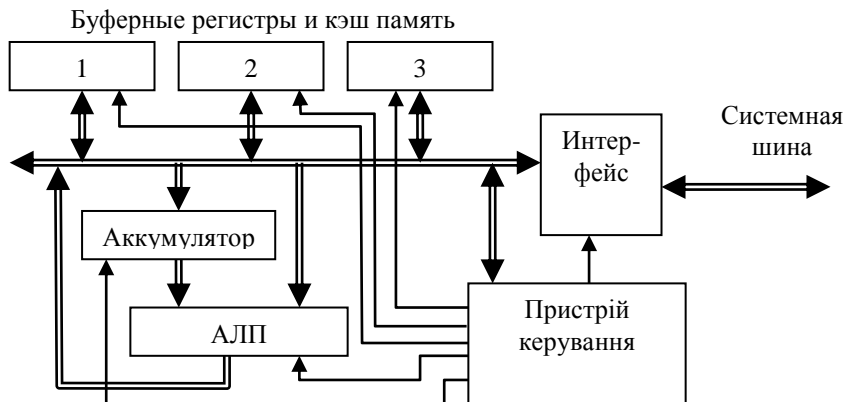
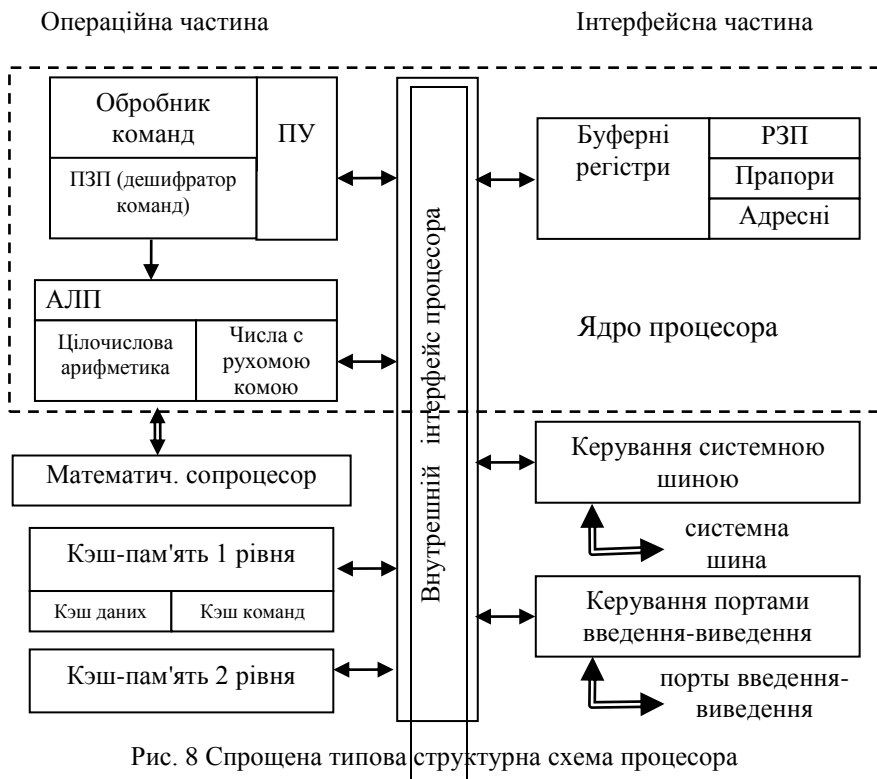


Рис. 7 Логічна організація процесора с одним акумулятором

На рис. 8 представлена спрощена типова структурна схема одного з можливих варіантів процесорів.





Пристрій управління (ПУ) містить: дешифратор операцій, блок випереджаючого їх виконання і прогнозу галузень, ПЗП мікропрограм, вузол формування адреси. ПЗП мікрокоманд призначений для зберігання і дешифрування машинних команд і формування алгоритмів їх виконання.

Арифметико-логічний пристрій (АЛП або операційний блок) здійснює арифметичні і логічні операції, вон містить: регістри, суматор, схему управління. У різних процесорах АЛП виконує різні функції, але основними є наступні:

- Складання з перенесенням і віднімання із заемом.
- Зеув вліво і управо.
- Логічне множення і складання.
- Порівняння цифрових кодів.

Деякі процесори мають цільове призначення, тому їх операції теж специфічні.

Регістри і кеш-пам'ять служать для тимчасового зберігання даних з дуже малим часом доступу. Всі пристрої взаємодіють між собою через внутрішній

шинний інтерфейс, а із зовнішніми пристроями через блоки управління системною шиною і операціями введення-виведення.

Основні властивості процесора визначаються його системою команд, наявністю співпроцесора, тактовою частотою, числом буферних регістрів мікропроцесорної пам'яті і об'ємом кеш-пам'яті.

Процесори класифікуються по типу і розрядності системної шини і розрядності буферних регістрів. Сучасні процесори Pentium мають розрядність шини 64, а регістрів в основному - 32 (тільки найновіші 64). Таким чином, майже всі процесори працюють з 32 розрядними даними і командами при 64 розрядній шині даних. Важливими параметрами є об'єм кеш-пам'яті і об'єм оперативної пам'яті, що адресується процесором, цей об'єм досягає 64 Гбайт.

Для розширення функціональних можливостей процесора він доповнюється математичним співпроцесором. Основні функції співпроцесора: прискорення операцій над числами з фіксованою і рухомою комою, обчислення масивів і функцій, обробка мультимедійної інформації.

### **1.2.2 Мікропроцесорна пам'ять**

Мікропроцесорна пам'ять має у своєму складі: регістри загального призначення (РЗП) або універсальні регістри, сегментні регістри, регістри звуку, регістр прапорів. Вони призначені для тимчасового зберігання і швидкого доступу до адрес і проміжних даних при виконанні операцій процесором. Регістри поділяють на доступні програмісту і внутрішні. Сукупність регістрів ЦП, які доступні програмісту, називають реєстровим іайлом або набором регістрів.

У базовій моделі процесора розглядається 14 регістрів пам'яті (рис.9 ). Реально в різних процесорах буває до 256 регістрів розрядністю до 8 байт (Pentium)

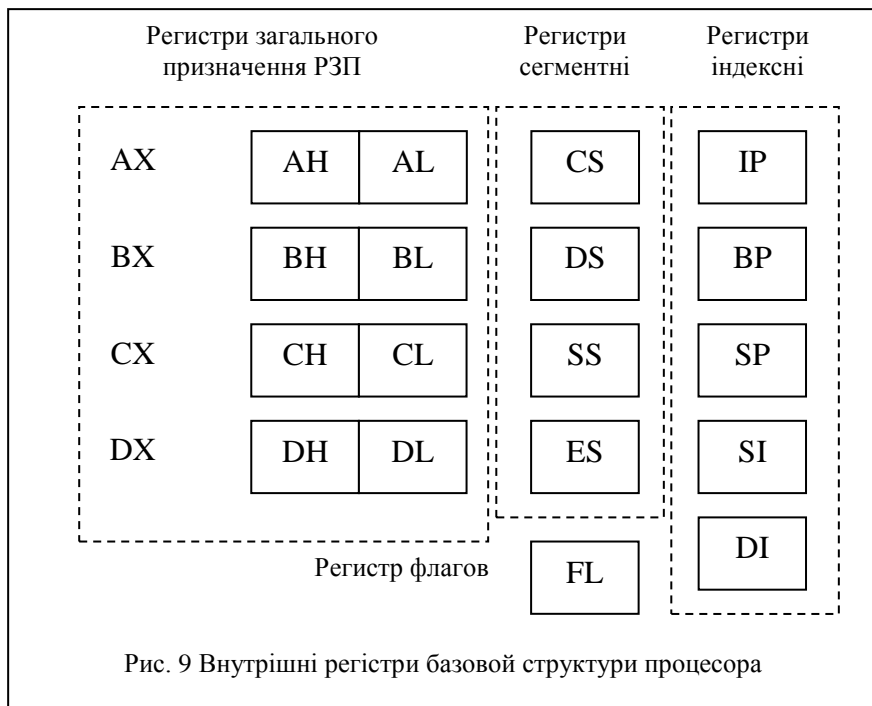


Рис. 9 Внутрішні реєстри базової структури процесора

### Регистри загального призначення (РЗП).

Однорозрядні позначаються однією буквою А, В, С, D

16-ти розрядні (2-байтні) подвійними буквами AX, BX, CX, DX

8-ми розрядні половинки 16-ти розрядних реєстрів - додається буква L, H. Наприклад, AL, AH .

4-байтні - додається буква E: наприклад, EAX, EBX .

Спеціалізовані реєстри

A – акумулятор- накопичує результати обчислень. Це основний робочий реєстр, в ньому зберігаються операнди майже всіх операцій і через нього проводиться основне звернення до пам'яті.

B – зберігання бази сегментної частини адреси,

C – лічильник повторень операцій,

D – реєстр даних, використовується самостійно або спільно з A.

### Регистр прапорів (станів) - FL

Служить для індикації стану процесора. Стан відбивається в окремих бітах, які можуть бути умовами для виконання різних дій. Біти умов називаються ще логічними змінними або прапорами. У різних процесорах можуть бути різні прапори, але більшість має наступні прапори:

Статусні прапори:

CF (Carry) – прапор перенесення із старшого розряду при арифметичних операціях і зсуві.

PF (Parity) – парність, результат має парне число одиниць.

AF (Auxiliary Carry) – перенесення в двійково-десяткових операціях.

ZF (Zero) – результат операції рівний нулю.

SF (Sing) – знак отриманого результату

OF (Overflow) – переповнювання, результат операції дуже великий (наприклад, при діленні на нуль)

Прапори, що управляють:

TF (Trap) – трасування, покрокове виконання програми.

IF (Interrupt) – дозволяються переривання.

DF (Direction) – напрям обробки строкових масивів шляхом зміни вмісту регістрів SI і DI на +1 або - 1.

### Адресні регістри

IP -Указатель (лічильник) команд (IP – instruction pointer) - Містить зсув адреси команди, яка виконується. Спільно з регістром сегменту і базовим регістром вказує фізичний елемент пам'яті, що адресується. Після виконання чергової команди стан лічильника автоматично збільшується і він вказує на наступний елемент пам'яті, де починається наступна команда.

BP – (base pointer) регістр базової частини адреси.

SP (stack pointer) – покажчик стека, зсув вершини стека.

SI (source index), DI (destination index) – регістри індексу джерела і призначення.

CS, DS, ES, SS – регістри сегментів коду, даних, додатковий і стека.

### 1.2.3 Кеш-пам'ять процесора.

Швидкодія процесора на багато більше, ніж основній оперативній пам'яті ОЗП. А оскільки звернення до пам'яті відбувається постійно, то вона обмежує реальну швидкість процесора. Проблема вирішується використанням швидкодіючої кеш-пам'яті невеликого об'єму, в якій дублюється вміст тієї частини ОЗУ, до якої чергове звернення процесора буде найбільш вірогідне. Після цього процесор звертається вже не до ОЗП, а до кеш-пам'яті, швидкодія якої у багато разів більша ніж у ОЗП. Але можливі випадки, коли прогнози не виправдовуються, тоді доводиться наново завантажувати інформацію з ОЗП у кеш-пам'ять. На це йде час, в перебігу якого процесор знаходиться в режимі очікування.

В результаті при збільшенні об'єму кеш-пам'яті реальна швидкість роботи процесора значно збільшується, але не нескінченно. Природно, що вона обмежена можливостями самого процесора. На рис.10 показана орієнтовна залежність швидкодії комп'ютера від місткості кеш-пам'яті.

Кеш-пам'ять має три рівні L1 – усередині процесора, L2 – поза процесором і працює з внутрішньою частотою процесора (у процесорах після 1999 року інтегрована в ядро), L3 - на системній платні. Перший рівень самий швидкодіючий, але обмежений за об'ємом розмірами кристала процесора. Зовнішні пристрої пам'яті і пристрої введення-виведення також мають свою кеш пам'ять для прискорення їх роботи.

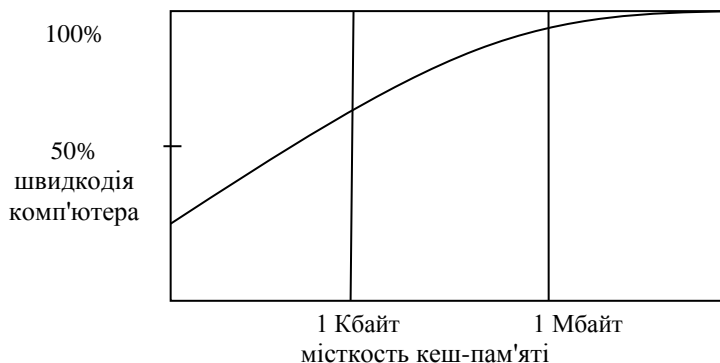


Рис. 10. Залежність швидкодії комп'ютера від місткості кеш-пам'яті

#### 1.2.4 Сучасні процесори

Сучасні процесори мають тактову частоту до 3-4 ГГц, розрядність до 128 біт, об'єм кеш-пам'яті знаходиться у межах 8 – 256 кбайт, кількість регістрів до 256. В сучасних процесорах застосовується величезна кількість різних технологічних новацій, які підняли як їх продуктивність, так і складність на висоту, яка ще 10 років назад здавалася неймовірною.

До першої групи новацій слід віднести технології обробки чисел з плаваючою комою. Другу групу складають технології паралельного і конвейєрного виконання операцій. Так, наприклад, технологія EPIC та інші дозволяють виконувати до 20 операцій за один такт і до 8 операцій над числами з рухомою комою. Один із способів досягти цього полягає в збільшенні числа суматорів арифметичного пристрою, числа регістрів і т.д. Число регістрів вже досягло 128 для цілих чисел 128 для чисел з рухомою комою.

Операції виконуються не тільки паралельно, але навіть забігаючи вперед. Спеціальні блоки і алгоритми дозволяють передбачати команди і галуження програми, що прискорює в середньому роботу комп'ютера.

Детальніше деякі технології описані у розділі 3.

### 1.3 Пам'ять комп'ютера

Пам'ять другий після процесора компонент комп'ютера по важливості і вартості. У комп'ютері є багато різних видів пам'яті, їх наявність і параметри визначають основні властивості всього комп'ютера.

### **1.3.1 Типи пам'яті**

Коротка класифікація основних властивостей усіх типів пам'яті зображена на рис.11.

За способом запису і вибірки даних з пам'яті розрізняють запам'ятовуючі пристрої (ЗП) з довільною вибіркою (Random Access Memory - RAM) і пам'ять з послідовною вибіркою. Перше означає, що записувати і читати можна довільні комірки пам'яті у будь-якій послідовності. Для цього досить указувати відповідні адреси комірок. При послідовній вибірці звернення до даних проводиться тільки строго у певному порядку, тому тут немає необхідності указувати адреси, що спрощує роботу з пам'яттю. Наприклад, послідовну вибірку мають пристрої пам'яті із стрічковими носіями. Іншим прикладом послідовної вибірки є стекова пам'ять.

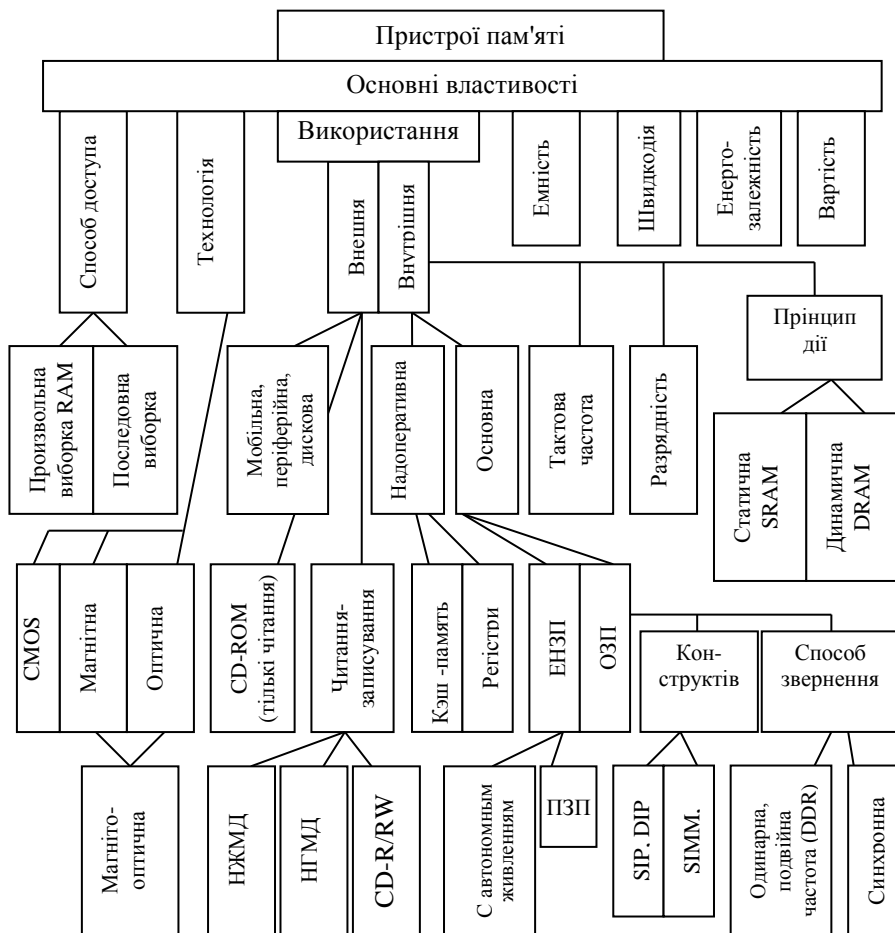


Рис. 11. Коротка класифікація основних властивостей пам'яті.

По місцю використання у комп'ютері і функціям можна виділити наступні типи пам'яті:

- Мікропроцесорні регістри (МПР)
- Кеш-пам'ять (КЕШ)
- Основна пам'ять (ОЗП – оперативний ЗП, ЕНЗП – енергонезалежний ЗП, ПЗП – постійний ЗП)
- Зовнішня пам'ять (змінна, флэш, периферійна, дискова, стрічкова – ПМЛ, оптична, - CD)

Дискова пам'ять: НЖМД – накопичувач на жорстких магнітних дисках («вінчестер»), НГМД - накопичувач на гнучких магнітних дисках.

CD – ROM, CD - R, CD - RW – накопичувачі на лазерних компакт-дисках (ROM - тільки для читання, R - записуваний, RW - перезаписуваний).

CMOS – пам'ять на основі технології малоспоживаючих структур працює з автономним живленням.

Всі типи пам'яті мають різні параметри і характеристики, достоїнства і недоліки. Немає жодного типу пам'яті, що повністю задовольняє всім вимогам. Саме це і обумовлює їх різноманіття у комп'ютері. На рис.12 представлено порівняння різних типів пам'яті по параметрам об'єм і швидкодія. Від параметрів залежить область застосування кожного типу пам'яті.

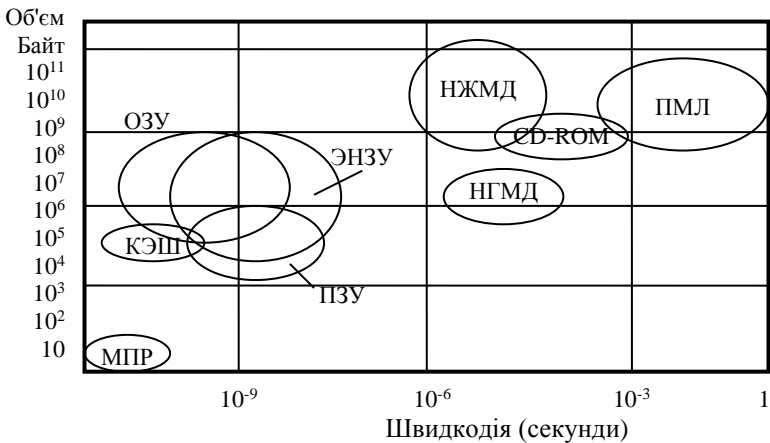


Рис.12. Зрівняння різних видів пам'яті по об'єму та швидкодії

Надоперативна пам'ять має дуже високу швидкодію і невисоку місткість. Місткість основної пам'яті значно вищий, але швидкодія гірше. Ці види пам'яті використовують для тимчасового зберігання інформації при включеному комп'ютері.

Зовнішня пам'ять має значні об'єми і дуже низьку швидкодію. Позитивна властивість цієї пам'яті в тому, що її вміст зберігається при виключенні живлення. Всі ці якості визначають її застосування для довготривалого зберігання інформації.

Пам'ять на магнітній стрічці застосовують іноді для резервного копіювання даних, де не потрібен швидкий пошук інформації.

З принципу дії пам'ять комп'ютера буває наступних типів:



1. Статична пам'ять (SRAM) побудована на статичних тригерах, має високу вартість і застосовується там, де потрібен висока швидкодія і невелика місткість (регістрова пам'ять і кеш-пам'ять з швидкодією 4 нсек). Використовується також в енергонезалежній пам'яті з автономним живленням.

2. Динамічна (DRAM) – інформація зберігається у вигляді електричного заряду і потрібне постійне її оновлення. Вона дешевша і може бути більшого об'єму чим статична. Використовується в пристроях оперативної пам'яті (ОЗП).

Порівняльна характеристика двох видів пам'яті:

Тип пам'яті	параметри				
	Швидкодія	Об'єм	Енерго-споживання	Вартість	Застосування
SRAM	висока	низький	високе	висока	МПР, КЕШ
DRAM	мала	високий	низьке	низька	ОЗП

Окрім цих параметрів при виборі враховуються експлуатаційні, заводстійкість і інші характеристики.

З рис. 14 видно, що динамічний ОЗП має високу швидкодію (близько 16 - 533 МГц або 2 - 60 нсек), але об'єм істотно менше, ніж у НЖМД, окрім того НЖМД зберігає інформацію при виключенні живлення. Тому вони застосовуються завжди у парі. Спочатку інформація завантажується з НЖМД в ОЗП і лише після цього процесор з високою швидкістю її обробляє. Програми і дані, що знаходяться в НЖМД просто зберігаються, а завантажені в ОЗУ вважаються запущеними або активованими. При виключенні живлення інформація з ОЗП знову перевантажується в НЖМД. Точно також здійснюється перевантаження з ОЗП в кеш-пам'ять, швидкодію і вартість якої ще вище, а об'єм істотно нижче, ніж у ОЗП.

В процесі роботи комп'ютера відбувається постійний обмін даними між ОЗП і НЖМД: деякі дані додатково прочитуються з НЖМД, якісь тимчасово розташовуються в НЖМД щоб звільнити місце в ОЗП (файли підкачки).

Не дивлячись на високу швидкодію динамічного ОЗП, воно у декілька разів менше швидкодії процесора (1 - 3 ГГц). Тому для роботи з ОЗУ використовується спеціальний інтерфейс з тактовою частотою нижче, ніж у процесора, а при зверненні до ОЗП процесору доводиться якийсь час чекати відповіді. Такі прості знижують загальну швидкодію комп'ютера.

З тактовою частотою процесора може працювати тільки статична пам'ять, яка має час доступу менше 2 нсек, тому вона використовується в кеш-пам'яті. При зверненні до кеш-пам'яті процесор відразу ж одержує відповідь, якщо в ній міститься необхідна інформація. Якщо ж контроллер помилився і

неправильно передбачив кешуєму область пам'яті то процесору доведеться звертатися в основну пам'ять. Такі помилки знижують ефективність роботи кеш-пам'яті.

### 1.3.2 Основна пам'ять

Основна пам'ять комп'ютера це пам'ять з довільною вибіркою (RAM) призначена для зберігання даних і програм під час роботи комп'ютера. Як видно з рис. 4, основна пам'ять складається з оперативної (ОЗП) і енергонезалежної (ЕНЗП) пам'яті. ОЗП це робоча область пам'яті процесора призначена для тимчасового запису і читання даних і програм тільки під час роботи комп'ютера, при виключенні живлення інформація в ній втрачається. На відміну від цього, ЕНЗП зберігає дані і програми також і при вимкненому живленні. Для цього пам'ять має власне автономне живлення, а деякі типи пам'яті (які виконані за спеціальною технологією) для зберігання інформації живлення не вимагають.

Постійна пам'ять (ПЗП / ROM - Read Only Memory – тільки для читання) незалежно від живлення постійно зберігає дані і програми, які не передбачається часто оновлювати, а іноді взагалі неможливо відновити. Існують різні типи ПЗП. Наприклад, масочные ПЗП (ROM) – інформація в них заноситься при виробництві і не змінюється (не перепрограмується) в процесі роботи. ППЗП (PROM) – пам'ять, одноразово перепрограмована на спеціальних пристроях. EPROM – багато разів перепрограмована пам'ять. Сюди відноситься електрично перепрограмована пам'ять EEPROM, використовується в пристроях флеш-пам'яті.

Конструктивно мікросхеми пам'яті випускаються в різних типах корпусів:

SIP – корпус з однорядним розташуванням виводів,

DIP – корпус з дворядним розташуванням виводів.

