

**Тема 2.** Обчислювальні машини з програмою, що зберігається в пам'яті.  
Фон-нейманівська ЕОМ. IBM PC – сумісний персональний комп'ютер

### **1.1. Концепція обчислювальних машин з програмою, що зберігається в пам'яті**

В основі архітектури сучасних ОМ лежить представлення алгоритму розв'язання задачі у вигляді програми послідовних обчислень. Згідно зі стандартом ISO 2382/1-84, програма для ОМ – це «впорядкована послідовність команд, що підлягає обробці».

ОМ, де певним чином закодовані команди програми зберігаються в пам'яті, відома під назвою обчислювальної машини з програмою, що зберігається в пам'яті. Ідея належить творцям обчислювача ENIAC Еккерту, Мочлі і фон Нейману. Ще до завершення робіт над ENIAC вони приступили до нового проекту – EDVAC, головною особливістю якого стала концепція програми, що зберігалася в пам'яті. Ця концепція на довгі роки визначила базові принципи побудови подальших поколінь обчислювальних машин. Щодо авторства існує декілька версій, але оскільки в закінченому вигляді ідея вперше була викладена в 1945 році в статті фон Неймана, саме його прізвище фігурує в позначенні архітектури подібних машин, які складають переважну частину сучасного парку ОМ і ОС.

Суть фон-нейманівської концепції обчислювальної машини можна звести до чотирьох принципів: двійкового кодування; програмного управління; однорідності пам'яті; адресності.

Принцип двійкового кодування. Згідно з цим принципом, вся інформація, як дані, так і команди, кодуються двійковими цифрами 0 і 1. Кожен тип інформації подається двійковою послідовністю і має свій формат. Послідовність бітів у форматі, що має певний сенс, називається полем. У числовій інформації зазвичай виділяють поле знака і поле значущих розрядів. У форматі команди можна виділити два поля (рис. 2.1): поле коду операції (КОп) і поле адрес (адресну частину – АЧ).

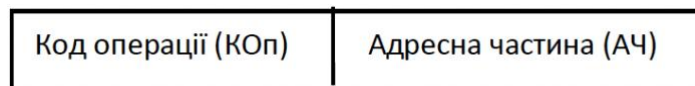


Рисунок 2.1 – Структура команди

Кодом операції є вказівка, яка операція повинна бути виконана, і задається за допомогою  $r$ -розрядної двійкової комбінації.

Вид адресної частини і число складових її адрес залежать від типу команди: в командах перетворення даних АЧ містить адреси об'єктів обробки (операндів) і результату; в командах зміни порядку обчислень – адреса наступної команди програми; в командах вводу/виводу – номер пристрою вводу/виводу. Адресна частина також подається двійковою послідовністю, довжину якої позначимо через  $p$ . Таким чином, команда в обчислювальній машині має вид  $(r + p)$  – розрядної двійкової комбінації.

**Принцип програмного управління.** Всі обчислення, передбачені алгоритмом розв'язання задачі, повинні бути подані у вигляді програми, що складається з послідовності управляючих слів – команд. Кожна команда передбачає деяку операцію з набору операцій, що реалізуються обчислювальною машиною. Команди програми зберігаються в послідовних комірках пам'яті обчислювальної машини і виконуються в природній послідовності, тобто в порядку їх розміщення в програмі. За необхідності, за допомогою спеціальних команд, ця послідовність може бути змінена. Рішення про зміну порядку виконання команд програми ухвалюється або на підставі аналізу результатів попередніх обчислень, або безумовно.

**Принцип однорідності пам'яті.** Команди і дані зберігаються в одній і тій же пам'яті і зовні не відрізняються. Розпізнати їх можна тільки за способом використання. Це дозволяє проводити над командами ті ж операції, що і над числами, і, відповідно, відкриває ряд можливостей. Так, циклічно змінюючи адресу частину команди, можна забезпечити звернення до послідовних елементів масиву даних. Такий прийом носить назву модифікації команд і з позицій сучасного програмування не є оптимальним. Більш корисним є інший

наслідок принципу однорідності, коли команди однієї програми можуть бути отримані як результат виконання іншої програми. Ця можливість лежить в основі трансляції – перекладу тексту програми з мови високого рівня на мову конкретної ОМ.

Концепція обчислювальної машини, викладена у статті фон Неймана, припускає єдину пам'ять для зберігання команд і даних. Такий підхід був прийнятий в обчислювальних машинах, що створювалися в Принстонському університеті, через що і отримав назву принстонської архітектури. Практично одночасно в Гарвардському університеті запропонували іншу модель, в якій ОМ мала окрему пам'ять команд і окрему пам'ять даних. Цей вид архітектури називають гарвардською архітектурою. Довгі роки переважаючою була і залишається принстонська архітектура, хоча вона породжує проблеми пропускної спроможності тракту «процесор - пам'ять». Останнім часом у зв'язку з широким використанням кеш-пам'яті розробники ОМ все частіше звертаються до гарвардської архітектури.

