

Архітектурні особливості сучасних мікропроцесорів

[1 Архітектура 16-розрядних мікропроцесорів](#)

[2 Архітектурні особливості мікропроцесорів наступних поколінь](#)

[3 Напрямки підвищення продуктивності процесорів](#)

1 Архітектура 16-розрядних мікропроцесорів

До 16-розрядних МП першого покоління належать МП i8086/i8088 та i80186/i80188, до МП другого покоління - i80286. Мікропроцесор має внутрішній надоперативний запам'ятовувальний пристрій (НОЗП) ємністю 14x16 байт. Шина адреси 20-розрядна, що дозволяє безпосередньо адресувати до $2^{20} = 1048576$ комірок пам'яті (1 Мбайт).

У МП i8086 застосовано конвеєрну архітектуру, що дозволяє сумістити у часі цикли вибірки команди та вибірки з пам'яті кодів наступних команд. Це досягається паралельною роботою двох порівняно незалежних пристроїв - операційного пристрою та шинного інтерфейсу. Структурну схему МП i8086 показано на рисунку 1. Операційний пристрій виконує команду, а шинний інтерфейс здійснює взаємодію із зовнішньою шиною: виставляє адреси, зчитує коди команд, записує результати обчислень у пам'ять або пристрої введення-виведення.

Операційний пристрій складається з регістрів загального призначення (РЗП), призначених для зберігання проміжних результатів - даних та адрес; АЛП з буферними регістрами; регістра ознак; блока керування та синхронізації (БК та С), який дешифрує коди команд і генерує сигнали керування для всіх блоків схеми МП. Шинний інтерфейс складається з шести байтової регістрової пам'яті, яка називається чергою команд, чотирьох сегментних регістрів: *CS*, *DS*, *ES*, *SS*, вказівника команд *IP*, суматора, а також допоміжних регістрів зв'язку і буфера шин (БШ) адреси/даних. Черга команд працює за принципом *FIFO* (*First Input - First Output*, тобто перший прийшов - перший пішов) і зберігає на виході порядок надходження команд. Довжина черги 6 байт.

Коли операційний пристрій зайнятий виконанням команди, шинний інтерфейс самостійно ініціює випереджаючу вибірку кодів команд з пам'яті у чергу команд. Вибирання з пам'яті чергового командного слова

здійснюється тоді, коли в черзі виявляється два вільні байти. Черга збільшує швидкість процесора у випадку послідовного виконання команд. У разі вибирання команд переходів, викликів і повернень з підпрограм та обробленні запитів переривань черга команд скидається і вибирання починається з нового місця програмної пам'яті.