Декілька мікросхем розташовані на платні утворюють модуль пам'яті:

SIMM - Модулі (плата) пам'яті з однорядним розташуванням контактів роз'єму (застарілі)

DIMM – Нові модулі з контактами в два ряди мають високу місткість і тактову частоту 100МГц і 133 МГц

RIMM – Новітні швидкодійні модулі пам'яті.

Нові типи оперативної пам'яті відрізняються способом обміну даними:

FPM DRAM, RAM EDO - застарілі типи пам'яті.

BEDO DRAM – пам'ять, в якій обмін з процесором відбувається за один такт блоками постійної максимальної довжини.

SDRAM – новіший тип пам'яті, синхронізований з роботою процесора, має блоковий спосіб обміну і конвейєрну технологію, при якій поки встановлюється адреса в одній частині пам'яті (банку) відбувається вибірка з іншого банку.

DDR SDRAM – синхронна динамічна пам'ять з подвійною частотою передачі даних по двом фронтам тактового сигналу. Випускаються у вигляді модуля DIMM.

DRDRAM – перспективний тип пам'яті з двобайтовою шиною Rambus з частотою 800 МГц. FERAM, MRAM - ферроелектрична і магнітна пам'ять високої швидкодії.

### 1.3.3 Принципи організації основної пам'яті

Основна пам'ять складається з комірок, кожна з яких зберігає інформацію 1 біт. Якби комірка була одна, до неї можна було б звертатися безпосередньо, але коли їх багато їх вибирають за адресою, а звертаються через шину адреси і дешифратор адреси ДШ (рис.8).

Для М-розрядних шин даних 1-розрядні комірки об'єднуються в М-розрядні із зверненням до них за однією адресою, а обмін даними здійснюється по М проводам шини даних. Обмін інформацією процесора з пам'яттю відбувається за час від одного до трьох тактів плюс час очікування.

ОЗП і ПЗП мають єдиний адресний простір, який визначає максимальну кількість комірок основної пам'яті, що безпосередньо адресуються. При розрядності адресної шини N, місткість адресного простору рівно  $2^N$ .

На рис. 13 приведена таблиця адрес для 4- розрядної шини адреси. В цьому випадку, місткість адресного простору пам'яті рівна 16.

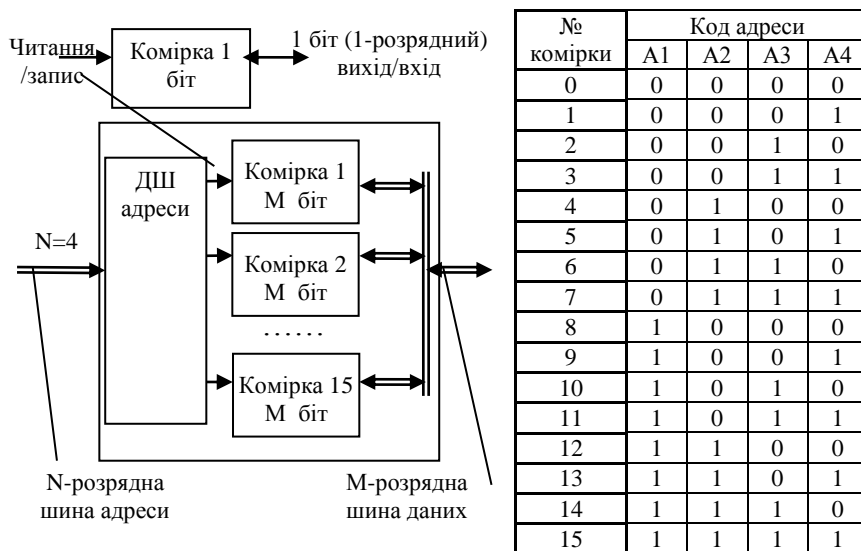


Рис. 13. Принцип доступу до комірок пам'яті

### 1.3.4 Сегментація пам'яті

Якщо процесор оперує  $N$ -розрядними адресами, а модуль пам'яті має  $Z$  комірок тобто його розрядність  $K = \log Z$ , тоді можливі 3 варіанти:

1. Місткість ОЗП рівна місткості адресного простору тобто  $N=K$ . Підключення пам'яті проводиться по схемі рис. 14.

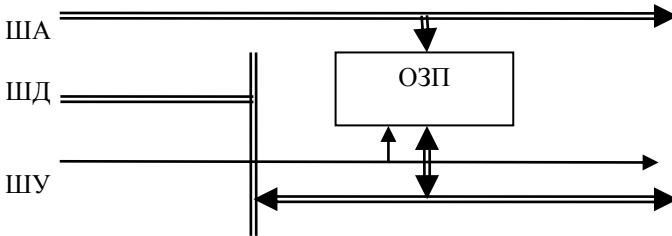


Рис. 14. Об'єм ОЗП дорівнює ємності адресного простору

2. Місткість одного модуля ОЗП менше місткості адресного простору ( $N > K$ ). В цьому випадку пам'ять складається з декількох модулів або сегментів. Для вибору комірки усередині модуля або сегменту використовуються  $K$  розрядів (зсув), а  $N-K$  розрядів адреси, що залишилися, використовуються через дешифратор для вибору модуля або сегменту (рис. 15).

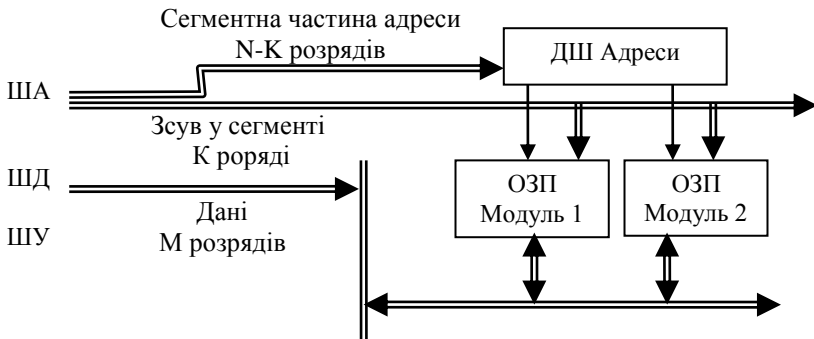


Рис. 15. Ємність одного модуля ОЗП менше ємності адресного простору

3. Місткість ОЗП більше місткості адресного простору. В цьому випадку пам'ять ділиться на сегменти по  $2^N$  комірок. Усередині сегменту адресація проводиться по  $N$ -розрядній шині, ця частина адреси називається зсувом, а

для адресації самих сегментів створюються додаткові розряди адресної шини (рис. 16).

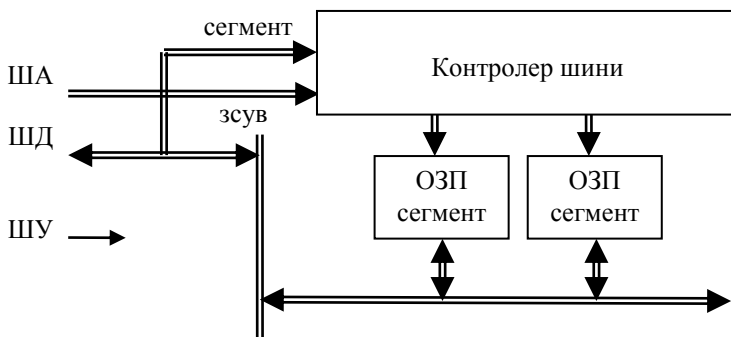


Рис. 16. Ємність ОЗП більша за ємність адресного простору

Наприклад, при  $N=16$  і необхідному об'ємі пам'яті 1 Мбайт ( $K=20$ ) в адресній шині не вистачає 4 розрядів. Реальна фізична 20-розрядна адреса реалізується за допомогою схеми рис.17. Маємо  $2^4=16$  сегментів з числом комірок у сегменті  $2^{16} = 65536$  (64 Кбайт).

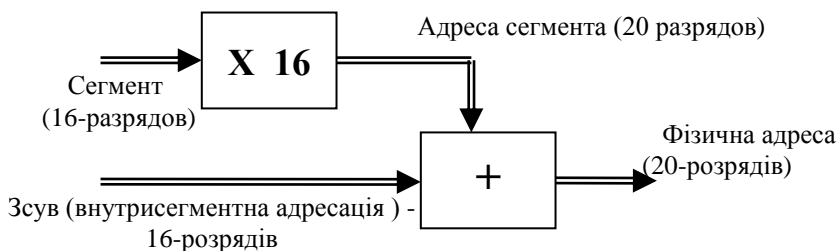


Рис. 17. Розрахунок 20-розрядної фізичної адреси

Реально частіше буває випадок – 3. Для вибору реальної комірки пам'яті треба вказати у якому сегменті вона знаходиться (адреса сегменту) і яке місце займає комірка у цьому сегменті (адреса зсуву). Кожна програма звично має свій сегмент пам'яті і не має доступу в іншій. Іноді для зручності програмування і оптимізації програм адресу даних обчислюють з урахуванням інших даних, наприклад, адреси базової і індексної.

### 1.3.5 Структура основної пам'яті

Загальний об'єм пам'яті може нарощуватися, при цьому її початкова частина розбивається стандартним чином на області, які мають своє призначення і користувачів.

Область пам'яті (адреса)	Тип	Назва	Розмір	Призначення
- 512 Мбайт і більш				Програми і дані ОС і користувача
- 64 Мбайт	Розширена пам'ять			
- 1088 Кбайт		Висока пам'ять		
- 1024 Кбайт (1 Мбайт)	Пам'ять, що безпосередньо адресується (1 Мбайт)	Верхня пам'ять 384 Кбайт	128 Кбайт ПЗП	Програми BIOS
			256 Кбайт ОЗУ	Службова пам'ять
- 640 Кбайт		Стандартна пам'ять 640 Кбайт	576 Кбайт ОЗУ	Програми і дані ОС і користувача
0 – 64 Кбайт			64 Кбайт ОЗУ	Службові програми і дані ОС

#### 1.4 Система введення-виведення

Уся інформація, яка поступає у комп'ютер або виходить за межі комп'ютера, проходить через систему пристроїв введення-виведення (ПВВ) або порти введення-виведення. Ця система включає пристрої для зв'язку, спостереження і управління зовнішніми (периферійними) пристроями, а також програмно-апаратні засоби забезпечення (адаптери, драйвери, контроллери і ін.). Периферійними пристроями є термінали, принтери, клавіатура, мишка, модеми, датчики і виконавчі пристрої, а також пам'ять великої місткості.

##### 1.4.1 Принцип роботи пристрою введення-виведення

Порти введення-виведення дозволяють встановити зв'язок між пристроями і програмним забезпеченням у комп'ютері. Використовуючи технологію прямого доступу до пам'яті, іноді інформація може передаватися безпосередньо між периферійними пристроями і пам'яттю, минувши процесор.

Звернення до пристроїв введення-виведення відбувається через систему інтерфейсних шин системної плати аналогічно зверненню до пам'яті. Наприклад, якщо ПВВ підключаються по 16-розрядній адресній шині, то всього може бути  $2^{16}=65536$  портів введення-виведення, які мають адреси від 0000h до FFFFh. Системна плата і набір мікросхем системної логіки використовують адреси портів від 0h до FFh, а всі інші від 100h до FFFFh.

Всі пристрої у комп'ютері використовують ці адреси, їх кількості достатньо для всіх випадків. При призначенні однієї адреси двом пристроям можуть виникати конфлікти. У системах, що підтримують технологію Plug and Play такі конфлікти вирішуються автоматично.

Порти використовуються для обміну даними і управління всіма пристроями, адаптерами і контроллерами системної платні, зовнішніх пристроїв і інтерфейсів.

Порти можуть бути паралельними і послідовними.

#### **1.4.2 Паралельний порт**

На рис. 18 зображена схема паралельного інтерфейса введення-виведення. Вибір ПВВ здійснюється подачею адреси по шині адреси (ША) і сигнала управління режимом (прийом або передача) по шині управління (ШУ). Після цього через шину даних ПВВ виконує обмін даними з процесором. Тимчасово дані зберігаються у регістрі RG. З іншого боку через логічну схему і блок СУ здійснюється обмін інформацією з відповідним зовнішнім пристроєм.

Логічна схема формує сигнали, які обумовлені протоколом конкретного типу порта введення-виведення. Схема узгодження рівнів сигналів (УС) змінює електричні параметри сигналів (напруга, струм і т.і.) якщо зовнішні пристрої користуються іншими стандартами. Наприклад, сигнали контроллера на системній платні звично мають рівні напруги 0+5 В, а рівні напруги сигналів в периферійному пристрої можуть бути -12+12 В.

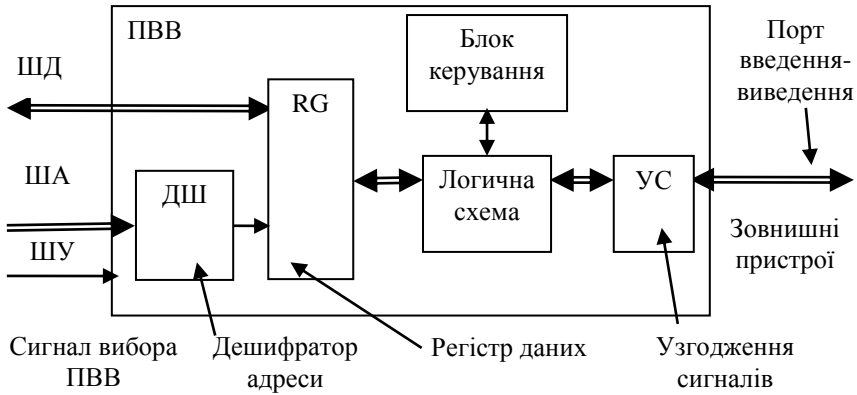


Рис. 18. Схема паралельного інтерфейса введення-виведення.

Розглянемо приклад деякого паралельного порту для зовнішнього пристрою (наприклад, принтера). Припустимо, що він має 8-розрядний вихідний сигнал (дані), 4 вихідних сигналів для управління зовнішнім пристроєм і 4 розряди вхідних сигналів для контролю стану зовнішнього пристрою. Структура проводів в такому інтерфейсі має вигляд:

№ проводу	Назва проводу/сигналу	Призначення
1	1 шина даних	Вихід
2	2 шина даних	Вихід
3	3 шина даних	Вихід
4	4 шина даних	Вихід
5	5 шина даних	Вихід
6	6 шина даних	Вихід
7	7 шина даних	Вихід
8	8 шина даних	Вихід
9	1 шина управління	Вихід
10	2 шина управління	Вихід
11	3 шина управління	Вихід
12	4 шина управління	Вихід
13	1 шина контролю	Вхід
14	2 шина контролю	Вхід
15	3 шина контролю	Вхід
16	4 шина контролю	Вхід

По шині даних здійснюється обмін даними з зовнішнім пристроєм. При цьому процесор керує його роботою по шині управління. Наприклад, у



принтері треба перемістити картридж, просунути папір, вибрати кольор і т.і. Виконання цих дій контролюється читанням інформації по шині контролю.

### 1.4.3 Послідовний порт

На рис. 19 представлена схема принципу реалізації послідовного порту. Процесор звертається до адаптера послідовного порту через шину розширення за його адресою і передає йому байт інформації. Одержаний байт записується у реєстр зсуву і послідовно біт за бітом видається у канал зв'язку. Після закінчення передачі цього байта виникає сигнал переривання і процесор видає наступний байт.

При прийомі з каналу зв'язку відбувається зворотний процес. Біти, що послідовно поступають з каналу зв'язку, збираються реєстром у байти і після сигналу переривання видаються у процесор. Прочитування байта процесором здійснюється також за адресою даного порту.

Цей адаптер може мати ще один або декілька адрес введення-виведення для передачі сигналів (даних) управління або прочитування даних про стан і режим роботи адаптера. Сигналами управління можуть бути завдання швидкості видачі біт у канал зв'язку, способу синхронізації, методу кодування і захисту від помилок і т.і.

Звідки процесор знає адресу даного пристрою? Ця інформація записана в спеціальних програмах- драйверах, які запускаються при обробці переривань цього пристрою. Через драйвер пристрій взаємодіє з операційною системою. Звичайно, драйвер розробляється підприємством – виготівником пристроїв і продається разом з ним.

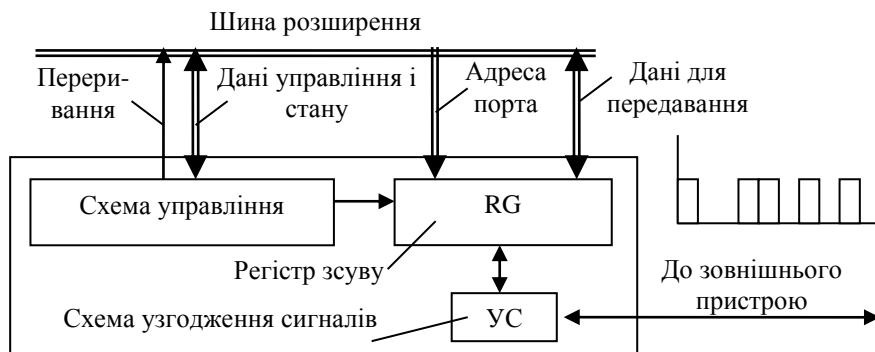


Рис. 19 Схема адаптера послідовного порту.

Дані про те, які адреси використовуються різними пристроями в даному комп'ютері можна подивитися в диспетчері пристроїв Windows. У Windows

2000/XP конфліктуючі пристрої мають позначку жовтою або червоною піктограмою.

Спеціалізовані порти введення-виведення застосовуються для підключення мишки і клавіатури. На рис. 20 представлений принцип підключення клавіатури.

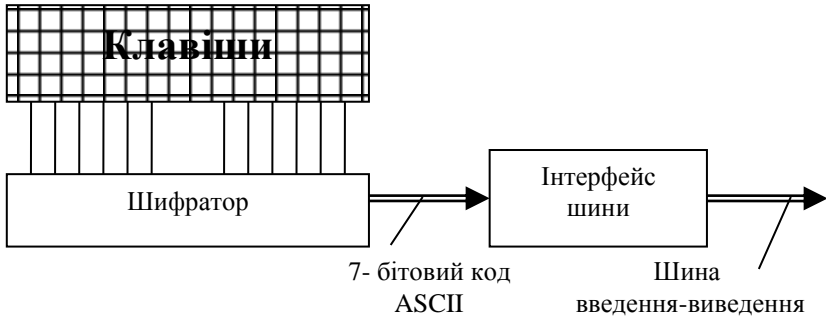


Рис. 20 Схема підключення клавіатури.

У деяких приладах існує виведення цифрових даних на спеціальну цифрову індикаторну панель. У цьому випадку використовується схема, яка зображена на рис. 21.

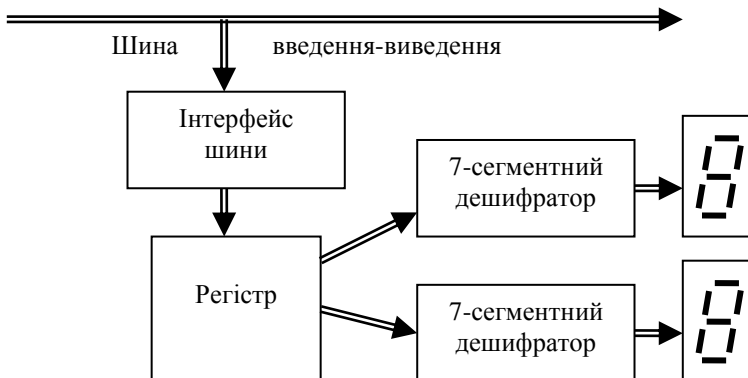


Рис. 21 Виведення даних на індикаторну панель

#### 1.4.4 Система переривань

Переривання слугують для того, щоб привернути увагу процесора до якогось пристрою або процесу. Система переривань дозволяє суміщати у часі роботу декількох програм у процесорі і зовнішніх пристроях, ефективно організувати обчислювальний процес і підвищувати оперативність обробки інформації, що поступає ззовні.

Система переривань це программо-апаратний комплекс, що забезпечує реакцію комп'ютера на зовнішні події. Реакція це виконання певних дій, а саме переривання поточної програми, виконання деякої програми обробки переривання і повернення на виконання перерваної поточної програми. При цьому, важливими є такі параметри як час реакції на подію і час обробки (обслуговування) переривання (т.е. події). Переривання можуть бути зовнішніми (або апаратними) і внутрішніми (апаратними або програмними).

Апаратні переривання поступають від пристроїв введення-виведення, програмні генеруються програмою, і теж використовуються для обслуговування операцій введення-виведення. Переривання розділені по пріоритетам, деякі з них мають переваги. Всі зовнішні переривання є маскованими, тобто відключаються на час виконання більш пріоритетної програми.

Кожен зовнішній пристрій підключається на один вхід контроллера апаратних переривань (канали переривання), але число входів завжди обмежено і на цей ресурс завжди багато претендентів. Крім того, кожне переривання вимагає пам'яті для розміщення відповідних програм обробки переривання (реагування). Тому розділення цього ресурсу між всіма апаратними засобами вимагає особливої уваги і іноді викликає конфлікти. Наприклад, коли деяким пристроям призначається одне і теж переривання. В цьому випадку, необхідно звільнити невживані канали переривань шляхом настройки у BIOS.

Як же працює система переривань. Коли який-небудь зовнішній пристрій готовий прийняти або передати дані він посилає сигнал на контроллер переривання (рис.22). Контроллер ідентифікує даний зовнішній пристрій і передає переривання процесору (по ШУ). Процесор припиняє роботу з поточною програмою, записує поточні дані у стек і запрошує від контроллера адресу підпрограми обробки даного переривання (по ШД). Ця підпрограма указує процесору, як треба діяти в даній ситуації. Наприклад, треба переслати дані з адаптера введення-виведення у процесор або пам'ять.

Після завершення роботи з підпрограмою процесор повертається до роботи з перерваною програмою і для цього прочитує із стека всі дані, на яких була перервана робота.

Переривання використовуються для завдання часових відміток для синхронізації і організації введення-виведення. Наприклад, переривання можуть поступати від системного годинника, співпроцесора, контроллера клавіатури, мишки, принтера і ін. Переривання дозволяють процесору своєчасно реагувати на зовнішні події. Наприклад, якщо по каналу зв'язку

поступив черговий байт або група байт інформації, їх необхідно прийняти і зберегти в пам'яті до надходження наступної порції.

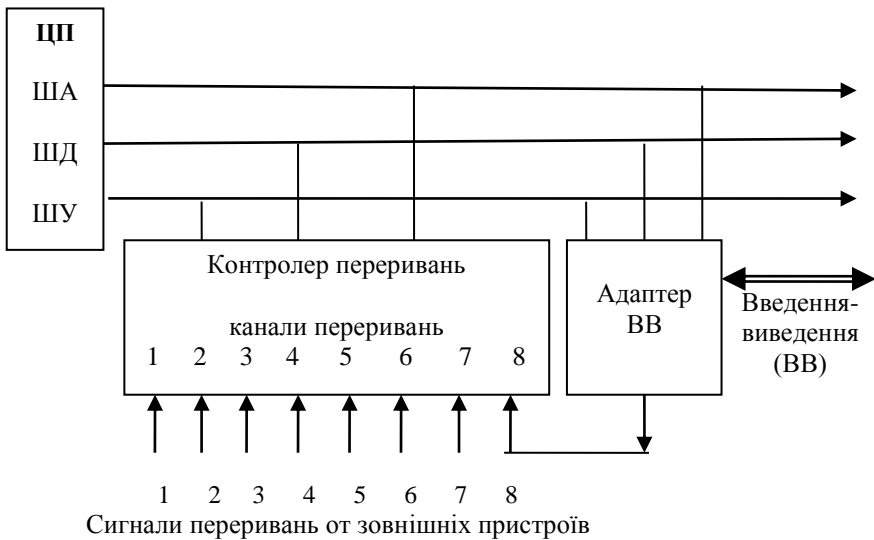


Рис. 22 Схема організації апаратних переривань у комп'ютері на прикладі операції введення-виведення.

#### 1.4.5 Канали прямого доступу до пам'яті

Канали прямого доступу до пам'яті (direct memory access – DMA) призначені для пристроїв, що здійснюють високошвидкісний обмін даними з пам'яттю. У комп'ютері є декілька (4 або 8) таких каналів, і до кожного можуть бути підключені декілька пристроїв. Одночасно мають доступ до пам'яті тільки пристрої, підключені до різних каналів. Наприклад, прямий доступ до пам'яті надається при регенерації динамічної пам'яті, контроллерам гнучких і жорстких магнітних дисків, звуковій платі, зовнішнім портам.

Процес надання прямого доступу до пам'яті відбувається таким чином. При запиті, будь-яким пристроєм, прямого доступу до пам'яті, відповідний контролер приймає цей запит і, якщо доступ дозволений, видає всій решті пристроїв команду відключитися від пам'яті. Пристрій, який запитав доступ, здійснює передавання або прийом інформації. Після закінчення сеансу обміну даними з пам'яттю вирішується доступ іншим пристроям.