**Принцип адресності.** Структурно основна пам'ять складається з пронумерованих комірок, причому процесору в довільний момент доступною є будь-яка комірка. Двійкові коди команд і даних розділяються на одиниці інформації, звані словами, і зберігаються в комірках пам'яті, а для доступу до них використовуються номери відповідних комірок – адреси.

**Фон-нейманівська архітектура.** У статті фон Неймана визначені основні пристрої ОМ, за допомогою яких повинні бути реалізовані вищеперелічені принципи. Більшість сучасних ОМ за своєю структурою відповідають принципу програмного управління. Типова фон-нейманівська ОМ містить: пам'ять, пристрій управління, арифметико-логічний пристрій і пристрій вводу/виводу (рис. 2.2).

У будь-якій ОМ є засоби для введення програм і даних до них. Інформація поступає з приєднаних до ЕОМ периферійних пристроїв (ПП) введення.

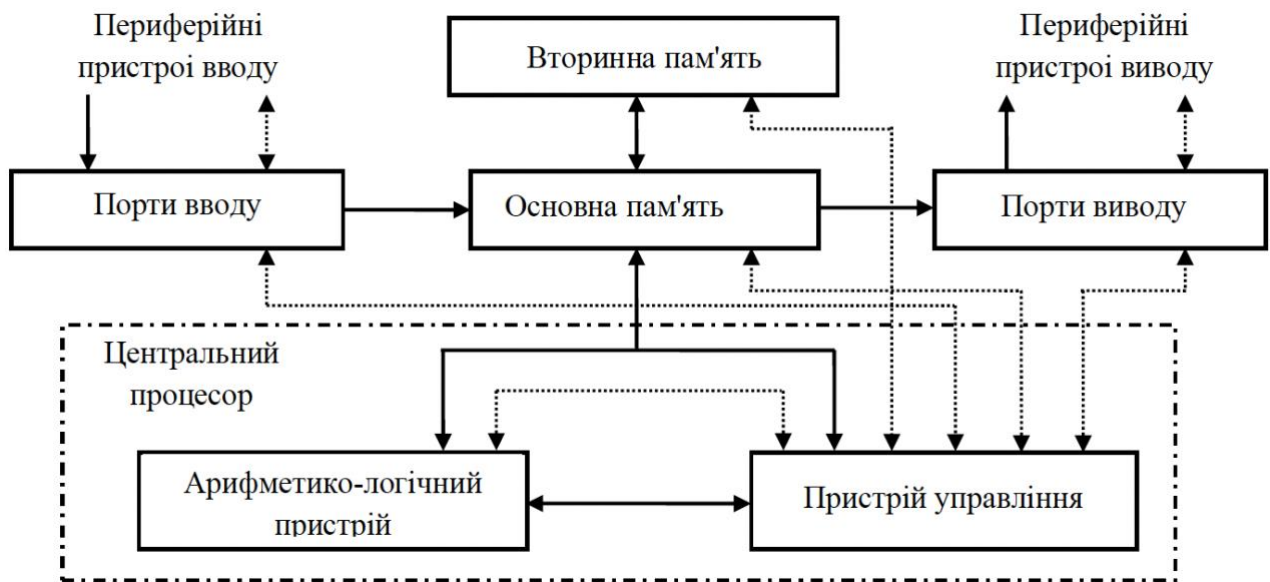


Рисунок 2.2 – Структура фон-нейманівської обчислювальної машини

Результати обчислень виводяться на периферійні пристрої виводу. Зв'язок і взаємодія ОМ і ПП забезпечують порти вводу і порти виводу. Терміном порт позначають апаратуру сполучення периферійного пристрою з ОМ і управління ним. Сукупність портів вводу і виводу називають пристроєм вводу/виводу (ПВВ) або модулем вводу/виводу ОМ (МВВ).

Введена інформація спочатку запам'ятовується в основній пам'яті, а потім переноситься у вторинну пам'ять, для тривалого зберігання. Щоб програма могла виконуватися, команди і дані повинні розташовуватися в основній пам'яті (ОП), організованій таким чином, що кожне двійкове слово зберігається в окремій комірці, що ідентифікується адресою, причому сусідні комірки пам'яті мають наступні за порядком адреси. Доступ до будь-яких комірок запам'ятовуючого пристрою (ЗП) основної пам'яті може проводитися в довільній послідовності. Такий вид пам'яті відомий як пам'ять з довільним доступом. ОП сучасних ОМ в основному складається з напівпровідникових оперативних запам'ятовуючих пристроїв (ОЗП), які забезпечують як зчитування, так і запис інформації. Для таких ЗП характерна енергозалежність – інформація, що зберігається, втрачається при відключенні електроживлення. Якщо необхідно, щоб частина основної пам'яті була енергонезалежною, в склад ОП включають постійні

запам'ятовуючі пристрої (ПЗП), які також забезпечують довільний доступ. Інформація, що зберігається в ПЗП, може тільки зчитуватися (але не записуватися).

