

Р.Рикалюк

АРХІТЕКТУРА КОМП'ЮТЕРНИХ СИСТЕМ

Тексти лекцій

Львів-2015

Список рекомендованої літератури:

1. Бройдо В.Л., Ильина О.П. Архитектура ЭВМ и систем: Учебник. – СПб.: Питер, 2009. – 720 с.
2. Воробйова О.М., Иванченко В.Д. Основы схемотехники: Навч. посібник. У 2-х ч. – Одеса: ОНАЗ ім. О.С. Попова, 2004. – Ч. 2. –172 с.
3. Горбунов В.Л., Панфилов Д.И., Преснухин Д.Л. Справочное пособие по микропроцессорам и микроЭВМ. – М., 1988.
4. Гук М. Аппаратные средства IBM PC. Энциклопедия 3-е изд – СПб.,: Питер, 2006. - 1072 с: ил..
5. Жмакин А.П. Архитектура ЭВМ: Учебник. – СПб.: ВHV-Петербург, 2006. – 320 с.
6. Злобін Г.Г, Рикалюк Р.Є. Архітектура та апаратне забезпечення ПЕОМ: Навч.посіб. –К., 2006., -К.,2012.
7. Енциклопедія кібернетики у 2 т. – К., 1977.
8. Каган Б.М. Электронные вычислительные машины и системы. – М., 1991.
9. Колодницький М.М. Технічне та програмне забезпечення комп'ютерних інформаційних технологій. – Житомир, 1995.
10. Кравчук С.О., Шонін В.О. Основы комп'ютерної техніки: Навч. посібн. – К.: Каравела, 2008. – 344 с.
11. Ланцов А.Л., Зворыкин Л.Н., Осипов И.Ф. Цифровые устройства на комплементарных МПД интегральных микросхемах. – М., 1983.
12. Локазюк В.М. Мікропроцесори та мікроЕОМ у виробничих системах. – К., 2002.

13. Мельник А.О. Архітектура комп'ютера. – Луцьк: Волинська обласна друкарня, 2008. – 470 с.
14. Мячев А.А., Степанов В.Н. Персональные ЭВМ и микроЭВМ. Основы организации: Справочник. – М., 1991.
15. Рикалюк Р.Є. Архітектура комп'ютерів.Текст лекцій – Львів, 2002.
16. Смирнов А.Д., Архитектура вычислительных систем. – М., 1990.
17. Таненбаум Э. Архитектура компьютера. – 5-е изд. - СПб.: Питер, 2007. — 844 с..
18. Толковый словарь по вычислительным системам / Под ред. В. Иллингуорта. – М., 1991.
19. Дэвид М. Харрис и Сара Л. Харрис. Цифровая схемотехника и архитектура компьютера/второе издание. - Morgan Kaufman/ © English Edition 2013.

ВСТУПНА ЛЕКЦІЯ. ПРИНЦИПИ фон-НЕЙМАНА.

Мабуть у кожної людини, яка звернула погожого вечора погляд до неба, виникає зацікавлення, як усе це у Світі влаштовано, що так чітко функціонує. Сподіваємось, що не менша цікавість виникає у кожного, хто пізнав чи пізнає можливості сучасного комп'ютера, а особливо у майбутніх фахівців комп'ютерної справи. Мета цього курсу власне полягає у вивченні основ побудови сучасних комп'ютерних систем, принципів функціонування окремих вузлів та можливостей використання цих знань для програмування. Передусім варто ознайомитися з короткою історією розвитку обчислювальних машин. Вона сягає середньовіччя й умовно поділена на кілька напрямів.

А. Механічні машини

Зовсім недавно (1967р.) дослідники знайшли у щоденниках Леонардо да Вінчі (1452-1519) ескіз сумуючої машини на зубчатих колесах, спроможної додавати 13-розрядні десяткові числа. Спеціалісти ІВМ відтворили цей ескіз у металі і машина виявилась працездатною!

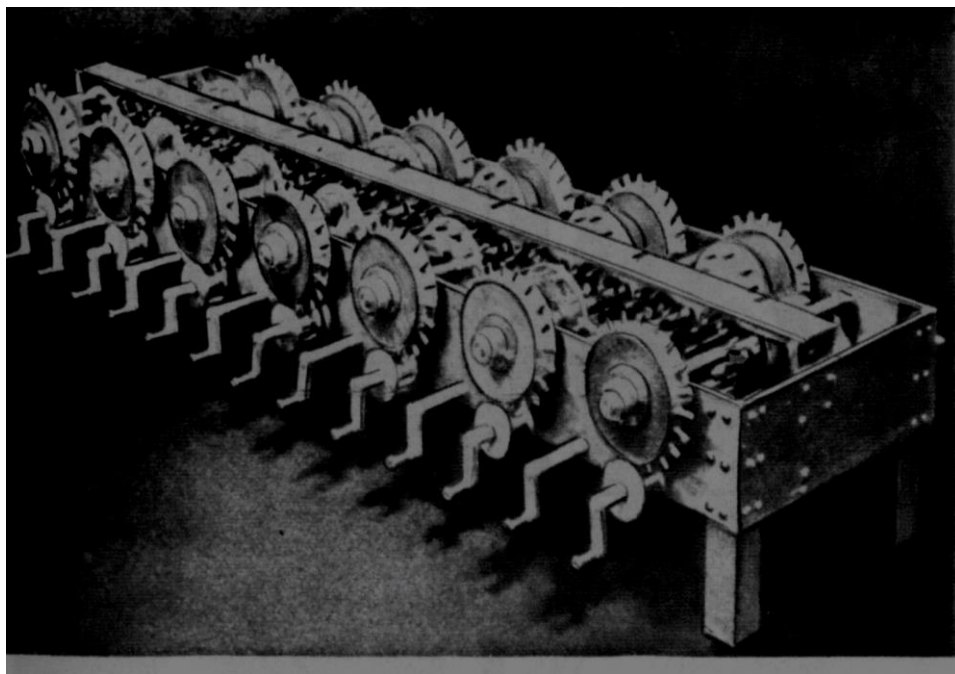


Рис.1.1. Сумуюча машина Леонардо да Вінчі

На початку 17-го сторіччя німецький вчений Вільгельм Шіккард (1592-1636) у листі Йоганну Кеплеру подає малюнок лічильної машини для сумування і множення 6-розрядних десяткових чисел. Про це також стало відомо у наші дні. У 1617 р. Джон Непер розробив дерев'яний арифмометр, однак найвідомішою стала обчислювальна машина Блеза Паскаля (1643-45), у якій йому вдалося розв'язати задачу перенесення десятків. Першу модель Б. Паскаль виготовив ще у 17-річному віці, а всього їх близько 50. Головна ідея механічної обчислювальної машини полягала у тому, що числа відображали за допомогою лічильних коліщаток або валиків з зубцями. Коліщаткам

надавали руху за допомогою конічних чи циліндричних зачеплень. Розряд фіксували за допомогою штифтів. Механізм перенесення розрядів працював подібно до того, як у механічному годиннику секундний механізм приводить у дію хвилинний.



Рис.1.2. Машина Паскаля

Машина Готфріда Вільгельма Лейбніца (1673) – це арифмометр на чотири дії, в основі якого був валик із зубцями (виконував додавання, множення і ділення 12-розрядних десяткових чисел).

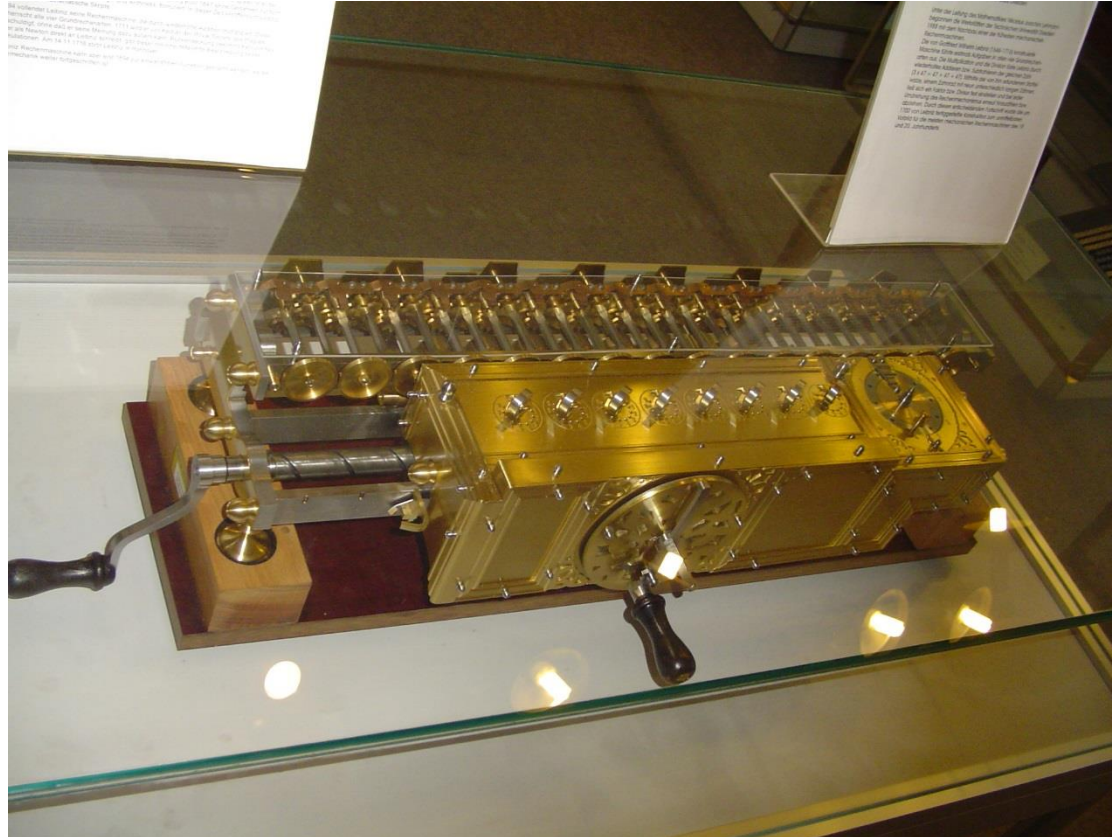


Рис.1.3. Машина Лейбніца (копія з Німецького музею).

Майже через століття з'явився суматор Є. Якобсона (1770). У Франції в 1799 р. Жозеф Марі Жакар винайшов ткацький верстат з перфокартами для задання узору — майже через два століття перфокарти масово використовувались для уведення даних і програм в ЕОМ.

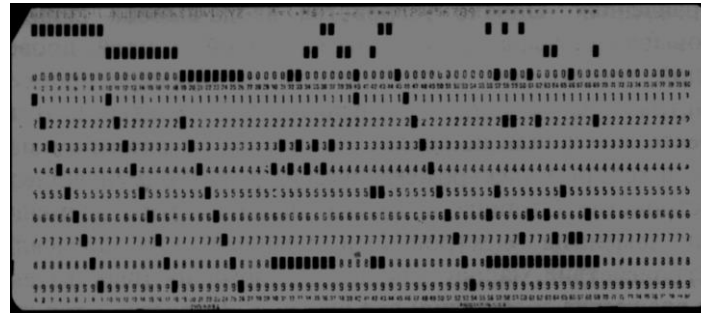


Рис.1.4. Перфокарта для ЕОМ серії ЕС-10хх

У 1885 р. Д. Фельт сконструював комптометр, у якому числа вводили з клавіатури, а у 1885 р. американець У. Берроуз в виготовив машину, яка друкувала вхідні дані та результати обчислень і у 1887 р. створив першу компанію з виробництва арифмометрів.

Епоха арифмометрів тривала досить довго. У 1709 р. Джованні Пелен сконструював колесо зі змінною кількістю зубців. Більше ніж через 150 років російський службовець В.Т.Однер (1876) удосконалив конструкцію колеса й такі арифмометри (старшому поколінню людей відомі під маркою “Фелікс”) випускали у колишньому СРСР з 1929 до 1978р. на заводах лічильних машин в Курську, Пензі і Москві.

Технологія арифмометрів була настільки добре відпрацьована, що були створені навіть кишенькові моделі



Рис.1.5. Арифмометр “Фелікс”

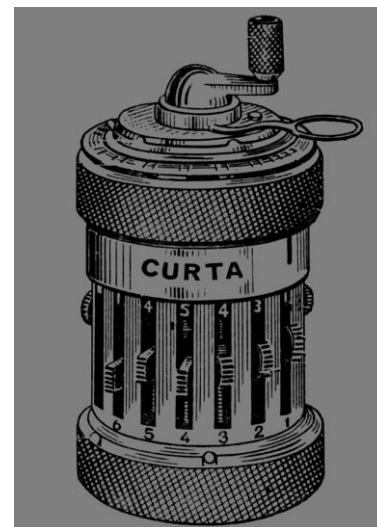


Рис.1.6. Кишеньковий арифмометр

Б. Аналогова техніка

Ера механічних лічильних машин представлена не тільки дискретними механізмами, а й пристроями безперервної дії (аналоговими). У XIX ст. розроблено пристрої для вимірювання довжин (лонгиметри) та площ (планиметри). Цікавим і дуже важливим винаходом того часу був механічний інтегратор для розв'язування диференціальних рівнянь до четвертого порядку, який розроблений 1912 р. російським математиком та інженером О.М.Криловим.

на

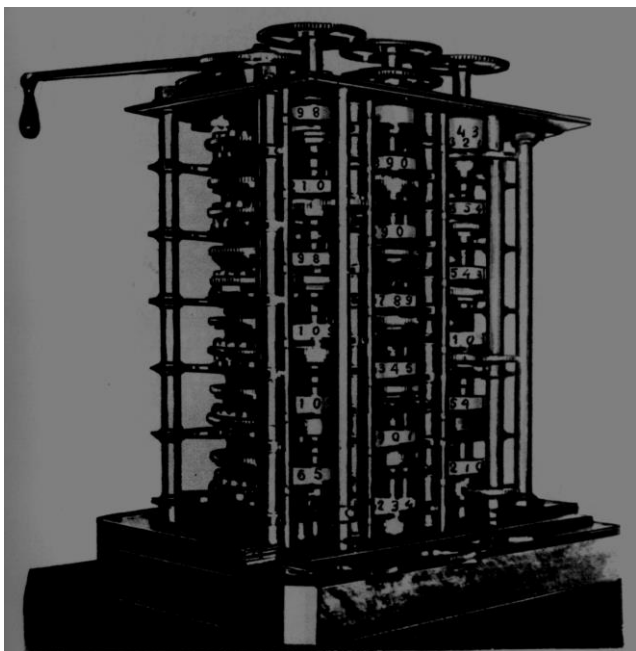


Суть машин безперервної дії полягає у зіставленні кожному миттєвому значенню змінної математичної величини, яка приймає участь у вихідних співвідношеннях, миттєвого значення іншої (машинної) величини. Часто ця машинна величина має іншу природу і потребує масштабного коефіцієнта. Кожний елемент аналогової машини виконує чітко визначену математичну операцію над машинними величинами. Цій операції відповідає певний фізичний закон, який задає математичні залежності між фізичними величинами виході і вході аналогового елемента (напр., закони Ома, Кірхгофа для електричних кіл).

Аналогова техніка розвивалася під впливом задач астрономії, балістики, навігації та ін. Найбільшого розвитку аналогові машини набули після відкриття та осмислення явищ електрики, особливо з розвитком електротехніки. Аналогові машини будували для реалізації сіткових моделей (у тому числі й економічних), для обчислення інтегралів, лінійних та нелінійних рівнянь тощо.

Рис.1.7. Польський аналоговий комп'ютер “АКАТ-1”.

Для моделювання складних задач з підвищеною точністю використовували аналогово-цифрові обчислювальні системи, побудовані за принципом комбінування у єдиному комплексі аналогової та цифрової форми зображення даних.



В. Лічильно-перфораційні машини

Появу лічильно-перфораційних машин зумовила потреба опрацювання банківської та статистичної інформації. Заради справедливості необхідно зазначити, що піонером в автоматизації обчислень був відомий англійський учений Чарльз Бебедж, який запропонував ідею побудови обчислювальної машини для табулювання функцій за допомогою методу різниць. Бебедж у 1820 – 22 рр. власноручно виготовив модель такої машини, яка могла табулювати функцію з точністю до восьми знаків.

Рис.1.8. Різницева машина Бебеджа.

У 1888 р. американський інженер Герман Голлерит створив табулятор, у якому носієм інформації слугували перфокарти. Отвори на перфокартах пробивалися за допомогою електричного струму. Головна ідея Голлерита полягала у тому, що різні характеристики відображали отворами (пробивками) у певних місцях перфокарти. Цю ідею згодом використали у пристроях введення-виведення інформації цифрових обчислювальних машин. Електромеханічні табулятори застосовували ще донедавна (70-80-ті роки XX ст.). У 1920-1940 рр. зроблено успішні спроби застосувати лічильно-перфораційні машини для науково-технічних розрахунків.

Г. Електронно-обчислювальні машини (ЕОМ)

Створенню першої електронної цифрової машини, безумовно, передувало опублікування 1854 р. відомим математиком Джорджем Булем теорії алгебри логіки, яку пізніше названо його іменем. Уже зазначено, що у 1882-83 рр. у Кембриджі Чарльз Бебедж розробив проект аналітичної машини, що містила такі складові, як арифметичний пристрій, пам'ять на 1000 п'ятидесятизначних десяткових чисел, програмний пристрій керування (програми готували на перфокартах). Машина обчислювала значення поліномів другого степеня з точністю до восьми знаків. Програми для машини Бебеджа та коментарі до неї складала леді Ада Лавлейс, дочка Джорджа Байрона; на честь цієї славної жінки названо одну з мов програмування – Ада.

У часи другої світової війни у Німеччині, Англії і США велися роботи по створенню обчислювальних пристроїв для військових потреб. У Німеччині в 1937 р. Конрад Цузе розробив механічну машину **Z1** (двійкова система числення, мантиса — 15 двійкових розрядів, порядок — 7 двійкових розрядів, оперативна пам'ять 64 слова, площа 2 кв.м.). У 1938р. **Z2** — додано пристрій вводу даних і програми з перфорованої кінострічки, а арифметичний пристрій був побудований на телефонних реле.

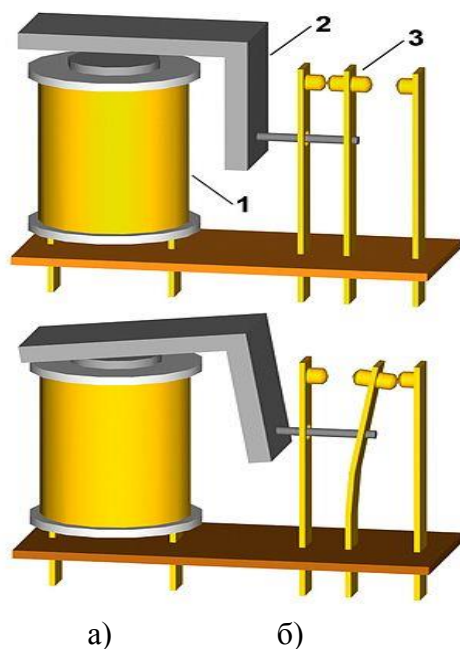


Рис.1.9. Реле — елементна база ЕОМ **Z1 - Z5**

а — через котушку реле струм не проходить, контакти розімкнені, б — через котушку реле проходить струм, контакти замкнені

У 1937 р. К. Цузе отримав німецький патент **Z23139/GMD Nr.005/021** на архітектуру обчислювача з двійковою системою числення. Обчислювач складався з процесора та оперативної пам'яті програм і даних. У 1941 р. була створена машина **Z3** (2000 реле), а у 1945 р. — **Z4**. Роботи К. Цузе були засекречені, про них довілися лише через кілька років після другої світової війни. Усього К. Цузе спроектував сім моделей ЕОМ (**Z1, Z2, Z3, Z4, Z5, Z11, Z22**). ЕОМ **Z11** і **Z22** були побудовані на електровакуумних лампах, у 1961 р. під назвою **Z23** був реалізований напівпровідниковий варіант **Z22**.



а

б

Рис.1.10. ЕОМ, спроектовані Конрадом Цузе: а — релейна **Z4** (тактова частота 30 Гц, оперативна пам'ять 64 слова, довжина слова 32 біти, з них 24 біти — мантиса, 7 бітів — порядок числа, 1 біт — знак числа, вага 1000 кг, споживана потужність 4кВт, експонується в **Deutsches Museum** в Мюнхені), б — напівпровідникова **Z23** (2700 транзисторів, 6800 діодів, тактова частота 150 кГц, довжина слова 40 біт, два арифметико логічних пристрої [плаваючою комою і цілочисельний], вага 1000 кг, споживана потужність 3кВт).

Першу вдалу спробу побудувати універсальну цифрову машину зроблено у 1944 р. Це була машина МАРК-1, проєктована Говардом Айкеном (Ейкеном) у Гарвардському університеті і реалізована фірмою ІВМ. Елементна база — електромеханічні реле, розрядність — 23 десяткові числа. Швидкодія: операції додавання (віднімання) — 0,3 с, множення — 6 с, ділення — 11 с.

Головним недоліком релейних машин була їхня невисока швидкодія та відсутність збереженої у пам'яті програми. Тому їх дуже швидко витіснили електронні машини.

Ще у 1906 р. Лі Ді Форест розробив вакуумний тріод, а у 1918 р. професор М. А. Бонч-Бруєвич сконструював двостабільну електронну схему на двох вакуумних тріодах — тригер.

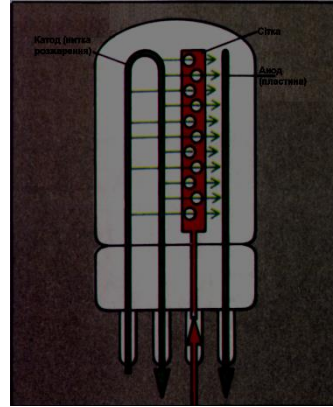


Рис. 1.11. Вакуумний тріод

Роботи зі створення першої обчислювальної машини на електронних елементах (лампах) розпочато у США на замовлення артилерійського управління для розрахунку балістичних таблиць. Це був проект ENIAC (Electronic Numerical Integrator and Computer), початий у 1943 р. і завершений у 1945 р. Джоном Мочлі та Преспером Екертом у Пенсільванському університеті.

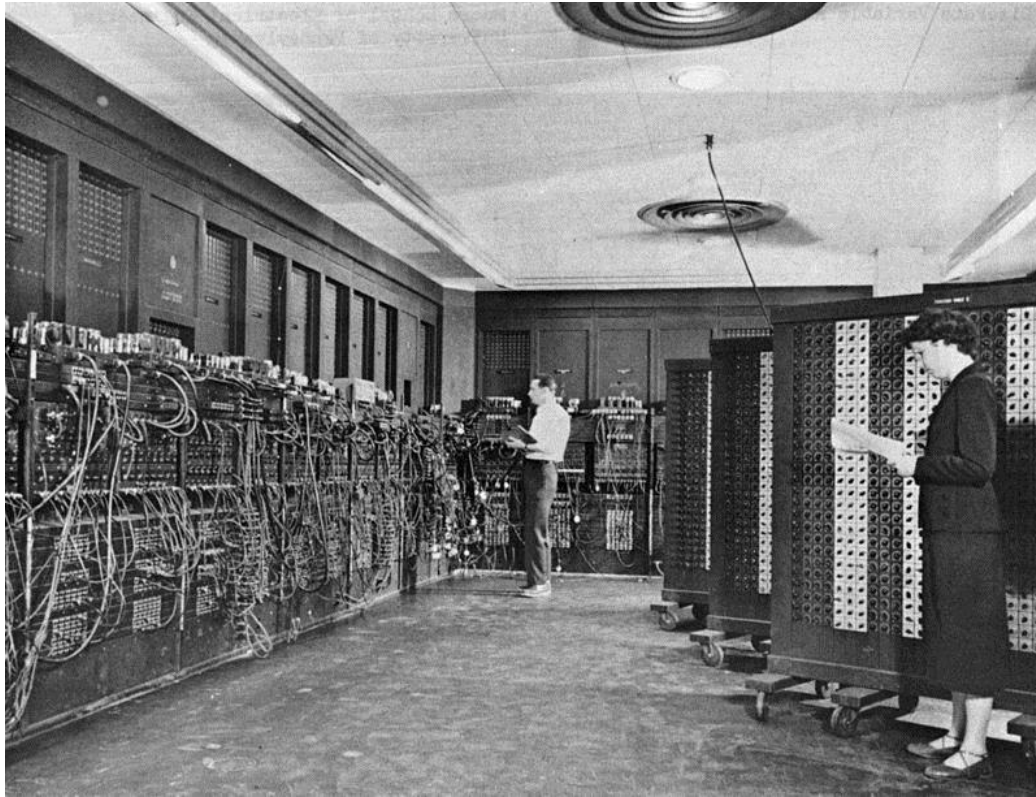


Рис. 1.14. ЕОМ Еніак

Замість реле конструктори використали тригери, збудовані на електронних лампах. Машина складалася більше ніж з 18 тис. електронних ламп і 1500 реле, споживала потужність близько 150 кВт. Довжина слова – 11 десяткових розрядів. Швидкодія: операції додавання (віднімання) – 200 мкс, множення – 2,8 мс, ділення – 6,0 мс. Зазначимо, що Д. Мочлі і П. Еккерт конструювали машину, яка б запам'ятовувала програму.