Як бачимо, кількість пристроїв, що потребують доступу до пам'яті, значна, а кількість каналів DMA обмежена, тому у комп'ютері постійно здійснюється перерозподіл даних ресурсів. Інколи виникають конфлікти між пристроями. Конфлікт – це коли двом пристроям призначений один і той же канал прямого доступу до пам'яті (DMA).

### 1.4.6 Відеоадаптер

Основна задача відеоадаптера це формування сигналів управління монітором. Він міститься у спеціалізованих схемах на системній платі або на окремій відеокарті (платі). Основними параметрами відеоадаптера, схема якого зображена на рис. 23, є розподільна здатність, глибина кольору і стандарт, що підтримується.

Відеоадаптер має свою власну BIOS, яка містить основні команди його інтерфейсу з програмним забезпеченням. Вона дозволяє вивести інформацію на монітор під час загрузки та процедури POST.

Графічний процесор здійснює графічну обробку відеоданих, які зберігаються у відеопам'яті.

Відеопам'ять на відеоадаптері своя власна, но може використовуватися і частина оперативної пам'яті. Вона зберігає зображення плоских та трьохмерних структур. Збільшення її об'єму дозволяє збільшити швидкість зміни зображення. Сучасний об'єм відеопам'яті досягає 256 Мбайт.

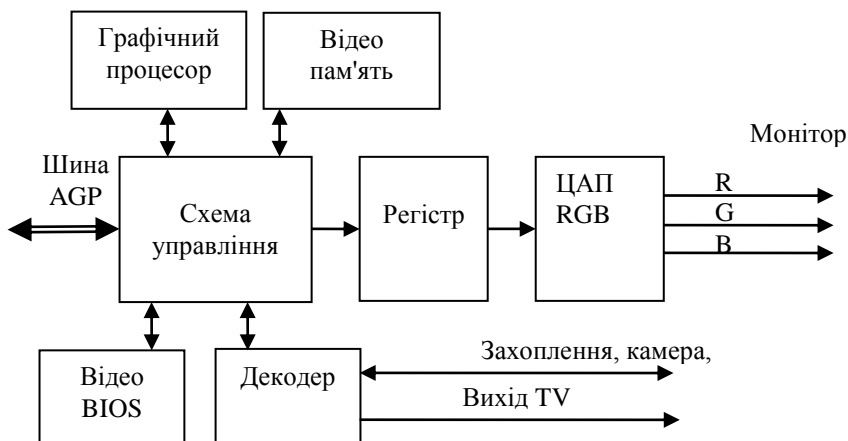


Рис. 23 Типова структура відеоадаптера

Відеоадаптер з'єднується з монітором через 15-контактні аналогові роз'єми стандарту VGA або через аналого-цифрові роз'єми DVI, які теж відповідають стандарту VGA. Існує кілька модифікацій цього стандарту під назвою SVGA, програмний інтерфейс яких відповідає стандарту VESA BIOS Extension, що дозволяє використовувати єдиний драйвер. Цій стандарт передбачає використання усіх форматів зображення (максимум 1280x1024 пікселів) та 16 777 216 кольорів (24-біт). Відповідний кольоровий сигнал формується у цифро-аналогових перетворювачах.

З системною платою відеоадаптер з'єднується через шину PCI або AGP, яка пропонує високошвидкісний доступ до пам'яті з пропускнуою здатністю у 16 разів більшу ніж перша. Це дозволяє збільшити швидкість виводу трьохмерних зображень. Існує 4 швидкості AGP 1x, 2x, 3x, і 4x. Остання має частоту 533 МГц і швидкість передачі даних 2,132 Гбайт/с. Більш сучасні відеоадаптери з'єднується через послідовну шину PCI Express. Її максимальна швидкість 16x відповідає 4 Гбайт/с.

Зв'язок відеоадаптера з програмним забезпеченням виконує відеодрайвер, який також дозволяє налаштовувати параметри відеосистеми.

Графічний акселератор зменшує навантаження на центральний процесор і збільшує швидкість виводу трьохмерної графіки. ЦП видає тільки абстрактне зображення, з якого реальне створює акселератор. Він також створює анімаційну послідовність кадрів та їх кольор між ключевими позиціями, які видає центральний процесор.

Інколи відеоадаптери підтримують роботу з двома моніторами і функцію TV-out, яка дозволяє підключати телевізор. Інколи є відеовхід для перетворення аналогового сигналу з відеомагнітофону або аналогової камери у цифровий сигнал і записоти його у пам'ять (відеозахват).

#### **1.4.7 Звуковий пристрій**

Звуковий пристрій призначений для перетворення цифрових сигналів, у яких представлений звук, в аналоговий сигнал, який може сприймати людське вухо. Також здійснюється зворотний процес перетворення аналогового звуку у цифрову форму для збереження і обробки цифровим шляхом.

Звуковий пристрій один з найбільш складних пристроїв комп'ютера. Він може бути розташований на окремій платі (звукова плата або карта), інтегрований у системну плату у вигляді окремої мікросхеми або у складі мікросхем системної логіки. Звукові пристрої використовують декілька системних ресурсів: хоч би одна лінія запитів переривання, два канали прямого доступу до пам'яті, декілька портів введення-виведення (рис.24). Карти для шини PCI канали прямого доступу до пам'яті не використовують.

Більшість пристроїв має якість звуку, яка відповідає компакт-дискам з частотою дискретизації 44,1 кГц. Для обробки звуку використовується власний процесор, який здійснює стиснення звуку, очищення його від шуму, накладення ефектів, синтез трьохмірного звуку, перетворення тексту у синтезовану мову, синтез багатьох звуків и їх синхронізацію з зображенням. Звуковий процесор знижує навантаження на центральний процесор і підвищує загальну швидкість комп'ютерної системи.

Для зв'язку звукового адаптера с програмним забезпеченням використовується програмний драйвер.

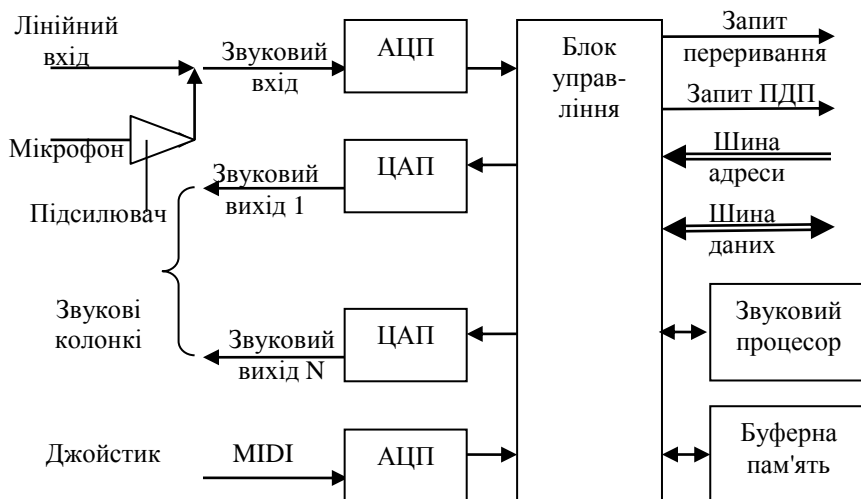


Рис. 24 Структурна схема звукової карти

## 1.5 Системна плата

### 1.5.1 Структура системної плати

Системна плата (system board, mother board) це головна плата комп'ютера. На системній платі розташовані основні взаємодіючі між собою компоненти і функціональні вузли комп'ютера. На сучасній системній платі встановлені різні комбінації наступних компонентів:

- Центральний процесор і співпроцесор
- Набір мікросхем системної логіки (північний, південний міст)
- Мікросхема Super I/O
- Тактовий генератор і таймер
- Мікросхема базової системи введення-виведення (BIOS)
- Модулі основної пам'яті, кеш-пам'ять, CMOS - пам'ять.
- Системні шини і шини розширення, локальні шини
- Відеоадаптер
- Звуковий адаптер (Sound Blaster)
- Мережевий адаптер
- Інтерфейсні роз'єми

Системні плати відрізняються типом процесора, набором системних мікросхем, тактовою частотою, типом BIOS, типом і об'ємом оперативної пам'яті і іншими параметрами.

У негонебезпечній CMOS – пам'яті зберігається поточний час, дата і конфігурація всього комп'ютера, які прочитуються при його завантаженні.

### 1.5.2 Конструкція ATX

В даний час найбільшого поширення набули комп'ютери конструкції ATX, яка розроблена в 1995 році фірмою Intel. Типова схема розташування вузлів на системній платі і комп'ютері системи ATX показана на рис.25. Процесор розташований близько від вентилятора блоку живлення і має свій вентилятор (кулер), що покращує його тепловий режим. Зовнішні роз'єми розташовані на системній платі. Модулі розширення перпендикулярні до системної платі.

Існує цілий ряд модифікованих плат сімейства ATX. Їх специфікації і технічні характеристики, а також огляд нових технологій можна подивитися на сайті <http://www.formfactors.org>.

З 2003 року існує нова специфікація системної платі ВТХ, яка має кращі умови для охолодження і дозволяє розмістити більшу кількість сучасних компонентів.

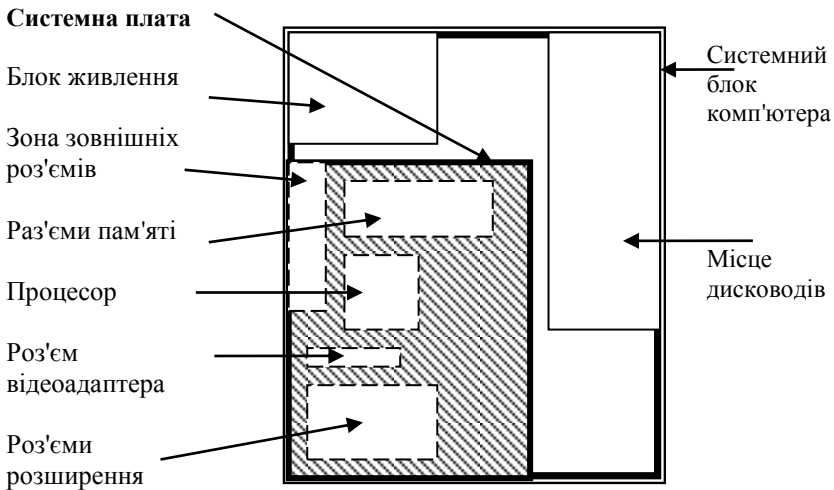


Рис. 25 Типова схема розташування вузлів на системній платі і комп'ютері ATX

### 1.5.3 Системна логіка

Набори мікросхем системної логіки (чіпсети) забезпечують нормальну роботу процесора і його зв'язок з рештою вузлів комп'ютера. Вони беруть на себе допоміжні управлінські функції, знижують навантаження і підвищують ефективну роботу процесора. Сучасні системні набори складаються з двох



базових і додаткових мікросхем системної логіки. З них будується багаторівнева архітектура комп'ютера.

Перша базова мікросхема це високопродуктивний контроллер, який відповідає за зв'язок швидкодіючої системної шини процесора з повільнішими але високопродуктивними пристроями (пам'ять, локальні інтерфейси, відеоадаптер). У різних виробників ці мікросхеми мають різні назви, але виконують схожі функції. Наприклад, їх називають північний міст (North bridge) або контроллер пам'яті (Memory Control Hub - MCH) або контроллер графіки і пам'яті. Північний міст забезпечує шлях для сполучення швидкодіючої шини процесора, що працює з частотою 66/100/133/200/266/400 МГц, з шиною оперативної пам'яті, шинами PCI (частоти 33/66 МГц), AGP (133/266/533 МГц).

Друга базова мікросхема відповідає за обслуговування пристроїв з низькою швидкістю. До цих мікросхем відносяться південний міст (South bridge), контроллер введення виведення (I/O Control Hub – ICH). Цій функціональний контроллер забезпечує міст між першою базовою мікросхемою (або шиною PCI) і пристроями низької швидкості або шиною ISA (8 МГц).

Мікросхеми системної логіки реалізує інтерфейс контроллера жорсткого диска IDE, інтерфейс USB, годинник і пам'ять CMOS, контроллери прямого доступу до пам'яті і контроллер переривань, контроллери інтерфейсів IDE/ATA, SCSI, USB, SATA, SAS, IEEE 1284, PCI Express.

В архітектурі северний-южный міст системні мікросхеми здійснюють перетворення системної шини в шину PCI, а потім PCI в шину ISA. Зв'язок між північним і південним мостом по шині PCI. Проте з часом продуктивність шини PCI перестала задовольняти збільшені потреби. У Наборі архітектурі для зв'язку цих контроллерів фірмою Intel розроблений спеціальний швидкісний Хаб-інтерфейс, що значно підвищує швидкість. А інтерфейс DMI (direct media interface) підвищує швидкість передачі ще в чотири рази. Існують також швидкісні інтерфейси інших виробників, такі як HyperTransport (16 біт, 1 ГГц), A-Link, V-Link.

Можуть бути ще додаткові мікросхеми для пристроїв введення-виведення. Наприклад, мікросхема Super I/O, контроллер IEEE-1394, контроллер SCSI, звукові, відео і мережеві адаптери. Дуже часто ці додаткові функції інтегровані в базові мікросхеми.

Мікросхема Super I/O підтримує паралельний порт, два послідовні порти, контроллер гнучких дисків, інтерфейс клавіатури, миші і інтерфейс ігрового порту. У неї може знаходитися пам'ять CMOS, годинник. У сучасних наборах мікросхема Super I/O включена в мікросхему південного моста.

Існує безліч різних варіантів системної логіки і відповідно структурних схем комп'ютера. Набори відрізняються типом підтримуваного процесора, пам'яті, частотою шини, протоколом, технологією і ін. Нижче розглянуті найбільш популярні і типові функції і технології. На рис.26 приведені

узагальнені типові схеми, що ілюструють принципи застосування системної логіки і способи побудови схеми комп'ютера.

Функції системних наборів значно знижують завантаження процесора і підвищують швидкодію комп'ютера.

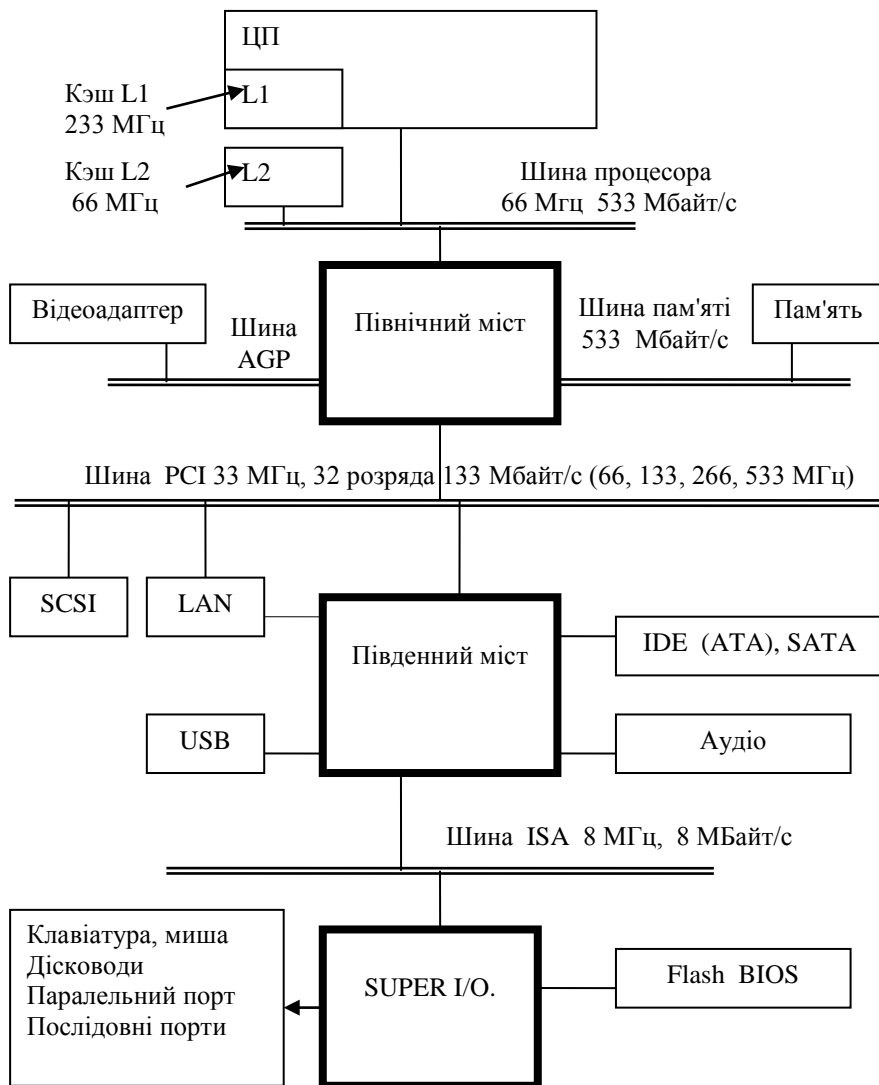


Рис. 26 Архітектура Північний / Південний міст

Розвитком архітектури Північний / Південний міст є HUB –архітектура (рис.27). У ній Південний міст приєднується до північного не через шину PCI, а через Hub-інтерфейс, який в два рази швидкісніший чим PCI, а також

надійніший і економічніший за рахунок зниження числа сигналів. При цьому за рахунок розвантаження шини PCI підвищується швидкість обслуговування приєднаних до неї пристроїв. А Південний міст також дістає можливість обслуговувати швидкісні інтерфейси ATA – 100 і USB 2.0 та інші.

У Південному мосту є шина LPC (Low Pin Count) – це 4 розрядна версія шини PCI. Вона служить для підтримки мікросхем ROM BIOS і Super I/O. По параметрам вона аналогічна шині ISA, але економічніша і дешевша. Застаріла шина ISA в даній архітектурі відсутній.

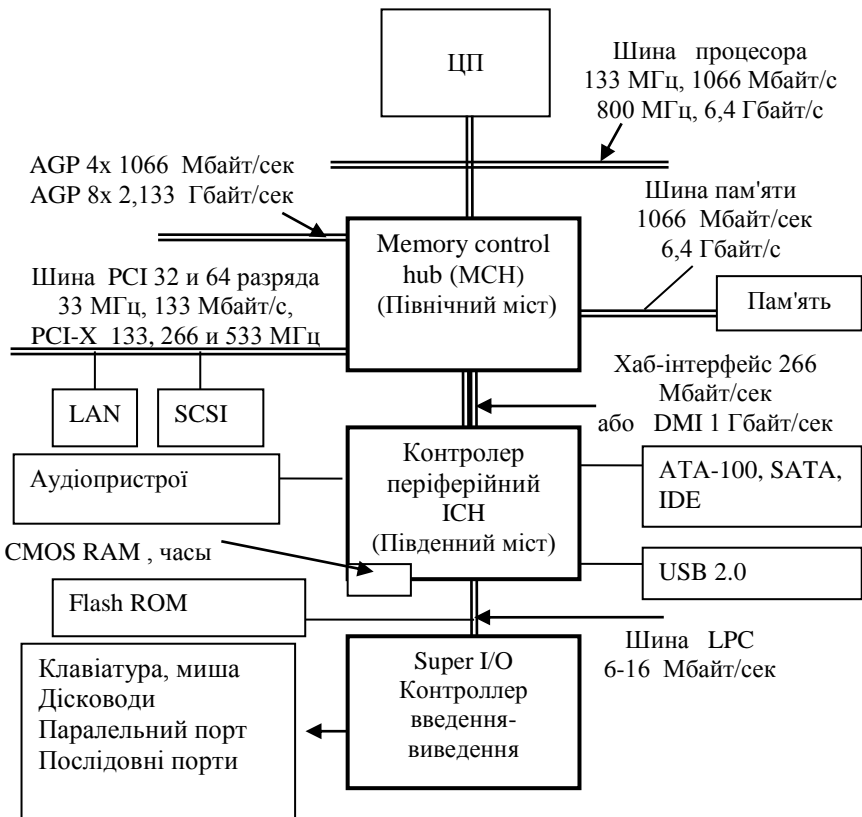


Рис. 27 Hub –архітектура (Pentium 3,4)

У наборі Intel 810 північному мосту відповідає контроллер GMCH (Graphics Memory Controller Hub) з інтегрованим графічним адаптером, модемом і аудіоадаптером. Тому в цьому наборі відсутній шина AGP. А в наборі Intel 815G при інтегрованій графіці вже з'являється інтерфейс AGP 4x

для розширення графічних можливостей. Набір має інтегрований контроллер локальної мережі (LAN).

Південний міст дуже часто включає функції схеми Super I/O. Такі мікросхеми одержали назву Super South Bridge.

Розвиток технологій пред'являє все більш високі вимоги до швидкості передачі і обробки даних. Тому подальший розвиток системної логіки йде по шляху підвищення розрядності і швидкодії обслуговуваних пристроїв. Швидко упроваджуються швидкісні послідовні інтерфейси. Наприклад, з'явилася підтримка пам'яті DDR-II і шина PCI-Express для підключення відеоадаптерів і високошвидкісних мережевих адаптерів (набір Intel 9xx).

Всі пристрої, які не розташовані на системній платі приєднуються до неї через роз'єми. Приклад розташування всіх роз'ємів на системній платі показаний на рис.28.

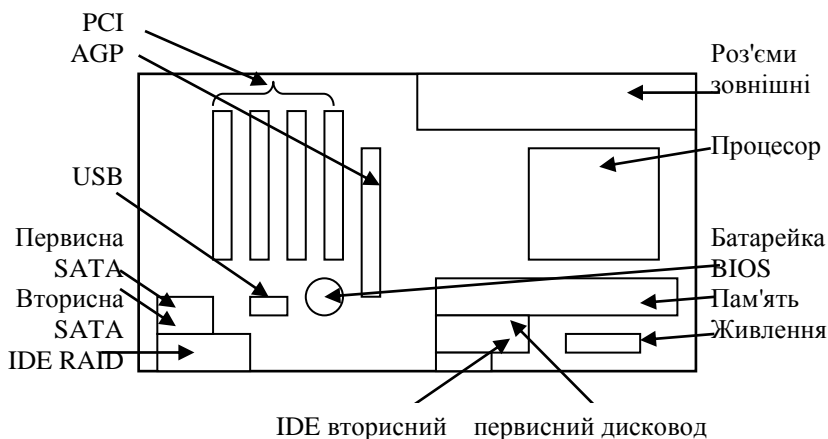


Рис. 28 Один з варіантів розташування раз'ємів на системній платі

### 1.5.4 Шинні інтерфейси

У комп'ютері є ціла система шинних інтерфейсів, які слугують для передачі сигналів між компонентами системи. Існує певна ієрархія шин: кожна повільніша шина пов'язана з процесором через більш швидку шину. Системна логіка виконує роль моста між шинами різних ієрархічних рівнів. Всі вони виходять на системну шину і відповідні схеми управління. Кожен системний пристрій з'єднаний з процесором через відповідну шину.

Є шини розширення для підключення різних пристроїв, локальні шини для обслуговування швидкісних пристроїв, шини для обслуговування периферійних пристроїв.

Більшість шин, як правило, паралельні і складаються з трьох шин: ША – шина адреси, ШД – шина даних, ШУ – шина управління. Кожна шина це набір проводів, до яких одночасно підключені багато пристроїв, активізація кожного пристрою здійснюється по команді процесора або контролера прямого доступу до пам'яті подачею сигналів управління (по ШУ) і відповідної адреси (по ША). Активізовані пристрої починають передавати або приймати дані (по ШД) поки сигнали активізації не будуть зняті. Шинная організація дозволяє значно спростити апаратні засоби взаємодії.

Шина характеризується розрядністю, тактовою частотою і швидкістю передачі даних. Розрядність буває 8, 16, 32, 64 розряди при частоті 8-533 МГц і більш. Швидкість передачі даних визначається добутком розрядності і тактової частоти і досягає в сучасних комп'ютерах величини в декілька мільярдів біт в секунду.

Нарис.29 показаний приклад схеми 8-розрядної шини даних, яка сполучає деякий контроллер з трьома пристроями введення-виведення (ПВВ).

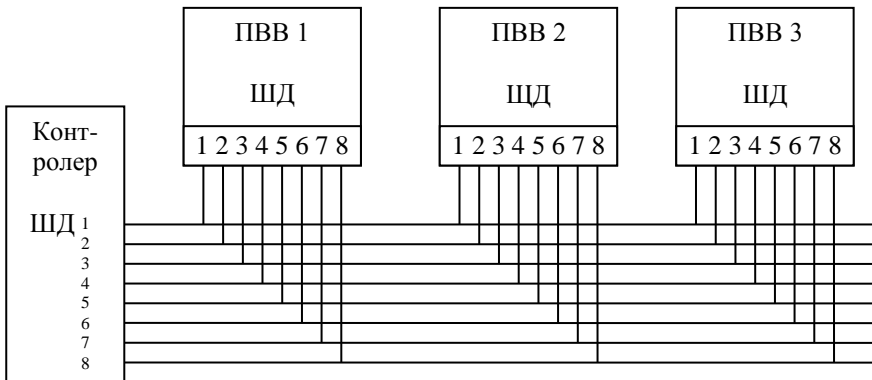


Рис. 29. Схема паралельної 8-розрядної шини даних (ШД)

Условні зображення трьох шин (даних, адреси і управління), що сполучають контроллер і ПВВ, показані на рис. 30. Вони для спрощення зображені однією потовщеною або подвійною лінією.