Розмір комірки основної пам'яті зазвичай приймається рівним 8 двійковим розрядам – байту. Для зберігання великих чисел використовуються 2, 4 або 8 байтів, що розміщуються в комірках з послідовними адресами.

Для довготривалого зберігання великих програм і масивів даних в ОМ зазвичай є додаткова пам'ять, відома як вторинна. Вторинна пам'ять енергонезалежна і найчастіше реалізується на базі магнітних дисків. Інформація в ній зберігається у вигляді спеціальних програмно підтримуваних об'єктів – файлів (згідно зі стандартом ISO, файл – це «ідентифікована сукупність екземплярів повністю описаного в конкретній програмі типу даних, що знаходяться поза програмою в зовнішній пам'яті і доступних програмі за допомогою спеціальних операцій») [25].

Пристрій управління (ПУ) – найважливіша частина ОМ, що організує автоматичне виконання програм (шляхом реалізації функцій управління) і забезпечує функціонування ОМ як єдиної системи. Для пояснення функцій ПУ ОМ слід розглядати як сукупність елементів, між якими відбувається пересилання інформації, в ході якої ця інформація може піддаватися певним видам обробки. Пересилання інформації між будь-якими елементами ОМ ініціюється своїм сигналом управління (СУ) в потрібній тимчасовій послідовності. Ланцюги СУ показані на рис. 1.2 півтоновими лініями. Основною функцією ПУ є формування управляючих сигналів, що відповідають за витягання команд з пам'яті в порядку, який визначається програмою, і подальше виконання цих команд. Крім того, ПУ формує СУ для синхронізації і координації внутрішніх і зовнішніх пристроїв ОМ.

Ще однією невід'ємною частиною ОМ є арифметико-логічний пристрій (АЛП). АЛП забезпечує арифметичну і логічну обробку двох вхідних змінних, в результаті якої формується вихідна змінна. Функції АЛП зазвичай зводяться до простих арифметичних і логічних операцій, а також операцій зсуву. Крім

результату операції АЛП формує ряд ознак результату (прапорів), що характеризують отриманий результат і події, які сталися в процесі його отримання (рівність нулю, знак, парність, перенесення, переповнення і т. д.). Прапори можуть аналізуватися в ПУ з метою ухвалення рішення про подальшу послідовність виконання команд програми.

ПУ і АЛП тісно взаємозв'язані і їх зазвичай розглядають як єдиний пристрій, відомий як центральний процесор (ЦП) або просто процесор. Крім ПУ і АЛП в процесор входить також набір регістрів загального призначення (РЗП), які виконують функцію проміжного зберігання інформації в процесі її обробки.

## **2.2 Принцип дії фон-нейманівської електронної обчислювальної машини**

Програма у фон-нейманівській ЕОМ реалізується центральним процесором (ЦП) за допомогою послідовного виконання створюючих цю програму команд. Дії, потрібні для вибірки і виконання команди, називають циклом команди. В загальному випадку цикл команди включає декілька складових (етапів):

- вибірку команди;
- формування адреси наступної команди;
- декодування команди;
- обчислення адрес операндів;
- вибірку операндів;
- виконання операції; запис результату.

Перераховані етапи виконання команди надалі називатимемо стандартним циклом команди. Відзначимо, що не всі з етапів присутні під час виконання будь-якої команди (залежить від типу команди), проте етапи вибірки, декодування, формування адреси наступної команди і виконання мають місце завжди.

Стисло охарактеризуємо кожен з вищеперелічених етапів стандартного циклу команди.

**Етап вибірки команди.** Цикл будь-якої команди починається з того, що центральний процесор витягує команду з пам'яті, використовуючи адресу, що зберігається в лічильнику команд (ЛК). Двійковий код команди поміщається в регістр команди (РК) і з цієї миті стає «видимим» для процесора. Лічильник команд і регістр команд розташовані в пристрої управління. Система команд багатьох ОМ припускає декілька форматів команд, причому в різних форматах команда може займати 1, 2 або більше комірок. У цьому випадку етап вибірки команди можна вважати завершеним лише після того, як в РК буде поміщений повний код команди. Інформація про фактичну довжину команди міститься в полях коду операції і способу адресації. Зазвичай ці поля розташовують у першому слові коду команди, і для з'ясування необхідності продовження процесу вибірки необхідне попереднє декодування їх вмісту. Таке декодування може бути проведене після того, як перше слово коду команди опиниться в РК. У разі багатослівного формату команди процес вибірки продовжується аж до занесення в РК всіх слів команди.

