Розділ 2 - Мікропроцесор

2.1 Функції мікропроцесору

2.2 Класифікація мікропроцесорів

2.3 Архітектура мікропроцесорів

2.3.1 Типи архітектур

2.3.2 Структура типового МП

2.4 Параметри процесорів

2.4.1 Швидкодія процесору

2.4.2 Розрядність процесора

Шина даних

Шина адреси

Внутрішні регістри

2.4.3 Режими процесора

Реальний режим

Захищений режим

Віртуальний реальний режим

Ключові терміни

мікропроцесор, мікроконтролер, мікроархітектура мікропроцесора, макроархітектура макропроцесора, внутрішня реєстрова пам'ять процесора, переривання, шина даних, шина адреси, регістр

"Мізком" персонального комп'ютера є мікропроцесор, або центральний процесор - CPU (Central Processing Unit). Мікропроцесор виконує обчислення і обробку даних (за винятком деяких математичних операцій, здійснюваних в комп'ютерах, що мають співпроцесор) і, як правило, є найдорожчою мікросхемою комп'ютера. У всіх PC-сумісних комп'ютерах використовуються процесори, сумісні з сімейством мікросхем Intel, але випускаються і проектуються вони як самою Intel, так і компаніями AMD, Cyrix, IDT і Rise Technologies.

**Мікропроцесор** ‒ це центральний блок персонального комп'ютера, призначений для управління роботою всіх інших блоків і виконання арифметичних і логічних операцій над інформацією.

**2.1 Функції мікропроцесору**

Мікропроцесор виконує такі основні функції:

- читання і дешифрування команд з основної пам'яті;

- читання даних з основної пам'яті і регістрів адаптерів зовнішніх пристроїв;

- прийом та обробку запитів і команд від адаптерів на обслуговування зовнішніх пристроїв;

- обробку даних і їх запис в основну пам'ять і регістри адаптерів зовнішніх пристроїв;

- вироблення керуючих сигналів для всіх інших вузлів і блоків мікропроцесорної системи.

**2.2 Класифікація мікропроцесорів**

Всі мікропроцесори поділяють на окремі класи відповідно до їх архітектури, структури і функціонального призначення. На рис. 2.1 – ви бачите схему класифікації МП за функціональним призначенням.

**Мікропроцесори загального призначення** призначені для вирішення широкого кола завдань обробки різноманітної інформації. Їх основною областю використання є персональні комп'ютери, робочі станції, сервери та інші цифрові системи масового застосування.

**Спеціалізовані мікропроцесори** орієнтовані на вирішення специфічних завдань управління різними об'єктами. Містять додаткові мікросхеми (інтерфейси), що забезпечують спеціалізоване використання. Мають особливу конструкцію, підвищену надійність.

**Мікроконтролери** є спеціалізованими мікропроцесорами, які орієнтовані на реалізацію пристроїв керування, вбудованих у різноманітну апаратуру. Характерною особливістю структури мікроконтролерів є розміщення на одному кристалі з центральним процесором внутрішньої пам'яті і великого набору периферійних пристроїв.

**Цифрові процесори сигналів** (ЦПС) представляють клас спеціалізованих мікропроцесорів, орієнтованих на цифрову обробку вхідних аналогових сигналів. Специфічною особливістю алгоритмів обробки аналогових сигналів є необхідність послідовного виконання ряду команд множення-підсумовування з накопиченням проміжного результату в регістрі-акумуляторі. Тому архітектура ЦПС орієнтована на реалізацію швидкого виконання операцій такого роду. Набір команд цих процесорів містить спеціальні команди MAC (Multiplication with Accumlation), які реалізують ці операції.



Рисунок 2.1 – Класифікація мікропроцесорів по функціональному призначенню

**2.3 Архітектура мікропроцесорів**

**Мікроархітектура мікропроцесора** ‒ це апаратна організація і логічна структура мікропроцесора, регістри, керуючі схеми, арифметико-логічні пристрої, запам’ятовуючі пристрої і пристрої, які зв'язують їхні інформаційні магістралі.

**Макроархітектура макропроцесора** ‒ це система команд, типи оброблюваних даних, режими адресації і принципи роботи мікропроцесора.