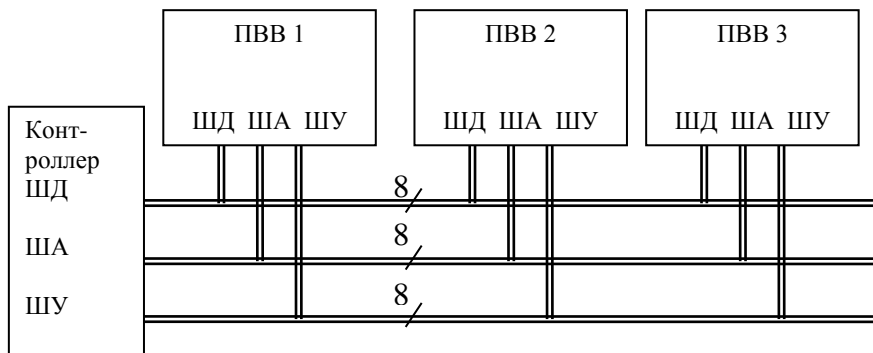


Рис. 30. Умовне позначення схеми паралельної 8-розрядної шини

У комп'ютері є наступні ієрархії шинних інтерфейсів:

- Процесорна або системна шина – сама високошвидкісна шина в комп'ютері, забезпечує обмін інформацією між процесором, північним мостом, контролером пам'яті, зовнішнім кешем і іншими пристроями. Працює на частоті до 800 МГц і має ширину до 64 розрядів.

- Відеошина AGP працює на частоті до 533 МГц і розрядності до 32 біт. Призначена для підключення відеоадаптера. Підключається до північного моста системної логіки.

- Локальні шини – обслуговують невелику групу однотипних високошвидкісних пристроїв в обхід шини розширення.

- Шина розширення – універсальна шина для підключення різних додаткових (низькошвидкісних) пристроїв в комп'ютері.

- Периферійні шини – для обслуговування зовнішніх пристроїв.

Найвідоміші шини розширення ISA (16 розрядна), EISA (32 розрядна), а також MCA вже застаріли, оскільки мають малу розрядність і частоту. Зараз як шину розширення використовують локальну шину PCI.

Сучасна універсальна комп'ютерна шина PCI (Peripheral Component Interconnect), розроблена в 1992 році. Перша версія шини PCI 1.0:

- Допускає підключення до 10 пристроїв.

- Має адаптер, що дозволяє налаштуватися для роботи з будь-яким процесором.

- Тактова частота 33 МГц.

- Розрядність даних – 32, розрядність адреси – 32 з можливістю розширення до 64.

Версія PCI 2.1 дозволяє працювати з частотою 66 МГц при розрядності 64 з підтримкою режимів Plug&Play, Bus Mastering і автоконфігурації адаптерів.

Використовується для підключення Відеокарти, Звукової карти, Модему, Контролера SCSI і інших. Ця шина може працювати одночасно з шиною процесора.

Є версії з частотою 133 МГц (PCI 2.2), а також з подвоєною 266 МГц (PCI DDR) і почотвереною частотою 533 МГц (PCI QRD).

На базі шини PCI 2.1 розроблена локальна шина AGP (Accelerated Graphics Port – прискорений графічний порт) для підключення відеоадаптера. Завдяки прямому виходу на системну пам'ять ця шина має високу швидкість передачі графічних даних. У цій шині реалізовані конвейєрні операції читання-запису, що дозволяє нейтралізувати вплив затримок в модулях пам'яті. Унаслідок незалежній роботі шини AGP від PCI, остання розвантажується і підвищується її швидкодія.

Існують версії 1x, 2x (1-кратна і 2-кратна швидкість з напругою сигналів 3,3В), 4x, 8x (4-х і 8-кратна швидкість з напругою 1,5 В). Із-за відмінності в напрузі сигналів вони частково відрізняються роз'ємами, щоб не можна було вставити плату в роз'єм з невідповідною напругою.

Розрядність шини 32 бита, тактова частота 66 МГц. Кратність швидкості показує, скільки біт даних передається за один такт. Так сама швидкодіюча версія AGP 8x забезпечує швидкість передачі даних 2,133 Гбайт/с.

### 1.5.5 Послідовні Інтерфейси

На початку століття з розвитком напівпровідникової техніки і підвищенням її швидкодії відбувається перехід на послідовні інтерфейси, які в порівнянні з паралельними мають суттєві переваги. У даний час для всіх існуючих паралельних інтерфейсів розроблені послідовні аналоги.

Основні переваги послідовного інтерфейсу:

1. Зручність, гнучкість, функціональність.
2. Висока пропускна спроможність.
3. Відсутність складної синхронізації
4. Зниження вартості і складності монтажу.
5. Підвищення надійності і захисту від перешкод.
6. Динамічна конфігурація і масштабування системи.
7. Полегшення арбітражу і переривань.

Послідовний інтерфейс SATA працює по восьмижильному кабелю при пропускній спроможності 1500 – 3000 – 6000 Мбит/сек. Замінює паралельний інтерфейс АТА (IDE), який має плоский широкий 80-ти проводний кабель (шлейф). Кращий захист від перешкод забезпечується за рахунок методів виявлення і виправлення помилок. Використовується для підключення дискових накопичувачів.

З 2001 року розроблений новий послідовний інтерфейс PCI Express, який забезпечує:

- Сумісність з шиною PCI і всіма драйверами пристроїв.
- Просту зміни пропускної спроможності зміною частоти і розрядності.



- Максимальну пропускну спроможність і низький час очікування.
- Спрощення конструкції плати, управління режимом живлення і гарячу комутацію, тобто без виключення живлення.

Тактова частота шини PCI Express 2,5 ГГц. Використовується кодування типу 8-10 з автосинхронізацією. Кожні 8 біт кодуються 10 бітами, що підвищує захист від перешкод. За один такт передається 1, 2, 4, 8, 16 або 32 біт даних по відповідному числу пропускних смуг (послідовних каналів). Кожна смуга це дві пари проводів для повнодуплексної передачі, тобто одночасного прийому і передачі. Швидкість передачі в одній смузі 250 Мбайт/с, в 32 смугах – 8 Гбайт/с.

Шина PCI Express 16x більш швидкодіюча чим звичайна PCI і AGP 8x, її можна використовувати для передачі відеоданих замість шини AGP 8x і як шина розширення до інших інтерфейсів SATA, USB і т.д.

Підключення до послідовного інтерфейсу здійснюється за схемою, яка зображена на рис.31. Загальна схема системи з шиною PCI зображена на рис.32.

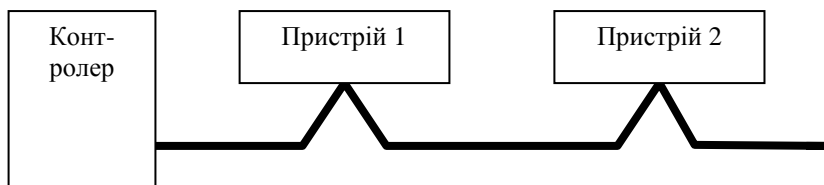


Рис. 31 Схема підключення пристроїв через послідовний інтерфейс

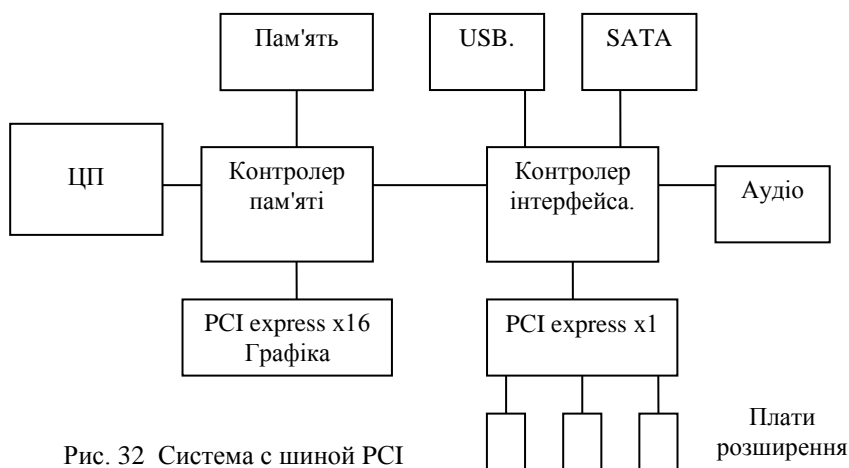


Рис. 32 Система с шиною PCI

Серед периферійних інтерфейсів найпопулярніша послідовна шина –USB (Universal Serial Bus). Вона дозволяє проводити гарячу заміну пристроїв без виключення і перезавантаження комп'ютера. Шина замінила застарілі інтерфейси RS-232, COM, IEEE-1284 (LPT). Шина дозволяє одночасно підключати до неї до 128 пристроїв, але для цього потрібні спеціальні розгалужуючі пристрої – HUB з одним входом і декількома виходами (рис.33).

Інтерфейс IEEE 1394 (FireWire) (iLink). Характеризується високою надійністю і якістю. Реалізований восьмижильним кабелем він має пропускну спроможність 400-1600 Мбит/сек. До контроллера допускається підключення до 63 пристроїв по технології Plug&Play у будь-якій конфігурації. Використовується для підключення високошвидкісних пристроїв зовнішньої пам'яті, відеокамер і ін.

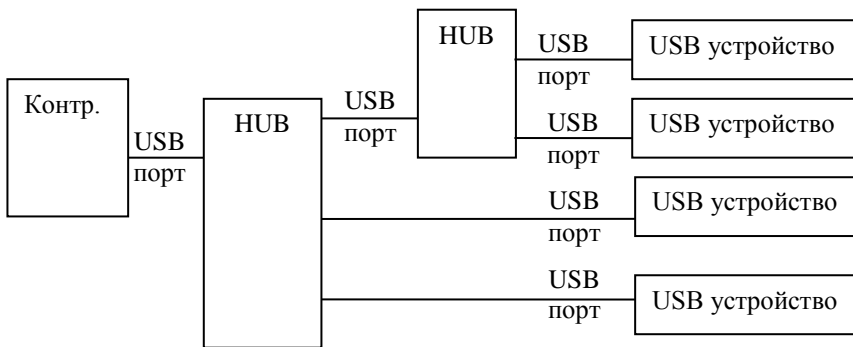


Рис. 33 Пример разветвленной структуры USB портов

Існують так звані бездротові інтерфейси, які побудовані за принципом послідовних і не вимагають кабелю для свого з'єднання (рис. 34).

На базі радіоканалу створений інтерфейс Bluetooth, він працює на частотах в діапазоні 2,4 – 2,485 ГГц. Дальність дії пристрою визначає його клас до 100 м, до 10 м, до 1 м. Більшість пристроїв мають 2-й клас. Швидкість передачі версії 1.2 до 1,0 Мбит/сек, для версії 2.0 до 3Мбит/сек. Інтерфейс застосовується в локальних мережах, а також для бездротових пристроїв введення і виведення даних. До кожного пристрою підключаються до 7 пристроїв, утворюючи мережу. Несуча частота автоматично міняється для забезпечення безпеки.

За принципом оптичного каналу створений інтерфейс IRDA. Він працює в інфрачервоному діапазоні на довжині хвилі 880 нм, дальність до 1 м, швидкість передачі від 115 Кбит/сек до 1152 Кбит/сек. або 16 Мбит/сек.

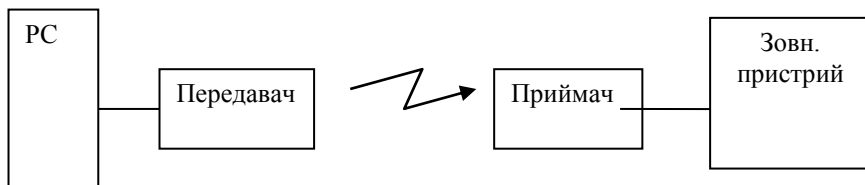


Рис. 34 Бездротовий канал зв'язку.

### 1.5.6 Загальні відомості про профілі устаткування

Кожен пристрій, який є у комп'ютері, відповідним чином реєструється і для нього встановлюється драйвер, який відповідає за його зв'язок з програмними засобами. Процесор має знати які у нього є пристрої і як з ними з'єднатися. Сукупність пристроїв, їх параметрів, інструкцій і драйверів у комп'ютері це профіль устаткування.

Профіль устаткування, який використовується Windows завантажується при запуску комп'ютера. Для кожного режиму роботи комп'ютера використовується свій профіль. При встановленні Windows створюється профіль устаткування «Profile 1», у який, за умовчанням, занесени всі пристрої, які присутні на комп'ютері на момент установки.

Профілі устаткування особливо зручні при роботі з переносними комп'ютерами. Звично такі комп'ютери використовуються в різних місцях, і профілі устаткування дозволяють змінювати перелік пристроїв, вживаних в конкретній ситуації. Наприклад, у профіль, призначений для підключення переносного комп'ютера, входять такі компоненти, як пристрій читання компакт-дисків і мережевий адаптер. Другий профіль слугує для використання переносного комп'ютера в готелі або літаку, коли замість названих пристроїв використовуються модем і переносний принтер.

Якщо в системі є декілька профілів устаткування, можна вказати серед них той, який використовуватиметься за умовчанням при кожному запуску комп'ютера. Windows дозволяє також відображати при запуску питання, який профіль слід використовувати. Після створення профілю устаткування пристрої, що входять в нього, можна відключати і включати з допомогою диспетчера пристроїв. При відключенні пристрою в профілі устаткування відповідні драйвери не завантажуються при запуску комп'ютера.

## МОДУЛЬ 2

### Програмне управління пристроями

Принцип програмного управління є фундаментальним в концепції побудови універсальних комп'ютерних систем. Він дозволяє відносно дешевими засобами значно розширити круг вирішуваних задач і легко перемикається з рішення однієї задачі на іншу.

Програми можуть бути написані на різних алгоритмічних мовах, з якими людині працювати зручніше. Але зрештою усі вони представляються у вигляді послідовності машинних команд, з якими тільки і працює процесор. Для перекладу одного в інше служать спеціальні програми-транслятори.

#### 2.1 Поняття про машинну команду

##### 2.1.1 Представлення інформації

Вся інформація у комп'ютері (дані і програми) представляється у вигляді двійкових чисел. Мінімальна одиниця інформації – 1 біт, яка представляється одним розрядом двійкового коду. Мінімальна одиниця інформації, що адресується, – слово, яке складається з декількох байт (байт це 8 біт). Інформація в комп'ютері представляється різними способами, які залежать від виду інформації, способу кодування, діапазону чисел і їх точності, способу представлення коми, форми негативних чисел.

Для кодування алфавітно-цифрових символів застосовуються спеціальні коди, найбільш поширений з них – ASCII. Кожен символ представляється одним байтом.

При передачі інформації для підвищення перешкодостійкості застосовують різні види кодування. Для дуже високої перешкодостійкості, яка необхідна в складних і довгих каналах зв'язку, застосовують складні коди, які значно збільшують об'єм передаваної інформації.

При передачі усередині комп'ютера або по якісним каналам зв'язку, для виявлення помилок застосовують контроль з парності. Для цього до слів або байтів додають один або декілька біт, щоб загальне число одиниць у всій комбінації або її частинах стало парним. При виникненні помилок парність порушується і це слово або блок інформації повторюється ще раз.

Мінімальна функціональна одиниця для представлення програм це машинна команда. Машинна команда містить дані або їх адреси, а також інструкції по обробці цих даних.

Виконання команди складається з послідовності елементарних операцій з даними і командами. Виконання цих елементарних операцій проводиться процесором за один машинний цикл, який реалізується в декілька тактів (етапів). Виконання кожного такту синхронізується спеціальними тактовими сигналами, які генеруються тактовим генератором. Наприклад, деякі команди можуть виконуватися у вигляді послідовності наступних етапів:

1. Передача адреси команди з процесора в пам'ять програм.
2. Прочитування з пам'яті і дешифрування команди в процесорі.
3. Виконання команди в процесорі.
4. Передача результату виконання команди в пам'ять.
5. Занесення в лічильник команд адреси чергової команди.

### 2.1.2 Структура машинної команди

Машинні команди це функції, які безпосередньо сприймаються і виконуються апаратними засобами комп'ютера. Кожен тип процесора, залежно від його структури і функціональних можливостей має свій набір машинних команд, які він може виконати. Система команд може бути повна (CISC – Complex Instruction Set Command), усічена (RISC – Reduced ) і мінімальна (MISC – Minimum ).

Машинні команди це послідовність двійкових цифр (код), записаних в пам'яті комп'ютера. Процесор здійснює вибірку машинних команд з пам'яті і їх виконання. Команди можуть бути однокбайтними і многобайтними, а останні можуть бути одно-, двух- і триадресними.

Команда містить операційну частину і операнд (дані чи адреса).. Операційна - це код операції, він визначає операцію, яку необхідно виконати. Після неї пристрій управління процесора визначає, чи є у цієї команди адресні байти і який алгоритм дії всіх блоків процесора. Кількість різних команд, які закодовані в операційній частині з  $N$  біт, дорівнює  $2^N$ . Наприклад, якщо операційна частина процесора складається з одного байта, то система команд такого процесора може мати 256 різних команд.

Операнд – це дані для обробки або їх адреса. Адресна частина команди містить адреси операндів і адрес результату виконуваних дій. Крім того, в команді міститься інформація які команди виконуються після даної, т.е чи зберігається природна послідовність або порушується.

Наприклад, розглянемо структуру машинної команди для деякого процесора з 8 розрядною шиною даних і 16 розрядною шиною адреси:

Однокбайтна команда має, наприклад, такий вигляд:

Код операції
10100010

П'ятибайтна команда:

Операційна частина	Адресна частина команди			
Код операції	Адреса 1 операнда		Адреса 2 операнда	
10100011	01011101	00110011	11110000	00000011

Припустимо, що ця команда є підсумовуванням двох операндів, тоді при її виконанні можливий наступний алгоритм:

1. Прочитування чергової команди з комірки пам'яті, адреса якої знаходиться у регістрі IP.
2. Дешифрування команди, після чого з ПЗП мікропрограм видається послідовність сигналів, що визначають послідовність наступних дій.
3. Збільшення на 1 вмісту регістра IP і прочитування першого байта адреси 1 операнда і запис його в регістр адреси.
4. Збільшення на 1 вмісту регістра IP і прочитування другого байта адреси 1-го операнда і запис його в регістр адреси.
5. У регістр IP заноситься адреса першого операнда і відбувається прочитування 1-го операнда і запис його в регістр даних.
6. Збільшення на 1 вмісту регістра IP і прочитування першого байта адреси 2-го операнда і запис його в регістр адреси.
7. Збільшення на 1 вмісту регістра IP і прочитування другого байта адреси 2-го операнда і запис його в регістр адреси.
8. У регістр IP заноситься адреса другого операнда і відбувається прочитування 2-го операнда і запис його в другий регістр даних.
9. Складання даних, що знаходяться в двох регістрах даних, результат знаходиться в акумуляторі.
10. Обчислення адреси наступної команди. Перехід до виконання наступної команди.

### **2.1.3 Типи адресації**

Оптимальний розподіл пам'яті забезпечує найбільшу продуктивність обчислювальної системи. Розподіл пам'яті може бути статичним і динамічним. При статичному - пам'ять наперед розподіляється між програмами і даними, при динамічному – області пам'яті розподіляються у міру виникнення необхідності при виконанні програм. Динамічний розподіл більш ефективний і реалізується операційною системою. Динамічний розподіл має два види сторінкову і сегментну організацію.

Обмежене число розрядів, що відводиться у форматі команди для адреси операнда, змусило використовувати розподіл пам'яті на сторінки або сегменти і застосовувати дві або три частини адреси при зверненні до комірки. При сторінковій організації в оперативну пам'ять завантажується необхідна сторінка і лише після цього до неї може звертатися процесор. При сегментній організації вся пам'ять розбивається на участки - сегменти і звернення до пам'яті відбувається по номеру сегменту і адреси усередині сегменту. Іноді фізична адреса складається з трьох частин – номери сегменту, номери сторінки і адреси усередині сторінки.

Адреса комірки, до якої відбувається звернення обчислюється підсумовуванням адреси усередині сторінки з адресою поточної сторінки, яка задається неявно і зберігається в одному з регістрів процесора.

Сторінкова і сегментна адресація дозволила вирішити завдання колективного використання процесора і організації мультипрограмування

режиму, а також лягла в основу могутнього механізму – віртуальній адресації.

Віртуальна адресація полягає в тому, що кожна програма - користувач маніпулює з віртуальними адресами, завдяки чому створюється ілюзія пам'яті великого об'єму, хоча реально оперативна пам'ять може мати значно менший об'єм. Це створюється за рахунок динамічного перерозподілу сторінок пам'яті між основною і зовнішньою пам'яттю. Якщо відбувається звернення до фізичної сторінки, якої немає в основній пам'яті, то вона витягується із зовнішньої пам'яті і завантажується в основну. У кожному сегменті пам'яті зберігаються функціональні дані – в одних команди, ядро операційної системи, в іншому дані, в третьому стек. У деяких сегментах дозволено читання і запис, в інших – тільки читання. Таким чином, за допомогою механізму сегментації вирішуються завдання захисту пам'яті.

У машинних командах використовуються наступні типи адресації:

- Безпосередня – в команді поміщено значення операнда.
- Пряма – в команді вказаний адреса операнда.
- Непряма – вказується регістр або адреса комірки пам'яті, в якій знаходиться адреса операнда.
- Асоціативна – ознака операнда для асоціативної пам'яті.
- Неявна – в команді адреса операнда не вказаний, але він підраховується кодом операції.
- Використовуються два види адресації:
- Відносна – адреса обчислюється як сума адреси сегменту і зсуву, а також іноді вмісту індексних і базових регістрів.
- Стекова – адреса обчислюється як сума адреси сегменту, бази стека і зсуву вершини стека.

При адресації даних ще може використовуватися адреса базова і індексна. Тому така адресація називається відносною. Дані про сегмент знаходяться в регістрі процесора CS, зсув – в регістрі IP. При стековій адресації абсолютна адреса також обчислюється на підставі адреси сегменту стека (регістр SS), зсуву бази (регістр BP), зсуву або вершини стека (регістр SP).

Адреси команд, з якими оперує програма, називаються виконуваними (або зсувом) і вони звичайно не співпадають з реальними фізичними (або абсолютними) адресами комірок пам'яті. Як було показано вище, фізична адреса визначається як сума адреси сегменту і зсуву.

Варіантом непрямой регістрової адресації є індексна адресація. Вона забезпечує зручність роботи з масивами. При цьому адреса операнда обчислюється підсумовуванням адреси першого елементу масиву, який знаходиться у коду команди, цілого позитивного зсуву, що зберігається в індексному регістрі, ім'я якого містить команда.

Базова адресація забезпечує переміщуваність програм в пам'яті без їх зміни і модифікації. У коді команди знаходиться ім'я базового регістра,

значення якого складаються з адресою, що знаходиться в кодї команди і таким чином визначається фізична адреса операнда.

Адресація з автозбільшенням і автозменшенням також є окремим випадком непрямї регістрової адресації. При цьому або перед виконанням команди вміст регістра, в якому знаходиться адреса операнда, збільшується або зменшується на одиницю. Така адресація застосовується при організації циклів і в операціях із строковими змінними. Одним із застосувань цього виду адресації є стекова пам'ять. Регістр – покажчик стека зберігає адресу верхівки стека, так зветься поточна комірка, в якій проводиться запис і читання. Стекова пам'ять заснована на принципі «останній прийшов – перший вийшов». Є спеціальні команди для заштовхування даних в стек і витягання їх звідти.

Іноді застосовується складніша адресація індексний-базова, в якій використовується комбінація розглянутих методів.

#### 2.1.4 Система машинних команд процесора

Сучасний ІВМ сумісний комп'ютер має набір з більш ніж 240 машинних команд. Їх можна об'єднати у кілька груп по функціям.

Групи операцій виконуваних машинними командами:

1. Пересилка даних (однонаправлена, обміни, звернення до пам'яті, звернення до регістрів, уведення-виведення).
2. Арифметичні команди.
3. Логічні команди.
4. Строкові команди.
5. Звернення до зовнішніх пристроїв.
6. Передача управління (галуження, виклик підпрограм)
7. Спеціальні і допоміжні (управління процесором і його частинами, обнулення і зрушення акумулятора).