Однак у процесі роботи виникли певні труднощі, і в 1945 р. до проекту було залучено відомого математика Джона фон

Неймана. Унаслідок співпраці у 1946 р. у науковій статті Неймана, Голдстайна і Бернса були викладені основні принципи побудови електронно-обчислювальних машин.

Це відомі нам *принципи фон Неймана*:

А. Щодо пристроїв комп'ютера, то він повинен містити:

- ☐ арифметико-логічний пристрій, який виконує арифметичні та логічні операції;
- ☐ пристрій керування, який організовує процес виконання програм;
- ☐ запам'ятовувальний пристрій, або пам'ять для зберігання програм і даних;
- ☐ зовнішні пристрої для введення і виведення інформації.

Ці пристрої повинні взаємодіяти між собою за такою схемою:



Б. Щодо принципу функціонування комп'ютера:

- ☐ програма розташовується у пам'яті, причому пам'ять повинна бути двох типів – швидка (оперативна) та повільніша (зовнішня);
- ☐ програмне керування виконанням завдань, тобто пристрій керування зчитує вміст комірки пам'яті, де є перша команда (інструкція), і організовує її виконання;
- ☐ наявність умовного переходу, що дає змогу змінювати послідовність виконання команд, розташованих у пам'яті підряд;
- ☐ для відображення інформації необхідно застосовувати двійкову систему числення.

Звичайно, що сучасні комп'ютери дещо відрізняються у побудові від наведених вище вимог. Наприклад, арифметико-логічний пристрій і пристрій керування об'єднано в один пристрій – центральний процесор. Крім цього, введено низку додаткових можливостей, зокрема переривання роботи процесора, паралельне опрацювання інструкцій та ін.

ЕОМ, у якій повністю втілено принципи фон Неймана, побудував англійський конструктор Моріс Уїлкс у 1949 р. (ЕДСАК) та 1952 р. (ЕДВАК). У цьому ж році Дж. Мочлі, Дж. Преспер Еккерт і Джон фон Нейман розпочали розробку нової машини — **ЕДВАК**.

Слід зауважити, що ще у 1941 р. Джон Атанасов і Кліффорд Беррі розробили проект електронної обчислювальної машини з двійковою системою числення і оперативною пам'яттю на конденсаторах (сучасна динамічна оперативна пам'ять є великим масивом мікроскопічних конденсаторів).

В Англії у 1936 р. Алан Тьюрінг запропонував теоретичні основи побудови обчислювальних машин (машина Тьюрінга), а у 1942-1943 рр. під керівництвом Алана Тьюрінга була створена спеціалізована електронна машина **Колосс** (2000 ламп) для розшифрування німецьких радіограм

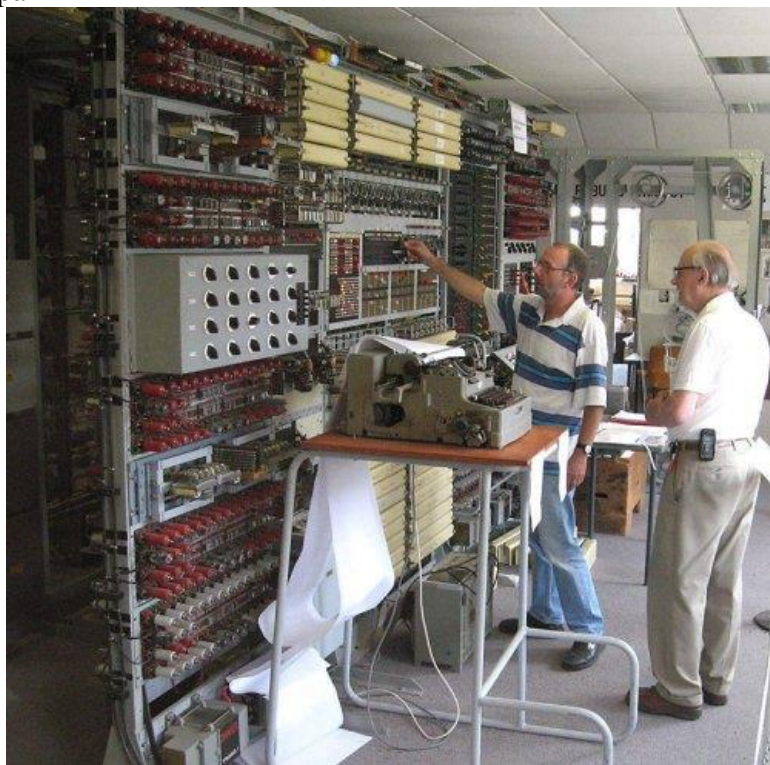


Рис.1.15. Спеціалізована машина **Колосс**

Окремо опишемо розвиток ЕОМ на теренах колишнього Радянського Союзу. У 1947 р. академік С.О.Лебедев очолив роботи з розробки ЕОМ (МЭСМ), яка повністю ґрунтувалась на принципах Неймана. Її дослідну експлуатацію розпочато 1950 р., а промислову – 1951 р. в Україні, в околицях Києва (Феофанія) [3, 8]. Це була ЕОМ першого покоління. Швидкодія такої машини становила 50 операцій за секунду. ЕОМ **МЕСМ** була створена незалежно від розробок західних вчених, адже усі роботи по розробці електронно-обчислювальних машин як у західних країнах, так і у СРСР були глибоко засекречені. Хронологічний список ЕОМ першого покоління виглядає приблизно так:

- 1952 р. – дослідний варіант ЕОМ БЭСМ (С.Лебедев). Швидкодія – 800 операцій за секунду. Оперативна пам'ять на ртутних лініях затримки;
- 1953 р. – БЭСМ (С.Лебедев, М.Сулим, В.Мельников). Швидкодія – 10 тис. операцій за секунду; “Стрела” (Ю.Базилевський) - триадресна ЕОМ;
- 1954 – 1957 рр. (Б.Рамеев) – "Урал";
- 1958 р. (М.Бруснецов) – єдина у світі ЕОМ з використанням трійкової системи числення “Сетунь”;
- 1959 р. (Г.Лопато, М.Матюхін, І.Брук) – “Минск-1”;
- 1960 р. (С.Лебедев, М.Сулим, М.Шура-Бура та ін.) – М-20.

Швидкодія цих машин не перевищувала 20 тис. операцій за секунду, а ємність оперативної пам'яті – 1 Кбайт.

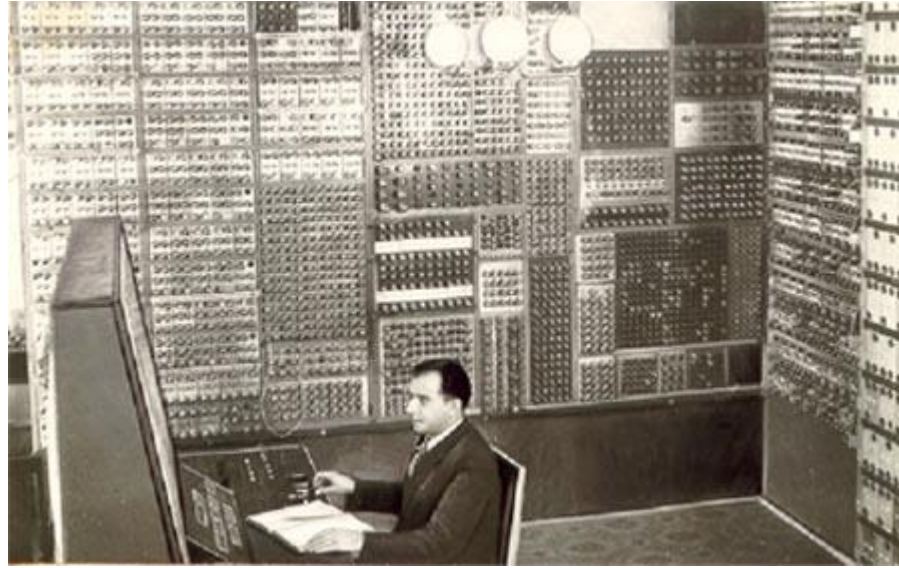


Рис.1.16. Електронно-обчислювальна машина МЕСМ За пультом В.Крайницький.

На цей час припадає становлення обчислювального центру Львівського національного університету імені Івана Франка, де у 1959-1960 рр. встановлено і введено в дію електронну обчислювальну машину "Урал-1". Це була перша машина такого класу на території Західної України. Швидкодія ЕОМ не перевищувала 100 операцій за секунду, а споживана потужність була близько 10 кВт [4].

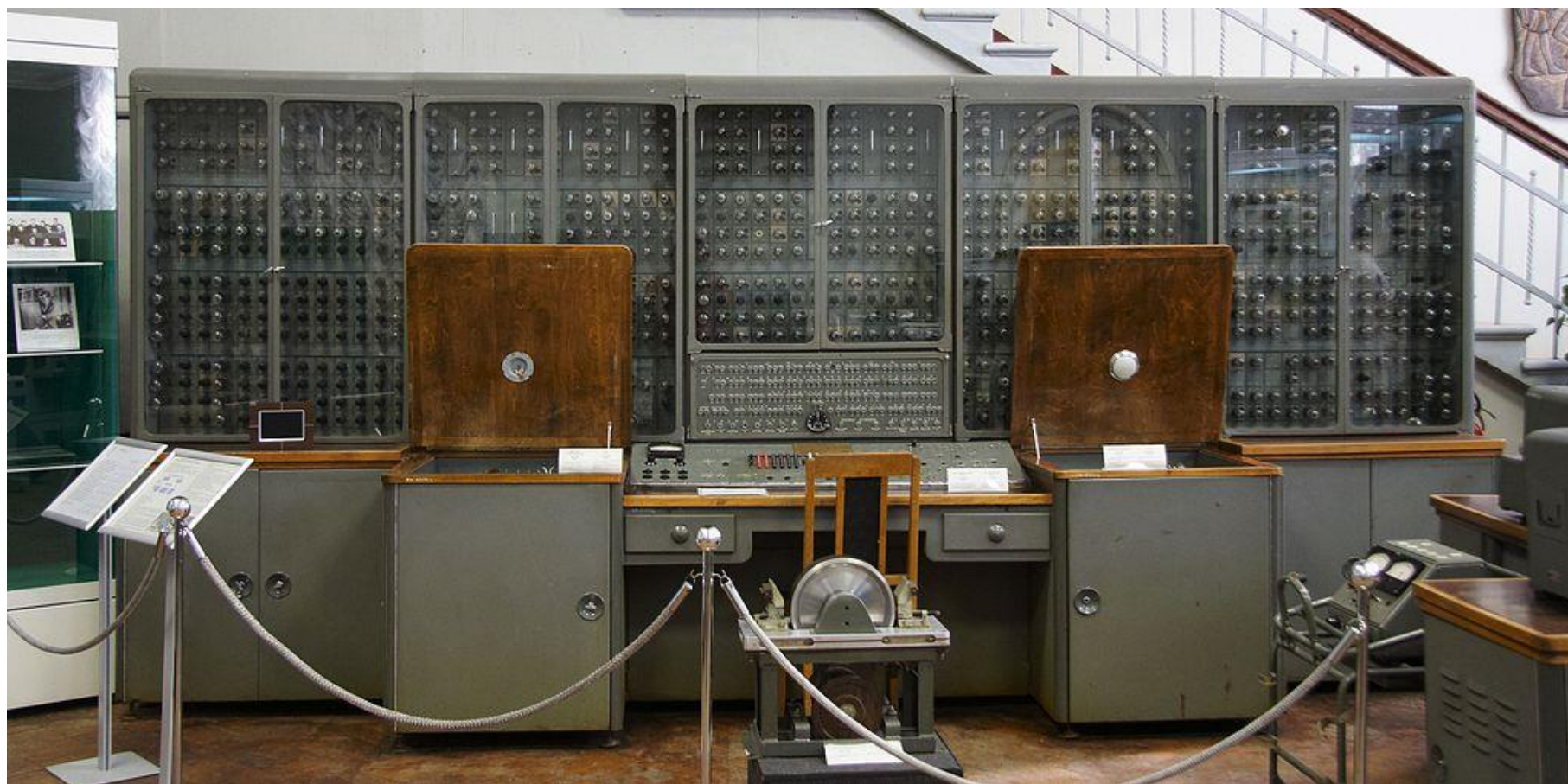


Рис. 1.17. ЕОМ “Урал-1”.

У 1947 р. Волтер Браттейн, Джон Бардін і Вільям Шоклі створили експериментальний зразок транзистора — основного елемента для ЕОМ II покоління.

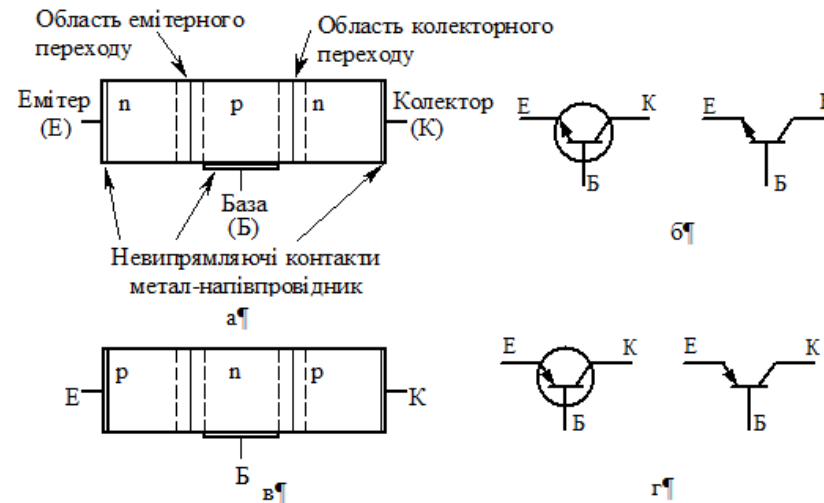
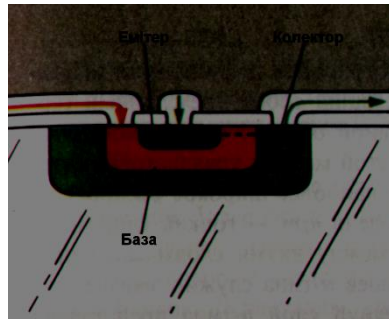


Рис.1.18. Транзистор — элементная база ЕОМ второго поколения

Початок другого покоління ЕОМ (напівпровідникова елементна база) умовно починається з вірменської розробки “Раздан-1”, яка з'явилася у промисловій експлуатації 1961 р. Після неї створено “Минск-2”, “Минск-22”, “Минск-32”, М-220, М-222, БЭСМ-4, Урал-11, -14, -15, -16, “Наири”, “Промінь”, “Мир-1”, “Мир-2”. Машина "Мир-2" мала вмонтовану мову програмування АНАЛІТИК, що давало змогу розв'язувати задачі в аналітичному вигляді. Швидкодія цих машин не перевищувала 80 тис. операцій за секунду (крім ЕОМ "БЭСМ-6") а ємність оперативної пам'яті була в межах 16–32 Кбайти. Завершує ряд ЕОМ другого покоління БЭСМ-6, яку випускали до 1981 р. Її швидкодія була 1 млн. операцій за секунду. Поряд з цими універсальними машинами розробляли низку ЕОМ для керування технологічними процесами (“Київ”, М-6000 та ін.) [5, 8].



Рис.1.19. ЕОМ БЭСМ-6.

В Україні роботи із створення ЕОМ продовжив В.М.Глушков, який після від'їзду С.О. Лебедєва в Москву очолив інститут кібернетики АН УРСР. Наступний короткий перелік (повний список розробок набагато більший) ілюструє розробки київського інституту кібернетики:

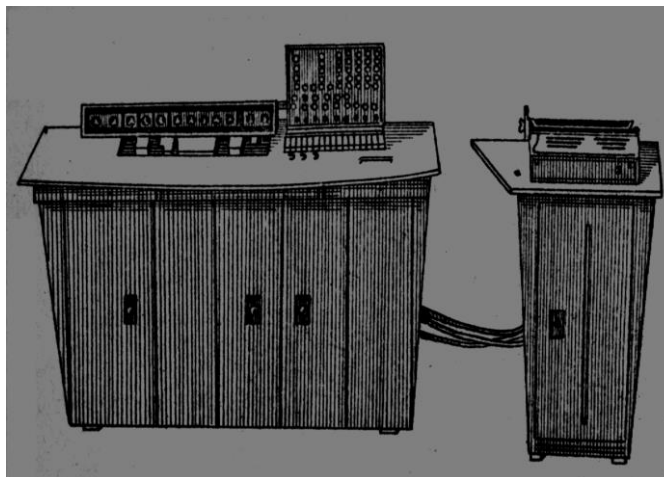
1961 р. — створена напівпровідникова ЕОМ **Днепр** для керування технологічними процесами. Ця ЕОМ випускалась 10 років. Дві ЕОМ **Днепр** використовувались у Байконурі для керування великим екраном під час польоту **Союз-Аполон** (основна частина розробок інституту кібернетики була скерована на створення керуючих ЕОМ різного [в основному спецпризначення]);



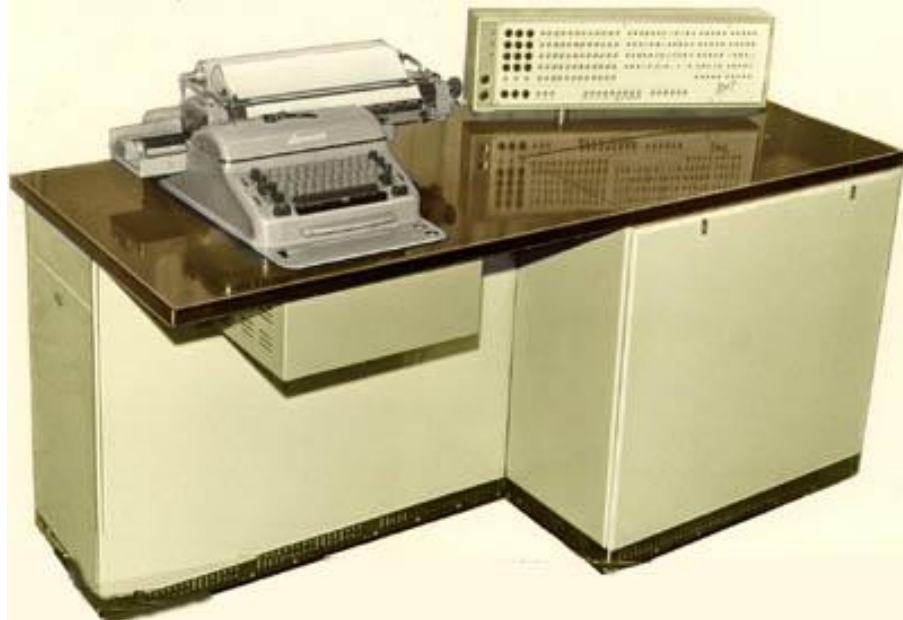
Рис.1.20. ЕОМ **Днепр** - перший вітчизняний напівпровідниковий керуючий комп'ютер широкого призначення. В серії з грудня 1961 року. Розробка Обчислювального центру АН УРСР.

1963 р. — створена напівпровідникова ЕОМ **Промінь** для інженерних розрахунків (100 команд, російськомовна мова програмування, вивід на електричну друкарську машинку);

Рис.1.21. ЕОМ **Промінь**



1965 р. — створена напівпровідникова ЕОМ **Мир-1** для інженерних розрахунків (фірма **ІВМ** придбала одну **Мир-1** для того, щоб довести своїм конкурентам, що у СРСР був створений мікропрограмний принцип керування)



1969 р. — створена напівпровідникова ЕОМ **Мир-2** для інженерних розрахунків. На цій ЕОМ вперше у світі були виконані аналітичні обчислення.



Рис.1.22. Електронно-обчислювальна машина **Мир-2**

У 1958 р. в Московському університеті була створена ЕОМ **Сетунь** (П.П. Брусенцов — апаратне забезпечення, Є.А. Жоголев — програмне забезпечення), яка ґрунтувалась на трійковій системі числення. Використання трійкової системи числення і особливої елементної бази (феритових кілець із обмотками — трансфлюксорів) дозволило створити надзвичайно просту архітектуру ЕОМ.

В 1970 р. в Московському університеті була створена ЕОМ **Сетунь-70**, яка мала особливу систему команд (команди не мали фіксованої довжини. Команду можна було скласти з окремих частин — складів адресації і складів операцій). В **Сетунь-70** були реалізовані принципи структурного програмування. Вона викликала величезний інтерес на Заході, але не була запущена у серійне виробництво в СРСР.

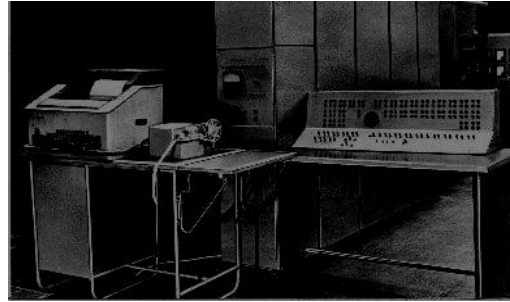


Рис.1.23. Електронно-обчислювальна машина **Сетунь**.

Третє покоління ЕОМ — це не тільки нова елементна база для побудови електронних компонент комп'ютера (інтегральні схеми), а й нова ідеологія, запозичена у фірми ІВМ. З 1972 р. держави — члени Ради Економічної Взаємодопомоги розпочали продукувати так звану серію ЕС ЕОМ: 1010 — Угорщина, 1020, 1021 — Болгарія, 1030 — Польща, 1040 — НДР, 1050, 1060 і проміжні серії (1022, 1035, 1045, 1061 та ін.) — СРСР. Передові розробки радянських вчених були перервані після прийняття ЦК КПРС секретної постанови про заборону власних розробок універсальних ЕОМ і розробку аналогів лише двох ліній:

**IBM 360/370 — серія EC10xx;
PDP 11 — серія CM.**

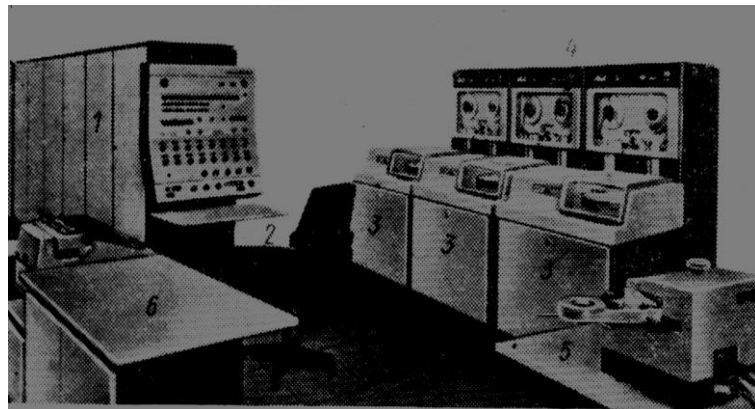


Рис.1.24. Електронно-обчислювальна машина **ЕС 1040**

1 — процесорна стійка, 2 — пульт оператора, 3 — запам'ятовуючий пристрій на магнітних дисках, 4 — цифрові магнітофони, 5 — пристрій читання інформації із перфострічки.

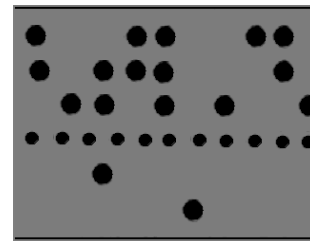


Рис. 1.25. Перфострічка

З цього моменту виникло поняття *архітектура ЕОМ*. Одні автори підручників (Карцев, 1978, Корольов, 1978) під архітектурою ЕОМ розуміють логічну організацію машини, інші (Каган, Каневський, 1979) у поняття архітектури вкладають інженерну реалізацію, тобто сукупність рішень, які реалізують логічні та функціональні можливості ЕОМ. Ми будемо використовувати об'єднане поняття, що дасть змогу повніше зрозуміти механізми функціонування сучасних ЕОМ.