**2.3.1 Типи архітектур**

При описі архітектури та функціонування процесора звичайно використовується його подання у вигляді сукупності програмно- доступних регістрів, що утворюють реєстрову або програмну модель. У цих регістрах містяться оброблювані дані (операнди) і керуюча інформація. Відповідно, в реєстрову модель входить група регістрів загального призначення, службовців для зберігання операндів, і група службових регістрів, що забезпечують управління виконанням програми і режимом роботи процесора, організацію звернення до пам'яті (захист пам'яті, сегментна і сторінкова організація та ін.).

Регістри загального призначення утворюють РЗП ‒ **внутрішню реєстрову пам'ять процесора**. Склад і кількість службових регістрів визначається архітектурою мікропроцесора. Зазвичай в їх склад входять:

\* Програмний лічильник PC (або CS + IP в архітектурі мікропроцесорів Intel);

\* Регістр стану SR (або EFLAGS);

\* Регістри управління режимом роботи процесора CR (Control Register);

\* Регістри, що реалізують сегментну і сторінкову організацію пам'яті;

\* Регістри, що забезпечують налагодження програм і тестування процесора.

Склад пристроїв і блоків, що входять в структуру мікропроцесора, і реалізуються механізми їх взаємодії визначаються функціональним призначенням і областю застосування мікропроцесора.

Архітектура та структура мікропроцесора тісно взаємопов'язані. Реалізація тих чи інших архітектурних особливостей вимагає введення в структуру мікропроцесора необхідних апаратних засобів (пристроїв і блоків) і забезпечення відповідних механізмів їх спільного функціонування. У сучасних мікропроцесорах реалізуються наступні варіанти архітектур.

Всі мікропроцесори можна розділити на групи:

- мікропроцесори типу CISC з повним набором системи команд;

- мікропроцесори типу RISC з усіченим набором системи команд;

- мікропроцесори типу VLIW з надвеликим командним словом;

- мікропроцесори типу MISC з мінімальним набором системи команд і вельми високою швидкодією та ін.

Основною відмінністю процесорів з CISC-архітектурою та RISC-архітектурою є спосіб звернення до памяті команд.

**CISC** (Complex Instruction Set Computer) ‒ архітектура реалізована в багатьох типах мікропроцесорів, що виконують великий набір різноформатних команд з використанням численних способів адресації. Вони виконують більше 200 команд різного ступеня складності, які мають розмір від 1 до 15 байт і забезпечують більше 10 різних способів адресації. Таке велике різноманіття виконуваних команд і способів адресації дозволяє програмісту реалізувати найбільш ефективні алгоритми вирішення різних завдань. Проте при цьому суттєво ускладнюється структура мікропроцесора, особливо його пристрій управління, що приводить до збільшення розмірів і вартості кристалу, зменшенню продуктивності. У той же час багато команд і способів адресації використовуються досить рідко. Також аналіз кодів програм, які генеруються компіляторами мов вищого рівня, показав, що компілятори з усієї системи команд МП використовують тільки обмежений набір простих команд. Це команди типу "регістр-регістр", "регістр-пам'ять".

**RISC** (Reduced Instruction Set Computer) ‒ архітектура відрізняється використанням обмеженого набору команд фіксованого формату. Сучасні RISC ‒ процесори зазвичай реалізують близько 100 команд, що мають фіксований формат довжиною 4 байта. Також значно скорочується число використовуваних способів адресації. Зазвичай в RISC - процесорах всі команди обробки даних виконуються тільки з реєстрової або безпосередньою адресацією. Для зменшення кількості звертань до пам'яті RISC-процесори мають збільшений об'єм внутрішніх регістрів - від 32 до декількох сотень, тоді як в CISC-процесорах кількість регістрів загального призначення переважно становить 8-16.

**VLIW** (Very Large Instruction Word) ‒ архітектура з'явилася відносно недавно - в 1990 -х роках. Її особливістю є використання дуже довгих команд (до 128 біт і більше), окремі поля яких містять коди, що забезпечують виконання різних операцій. Таким чином, одна команда викликає виконання відразу декількох операцій паралельно в різних операційних пристроях, що входять в структуру мікропроцесора.