**Етап формування адреси наступної команди.** Для фон-нейманівських машин характерне розміщення сусідніх команд програми в суміжних комірках пам'яті. Якщо вибрана з пам'яті команда не порушує природного порядку виконання програми, то для обчислення адреси наступної виконуваної команди досить збільшити вміст лічильника команд на довжину поточної команди, яка подана кількістю займаних кодом команди комірок пам'яті. Довжина команди, а також те, чи здатна вона змінити природний порядок виконання команд програми, з'ясовуються в ході раніше згадуваного попереднього декодування. Якщо вибрана з пам'яті команда здатна змінити послідовність виконання програми (команда умовного або безумовного переходу, виклику процедури і т. п.), процес формування адреси наступної команди переноситься на етап виконання операції. В силу сказаного, в ряді ОМ даний етап циклу команди йде не за вибіркою команди, а знаходиться в кінці циклу.

**Етап декодування команди.** Після вибірки команди вона повинна бути декодована, для чого ЦП розшифровує код команди, що знаходиться в РК. В результаті декодування з'ясовуються такі моменти:

- чи знаходиться в РК повний код команди або потрібне дозавантаження решти слів команди;
- які подальші дії потрібні для виконання даної команди;
- якщо команда використовує операнди, то звідки вони повинні бути узяті (номер регістра або адреса комірки основної пам'яті);
- якщо команда формує результат, то куди цей результат повинен бути направлений.

Відповіді на два перші питання дає розшифровка коду операції, результатом якої може бути унітарний код, де кожен розряд відповідає одній з команд. На практиці замість унітарного коду можуть зустрітися найрізноманітніші форми подання результатів декодування, наприклад адреса комірки спеціальної управляючої пам'яті, де зберігається перша мікрокоманда мікропрограми для реалізації вказаної в команді операції.

Повне з'ясування всіх аспектів команди, крім розшифровки коду операції, вимагає також аналізу адресної частини команди, включаючи поле способу адресації.

За наслідками декодування проводиться підготовка електронних схем ОМ до виконання наказаних командою дій.

**Етап обчислення адрес операндів.** Етап має місце, якщо в процесі декодування команди з'ясовується, що команда використовує операнди. Якщо операнди розміщуються в основній пам'яті, здійснюється обчислення їх виконавчих адрес, з урахуванням вказаного в команді способу адресації. Так, у разі індексної адресації для отримання виконавчої адреси проводиться підсумовування вмісту адресної частини команди і вмісту індексного регістра.

**Етап вибірки операндів.** Обчислені на попередньому етапі виконавчі адреси використовуються для зчитування операндів з пам'яті і занесення в певні регістри процесора. Наприклад, у разі арифметичної команди операнд після



вибору з пам'яті може бути завантажений у вхідний регістр АЛП. Проте частіше операнди заздалегідь заносяться в спеціальні допоміжні регістри процесора, а їх пересилка на вхід АЛП відбувається на етапі виконання операції.

**Етап виконання операції.** На цьому етапі в АЛП реалізується вказана в команді операція. Через відмінність суті кожної з команд ОМ, зміст цього етапу також суто індивідуальний.

**Етап запису результату.** Етап запису результату присутній в циклі тих команд, які припускають занесення результату в регістр або комірку основної пам'яті. Якщо отриманий результат буде використаний в наступній команді, то він може залишатися в регістрі АЛП.

### **2.3 Етапи розвитку архітектури фон неймана**

У 1946 році відомий американський математик Дж. фон Нейман вперше сформулював основні принципи побудови ЕОМ, що програмно управляються:

- 1) принцип програмного управління – ЕОМ може автоматично перетворювати вихідні дані відповідно до заданої програми;
- 2) принцип умовного переходу – надає гнучкість та універсальність програм за рахунок забезпечення можливості переходу в процесі розв'язання задачі на певну ділянку програми залежно від результатів проміжних обчислень або вихідних даних;
- 3) принцип збереженої програми – програму розміщують у запам'ятовуючій пристрій ЕОМ;
- 4) принцип довільного доступу до елементів пам'яті;
- 5) принцип використання двійкової системи числення;
- 6) принцип багаторівневої (ієрархічної) пам'яті.

Ці принципи актуальні і для сучасних ЕОМ, але зі створенням нових поколінь та сімейств машин вони доповнювались та уточнювались.

