

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УЖГОРОДСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інформаційних управляючих систем та технологій**

О.М. ЛЕВЧУК

АРХІТЕКТУРА ОБЧИСЛЮВАЛЬНИХ СИСТЕМ

Конспект лекцій

УЖГОРОД – 2023

Архітектура обчислювальних систем: Конспект лекцій до вивчення курсу для студентів спеціальностей: 126 Інформаційні системи та технології, 122 Комп'ютерні науки / О.М. Левчук. – Ужгород: 2023. 130 с.

Посібник з курсу «Архітектура обчислювальних систем» містить теоретичні та практичні відомості щодо принципів побудови, організації та функціонування сучасних обчислювальних систем. Розглядаються моделі обробки даних, структурні компоненти процесорів, пам'яті та пристроїв введення-виведення, а також наведено перелік використаних джерел.

Укладачі:

Левчук О.М., к.т.н., доцент кафедри інформаційних управляючих систем та технологій факультету інформаційних технологій УжНУ.

Рецензенти:

- Кондрат О.Б., к.ф-м.н., доцент кафедри інформаційних управляючих систем та технологій факультету інформаційних технологій УжНУ;
- Гуранич П.П., к.ф-м.н., доцент, завідувач кафедри оптики фізичного факультету УжНУ.

Рекомендовано кафедрою інформаційних управляючих систем та технологій.

Протокол № 1 від 30 серпня 2023 року.

Рекомендовано Вченою радою факультету інформаційних технологій.

Протокол № 2 від 26 вересня 2023 року.

© Левчук О.М., 2023

ЗМІСТ

Тема 1. Основні характеристики ЕОМ. Класифікація засобів ЕОТ.	3
Тема 2. Загальні принципи побудови сучасних ЕОМ. Апаратні і програмні засоби.	11
Тема 3. Процесор.....	21
Тема 4. Режими роботи процесора.	29
Тема 5. Покоління процесорів	32
Тема 6. Організація мікроконтролерів. Процесорне ядро мікроконтролера.....	45
Тема 7. Пам'ять програм і даних МК	54
Тема 8. Таймери і процесори подій.....	58
Тема 9. Тактові генератори МК.....	65
Тема 10. Цифрова логіка і цифрові системи	68
Тема 11. Представлення даних на машинному рівні.....	75
Тема 12. Елементи та вузли цифрового комп'ютера.....	82
Тема 13. Архітектура і організація систем пам'яті	95
Тема 14. Інтерфейсні системи комп'ютера	102
Тема 15. Обчислювальні системи.....	112
Тема 16. Паралельні та нетрадиційні архітектури.....	120
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	130

Тема 1. Основні характеристики ЕОМ. Класифікація засобів ЕОТ.

1. Основні характеристики ЕОМ.

Перші електронні обчислювальні машини (ЕОМ) з'явилися усього лише 60 років тому. За цей час мікроелектроніка, обчислювальна техніка і вся індустрія інформатики стали одними з основних складових світового науково-технічного прогресу. Вплив обчислювальної техніки на всі сфери діяльності людини продовжує розширюватися вшир і всередину. В даний час ЕОМ використовуються не тільки для виконання складних розрахунків, але й у керуванні виробничими процесами, в утворенні, охороні здоров'я, екології і т.д. Мається на увазі, що ЕОМ здатно обробляти будь-які види інформації: числову, текстову, табличну, графічну, відео, звукову.

Електронна обчислювальна машина – комплекс технічних і програмних засобів, призначений для автоматизації підготовки і рішення задач користувачів. Під користувачем розуміють людину, в інтересах якої проводиться обробка даних на ЕОМ. У якості користувача можуть виступати замовники обчислювальних робіт, програмісти, оператори. Як правило час підготовки задач у багато разів перевищує час їхнього розв'язування.

Вимоги користувачів до виконання обчислювальних робіт задовольняються спеціальним підбором і налаштуванням технічних і програмних засобів. Звичайно, ці засоби взаємозалежні і поєднуються в одну структуру.

Структура – сукупність елементів і їхніх зв'язків. Розрізняють структури технічних, програмних і апаратно-програмних засобів. Вибираючи ЕОМ для рішення своїх задач, користувач цікавиться функціональними можливостями технічних і програмних модулів (як швидко може бути вирішена задача, наскільки ЕОМ підходить для рішення даного кола задач, який сервіс програм є в ЕОМ, можливості діалогового режиму, вартість підготовки і рішення задач і т. д.). При цьому користувач цікавиться не конкретною технічною і програмною реалізацією окремих модулів, а більш загальними питаннями можливості організації обчислень.

Архітектура ЕОМ – це багаторівнева ієрархія апаратно-програмних засобів, з яких будується ЕОМ. Кожний з рівнів допускає різноманітну побудову і застосування. Конкретна реалізація рівнів визначає особливості структурної побудови ЕОМ.

Деталізацією архітектурної і структурної побудови ЕОМ займаються різні категорії фахівців обчислювальної техніки. Інженери-схемотехніки проектують окремі технічні пристрої та розробляють методи їхнього сполучення один з одним. Системні програмісти створюють програми керування технічними засобами, інформаційного взаємодії між рівнями, організації обчислювального процесу. Програмісти-прикладники розробляють пакети програм більш високого рівня, що забезпечують взаємодію користувачів з ЕОМ та необхідний сервіс при рішенні ними своїх задач.

Самого ж користувача цікавлять звичайно більш загальні питання, що стосуються його взаємодії з ЕОМ, починаючи з наступних груп характеристик ЕОМ, що визначають її структуру:

- технічні й експлуатаційні характеристики ЕОМ (швидкодія і продуктивність, показники надійності, вірогідності, точності, ємність оперативної і зовнішньої пам'яті, габаритні розміри, вартість технічних і програмних засобів, особливості експлуатації й ін.);
- характеристики і склад функціональних модулів базової конфігурації ЕОМ; можливість розширення складу технічних і програмних засобів; можливість зміни структури;
- склад програмного забезпечення ЕОМ і сервісних послуг (операційна чи система середовище, пакети прикладних програм, засобу автоматизації програмування).

Однією з найважливіших характеристик ЕОМ є її швидкодія, що характеризується числом команд, виконуваних ЕОМ за одну секунду. Оскільки до складу команд ЕОМ включаються операції, різні по тривалості виконання і по імовірності їхнього використання, то має сенс характеризувати його чи середньою швидкістю ЕОМ, чи граничним (для самих "коротких" операцій типу "регістр-регістр"). Сучасні обчислювальні машини мають дуже високі характеристики по швидкодії, вимірювані міліярдами операцій у секунду.

Реальна чи ефективна швидкодія, забезпечувана ЕОМ, значно нижче, і вона може сильно відрізнятися в залежності від класу розв'язуваних задач. Порівняння по швидкодії різних типів ЕОМ, що різко відрізняються один від одного своїми характеристиками, не забезпечує достовірних оцінок. Тому дуже часто замість характеристики швидкодії використовують зв'язану з нею характеристику продуктивності – обсяг робіт, здійснюваних ЕОМ за одиницю часу. Наприклад, можна визначати цей параметр числом задач, виконуваних за визначений час. Однак, порівняння по даній характеристиці ЕОМ різних типів може викликати складнощі, оскільки оцінка продуктивності різних ЕОМ є важливою практичною задачею.

Іншою найважливішою характеристикою ЕОМ є **ємність запам'ятовуючих пристроїв**. Ємність пам'яті вимірюється кількістю структурних одиниць інформації, що може одночасно знаходитися в пам'яті. Цей показник дозволяє визначити, який набір програм і даних може бути одночасно розміщений у пам'яті.

Найменшою структурною одиницею інформації є біт – одна двійкова цифра. Як правило, ємність пам'яті оцінюється в більш великих одиницях виміру – байтах (байт дорівнює вісьмом бітам). Наступними одиницями виміру є 1 Кбайт - 1024 байта, 1 Мбайт, 1 Гбайт.

Звичайно окремо характеризують ємність оперативної пам'яті і ємність зовнішньої пам'яті. В даний час персональні ЕОМ можуть мати ємність оперативної пам'яті, рівну 4-32 МБайтам і навіть більше. Цей показник дуже важливий для визначення.

Надійність – це здатність ЕОМ за певних умов виконувати необхідні функції протягом заданого періоду часу (стандарт ISO 23 82/14-78).

Висока надійність ЕОМ закладається в процесі її виробництва. Перехід на нову елементну базу – надвеликі інтегральні схеми (НВІС) різко скорочує число використовуваних інтегральних схем, а виходить, і число їхніх з'єднань один з одним. Добре продумане компонування комп'ютера і забезпечення необхідних режимів роботи (охолодження, захист від пилу), модульний принцип побудови дозволяє легко перевіряти і контролювати роботу всіх пристроїв, проводити діагностику й усунення несправностей.

Точність - можливість розрізняти майже рівні значення (стандарт 150 - 2382/2-76). Точність одержання результатів обробки в основному визначається розрядністю ЕОМ, а також з використовуваними структурними одиницями представлення інформації (байтом, словом, подвійним словом).

У багатьох застосуваннях ЕОМ не потрібно великої точності, наприклад, при обробленні текстів і документів, при керуванні технологічними процесами. У цьому випадку досить використовувати 8-и, 16- розрядні двійкові коди.

При виконанні складних розрахунків потрібно використовувати більш високу розрядність (32, 64 і навіть більш). Тому всі сучасні ЕОМ мають можливість роботи з 16- і 32- розрядними машинними словами. За допомогою засобів програмування мов високого рівня цей діапазон може бути збільшений у кілька разів, що дозволяє досягати дуже високої точності.

Вірогідність - властивість інформації бути правильно сприйнятої. Вірогідність характеризується імовірністю одержання безпомилкових результатів. Заданий рівень вірогідності забезпечується апаратурно-програмними засобами контролю самої ЕОМ. Можливі методи контролю вірогідності шляхом рішення еталонних задач і повторних розрахунків. В особливо відповідальних випадках проводяться контрольні рішення на інших ЕОМ і порівняння результатів.

2. Класифікація засобів ЕОТ.

В даний час у світі зроблені, працюють і продовжують випускатися мільйони обчислювальних машин, що відносяться до різних поколінь, типам, класам, що відрізняються своїми областями застосування, технічними характеристиками й обчислювальними можливостями. Традиційно електронну обчислювальну техніку (НОТ) підрозділяють на аналогову і цифрову.

В аналогових обчислювальних машинах (АОМ) оброблювана інформація представляється відповідними значеннями аналогових величин: струму, напруги, кута повороту якогось механізму і т. п. Ці машини забезпечують прийнятну швидкість, але не дуже високу точність обчислень (0.001-0.01). Поширено подібні машини не дуже широко. Вони використовуються в основному в проектних і науково-дослідних установах у складі різних стендів по відпрацюванню складних зразків техніки. По

своєму призначенню їх можна розглядати як спеціалізовані обчислювальні машини.

В даний час під словом ЕОМ звичайно розуміють цифрові обчислювальні машини, у яких інформація кодується двійковими кодами чисел. Саме ці машини завдяки універсальним можливостям і є самою масовою обчислювальною технікою.

Ринок сучасних комп'ютерів відрізняється розмаїтістю і динамізмом, яких ще не знала жодна область людської діяльності. Щороку вартість обчислень скорочується приблизно на 25-30%, вартість збереження одиниці інформації - до 40%. Практично кожне десятиліття міняється покоління машин, кожні два роки - основні типи мікропроцесорів - НВІС, що визначають характеристики нових ЕОМ. Такі темпи зберігаються вже багато років.

Те, що 10-15 років тому вважалося сучасною великий ЕОМ, у даний час є застарілою технікою з дуже скромними можливостями. Сучасний персональний комп'ютер зі швидкодією в десятки і сотні мільйонів операцій у секунду стає доступним засобом для масового користувача.

У цих умовах будь-яка запропонована класифікація ЕОМ дуже швидко застаріває і має потребу в коректуванні. Наприклад, у класифікаціях десятилітньої давнини широко використовувалися назви «мини», «міди» і **мікроЕОМ**. Разом з тим існує цілий ряд закономірностей розвитку обчислювальної техніки, що дозволяють передбачати і пророкувати основні результати цього поступального руху. Необхідно аналізувати традиційні і нові області застосування ЕОМ, класи і типи використовуваних обчислювальних засобів, що склалася кон'юнктура ринку інформаційних технологій і його динаміку, кількість і якість обчислювальної техніки, що випускається визнаними лідерами - виробниками засобів ЕОТ і т.д. Коротко розглянемо ці основні питання, з'ясування яких дозволить зрозуміти, яка обчислювальна техніка потрібна для рішення визначених задач.

Академік В.М. Глушков указував, що існують три глобальні сфери діяльності людини, що вимагають використання якісно різних типів ЕОМ.

Перший напрямок є традиційним - застосування ЕОМ для автоматизації обчислень. Науково-технічна революція у всіх областях науки і техніки постійно висуває нові наукові, інженерні, економічні задачі, що вимагають проведення великомасштабних обчислень (задачі проектування нових зразків техніки, моделювання складних процесів, атомна і космічна техніка й ін.). Відмінною рисою цього напрямку є наявність гарної математичної основи, закладеної розвитком математичних наук і їхніх додатків. Перші, а потім і наступні обчислювальні машини класичної структури в першу чергу і створювалися для автоматизації обчислень.

Друга сфера застосування ЕОМ зв'язана з використанням їх у системах керування. Вона народилася приблизно в 60-і роки, коли ЕОМ стали інтенсивно впроваджуватися в контури керування автоматичних і автоматизованих систем. Математична база цієї нової сфери практично була відсутня, протягом наступних 15-20 років вона була створена.

Нове застосування обчислювальних машин зажадало видозміни їхньої структури. ЕОМ, використовувані в керуванні, повинні були не тільки забезпечувати обчислення, але й автоматизувати збір даних і розподіл результатів обробки.

Сполучення з каналами зв'язку зажадало ускладнення режимів роботи ЕОМ, зробило їх багатопрограмними і **багатокористувацькими**. Для виключення взаємних перешкод між програмами користувачів у структуру машин були введені засоби розмежування: блоки переривань і пріоритетів, блоки захисту і т.і. Для керування різноманітною периферією стали використовуватися спеціальні процесори введення-висновку чи даних канали. Саме тоді і з'явився дисплей як засіб оперативної людино машинної взаємодії користувача з ЕОМ.

Новій сфері робіт найбільшою мірою відповідали **міні ЕОМ**. Саме вони стали використовуватися для керування галузями, підприємствами, корпораціями. Машини нового типу задовольняли наступним вимогам:

- були більш дешевими в порівнянні з великими ЕОМ, що забезпечують централізовану обробку даних;
- були більш надійними, особливо при роботі в контурі керування;
- мали велику гнучкість і адаптованість налаштування на конкретні умови функціонування;
- мали архітектурну прозорість, тобто структура і функції ЕОМ були зрозумілі користувачам.

Початок випуску подібних ЕОМ зв'язано з малими керуючими машинами РОР фірми ДЕС. Термін "**міні ЕОМ**" з'явився в 1968 р. стосовно до моделі РОР-8. В даний час використання міні-ЕОМ скорочується. Зникає і термін міні-ЕОМ. На зміну їм приходять ЕОМ інших типів: сервери, що забезпечують диспетчерські функції в мережах ЕОМ, середні ЕОМ чи старші моделі персональних ЕОМ (ПЕОМ).

Одночасно зі структурними змінами ЕОМ відбувалася і якісна зміна характеру обчислень. Частка чисто математичних розрахунків постійно скорочувалася, і в даний час вона складає близько 10% від всіх обчислювальних робіт. Машини усе більше стали використовуватися для нових видів обробки: текстів, графіки, звуку й ін.

Третій напрямок зв'язаний із застосуванням ЕОМ для рішення задач штучного інтелекту. Нагадаємо, що задачі штучного інтелекту припускають одержання не точного результату, а найчастіше усередненого в статистичному, ймовірнісному змісті. Прикладів подібних задач багато: задачі робототехніки, доказу теорем, машинного перекладу текстів з однієї мови на іншій, планування з урахуванням неповної інформації, складання прогнозів, моделювання складних процесів і явищ і т.д. Цей напрямок усе більше набирає силу. У багатьох областях науки і техніки створюються й удосконалюються бази даних і бази знань, експертні системи. Для технічного забезпечення цього напрямку потрібні якісно нові структури ЕОМ з великою кількістю обчислювачів (ЕОМ чи процесорних елементів), що забезпечують паралелізм

в обчисленнях. Власне кажучи, ЕОМ поступаються місцем найскладнішим обчислювальним системам.

Уже це невелике перерахування областей застосування ЕОМ показує, що для рішення різних задач потрібна відповідно і різна обчислювальна техніка. Тому ринок комп'ютерів постійно має широку градацію класів і моделей ЕОМ. Фірми-виробники засобів ОТ дуже уважно відслідковують стан ринку ЕОМ. Вони не просто констатують окремі факти і тенденції, а прагнуть активно впливати на них і випереджати потреби споживачів. Так, наприклад, фірма IBM, що випускає приблизно 80% світового машинного "парку", у даний час випускає в основному чотири класи комп'ютерів, перекриваючи ними широкий клас задач користувачів.

- **Великі ЕОМ (mainframe)**, що являють собою багатокористувацькі машини з центральною обробкою, з великими можливостями для роботи з базами даних, з різними формами вилученого доступу. Здавалося, що з появою швидко прогресуючих ПЕОМ великі ЕОМ приречені на вимирання. Однак вони продовжують розвиватися і випуск їх знову став збільшуватися, хоча їхня частка в загальному парку постійно знижується. По оцінках IBM, біля половини всього обсягу даних в інформаційних системах світу повинне зберігатися саме на великих машинах. Нове їхнє покоління призначене для використання в мережах як великі сервери. Початок цього напрямку було покладено фірмою IBM ще в 60-і роки випуском машин IBM/360, IBM/370. Ці машини одержали широке поширення у світі. Нова серія машин S/390 продовжує цю лінію. Вона нараховує більш двох десятків моделей.

- Машини RS/6000 - дуже могутні по продуктивності і призначені для побудови робочих станцій для роботи з графікою, Unix-серверів, кластерних комплексів. Спочатку ці машини передбачалося застосовувати для забезпечення наукових досліджень.

- Середні ЕОМ, призначені в першу чергу для роботи у фінансових структурах (ЕОМ типу AS/400 - "бізнеси-комп'ютери", 64-розрядні). У цих машинах особлива увага приділяється збереженню і безпеці даних, програмної сумісності і т.д. Вони можуть використовуватися як сервери в локальних мережах.

- Комп'ютери на платформі мікросхем фірми Intel. IBM-сумісні комп'ютери цього класу складають приблизно 50% ринку всієї комп'ютерної техніки. Більш половини їх надходить у сферу малого бізнесу. Незважаючи на настільки значний обсяг випуску персональних комп'ютерів цієї платформи, фірма IBM проводить великі дослідження і розвиток власної альтернативної платформи, що одержала назву Power PC. Цей напрямок дозволило б значно поліпшити структуру апаратних засобів ПК, а виходить, і ефективність їхнього застосування. Однак нові моделі цієї платформи поки не витримують конкуренції з IBM PC. Немаловажним тут є і нерозвиненість ринку програмного забезпечення. Тому в масового користувача цей напрямок попиту не знаходить, і частка комп'ютерів із процесорами Power PC незначна.

Крім перерахованих типів обчислювальної техніки, необхідно відзначити клас обчислювальних систем, що одержав назву "**суперЕОМ**". З розвитком науки і техніки постійно висувуються нові великомасштабні задачі, що вимагають виконання великих обсягів обчислень. Особливо ефективне застосування суперЕОМ при рішенні задач проектування, у яких натурні експерименти виявляються дорогими, недоступними чи практично нездійсненними. У цьому випадку ТОМ дозволяє методами чисельного моделювання одержати результати обчислювальних експериментів, забезпечуючи прийнятний час і точність рішення, тобто вирішальним умовою необхідності розробки і застосування подібних ЕОМ є економічний показник "продуктивність /вартість". СуперЕОМ дозволяють у порівнянні з іншими типами машин точніше, швидше і якісніше вирішувати подібні задачі, забезпечуючи необхідний пріоритет у розробках перспективної обчислювальної техніки. Подальший розвиток суперЕОМ зв'язується з використанням напрямку масового паралелізму, при якому одночасно можуть працювати сотні і навіть тисячі процесорів. Зразки таких машин уже випускаються декількома фірмами:

Необхідно відзначити і ще один клас найбільш масових засобів ЕОТ - **мікропроцесори, що вбудовуються**. Успіхи мікроелектроніки дозволяють створювати мініатюрні обчислювальні пристрої, аж до однокристальних ЕОМ. Ці пристрої, універсальні по характері застосування, можуть вбудовуватися в окремі машини, об'єкти, системи. Вони знаходять усе більше застосування в побутовій техніці (телефонах, телевізорах, електронних годинник, мікрохвильових печах і т.д.), у міському господарстві (енерго-, тепло-, водопостачанні, регулюванні руку транспорту і т.д), на виробництві (робототехніці, керуванні технологічними процесами). Поступово вони входять у наше життя, усе більше змінюючи середовище існування людини.

Таким чином, можна запропонувати наступну класифікацію засобів обчислювальної техніки, в основу якої покладено їхній поділ по швидкодії.

СуперЕОМ для рішення великомасштабних обчислювальних задач, для обслуговування найбільших інформаційних банків даних.

Великі ЕОМ для комплектування відомчих, територіальних і регіональних обчислювальних центрів.

Середні ЕОМ широкого призначення для керування складними технологічними виробничими процесами. ЕОМ цього типу можуть використовуватися і для керування розподіленою обробкою інформації як мережні сервери.

Персональні і професійні ЕОМ, що дозволяють задовольняти індивідуальні потреби користувачів. На базі цього класу БОМ будуються автоматизовані робочі місця (АРМ) для фахівців різного рівня.

Мікропроцесори, що вбудовуються, здійснюючу автоматизацію керування окремими пристроями і механізмами.

Високі швидкості обчислень, забезпечувані ЕОМ різних класів, дозволяють переробляти і видавати усе більша кількість інформації, що, у

свою чергу, породжує потреби в створенні зав'язків між окремо використовуваними ЕОМ. Тому всі сучасні ЕОМ у даний час мають засоби підключення до мереж зв'язку і комплексування в системи.

Перераховані типи ЕОМ, що повинні використовуватися в індустріально розвинутих країнах, утворюють деяку подобу піраміди з визначеним співвідношенням чисельності ЕОМ кожного шару і набором їхніх технічних характеристик. Розподіл обчислювальних можливостей по шарах повинний бути збалансовано. Наприклад, система обробки даних, використовувана на Олімпійських іграх в Атланті, містила: 4 великих ЕОМ S/390, 16 систем RS/6000, більш 80 систем AS/400, більш 7000 IBM PC, більш 1000 лазерних принтерів, більш 250 локальних мереж Token Ring і ін. Багато ПЕОМ мали сполучення з датчиками швидкості, часу і т.д.

Необхідна кількість суперЕОМ для окремо розвинутих країн, таких, як Україна, повинна складати 100-200 шт., великих ЕОМ - тисячі, середніх - десятки і сотні тисяч, ПЕОМ - мільйони, що вбудовуються микробОМ - мільярди. Усі використовувані ЕОМ різних класів утворюють машинний парк країни, життєдіяльність якого і його інформаційне насичення визначають успіхи інформатизації суспільства і науково-технічного прогресу країни. Формування збалансованого машинного парку є складною політичною, економічною і соціальною проблемою, рішення якої вимагає багатоміліардних інвестицій. Для цього повинна бути розроблена відповідна структура: створення спеціальних виробництв (елементної бази ЕОМ, програмного забезпечення і технічних зав'язків), зміна поколінь машин і технологій, зміна форм економічного й адміністративного керування, створення нових робочих місць і т.д.

Лекція 2. Загальні принципи побудови сучасних ЕОМ. Апаратні і програмні засоби.

1. Загальні принципи побудови сучасних ЕОМ.

Основним принципом побудови всіх сучасних ЕОМ є програмне керування. В основі його лежить представлення алгоритму рішення будь-якої задачі у виді програми обчислень.

"Алгоритм - кінцевий набір розпоряджень, що визначає розв'язок задачі за допомогою кінцевої кількості операцій". "Програма – упорядкована послідовність команд, що підлягає обробці". Варто помітити, що однозначного визначення алгоритму не існує. Принцип програмного керування може бути здійснений різними способами. Стандартом для побудови практично всіх ЕОМ став спосіб, описаний Дж. фон Нейманом у 1945 р. при побудові ще перших зразків ЕОМ. Суть його полягає в наступному.

Всі обчислення, запропоновані алгоритмом рішення задачі, повинні бути представлені у виді програми, що складає з послідовності керуючих слів-команд. Кожна команда містить указівки на конкретну виконувану операцію, місце перебування (адреси) операндів і ряд службових ознак. Операнди – змінні, значення яких беруть участь в операціях перетворення даних. Список (масив) усіх перемінних (вхідних даних, проміжних значень і результатів обчислень) є ще одним невід'ємним елементом будь-якої програми.

Для доступу до програм, команд і операндів використовуються їхні адреси. У якості адреси виступають номери комірок пам'яті ЕОМ, призначені для збереження об'єктів. Інформація кодується двійковими цифрами 0 і 1. Тому різні типи інформацій, розміщені в пам'яті ЕОМ, практично нерозрізнені, ідентифікація їх можлива лише при виконанні програми, відповідно до її логіки, по контексту.

Кожен тип інформації має формати – структурні одиниці інформації, закодовані двійковими цифрами 0 і 1. Звичайно усі формати даних, використовувані в ЕОМ, кратні байту, тобто складаються з цілого числа байтів.

Послідовність бітів у форматі, що має визначений зміст, називається полем. Наприклад, у кожній команді програми розрізняють поле коду операцій, поле адрес операндів. Стосовно до числової інформації виділяють знакові розряди, поле значущих розрядів чисел, старші і молодші розряди.

Послідовність, що складається з визначеного для даної ЕОМ числа байтів, називається словом. Для великих ЕОМ розмір слова складає чотири байти, для ПЕОМ – два байти. Як структурні елементи інформації розрізняють також півслово, подвійне слово й ін.

Схема ЕОМ, що відповідає програмному принципу керування, логічно впливає з послідовного характеру перетворень, виконуваних людиною по деякому алгоритмі. Узагальнена структурна схема ЕОМ перших поколінь представлена на рис. 1.

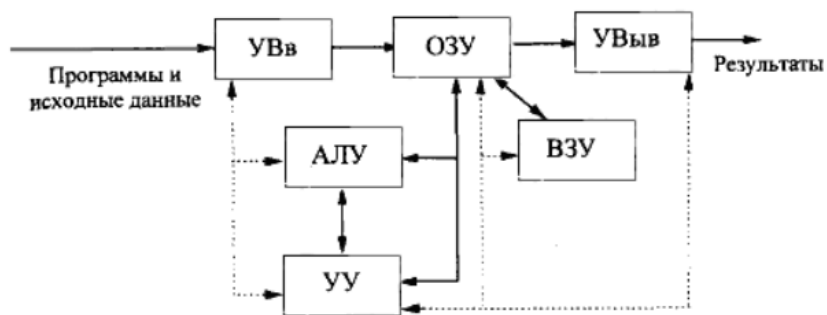


Рис. 1. Структурна схема ЕОМ першого і другого поколінь.

У будь-який ЕОМ маютья пристрої введення інформації (ПВв), за допомогою яких користувачі вводять в ЕОМ програми розв'язування задач і дані до них. Введена інформація цілком чи частково спочатку запам'ятовується в оперативному запам'ятовуючому пристрої (ОЗП), а потім переноситься в зовнішнє запам'ятовуюче пристрій (ЗП), призначене для тривалого збереження інформації, де перетвориться в спеціальний програмний об'єкт - файл. "Файл - ідентифікована сукупність екземплярів цілком описаного в конкретній програмі липу даних, що знаходяться поза програмою в зовнішній пам'яті і доступних програмі за допомогою спеціальних операцій (ДСТ 20866 - 85).

При використанні файлу в обчислювальному процесі його вміст переноситься в ОЗП. Потім програмна інформація команда за командою зчитується в пристрій керування (ПК).

Пристрій керування призначається для автоматичного виконання програм шляхом примусової координації всіх інших пристроїв ЕОМ. Ланцюга сигналів керування показані на рис. 1 штриховими лініями. Викликувані з ОЗП команди дешифруються пристроєм керування: визначаються код операції, яку необхідно виконати наступної, і адреси операндів, що приймають участь у даній операції.

У залежності від кількості використовуваних у команді операндів розрізняються одне-, двох-, триадресні і безадресні команди. В одноадресних командах указується, де знаходиться один із двох оброблюваних операндів. Другий операнд повинний бути поміщений заздалегідь в арифметичний пристрій (для цього в систему команд вводяться спеціальні команди пересилання даних між пристроями).

Двоадресні команди містять указівки про двох операндів, розташовуваних у пам'яті (чи в регістрах і пам'яті). Після виконання команди в одну з цих адрес засилається результат, а операнд який знаходився там губиться.

У триадресних командах звичайно дві адреси вказують, де знаходяться вихідні операнди, а третій - куди необхідно помістити результат.

У безадресних командах звичайно обробляється один операнд, що до і після операції знаходиться на одному з регістрів арифметико-логічного пристрою (АЛП). Крім того, безадресні команди використовуються для

виконання службових операцій (очистити екран, заблокувати клавіатуру, зняти блокування й ін.)

Усі команди програми виконуються послідовно, команда за командою, у тім порядку, як вони записані в пам'яті ЕОМ (природний порядок проходження команд). Цей порядок характерний для лінійних програм, тобто програм, що не містять розгалужень. Для організації розгалужень використовуються команди, що порушують природний порядок проходження команд. Окремі ознаки результатів г пристрій керування використовує для зміни порядку виконання команд програми.

АЛП виконує арифметичні і логічні операції над даними. Основною частиною АЛП є операційний автомат, до складу якого входять суматори, лічильники, регістри, логічні перетворювачі й ін. Воно щораз перенастроюється на виконання чергової операції. Результати виконання окремих операцій зберігаються для наступного використання на одному з регістрів АЛП чи записуються в пам'ять. Результати, отримані після виконання всієї програми обчислень, передаються на пристрої виводу (ПВВ) інформації. У якості ПВВ можуть використовуватися екран дисплея, принтер, графобудівник і ін.

Сучасні ЕОМ мають досить розвинуті системи машинних операцій. Наприклад, ЕОМ типу IBM PC мають близько 200 різних операцій (170 - 230 у залежності від типу мікропроцесора).

Будь-яка операція в ЕОМ виконується по визначеній мікропрограмі, реалізованої в схемах АЛП відповідною послідовністю сигналів керування (мікрокоманд). Кожна окрема мікрокоманда - це найпростіше елементарне перетворення даних типу алгебраїчного додавання, зрушення, перезапису інформації і т.п.

Вже в перших ЕОМ для збільшення їхньої продуктивності широко застосовувалося сполучення операцій. При цьому послідовні фази виконання окремих команд програми (формування адрес операндів, вибірка операндів, виконання операції, відсилення результату) виконувалися окремими функціональними блоками. У своїй роботі вони утворювали своєрідний конвеєр, а їхня рівнобіжна робота дозволяла обробляти різні фази цілого блоку команд. Цей принцип одержав подальший розвиток в ЕОМ наступних поколінь. Але все-таки перші ЕОМ мали дуже сильну централізацію керування, єдині стандарти форматів команд і даних, "тверде" побудова циклів виконання окремих операцій, що багато в чому порозумівається обмеженими можливостями використовуваної в них елементної бази. Центральне ПК обслуговувало не тільки обчислювальні операції, але й операції введення-висновку, пересилань даних між ЗП й ін. Усе це дозволяло в якомусь ступені спростити апаратуру ЕОМ, але сильно стримувало ріст їхньої продуктивності.

В ЕОМ третього покоління відбулося ускладнення структури за рахунок поділу процесів введення-виведення інформації і її обробки (рис. 1.2).

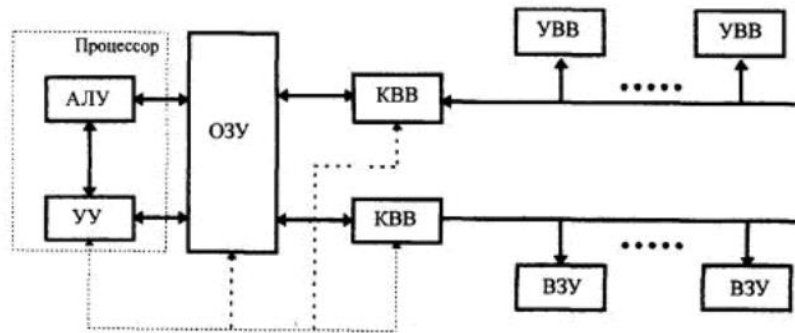


Рис. 2. Структурна схема ЕОМ третього покоління.

Сильнозв'язні пристрої АЛП і ПК одержали назва процесор, тобто пристрій, призначена для обробки даних. У схемі ЕОМ з'явилися також додаткові пристрої, що мали назви: процесори введення-висновку, пристрою керування обміном інформацією, канали введення-виведення (КВВ). Остання назва одержала найбільше поширення стосовно до великих ЕОМ. Тут намітилася тенденція до децентралізації керування і рівнобіжній роботі окремих пристроїв, що дозволило різко підвищити швидкодію ЕОМ у цілому.

Серед каналів уведення-виведення виділяли мультимплексні канали, здатні обслуговувати велику кількість повільно працюючих пристроїв уведення- виведення (ПІВ), і селекторні канали, що обслуговують у багатоканальних режимах швидкісні зовнішні запам'ятовуючі пристрої (ЗЗП).

У персональних ЕОМ, що відносяться до ЕОМ четвертого покоління, відбулася подальша зміна структури (рис. 1.3).

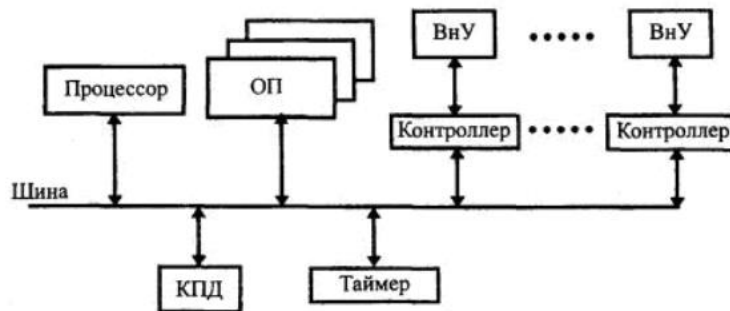


Рис. 3. Структурна схема ПЕОМ.

З'єднання всіх пристроїв у єдину машину забезпечується за допомогою загальної шини, що представляє собою лінії передачі даних, адрес, сигналів керування і харчування. Єдина система апаратурних з'єднань значно спростила структуру, зробивши її ще більш децентралізованою. Усі передачі даних по шині здійснюються під керуванням сервісних програм.

"Ядро ПЕОМ утворюють процесор і основна пам'ять (ОП), що складається з оперативної пам'яті і постійного запам'ятовуючого пристрою (ПЗП). ПЗП призначається для запису і постійного збереження найбільше часто використовуваних програм керування. Підключення всіх зовнішніх пристроїв (ЗП), дисплея, клавіатури, зовнішніх ЗП й інших забезпечується через

відповідні адаптери – узгоджувачі швидкостей роботи пристроїв, що сполучаються, чи контролери – спеціальні пристрої керування периферійною апаратурою. Контролери в ПЕОМ відіграють роль каналів уведення-висновку. Як особливі пристрої варто виділити таймер – пристрій виміру часу і контролер прямого доступу до пам'яті (КПД) – пристрій, що забезпечує доступ до ОП, міняючи процесор.

Спосіб формування структури ПЕОМ є досить логічним і природним стандартом для даного класу ЕОМ.

Децентралізація побудови і керування викликала до життя такі елементи, що є загальним стандартом структур сучасних ЕОМ:

- модульність побудови;
- магістральність;
- ієрархія керування.

Модульність побудови припускає виділення з структури ЕОМ досить автономних, функціонально і конструктивно закінчених пристроїв (процесор, модуль пам'яті, нагромаджувач на твердому чи гнучкому магнітному диску).

Модульна конструкція ЕОМ робить її відкритою системою, здатною до адаптації й удосконалювання. ДО ЕОМ можна підключати додаткові пристрої, поліпшуючи її технічні й економічні показники. З'являється можливість збільшення обчислювальної потужності, поліпшення структури шляхом заміни окремих пристроїв на більш зроблені, зміни і керування конфігурацією системи, пристосування її до конкретних умов застосування відповідно до вимог користувачів.

У сучасних ЕОМ принцип децентралізації і рівнобіжної роботи розповсюджений як на периферійні пристрої, так і на самі ЕОМ (процесори). З'явилися обчислювальні системи, що містять трохи обчислювачів (ЕОМ чи процесори), що працюють узгоджено і паралельно. Усередині самої ЕОМ відбувся ще більш різкий поділ функцій між засобами обробки. З'явилися окремі спеціалізовані процесори, наприклад співпроцесори, що виконують обробку чисел із крапкою, що плаває, матричні процесори й ін.

Всі існуючі типи ЕОМ випускаються сімействами, у яких розрізняють старші і молодші моделі. Завжди мається можливість заміни більш слабкої моделі на більш могутню. Це забезпечується інформаційною, апаратурною і програмною сумісністю. Програмна сумісність у сімействах установлюється за принципом нагору, тобто програми, розроблені для ранніх і молодших моделей, можуть оброблятися і на старших, але не обов'язково навпаки.

Модульність структури ЕОМ вимагає стандартизації й уніфікації устаткування, номенклатури технічних і програмних засобів, засобів сполучення - інтерфейсів, конструктивних рішень, уніфікації типових елементів заміни, елементної бази і нормативно-технічної документації. Усе це сприяє поліпшенню технічних і експлуатаційних характеристик ЕОМ, росту технологічності їхнього виробництва.

Децентралізація керування припускає ієрархічну організацію структури ЕОМ. Централізоване керування здійснює пристрій керування головного, чи

центрального, процесора. Модулі, що підключаються до центрального процесора, (контролери і КВВ) можуть, у свою чергу; використовувати спеціальні чи шини магістралі для обміну керуючими сигналами, адресами і даними. Ініціалізація роботи модулів забезпечується по командах центральних пристроїв, після чого вони продовжують роботу з власних програм керування. Результати виконання необхідних операцій представляються ними "нагору по ієрархії" для правильної координації всіх робіт.

Ієрархічний принцип побудови і керування характерний не тільки для структури ЕОМ у цілому, але і для окремих її підсистем. Наприклад, по цьому ж принципі будується система пам'яті ЕОМ.

Історія розвитку ЕОТ дала не дуже широкий спектр основних структур ЕОМ. Усі приведені структури не виходять за межі класичної структури фон Неймана. Їх поєднують наступні традиційні ознаки:

- ядро ЕОМ утворить процесор - єдиний обчислювач у структурі, доповнений каналами обміну інформацією і пам'яттю;
- лінійна організація осередків усіх видів пам'яті фіксованого розміру;
- однорівнева адресація комірок пам'яті, що стирає розходження між усіма типами інформації
- внутрішня машинна мова низького рівня, при якому команди містять елементарні операції перетворення простих операндів
- послідовне централізоване керування обчисленнями:
- досить примітивні можливості пристроїв введення-виведення.

Незважаючи на всі досягнуті успіхи, класична структура ЕОМ не забезпечує можливостей подальшого збільшення продуктивності. Намітилася криза, обумовлена появою істотних недоліків:

- погано розвиті засоби обробки нечислових даних (структури, символи, пропозиції, графічні образи, звук, дуже великі масиви даних і ін.);
- невідповідність машинних операцій операторам мов високого рівня;
- примітивна організація пам'яті ЕОМ;
- низька ефективність ЕОМ при рішенні задач, що допускають рівнобіжну обробку і т.п.

Усі ці недоліки приводять до надмірного ускладнення комплексу програмних засобів, використовуваного для підготовки і рішення задач користувачів.

В ЕОМ майбутніх поколінь, з використанням у них "убудованого штучного інтелекту", передбачається подальше ускладнення структури; В-першу чергу, це стосується удосконалювання процесів спілкування користувачів з ЕОМ (використання аудіо-, відеоінформації, систем мультимедіа й ін.), забезпечення доступу до баз даних і баз знань, організації рівнобіжних обчислень. Безсумнівно, що цьому повинні відповідати нові рівнобіжні структури з новими принципами їхньої побудови. Як приклад,

найшвидша ЕОМ фірми IBM у даний час забезпечує швидкодію 600 MBPS (мільйонів команд у секунду), сама ж велика гіперкубічна система nCube дає швидкодію 123.103 MBPS. Розрахунки показують, що вартість однієї машинної операції в гіперсистемі приблизно в тисячу разів менше. Імовірно, подібними системами будуть обслуговуватися великі інформаційні сховища.

2. Функції програмного забезпечення.

Електронні обчислювальні машини є універсальними технічними засобами автоматизації обчислювальних робіт, тобто вони здатні вирішувати будь-які задачі, зв'язані з перетворенням інформації. Однак підготовка задач до рішення на ЕОМ була і залишається дотепер досить трудомістким процесом, що вимагає від користувачів у багатьох випадках спеціальних знань і навичок.

Для зниження трудомісткості підготовки задач до рішення, більш ефективного використання окремих технічних, програмних засобів і ЕОМ у цілому, а також полегшення їхньої експлуатації кожна ЕОМ має спеціальний комплекс програмних засобів регулярного застосування. Ці засоби забезпечують взаємодію користувачів з ЕОМ і є своєрідним "посередником" між ними. Вони одержали назву програмного забезпечення (ПЗ) ЕОМ.