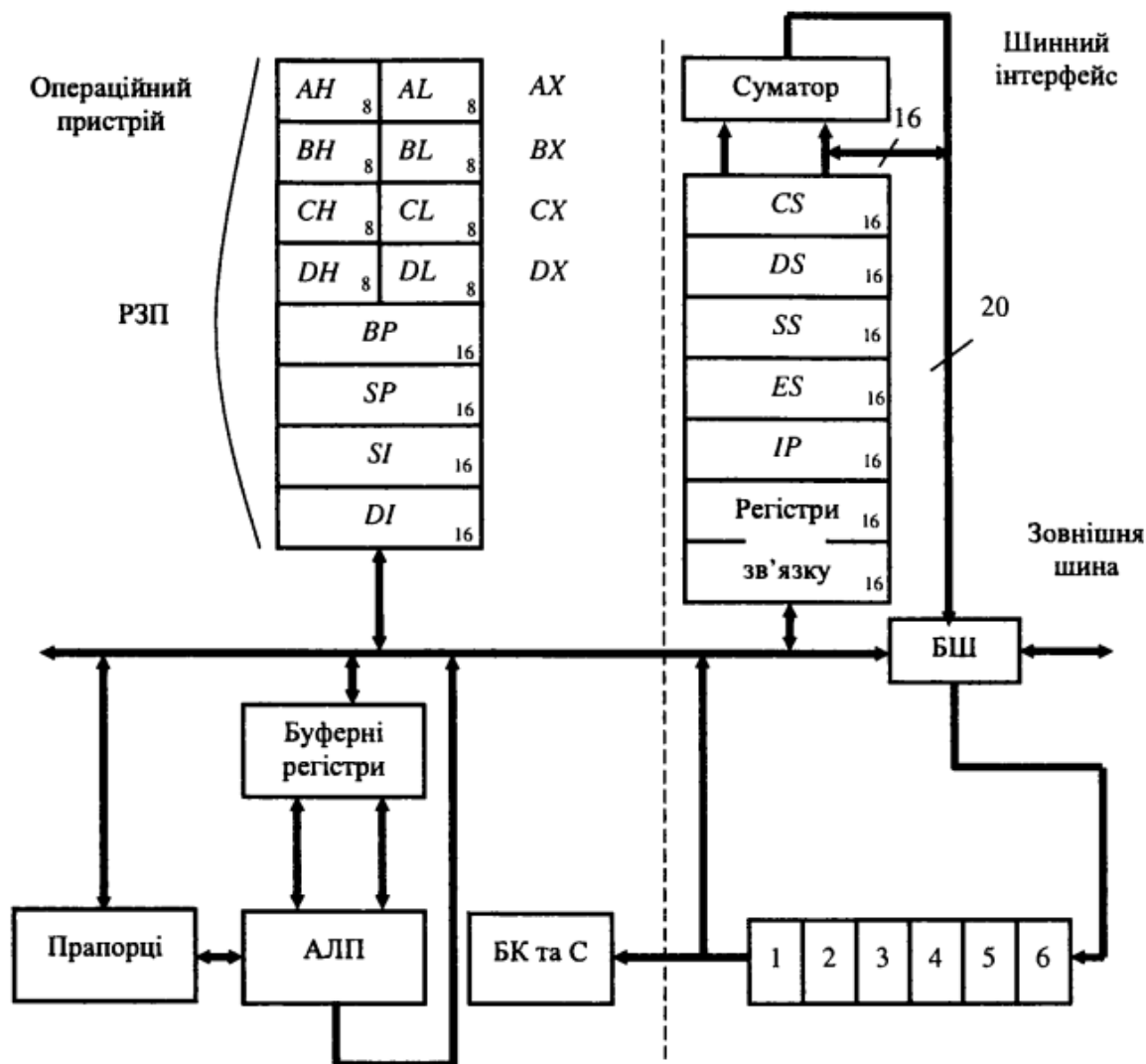


Рисунок 1 – Структурна схема мікропроцесора i8086

Ще одним із завдань шинного інтерфейсу є формування фізичної 20-розрядної адреси із двох 16-розрядних слів. Першим словом є вміст одного з сегментних регістрів *CS*, *SS*, *DS*, *ES*, а друге слово залежить від типу адресації операнда або коду команди. Складання 16-розрядних слів відбувається зі зміщенням на 4 розряди і здійснюється за допомогою суматора, що входить до складу шинного інтерфейсу.

Пам'ять являє собою масив ємністю 1 Мбайт (рисунок 2). У пам'яті зберігаються як байти, так і двобайтові слова. Слова розміщуються у двох

сусідніх комірках пам'яті; старший байт зберігається у комірці зі старшою адресою, молодший - з молодшою. Адресою слова вважається адреса його молодшого байта. На рисунку 2 показано приклад, коли з адресою 00000 зберігається байт 35H, а з адресою 00001 - слово 784AH. Початкові (00000H-003FFFH) і кінцеві (FFFF0H-FFFFFH) адреси зарезервовані для системи переривань та початкового встановлення відповідно.

У МП i8086 застосовано сегментну організацію пам'яті, яка характеризується тим, що програмно доступною є не вся пам'ять, а лише деякі сегменти, тобто області пам'яті. Усередині сегмента використовують лінійну адресацію.