В ЕОМ, починаючи з третього покоління, додатково застосовуються такі принципи:

- мультипрограмування – сумісне використання різних команд однієї й тієї ж або різних, незалежно одна від одної, програм, які зберігаються в оперативній пам'яті;
- інформаційна та програмна сумісність – дає змогу виконувати наявні програми на різних моделях сімейства;
- високий рівень технічної стандартизації – єдина для всіх машин номенклатура зовнішніх та інших пристроїв;
- можливість організації багатоетапної роботи зі створення та удосконалення ЕОМ.

Комп'ютери четвертого покоління будуються за принципами:

- багатопроцесорності – комутація декількох процесорів під час роботи зі спільною пам'яттю;
- організації віртуальної пам'яті – забезпечує практично необмежений об'єм адресного простору ОЗП;
- широкого використання ВІС та НВІС і макромодульної структури, в основі якої – ідея побудови з великих стандартизованих блоків (макромодулів) функціонально гнучких обчислювальних систем;
- використання внутрішніх мов високого рівня.

Комп'ютери п'ятого покоління відрізняються:

- подальшим розвитком функції введення-виведення графіки, зображень, документів, мови;
- можливістю діалогової обробки інформації за допомогою природної мови;
- здатністю до самовдосконалення, до асоціативних побудов та отримання висновків.

Мови програмування в процесі форматування програм можуть реалізовувати безпосередній інтерфейс між людиною і машиною. Це мови надвисокого рівня, які забезпечують цю можливість:

- підтриманням засобів верифікації та підвищенням загальної надійності програм;

- забезпеченням розумної взаємодії користувачів з обчислювальною системою на різному рівні доступу до бази даних для вибору потрібної інформації та до бази знань для отримання нових уявлень, необхідних для розв'язання незнайомих задач;
- використанням існуючих програмних фондів, орієнтованих на традиційну архітектуру ЕОМ.

## **2.4 Структурна схема IBM PC – сумісного персонального комп'ютера**

Персональний комп'ютер (ПК) є сукупністю двох важливих частин: апаратного та програмного забезпечення. На сьогоднішній день найбільше розповсюдження отримали IBM PC – сумісні персональні комп'ютери, структурна схема яких подана на рис. 2.3.

Архітектура PC – сумісного комп'ютера визначається рядом властивостей, які забезпечують можливість функціонування програмного забезпечення, що управляє периферійним обладнанням. Програми можуть взаємодіяти з пристроями різними способами:

- через виклики функцій операційної системи;
- через виклики функцій базової системи вводу/виводу (BIOS);
- безпосередньо взаємодіючи з відомими їм пристроями.

Апаратне забезпечення персонального комп'ютера складається з таких частин:

- системний блок комп'ютера – пристрій розміщений в компактному металевому чи пластмасовому корпусі, в якому містяться всі електронні компоненти ПК, а також блок живлення;
- клавіатура – основний пристрій ручного введення інформації в ПК;
- монітор – найголовніший пристрій вводу/виводу візуальної інформації ПК.

Крім того, в ПК можна використовувати додаткові пристрої вводу/виводу інформації (ПВВ) і засоби покращання якості електроживлення (наприклад,

БДЖ – безперебійне джерело живлення для захисту від раптового зникнення напруги в мережі).