Під програмним забезпеченням будемо розуміти комплекс програмних засобів регулярного застосування, призначений для підготовки і рішення задач користувачів.

Програмне забезпечення окремих ЕОМ і ВР може сильно розрізнятися складом використовуваних програм, що визначається класом використовуваної обчислювальної техніки, режимами її застосування, змістом обчислювальних робіт користувачів і т.п. Розвиток ПО сучасних ЕОМ і ВР у значній мірі носить еволюційний і емпіричний характер, але можна виділити закономірності в його побудові.

У загальному випадку процес підготовки і рішення задач на ЕОМ користувачами передбачає виконання наступної послідовності етапів (рис. 1.4):

Помі- чення ЭВМ	Етапы подготовки и решения задач					
	Постановка задачи	Выбор алгоритма	Программи- рование на входном языке	Организация вычислитель- ного процесса	Получение машинной программы	Решение задачи
I	П О Л Ь З О В А Т Е Л Ь					Аппаратура
II	П О Л Ь З О В А Т Е Л Ь				Программные средства	Аппаратура
III	П О Л Ь З О В А Т Е Л Ь				Программное обеспечение	Аппаратура
IV	П О Л Ь З О В А Т Е Л Ь				Программное обеспечение	Аппаратура
V	П О Л Ь З О В А Т Е Л Ь				Программное обеспечение	Аппаратура
VI						

Рис. 4. Автоматизація підготовки і рішення задач в ЕОМ.

- формулювання проблеми і математична постановка задачі;
- вибір методу і розробка алгоритму рішення;
- програмування (запис алгоритму) з використанням деякої алгоритмічної мови;

- планування й організація обчислювального процесу - порядку і послідовності використання ресурсів ЕОМ і ВР;
- формування "машинної програми", тобто програми, которую безпосередньо буде виконувати ЕОМ;
- власне рішення задачі - виконання обчислень по готовій програмі.

В міру розвитку обчислювальної техніки автоматизація цих етапів йде нагору. В ЕОМ 1-го покоління автоматизації підлягав тільки останній етап. Усі п'ять попередніх етапів користувач повинний був готувати вручну самостійно. Трудомісткий і рутинний характер цих робіт був джерелом великої кількості помилок у завданнях. Тому в ЕОМ наступних поколінь з'явилися спочатку елементи, а потім цілі системи, що полегшують процес підготовки задач до рішення.

Для ЕОМ 2-го покоління характерно широке застосування алгоритмічних мов (Автокоди, Алгол, Фортран і ін.) і відповідних трансляторів, що дозволяють автоматично формувати машинні програми по їхньому описі алгоритмічною мовою. Тут же широко стали впроваджуватися бібліотеки стандартних програм, що дозволило будувати машинні програми блоками, використовуючи накопичений і придбаний програмістами досвід. Відзначимо, що тимчасові границі появи всіх нововведень досить розмиті. Звичайно їхні джерела можна знайти в надрах ЕОМ попередніх поколінь.

ЕОМ 3-го покоління характеризуються розквітом операційних систем (ОС), що відповідають за організацію і керування обчислювальним процесом. Саме тут слово "ЕОМ" усі частіше стало замінятися поняттям "обчислювальна система", що в більшому ступені відбивало ускладнення як апаратної, так і програмної частин ЕОМ. Вартість програмного забезпечення стала рости і в даний час набагато випереджає вартість апаратних засобів.

Операційна система планує послідовність розподілу і використання ресурсів обчислювальної системи, а також забезпечує їхню погоджену роботу. Під ресурсами звичайно розуміють ті засоби, що використовуються для обчислень: машинний час окремих чи процесорів ЕОМ, що входять у систему; обсяги оперативної і зовнішньої пам'яті; окремі пристрої, інформаційні масиви; бібліотеки програм; окремі програми як загального, так і спеціального застосування і т. п. Найбільш уживані функції ОС у частині обробки позаштатних ситуацій (захист програм від взаємних перешкод, системи переривань і пріоритетів, служба часу, сполучення з каналами зв'язку і тд.) були чи цілком частково реалізовані апаратно. Одночасно були реалізовані більш складні режими роботи: колективний доступ до ресурсів, мультипрограмні режими. Частина цих рішень стала своєрідним стандартом і початку використовуватися повсюдно в ЕОМ різних класів. Це дозволило в значній мірі підвищити ефективність застосування ЕОМ і ВР у цілому.

В ЕОМ 4-го покоління продовжується ускладнення технічних і програмних структур. Слід зазначити помітне підвищення "інтелектуальності" машин. Особливо це стало видно з появою персональних ЕОМ, орієнтованих на визначені категорії користувачів. Програмне забезпечення цих машин

створює дружнє середовище спілкування людини і комп'ютера. Воно, з одного боку, керує процесом обробки інформації, а з іншого боку – створює необхідний сервіс для користувача, знижуючи трудомісткість його рутинної роботи і надаючи йому можливість більше уваги приділяти творчості.

Подібні тенденції будуть зберігатися й в ЕОМ наступних поколінь. Так, на думку дослідників, машини наступного сторіччя будуть мати убудований у них штучний інтелект, що дозволить користувачам звертатися до машин (систем) природною мовою, вводити й обробляти тексти, документи, ілюстрації, створювати системи обробки знань і т. д. Усе це приводить до необхідності розробки складного, багатошарового ієрархічного програмного забезпечення систем обробки даних.

3. Персональні ЕОМ.

В даний час ПЕОМ є самим масовим типом. Саме їм приділяється вирішальна роль при переході суспільства до інформатизації - найбільш повному використанню інформаційних технологій.

Цікаві причини появи і розвитку цього класу ЕОМ. Структура і динаміка розвитку світового парку ЕОМ показані на рис 5.

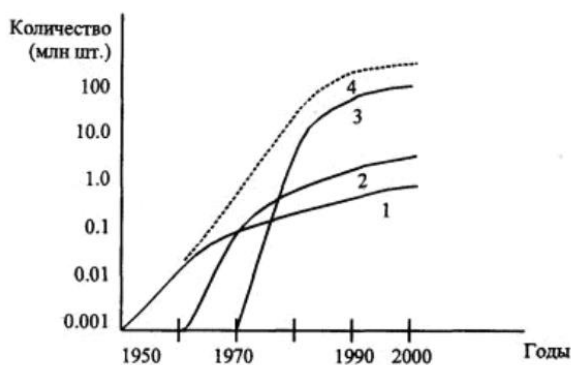


Рис. 1.6. Структура і динаміка розвитку світового парку ЕОМ. Класи машин: 1 - великі ЕОМ; 2 - міні ЕОМ; 3 - персональні ЕОМ; 4 - сумарний парк.

В даний час частка ПЕОМ у світовому парку складає близько 80%. Частки великих ЕОМ і міні ЕОМ (останнім часом вони замінюються середніми ЕОМ нових поколінь) оцінюються приблизно по 10%. Розвиток ПЕОМ визначається насамперед економічними факторами, тому що вартість одиниці обчислювальної потужності в них обходиться значно дешевше. Поява ПЕОМ закономірно і порозумівається зміною характеру обчислювальних робіт, у яких велику роль грає нечислова обробка.

Великі ЕОМ в основному використовувалися і використовуються для централізованої обробки інформації. У першу чергу вони застосовувалися для великомасштабних обчислень по програмах, розробленим колективами фахівців. Тому дорогі великі машини встановлювалися у великих академічних обчислювальних центрах.

Міні ЕОМ стали застосовуватися для розподіленої обробки даних і для керування об'єктами, технологічними процесами, підприємствами.

З появою персональних ЕОМ намітився новий етап в організації і забезпеченні обчислень - етап "персональних обчислень". Суть його виражається девізом "One man – one job – one computer" (людина - робота - комп'ютер). Таким чином, персональні ЕОМ покликані вирішувати в першу чергу ті задачі, що виникають у фахівців різного профілю у визначені моменти часу, безпосередньо на робочих місцях, тобто там, де знаходяться джерела даних, підлягаючі обробці.

При цьому найпоширенішим режимом роботи є режим безпосереднього доступу до ресурсів ЕОМ, "один на один з комп'ютером". Подібний режим роботи уже використовувався при роботі з першими ЕОМ, однак при централізованому керуванні він був у край неефективний. Якщо раніше за пультом великий ЕОМ повинний був знаходитися професійний програміст, то за персональним комп'ютером звичайно знаходиться "непрограмуєчий професіонал". Так звичайно називають фахівця конкретної предметної області (бухгалтера, економіста, інженера-дослідника і т.п.), але не фахівця в обчислювальній техніці і програмуванні. Тому повернення до режиму безпосереднього доступу відбувається на якісно новій основі.

При широкому застосуванні ПЕОМ у різних сферах діяльності людини висуваються вимоги до їх належного програмного забезпечення. В даний час число професійних програмістів в індустріально розвинутих країнах складає не більш 0,390 населення. Фірми - розроблювачі програмного забезпечення не можуть забезпечити кожного користувача ПЕОМ необхідним набором програм. Їхні зусилля зосереджені на виробництві пакетів прикладних програм і систем програмування, розрахованих на масового користувача. Саме тому такий вибуховий характер мають попит, виробництво і поширення подібних пакетів. Вони складають фундамент для наступної розробки власних програм користувача, що враховують усю специфіку необхідних обчислень, тобто, як і у всіх науках, спеціалізація є надбудовою уніфікації. Це дозволяє користувачам - фахівцям з невисокою математичною, обчислювальною і програмістською підготовкою необов'язково найефективнішими засобами і способами ставити і вирішувати задачі спеціальної обробки даних.

МікроЕОМ, орієнтована на розробку і використання прикладних програм "непрограмуєчим професіоналом", одержала назву персонального комп'ютера, а відповідний режим використання обчислювальної техніки - режим персональних обчислень.

Основна мета використання ПЕОМ - формалізація професійних знань. Тут, у першу чергу, автоматизується рутинна частина робіт фахівців, що займає більш 75% їхнього робочого часу. Застосування ПЕОМ дозволяє зробити праця фахівців більш творчим, цікавим, ефективним. Персональні ЕОМ використовуються повсюдно, у всіх сферах діяльності людей. Нові сфери застосування змінили і характер обчислювальних робіт. Так, інженерно-технічні розрахунки складають не більш 9%, автоматизація керування збутому, закупівлями, керування запасом – 16%, фінансово-економічні розрахунки -15%, діловодство - більш 10%, ігрові задачі – 8% і т.д.

Лекція 3. Процесор.

1. Поняття центрального процесору.

Основним пристроєм для обробки інформації є мікропроцесор або центральний процесор.

Центральний процесор (CPU, від англ. Central Processing Unit) – це основний робочий компонент комп'ютера, який виконує арифметичні і логічні операції, задані програмою, управляє обчислювальним процесом і координує роботу всіх пристроїв комп'ютера.

Центральний процесор в загальному випадку містить у собі:

- арифметико-логічний пристрій;
- шини даних і шини адрес;
- регістри;
- лічильники команд;
- кеш - дуже швидко пам'ять малого обсягу (від 8 до 512 Кбайт);
- математичний співпроцесор чисел з плаваючою точкою.

Сучасні процесори виконуються у вигляді мікропроцесорів. Фізично мікропроцесор являє собою інтегральну схему – тонку пластинку кристалічного кремнію прямокутної форми площею всього кілька квадратних міліметрів, на якій розміщені схеми, що реалізують всі функції процесора. Кристал-пластинка зазвичай міститься в пластмасовому або керамічному плоскому корпусі і з'єднується золотими проводками з металевими штирями, щоб його можна було приєднати до системної плати комп'ютера.

У сучасних персональних комп'ютерах різних фірм застосовуються процесори двох основних архітектур:

- Повна система команд змінної довжини - Complex Instruction Set Computer (CISC);
- Скорочений набір команд фіксованої довжини - Reduced Instruction Set Computer (RISC).

Весь ряд процесорів фірми Intel, що встановлюються в персональні комп'ютери IBM, мають архітектуру CISC, а процесори Motorola, використовувані фірмою Apple для своїх персональних комп'ютерів, мають архітектуру RISC. Обидві архітектури мають свої переваги і недоліки. Так CISC-процесори мають великий набір команд (до 400), з яких програміст може вибрати команду, найбільш підходящу йому для даного випадку. Недоліком цієї архітектури є те, що великий набір команд ускладнює внутрішній устрій управління процесором, збільшує час виконання команди на мікропрограмному рівні. Команди мають різну довжину і час виконання.

RISC-архітектура має обмежений набір команд і кожна команда виконується за один такт роботи процесора. Невелике число команд спрощує пристрій управління процесора. До недоліків RISC-архітектури можна віднести те, що якщо необхідної команди в наборі немає, програміст змушений реалізувати її за допомогою декількох команд з наявного набору, збільшуючи розмір програмного коду.

2. Пристрій керування виконанням команд.

Найбільш складним функціональним пристроєм процесора є пристрій керування виконанням команд. Він містить:

- буфер команд, який зберігає одну або кілька чергових команд програми; читає команди з запам'ятовуючого пристрою, поки виконується чергова команда, зменшуючи час її вибірки з пам'яті;
- дешифратор команд розшифровує код операції чергової команди і перетворює його на адресу початку прошивки, яка реалізує виконання команди;
- управління вибіркою чергової мікрокоманди – невеликий процесор, що працює за принципом фон Неймана, має свій лічильник мікрокоманд, який автоматично вибирає чергову мікрокоманду з ПЗП мікрокоманд;
- постійний запам'ятовуючий пристрій (ПЗП) мікрокоманд – це запам'ятовуючий пристрій, в який інформація записується одноразово і потім може тільки зчитуватися; відмінною рисою ПЗП є те, що записана в нього інформація зберігається скільки завгодно довго і не вимагає постійної напруги живлення.

Адреса, яка поступає від дешифратора команд, записується в лічильник мікрокоманд пристрою вибірки, і починається процес обробки послідовності мікрокоманд. Кожен розряд мікрокоманд пов'язаний з одним керуючим входом будь-якого функціонального пристрою. Так, наприклад, керуючі входи регістра зберігання «Скидання», «Запис», «Читання» з'єднані з відповідними розрядами мікрокоманд. Загальна кількість розрядів мікрокоманд може становити від декількох сотень до декількох тисяч і дорівнює загальній кількості керуючих входів всіх функціональних пристроїв процесора. Частина розрядів мікрокоманд подається на пристрій управління вибіркою чергової мікрокоманди і використовується для організації умовних переходів і циклів, так як алгоритми обробки команд можуть бути досить складними.

Вибірка чергової мікрокоманди здійснюється через певний інтервал часу, який, в свою чергу, залежить від часу виконання попередньої мікрокоманди. Частота, з якою здійснюється вибірка мікрокоманд, називається тактовою частотою процесора. Тактова частота є важливою характеристикою процесора, так як визначає швидкість виконання процесором команд, і, в кінцевому підсумку, швидкодію процесора.

Арифметико-логічний пристрій (АЛП) призначено для виконання арифметичних і логічних операцій перетворення інформації. Функціонально АЛП складається з декількох спеціальних регістрів, повнорозрядного сумарного і схем місцевого управління.

Регістри загального призначення (РЗП) використовуються для тимчасового зберігання операндів команди, яка виконується, і результатів обчислень, а також зберігають адреси комірок пам'яті або портів введення-виведення для команд, які звертаються до пам'яті і зовнішніх пристроїв. Необхідно відзначити, що якщо операнди команди зберігаються в РЗП, то час виконання команди значно скорочується. Одна з причин, чому програмісти

іноді звертаються до програмування на мові машинних команд, це найбільш повне використання РЗП для отримання максимальної швидкодії при виконанні програм, критичних за часом.

3. Пам'ять процесорів.

Пам'ять процесора призначена для короткочасного і довготривалого зберігання інформації - кодів команд і даних. Інформація в пам'яті зберігається в двійкових кодах, кожен біт - елементарна комірка - може приймати значення «0» або «1». Кожна комірка пам'яті має свою адресу, однозначно її ідентифікує в певній системі координат. Мінімальною одиницею зберігання інформації в пам'яті зазвичай є байт, що складається, як правило, з 8 біт.

Існують процесори і комп'ютери з розрядністю оброблюваного слова не кратного 8 (наприклад, 5, 7, 9 ...), і їх байти не восьмибітні, але в світі РС зіткнення з ними малоймовірно. Також в деяких системах сукупність восьми сусідніх біт даних називають октетом. Назва «октет» зазвичай має на увазі, що ці 8 біт не мають явної адреси, а характеризуються тільки своїм місцем розташування в довгому ланцюжку біт.

З появою великих (за розмірами) комп'ютерів відбувся розподіл пам'яті на внутрішню і зовнішню. Під внутрішньою мається на увазі пам'ять, розташована всередині процесорної «шафи». Сюди входила і електронна і магнітна пам'ять (на магнітних сердечниках). Зовнішня пам'ять представляла собою окремі пристрої з рухомими носіями - накопичувачі на магнітних дисках (а спочатку - на барабанах) і стрічці. Згодом всі пристрої комп'ютера вдалося вмістити в один невеликий корпус, і колишню класифікацію пам'яті стосовно РС можна переформулювати так:

- Внутрішня пам'ять - електронна (напівпровідникова) пам'ять, що встановлюється на системній платі або на платах розширення;
- Зовнішня пам'ять - пам'ять, реалізована у вигляді пристроїв з різними принципами збереження інформації і зазвичай з рухомими носіями. В даний час сюди входять пристрої магнітної (дискової і стрічкової) пам'яті, оптичної і магнітооптичної пам'яті. Пристрої зовнішньої пам'яті можуть розміщуватися як в системному блоці комп'ютера, так і в окремих корпусах, що досягають іноді і розмірів невеликої шафи.

Для процесора безпосередньо доступною є внутрішня пам'ять, доступ до якої здійснюється за адресою, заданої програмою. Для внутрішньої пам'яті характерна одновимірна (лінійний) адреса, яка є одним двійковим числом певної розрядності. Внутрішня пам'ять поділяється на оперативну, інформація в якій може змінюватися процесором в будь-який момент часу, і постійну, інформацію якої процесор може тільки зчитувати. Звернення до комірок оперативної пам'яті може відбуватися в будь-якому порядку, причому як за читанням, так і за записом, і оперативну пам'ять називають пам'яттю з довільним доступом - Random Access Memory (RAM) - на відміну від постійної пам'яті (Read Only Memory, ROM). Зовнішня пам'ять адресується складнішим чином - кожна її комірка має свою адресу всередині деякого блоку, який, в

свою чергу, має багатомірну адресу. Під час фізичних операцій обміну даними блок може бути зчитаний або записаний тільки цілком.

4. Принцип дії процесора.

Процесор є головним елементом ЕОМ. Він прямо або побічно управляє всіма пристроями і процесами, що відбуваються в ЕОМ. У конструкції сучасних процесорів чітко проглядається тенденція постійного збільшення тактової частоти. Це природно, чим більше операцій виконує процесор, тим вище його продуктивність. Гранична тактова частота багато в чому визначається існуючою технологією виробництва мікросхем (найменшими досяжними розмірами елементів, які визначають мінімальний час передачі сигналів).

Крім підвищення тактової частоти збільшення продуктивності процесорів досягається розробниками менш очевидними прийомами, пов'язаними з винаходом нових архітектур і алгоритмів обробки інформації. Деякі з них розглянемо на прикладі процесора Pentium і наступних моделей.

Перелічимо основні особливості процесора Pentium:

- конвеєрна обробка інформації;
- суперскалярна архітектура;
- наявність роздільних кеш-пам'ятей для команд і даних;
- наявність блоку передбачення адреси переходу;
- наявність блоку обчислень з плаваючою точкою;
- підтримка многопроцесорного режиму роботи;
- наявність кошти завдання розміру сторінки пам'яті;
- наявність засоби виявлення помилок.

Термін суперскалярна архітектура означає, що процесор містить більше одного обчислювального блоку. Ці обчислювальні блоки частіше називають конвеєрами. Наявність в процесорі двох конвеєрів дозволяє йому одночасно виконувати дві команди. Кожен конвеєр розділяє процес виконання команди на п'ять етапів:

- зчитування команди з ОЗУ або кеш-пам'яті;
- декодування (дешифрування) команди, тобто визначення коду виконуваної операції;
- виконання команди;
- звернення до пам'яті;
- запам'ятовування отриманих результатів.

Для реалізації кожного з перерахованих етапів (кожної операції) служить окремий пристрій - щабель. Таким чином, в кожному конвеєрі процесора Pentium є п'ять ступенів. При конвеєрній обробці на виконання кожного етапу одні такт синхронізующої (тактовою) частоти. У кожному новому такті закінчується виконання однієї команди і починається виконання нової команди. Таке виконання команди називається потоковою обробкою.

Образно її можна порівняти з виробничим конвеєром, де на кожній дільниці з різними виробами виконують завжди одну і ту ж операцію. При

цьому, коли готовий виріб сходить з конвеєра, то на нього заходить нове, а решта виробів в цей час перебувають на різних стадіях готовності. Перехід виробів, що виготовляються з дільниці на дільницю повинен відбуватися синхронно, за спеціальними сигналами (в процесорі це - такти, відлічувані тактовим генератором).

Загальний час виконання однієї команди в конвеєрі з п'ятьма ступенями становитиме п'ять періодів тактової частоти. У кожному такті конвеєр буде одночасно виконувати п'ять команд. В результаті за п'ять тактів буде виконано п'ять команд. Таким чином, конвейеризація збільшує продуктивність процесора, але вона не скорочує час виконання окремої команди. Виграш виходить за рахунок того, що одночасно обробляється відразу кілька команд.

Насправді конвеєризація навіть збільшує час виконання кожної окремої команди через появу "накладних витрат", пов'язаних з організацією роботи конвеєра. При цьому тактова частота обмежується швидкістю роботи найповільнішої ступені.

5. Історія розвитку мікропроцесорів.

15 листопада 1971 року компанія Intel приступила до випуску першого в світі мікропроцесора моделі intel 4004. Він був чотири розрядний, тобто за одну операцію (такт) обробляв одне 4-розрядне число. Робоча частота становила 108 кГц (0,108 МГц). Цей процесор містив 2300 транзисторів і вироблявся за 10-мікронною технологією. Шина даних мала ширину 4 розряди, що дозволило адресувати 640 байт пам'яті. Цей процесор призначався для використання в програмованих калькуляторах, в схемах управління світлофорів, аналізаторі крові і навіть на міжпланетній науково-дослідницькій станції NASA Pioneer 10!

У 1972 р Intel розробила 8-розрядний мікропроцесор моделі intel 8008. Він містив 3500 транзисторів і вироблявся за технологією 10-мікрон. Шина даних була 8-розрядною, що дозволяло адресувати 16 Кбайт пам'яті. Цей процесор призначався для використання в терміналах і програмованих калькуляторах.

Наступна модель мікропроцесора, intel 8080, була анонсована компанією intel в квітні 1974 року. Цей процесор містив 6000 транзисторів і міг адресувати вже 64 Кбайт пам'яті. На ньому був зібраний перший персональний комп'ютер Altair 8800 (не PC). В цьому комп'ютері використовувалася операційна система CP/M, а фірма Microsoft розробила для нього інтерпретатор мови BASIC.

Завдяки популярності процесора intel 8080, деякі фірми почали випуск його клонів. У червні 1976 року з'явився процесор Z-80 (Zilog), який працював на частоті 2,5 МГц (більш пізні моделі працювали на частоті 10 МГц). Даний процесор був несумісний з intel 8080, але міг виконувати всі написані для нього програми. Він став використовуватися в комп'ютерах TRS-80 Model 1, випущеної фірмою Radio Shack. Цей же процесор встановлювався в комп'ютери Osborne і Kaypro.

У березні 1976 року фірма intel випустила процесор 8085, який містив 6500 транзисторів, працював на частоті 5 МГц і проводився по 3-мікронній технології. В цьому ж році фірма MOS Technology випустила процесор 6205, який був абсолютно не схожий на процесори фірми intel. Він був розроблений групою інженерів фірми Motorola. Ця ж група працювала над створенням процесора 6800, який в майбутньому трансформувався в сімейство процесорів 68000. Відмінною рисою процесора 6205 від intel 8080 його ціна 25 доларів, на відміну від 300 доларів. Саме на цьому процесорі були створені перші моделі комп'ютерів Apple I Apple II, а також ігрові приставки Nintero. Процесори серії 68000, які тепер називаються PowerPC, в даний час використовуються в комп'ютерах Apple Macintosh.

У червні 1978 року компанія intel випустила перший 16-розрядний процесор intel 8086, який містив набір команд x86 (цей набір команд підтримується в найсучасніших комп'ютерах). Він став базою для персональних комп'ютерів IBM PC XT, що стали стандартом де-факто для всієї комп'ютерної індустрії. Цей процесор був повністю 16-розрядним - внутрішні регістри і шина даних. Він містив 29000 транзисторів і працював на частоті 5 МГц. Завдяки 20-розрядній шині адреси він міг адресувати 1 Мбайт пам'яті. Коштував цей процесор дорого і в зв'язку з цим компанія intel випустила в 1979 р. "дешеву" версію цього процесора під кодовою назвою intel 8088. Оригінальний процесор intel 8088 містив 30000 транзисторів і працював на частоті 5 МГц. Цей процесор відрізнявся від попереднього 8-розрядною шиною даних.

Потім з'явилися процесори 80286, 80386 (перший 32-розрядний процесор), 80486. З появою у 1995 р. процесорів Pentium почався новий етап розвитку персональних комп'ютерів, коли вони стали не тільки робочим інструментом, а й домашнім, побутовим пристроєм повсякденного використання.

У 1995 р. компанія AMD випускає процесори сімейства 5x86. При розробці даної архітектури компанія AMD за великим рахунком, слідувала у фарватері технічних рішень корпорації Intel. У 1997 р. компанія AMD вперше використовує мікро-архітектуру RISC для побудови виконавчих конвеєрів. Тому довелося вводити блок перекомпілювання звичайних CISC-інструкцій довільної довжини у набори RISC-команд фіксованої довжини. Ця політика була запропонована в процесорах сімейства K6-III випущених 1999 р. Однак тільки з виходом процесора Athlon (ядро мікро-архітектури K7) компанія продемонструвала повністю незалежні технічні рішення.

З 1995 р і по теперішній час різні фірми випустили понад 150 моделей процесорів для персональних комп'ютерів. Деякі з них стали знаковими явищами в комп'ютерній індустрії. Основними виробниками процесорів для ПК стали компанії Intel і AMD. Корпорація Intel з 1995 р і по теперішній час використовувала сім моделей платформ з різними інтерфейсами процесорів: від Socket 5 до Socket 775. Компанія AMD використовувала трохи менше -

п'ять платформ (від Socket 7 до Socket 939). Таким чином, в середньому актуальність платформи зберігалася протягом півтора-двох років.

6. Характеристики процесора.

При описі мікропроцесора виділяють наступні характеристики:

- 1) швидкодія (Тактова частота);
- 2) технологічна норма;
- 3) розрядність (шини даних, адреси, регістрів);
- 4) КЕШ пам'ять;
- 5) інтерфейс (процесорний роз'єм);
- 6) кількість транзисторів.

1) Швидкодія.

Швидкодія - це одна з характеристик процесора, яка багато в чому залежить від тактової частоти, зазвичай вимірюється в мегагерцах (МГц). Вона визначається параметрами кварцового резонатора, що представляє собою кристал кварцу, укладений в невеликий олов'яний контейнер. Під впливом електричної напруги в кристалі кварцу виникають коливання електричного струму з частотою, яка визначається формою і розміром кристала. Частота цього змінного струму і називається тактовою частотою. Мікросхеми звичайного комп'ютера працюють на частоті декількох мільйонів герц. (Герц - одне коливання в секунду.) Швидкодія вимірюється в мегагерцах, тобто в мільйонах циклів у секунду.

2) Технологічна норма.

Технологічна норма виробничого процесу визначає характерну мінімальну відстань між двома сусідніми елементами в одному шарі мікросхеми, тобто своєрідний крок сітки, до якої здійснюється прив'язка елементів мікросхеми.

3) Розрядність (шини даних, адреси, регістрів).

Шина даних.

Продуктивність і розрядність зовнішньої шини даних є основними характеристиками центрального процесора, що визначають швидкодію, з яким дані передаються в процесор або з нього.

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. Розрядність шини даних подібна кількості смуг руху на швидкісній автомагістралі; точно так же, як збільшення кількості полос дозволяє збільшити потік машин по трасі, збільшення розрядів дозволяє підвищити продуктивність.

Шина адреси.

Шина адреси являє собою набір провідників; по ним 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), що використовуються для формування адреси, дозволяє збільшити кількість адресованих комірок. Розрядність шини адреси визначає максимальний обсяг пам'яті, що адресується процесором.

Регістри.

Кількість бітів даних, які може обробити процесор за один прийом, характеризується розрядністю внутрішніх регістрів. Регістр - це, по суті, комірка 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.

4) Кеш-пам'ять.

Кеш-пам'ять першого рівня.

Кеш - це швидкодіюча пам'ять, призначена для тимчасового зберігання програмного коду і даних. Звернення до вбудованої кеш-пам'яті відбуваються без станів очікування, оскільки її швидкодія відповідає можливостям процесора, тобто кеш-пам'ять першого рівня) працює на частоті процесора.

Завдяки цьому обмін даними з відносно повільної системної пам'яттю значно прискорюється. Процесору не потрібно чекати, поки чергова порція програмного коду або даних надійде з основної області пам'яті, а це призводить до відчутного підвищення продуктивності комп'ютера. При відсутності кеш-пам'яті такі паузи виникали б досить часто.

Кеш-пам'ять другого рівня.

Коли в кеш-пам'яті першого рівня дані відсутні, на сцену виходить кеш-пам'ять другого рівня. Використовуючи аналогію з обідом, можна сказати, що кеш-пам'ять другого рівня являє собою візок офіціанта, на якому знаходяться страви, що найбільш часто замовляються. Для того щоб отримати одне з них, потрібно 15 с. У системах на базі процесора Pentium (Socket 7) кеш-пам'ять другого рівня працює на частоті системної плати-66 МГц (15 нс).

5) Інтерфейс (процесорний роз'єм).

Корпуси процесорів:

- корпус PGA
- Корпуси SEC і SEP
- Slot 1
- Socket 7
- Socket A
- Socket 755 та інші

Лекція 4. Режими роботи процесора.

1. Реальний режим роботи процесора (Real Mode - RM).

Широко відомо, що першим мікропроцесором, на базі якого було створено IBM PC, був Intel 8088. Цей мікропроцесор відрізнявся від першого 16-розрядного мікропроцесора фірми Intel - 8086 - перш за все тим, що у нього була 8-бітова шина даних, а не 16-бітова (як у 8086). Обидва ці мікропроцесори призначалися для створення обчислювальних пристроїв, які б працювали в однозадачних режимі, тобто спеціальних апаратних засобів для підтримки надійних і ефективних мультипрограмних ОС в них не було.

Однак на той час, коли розробники усвідомили необхідність включення в мікропроцесор спеціальної апаратної підтримки для мультипрограмних обчислень, вже було створено дуже багато програмних продуктів. Тому для сумісності з першими комп'ютерами в наступних версіях мікропроцесорів була реалізована можливість використовувати їх в двох режимах – реальному (Real mode - так називали режим роботи перших 16-бітових мікропроцесорів) і захищеному (Protected mode означає, що паралельні обчислення можуть бути захищені апаратно-програмними механізмами).

В перших 16-бітових процесорах i8086 / i8088 (а значить, і в інших мікропроцесорах сімейства i80x86 при роботі їх в реальному режимі) звернення до пам'яті з можливим адресним простором в 1 Мбайт здійснюється за допомогою механізму сегментної адресації. Цей механізм був використаний для збільшення кількості розрядів, що беруть участь в адресі комірки пам'яті, з якої в даний момент здійснюється робота, з 16 до 20 і тим самим збільшення обсягу пам'яті.

Для адресації операндів використовується аналогічний механізм, тільки беруть участь в цьому випадку інші сегментні регістри. Нагадаємо, що для визначення фізичної адреси команди вміст сегментного регістра CS (code segment) множиться на 16 за рахунок додавання праворуч (до молодших бітів) чотирьох нулів, після чого до отриманого значення додається вміст покажчика команд (реєстр IP, instruction pointer). Виходить двадцяти бітове значення, яке і дозволяє вказати будь-який байт з 2^{20} .

Real Mode – режим реальної адресації, відповідний роботі системи з CPU 8086, використовується тільки в MS DOS. Область адрес, доступних системі (1 Мб), не захищена. Реалізовано двадцять адресних ліній, режим однокористувацький. Однак при цьому працюють 32-бітові регістри CPU. За замовчуванням використовуються всі наявні команди, довжина операндів 16 біт. Для роботи з 32-розрядними операндами та використання додаткових режимів адресації застосовують спеціальний префікс переадресації. 32-розрядний адреса не перевищує межі сегменту 64 Кбайт (0000H-FFFFH), в іншому випадку констатується особлива ситуація – 13H. Виконавча адреса завжди відповідає фізичній, сторінковий механізм відключений. Всі сегменти можуть перебувати в стані запису, зчитування або виконання.

2. Захищений режим роботи (Protected Mode – PM).

У захищеному режимі роботи визначення фізичної адреси здійснюється зовсім інакше. Перш за все використовується сегментний механізм для організації віртуальної пам'яті. При цьому адреси задаються 32-бітовими значеннями. Крім цього, можлива сторінкова трансляція адрес, також з 32-бітовими значеннями.

Protected Mode – режим віртуальної адресації.

Віртуальна адресація – це спосіб організації доступу до інформації, при якому більша її частина розташовується не в фізичному ОЗП, а в пристроях зовнішньої пам'яті (ПЗП), звідки вона періодично підкачується в ОЗП (swapping), що створює ілюзію розширення його розмірів.

Нарешті, при роботі в захищеному режимі, який за умовчанням передбачає 32-бітовий код, можливо виконання програм, створених для роботи мікропроцесора в 16-бітовому режимі. Для цього введено режим віртуальної 16-бітової машини і 20-бітові адреси реального режиму, які трансльються за допомогою сторінкового механізму в 32-бітові значення захищеного режиму. Нарешті, є ще один режим - 16-бітовий захищений, що дозволяє 32-бітовим мікропроцесорам виконувати захищений 16-бітовий код, який був характерний для мікропроцесора 80286. Слід зазначити, що це останній режим практично не використовується, оскільки програм, створених для нього, не так вже й багато.

Захищений режим дозволяє системі запустити кілька захищених один від одного процесів. Структура цього режиму наведена на рис. 1.

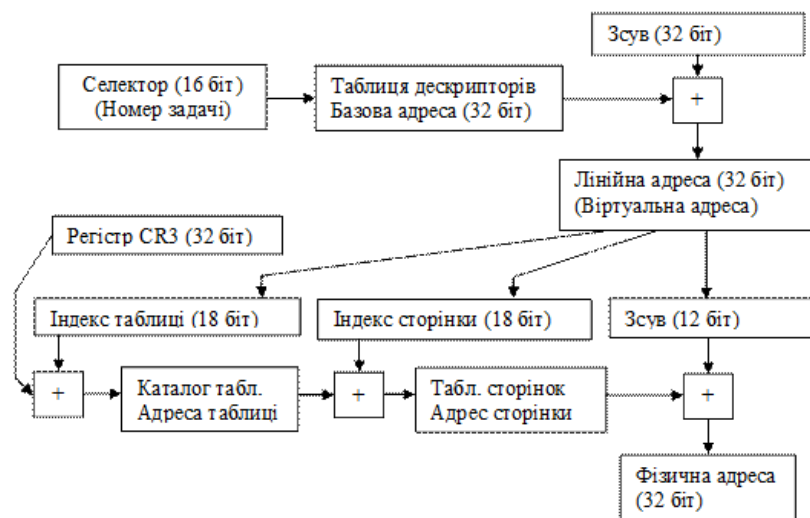


Рис. 1. Формування лінійної адреси в захищеному режимі.

Захист пам'яті за допомогою сегментації забороняє:

- використовувати сегменти не за призначенням (наприклад, намагатися трактувати область даних як коди інструкцій);
- порушувати права доступу (намагатися модифікувати сегмент, призначений тільки для читання, звертатися до сегмента, не маючи достатніх привілеїв, та т.п.);

- адресуватися до елементів, що виходять за ліміт сегмента;
- змінювати вміст таблиць дескрипторів (тобто параметрів сегментів), не маючи достатніх привілеїв.

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 (багатовимірність);
- 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.

3. Віртуальний реальний режим (Protected Mode 86 – VM 86).

Для зворотної сумісності 32-розрядна система Windows 9x використовує третій режим в процесорі – віртуальний реальний режим. Віртуальний реальний, по суті, є режимом виконання 16-розрядного середовища (реальний режим), який 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 багатовимірним, 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 РС 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.

Тема 5. Покоління процесорів.

1. Склад процесору.

Центральний процесор (CPU – Central processing unit) – це основний робочий компонент комп'ютера, який виконує арифметичні та логічні операції (задані програмою), керує обчислювальним процесом та координує роботу усіх пристроїв комп'ютера.

Центральний процесор містить у собі:

- 1) арифметико – логічні пристрої;
- 2) шини даних і шини адрес;
- 3) регістри;
- 4) лічильники команд;
- 5) кеш;
- 6) математичний співпроцесор з плаваючою комою.

2. Історичні процесори. 4-бітові процесори.

Перший 4-х бітовий процесор **Intel 4004** представлений 15 листопада 1971 року.

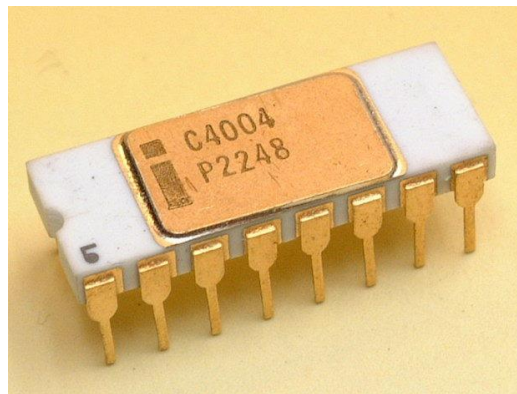


Рис. 1. Intel 4004.

Основні характеристики:

- максимальна тактова частота 740 кГц. У багатьох джерелах наводиться інше, невірне значення максимальної тактової частоти — 108 кГц; ця цифра наводиться на деяких інтернет-сторінках самої фірми Intel. Мінімальний час циклу інструкції 4004 становить 10,8 мікросекунд і, швидше за все, хтось колись переплутав цю цифру з максимальною тактовою частотою. На жаль, ця помилка отримала дуже широке поширення.
- містив у собі 2300 транзисторів;
- технологія виробництва 10 мкм PMOS;
- ширина шини: 4 біта;
- адресна пам'ять: 640 байт;
- пам'ять для програми: 4 Кбайта;
- використовувався в калькуляторі Busicom;
- 8-ми бітові процесори.

3. I покоління МП.

1 квітня 1972 року представлений 8-ми бітовий процесор **Intel 8008**.



Рис. 2. Intel 8008.

Основні характеристики:

- містив 3500 транзисторів;
- тактова частота від 200 до 800 кГц;
- технологія виробництва 10 мкм;
- ширина шини даних 8 біт;
- адресна пам'ять 16 кілобайт;
- використовувався терміналах, калькуляторах, розливочних машинах.

1-го квітня 1974 року був представлений **Intel 8080** з характеристиками:

- тактова частота 2 МГц;
- ширина шини даних 8 біт, адреси 16 біт (максимальний обсяг пам'яті 64 КВ);
- 6000 транзисторів;

На його базі був зібраний мікрокомп'ютер **Altair 8080**, також використовувався у деяких контролерах світлофорів та як вбудований процесор у ракетах

16-бітові процесори x86.

8 червня 1978 року виходить **16-бітовий 8086** (позначення x86).

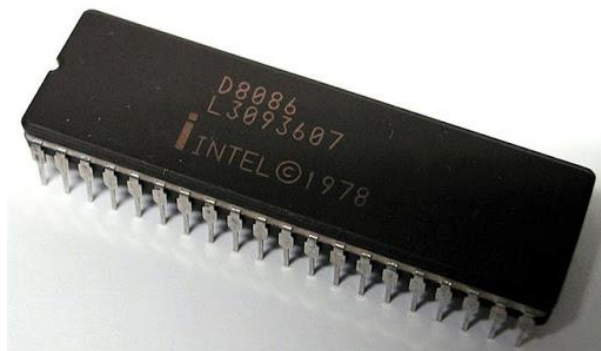


Рис. 3. Мікропроцесор 8086.

Основні характеристики:

- тактові частоти: 5 МГц з швидкодією 0,33 MIPS; 8 МГц з швидкодією 0,66 MIPS; 10 МГц з швидкодією 0,75 MIPS;
- ширина шини: 16 біт – дані, 20 біт – адреси;
- кількість транзисторів: 29000;
- технологія: 3 мкм;

- адресна пам'ять: 1 Мбайт.

Використовувався в портативних обчисленнях

Intel 8088 представлений 1 червня 1979 року.

- внутрішня архітектура 16 біт;
- ширина зовнішніх шин: даних 8 біт, адреси 20 біт;
- кількість транзисторів: 29000;
- технологічний процес 3 мкм;
- адресна пам'ять 1 мегабайт;
- ідентичний до 8086 крім того, що у 8088 зовнішня шина 8 біт;

Використовувався у комп'ютерах IBM PC та інших клонах IBM PC.

4. II покоління МП.

1-го лютого 1982 року представлений **Intel 80286**.



Рис. 4. Intel 80286.