Оскільки людині складно сприймати двійкові коди машинних команд, були розроблені мови символічного кодування, одна з них мова Асемблер. Кожній машинній команді відповідає буквене позначення - мнемоніка. Нижче розглянуті приклади команд всіх груп, їх типи адресації і відповідний буквений код команди Асемблера.

Для запису чисел в Асемблері використовуються двійкові, десяткові і шістнадцятиричні числа. Двійкові мають позначку - букву В, десяткові – D, шістнадцятиричні - мітка Н.

Приклади: 1010В – 1010 двійкове, 1234D – 1234 десяткове, 1234Н – 1234 шістнадцятиричне, 50ADH - 50AD шістнадцятиричне.

Шістнадцятиричні цифри представлені в наступній таблиці:

Десятковий еквівалент	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Шістнадцятиричні цифри	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F



1. Пересилка даних між регістрами або між регістрами і комірками пам'яті.

Приклади:

№	Код операції	Адр. частина	Асемблер	Опис	Адресація
1	10110 reg	Data8	MOV R, data8	Занести в R число data8	Безпосередня
2	10001010	11dddsss	MOV R2, R1	Переслати з регістру R1 в R2	Пряма
3	11000110	xxxxxxx	MOV adr, data8	Занести в комірку з адресою adr число data8	
		Adr			Пряма
		Adr			Пряма
		Data8			Безпосередня
4	01010 reg		PUSH R	Записати вміст регістру R в стек	Неявна

2. Арифметичні операції.

У базовому процесорі X86 є операцій складання і віднімання – по 24 команди, множень і ділень - по 8 команд.

Приклад:

№	Код операції	Адр. частина		Асемблер	Опис	Адресація
1	10000000	11000d	Data 8	ADD R, data8	Складання вмісту регістра R з даними data8	Непосредств. Пряма

3. Логічні операції (49 команд)

№	Код операції	Адр. частина		Асемблер	Опис	Адресація
1	10000000	11100d	Data 8	AND R, data8	Поразрядное логич. умножение содержимого регистра с данными	Непосредств. Прямая

## 4. Звернення до зовнішніх пристроїв (8 команд)

Приклади:

№	Код операції	Адр. частина	Асемблер	Опис	Адресація
1	11100100	port	IN A, port	Ввести в рег. А число з порту port	Пряма
	11100110	port	OUT A, port	Вивести число А в порт port	Пряма

Як правило, ми не знаємо адрес всіх портів. У цьому випадку використовуються програмні або системні переривання, які посилають на відповідні підпрограми обробки переривань, вони і проводять введення і виведення даних у зовнішні порти.

Наприклад, для роботи з відеомонітором служить переривання 10H BIOS, для роботи з клавіатурою – переривання 16H BIOS.

5. Передача управління. Існують наступні групи команд, які слугують для змінення послідовного ходу виконання команд.

1. Безумовний перехід – обов'язковий перехід за вказаною адресою (три команди):

2. Перехід на підпрограму і повернення з підпрограми.

3. Умовний перехід – перехід здійснюється тільки при виконанні деяких умов (їх декілька десятків).

№	Код операції	Адр. частина	Асемблер	Опис	Адресація
1	11111111	11100ddd	--	JMP RW	Безумовний перехід до команди, адреса якої RW

6. Спеціальні команди були додані в пізніших процесорах з технологією MMX і її розширеннями. Ці команди призначені для прискорення складних математичних операцій, виконання операцій з плаваючою комою, управління кеш-пам'яттю, обробки тривимірної графіки, кодування відео, обробки потоків даних. Вони призначені для ефективнішої обробки з високим ступенем паралелізму деяких мультимедійних даних, що часто зустрічаються (графічних і потоками аудіо- і відеоданих, розпізнавання мови). Паралелізм означає, що одні і ті ж алгоритми застосовуються до групи даних, наприклад, при перетворенні графічного зображення (повороти, переміщення).

Необхідно відзначити, що математичний співпроцесор має свою систему команд відмінну від основного процесора. Співпроцесори виконують складні математичні операції з довгими операторами і числами з рухомою комою, обчислення функцій і т.д. Виконувана програма повинна бути розрахована на використання співпроцесора і мати відповідні команди, інакше співпроцесор задіяний не буде.

### **2.1.5 Час виконання команд**

Продуктивність комп'ютера залежить від швидкості виконання команд процесором і часу звернення до пам'яті. Швидкість визначається складністю команди. Час виконання команд визначається наступними чинниками:

1. Тактова частота процесора.
2. Швидкодії оперативної пам'яті.
3. Наявність і місткості надоперативної і буферної пам'яті.
4. Формат і розрядність машинного слова.
5. Розвиненість системи команд процесора і наявність співпроцесора.

Тактова частота процесора визначає час виконання кожного такту, з яких складається машинний цикл.

В процесі виконання команди відбувається багатократне звернення до оперативної пам'яті. Тому від часу виконання цієї операції істотно залежить швидкість роботи комп'ютера. Швидкодія пам'яті постійно збільшується, але відстає від швидкодії процесора. Для подолання відставання застосовують надоперативну або кеш-пам'ять для тимчасового зберігання даних. Ця пам'ять має обмежений об'єм, але підвищену швидкодію. В кеш-пам'ять заноситься частина інформації оперативної пам'яті для зменшення часу доступу до цієї інформації. Підвищення швидкодії досягається за рахунок застосування статичного типу пам'яті що працює на підвищеній частоті, а також за рахунок розміщення такої пам'яті в корпусі процесора.

Чим більше розрядність машинного слова, тим більше інформації прочитує процесор за один такт. При меншій розрядності для цього буде потрібно декілька тактів.

Розвинена система команд дозволяє виконувати деякі операції однією командою замість декількох. Наявність співпроцесора значно розширює систему доступних команд в область більш складних операцій. Наприклад, операція множення в розвиненій системі виконується як одна команда, а в менш розвинених - як послідовність команд додавання.

## **2.2 Машинна програма**

### **2.2.1 Структура машинної програми**

Програма це певна послідовність команд (інструкцій), яка повинна бути виконана на комп'ютері для вирішення завдань. Програма кодується в цифровому вигляді і зберігається в пам'яті комп'ютера.

Машинні коди, які складають програму, записуються в певну область пам'яті комп'ютера. Програма може знаходитися в пам'яті необмежено довго і не виконувати ніяких дій. Щоб виконати програму процесору указують адресу першої комірки цієї області, він читає всі комірки по черзі і виконує команди. При цьому з іншої області пам'яті прочитуються або записуються дані, необхідні для вирішення завдання. Після виконання однієї програми процесору указується наступне завдання і т.д. Розподілом завдань і ресурсів (пам'яті і ін.) займається операційна система.

Коди машинних команд, які записані в пам'ять складають машинну програму. При цьому команди в цій програмі можуть виконуватися послідовно у порядку розташування в пам'яті (лінійна програма) або цей порядок може змінюватися (проводиться передача управління). Тому структурні типи програм: лінійні і розгалужені.

Лінійна програма – виконується в природному порядку розташування команд в програмі і пам'яті.

Приклад:

1. Початок програми. У реєстр адреси записується адреса початку програми в пам'яті.

2. Прочитування всіх байтів першої команди і виконання її.

3. Приріст реєстра адреси і перехід до наступної команди.

4. Прочитування всіх байтів другої команди, виконання її і т.д.

Галуження або передача управління – зміна природного порядку виконання команд. Типи галужень:

1. Перехід на виконання команди, розташованої за вказаною в програмі адресою. Перехід може бути безумовний або умовний.

2. Виклик підпрограми - виконання ряду команд, які складають підпрограму, і повернення в точку виклику. Виклик може бути умовний або безумовний.

3. Переривання - виконання ряду команд, які складають підпрограму обробки переривання, і повернення в точку виклику.

При передачі управління в лічильник команд записується адреса команди, на яку передається управління. При переході до виконання підпрограм, адреса чергової команди заноситься в стек і використовується для продовження виконання основної програми після виконання підпрограми.

На рис.35 приведений приклад програми, в якій спочатку виконуються команди в порядку їх зберігання в пам'яті 1-2, а потім відбувається галуження програми.

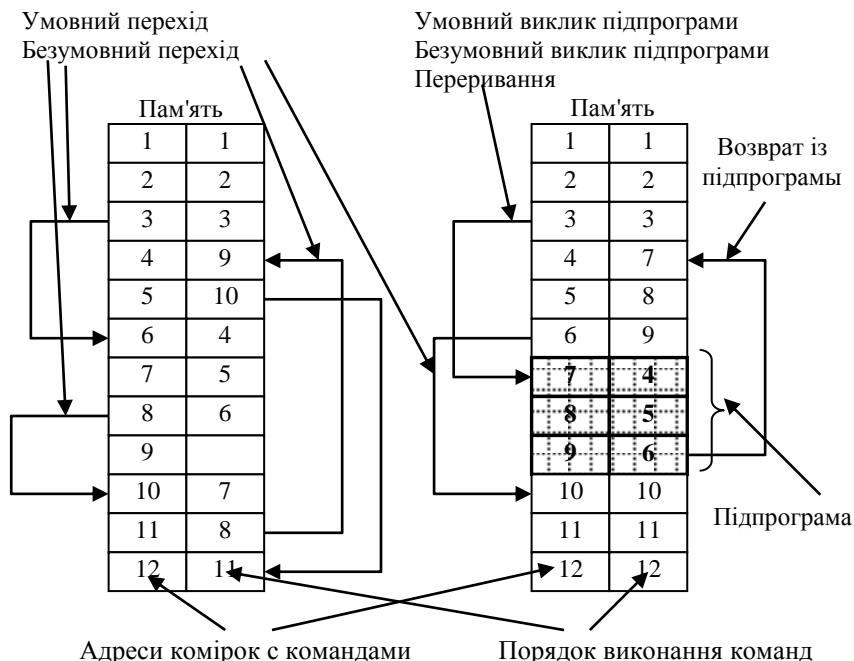


Рис. 35. Приклади програм з галуженням

### 2.2.2 Обробка переривань

При роботі комп'ютера виникають ситуації коли треба терміново приймати заходи. Сигналом виникнення такої ситуації є електричний сигнал на спеціальному вході запиту переривань. Заходи здійснюються шляхом виконання відповідної підпрограми обробки переривань.

Процесори мають фізичні входи апаратних переривань і програмну систему обробки переривань. Програмним способом можна збільшити кількість точок входу в підпрограми обробки переривань (векторів переривань), тобто кількість варіантів реагування на різні ситуації, що викликають переривання.

Підпрограми обробки переривань передбачають два типи переривань внутрішні і зовнішні. Зовнішні переривання відбуваються під час виникнення сигналу на одному з апаратних входів процесора. Внутрішні переривання відбуваються в наступних випадках:

- при діленні на нуль під час виконання операцій ділення,
- при реалізації покрокового режиму для відладки програм,

- при переповнюванні реєстрів,
- при виконанні команди переривання «INT адреса».

При програмній обробці переривань повинні виконуватися наступні обов'язкові дії:

1. Зупинення виконання попередньої програми. Занесення в стек поточних значень основних реєстрів процесора. (Після закінчення обробки ці дані прочитуються із стека і виконання попередньої програми продовжується).
2. Блокування інших запитів на переривання.
3. Занесення адреси підпрограми обробки переривання в лічильник команд.
4. Виконання програми обробки.
5. Виконання команди виходу з підпрограми обробки. Прочитування даних із стека, відновлення лічильника команд і продовження виконання перерваної програми.

Окрім переривання можуть бути замасковані, заборонені, а також виконуватися відповідно до їх пріоритету.

### 2.2.3 Час виконання програми

Час виконання програми визначається часом виконання

- завантаження програми із зовнішнього пристрою пам'яті в оперативну;
- окремих команд програми;
- звернення до пам'яті;
- процедур введення-виведення, якщо вони передбачені у програмі;
- системних процедур і процедур BIOS.

Час виконання команд розглянуто у попередньому розділі.

Якщо об'єм оперативної пам'яті не вистачає для всієї інформації, то частина її перекидається у зовнішню пам'ять, а потім, при необхідності, повертається назад. Ясно, що час такої операції залежить від об'єму оперативної пам'яті і швидкодії зовнішньої пам'яті.

Будь-яка програма не обходиться без операцій введення-виведення, які займають іноді значний час. Під час цих операцій процесор знаходиться в режимі очікування або ці операції доручаються додатковому процесору (контроллеру). Контроллери значно підвищують швидкодію центрального процесора.

Існує обробка інформації у реальному часі, коли виконання програм синхронізовано з протіканням реальних фізичних процесів. У цьому випадку час виконання програм обмежений як зверху, так і знизу. Щоб збільшити час виконання програми, додають порожні команди, які не виконують ніяких корисних дій. Для зменшення часу виконання програми її розділяють на частини, які допускають одночасне їх виконання різними процесорами або різними ядрами процесора.

Робота програми контролюється, забезпечується і управляється системними засобами, драйверами і засобами базової системи введення-

виведення. Всі ці засоби також вимагають часу на їх виконання і інших ресурсів.

### **2.3 Взаємодія програмних і апаратних засобів**

Як видно з викладеного вище, обидві складові частини комп'ютера апаратна і програмна тісно взаємодіють між собою і не можуть функціонувати самостійно, окремо один від одного. Крім того, будь-яка складна програма високого рівня виконується після перетворення у машинні коди, які безпосередньо виконуються апаратними засобами. Для їх виконання необхідний цілий комплекс ресурсів апаратних засобів відповідним чином налаштованих на спільну роботу. Безпосередню взаємодію програми з апаратними засобами, їх настройку і управління ресурсами здійснює базова система введення-виведення (BIOS).

#### **2.3.1 Ієрархія взаємодії**

У взаємодії будь-яких програмно керованих пристроїв можна виділити ієрархічні рівні цієї взаємодії. Для опису структури і призначення цих рівнів можна скористатися моделлю відкритих систем. Найвищий рівень цієї моделі мають користувачі, тобто ті, хто використовує технічну систему в своїх цілях. Це може бути людина або будь-яка система управління. Потім слідує програмні рівні і найнижчий рівень це рівень фізичних пристроїв і сигналів. У тому або іншому вигляді елементи цієї моделі присутні при взаємодії будь-яких технічних систем.

Інформація переміщується по рівням з передавальної сторони до приймальної. Кожен рівень передавальної сторони віртуально взаємодіє з відповідним аналогічним рівнем приймальної сторони, тобто так, якби відсутні всі нижчестоячі рівні. Будь-який рівень виконує замовлення вищестоячих рівнів і нічого не знає про нижчестоячі рівні, тобто ніяк не залежить від них: їх кількості, типу і алгоритму роботи.

Кожен рівень виконує свої функції і пов'язаний з вищестоящим і нижчестоячим рівнями через програмні або апаратні інтерфейси. Прикладні програми взаємодіють з операційною системою через інтерфейс API (Application Programming Interface), який визначає, як виконується запис і прочитування даних на диск, друк і інші функції. Інтерфейс API складається з набору системних викликів. Системний виклик – механізм, наданий прикладним програмам для запиту виконання ОС тієї чи іншої системної функції. Для організації системних викликів резервують одне або кілька програмних переривань (наприклад, `int 21h` у MS-DOS). На цьому рівні інтерфейс ніяк не залежить від апаратного забезпечення.

Операційна система взаємодіє з пристроями через відповідні драйвери, що входять до складу BIOS. Інтерфейс операційної системи з BIOS є стандартним, а ось зв'язок з сотнями різних типів зовнішніх пристроїв забезпечують відповідні програми – драйвери пристроїв. Драйвери унікальні

для кожного пристрою і поставляються разом з ним. Драйвери повинні відповідати як особливостям апаратного забезпечення, так і операційній системі.

Наприклад, розглянемо процес передачі документа факсом. Ми вставляємо документ у факс-апарат і не бажаємо знати, що відбудеться далі. А подальша доля документа це вже турбота нижчестоячих рівнів. Апарат встановлює зв'язок з іншим апаратом, сканує документ і передає інформацію в передавачик. Подальша доля документа це вже турбота не цього рівня, а нижчестоячих рівнів. На цих рівнях відбувається формування сигналів і передача їх по різних каналах і лініях зв'язку (кабельним, радіорелейним, оптичним, космічним, а також аналоговим або цифровим).

На рис. 36 показана структура рівнів взаємодії комп'ютера і периферійного пристрою з урахуванням моделі відкритих систем. Наприклад, Користувач\_1 виводить на друк документ для Користувача\_2. Логічне або віртуальне з'єднання полягає у тому, що Пользователь\_1 як би безпосередньо передає документ Пользователю\_2, а як реально відбувається передача і що твориться на нижчестоячих рівнях його не цікавить. Реально ж документ передається засобами спочатку операційної системи, а потім засобами BIOS, драйверів і фізичного з'єднання у відповідний пристрій друку. У пристрої друку відбувається зворотний процес перетворення документа від низького рівня у вигляді сигналів до найвищого рівня зображення на папері, зручного для сприйняття людиною – Користувачем\_2.

Також проводиться зворотна передача по всім рівням, наприклад, при опитуванні клавіатури. При натисненні на кнопку клавіатури, згідно програми роботи даного пристрою, формується відповідний код. Згідно алгоритму взаємодії з інтерфейсом код передається через адаптер і кабель на відповідний адаптер комп'ютера. Тут цей сигнал приймає базова система введення-виведення і передає його в операційну систему, звідки він поступає в прикладну програму. Таким чином, в тексті документа з'являється відповідний символ. Щоб цей символ відбився на екрані монітора, сигнали повинні пройти аналогічний шлях рівнів, але вже у бік монітора.



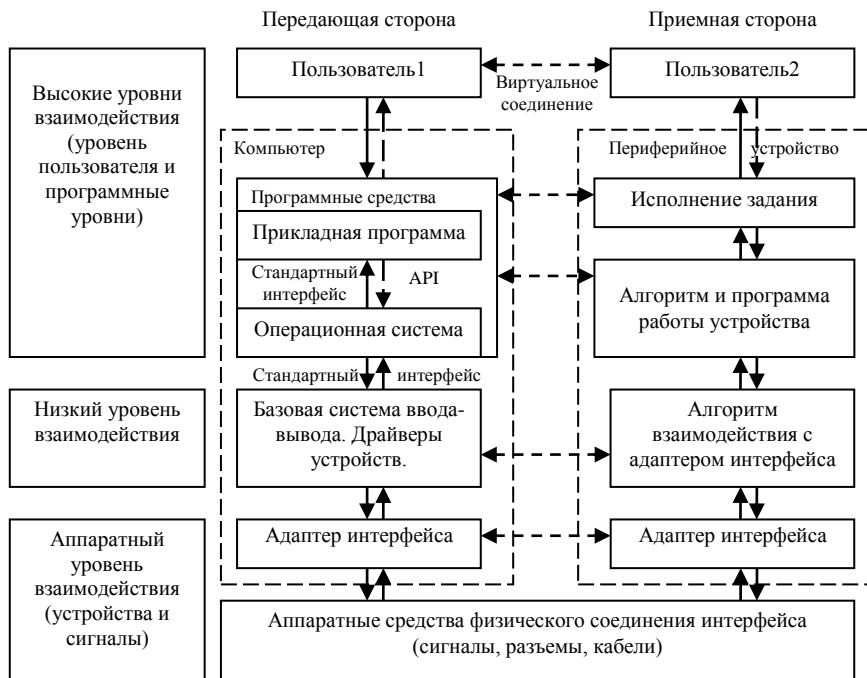


Рис. 36 Иерархическая модель взаимодействия в компьютерной системе

### 2.3.2 Базава система введення-виведення

Основне призначення базової системи введення-виведення (BIOS) полягає у виконанні функцій сполучної ланки програмних і апаратних засобів комп'ютера. Наприклад, вона опитує клавіатуру, мишку і надає результат операційній системі, звідки дані поступають в прикладну програму. Виведення зображення на монітор також здійснюється за участю BIOS.. Таким чином, здійснюється взаємодія клавіатури, мишки і монітора з прикладною програмою.

Різні пристрої або програми працюють по різному. Їм потрібні різні вхідні сигнали, різний час реагування, вони видають різні вихідні сигнали і т.д. Всі ці особливості «знає» тільки відповідна програма - драйвер. Драйвер це процедура (програма) низького рівня, яка безпосередньо взаємодіє з системними пристроями.

BIOS здійснює низькорівневе (т.е. безпосередньо пов'язане з апаратними засобами) обслуговування всіх компонентів комп'ютера, визначення і установка взаємозв'язку між ними. Виконання цих функцій пов'язано з

параметрами апаратних компонентів і режимами їх роботи, які називаються параметрами BIOS. Настройкою параметрів для кожного пристрою задається конкретна конфігурація універсального комп'ютера для виконання певних класів завдань.

Раніше для настройки параметрів і режимів апаратних засобів використовувалися механічні перемикачі або перемички на платні і пристроях. Зараз вони ще подекуди використовуються, але зручнішими і сучаснішими є програмно керовані електронні перемикачі, стан яких необхідно зберігати, коли комп'ютер вимкнений, тобто в енергонезалежній пам'яті. Ці настройки апаратних засобів є невід'ємною частиною базової системи введення-виведення. У комп'ютері є мікросхема з живленням від батареї, що містить годинник і спеціальну CMOS пам'ять, де зберігаються всі настройки, дата і час. Там же знаходиться програма BIOS і програма BIOS Setup – для налаштування параметрів пристроїв.

Програма BIOS включає багато компонентів, які здійснюють взаємодію прикладних і системних програм з блоками і пристроями комп'ютера. Одна частина BIOS (системна) розташована в ПЗП на системній платні, містить драйвери основних компонентів (клавіатура, дисководи) і все необхідне для початкового завантаження. Системна BIOS здійснює тестування, запуск всіх компонентів, пошук і ініціалізацію драйверів пристроїв і інших частин BIOS, розташованих на платах розширення, адаптерах, дисководів і ін. Третьою частиною BIOS є драйвери пристроїв, які встановлюються на жорсткому диску і завантажуються разом з операційною системою. Завантажувані драйвери завжди новіші і вони заміщують драйвери початкового завантаження.

При включенні живлення комп'ютера і самотестуванні блоку живлення видається сигнал „сброс” і запускається стартовий блок (програма) BIOS.. Проводиться перевірка характеристик процесора, напруги живлення, температури, частоти системної шини і ін. При відхиленнях параметрів від норми включення комп'ютера може бути припинено. Далі запускаються програми ініціалізації і тестування всіх компонентів комп'ютера – програма POST (Power On Self Test). Ведеться пошук і активізація драйверів з ПЗП адаптерів (відеоадаптера, адаптера дисководів, мережевого адаптера і ін.). Після цього відбувається запуск завантажуваних драйверів і операційної системи.

Тестування комп'ютера проводиться всякий раз при його включенні. Перед початком тестування будь-якого пристрою системою BIOS за адресою 080 або 0300 видається код цього пристрою. Якщо тестування має негативні результати, то подальше тестування припиняється, а за виданим кодом пристрою можна судити про те, який пристрій несправно. Для індикації коду пристрою застосовується спеціальна POST плата.

Програма BIOS Setup запускається клавішею Del при завантаженні комп'ютера. У головному меню програми, що відкрилося, представлені

тематичні меню для вибору параметрів. Тут можна встановити системний час і дату, проконтролювати і встановити частоту процесора та інші його параметри, вибрати з якого приводу здійснювати завантаження операційної системи і послідовність пошуку завантажувального модуля і багато що інше. Докладніше робота з програмою Setup BIOS розглянута у спеціальній літературі. Щоб провести настройки з операційної системи в деяких випадках можна використовувати програму tweakbios, але це слід робити дуже обережно щоб не вивести комп'ютер з ладу.

Таким чином BIOS містить:

- Набір драйверів для взаємодії операційної системи з пристроями при завантаженні системи;
- POST програма самотестування при включенні живлення всіх важливих компонентів системи.
- Завантажувач операційної системи – програма виконує пошук завантажувального сектора на диску і виконує код початкового завантаження.

При завантаженні операційної системи BIOS шукає в кореневому каталозі файл CONFIG.SYS, який інтерпретується як набір команд конфігурації. Це ще один файл, що відповідає за взаємодію програмних і апаратних засобів. Некоректності в його командах можуть приводити до діагностики апаратних помилок.

### **2.3.3 Розподіл системних ресурсів**

Таким чином, у комп'ютері є багато різних пристроїв, які приєднані до процесора через систему інтерфейсів. При цьому всі пристрої використовують спільні програмні і апаратні засоби - системні ресурси. До системних ресурсів відносять:

- Адресний простір пам'яті;
- Канали запитів переривання;
- Канали прямого доступу до пам'яті;
- Адреси портів введення-виведення.