**MISC** (англ. minimal instruction set computer — «комп'ютер з мінімальним набором команд»). Процесори з MISC, як і процесори RISC, характеризуються невеликим числом команд, які найчастіше зустрічаються. Разом з цим принцип «дуже довгих командних слів» (VLIW) забезпечує виконання групи несуперечливих команд за один цикл роботи процесора. Порядок виконання команд встановлюється таким чином, щоб досягти максимального завантаження маршрутів, за якими проходять потоки інструкцій та даних.

Крім набору виконуваних команд і способів адресації важливою архітектурною особливістю мікропроцесорів є варіант реалізації пам'яті, який використовується, і організація вибірки команд і даних. За цими ознаками розрізняються процесори з Принстонскою і Гарвардською архітектурою.



Рисунок 2.2 - Основні типи архітектури:

*а* — фоннейманівська; *6* — гарвардська

**Принстонська архітектура**, яка часто називається архітектурою **Фон - Неймана**, характеризується використанням загальної оперативної пам'яті для зберігання програм, даних, а також для організації стека. Для звернення до цієї пам'яті використовується загальна системна шина, по якій в процесор надходять і команди, і дані. Ця архітектура має ряд важливих переваг. Наявність загальної пам'яті дозволяє оперативно перерозподіляти її обсяг для зберігання окремих масивів команд, даних і реалізації стека в залежності від розв'язуваних завдань. Використання спільної шини для передачі команд і даних значно спрощує відладку, тестування і поточний контроль функціонування системи, збільшує її надійність. Тому Принстонська архітектура протягом довгого часу домінувала в обчислювальній техніці.

Основний недолік - необхідність послідовної вибірки команд і даних по спільній системній шині. При цьому шина стає вузьким місцем, яке обмежує продуктивність системи. Постійно зростаючі вимоги до продуктивності мікропроцесорних систем викликали в останні роки більш широке застосування Гарвардської архітектури при створенні багатьох типів сучасних мікропроцесорів.

**Гарвардська архітектура** характеризується фізичним поділом пам'яті команд (програм) і пам'яті даних. У її оригінальному варіанті використовувався також окремий стек для зберігання вмісту програмного лічильника, який забезпечував можливості виконання вкладених підпрограм. Кожна пам'ять з'єднується з процесором окремої шиною, що дозволяє одночасно з читанням/записом даних при виконанні поточної команди робити вибірку і декодування наступної команди. Завдяки такому поділу потоків команд і даних і поєднанню операцій їх вибірки реалізується більш висока продуктивність, ніж при використанні Принстонской архітектури.

Недоліки Гарвардської архітектури пов'язанні з необхідністю проведення більшого числа шин, а також з фіксованим об'ємом пам'ті, виділеної для команд і даних, призначення якої не може оперативно пеперозподілятися. Тому потрібно використовувати пам'ять більшого об'єму, коефіцієнт використання якої при вирішенні різноманітних задач виявляється нижчим, ніж в системах з Прінстонською архітектурою. Проте розвиток мікроелектроніки дозволив в значній мірі подолати вказані недоліки, тому Гарвардська архітектура широко застосовується у внутрішній структурі сучасних мікропроцесорів, де використовується окрема кеш-пам'ять для зберігання команд і даних. Разом з цим, у зовнішній структурі більшості мікропроцесорних систем реалізуються принципи Прінстонської архітектури.

Гарвардська архітектура отримала також широке застосування в мікроконтролерах та цифрових сигнальних процесорах.

**2.3.2 Структура типового МП**

До складу мікропроцесора входять такі пристрої.

Схема процесора містить пристрій керування, АЛП і ре­гістри: адреси, даних, команд, а також стану, акумулятор, лічиль­ник команд, покажчик стеку.

***Пристрій керування*** відповідно до кодів команд і зовнішніх керуючих сигналів та сигналів синхронізації фор­мує керуючі сигнали для всіх блоків структурної схеми МП, а також керує обміном інформацією між МП, пам'яттю і ПВВ. Пристрій керування реалізує такі функції: початкового вста­новлення МП, синхронізації, переривань, узгодження швид­кодії модулів МП системи.