Основні характеристики:

- тактові частоти: 6 МГц з швидкодією 0,9 MIPS; 8 МГц, 10 МГц з швидкодією 1,5 MIPS; 12,5 МГц з швидкодією 2,66 MIPS;
- ширина шини даних: 16 біт;
- ширина шини адреси: 24 біт;
- кількість транзисторів: 134000;
- технологія: 1,5 мкм;
- адресна пам'ять: 16 Мбайт;
- включав апаратний захист пам'яті для підтримки багатозадачних операційних систем з окремим адресним простором для кожного процесу. Захист базувався на сегментних дескрипторах і, через відсутність прямої адресації блоків пам'яті більше 64-х кілобайт, сегментація пам'яті була явною для програм: програмі доводилося перемикаати сегменти.
- 3-6-разова продуктивність 8086;

Широко використовувався в клонах PC того часу.

5. III покоління МП.

32 – бітові процесори i386.

Моделі i386 : i386DX, i386SX, i386SL, i386EX.

Перший **32-бітовий** процесор Intel i386DX виходить 17 жовтня 1985 року.

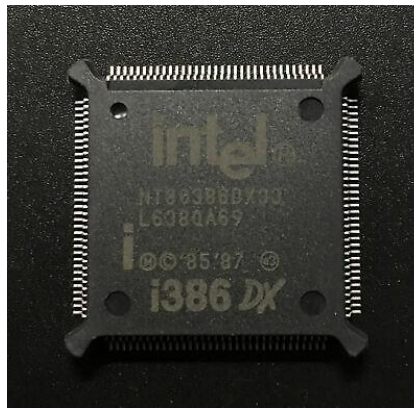


Рис. 5. Intel i386DX.

Основні характеристики:

- тактові частоти: 16 МГц з швидкістю від 5 до 6 MIPS; 16 лютого 1987 20 МГц з швидкістю від 6 до 7 MIPS; 4 квітня 1988 25 МГц з швидкістю 8,5 MIPS; 10 квітня 1989 33 МГц з швидкістю 11,4 MIPS (9.4 SPECint92 на Compaq / i 16K L2);
- ширина шини: 32 біт;
- кількість транзисторів: 275000;
- технологія: 1 мкм;
- адресна пам'ять: 4 Гбайта;
- перший чип x86 для підтримки 32-бітних наборів даних
- перероблена і розширена підтримка захисту пам'яті, включає сторінкову віртуальну пам'ять і режим віртуального 86 (особливості, які в майбутньому будуть потрібні для Windows 95, OS/2 Warp і Unix); Використовувався в настільних комп'ютерах.

16 червня 1988 року представлений **Intel i386SX** з характеристиками:

- Тактові частоти: 16 МГц з швидкістю 2.5 MIPS; 25 січня 1989 20 МГц з швидкістю 2.5 MIPS, 25 МГц з швидкістю 2.7 MIPS; 26 жовтня 1992 33 МГц з швидкістю 2.9 MIPS
- Внутрішня архітектура: 32 біт
- Зовнішня шина: 16 біт
- Кількість транзисторів: 275,000
- Технологія: 1 мкм
- Адресна пам'ять: 16 Мбайт
- Вбудована багатозадачність
- Використовувався в настільних комп'ютерах початкового рівня і портативних обчислювальних пристроях

15 жовтня 1990 року представлений **Intel i386SL** (високоінтегрований, включає контролери кешу, шини і пам'яті).

У серпні 1994 року представлений **Intel i386SX**.

6. IV покоління МП.

10 квітня 1989 року представлений **Intel i486** – 32-бітний МП.

Основні характеристики:

- тактові частоти: 25 МГц з швидкодією 20 MIPS; 7 травня 1990 33 МГц з швидкодією 27 MIPS; 24 червня 1991 50 МГц з швидкодією 41 MIPS;
- ширина шини: 32 біта;
- кількість транзисторів: 1,2 мільйона;
- технологія: 1 мкм; 50 МГц версія була на 0,8 мкм;
- адресна пам'ять: 4 Гбайта;
- кеш першого рівня на чипі;
- вбудований математичний співпроцесор;
- 50-разова продуктивність 8088.

Використовувався в настільних і серверних системах.

7. МП V покоління – Pentium, P5.

22 березня 1993 року представлений МП V покоління – Pentium.

Основні характеристики:

- Мікропроцесор для настільних систем з підтримкою симетричної багатопроцесорності (SMP), обмеженою двома мікропроцесорами;
- ширина шини: 64 біт;
- частота системної шини: 60 або 66 МГц;
- ширина шини адреси: 32 біт;
- адресна пам'ять: 4 гігабайти;
- напруга живлення: 5 В;
- кеш L1: 16 КБ.

Використовувався в настільних комп'ютерах.

Основні відмінності від процесора i486:

А) Суперскалярна архітектура. Завдяки використанню суперскалярної архітектури процесор може виконувати 2 команди за 1 такт. Така можливість існує завдяки наявності двох конвеєрів – u-і v-. u-конвеєр – основний, виконує всі операції над цілими і дійсними числами; v-конвеєр – допоміжний, виконує тільки прості операції над цілими і частково над дійсними. Щоб старі програми (для 486) повною мірою використали можливості такої архітектури, необхідно було їх перекомпілювати. Pentium – перший CISC-процесор, що використовує багатоконвеєрну архітектуру.

Б) 64-бітна шина даних дозволяє процесору Pentium за один шинний цикл обмінюватися вдвічі більшим обсягом даних з оперативною пам'яттю, ніж 486.

В) Роздільне кешування програмного коду і даних. У процесорах Pentium використовується кеш-пам'ять першого рівня (кеш L1) обсягом 16 Кб, розділена на 2 сегменти: 8 Кб для даних і 8 Кб для інструкцій. Це покращує

продуктивність і дозволяє робити подвійне кешування доступним частіше, ніж це було можливо раніше. Крім того, змінено механізм кешування.

Г) Покращений блок обчислень з плаваючою крапкою (FPU).

Д) Симетрична багатопроцесорна робота (SMP).

8. МП VI покоління - Pentium II, Pentium III.

Відлік шостого покоління процесорів почався з Pentium Pro, випущеного в 1995 році. Зараз до цього покоління відносяться Pentium II (1997 р.), Celeron, Xeon (1998 р.) і, нарешті, Pentium III (1999 р.). Від попереднього покоління ці процесори головним чином відрізняє застосування "динамічного виконання" (зміни порядку виконання інструкцій) і архітектура подвійної незалежної шини. Тут вторинному кешу, введеному в процесор (але не в усі моделі), виділяється окрема високошвидкісна магістраль. В ході еволюції покоління до системи команд Pentium Pro, розширеної щодо Pentium з метою скорочення умовних переходів, було додано розширення MMX - так з'явився Pentium II. Тепер ідею MMX – одночасне виконання однієї інструкції над групою операндів -розповсюдили і на інструкції з плаваючою точкою: SSE (Streaming SIMD Extensions) – основний козир Pentium III. Правда, трохи раніше те ж саме (але в меншому обсязі) було зроблено фірмою AMD – розширення 3DNow! було реалізовано вже в процесорах K6-2 для сокета 7.

Основні характеристики:

- поява першого мп 7 травня 1997 року;
- частота ядра 233 - 450 МГц;
- розрядність шини адрес: 36 біт;
- розрядність шини даних: 64 біт;
- кількість транзисторів: 7,5 млн.;
- кеш першого рівня: 16 Кб (кеш даних) + 16 Кб (кеш інструкцій);
- кеш другого рівня 512 Кб;
- технологія виробництва 350 нм.;
- інтерфейс – Slot 1;

Кодові імена: Klamath, Deschutes.

26 лютого 1999 року був анонсований Intel Pentium III. Ядро Pentium III являє собою модифіковане ядро Deschutes. У порівнянні з попередником розширено набір команд і оптимізована робота з пам'яттю. Це дозволило підвищити продуктивність як в нових програмах, що використовують розширення SSE, так і в існуючих (за рахунок зрослої швидкості роботи з пам'яттю). Також був введений 64-бітний серійний номер, унікальний для кожного процесора.

9. МП VII покоління - Pentium IV, Core Duo, Duo Core, Core II Duo.

Intel Pentium 4 - одноядерний x86-сумісний процесор компанії Intel, представлений 20 листопада 2000 року, який став першим мікропроцесором, в основі якого лежала принципово нова в порівнянні з попередниками архітектура сьомого покоління (за класифікацією Intel) - NetBurst. Крім різних

варіантів Pentium 4, до процесорів архітектури NetBurst відносяться двоядерні процесори Pentium D, а також деякі процесори Xeon, призначені для серверів. Крім того, частина процесорів Celeron для систем нижнього цінового рівня являє собою Pentium 4 з частково відключеним кешем другого рівня.

Виробництво процесорів Pentium 4 було розпочато в 2000 році. З середини 2005 року почалося їх поступове витіснення в нижню цінову категорію двоядерними процесорами Pentium D. 27 липня 2006 з'явилися перші процесори сімейства Core 2 Duo, що прийшли на зміну процесорам архітектури NetBurst, а 8 серпня 2007 року компанія Intel оголосила про запуск програми по зняттю з виробництва всіх процесорів архітектури NetBurst.

Однією з особливостей архітектури процесора Pentium 4 є Гарвардська внутрішня структура, що реалізується шляхом розділення потоків команд і даних, що надходять від системної шини через блок зовнішнього інтерфейсу і розміщену на кристалі процесора загальну кеш-пам'ять 2-го рівня L2.

Блок зовнішнього інтерфейсу реалізує обмін процесора з системною шиною, до якої підключається пам'ять, контролери введення / виведення і інші активні пристрої системи. Обмін по системній шині здійснюється за допомогою 64-розрядної двобічної шини даних, 41-розрядної шини адреси (33 адресних ліній і 8 ліній вибору байтів), що забезпечує адресацію до 64 Гб зовнішньої пам'яті.

Архітектура NetBurst (робоче найменування - P68), що лежить в основі процесорів Pentium 4, розроблялася компанією Intel, в першу чергу, з метою досягнення високих тактових частот процесорів. NetBurst не є розвитком архітектури P6, що використовувалася в процесорах Pentium III, а являє собою принципово нову в порівнянні з попередниками архітектуру. Характерними особливостями архітектури NetBurst є гіперконвейерізація і застосування кеша послідовностей мікрооперацій замість традиційного кешу інструкцій.

На зміну NetBurst в 2006 році прийшла архітектура Intel Core. Однією з причин розробки цієї архітектури була неможливість збільшення частоти в NetBurst, а також її дуже велике тепловиділення. Ця архітектура була розрахована на розробку багатоядерних процесорів, розмір кеша першого рівня був збільшений до 64 Кб. Частота залишилася на рівні 3 ГГц, але зате була сильно знижена споживана потужність, а також техпроцес, до 60 нм.

Процесори на архітектурі Core підтримували апаратну віртуалізацію Intel-VT, та деякі розширення команд, але не підтримували Hyper-Threading, оскільки були розроблені на основі архітектури P6, де такої можливості ще не було.

10.Сучасні процесори: Core i3, Core i5, Core i7, Core i9.

Core i3 – один з найдешевших процесорів ,має два фізичних ядра для збільшення швидкодії застосовується технологія Intel Hyper Threading Technology. Ця технологія створює з 2 фізичних ядер 2 віртуальні, таким чином операційна система визначає, що процесор має 4 ядра. Кеш пам'яті 3-4 МБ

Core i5 – бюджетний процесор середнього рівня, зазвичай у цих процесорах 4 фізичних ядра, тільки деякі моделі можуть мати 2 фізичних ядра і 2 віртуальних. Кеш пам'яті зазвичай 6МБ, але бувають і моделі, що мають від 4 і до 8 МБ. Більш висока швидкодія досягається наявністю 4-х фізичних ядер і збільшеним об'ємом кеш пам'яті.

Core i7 – процесори мають від 4 до 8 фізичних ядер, з обов'язковим використанням технології INTEL HYPER THREADING TECHNOLOGY. Кеш пам'яті від 8МБ до 20МБ в останніх моделях процесорів. Швидкодія збільшена віртуальними ядрами і великим об'ємом кеш пам'яті. Процесори для мобільних пристроїв використовують 2 фізичні ядра.

Core i9 – від 6 до 8 фізичних ядер, кеш пам'яті від 10 МБ.

11. Покоління процесорів Intel Core i5.

Перше покоління (2009, Nehalem). Випущений процесор Intel Core i5 з архітектурою Nehalem в кінці 2009 року. Побудований за старим технологічним процесом 45 нм, але мав 4 ядра на одному кристалі (у C2Q було 2 кристали по 2 ядра). Моделі, які були випущені: i5 – 750S, 750 і 760. Чіпи цього покоління не мали вбудованої графіки, вбудовувалися в плати з сокетом 1156, і працювали з пам'яттю DDR3. Також Intel Core i5 вперше отримали підтримку автоматичного розгону Turbo Boost, який дозволяв збільшити частоту при нерівномірному навантаженні на ядра.

Перше покоління (2010, Westmere). У 2010 році світ побачив процесори Core i5 Westmere, створені за технологічним процесом 32 нм. Проте, вони мали по 2 фізичних ядра з підтримкою НТ (HyperThreading – технологія опрацювання 2 потоків обчислень на 1 ядрі, яка дозволяла процесору працювати в 4 потоки) і мали нумерацію i5-6xx. Моделі: 650, 655K, 660, 661, 670 і 680. Особливістю Core i5 цієї серії стала поява вбудованого GPU.



Brand Name & Processor Number ¹	Base Clock Speed	Turbo Frequency ²	Cores/Threads	Cache	Memory Speed Support	TDP	Pricing (1Ku)
Intel® Core™ i5-670	3.46	Up to 3.73 GHz	2/4	4 MB	DDR3-1333	73W	\$284
Intel® Core™ i5-661	3.33	Up to 3.60 GHz	2/4	4 MB	DDR3-1333	87W	\$196
Intel® Core™ i5-660	3.33	Up to 3.60 GHz	2/4	4 MB	DDR3-1333	73W	\$196
Intel® Core™ i5-650	3.20	Up to 3.46 GHz	2/4	4 MB	DDR3-1333	73W	\$176
Intel® Core™ i3-540	3.06	N/A	2/4	4 MB	DDR3-1333	73W	\$133
Intel® Core™ i3-530	2.93	N/A	2/4	4 MB	DDR3-1333	73W	\$113

Рис. 6. Характеристики Core i3 та Core i5 першого покоління.

Друге покоління (2011, Sandy Bridge).

Архітектура Sandy Bridge – одна з найважливіших сторінок історії Intel. Чіпи на ній випускались за старим технологічним процесом 32 нм, але отримали більше внутрішньої оптимізації. Це дозволило їм суттєво перевершити своїх попередників. При однаковій частоті новий чіп був набагато швидшим ніж попередні. Процесори цієї серії мали нумерації Intel Core i5 – 2xxx. Модель під номером 2390Т мала два ядра з підтримкою HT, а інші (від 2300 до 2550К) – 4 ядра без HT. Для процесорів другого покоління був створений новий сокет 1155. Також було перенесено GPU на один кристал з CPU. Контролер пам'яті працював з планками DDR3.

Третє покоління (2012, Ivy Bridge).

Ivy Bridge – це друга версія попередньої архітектури. Процесори цієї серії відрізнялися від попередніх технологічним процесом 22 нм. Проте, їх внутрішній пристрій залишався таким самим, тому приріст ефективності тільки +5%, він досягався за рахунок підняття частот. Номера моделей- від 3330 до 3570К. i5 - 3470Т мав 2 ядра з підтримкою HT, а інші – 4 ядра без HT.

PROCESSOR AND BENEFITS	Processor Number	Clock Speed (GHz)	Cores/Threads	with Intel® Turbo Boost Technology 2.0	Intel® Turbo Boost Technology ¹	Max Turbo Frequency	with Intel® vPro™	Intel® Hyper-Threading ¹	Cache	Intel® HD Graphics	NM
 3rd Generation Intel® Core™ i5 Processors	i5-3570	3.4	4/4	✓	–	3.8	✓	–	6MB	2500	22
	i5-3570K	3.4	4/4	✓	–	3.8	✓	–	6MB	4000	22
	i5-3570S	3.1	4/4	✓	–	3.8	✓	–	6MB	2500	22
	i5-3570T	2.3	4/4	✓	–	3.3	✓	–	6MB	2500	22
	i5-3550	3.3	4/4	✓	–	3.7	✓	–	6MB	2500	22
	i5-3550S	3	4/4	✓	–	3.7	✓	–	6MB	2500	22
	i5-3475S	2.9	4/4	✓	–	3.6	✓	–	6MB	4000	22
	i5-3470	3.2	4/4	✓	–	3.6	✓	–	6MB	2500	22
	i5-3470T	2.9	2/4	✓	–	3.6	✓	✓	3MB	2500	22
	i5-3450	3.1	4/4	✓	–	3.5	✓	–	6MB	2500	22
	i5-3450S	2.9	4/4	✓	–	3.5	✓	–	6MB	2500	22
	i5-3330	3	4/4	✓	–	3.2	✓	–	6MB	2500	22
 2nd Generation Intel® Core™ i5 Processors	i5-2500	3.3	4/4	✓	–	3.7	✓	–	6MB	2000	32
	i5-2500K	3.3	4/4	✓	–	3.7	✓	–	6MB	3000	32
	i5-2500S	2.7	4/4	✓	–	3.7	✓	–	6MB	2000	32
	i5-2500T	2.3	4/4	✓	–	3.3	✓	–	6MB	2000	32
	i5-2405S	2.5	4/4	✓	–	3.3	✓	–	6MB	3000	32
	i5-2400	3.1	4/4	✓	–	3.4	✓	–	6MB	2000	32
	i5-2400S	2.5	4/4	✓	–	3.3	✓	–	6MB	2000	32
	i5-2390T	2.7	2/4	✓	–	3.5	✓	✓	3MB	2000	32
	i5-2320	3	4/4	✓	–	3.3	✓	–	6MB	2000	32
	i5-2310	2.9	4/4	✓	–	3.2	✓	–	6MB	2000	32
	i5-2300	2.8	4/4	✓	–	3.1	✓	–	6MB	2000	32

Рис. 7. Характеристики Core i5 другого та третього поколінь.

Четверте покоління (2013, Haswell).

Процесори Intel Core i5 4-го покоління були випущені на технологічному процесі 22 нм, але архітектура була покращеною. Великого збільшення ефективності це не дало, знову 5%, тільки ЦП став трошки енергоефективнішим. Процесори Intel Core i5 четвертого покоління називалися в форматі i5- 4xxx, з номерами від 4430 до 4690. Моделі i5 – 4570Т і ТУ були мали 2 ядра, інші – 4. Чіпи перевели на новий сокет 1150. Моделі 4570R I 4670R мали покращену графіку Iris Pro.



Рис. 8. Core i5 четвертого покоління.

П'яте покоління (2015, Broadwell).

Це покоління фактично було перехідним етапом, оскільки чіпи являли собою той же Haswell, тільки переведений на новий технологічний процес 14 нм. В цій серії вийшло тільки 3 чотириядерні моделі: i5-5575R, 5675C, 5675R. Всі моделі мали покращений графічний процесор Iris Pro, 128 Мб eDRAM пам'яті.

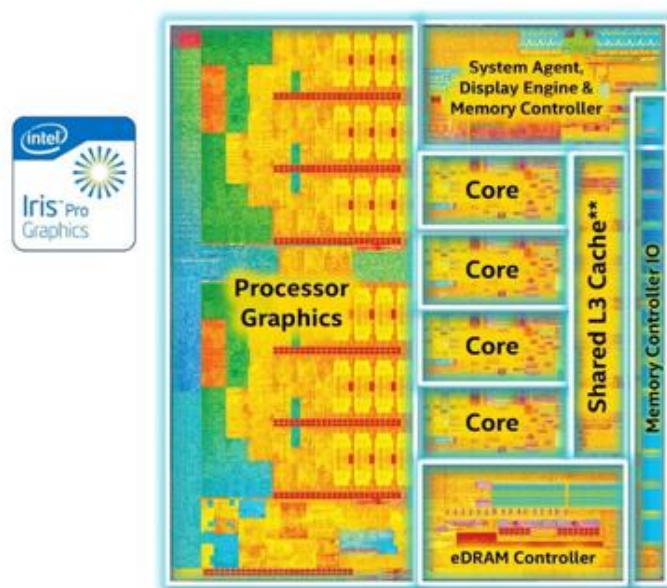


Рис. 9. Core i5 п'ятого покоління (Iris Pro Graphics 6200).

Шосте покоління (2015, Skylake).

Чіпи з архітектурою Skylake випускалися за технологічним процесом 14 нм та мали 4 ядра. Модельні номери процесорів від i5-6400 до 6600K, всі ЦП чотирьох ядерні. Великого приросту нова архітектура не дала, але чіпи мали ряд змін. По-перше, вони встановлювались на новий Socket 1151, по-друге – отримали комбінований контролер пам'яті DDR3/DDR4. У шостому поколінні теж виходили чіпи з графікою Iris Pro – i5-6585R I 6685R.



Рис. 10. Intel Core i5-6600K.

Сьоме покоління (2017, Kaby Lake).

Сьоме покоління майже нічим не відрізнялося від шостого. Технологічний процес залишився таким же, 14 нм, архітектура отримала тільки косметичні покращення, а невелике збільшення ефективності отримано тільки через підвищення частот. Чіпи цієї серії носять індекси i5-7xxx, номери моделей – від 7400 до 7600K. Сокет залишився таким самим (1151), контролер пам'яті теж не змінився. Виключення – модель i5-7640K, розрахована на Socket 2066.



Рис. 11. Intel Core i5-7600K.

Восьме покоління (2017, Coffee Lake, Cannon Lake).

Процесори Core i5 з архітектурою Coffee lake виготовлені за технологічним процесом 14 нм, архітектурно мінімально відрізняються від Skylake і Kaby Lake, мають приблизно таку ж саму ефективність в розрахунку на одне ядро. Проте, збільшилося число ядер з 4 до 6, тим самим це збільшило швидкодію до 1.5 рази на фоні попередників. У серії випущено чіпи з іменами формату i5-8xxx, і номерами від 8400 до 8600K. Сокет чіпів залишився таким самим (Socket 1151).

NEW 8 TH GEN INTEL® CORE™ PROCESSOR FAMILY									
Processor number ¹	Base clock speed (GHz)	Intel® Turbo Boost Technology 2.0 maximum single core turbo frequency (GHz)	Cores/Threads	Thermal Design Power	Unlocked ¹	Intel® Smart Cache	Memory support	Intel® Optane™ Memory Support ⁴	RCP pricing (USD 1K)
Intel® Core™ i7-8700K	3.7	4.7	6/12	95	✓	12 MB	Two channels DDR4-2666 ²	✓	\$359
Intel® Core™ i7-8700	3.2	4.6	6/12	65		12 MB	Two channels DDR4-2666 ²	✓	\$303
Intel® Core™ i5-8600K	3.6	4.3	6/6	95	✓	9 MB	Two channels DDR4-2666 ²	✓	\$257
Intel® Core™ i5-8400	2.8	4	6/6	65		9 MB	Two channels DDR4-2666 ²	✓	\$182
Intel® Core™ i3-8350K	4	N/A	4/4	91	✓	6 MB	Two channels DDR4-2400 ³	✓	\$168
Intel® Core™ i3-8100	3.6	N/A	4/4	65		6 MB	Two channels DDR4-2400 ³	✓	\$117

Рис. 12. Порівняльна характеристика процесорів 8 покоління.

Дев'яте покоління (2018, Coffee Lake).

9 покоління – технологія 14nm, графіка Intel UHD 630, реліз 2018-2019 роки. Конструктор в різних поєднаннях в залежності від типу процесора:

- Технологія Intel® Turbo Boost 2.0;
- Технологія Intel® Hyper-Threading;
- Технологія Intel® Smart Cache;
- Інтегрований контролер пам'яті;
- Графічне рішення Intel® UHD;
- Технологія Intel® Quick Sync Video;
- Оверклокінг ядер процесора, пам'яті і графіки;
- Інтерфейс PCI Express * 3.0;
- Підтримка пам'яті Intel® Optane™ Intel® Power Optimizer.

Десяте покоління (кінець 2019, Ice Lake).

Ice Lake – кодове ім'я сімейства процесорів Intel 10-го покоління, на мікро архітектурі Sunny Cove. Чіпи виготовляються на техпроцесі 10 нм. Сімейство IceLake–послідовник мобільної лінійки CannonLake, чіпи якої виготовлялися за нормою 10 нм, але не мали робочого вбудованого графічного ядра.

Серия ⇅	Модель ⇅	Ядра (потоки) ⇅	Штатная частота ЦП ⇅	Turbo частота ЦП [ГГц]			GPU			Кэш L3 ⇅	
				Кол-во ядер			Серия ⇅	EUs ⇅	Макс. частота ⇅		
				1 ⇅	2 ⇅	4 ⇅					
Core i7	1068G7	4 (8)	2,3 ГГц	4,1		3,6	Iris Plus	64	1,1 ГГц	8 МБ	
	1065G7 ↗		1,3 ГГц	3,9		3,5					
	1060G7 ↗		1,0 ГГц	3,8		3,4					
Core i5	1035G7 ↗		1,2 ГГц	3,7		3,3	UHD	48	1,05 ГГц	6 МБ	
	1035G4 ↗		1,1 ГГц								
	1035G1 ↗		1,0 ГГц	3,6	3,5		3,2	Iris Plus			
	1030G7 ↗		0,8 ГГц								
	1030G4 ↗		0,7 ГГц								
Core i3	1005G1 ↗	2 (4)	1,2 ГГц	3,4		N/A	UHD	32	0,9 ГГц	4 МБ	
	1000G4 ↗		1,1 ГГц	3,2			Iris Plus	48			
	1000G1 ↗						UHD	32			

Рис. 13. Мобільні процесори Ice Lake.

Одинадцять поколінь (2020, Tiger Lake).

Tiger Lake – кодове ім'я мікроархітектури 11-го покоління процесорів фірми Intel. Чіпи на цій мікроархітектурі виготовляються на технологічному процесі 10 нмв. Мікроархітектура Tiger Lake – спадкоємиця мікроархітектури Ice Lake, чіпи якої також виготовляються за нормою 10 нанометрів. Згідно етапів розробки процесорів Intel – це так званий етап «так», на якому виробничий техпроцес не змінюється, а оптимізується.

Важливе нововведення – вбудована підтримка інтерфейсу Thunderbolt 4 з пропускною спроможністю 40 Гбіт/с. Ніяких подробиць про цей інтерфейс і його ключові відмінності від Thunderbolt 3 немає. Ще заявлена підтримка Wi-Fi 6.

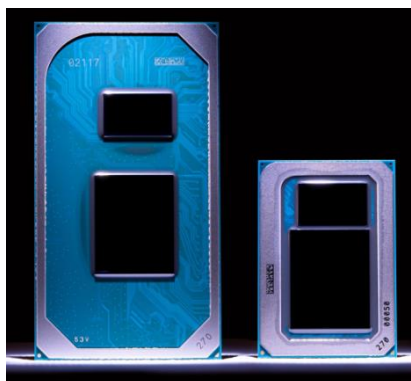


Рис. 13 Мікропроцесор Tiger Lake.

Тема 6. Організація мікроконтролерів. Процесорне ядро мікроконтролера.

1. Мікроконтролер та мікропроцесор.

Основною особливістю сучасного етапу розвитку мікропроцесорних систем (МПС) є завершення переходу від систем, виконаних на основі декількох великих ІС, до однокристальних мікроконтролерів (МК), які поєднують в одному кристалі всі основні елементи такі, як: центральний процесор (ЦП), постійний запам'ятовуючий пристрій (ПЗП), оперативний запам'ятовуючий пристрій (ОЗП), порти вводу/виводи, таймери. **Що ж таке мікроконтролер?**

Мікроконтролер (МК) — це програмований електронний пристрій, який містить у своєму складі центральний процесор, оперативну пам'ять, постійну пам'ять та периферійні модулі на одному кристалі. Завдяки цьому мікроконтролери є автономними та не потребують додаткових компонентів для виконання базових обчислювальних завдань.

На відміну від персональних комп'ютерів або мікропроцесорних систем, які призначені для широкого спектра завдань, мікроконтролери зазвичай застосовуються для виконання конкретних завдань у вбудованих системах.

Відмінність мікроконтролера від мікропроцесора

Часто мікроконтролери плутають із мікропроцесорами, але між ними є суттєва різниця:

Параметр	Мікроконтролер	Мікропроцесор
Основна функція	Виконує конкретні завдання у вбудованих системах	Обробка даних у багатофункціональних системах
Компоненти	Вбудовані пам'ять, інтерфейси та периферія	Потребує зовнішніх модулів для роботи
Використання	Керування пристроями, автоматика, IoT	Комп'ютери, сервери, складні обчислення
Споживання енергії	Мінімальне (енергоефективні рішення)	Високе (потужні процесори)

Таким чином, мікроконтролери є більш компактними, енергоефективними та орієнтованими на спеціалізовані завдання, тоді як мікропроцесори використовуються для високопродуктивних обчислень у комп'ютерах.

Галузі застосування мікроконтролерів

Завдяки універсальності та низькому енергоспоживанню мікроконтролери широко використовуються у багатьох сферах:

1. **Побутова техніка** — управління пральними машинами, холодильниками, мікрохвильовими печами.

2. **Автомобільна електроніка** – системи впорскування палива, ABS, подушки безпеки, бортові комп'ютери.
3. **Промисловість** – автоматизовані системи керування виробництвом, роботи, датчики контролю якості.
4. **Медицина** – кардіостимулятори, електронні термометри, діагностичне обладнання.
5. **Інтернет речей (IoT)** – розумні будинки, датчики безпеки, автоматичне освітлення.

Від простих електронних пристроїв до складних автоматизованих систем – мікроконтролери стали незамінним компонентом сучасної техніки.

2. Класифікація і структура мікроконтролерів.

В даний час промисловістю випускається цілий ряд типів МК. Усі ці прилади можна умовно поділити на три основних класи:

- 8-розрядні МК для вбудовуваних задач;
- 16- і 32-розрядні МК;
- цифрові сигнальні процесори (DSP).

Найрозповсюдженішим представником сімейства МК є 8-розрядні прилади, які широко застосовуються в промисловості, побутовій і комп'ютерній техніці. Вони пройшли у своєму розвитку шлях від найпростіших приладів з відносно слабorozвиненою периферією до сучасних багатофункціональних контролерів, що забезпечують реалізацію складних алгоритмів управління в реальному масштабі часу. Причиною життєздатності 8-розрядних МК є використання їх для управління реальними об'єктами, де застосовуються, в основному, алгоритми з перевагою логічних операцій, швидкість обробки яких практично не залежить від розрядності процесора.

Зростанню популярності 8-розрядних МК сприяє постійне розширення номенклатури виробів, що випускаються такими відомими фірмами, як Motorola, Microchip, Intel, Zilog, Atmel і багатьма іншими. Сучасні 8-розрядні МК володіють, як правило, кількома відмітними ознаками. Перелічимо основні з них:

- **модульна організація**, при якій на базі одного **процесорного ядра** (центрального процесора) проектується ряд (лінійка) МК, що відрізняються об'ємом і типом **пам'яті програм**, об'ємом **пам'яті даних**, набором периферійних модулів, частотою синхронізації;
- **використання закритої архітектури МК**, що характеризується відсутністю ліній магістралей адреси і даних на виводах корпусу МК. Таким чином, МК – це закінчена система обробки даних, нарощування можливостей якої з використанням паралельних магістралей адреси і даних не передбачається;
- **використання типових функціональних периферійних модулів** (таймери, процесори подій, контролери послідовних інтерфейсів, аналого-цифрові перетворювачі й ін.), які мають незначні відмінності в алгоритмах роботи в МК різних виробників;

- **розширення кількості режимів роботи периферійних модулів**, які задаються в процесі ініціалізації регістрів спеціальних функцій МК.

При **модульному принципі** побудови всі МК одного сімейства містять процесорне ядро, однакове для всіх МК даного сімейства, і змінюваний функціональний блок, що відрізняє МК різних моделей. Структура модульного МК наведена на рис. 1.

Процесорне ядро містить у собі:

- центральний процесор;
- внутрішню контролерну магістраль (ВКМ) у складі шин адреси, даних і управління;
- схему синхронізації МК;
- схему управління режимами роботи МК, включаючи підтримку режимів зниженого енергоспоживання, початкового запуску (ініціалізації) і т.д.

Змінюваний функціональний блок містить у собі модулі пам'яті різного типу й об'єму, порти вводу/виводу, модулі тактових генераторів (Г), таймери. У відносно простих МК модуль обробки переривань входить до складу процесорного ядра. У більш складних МК він виготовляється як окремий модуль з розвинутими можливостями. До складу змінюваного функціонального блоку можуть входити і такі додаткові модулі як компаратори напруги, аналого-цифрові перетворювачі (АЦП) і інші. Кожен модуль проектується для роботи в складі МК з урахуванням протоколу ВКМ. Даний підхід дозволяє створювати різноманітні за структурою МК у межах одного сімейства.

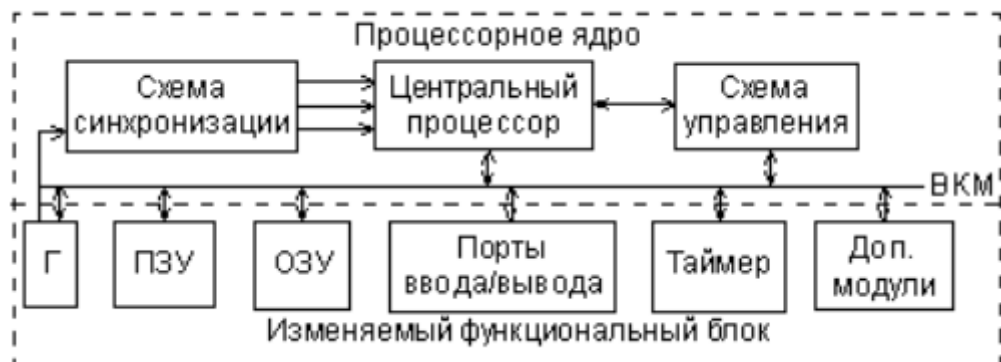


Рис. 1. Модульна організація МК.

Схема синхронізації МК забезпечує формування сигналів синхронізації, необхідних для виконання командних циклів центрального процесора, а також обміну інформацією з внутрішньою магістраллю. У залежності від виконання центрального процесора командний цикл може містити в собі від одного до декількох (4 - 6) тактів синхронізації. Схема синхронізації формує також мітки часу, необхідні для роботи таймерів МК. До складу схеми синхронізації входять дільники частоти, які формують необхідні послідовності синхросигналів.

3. Структура процесорного ядра МК.

Основними характеристиками, які визначають продуктивність процесорного ядра МК, є:

- набір регістрів для збереження проміжних даних;
- система команд процесора;
- способи адресації операндів у просторі пам'яті;
- організація процесів вибірки і виконання команди.

З погляду системи команд і способів адресації операндів процесорне ядро сучасних 8-розрядних МК реалізує один із двох принципів побудови процесорів:

- процесори з CISC-архітектурою, які реалізують так звану повну систему команд (Complicated Instruction Set Computer);
- процесори з RISC-архітектурою, що реалізують скорочену систему команд (Reduced Instruction Set Computer).

CISC-процесори виконують великий набір команд із розвитими можливостями адресації, даючи виробнику можливість вибрати найбільше придатну команду для виконання необхідної операції. У застосуванні до 8-розрядних МК процесор з CISC-архітектурою може мати однобайтний, двобайтний і трьобайтний (рідко чотирьобайтний) формат команд. При цьому система команд, як правило, неортогональна, тобто не всі команди можуть використовувати кожен зі способів адресації стосовно до кожного з регістрів процесора. Вибірка команди на виконання здійснюється побайтно протягом декількох циклів роботи МК. Час виконання команди може складати від 1 до 12 циклів. До МК із CISC-архітектурою відносяться МК фірми Intel з ядром MCS-51, які підтримуються в даний час цілим рядом виробників, МК сімейств HC05, HC08 і HC11 фірми Motorola і ряд інших.

У процесорах з RISC-архітектурою набір команд, які виконуються, скорочений до мінімуму. Для реалізації більш складних операцій приходить комбінувати команди. При цьому всі команди мають формат фіксованої довжини (наприклад, 12, 14 або 16 біт), вибірка команди з пам'яті і її виконання здійснюється за один цикл (такт) синхронізації. Система команд RISC-процесора припускає можливість рівноправного використання всіх регістрів процесора. Це забезпечує додаткову гнучкість при виконанні ряду операцій. До МК із RISC-процесором відносяться МК AVR фірми Atmel, МК PIC16 і PIC17 фірми Microchip і інші.

CISC	RISC
Акцент на апаратному забезпеченні	Акцент на програмне забезпечення
Кілька розмірів і форматів інструкцій	Інструкції того самого набору з кількома форматами
Менше реєстрів	Використовує більше регістрів
Більше режимів адресації	Менше режимів адресації
Широке використання мікропрограмування	Складність компілятора
Інструкції займають різний час циклу	Інструкції займають один цикл
Конвеєрна робота складна	Конвеєрна робота проста

На перший погляд, МК із RISC-процесором повинні мати більш високу продуктивність у порівнянні з CISC МК при одній і тій же тактовій частоті внутрішньої магістралі. Однак на практиці питання про продуктивність більш складне і неоднозначне.

По-перше, оцінка продуктивності МК за часом виконання команд різних систем (RISC і CISC) не зовсім коректна. Звичайно продуктивність МП і МК прийнято оцінювати числом операцій пересилання "регістр-регістр", що можуть бути виконані протягом однієї секунди. У МК із CISC-процесором час виконання операції "регістр-регістр" складає від 1 до 3 циклів, що, здавалося б, поступається продуктивності МК із RISC-процесором. Однак прагнення до скорочення формату команд при збереженні ортогональності системи команд RISC-процесора приводить до змушеного обмеження кількості доступних в одній команді регістрів. Так, наприклад, системою команд МК PIC16 передбачена можливість пересилання результату операції тільки в один із двох регістрів - регістр-джерело операнда f або робочий регістр W . Таким чином, операція пересилання вмісту одного з доступних регістрів в інший (не джерело операнда і не робочий) зажадає використання двох команд. Така необхідність часто виникає при пересиланні вмісту одного з регістрів загального призначення (РОН) в один з портів МК. У той же час, у системі команд більшості CISC-процесорів присутні команди пересилання вмісту РОН в один з портів вводу/виводу. Тобто більш складна система команд іноді дозволяє реалізувати більш ефективний спосіб виконання операції.

По-друге, оцінка продуктивності МК за швидкістю пересилання "регістр-регістр" не враховує особливостей конкретного реалізованого алгоритму управління. Так, при розробці швидкодіючих пристроїв автоматизованого керування основну увагу варто приділяти часові виконання операцій множення і ділення при реалізації рівнянь різних передаточних функцій. А при реалізації пульта дистанційного керування побутовою технікою варто оцінювати час виконання логічних функцій, що використовуються при опитуванні клавіатури і генерації послідовної кодової посилки управління. Тому в критичних ситуаціях, що вимагають високої швидкодії, варто оцінювати продуктивність на безлічі тих операцій, що переважно використовуються в алгоритмі управління і мають обмеження за часом виконання.

По-третє, необхідно ще враховувати, що зазначені в довідникових даних на МК частоти синхронізації зазвичай відповідають частоті кварцового резонатора, який підключається, у той час як тривалість циклу центрального процесора визначається частотою обміну по ВКМ. Співвідношення цих частот індивідуальне для кожного МК і повинне бути прийняте в розрахунок при порівнянні продуктивності різних моделей контролерів.

З погляду організації процесів вибірки і виконання команд в сучасних 8-розрядних МК застосовується одна з двох уже згадуваних архітектур МКС: фон-нейманівська (принстонська) або гарвардська.

Основною особливістю фон-нейманівської архітектури є використання загальної пам'яті для збереження програм і даних.

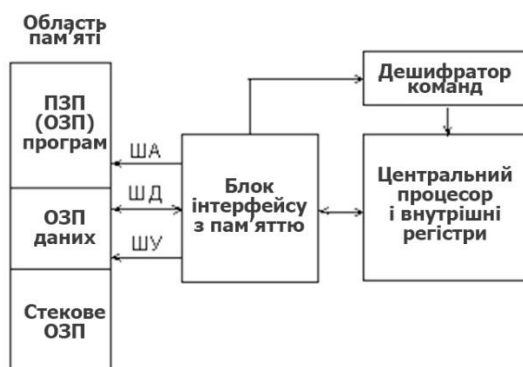


Рис. 2. Структура МКС із фон-нейманівською архітектурою.

Основна перевага архітектури Фон-Неймана – спрощення пристрою МКС, тому що реалізується звертання тільки до однієї загальної пам'яті. Крім того, використання єдиної області пам'яті дозволяло оперативно перерозподіляти ресурси між областями програм і даних, що істотно підвищувало гнучкість МШС із погляду виробника програмного забезпечення. Розміщення стека в загальній пам'яті полегшувало доступ до його вмісту. Невипадково тому фон-нейманівська архітектура стала основною архітектурою універсальних комп'ютерів, включаючи персональні комп'ютери.

Основною особливістю гарвардської архітектури є використання розділних адресних просторів для збереження команд і даних.

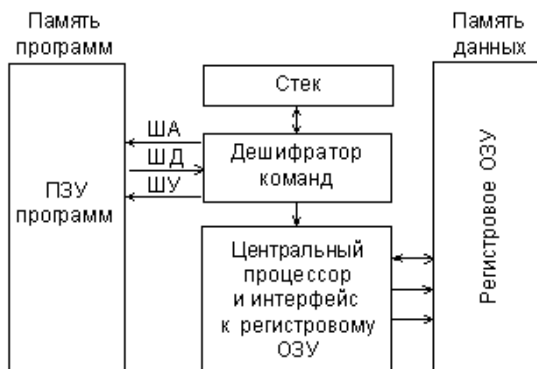


Рис. 3. Структура МКС із гарвардською архітектурою.

