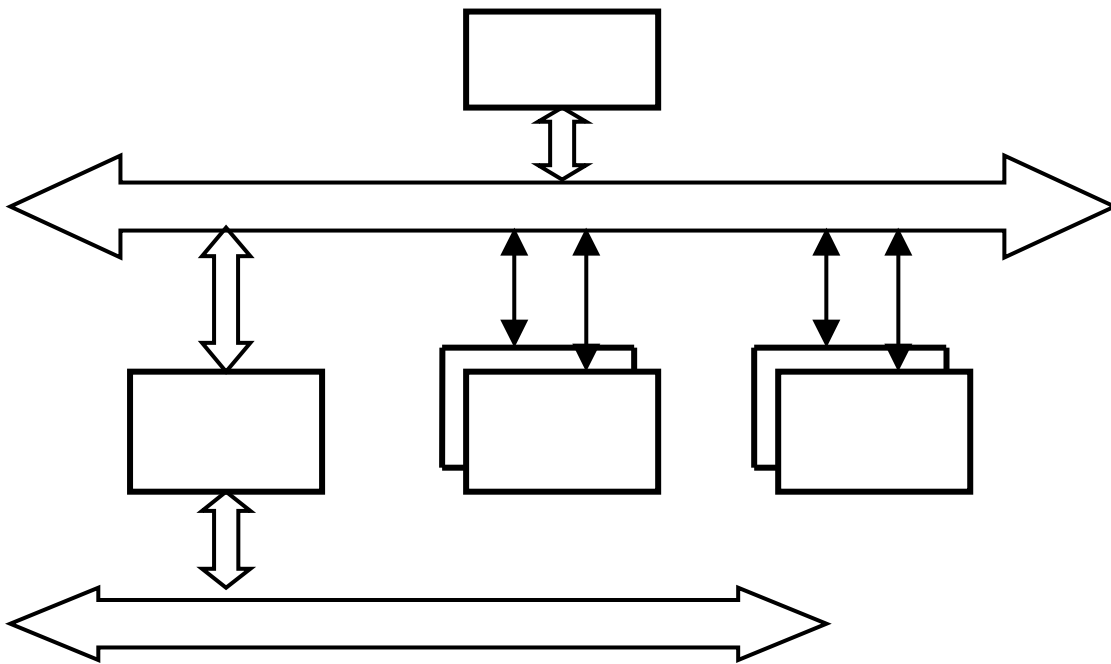


А. Карачка, О. Дудко



АРХІТЕКТУРА КОМП'ЮТЕРІВ

НАВЧАЛЬНИЙ ПОСІБНИК



*До 50-річчя
Тернопільського національного
економічного університету*

Тернопіль

ТНЕУ
2010

Міністерство освіти і науки України
Тернопільський національний економічний університет

А. Карачка, О. Дудко

АРХІТЕКТУРА КОМП'ЮТЕРІВ

За редакцією доктора технічних наук, професора А. О. Саченка

Економічна думка
Тернопіль

2010

Карачка А. Ф., Дудко О. І. Архитектура комп'ютерів: Навч. посіб. / За ред. А. О. Саченка. – Тернопіль: Економічна думка, 2009. – 120 с.

Пропонований навчальний посібник присвячений питанням організації і функціонування сучасних комп'ютерів та їхніх основних складових частин. Розглянуто структуру і функціонування комп'ютерів нейманівського типу, принципи організації шин, внутрішньої та зовнішньої пам'яті, операційних пристроїв і систем введення-виведення. Висвітлено основні тенденції в архітектурі сучасних процесорів, принципи побудови й архітектурні особливості комп'ютерів ненеіманівського типу, принципово нові архітектури комп'ютерів і особливості їхньої побудови. Дано визначення поняття «нейрокомп'ютер» і описано його архітектурні особливості.

Посібник буде корисним студентам і викладачам цієї дисципліни, а також слухачам інститутів післядипломної освіти та курсів підвищення кваліфікації спеціалістів.