Системні ресурси розподіляються між працюючими пристроями і процесами у комп'ютері. Ресурси обмежені, тому при їх розподілі один ресурс може призначатися різним пристроям, тоді виникають конфлікти. У наведеному вище списку ресурси розташовані по ступені убування частоти конфліктів. Найчастіше виникають конфлікти при розподілі пам'яті і каналів переривання. Це пов'язано з тим, що пам'ять і канали переривання найбільш інтенсивно використовуються всіма пристроями, а кількість їх обмежена.. Канали прямого доступу використовуються рідко, тому конфлікти майже не виникають. Операція введення-виведення використовується часто, але для неї відведено 64 Кбайта адресного простору і це досить всім пристроям.

Кожний платні адаптера потрібен свій набір ресурсів. Наприклад, для аудіопристрою потрібні все їх види.

### 2.3.4 Адаптери, драйвери, контроллери

Адаптери – це апаратні засоби для узгодження логіки роботи і фізичних сигналів двох різних пристроїв.

Драйвери – це програмні засоби, що виконуються процесором, для узгодження алгоритмів і сигналів стандартних і нестандартних інтерфейсів.

Контроллери - це апаратні засоби для управління системними і периферійними пристроями. Контроллер працює за заданою програмою, яка може бути жорстко представлена у вигляді логічної схеми контроллера або у вигляді програми в пам'яті контроллера. У останньому випадку контроллер є локальним процесором, призначеним для виконання обмеженого набору функцій.

У комп'ютері є багато різних адаптерів, драйверів та контролерів. Розглянемо деякі з них. Раніше вже розглянуті звукові і відеоадаптери.

Адаптер SCSI використовується для високошвидкісного введення-виведення інформації, частіше для зовнішніх пристроїв пам'яті. Адаптер вимагає один канал переривання, один канал прямого доступу до пам'яті і деякий адресний простір.

Адаптер USB використовується для широкого спектру пристроїв високошвидкісного введення-виведення. Наприклад, для принтерів, сканерів, змінної пам'яті і ін. До шини USB може бути підключене до 127 пристроїв і при цьому для всіх потрібне тільки одне переривання.

Адаптер послідовних портів COM використовується для зв'язку з низькошвидкісними пристроями: модем, мишка і ін.

При підключенні зовнішніх пристроїв і адаптерів широко використовується технологія Plug and Play (PNP). Раніше при додаванні нового пристрою необхідно було проводити безліч настройок і перерозподіли системних ресурсів. Нова технологія PNP дозволяє спростити і автоматизувати цей процес. Але для її реалізації потрібна відповідна підтримка. Це означає, що повинні бути відповідні апаратні засоби на платі адаптера, програми в BIOS, засоби підтримки в операційній системі.

При установці плата адаптера інформує BIOS і ОС про необхідні системні ресурси. У відповідь їм повідомляється про вільні ресурси і плата адаптера проводить необхідні настройки.

Будь-яка функція, що виконує комп'ютер, може бути реалізована кількома способами:

- Апаратна реалізація: функція жорстко задана електричною і логічною схемою приладу. Цій спосіб має низку гнучкість, високу вартість, але і високу швидкодію.

- Програмна реалізація: дозволяє багатократно використовувати одні апаратні засоби для виконання багатьох функцій. Має високу гнучкість, низку вартість, але дуже низку швидкодію.

– Комбінована програмно-апаратна реалізація: частина функції реалізована апаратно, а частина – програмно. Це оптимальний спосіб, який має не високу вартість і досить хорошу швидкодію.

Більшість функцій комп'ютера реалізовані комбінованим способом. Усі адаптери, драйвери та контролери є складовими частинами, що реалізують цей комбінований спосіб виконання певних функцій.

## **2.4 Програмне управління об'єктами**

У діяльності людини постійно виникають задачі, вирішити які допомагають автоматичні пристрої. Наприклад:

- Виконання важкої, небезпечної і монотонної роботи.
- Розширення можливості людини за швидкістю, точність і ін.
- Підвищення продуктивності і ефективності праці.
- Збільшення якості виконання робіт.
- Створення складних виробничих і технологічних систем

Управління автоматичними об'єктами може бути жорстким (апаратним) або програмним. При першому способу алгоритм управління закладається у структуру і схему автоматичного пристрою, змінити його не можна. При програмному способу алгоритм задається програмою, його легко змінити і налаштувати на певні умови.

### **2.4.1 Структура системи управління**

За допомогою програмно керованого комп'ютера можна не тільки проводити обчислення, але і виконувати фізичні дії, управляти об'єктами і процесами. Наприклад, переміщати вантажі, обробляти деталі, плавити сталь, управляти машинами, прати білизну і багато що інше. Всі фізичні дії, які виконує чоловік в своїй практичній діяльності, можуть виконуватися під контролем і управлінням спеціальних програм, що виконуються на комп'ютері. Для цього програма повинна в темпі протікання процесу з'ясувати стан об'єкту управління у вигляді набору значень різних його параметрів (вихідних параметрів), порівняти стан із заданими вимогами і видати на виконуючий пристрій набір команд управління, для змінення вхідних параметрів об'єкта і приведення його в заданий, на даний момент часу, стан (рис.37).

Виконання фізичних дій для змінення вхідних параметрів об'єкта, здійснюється виконавчими пристроями, які реалізуються апаратними методами у вигляді електричних схем з жорсткою логікою функціонування.

Задані вимоги можуть гнучко змінюватися залежно від виконуваних завдань, типу і параметрів продукції. Гнучкість програмного управління дозволяє швидко переходити з випуску однієї продукції на іншу.

Таким чином, основні властивості програмних систем управління:

1. Обробка даних в реальному масштабі часу.
2. Гнучке програмне управління залежить від завдання.

3. З'ясування стану об'єкту і виконання керуючих дій здійснюється за допомогою спеціальних периферійних засобів.

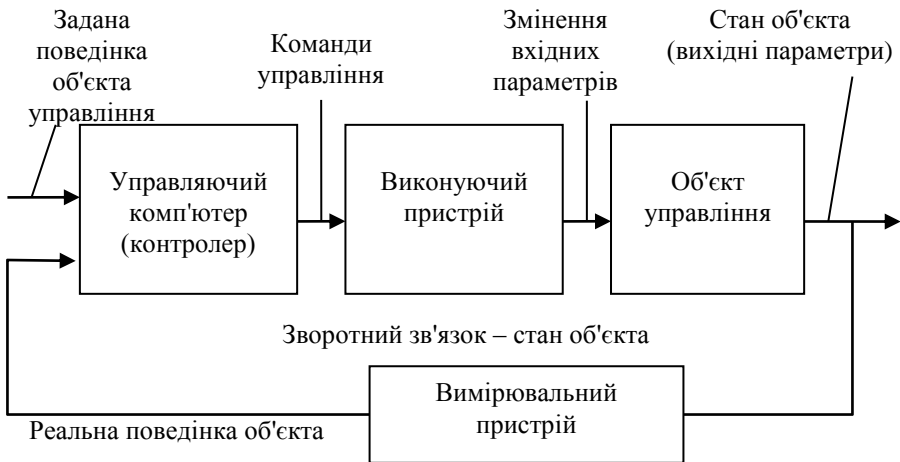


Рис. 37 Загальна структура системи управління.

#### 2.4.2 Реалізація програмного управління устаткуванням

Для того, щоб програма могла виконувати вказані дії комп'ютер повинен бути оснащений спеціальними вимірювальними і виконуючими апаратними засобами (рис.38). Ці засоби є органами чуття, а також руками і ногами системи. Такі функції виконують датчики різних фізичних величин і виконавчі механізми, які дозволяють змінювати відповідні параметри різних фізичних процесів. Можливості цих пристроїв значно більше, чим у органів чуття і кінцівок людини. Тому такі автоматичні системи є надійним помічником, що розширює наші можливості.

Для реалізації виконавчих і вимірювальних ланок потрібні спеціальні апаратні периферійні засоби, які забезпечують зв'язок комп'ютера з керованим процесом, узгодження видів сигналів, видів енергії і потужності. Схема взаємодії керуючого комп'ютера з технологічним обладнанням зображена на рис. 39.

Обробка інформації в комп'ютерах, що управляють, проводиться в реальному масштабі часу. В певні моменти керованого процесу, проводяться всі необхідні вимірювання и керуючи дії. Для реалізації цього режиму потрібна спеціальна організація програмного забезпечення. Тому, що програма повинна виконуватися у строго означені проміжки часу.

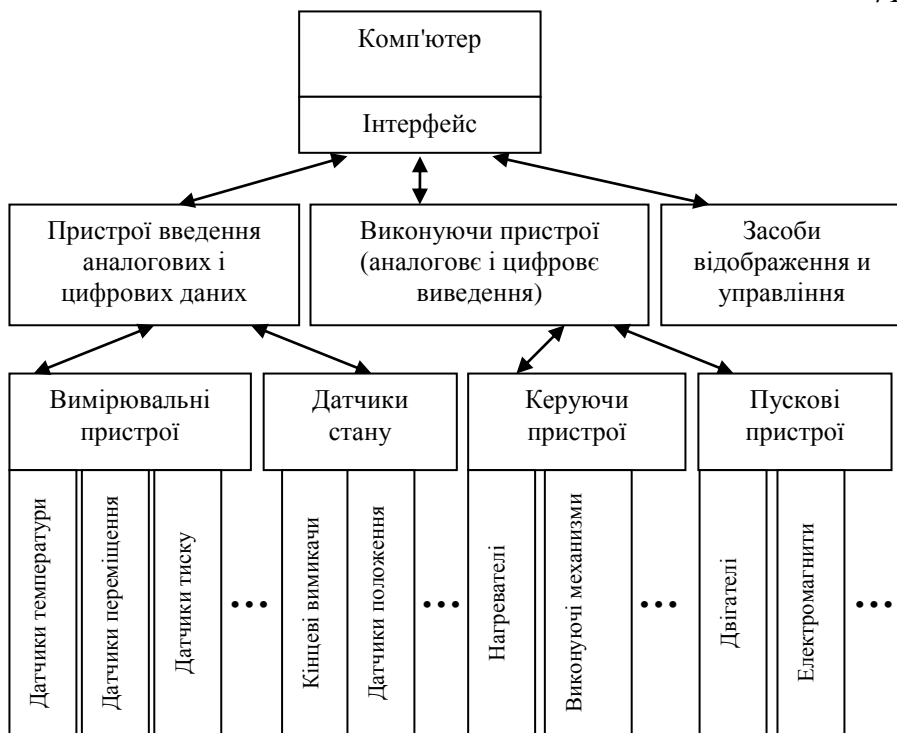


Рис. 38 Структура керуючої системи.

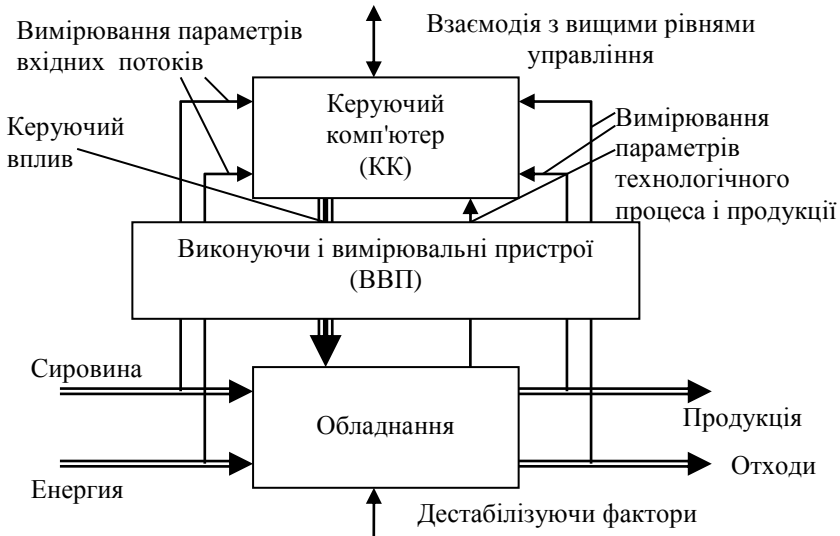


Рис. 39 Структура комп'ютерного програмного управління технологічним обладнанням.

### 2.4.3 Розповсюджені системи управління

Для керування складними і розповсюдженими процесами окремі керуючі комп'ютери об'єднуються у системи. Таки системи мають ієрархічний характер і розповсюджені у просторі (рис.40). Кількість рівнів управління не обмежена. Система будується так, що б в аварійних умовах кожен рівень міг функціонувати автономно без зв'язку з вищестоящим. При цьому, тільки знижується якість і ефективність управління, без втрати працездатності всієї системи.

У розповсюджених системах значну роль мають системи передавання інформації і комунікаційні мережі. Крім того, системи управління повинні мати високу надійність і ремонтоздатність, тому що збитки от виходів з ладу дуже велика.

Комп'ютери, що управляють, обробляють потоки інформації від трьох основних джерел:

1. Вищестоящого рівня керування (дані планування).
2. Обслуговуючого персоналу (дані управління).
3. Керованого процесу (дані процесу).

При необхідності пристрої апаратного рівня здатні керувати процесом автономно без втручання інших рівнів керування.



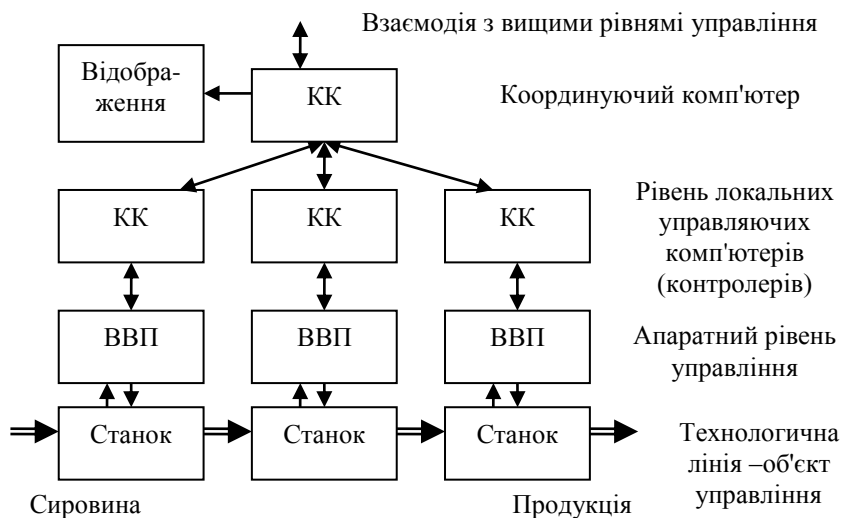


Рис. 40 Структура програмного управління технологічної лінії.

## МОДУЛЬ 3

### Комп'ютерні системи

Останніми роками рівень складності завдань, що вирішуються засобами комп'ютерної техніки, значно зріс і вимагає високої продуктивності і ефективності обчислень. Підвищення якості досягається поліпшенням характеристик апаратних засобів, вдосконаленням алгоритмів обробки інформації і нарощуванням програмно-апаратних засобів. Це приводить до створення достатньо складних технічних рішень.

Тому при створенні комп'ютерної системи для вирішення складних завдань треба правильно вибрати її архітектуру і склад апаратних і програмних засобів. А для цього треба знати якісні характеристики комп'ютерної системи, режими і технології її роботи, шляхи підвищення її ефективності і надійності, знати принципи тестування і обслуговування.

#### 3.1 Класифікація комп'ютерних систем

Існує цілий ряд класифікацій по різним критеріям, властивостям і ознакам, залежно від суб'єктивних потреб і переваг різних категорій користувачів. За архітектурними ознаками можна розглядати наступні групи систем (рис.41). Зосереджені – одномашинні, обчислювальні комплекси і системи. Розповсюджені – телеобробка і обчислювальні мережі.

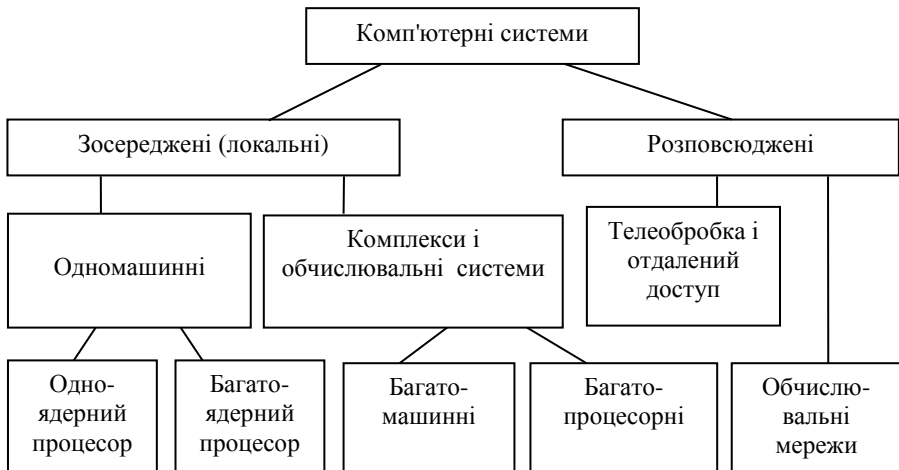


Рис. 41. Класифікація комп'ютерних систем

Одномашинні системи – це автономний комп'ютер з єдиним процесором. Як правило, це універсальні комп'ютери різної потужності від супер- до мікро-комп'ютерів.

Більшість комп'ютерів мають цифровий принцип дії. Але існують як цілі комп'ютери так і окремі обчислювальні пристрої з аналоговим принципом дії. Аналогові це завжди спеціалізовані пристрої з обмеженими функціями. Цифрові комп'ютери це більш універсальні і складніші пристрої.

Одним з різновидів одномашинних систем є персональні комп'ютери (PC або ПК), призначені для індивідуального використання. Такі комп'ютери можуть бути стаціонарними і мобільними (ноутбуки, наладонные і ін.). Іншим різновидом одномашинних систем є мікрокомп'ютери або мікроконтролери, які використовуються для управління в технологічних процесах, промислового устаткуванні і побутовій техніці.

В даний час дуже рідко комп'ютери використовуються автономно, тим більше індивідуально. Частіше вони включені в різні комплекси, локальні і глобальні мережі і використовуються корпоративно на користь групи користувачів, що значно підвищує ефективність їх використання. Це означає, що будь-який сучасний комп'ютер це або цілком комп'ютерна система або частина складнішої системи.

<b>Достоїнства автономних комп'ютерів</b>	<b>Недоліки</b>
Величезний досвід застосування	Низька надійність
Простота і низька вартість	Низька продуктивність
Великий досвід створення апаратних і програмних засобів	Низкі функціональні можливості
Великий парк апаратних і програмних засобів	

Обчислювальні комплекси – сукупність декількох комп'ютерів або процесорів і загальносистемного програмного забезпечення. Комплекси бувають багатомашинні і багатопроекторні. Декілька процесорів обмінюються між собою інформацією через ПВВ або загальну пам'ять.

<b>Достоїнства комплексів</b>	<b>Недоліки</b>
Підвищується надійність	Потребує спеціального програмного забезпечення
Підвищується продуктивність	Складне апаратне забезпечення
Розширюються функціональні можливості	Недостатні функціональні можливості

Обчислювальні системи – спеціальне АС і ПО для вирішення завдань конкретної області.

Наприклад, системи для обробки відео і графічної інформації, проектних і наукових завдань, рішення інтегральних і диференціальних рівнянь. Системи бувають проблемно-орієнтовані і спеціалізовані. Такі системи, як

правило, менш універсальні і тому дешевші (при масовому виробництві), надійні і більш пристосовані до умов роботи. Адаптивні Обчислювальні системи – в яких конфігурація змінюється залежно від структури алгоритму вирішуваної задачі.

Наприклад, системи інформаційні (у метро і ін.), для управління підприємством або технологічними процесами.

Системи телеобробки – для обробки даних переданих по каналах зв'язку.

Складаються з центрального комп'ютера і менш потужних периферійних комп'ютерів або терміналів для введення-виведення інформації.

Наприклад, система замовлення квитків на потяги або банківська система збору інформації з філіалів.

<b>Достоїнства системи</b>	<b>Недоліки</b>
Простота і зручність роботи	Обмежені функціональні можливості
Висока продуктивність	Висока вартість при виробництві малими серіями.
Низька вартість	Індивідуальна розробка
Висока надійність	

Обчислювальні мережі – це сукупність комп'ютерів, що взаємодіють між собою через засоби передачі інформації. Потрібні спеціальні засоби передачі інформації і мережеве програмне забезпечення.

Мережі бувають локальні і глобальні. Локальні обчислювальні мережі розташовані в обмеженому просторі, що дозволяє одержувати дуже високу швидкість передачі інформації і отже продуктивність обчислювальної системи. Глобальні обчислювальні мережі – не обмежені в просторі. Швидкість передачі в каналах зв'язку обмежена їх дальністю, типом і якістю.

<b>Достоїнства обчислювальних мереж</b>	<b>Недоліки</b>
Висока надійність і резервування обчислювальних і інформаційних ресурсів.	Надійність залежить від надійності каналів зв'язку.
Висока продуктивність і функціональні можливості.	Производительность обмежена пропускной здатністю каналів.
Групове використання знань і досвіду.	Необхідність стиковки різних операційних систем.
Сумісне використання ресурсів.	Труднощі управління великою системою.
Постійне збільшення інформаційних ресурсів.	Потрібні спеціальні засоби передачі інформації і мережеве програмне забезпечення.

### 3.2 Проектування і системна інтеграція

Проектування системи полягає в завданні її необхідних параметрів і характеристик, розробки кількох варіантів системи з обмеженого набору існуючих програмно-апаратних компонентів, вибору критеріїв оцінки варіантів і остаточного вибору варіанта проектованої системи. Комп'ютерні системи будуються із крупних функціональних блоків або комплексів (системна платня, операційна системи, програмні і апаратні платформи). Тому проектування полягає у виборі серед дискретного і обмеженого числа варіантів. Такий вибір називається системною інтеграцією, тобто об'єднання програмних і апаратних засобів в єдину працездатну систему із заданими характеристиками. Системна інтеграція базується на системному підході і охоплює всі етапи життєвого циклу виробу від цього створення до утилізації.

Згідно архітектурним рівням і системному підходу системна інтеграція також складається з рівнів: рівень користувача, архітектурний рівень, рівень інтеграції даних і застосувань, рівень інтеграції апаратних і програмних платформ, рівень інтеграції компонентів, технологічно-конструктивний рівень. Інтеграція на кожному рівні оперує крупними об'єктами, які також є продуктом інтеграції на нижчому рівні.

Таким чином, проектування комп'ютерної системи полягає у виборі такої її конфігурації, яка забезпечує задану якість при певних обмеженнях. Часто критерієм якості є ефективність системи. Академік В.М.Глушков запропонував використовувати як критерій ефективності вартість ефективної машинної операції

$K_э = S(t) / N_э(t)$ , де:  $S(t)$  - витрати на амортизацію і експлуатацію комп'ютера за певний період,  $N_э(t)$ , - число ефективних операцій, виконаних комп'ютером за цей період.

При збільшенні продуктивності цей показник знижується. Існують і інші критерії: обчислювальна потужність, швидкодія і ін.

Обмеженнями є задані значення надійності, швидкодії, об'єму пам'яті, набору операцій, розрядності, точності, можливостей введення-виведення даних. Обмеженнями можуть бути структура системи, елементна база і програмні засоби.

Окрім фундаментальних принципів Неймана (см. розділ 1.1) при розробці комп'ютерної техніки використовуються ще ряд принципів, одержаних на основі багаторічної практики розробників комп'ютерних систем:

- Оптимізація усього життєвого циклу системи з урахуванням екологічних факторів.
- Надлишковість апаратури і інформації збільшує надійність системи.
- Надлишковість створюється шляхом дублювання і резервування, а також паралельною роботою окремих блоків.
- Використання стандартизованих і уніфікованих блоків, програм, методів для проектування і обслуговування систем.

– Авторизація процесу проектування та обслуговування.

### **3.3 Якість комп'ютерних систем**

#### **3.3.1 Характеристики якості**

Якість і ефективність є комплексними характеристиками, залежними від ряду параметрів і властивостей системи. Для користувача якість роботи комп'ютера полягає, перш за все, в підвищенні його швидкодії при рішенні складних задач, розширенні функціональних можливостей, підвищенні надійності, поліпшенні екологічних параметрів і зручності роботи з комп'ютером. Важливим параметром є вартість.

Ефективність роботи комп'ютера визначається цілим набором параметрів. Це, в першу чергу, швидкодія процесора, яка, у свою чергу, визначається його тактовою частотою і розрядністю. Істотним параметром є тактова частота системної шини, шин розширення і локальних шин. Ефективність визначається також наявністю і величиною системних ресурсів: різних видів пам'яті, можливостей підсистем введення-виведення, переривань, прямого доступу до пам'яті і ін.

Ефективність використання комп'ютера істотно залежить від архітектури і конфігурації його апаратних і програмних засобів, настройки і ефективності їх взаємодії, базової системи введення-виведення. Ефективність також значно залежить від надійності роботи всіх систем і достовірності інформації.

Над поліпшенням цих параметрів працюють сотні фірм і мають в цій справі істотний прогрес. Вдосконалення йде двома шляхами: одномашинні системи, а також комплекси та обчислювальні системи і мережі.