Нова епоха у розвитку ЕОМ розпочалася з появою схем високого ступеня інтегрування, а на їхній базі – персональних ЕОМ. Ще 1959 р. Роберт Нойс, майбутній засновник фірми “Intel”, винайшов спосіб інтегрування всіх необхідних електронних елементів на одній пластинці (те, що сьогодні називаємо чіпами). Однак перший комп'ютер на інтегральних схемах з'явився тільки 1968 р. (фірма Burroughs), а у 1970 р. Маршіан Едвард Хофф (фірма Intel) сконструював перший мікропроцесор Intel-4004, який виконував усі функції центрального процесора великої ЕОМ. На початку 1975 р. з'явився перший комерційний комп'ютер “Альтаір-8800” на базі процесора Intel-8080, а наприкінці цього року Пол Аллен і Білл Гейтс (майбутні засновники фірми Microsoft) створили для нього інтерпретатор з мови Basic, що дало змогу спростити доступ до використання цього і наступних комп'ютерів, зробивши їх власне персональними. Особливого розвитку виробництво персональних комп'ютерів набуло після того, як фірма IBM 1979 р. зрозуміла, що майбутнє за ринком власне таких ЕОМ, і приступила до їхнього випуску. Повноцінний 8-бітовий IBM PC офіційно презентований у 1981 р., і відтоді марка цієї фірми стала стандартом у виробництві комп'ютерної техніки.



Рис.1.26. Мікро-ЕОМ **Altair** .

Після створення фірмою **Intel** у 1971 р. мікропроцесора **I4004** (тактова частота 108 кГц, 2300 транзисторів, 10 мікронна технологія) розпочалися розробки мікро-ЕОМ (**LSI-11**, **microVAX**, **Електроніка-60**, **ДБК-2**). У 1975 р. на мікропроцесорі **I8080** (6000 транзисторів, 10 мікронна технологія) був створений „електронний конструктор” **Altair** (\$397), з якого кожний бажаючий міг зібрати мікро-ЕОМ з 4Кб оперативної пам’яті.

Перші зразки мікропроцесора **I8080** продавались по ціні 300 \$. У 1975 р. фірма **Mos Technologies** випустила мікропроцесор 6502, який продавався по ціні 25\$. На цьому мікропроцесорі у 1976 р. Стефан Возняк і Стівен Джобс створили першу у світі мікро-ЕОМ **Apple I**, яка була орієнтована на некваліфікованого користувача — **персональний комп’ютер (personal computer)**. У 1977 р. розпочався промисловий випуск **Apple II** (процесор 6502, оперативна пам’ять 4Кб, пристрій для читання/запису даних на ГМД, кольорова графіка, електронна таблиця **VisiCalc**).



Рис. 1.27. Персональний комп'ютер **Apple II**

Фірма **Apple** встановила обмеження на ціну своїх ПЕОМ — не більше \$6000 (модель Lisa, яка мала собівартість \$8000 комерційного успіху не мала). Поява на ринку ПЕОМ **Apple II** призвела до небаченого в історії США явища — службовці купували **Apple** за власні кошти для використання їх на службі! У 1977 р. фірма **Commodor International** випустила ПК **PET**, а фірма **Tandy Radio Shack** — **TRS-80** на мікропроцесорі **Z-80**. У 1982 р. фірма **Commodor Business Machines** випустила ПК **Commodor-64**.

У 1981 р. дванадцять інженерів фірми **IBM** під керівництвом Дона Естріджа виставила на загальний огляд свою розробку — **IBM PC** (процесор **i8088** — 16-бітний з 8-бітною шиною даних і тактовою частотою 4.77 МГц, 16Кб [з можливістю розширення до 256Кб] оперативної пам'яті, вбудований Бейсік, один [два] пристрої для роботи з 160Кб гнучкими магнітними дисками, монохромний або кольоровий дисплей **Zenith**, можливість використання кольорового телевізора замість дисплея). Ціна **IBM PC** з монохромним дисплеєм складала \$3000, з кольоровим - \$6000.



Рис. 1.28. Зовнішній вигляд **IBM PC** (фірмове позначення **IBM 5051**)

Архітектура **IBM PC** була оголошена відкритою — фірма **IBM** сподівалася у такий спосіб заощадити кошти на розробці периферійних пристроїв до своєї ПЕОМ — незалежні виробники, маючи документацію про архітектуру **IBM PC**, могли незалежно від фірми **IBM** (і за власні кошти) розробляти до **IBM PC** пристрої друку, дисплеї, пристрої читання/запису даних на магнітні диски. Однак, попри очікуваний ефект, фірма **IBM** отримала несподіваний результат — незалежні виробники, користуючись відкритою архітектурою **IBM PC**, почали виробляти **IBM**-подібні (**IBM PC compatible**) ПЕОМ. Певні зміни в архітектурі **IBM**-подібної ПЕОМ, які робилися для того, щоб уникнути судових переслідувань з боку фірми **IBM**, узгоджувались з **BIOS** (базовою системою вводу-виводу), і для користувача ПЕОМ ставала дуже подібною на **IBM PC**. Скромні можливості **IBM PC** (а отже і низька конкурентноздатність) змусили фірму **IBM** незабаром виставити **IBM PC-XT** (**eXtended arhiTecture**) та **IBM PC-AT** (**Advanced arhiTecture**). Не отримавши очікуваного результату від відкритості архітектури **IBM PC**, фірма **IBM** спробувала закрити архітектуру своєї нової лінії **IBM PS/2** (до речі усі ПЕОМ фірми **Apple** мають закриту архітектуру), однак очікуваного результату це також не дало — світовий ринок вже був наповнений дешевшими **IBM**-подібними ПЕОМ з співмірними або й кращими характеристиками. Скористався з відкритості архітектури **IBM PC** і Радянський Союз — у другій половині вісімдесятих років були розроблені ПЕОМ **ЕС-1840** (Мінськ), **Іскра-1030** (Смоленськ), **Нейрон** (Київ). Багатоплатна конструкція цих ПЕОМ стала причиною їх низької надійності. Особливе місце серед цих розробок займають **Поиск-1**, **Поиск-2** ” (копія **IBM PC-XT**), які вироблялись на Київському виробничому об’єднанні **Електронмаш**, і **Практик**, **ЕС 7978** (копія **IBM PC-XT**), які вироблялись на Канівському електромеханічному заводі **Магніт**. Слід наголосити на високій надійності ПЕОМ **ЕС 7978** і **Практик** — в одній із львівських шкіл **ЕС 7978** пропрацювала понад 10 років без жодного ремонту (у радянські часи Канівський завод **Магніт** виробляв прилади нічного бачення для танків і мав добре відпрацьовану технологію). Причиною низької надійності ПЕОМ **Поиск-2** у першу чергу було використання неякісних мікросхем оперативної пам’яті (білоруського виробництва). В/о **Електронмаш** і завод **Магніт** освоїли випуск ПЕОМ з процесором **I80386**, однак не змогли витримати конкуренції з невеличкими комп’ютерними фірмами, які масово розпочали збирати **IBM**-подібні ПЕОМ з комплектуючих, виготовлених за кордоном (головно у південносхідній Азії).



Рис. 1.29. ПК “Львів”, [персональний комп'ютер](#), розроблений у 1986–1987 роках у [Львівському політехнічному інституті](#) (Конструктор доцент В. Я. Пуйда). Виготовлявся на заводі [ЛОРТА](#).

У 1980 р. на ринку професійних мікро-ЕОМ з'явилась робоча станція **Apollo** (32-бітний мікропроцесор, власна **Unix**-подібна операційна система **Domain OS** з графічним інтерфейсом), яка була набагато дешевшою від великих і міні-ЕОМ (кілька десятків тисяч \$ замість сотень тисяч). Нижче подані результати виконання Фортран-тесту на великій ЕОМ, суперміні-ЕОМ і робочій станції:
IBM-360 -780 сек.; **VAX 11/780** - 667 сек.; робоча станція з мікропроцесором **Motorola 68020** (f-20МГц) - 748 сек.

Поява у 1985 р. 32-бітного мікропроцесора I80386 призвела до появи співмірних по потужності з робочими станціями, але значно дешевших IBM—подібних ПЕОМ.

Сучасний ринок пропонує надзвичайно широкий спектр як персональної, так і високопрофесійної комп'ютерної техніки. Тому для нас важливо знати, які характеристики сучасної ЕОМ є найсуттєвішими у разі вибору відповідного обладнання для виконання певного класу задач. Як і для ЕОМ загального

призначення, це швидкодія процесора (або його продуктивність), ємність і швидкодія оперативної пам'яті та наявність відповідного периферійного обладнання. З появою інтегральних схем пам'яті ситуація постійно змінюється на краще, і сьогодні оперативна пам'ять звичайного персонального комп'ютера **1 чи 2** Гбайти є нормою.

Продуктивність ЕОМ збільшується кількома шляхами. Спочатку (перші два десятиліття) їхня продуктивність зростала завдяки зміні елементної бази (реле, лампи, транзистори, мікросхеми). Сьогодні для виготовлення мікросхем використовують кремній, рідше – германій. Час зміни логічного стану елементарної електронної схеми на кремнієвій основі становить від 350 пікосекунд до 5 наносекунд. Відомо, що виконання операцій (команд) у цифровій обчислювальній машині розділено на кілька етапів (вибирання чи пересилання операндів у пам'ять чи регістри, операції з регістрами тощо). Процес виконання синхронізований відповідно до сигналів, які виробляє спеціальний генератор. Частоту коливань генератора називають тактовою (одному тактові відповідає один період коливання генератора). Повна операція виконується за кілька тактів, кількість яких не завжди збігається з кількістю етапів.

Головний машинний такт у середньому становить 10–15 нс (для об'ємної кремнієвої технології це близько до межі). Перехід від кремнію до арсеніду галію (GaAs) може підвищити тактову частоту на півтора-два порядки (16 МГц – 62,5 нс). Сучасні розробки, що ґрунтуються на плівкових технологіях (товщина шару приблизно три атомних перерізи), дають змогу досягнути тактових частот кілька десятків гігагерц.

Чому неможливо підвищити продуктивність простим збільшенням тактової частоти? Відповідь на це питання така: за 1 нс сигнал проходить 300 мм (швидкість світла). Практично 1 м сигнал проходить за 4–5 нс по коаксіальному кабелю або за 7,0–7,5 нс по смужковому провіднику (за 1 нс 130–250 мм). Якщо окремі

інтегральні схеми (ІС) в пристрої будуть віддалені на 1м, то сигнал запізнюватиметься на 1 такт. Отже, потрібно:

- □ зменшувати розміри, тобто збільшувати ступінь інтеграції на кристалі;
- □ компактно розташовувати ІС на платі ;
- □ виготовляти якомога тонші провідники та зменшувати відстані між ними (багатошарові плати).

Ще один момент – виділення енергії, тобто перегрівання мікросхем, унаслідок чого потрібне охолодження.

Далі – суперечність між швидкодією логічних схем, арифметико-логічних пристроїв і можливостями запам'ятовувальних пристроїв .

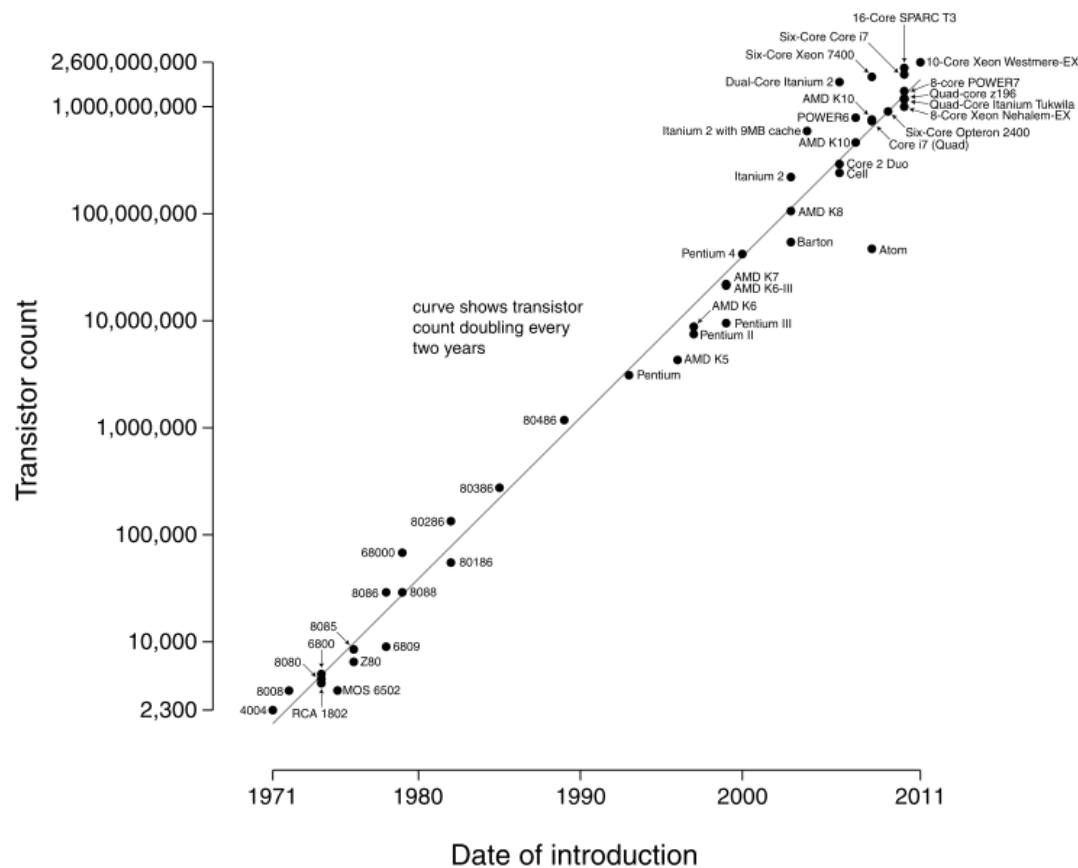
Є також головна семантична суперечність між мовами програмування високого рівня і кодом машинних команд.

Нещодавно комп'ютерний світ відсвяткував 40-річчя появи мікропроцесорів (1971). Перший шістнадцятирозрядний мікропроцесор Intel8086 фірма Intel випустила у 1978 р. а тридцятидво-розрядний Intel 586-й отримав навіть власну назву – Pentium. Pentium і PentiumPro (шосте покоління) містять відповідно 2,3 та 5,5 млн. транзисторів, а відстані між доріжками 1,0 і 0,8 мікрона. Процесор MMX уміщує 4,5 млн. транзисторів. Тактова частота зросла до 300–400 МГц. Швидкодія сягає 200 млн. операцій за секунду. Процесор PentiumII виготовлений за 0,25-мікронною технологією, а PentiumIII (Katmai) – за 0,18-мікронною. Його тактова частота сягає близько 800 МГц і системної шини 200 МГц. У серпні 2000 року оголошено про випуск процесора PentiumIV, тактова частота якого вже перейшла межу 2 ГГц.

У 2000 р. президент компанії Intel Ендрю Гроув був переконаний, що у 2011 році процесор уміщуватиме близько мільярда транзисторів, його швидкість зросте до 100 000 мільйонів (10^{11}) операцій за секунду, а тактова частота – до 10 ГГц. Повідомлення від компанії Intel 2001 р. були ще оптимістичнішими. Їхні лабораторні дослідження давали змогу стверджувати, що тактова частота процесора вже у 2007 р. досягне межі 20 ГГц. Реально на сьогодні маємо тактові частоти порядку 6-8 ГГц, а кількість транзисторів на кристалі перейшла 2 млрд. рубіж.

Закон Мура — [емпіричне](#) спостереження, зроблене в [1965](#) році (через шість років після винаходу [інтегральної схеми](#)), у процесі підготовки виступу [Гордоном Муром](#) (одним із засновників компанії [Intel](#)). Він висловив припущення, що кількість [транзисторів](#) на кристалі [мікросхеми](#) буде подвоюватися кожні 24 місяці. Створивши [графік](#) зростання продуктивності запам'ятовуючих мікросхем, він виявив закономірність: нові моделі мікросхем розроблялися через більш-менш однакові періоди (18-24 міс.) після появи їхніх попередників. При цьому їхня місткість зростала щоразу приблизно вдвічі. Якщо така тенденція продовжиться, уклав Мур, то потужність [обчислювальних пристроїв](#) експоненціально зросте протягом відносно короткого проміжку часу.

Microprocessor Transistor Counts 1971-2011 & Moore's Law



Список літератури до теми

1. Енциклопедія кібернетики у 2 т.: –К., 1973.
2. *Аладьев В.З., Хунт Ю.Я., Шишаков М.Л.* Основы информатики: Учеб. пособие.– М., 1999.
2. *Малиновский Б.М.* История вычислительной техники в лицах.–К., 1991.
3. *Кардаш А.І., Костовський О.М., Рикалюк Р.Є.* 40 років обчислювальному центру університету // Вісник Львів. ун-ту. Сер. прикл. матем. та інформатика, 1999. –Вип.1.–с.
4. *Грубов В.И., Курдан В.С.* Справочник по ЭВМ и аналоговым устройствам. –К., 1977.
5. <http://www.intel.com>.
6. <http://www.motorola.com>.
7. <http://www.icfst.kiev.ua/museum/>

Крім **IBM**-подібних ПЕОМ та ПЕОМ фірми **Apple** випускались ПЕОМ інших ліній. Слід згадати **Commodor** (фірми **Amiga**), **Atari** (фірми Atari), **ZX-Spectrum** (фірми **Sinclair Research Ltd**) та його радіолюбительські аналоги, зроблені у Львові, **ПК-Львів** (розробник В.Пуйда, Львівський політехнічний інститут), **Правець-8**, **БК-0010**, **Корвет**, **Агат**. Всі перераховані ПЕОМ випускались в одному корпусі з клавіатурою, використовували 8-бітні мікропроцесори різних виробників і були непристосовані до модернізації, саме тому з появою більш потужних мікропроцесорів ці ПЕОМ просто зійшли з арени.

Для ілюстрації розвитку мікропроцесорів (і ПЕОМ) у таблиці подані основні характеристики мікропроцесорів, які використовувались для побудови ПЕОМ

Мікропроцесор	Розрядність	Швидкодія (тактова частота) МГц	Назва ПЕОМ
Z80A	8 біт	6-8	ZX-Spectrum, Robotron, Arat
Intel 8088	8/16 біт	4.77	IBM-PC, IBM-PC/XT
K1810BM 88 (аналог I8088)	8/16 біт	4.77	Поиск-1
K1810BM 86 (аналог I8086)	16 біт	8	Поиск-2, Практик, ЕС-7978
Intel 80286	16 біт	6-25	IBM-PC/AT
Motorola 68000	16 біт	6-16	Apple, Macintosh
Intel 80386	32 біт	16-40	IBM-PC.386
Motorola 68020	32 біт	16-20	Macintosh***** Беста (виробн. СРСР)
Intel 80486	32 біт	16-120	IBM-PC.486
Motorola 68030	32 біт	16-**	Macintosh*****
Pentium I	32/64 біт	60-133	IBM-PC.PI
Motorola 68040	32/64 біт	16-60	Macintosh*****
Pentium II	32/64 біт	233-500	IBM-PC.PII
Power PC	32/64 біт	100-133	Power Macintosh
Pentium III	32/64 біт	450-1000	IBM-PC.PIII
Power PC G3	32/64 біт	350-500	I Mac
Power PC G4	32/64 біт	400-733	Power Macintosh
Pentium IV	32/64 біт	3700	IBM-PC.PIV

Примітка: ***** означає фірмове позначення моделі

Сучасні ПЕОМ можна розділити на дві групи:

ПЕОМ від фірми **Apple (Macintosh)**

IBM-подібні ПЕОМ від різних виробників (у тому числі українських).

ПЕОМ від фірми **Apple (Macintosh)** завжди відрізнялись високими споживчими характеристиками, які забезпечувались високим рівнем схемних рішень та високою якістю програмного забезпечення. Одному з авторів доводилось спостерігати у роботі операційну систему **Mac OS 5** з графічним інтерфейсом, яка вантажилась з 3,5” дискети (порівняйте 700 Кб з мінімумом у 100 Мб

для **MS-Windows 95**). Політика закритої архітектури (**Apple** та **Macintosh** випускають лише 4 заводи в світі) дозволяла випускати ПЕОМ лише високої якості. Розробники програмного забезпечення, які бажають писати програми для **Apple**, повинні пройти обов'язкову сертифікацію (у такий спосіб забезпечується високий рівень програмного забезпечення для **Apple**).

1.4. Типи сучасних ЕОМ

Електронно-обчислювальна машина (ЕОМ, англ. computer [обчислювач]) — електронний пристрій для виконання обчислення з використанням електричних сигналів. За використовуваною елементною базою ЕОМ поділяють на п'ять поколінь:

I покоління — лампові;

II покоління — транзисторні;

III покоління — на інтегральних схемах низького ступеню інтеграції;

IV покоління — на інтегральних схемах високого ступеню інтеграції;

V покоління — на інтегральних схемах надвисокого ступеню інтеграції.

Однак, у цьому поділі відсутні релейні ЕОМ (**Z3-Z5, Марк-1**) та на магнітних елементах з трьома станами (**Сетунь**).

Сучасні ЕОМ поділяють на:

суперЕОМ (**Cray, IBM BlueGene, Эльбрус**);

робочі станції;

серверні ЕОМ;

персональні ЕОМ, у тому числі стаціонарні, міні, переносні, планшетні, ультрамобільні, кишенькові.

Термін суперЕОМ був введений в обіг через нестримний потяг людей до надзвичайного — суперзірка, суперхіт, суперкнижка, суперфільм, супер, супер, супер. Чіткого визначення терміну суперЕОМ немає. Вважають, що суперЕОМ — це ЕОМ, яка має технічні параметри (швидкість обчислень, обсяг оперативної пам'яті, обсяг дискової пам'яті) значно вищі, ніж доступні для загалу ЕОМ і ПЕОМ. В останній чверті минулого сторіччя до суперЕОМ зараховували ЕОМ **Cray-1 — Cray-4** (США) і **Эльбрус-1 — Эльбрус-3, ПС-2100** (Радянський Союз).



а)



б)

Рис. 1.37. СуперЕОМ (а — **PC-2100**, б — **Cray-2**)

Сьогодні суперЕОМ виробляють такі фірми як **IBM, HP, NEC, Sun, SGI, Dell**. Сучасні суперЕОМ є багатопроцесорними системами, які побудовані на мікропроцесорах фірм **Intel (CISC-архітектура), AMD, IBM, DEC (RISC-архітектура)** та ін. У таблиці подано список першої десятки суперЕОМ (станом на червень 2006 р.)

Місце	ЕОМ	Кількість процесорів	Країна
1	IBM BlueGene/L - eServer Blue Gene Solution	131072	США
2	IBM eServer Blue Gene Solution	40960	США
3	IBM eServer pSeries p5 575 1.9 GHz	12208	США
4	SGI Altix 1.5 Ghz	10160	США

Місце	EOM	Кількість процесорів	Країна
5	Tera-10 - NovaScale 5160, Itanium2 1.6 GHz	8704	Франція
6	Dell Thunderbird - PowerEdge 1850, 3.6 GHz	9024	США
7	NEC/Sun Sun Fire X64 Cluster, Opteron 2.4/2.6 Ghz,	10368	Японія
8	IBM eServer Blue Gene Solution	16384	Німеччина
9	Cray Inc. Red Storm Cray XT3, 2.0 GHz	10880	США
10	NEC Earth-Simulator	5120	Японія

Більш докладну інформацію про сучасні суперЕОМ можна отримати за адресою: **www.top500.org**

СуперЕОМ 70-х і 80-х років минулого сторіччя будувались з використанням векторних процесорів. На початку 90-х років відбувся перехід до використання великої кількості паралельно з'єднаних скалярних процесорів. Наприкінці двадцятого сторіччя для створення суперЕОМ стали широко використовувати кластери, які забезпечують високу продуктивність при мінімальній вартості.

В якості операційних систем для суперЕОМ використовують Unix-подібні ОС (**Unix, Linux, Solaris** тощо).

Робочі станції належать до професійних ПЕОМ. Значно менші обсяги виробництва робочих станцій, значно вищі їх технічні характеристики зумовлюють істотно вищі ціни на робочі станції порівняно з цінами на персональні ЕОМ. На рис. 1.38 зображені робочі станції фірм **Silicon Graphics** і **Sun Microsystems**



а



б



в



г

Рис. 1.38. Робочі станції
а - Silicon Graphics Fuel, б - Silicon Graphics Onyx 4, в - Sun Ultra 40, г - Sun Ultra 45

На серверні ЕОМ покладають завдання опрацювання запитів від багатьох ПЕОМ у мережі.

Залежно від завдань, які покладаються на сервер, їх можна розділити на сервери:

початкового рівня (файл-сервер, Інтернет-сервер, термінальний сервер, сервер баз даних для малих та середніх робочих груп [від кількох ПЕОМ до кількох десятків ПЕОМ]) ;

середнього рівня (файл-сервер, Інтернет-сервер, термінальний сервер, сервер баз даних, сервер програмного забезпечення для корпорації [від кількох десятків ПЕОМ до кількох сотень ПЕОМ]) ;

верхнього рівня (сервер розподілених баз даних, центральний корпоративний термінальний сервер, **SQL**-сервер, **Exchange**-сервер).