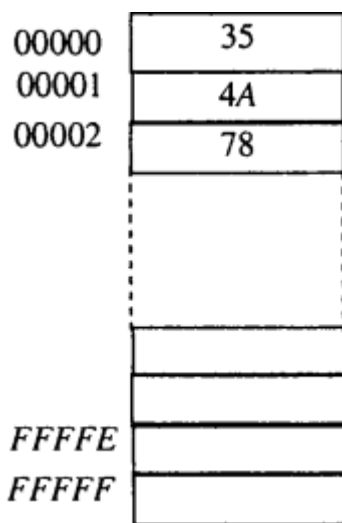


Рисунок 2 – Програмна модель пам'яті

Упровадження сегментної організації можна пояснити таким чином. Мікропроцесор i8086 являє собою 16-розрядний процесор, тобто він має 16-розрядну внутрішню шину, 16-розрядні регістри і суматори. Прагнення розробників ВІС адресувати якомога більший масив пам'яті зумовило використання 20-розрядної шини даних.

Для формування 20-розрядної адреси у 16-розрядному процесорі використовують інформацію двох 16-розрядних регістрів. У МП i8086 20-розрядна адреса формується з двох 16-розрядних адрес, які називають *логічними*. Перша логічна адреса, доповнена праворуч чотирма нулями, являє собою початкову адресу сегмента ємністю 64 кбайт. Друга логічна адреса визначає зміщення у сегменті, тобто відстань від початку сегмента до адресованої комірки. Якщо вона дорівнює 0000, то адресується перша комірка сегмента, якщо FFFFH - то остання. Отже, логічний адресний простір розподілено на блоки суміжних адрес розміром 64 кбайт, тобто сегменти.

Такий підхід до організації пам'яті зручний ще й тому, що пам'ять логічно поділяється на області коду (програмної пам'яті), даних і стека. Фізична 20-розрядна адреса комірки пам'яті формується з двох 16-розрядних адрес – адреси сегмента *Seg* і виконавчої адреси *EA* (*Executive Address*), які додаються зі зміщенням на чотири розряди (рисунок 3).

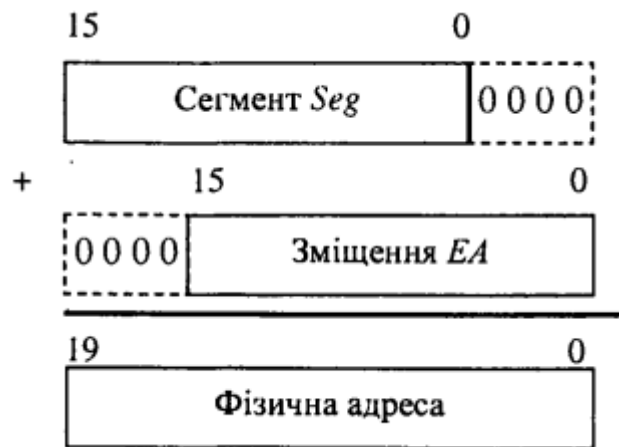


Рисунок 3 – Формування фізичної адреси

Зміщення адреси сегмента на 4 розряди ліворуч еквівалентне його множенню на 2^4 . Тоді фізична адреса дорівнює $16 \times Seg + EA$. Як перша логічна адреса *Seg* використовується вміст одного із чотирьох сегментних регістрів: *CS* (*Code Segment* - сегмент кодів), *DS* (*Data Segment* - сегмент даних), *ES* (*Extended Segment* - додатковий сегмент даних), *SS* (*Stack Segment* - сегмент стека). Друга логічна адреса *EA* або зміщення залежить від сегмента. Так, у сегменті кодів як *EA* використовується вміст лічильника інструкцій *IP*, у сегментах даних значення *EA* залежить від засобу адресації операнда, у сегменті стека використовуються регістри *SP* або *BP*.

Перетворення логічних адрес на фізичні завжди однозначне, тобто парі *Seg* і *EA* відповідає єдина фізична адреса. Зворотне перетворення не є однозначним: фізичну адресу можна подати за допомогою 4096 пар логічних адрес. Фізична адреса позначається у вигляді *Seg:EA*, де як *Seg* і *EA* можуть використовуватися і позначення регістрів, і 16-розрядні дані.