По першому шляху збільшується частота процесора, швидкість шини, розміри кеш-пам'яті, удосконалюються режими роботи комп'ютера, створюються нові технології обробки інформації, розробляються нові інтегральні схеми і розвивається технологія виробництва інтегральних схем, зокрема процесорів, поліпшується архітектура. Проте останнім часом нарощування швидкодії сповільнилося. Це викликано тим, що традиційні методи наблизилися до межі своїх можливостей: частота процесорів за останній рік збільшилася трохи, збільшення швидкості шини і розмірів кеш-пам'яті не дають значного результату. Все це приводить до збільшення ролі нових технологій обробки інформації.

Обмеження можливостей апаратних засобів привело до того, що перспективним став другий шлях збільшення ефективності за рахунок розширення складу апаратних засобів. Це створення комплексів, систем і мереж. Потужність локальних комп'ютерів збільшується шляхом створення багатопроесорних і багатомашинних комплексів і обчислювальних систем. Останнім часом найбільш значний прорив відбувся при переході на двудерну архітектуру процесорів і створенні багатопроесорних структур.

#### **3.3.2 Основні параметри комп'ютера.**

Основними параметрами комп'ютера можна назвати такі параметри, які дають загальне уявлення про можливість введення, виведення і зберігання інформації, а також швидкодії і ефективності її обробки. До основних параметрів можна віднести, принаймні, наступні:

- Функціональні можливості системи.
- Продуктивність – кількість операцій за одиницю часу.
- Розрядність процесора і системної шини, швидкодія, тактова частота.
- Структура машинних команд процесора.
- Форми представлення чисел.
- Адресний простір.
- Номенклатура і швидкодія ПВВ і пристроїв зв'язку.
- Номенклатура, місткість і швидкодія різних видів пам'яті.
- Робота в багатозадачному режимі.
- Завантаження пристрою в режимі розділення времени- відсоток зайнятості в процесах.
- Надійність – інтенсивність відмов, напрацювання на відмову, середній час відновлення.
- Вартість одиниці ресурсу.

Час виконання команд процесором в різних поколіннях постійно знижується. У раних процесорах на виконання однієї команди витрачалося в середньому 12 тактів. У процесорах серії 286 і 386 швидкодія підвищилася і на одну команду витрачалося вже 4,5 такту. У процесорах 486 і аналогічних цей параметр понижений до 2 тактів. Сучасні процесори шостого і сьомого покоління виконують до трьох команд за один такт.

Продуктивність визначається як частотою так і швидкодією процесора, причому у різних процесорів по- різному. Тому для порівняння процесорів фірмою Intel запропонована система тестів під назвою індекс iCOMP. Якість процесора оцінюється його індексом. Наприклад, процесори Pentium мають індекс від 100 до 3000. Сучаснішим і достовірнішим набором еталонних тестів є набір SYSmark 2004, який дозволяє проводити перевірку будь-яких апаратних засобів.

Сучасні процесори працюють на частоті, яка рівна добутку деякого множника на тактову частоту системної шини. Значення останній швидко зростає і складає зараз величину від 266 до 533 і навіть 800 Мгц. Тактова частота процесора досягає 3.4 Ггц. Значення множника можна встановити за допомогою перемичок на системній платні або в BIOS. Для отримання інформації про поточну конфігурацію і параметри системи можна скористатися спеціальними програми і тестами.

### **3.3.3 Конфігурація комп'ютера**

Загальне уявлення про комп'ютер і його властивості дає склад основних блоків і вузлів комп'ютера, а також їх параметри, тобто конфігурація

комп'ютера. Конфігурація ПК використовується при виборі і порівнянні різних комп'ютерів між собою. Залежно від призначення конфігурацію можна представити детально або скорочено у вигляді переліку основних пристроїв і їх параметрів.

Приклад короткого запису конфігурації комп'ютера:

Intel Pentium IV 3.0 / 800 MHz/ DDR 1 Gb 667 MHz / MB Asus ATX / HDD 100 Gb / FDD 1.44 / GeForce 6600GT PCI-E 256 MB / 17" Samsung / DVD-RW x 24/24/52 /

Цей короткий запис розшифровується таким чином:

№	Позначення	Опис
1.	Intel Pentium IV	Фірма виробник і тип процесора
2.	3.0	Тактова частота процесора 3 Гигагерца (1 ГГц=10 <sup>9</sup> Гц)
3.	800 MHz	Тактова частота процесорної шини 800 MHz
4.	DDR 1 Gb 533 MHz	ОЗП типу DDR, місткістю 1 Гігабайт, тактова частота 533 MHz
5.	MB Asus ATX	Системна платня фірми Asus формфактор ATX
6.	100 Gb	Вінчестер об'ємом 100 Гігабайт
7.	1.44	Дисковод для гнучких дисків об'ємом 1.44 Мбайт
8.	GeForce 6600GT PCI-E 256 MB	Відеоадаптер GeForce 6600GT з шиною PCI-E і пам'яттю 256 MB
9.	17" Samsung	Розмір екрану монітора 17 дюймів фірми Samsung
10.	CD-RW x 10/24/52	Привід DVD-RW (для читання та запису) з швидкостями для запису, перезапису і читання

### 3.4 Режими і технології роботи процесора

Створення нових процесорів супроводжується появою у них нових властивостей, підвищенням ефективності, швидкодії, надійності. З'являються нові технології і режими управління обчислювальною системою. Підвищення продуктивності процесора досягається різними шляхами:

- Апаратні засоби (нарошування тактової частоти, збільшення об'єму кеш-пам'яті, збільшення кількості виконавчих пристроїв).
- Розробка нових режимів і технологій роботи.
- Додавання в систему нових команд і інструкцій прискорюючих обробку певних видів даних.
- Мікроархітектурні прийоми – попередня вибірка команд і прогноз галужень, конвейерна обробка.



Апаратні засоби розглянуті вище, їх перспективи дуже обмежені. Розробка інших шляхів з кожним роком розширюється.

### **3.4.1 Режими роботи**

Комп'ютер має два основних режими роботи: реальний або розрахований на одного користувача і захищений або розрахований на багато користувачів, мультипрограмний (так називається тому, що виконується одночасно декілька програм і приймаються заходи по захисту пам'яті від несанкціонованого звернення програм до недозволених областей). Система захисту забезпечує безпеку програм і даних від втручання інших користувачів. Захисту також потребують програми операційної системи. Спотворення інформації може виникнути унаслідок помилок в програмах при розділенні часу.

Для захисту використовуються програмно-апаратні засоби, які перевіряють всі адреси звернення до пам'яті. Якщо адреса відноситься до виділеного даній програмі масиву пам'яті, то звернення дозволяється, інакше команда припиняється.

У розрахованому на багато користувачів режимі завдання можуть виконуватися в пакетному режимі (по черзі) або в режимі розділення часу, коли процесор перемикається між завданнями, кожного разу займаючись одним завданням в перебігу невеликого інтервалу часу. Створюється враження, що завдання вирішуються одночасно.

Діалоговий режим дозволяє користувачу контролювати хід виконання програми і втручатися, при необхідності. Режим реального часу забезпечує видачу результатів в темпі протікання процесів. Наприклад, система управління рухом транспорту.

Всі ці режими забезпечують операційні системи, зокрема сучасні системи Windows.

### **3.4.2 Нові технології обробки інформації**

З кожним роком все стрімкіше росте швидкодія і ефективність роботи персональних і інших типів комп'ютерів. В першу чергу, росте тактова частота процесора, зростає об'єм всіх видів пам'яті. Але найголовніше, виникають нові хитромудрі методи і технології обробки інформації, які дозволяють при існуючих частотах і об'ємах пам'яті істотно підвищити швидкодію комп'ютера. Саме у цих інтелектуальноемких технологіях бачиться перспектива розвитку комп'ютерної техніки, тому, що не можна нескінченно нарощувати частоти і об'єми пам'яті.

В даний час найбільш довершеними є процесори Pentium п'ятого покоління фірми Intel і його аналоги інших виробників. Розглянемо їх основні якості, що підвищують ефективність роботи комп'ютера.

Суперскалярна технологія це одночасне виконання декількох команд в процесорі. Сучасні могутні процесори мають повну систему команд (CISC).

При виконанні деякі з цих команд розбиваються на простіші команди (RISC), які реалізуються конвейсним способом. Тому декілька команд виконується одночасно, що збільшує швидкодію процесора.

Технологія MMX і її оновлення (SSE, SSE2, SSE3) дозволяють прискорити компресію і декомпресію відеоданих, маніпулювання зображенням, шифрування і операції введення-виведення. Це досягається розширенням кеш-пам'яті і набору команд процесора, введенням технології SIMD (single instruction – multiple data), що дозволяє одній команді здійснювати одну операцію над декількома даними. Нові команди призначені для паралельного виконання команд, прискорення складних математичних операцій, розпізнавання мови, обробки графіки, відео і потоків даних. Нові команди часто застосовуються при обробці мультимедійної інформації, наприклад, обробка різних точок зображення по одному алгоритму.

Динамічне виконання команд дозволяє одночасно обробляти декілька команд. Якщо при виконанні команди відбувається затримка (наприклад, із-за звернення до пам'яті), то виконується наступна команда. При цьому проводиться прогноз всіх переходів і команд, які виконуватимуться, проводиться аналіз потоку даних і попереджувальне виконання команд. Для прогнозу застосовуються спеціальні алгоритми.

Технологія HT (hyper treading) – багатопотокове (гіперпотокове) виконання програм. Технологія HT вперше з'явилася в процесорі Pentium 4 3,06 ГГц. На базі одного процесора формується два або більше логічних (віртуальних) процесора. У процесорі що підтримує технологію HT повинні бути два набори регістрів, а основні обчислювальні блоки, кеш-пам'ять і введення-виведення використовуються спільно. Один фізичний процесор одночасно обробляє два і більш незалежних потоків команд. При обробці одного потоку завжди є інтервали часу простою і очікування, коли процесор не зайнятий, саме в ці моменти проводиться обробка другого потоку. Таким чином, підвищення продуктивності засновано на підвищенні завантаження і ефективності використання фізичного процесора.

Ця технологія підвищує продуктивність до 30 %, тоді як блоки, що її реалізують, займають менше 5 % площі кристала процесора. Таке збільшення продуктивності досягається тільки в багатозадачних режимах в операційних системах Windows XP, Linux або для завдань, які допускають їх виконання в багатопотоковому режимі. Це означає, що програми повинні спеціально створюватися для виконання в цьому режимі. Операційна система «бачить» два процесори і звертається до них як до двох різних. Технологія HT підтримується не тільки процесором, але і апаратними засобами системної логіки, засобами BIOS і операційної системи.

Проте технологія HT для завдань, які є принципово однопотоковими, не дає вигоди, а навпаки збільшує час виконання завдань, оскільки другий віртуальний процесор ніколи не використовується. Тому в BIOS передбачається можливість відключати засоби технології HT.

Конвейерна обробка – це одночасне виконання різних тактів послідовних команд різними підсистемами процесора, тобто одночасне виконання декількох команд. Конвейери з'явилися у 286 процесорів, вони мали 6 байт регістрів, куди заносилася черга виконуваних команд, які виконуються одночасно. У 486 процесора черга вже досягла 16 байт. Конвейерна технологія збільшує ефективну швидкість процесора до 5 разів.

Архітектура подвійної незалежної шини (DIB – dual independent bus) збільшує пропускну спроможність шини процесора. Разом з основною системною шиною є друга або фоновна шина, яка застосовується для роботи кеш-пам'яті другого рівня. Перша основна шина використовується для роботи з основною пам'яттю, а друга додаткова шина використовується кеш-пам'яттю другого рівня. В результаті швидкість збільшується, оскільки можна одночасно працювати з основною і кеш пам'яттю. При необхідності, можна використовувати одночасно обидві шини, що теж збільшує швидкість передачі.

Необхідно відзначити, що для ефективного використання всіх цих нововведень і технологій в комп'ютері повинні бути передбачені засоби апаратної і програмної підтримки (системна плата, BIOS, операційна система і т.д.), інакше можливості залишаться нереалізованими.

### **3.4.3 Конвейерний обмін з пам'яттю.**

Одна з конвейерних технологій застосовується при обміні інформацією з пам'яттю. Наприклад, такий конвейер застосовується в шині AGP (відеоадаптер). На рис.42 зображена загальна схема обміну: контроллер видає адресу, яка по шині адреси поступає в пам'ять, активізується потрібна комірка і видає дані, а вони по шині даних поступають в контроллер. Для виконання кожної з цих операцій потрібен час, тому дані поступають пізніше, ніж була видана їх адреса. Це запізнювання визначає швидкість пам'яті. Запізнювання може бути більше або менше, але воно принципово є у будь-яких пристроїв.



Рис. 42 Схема обміну інформацією

На рис.43 зображені сигнали в шинах за відсутності конвейєра. У момент часу  $T_1$  був виданий адреса контроллером, а у момент  $T_2$  в контроллер поступили дані. Через деякий час, потрібний на прочитування цих даних, адресу можна зняти з шини адреси – момент часу  $T_3$ . Через час, потрібний на перемикання можна виставити нову адресу, –  $T_4$ . Таким чином, контроллер не змінює адреси ША(К) поки не одержить своїх даних ШД(К), а з моменту появи першої адреси до появи другої повинно пройти час  $T_4-T_1$ . Цей час і визначає швидкодію процесів обміну інформацією з пам'яттю. Воно залежить від фізичних властивостей пристроїв і ліній зв'язку, які принципово не можуть працювати нескінченно швидко.

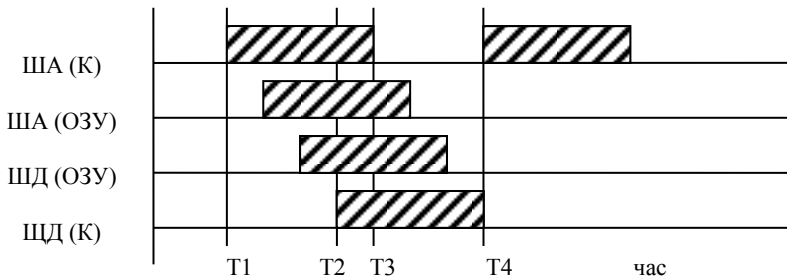


Рис. 43 Сигнали у шинах без конвейєра.

За наявності конвейєра, контроллер не чекає надходження даних за поточною адресою, а безперервно видає наступні адреси (рис.44). При цьому виходить, що якщо у момент  $T_1$  була видана перша адреса, то відповідні дані поступають тільки у момент  $T_2$  коли контроллер вже видає наступні адреси.

Тому при читанні даних враховується той факт, що це дані не з поточної адреси, а з попередньої, тобто враховується затримка.

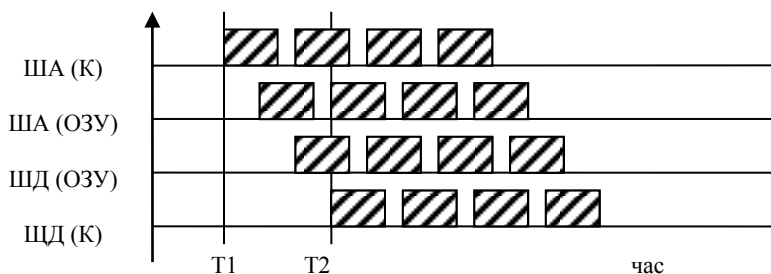


Рис. 44 Сигнали у шинах з конвейером.

### 3.5 Багатопроцесорні системи

Класична багатопроцесорна система це однорідна система, яка містить декілька однотипних і рівноправних процесорів, що взаємодіють між собою і керованих єдиною операційною системою. Крім того, до багатопроцесорних можна віднести системи з кількома нерівноправних процесорів. У цих системах є основний або центральний процесор і залежні від нього процесори: математичний співпроцесор, графічний процесор відеоадаптера, сигнальний процесор аудіопристрою, контролери і т.і.

#### 3.5.1 Однорідні багатопроцесорні системи

В однорідних багатопроцесорних системах взаємодія між процесорами може бути на рівні регістрів або через загальну пам'ять. Декілька процесорів підвищують надійність комп'ютера і підвищують його продуктивність. При виході з ладу одного процесора інші можуть взяти на себе його функції. Швидкість рішення збільшується для достатньо складних завдань, які дозволяють організувати їх рішення за допомогою паралельних обчислень. Простим збільшенням числа процесорів можна з обмежених по можливостям компонентів будувати могутні суперкомп'ютери.

В даний час існують три методи організації багатопроцесорних структур: конвейерні, векторні і матричні. На рис.45,а представлена базава однопроцесорна структура. Тут один процесор виконує всі команди деякої програми і послідовно здійснює перетворення ряду вхідних даних у вихідні.

У векторній структурі (рис.45,в ) кожен процесор також виконує всі команди програми, але таких процесорів  $N$  і вони одночасно обробляють  $N$  вхідних даних.

У конвейерній структурі (рис.45,б) вхідні дані обробляються послідовністю  $M$  процесорів, що виконують кожен свою команду, які

слідують одна за одною в програмі. Одночасно проводиться обробка  $M$  вхідних даних в  $M$  процесорах, тобто швидкість роботи підвищується в  $M$  разів.

У матричній структурі (рис.45,г) реалізована обробка векторних  $N$ -мірних даних конвейером у  $M$  процесорах. Продуктивність збільшується в  $N \times M$  разів.

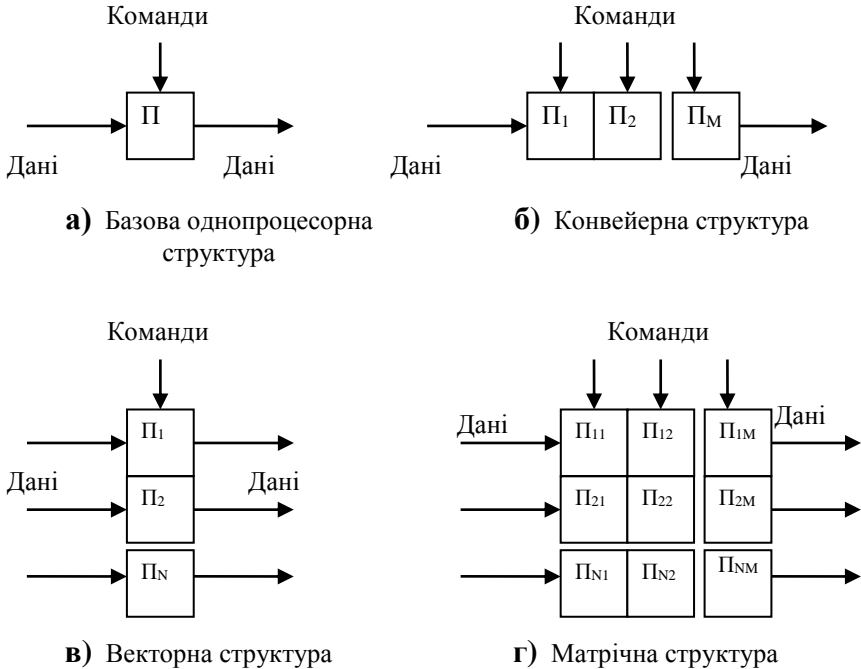


Рис. 45 Багатопроцесорні структури

### 3.5.2 Розподілені функції управління

В однопроцесорному комп'ютері центральний процесор програмним шляхом може виконати будь-яку операцію і будь-який алгоритм. Але, якщо він буде виконувати усі функції, то для цього треба багато часу і інших ресурсів. Час виконання задачі може значно збільшитися. Тому в комп'ютері деякі найбільш прості, але трудомісткі операції доручаються апаратним засобам, це звільняє процесор від рутинної роботи і підвищує швидкість всієї системи. Такими апаратними засобами є співпроцесори, контролери, адаптери, системна логіка і т.д. Деякі з цих апаратних засобів є процесорами по потужності такі ж, як центральний процесор або навіть могутніші.

Таким чином, будь-який сучасний персональний комп'ютер є по суті багатопроцесорним, оскільки окрім основного або центрального процесора має математичний співпроцесор, який працює спільно з основним і взаємодіє з ним через реєстри або внутрішню пам'ять і значно збільшує швидкість рішення математичних задач. Є також свій процесор на платі відеоадаптера, він займається обробкою графічної інформації. Крім того, в комп'ютері є безліч різних контролерів: контролер пам'яті, переривань, прямого доступу до пам'яті, введення-виведення і ін. Всі ці окремі процесори (рис.46), виконуючи свої вузькі завдання, знижують навантаження на центральний процесор, підвищують ефективність його функціонування. Всі ці пристрої взаємодіють з центральним процесором через загальну шину або пам'ять.

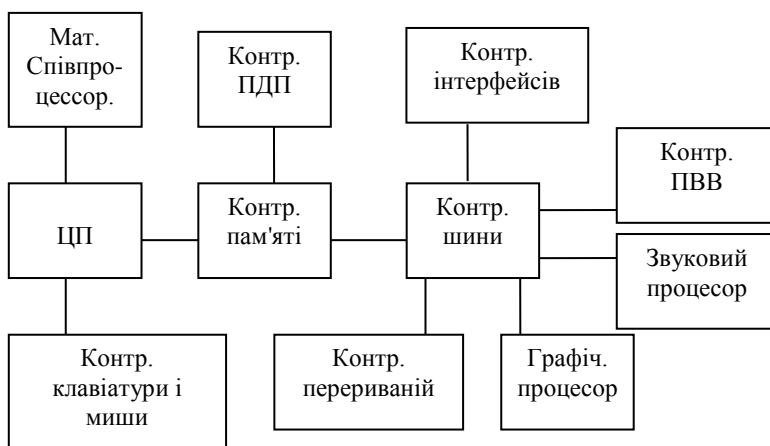


Рис. 46 Структура функцій управління (контролерів) у комп'ютері

Нова ера в розвитку комп'ютерів наступила із створенням багатоядерних процесорів. Поки на ринку з'явилися тільки двужадерні, що дають майже двократне збільшення продуктивності, але це тільки початок. Багатоядерні процесори мають декілька комплексів основних вузлів тих, що відповідають за обробку інформації (ядер), які працюють одночасно і мають швидкий обмін даними між собою. Загальними є тільки кеш-пам'ять з комутатором і контролером пам'яті, через яку ядра обмінюються інформацією, і процесорна шина, через яку вони взаємодіють із зовнішніми пристроями.

Історію розвитку процесорів можна представити наступним ланцюжком: один процесор – один процесор з технологією НТ – один процесор з двома ядрами – два процесори – багато процесорів. У міру розвитку ресурси одного процесора все більше розділяються на дві і більше частин. У технології НТ з'явилися два набори реєстрів, в двужадерних додалися два набори основних обчислювальних засобів, і нарешті, при повному розділенні маємо два

незалежні процесори. Чому не можна було відразу замість одного поставити два процесори. Річ у тому, що двух- і багатопроцесорні системи на багато складніші і дорожчі чим однопроцесорні. Ця технологія застосовується для могутніх і дорогих серверів і суперкомп'ютерів, але очевидно з часом вона з'явиться і в персональних комп'ютерах.

У багатоядерних процесорах і багатопроцесорних системах продовжують розвиватися всі традиційні технології підвищення якості обробки інформації. Підвищення ефективності можливе за рахунок комбінування технологій. Так можуть бути комп'ютери з декількома багатоядерними процесорами або є процесор з двома ядрами і технологією HT, що еквівалентно чотирьом процесорам (Pentium Extreme Edition).

Багатоядерна технологія робить перші кроки, тому є окремі тимчасові недоліки. Наприклад, перші розробки (Pentium D) мають два ядра, які є двома окремими процесорами, що знаходяться в одному корпусі. В цьому випадку виникає проблема синхронізації і актуалізації інформації в кеш-пам'яті кожного ядра, якщо обидва ядра кэшируют одну і ту ж область оперативної пам'яті. Зміна змісту кеш-пам'яті одного ядра повинна супроводжуватися відповідними змінами в другому, а така синхронізація істотно знижує ефективність роботи.

У більш пізніх версіях двоядерних процесорів Intel Core Duo використовується одна загальна кеш-пам'ять з динамічним розподіленням. У цьому процесорі додано ще 10 команд для реалізації складної арифметики, графіки і кодування відео.

Для ефективної роботи всіх цих апаратних технологій програмні засоби повинні бути адаптовані і оптимізовані під багатопотокову технологію.

### **3.5.3 Розподілені процесори**

Одна з головних причин зниження ефективності обчислень в комп'ютерах це великий час доступу до пам'яті, який може бути у декілька разів більшим ніж час обробки. Тому прагнуть наблизити і спростити взаємодію засобів обробки і пам'яті. Крім того, швидкодія пам'яті вже зараз відстає від швидкодії процесорів, і це відставання збільшуватиметься.