Як уже згадувалося, персональні ЕОМ сьогодні поділяють на стаціонарні (рис. 1.39-1.40), переносні (рис. 1.41-1.43) і кишенькові (рис. 1.44-1.45).

Стаціонарні ПЕОМ складаються із системного блоку, дисплея, клавіатури і мишки до яких за потреби додають інші пристрою вводу/виводу даних (сканери, пристрою друку, Веб-камери, голосники і т.ін.). Останнім часом виробники стали пропонувати системні блоки із зменшеними габаритами, що дозволяє закріплювати їх на задній стінці рідкокристалічного дисплея (**Imac G5**) або у підставці рідкокристалічного дисплея (рис. 1.40)



Рис. 1.39. Стаціонарні ЕОМ



Рис. 1.40. Стационарна ПЕОМ **LLUON LXMA52-1**



Рис. 1.41. Фотографія переносної ПЕОМ

Планшетна ПЕОМ (англ. **Tablet PC**) повинна мати розмір товстого журналу, сенсорний екран, голосовий ввід, бездротовий і потужність сучасної ПЕОМ. Їх розробки ведуться кілька останніх років.



Рис. 1.42. Планшетна ПЕОМ **Pepper Pad**

Для уведення алфавітно-цифрових даних використовують рукописний ввід або вбудовану клавіатуру (не усі моделі планшетних ПЕОМ мають клавіатуру). Для обміну інформацією з стаціонарними ПЕОМ використовують бездротові інтерфейси, інтерфейс **USB**, локальні мережі. Планшетні ПЕОМ працюють під керуванням ОС **Linux (Pepper Pad)** або **Microsoft Windows**. Подальшим розвитком ідеї планшетних ПЕОМ можна вважати розробки ультрамобільних ПЕОМ (**UMPC**). На рис. 1.43 зображено ультрамобільну ПЕОМ **OQO model1**



Рис.1.43. Ультрамобільна ПЕОМ **OQO model1**

Усі кишенькові ПЕОМ обладнані сенсорними дисплеями з функцією рукописного вводу. До них можна приєднати клавіатуру (складану або гнучку), деякі моделі мають вбудовану клавіатуру зменшених розмірів. На рис. 1.44 зображена кишенькова ПЕОМ фірми **Asus**



Рис. 1.44. Кишенькова ПЕОМ Asus MyPal A600

Поєднання в одному пристрої кишенькової ПЕОМ і мобільного телефону призвело до появи комунікаторів — ви можете розмовляти по телефону або працювати з цим пристроєм як з кишеньковою ПЕОМ. На рис. 1.45 зображені комунікатори.



Рис. 1.45. Кому́нікатори, планшети

Тема 1. Загальні принципи побудови ЕОМ

Почнемо з визначення деяких понять.

- ЕОМ (ЕЛЕКТРОННА ОБЧИСЛЮВАЛЬНА МАШИНА) – це, мабуть, найпоширеніше поняття до зовсім недавнього часу. Спочатку воно стосувалося перших електронних автоматичних пристроїв для опрацювання інформації. Однак з часом на ЕОМ почали щораз більше накладати функції інформаційного плану: зберігання, шукання, сортування, опрацювання інформації (в тім числі графічної, текстової, образної (передавання зображень), звукової). Розрізняють спеціалізовані ЕОМ; комплекси ЕОМ; універсальні ЕОМ; міні-, мікро-ЕОМ.

- ОС – обчислювальна система – пристрій, що сприймає інформацію у вигляді даних, які можуть бути зображені у аналоговій або цифровій формі, зберігає дані, опрацьовує їх з великою швидкістю, пересилає дані всередині системи через лінію зв'язку і видає результат цих дій як інформацію. (Не треба плутати з поняттям операційна система, що також має скорочене позначення ОС). Обчислювальні системи можуть бути:

- аналоговими (безперервної дії);
- цифровими (дискретними);
- гібридними.

- Комп'ютер (англ. *computer* – обчислювач) – спочатку в англomовних країнах, а тепер майже всюди, назва електронної обчислювальної машини. Головно слугує для передавання, опрацювання і зберігання інформації. Найбільше вживана форма випуску – персональні комп'ютери (ПК).

1.1. Поняття про інформацію

Про сучасні комп'ютери, їхню будову, можливості та принципи функціонування йтиметься дещо пізніше, спочатку ж розглянемо, що таке інформація і як її можна відображати в ЕОМ. Слово *інформація* походить від лат. *informatio*, що означає роз'яснення, виклад, обізнаність. Це одне з найзагальніших понять науки. Поняття інформації – одне з головних понять кібернетики (подібно до поняття енергії у фізиці). Розділ науки, присвячений технічним проблемам

інформації (окрім семантичних та прагматичних), називають *теорією інформації*. Різні способи зображення інформації, спеціально пристосовані для конкретних випадків, пов'язаних з передаванням, зберіганням та опрацюванням інформації, розглядає теорія кодування.

У комп'ютері маємо справу з дискретними повідомленнями, або машинними одиницями інформації. Насамперед постає питання, як виразити, поміряти і передати інформацію? Вперше цю задачу розв'язано у працях Клода Шеннона [1], де запроваджено поняття кількості інформації. Дещо спрощено схема викладу цього поняття може бути така.

Нехай нам потрібно скласти слово як комбінацію з букв алфавіту. Алфавіт містить P букв (чи символів), а слово – x символів. Очевидно, таких слів можна скласти N (правда, невідомо, чи всі вони будуть мати мовний зміст):

$$N=P^x \text{ (кількість розміщень).} \quad (1.1)$$

Припустимо, що кількість інформації у якомусь повідомленні пропорційна до його довжини. Тоді з (1.1) після логарифмування отримаємо

$$\log_a N = x \log_a P \quad (1.2)$$

Величину $x \log_a P$ визначимо як кількість інформації I . Знайдемо найменше I , яке можна було б прийняти за одиницю інформації. Очевидно, це буде при $P=2$, бо при $P=1$, $I=0$. Отже,

$$I_{\min} = x \log_a 2. \quad (1.3)$$

Найпростіше слово складається з однієї букви, тоді

$$I = \log_a 2. \quad (1.4)$$

Це і є кількість інформації найпростішого слова.

Вибір основи логарифма – довільний. Зважаючи на те, що в обчислювальних системах головно послуговуються двійковою арифметикою, прийемо основу логарифма $a=2$. Тоді

$$I = \log_2 2 = 1. \quad (1.5)$$

Ця одиниця інформації у двійковому алфавіті називається *біт* і може мати значення 0 і 1 (походить від англ. **binary digit** і, як стверджує К. Шеннон [1], це слово запропонував уживати відомий американський вчений-статистик Джон Тьюкі). Отже, кількість інформації у будь-якому слові двійкового алфавіту завжди дорівнює кількості бітів у ньому, тобто

$$I = x \log_2 2 = x, \quad (1.6)$$

а кількість можливих повідомлень N легко визначити з (1.2):

$$\log_2 N = x \log_2 2, \text{ тобто } N = 2^x. \quad (1.7).$$

Ще раз зазначимо, що таке поняття кількості інформації виникло з задач теорії зв'язку і, по суті, застосовне саме до них.

Приклад: Кількість можливих слів, які містять три біти, є 8. Ось вони: 000, 001, 010, 011, 100, 101, 110, 111. Це не що інше, як числа вісімкової системи числення.

Кількість можливих слів, які містять чотири біти $N = 2^4 = 16$. Це такі комбінації: 0000, 0001, 0010, 0011, 0100, 0101, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, тобто числа шістнадцяткової системи числення.

Біт є найменшою одиницею вимірювання інформації. На практиці частіше використовують похідні одиниці: *байт*, *Кбайт*, *Мбайт*, *Гбайт*, *Тбайт*. Між ними є такі співвідношення:

1 *байт* = 8 *бітів* (може бути $2^8 = 256$ різних байтів)

1 *Кбайт* = 2^{10} *бітів* = 1024 *байти* (скорочено позначають К)

1 *Мбайт* = 1024 *Кбайти*

1 *Гбайт* = 1024 *Мбайти*

1 *Тбайт* = 1024 *Гбайти*

До машинних одиниць інформації належать також: слово, запис, блок, файл. Деякі машинні одиниці мають аналоги з натуральними одиницями інформації: розряд, символ, поле, масив, запис.

1.2. Основи алгебри логіки

Усі пристрої ЕОМ складаються з елементарних логічних схем. Їхнє функціонування ґрунтується на законах і правилах алгебри логіки, яка оперує двома поняттями: істина і фальш. На честь її винахідника, англійського математика Джорджа Буля, алгебру логіки ще називають булевою алгеброю. Основу цієї алгебри становлять дві бінарні операції (*кон'юнкція* та *диз'юнкція*) і одна унарна

(заперечення). Крім цих трьох, вводять і інші, однак доведено, що будь-яку з них можна виразити за допомогою формули, у якій використано тільки три базові. (Наприклад, для функції від двох змінних таких логічних функцій є 16).

Мовою алгебри логіки будь-яку функцію зображають у вигляді таблиці відповідності всіх можливих логічних змінних та вихідних логічних функцій. Це так звана *таблиця істинності*.

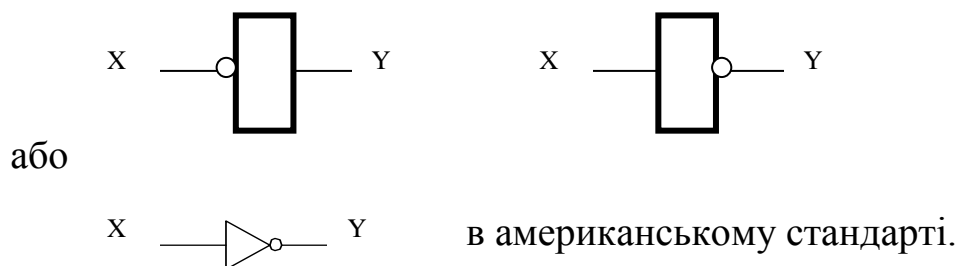
Розглянемо головні логічні функції.

Логічне заперечення. Позначають НЕ.

Означення. Логічне заперечення НЕ змінної X є логічна функція Y , яка істинна тільки тоді, коли X хибна, і навпаки. Інша назва: *інверсія*.

У символах алгебри логіки записують: $Y = \bar{X}$.

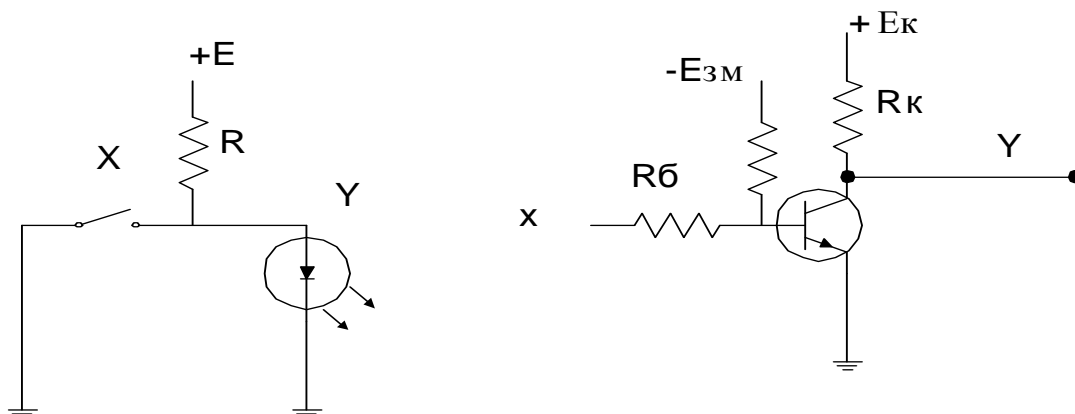
Графічно на схемах позначають кружечком на вході, чи виході логічного символу (у системі ЄСКД):



Таблиця істинності:

X	0	1
Y	1	0

Схемна реалізація – інвертор

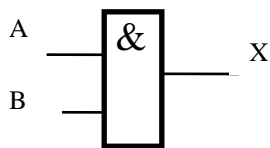


Логічне множення. Позначають І.

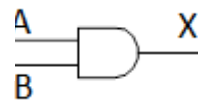
Означення. Логічним множенням двох змінних A і B є логічна функція X , яка істинна тільки тоді, коли одночасно істинні вхідні змінні. Інша назва: *кон'юнкція*.

Запис: $X = A * B$, або $X = A \wedge B$.

Графічне зображення:



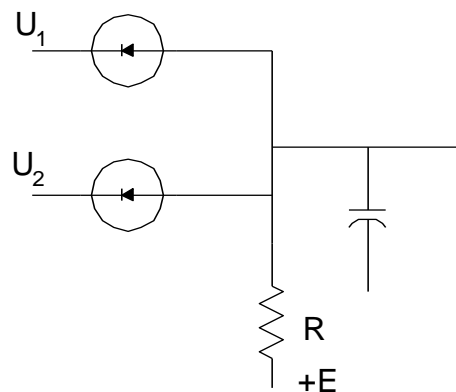
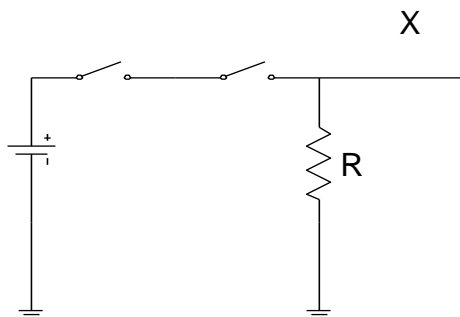
або у американському стандарті



Таблиця істинності:

A	0	0	1	1
B	0	1	0	1
X	0	0	0	1

Схемна реалізація:



Логічне додавання. Позначають АБО.

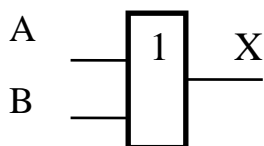
Означення. Логічною сумою змінних A і B є логічна функція X , яка істинна, якщо хоча б одна із вхідних величин істинна.

Інша назва: *диз'юнкція*.

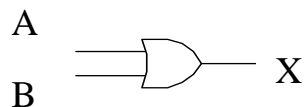
Запис:

$X = A + B$, або $X = A \vee B$.

Графічне зображення:



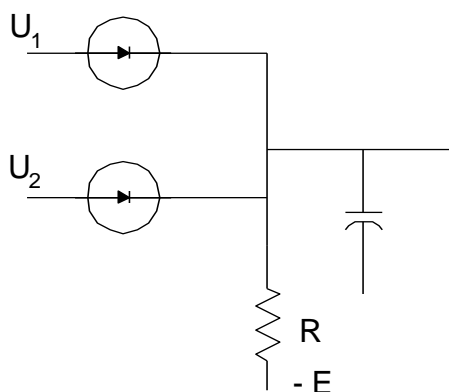
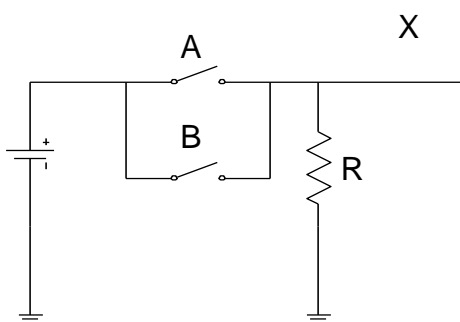
або у американському стандарті



Таблиця істинності:

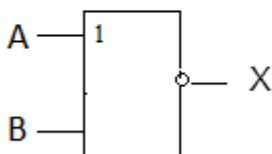
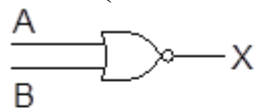
A	0	0	1	1
B	0	1	0	1
X	0	1	1	1

Схемна реалізація:

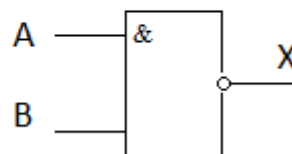
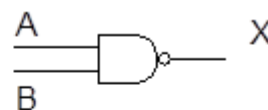


У реальних схемах символічні ключі замінюють діодами, транзисторами чи просто використовують напilenі *p-n* - переходи у схемах більшої інтеграції. Схеми також можуть об'єднуватись відповідно до потреби реалізації певної логічної функції. Наприклад, дуже часто застосовують елементи

АБО-НЕ (елемент Пірса)



чи І-НЕ (елемент Шеффера):



Функції, утворені із логічних змінних, можна перетворювати відповідно до правил або законів алгебри логіки.

Приклад.

1. Логічний добуток будь-якого аргументу на нуль завжди дорівнює нулю (використовують для "маскування" вхідних змінних).
2. Логічний добуток будь-якого аргументу на 1 дорівнює значенню самого аргументу.
3. Логічний добуток аргументу з його інверсією дорівнює нулю.
4. Правила де Моргана (закони інверсії) :

$$\overline{A * B} = \bar{A} + \bar{B}$$

$$\overline{A + B} = \bar{A} * \bar{B}$$

і т. д.

Ці правила і закони застосовують для спрощення логічних функцій і зведення їх до вигляду, що полегшує схемну реалізацію. Для забезпечення мінімізації логічних рівнянь використовують запис вхідних даних у вигляді діаграм Карно (Карнау).

На рисунку нижче показана діаграма Карно для трьох змінних.

	B		\bar{B}	
A	6	7	5	4
\bar{A}	2	3	1	0
	C		\bar{C}	

Кожна клітинка діаграми відповідає логічному добутку прямого чи інверсного значення змінних, які присвоєно стовпцеві або рядкові, на перетині яких вона розміщена. Наприклад, клітинка з номером 7 є на перетині рядка зі значенням змінної A і стовпця зі значенням змінних B і C та відповідає логічному добутку ABC , клітинка з номером 0 відповідатиме значенню $\bar{A} \bar{B} \bar{C}$. В діаграмах Карно значення змінних присвоюються так, щоб сусідні клітинки відрізнялися між собою тільки на одну змінну. Після цього згідно з описом логічної функції кількох змінних у клітинки записують значення логічних добутків з таблиці істинності. Якщо добутки є у сусідніх клітинках, то з загального виразу можна вилучити одну змінну, яка трапляється у прямому та інверсному кодах. Якщо добутки утворюють квадрат, то з загального виразу можна вилучити дві змінні.

1.3. Елементна база ЕОМ

Головні логічні елементи та вузли

Елементами ЕОМ називають пристрої, які виконують логічні функції, запам'ятовують інформацію, перетворюють її, а також формують та підсилюють сигнали.

З деякими найпростішими елементами ми вже ознайомлені. Це інвертор, пристрій для логічного додавання і множення. Відомо також, що використовуючи їхню різноманітну комбінацію, можна будувати складні логічні схеми.

Головним елементом пам'яті ЕОМ є **тригери** - логічні пристрої, які мають два стійкі стани. Для перемикання тригерів з одного стану в інший використовують вхідні логічні схеми. За способом перемикання розрізняють такі тригери:

- *RS*-тригери з роздільним установленням 0 і 1;
- *D*-тригер із затримкою;
- *JK*-тригери (універсальні);
- *T*-тригери (лічильні);

Ці назви походять від перших букв вхідних сигналів: *S* (set – встановити); *R* (reset – вимкнути); *T* (toggle – релаксатор); *J* (jerk – різко увімкнути); *K* (kill – різко вимкнути); *D* (delay – затримка). Вихідний сигнал тригера прийнято позначати буквою *Q*.

***RS*-тригери з роздільним установленням 0 та 1**

Найпростіший запам'ятовувальний елемент має два входи: *S* і *R*. У разі комбінації сигналів $S=1$, $R=0$ тригер буде встановлений в одиничний стан, тобто $Q=1$.

Якщо ж комбінація вхідних сигналів $S=0$, $R=1$, то тригер встановиться в нуль, $Q=0$. Комбінація $S=0$, $R=0$ залишить тригер у попередньому стані, а от комбінація $S=1$, $R=1$ буде невизначеною, отже, забороненою для такого елемента ($Q=\bar{Q}=0$).

Функціонування *RS*-тригера описує вираз

$$Q(t+1) = S(t) \vee Q(t)R'(t),$$

де $S(t)R(t)=0$, t – момент часу, який передуює зміні стану. (Тут і надалі в тексті позначення інверсії $\bar{Y} \equiv Y'$).

Такі тригери реалізують на вже відомих нам логічних елементах І-НЕ, АБО-НЕ, з'єднуючи їхні входи і виходи навхрест. На схемах їх зображають прямокутником, у якому з лівого боку позначають вхідні сигнали S , R , а з правого – вихідні. Вхід S називають установлювальним, а вхід R – скидальним. Приклади таких RS -тригерів показано на рис.1.1.

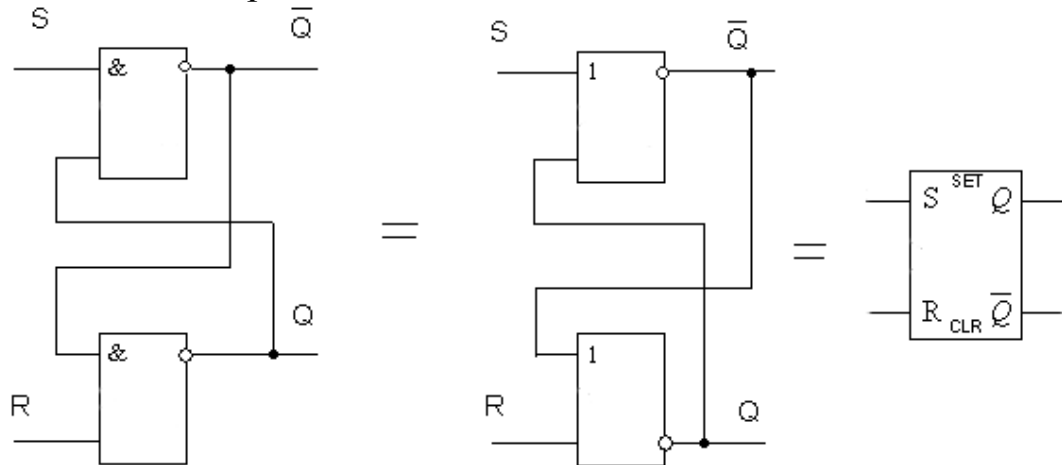


Рис. 1. 1. RS - тригери на базі елементів І та АБО

Іноді вводять синхронізацію зміни стану тригера (позначають вхід C (clocking – синхронізація)). При $C=1$ тригер перемикається за законом RS -тригера, при $C=0$ він зберігає попередній стан.

D -тригер (затримки)

Такий тригер має один інформаційний вхід і вхід для синхроімпульсу. Головне призначення такого тригера – затримка і збереження сигналу, який подають на вхід за умови $C=1$. На рис.1.2 показана схема синхронізованого двотактного D -тригера, який затримує сигнал на один період; і його описує формула:

$$Q(t+1) = D(t).$$

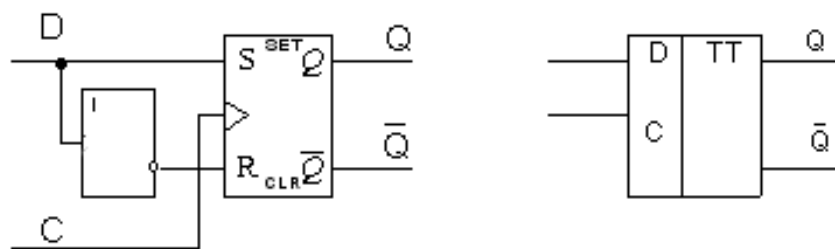


Рис. 1. 2. Синхронізований D -тригер

Різновидом D -тригера є DV -тригер (V -value – вентиль), у якому через вхід керування V дозволено перемикання, або тригер не реагує на перемикання при $V=0$.

JK -тригер (універсальний)

Працює за принципом RS -тригера, проте в ньому комбінація $J=K=1$ не заборонена, у цьому разі він змінює свій стан на протилежний до того, у якому був. Переважно його реалізують за двоступеневою схемою (див. рис.1.3).