*Функція початкового встановлення МП.* Зовнішній сигнал початкового встановлення процесора *RESET* формується під час ввімкнення джерела живлення МП або під час натискан­ня кнопки *RESET.* Після появи цього сигналу пристрій керування забезпечує завантаження нульового значення у про­грамний лічильник, що ініціює вибирання з пам'яті байта команди з нульовою адресою. Наприкінці вибирання вміст лічильника команд збільшується на одиницю, після чого ви­бирається байт команд з наступною адресою. Так виконуєть­ся вся записана у пам'яті програма.



Рис. 1.7. Структур­на схема з процесо­ром фоннейманівської архітектури

*Функція синхронізації.* Згідно із зовнішніми керуючими сигналами і сигналами синхросигналізації пристрій керуван­ня синхронізує роботу всіх блоків МП.

*Функція переривань.* З надходженням сигналу перери­вання пристрій керування ініціює роботу підпрограми об­робки відповідного переривання. Потреба у реалізації функцій переривань виникає тоді, коли під час виконання основної програми треба перевести МП на розв'язання іншої задачі, наприклад оброблення аварійної ситуації або роботи з ПВВ.

*Функція узгодження швидкодії модулів мікропроцесор­ної системи.* Під час обслуговування пам'яті та **ПВВ** із значно меншою швидкодією, ніж МП, узгодження швидкодії вирі­шується генерацією тактів очікування МП, а під час обслуго­вування пристроїв з більшою швидкодією, ніж МП, викорис­товується режим безпосереднього доступу до пам'яті.

***Арифметико-логічний пристрій*** (АЛП) - комбіна­ційна схема на основі суматора, який сигналами з виходів пристрою керування налагоджується на виконання певної арифметичної або логічної операції — додавання, відніман­ня, ЛОГІЧНЕ І, ЛОГІЧНЕ АБО, логічне НІ, ВИКЛЮЧАЛЬНЕ АБО, зсуву, порівняння, десяткової корекції. Отже, АЛП виконує арифметичні або логічні операції над операндами, які пересилаються з пам'яті і (або) регістрів МП.

***Операнд*** - це об'єкт у вигляді значення даних, вмісту регістрів або вмісту комірки пам'яті, з яким оперує команда, наприклад, у команді додавання операндами є доданки. Опе­ранд може задаватися у команді у вигляді числа або знахо­дитися в регістрі чи комірці пам'яті. Отриманий після вико­нання команди в АЛП результат пересилається в регістр або комірку пам'яті.

***Регістри*** призначені для зберігання га-розрядного двійко­вого числа. Це *п* тригерів зі схемами керування читанням-записом та вибірки. Регістри створюють внутрішню пам'ять МП і використовуються для зберігання проміжних резуль­татів обчислень.

***Акумулятор*** — регістр, в якому зберігається один з операндів. Після виконання команди в акумуляторі замість операнду розміщується результат операції. У 8-розрядних процесорах акумулятор бере участь у всіх операціях АЛП пристрою. Однак у 16-розрядних МП більшість команд ви­конуються без участі акумулятора, але в деяких командах (введення, виведення, множення, ділення) акумулятор діє так само, як і в 8-розрядних МП, тобто зберігає один з операндів, а після виконання команди - результат операції.

***Лічильник команд,*** або ***програмований лічильник,*** призначений для зберігання адреси комірки пам'яті, яка містить код наступної команди. Програма дій МП записана в пам'яті у вигляді послідовності кодів команд. Для переходу до наступної команди вміст лічильника збільшується на оди­ницю у момент вибирання команди з пам'яті. Отже, наприкінці виконання команди в лічильнику команд зберігається адреса наступної команди.

***Покажчик стеку*** — це регістр, який зберігає адресу останньої зайнятої комірки стеку. *Стеком,* або *степовою па­м'яттю,* називають область пам'яті, яка організована за прин­ципом «останній прийшов — перший пішов».

***Регістр команд*** зберігає код команди впродовж усього часу виконання команди.

***Регістр адреси*** і ***регістри даних*** призначені для збе­рігання адрес та даних, що використовуються під час вико­нання поточної команди в МП.