Одним з шляхів прискорення взаємодії є використання кеш-пам'яті, яка максимально наближена до процесора або навіть розташована з ним в одному корпусі. Іншим шляхом є технологія «процесор в пам'яті». Його суть в тому, що частина обчислювальних функцій процесора переміщується в пам'ять. Такі комп'ютерні системи містять могутній центральний процесор і менш могутні процесори усередині чіпів пам'яті. Тим самим розподіляються обчислювальні операції між різними ресурсами системи.

Процесор в пам'яті дозволяє прискорити виконання простих операцій, які часто зустрічаються. Це операції: збірка і розбирання фреймів, індексований доступ і доступ по покажчику, обробка списків і т.і. Локалізація



цих операцій дозволяє істотно прискорити доступ до оперативної пам'яті і зменшити потоки даних між пам'яттю і головними процесорами.

### **3.6 Надійність системи і достовірність інформації**

Надійність найважливіша характеристика якості будь-якої системи. Існує цілий ряд параметрів надійності, а також математична теорія надійності. В цілому, надійність це здатність системи виконувати свої функції в перебігу заданого часу. Надійність системи залежить від надійності кожного його елементу і способу їх взаємодії.

Для забезпечення надійності систем застосовують ряд заходів: при проектуванні - шляхом вибору архітектури, при виробництві - підвищенням надійності компонентів, при експлуатації - відповідним обслуговуванням, ремонтом і навчанням персоналу.

З погляду архітектури, підвищення надійності забезпечується введенням резервування елементів і збільшенням надлишкових ресурсів. В цьому випадку при виході з ладу частини ресурсів, ті що залишилися зможуть їх замінити і виконувати задані функції. Наприклад, система RAID, яка застосовується для резервування накопичувачів на жорстких магнітних дисках.

Ефективний спосіб забезпечення рівня надійності це профілактичний ремонт і поточне обслуговування. На рис. 47 зображено зниження рівня надійності системи без ремонту і з ремонтом, але різними міжремонтними періодами Tr1 і Tr2.

У комплекс заходів щодо підвищення надійності входить захист, як від внутрішніх збоїв, так і від зовнішніх дій і перешкод. Основний об'єкт захисту це інформація. Для її захисту і підвищення достовірності існують різні методи. В першу чергу це методи ослаблення зовнішніх електро-магнітних впливів: фільтри, екрани, симетричні лінії зв'язку.

Якщо похібка всеж виникла, захист від неї здійснюється методами кодування цифрових сигналів. Найпростіший метод кодування це метод контрольної суми. Будь-яка група даних при зберіганні або передачі містить кілька біт контрольної суми. При отриманні даних біти контрольної суми порівнюються з реальною підрахованою контрольною сумою. Якщо вони не співпадають, то цей блок даних відкидається. Більш складні методи дозволяють відновити помилки, але для цього потрібна велика надлишкова інформація, тобто інформації повинно значно більше ніж спочатку.

Одним з видів зовнішнього впливу є дія вірусів і несанкціонований доступ в систему. Для захисту в архітектуру комп'ютера додаються спеціальні програмні і апаратні засоби.

Дослідження працездатності апаратних і програмних елементів комп'ютера проводиться за допомогою тестових програм. Тести запускаються при включенні системи, але можна тестувати і в процесі роботи, якщо виникає підозра в несправності.

Великий розвиток одержали системи управління живленням. При цьому системні плати повинні підтримувати функції процесора АРМ (Advanced Power Management) і SMM (System Management Mode). Вони дозволяють переводити різні вузли комп'ютера на різні рівні готовності і енергоспоживання. Ці процеси регламентує стандарт управління живленням ACPI (Advanced Configuration and Power Interface).

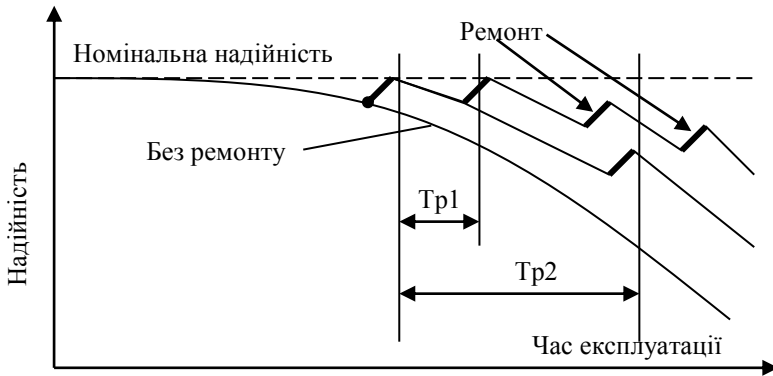


Рис. 47 Змінення надійности при експлуатації комп'ютера.

### 3.7 Дослідження і тестування комп'ютера

Для тестування комп'ютера і дослідження його параметрів існує багато різних програм, які поставляються спільно з операційною системою або окремо.

#### 3.7.1 Диспетчер пристроїв

Диспетчер пристроїв встановлюється на комп'ютер водночас з операційною системою Windows. Диспетчер дозволяє отримати список усіх пристроїв, які встановлені на комп'ютері, а також їх параметрів. На рис. 48 зображено вікно зі списком пристроїв.

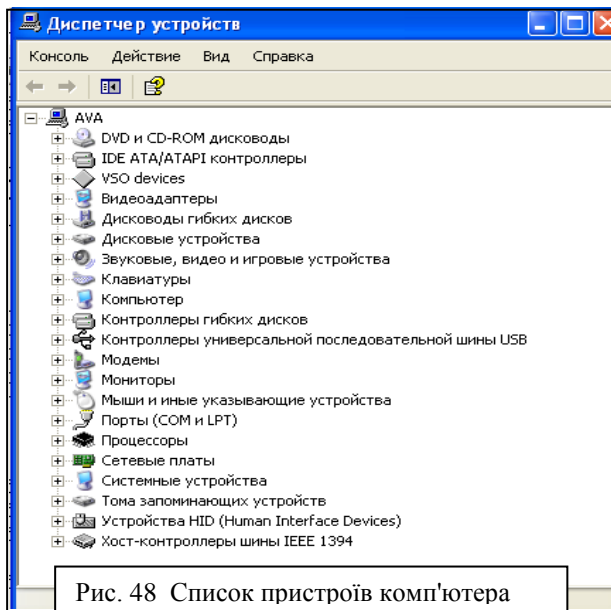


Рис. 48 Список пристроїв комп'ютера

Диспетчер пристроїв використовують для оновлення драйверів (або програмного забезпечення) устаткування, зміни настройки устаткування, а також для усунення неполадок.

#### Увага!

Неправильне зміни конфігурації устаткування може пошкодити систему. Перед внесенням змін в конфігурацію устаткування слід прочитати відповідні інструкції.

Диспетчер пристроїв дозволяє:

- визначати правильність роботи устаткування комп'ютера;
- змінювати параметри конфігурації обладнання;
- визначати драйвери для кожного пристрою і одержувати відомості про кожен драйвер;
- змінювати додаткові параметри і властивості пристроїв;
- встановлювати оновлені драйвери пристроїв;
- отключати і удаляти пристрої;
- здійснювати повернення до попередньої версії драйвера;
- роздруковувати список пристроїв, встановлених на комп'ютер.

Звичайно диспетчер пристроїв використовується для перевірки стану устаткування і оновлення драйверів пристроїв на комп'ютері. Можливості діагностики диспетчера пристроїв також можуть використовуватися досвідченими користувачами, що володіють глибокими знаннями про

комп'ютерне устаткування, для вирішення конфліктів пристроїв і зміни параметрів ресурсів.

**Увага!**

Неправильна зміна настройок ресурсів може привести до відключення пристроїв або послужити причиною неправильної роботи комп'ютера. Параметри ресурсів повинні змінюватися тільки користувачами, що володіють глибокими знаннями комп'ютерного устаткування і його настройки.

Звичайно для зміни параметрів ресурсу не потрібно використовувати диспетчер пристроїв, оскільки ресурси розподіляються автоматично операційною системою в процесі настройки устаткування.

Для управління драйверами пристроїв необхідні дозволи і привілеї, перераховані нижче. Такі дозволи мають адміністратори.

- Привілей завантаження/вивантаження драйвера.
- Дозволи, необхідні для копіювання файлів в каталог system32\drivers.
- Дозволи, необхідні для запису параметрів в реєстр.

Вхід в систему з такими привілеями робить систему безпеки комп'ютера уразливою. Тому небажано використання привілеїв адміністратора без необхідності. Управляти пристроями за допомогою диспетчера пристроїв можна тільки на локальному комп'ютері. Для удаленого комп'ютера диспетчер пристроїв працюватиме тільки в режимі читання.

У меню Вид можна вибрати, що відображається у вікні диспетчера:

Параметр	Описание
Пристрої по типу	Отображение устройств по их типу, например монитор или мышь. Имя подключения отображается под типом. Это режим просмотра устанавливается по умолчанию. Отображение устройств по способу подключения к компьютеру. Каждое устройство отображается под тем основным устройством компьютера, к которому оно подключено. Например, при наличии контроллера SCSI все подключенные к нему устройства войдут в семейство этого контроллера.
Пристрої по підключенню	Отображение состояния всех распределенных ресурсов по типу устройства, использующего эти ресурсы. Это могут быть каналы прямого доступа к памяти (DMA), порты ввода-вывода, запросы на прерывание (IRQ) и адреса памяти.
Ресурси по типу	Отображение состояния всех распределенных ресурсов по типу подключения. Это могут быть: каналы DMA, порты ввода-вывода и адреса памяти.
Ресурси по підключенню	Отображение состояния всех распределенных ресурсов по типу подключения. Это могут быть: каналы DMA, порты ввода-вывода и адреса памяти.

### 3.7.2 Тестові програми

Крім штатних засобів Windows існує багато програм для тестування апаратних засобів комп'ютера. Наприклад, програма Everest, Dr. Hardware, SiSoft Sandra. Вікна відповідних програм зображені на рис. 49, 50.

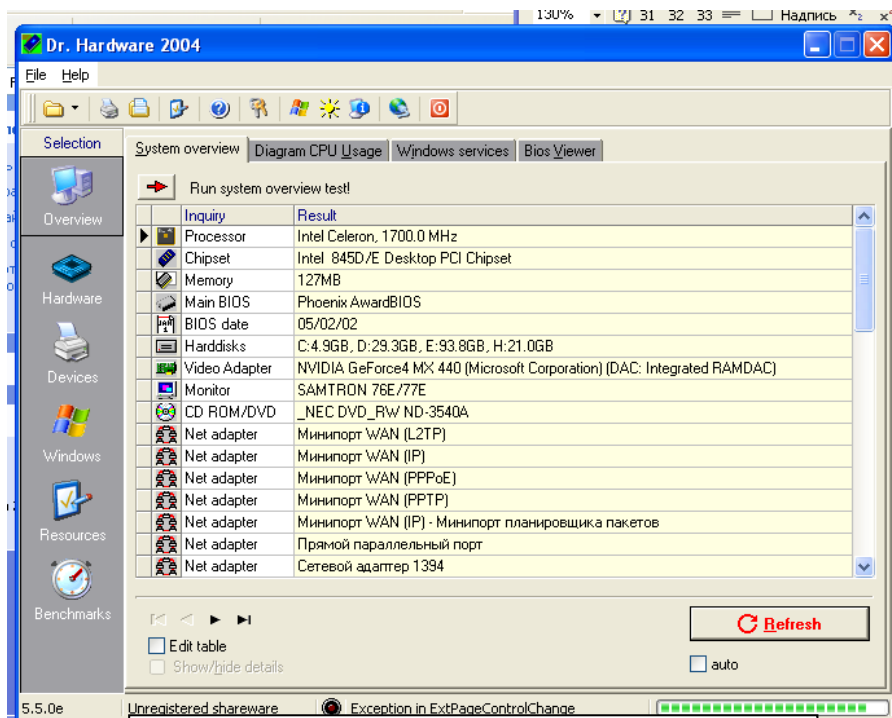


Рис. 49 Вікно програми Dr. Hardware

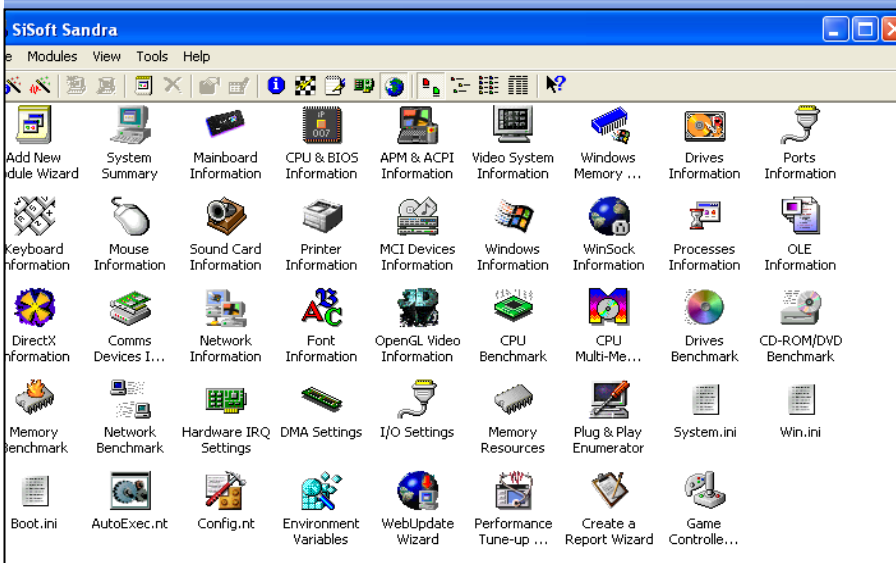


Рис. 50 Вікно програми SiSoft Sandra

### 3.7.3 Програма Debug.exe

Проглядання вмісту пам'яті можна провести за допомогою програми Debug.exe, вона є в будь-якому комп'ютері і легко запускається з командного рядка. Введенням знаку ? можна вивести на екран і проглянути список всіх команд цієї програми:

```

C:\WINDOWS\system32\debug.exe
-?
assemble      A [адрес]
compare       C диапазон адрес
dump          D [диапазон]
enter         E адрес [список]
fill          F диапазон список
go            G [=адрес] [адреса]
hex           H значение1 значение2
input         I порт
load          L [адрес] [диск] [первый_сектор] [число]
move         M диапазон адрес
name          N [путь] [список_аргументов]
output        O порт байт
proceed       P [=адрес] [число]
quit          Q
register       R [регистр]
search        S диапазон список
trace         T [=адрес] [значение]
unassemble    U [диапазон]
write         W [адрес] [диск] [первый_сектор] [число]
выделение памяти EMS      XA [#число_страниц]
освобождение памяти EMS  XD [дескриптор]
сопоставление страниц EMS XM [Lстраница] [Rстраница] [дескриптор]
вывод состояния памяти EMS XS
-

```

Наприклад, як показано нижче, команда `r` виводить вміст всіх регістрів, а команда `r ax` тільки регістра `AX`.

```

r ax
AX 0000
:1111
-r
AX=1111 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1525 ES=1525 SS=1525 CS=1525 IP=0100 NU UP EI PL NZ NA PO NC
1525:0100 0000          ADD     [BX+SI],AL          DS:0000=CD

```

По команді `d [адреса]` виводиться вміст 128 елементів пам'яті починаючи із заданої адреси, а також інтерпретація вмісту в кодах ASCII. Адреса задається з вказівкою сегменту і зсуву.

```

-d 0000:1C80
0000:1C80 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0000:1C90 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0000:1CA0 00 00 43 3A 5C 57 49 4E-44 4F 57 53 5C 53 59 53 ...C:\WINDOWS\SYS
0000:1CB0 54 45 4D 33 32 5C 43 4F-55 4E 54 52 59 2E 53 59 TEM32\COUNTRY.SY
0000:1CC0 53 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 S.....
0000:1CD0 00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0000:1CE0 00 00 B5 01 06 00 02 AF-0A A7 00 04 31 0B A7 00 .....1...
0000:1CF0 05 B3 0B A7 00 06 E3 0B-A7 00 07 E5 0C A7 00 01 .....

```

Якщо вміст пам'яті є програмою, то її можна подивитися, провівши дисасемблювання командою U. Щоб знайти якусь програму в пам'яті можна викликати список всіх програм командою MEM /D /P.

```

u0000:11f1
0000:11F1 0000      ADD     [BX+SI ],AL
0000:11F3 0000      ADD     [BX+SI ],AL
0000:11F5 00A705E2    ADD     [BX+E205 ],AH
0000:11F9 0201      ADD     AL,[BX+DI ]
0000:11FB 00E2      ADD     DL,AH
0000:11FD 024701    ADD     AL,[BX+01 ]
0000:1200 E202      LOOP   1204
0000:1202 46        INC     SI
0000:1203 30FF      XOR     BH,BH
0000:1205 FF00      INC     WORD PTR [BX+SI ]
0000:1207 00980088  ADD     [BX+SI +8800 ],BL
0000:120B 0000      ADD     [BX+SI ],AL
0000:120D 40        INC     AX
0000:120E 56        PUSH   SI
0000:120F 0F        DB     0F
0000:1210 1F        POP    DS

```

Вміст пам'яті можна проглянути і змінити командою E. Клавішею пропуск перемикаються комірки пам'яті.

```

e0000:11f1
0000:11F1 00.00 00. 00. 00. 00. A7. 05.
0000:11F8 E2. 02. 01. 00. E2. 02. 47. 01.
0000:1200 E2. _

```

### 3.8 Профілактика і обслуговування комп'ютера

Комп'ютери, як правило, працюють цілком надійно, але рано чи пізно виходять з ладу. У перебігу терміну служби відповідним регулярним профілактичним обслуговуванням можна забезпечити якісну, надійну і тривалу його роботу.

Обслуговування полягає в забезпеченні необхідних умов роботи, ремонті і заміні апаратних засобів, установці і оновленні програмних засобів. Для забезпечення нормальних умов роботи необхідно враховувати наступні зовнішні чинники:

1. Температурні умови.
2. Пил і забруднення робочих поверхонь.
3. Електромагнітні умови.
4. Якість електроживлення .
5. Вплив хімічних речовин, механічних і інших факторів.

Основними тепловиділяючими елементами є процесор, блок живлення, відеоадаптер, вінчестер. Перегрів цих пристроїв може привести до їх поломки, тому на них в першу чергу встановлюються вентилятори (кулери).



Необхідно проводити наступні заходи щодо забезпечення температурного режиму:

- Проводити діагностику і контролювати тепловий режим основних компонентів.
- Контролювати справність вентиляторів і звільняти від пилу вентиляційні отвори.
- Додавати вентилятори при нестійкій роботі системи.
- Регулярно видаляти пил зовні і усередині комп'ютера.
- Не експлуатувати апаратуру в жарких приміщеннях і біля опалювальних приладів.
- Не зберігати і не експлуатувати устаткування при температурах +5 і нижче.

Пил і забруднення впливають через підвищення теплоізоляції, окислювання контактів, абразивна дія на рухомі поверхні, порушення контактів, помилки при читанні інформації з магнітних носіїв. Пил легко всмоктується повітряним потоком і притягується електричними полями. Треба застосовувати наступні заходи профілактики:

- Не експлуатувати устаткування в запилених, вологих і виробничих приміщеннях, використовувати всі доступні заходи боротьби з пилом.
- Не експлуатувати устаткування в приміщеннях з високим рівнем електро-магнітних полів.
- Не торкатися пальцями поверхні магнітних дисків, екрану монітора, використовувати захисні чохла.
- Використовувати для очищення комп'ютера від пилу спеціальні очищаючі і антистатичні засоби.

Електромагнітні поля можуть викликати перешкоди і збої в роботі пам'яті і інших пристроїв, руйнування даних на магнітних носіях, спотворення даних в кабелях, вихід з ладу мікросхем. Дуже небезпечні електростатичні розряди, які можуть руйнувати окремі елементи або викликати помилки в двійкових кодах. Не можна розміщувати комп'ютер поблизу джерел могутніх електромагнітних полів: потужних двигунів, потужної електропроводки, електромагнітів, радіоустановок, блоків живлення.

Стрибки і пропажа напруги у мережі живлення можуть визвати руйнування візлів комп'ютера. Для захисту від них необхідно встановлювати мережеві фільтри і блоки резервного живлення (UPS).

Хімічній дії агресивних речовин (окисленню) піддаються механічні контакти в роз'ємах і мікросхемах. При цьому виникають пошкодження сигналів і помилки. Порушення контактів є найбільш частою несправністю електронних приладів. Заходи боротьби з окисленням:

- Не експлуатувати устаткування в приміщеннях з агресивним і вологим середовищем.

– Видаляти окислення спеціальними неабразивними чистячими засобами або спеціальними розчинами.

Найменш надійні блоки у будь-якій системі це ті, які містять пересувні механічні елементи. У комп'ютері такими блоками є дисководи і найбільш відповідальний з них - вінчестер. Для страхування інформації необхідно регулярно виконувати резервне копіювання файлів.

## **Лабораторні роботи**

### **Лабораторна робота №1.**

Визначення конфігурації і загальної інформації про комп'ютер.

Мета роботи: Навчитися одержувати загальну інформацію про конфігурацію комп'ютера.

Для отримання інформації використовувати штатні засоби операційної системи і інші допоміжні утиліти.

Завдання:

1. Відкрити Панель управління і запустити утиліту Система (для Windows XP в групі Продуктивність і обслуговування).
2. Визначите основні параметри комп'ютера (які можна вважати основними?)
3. Використовуйте інші допоміжні утиліти (які ще бувають?)
4. Запишіть конфігурацію комп'ютера у вигляді короткого запису.
5. Поясніть зроблений запис.
6. Оформите звіт про виконання лабораторної роботи.

### **Лабораторна робота №2.**

Контроль і настройка параметрів BIOS.

**УВАГА!** Після зміни параметрів і завершення лабораторної роботи слід закрити програму без збереження змін.

Мета роботи: Навчитися одержувати загальну інформацію про параметри конфігурації комп'ютера, змінювати їх і налаштовувати режими роботи компонентів комп'ютера.

Для отримання інформації і настройки використовувати штатні засоби базової системи введення-виведення – Setup BIOS.

Завдання:

1. Презавантажити комп'ютер кнопками Ctrl+Alt+Del.
2. При появі на екрані запрошення нажимати кнопку Del .
3. Виділите по черзі всі категорії і прочитайте їх вміст в нижній частині вікна.
4. Знайдіть, де встановлюються час і дата. Увійдіть до цього меню. Виділите по черзі всі рядки – команди і прочитайте їх призначення в правій частині вікна.

5. Знайдіть категорію з установками черговості завантаження. Спробуйте зрадити черговість. Для чого це зроблено?

6. Знайдіть категорію з інформацією про напругу і температурні режими. Відкрийте її і вивчіть інформацію, що є там. Що можна сказати про режими комп'ютера?

7. Які ще категорії привернули вашу увагу? Чому?

8. Закрийте програму без збереження змін.

9. Поясніть отримані результати.

10. Оформіть звіт про виконання лабораторної роботи.

### **Лабораторна робота №3.**

Тестування пам'яті.

Мета роботи: Навчитися одержувати інформацію про вміст пам'яті і змінювати його при необхідності.

Для отримання роботи використовувати штатні засоби операційної системи утиліту DEBUG.EXE.

Завдання:

1. Вивчіть опис утиліти DEBUG.EXE у розділі .

2. У меню Пуск командою Виконати відкрити вікно Запуск програми для введення командного рядка.

3. Ввести DEBUG.EXE і нажирати кнопку ОК.

4. Ввести знак ? і нажирати клавішу Enter.

5. Вивчіть склад команд цієї утиліти.

6. Проглянете зміст довільної області пам'яті командою D.

7. Проглянете зміст регістрів процесора.

8. Введіть які-небудь дані в елементи пам'яті.

9. Знайдіть в пам'яті область з програмним кодом і дисасемблюйте частина коду.

10. Поясніть отримані результати.

11. Оформіть звіт про виконання лабораторної роботи.

### **Рекомендована література**

1. С. мюллер Модернізація і ремонт ПК, 16-е изд.: Пер з англ. - М.: Изд.дом «Вільямс», 2006.-1328.

2. У. Л. Бройдо, О.П. Ільїна. Архітектура ЕОМ і систем: Підручник для вузів.-СПб.:Питер, 2006.-718с.

3. П. Нортона, Д. М. Гудмен. Внутрішній світ персональних комп'ютерів. Пер. з англ.- К.: Діасофт, 2003.

4. А. Траськовській. BIOS. Експрес - курс. - Спб.: БХВ - Петербург. 2005. – 288 с.

5. Е. М. Міхлін. Як самому зібрати, набудувати і модернізувати ПК.: М.: ТОВ Діасофтнп, 2005.-544 с.

6. Дж. Уокерлі. Архітектура і програмування ЕОМ. М.:Мир.- 1984.

7. А.в.палагин, Ю.с.яковльов. Системна інтеграція засобів комп'ютерної техніки. Монографія. - Вінниця: „УНІВЕРСУМ-Вінниця”, 2005.-680 с.

8. Енциклопедія кібернетики: у 2 т. – До: - Головна редакція Української Радянської енциклопедії, 1973.