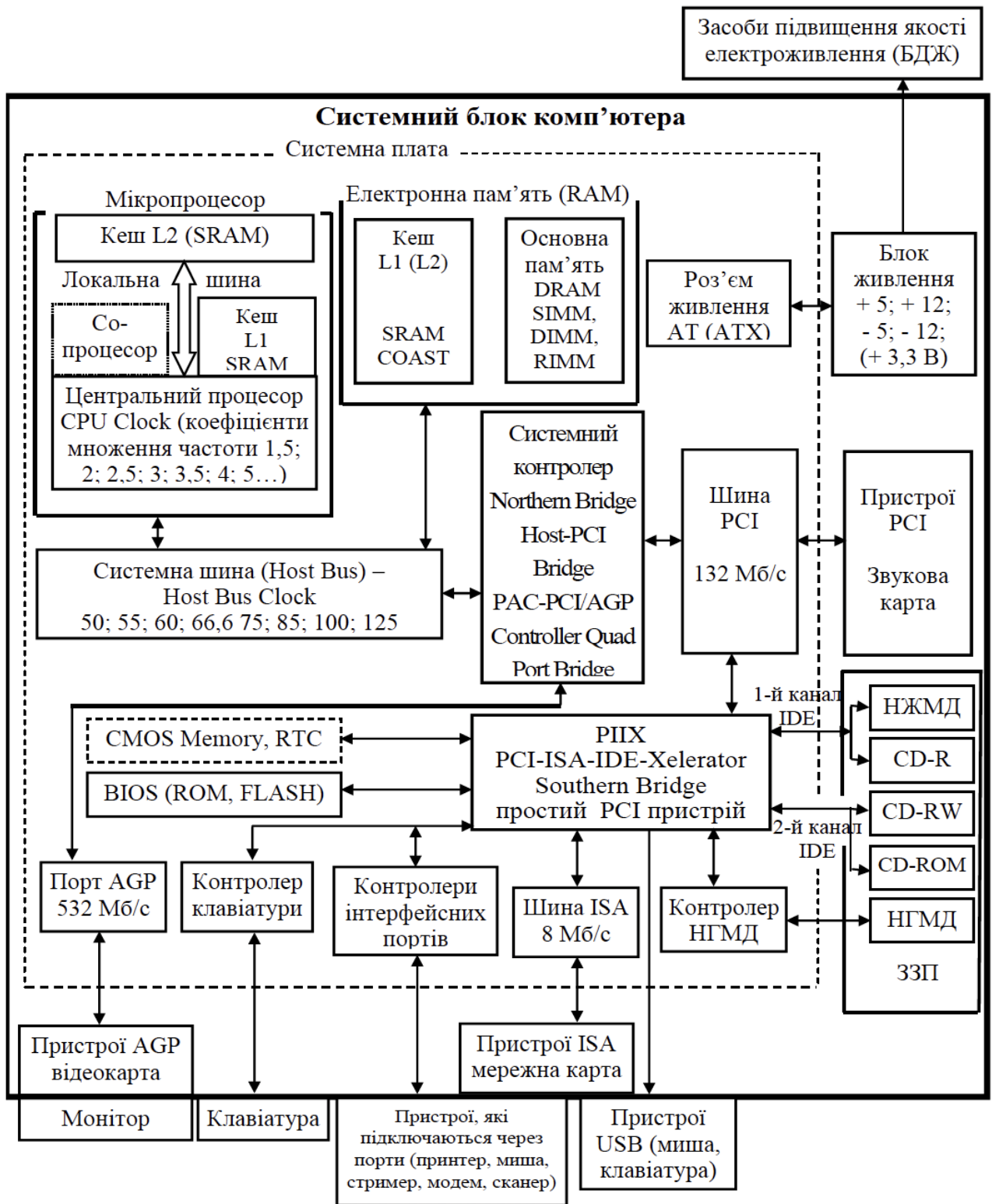


Рисунок 1.3 – Структурна схема IBM PC – сумісного комп'ютера

У системному блоці (рис. 2.3) розташовуються такі компоненти:

Системна плата (об'єднуюча, материнська) є основним компонентом комп'ютера. Вона, як правило, містить:

- сокет (слот) процесора;
- сокети (слоти) пам'яті SIMM і DIMM;
- слоти шини; ПЗП BIOS (ROM BIOS);
- кеш 2-го рівня (кеш L2, Level 2 cache);
- набір мікросхем системної логіки (чіпсет) системної плати;
- чіп вводу/виводу;
- перетворювачі напруги живлення процесора;
- батарею для живлення годинника і CMOS.

Набір мікросхем системної логіки містить усі схеми, які входять до складу системної плати. Він управляє центральним процесором, системною шиною процесора, кешем L2, оперативною пам'яттю, шиною PCI (Peripheral Component Interconnect), шиною ISA (Industry Standard Architecture) чи іншою, ідентичною за функціональним призначенням, ресурсами системи та ін.

Набір мікросхем системної логіки визначає також первинні можливості та специфікації системної плати – типи підтримуваних процесорів, пам'яті, плат розширення, дисководів та ін.

Системні плати випускають у кількох варіантах, які відрізняються розмірами та форм-факторами. Форм-фактор системної плати визначає тип корпусу, в який її можна вставити.

Мікропроцесор (МП) – це основний «мозковий» вузол ПК, в задачу якого входить виконання програмного коду, що знаходиться в пам'яті, і керування іншими пристроями. До складу мікропроцесора сучасного ПК крім центрального мікропроцесора – пристрою обробки CPU (Central Processing Unit) – входить математичний співпроцесор CoP (CoProcessor), який ефективно обробляє числові дані у форматі з плаваючою комою (крапкою), і невелика за обсягом швидкодіюча кеш-пам'ять, реалізована за одно- або дворівневою схемою (від англ. cache – склад, схованка; кеш-пам'ять призначена для пом'якшення

наслідків, що викликані розузгодженням швидкості роботи швидкого МП і повільної оперативної пам'яті).

Мікропроцесор – це найчастіше одна надвелика інтегральна схема (НВІС), реалізована в єдиному напівпровідниковому кристалі. Ступінь інтеграції визначається розміром кристала і кількістю реалізованих у ньому транзисторів. Деякі МП, строго кажучи, не є однокристальними – кристал центрального процесора зі співпроцесором і кілька кристалів вторинного кеша зібрані на загальному картриджі.