Рецензенти:

доктор технічних наук, професор *М. О. Корчемний*;
доктор технічних наук, професор *М. В. Приймак*

Рекомендовано до друку Вченою радою Тернопільського національного економічного університету (протокол №7 від 25.11.2009р.)

© А. Ф. Карачка, 2010

© О. І. Дудко, 2010

ЗМІСТ

1. ЗАГАЛЬНІ ПОНЯТТЯ ПРО АРХІТЕКТУРУ КОМП'ЮТЕРІВ.....	6
Визначення поняття «архітектура» комп'ютера.....	6
1.2. Історичний огляд архітектур комп'ютерів і їхня класифікація.	10
1.3. Особливості архітектури найманівського комп'ютера.....	14
1.4. Основні типи комп'ютерів та сфери їхнього застосування.....	15
1.5. Основні характеристики комп'ютерів.....	19
1.6. Вимірювання і визначення порівняльних характеристик продуктивності комп'ютерів.....	20
2. СИСТЕМИ ЧИСЛЕННЯ.....	26
2.1. Системи числення та їхня класифікація.....	26
2.2. Переведення чисел з однієї системи числення в іншу.....	28
2.3. Арифметичні дії над числами в різних системах числення ...	33
3. АЛГОРИТМИ ОБРОБКИ ІНФОРМАЦІЇ ТА ЇХНІ АРХІТЕКТУРНІ РІШЕННЯ.....	35
3.1. Загальні відомості про операції та їхній розподіл на групи...	35
3.2. Алгоритми виконання арифметичних операцій над двійковими числами та їхні архітектурні рішення.....	36
3.3. Алгоритми і пристрої для виконання логічних операцій.....	42
3.4. Особливості виконання операцій десяткової арифметики...	42
3.5. Алгоритм додавання і віднімання чисел з плаваючою крапкою...	45
4. ПОДАННЯ ІНФОРМАЦІЇ В КОМП'ЮТЕРАХ.....	49
4.1. Форми подання чисел і алфавітної інформації в комп'ютера...	49
4.2. Комп'ютерні коди: прямий, обернений і доповняльний.....	51
4.3. Кодування десяткових чисел і алфавітно-цифрової інформації...	54
5. ПАМ'ЯТЬ КОМП'ЮТЕРІВ.....	56
5.1. Ієрархічний принцип побудови пам'яті.....	56
5.2. Види запам'ятовувальних пристроїв, їхні особливості.....	59
5.3. Віртуальна пам'ять та її організація.....	62
5.4. Молекулярна і голографічна пам'ять та основні принципи їхньої побудови.....	65
5.5. Адресація пам'яті комп'ютерів.....	68
6. АРХІТЕКТУРА СУЧАСНИХ ПРОЦЕСОРІВ.....	74

6.1. Класифікація, архітектура і характеристика сучасних CISC-мікропроцесорів.....	74
6.2. Мікропроцесори з RISC-архітектурою.....	79
6.3. Архітектура і характеристика 64-розрядних мікропроцесорів	104
6.4. Мікропроцесори з довгим командним словом (VLIW).....	114
7. АРХІТЕКТУРА ТРАНСП'ЮТЕРІВ І ТРАНСП'ЮТЕРНИХ СИСТЕМ...	118
7.1. Основні поняття і концепції архітектури трансп'ютерів.....	118
7.2. Типи трансп'ютерів та їхня організація.....	120
7.3. Трансп'ютерна організація паралельних обчислень.....	121
7.4. Мультипроцесорні трансп'ютерні системи.....	123
8. ІНТЕРФЕЙСИ ТА ШИНИ КОМП'ЮТЕРІВ.....	129
8.1. Загальна характеристика інтерфейсів.....	129
8.2. Класифікація інтерфейсів периферійних пристроїв.....	131
8.3. Канали введення-виведення, та їхні функції.....	132
8.4. Мультиплексний і селекторний канали, їхня структура й організація роботи.....	134
8.5. Шини міні-, мікро- і персональних комп'ютерів, їхня організація й характеристика.....	136
9. АРХІТЕКТУРА СУПЕРКОМП'ЮТЕРІВ.....	142
9.1. Поняття про суперкомп'ютери та їхню організацію.....	142
9.2. Класифікація суперкомп'ютерів.....	144
9.3. Типові архітектури суперкомп'ютерів.....	146
10. АРХІТЕКТУРА НЕНЕЙМАНІВСЬКИХ КОМП'ЮТЕРІВ.....	150
10.1. Рекурсивна архітектура.....	150
10.2. Машина Айліфа: її особливості і практичне застосування...	151
10.3. РЕФАЛ: мова і машина	153
10.4. Асоціативні паралельні процесори.....	156
11. АРХІТЕКТУРА НЕЙРОКОМП'ЮТЕРІВ.....	158
11.1. Визначення поняття «нейрокомп'ютер» і передумови його виникнення.....	158
11.2. Архітектурні особливості й апаратне забезпечення нейрокомп'ютерів.	159
11.3. Нейрокомп'ютерні мережі.....	163
ЛІТЕРАТУРА.....	166