***Регістр стану,*** або ***регістр прапорців*** (ознак), при­значений для зберігання інформації про результат операції в АЛП і складається з кількох тригерів, які набувають одинич­них або нульових значень. Наприклад, прапорець нуля встановлюється у стан логічної одиниці за нульового результату операції.



Рисунок 2.3 – Архітектура типового мікропроцесору

**2.4 Параметри процесорів**

Робота процесора полягає і в послідовному виконанні команд з оперативної пам'яті, і в швидкості виконання команд. Чим швидше процесор виконує команди, тим вища продуктивність комп'ютера в цілому. Швидкість роботи процесора залежить від декількох параметрів (швидкодія, розрядність, режими роботи, тощо ).

**2.4.1 Швидкодія процесору**

Швидкодія процесора ‒ досить простий параметр. Вона вимірюється в мегагерцах (МГц); 1 МГц дорівнює мільйону тактів в секунду. Чим вище швидкодія, тим краще (тим швидше процесор). Тактова частота визначає максимальний час виконання перемикання між елементами ЕОМ.

Реальна частота роботи ядра процесора може становити 1,5 – 4 ГГц. Тактова частота визначається множенням частоти зовнішньої шини процесора на коефіцієнт множення. Зовнішня шина використовується для обміну даними з іншими пристроями і може мати позначення FSB ‒Front Side Bus‒ (наприклад: для процесора Intel Core 2DUO E6600 частота FSB - 266,6 МГц, множник - 9, в результаті тактова частота буде рівна 2400 МГЦ)

**2.4.2 Розрядність процесора**

Розрядність процесора визначає максимальну кількість двійкових розрядів, які можуть бути оброблені одночасно. У процесор входить три важливих пристрої, основною характеристикою яких є розрядність:

- шина вводу і виводу даних;

- внутрішні регістри;

- шина адреси пам'яті.

**Шина даних**

Коли говорять про шину процесора, найчастіше мають на увазі шину даних, представлену як набір з'єднань (або виводів) для передачі або прийому даних. Чим більше сигналів одночасно надходить на шину, тим більше даних передається по ній за певний інтервал часу і тим швидше вона працює.

Дані в комп'ютері передаються у вигляді цифр через однакові проміжки часу. Для передачі одиничного біта даних в певний часовий інтервал посилається сигнал напруги високого рівня (близько 5 В), а для передачі нульового біта даних ‒ сигнал напруги низького рівня (близько 0 В).

**Шина адреси**

Шина адреси являє собою набір провідників, по яким передається адреса комірки пам'яті, в яку або з якої пересилаються дані. Як і в шині даних, по кожному провіднику передається один біт адреси, відповідний одній цифрі в адресі. Збільшення кількості провідників (розрядів), які використовуються для формування адреси, дозволяє збільшити кількість адресованих осередків. Розрядність шини адреси визначає максимальний обсяг пам'яті, що адресується процесором.

Шини даних і адреси незалежні, і розробники мікросхем вибирають їх розрядність на свій розсуд, але, чим більше розрядів в шині даних, тим більше їх і в шині адреси. Розрядність цих шин є показником можливостей процесора: кількість розрядів в шині даних визначає здатність процесора обмінюватися інформацією, а розрядність шини адреси ‒ об'єм пам'яті, з яким він може працювати.

**Внутрішні регістри**

Кількість бітів даних, які може обробити процесор за один прийом, характеризується розрядністю внутрішніх регістрів. Регістр ‒ це, по суті, елемент пам'яті всередині процесора; наприклад, процесор може складати числа, записані в двох різних регістрах, а результат зберігати в третьому регістрі. Розрядність регістра визначає кількість розрядів оброблюваних процесором даних, а також характеристики програмного забезпечення і команд, які виконуються чіпом.

**2.4.3 Режими процесора**

Режими процесора призначені для виконання програм у різних середовищах; в різних режимах можливості чіпа неоднакові, тому що команди виконуються по-різному. В залежності від режиму процесора змінюється схема управління пам'яттю системи і завданнями.

***Процесори можуть працювати в трьох режимах***: реальному, захищеному і віртуальному реальному режимі (реальному всередині захищеного).