На рисунку 4 показано розміщення у просторі 1 Мбайт чотирьох сегментів по 64 кбайт без перекриття. Початкові адреси сегментів визначаються вмістом 16-розрядних сегментних регістрів, які доповнено праворуч чотирма нульовими бітами. Зміщення в сегменті кодів визначається вмістом регістра *IP*; зміщення в сегменті даних і додатковому сегменті даних - ефективною адресою *EA*, яка наводиться в команді; у сегменті стека - вмістом регістра *SP*.

У сегментах кодів розміщено коди команд, тобто програму у машинних кодах; у решті сегментів – дані. Програма може звертатися тільки до даних у

сегментах (рисунок 4).

Змінюючи вміст сегментних реєстрів, можна пересувати сегменти в межах усєї пам'яті 1 Мбайт.

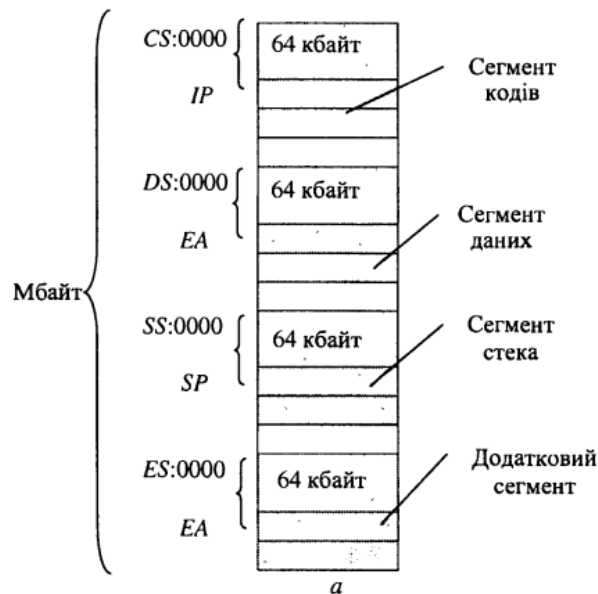


Рисунок 4 – Розміщення сегментів у просторі пам'яті 1 Мбайт

Реєстри загального призначення поділяються на реєстри даних і реєстри-вказівники. До реєстрів даних відносять чотири 16-розрядні реєстри: *AX*, *BX*, *CX*, *DX*. Кожний із цих реєстрів складається з двох 8-розрядних реєстрів, які можна незалежно адресувати за символічними іменами *AH*, *BH*, *CH*, *DH* (старші байти - *High*) та *AL*, *BL*, *CL*, *DL* (молодші байти - *Low*). Реєстри-вказівники *SP* (*Stack Pointer* - вказівник стека), *BP* (*Base Pointer* - базовий реєстр), *SI* (*Source Index* - індекс джерела), *DI* (*Destination Index* - індекс призначення) є 16-розрядними. Усі РЗП можна використати для зберігання даних, але в деяких командах допускається використання певного реєстра за замовчуванням. На відміну від 8-розрядних МП реєстр *SP* зберігає зміщення останньої зайнятої комірки стека відносно початку сегмента стека, а повна адреса стека визначається як *SS:SP*.

Сегментні реєстри *CS*, *DS*, *ES*, *SS* визначають початкові адреси чотирьох сегментів пам'яті. Використання сегментних реєстрів визначається типом звернення до пам'яті.

2 Архітектурні особливості мікропроцесорів наступних поколінь

Аналіз коду програм, що генерується компіляторами мов високого рівня, показав, що найчастіше використовується тільки обмежений набір простих команд форматів "реєстр, реєстр → реєстр" і "реєстр ↔ пам'ять". Компілятори не в змозі ефективно використовувати складні команди. Це

спостереження сприяло формуванню концепції процесорів з скороченим набором команд, так званих RISC-процесорів (RISC - Reduced Instruction Set Computer).