ПЕРЕЛІК СКОРОЧЕНЬ

АЛП – арифметико-логічний пристрій
АРМ – автоматизоване робоче місце
АСУ – автоматизована система управління
ВВ – введення-виведення
ЗЗП – зовнішній запам'ятовувальний пристрій
ЗП – запам'ятовувальний пристрій
ІБ – інтерфейсний блок
К – контролер
КБ – керуючий блок
КВВ – канал введення-виведення
КОП – код операції
КСК – керуюче слово каналу
МК – мультиплексний канал
МП – мікропроцесор
МТС – мультитрансп'ютерна система
НЖМД – накопичувач на жорсткому магнітному диску
НМД – накопичувач на магнітному диску
НМС – накопичувач на магнітній стрічці
НОЗП – надоперативний запам'ятовувальний пристрій
ОЗП – оперативний запам'ятовуючий пристрій
ОП – оперативна пам'ять
ОС – операційна система
ПК – персональний комп'ютер
ПП – периферійний пристрій
ППЗП – перепрограмовуючий запам'ятовувальний пристрій
ППК – процесор з плаваючою крапкою
ПУКМ – пристрій управління конфігурацією мережі
РЗП – регістр загального призначення
РОМ – рекурсивна обчислювальна машина
РПК – регістр плаваючої крапки
СК – селекторний канал
СОЛО – схема однобайтних логічних операцій
ФБ – функціональний блок

ФП – функціональний виконавчий пристрій
ЦП – центральний процесор
ШНС – штучна нейронна система

1. ЗАГАЛЬНІ ПОНЯТТЯ ПРО АРХІТЕКТУРУ КОМП'ЮТЕРІВ

1.1. Визначення поняття «архітектура» комп'ютера

Спеціалісти з архітектури комп'ютерів називаються архітекторами комп'ютерів чи комп'ютерних систем.

Завданням такого архітектора є розробка комп'ютера, який відповідав би конкретним вимогам його користувачів чи загальним вимогам сьогодення. Користувачі комп'ютерів, як правило, хочуть мати машину (комп'ютер) якомога більш універсальну, яка може виконувати найскладніші завдання, супершвидкодіючу, високонадійну і дешеву.

У комбінації одні з одними ці вимоги є взаємовиключними. Наприклад, якщо намагатися збільшити продуктивність машини, зберігши при цьому її функціональні можливості, то, беззаперечно, зростає її вартість. Аналогічно можна сказати і про інші перераховані вимоги чи їхні комбінації.