**Реальний режим**

У першому IBM PC використовувався процесор 8088, який міг виконувати 16-розрядні команди, застосовуючи 16-розрядні внутрішні регістри, і адресувати тільки 1 Мбайт пам'яті, використовуючи 20 розрядів для адреси. Все програмне забезпечення PC спочатку було призначено для цього процесора; воно було розроблено на основі 16-розрядної системи команд і моделі пам'яті об'ємом 1 Мбайт. Наприклад, DOS, все програмне забезпечення DOS, Windows від 1.x до 3.x і всі додатки для Windows від 1.x до 3.x написані в розрахунку на 16-розрядні команди. Ці 16-розрядні операційні системи і додатки були розроблені для виконання на первинному процесорі.

Всі програми, що виконуються в реальному режимі, повинні використовувати тільки 16-розрядні команди, 20-розрядні адреси і підтримуватися архітектурою пам'яті, розрахованою на ємність до 1 Мбайт.

Для програмного забезпечення цього типу зазвичай використовується однозадачний режим, тобто одночасно може виконуватися тільки одна програма. Немає ніякого вбудованого захисту для запобігання перезапису елементів пам'яті однієї програми або навіть операційної системи іншою програмою; це означає, що при виконанні декількох програм цілком можуть бути зіпсовані дані або код однієї з них, а це може привести всю систему до краху (або зупинки).

**Захищений режим**

Першим 32-розрядним процесором, призначеним для PC, був 386. Цей чіп міг виконувати абсолютно нову 32-розрядну систему команд. Щоб повністю використовувати перевагу 32-розрядної системи команд, були необхідні 32-розрядна операційна система і 32-розрядні додатки. Цей новий режим називався захищеним, тому що працюють у ньому програми захищені від перезапису своїх областей пам'яті іншими програмами. Такий захист робить систему більш надійною, оскільки жодна програма з помилками вже не зможе так легко пошкодити інші програми або операційну систему. Крім того, програму, "потерпілу крах", можна досить просто завершити без збитку для всієї системи.

Захищений режим має багато переваг:

У захищеному режимі доступна вся системна пам'ять ( не існує межі 1 Мбайт).

У захищеному режимі операційна система може організувати одночасне виконання декількох завдань ( багатозадачність ) .

У захищеному режимі підтримується віртуальна пам'ять - операційна система при необхідності може використовувати жорсткий диск в якості розширення оперативної пам'яті.

У захищеному режимі здійснюється швидкий ( 32/64-розрядних ) доступ до пам'яті і підтримується робота 32 -х розрядних операцій введення-виведення

**Віртуальний реальний режим**

Для зворотної сумісності 32-розрядна система Windows 9x використовує третій режим в процесорі ‒ віртуальний реальний режим. Віртуальний реальний, по суті, є режимом виконання 16-розрядної середовища (реальний режим), яке реалізовано всередині 32-розрядного захищеного режиму (тобто віртуально, а не реально). Виконуючи команди у вікні підказки DOS усередині Windows, ви створюєте віртуальний сеанс реального режиму. Оскільки захищений режим є справді багатозадачним, фактично можна виконувати декілька сеансів реального режиму, причому в кожному сеансі власне програмне забезпечення працює на віртуальному комп'ютері. І всі ці додатки можуть виконуватися одночасно, навіть під час роботи інших 32-розрядних програм.

Віртуальне реальне вікно повністю імітує середовище процесора 8088, і, якщо не враховувати швидкодію, програмне забезпечення буде виконуватися так, як воно виконувалося першим PC в реальному режимі. Кожна віртуальна машина отримує власний 1 Мбайт адресного простору і власний екземпляр реальних апаратних підпрограм управління апаратурою (базову систему вводу-виводу), причому при цьому емулюються всі регістри і можливості реального режиму.

Віртуальний реальний режим використовується при виконанні програм у вікні DOS, а також при виконанні 16-розрядних програм, написаних для DOS або Windows. При запуску програми DOS операційна система Windows створює віртуальну машину DOS, на якій цей додаток може виконуватися.

Важливо відзначити, що всі процесори Intel (а також Intel-сумісні AMD і Cyrix) при включенні живлення починають працювати в реальному режимі. При завантаженні 32-розрядна операційна система автоматично перемикає процесор в 32-розрядний режим і управляє ним в цьому режимі.