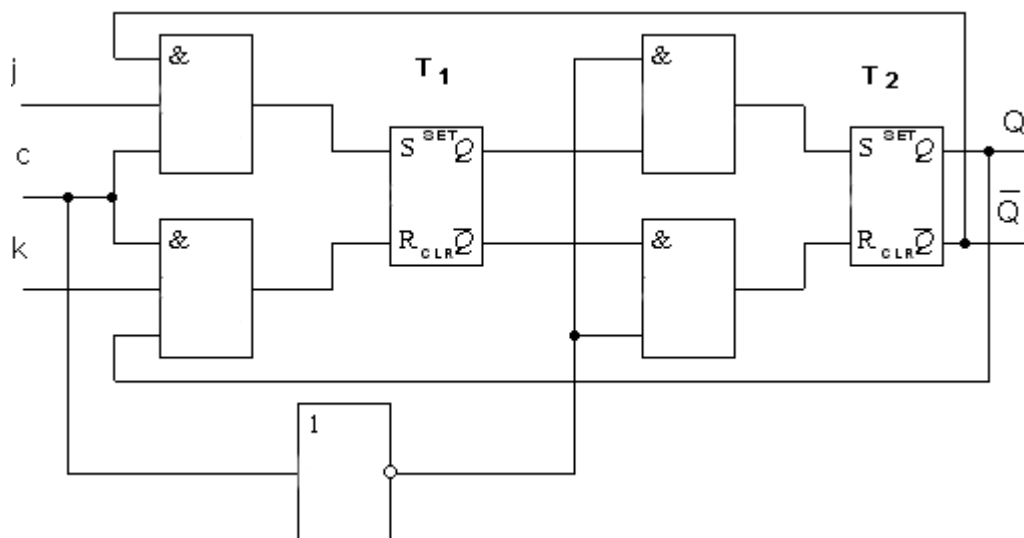
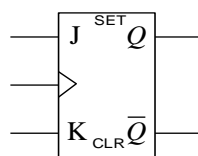


Рис. 1. 3. Схемна реалізація JK -тригера

На схемах позначають так:



При $C=1$ відбувається записування інформації в $T1$, при $C=0$ – переписування інформації з $T1$ в $T2$. Функція переходів JK за умови $R=S=1$ має вигляд

$$Q(t+1) = Q(t)K'(t) \vee Q'(t)J(t).$$

T -тригер (лічильний)

Цей тригер змінює свій стан у разі надходження кожного вхідного імпульсу. Він може бути реалізований на базі JK -тригера. При значеннях змінних $J=K=1$ сигналом C можна змінювати стан тригера. Позначають як на рис. 1. 4.

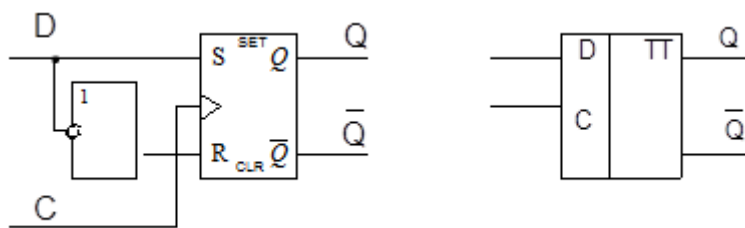


Рис. 1. 4. Схемна реалізація лічильного тригера

T-тригер реалізує таку функцію:

$$Q(t+1) = Q(t)T'(t) \vee Q'(t)T(t).$$

За допомогою описаних логічних пристроїв можна будувати різні блоки чи пристрої ЕОМ. До них належать: регістри, лічильники, шифратори, дешифратори, мультиплексори, суматори та арифметико-логічні пристрої як цілочислової так і плаваючої арифметики.

Мультиплексор – пристрій, який виконує передавання сигналів від однієї з вхідних ліній у вихідну. Вхідну лінію вибирають кодом, який надходить на входи керування мультиплексора. Уживається також інша назва – комутатор (рис. 1. 5).

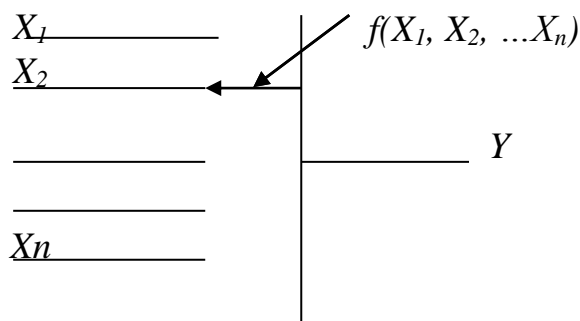


Рис. 1. 5. Схема комутування вхідних сигналів X на вихідну лінію Y за допомогою функції $f(X_1, X_2, \dots, X_n)$

Регістр – це вузол ЕОМ, який виконує тимчасове збереження та перетворення інформації. Регістри будують на основі тригерних схем. Кількість тригерів визначає розрядність слів, які записують чи зберігають у регістрі. Регістри є послідовні і паралельні, одно- і двотактні, зсувні і перетворювальні. Розглянемо кілька прикладів схем регістрів. Побудуємо регістр, наприклад, трирозрядний. Очевидно, що для цього треба мінімум три тригери, які будемо використовувати для запису кожного розряду.

Використаємо звичайні *RS*-тригери. Трохи добудуємо схему ліворуч і праворуч. Ліворуч організуємо синхронний запис

інформації, а праворуч – формування прямого та оберненого коду. Перед записом інформації всі тригери сигналом керування R поставило в нуль $\Rightarrow (Q=0)$. Запис у тригери виконується за тактовим імпульсом Ti_1 . Власне ці два сигнали (R і Ti_1) визначають тип регістра, який називають двотактовим паралельної дії (рис. 1. 6).

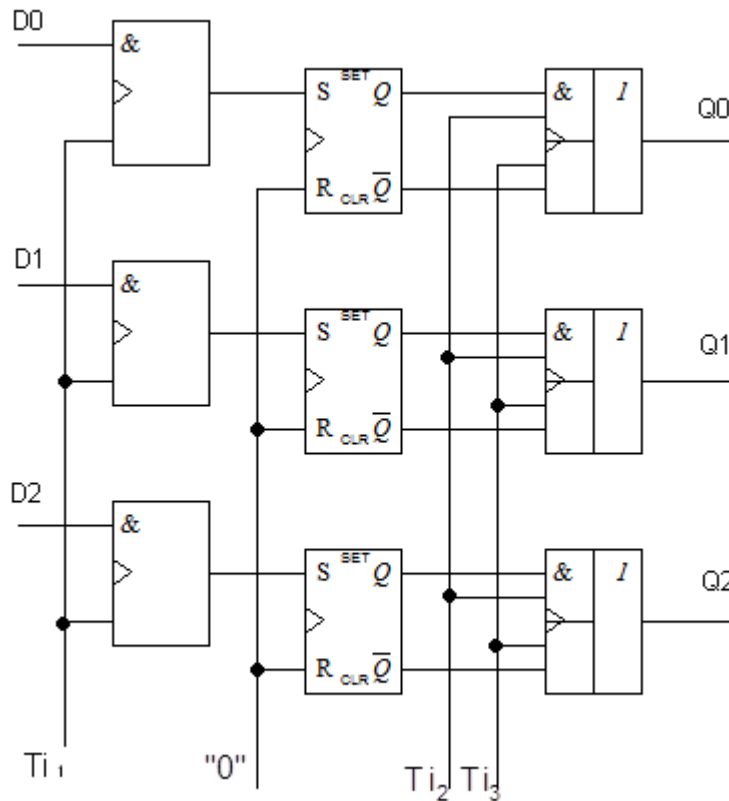


Рис. 1.6. Двотактовий регістр паралельної дії

Код регістра видають за допомогою другого і третього тактових імпульсів: Ti_2 - прямий код, Ti_3 – обернений код.

Якщо на вхід можна подавати парафазний код (тобто вхідне значення подають у прямому й оберненому коді), то відпадає потреба у такті установлення в "0", тобто отримаємо одноктактний паралельний регістр.

Регістр є дуже зручним пристроєм для "зсування" інформації праворуч чи ліворуч або перетворення послідовного коду у паралельний (рис. 1. 7). Для цього тригери регістра потрібно з'єднати послідовно через невелику затримку.

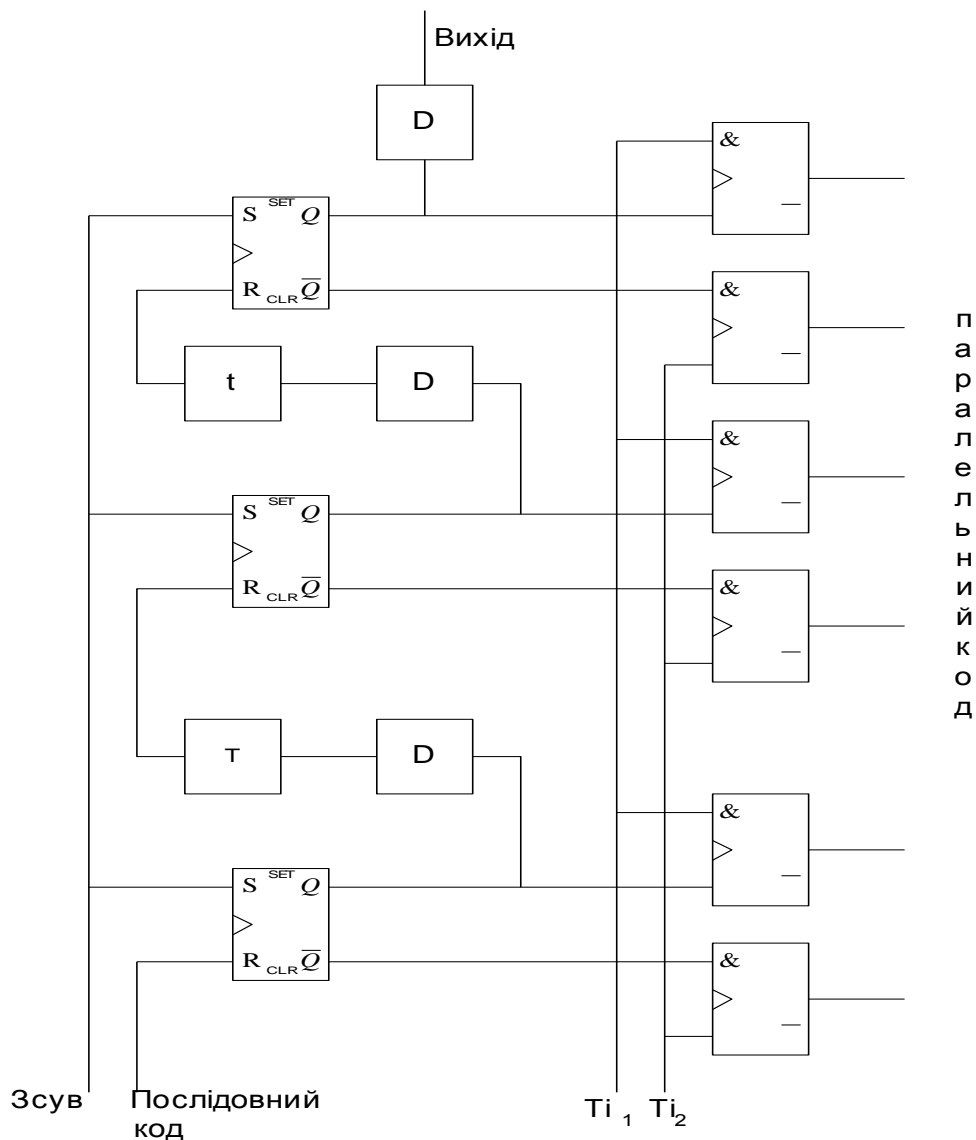


Рис. 1. 7а. Регістр для перетворення послідовного коду у паралельний. D – диференціювальна ланка, яка формує імпульс для встановлення тригера по входу R ; t – невелика лінія затримки

На вхід R першого тригера надходить послідовний код, наприклад, 101. Входи S усіх тригерів об'єднані і на них надходить імпульс зсуву, який одночасно виконує функцію встановлення в "0" регістра. Тригери спрацьовують тільки від додатних імпульсів. Це означає, що встановлення наступного тригера можливе лише у разі перекидання попереднього з "1" у "0". Не важко зауважити, що через три кроки послідовний код 101 буде занесений у регістр і може бути прочитаний у паралельному (прямому або інверсному) коді.

Таку ж функцію виконує регістр, зображений на рис 1.7б.

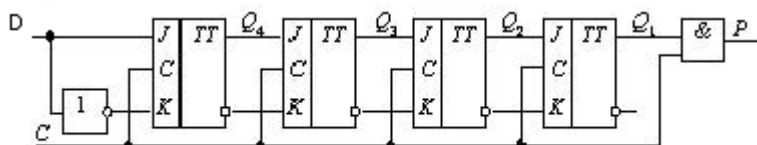


Рис. 1. 76. Регістр для перетворення послідовного коду у паралельний на базі лічильних тригерів.

Ще розглянемо роботу однотактового регістра зі зсуванням числа праворуч (рис. 1. 8).

Тут для спрощення схеми ланки зчитування числа пропущено. Для зсуву числа потрібно подати n імпульсів зсуву.

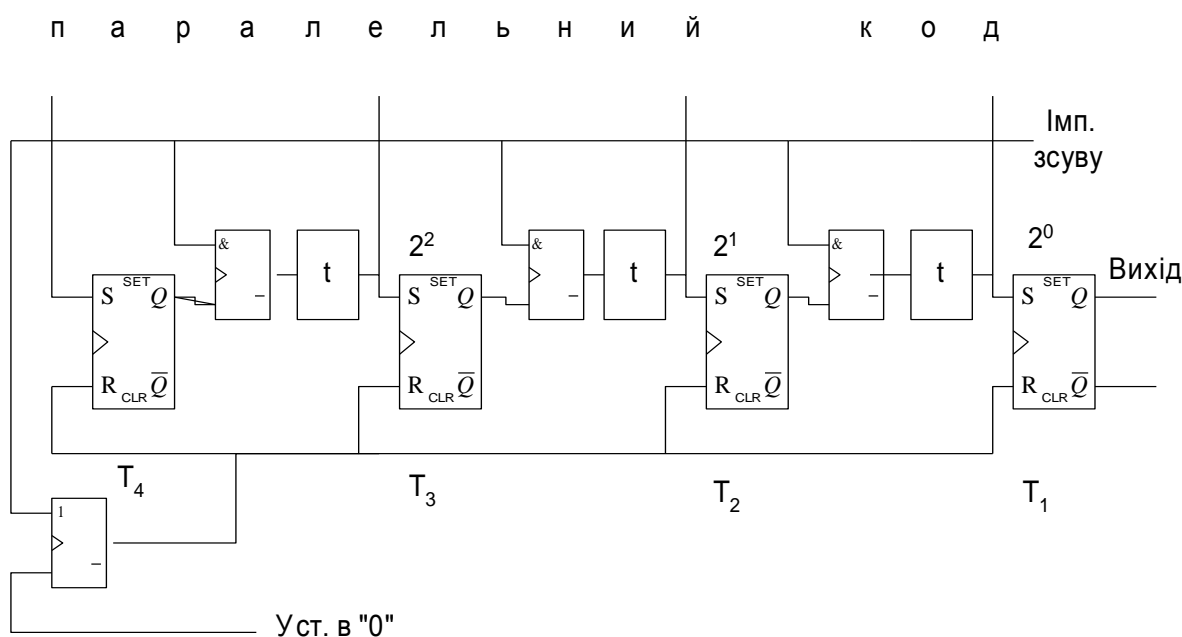


Рис. 1. 8. Однотактовий регістр для зсування коду праворуч

Нехай у регістрі є код 1011. Стан тригерів після кожного імпульсу зсуву t_i буде змінюватися відповідно до таблиці:

Імпульс	t_1	t_2	t_3	t_4	
T ₄	1	0	0	0	$t_2 = t_1 + t_{затр.}$
T ₃	0	1	0	0	$t_3 = t_2 + t_{затр.}$
T ₂	1	0	1	0	$t_4 = t_3 + t_{затр.}$
T ₁	1	1	0	1	

Якщо використовувати парафазний (такий, у якому присутні одночасно прямі та інверсні значення) код, то нема потреби у лініях затримки.

Затримку можна виконати також за допомогою тригера, що значно підвищує надійність роботи схеми. Зсув можна робити як праворуч, так і ліворуч. Такі регістри називають *реверсивними*.

Якщо вихід T_1 подати на вхід T_4 то легко зауважити, що в разі подання імпульсів зсуву інформація у регістрі буде циркулювати. Такі регістри називають *кільцевими*.

Лічильник – пристрій, призначений для підрахунку кількості імпульсів. Лічильники бувають підсумовувальні, віднімальні та реверсивні. Реверсний лічильник залежно від перекомутації може бути підсумовувальним або віднімальним. Будують лічильники на основі тригерів, використовують у пристроях керування та в арифметичних пристроях для рахунку номерів команд, кількості циклів програми, кількості тактів у разі множення і ділення, а також як суматори. Приклад 4-розрядного підсумовувального двійкового лічильника показано на рис. 1. 9.

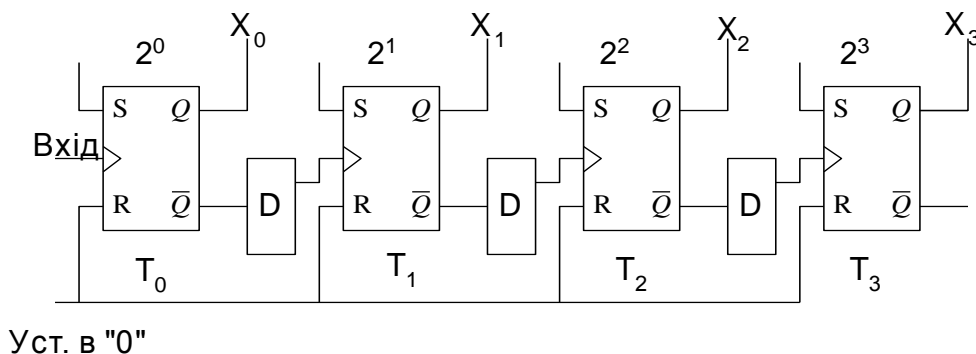


Рис. 1. 9. Схема 4-розрядного підсумовувального двійкового лічильника

Такий лічильник порахує до 16 (1111) і знову стане в "0". Максимальна кількість імпульсів, яку може підрахувати двійковий лічильник, що складається з n розрядів, дорівнює $2^n - 1$.

Віднімальний лічильник отримаємо тоді, коли сигнал знімати з інверсного виходу тригера.

Лічильники, які однаково можна використовувати як для додавання, так і для віднімання імпульсів, називають *реверсивними*.

Іноді потрібно отримати значення $K \neq 2^n$. Тоді будують спеціальні перерахункові схеми, де від певних розрядів уводять обернений зв'язок. Перерахункові схеми з $K=10$ називають декадними лічильниками, які застосовують для побудови десяткових лічильників (рис. 1.10).

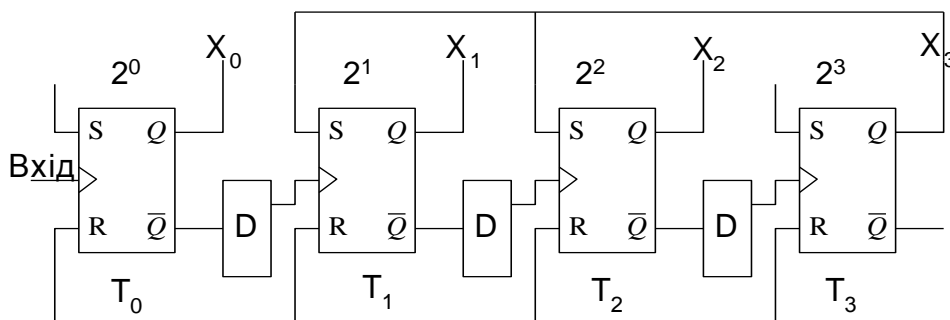
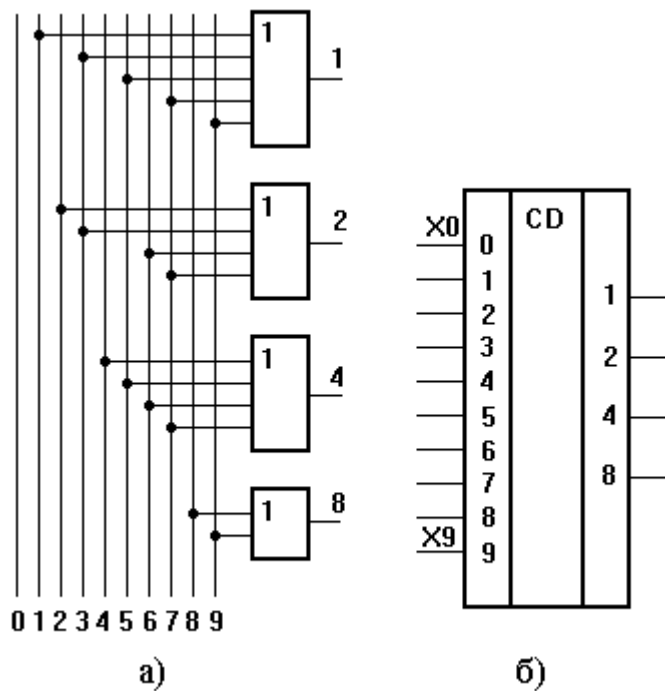


Рис. 1. 10. Декадний лічильник

Шифратор – це вузол ЕОМ, що виконує операцію формування відповідного двійкового коду в разі появи сигналу на одному з входів шифратора. У обчислювальній техніці застосовують переважно багатоімпульсні шифратори, які дозволяють кодувати дані при записуванні програм на носії інформації чи дані, виражені якоюсь фізичною величиною (напр., напругою). Приклад шифратора на 10 входів показано на наступному рисунку (а – принципова схема, б – схемне позначення стандартних шифраторів в документації).



Дешифратор – пристрій який виконує перетворення n -розрядного двійкового коду в однорозрядний з основою $p=2^n$, тобто функцію обернену до шифрування. Використовують у багатьох пристроях, зокрема у пристроях керування, для розшифровування коду операції та видавання сигналів керування у ті кола машин, які повинні працювати під час виконання цієї операції; у запам'ятовувальних пристроях для розшифрування адреси чи команди, записування або читання коду з певної комірки пам'яті.

Розглянемо роботу простої схеми дешифрування на два розряди. Як і в попередніх випадках, побудуємо її на основі тригерних схем (рис. 1. 11). Пояснення роботи такого пристрою не потребує особливих зусиль.

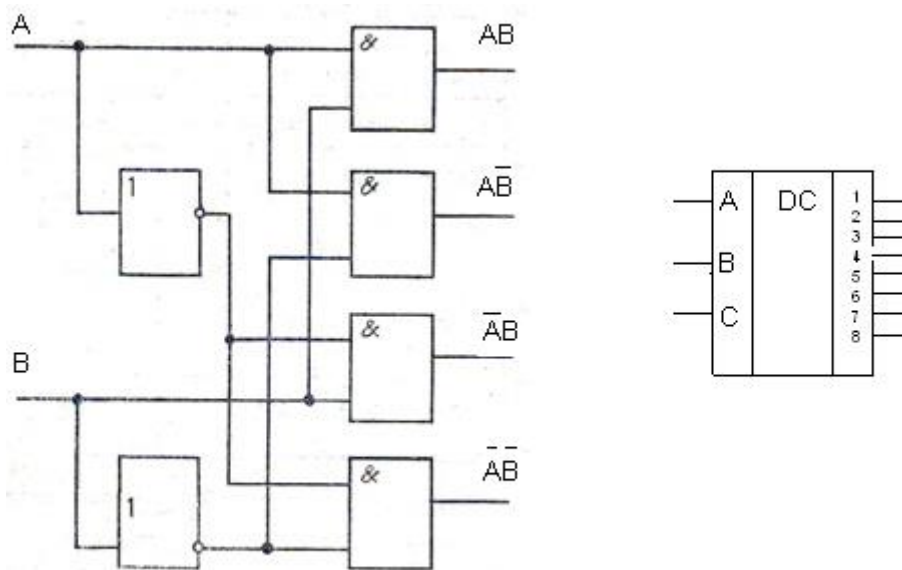


Рис. 1. 11. Схема дворозрядного дешифратора (з правого боку показано схемне позначення трирозрядного дешифратора)

Суматор – пристрій, що виконує сумування чисел на підставі правил порозрядного додавання з урахуванням переносів, які спрямовують у старші розряди.

Суматори класифікують:

1. за виглядом елементів, які використовують – комбінаційного та накопичувального типів;
2. за способом введення-виведення чисел – паралельної та послідовної дії;
3. за способом зображення чисел – двійкові та десяткові;
4. за способом організації перенесення – з послідовним та наскрізним перенесенням.

Розглянемо суматори комбінаторного типу (комбінація логічних елементів І, АБО, НЕ)

Сигнал на виході (сума) утворюється тільки в разі визначеної комбінації вхідних сигналів (доданків), які подаються одночасно. Після зникнення вхідних сигналів вихідний сигнал зникає (запам'ятовувальних властивостей немає). Тому такі суматори працюють з регістром, у який записується результат.