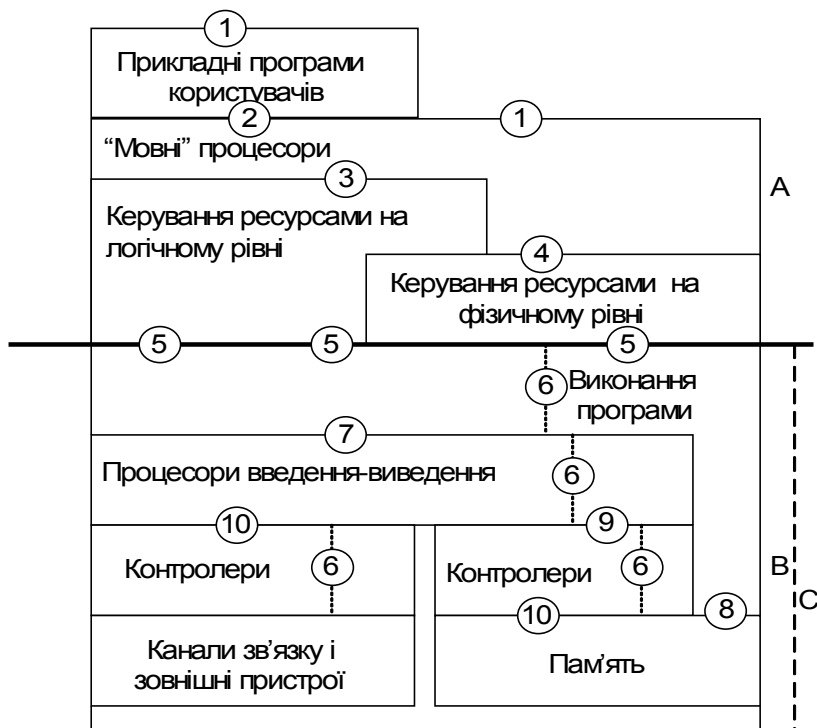
Архітектори комп'ютерів ставлять перед собою завдання вдосконалення й оновлення парку комп'ютерів і побудованих на їхній основі обчислювальних систем у кожному з перерахованих напрямків та водночас намагаються створити систему з оптимальним співвідношенням вимог, тобто оптимальне співвідношення є «пошуком компромісу», а для архітекторів комп'ютерів – важливим видом діяльності.

Щодо комп'ютерів і обчислювальних систем (комп'ютерних систем) термін «архітектура» може бути визначений як розподіл функцій, які реалізуються комп'ютером (системою), за окремими його (її) рівнями і точне визначення меж між цими рівнями. Таким чином, якщо архітектурі відведений деякий рівень, то передусім необхідно встановити, які системні функції повністю або частково виконуються компонентами комп'ютера (системи), що знаходяться вище і нижче від заданого рівня. Після виконання цього завдання наступний крок полягає у точному визначенні інтерфейсів для рівня, який розглядається.

Таким чином, архітектура комп'ютера (комп'ютерної системи) передбачає багаторівневу організацію. Архітектуру комп'ютера (комп'ютерної системи) можна визначити шляхом виявлення її відмінностей від інших видів архітектури. Специфічною властивістю архітектури комп'ютера (комп'ютерної системи) є можливість виділення в ній набору рівнів абстракції (рис. 1.1).

Архітектура першого рівня, що називається *архітектурою системи* (цифра 1), визначає, які функції при обробці даних виконуються системою загалом, а які покладаються на користувачів, операторів

комп'ютерів, адміністраторів мереж, адміністраторів баз даних і т. ін. Система взаємодіє із зовнішнім світом через два набори інтерфейсів: мови і системні програми. Розробка архітектури системи передбачає визначення обох груп інтерфейсів.



A – програмні засоби; B – мікропрограмні засоби; C – апаратні засоби.

Рис. 1.1. Багаторівнева організація архітектури комп'ютерної системи

Інтерфейси 2 – 4 розмежовують певні рівні в середині програмного забезпечення. Якщо програми, які реалізують прикладні завдання, написані мовами програмування, що не входять до числа тих, які надані в розпорядження користувача, то можна вести мову про архітектуру рівня, призначення котрого полягає у визначенні вказаних мов. Транслятори таких мов взаємодіють з нижчими рівнями програмного забезпечення, які позначено на абстрактній моделі архітектури як рівні 3 і 4. Рівень керування логічними ресурсами може охоплювати реалізацію таких функцій, як керування базою даних, файлами, віртуальною пам'яттю, мережною телеобробкою. До рівня керування фізичними ресурсами належать функції керування зовнішньою й оперативною пам'яттю, внутрішніми процесами, що відбуваються в системі (наприклад, процесами планування і синхронізації), а також іншими

апаратними засобами. Рівні 2–4 – це *архітектура програмного забезпечення*.