Гарвардська архітектура майже не використовувалася до кінця 70-х років, поки виробники МК не зрозуміли, що вона дає певні переваги виробникам автономних систем управління.

Справа в тому, що, судячи з досвіду використання МШС для управління різними об'єктами, для реалізації більшості алгоритмів управління такі переваги фон-нейманівської архітектури як гнучкість і універсальність не мають великого значення. Аналіз реальних програм управління показав, що необхідний об'єм пам'яті даних МК, використовуваний для збереження проміжних результатів, як правило, на порядок менший необхідного об'єму

пам'яті програм. У цих умовах використання єдиного адресного простору викликало збільшення формату команд за рахунок збільшення кількості розрядів для адресації операндів. Застосування окремої невеликої за об'ємом пам'яті даних сприяло скороченню довжини команд і прискоренню пошуку інформації в пам'яті даних.

Крім того, гарвардська архітектура забезпечує потенційно більш високу швидкість виконання програми в порівнянні з фон-нейманівською за рахунок можливості реалізації паралельних операцій. Вибірка наступної команди може відбуватися одночасно з виконанням попередньої, і немає необхідності зупиняти процесор на час вибірки команди. Цей метод реалізації операцій дозволяє забезпечувати виконання різних команд за однакове число тактів, що дає можливість більш просто визначити час виконання циклів і критичних ділянок програми.

Більшість виробників сучасних 8-розрядних МК використовують гарвардську архітектуру. Однак гарвардська архітектура є недостатньо гнучкою для реалізації деяких програмних процедур. Тому порівняння МК, виконаних за різними архітектурами, варто проводити стосовно до конкретної задачі.

4. Управління живленням та тактуванням

Управління живленням та тактуванням є важливими аспектами в проектуванні обчислювальних систем. Вони забезпечують ефективну роботу компонентів системи, оптимізуючи витрати енергії та забезпечуючи стабільну роботу процесора.

1. **Управління живленням:** це процес оптимізації енергоспоживання компонентів системи. У сучасних обчислювальних системах використовуються різні методи зниження споживаної потужності, зокрема через динамічне регулювання частоти та напруги (DVFS — Dynamic Voltage and Frequency Scaling), вимикання неактивних блоків і використання енергоефективних компонентів.
2. **Управління тактуванням:** це метод контролю частоти такту процесора, що визначає швидкість виконання інструкцій. Зміна частоти такту може впливати на продуктивність і енергоспоживання. Висока частота такту збільшує швидкість виконання, але і збільшує споживання енергії. Управління тактуванням дозволяє балансувати між цими двома чинниками, що особливо важливо для мобільних пристроїв.
3. **Інтелектуальні методи управління:** сучасні мікропроцесори оснащуються алгоритмами, що дозволяють самостійно налаштовувати параметри живлення та тактування в залежності від навантаження. Це дає змогу не тільки зменшити енергоспоживання, але й продовжити термін служби батарей у мобільних пристроях.

Управління живленням і тактуванням відіграє критичну роль у забезпеченні ефективності роботи обчислювальних систем, особливо в умовах

обмежених ресурсів енергії, як у мобільних пристроях та вбудованих системах.

5. Система команд процесора МК.

Так само, як і в будь-якій мікропроцесорній системі, набір команд процесора мікроконтролера включає чотири основні групи:

- команди пересилання даних;
- арифметичні команди;
- логічні команди;
- команди переходів.

Для реалізації можливості незалежного управління розрядами портів (регістрів) у більшості сучасних МК передбачена також група команд бітового управління (булевий або бітовий процесор). Наявність команд бітового процесора дозволяє істотно скоротити об'єм коду управляючих програм і час їхнього виконання.

У ряді МК виділяють також групу команд управління ресурсами контролера, які використовуються для налаштування режимів роботи портів вводу/виводу, управління таймером і т.п. У більшості сучасних МК внутрішніх ресурсів контролера відображаються в пам'яті даних, тому для мети управління ресурсами використовуються команди пересилання даних.

Система команд МК у порівнянні із системою команд універсального МП має, як правило, менш розвинуті групи арифметичних і логічних команд, зате більш потужні групи команд пересилання даних і управління. Ця особливість пов'язана зі сферою застосування МК, яка вимагає, насамперед, контролю навколишнього оточення і формування управляючих впливів.

6. Приклад мікроконтролера: розбір архітектури ATmega328P.

Приклад мікроконтролера: ATmega

Мікроконтролер ATmega (наприклад, ATmega328P) має типову архітектуру RISC (Reduced Instruction Set Computing). Він складається з кількох основних блоків:

1. **Центральний процесор (CPU)** — обробляє інструкції.

Арифметико-логічний блок (ALU): відповідає за виконання арифметичних і логічних операцій.

Регістрами управління: включає програмні лічильники та регістри, що зберігають тимчасові значення під час виконання програм.

Порти введення/виведення: дозволяють мікроконтролеру взаємодіяти із зовнішніми пристроями через цифрові або аналогові сигнали.

2. **Пам'ять:**

- **Flash-пам'ять:** для зберігання коду програми. Ця пам'ять є перезаписуваною, що дозволяє оновлювати програмне забезпечення.

- **SRAM (статична оперативна пам'ять):** зберігає змінні під час виконання програми. Це оперативна пам'ять, доступна для змінних і тимчасових даних.
- **EEPROM (електричноперезаписувана пам'ять):** забезпечує збереження даних навіть після вимкнення живлення.

3. **Вхідно-вихідні порти** — для взаємодії з зовнішніми пристроями.

Цифрові порти: дозволяють зчитувати сигнали з кнопок або переключати статус світлодіодів.

Аналогові порти: використовуються для зчитування аналогових сигналів за допомогою АЦП.

4. **Таймери/лічильники** — для вимірювання часу та генерації подій.

Використовуються для генерування переривань, визначення часу та синхронізації подій. Наприклад, для вимірювання тривалості подій або генерування переривань із заданим інтервалом.

5. **Аналого-цифровий перетворювач (ADC).** Перетворює аналогові сигнали на цифрові значення для обробки процесором. Це важливо для застосувань, де необхідно працювати з датчиками, що надають аналогові сигнали (наприклад, температурні датчики).

6. **Переривання:**

ATMega має систему переривань, що дозволяє обробляти події (наприклад, натискання кнопки) без постійного моніторингу процесором. Це дозволяє зекономити енергоспоживання і підвищити ефективність системи.

Режими енергозбереження:

7. Мікроконтролер підтримує різні режими економії енергії, зокрема сплячі режими, коли неактивні блоки вимикаються для зниження споживаної потужності.

Архітектура ATMega дозволяє ефективно управляти різними функціями при обмежених ресурсах енергії.

Тема 7. Пам'ять програм і даних МК.

1. Пам'ять програм (ПЗП).

Основною властивістю пам'яті програм є її енергонезалежність, тобто можливість зберігання програми за відсутності живлення. З погляду користувачів МК слід розрізняти наступні типи енергозалежної пам'яті програм:

ПЗП масочного типу - mask-ROM. Вміст комірок ПЗП цього типу заноситься при її виготовленні за допомогою масок і не може бути згодом замінене або перепрограмоване. Тому МК з таким типом пам'яті програм слід використовувати тільки після достатньо тривалої експлуатації. Основним недоліком даної пам'яті є необхідність значних витрат на створення нового комплексу фотошаблонів і їх впровадження у виробництво. Звичайно такий процес займає 2-3 місяці і є економічно вигідним тільки при випуску десятків тисяч приладів. ПЗП масочного типу забезпечують високу надійність зберігання інформації внаслідок програмування в заводських умовах з подальшим контролем результату.

ПЗП, програмовані користувачем, з ультрафіолетовим стиранням - EPROM (Erasable Programmable ROM). ПЗП даного типу програмуються електричними сигналами і стираються за допомогою ультрафіолетового опромінювання. Елемент пам'яті EPROM є МОП-транзистором з «плаваючим» затвором, заряд на який переноситься з управляючого затвора при подачі відповідних електричних сигналів. Для стирання вмісту комірки вона опромінюється ультрафіолетовим світлом, яке надає заряду на плаваючому затворі енергію, достатню для подолання потенційного бар'єру і стікання на підкладку. Цей процес може займати від декількох секунд до декількох хвилин. МК з EPROM допускають багатократне програмування і випускаються в керамічному корпусі з кварцовим віконцем для доступу ультрафіолетового світла. Такий корпус вартує досить дорого, що значно збільшує вартість МК. Для зменшення вартості МК з EPROM його вкладають в корпус без віконця (версія EPROM з однократним програмуванням).

ПЗП, одноразово програмовані користувачем, - OTPROM (One-Time Programmable ROM). Є версією EPROM, виконаною в корпусі без віконця для зменшення вартості МК на його основі. Скорочення вартості при використуванні таких корпусів настільки значне, що останнім часом ці версії EPROM часто використовують замість масочних ПЗП.

ПЗП, програмовані користувачем, з електричним стиранням - EEPROM (Electrically Erasable Programmable ROM). ПЗП даного типу можна вважати новим поколінням EPROM, в яких стирання елементів пам'яті проводиться також електричними сигналами за рахунок використання тунельних механізмів. Вживання EEPROM дозволяє стирати і програмувати МК, не знімаючи його з плати. У такий спосіб можна проводити відладку і модернізацію програмного забезпечення. Це дає величезний вигаш на початкових стадіях розробки МК-систем або в процесі їх вивчення, коли

багато часу йде на пошук причин непрацездатності системи і виконання циклів стирання-програмування пам'яті програм. За ціною EEPROM займають середнє положення між OTPROM і EPROM. Технологія програмування пам'яті EEPROM допускає побайтове стирання і програмування комірок. Не дивлячись на очевидні переваги EEPROM, тільки в небагатьох моделях МК така пам'ять використовується для зберігання програм. Зв'язано це з тим, що, по-перше, EEPROM мають обмежений об'єм пам'яті. По-друге, майже одночасно з EEPROM з'явилися FLASH-ПЗП, які при схожих споживчих характеристиках мають більш низьку вартість;

ПЗП з електричним стиранням типу Flash - Flash-ROM. Функціонально Flash-пам'ять мало відрізняється від EEPROM. Основна відмінність полягає в способі стирання записаної інформації. В пам'яті EEPROM стирання проводиться окремо для кожної комірки, а в Flash-пам'ять стерти можна тільки цілими блоками. Якщо необхідно змінити вміст однієї комірки Flash-пам'яті, буде потрібно перепрограмувати весь блок. Спрощення декодуючих схем в порівнянні з EEPROM привело до того, що МК з Flash-пам'яттю стають конкурентоздатними по відношенню не тільки до МК з однократно програмованими ПЗП, але і з масочними ПЗП також.

2. Пам'ять даних.

Пам'ять даних МК виконується, як правило, на основі статичного ОЗП. Термін «статичне» означає, що вміст комірок ОЗП зберігається при зниженні тактової частоти МК до скільки завгодно малих значень (з метою зниження енергоспоживання). Більшість МК має такий параметр, як «напруга зберігання інформації» - U_{standby} . При зниженні напруги живлення нижче мінімального допустимого рівня U_{DDMIN} , але вище за рівень U_{standby} робота програми МК виконуватися не буде, але інформація в ОЗП зберігається. При відновленні напруги живлення можна буде скинути МК і продовжити виконання програми без втрати даних. Рівень напруги зберігання складає звичайно близько 1 В, що дозволяє у разі потреби перевести МК на живлення від автономного джерела (батарей) і зберегти в цьому режимі дані ОЗП.

Об'єм пам'яті даних МК, як правило, невеликий і складає звичайно десятки і сотні байт. Цю обставину необхідно враховувати при розробці програм для МК. Так, при програмуванні МК константи, якщо можливо, не зберігаються як змінні, а заносяться в ПЗП програм. Максимально використовуються апаратні можливості МК, зокрема, таймери. Прикладні програми повинні орієнтуватися на роботу без використання великих масивів даних.

3. Регістри МК.

Як і всі МПС, МК мають набір регістрів, які використовуються для управління його ресурсами. До числа цих регістрів входять звичайно регістри процесора (акумулятор, регістри стану, індексні регістри), регістри управління (регістри управління перериваннями, таймером), регістри, що забезпечують

ввід/вивід даних (реєстри даних портів, реєстри управління паралельним, послідовним або аналоговим вводом/виводом). Звернення до цих реєстрів може проводитися по-різному.

В МК з RISC-процесором всі реєстри (часто і акумулятор) розташовуються за адресами, що явно задаються. Це забезпечує більш високу гнучкість при роботі процесора.

Одним з важливих питань є розміщення реєстрів в адресному просторі МК. В деяких МК всі реєстри і пам'ять даних розташовуються в одному адресному просторі. Це означає, що пам'ять даних суміщена з реєстрами. Такий підхід називається «відображенням ресурсів МК на пам'ять».

В інших МК адресний простір пристроїв вводу/виводу відокремлений від загального простору пам'яті. Окремий простір вводу/виводу дає деяку перевагу процесорам з гарвардською архітектурою, забезпечуючи можливість прочитувати команду під час звернення до реєстра вводу/виводу.

4. Стек МК.

В мікроконтролерах ОЗП даних використовується також для організації виклику підпрограм і обробки переривань. При цих операціях вміст програмного лічильника і основних реєстрів (акумулятор, реєстр стану і інші) зберігається і потім відновлюється при поверненні до основної програми.

У фон-нейманівській архітектурі єдина область пам'яті використовується, у тому числі, і для реалізації стека. При цьому знижується продуктивність пристрою, оскільки одночасний доступ до різних видів пам'яті неможливий. Зокрема, при виконанні команди виклику підпрограми наступна команда вибирається після того, як в стек буде поміщений вміст програмного лічильника.

В гарвардській архітектурі стекові операції проводяться в спеціально виділеній для цієї мети пам'яті. Це означає, що при виконанні програми виклику підпрограм процесор з гарвардською архітектурою проводить декілька дій одночасно.

Необхідно пам'ятати, що МК обох архітектур мають обмежену ємність пам'яті для зберігання даних. Якщо в процесорі є окремий стек і об'єм записаних в нього даних перевищує його місткість, то відбувається циклічна зміна вмісту покажчика стека, і він починає посилатися на раніше заповнену комірку стека. Це означає, що після дуже великої кількості викликів підпрограм в стеку опиниться неправильна адреса повернення. Якщо МК використовує загальну область пам'яті для розміщення даних і стека, то існує небезпека, що при переповнюванні стека відбудеться запис в область даних або буде зроблена спроба запису завантажуваних в стек даних в область ПЗП.

5. Зовнішня пам'ять.

Не дивлячись на існуючу тенденцію по переходу до закритої архітектури МК, в деяких випадках виникає необхідність підключення додаткової зовнішньої пам'яті (як пам'яті програм, так і даних).

Якщо МК містить спеціальні апаратні засоби для підключення зовнішньої пам'яті, то ця операція проводиться штатним способом (як для МП).

Другий, більш універсальний, спосіб полягає в тому, щоб використовувати порти вводу/виводу для підключення зовнішньої пам'яті і реалізувати звернення до пам'яті програмними засобами. Такий спосіб дозволяє задіювати прості пристрої вводу/виводу без реалізації складних шинних інтерфейсів, проте приводить до зниження швидкодії системи при зверненні до зовнішньої пам'яті.

Тема 8. Таймери і процесори подій.

1. Поняття таймера в МК

Більшість задач управління, які реалізуються за допомогою МК, вимагає виконання їх в реальному часі. Під цим розуміється здатність системи одержати інформацію про стан керованого об'єкту, виконати необхідні розрахункові процедури і видати управляючі дії протягом інтервалу часу, достатнього для бажаної зміни стану об'єкту.

Покладати функції формування управління в реальному часі тільки на центральний процесор неефективно, оскільки це займає ресурси, необхідні для розрахункових процедур. Тому в більшості сучасних МК використовується апаратна підтримка роботи в реальному часі з використанням таймера (таймерів).

Модулі таймерів служать для прийому інформації про час настання тих або інших подій від зовнішніх датчиків подій, а також для формування управляючих дій в часі.

Модуль таймера 8-розрядного МК є 8-и або 16-розрядний лічильник з схемою управління. Схемотехнікою МК звичайно передбачається можливість використання таймера в режимі лічильника зовнішніх подій, тому його часто називають таймером/лічильником. Структура типового 16-розрядного таймера/лічильника у складі МК приведена на рис. 1. В пам'яті МК 16-розрядний лічильник відображається двома регістрами: ТН - старший байт лічильника, ТЛ - молодший байт. Регістри доступні для читання і для запису. Напрямок рахунку - тільки прямий, тобто під час надходження вхідних імпульсів вміст лічильника збільшується.

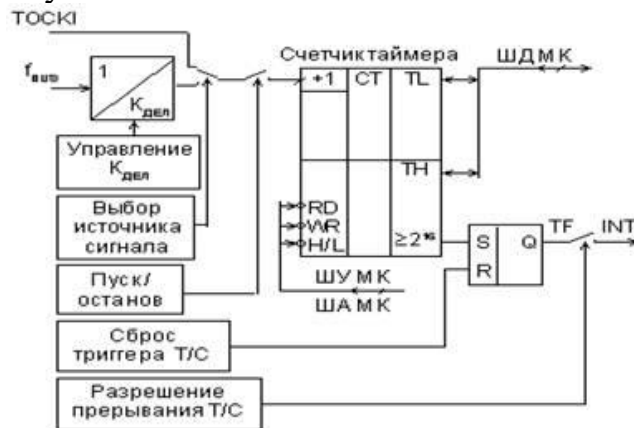


Рис. 1. Структура модуля таймера/лічильника.

Залежно від налаштувань лічильник може використовувати одне з джерел вхідних сигналів:

- імпульсну послідовність з виходу керованого дільника частоти F_{bus} ;
- сигнали зовнішніх подій, що поступають на вхід ТОСКІ контролера.

В першому випадку говорять, що лічильник працює в режимі таймера, в другому – в режимі лічильника подій. При переповнюванні лічильника встановлюється в «одиницю» тригер переповнювання ТФ, який генерує запит на переривання, якщо переривання від таймера дозволені. Пуск і зупинка таймера можуть здійснюватися тільки під управлінням програми. Програмним способом можна також встановити старший і молодший біти лічильника в довільний стан або зчитати поточний код лічильника.

Розглянутий «класичний» модуль таймера/лічильника широко застосовується в різних моделях щодо простих МК. Він може використовуватися для вимірювання тимчасових інтервалів і формування послідовності імпульсів. Основними недоліками «класичного» таймера/лічильника є:

- втрати часу на виконання команд пуску і зупинки таймера, що приводять до появи помилки при вимірюванні тимчасових інтервалів і обмежуючі мінімальну тривалість вимірюваних інтервалів часу одиницями мс;
- складнощі при формуванні тимчасових інтервалів (міток часу), відмінних від періоду повного коефіцієнта рахунку, рівного $(K_{діл}/F_{bus}) \cdot 2^{16}$;
- неможливість одночасного обслуговування (вимірювання або формування імпульсного сигналу) відразу декількох каналів.

Перші два недоліки були усунені у вдосконаленому модулі таймера/лічильника, що використовується в МК сімейства MCS-51 (Intel). Додаткова логіка рахункового входу дозволяє тактовим імпульсам поступати на вхід лічильника, якщо рівень сигналу на одній з ліній вводу рівний «1». Таке рішення підвищує точність вимірювання тимчасових інтервалів, оскільки пуск і зупинка таймера проводиться апаратно. Також у вдосконаленому таймері реалізований режим перезавантаження лічильника довільним кодом у момент переповнювання. Це дозволяє формувати тимчасові послідовності з періодом, відмінним від періоду повного коефіцієнта рахунку.

Проте ці удосконалення не усувають головного недоліку модуля «класичного» таймера – одноканального режиму роботи. Вдосконалення підсистеми реального часу МК ведеться по наступних напрямках:

- збільшення числа модулів таймерів/лічильників. Цей шлях характерний для фірм, випускаючих МК із структурою MCS-51, а також для МК компаній Mitsubishi і Hitachi;
- модифікація структури модуля таймера/лічильника, при якій збільшення числа каналів досягається не за рахунок збільшення числа лічильників, а за рахунок введення додаткових апаратних засобів вхідного захоплення (input capture - IC) і вихідного порівняння (output compare - OC). Такий підхід використовується, зокрема, в МК компанії Motorola.

Принцип дії каналу вхідного захоплення таймера/лічильника ілюструє рис 1.

Схема детектора події «спостерігає» за рівнем напруги на одному з входів МК. Частіше всього це одна з ліній порту вводу/виводу. При зміні рівня логічного сигналу з «0» на «1» і навпаки виробляється строб запису, і поточний стан лічильника/таймера записується в 16-розрядний регістр входного захоплення. Описану дію в мікропроцесорній техніці називають подією захоплення.

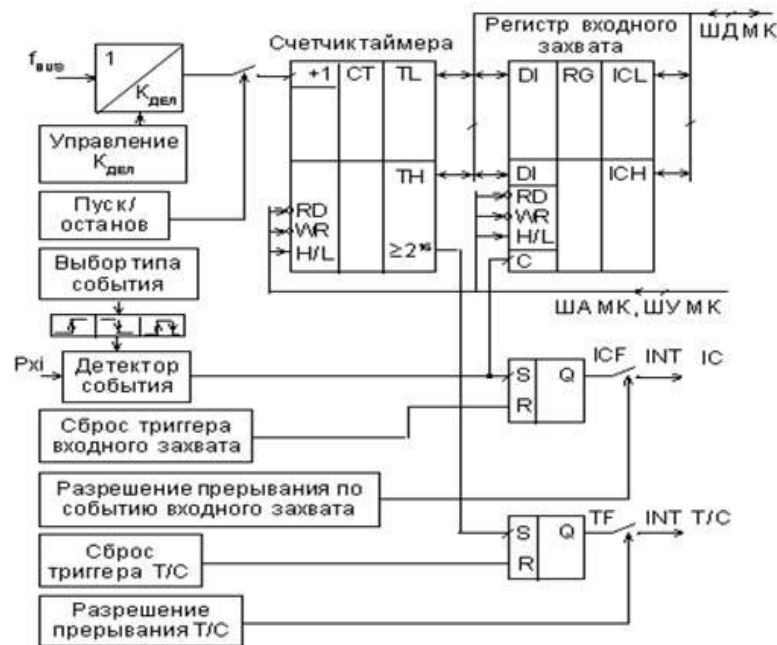


Рис. 2 Структурна схема каналу входного захоплення таймера.

Передбачена можливість вибору типу сигналу на вході, і це сприймається як подія:

- позитивний (передній) фронт сигналу;
- негативний (задній) фронт сигналу;
- будь-яка зміна логічного рівня сигналу.

Вибір типу події захоплення встановлюється в процесі ініціалізації таймера і може неодноразово змінюватися в ході виконання програми. Кожна подія захоплення приводить до установки в «1» триггера входного захоплення і появи на його виході прапора (ознаки) входного захоплення ICF. Стан триггера входного захоплення може бути лічений програмно, а якщо переривання по події захоплення дозволені - формується запит на переривання INT IC.

Використовування режиму входного захоплення дозволяє виключити помилки вимірювання входного інтервалу часу, зв'язані з часом переходу до підпрограми обробки переривання, оскільки копіювання поточного стану лічильника здійснюється апаратними, а не програмними засобами. Проте час переходу на підпрограму обробки переривання накладає обмеження на тривалість вимірюваного інтервалу часу, оскільки передбачається, що друга подія захоплення відбудеться пізніше, ніж код першої події буде прочитаний МК.

Структура апаратних засобів каналу вихідного порівняння представлена на рис. 3.

Цифровий компаратор безперервно порівнює поточний код лічильника таймера з кодом, який записаний в 16-розрядному регістрі вихідного порівняння. У момент рівності кодів на одному з виходів МК (Pxj на мал. 3) встановлюється заданий рівень логічного сигналу. Звичайно передбачено три типи зміни сигналу на виході Pxj у момент події вихідного порівняння:

- установка високого логічного рівня;
- установка низького логічного рівня;
- інвертування сигналу на виході.

При настанні події порівняння встановлюються в «1» тригер вихідного порівняння і відповідну йому ознаку вихідного порівняння OCF. Аналогічно режиму вхідного захоплення стан тригера вихідного порівняння може бути прочитано програмно, а якщо переривання по події порівняння дозволені - формується запит на переривання INT OC.

Режим вихідного порівняння призначений, перш за все, для формування тимчасових інтервалів заданої тривалості. Тривалість сформованого тимчасового інтервалу визначається тільки різницею кодів, послідовно завантажуваних в регістр вихідного порівняння, і не залежить від програмного забезпечення МК. Час, необхідний для запису нового значення коду в регістр каналу порівняння, обмежує мінімальну тривалість формованого тимчасового інтервалу.

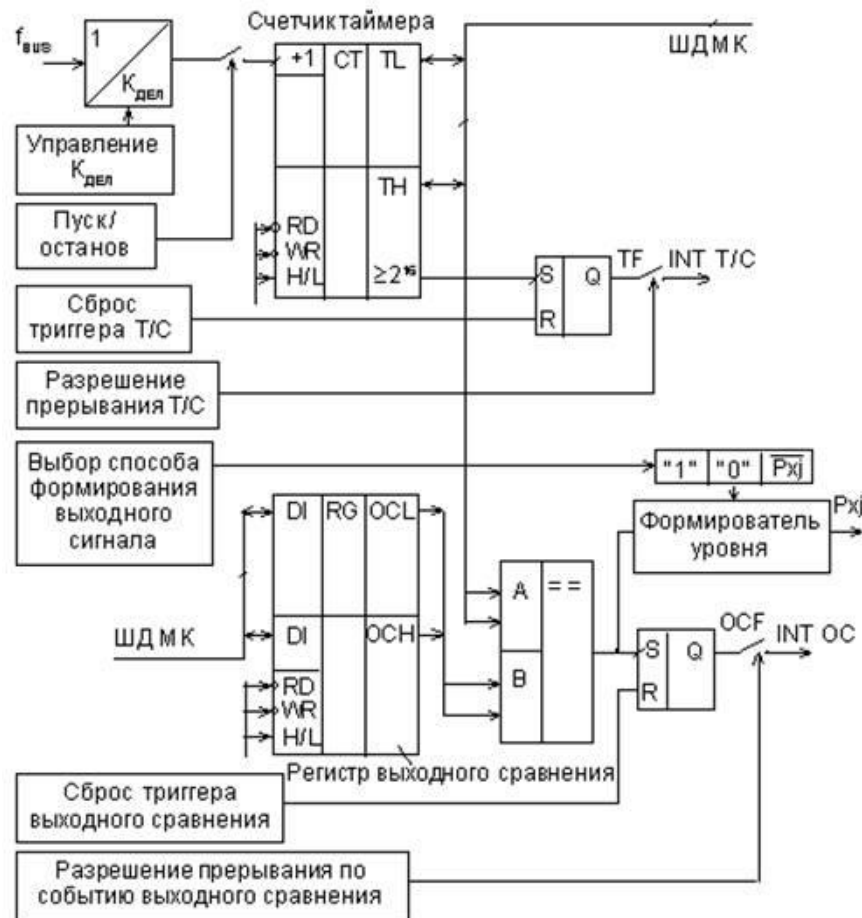


Рис. 3 Структурна схема каналу вихідного порівняння таймера.

Модулі вдосконаленого таймера використовуються у складі МК в різних модифікаціях. При цьому число каналів вхідного захоплення і вихідного порівняння в модулі може бути різним. Так, в МК сімейства HC05 фірми Motorola типовими рішеннями є модулі 1IC+1OC або 2IC+2OC, а модуль таймера у складі МК тільки один. У ряді модулів канали можуть бути довільно налаштовані на функцію вхідного захоплення або вихідного порівняння за допомогою ініціалізації. Лічильник модуля вдосконаленого таймера може не мати функції програмної зупинки. В цьому випадку стан лічильника не можна синхронізувати з яким-небудь моментом роботи МК, і такий лічильник характеризується як вільно підраховуючий (free counter).

Апаратні засоби вдосконаленого таймера дозволяють вирішити багато задач управління в реальному часі. Проте у міру зростання складності алгоритмів управління виразно виявляються обмеження модулів вдосконаленого таймера, а саме:

- недостатнє число каналів захоплення і порівняння, що належить одному лічильнику тимчасової бази. Це не дозволяє сформувати синхронізовані між собою багатоканальні імпульсні послідовності;
- однозначно певна конфігурація каналу (або захоплення або порівняння) часто не задовольняє потребам вирішуваної задачі;
- формування сигналів по методу широтно-імпульсної модуляції (ШІМ) вимагає програмної підтримки, що знижує максимально досяжну частоту вихідного сигналу.

Тому наступним етапом розвитку модулів підсистеми реального часу МК стали модулі процесорів подій. Вперше модулі процесорів подій були використані компанією Intel в МК сімейства 8xC51Fx. Цей модуль одержав назву програмованого рахункового масиву (Programmable Counter Array – PCA).

2. Структура процесора подій.

PCA забезпечує більш широкі можливості роботи в реальному масштабі часу і у меншій мірі витрачає ресурси центрального процесора, ніж стандартний і вдосконалений таймери/лічильники. До переваг PCA також можна віднести більш просте програмування і більш високу точність. Наприклад, PCA може забезпечити кращу часову роздільну здатність, ніж таймери 0, 1 і 2 МК сімейства MCS-51, оскільки лічильник PCA здатний працювати з тактовою частотою, втричі більшою, ніж у цих таймерів. PCA також може вирішувати багато задач, виконання яких з використанням таймерів вимагає додаткових апаратних витрат (наприклад, визначення фазового зсуву між імпульсами або генерація ШІМ-сигналу). PCA складається з 16-бітового таймера-лічильника і п'яти 16-бітових модулів порівняння, як показано на рис. 4.

Таймер-лічильник PCA використовується як базовий таймер для функціонування всіх п'яти модулів порівняння. Вхід таймера-лічильника PCA може бути запрограмований на підрахунок сигналів від наступних джерел:

- вихід діляника на 12 тактового генератора МК;
- вихід діляника на 4 тактового генератора МК;
- сигнал переповнювання таймера 0;
- зовнішній вхідний сигнал на виведення ECI (P1.2).

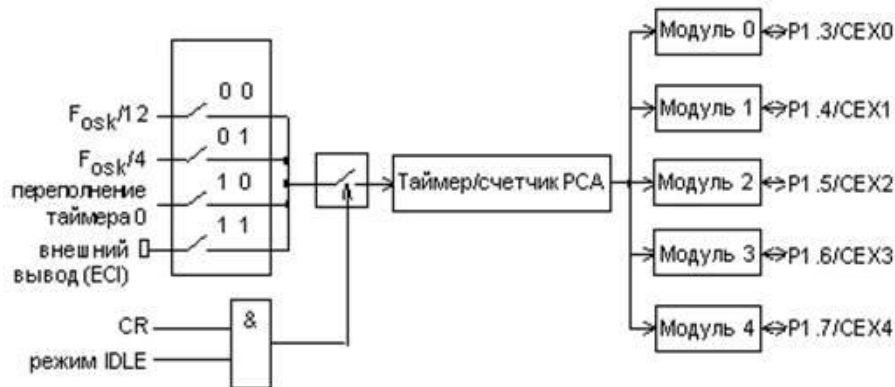


Рис. 4. Структура процесора подій МК сімейства Intel 8xC51Fх.

Будь-який з модулів порівняння може бути запрограмований для роботи в наступних режимах:

- спрацювання по фронту і/або спаду імпульсу на вході CEXi;
- програмованого таймера;
- високошвидкісного виходу;
- широтно-імпульсного модулятора.

Модуль 4 може бути також запрограмований як сторожовий таймер (Watchdog Timer - WDT).

Режим спрацювання по імпульсу на вході МК еквівалентний режиму вхідного захоплення (IC) вдосконаленого таймера. Режими програмованого таймера і високошвидкісного виходу близькі по своїх функціональних можливостях до режиму вихідного порівняння (OC).

В режимі ШІМ на відповідному виводі МК формується послідовність імпульсів з періодом, рівним періоду базового таймера/лічильника PCA. Значення 8-розрядного коду, записане в молодший байт регістра відповідного модуля задає скважність формованого сигналу. При зміні коду від 0 до 255 шпаруватість міняється від 100 % до 0,4 %.

Режим ШІМ дуже простий з погляду програмного обслуговування. Якщо зміни скважності не передбачається, то достатньо один раз занести відповідний код в регістр даних модуля, проініціювати режим ШІМ, і імпульсна послідовність відтворюватиметься із заданими параметрами без втручання програми.

Призначення і особливості роботи сторожового таймера будуть розглянуто далі окремо.

При роботі модуля порівняння в режимі заціпки, програмованого таймера або високошвидкісного виходу модуль може сформувати сигнал

переривання. Сигнали від всіх п'яти модулів порівняння-защипки і сигнал переповнювання таймера РСА розділяють один вектор переривання. Іншими словами, якщо переривання дозволені, то і сигнал переповнювання таймера РСА і сигнал від будь-якого з модулів викликають одну і ту ж підпрограму переривань, яка повинна сама ідентифікувати джерело, що викликало її.

Для роботи із зовнішніми пристроями таймер-лічильник РСА і модулі порівняння використовують виводи P1 порту МК. Якщо який-небудь вивід порту не використовується при роботі РСА, або РСА не задіяний, порт може застосовуватися стандартним чином.

Реалізований в 8xC51FX РСА опинився настільки вдалим, що архітектура даних МК стала промисловим стандартом де-факто, а сам РСА багато разів відтворювався в різних модифікаціях мікроконтролерів різних фірм.

Тенденція розвитку підсистеми реального часу сучасних МК знаходить своє віддзеркалення в збільшенні числа каналів процесорів подій і розширенні їх функціональних можливостей.

Тема 9. Тактові генератори МК.

1. Поняття генератора тактової частоти.

Генератор тактової частоти (генератор тактових імпульсів) — електронний генератор, призначений для генерації електричних тактових імпульсів заданої частоти (зазвичай прямокутної форми) для синхронізації сигналів та процесів в цифрових пристроях — ЕОМ, електронних годинниках і таймерах, в мікропроцесорній та іншій цифровій техніці. Тактові імпульси часто використовуються як еталонна частота — рахуючи їх кількість, можна, наприклад, вимірювати часові інтервали.

В мікропроцесорній техніці один тактовий імпульс, як правило, відповідає одній базовій операції. Обробка однієї інструкції може проводитися за один або кілька тактів роботи мікропроцесора, в залежності від архітектури і типу інструкції. Частота тактових імпульсів визначає швидкість обчислень.

В складних пристроях тактовий генератор одночасно генерує декілька імпульсів, зміщених один відносно одного по фазі, що дозволяє синхронізувати елементи таких пристроїв та уникати «перегонів» сигналів. Багатофазні тактові генератори, наприклад, використовуються в апаратних системах з конвеєром команд. Зміщення імпульсів по фазі досягається поділом частоти або лініями затримки сигналу.

2. Типи генераторів.

RC-генератор.

У нескладних конструкціях, не критичних до стабільності тактового генератора, часто використовується послідовне включення декількох інверторів через RC-компоненти. Частота коливань залежить від номіналів резистора і конденсатора. Основний мінус даної конструкції — низька стабільність частоти. Плюс — гранична простота.

Кварцовий генератор.

Складовою кварцового генератора є кварцовий резонатор, який забезпечує стабільність частоти. Мікросхема генерації являє собою спеціальну мікросхему, яка при підключенні до її входів кварцового резонатора буде видавати на інших виводах частоту, поділену або помножену на вхідну. Таке рішення використовується в годинниках, а також на старих материнських платах (де частоти шин були заздалегідь відомі, тільки внутрішня частота центрального процесора множилася коефіцієнтом множення).

В простому випадку тактовий генератор може бути побудований без використання спеціальних мікросхем. Для побудови тактового генератора, який виробляє прямокутні імпульси, можна застосувати такі мікросхеми, як K155ЛА3 (містить чотири елементи 2 І-НЕ), яка виконана за технологією ТТЛ або K555ЛА3 (SN74LS00), яка виконана за технологією ТТЛШ і збігається за виводами з мікросхемою K155ЛА3. Крім мікросхеми буде потрібен кварцовий резонатор (частота на яку він розрахований і буде вихідною частотою генератора), а також кілька резисторів. Схеми даних генераторів прості, вони

описані у всіх підручниках по електроніці. А надійність їх незаперечна багаторічним застосуванням у всіх ЕОМ, ПЕОМ та інших конструкціях, таких як частотоміри, цифрові шкали, годинник і ін.

В більш складних випадках використовують спеціальні мікросхеми тактового генератора, які простими засобами дозволяють отримати багатofазні тактові генератори.

Програмована мікросхема генерації.

У сучасних материнських платах необхідно велика кількість різних частот, крім опорної частоти системної шини, які, по можливості, не повинні бути залежні один від одного. Хоча базова частота все ж формується кварцовим резонатором (частота — 14,3 МГц), вона необхідна лише для роботи самої мікросхеми. Вихідні ж частоти коригуються самою мікросхемою. Наприклад, частота системної шини може дорівнювати стандартним 33 МГц, AGP — 66 МГц і не залежить від частоти FSB процесора.

Якщо в електронній схемі необхідно розділити частоту на 2 використовують Т-тригер в режимі лічильника імпульсів. Відповідно, для збільшення дільника збільшують кількість лічильників (тригерів).

3. Тактові генератори МК.

Сучасні МК містять вбудовані тактові генератори, які вимагають мінімального числа зовнішніх часозадаючих елементів. На практиці використовуються три основні способи визначення тактової частоти генератора: за допомогою кварцового резонатора, керамічного резонатора і зовнішнього RC-ланцюга.

Типова схема підключення кварцового або керамічного резонатора приведена на малюнку:

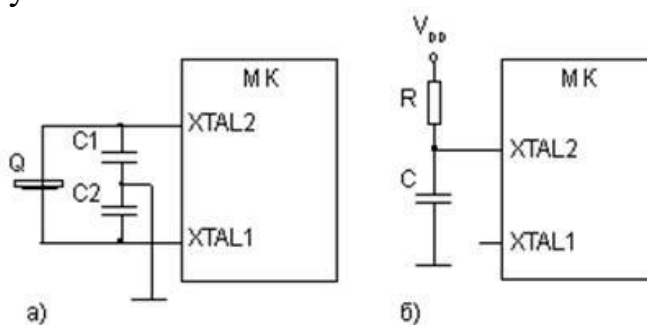


Рис. 1. Тактування МК з використанням кварцового або керамічного резонаторів (а) і з використанням RC-ланцюга (б).

Кварцовий або керамічний резонатор Q підключається до виводів XTAL1 і XTAL2, які звичайно є входом і виходом інвертуючого підсилювача. Номінали конденсаторів C1 і C2 визначаються виробником МК для конкретної частоти резонатора. Іноді вимагається включити резистор порядку декількох мегаом між виводами XTAL1 і XTAL2 для стабільної роботи генератора.

Використовування кварцового резонатора дозволяє забезпечити високу точність і стабільність тактової частоти (розкид частот кварцового резонатора звичайно складає менш 0,01%). Такий рівень точності потрібен для

забезпечення точного ходу годинника реального часу або організації інтерфейсу з іншими пристроями. Основними недоліками кварцового резонатора є його низька механічна міцність (висока крихкість) і відносно висока вартість.

При менш жорстких вимогах до стабільності тактової частоти можливе використання більш стійких до ударного навантаження керамічних резонаторів. Багато керамічних резонаторів мають вбудовані конденсатори, що дозволяє зменшити кількість зовнішніх елементів, що підключаються, з трьох до одного. Керамічні резонатори мають розкид частот порядку декількох десятих часток відсотка (звичайно близько 0,5 %).

Найдешевшим способом задання тактової частоти МК є використання зовнішнього RC-ланцюга, як показано на мал. (б). Зовнішній RC-ланцюг не забезпечує високої точності задання тактової частоти (розкид частот може доходити до десятків відсотків). Це неприйнятно для багатьох програм, де потрібен точний підрахунок часу. Проте є маса практичних задач, де точність завдання тактової частоти не має великого значення.

Залежність тактової частоти МК від номіналів RC-ланцюга залежить від конкретної реалізації внутрішнього генератора і приводиться в керівництві по вживанню контролера.

Практично всі МК допускають роботу від зовнішнього джерела тактового сигналу, який підключається до входу XTAL1 внутрішнього підсилювача. За допомогою зовнішнього тактового генератора можна задати будь-яку тактову частоту МК (в межах робочого діапазону) і забезпечити синхронну роботу декількох пристроїв.

Деякі сучасні МК містять вбудовані RC або кільцеві генератори, які дозволяють контролеру працювати без зовнішніх ланцюгів синхронізації. Робота внутрішнього генератора звичайно дозволяється шляхом програмування відповідного біта регістра конфігурації МК.

В більшості моделей МК частота часозадаючого елемента (резонатора або RC-ланцюга) і частоти тактування fBUS жорстко зв'язані коефіцієнтом поділу вбудованого подільника частоти. Тому зміна частоти програмним шляхом не представляється можливою.

В деяких МК сімейства AVR фірми Atmel тактова частота контролера, що задається внутрішнім RC-ланцюгом, також може змінюватися програмними засобами.

Тема 10. Цифрова логіка і цифрові системи.

1. Основні елементи цифрової логіки.

Сигнал – це будь-яка фізична величина (наприклад, температура, тиск повітря, інтенсивність світла, сила струму і т.д.), що змінюється з часом. Саме завдяки цій зміні сигнал може нести в собі якусь інформацію.

Електричний сигнал – це електрична величина (наприклад, напруга, струм, потужність), що змінюється з часом. Вся електроніка в основному працює з електричними сигналами, хоча зараз все більше використовуються світлові сигнали, які представляють собою інтенсивність світла, що змінюється в часі.