Дейв Паттерсон і Карло Секуїн сформулювали 4 основних принципи побудови RISC-процесорів:

1. Будь-яка операція повинна виконуватися за один такт, незалежно від її типу.

2. Система команд повинна містити мінімальну кількість найпростіших інструкцій однакової довжини, що найбільш часто використовуються.

3. Операції обробки даних реалізуються тільки у форматі "регістр←регістр" (операнди вибираються з оперативних регістрів процесора, і результат операції записується також у регістр, а обмін між оперативними регістрами й пам'яттю виконується тільки за допомогою команд завантаження записи).

4. Склад системи команд повинен бути "зручний" для компіляції операторів мов високого рівня.

Таким чином, RISC-процесори комп'ютерів зі скороченим набором команд мають команди обробки типу "регістр ←регістр" і команди збереження (store) і завантаження (load) типу "пам'ять ← регістр" і "регістр ← пам'ять" відповідно. Функціональні перетворення можуть виконуватися лише над вмістом регістрів, а результат поміщається тільки в регістр.

Після відокремлення RISC-процесорів в окремий клас процесори з традиційними наборами команд стали називатися "CISC-процесорами" (CISC – Complicated Instruction Set Computer) з повним набором команд. Як правило, в цих процесорах команди мають багато різних форматів і вимагають для свого представлення різне число байтів пам'яті. Це обумовлює визначення типу команди в ході її дешифрування при виконанні, що ускладнює пристрій управління процесора і перешкоджає підвищенню тактової частоти до рівня, досяжного в RISC-процесорах на тій же елементній базі.

Мікропроцесори з класичною CISC - архітектурою реалізують на рівні машинної мови комплексні набори команд різної складності (від простих, характерних для мікропроцесора першого покоління, до складних, характерних для 32-розрядних мікропроцесорів типу 80486, 68040 та ін.) Лідером у цій галузі є фірма Intel і її клонмейкери, мікропроцесорами якої комплектується більш 80% випускаються персональних комп'ютерів. Мікропроцесори сімейства M60xxx фірми Motorola використовувалися в персональних комп'ютерах типа Macintosh. Мікропроцесори цього сімейства

широко застосовуються також в пристроях управління, вбудованих в різні прилади і системи: контрольну вимірювальну та зв'язкову апаратуру, лазерні принтери та контролери дисководів, роботи і системи промислової автоматики.

Мікропроцесори з RISC - архітектурою застосовуються, в основному, в робочих станціях і потужних серверах. Широке застосування знаходять RISC - мікропроцесори сімейств SPARC фірми Sun Microsystems і Rх000 фірми MIPS Computer Systems. Останнім часом дуже активно впроваджуються в різну апаратуру RISC - мікропроцесори сімейства PowerPC – спільна розробка фірм IBM, Motorola і Apple Computers (альянс IMA). Процесори PowerPC 7XX (G3), PowerPC 74XX (G4), PowerPC 970 (G5) являються основою побудови персональних комп'ютерів iMac фірми Apple Computers. Серед фірм, що випускають RISC-мікропроцесори, знаходяться також Intel, Hewlett-Packard, Digital Equipment. Необхідно також відмітити трансп'ютери – оригінальні RISC-мікропроцесори, що розроблені фірмою Immos для побудови мультипроцесорних систем.

Розвиток мікропроцесорів відбувається при постійному прагненні збереження наступності програмного забезпечення (ПЗ) і підвищення продуктивності за рахунок вдосконалення архітектури та збільшення тактової частоти. Збереження наступності ПЗ та підвищення продуктивності, взагалі суперечать один одному.

Так, наприклад, процесори з системою команд x86, що відносяться до класу CISC-процесорів аж до Pentium Pro, мали більш низькі тактові частоти в порівнянні з мікропроцесорами провідних компаній виробників RISC-процесорів, що виготовлялись за одними і тими же технологічними нормами. Для цих процесорів існували програми, на яких продуктивність x86 мікропроцесорів була значно нижче, ніж у RISC-процесорів, реалізованих на тій же елементній базі. Однак можливість використання сумісного програмного забезпечення для різних поколінь x86 процесорів забезпечувала їм стійке домінуюче становище на ринку.