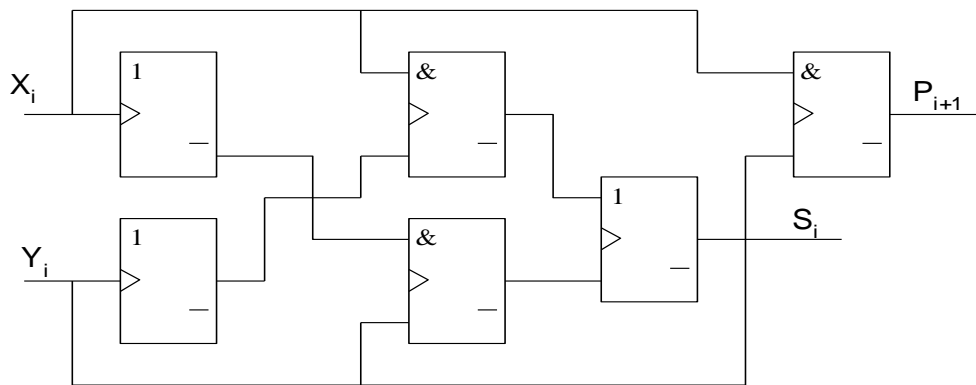
Процес сумування і перенесення з сусіднього розряду в однорозрядній сумувальній схемі розбито на дві аналогічні операції: сумування дворозрядних доданків і сумування з отриманим результатом одиниці перенесення. Кожну з функцій виконує схема, яку називають *напівсуматором*.

Розглянемо роботу напівсуматора, яку описують логічними виразами для суми:

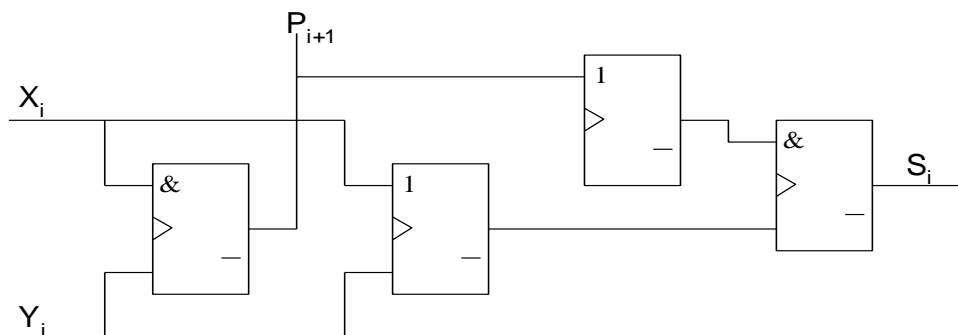
$$\begin{aligned}
 S_i &= X_i \bar{Y}_i + \bar{X}_i Y_i = X_i \bar{Y}_i + \bar{X}_i Y_i + X_i \bar{X}_i + Y_i \bar{Y}_i = \\
 &= X_i(\bar{X}_i + \bar{Y}_i) + Y_i(\bar{X}_i + \bar{Y}_i) = (X_i + Y_i)(\bar{X}_i + \bar{Y}_i) = \\
 &= \overline{X_i Y_i}
 \end{aligned}$$

та перенесення $P_{i+1} = X_i Y_i$.

Результат для суми отримано внаслідок певних алгебричних перетворень. Для прикладу розглянемо схеми, які реалізують проміжний і кінцевий етапи перетворень (рис. 1. 12).



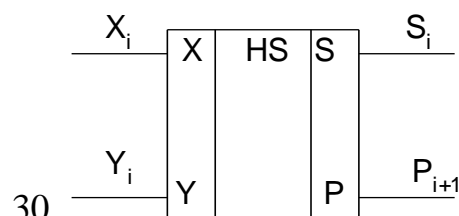
а



б

Рис. 1.12. Схеми напівсуматорів комбінаційного типу:
а – проміжний, б – кінцевий

На схемах позначають



Якщо додати ще один напівсуматор, то отримаємо схему повного суматора на три входи (рис. 1. 13).

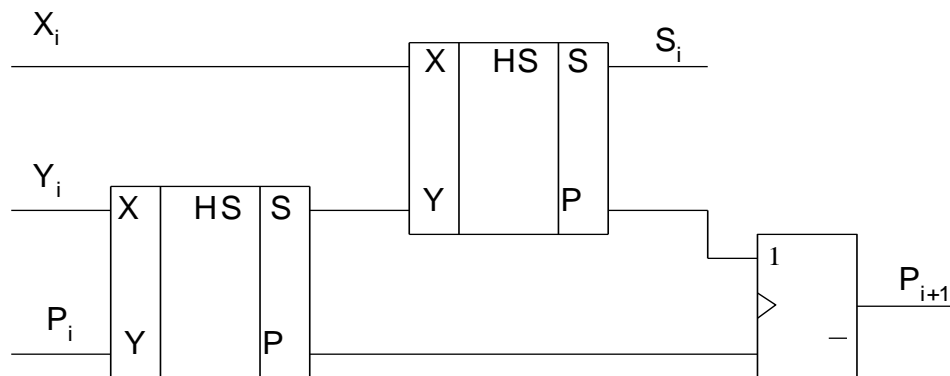
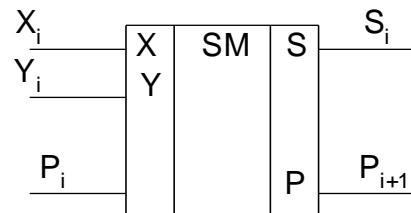


Рис.1.13. Схема повного суматора на три входи

Повний суматор на схемах позначають так:



Багаторозрядний суматор можна отримати простим послідовним з'єднанням однорозрядних суматорів (рис. 1. 14).

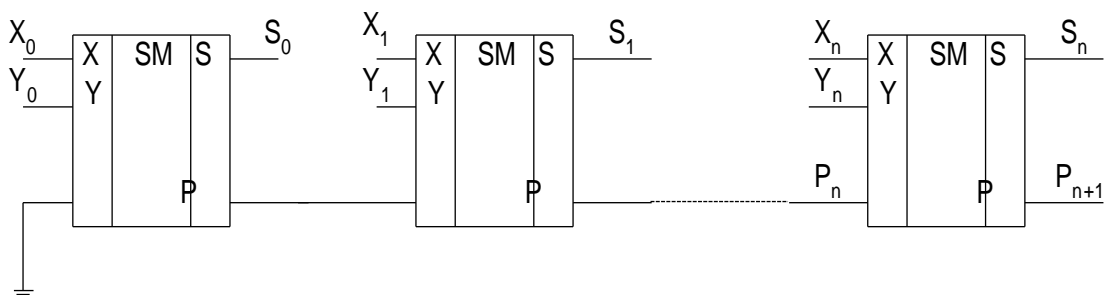


Рис. 1. 14. Схема багаторозрядного суматора

Для зменшення часу поширення сигналу перенесення використовують схеми прискореного перенесення.

Однорозрядний суматор можна використати для виконання операції віднімання. Для цього на вхід Y подають або прямий, або обернений код доданка, який відповідає його від'ємному значенню (рис. 1. 15).

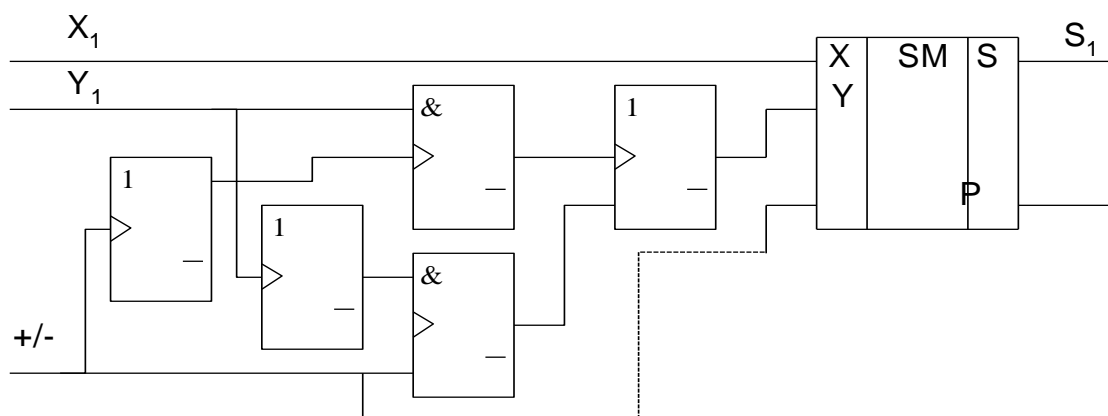


Рис. 1. 15. Використання однорозрядного суматора для операції віднімання

Залежно від сигналу керування +/- на вхід суматора подається прямий код Y (при +/- =0), або обернений код Y (при +/- =1). Для формування додаткового коду доданка до його інверсного коду треба додати 1 молодшого розряду. Це досягають з'єднанням входу керування +/- з входом суматора молодшого розряду.

Трохи складніше, однак можна зібрати з наведених вище логічних схем і вузлів пристрої для виконання операції множення та ділення двійкових чи десяткових чисел.

Список літератури до теми

1. Злобін Г.Г., Рикалюк Р.Є. Архітектура та апаратне забезпечення ПЕОМ: Навч.посіб. –К., 2006, 2012.
2. Шеннон К. Работы по теории информации и кибернетике / Пер. с англ. – М., 1963.
3. Хэмминг Р.В. Теория кодирования и теория информации / Пер. с англ. – М., 1983.
4. Биркгоф Г., Барти Т. Современная прикладная алгебра / Пер.с англ. –М.; 1976.
5. Ланцов А.Л., Зворыкин Л.Н., Осипов И.Ф. Цифровые устройства на комбинентарных МПД интегральных микросхемах. –М., 1983.
6. Горбунов В.Л., Панфилов Д.И., Преснухин Д.Л. Справочное пособие по микропроцессорам и микроЭВМ. –М., 1988.

7. *Каган Б.М. Электронные вычислительные машины и системы: Учеб.пособие для вузов.*—М., 1991.
8. *Харрис Д. М., Харрис С.Л. Цифровая схемотехника и архитектура компьютера/второе издание. - Morgan Kaufman/ © English Edition 2013.*

Контрольні запитання до теми

1. Яка відмінність у термінах ЕОМ і ПК?
2. Що називаємо інформацією?
3. Дайте означення біта.
4. Назвіть три головні функції алгебри логіки.
5. Запишіть закони інверсії.
6. Що таке тригер і які типи тригерів Ви знаєте?
7. Які стани дозволені, а які заборонені у *RS*-тригері?
8. Які функції регістрів і які типи регістрів використовують у побудові ЕОМ?
9. Як побудувати десятковий лічильник?
10. Для чого використовують шифратори та дешифратори?
11. Які типи суматорів Ви знаєте?

Тема 2. Класифікація архітектур ОС

Для ЕОМ першого і другого покоління характерною була класифікація за системою команд. Практично перехід від однієї системи до іншої означав повну перекваліфікацію програміста. У ЕОМ третього і четвертого покоління щораз більше застосовували мови програмування високого рівня (Фортран, ПЛ1, Алгол та ін.), і програмісти головно не потребували знання системи команд, структури пам'яті, способів адресування, розрядності регістрів тощо. Однак для системного програміста і сьогодні актуальними є питання структури ЕОМ, ємність пам'яті і взаємодії між компонентами, особливо між процесором і периферією, тому, погоджуючись зі А. Смирновим [1], розглянемо таку класифікацію.

2.1. Класифікація за інтегральними ознаками

Доцільно скористатися поняттям архітектури ЕОМ як певним набором інтегральних характеристик, а саме:

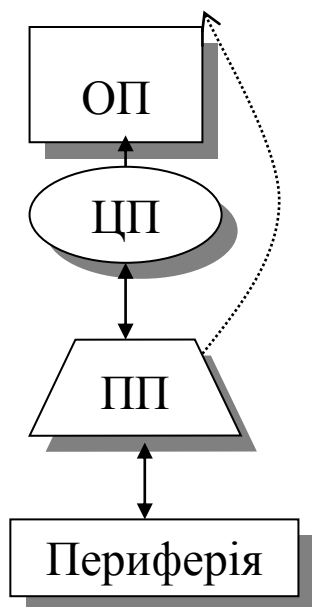
- сукупність властивостей і характеристик, що визначають інформаційні зв'язки між ЦП, периферійними процесорами (ПП), ОП, зовнішньою пам'яттю і периферією;
- взаємодією і структурою потоків команд та потоків даних.

Перша характеристика дає змогу врахувати такі особливості ЕОМ третього і четвертого поколінь, як багатопроцесорність, моношинна структура зв'язків, швидкодія, розподілена пам'ять.

2.1.1. Класифікація за взаємодією ЦП, ОП, ПП

Розглянемо інформаційні зв'язки між трьома головними типами пристроїв ЕОМ, обмежившись лише структурними зв'язками в архітектурі ОС. Є такі структури:

- однопроцесорна (з одним ЦП і периферійним процесором чи каналом) зі зв'язком через ЦП;
- з одним потужним ЦП і кількома периферійними процесорами, що його обслуговують;
- багатопроцесорна (кілька ЦП, ПП і секціонована ОП);
- з магістральною шиною;
- мережна;
- функціонально переналагоджувальна багатопроцесорна;
- мас-процесорна (дуже значна кількість процесорів).



В однопроцесорній структурі ЦП (спільно з операційною системою) координує всю роботу системи. Потрібні прямі зв'язки з периферійними пристроями ЦП виконує через канали або ПП (рис. 2. 1). Така структура була характерною для ЕОМ третього покоління: IBM 360/370, ЕС ЕОМ, ICL System4 та інші. Ця архітектура застосовується також в сучасних пристроях, які забезпечують виконання спеціальних функцій у комп'ютерних мережах.

Рис. 2. 1. Однопроцесорна архітектура ЕОМ

Однак для продуктивніших систем такі структури, коли в разі кожного звертання необхідно переривати роботу ЦП, не вигідні. Тому виникла ідея поєднати роботу потужного ЦП із кількома менш потуж-

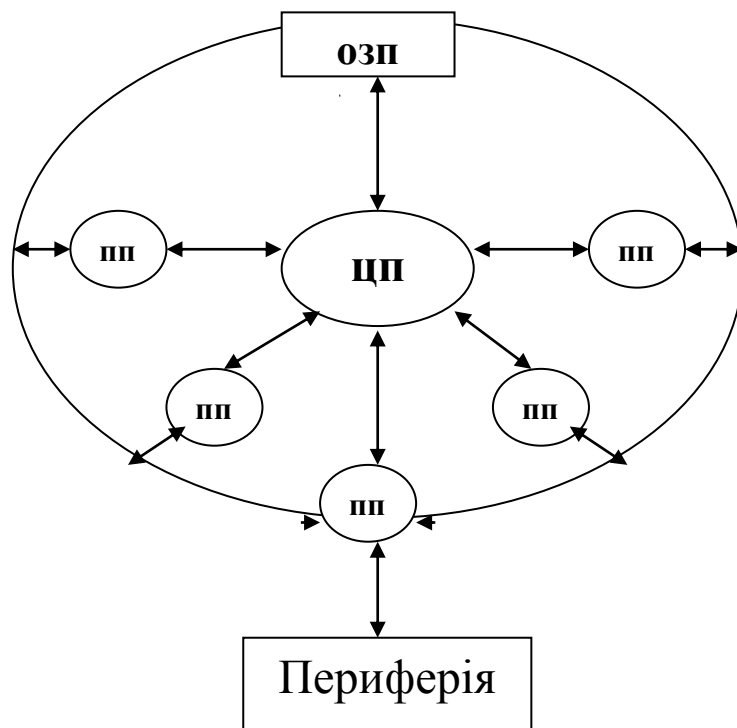


Рис. 2. 2. Архітектура з периферійними процесорами

нішими, функція яких полягає у обслуговуванні переривань від зовнішніх пристроїв, а також ОЗП. Наприклад, це може бути структура з концентричним розташуванням периферійних процесорів (рис. 2.2). Подібну структуру мають високопродуктивні системи CDC.

А от ЕОМ "Ельбрус" (кількість ЦП від 1 до 10, ПП від 1 до 4, кількість секцій ОП від 1 до 8), яку донедавна використовували тільки для військових цілей, була побудована за принципом збільшення кількості ЦП з однією ОП. Система може працювати за алгоритмом розпаралелювання задачі на кілька процесів або розв'язувати на кожному з них свою задачу. Ця система застосована також у Burroughs-700 (США), Iris-80 (Франція) та ін. (рис. 2. 3).

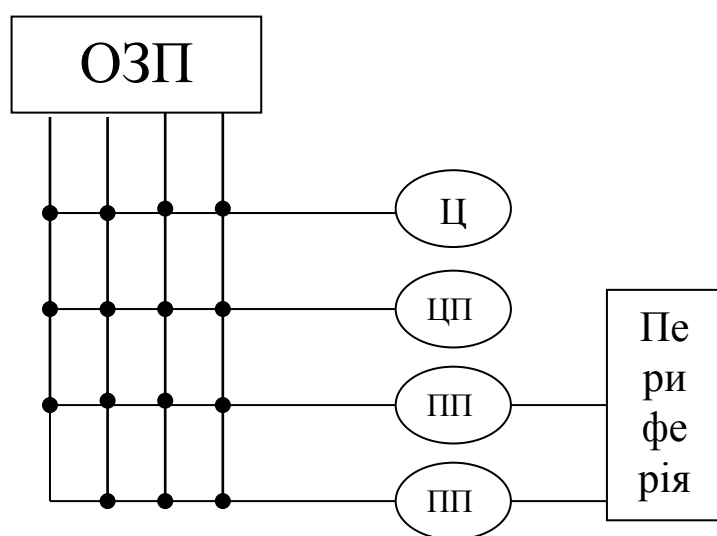


Рис. 2. 3. Багатопроцесорна архітектура ЕОМ

Збільшення продуктивності такої системи близьке до лінійного до трипроцесорного варіанта (рис. 2. 4). Тому серед ОС високої продуктивності популярні дво- і чотирипроцесорні системи (Cray-X-MP; IBM "Сьєрра" та ін.). Завдяки використанню комплексу додаткових засобів і архітектурних знахідок з опорою на кеш-пам'ять великої потужності вдалося отримати майже лінійне зростання продуктивності зі збільшенням кількості процесорів до п'яти-десяти для широкого класу задач на ОС родини "Ельбрус".

Нові можливості відкриваються для паралельних ОС, якщо звернутися до різного типу сіткових алгоритмів. Такими алгоритмами розв'язують багато задач неперервного середовища, що їх описують системи рівнянь у частинних похідних, лінійні системи алгебричних рівнянь. Процесори, які опрацьовують інформацію та

Продуктивність,
умовних одиниць

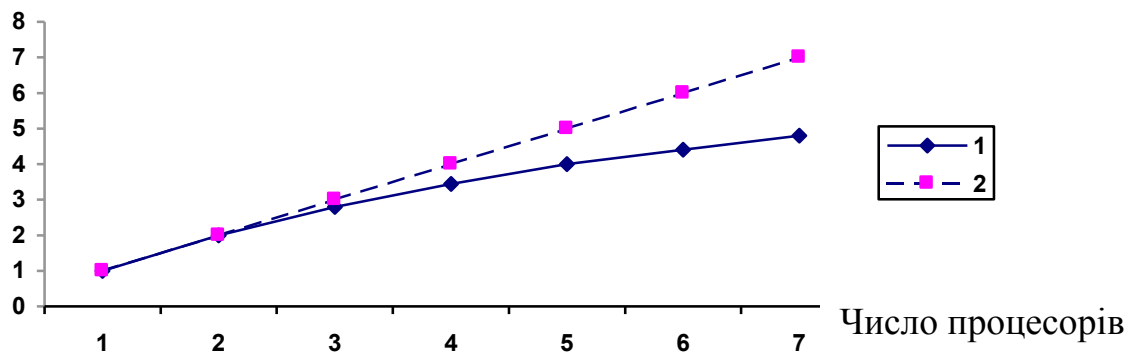


Рис.2.4. Залежність продуктивності ОС від числа процесорів:
1 – реальна, 2 – очікувана.

обмінюються інформацією між собою, розміщені у вузлах решітки або сітки. Таких процесорів у ОС міститься вже декілька сотень або навіть тисяч. Основою ОС стають мікропроцесори. На відміну від багатопроцесорних систем з великими універсальними процесорами, такі системи називають *мас-процесорними*.

Операції розподіляють за окремими функціональними пристроями, які працюють паралельно, проте в самих функціональних пристроях застосована конвеєризація (див. стор. 35), а також відбувається попередній перегляд програм, а їхньому швидкому попередньому і паралельному виклику з ОП допомагає глибоке розшарування пам'яті. Детальніше про це див. напр., у [1].

Розрізняють такі мас-процесорні ОС:

- багатопроцесорні скалярні (IBM 3090/200, ЕС 1065);
- векторні і векторно-конвеєрні (CDC 6600, Cray -1);
- мас - процесорні з динамічною архітектурою;
- спеціальні процесорні.

Про *магістральну шину* детальніше йтиметься під час розгляду архітектури мікропроцесорних систем. Тут зазначимо лише таке: магістрально-шинна структура передбачає, що всі частини ОС можуть бути приєднаними до однієї багатопровідної шини (або коаксіального чи оптоволоконного кабелю), захоплюючи її на короткий період машинного такту для обміну інформацією (рис. 2. 5). Конфліктну ситуацію вирішує арбітр шини.

Розрізняють шину адреси, шину даних та шину керування.

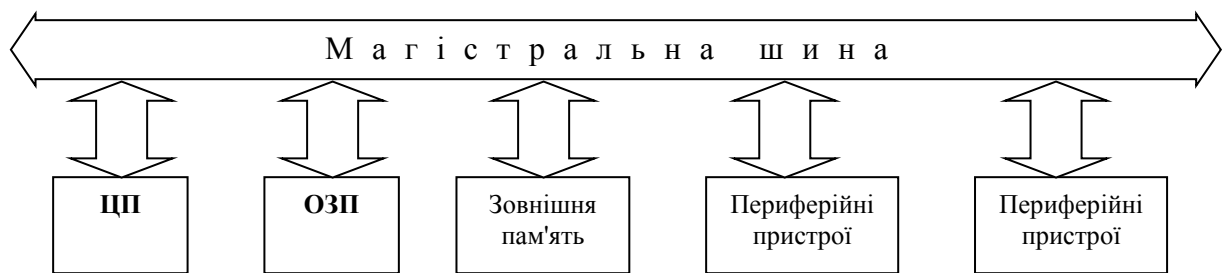


Рис. 2.5. Приклад архітектури ОС з магістральною шиною

Окремий напрям становить мережна архітектура ОС. Розрізняють глобальні, регіональні та локальні комп'ютерні мережі. Вони мають свою топологію і їхня робота регламентована певним набором правил (протоколів). Про це детальніше див. напр. у [4,5].

2.1.2. Класифікація за взаємодією потоку команд і потоку даних

Розглянемо дещо іншу класифікацію, яка властива головно суперЕОМ. Відомо, що у ЕОМ з класичною архітектурою опрацювання даних є послідовним. Команди надходять одна за одною, а для них з ОП чи з регістрів так само послідовно надходять операнди. Одній команді відповідає один необхідний для неї набір операндів; для бінарних операцій, як звичайно – два набори. Скорочено такий тип архітектури називають ОКОД (одна команда – одні дані); англійська аббревіатура SISD (Single Instruction, Single Data) (рис. 2. 6).

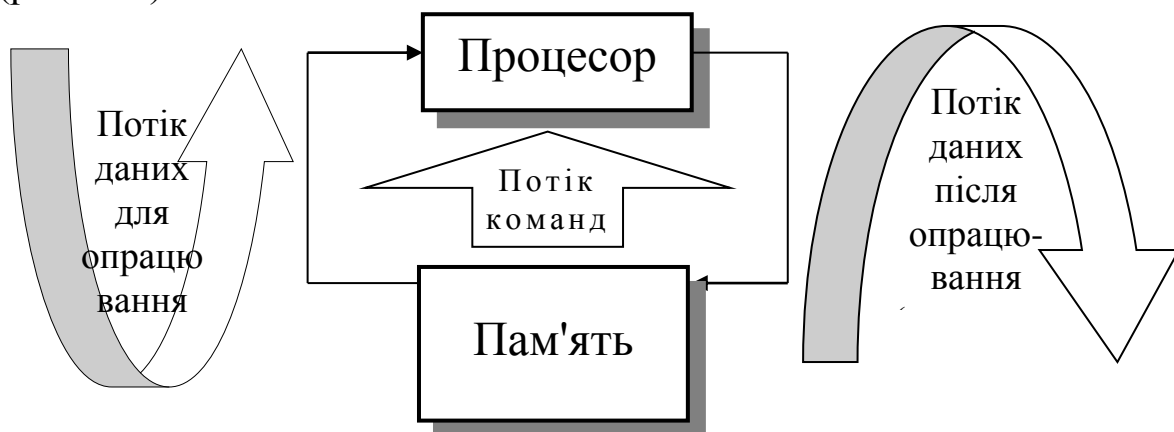


Рис.2.6. Тип архітектури ОКОД

Для спеціальних процесорів, що опрацьовують зображення, або асоціативних процесорів опрацьовується набір з множини даних. Це тип ОКБД (одна команда – багато даних); англійська аббревіатура SIMD (Single Instruction, Multiple Data) (рис. 2.7). Кількість елементів, що опрацьовуються, може бути дуже великою (тисячі), проте їх опрацювання відбувається синхронно (приклад: сучасні процесори з технологією MMX).