Аналоговий сигнал – це сигнал, який може приймати будь-які значення в певних межах (наприклад, напруга може плавно змінюватися в межах від нуля до десяти вольт). Пристрої, що працюють тільки з аналоговими сигналами, називаються аналоговими пристроями. Назва "аналоговий" має на увазі, що сигнал змінюється аналогічно фізичній величині, тобто безперервно.

Цифровий сигнал – це сигнал, який може приймати тільки два (іноді - три) значення, причому дозволені деякі відхилення від цих значень (рис.1). Наприклад, напруга може приймати два значення: від 0 до 0,5 В (рівень нуля) або від 2,5 до 5 В (рівень одиниці). Пристрої, що працюють виключно з цифровими сигналами, називаються цифровими пристроями.

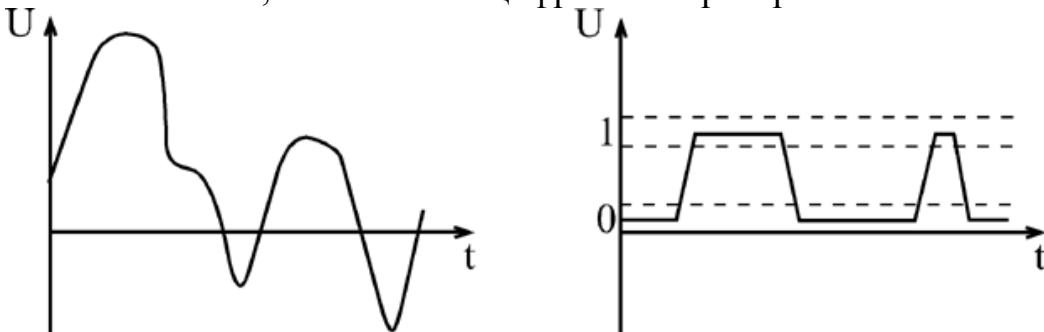


Рис. 1. Електричні сигнали: аналоговий (зліва) і цифровий (праворуч).

Можна сказати, що в природі практично всі сигнали – аналогові, тобто вони змінюються безперервно в якихось межах. Саме тому перші електронні пристрої були аналоговими. Вони перетворювали фізичні величини в пропорційні їм напругу або струм, виробляли над ними якісь операції і потім виконували зворотні перетворення в фізичні величини. Наприклад, голос людини (коливання повітря) за допомогою мікрофона перетворюється в електричні коливання, потім ці електричні сигнали посилюються електронним підсилювачем і за допомогою акустичної системи знову перетворюються в коливання повітря - в сильніший звук.

Однак аналогові сигнали і аналогова електроніка, яка з ними працює, мають великі недоліки, пов'язані саме з природою аналогових сигналів. Справа в тому, що аналогові сигнали чутливі до дії всіляких паразитних сигналів – шумів, наведень, перешкод. Шум – це внутрішні хаотичні слабкі сигнали будь-якого електронного пристрою (мікрофона, транзистора, резистора і т. д.).

Наведення і перешкоди – це сигнали, що приходять на електронну систему ззовні і спотворюють корисний сигнал (наприклад, електромагнітні випромінювання від радіопередавачів або від трансформаторів).

Всі операції, вироблені електронними пристроями над сигналами, можна умовно розділити на три великі групи:

- обробка (або перетворення);
- передача;
- зберігання.

У всіх цих трьох випадках корисні сигнали спотворюються паразитними - шумами, перешкодами, наводками. Крім того, при обробці сигналів (наприклад, при посиленні, фільтрації) ще і спотворюється їх форма - через недосконалість, неідеальності електронних пристроїв. А при передачі на великі відстані і при зберіганні сигнали до того ж послаблюються.

У разі аналогових сигналів все це істотно погіршує корисний сигнал, тому що всі його значення дозволені (рис. 1.2). Тому кожне перетворення, кожне проміжне зберігання, кожна передача по кабелю або ефіру погіршує аналоговий сигнал, іноді аж до його повного знищення. Треба ще врахувати, що всі шуми, перешкоди і наведення принципово не піддаються точному розрахунку, тому точно описати поведінку будь-яких аналогових пристроїв абсолютно неможливо. До того ж з часом параметри всіх аналогових пристроїв змінюються через старіння елементів, тому характеристики цих пристроїв не залишаються постійними.

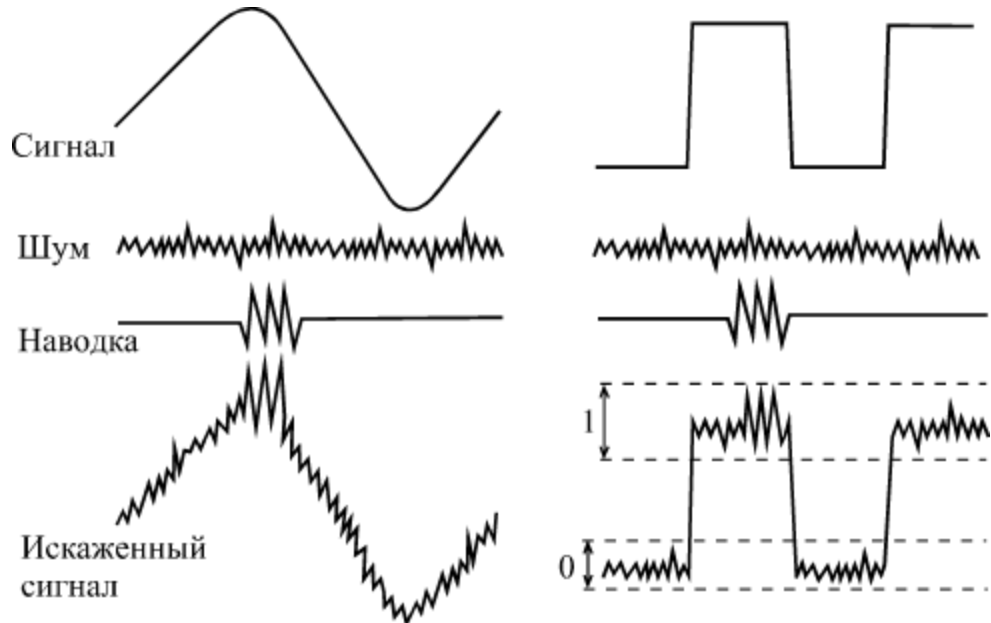


Рис. 2. Спотворення шумами і наведеннями аналогового (зліва) і цифрового (праворуч) сигналів.

На відміну від аналогових, цифрові сигнали, які мають лише два дозволених значення, захищені від дії шумів, наведень і перешкод набагато краще. Невеликі відхилення від дозволених значень ніяк не спотворюють цифровий сигнал, тому що завжди існують зони допустимих відхилень (рис. 1.2). Саме тому цифрові сигнали допускають набагато більш складну і

багатоступеневу обробку, набагато більш тривале зберігання без втрат і набагато більш якісну передачу, ніж аналогові. До того ж поведінку цифрових пристроїв завжди можна абсолютно точно розрахувати і передбачити. Цифрові пристрої набагато менше схильні до старіння, так як невелика зміна їх параметрів ніяк не відбивається на їх функціонуванні. Крім того, цифрові пристрої простіше проектувати і налагоджувати. Зрозуміло, що всі ці переваги забезпечують бурхливий розвиток цифрової електроніки.

Однак у цифрових сигналів є і великий недолік. Справа в тому, що на кожному зі своїх дозволених рівнів цифровий сигнал повинен залишатися хоча б протягом якогось мінімального тимчасового інтервалу, інакше його неможливо буде розпізнати. А аналоговий сигнал може приймати будь-яке своє значення нескінченно малий час. Можна сказати й інакше: аналоговий сигнал визначений в безперервному часу (тобто в будь-який момент часу), а цифровий - в дискретному (тобто тільки в виділені моменти часу). Тому максимально досяжна швидкодія аналогових пристроїв завжди принципово більше, ніж цифрових. Аналогові пристрої можуть працювати з більш мінливими сигналами, ніж цифрові. Швидкість обробки і передачі інформації аналоговим пристроєм завжди може бути вище, ніж швидкість обробки і передачі цифровим пристроєм.

Крім того, цифровий сигнал передає інформацію тільки двома рівнями і зміною одного свого рівня на інший, а аналоговий - ще і кожним поточним значенням свого рівня, тобто він більш ємний з точки зору передачі інформації. Тому для передачі того обсягу інформації, що міститься в одному аналоговому сигналі, найчастіше доводиться використовувати кілька цифрових (найчастіше від 4 до 16).

До того ж, як уже зазначалося, в природі всі сигнали - аналогові, тобто для перетворення їх в цифрові і зворотного перетворення потрібне застосування спеціальної апаратури (аналого-цифрових і цифро-аналогових перетворювачів).

Цифрова логіка, елементи, які її представляють, працюють з так званими цифровими сигналами. На відміну від аналогових, цифрові сигнали приймають два можливих значення: логічна одиниця і логічний нуль. Логічна одиниця позначається «1» або, в деяких випадках, «високим» рівнем («В»). Логічний нуль, відповідно, позначається «0» або «низьким» рівнем («Н»). Логічні елементи або елементи цифрової логіки побудовані на біполярних і польових транзисторах, що працюють в режимах насичення і відсічення.

Найбільшого поширення набули перевірені часом цифрові логічні елементи на основі біполярних транзисторів - ТТЛ-елементи (транзисторно-транзисторна логіка) і на основі польових транзисторів - КМОН-елементи (комплементарні, на основі переходів метал-оксидних-напівпровідників).

Логічні елементи ТТЛ, асортимент яких налічує до 200 найменувань різного ступеня інтеграції і функціонального призначення, працюють при напрузі живлення 5 В. Ці мікросхеми здатні працювати до частот 20 ... 100 МГц і споживають від джерела живлення значний струм.

Нижче розглянуті основні логічні елементи цифрової логіки.

sdobryday.net

ЛОГИЧЕСКИЙ ЭЛЕМЕНТ	УСЛОВНОЕ ОБОЗНАЧЕНИЕ	ГРАФИКИ СИГНАЛОВ	СХЕМНЫЙ ЭКВИВАЛЕНТ	ТАБЛИЦА ИСТИННОСТИ	ПРИМЕРЫ ЭЛЕМЕНТОВ				
	Россия	зарубежн.			ТТЛ	Зарубежные русских КМОП			
ПОВТОРИТЕЛЬ Repeater					A 0 1	Y 0 1	7407 K155/1H4	CD4010 K176/1H3	
НЕ NOT					A 0 1	Y 1 0	7424 K155/1H1	CD4049 K176/1H2	
ИЛИ OR					A 0 0 1 1	B 0 1 0 1	Y 0 1 1 1	7432 K155/1H1	
ИЛИ-НЕ OR-NOT					A 0 0 1 1	B 0 1 0 1	Y 1 0 0 0	7422 K155/1E1	CD4001 K176/1E5
И AND					A 0 0 1 1	B 0 1 0 1	Y 0 0 1 1	7428 K155/1H1	CD4081B K1561/1H2
И-НЕ AND-NOT					A 0 0 1 1	B 0 1 0 1	Y 1 1 0 0	7420 K155/1A3	4011 K176/1A7
ЭКВИВАЛЕНТНОСТЬ Equivalence		—			A 0 0 1 1	B 0 1 0 1	Y 1 0 1 1		
ИСКЛЮЧАЮЩЕЕ ИЛИ Excluding OR					A 0 0 1 1	B 0 1 0 1	Y 0 1 1 0	7486 K155/1H5	CD4030 K176/1H2

Page 22 of 22

Рис. 26.1

Рис. 3. Основні логічні елементи цифрової логіки.

На основі найпростіших елементів цифрової логіки можуть бути синтезовані практично будь-які і як завгодно більш складні пристрої цифрової логіки - тригери, лічильники, шифратори, дешифратори і інші. У той же час з більш складних елементів можуть бути отримані більш прості. У цьому можна легко переконатися, аналізуючи інформацію, наведену на рис. 3, або експериментально. Так, наприклад, з'єднавши разом входи А і В елементів АБО-НЕ або І-НЕ, можна отримати елемент НЕ.

Відзначимо попутно, що найчастіше «зайві» невикористовувані входи логічних елементів об'єднують з іншими виводами, або з'єднують із загальною шиною «земля» або шиною живлення.

2. Комбінаційна логіка її поняття та характеристики.

В теорії цифрових пристроїв **комбінаційною логікою (комбінаційною схемою)** називають логіку функціонування пристроїв комбінаційного типу. У комбінаційних пристроїв стан виходу однозначно визначається набором вхідних сигналів. Це відрізняє комбінаційну логіку від секвенційної логіки, в рамках якої вихідне значення залежить не тільки від поточного вхідного впливу, але й від передісторії функціонування цифрового пристрою. Іншими словами, секвенційна логіка припускає наявність пам'яті, яку комбінаційна логіка не передбачає.

Характеристика.

Комбінаційна логіка використовується в обчислювальних схемах для формування вхідних сигналів і для підготовки даних, які підлягають збереженню. На практиці обчислювальні пристрої зазвичай поєднують комбінаційну та секвенційну логіку. Наприклад, Арифметико-логічний пристрій (АЛП) для математичних обчислень містить комбінаційні вузли. Математику комбінаційної логіки забезпечує булева алгебра. Базовими операціями є: кон'юнкція $x \wedge y$, диз'юнкція $x \vee y$ і заперечення (інверсія) $\neg x$ або $\bar{x}$. У комбінаційних схемах використовуються логічні елементи: кон'юнктор (І), диз'юнктор (АБО), інвертор (НЕ), а також похідні елементи: І-НЕ, АБО-НЕ і «Рівнозначність». Найбільш відомі комбінаційні пристрої — це суматор, напісуматор, шифратор, дешифратор, мультиплексор і демультимплексор.

Представницькі норми.

Форми представлення логічних виразів засновані на поняттях «істина» (Т — true) і «хибність» (F — false). У двійковому обчисленні — це відповідно значення 1 і 0, якими кодуються пропозиціональні змінні. Вирази комбінаційної логіки можуть бути представлені у формі таблиці істинності, або у вигляді формули булевої алгебри. Нижче показаний приклад таблиці істинності для трьох змінних.

Таблиця істинності служить основою для подання логічного виразу у вигляді алгебраїчної формули:

$$x \wedge \bar{y} \wedge \bar{z} \vee x \wedge y \wedge z.$$

На відміну від таблиці, логічна формула здатна перетворюватися за правилами булевої алгебри. Таким чином знаходиться скорочений вираз:

$$x \wedge (\bar{y} \wedge \bar{z} \vee y \wedge z).$$

З точки зору комбінаційної логіки представлені формули визначають одну і ту ж функцію. Різниця лише в тому, що скорочена формула дозволяє реалізувати відповідну комбінаційну схему в більш компактному вигляді.

x	y	z	Логічна формула	Результат
F	F	F	$\bar{x} \wedge \bar{y} \wedge \bar{z}$	T
F	F	T	$\bar{x} \wedge \bar{y} \wedge z$	T
F	T	F	$\bar{x} \wedge y \wedge \bar{z}$	T
F	T	T	$\bar{x} \wedge y \wedge z$	F
T	F	F	$x \wedge \bar{y} \wedge \bar{z}$	T
T	F	T	$x \wedge \bar{y} \wedge z$	F
T	T	F	$x \wedge y \wedge \bar{z}$	F
T	T	T	$x \wedge y \wedge z$	T

3. Секвенційна логіка її поняття та характеристики.

Секвенційна логіка — це логіка пам'яті цифрових пристроїв. Назва «секвенційна» походить з англ. sequential. Відповідна логіка може називатися

також «послідовна», хоча останній термін переважно вживається у зв'язку з логічними автоматами.

Секвенційна логіка відрізняється від комбінаційної логіки тим, що моделює цифрові пристрої з урахуванням передісторії їх функціонування.

Характеристика.

Секвенційна логіка є розділом математичної логіки. Вона розвивається в рамках теорії цифрових схем в тісному зв'язку з комбінаційною логікою, булевою алгеброю і кінцевими автоматами. В залежності від регламенту функціонування цифрові пристрої підрозділяються на синхронні і асинхронні. Відповідно їх поведінка підкоряється або синхронній, або асинхронній логіці.

Види секвенційної логіки

- Синхронна секвенційна логіка
- Асинхронна секвенційна логіка

Синхронна секвенційна логіка.

При логічному моделюванні пристроїв з пам'яттю особлива роль відводиться фактору часу, який в синхронних схемах природним чином враховується тактами кінцевого автомата. Такти визначають моменти зміни станів автомата, тобто, синхронізують відповідну функцію. Математичний апарат синхронної логіки задають автоматні моделі Мілі і Мура.

Асинхронна секвенційна логіка.

Асинхронна секвенційна логіка для вираження ефекту запам'ятовування використовує моменти зміни станів, які задаються не в явному вигляді, а виходячи із зіставлення логічних величин за принципом «раніше-пізніше». Для асинхронної логіки достатньо встановити черговість зміни станів безвідносно будь-яких прив'язок до реального або віртуального часу.

Теоретичний апарат секвенційної логіки складають математичні інструменти секвенції і вен'юнкції, а також логіко-алгебраїчні рівняння на їх основі.

4. Асинхронна логіка її поняття та характеристики.

Асинхронна логіка — різновид взаємодії логічних елементів цифрових пристроїв. Відрізняється від синхронної тим, що її елементи діють асинхронно в часі, не підкоряючись глобальному генератору тактових імпульсів.

Характеристика

Асинхронна логіка являє собою свого роду симбіоз комбінаційної логіки і асинхронної секвенційної логіки. У теорії цифрових пристроїв асинхронна логіка відрізняється від синхронної тим, що її пропозиційні елементи діють асинхронно в часі, не підкоряючись загальному регулятору, або датчику часу. Це означає, що кожен вхід модельованого цифрового пристрою керується своїм власним часом, своїм генератором тактових імпульсів. Асинхронна логіка є розділом дискретної математики, і може розглядатися як прикладна дисципліна математичної логіки. Математичним апаратом асинхронної логіки служить Булева алгебра, яка доповнюється алгебраїчними інструментами асинхронної секвенційної логіки - вен'юнкцією і секвенцією.

Формули асинхронної логіки – це аналітичні вирази, в яких пропозиційні змінні з'єднані булевими операціями кон'юнкції, диз'юнкції і заперечення з участю логіко-динамічної операції вен'юнкція. Крім того, поряд з двійковими змінними, в формулах можуть бути присутніми секвенції. Перетворення формул асинхронної логіки підпорядковані певним правилам.

5. Цифрові системи передавання.

Цифрові системи передавання — системи передавання інформації, які працюють з цифровими даними — певною послідовністю електричних імпульсів. Для передавання аналогового сигналу по ЦСП на передавальній стороні сигнал спочатку перетворюється в цифрову (дискретну) форму за допомогою аналогово-цифрового перетворювача (АЦП). Якщо на приймальній стороні користувачу чи іншій системі потрібен той же сигнал в аналоговій формі — то використовується зворотній перетворювач — цифрово-аналоговий (ЦАП).

Основні схеми взаємодії апаратури

Співнаправлений інтерфейс (codirectional interface) — інформаційний і синхросигнал передаються від одного терміналу до іншого. Термінали між собою рівні і симетричні.

Різнонаправлений інтерфейс (contradirectional interface) — термінали нерівноправні, синхросигнали передаються від керуючого терміналу до керованого. Інформаційні сигнали симетричні.

Інтерфейс з центральним тактовим генератором (centralized clock interface) — синхросигнали приходять від тактового генератора, інформаційні сигнали симетричні.

Переваги

- Висока завадозахищеність
- Слабка залежність якості передавання від довжини лінії зв'язку
- Стабільність параметрів каналу зв'язку
- Ефективне використання пропускної здатності каналів зв'язку
- Можливість створення цифрової мережі зв'язку
- Високі техніко-економічні показники
- Висока енергоефективність

Недоліки

- Більша складність реалізації
- Забезпечення синхронізації

Тема 11. Представлення даних на машинному рівні.

1. Організація даних.

Розглянемо представлення даних, що відносяться до прикладної області на системному рівні. Воно визначається поглядом на дані із точки, що перебуває поза програмним середовищем, тобто або з боку користувача, або з боку системи. Зовнішнє представлення даних – це спосіб організації вхідних і вихідних даних. Одні дані утримуються в системі, інші – надходять на її вхід, треті – призначені для виводу й передачі користувачам. Внутрішнє представлення даних – це представлення даних усередині програм, доступне тільки програмісту.

Програміст, проектувальник і користувач мають свої власні погляди на організацію даних. Відповідно до цього можуть бути виділено три рівні організації даних:

- логічна організація даних: проектний рівень.
- представлення даних: рівень мови реалізації.
- фізична організація даних: машинний рівень.

Логічна організація даних відображає погляд користувача на дані. У її основі лежать вимоги користувача й внутрішньо властиві даним зв'язки. Це найбільш важливий рівень абстракції, використовуваний представленні даних, оскільки саме вимоги користувачів визначають вигляд проекрованої системи. Якщо на етапі проектування системи вдало обрана логічна організація даних, зміни системних вимог, що не приводять до модифікації логічної структури даних, не спричинять реорганізації на більш низьких рівнях представлення даних. Тільки на логічному рівні можуть застосовуватися формальні методи опису динамічно-змінюваних структур.

Логічна організація структур даних – це моделі структур, які не залежать від способу їх зберігання у комп'ютерній пам'яті. Логічну модель даних називають абстрактною моделлю.

Опис даних мовою програмування належить до рівня представлення даних. Відношення між даними задаються у вигляді, характерному для конкретної мови. На цьому рівні оперують масивами й вказівниками.

Інформація про представлення даних може бути розподілена по окремих програмних модулях, причому можна використовувати як зовнішню, так і внутрішню форми представлення даних. Під зовнішнім представленням розуміється погляд на дані з боку інших програм, тобто представлення на рівні потоків даних. При зовнішньому представленні головним є визначення можливих шляхів доступу.

Внутрішнє представлення – це представлення у вигляді внутрішніх областей зберігання даних, тобто структура даних може бути в зовнішньому представленні стеком, а у внутрішньому – масивом або зв'язаним списком. Припустимо, що дані про покупця включають номер рахунку, ім'я, адресу, дати попередніх витрат, платежі й рахунку. Зовнішньою формою їх представлення є окремі агрегати даних про особу покупця й про його платежі.

Внутрішнє представлення може складатися з таблиць, полів ознак і вказівників.

Фізична організація даних вказує на те, у якому вигляді дані зображуються в пам'яті комп'ютера. Фізична організація даних суттєво залежить від типу пам'яті, на якій вони записуються. Фізичну модель даних називають конкретною моделлю.

Рівень фізичної організації пов'язаний із системним програмним забезпеченням. На цьому рівні доводиться оперувати із межами слів, розмірами полів, двійковими кодами й фізичними записами. Більшість засобів системного програмного забезпечення дають можливість програмісту здійснити вибір з досить вузького кола способів представлення даних, які мають більш-менш ясну фізичну організацію. Агрегати даних, що зберігаються у швидкодіючій пам'яті, можуть бути представлені масивами або стеками. До записів файлу можна звертатися в послідовному або довільному порядку, використовуючи ключі різних типів.

Для подання інформації в пам'яті ЕОМ (як числовий, так і не числовий) використовується двійковий спосіб кодування.

Елементарна комірка пам'яті ЕОМ має довжину 8 біт (байт). Кожен байт має свій номер (його називають адресою). Найбільшу послідовність біт, яку ЕОМ може обробляти як єдине ціле, називають машинним словом. Довжина машинного слова залежить від розрядності процесора і може бути рівною 16, 32, 64 бітам і т.д.

2. Кодування символів.

Для кодування символів досить одного байту. При цьому можна уявити 256 символів (з десятковими кодами від 0 до 255). Набір символів персональних ЕОМ, сумісних з IBM PC, найчастіше є розширенням коду ASCII (American Standard Code for Information Interchange – стандартний американський код для обміну інформацією). В даний час використовуються і двобайтові представлення символів.

3. Двійково-десятькове кодування.

У деяких випадках при поданні чисел в пам'яті ЕОМ використовується змішана двійковий-десятьковий "система числення", де для зберігання кожного десятикового знаку потрібен напівбайт (4 біта) і десятикові цифри від 0 до 9 представляються відповідними двійковими числами від 0000 до 1001. Наприклад, упакований десятиковий формат, призначений для зберігання цілих чисел з 18-ю значущими цифрами і займає в пам'яті 10 байт (старший з яких знаковий), використовує саме цей варіант.

4. Подання цілих чисел в додатковому коді.

Інший спосіб представлення цілих чисел - додатковий код. Діапазон значень величин залежить від кількості біт пам'яті, відведених для їх зберігання. Наприклад, величини типу Integer (всі назви типів даних тут і

нижче представлені в тому вигляді, в якому вони прийняті в мові програмування Turbo Pascal. У інших мовах такі типи даних теж є, але можуть мати інші назви) лежать в діапазоні від -32768 (-215) до 32767 (215 - 1) і для їх зберігання відводиться 2 байти (16 біт); типу LongInt - в діапазоні від -231 до 231 - 1 і розміщуються в 4 байтах (32 біта); типу Word - в діапазоні від 0 до 65535 (216 - 1) (використовується 2 байти) і т.д.

Як видно з прикладів, дані можуть бути інтерпретовані як числа зі знаком, так і без знаку. У разі подання величини зі знаком найлівіший (старший) розряд вказує на додатне число, якщо містить нуль, і на від'ємне, якщо - одиницю.

Взагалі, розряди нумеруються справа наліво, починаючи з 0. Нижче показана нумерація біт в двобайтовому машинному слові.

Додатковий код додатного числа збігається з його прямим кодом. Прямий код цілого числа може бути отриманий таким чином: число переводиться в двійкову систему числення, а потім його двійкову запис зліва доповнюють такою кількістю незначних нулів, скільки вимагає тип даних, до якого належить число.

Наприклад, якщо число $37_{(10)} = 100101_{(2)}$ оголошено величиною типу Integer (шестнадцятибітове зі знаком), то його прямим кодом буде 0000000000100101, а якщо величиною типу LongInt (тридцятидвохбітове зі знаком), то його прямий код буде 00000000000000000000000000100101. Для більш компактного запису частіше використовують шістнадцяткове представлення коду. Отримані коди можна переписати відповідно як 0025 (16) і 00000025 (16).

Додатковий код цілого від'ємного числа може бути отриманий за наступним алгоритмом:

- 1.записати прямий код модуля числа;
- 2.інвертувати його (замінити одиниці нулями, нулі - одиницями);
- 3.додати до інверсного коду одиницю.

Наприклад, запишемо додатковий код числа -37, інтерпретуючи його як величину типу LongInt (тридцятидвохбітове зі знаком):

прямий код числа 37 є 00000000000000000000000000100101;

інверсний код 1111111111111111111111111011010;

додатковий код 1111111111111111111111111011011 або FFFFFFFD (16).

При отриманні числа по його додатковому коду перш за все необхідно визначити його знак. Якщо число виявиться додатнім, то просто перевести його код в десяткову систему числення. У разі від'ємного числа необхідно виконати наступний алгоритм:

- 1.відняти від коду числа 1;
- 2.інвертувати код;
- 3.перевести в десяткову систему числення. Отримане число записати зі знаком мінус.

Приклади. Запишемо числа, відповідні додатковим кодам:

0000000000010111.

Оскільки в старшому розряді записаний нуль, то результат буде додатнім. Це код числа 23.

111111111000000. Тут записаний код від'ємного числа. Виконуємо алгоритм: 1) 111111111000000 (2) - 1 (2) = 1111111110111111 (2); 2) 0000000001000000; 3) 1000000 (2) = 64 (10).

Відповідь: -64.

Кодування дійсних чисел.

Дещо інший спосіб застосовується для подання в пам'яті персонального комп'ютера дійсних чисел. Розглянемо уявлення величин з плаваючою точкою.

Будь-яке дійсне число можна записати в стандартному вигляді $M \times 10^p$, де $1 \leq M < 10$, p - ціле. Наприклад, $120100000 = 1,201 \times 10^8$. Оскільки кожна позиція десяткового числа відрізняється від сусідньої на ступінь числа 10, множення на 10 еквівалентно зсуву десяткової коми на одну позицію вправо. Аналогічно розподіл на 10 зрушує десяткову кому на позицію вліво. Тому наведений вище приклад можна продовжити: $120100000 = 1,201 \times 10^8 = 0,1201 \times 10^9 = 12,01 \times 10^7$. Десяткова кома "плаває" в числі і більше не позначає абсолютне місце між цілою і дробовою частинами.

У наведеній вище записи M називають мантисою числа, а p - його порядок. Для того щоб зберегти максимальну точність, обчислювальні машини майже завжди зберігають мантису в нормалізованому вигляді, що означає, що мантиса в даному випадку є число, яке лежить між 1 (10) і 2 (10) ($1 \leq M < 2$). Підстава системи числення тут, як уже зазначалося вище, - число 2. Спосіб зберігання мантиси з плаваючою точкою на увазі, що двійкова кома знаходиться на фіксованому місці. Фактично мається на увазі, що двійкова кома слід після першої двійковій цифри, тобто нормалізація мантиси робить одиничним перший біт, поміщаючи тим самим значення між одиницею і двійкою.

Місце, відведене для числа з плаваючою точкою, ділиться на два поля. Одне поле містить знак і значення мантиси, а інше містить знак і значення порядку.

Сучасний персональний комп'ютер дозволяє працювати з наступними дійсними типами (діапазон значень вказано за абсолютною величиною, а в деяких випадках перелік типів даних може бути розширений):

Тип	Діапазон	Мантиса	Байти
Real	$2,9 \times 10^{-39} \dots 1,7 \times 10^{38}$	11-12	6
Single	$1,5 \times 10^{-45} \dots 3,4 \times 10^{38}$	7-8	4
Double	$5,0 \times 10^{-324} \dots 1,7 \times 10^{308}$	15-16	8
Extended	$3,4 \times 10^{-4932} \dots 1,1 \times 10^{4932}$	19-20	10

Покажемо перетворення дійсного числа для подання його в пам'яті ЕОМ на прикладі величини типу Double.

Як видно з таблиці, величина цього типу займає в пам'яті 8 байт. На малюнку нижче показано, як тут представлені поля мантиси і порядку (нумерація бітів здійснюється справа наліво):

S	Смещенный порядок	Мантисса
	63	62..5251..0

Можна помітити, що старший біт, відведений під мантиссу, має номер 51, тобто мантиса займає молодші 52 біта. Риса вказує тут на положення двійковій коми. Перед коми повинен стояти біт цілої частини мантиси, але оскільки вона завжди дорівнює 1, тут даний біт не потрібно і відповідний розряд відсутній в пам'яті (але він мається на увазі). Значення порядку зберігається тут не як ціле число, представлене в додатковому коді.

Для спрощення обчислень і порівняння дійсних чисел значення порядку в ЕОМ зберігається у вигляді зміщеного числа, тобто до теперішнього значення порядку перед записом його в пам'ять додається зсув. Зсув вибирається так, щоб мінімального значення порядку відповідав нуль. Наприклад, для типу Double порядок займає 11 біт і має діапазон від 2-1023 до 2¹⁰²³, тому зсув дорівнює 1023 (10) = 1111111111 (2). Нарешті, біт з номером 63 вказує на знак числа.

Таким чином, з вищесказаного випливає наступний алгоритм для отримання уявлення дійсного числа в пам'яті ЕОМ:

- Перевести модуль даного числа в двійкову систему числення;
- нормалізувати двійкове число, тобто записати у вигляді $M \times 2^p$, де M - мантиса (її ціла частина дорівнює 1 (2)) і p - порядок, записаний в десятковій системі числення;
- додати до порядку зміщення і перевести зміщений порядок в двійкову систему числення;
- враховуючи знак заданого числа (0 - позитивне; 1 - негативне), виписати його подання до пам'яті EOM.

Приклад. Запишемо код числа -312,3125.

Двійковий запис модуля цього числа має вигляд $+100111000,0101$.

$$\text{Maemo} + 100111000,0101 = +1,001110000101 \times 28.$$

Отримуємо зміщений порядок $8 + 1023 = 1031$. Далі маємо $1031 (10) = 10000000111 (2)$.

[illegible]

Очевидно, що більш компактно отриманий код варто записати в такий спосіб: C073850000000000 (16).

5. Подання цілих чисел в беззнакових цілих типах.

Для беззнакового уявлення все розряди осередки відводяться під уявлення самого числа. Наприклад, в байті (8 біт) можна уявити беззнакові числа від 0 до 255. Тому, якщо відомо, що числова величина є невід'ємне, то вигідніше розглядати її як беззнакову.

6. Подання цілих чисел в знакових цілих типах.

Для уявлення зі знаком найстарший (лівий) біт відводиться під знак числа, інші розряди - під саме число. Якщо число позитивне, то в знаковий розряд поміщається 0, якщо негативне - 1. Наприклад, в байті можна уявити знакові числа від -128 до 127.

7. Подання цілих чисел в комп'ютері.

Цілі числа є простими числовими даними, з якими оперує ЕОМ. Для цілих чисел існують два подання: беззнакове (тільки для невід'ємних цілих чисел) і зі знаком. Очевидно, що негативні числа можна представляти тільки в знаковому вигляді. Цілі числа в комп'ютері зберігаються в форматі з фіксованою комою.

8. Прямий код числа.

Подання числа в звичній формі "знак" - "величина", при якій старший розряд осередку відводиться під знак, а решта - під запис числа в двійковій системі, називається прямим кодом двійкового числа. Наприклад, прямий код двійкових чисел 1001 і -1001 для 8-розрядної комірки дорівнює 00001001 та 10001001 відповідно.

Позитивні числа в ЕОМ завжди представляються за допомогою прямого коду. Прямий код числа повністю збігається із записом самого числа в комірці машини. Прямий код негативного числа відрізняється від прямого коду відповідного позитивного числа лише вмістом знакового розряду. Але негативні цілі числа не подаються у ЕОМ за допомогою прямого коду, для їх подання використовується так званий додатковий код.

9. Додатковий код числа.

Додатковий код позитивного числа дорівнює прямому коду цього числа. Додатковий код негативного числа m дорівнює $2^k - |m|$, де k - кількість розрядів в осередку.

Як вже було сказано, при поданні невід'ємних чисел в беззнакову форматі все розряди осередки відводяться під саме число. Наприклад, запис числа $243 = 11110011$ в одному байті при беззнакову поданні буде виглядати наступним чином:

1 1 1 1 0 0 1 1

При поданні цілих чисел зі знаком старший (лівий) розряд відводиться під знак числа, і під власне число залишається на один розряд менше. Тому, якщо наведене вище стан осередку розглядати як запис цілого числа зі знаком, то для комп'ютера в цьому осередку записано число -13 ($243 + 13 = 256 = 2^8$).

Але якщо це саме від'ємне число записати в осередок з 16-ти розрядів, то вміст комірки буде наступним:

1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1

Подання чисел з плаваючою комою.

При поданні чисел з плаваючою комою частина розрядів осередку відводиться для запису порядку числа, інші розряди - для запису мантиси. По одному розряду в кожній групі відводиться для зображення знаку порядку і знаку мантиси. Для того, щоб не зберігати знак порядку, був придуманий так званий зміщений порядок, який розраховується за формулою $2a-1 + \text{ПП}$, де a - кількість розрядів, що відводяться під порядок.

Приклад: Якщо істинний порядок дорівнює -5, тоді зміщений порядок для 4-байтового числа буде дорівнює $127-5 = 122$.

10. Алгоритм представлення числа з плаваючою комою.

1. Перевести число з r -ічної системи числення в двійкову;
2. Уявити двійкове число в нормалізованій експоненційній формі;
3. Розрахувати зміщений порядок числа;
4. Розмістити знак, порядок і мантису в відповідні розряди сітки.

Тема 12. Елементи та вузли цифрового комп'ютера.

1. Алгебра логіки. Логічні вентиля.

Логіка як наука про міркуваннях в класичному вигляді зародилася в стародавній Греції і спочатку розумілася як атрибут психіки людини. Розвиток Європейської культурної традиції, орієнтованої на об'єктивізації науки, привело до появи в середині XIX століття математичної формалізації логіки - Булевої алгебри, що розуміється як алгебра логіки.

У булевій логіці, що формалізувало Арістотелеву логіку, операнди і результати операцій (висловлювання в логіці висловлювань) брали тільки два значення: 0 - брехня і 1 - істина. Основні логічні операції:

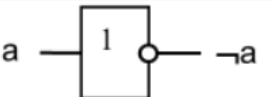
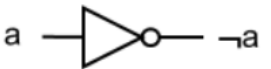
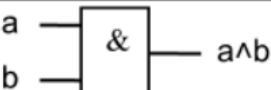
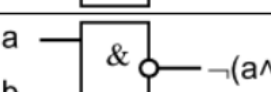
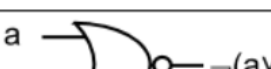
1. унарна (виконується над одним операндом) операція заперечення, позначається рівноправними символами $\neg$, $\neg$ або рискою над операндом;
2. бінарна операція кон'юнкції («І» або логічне множення), позначається символом $\wedge$;
3. бінарна операція диз'юнкції («АБО» або роз'єднання), позначається символом $\vee$.

На основі цих операцій можуть бути сконструйовані і інші, вже більш складні логічні операції.

На початку XX ст. з'явилися багатозначні логіки, в яких значень істинності було більше. А в другій половині XX ст. з'явилася теорія нечітких множин Л.Заде і, на її основі, широкий клас нечітких логік, в яких значення істинності можуть описуватися нечіткими множинами. Елементна база сучасних масових цифрових комп'ютерів реалізує двозначну, тобто булеву логіку, тому в курсі ми обмежимося тільки двозначною чи суворою логікою.

Пристрої, що виконують окремі логічні операції, називаються вентилями. Нижче наведено приклади умовних графічних позначень (УДО) логічних вентилів на схемах по ДСТУ 2.734-91 ЕСКД.

Таблиця 1. Умовно-графічні позначення основних логічних вентилів.

Логічний вентиль	ГОСТ 2.734-91	US ANSI 91-1984
Інвертор (НЕ)		
Кон'юнктор (И)		
Диз'юнктор (ИЛИ)		
Елемент Шеффера (И-НЕ) (Штрих Шеффера)		
Елемент Пірса (ИЛИ-НЕ) (Стрілка Пірса)		

Таблиця 2. Таблиці істинності вентилів Таб.1 в Булевій логіці.

Таблиця істинності схеми «НЕ»					
a	$\neg a$				
0	1				
1	0				
Таблиця істинності схеми «І»			Таблиця істинності схеми «АБО»		
a	b	$a \wedge b$	a	b	$a \vee b$
0	0	0	0	0	0
0	1	0	0	1	1
1	0	0	1	0	1
1	1	1	1	1	1
Таблиця істинності схеми «І-НЕ»			Таблиця істинності схеми «АБО-НЕ»		
a	b	$\neg ab$	a	b	$\neg (a \vee b)$
0	0	1	0	0	1
0	1	1	0	1	0
1	0	1	1	0	0
1	1	0	1	1	0

2. Тригер.

Апаратно, тобто «В залізі» логічні вентиля реалізуються тригерами.

Тригер – найпростіший послідовний пристрій, який може тривалий час перебувати в одному з декількох стійких станів і переходити з одного в інше під впливом вхідних сигналів. Послідовними називають такі логічні пристрої, вихідні сигнали яких визначаються не тільки сигналами на їх входах, а й передісторією їх роботи, тобто станом елементів пам'яті. Тригер – один з базових елементів цифрової техніки.

Тригерні схеми класифікують за такими ознаками:

- способу прийому логічних сигналів;
- функціональними можливостями;
- принципу побудови;
- числу стійких станів (зазвичай стійких станів два, рідше - більше);
- числу рівнів - два рівня (високий, низький) в дворівневих елементах, три рівня (позитивний, нуль, негативний) в тривірневої елементах.

За способом прийому сигналів розрізняють асинхронні, синхронні і змішані тригерні схеми, статичні і динамічні.

Асинхронний тригер змінює свій стан безпосередньо в момент появи відповідного інформаційного сигналу.

Синхронні тригери реагують на інформаційні сигнали тільки при наявності відповідного сигналу на так званому вході синхронізації С (від англ. Clock). Цей вхід також позначають термінами «стрибок», «такт». Синхронні

тригери в свою чергу поділяють на тригери із статичним (статичні) і динамічним (динамічні) управлінням по входу синхронізації С.

Статичні тригери сприймають інформаційні сигнали при подачі на вхід з логічної одиниці (прямий вхід) або логічного нуля (інверсний вхід).

Динамічні тригери сприймають інформаційні сигнали при зміні (перепаді) сигналу на вході С від 0 до 1 (прямий динамічний С-вхід) або від 1 до 0 (інверсний динамічний С-вхід).

Статичні тригери в свою чергу поділяють на одноступінчасті (однотактний) і двох-ступінчасті (двотактні).

У одноступенчатому тригері є один щабель запам'ятовування інформації, а в двохступенчатому – дві такі ступені. Спочатку інформація записується в першу щабель, а потім переписується в другу і з'являється на виході. Двоступеневий тригер позначають ТТ.

По структурній побудові – однотактний (тригери засувки), двотактні і тригери з динамічним управлінням. За способом реакції на перешкоди - прозорі і непрозорі. Непрозорі, в свою чергу, діляться на проникні і непроникні. За функціональним призначенням - RS, D, JK, T, RR, SS, EE, DV.

При виготовленні тригерів застосовуються переважно напівпровідникові прилади (зазвичай польові транзистори), в минулому – електронні лампи, реле. В даний час в основному розробляються програмовані логічні інтегральні схеми (ПЛІС).

Тригери використовуються в обчислювальній техніці як основний елемент для побудови більш складних компонентів обчислювальних систем: процесорів, регістрів, лічильників, ОЗУ.