Потім на основі розробок компаній NexGen і AMD, пізніше підхоплених компанією Intel, була реалізована успішна спроба вирішення проблеми підвищення продуктивності в рамках архітектури x86. Ці компанії, зберігаючи наступність за системою команд із CISC-мікропроцесорами сімейства x86, створили нові пристрої з використанням елементів RISC-архітектури. Першими прикладами такого підходу можуть служити мікропроцесори Nx586 (NexGen), K5, K6 (AMD), які використовують концепцію RISC-ядра. У мікропроцесор вбудовується апаратний транслятор, що перетворює команди x86 в команди внутрішнього RISC-процесора.

Компанія Intel вперше використала цей підхід у своїх мікропроцесорах з архітектурою P6. Важливим елементом архітектури, що з'явилися в МП i80486 фірми Intel, став конвеєр – спеціальний пристрій, що реалізує такий метод обробки команд усередині мікропроцесора, при якому виконання команди розбивається на кілька етапів, i80486 має п'ятиступінчастий конвеєр. Відповідні п'ять етапів включають:

- вибірку команди з кеш - пам'яті або оперативної пам'яті;
- декодування команди;
- генерацію адреси, за якої визначаються адреси операндів в пам'яті;
- виконання операції за допомогою АЛП;
- запис результату (куди буде записаний результат, залежить від алгоритму роботи конкретної машинної команди).

Таким чином, на стадії виконання кожна машинна команда як би розбивається на елементарні операції. У чому перевага такого підходу? Чергова команда після її вибірки потрапляє в блок декодування. Таким чином, блок вибірки вільний і може вибрати наступну команду. В результаті на конвеєрі можуть знаходитися у різній стадії виконання п'ять команд. Швидкість обчислення в результаті істотно зростає. Мікропроцесори, що мають один конвеєр, називаються скалярними, а два і більше – супер скалярними. Мікропроцесор Pentium має два конвеєра, тобто використовує суперскалярну архітектуру, тому може виконувати дві команди за машинний такт. Внутрішня структура конвеєра така ж, як і у i486. Мікропроцесори сімейства P6 (Pentium Pro/II/III) мають іншу структуру конвеєра.

Сучасні мікропроцесори можуть містити десять і більше конвеєрів. У разі ефективного завантаження паралельно функціонуючих пристроїв можливе отримання в одному такті декількох результатів операцій, представлених скалярами: цілочисельними операндами або операндами з плаваючою крапкою.

Ефективне завантаження паралельно функціонуючих конвеєрів забезпечується або апаратурою процесора, або компілятором, на вході якого надходять програми з традиційною послідовною мовою програмування, або спільно апаратурою та компілятором. У компіляторах використовується особлива техніка вилучення паралелізму з послідовних програм.

Є два крайніх підходи до відображення властивого мікропроцесору внутрішнього паралелізму обробки даних на архітектурному рівні в системі команд.

Перший підхід більш консервативний і полягає в тому, що ніякої вказівки на паралельну обробку усередині процесора система команд не містить. Саме такі процесори відносяться до класу суперскалярних. Така

назва, з одного боку, відрізняє ці процесори від векторних процесорів, а з іншого боку, підкреслює властивий цим процесорам внутрішній паралелізм, що забезпечує одержання в одному такті декількох скалярних результатів.

Другий підхід, навпаки, повністю відкриває користувачеві усі можливості паралельної обробки. У спеціально відведених полях команди кожному з паралельно працюючих пристроїв обробки пропонується дія, яку пристрій повинен зробити.

Такі процесори називаються процесорами з довгим командним словом (VLIW - Very Long Instruction Word). Передбачається, що існують компілятори з мов високого рівня, які готують програми для завантаження їх в мікропроцесори. Представники цієї архітектури: Crusoe від Transmeta, Itanium від Intel і російський Ельбрус 2000 (архітектура E2K).