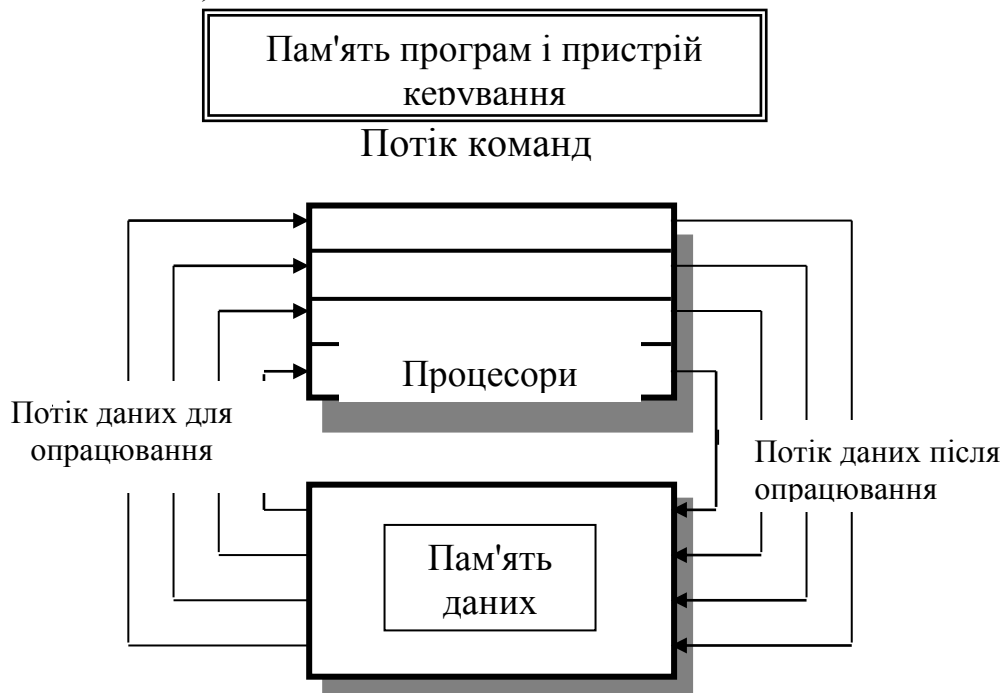


Рис. 2.7. Тип архітектури ОКБД

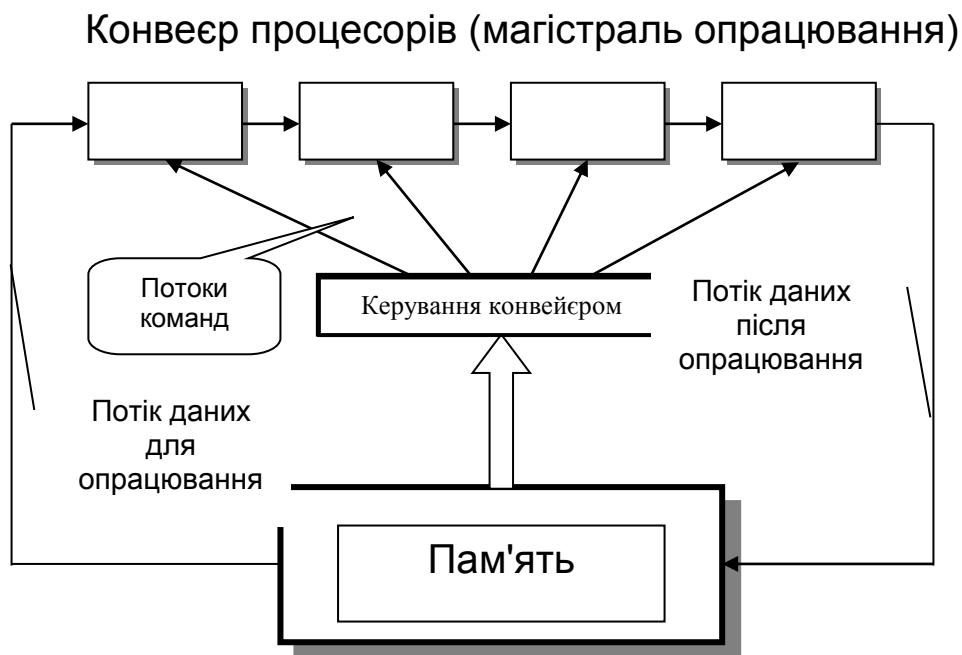


Рис. 2. 8. Тип архітектури БКОД

Прикладом типу БКОД (багато команд – одні дані) може бути конвеєр, тобто послідовне опрацювання одного потоку даних багатьма опрацьовувальними ступенями конвеєра (рис.2.8).

Тип БКБД (багато команд – багато даних) передбачає найповніше і незалежне розпаралелювання процесу (рис. 2. 9).

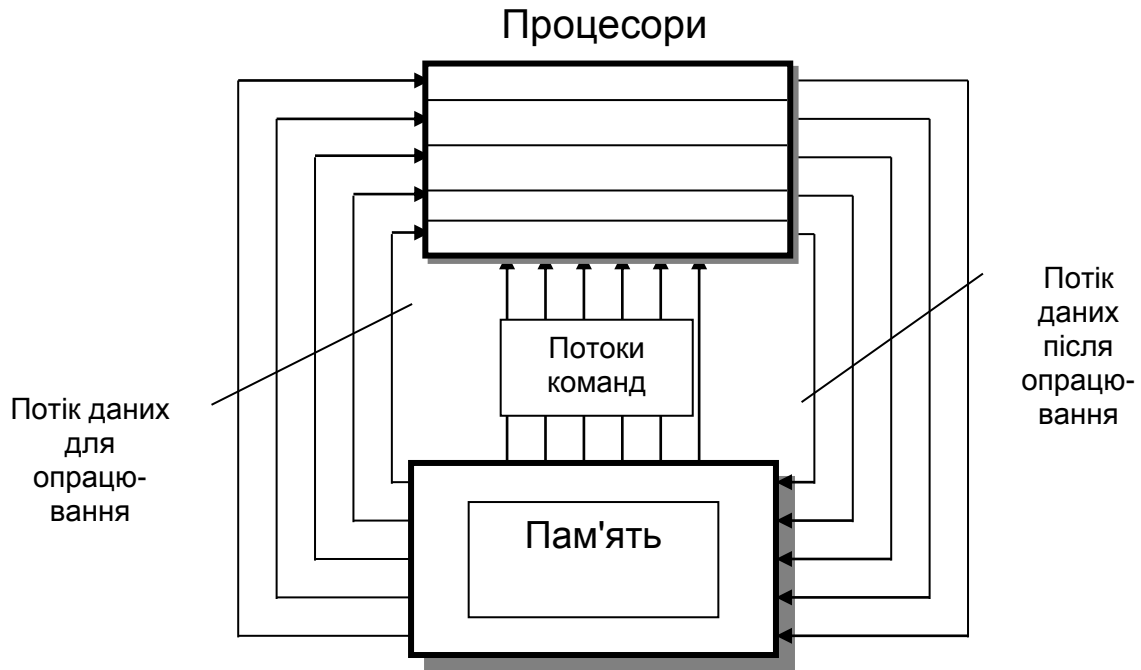


Рис. 2. 9. Архітектура БКБД

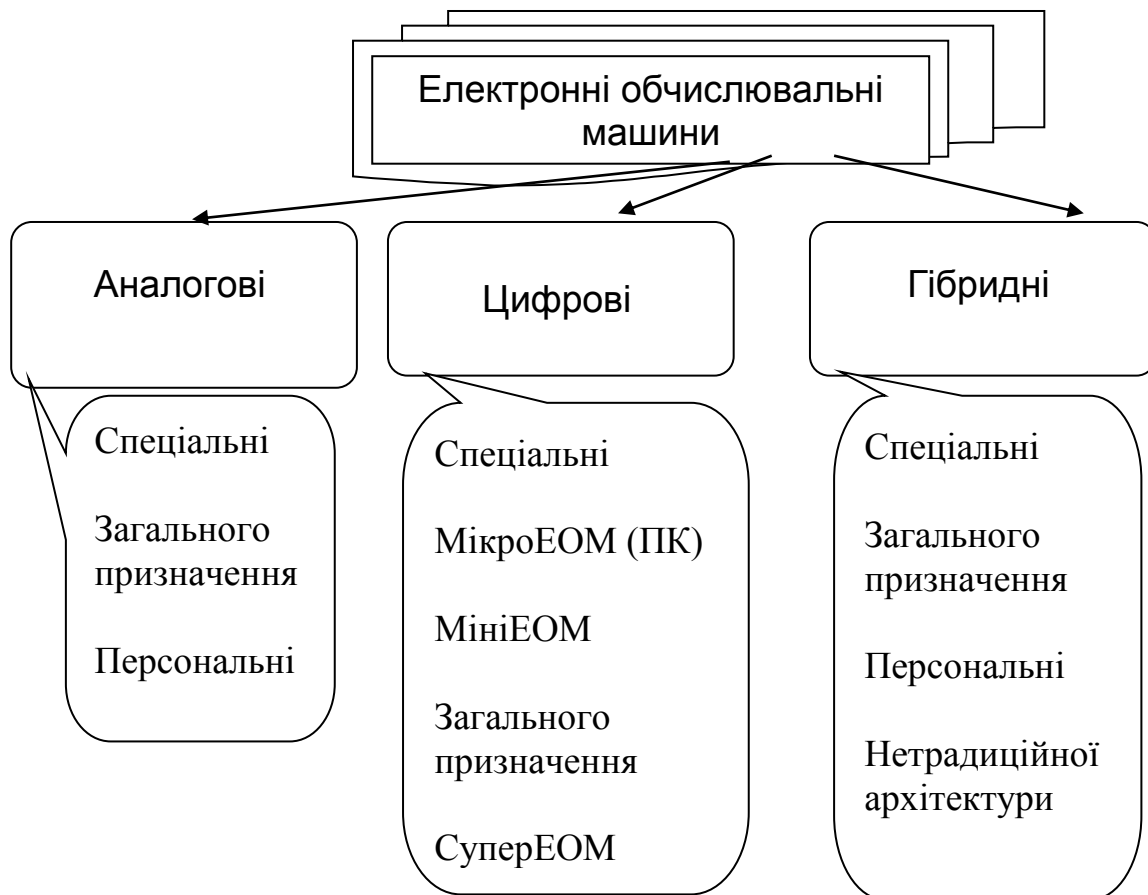
Розглянута вище класифікація є в багатьох працях, і хоча вона й відображає головну взаємодію потоків даних і команд, однак все ж не охолює всіх можливих взаємодій, які трапляються на практиці.

Наприклад, проаналізуємо конвеєрно-векторне опрацювання. Щодо потоку даних та інтегрального опрацювання дії – це тип ОКБД. Проте опрацьовувальну дію можуть виконувати підконвеєри, тобто множина послідовно виконуваних команд над кожною парою операндів, а це тип БКОД.

Можна, звичайно, ввести проміжні групи нижчого рангу й отримати ранговано однозначно визначену систему архітектур. Однак життя, як звичайно, не вписується навіть у найретельніше відпрацьовані схеми. Конструктори ОС прагнуть досягти не найгармонійнішої одноманітної архітектури, а максимальної продуктивності в заданому класі задач чи найліпшого співвідношення продуктивності ОС до її вартості, застосовуючи всі можливі архітектурні й апаратні хитрощі [1].

2.2. Класифікація за функціональним призначенням

Як уже зазначено, сучасні обчислювальні системи умовно поділяють на аналогові, дискретні та гібридні. Кожна з цих груп має ще свій внутрішній поділ, який виглядає приблизно так [2]:



2.3. Класифікація за способом організації виконання команд

CISC (Complete Instruction Set Computer) — ЕОМ з повним набором інструкцій (команд), які виконує мікропроцесор. До цього типу належить сімейство ПЕОМ на базі мікропроцесорів **x86**. Внаслідок широкого набору команд архітектура таких процесорів страждає від надміру апаратних ресурсів, склад і призначення регістрів досить неоднорідні. На виконання команд необхідно затрачати значний час, причому час виконання різних команд не однаковий (різна кількість машинних тактів). Ці фактори негативно впливають на продуктивність ПЕОМ, тому починаючи з МП **Intel486** застосовують комбіновану архітектуру, а саме у **CISC**-процесор

вкраплюють **RISC**-ядро.

Мікропроцесори з архітектурою **RISC (Reduced Instruction Set Computer)** використовують порівняно невеликий (скорочений) набір найуживаніших команд. Цей набір визначено внаслідок статистичного аналізу значної кількості програм для головних галузей застосування **CISC**-процесорів. Особливістю **RISC**-архітектури є те, що всі команди мають однаковий формат, працюють з операндами, які розташовані у регістрах процесора. Звертання до пам'яті виконується за допомогою спеціальних команд завантаження регістра і запису.

Невеликий набір команд і простота їхньої структури дають змогу реалізувати повністю апаратне виконання й ефективний конвеєр з порівняно незначним об'ємом обладнання. Завдяки цьому команди виконуються в 2-4 рази швидше, ніж у звичайних **CISC** з тою ж тактовою частотою. Продуктивність такого процесора на 30% більша ніж звичайного. Перші експериментальні МП (**RISC II, MIPS, IBM 801**) з'явилися ще у 1980-81 рр. Мікропроцесорну революцію продовжили провідні фірми у 1985-86 рр. (**Acorn, AMD, IBM, Sun Microsystems**). Сьогодні **RISC**-процесори здебільшого використовують для побудови співпроцесорів та спецпроцесорів, інтелектуальних контролерів, багатопроцесорних систем та ін. До сімейства **RISC**-процесорів належить також архітектура **MIPS** (англ. *Microprocessor without Interlocked Pipeline Stages* — мікропроцесор без блокувань в конвеєрі), яку розробляє компанія **MIPS Technologies**. Архітектуру **MIPS** використовують у комп'ютерах **SGI**, в ігрових консолях **Nintendo 64**, **Sony PlayStation**, **Sony PlayStation 2** і **Sony PSP** та вмонтованих системах.

Головна ідея **MIPS**-архітектури полягає у використанні простих команд (не передбачається апаратна підтримка операцій множення і ділення), що забезпечує високу швидкість їх виконання (висока тактова частота). Це значно спрощує будову процесора і дає змогу уникати блокувань конвеєра. На сьогодні **MIPS** використовують у вмонтовуваних пристроях з критичною продуктивністю.

В деяких серверних процесорах та графічних процесорах (англ. GPU -Graphical Processor Unit) використовують архітектуру **VLIW** (англ. very long instruction word — «дуже довга машинна команда») — архітектуру процесорів з декількома обчислювальними пристроями. Одна інструкція **VLIW**-процесора містить декілька операцій, які повинні виконуватись паралельно. В суперскалярних процесорах також є декілька обчислювальних модулів, але завдання

розподілу між ними роботи розв'язується апаратно. Це дуже ускладнює дизайн процесора і може призвести до помилок. У VLIW-процесорах завдання розподілу виконується під час компіляції і в інструкціях явно вказано, який обчислювальний пристрій має виконувати яку команду.

VLIW можна вважати логічним продовженням ідеології RISC, яке поширює її на архітектури з декількома обчислювальними модулями.

Як і в RISC, в інструкції явно вказується, що саме має робити кожний модуль процесора. Через це довжина інструкції може досягати 128 або ж 256 біт.

Розглянемо приклад VLIW-процесора з двома арифметико-логічними пристроями (АЛПр). Додамо чотири числа, які знаходяться в регістрах R1, R2, R3 і R4. Тоді псевдокод може виглядати так:

$R5=R1+R2$, $R6=R3+R4$; кожний АЛПр додає свою пару чисел
 $R0=R5+R6$, NOP ; перший АЛПр додає проміжні суми, другий простоює.

З цього прикладу можна зробити такі висновки:

1 (переваги) Підхід VLIW значно спрощує архітектуру процесора за рахунок розподілу обчислювальних пристроїв компілятором.

Оскільки з процесора видалено великі і складні вузли, які виконували розподіл обчислювальних пристроїв, значно зменшується енергоспоживання;

2 (недоліки) Через велику кількість порожніх інструкцій для невикористовуваних пристроїв програми для VLIW-процесорів можуть бути значно довшими, ніж аналогічні програми для традиційних архітектур.

Реалізації VLIW-процесорів:

1. Мікропроцесор Intel Itanium має 64-бітну систему команд «з явним паралелізмом» (англ. explicitly parallel instruction computing, EPIC), яка є одним із варіантів VLIW;

2. Процесори фірми Tilera також мають VLIW-архітектуру;

3. Графічні процесори AMD/ATI Radeon, починаючи з R600, мають VLIW-архітектуру.

Ще один тип ЕОМ — з **MISC-архітектурою (Minimise Instruction Set Computer)**, які мають мінімальний набір інструкцій (Сетунь, Forth).

Список літератури до теми

1. *Смирнов А.Д.*, Архитектура вычислительных систем. Учеб. пособие для вузов. – М., 1990.
2. *Аладьев В.З., Хунт Ю.Я., Шишаков М.Л.* Основы информатики.– Учеб. пособие.– М., 1999.
3. *Каган Б.М.* Электронные вычислительные машины и системы: Учеб.пособие для вузов. – М., 1991.
4. *Рикалюк Р.Є., Стягар О.М., Данчак П.В.* Вступ до комп'ютерних мереж: Текст лекцій. – Львів, 1996.
5. *Буров Є.* Комп'ютерні мережі. – Львів, 1999.
6. *Колодницький М.М.* Технічне та програмне забезпечення комп'ютерних інформаційних технологій. – Житомир, 1995.

Контрольні запитання до теми

1. У чому полягає суть класифікації за інтегральними ознаками?
2. Яка залежність продуктивності ЕОМ від кількості процесорів?
3. Наведіть приклади мас-процесорних систем.
4. Які типи взаємодії потоку команд і потоку даних?
5. Як класифікують обчислювальні системи за функціональним призначенням?



IBM Blue Gene P supercomputer [CC BY-SA 2.0](#) [view terms](#)

[Blue Gene / P](#) From [Argonne National Laboratory](#) Uploaded using [F](#)

Applications of supercomputers

The stages of supercomputer application may be summarized in the following table:

Decade	Uses and computer involved
1970s	Weather forecasting, aerodynamic research (Cray-1). ^[75]
1980s	Probabilistic analysis, ^[76] radiation shielding modeling ^[77] (CDC Cyber).
1990s	Brute force code breaking (EFF DES cracker). ^[78]
2000s	3D nuclear test simulations as a substitute for legal conduct Nuclear Non-Proliferation Treaty (ASCI Q). ^[79]
2010s	Molecular Dynamics Simulation (Tianhe-1A). ^[80]

Тема 3. Загальні принципи побудови мікро-ЕОМ та ПЕОМ

3.1. Принципи побудови ЕОМ

Базові принципи організації мікропроцесора. Канонічна схема мікропроцесора. Системи, види і формати команд універсальних мікропроцесорів. CISC-і RISC-архітектури. Вибірка, дешифрування та виконання команд. Подання роботи обчислювального тракту процесора на мікроархітектурному рівні. Режими адресації пам'яті та пристроїв вводу-виводу. Система переривань. Механізми звернення до підпрограм. Мікропроцесор Intel 8086(88). Машинна мова. Архітектура процесора. Регістри загального призначення. Індексні регістри та регістри-вказівники. Регістри сегменту. Шини мікропроцесора. Зв'язок з магістраллю. Переривання (внутрішні та зовнішні, масковані та немасковані).

Архітектура ЕОМ залежить від типу центрального процесора чи мікропроцесора (МП). Визначальним моментом є розрядність МП (8, 16, 32, 64 і т. д.). ЕОМ, зібрані на базі цих МП, мають не тільки певний тип МП, а й функціональні розширювачі, контролери, гнучкі та жорсткі диски, клавіатуру, дисплей та інші вузли.

Відомо також, що розв'язування певної задачі на ЕОМ поділене на частини. Найпростіша частина – це машинна команда. Набір машинних команд, які записані у певній послідовності, становлять машинну програму. Для реалізації тої чи іншої програми необхідні арифметико-логічний пристрій, пристрій пам'яті, пристрій керування, пристрій введення/виведення інформації.

Взаємодія цих пристроїв відбувається через так звану системну магістраль (СМ), або шину. У переважній більшості випадків СМ – це розширення шини МП, однак, не завжди (напр. МП – шістнадцятирозрядний, а зовнішня шина – восьмирозрядна.)

Незалежно від архітектурного виконання структурну організацію взаємодії компонентів ЕОМ можна зобразити, як на рис. 3. 1.

Структурна схема може дещо видозмінюватися залежно від призначення ЕОМ: побутова, професійна чи спеціалізована. (Робоча станція, автоматизоване робоче місце тощо).

Між МП та ОП є зв'язок, тобто розрядність МП пов'язана з обсягом ОП. Наприклад, 8-розрядний МП – ОП 64 Кбайт, для 16 розрядної – від 1 до 16 Мбайт.

Швидкість виконання операцій визначена тактовою частотою МП та магістральної шини. Прості операції додавання цілих чисел типу "пам'ять – пам'ять" виконуються за 20-30 тактів, а множення-ділення – за 130-150. Знаючи тактову частоту процесора, легко визначити його швидкодію.

3.2. *Склад і характеристика ПЕОМ*

Базовий комплект сучасної ПЕОМ містить такі головні компоненти: системний блок, клавіатуру, монітор (дисплей), пристрій для друкування (принтер), маніпулятор. Засоби зв'язку з пристроями, що не належать до базового комплекту, містить також системний блок.

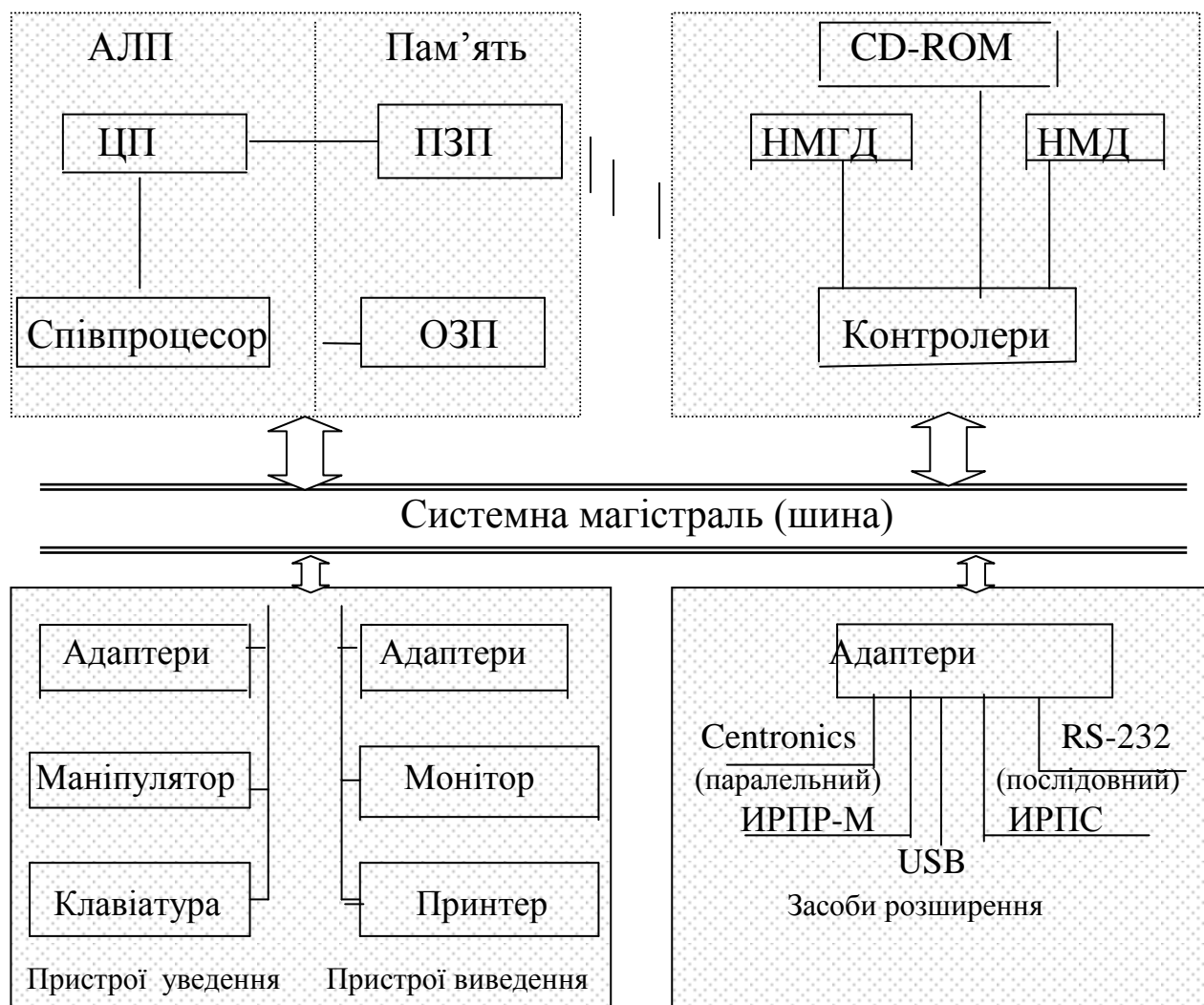


Рис. 3. 1. Структурна схема сучасної ЕОМ

Системний блок містить: електронні модулі (системна плата, оперативний запам'ятовувальний пристрій (ОЗП), постійний запам'ятовувальний пристрій (ПЗП), контролери пам'яті та зовнішні пристрої, блок живлення, спеціальні з'єднувачі (слоти) для модулів розширення ОЗП, адаптери для приєднання до мережі (LAN-карти), контролери складних зовнішніх пристроїв, наприклад, відеоадаптер. Залежно від типу корпусу системного блоку розташування окремих вузлів та рознять для під'єднання зовнішніх пристроїв не однакове. На рис. 3. 2 показано зовнішній вигляд передньої та задньої панелей системного блоку типу АТ та АТХ.