У багатопроцесорній системі для підвищення загальної продуктивності системи функції центрального процесора розподіляються між декількома, звичайно ідентичними мікропроцесорами, один із яких призначається головним.

Співпроцесор – це спеціалізований процесор, призначений для розвантаження центрального процесора від складних обчислень з плаваючою комою. Співпроцесор в МП класу 486 і старше вбудований у середину кристала мікропроцесора.

Кеш-пам'ять першого рівня (Level 1 – L1) у процесорів класу 486 і старше вбудована у середину кристала і працює на однаковій з ним частоті. Якщо МП використовує дворівневу модель кеш-пам'яті (L1, L2), то застосовується архітектура подвійної незалежної шини (ПНШ – Dual Independent Bus). Одна з шин МП архітектури ПНШ – локальна шина – використовується тільки для зв'язку з кристалами вторинного кеша, які розташовані у тому ж корпусі мікросхеми або на загальному картриджі. Ця шина є локальною і у геометричному значенні – провідники мають довжину одиниць сантиметрів, що дозволяє її використовувати навіть на частоті ядра процесора.

Друга шина МП виходить на зовнішні виводи мікросхеми, вона є системною шиною МП. Ця шина працює на зовнішній частоті незалежно від внутрішньої шини.

МП керує роботою ПК, отримуючи і посылаючи керуючі сигнали, адреси пам'яті і дані від одних компонентів ПК до інших компонентів, використовуючи

для цього групу сполучаючих електронних шляхів, які називаються системною шиною (СШ).

Системна шина - це електронний шлях на системній платі, що спільно використовується, до якого підключені всі керовані компоненти ПК. Коли дані передаються від одного компонента до іншого, вони переміщуються вздовж цього загального шляху до місця призначення. СШ поділяється на чотири складові частини: лінії передачі електроживлення, керуюча шина (для передачі керуючої інформації, наприклад, такої, як тактові сигнали від системного генератора тактових сигналів, сигнали переривання і т.д.), адресна шина (виконує передачу адрес комірок пам'яті і пристроїв, приєднаних до шини) і шина даних (разом із шиною адреси здійснює перенесення даних в середині ПК).

Оскільки швидкодія різних компонентів ПК (мікропроцесора, пам'яті й інших пристроїв) істотно розрізняється, в комп'ютерах на МП класу 486 і старше застосовується внутрішнє множення частоти. Розрізняють такі частоти:

- Host Bus Clock – частота системної шини (зовнішня частота процесора), опорна для всіх інших частот. МП класу Pentium і старше використовують частоти 50, 55, 60, 66,6, 75, 83, 100, 125 МГц і вище.
- CPU Clock чи Core Speed - внутрішня частота МП, на якій працює його обчислювальне ядро (центральний процесор, співпроцесор, кеш-пам'ять L1). Сучасні технології дозволили істотно підвищити граничні частоти інтегральних компонентів, у зв'язку з чим широко застосовується внутрішнє множення частоти на 1,5, 2, 2,5, 3, 3,5, 4 і деякі інші значення.

Основна чи оперативна пам'ять ПК призначена для оперативного обміну (збереження, запису і зчитування) інформацією (кодами і даними) між МП, зовнішньою пам'яттю і периферійними пристроями. Для побудови основної пам'яті в ПК використовують мікросхеми динамічної пам'яті (мікросхеми DRAM – пам'яті – Dynamic Random Access Memory), що мають найкраще сполучення обсягу, щільності упакування, енергоспоживання і ціни. Мікросхеми DRAM – пам'яті в сучасному ПК установлюють на спеціальні модулі пам'яті SIMM (Single

In – Line Memory Module), DIMM (Dual In – Line Memory Module), RIMM (Rambus In – Line Memory Module) у відповідні гнізда системної плати.

Важливою особливістю основної пам'яті ПК є її ієрархічний спосіб побудови, що прийшов в архітектуру ПК з появою процесора 386, і полягає в сполученні основної пам'яті великого обсягу на мікросхемах динамічної пам'яті з відносно невеликою кеш-пам'яттю на швидкодіючих мікросхемах статичної пам'яті SRAM (Static RAM).

Кеш-пам'ять є додатковим і швидкодіючим сховищем копій блоків інформації основної пам'яті, до яких, імовірно, найближчим часом буде звернення. У сучасному комп'ютері кеш-пам'ять побудована за трирівневою схемою:

- кеш L1 – кеш на системній платі 386 процесорів, працює на Host Bus Clock;
- кеш, вбудований в кристал мікропроцесора класу 486 і старше, працює на CPU Clock;
- кеш L2 – кеш на системній платі мікропроцесора класу 486 і старше (за винятком МП Pentium Pro, Pentium II Xeon, Pentium II, Celeron 300A і старше, K6–3 і їхніх мобільних варіантів) працює на Host Bus Clock;
- кеш вбудований в корпус МП чи встановлений на загальному картриджі (для МП Pentium Pro, Pentium II Xeon, Pentium II, Celeron 300A і старше, K6–3) працює на CPU Clock чи на половині цієї частоти;
- кеш L3 – кеш на системній платі мікропроцесора K6–3 працює на Host Bus Clock.