Рівень 5 відображає основну лінію розмежування системи, а саме межу між системним програмним (операційною системою) і апаратним забезпеченням (мікропрограми й електронні логічні схеми).

Мікропрограма – це записана в пам'яті програма, що фактично керує передачею всіх символів і даних у фізичних компонентах системи, таких як шини, регістри, суматори чи процесор; альтернативним засобом керування передачею сигналів і даних є суто апаратний спосіб (використання комбінацій логічних схем).

Рівень (інтерфейс) 5 дає змогу відображати фізичну структуру системи абстрактно, незалежно від способу реалізації. Розмежування функцій, які виконуються вище і нижче від цього рівня, та визначення інтерфейсу 5 – одна зі складових частин процесу розробки *архітектури комп'ютера*.

Інтерфейс 7 визначає, які функції реалізують ЦП, що виконують програми, а які – процесори введення-виведення (тобто канали).

Архітектура другого рівня визначає розмежування між процесорами введення-виведення і контролерами зовнішніх пристроїв. При цьому можна розмежувати функції, які реалізуються контролерами і самими пристроями введення-виведення. Архітектура таких рівнів (інтерфейси 7, 9, 10) може бути названа *архітектурою фізичного введення-виведення*.

Інтерфейс 6 представляє собою інтерфейс мікропрограми (тобто забезпечує узгодження потоку даних і керуючих сигналів з форматом мікрокоманд) у середині процесора. Архітектура рівня 6 може бути названа *архітектурою мікропрограмного керування*. Архітектуру рівнів 6 і 8 (інтерфейс між процесором та основною пам'яттю) часто називають *архітектурою процесора або організацією процесора*.

Останній різновид архітектури наочно на рис. 1.1 не показано. *Мультипроцесорна архітектура* передбачає розподіл функцій між групою процесорів з розмежуванням функцій комп'ютерної системи між цими процесорами і визначенням відповідних інтерфейсів.

Архітектуру здебільшого розуміють як внутрішню структуру обчислювальної машини (комп'ютера). З цієї причини в Японії та інших країнах важливе значення надають апаратним засобам комп'ютера. Вихідним (першочерговим) завданням архітекторів комп'ютерів, яке було відправною точкою щодо розробки нових архітектурних рішень, було і є завдання модернізації й оновлення шляхом пошуку компромісів між

зростаючими вимогами до комп'ютерів таким чином, щоб можна було максимально їх задовольнити.

Архітектор комп'ютера (інженер-системотехнік) при розробці шукає шляхи вирішення завдань за трьома основними напрямками:

- визначає форму подання програм для комп'ютера і правила інтерпретації програм цього комп'ютера;
- встановлює способи адресації даних цією програмою;
- задає форму подання даних.

У процесі розробки необхідно вирішувати такі завдання: визначення мінімально адресованої області пам'яті, типів і форматів даних, кодів операцій та форматів, машинних команд, способів адресації і захисту пам'яті, механізму управління послідовністю виконання команд, інтерфейсу машини з пристроями введення-виведення.

Завдання етапу розробки архітектури

Розробка архітектури комп'ютера переважно зводиться до встановлення межі між окремими рівнями обчислювальної системи з багаторівневою організацією. Після попереднього визначення функцій, які виконуються системою загалом, завдання архітектора комп'ютера полягає в розподілі функцій системи між її окремими рівнями.

Можна сформулювати такий порядок розробки архітектури комп'ютера:

1. Аналіз вимог, які пред'являються до комп'ютера. На цьому етапі визначається загальна архітектура комп'ютера, встановлюються фактори, що несуть основне функціональне навантаження: кількість і специфіка мов програмування, характеристики периферійних пристроїв; спосіб та характер взаємодії з навколишнім середовищем (наприклад, режим реального часу, обробка даних, розподіл часу і т. ін.); вимоги до операційної системи і т. ін.