Подальше підвищення продуктивності мікропроцесорів зв'язується в даний час зі статичним і динамічним аналізом коду з метою виявлення паралелізму рівня програмних сегментів з використанням інформації, що надається компілятором мови високого рівня. Дослідження в даному напрямку привели до розробки мультитредової архітектури процесорів, які є подальшим розвитком супер скалярної архітектури.

Суперскалярні мікропроцесори і мікропроцесори з довгим командним словом мають один лічильник команд і в силу цього можуть бути названі однитредовими. У цих мікропроцесорах команди, що аналізуються на предмет можливості їх паралельного спільного виконання, прив'язані до лічильника команд процесора або вікном виконання як в суперскалярних мікропроцесорах, або довгою командою як в мікропроцесорах з довгим командним словом. Для того щоб більш агресивно вибирати для паралельного виконання команди однієї або кількох програм, в мікропроцесор вводиться кілька лічильників команд.

Мікропроцесори з декількома лічильниками команд отримали назву мультитредових. Мультитредовість в термінології Intel отримала назву Hyper-Threading (гіперточність). Процесори, повною мірою використовують усі переваги, надані мультитредовою архітектурою, розробляються фірмами IBM і SUN.

Суть технології **Hyper-Threading** полягає в тому, що в кристал процесора додано кілька блоків, що дозволяють одному фізичному процесору розпізнаватися і працювати в системі як два логічних процесора, кожен з яких може бути завантажений своїм завданням. Основна частина блоків процесора використовується спільно, але деякі про дубльовані і можуть виконувати різні завдання.

Технологія Hyper-Threading допомагає скоротити періоди простою процесора шляхом використання ресурсів, не зайнятих одним завданням, виконанням інструкцій іншого завдання, наприклад, у разі:

- затримок при доступі до пам'яті;
- виконання послідовності взаємозалежних інструкцій;
- помилок передбачення розгалужень;
- одночасних обчислень в цілочисленому і експоненційному форматах.

У результаті пропускну здатність основних ресурсів процесора зростає, а сумарний час виконання двох завдань скорочується.

3 Напрямки підвищення продуктивності процесорів

Розвиток мікропроцесорної техніки в області універсальних мікропроцесорів йде шляхом постійного підвищення їх продуктивності. Традиційними напрямками такого розвитку є *підвищення тактової частоти роботи МП і збільшення кількості одночасно виконуваних команд за рахунок збільшення числа конвеєрів* (виконавчих пристроїв) в МП.

Проте обоє ці напрями слід визнати екстенсивними, такими, що мають природні обмеження.

Підвищення тактової частоти, яке в основному забезпечується шляхом збільшення кількості сходинок в конвеєрі, призводить до великих втрат часу при необхідності перезавантаження конвеєра внаслідок конфліктів по управлінню або при перемиканні на нове завдання. Таке збільшення має також і фізичні обмеження, пов'язані з схмотехнікою кристала ВІС. Обмеження визначаються також впливом накладних витрат при передачі частково обробленої команди на наступний ступінь конвеєра. На великих частотах ці витрати стають сумірними з тривалістю обробки на черговому етапі. Багато в чому цей напрям вичерпав себе в мікропроцесорі Pentium 4, що працює на частотах, близьких до 4 ГГц.

Підвищення продуктивності за рахунок збільшення числа конвеєрів в мікропроцесорі можна оцінити збільшенням числа команд, що виконуються програмами за такт (IPC - INsTRuctions Per Cycle). Так, для МП Alpha 21264 цей показник дорівнює 6, стільки ж мікрооперацій за такт може видати Pentium 4. Але це граничні значення, а реальні програмні коди, зокрема, із-за різних взаємозалежностей, дають набагато нижче значення IPC. Подальше збільшення числа виконавчих пристроїв веде до ускладнення розташованого у ВІС пристрою управління, що розподіляє команди по конвеєрах, а також до складних взаємозалежностей між даними. До того ж реальні коди програм не дозволяють забезпечити ефективне завантаження усіх наявних в МП

виконавчих пристроїв, що призводить до їх простоїв. Слід зазначити також, що зростання продуктивності мікропроцесора не є прямо пропорційним зростанню кількості конвейерів, а зазвичай істотно нижче.