За функціональними можливостями тригери поділяють на такі класи:

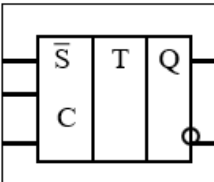
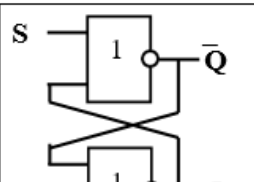
- з роздільним установкою стану 0 і 1 (RS-тригери). Якщо тригер є синхронним - додається вхід синхронізації С;
- універсальні (JK-тригери);
- з прийомом інформації по одному входу D (D-тригери, або тригери затримки);
- з рахунковим входом Т (Т-тригери). Кожен тип тригера має власну таблицю роботи (таблицю істинності). Вихідна стан тригера зазвичай позначають буквою Q. Індекс біля букви означає стан до подачі сигналу (t) або після подачі сигналу (t + 1).

Якщо тригер синхронний, то існує також додатковий вхід синхронізації. Для того, щоб такий тригер врахував інформацію на синхронних входах, на вході синхронізації необхідно сформувати активний фронт (зазвичай позитивний).

Тригери зазвичай реалізуються на основі логічних схем «І-НЕ» або «АБО-НЕ». Нижче наведено основні варіанти реалізації тригерів за допомогою розглянутих вище логічних вентилів і їх УДО.

3. Види тригерів.

RS-тригер (від англ. set - установка, і reset - відновлення, тобто тригер з установочними входами). Це тригер з роздільними установками стану 0 і 1, має два симетричних входи S і R і два симетричних виходи Q і $\bar{Q}$, причому вихідний сигнал $\bar{Q}$ є логічним запереченням сигналу Q. Якщо тригер є синхронним, то додається вхід синхронізації C. Приклад УДО RS тригера, його реалізація на елементах «2АБО-НЕ» і таблиця істинності дані нижче.

		<table><tr><th>S</th><th>R</th><th>Q</th><th>¬Q</th></tr><tr><td>0</td><td>0</td><td colspan="2">Зберігання біта</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td colspan="2">Заборонений стан! Порушення закону несуперечності і невизначеність стану.</td></tr></table>	S	R	Q	¬Q	0	0	Зберігання біта		1	0	1	0	0	1	0	1	1	1	Заборонений стан! Порушення закону несуперечності і невизначеність стану.	
S	R	Q	¬Q																			
0	0	Зберігання біта																				
1	0	1	0																			
0	1	0	1																			
1	1	Заборонений стан! Порушення закону несуперечності і невизначеність стану.																				
Мал.5.1. УДО RS тригера (синхронного)	Мал.5.2 RS тригер на елементах «2АБО-НЕ»																					

Тригер може бути реалізований і на елементах «2 І-НЕ».

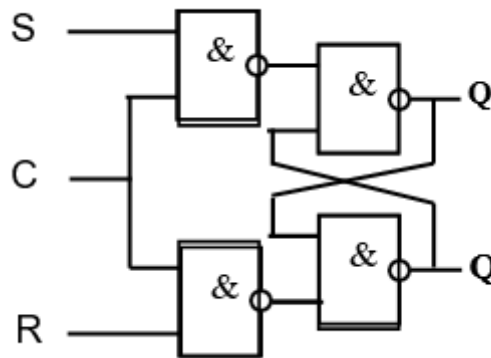
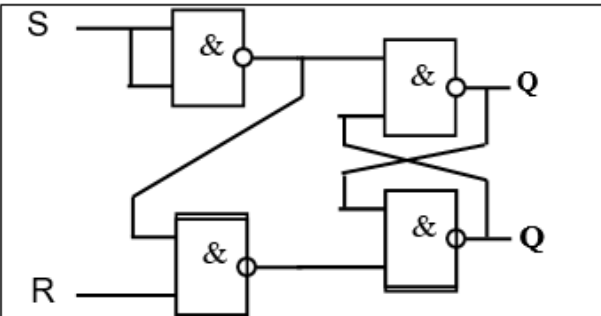


Рис. 3. RS тригер (синхронний) на елементах «2 І-НЕ».

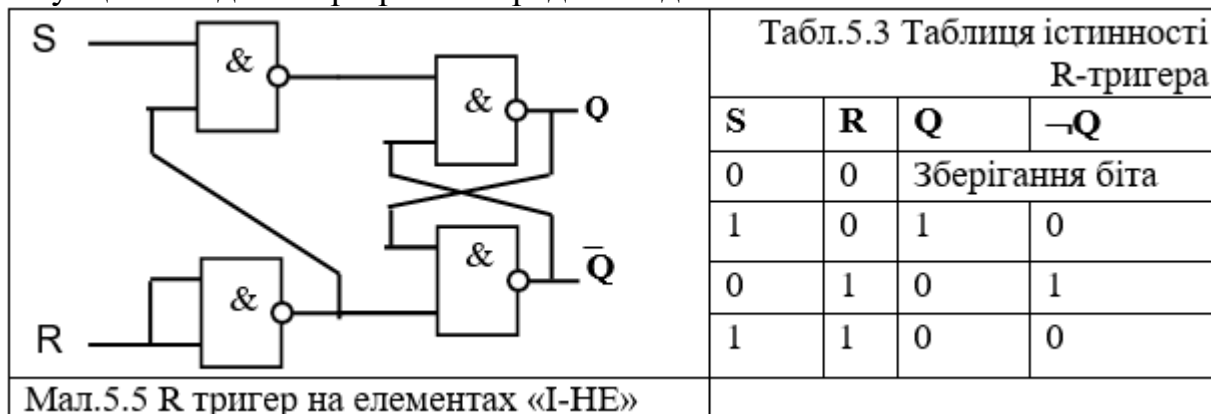
S-тригер - модифікація RS-тригера, виходи якого при комбінації сигналів $R = S = 1$ приймають одиничний стан, а при всіх інших комбінаціях вхідних сигналів функціонує як RS-тригер. S - тригер - пріоритетний тригер, тому що S вхід має пріоритет перед R входом.

	Табл.5.2 Таблиця істинності S-тригера			
	S	R	Q	$\bar{Q}$
	0	0	Зберігання біта	
	1	0	1	0
	0	1	0	1
	1	1	1	1
Мал.5.4 S тригер на елементах «І-НЕ»				

Стан виходу S-тригера на $n+1$ кроці може бути описано наступним чином:

$$Q^{n+1} = S^n + \bar{R}^n Q^n = S^n (1 + Q^n) = S^n + S^n Q^n + \bar{R}^n Q^n = S^n (\overline{S^n R^n}) Q^n \quad (5.1)$$

R-тригер - модифікація RS-тригера, виходи якого при комбінації сигналів $R = S = 1$ приймають нульовий стан, а при всіх інших комбінаціях вхідних сигналів функціонує як RS-тригер. R - тригер - пріоритетний тригер, тому що R вхід має пріоритет перед S входом.



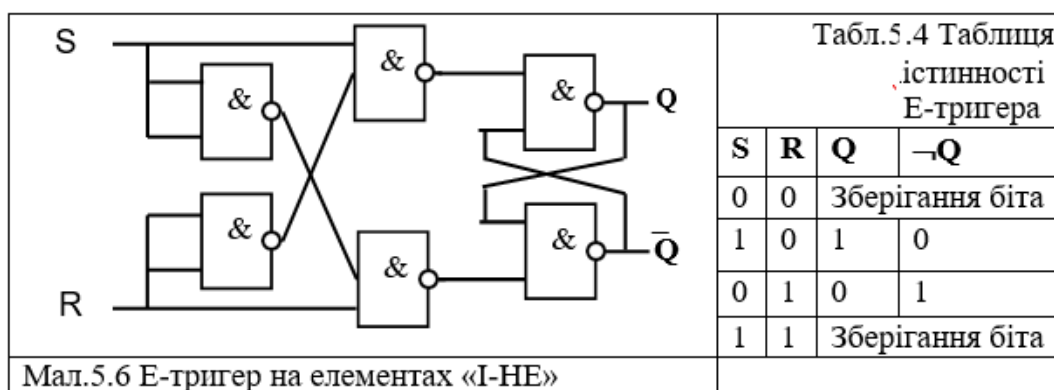
Стан виходу R-тригера на $n+1$ кроці може бути описано наступним чином:

$$Q^{n+1} = \bar{R}^n S^n + \bar{R}^n Q^n = \overline{(R^n S^n)} \overline{(R^n Q^n)} \quad (5.2)$$

Е-тригер - модифікація RS-тригера, виходи якого при комбінації сигналів $R = S = 1$ не змінюють свій стан, а при всіх інших комбінаціях вхідних сигналів функціонує як RS-тригер. Е - тригер - пріоритетний тригер, тому що той вхід, на якому сигнал з'явився першим, має пріоритет перед іншим входом.

Стан виходу Е-тригера на $n+1$ кроці може бути описано так:

$$Q^{n+1} = S^n Q^n + S^n \bar{R}^n + \bar{R}^n Q^n = \overline{(S^n R^n)} \overline{(S^n R^n)} Q^n \quad (5.3)$$



D-тригер (від англ. Delay - затримка). В RS-тригерах для запису логічного нуля і логічної одиниці потрібні різні входи, що не завжди зручно. При записі і зберіганні даних один біт може приймати значення, як нуля, так і одиниці. Для його передачі досить одного проводу. Оскільки сигнали установки і скидання тригера не можуть з'являтися одночасно, то можна

об'єднати ці входи за допомогою інвертора. D- тригер здатний запам'ятовувати по синхросигналам і зберігати один біт двійкової інформації.

Табл.5.5 Таблиця істинності синхронного D-тригера

C	D	Q(t)	Q(t+1)
0	Довільне	0	0
0	Довільне	1	1
1	0	Довільне	0
1	1	Довільне	1

Мал.5.7 D-тригер на елементах «I-HE»

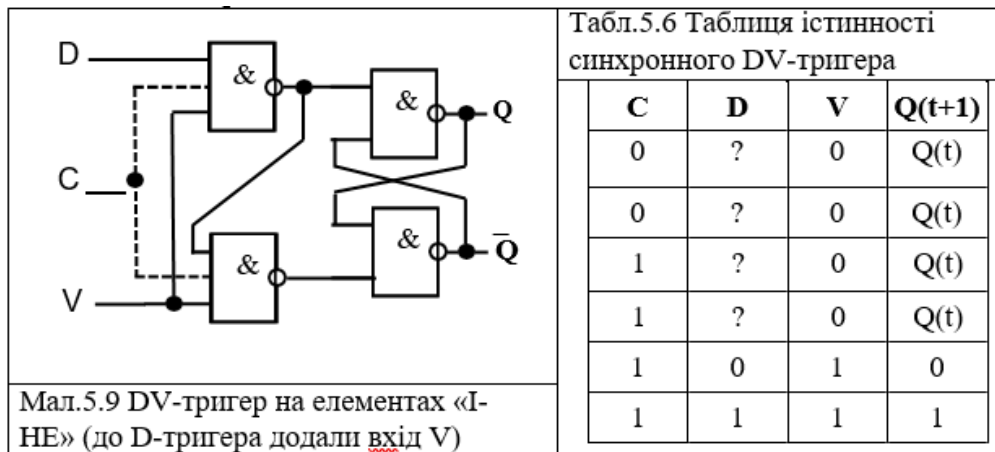
У Таблиці 5. першим двом рядкам відповідає режим зберігання інформації, а двом останнім - режим запису інформації.

D-тригери, що працюють по фронту. Вище наведено D-тригер, що працює за потенціалом. Потенціал, високий або низький, може зберігатися тривалий час. У зв'язку з цим інтерес представляють D тригери, що працюють по фронту, так як фронт сигналу синхронізації завжди обмежений у часі. В ідеалі тривалість фронту дорівнює нулю. Тому в тригері, що запам'ятовує вхідну інформацію по фронту не потрібно пред'являти вимоги до тривалості тактового сигналу. Тригер, запам'ятовує вхідну інформацію по фронту сигналу синхронізації, може бути побудований з двох тригерів, які працюють за потенціалом.

Мал.5.8 D-тригер, що працює по фронту

Мал.5.9 Тимчасові діаграми D-тригера, що працює по фронту

DV - тригер - це D-тригер, який має додатковий керуючий вхід V. При V = 1 DV-тригер працює за правилами D-тригера. При сигналі V = 0 тригер зберігає свій стан незмінним - «замикається» і зберігає інформацію незалежно від стану входу D. Наявність додаткового входу V розширює функціональні можливості D-тригера - інформація на виходах в потрібні моменти може бути збережена протягом необхідного числа тактів.



У **Т-тригера** тільки один вхід. (У TV - два входи, доданий керуючий вхід, як у DV-тригера.).

Після надходження на вхід Т імпульсу, стан тригера змінюється на прямо протилежний. При отриманні другого імпульсу Т-тригер скидається в початковий стан. Таким чином, Т-тригер лічильник вельми своєрідний - рахувати він вміє тільки до одного.

Використовуються Т-тригери для побудови схем різних лічильників. Т тригери можна реалізувати тільки на базі двоступеневих тригерів, подібних D тригеру. Використання двох тригерів дозволяє уникнути невизначеного стану схеми при дозвільному потенціал на вході синхронізації С.

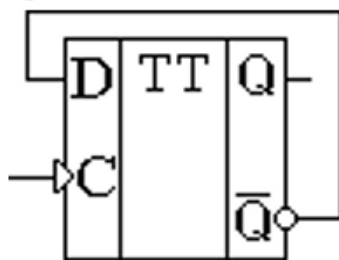


Рис.5.10 Т-тригер на основі D-тригера

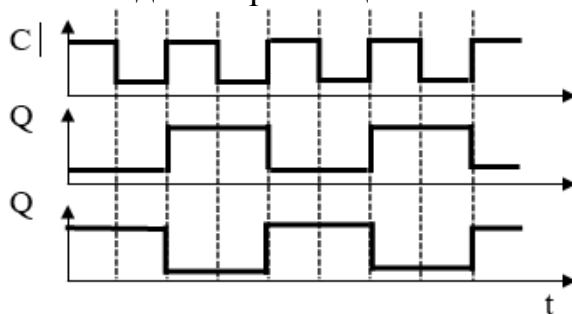


Рис.5.11 Тимчасові діаграми Т-тригера, побудованого на основі D-тригера, що працює по задньому фронту сигналу

JK-тригери - універсальні (Jump-Kill). В RS тригері є заборонені комбінації входів - одночасна подача одиничних сигналів на входи R і S. У JK тригері заборонений стан виключено.

Для цього його схема змінена так, щоб при подачі двох одиниць JK тригер перетворився в рахунковий (Т-тригер). Тоді, при подачі на вхід С імпульсів синхронізації цей тригер повинен змінити свій стан на протилежний. Для реалізації рахункового режиму в схемі (рис.12) введений перехресний зворотній зв'язок з виходів другого тригера на входи R і S першого тригера. Зворотній зв'язок на R і S входах першого тригера гарантує неможливість забороненої комбінації, а її перехресного дає новий режим роботи - рахунковий. В результаті, при подачі на входи J і K логічної одиниці, JK-тригер переходить в рахунковий режим, подібно Т тригеру. Таблиця істинності JK тригера практично збігається з таблицею істинності синхронного RS-тригера.

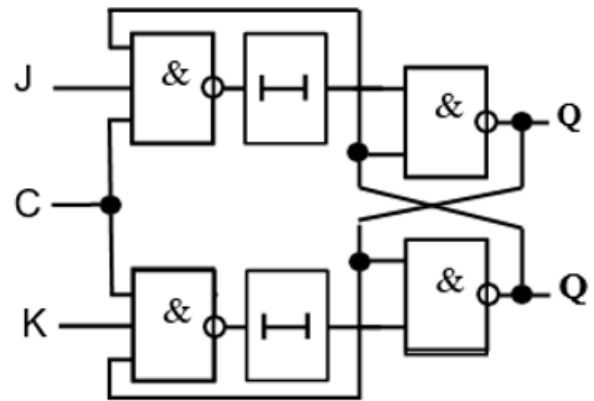
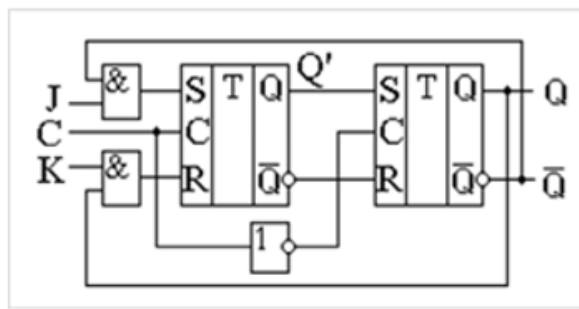


Рис. 12. JK-тригер.

4. Регістри.

Регістр – це послідовний логічний пристрій, що використовується для зберігання n -розрядних двійкових чисел і виконання перетворень над ними. Регістр являє собою упорядковану послідовність тригерів, число яких відповідає числу розрядів в машинному слові. З кожним регістром зазвичай пов'язаний комбінаційний цифровий пристрій, за допомогою якого забезпечується виконання необхідних операцій над словами. Фактично будь-який цифровий пристрій можна представити у вигляді сукупності регістрів, з'єднаних один з одним за допомогою комбінаційних цифрових пристроїв. Регістри будуються зазвичай на основі D-тригерів (з прийомом інформації по одному входу).

Основні операції, що виконуються регістром:

- прийом слова в регістр;
- передача слова з регістру;
- порозрядні логічні операції;
- зрушення слова вліво або вправо на задане число розрядів;
- перетворення послідовного коду слова в паралельний і *послідовний*;

- установка регістра в початковий стан (скидання).

Класифікація регістрів спочатку зазвичай проводиться в два класи:

- накопичувальні (регістри пам'яті, зберігання);
- зсувні, що використовуються для обчислень.

У свою чергу, зсуваються регістри класифікуються:

- за способом введення-виведення інформації:
 - паралельні - запис інформації проводиться одночасно по всьому входу, аналогічно і зчитування йде паралельно зі всіх виходів;
 - послідовні - запис і зчитування інформації відбувається в перший тригер, а інформація з цього тригера перезаписується в наступний, аналогічна процедура застосовується і до решти тригерів, формує регістр;
- комбіновані;
- за напрямком передачі інформації:
 - односпрямовані;
 - реверсивні.

- за основою системи числення;
- виконавчі;
- трійчастий;
- десяткові.

Паралельні або статичні регістри. У паралельних регістрах прийом і видача слів виробляються в усіх розрядах одночасно. Використовуються для зберігання слів, які підлягають порозрядним логічним перетворенням. Схеми розрядів в цих регістрах не обмінюється даними між собою. Загальні для розрядів зазвичай ланцюга управління: тактирування, скидання / установки, дозволу виходу або прийому.

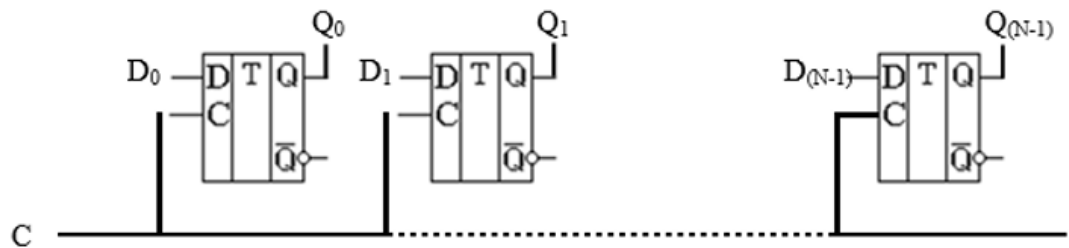


Рис. 13. Паралельний регістр.

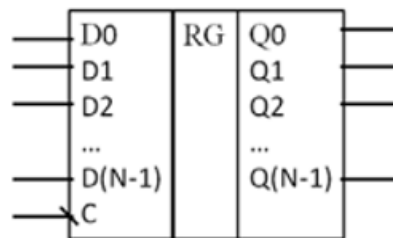


Рис. 14. УГО паралельного регістру.

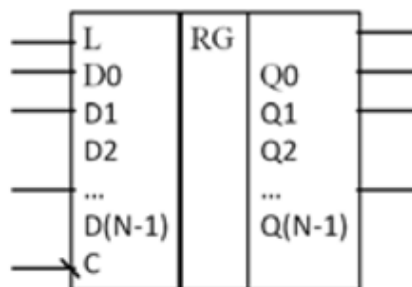


Рис. 15. УГО паралельного регістру зі входом дозволу запису.

Пристрої (в тому числі - регістри), в яких для запису вхідного паралельного коду D_i використовується сигнал дозволу запису L , а тактовий сигнал C не використовується, називаються пристроями з **асинхронним паралельним записом коду** (Рис.5.14).

Пристроями з **синхронним паралельним записом коду** називаються такі пристрої, в тому числі регістри (рис.15), в яких для запису вхідного паралельного коду D_i необхідні:

1. сигнал дозволу запису L .

2. перепад синхросигналу на тактовому вході С.

Послідовні (зсувні) регістри представляють собою ланцюжок розрядних схем, пов'язаних ланцюгами переносу. У одноканальних регістрах слово зсувається при надходженні синхросигналу. Вхід і вихід - послідовні. Діляться на:

1. З зрушенням вправо, позначаються DSR - Data Serial Right;
2. З зрушенням вліво, позначаються DSL - Data Serial Left;
3. Реверсивні регістри (зрушення вправо або вліво);
4. Послідовно-паралельні регістри - в них паралельний або послідовний режим запису-зчитування інформації визначається сигналом на спеціальному вході F.

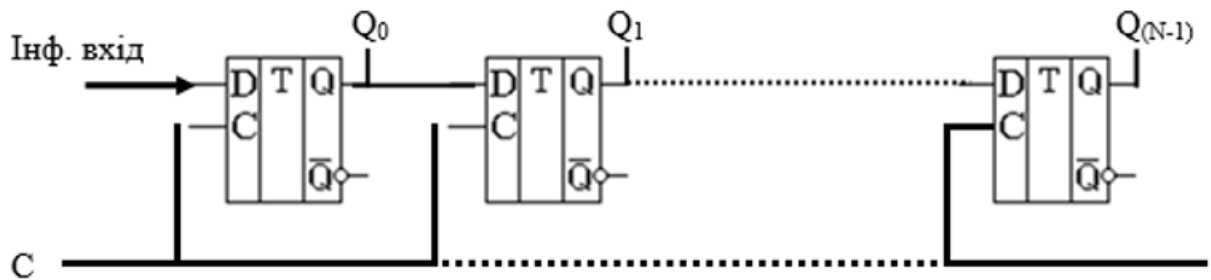


Рис. 16. Послідовний регістр із зсувом вправо (DSR).

5. Шифратор.

Шифратор (CD - coder) перетворює необхідний десятковий номер активного елементу в двійковий код. Виділяють шифратори:

- **пріоритетні**, які виконують перетворення максимального десяткового номера активного входу в двійковий еквівалент цього номера;
- **непріоритетні**, що виконують перетворення десяткового номера активного входу в двійковий еквівалент цього номера.

Непріоритетний шифратор перетворює з унітарного коду (прямого або інверсного) в натуральний двійковий код (n-розрядний двійковий код).

Застосування шифраторів:

1. стиснення інформації (завдяки тому, що число розрядів натурального двійкового коду менше числа розрядів унітарного коду);
2. введення даних в ЦК з клавіатури: введення числових даних з клавіатури виконується натисненням однієї з десяти клавіш, що кодується в унітарний код, а введення цих даних в процесор виробляється вже в двійковому коді, який формується шифратором - при натисканні клавіші на одному з входів шифратора з'являється логічна 1 і на виходах встановлюється двійковий код, відповідний символу цієї кнопки.

Пріоритетний шифратор - на його вхід може бути поданий довільний двійковий код. ГДК може містити довільне число одиниць (або нулів для інверсного), розташованих в довільному порядку. На виході пріоритетного шифратора формується натуральний двійковий код, що визначає номер позиції пріоритетної одиниці, тобто одиниці, що стоїть в самому старшому розряді.

Пріоритетні шифратори можуть використовуватися в якості звичайних шифраторів, оскільки унітарний код є окремим випадком довільного. Застосовуються для перетворювача двійкових чисел з формату з фіксованою комою в формат з плаваючою комою (порядок визначається за положенням старшої одиниці), і в аналого-цифрових перетворювачах паралельного типу.

Логіка роботи шифратора 4/2 (Рис.17 - 19) описується так:

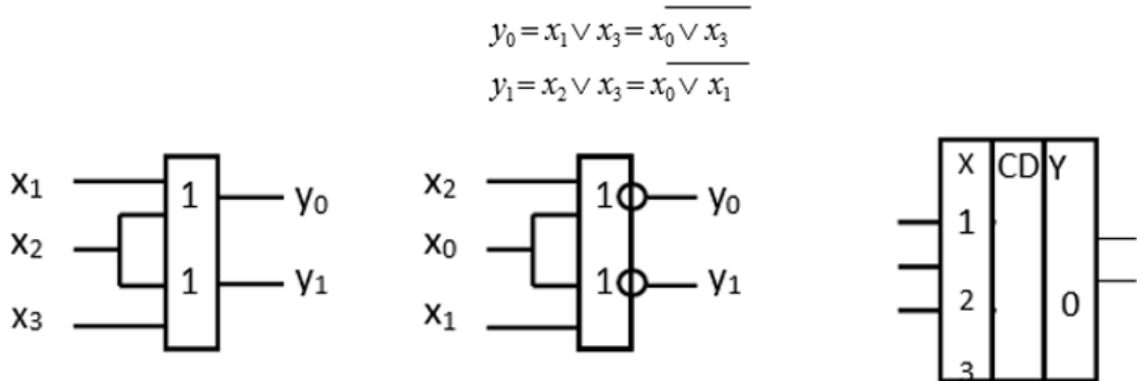


Рис. 17. Побудова шифратора 4/2 на елементах «АБО», «АБО-НІ», УГП шифратора.

6. Дешифратор.

Дешифратор (DC - decoder) – це елемент, функціонально зворотний **шифратору**, тобто комбінаційний пристрій, що перетворює n-разрядний двійковий код в логічний сигнал, що з'являється на тому виході, десятковий номер якого відповідає бінарного коду.

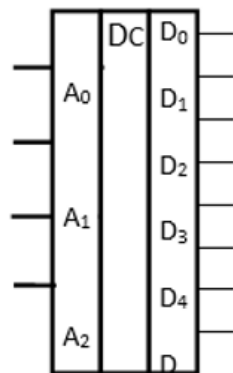
Дешифратор працює наступним чином: двійкове слово $x_{N-1}x_{N-2} \dots x_0$ поразрядно подається на N входів, тоді на виході формується такий код розрядності менший або рівний $2N$, що розряд, номер якого дорівнює вхідному слову, приймає значення одиниця, а всі інші розряди дорівнюють нулю. Можна бачити, що максимально можлива розрядність вихідного слова дорівнює $2N$. Такий дешифратор називається повним. Якщо частина вхідних наборів не використовується, то число виходів менше $2N$, і дешифратор називається неповним.

Часто дешифратори доповнюються входом дозволу роботи E (від enable). Якщо на E-вхід надходить одиниця, то дешифратор функціонує, інакше на виході дешифратора виробляється логічний нуль незалежно від вхідних сигналів. Варіант - дешифратор з входом V (veto). Вхід V дозволяє не тільки забороняти роботу пристрою, але і нарощувати його розрядність. Коли на цей вхід подається логічна одиниця, незалежно від стану інформаційних входів на виході утворюються нулі.

Якщо на вході V дешифратора логічний нуль, то на виході, номер якого відповідає бінарного коду адреси, позначеного на вході, формується логічна одиниця, а на інших виходах нулі.

Існують дешифратори з інверсними виходами, у такого дешифратора обраний розряд показаний нулем.

Логіка роботи дешифратора описується системою кон'юнкція, приклад наведений нижче.



$$D_0 = \bar{V} \cdot \bar{A}_2 \cdot \bar{A}_1 \cdot A_0$$

$$D_1 = \bar{V} \cdot \bar{A}_2 \cdot \bar{A}_1 \cdot A_0$$

$$D_2 = \bar{V} \cdot \bar{A}_2 \cdot A_1 \cdot \bar{A}_0$$

$$D_3 = \bar{V} \cdot \bar{A}_2 \cdot A_1 \cdot A_0$$

$$D_4 = \bar{V} \cdot A_2 \cdot \bar{A}_1 \cdot \bar{A}_0$$

$$D_5 = \bar{V} \cdot A_2 \cdot \bar{A}_1 \cdot A_0$$

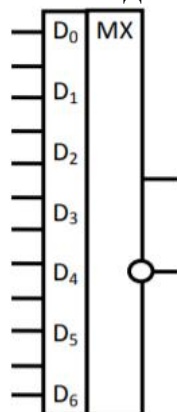
$$D_6 = V \cdot A_2 \cdot A_1 \cdot \bar{A}_0$$

$$D_7 = \bar{V} \cdot A_2 \cdot A_1 \cdot A_0$$

Рис. 20. УГП роботи дешифратора 3/8 та логіка його роботи.

7. Мультиплексор та демультиплексор.

Мультиплексор (multiplexer - комутатор, multi - багато, plex - мережа) передає сигнал з одного з входів, зазначеного в адресі, на загальний вихід, тобто виконує роль селектора каналів. Застосовуються як універсальні перемикачі та пристрої, що реалізують довільні логічні функції. Можуть будуватися на основі дешифраторів.



$$y = V \cdot \left(D_0 \cdot \bar{A}_2 \cdot \bar{A}_1 \cdot \bar{A}_0 \vee D_1 \cdot \bar{A}_2 \cdot \bar{A}_1 \cdot A_0 \vee D_2 \cdot \bar{A}_2 \cdot A_1 \cdot \bar{A}_0 \vee D_3 \cdot \bar{A}_2 \cdot A_1 \cdot A_0 \vee D_4 \cdot A_2 \cdot \bar{A}_1 \cdot \bar{A}_0 \vee D_5 \cdot A_2 \cdot \bar{A}_1 \cdot A_0 \vee D_6 \cdot A_2 \cdot A_1 \cdot \bar{A}_0 \vee D_7 \cdot A_2 \cdot A_1 \cdot A_0 \right)$$

Рис. 21. УГП мультиплексора та логіка його роботи.

Демультиплексор перемикає сигнал з єдиного входу на один з виходів – той, який зазначений в адресі. Демультиплексор складається з дешифратора, керованого адресою, і кон'юнкторів, керованих інформаційним сигналом і виходами дешифратора.

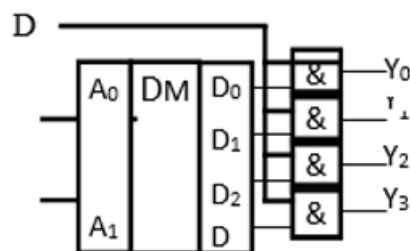


Рис. 22. Демультиплексор.

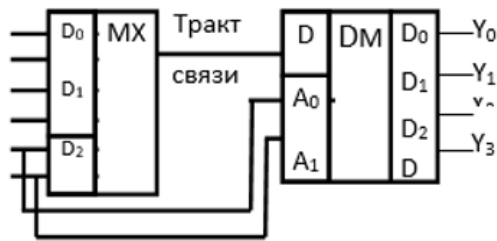


Рис. 23. Мультиплексована лінія зв'язку.

Лічильник – це пристрій, на виходах якого формується двійковий (двійково-десятковий) код, який визначається кількістю імпульсів, що надійшли до нього. Лічильники можуть будуватися на Т-тригерах. Основний параметр лічильника - модуль рахунку, він визначається як максимальне число одиничних сигналів, яке може бути пораховано лічильником. Лічильники на схемах позначають аббревіатурою СТ (від англ. Counter).

по модулю рахунку	по напрямку рахунку	по способу формування внутрішніх зв'язків
двійково-десяткові; двійкові; з довільним постійним модулем рахунку; зі змінним	підсумовують; віднімаються; реверсивні;	з послідовним переносом; паралельним переносом; комбінованим переносом; кільцеві.

Перетворювачі кодів (ПК) діляться на:

- вагові - перетворюють інф. з однієї системи числення в іншу;
- невагові - перетворюють інформацію для подальшого відображення.

Приклад: **перетворювач двійково-десяткового коду** в код для семисегментних світлодіодних індикаторів. На рис. 24 показаний фрагмент підключення одного сегмента до виходу схеми із загальним емітером і приведені накреслення перших п'яти цифр.

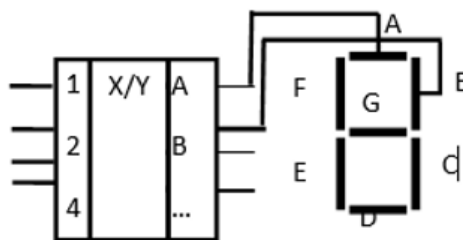


Рис. 24. Перетворювач двійково-десяткового коду в код для семисегментних світлодіодних індикаторів.

Тема 13. Архітектура і організація систем пам'яті.

1. Загальна інформація.

Персональний комп'ютер - це сукупність апаратних і програмних засобів, призначених для обробки інформації. Він зазвичай містить один основний процесор і можливо, кілька співпроцесорів, має фіксований склад і універсальне застосування

Архітектура - це множина ресурсів комп'ютера, доступних користувачеві на логічному рівні, без деталізації способів взаємодії процесорів, пристроїв пам'яті, зовнішніх пристроїв і програмних засобів.

Обчислювальні можливості комп'ютера в основному визначаються характеристиками її пам'яті.

Пам'ять складається з величезного числа елементів, кожен з яких може перебувати в одному з двох станів, що кодуються двійковою цифрою 0 або 1 (біти, що кодують стану елементів). Біти пам'яті групуються в більші одиниці зберігання інформації, мінімальної з яких є байт. Незалежно від моделі комп'ютера довжина байту інформації фіксована і складає 8 біт (двійкових розрядів).

За своїм функціональним призначенням пристрої, що запам'ятовують можна розділити на великі класи:

- реєстрові внутрішні запам'ятовуючі пристрої;
- Кеш-пам'ять;
- основна пам'ять;
- зовнішні запам'ятовуючі пристрої (ЗЗП).

Реєстрові запам'ятовуючі пристрої входять до складу центрального процесора, служать для проміжного зберігання даних, адресною і керуючою інформації. Ці ЗП володіють найбільшим швидкодією і зазвичай невеликі за обсягом. Сукупність реєстрів основного призначення (РОП), що утворюють локальну пам'ять процесора, призначена для підвищення швидкодії процесора, оскільки завантаження в АЛП операнди виконуваної команди з РОП набагато швидше, ніж з оперативної пам'яті.

До класу *зовнішніх запам'ятовуючих пристроїв (ЗЗП)* відносяться енергонезалежні пристрої з пам'яттю (магнітні та оптичні диски, флеш-карти, магнітні стрічки). Важливо не тільки те, що прямого доступу до них центральний процесор не має. Перш ніж скористатися інформацією, що зберігається на ЗЗП, вона повинна бути перенесена в оперативну пам'ять. ЗЗП служать для зберігання великих обсягів інформації, використовуються для створення бібліотек, архівів, баз даних.

Пристрої, що запам'ятовують, що входять до складу основної пам'яті, становлять найважливіший модуль будь-якого комп'ютера, в них зберігаються програми і дані, що обробляються центральним процесором. У складі основної пам'яті виділимо оперативний пристрій (ОЗП) і постійний запам'ятовуючий пристрій (ПЗП).

Далі розглянемо кілька типів запам'ятовуючих пристроїв, що входять до складу основної пам'яті комп'ютера.

Постійний запам'ятовуючий пристрій (ПЗП). В англійській термінології такого роду пам'ять називають ROM (Read Only Memory) – пам'ять, що працює тільки на зчитування. Інформація, що знаходиться в такій пам'яті, заздалегідь закладається при її виготовленні («зашивається») і при відключенні харчування не руйнується.

В постійній пам'яті зберігаються програми, що забезпечують роботу комп'ютера після його включення в мережу (BIOS), що забезпечують перевірку працездатності комп'ютера. Тут же зберігаються дані, які не змінюються в процесі експлуатації. Постійна пам'ять використовується тільки в режимі читання інформації. Система BIOS пов'язана з аббревіатурою CMOS. Ця назва дана постійної перепрограмованій пам'яті лежить в основі її виготовлення технології CMOS – Complementar Metal-Oxide-Semiconductor. В системі BIOS є програма Setup, яка може змінювати вміст CMOS пам'яті в залежності від конфігурації комп'ютера. У мікросхемі CMOS реалізовані також годинник реального часу RTS (Rial Time Clock). Вони працюють і при вимкненому з мережі комп'ютері від спеціальної батареї. Годинники дозволяють стежити за поточним часом, користувач комп'ютера завжди може дізнатися час, число, місяць, рік, скористатися програмами, які обмежать час використання комп'ютера для ігор дітьми. Комп'ютер може нагадати його господареві про необхідність вжити заздалегідь заплановані на певний час дії, включити в певний час електронну техніку, або вимкнути її і т.д. Залежно від конфігурації комп'ютера. У мікросхемі CMOS реалізовані також годинник реального часу RTS (Rial Time Clock). Вони працюють і при вимкненому з мережі комп'ютері від спеціальної батареї. Годинники дозволяють стежити за поточним часом, користувач комп'ютера завжди може дізнатися час, число, місяць, рік, скористатися програмами, які обмежать час використання комп'ютера для ігор дітьми. Комп'ютер може нагадати його господареві про необхідність вжити заздалегідь заплановані на певний час дії, включити в певний час електронну техніку, або вимкнути її і т.д.

Пам'ять комп'ютера побудована з двійкових запам'ятовуючих елементів - **бітів**, об'єднаних в групи по 8 бітів, які називаються **байтами**. (Одиниці виміру пам'яті збігаються з одиницями виміру інформації). Всі байти пронумеровані. Номер байта називається його адресою.

Байти можуть об'єднуватися в осередки, які називаються також **словами**. Для кожного комп'ютера характерна певна довжина слова - два, чотири або вісім байтів. Це не виключає використання елементів пам'яті іншої довжини (наприклад, півслова, подвійне слово). Як правило, в одному машинному слові може бути представлено або одне ціле число, або одна команда. Однак, допускаються змінні формати подання інформації. Розбиття пам'яті на слова для чотирьох байтового комп'ютерів представлено в таблиці:

Таблиця 1. Розбиття пам'яті на слова.

байт 0	байт 1	байт 2	байт 3	байт 4	байт 5	байт 6	байт 7
півслова		півслова		півслова		півслова	
СЛОВО				СЛОВО			
ПОДВІЙНЕ СЛОВО							

Сучасні комп'ютери мають багато різноманітних запам'ятовуючих пристроїв, які сильно відрізняються між собою за призначенням, тимчасовим характеристикам, обсягом інформації, що зберігається і вартості зберігання однакового обсягу інформації. Розрізняють два основних види пам'яті - **внутрішню і зовнішню**.

До складу внутрішньої пам'яті входять: **оперативна пам'ять, кеш-пам'ять і спеціальна пам'ять**.

2. Запам'ятовувальні пристрої.

За критерієм розташування щодо елементів архітектури комп'ютера можна виділити наступні види комп'ютерної пам'яті:

1. **Процесорна пам'ять** – це регістри команд 1-го рівня, вони розміщені на одному кристалі з процесором. РОП – частина центрального процесора (супероперативна (СОЗУ) або кеш-пам'ять), вона програє по місткості оперативної пам'яті, але перевершує її по швидкодії;

2. **Внутрішня пам'ять** – регістри розташовані на одній платі з процесором (кеш-пам'ять 2-го і наступних рівнів, а також основна пам'ять **ОП і ПЗП**); **ОП** або **ОЗП** забезпечують і читання, і запис, а **ПЗП** - тільки читання, тому що для запису потрібна особлива процедура - «прожиг»; **ПЗП** призначено для зберігання інформації, що підлягає обов'язковому збереженню при виключенні комп'ютера: стандартних програм, констант, таблиць символів і т.п.

3. **Зовнішня пам'ять** – ЗП великої ємності, розташовані окремо від основної плати, зв'язок з цими пристроями організовується за допомогою контролерів: магнітні і оптичні диски, твердотільна пам'ять, etc.

Перші два види пам'яті іноді об'єднують разом, називаючи внутрішньою пам'яттю.

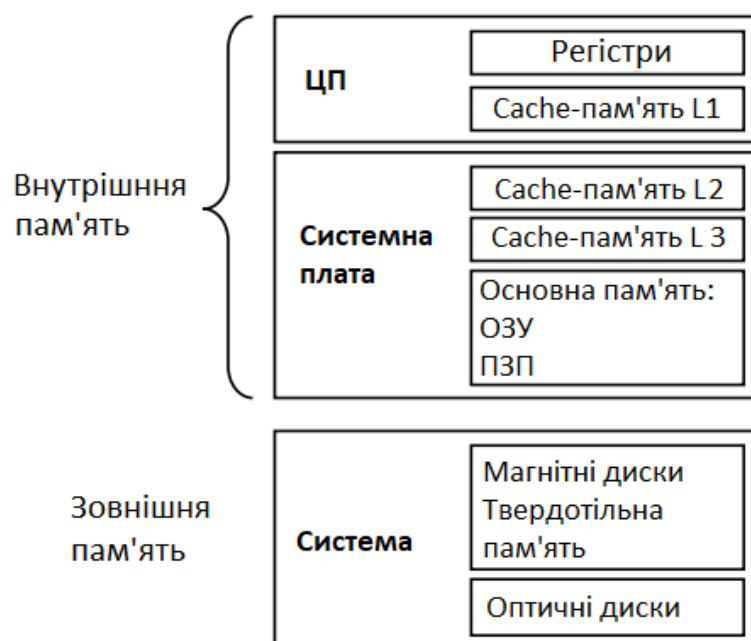


Рис. 1. Класифікація запам'ятовуючих пристроїв за критерієм їх положення щодо основних елементів архітектури комп'ютера.