2. Складання специфікацій. Тут формулюються критерії, що використовуються, і оцінюється їхня значущість, визначаються функції комп'ютера відповідно до прийнятих критеріїв. Критерії мають враховувати вартість комп'ютера (системи), який проектується, надійність, трудомісткість реалізації, універсальність, здатність до розширення можливостей, швидкодію, простоту програмування і відлагодження програм.

3. Вивчення відомих рішень.

4. Розробка функціональної схеми. Це є основним етапом створення архітектури комп'ютера. На ньому встановлюються співвідношення між рівнями організації комп'ютера (системи) відповідно до необхідних функцій

і вимог, які пред'являються до комп'ютера (системи). Розподіляються функції між апаратними засобами і програмним забезпеченням.

5. Розробка структурної схеми. На цьому етапі строго визначаються всі елементи архітектури комп'ютера, включаючи визначення типів та форматів команд, способи подання даних і адресації.

Щодо комп'ютерів (комп'ютерних систем) термін «архітектура» визначається як розподіл функцій комп'ютера за окремими його рівнями при точному визначенні меж між цими рівнями. Отже, архітектура комп'ютера (комп'ютерної системи) передбачає існування багаторівневої організації.

Щоб описати комп'ютер (систему), необхідно визначити функцію і структуру комп'ютера (системи).

Функція системи – це правило отримання результатів, які визначаються призначенням системи. Інакше кажучи, функція системи виявляється в описі процесів, які відбуваються в цій системі. Функції системи доцільно описувати в математичній формі. Часто функції систем описують у вигляді (формі) алгоритмів.

Структура системи – це фіксована сукупність елементів і зв'язків між ними. Математично вигідно подавати структуру в формі графа. При цьому вершини графа відповідають елементам, а дуги і ребра відображають зв'язки між відповідними елементами.

Поняттям вищого рівня, ніж функція і структура, є *організація*. Один і той самий принцип організації для різних випадків приводить до систем з різними функціями та структурами. Можна зробити висновок, що функція і структура – це конкретизація принципу організації.

Визначимо термін «*архітектура комп'ютера*» дещо вужче у такій формі.

Архітектура комп'ютера – це абстрактне подання фізичної системи (апаратних засобів і мікропрограм) з точки зору програміста, який використовує машинноорієнтовану мову.

Архітектура визначає принципи організації комп'ютера (системи) і функції процесора й не відображає такі процеси, як керування та передавання даних всередині процесора, а також особливості систем елементів і технологію виробництва.

1.2. Історичний огляд архітектур комп'ютерів і їхня класифікація

Поява машини, здатної автоматично виконувати чотири арифметичні дії – додавання, віднімання, множення і ділення, датується початком XVII ст. У 1623 р. В. Шикард винайшов машину, здатну не тільки додавати і віднімати числа, а й частково перемножувати і ділити

їх. Про цю машину тоді мало знали, тому більш відомим вважається настільний арифмометр на основі зубчастого колеса, який у 1642 р. створив французький філософ і вчений Б. Паскаль.

XIX ст. було століттям значних обчислень, які виконувались вручну. Зокрема, це і складання таблиць логарифмів, і розрахунки в астрономії, і багато іншого. З метою автоматизувати процес складання подібних числових таблиць англійський математик Ч. Беббідж у 1823 р. почав працювати над створенням «різницевої машини». За задумом вченого ця машина із зубчастими колесами мала обчислювати значення поліномів і друкувати результати обчислень на негативній пластині для фотодруку.

Нова лічильна машина Беббіджа отримала назву «аналітична», а в 1834 р. він сформулював її основні принципи. За задумом Беббіджа така машина мала автоматично виконувати різні обчислення при послідовному введенні набору перфокарт, які вміщували пари команд і даних, та змінювати розташування отворів на карті, а також порядок розміщення карт відповідно до виконання обчислень. Аналітична машина – це програмована автоматична обчислювальна машина з послідовним управлінням, яка вміщувала арифметичний пристрій і пам'ять.