На сьогодні для підвищення продуктивності мікропроцесорів використовується ряд нових підходів, основними з яких є :

- CMP (Chip Multi ProcessINg) – створення на одному кристалі системи з декількох мікропроцесорів (багатоядерність);
- SMT (Simultaneous MultiThreadINg) – багатониткова архітектура;
- EPIC (Explicitly Parallel INsTRuction ComputINg) – обчислення з явним паралелізмом в командах.

Напрямок CMP забезпечується збільшеними технологічними можливостями, які дозволяють створити на одному кристалі декілька мікропроцесорів і організувати їх роботу за принципом мультипроцесорних систем.

Виробники чіпів вже не женуться за частотою, змістивши акцент на багатоядерну архітектуру, яка дозволяє нарощувати продуктивність, зберігаючи в прийнятних межах енергоспоживання і тепловиділення.

Багатоядерні процесори добре пристосовані для вимогливих мультимедійних завдань, таких як обробка відеозаписів, роботи з великими базами даних, одночасне виконання декількох ресурсоемних завдань, наприклад, комп'ютерної гри, запис DVD і завантаження файлів з Інтернету.

При такому підході завдання підвищення продуктивності роботи окремих застосувань вимагає розпаралелювання останніх, тобто проблема переміщається з апаратного на програмний рівень. На даний момент складнощі полягають в тому, що велика частина існуючого програмного забезпечення створювалася без розрахунку на використання в багатоядерних і багатопроцесорних конфігураціях. Іншими словами, прогрес в області апаратних засобів на якийсь час випередив прогрес в області програмного забезпечення.

Розвиток мікропроцесорної техніки в цьому напрямі йде дуже швидкими темпами. Так, компанія Tiler в 2007 році почала постачання спеціалізованих процесорів Tile64, що налічують 64 ядра

Напрямок SMT в розвитку архітектури мікропроцесорів базується на тому, що одне завдання не в змозі повністю завантажити усі зростаючі ресурси мікропроцесора. Тому на одному процесорі здійснюється запуск

декількох завдань одночасно, при цьому розпаралелювання програм здійснюється апаратними засобами МП.

Це дозволяє більше рівномірно завантажити ресурси процесора. Паралельно в різних пристроях МП можуть виконуватися команди з різних завдань. Так, мікропроцесор Alpha 21264 підтримує виконання до 4 завдань одночасно. За підтримки SMT на 4 нитки кожен процесор з точки зору операційної системи виглядає як 4 логічних процесора. Дослідження показали, що SMT дозволяє збільшити продуктивність цього процесора до двох разів, а додаткові схеми управління займають всього біля 10 % площі кристала.

Деякі мікропроцесори для максимального підвищення своєї продуктивності використовують обидва вищезазначені підходи. Так, компанія Sun Microsystems представила новий процесор ULTRaSPARC T2.

Новий чіп забезпечений вісьмома ядрами, кожне з яких може обробляти вісім потоків інструкцій. Таким чином, він одночасно здатний оперувати з 64 потоками. Тактова частота ULTRaSPARC T2 складає від 900 МГц до 1,4 ГГц.

Напрям EPIC фактично використовує відому технологію VLIW (Very Large INsTRuction Word) – дуже довгого командного слова.

Розпаралелювання алгоритму між виконавчими модулями виконується компілятором на етапі створення машинного коду, коли команди об'єднуються в зв'язки і не конкурують між собою за ресурси мікропроцесора. При цьому спрощується блок управління на кристалі.

Типовим представником архітектури EPIC є мікропроцесор Itanium фірми Intel.

1997 року фірми *Intel i Hewlett-Packard* розробили нову мікропроцесорну архітектуру EPIC (*Explicitly Parallel Instruction Computing* - явного паралельного обчислення команд), яку було покладено в основу 64-розрядних МП *IA-64, Itanium, Itanium 2*.