Кеш на системній платі сучасного ПК набирається мікросхемами статичної пам'яті фіксованого обсягу, запаяних на плату без застосування додаткових модулів і роз'ємів (для старих системних плат із МП Pentium широке поширення отримали модулі COAST (Cache On A Stick) – «кеш на полиці»; модуль із двостороннім друкованим роз'ємом, який встановлений у спеціальний слот).

CMOS Memory (Complementary Metal-Oxide-Semiconductor), RTC (Real Time Clock) – спеціальна мікросхема напівпостійної пам'яті (КМОП – пам'ять)



невеликого обсягу для збереження інформації про конфігурацію ПК разом з годинником і календарем. Живлення CMOS Memory, RTC при виключеному ПК здійснюється від батарейки.

BIOS (Basic Input Output System) – ключовий елемент системної плати, призначений для енергонезалежного збереження системної інформації. BIOS, користуючись засобами, наданими чипсетом, керує всіма компонентами і ресурсами системної плати. Код BIOS зберігається в мікросхемі енергонезалежної постійної пам'яті (Read Only Memory (ROM BIOS) чи флеш-пам'яті (Flash BIOS).

Шини розширення призначені для підключення різних адаптерів чи контролерів периферійних пристроїв, що розширюють можливості ПК. Під адаптером звичайно розуміють засіб сполучення якого-небудь пристрою з шиною ПК (контролер служить тим же цілям сполучення, але при цьому має на увазі його деяка активність, тобто здатність до самостійних дій після отримання команд від обслуговуючої його програми).

В сучасному ПК застосовують такі шини, розташовані на системній платі у виді слотів чи секцій розширення:

- PCI – Peripheral Component Interconnect Local Bus;
- AGP – Accelerated Graphic Port – прискорений графічний порт;
- ISA – Industry Standard Architecture.

Зовнішні інтерфейси ПК (рівнобіжний, послідовний і USB-шина), як і шини розширення, дозволяють значно розширити функціональні можливості ПК. Однак, на відміну від шин розширення, зовнішні інтерфейси дозволяють підключати різні периферійні пристрої прямо, без використання додаткових адаптерів.

Сучасні системні плати будують на основі чипсетів (Chipset) – набір з декількох надвеликих інтегральних схем, що реалізують усі необхідні функції зв'язку основних компонентів: МП, пам'яті і шин розширення. Майже всі сучасні чипсети є набором із двох мікросхем, що прийнято називати Northern Bridge (Північний міст) і Southern Bridge (Південний міст).

Northern Bridge відповідає за роботу з МП, системною шиною, PCI-шиною, AGP-портом, пам'яттю і кешем (які не входять до складу МП).

Southern Bridge – це фактично простий PCI – пристрій, що містить всередині себе контролер Bus Master IDE (який дозволяє пристроям, що знаходяться на такій шині, самим керувати процесом передачі даних по ній без участі процесора), міст PCI–ISA.

Периферія досить стабільна, тому архітектура Southern Bridge набагато менше піддається змінам, чим Northern Bridge, що дозволяє в нових чипсетах використовувати його попередні версії.

Міст PCI–IDE потрібний для роботи з існуючими накопичувачами з жорсткими магнітними дисками (НЖМД), міст PCI–ISA є рудиментом, що відмирає. Справа в тому, що дотепер BIOS, контролер гнучких дисків, послідовні і рівнобіжні порти, інші пристрої, яким цілком достатньо невеликої пропускної здатності, знаходяться на шині ISA. Однак уже відбувається відмова від старої шини ISA і повний перехід на PCI.

Контролери накопичувачів на гнучких магнітних дисках (НГМД), інтерфейсних портів, клавіатури, CMOS Memory, RTC можуть входити власне в чипсет, а можуть бути реалізовані на окремих «сторонніх» мікросхемах.

Зовнішні запам'ятовуючі пристрої комп'ютера (ЗЗП) – це пристрої, що дозволяють автономно зберігати інформацію для наступного її використання незалежно від стану ПК (включений чи виключений). В сучасному ПК в ЗЗП входять пристрої магнітної, оптичної і магнітооптичної пам'яті. ЗЗП можна розміщувати як у системному блоці комп'ютера, так і в окремому корпусі.

Блок живлення призначений для перетворення змінного електричного струму в постійний, який стабілізований у невеликих межах, а також для захисту електричних ланцюгів блока живлення і комп'ютера від впливу різних перешкод і несправностей.