У другій половині XIX ст. Г. Холлерит розробив машину з введенням з перфокарт, здатну автоматично класифікувати і складати таблиці даних. Ця машина використовувалася в 1890 р. у США при проведенні перепису населення. У 1896 р. Холлерит заснував фірму, яка стала попередницею знаменитої фірми «IBM». Після смерті Беббіджа значного прогресу в дослідному виробництві універсальних обчислень не простежувалося.

У Німеччині в 1938 р. К. Цузе створив механічну обчислювальну машину Z1. В ній, на відміну від наступних машин, які оперували десятковими числами, вперше було використано операції з двійковими числами. Наступна модель Z3 (1941 р.) стала універсальною програнованою обчислювальною машиною, її операційний пристрій був виконаний на електромагнітних реле, і в машині застосовувалися операції плаваючої арифметики.

Швидкість обчислень у механічних машинах на основі зубчастого колеса та в електричних машинах, виконаних на реле, була незначною, тому в кінці 30-х рр. і на початку 40-х рр. XX ст. почалися розробки електронних обчислювальних машин, елементною базою яких стала трьохелектродна електронна лампа. Перший універсальний комп'ютер (прообраз сучасного комп'ютера) (ЕНІАК, 1946 р., США) був призначений для складання числових таблиць з метою обчислення траєкторії польоту

снарядів чи ракет до цілі. Комп'ютер оперував десятковими числами, і його програмування здійснювалося шляхом встановлення перемикачів та комутації роз'ємів. З появою комп'ютера ЕНІАК швидкість виконання операцій збільшилась, але на програмування алгоритму обчислень витрачалося багато часу. Ще однією проблемою були численні помилки, які виникали при програмуванні за допомогою встановлення перемикачів чи комутації роз'ємів.

Відомий математик і фізик Д. фон Нейман, консультант проекту ЕНІАК, запропонував записувати алгоритм обчислень у пам'ять разом з даними так, щоб вміст його можна було вільно змінювати аналогічно до даних відповідно до отриманих результатів. Цей принцип отримав назву *«принцип збереження програми»* і використовувався при проектуванні нового комп'ютера під назвою ЕДВАК (EDVAC) в 1945 р.

У комп'ютері ЕДВАК застосовувалася двійкова арифметика, в її основній пам'яті були використані елементи на ультразвукових ртутних лініях затримки. Операції виконувалися порозрядно, починаючи з молодшого розряду двійкового числа.

Історію обчислювальних машин (комп'ютерів) та їхні архітектурні особливості прийнято розглядати за поколіннями: перше, друге, третє і т. д. На сьогоднішній день можна вести мову про четверте покоління комп'ютерів і частково про п'яте.

Проект створення комп'ютерів п'ятого покоління запропонувала Японія (80-ті рр. ХХ ст.), над розробкою яких інтенсивно працюють вчені не тільки Японії, а й США і європейських країн.

Перше покоління комп'ютерів (1945 – 1954 рр.) – це час становлення архітектури комп'ютерів нейманівського типу. Комп'ютер першого покоління складався з трьох структурних елементів: центрального процесора (ЦП), пам'яті і пристрою введення-виведення. До складу ЦП входили арифметико-логічний пристрій (АЛП) і пристрій керування. Керування було покладено на ЦП, і для пересилання даних послідовно між пристроєм введення-виведення також здійснювалось пряме керування з боку ЦП. Комп'ютерна програма була складена вже не машинною мовою, а мовою асемблера. До передових комп'ютерів цього покоління можна зарахувати комп'ютер АТЛАС Манчестерського університету, в якому зроблено спробу використання однорівневої пам'яті, власне кажучи, концепції віртуальної пам'яті й індексних регістрів.