Швидкодія пам'яті визначається наступними характеристиками:

1. **Час доступу** (інтервал від занесення адреси до занесення в пам'ять);
2. **Тривалість циклу** або період обертання. при довільному доступі це мінімальний час між двома послідовними зверненнями до пам'яті, визначається як час доступу плюс додатковий час, обумовлене загасанням сигналів на лініях або часом, необхідним для відновлення лічених даних;
3. **Швидкість передачі** (в пам'ять або з неї).

Розрізняють такі методи доступу до пам'яті:

1. **Послідовний** - для доступу до потрібного елементу необхідно прочитати всі попередні йому елементи (приклади: стрічковий магнітофон);
2. **Прямий** - кожен запис має свою адресу, який відображає її фізичне розміщення на носії, звернення до пам'яті здійснюється до початку запису з послідовним доступом до наступним одиницям (наприклад, магнітний або лазерний диск);
3. **Довільний** - кожна клітинка має унікальний фізичний адресу (наприклад, flash пам'ять) - пристрої, що запам'ятовують з довільною вибіркою (RAM - Random Access Memory);
4. **Асоціативний** - доступ не за адресою, а по вмісту – інакше називається контекстний доступ або контекстно-адресована пам'ять.

RAM (Random Access Memory) - «пам'ять з довільним доступом». Аббревіатурою RAM часто позначають основну пам'ять, звану також оперативним запам'ятовуючим пристроєм **ОЗП**.

ROM (Read-Only Memory) – «постійне запам'ятовуючий пристрій» (**ПЗП**), термін **ПЗП** стосується всіх видів незалежної пам'яті, включаючи і перезапису.

Виділяють наступні види **RAM** або **ОЗП**: статичну (**SRAM**), динамічну (**DRAM**), реєстрову (**RG**).

Більш детально:

SRAM (Static RAM) – статичний ОЗП. Енергозалежна пам'ять, регенерації не вимагає, але дорожча і менш ємна, ніж **DRAM**.

DRAM (Dynamic RAM) – пам'ять, яка вимагає постійного відновлення (регенерації) свого вмісту навіть при включеному живленні. Динамічний ОЗП або **ЗПДВ** («пристрій з довільною вибіркою»), останній термін застосовується зазвичай до динамічного різновиду **RAM**.

SDRAM – (Synchronous DRAM) «синхронна **DRAM**». Відрізняється від простої **DRAM** наявністю спеціального логічного блоку і дволанковою структурою. Всі операції запису / читання синхронізовані з основним тактовим сигналом.

VRAM (Video RAM) – «відео **RAM**». Спеціально розроблена для відеоадаптерів різновид **DRAM** з двопортовою організацією, допускає одночасне звернення до неї від двох різних пристроїв.

NRAM (Nano RAM) - незалежна пам'ять на нанотрубках.

FRAM, **FeRAM** (Ferroelectric RAM) – експериментальний різновид незалежної пам'яті на основі феро електричного принципу зберігання інформації.

MRAM (Magnetic RAM) – експериментальний різновид швидкісний незалежній пам'яті на основі магніторезистивного ефекту.

NVRAM (Nonvolatile RAM) – «незалежна **RAM**» (від латинського *volatilis* летучий). В принципі охоплює всі різновиди **EPROM** і **EEPROM** (в тому числі і **Flash**-пам'ять).

3. Види **ROM** або **ПЗП**:

OTPROM (One Time Programmable ROM) – «одноразово програмований ПЗП». Різновид **OTPROM** – «масочний ПЗП», він програмується («прошивається») виробником, користувач його перепрограмувати не може.

PROM (Programmable ROM) – «програмований ПЗП» (ППЗП) – іноді кажуть про одноразово програмованих користувачем ЗП.

EPROM (Erasable Programmable ROM) – «зі стиранням / програмована **ROM** - ПППЗП» Іноді вживається, як синонім **UV-EPROM**.

EEPROM (Electrically Erasable Programmable ROM) – **ЕСППЗП**.

UV-EPROM (Ultra-Violet EPROM) – «ультрафіолетова **EPROM**», **УФППЗП**. Історично це була перша комерційний різновид **EPROM**, операція стирання в цьому пристрої проводилася коротким опроміненням ультрафіолетом через спеціальне віконце - тому **flash**.

Flash memory – спочатку термін позначав новий різновид **EEPROM**, в якій читання / запис для прискорення процесу виробляються відразу цілими блоками. На сьогодні це фактично синонім **EEPROM**, термін позначає будь-які різновиди **ЕСППЗП**.

Необхідно згадати також програмовані логічні матриці і пристрої різних типів (**PLM, PML, PLA, PAL, PLD, FPGA** etc.), що включають в себе широкий вибір логічних елементів і пристроїв на одному кристалі.

Елемент пам'яті (**ЕП**) реалізується за допомогою тригера або транзистора з плаваючим затвором (flash-пам'ять). Осередок пам'яті (**ОП**) - це упорядкований набір ЕП.

Статична пам'ять (SRAM) зазвичай будується на D-тригерах - засувках.

Динамічна пам'ять (DRAM), що забезпечує велику місткість по у порівнянні з SRAM, будується на основі конденсаторів. Через те, що розмір конденсатора на інтегральній мікросхемі менше розміру D-тригера, при однакових геометричних розмірах обсяг динамічної пам'яті зазвичай вище, ніж місткість статичної. Разом з тим, схема **DRAM** складніше, тому що використовується мультиплексування адресних входів

і більш складна схема управління, що включає дві адресні лінії (старшу і молодшу), мультиплексори і демультиплексори, а також схему регенерації інформації, потреба у якій обумовлена тим, що конденсатори, на відміну від тригерів, з часом розряджаються.

Перепрограмувальне ПЗП (або репрограмувальне - РПЗП). РПЗП допускають багаторазовий перезапис інформації, в тому числі, перепрограмування, користувачем. Для цього при побудові РПЗП використовується відповідна елементна база, заснована на МОП транзисторах з "плаваючим затвором". У режимі запису для запису нуля на ЕП подається електричний імпульс. стирання інформації виробляється або УФ-випромінюванням (**EPROM**), або електрично (**EEPROM**). При стиранні все осередки переводяться в стан "1". термін зберігання інформації досягає декількох років.

FLASH-пам'ять будується на основі МДН-транзисторів. МДН-транзистор (Метал - Діелектрик - Напівпровідник) або МОН-транзистор (Метал - Оксид - Напівпровідник) – польовий транзистор з ізольованим затвором рис.2. Робота МОН-транзистора заснована на управлінні поверхневими властивостями напівпровідника шляхом зміни потенціалу затвора.

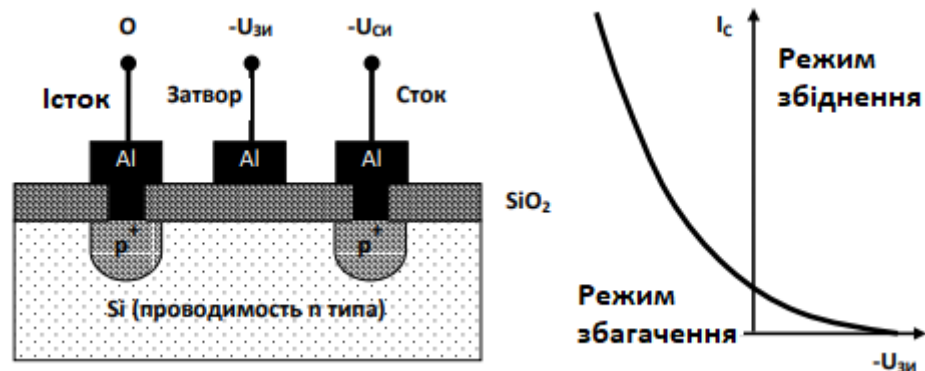


Рис. 2. Польовий транзистор з ізольованим затвором (зліва). Сток-затворна характеристика МДН-транзистору з вбудованим каналом (зправа).

Виділяють два типи МДН-транзисторів: з вбудованим каналом провідності між витоком і стоком, і з індукованим. В першому випадку при відсутності напруги на затворі між витоком і стоком існує область, яка містить носії заряду – це канал провідності. Зміна напруги на затворі дозволяє збільшувати або зменшувати число носіїв в цій області, тобто посилювати або послаблювати струм, аж до його повного припинення - замикання транзистора (рис. 2 зправа). У МДН-транзисторі з індукованим каналом навпаки - при відсутності струму на затворі між стоком і витоком носіїв недостатньо для протікання струму, подача напруги на затвор дозволяє збільшити число носіїв - відкрити транзистор. Для використання МДН-транзисторів в якості елементів пам'яті вони модифіковані – в схему введені додатковий ізольований, тобто оточений діелектриком, затвор. такий затвор називається плаваючим. При запису інформації затвор вводять або, навпаки, видаляють заряд на плаваючому затворі – для цього використовується тунельний ефект.

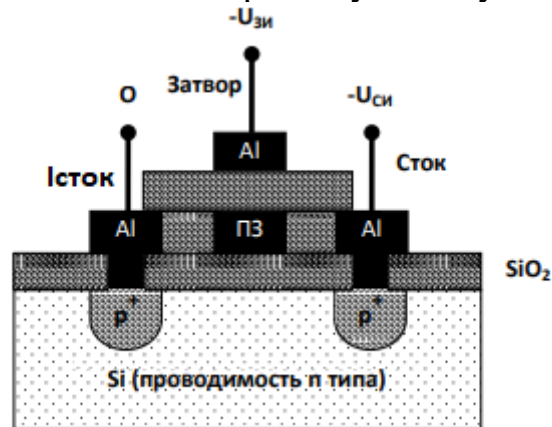


Рис. 3. МДН-транзистор з індукованим каналом.

Плаваючий затвор зберігає заряд, забезпечуючи необхідний режим: провідність відповідає зберіганню одиниці, а непровідність – нуля.

Тема 14. Інтерфейсні системи комп'ютера.

1. Системні, локальні і периферійні шини інтерфейсу, їх призначення та основні характеристики.

Інтерфейс (англ. *interface*):

- Сукупність засобів і правил, що забезпечують взаємодію комп'ютерів, периферійних пристроїв, пристроїв вводу/виводу та/або комп'ютерних програм.
- Сукупність описів і узгоджень щодо процедури передачі керування в підпрограму та повернення до вихідної програми.

Внутримашинний інтерфейс – система зв'язку і сполучення вузлів і блоків комп'ютера між собою. Являє собою сукупність мікросхем, ліній зв'язку, схем сполучення з компонентами комп'ютера, протоколів (алгоритмів) передачі і перетворення сигналів. Існують два варіанти організації внутримашинного інтерфейсу:

- багатозв'язний інтерфейс: кожен блок ПК з'єднаний з іншими блоками своїми локальними лініями зв'язку; багатозв'язна інтерфейс зазвичай застосовується в якості периферійного інтерфейсу, що доповнює системний, а іноді і в якості внутримашинного - як в деяких простих комп'ютерах, так і в ПК при використанні високошвидкісних послідовних інтерфейсів;
- однозв'язний інтерфейс: всі блоки ПК зв'язані один з одним через загальну або системну шину.

У більшості ПК в якості системного інтерфейсу використовується системна шина, але в перспективі очікуються використання багатозв'язного інтерфейсу на основі PCI Express.

Шина (bus) – сукупність ліній зв'язку, за якими інформація передається одночасно. Під **основною або системною шиною** зазвичай розуміється шина між процесором і підсистемою пам'яті. **Шини** характеризуються розрядністю і частотою. Найважливішими функціональними характеристиками системної шини є:

- кількість обслуговуваних нею пристроїв,
- пропускна здатність (максимально можлива швидкість передачі інформації).

Пропускна здатність шини залежить від її розрядності (існують шини 2-х, 4-х, 8-и, 16-и, 32-х і 64-розрядні) і тактової частоти, на якій шина працює.

Розрядність або ширина шини (bus width) – кількість каналів зв'язку в шині, тобто число бітів, яке може бути передано по шині одночасно.

Тактова частота шини (bus frequency) – частота, з якою передаються послідовні біти інформації по каналах зв'язку. В якості системної шини в різних ПК використовувались і можуть використовуватися:

- **шини розширень** – шини загального призначення, що дозволяють підключати велику кількість найрізноманітніших пристроїв;
- **локальні шини**, часто спеціалізуються на обслуговуванні

невеликої кількості пристроїв певного класу, переважно відеосистем.

У комп'ютерах широко використовуються і **периферійні шини** - інтерфейси для зовнішніх запам'ятовуючих і численних периферійних повільно діючих пристроїв.

2. Шини розширень.

Шина розширення ISA (Industry Standard Architecture – промислова стандартна архітектура) представлена в двох версіях: для IBM PC / XT (1981 рік) і для PC AT (1984 рік). Вона використовувалася в першому комп'ютері IBM PC, і тоді це була неофіційна назва шини IBM PC / XT, що дозволяла додавати в систему різні пристрої.

Шина PC/XT - 8-розрядна шина даних і 20-розрядна шина адреси, розрахована на тактову частоту 4,77 МГц; має 4 лінії для апаратних переривань і 4 канали для прямого доступу в пам'ять (канали DMA - Direct Memory Access). Шина адреси обмежувала адресний простір мікропроцесора величиною 1 Мбайт. Використовувалася з МП 8086, 8088.

Шина PC/AT -16-розрядна шина даних і 24-розрядна шина адреси, робоча тактова частота до 8 МГц, але може застосовуватися і МП з тактовою частотою 16 МГц, так як контролер шини здатний ділити частоту навпіл; має 7 ліній для апаратних переривань і 4 канали DMA. Використовувалася з МП 80286, сумісна і з МП з тактовою частотою більше 66 МГц (коефіцієнт ділення збільшений); збільшено кількість ліній апаратних переривань з 4 до 15 і каналів прямого доступу до пам'яті (DMA) з 4 до 7. Завдяки 24-розрядній шини адреси адресний простір збільшився з 1 Мбайт до 16 Мбайт. Теоретична пропускна здатність шини даних дорівнює 16 Мбайт/с, але реально вона близько 5,5 Мбайт/с, зважаючи на низку особливостей її використання.



Рис. 1. Конфігурація системи з шиною ISA.

ISA – основна шина на застарілих материнських платах. З появою 32-розрядних високошвидкісних МП шина ISA стала суттєвою перешкодою збільшення швидкодії ПК. Раніше за допомогою інтерфейсу ISA підключалися такі пристрої, як відеокарти, модеми, звукові карти і т. д. На сучасних материнських платах цей інтерфейс або зовсім відсутній, або є всього 1-2

слоти. Конструктивно слот ISA являє собою роз'єм, що складається з двох частин – 62-контактного і примикаючого до нього (в PC AT) 36-контактного сегменту.

Шина EISA (Extended ISA) – 32-розрядна шина даних і 32-розрядна шина адреси, створена в 1989 році як функціональне і конструктивне розширення ISA. Адресний простір шини 4 Гбайт, працює на частоті 8-10 МГц.

Теоретична пропускна здатність шини – 33 Мбайт/с, причому швидкість обміну по каналу МП – кеш – ОП визначається параметрами мікросхем пам'яті; збільшено число роз'ємів розширень - теоретично може підключатися до 15 пристроїв (практично до 10). Покращена система переривань, підтримується Bus Mastering – режим одноосібного управління шиною з боку будь-якого з пристроїв на шині, має систему арбітражу для управління доступом пристроїв до шини. Забезпечується автоматичне конфігурування системи і управління DMA. Шина підтримує багатопроцесорну архітектуру обчислювальних систем. **Шина EISA** надто дорога і застосовується в швидкісних ПК, мережевих серверах і робочих станціях. Зовні слоти шини на СП мають такий же вигляд, як і ISA, і в них можуть вставлятися плати ISA, але в глибині роз'єму знаходяться додаткові ряди контактів EISA, а плати EISA мають більш високу частину роз'єму з додатковими рядами контактів.

Шина MCA (Micro Channel Architecture) – 32-розрядна шина, створена фірмою IBM в 1987 році для машин PS/2, пропускна здатність 76 Мбайт/с, робоча частота 10-20 МГц. За своїм іншим характеристикам близька до шини EISA, але не сумісна ні з ISA, ні з EISA. Оскільки комп'ютери PS/2 не набули широкого поширення, в першу чергу через відсутність кількості прикладних програм, шина MCA також використовується не дуже широко. Другою важливою причиною відсутності попиту на MCA є несумісність плат адаптерів ISA з MCA. І не останню роль зіграв той факт, що IBM зажадала від всіх виробників, охочих придбати права на використання цієї шини, заплатити за використання ISA в усіх випущених раніше комп'ютерах. Власне кажучи, це і спричинило за собою розробку альтернативи - EISA.

3. Лока

4. льні шини.

Сучасні обчислювальні системи характеризуються:

- стрімким зростанням швидкодії мікропроцесорів і деяких зовнішніх пристроїв (так, для відображення цифрового повноекранного відео з високою якістю необхідна пропускна здатність 22 Мбайт/с);
- появою програм, що вимагають виконання великої кількості інтерфейсних операцій (наприклад програми обробки графіки в Windows, мультимедіа).

У цих умовах пропускної здатності шин розширення, що обслуговують одночасно кілька пристроїв, виявилось недостатньо для комфортної роботи користувачів, оскільки комп'ютери стали довгий час «думати». Розробники

інтерфейсів пішли по шляху створення локальних шин, що підключаються безпосередньо до шини МП, що працюють на тактовій частоті МП (але не на внутрішній робочій його частоті) і забезпечують зв'язок з деякими швидкісними зовнішніми по відношенню до МП пристроями: основний і зовнішньою пам'яттю, відеосистемами і т.д.

Зараз існують три основні стандарти універсальних локальних шин: **VLB, PCI і AGP.**

Шина VLB (VL-bus, VESA Local Bus) представлена в 1992 році асоціацією стандартів відеоелектроніки (VESA - торгова марка Video Electronics Standards Association) і тому часто її називають шиною VESA. Шина VLB, по суті, є розширенням внутрішньої шини МП для зв'язку з відеоадаптером і рідше - з жорстким диском, платами мультимедіа, мережним адаптером. Розрядність шини для даних - 32 біта, для адреси - 30, реальна швидкість передачі даних по VLB - 80 Мбайт/с, теоретично досяжна - 132 Мбайт/с (у версії 2 - 400 Мбайт/с).

Недоліки шини VLB:

- орієнтація тільки на МП 80386, 80486 (не адаптована для процесорів класу Pentium);
- жорстка залежність від тактової частоти МП (кожна шина VLB розрахована тільки на конкретну частоту до 33 МГц);
- мала кількість пристроїв, що підключаються до шини VLB, може підключатися тільки 4 пристрої;
- відсутня арбітраж шини - можуть бути конфлікти між пристроями.

Інтерфейси PCI.

Шина **PCI** (Peripheral Component Interconnect, з'єднання зовнішніх компонентів) - найпоширеніший і універсальний інтерфейс для підключення різних пристроїв.

Розроблена в 1991 році фірмою Intel з метою створити шину, здатну замінити всі існуючі, часто несумісні шинні інтерфейси, такі як ISA, EISA, MCA, VLB. Шина PCI є більш універсальною, ніж VLB; допускає підключення до 10 пристроїв; має свій адаптер, що дозволяє їй налаштовуватися на роботу з будь-яким МП від 80486 до сучасних Pentium. Тактова частота PCI - 33 МГц, розрядність - 32 розряду для даних і 32 розряду для адреси з можливістю розширення до 64 бітів, теоретична пропускна здатність 132 Мбайт / с, а в 64-бітовому варіанті - 264 Мбайт/с. Модифікація 2.1 локальної шини **PCI** працює на тактовій частоті до 66 МГц і при розрядності 64 має пропускну здатність до 528 Мбайт/с. Здійснено підтримка режимів Plug and Play, Bus Mastering і автоконфігурування адаптерів.

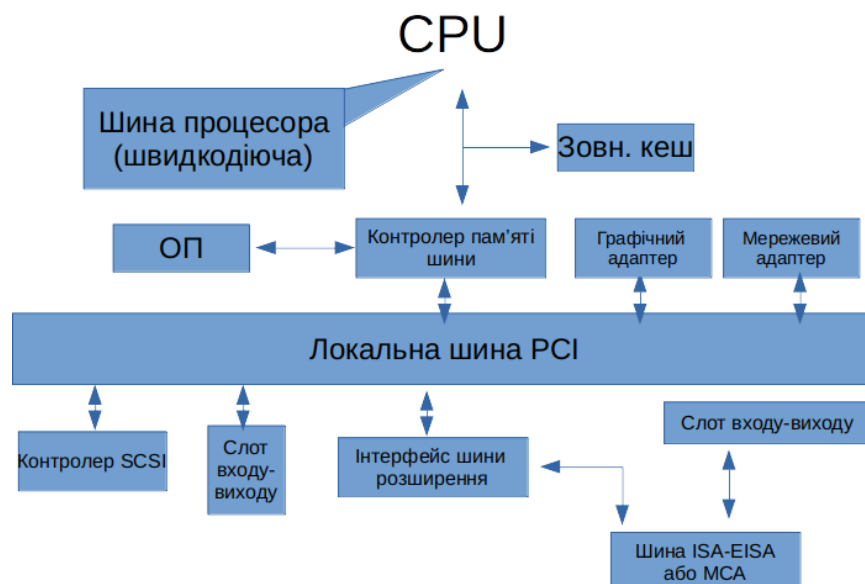


Рис. 2. Локальна шина PCI.

Конструктивно роз'єм шини на системній платі складається з двох наступних посліп секцій по 64 контакту (кожна зі своїм ключем). За допомогою цього інтерфейсу до материнської плати підключаються відеокарти, звукові карти, модеми, контролери SCSI і інші пристрої.

Інтерфейс AGP.

Шина **AGP** (Accelerated Graphics Port - прискорений графічний порт) - інтерфейс для підключення відеоадаптера до окремої магістралі **AGP**, що має вихід безпосередньо на системну пам'ять. Розроблена шина на основі стандарту PCI v2.1. Шина може працювати з частотою системної шини до 133 МГц і забезпечує найвищу швидкість передачі графічних даних.

Шина AGP має два режими роботи: **DMA** і **Execute**. У режимі **DMA** основною є пам'ять відеокарти. Графічні об'єкти зберігаються в системній пам'яті (ОЗУ), але перед використанням копіюються в локальну пам'ять відеокарти. Обмін ведеться великими послідовними пакетами. У режимі **Execute** системна пам'ять і локальна пам'ять відеокарти логічно рівноправні. Графічні об'єкти не копіюються в локальну пам'ять, а вибираються безпосередньо з системної.

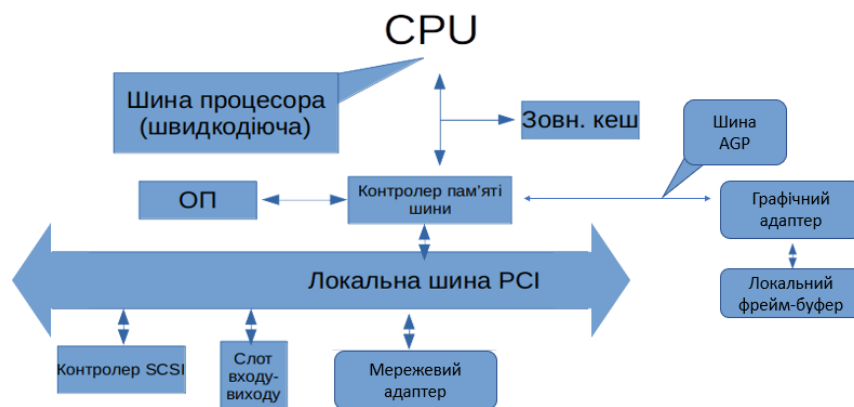


Рис. 3. Локальна шина PCI.

5. Периферійні шини.

Периферійні шини набагато більш різноманітні.

Інтерфейс АТА отримав назву Integrated Drive Electronics (IDE), запропонований в 1988 році користувачам ПК IBM PC AT. Він обмежує місткість одного накопичувача 504 Мбайт (через організації адресного простору в традиційній BIOS «головка-циліндр-сектор»: 16 головок $\times$ 1024 циліндра $\times$ 63 сектора $\times$ 512 байтів в секторі = 504 Мбайт = 528 482 304 байтів) і забезпечує теоретичну швидкість передачі даних 5-10 Мбайт/с.

Периферійні шини **IDE** (Integrated Drive Electronics), АТА (AT Attachment - підключається до АТ), EIDE (Enhanced IDE), SCSI (Small Computer System Interface) використовуються найчастіше в якості інтерфейсу тільки для зовнішніх запам'ятовуючих пристроїв.

Існує багато модифікацій і розширень інтерфейсів АТА/IDE. Є інтерфейси АТА з різними номерами, Fast АТА (теж з номерами), Ultra АТА (і їх кілька) і, нарешті, EIDE. Є також IDE-інтерфейси, що підтримують протоколи АТАPI, DMA і т. Д. Багато з наведених назв офіційно не затверджені, є торговими марками, але тим не менш в літературі зустрічаються часто. Така масовість назв пов'язана з тим, що в даний час більше 90% всіх використовуваних в персональних комп'ютерах дискових інтерфейсів відносяться до категорії IDE.

Коротко розглянемо деякі модифікації.

Fast **АТА-2** або Enhanced IDE (EIDE - розширений IDE), який використовує як традиційну (але розширену) адресацію за номерами головки, циліндра і сектора, так і адресацію логічних блоків (Logic Block Address - LBA), підтримує ємність диска до 2500 Мбайт і швидкість обміну до 16,7 Мбайт/с. До адаптера EIDE, що підтримує стандарт АТАPI, може підключатися до чотирьох накопичувачів, в тому числі і CD-ROM.

АТАPI (ATA Package Interface) – стандарт, створений для того, щоб безпосередньо підключати до інтерфейсу АТА не тільки жорсткі диски, але і дисководи, стримери, сканери і т.д. Версії інтерфейсу АТА-3 і Ultra АТА обслуговують диски більшої ємності, мають швидкість обміну до 33 Мб/с, підтримують технологію SMART (Self Monitoring Analysis and Report Technology – технологію самостійного стеження, аналізу і звіту), що дозволяє пристроям повідомляти про свої несправності, і ряд інших сервісів. Сучасні версії інтерфейсу АТА/АТАPI-5, АТА/АТАPI-6 по протоколам UDMA/66 і UDMA/100 забезпечують пікову пропускну здатність 66 і 100 Мб/с, відповідно.

UDMA (Ultra Direct Memory Access) – режим прямого доступу до пам'яті. Звичайний метод обміну з жорстким диском IDE – це програмний вхід-вихід (Programmed Input/Output), при якому процесор, використовуючи команди введення-виведення, зчитує або записує дані в буфер жорсткого диску, що забирає якусь частину процесорного часу. Введення-виведення шляхом прямого доступу до пам'яті йде під керуванням самого жорсткого

диску або його контролера в паузах між зверненнями процесора до пам'яті, що економить процесорний час, але трохи знижує максимальну швидкість обміну.

На материнських платах реалізовані два канали **IDE**, до кожного з яких можливе підключення до двох пристроїв.

SCSI (Small Computer System Interface) є більш складним і потужним інтерфейсом і широко використовується в трьох версіях: SCSI-1, SCSI-2 і SCSI-3. Це універсальні периферійні інтерфейси для будь-яких класів зовнішніх пристроїв. Фактично SCSI є спрощеним варіантом системної шини комп'ютера, що підтримує до восьми пристроїв. Така організація вимагає від пристроїв наявності певних контролерів - наприклад, в жорстких дисках SCSI всі функції кодування-декодування, пошуку сектора, корекції помилок і т.п. Покладаються на вбудовану електроніку, а зовнішній SCSI-контролер виконує функції обміну даними між пристроєм і комп'ютером - часто в автономному режимі, без участі центрального процесора (режими DMA - прямого доступу до пам'яті, або Bus Mastering - «загарбника», головного абоненташіни). Інтерфейси SCSI-1 мають 8-бітову шину; SCSI-2 і SCSI-3 - 16- або 32-бітову і розраховані на використання в потужних машинах-серверах і робочих станціях. Існує багато різних специфікацій даного інтерфейсу, що відрізняються пікової пропускною спроможністю, максимальним числом пристроїв, що підключаються, граничної довжиною кабелю. Так, максимальна пропускна здатність може досягати 80 і навіть 160 Мбайт / с. В інтерфейс SCSI: Plug and Play додані кошти підтримки технології Plug and Play - автоматичне впізнання типу та функціонального призначення пристроїв, настройка без допомоги користувача або при мінімальному його участі, можливість заміни пристроїв під час роботи і т. П. Все SCSI-пристрої управляються спеціальним SCSI-контролером, реалізованим частіше у вигляді окремої плати розширення, яка встановлюється в вільний роз'єм на материнській платі. Однак випускаються і материнські плати з вбудованими контролерами SCSI.

RS-232 - інтерфейс обміну даними по послідовному комунікаційному порту (COM-порту). Управління роботою COM-портів (число яких обмежено чотирма) здійснюється спеціальною мікросхемою UART16550A, розташованої на материнській платі. Фізично роз'єм COM-порту може бути 25- або 9-контактним. За допомогою даного інтерфейсу здійснюється робота і підключення таких пристроїв, як зовнішній модем, мишка і т. д.

IEEE 1284 {Institute of Electrical and Electronic Engineers 1284 - стандарт Інституту інженерів з електротехніки та електроніки 1284) - стандарт, що описує специфікації паралельних швидкісних інтерфейсів SPP (Standard Parallel Port - стандартний паралельний порт), EPP (Enhanced Parallel Port - покращений паралельний порт), ECP (Extended Capabilities Port - порт з розширеними можливостями), як правило, використовуються для підключення через паралельні порти комп'ютера (LPT-порти) таких пристроїв, як принтери, зовнішні пристрої, що запам'ятовують, сканери, цифрові камери. З боку LPT-порту встановлено стандартний роз'єм DB-25 (25 контактів), а з

боку пристрою використовується роз'єм типу Centronics. Контролер паралельного порту розміщений на материнській платі.

Таблиця 1. Основні характеристики шин:

Характеристика	Шина					
	ISA	EISA	MCA	VLB	PCI	AGP
Розрядність шини	16/24	32/32	32/32	32/32	32/32	32/32
для даних і адреси (біт)				64/64	64/64	64/64
Робоча частота (МГц)	8	8-33	10-20	До 33	До 66	66/133
Пропускна здатність(Мбайт/с)	16	33	76	132	132/264 /528	528/1056 /2112
Число підключень	6	15	15	4	10	1

6. Універсальні послідовні інтерфейси.

Послідовні інтерфейси використовують замість широченних багатожильних (до 64 жил) шлейфів і кабелів 2-8-жильні. Вони зручніше паралельних і, як це не парадоксально, істотно більш швидкісні. Пропускна здатність послідовних інтерфейсів збільшується через з'єднань з пристроями на кшталт «точка-точка» (багатозв'язна інтерфейс) замість загальної шини і зменшення паразитних індуктивностей і ємностей проводів, а отже, і можливості роботи на більш високих робочих частотах. Так, робочі частоти паралельних інтерфейсів лежать в межах десятків - кількох сотень МГц, а послідовних - до десятка ГГц (наприклад, послідовний інтерфейс PCI Express має робочу частоту 2,5 ГГц). Першими на послідовні інтерфейси перебралися клавіатури, миші, модеми, принтери і сканери, а з 2003 року ця тенденція спостерігається і для інших зовнішніх пристроїв, включаючи дискову пам'ять (інтерфейси USB, SATA, SAS) і відеосистему (інтерфейс PCI Express). Є спроби перекладу на ці інтерфейси і системи оперативної пам'яті (технологія Rambus).

Основні переваги послідовних інтерфейсів:

- велика гнучкість і функціональність шин;
- зручність налагодження і використання з огляду на перенесення «центру ваги» виконання цих технологій на мікросхеми;
- висока пропускна здатність, з огляду на багатозв'язна і зниження паразитних індуктивностей і ємностей в лініях зв'язку і відсутності складних процедур синхронізації;
- мініатюризація і зниження вартості монтажу, скорочення кількості контактів, проводів, екранів;
- можливість гарячого підключення пристроїв, тобто динамічного налаштування та її масштабування;
- полегшення арбітражу шин і організації переривань;

- найкраща перешкодозахищеність і надійність роботи.

Сімейство послідовних інтерфейсів PCI Express.

Мабуть, найбільш перспективно і представляє істотний інтерес сімейство послідовних інтерфейсів **PCI Express**, інформація про базову протоколі якого з'явилася в липні 2002 року. **PCI Express** використовує сукупність незалежних послідовних каналів передачі даних. Оскільки при передачі використовується перешкодозахищеність кодування кожен байт видається 10 бітами. Пропускна здатність одного каналу 200 Мб/с. Ліцензовані 1-о, 2-х, 4-х, 8-й, 16-й і 32-канальні версії (до 16 Гб/с). У режимі дуплексної передачі всі ці цифри пропускної здатності подвоюються.

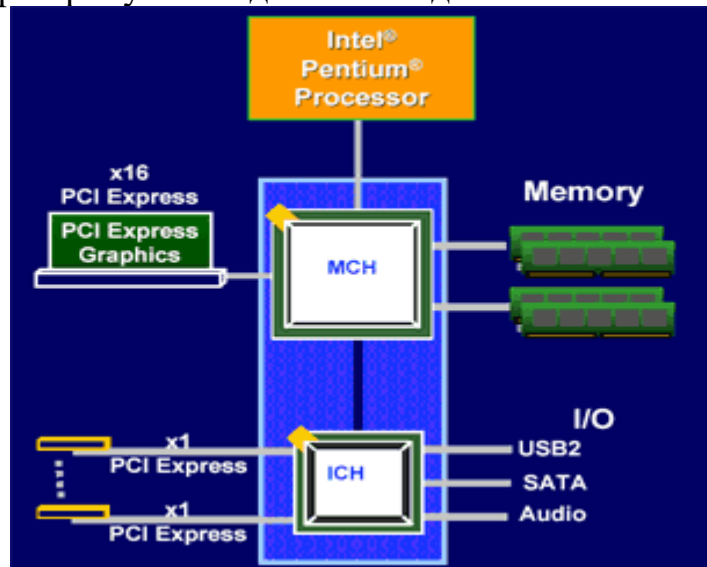


Рис.4. Сімейство послідовних інтерфейсів

7. Бездротові інтерфейси: Wi-Fi, WiMax, WiBro.

Сімейство інтерфейсів Wi-Fi.

Інтерфейси **Wi-Fi** відносяться до групи інтерфейсів, що забезпечують бездротовий доступ комп'ютерів до мереж. Базовий стандарт IEEE 802.11 або WiFi (Wireless Fidelity - «беззаперечна відданість») був розроблений в 1997 році і забезпечував передачу даних по радіоканалу 2,4 ГГц зі швидкістю до 250 Кбіт / с. Пізніше з'явилися версії IEEE 802.11a, IEEE 802.11b, IEEE 802.11g підвищили швидкість передачі до 7-56 Мбіт / с, а в неліцензованих режимі IEEE 802.11g turbo і до 108 Мбіт / с. Найбільша швидкість 320 Мбіт / с забезпечується в стандарті IEEE 802.11n.

Швидкість передачі залежить від дальності і інтенсивності перешкод. Максимальна дальність приблизно 100 м, в межах прямої видимості можливо і деяке збільшення дальності.

Для роботи з інтерфейсом **Wi-Fi** необхідні комп'ютер, укомплектований своєю мобільною точкою доступу - бездротової мережевої інтерфейсної картою NIC (Network Interface Card), і локальна мережа з точками доступу AP (Access Point), які будучи стаціонарними пристроями, виконують роль моста між бездротовим й провідним сегментами цієї мережі.

Сімейство інтерфейсів WiMax.

Бездротова технологія **WiMax** (Wireless for Microwave Access) - це комерційна назва стандарту IEEE 802.16a, заявленого в січні 2003 року. Це третя версія стандарту IEEE 802.16, вперше запропонованого в грудні 2001 року: цей стандарт працював в смузі частот 10-66 ГГц і забезпечував передачу даних зі швидкістю 134 Мбіт / с тільки в межах прямої видимості. Дальність зв'язку була до 5 км. У грудні 2002 року з'явився стандарт IEEE 802.16с, що відрізняється від попередника лише універсальністю обладнання.

Інтерфейс WiBro.

У Корейському інституті електронних і комунікаційних розробок спільно з компанією Samsung Electronics в 2002 році представлений проект мобільного радіоінтерфейсу **Wireless Broadband** (WiBro), який в 2005 році офіційно визнали багато компаній, в тому числі і компанія Intel. WiBro за багатьма параметрами перевершує мобільні інтерфейси WiMax. WiBro забезпечує надійний роумінг, і на швидкості пересування мобільного клієнта 120 км / год підтримує пропускну здатність 1 Мбіт/с.

8. Прикладні програмні інтерфейси

Для ефективної роботи з відео- і аудіоінформацією розроблені прикладні програмні інтерфейси API (Application Program Interface):

- Direct X - для роботи з графікою;
- Open GL - для роботи з 3D графікою;
- **Open ML** - для роботи з мультимедіа (відео-та аудіоінформацією, в тому числі і з 3D графікою).

API Open ML істотно полегшує роботу прикладним програмістам в таких областях як:

- кодування і декодування мультимедіа потоків за допомогою різноманітних ефективних кодеків;
- підтримка асинхронної передачі інформації між прикладною програмою і пристроями мультимедіа;
- управління пристроями мультимедіа і забезпечення їх сумісності шляхом використання спеціальних драйверів,
- створення гнучкого механізму буферизації, що забезпечує плавне відтворення цифрової мультимедіа інформації;
- обробка цифрових аудіо / відео потоків в реальному часі;
- відтворення і запис цифрового мультимедіа інформації;
- підтримка одночасної роботи з двовірною і тривимірною графікою, в тому числі і узгодження зі стандартами Open GL;
- ефективна розробка текстур в тривимірній графіці.

Тема 15. Обчислювальні системи.

1. Поняття системи.

Термін «система» в даний час використовується вкрай широко, але його популярність має і зворотну сторону - досить часто передбачувана інтуїтивна зрозумілість терміну має наслідком різний його розуміння співрозмовниками і, як результат, взаємне нерозуміння. Наведемо кілька найпоширеніших визначень системи:

- щось, більше, ніж просто сума її елементів.
- сукупність взаємопов'язаних сутностей, сприймається як єдине ціле.
- множина взаємопов'язаних елементів, відокремлених від середовища і які взаємодіють з нею, як єдине ціле.
- кінцева множина функціональних елементів і відносин між ними, виділена з середовища відповідно до визначеної мети в рамках певного часового інтервалу.
- відображення в свідомості суб'єкта властивостей об'єктів і їх відносин у вирішенні задачі дослідження, пізнання.

Неважко бачити, що ці визначення відрізняються розпливчастістю і неконструктивна в тому плані, що їх складно використовувати для побудови системи. Спробуємо визначити систему через її атрибути, тобто невід'ємні властивості.

Почнемо з **функціональних атрибутів**:

- синергетичність (справа, праця, робота);
- емерджентність (від emergent-виникає, несподівано з'являється);
- цілеспрямованість або телеологічності (наявність цільової функції);
- альтернативність (вибір шляхів досягнення мети і самоорганізації для досягнення мети як обмеження числа ступенів свободи).

У якості структурних атрибутів відзначимо два взаємопов'язаних:

- структурованість;
- ієрархічність.

Структурованість означає наявність зв'язків між елементами системи – її внутрішню зв'язність або взаємозалежність елементів системи.

Необхідно звернути увагу на те, що ієрархічність є не просто властивістю, але саме атрибутом, тобто невід'ємною властивістю системи. Це суперечить домінуючій в сучасній масовій свідомості думці про непотрібність ієрархії і загальну рівність, але математика стверджує, що стійкі тільки системи, що мають ієрархічну організацію, тобто не тільки горизонтальні зв'язки, але і вертикальні.

Оскільки нас цікавлять комп'ютерні системи, тобто системи, призначені для роботи з інформацією, то необхідно згадати закон Ешбі (*William Ross Ashby* 06.09.1903 - 15.11.1972), який говорить:

«Різноманітність (ентропію) системи можна знизити не більше ніж на величину кількості інформації в керуючій системі про керовану, що

дорівнює різноманітності (ентропії) управління за вирахуванням втрати інформації від неоднозначного управління».

Нехай X - керована система, тобто множина її станів x , $x \in X$, а U - множина управлінь, де $u \in U$ – управління.

Визначення 1: Управлінням називається відображення

$$u : x \rightarrow y \in Y \subseteq X, \quad (1)$$

де x і y - стани системи до і після управління. Введемо ентропію H .

Якщо має місце

$$H(y) \geq H(x),$$

то система некерована. Таким чином ціль управління визначається як зниження ентропії керованої системи

$$H(y) < H(x),$$

Стан системи після управління

$$H(y) \geq H(x) - I(u, x) \quad (2)$$

де $I(u, x) = H(u) - H(u | x)$ – кількість інформації (використана ентропійна міра інформації) в керуючій системі про керовану. Таким чином, для того, щоб керувати, треба знати більше, ніж об'єкт управління! Компетентність – необхідна умова, сам по собі адміністративний статус не забезпечує ефективність управління. Більш того, управління з боку некомпетентної особи неефективно.

2. Паралелізм обчислень та шляхи його досягнення. Закон Амдала.

Поняття обчислювальної системи нерозривно пов'язане з проблемою вирішення складних обчислювальних задач, що вимагають підвищення обчислювальної потужності. Один з основних методів – перехід до **паралельних обчислень**, під якими розуміються процеси обробки даних, що характеризуються одночасним виконанням декількох операцій. Можливий паралелізм на рівні програм і на рівні пристроїв.