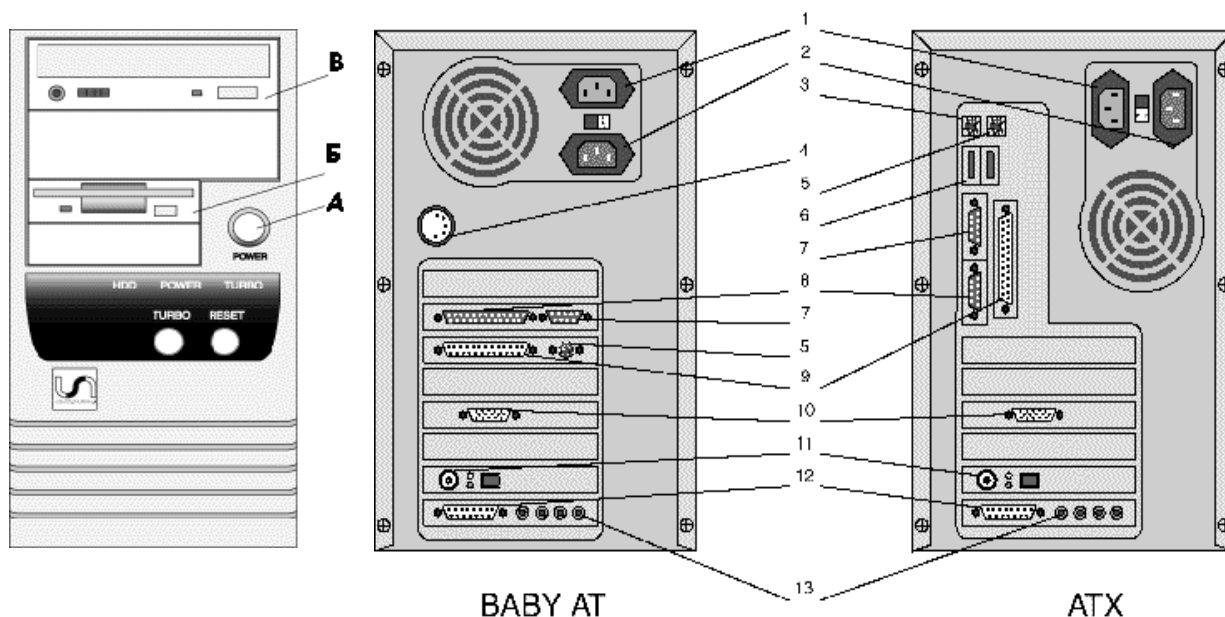


Рис. 3. 2. Зовнішній вигляд передньої (а) та задньої (б) панелей системного блоку ПЕОМ. А – кнопка для вмикання ПК, Б – пристрій для гнучких дисків, В – пристрій для компакт-дисків, 1 – роз'єм для під'єднання шнура живлення монітора, 2 – роз'єм для під'єднання ПК до мережі живлення, 3 – роз'єм для під'єднання клавіатури PS/2, 4 – роз'єм для під'єднання клавіатури, 5 – роз'єм для під'єднання маніпулятора (мишки), 6 – USB-порти, 7 – послідовний порт (COM2), 8 – послідовний порт (COM1), 9 – паралельний порт (LPT), 10 – вихід відеоадаптера, 11 – роз'єм для під'єднання до локальної мережі, 12 – ігровий порт, 13 – гнізда для під'єднання зовнішніх акустичних систем.

3.3. Типи даних МП

Для всіх сучасних МП базовою одиницею даних є байт, тобто 8-бітова величина. Після того, як дані локалізовані у пам'яті процесора, МП може інтерпретувати їх різними способами, залежно від змісту команди чи намірів програміста. Для кодування інформації використовують в основному двійкову, десяткову та шістнадцяткову системи числення. У сучасних МП зrealізована апаратна підтримка багатьох типів даних, які включають в себе числа, вказівники (pointer) та рядки (string):

1. *Біт* (bit) – поодинокий двійковий розряд.
2. *Бітове поле* (bit field) – група до 32 бітів.
3. *Рядок бітів* (bit string) – послідовність бітів довжини до 4 Гбіт.
4. *Байт* (byte) – ціле число, що займає 8 бітів (старший розряд відводиться під знак).
5. *Беззнаковий байт* (unsigned byte) – натуральне число, що займає 8 бітів.
6. *Ціле слово* (integer word) – ціле число, що займає 16 бітів.
7. *Беззнакове ціле* (unsigned integer, word) – натуральне число, що займає 16 бітів.
8. *Довге ціле* (long integer, long word) – ціле число, що займає 32 біти.
9. *Довге беззнакове ціле* (unsigned long integer) – натуральне число, що займає 32 біти.
10. *Чотирикратне ціле* (quad integer, quad word) – ціле число, що займає 64 біти.
11. *Чотирикратне беззнакове ціле* (unsigned quad integer) – натуральне число, що займає 64 біти.
12. *Символ* (char) – байтове подання керівних та алфавітно-цифрових символів ASCII.
13. *Рядок* (string) – неперервна послідовність, утворена із байтів слів чи подвоєних слів від 1 байта до 4 Гбайт.
14. *Двійково-десятковий код* (binary coded decimal, BCD) – байтове (незапаковане) подання десятикових цифр від 0 до 9.
15. *Запакований двійково-десятковий код* – байтове подання від 0 до 9 в одному байті.
16. *Короткий вказівник, зсув* (short pointer, offset) – 16- або 32-розрядна адреса, що непрямо визначає комірку пам'яті.
17. *Вказівник* (pointer) – повна адреса елемента пам'яті, яка містить 16-бітний або 32-розрядний зсув.

У тих випадках, коли МП взаємодіє з арифметичним співпроцесором, додатково використовують знакові 32-, 64- та 80-розрядні формати даних з плаваючою комою.

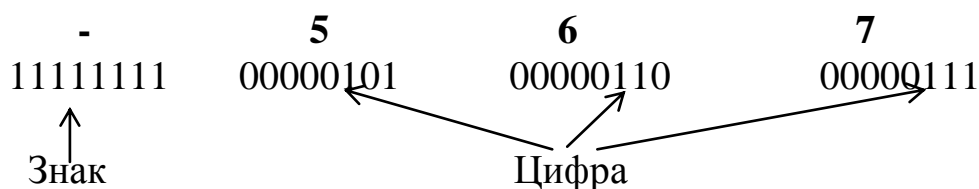
Арифметичні типи даних 32-розрядних ПК

Тип даних	Біти	Кількість десяткових значущих цифр	Діапазон значень
Коротке ціле	16	4	$-32768 \leq x \leq +32767$
Ціле	32	9	$-2 \cdot 10^9 \leq x \leq +2 \cdot 10^9$
Довге ціле	64	18	$-9 \cdot 10^{18} \leq x \leq +9 \cdot 10^{18}$
Запаковане десяткове	80	18	$-99..99 \leq x \leq +99..99$
Коротке дійсне	32	6-7	$8.43 \cdot 10^{-37} \leq x \leq 3.37 \cdot 10^{38}$
Подвійне дійсне	64	15-16	$4.19 \cdot 10^{-307} \leq x \leq +1.67 \cdot 10^{308}$
Довге подвійне дійсне	80	19	$3.4 \cdot 10^{-4932} \leq x \leq +1.2 \cdot 10^{4932}$

Пояснення запакованого та розпакованого формату:

а) У розпакованому форматі кожен байт містить одну десяткову цифру, яка зображується у молодшій тетраді. Знак числа кодується у різних ЕОМ по-своєму. У ПК знак десяткового числа кодується у крайньому лівому байті, причому знак “-” кодується 1, знак “+” - кодується 0.

Наприклад, число -567 у розпакованому форматі буде мати вигляд



В неупакованому форматі цифри, як і символи найчастіше зображують у американському коді ASCII (American Standart Code for Information Interchange). Розрізняють дві частини таблиці кодів: 1 - основна, 2 - додаткова (з урахуванням національних стандартів) [5, 6]:

- 1) 0-127
- 2) 128-255.

Десяткові числа від 0 до 9 у кодуванні ASCII відповідають шістнадцятковим кодам від 30h до 39h відповідно.

б) В упакованому форматі десяткове число займає половину байта, тобто у кожному байті міститься дві десяткові цифри з вагами розрядів 8421 у кожній тетраді. Діапазон чисел, який може бути представлений у одному байті знаходиться у межах від 0 до 99.

Наприклад, число +963,8 в упакованому форматі буде таким:

+ 9 6 3 , 8
00000000 1001 0110 0011 1000

Знак Цифри

Положення десяткової коми може бути довільним, зокрема, для ПК вона знаходиться після третього зліва розряду числа.

Типи даних, що підтримуються архітектурою IA-64

Архітектура IA-64 64-розрядна. Тому в ній з'явилися нові типи даних. Цілі числа представлені 1, 2, 4 і 8 байтами. Дійсні — 4 (single— стандартна точність), 8 (double— подвійна точність) і 10 (double extended — «розширене» дійсне число подвійної точності).

Усі вказівники мають ділянки пам'яті ємністю 8 байт. За замовмочанням більшість інструкцій для роботи з цілими числами оперують даними з ємністю 64 біт. Запис у регістри здійснюється такими самими порціями. В 128-бітовий пакет інструкцій компілятор вставляє шаблон з кількох бітів з метою маркування команд, що виконуватимуться паралельно. Кожна інструкція містить три 7-бітових поля GPR (general-purpose register), які задають стан універсальних регістрів.

Всього процесори з архітектурою IA-64 мають 128 універсальних цілочислових регістрів і стільки ж регістрів для чисел з плаваючою комою. Всі регістри доступні програмісту в будь-якому порядку. Це принципово новий набір регістрів порівняно з регістрами архітектури IA-32.

Єдиний універсальний лінійний адресний простір процесорів архітектури IA-64 має ємність 264 байт. Дані та інструкції можуть зберігатися одночасно в одному діапазоні пам'яті. В цьому сутність єдиного адресного простору (single). «Універсальний» (uniform) в даному випадку означає, що за діапазонами пам'яті не закріплюються наперед визначені функціональні обов'язки. «Лінійний» (linear)

означає, що в адресному просторі не виділяються сегменти. Всі 264 байт пам'яті можуть розподілятися довільними порціями.

Загалом інструкції зберігаються в сервісному центрі пам'яті так, щоб байт з найменшою значущістю був першим (так званий little-endian byte order). Але такий байт може розташовуватися в пам'яті як першим, так і останнім (так званий bigendian byte order).

Серед нових в архітектурі IA-64 — інструкції, призначені для роботи з числами з плаваючою комою. Їх називають «множинно-акумулюючими» (Multiply and Accumulate), наприклад множення і додавання (fma) — операція, яку вважають основною для всієї арифметики з плаваючою комою. Якщо цю операцію записати у вигляді $y = a*b+c$, то при $c=0$ виконується множення, при $b=1$ — додавання, а при $b = 1$ і $c = 0$ — нормалізація.

Множинні операції дають змогу виключити одну операцію округлення, тому точність операції вища, ніж точність при використанні двох стандартних операцій mul і add.

3.4. Мікропроцесор – основа ЕОМ

Тепер детальніше ознайомимося з архітектурою та принципом функціонування МП. Як тільки з'явилась технологічна можливість створювати схеми великої інтеграції (ВІС), які дали змогу стикувати МП з ОЗП, ПЗП та інтерфейси периферійних пристроїв, то відразу виникли так звані мікропроцесорні комплекти (МПК). Одним з перших таких комплектів був Intel 4001/4004 (1971 р). До нього входили:

- центральний 4-бітовий МП ЦП-4004, який розміщувався на одному кристалі;
- ПЗП (4001) ємністю 256 8-бітових слів, які можна було використовувати для зберігання програм і констант. Він містився на одній мікросхемі і підлягав перепрограмуванню;
- оперативна пам'ять 4002 ємністю 80 4-бітових слів. Це була окрема МС, через яку виконувалася ще й програма введення/виведення (чотири вхідні лінії і система керування введення/виведення);

- зсувовий регістр 4003. Це 10-бітовий регістр з послідовним входом, а виходи можуть бути або один послідовного коду, або паралельний з усіх розрядів регістра. Головно відігравав роль буфера для вхідних і вихідних даних.

МПК 4004 використовував одну двонапрямлену 4-розрядну шину даних, а для сигналів керування між ЦП, ОЗП, ПЗП ще п'ять додаткових ліній.

Один ЦП міг керувати від 1 до 16 ПЗП і кількома регістрами зсуву. Разом з портами 4001 та 4002 обслуговувалося 128 ліній входу/виходу.

Система команд ЦП складалася з 45 команд, розподілених на три групи:

- 16 машинних команд (5 подвійної довжини);
- 14 акумуляторних команд;
- 15 команд уведення/виведення і адресування портів.

На наступному етапі розвитку ЕОМ розрядність МП зростала до 8 (Z80, КР580ВМ80А), 16(8086,8088,80186), 32(80386,80486) і т. д.

Архітектура сучасних мікропроцесорів.

Центральний процесор (англ. CPU - Central processing unit) – функціональна частина ЕОМ, що призначена для виконання програм на ЕОМ. Для побудови сучасних ЕОМ використовують мікропроцесори – процесори, які розташовані в корпусі однієї мікросхеми.

На рис. 3.2 зображена спрощена блок-схема мікропроцесора

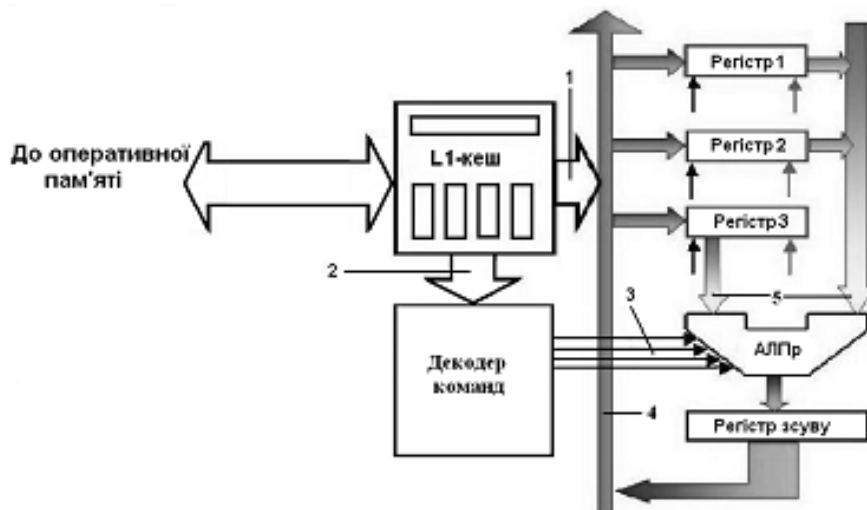


Рис. 3.2. Спрощена блок-схема мікропроцесора:

1 – внутрішня шина мікропроцесора, 2 – шина команд, 3 – шини зв'язку реєстрів із арифметико-логічним пристроєм (АЛПр), Регістр 1, Регістр 2, Регістр 3 – реєстри загального призначення, L1-кеш (див. далі) – швидка пам'ять для інструкцій і даних.

Через блок спраження із зовнішньою шиною мікропроцесора (Front Side Bus, FSB) у кеш-пам'ять мікропроцесора завантажують певну кількість даних і команд програми. Під час виконання програми чергова команда через шину команд завантажується в АЛПр, а дані через внутрішню шину мікропроцесора – у реєстри загального призначення. Виконання команд програми здійснюється за сигналами тактового генератора, який розташовують на системній платі. На рис. 3.3 зображено операції, які виконує мікропроцесор під час виконання одного такту.

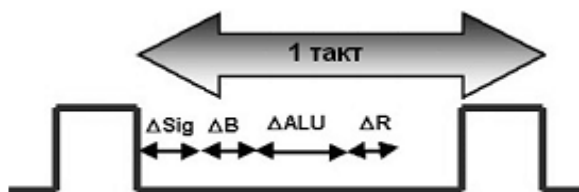


Рис.3.3.

ΔSig – час декодування команди мікропроцесором,
 ΔB – час завантаження даних із реєстрів загального призначення в АЛПр,
 ΔALU – час виконання команди арифметико-логічним пристроєм,
 ΔR – час запису результатів виконання команди (в реєстри загального призначення або у L1-кеш).

Для підвищення швидкодії мікропроцесорів у першу чергу підвищують тактову частоту (процесор Intel8086 мав тактову частоту 4.77 МГц). Для збільшення тактової частоти мікропроцесора можна:

- зменшити час декодування команди ΔSig ;
- зменшити часи ΔB , ΔR (збільшити швидкість обміну даними);
- зменшити час виконання команди ΔALU .

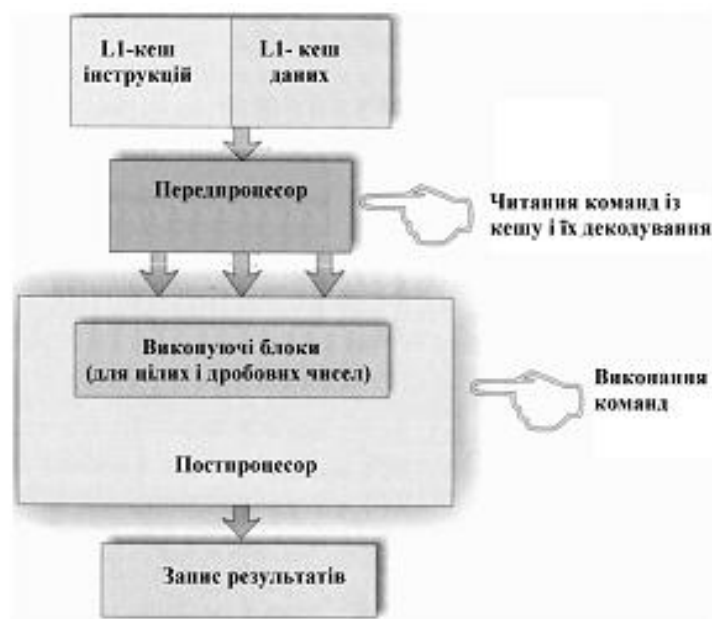
Для підвищення швидкості обміну даними виробники мікропроцесорів покращують розташування компонент мікропроцесора і зменшують їх геометричні розміри (чим менший шлях електрона, тим швидше він його проходить) – сучасні мікропроцесори виготовляють за техпроцесом 32, 22 нм, очікується перехід на 15 нм.

Час декодування команди ΔSig можна зменшити за рахунок скорочення числа команд мікропроцесора. Ще у вісімдесятих роках минулого сторіччя у Кебріджському і Стенфордському університетах були проведені числові експерименти із моделювання процесора зі скороченою системою команд (Reduced Instruction Set Computing – RISC), які довели перспективність RISC-архітектури. Мікропроцесори фірми Intel наразі проектуються за CISC-архітектурою (CISC – Complex Instruction Set Computing). Використання мікропроцесорів із великим набором команд ускладняє їх декодування (збільшує час ΔSig), але спрощує розробку програмного забезпечення для них.

Мікропроцесори фірми AMD мають RISC-архітектуру, а для використання їх в IBM PC- подібних ПЕОМ використовують програмну емуляцію системи команд $Ix86$. Набір команд мікропроцесорів Intel 8086, Intel 8088, Intel 80286 містив команди лише цілочисельної арифметики. Для обчислень з плаваючою комою використовували або програмну реалізацію (збільшувався час виконання команди), або зовнішній співпроцесор плаваючої арифметики.

Починаючи з мікропроцесора Intel 80486DX співпроцесор плаваючої арифметики розташовують в одному корпусі з мікропроцесором. Окрім підвищення тактової частоти розробники використовують різноманітні вдосконалення архітектури мікропроцесора (конвейєр, блок передбачення розгалуження, попереднє завантаження операндів, збільшення розмірів кеш-пам'яті [див. далі] різних рівнів [L1, L2, L3] і ін.). Конвейєр – це пристрій, у якому одночасно кілька команд програми проходять різні стадії обробки. В класичному п'ятирівневому конвейєрі п'ять команд одночасно проходять різні стадії виконання команди:

- читання [вибірка] команди з буфера;
- дешифрування команди;
- читання операндів;
- виконання команди;
- запис результатів виконання команди.



Виконання фрагменту програми мікропроцесором із конвейєром виглядає таким чином:

Такт	Читання команди	Дешифрування команди	Читання операндів	Виконання команди	Запис результатів виконання команди
1	Інструкція 1				
2	Інструкція 2	Інструкція 1			
3	Інструкція 3	Інструкція 2	Інструкція 1		
4	Інструкція 4	Інструкція 3	Інструкція 2	Інструкція 1	
5	Інструкція 5	Інструкція 4	Інструкція 3	Інструкція 2	Інструкція 1
6	Інструкція 6	Інструкція 5	Інструкція 4	Інструкція 3	Інструкція 2
7	Інструкція 7	Інструкція 6	Інструкція 5	Інструкція 4	Інструкція 3
8	Інструкція 8	Інструкція 7	Інструкція 6	Інструкція 5	Інструкція 4
9	Інструкція 9	Інструкція 8	Інструкція 7	Інструкція 6	Інструкція 5

Виконання п'яти команд на процесорі без конвейєра тривало би 5 тактів процесора, виконання цих же п'яти команд на процесорі з конвейєром триває один такт (після завантаження усіх сходинок конвейєра – у цьому прикладі на п'ятому такті і далі. В сучасних мікропроцесорах кількість „сходинок” конвейєра значно більша (операції читання команди з буфера, дешифрування команди, виконання команди, запису результатів виконання команди розділені на більш дрібні етапи).

Список літератури до теми

- 1 *Аладьев В.З., Хунт Ю.Я., Шишаков М.Л.* Основы информатики: Учеб. пособие. – М., 1999.
- 2 *Григорьев В.Л.* Микропроцессор i486. Архитектура и программирование. Кн.1. – М., 1993.
- 3 *Злобін Г.Г, Рикалюк Р.Є.* Архітектура та апаратне забезпечення ПЕОМ: Навч.посіб. –К., 2006., -К.,2012.
- 4 *3. Мячев А.А., Степанов В.Н.* Персональные ЭВМ и микроЭВМ. Основы организации: Справочник. – М., 1991.
- 5 *4. Каган Б.М.* Электронные вычислительные машины и системы: Учеб. пособие для вузов. – М., 1991.
- 6 *5. Фигурнов В. Э.* IBM PC для пользователя. Краткий курс. 7-е изд. – М., 1997.
- 7 *6. Дао Л.* Программирование микропроцессора 8088. Пер. с англ. – М., 1988.

Контрольні запитання до теми

1. Які компоненти входять до системного блоку?
2. Яка розрядність системної магістралі?
3. Якими типами даних оперує мікропроцесор?
4. Яка відмінність між запакованим і розпакованим форматом чисел?
5. Що таке командний цикл і як він пов'язаний з тактовою частотою процесора?
6. Яка функція тригера ознак?
7. Як використовуються тимчасові та акумуляторні регістри для організації виконання команд?
8. Як працює найпростіший мікропроцесор?