В цілому, можна виділити наступні режими виконання незалежних частин програми:

1. багатозадачність або режим поділу часу – тут використовується один процесор, тобто цей режим - псевдопаралельний, тому що виконується тільки одна частина програми, а решта чекають своєї черги; в цьому режимі проявляються ефекти «справжніх» паралельних обчислень;

2. паралельне виконання як на кількох процесорах, так і за допомогою конвеєрних і векторних процесорів;

3. розподілені обчислення на декількох процесорах або комп'ютерах, об'єднаних в мережу – цей режим пов'язаний з втратами часу на передачу даних, тому режим ефективний для паралельних алгоритмів з низькими вимогами до міжпроцесорного (міжкомп'ютерного) обміну даними.

Інтуїтивно очевидно, що паралелізм на рівні пристроїв може бути забезпечений наступними методами:

1. незалежністю функціонування окремих вузлів і елементів обчислювача (або системи);

2. надмірністю елементів обчислювача.

Останнє може бути досягнуто як їх дублюванням, так і використанням спеціалізованих для вирішення окремих завдань пристроїв (співпроцесорів).

Закон Амдала (Gene Amdahl, 1967) дає оцінку прискорення обчислень при використанні n процесорів. Нехай T_s - час виконання завдання на однопроцесорному пристрої, тоді вираз у часі вирішення завдання на пристрої з n процесорів може бути оцінений наступним чином:

$$S = \frac{T_s}{T_{pn}} = \frac{n}{1 + (n - 1) * f}, \quad (3)$$

де f – частка операцій, які повинні виконуватися послідовно одним з процесорів. Можна бачити, залежність (3) з ростом параметра f має тенденцію до насичення:

$$\lim_{n \rightarrow \infty} S = \frac{1}{f}$$

Таким чином, закон Амдала формулює фундаментальне обмеження на зростання продуктивності при розпаралелюванні обчислень. Разом з тим, оцінка (3), як було дещо пізніше (1988) показано Густафсоном і Барсісом, надмірно песимістична.

Закон Густафсона-Барсіса. При виведенні закону Амдала передбачалося, що зростання n не супроводжується зростанням обсягу завдання. Але в реальності більш потужні системи купуються для вирішення більш складних завдань або, якщо на новій системі автоматично вирішується стара задача, то користувач прагне покращити якість результату, наприклад, підвищенням точності. Це веде до зростання об'єму обчислень, при цьому зростання об'єму обчислень часто супроводжується зниженням частки послідовних операцій за рахунок зростання паралельних. Наприклад, в деяких завданнях формула, яка обчислюється, дозволяє розбити великий масив даних N на підмасиви N / n , для них паралельно виконати деякі операції, а вже потім їх результати піддати тій же операції. Таким чином може бути отримана економія.

Цей факт дозволив дещо покращити оцінку ефективності розпаралелюванню, дану законом Амдала (3), і сформулювати **закон масштабованого прискорення**.

Нехай t_s і t_p - час виконання послідовної і паралельної частин обчислень, тоді частка послідовно виконуваних розрахунків

$$g(n) = \frac{t_s}{t_s + t_p / n},$$

Звідси можна отримати оцінку досяжного при розпаралелюванні прискорення обчислень, відому як закон Густафсона - Барсіса:

$$S = \frac{T_s}{T_{pn}} = \frac{t_s + t_p}{t_s + t_p / n} = \frac{(t_s + t_p / n) (g(n) + n(1 - g(n)))}{t_s + t_p / n} = n + (1 - n) g(n) \quad .(4)$$

Оскільки закон Густафсона-Барсіса дає оцінку ефективності організації паралельних обчислень при збільшенні складності вирішуваних завдань (а не просто їх об'єму!), То він часто називається законом масштабування прискорення. Виграш, що дається законом Густафсона-Барсіса (4) щодо закону Амдала (3), оцінює творчу роль людини у вирішенні завдання – виграш досягається не «тупою силою» комп'ютера за принципом «комп'ютер є – розуму не треба», а організацією процесу розв'язку задачі, тобто саме розумом людини, формулює спосіб обчислювального рішення задачі - алгоритм і програму.

3. Класифікація обчислювальних систем. Систематика Флінна.

Для класифікації обчислювачів в контексті паралелізму обчислень часто використовується класична систематика (або класифікація) Флінна (M.J.Flynn, 1966, 1972), заснована на аналізі способів взаємодії потоків. Тому почнемо з розгляду концепції потоків.

У сучасних комп'ютерах використовується багатозадачний режим виконання програми, тобто робота процесора організована так, що він поперемінно виконує кілька програм. Для цього необхідно визначити одиниці роботи процесора і інших вузлів обчислювача, між якими і буде ділитися його ресурс (час). На сьогодні основний одиницею роботи виступає процес (іноді термін задача). Кілька процесів можуть бути об'єднані в завдання. Процеси можуть включати в себе кілька потоків.

Процес – це виконувана програма (в стадії виконання). Процес може перебувати в наступних станах:

Породження – готуються умови для першого виконання на процесорі;

Активний стан – програма виконується;

Очікування – програма не виконується на процесорі через зайнятості якогось ресурсу

Готовність – програма не виконується, але для її виконання представлені всі ресурси, крім часу процесора

Закінчення – нормальне або аварійне завершення програми

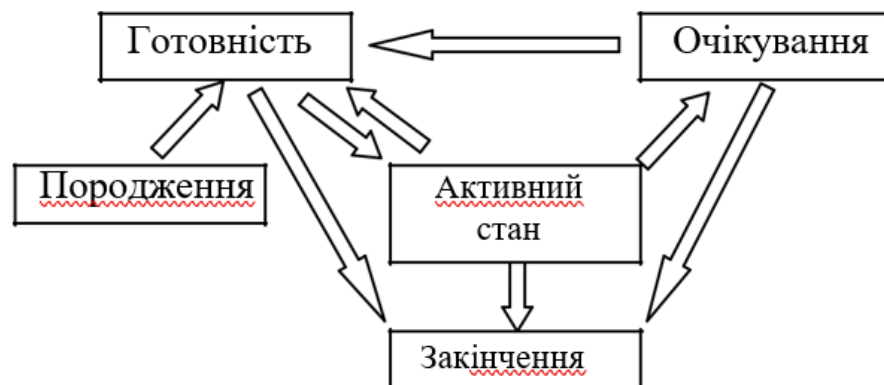


Рис. 1. Стани процесу та їх зв'язки.

Процеси обмінюються між собою даними по односпрямованим лініям комутації – **каналах**. Канали діляться на:

Синхронні – повідомлення може бути передано тільки після отримання підтвердження про прийом попереднього – приймаючий процес не виконується, поки не отримає дані, а передавальний – поки не отримає підтвердження про прийом даних.

Асинхронно/синхронні – повідомлення копіюється в буфер і пересилається звідти одночасно з роботою передавального процесу. Прийом повідомлень синхронний – процес блокується до моменту прийому.

Асинхронні – і передача і прийом завершуються відразу.

Можлива проблема «Deadlock», суть в тому, що один процес не може передати повідомлення іншому, тому що той чекає від нього підтвердження. В цьому випадку застосовується **семафор** – захищена змінна, значення якої можна опитувати і міняти тільки за допомогою спеціальних операцій **wait** і **signal** і операції ініціалізації **init**. Двійкові семафори можуть приймати тільки значення 0 і 1. Семафори з лічильниками можуть приймати невід'ємні цілі значення.

Процеси спілкуються між собою через ОС, споживають ресурси, так як:

- процеси вирішують загальну задачу;
- процеси працюють з одними і тими ж даними;
- процеси використовують одне і теж саме право доступу до ресурсів.

ОС розглядає такі пов'язані процеси як звичайні і ізолює їх один від одного. В результаті, так як на створення кожного такого процесу ОС витрачає системні ресурси, виникає проблема неефективного їх використання: у кожного процесу своя фізична пам'ять, свої пристрої введення-виведення, свій віртуальний простір.

Вирішення цієї проблеми можливе в рамках багатопотокової обробки, в основу якої покладено облік тісних зв'язків між окремими гілками обчислення одного і того ж завдання.

Багатопотокова обробка – це не багатопрограмна обробка!

Виділяють наступні потоки (або послідовності):

- потоки управління;
- потоки даних;
- потоки запитів.

Потік управління: чергова команда отримує можливість виконання відразу після того як їй передано управління. Таким чином, потік управління – це послідовність виконання команд в ході виконання програми. Застосовується потік управління у випадках, коли можливе одночасне (паралельне) виконання ряду операцій.

Особливості потоку управління:

- Дані розсилаються між командами не безпосередньо, а через осередки загальною пам'яті. Звернення проводиться до загальної пам'яті.

- Операнди можуть перебувати і в самих командах для поліпшення доступу до них.
- Потік управління за своєю суттю є послідовним, паралельне виконання команд організовується ввідними в нього спеціальними керуючими операндами, що забезпечують переходи.

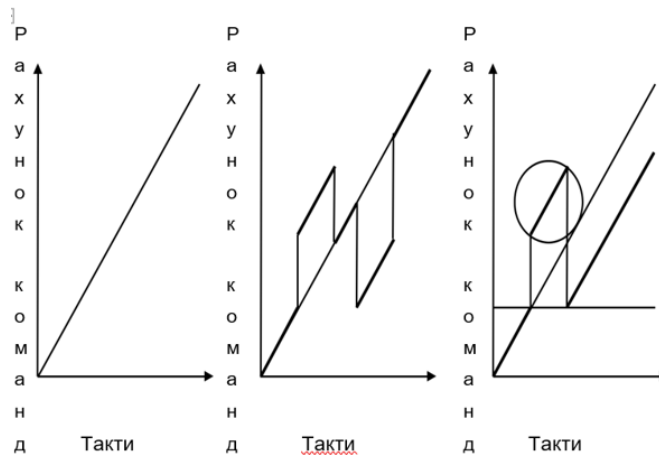


Рис. 2. а) Потік управління без переходів; б) потік управління з переходами; в) потік управління з процедурою (обведена).

Процедура. Виклик процедури аналогічний команді переходу в тому сенсі, що він також передає управління - змінює потік управління, але, на відміну від команд переходу, після виконання процедури управління повертається команді, що викликала процедуру (рис.2 в)). Тіло процедури можна розглядати як визначення нової команди на більш високому рівні.

Зазвичай процедури мають ієрархію. Якщо потрібно рівноправність процедур, наприклад, для паралельної обробки даних на одному процесорі, застосовуються **співпрограми**. Кожна співпрограма працює як би одночасно з іншими співпрограмами так, як ніби у неї є власний процесор.

Тут важливо розрізняти співпрограми і підпрограми.

	Підпрограма	Співпрограми
Точки ходу	Підпрограма завжди має одну точку входу.	Співпрограми має одну точку входу і всередині послідовність повернень і наступних за ними точок входу.
Повернення управління	Підпрограма повертає управління тільки один раз. Після повторного виклику підпрограми управління починається зі стартовою точки входу.	Співпрограми може повертати управління багато разів. після повторного виклику співпрограми управління починається з місця, в якому було передано управління.

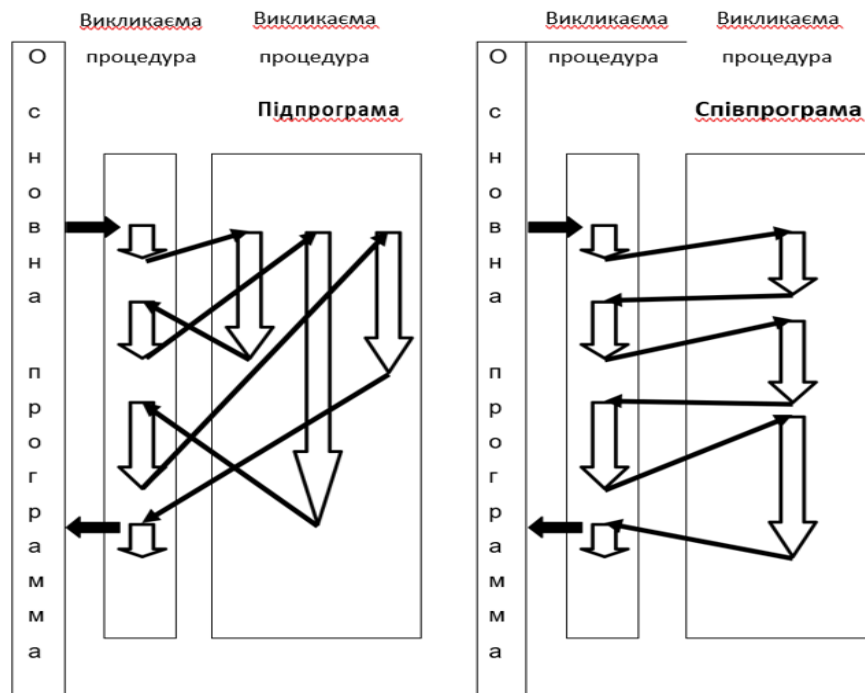


Рис. 3. а) організація управління з викликом підпрограми; б) організація управління з викликом співпрограми.

Виділяють три типи елементів потоку управління:

- **контейнери** – забезпечують структури в пакетах;
- **завдання** – забезпечують функціональність;
- **елементи керування черговістю** – з'єднують компоненти, що виконуються, контейнери і завдання в упорядкований потік управління.

Контейнери – це об'єкти, що містять структуру пакетів і служби для завдань. Підтримують повторення потоків управління в пакетах, а також групують завдання і контейнери в єдині робочі об'єкти. Крім задач, контейнери можуть включати в себе інші контейнери.

Пакети використовують контейнери для вирішення наступних завдань:

- повторення завдань для кожного елемента: файлів, папок, схем або керуючих об'єктів. Наприклад, пакет може виконувати інструкції, розміщені в декількох файлах.
- повторення завдань до тих пір, поки значення певного виразу не дорівнюватиме **false**. Наприклад, пакет може щодня посылати повідомлення по електронній пошті протягом заданого терміну.
- угруповання завдань і контейнерів, виконання яких, успішне або аварійне, враховується як для єдиного об'єкта. Наприклад, пакет може групувати завдання, що видаляють і додають рядки в таблиці бази даних, а потім фіксувати їх або ж проводити відкат всіх пов'язаних завдань в разі збою однієї з них.

Завдання – це елементи потоку управління, які визначають робочі модулі, що виконуються в потоці управління пакета. Якщо в пакеті кілька завдань, вони пов'язані і впорядковані в потоці управління з допомогою управління черговістю.

Елементи управління черговістю пов'язують виконувані об'єкти, контейнери і завдання в пакетах в потік управління і задають умови, які визначають, чи виконуються виконувані об'єкти.

Як виконуваного об'єкта можуть бути контейнери «цикл по елементам» і «цикл по кожному елементу», контейнери послідовності, завдання або **обробник події**. Обробники подій також використовують управління черговістю для зв'язування своїх виконуваних об'єктів в потік управління.

Потік даних: чергова команда отримує можливість виконання відразу після того як з'являються операнди.

Особливості потоку даних:

- наявний спеціальний механізм пересилання даних для розсилки проміжних даних безпосередньо між командами;
- операнди можуть бути вбудовані в команди;
- не використовується традиційна концепція загальної пам'яті даних, внаслідок немає конфліктів і тимчасових втрат;
- обмеження по послідовності виконання обчислювальних дій визначаються тільки залежностями між даними, що дозволяє повністю реалізувати властивий моделі паралелізм.

Тема 16. Паралельні та нетрадиційні архітектури.

1. Опис архітектури паралельних обчислювальних систем.

Застосування паралельних обчислювальних систем (ПВС) і суперкомп'ютерів є стратегічним напрямком розвитку обчислювальної техніки. Це викликано не тільки принциповим обмеженням максимально можливого швидкодії звичайних послідовних ЕОМ, але і практично постійною наявністю обчислювальних завдань, для вирішення яких можливостей існуючих засобів обчислювальної техніки завжди виявляється недостатньо. Вихід із цього становища напрошувався сам собою. Якщо один комп'ютер не справляється з вирішенням завдання за потрібний час, то спробуємо взяти два, три, десять комп'ютерів і змусимо їх одночасно працювати над різними частинами загальної задачі, сподіваючись отримати відповідне прискорення.

Існує безліч способів організації паралельно працюють обчислювальних систем. Паралельність обчислень, коли в один і той же момент виконується одночасно кілька операцій обробки даних, здійснюється в основному за рахунок введення надмірності функціональних пристроїв (багатопроцесорності). В цьому випадку можна досягти прискорення процесу вирішення обчислювальної задачі, якщо здійснити поділ вживаного алгоритму на інформаційно незалежні частини і організувати виконання кожної частини обчислень на різних функціональних пристроях (процесорах, ядрах, арифметико-логічних пристроях і т. д.). Подібний підхід дозволяє виконувати необхідні обчислення з меншими витратами часу в порівнянні з послідовною організацією обчислень. Можливість отримання максимального прискорення обмежується (в принципі) тільки числом наявних процесорів і кількістю паралельно виконуються частин в обчисленнях.

Розглядаючи проблеми організації паралельних обчислень, слід розрізняти такі відомі режими виконання незалежних частин програми:

- багатозадачність (режим поділу часу), при якому для виконання декількох процесів використовується єдиний процесор. Цей режим є псевдопаралельним: активним (виконуваним) може бути один єдиний процес, а всі інші процеси знаходяться в стані очікування своєї черги. Застосування режиму поділу часу може підвищити ефективність організації обчислень (наприклад, якщо один з процесів не може виконуватися через очікування даних, що вводяться, то процесор буде задіяний для виконання іншого, готового до виконання процесу). Крім того, в даному режимі проявляються багато ефекти паралельних обчислень (необхідність взаємовиключення і синхронізації процесів і ін.), і як результат, цей режим може бути використаний при початковій підготовці паралельних програм;
- паралельні обчислення, коли в один і той же момент може виконуватися кілька команд обробки даних. Такий режим обчислень може бути забезпечений не тільки при використанні декількох процесорів, але і за допомогою конвеєрних і векторних обробних пристроїв;

- розподілені обчислення – термін, який зазвичай застосовують для вказівки паралельної обробки даних, коли використовується кілька обробних пристроїв, досить віддалених один від одного, в яких передача даних по лініях зв'язку призводить до суттєвих тимчасових затримок. Як результат ефективна обробка даних при такому способі організації обчислень можлива тільки для паралельних алгоритмів з низькою інтенсивністю потоків міжпроцесорних передач даних. Перераховані умови є характерними, наприклад, при організації обчислень в багатомашинних обчислювальних комплексах, утворених об'єднанням декількох окремих ЕОМ за допомогою каналів зв'язку локальних або глобальних інформаційних мереж.

При розробці паралельних алгоритмів розв'язання задач принциповим моментом є аналіз ефективності використання паралелізму, що складається зазвичай в оцінці одержуваного прискорення процесу обчислення (скорочення часу рішення задачі). Формування подібних оцінок прискорення може здійснюватися стосовно до вибраного обчислювального алгоритму (оцінка ефективності розпаралелювання конкретного алгоритму). Інший важливий підхід може полягати в побудові оцінок максимально можливого прискорення процесу отримання рішення задачі конкретного типу (оцінка ефективності паралельного способу вирішення завдання).

Власне паралелізм передбачає наявність декількох (n) пристроїв для обробки даних і алгоритм, що дозволяє виробляти на кожному незалежну частину обчислень. В кінці обробки часткові дані збираються разом для отримання остаточного результату. У цьому випадку (нехтуючи накладними витратами на отримання та збереження даних) можна отримати прискорення процесу в n раз. Далеко не кожен алгоритм може бути успішно розпаралелен таким способом (природною умовою розпаралелювання є обчислення незалежних частин вихідних даних за однаковими, або подібним, процедурам; ітераційність або рекурсивність викликають найбільші проблеми при розпаралелюванні).

2. Класифікація паралельних архітектур.

Існують різні класифікації, які призначені для різних цілей. При розробці паралельного алгоритму важливо знати тип оперативної пам'яті, тому що вона визначає спосіб взаємодії між частинами паралельної програми. В залежності від організації підсистем оперативної пам'яті паралельні комп'ютери можна розділити на два наступних класи. Історично самою ранньою є класифікація М. Флінна (1966 р.). класифікація заснована на понятті потоку, під яким розуміється послідовність команд або даних, що обробляються процесором. На основі числа потоків команд та потоків даних відокремлюють чотири класи архітектур (рис. 1):

- SISD (Single Instruction, Single Data) — системи, в яких існує одиночний потік команд і одиночний потік даних; до даного типу систем можна віднести звичайні послідовні ЕОМ;

- SIMD (Single Instruction, Multiple Data) — системи з одним потоком команд і множинним потоком даних; подібний клас систем складають системи, в яких в кожний момент часу може виконуватися одна й та ж команда для обробки декількох інформаційних елементів;
- MISD (Multiple Instruction, Single Data) — системи, в яких існує множинний потік команд і одиночний потік даних; прикладів конкретних ЕОМ, що відповідають даному типу обчислювальних систем, не існує; введення подібного класу робиться для повноти системи класифікації;
- MIMD (Multiple Instruction, Multiple Data) — системи з множинним потоком команд і множинним потоком даних; до подібного класу систем належить більшість паралельних багатопроцесорних (багатоядерних) обчислювальних систем.

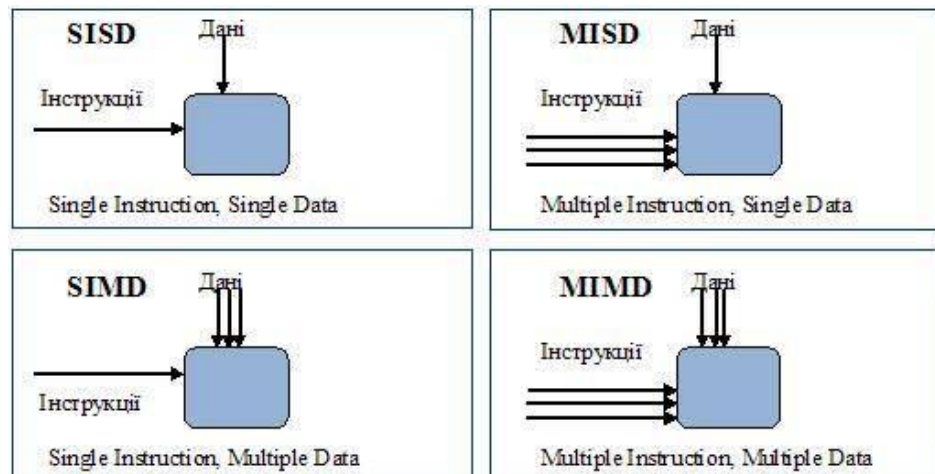


Рис.1. Чотири класи паралельних архітектур.

Класифікація Хокні.

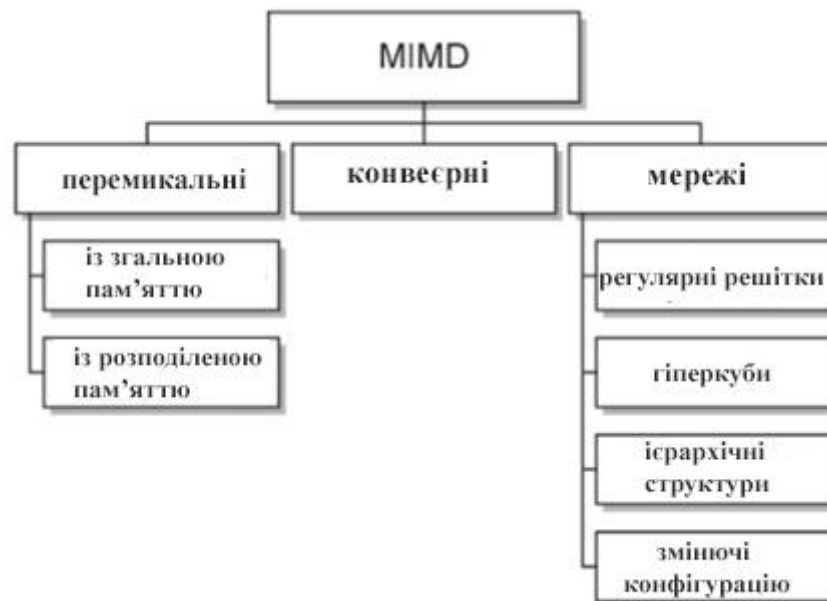


Рис. 2. Класифікація Хокні.

Класифікація Дункана:

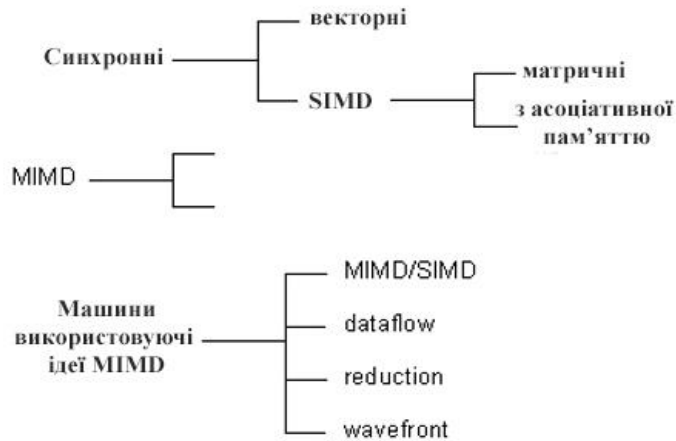


Рис. 3. Класифікація Дункана.

Класифікація паралельних архітектур

Паралельні обчислювальні архітектури можна класифікувати на основі різних критеріїв, таких як тип процесорів, структура пам'яті та спосіб обміну даними між процесорами. Основні типи паралельних архітектур включають:

1. Симетричні багатопроцесорні системи (SMP):

- У SMP-системах всі процесори мають спільний доступ до загальної пам'яті. Вони працюють під єдиною операційною системою і мають рівноправний доступ до всіх ресурсів системи, що забезпечує високу ефективність обробки завдань з високим рівнем взаємодії між процесорами.

2. Системи з розподіленою пам'яттю:

- У таких системах кожен процесор має свою власну локальну пам'ять, і обмін даними між процесорами здійснюється через мережеві з'єднання. Це дозволяє масштабувати систему на велику кількість процесорів, проте може викликати затримки при передачі даних між процесорами.

3. Кластери:

- Кластери складаються з набору окремих комп'ютерів (вузлів), з'єднаних мережею, кожен з яких виконує частину загального завдання. Кластери можуть використовувати як спільну, так і розподілену пам'ять, і зазвичай використовуються для вирішення великих наукових і інженерних задач.

4. Графічні процесори (GPU):

- GPU-архітектури спеціалізовані на обробці великих обсягів однотипних даних і широко застосовуються для паралельних обчислень у задачах машинного навчання, графіки та симуляцій. Вони мають велику кількість ядер, що дозволяє ефективно виконувати масивні паралельні обчислення.

5. Векторні процесори:

- Ці процесори здатні виконувати одну операцію над великими масивами даних одночасно. Векторні архітектури використовуються для задач, що потребують високопродуктивних обчислень, таких як обробка сигналів і чисельне моделювання.

Кожен тип паралельної архітектури має свої переваги та недоліки, і вибір конкретної архітектури залежить від характеру завдань, що розв'язуються, і вимог до продуктивності та ефективності системи.

Класифікація паралельних архітектур за типом процесорів

1. Багатоядерні процесори (Multicore Processors):

- Багатоядерні процесори містять декілька обчислювальних ядер в одному чипі. Кожне ядро може виконувати окремий потік команд, що дозволяє паралельно обробляти кілька завдань. Ці процесори широко використовуються в персональних комп'ютерах, серверах та мобільних пристроях.

2. Графічні процесори (GPU - Graphics Processing Units):

- GPU мають велику кількість ядер, спеціалізованих на паралельній обробці великих масивів однотипних даних. Вони ефективно використовуються для обчислень, пов'язаних з обробкою графіки, машинним навчанням, симуляціями та науковими розрахунками.

3. Векторні процесори (Vector Processors):

- Векторні процесори здатні виконувати одну операцію над великими векторами даних одночасно. Вони використовуються для задач, що потребують високопродуктивних обчислень, таких як обробка сигналів та чисельне моделювання.

4. Сопроцесори (Coprocessors):

- Сопроцесори – це додаткові процесори, які працюють разом з основним CPU, спеціалізуючись на певних типах обчислень. Наприклад, математичні сопроцесори прискорюють виконання складних обчислень з плаваючою комою.

5. Процесори цифрової обробки сигналів (DSP - Digital Signal Processors):

- DSP призначені для ефективної обробки цифрових сигналів у режимі реального часу. Вони широко застосовуються в телекомунікаціях, аудіо та відео обробці, а також у системах управління.

6. Процесори з архітектурою MIMD (Multiple Instruction, Multiple Data):

- MIMD процесори можуть виконувати різні інструкції над різними даними одночасно. Ця архітектура дозволяє гнучко розподіляти завдання між процесорами, що особливо корисно в багатопроекторних системах.

7. Процесори з архітектурою SIMD (Single Instruction, Multiple Data):

- SIMD процесори виконують одну інструкцію над кількома даними одночасно. Вони ефективні для задач, де однакові операції виконуються над великими наборами даних, такі як обробка зображень і відео.

8. Гібридні процесори (Hybrid Processors):

- Гібридні процесори об'єднують властивості різних типів процесорів, наприклад, CPU з інтегрованим GPU. Це дозволяє використовувати переваги обох типів обчислень в одному чипі, підвищуючи загальну продуктивність системи.

Кожен з цих типів процесорів має свої специфічні застосування і переваги, і вибір конкретного типу залежить від характеру завдань та вимог до системи.

Класифікація паралельних архітектур за способом обміну даними між процесорами

1. Системи зі спільною пам'яттю (Shared Memory Systems):

- У цих системах всі процесори мають доступ до спільної адресного простору пам'яті. Обмін даними між процесорами здійснюється через спільну пам'ять. Ці системи діляться на дві підкатегорії:
 - **Симетричні багатопроцесорні системи (SMP - Symmetric Multiprocessing):** В SMP-системах всі процесори мають рівноправний доступ до всієї пам'яті і можуть безпосередньо взаємодіяти один з одним через спільну пам'ять.
 - **Уніфіковані системи пам'яті (UMA - Uniform Memory Access):** В UMA-системах всі процесори мають рівний час доступу до всіх частин пам'яті. Це спрощує програмування, але може створювати проблеми з масштабуванням через обмеження на пропускну здатність пам'яті.

2. Системи з розподіленою пам'яттю (Distributed Memory Systems):

- У таких системах кожен процесор має свою локальну пам'ять, і обмін даними між процесорами здійснюється через мережеві з'єднання. Ці системи дозволяють масштабуватися на велику кількість процесорів. Основні типи таких систем включають:
 - **Кластери (Clusters):** Кластери складаються з набору окремих комп'ютерів (вузлів), з'єднаних мережею. Кожен вузол має власну пам'ять і обмінюється даними з іншими вузлами через мережу.
 - **Масово-паралельні процесори (MPP - Massively Parallel Processors):** В MPP-системах велика кількість процесорів з власною пам'яттю з'єднані високошвидкісною мережею. Ці системи призначені для виконання надзвичайно великих обчислювальних завдань.

3. Гібридні системи (Hybrid Systems):

- Гібридні системи поєднують елементи як спільної, так і розподіленої пам'яті. В таких системах можуть бути підсистеми зі спільною пам'яттю, які взаємодіють між собою через мережу як розподілені системи. Це дозволяє комбінувати переваги обох типів пам'яті для досягнення більшої продуктивності та масштабованості.

- **NUMA (Non-Uniform Memory Access):** В NUMA-системах процесори мають нерівномірний доступ до різних частин пам'яті, тобто кожен процесор має швидкий доступ до локальної пам'яті і повільніший доступ до віддаленої пам'яті інших процесорів. Це дозволяє масштабувати систему без втрати продуктивності.

4. Системи з обміном повідомленнями (Message Passing Systems):

- У таких системах обмін даними між процесорами здійснюється за допомогою передачі повідомлень через мережу. Кожен процесор має свою локальну пам'ять, і для взаємодії з іншими процесорами використовуються спеціальні протоколи передачі повідомлень. Цей підхід широко використовується в розподілених обчисленнях та кластерах.

5. Архітектури з спільним розділом пам'яті (Partitioned Global Address Space - PGAS):

- В PGAS-системах глобальний адресний простір розділений між процесорами, але кожен процесор може адресувати частину пам'яті інших процесорів. Це дозволяє поєднувати переваги спільної і розподіленої пам'яті, забезпечуючи більш ефективний обмін даними між процесорами.

Кожна з цих архітектур має свої переваги та обмеження, і вибір конкретної архітектури залежить від типу задач, що розв'язуються, та вимог до продуктивності і масштабованості системи.

Класифікація нетрадиційних архітектур обчислювальних систем

1. Квантові обчислювальні системи (Quantum Computing Systems):

- Використовують квантові біти (кубіти), які можуть перебувати у станах суперпозиції, дозволяючи виконувати обчислення набагато швидше для певних задач, ніж традиційні бінарні комп'ютери. Квантові комп'ютери можуть революціонізувати такі області, як криптографія, оптимізація та моделювання молекулярних процесів.

2. Обчислювальні системи з ДНК (DNA Computing Systems):

- Використовують молекули ДНК для зберігання та обробки інформації. Цей підхід дозволяє вирішувати складні задачі комбінаторики та оптимізації шляхом біологічних процесів. ДНК-обчислення є перспективними для задач, де потрібен паралелізм на молекулярному рівні.

3. Нейроморфні обчислювальні системи (Neuromorphic Computing Systems):	Імітують структуру та функції біологічних нейронних мереж. Використовують спеціалізовані апаратні рішення, такі як нейронні процесори, що забезпечують ефективну обробку задач штучного інтелекту, розпізнавання образів та обробки сенсорних даних.
4. Оптичні обчислювальні системи (Optical Computing Systems):	Використовують світло для передачі та обробки інформації. Оптичні комп'ютери можуть забезпечувати вищі швидкості обробки даних і менші затримки порівняно з електронними системами, що робить їх перспективними для високошвидкісних мереж та обробки великих обсягів даних.
5. Терагерцові обчислювальні системи (Terahertz Computing Systems):	Використовують терагерцові частоти для обробки інформації, що дозволяє досягти дуже високих швидкостей обчислень. Ця технологія знаходиться на стадії досліджень і розробок, але має потенціал для значного збільшення продуктивності обчислювальних систем.
6. Резистивні обчислювальні системи (Reconfigurable Computing Systems):	Використовують спеціальні про

3. Класифікація нетрадиційних архітектур.

ДНК процесори.

В даний час в пошуках реальної альтернативи напів-провідникових технологій створення нових обчислювальних систем вчені звертають дедалі більшу увагу на біотехнології, або біокомп'ютинг, який представляє собою гібрид інформаційних, молекулярних технологій, в тому числі біохімічних. Біокомп'ютинг дозволяє вирішувати складні обчислювальні задачі, використовувати методи, прийняті в біохімії і молекулярної біології, організатори розраховують за допомогою живих тканин, клітин, вірусів і біомолекул.

В основному елементі (процесорі) використовуються частки дезоксирибонуклеїнової кислоти.

Асоціативні процесори.

Асоціативні системи відносяться до класу: один потік команд - безліч потоків даних (SIMD = Single Instruction Multiple Data). Ці системи включають велике число операційних пристроїв, здатних одночасно по командам керуючого пристрою вести обробку декількох потоків даних. В асоціативних обчислювальних системах інформація на обчислення надходить від асоціативних запам'ятовуючих пристроїв (АЗП), що характеризуються тим, що інформація в них вибирається не по певній адресі, а за її змістом.

Матричні процесори.

Найбільш поширеними з систем, класу: один потік команд - безліч - потоків даних (SIMD), є матричні системи, які найкраще пристосовані для вирішення завдань, що характеризуються паралелізмом незалежних об'єктів або даних. Організація систем подібного типу на перший погляд досить проста. Вони мають загальне керуючий пристрій, що генерує потік команд і велике число процесорних елементів, які працюють паралельно і обробляють кожна свій потік даних.

Комунікаційні процесори.

Комунікаційні процесори – це мікрочіпи, які є чимось середнім між жорсткими спеціалізованими інтегральними мікросхемами і гнучкими процесорами загального призначення.

Комунікаційні процесори програмуються, як і звичайні процесори, але побудовані з урахуванням мережевих завдань, оптимізовані для роботи в мережі, і на їх основі виробники - як процесорів, так і обладнання - пишуть програмне забезпечення для специфічних додатків.

Процесори з багатозначною (нечіткою) логікою.

Ідея побудови процесорів з нечіткою логікою (fuzzy logic) ґрунтується на нечіткій математиці. Математична теорія нечітких множин, запропонована проф. Л.А. Заде, будучи предметом інтенсивних досліджень, відкриває все більші можливості перед системними аналітиками. Засновані на цій теорії різні комп'ютерні системи, в свою чергу, істотно розширюють сферу застосування нечіткої логіки. Завдання за допомогою нечіткої логіки вирішуються за наступним принципом:

- 1) чисельні дані (показання вимірювальних приладів, результати анкетування) фаззуються (переводяться в нечіткий формат);
- 2) обробляються за певними правилами;
- 3) дефаззуються і у вигляді звичної інформації подаються на вихід.

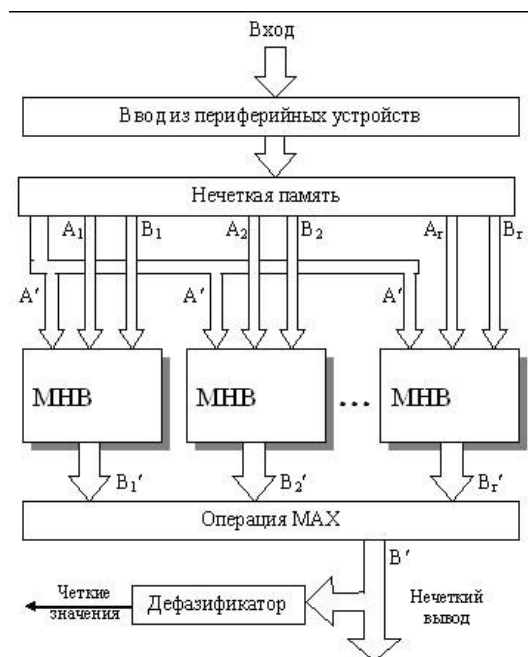


Рис. 4. Процесори з багатозначною (нечіткою) логікою.

Сигнальні процесори.

У відповідь на зростаючі запити споживачів фірма Motorola розробила нову архітектуру мікросхеми, орієнтовану як на виконання складних алгоритмів цифрової обробки сигналів, так і на вирішення завдань управління. Сімейство мікросхем DSP568xx побудовано на базі ядра 16-розрядного процесора DSP56800 з фіксованою точкою. Це ядро призначене для ефективного вирішення завдань управління і цифрової обробки сигналів. Реалізований в ньому набір команд забезпечує цифрову обробку сигналів з ефективністю кращих DSP загального призначення і відповідає вимогам простоти створення компактних програм управління.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Список основної літератури

1. Вичужанін В. В., Вичужанін О. В. Комп'ютерна схемотехніка та архітектура комп'ютерів: навч. посіб. для здобувачів першого (бакалаврського) рівня вищої освіти за спец. 126 "Інформаційні системи та технології", 174 "Автоматизація, комп'ютерно-інтегровані технології та робототехніка" / В. В. Вичужанін, О. В. Вичужанін. – Одеса : Нац. ун-т "Одеська політехніка", 2024. – 181 с.
2. Євсєєв С. П., Дженюк Н. В., Охрименко М. Ю. та ін. Цифрова схемотехніка та архітектура мікропроцесорів: навч. посіб. / С. П. Євсєєв, Н. В. Дженюк, М. Ю. Охрименко та ін. – Харків–Львів : Новий Світ–2000, 2024. – 513 с.
3. Батрак Є. О. Архітектура комп'ютерних систем : лаб. практикум для студ. спец. 126 "Інформаційні системи та технології" / Є. О. Батрак. – Київ : КПІ ім. Ігоря Сікорського, 2020. – 110 с.
4. Воробйова О. М., Іванченко В. Д. Основи схемотехніки : навч. посіб. / О. М. Воробйова, В. Д. Іванченко. – Одеса: Фенікс, 2009. – 388 с.
5. Демиденко М. І., Руденко О. А. Комп'ютерна схемотехніка та архітектура комп'ютерів: навч. посіб. для студ. спец. 122 "Комп'ютерні науки" / М. І. Демиденко, О. А. Руденко. – Полтава : ПолтНТУ, 2017. – 98 с.

Список додаткової літератури

1. Katz R.H., Borriello G. Contemporary Logic Design. — 2nd ed. — Pearson, 2004. — 608 p.
2. Sedra A.S., Smith K.C. Microelectronic Circuits. — 8th ed. — Oxford University Press, 2020. — 1360 p.
3. Floyd T.L. Digital Fundamentals. — 11th ed. — Pearson, 2015. — 816 p.
4. Baker R.J. CMOS: Circuit Design, Layout, and Simulation. — 3rd ed. — Wiley-IEEE Press, 2010. — 1040 p.
5. Brown S., Vranesic Z. Fundamentals of Digital Logic with Verilog Design. — 3rd ed. — McGraw-Hill, 2013. — 800 p.
6. Anderson D.V., Anderson J.D. Fundamentals of Semiconductor Devices. — McGraw-Hill, 2005. — 704 p.
7. Harris D.M., Harris S.L. Digital Design and Computer Architecture. — 2nd ed. — Morgan Kaufmann, 2012. — 712 p.
8. Nilsson J.W., Riedel S.A. Electric Circuits. — 11th ed. — Pearson, 2018. — 944 p.
9. Kuo B.C. Automatic Control Systems. — 9th ed. — Wiley, 2014. — 928 p.
10. Циганок О.О. Основи схемотехніки: навчальний посібник. — Київ: Кондор, 2011. — 320 с.
11. Мельник М.В. Мікроелектроніка і мікропроцесорна техніка: навчальний посібник. — Львів: ЛНУ імені Івана Франка, 2009. — 368 с.