Друге покоління комп'ютерів (1955 – 1964 рр.) засвідчило про початок застосування транзисторів, замість вакуумних ламп. На зміну ЗП на електронно-променевих трубках і ультразвукових лініях затримки

прийшли ЗП на магнітних сердечниках і магнітних барабанах, що стали використовуватися як основна пам'ять. Значного поширення набули також індексні реєстри й апаратні засоби для виконання операцій з плаваючою крапкою. Стали застосовуватися також спеціальні команди для виклику підпрограм. Водночас розроблялися мови програмування високого рівня, такі як Алгол, Кобол і Фортран, що давали змогу складати програми без врахування типу комп'ютера. Крім цього, були розроблені і впроваджені спеціалізовані процесори введення-виведення, здатні керувати різними за швидкодією пристроями введення-виведення, в результаті чого звільнявся ЦП для виконання основної роботи, тобто для виконання суто операцій опрацювання інформації. В комп'ютерах першого покоління системне програмне забезпечення не застосовувалося або майже не застосовувалося, а в комп'ютерах другого покоління вже стали використовуватися компілятори, бібліотеки підпрограм, режим пакетної обробки і т. ін.

Третє покоління комп'ютерів (1965 – 1974 рр.) характеризується тим, що, замість транзисторів, вузли комп'ютерів стали будувати на інтегральних мікросхемах (інтегральних схемах), які за ступенем інтеграції поділяються на малі ІС, середні ІС (СІС) і великі ІС (ВІС), а також тим, що, замість пам'яті, на магнітних сердечниках стала використовуватися напівпровідникова пам'ять. Були розроблені методи ефективного використання таких ресурсів, як час ЦП, простір пам'яті і т. ін. Значного поширення набув метод віртуальної пам'яті, мета якого полягала в ефективному використанні простору пам'яті. Для підвищення ефективності використання ЦП при паралельному опрацюванні множини програм виникла необхідність у системній програмі, яка керує роботою ЦП. Так була створена операційна система (ОС). У третьому поколінні комп'ютерів поряд з мікропрограмуванням були розвинені і поширені різні архітектури пам'яті. Зокрема, значного поширення набули архітектура з диспетчеризацією пам'яті, метою якої є підвищення ефективності використання простору пам'яті, розшарування пам'яті, метою якого є збільшення фактичної швидкості доступу до пам'яті, та кеш-пам'ять. Започаткуванням комп'ютерів третього покоління стала розробка фірмою «ІВМ» систем машин ІВМ-360. Ця система суттєво вплинула на логічну організацію комп'ютерів третього покоління, розроблених в інших країнах. Так, Радянський Союз та інші країни-члени РЕВ на початку 70-х рр. ХХ ст. спільно розробили й організували серійне виробництво Єдиної системи ЕОМ (ЄС ЕОМ) і Системи малих ЕОМ (СМ ЕОМ) – машин третього покоління на інтегральних мікросхемах.

Четверте покоління комп'ютерів (кінець 70-х рр. – початок 80-рр. ХХ ст. і до сьогодні) характеризується тим, що конструктивно-технологічною основою є ВІС і надвеликі (НВІС) інтегральні мікросхеми, які вміщують тисячі, десятки тисяч, а то й мільйони транзисторів на одному кристалі. До четвертого покоління належать реалізовані на НВІС такі засоби обчислювальної техніки, як мікропроцесори, мікрокомп'ютери і побудований на їхній основі новий тип машин – персональні комп'ютери.

У комп'ютерах четвертого покоління досягається подальше спрощення контактів людини з комп'ютером шляхом підвищення рівня машинної мови, починається практична реалізація мовного введення і виведення інформації. Використання ВІС дає змогу апаратними засобами реалізовувати деякі функції програм операційних систем (апаратна реалізація трансляторів з мов високого рівня й ін.), що сприяє збільшенню продуктивності комп'ютерів.

Прикладами великих обчислювальних систем, які слід зарахувати до четвертого покоління, є багатопроцесорний комплекс «Ельбрус-2» і багатопроцесорна обчислювальна система ПС-2000, що вміщує до 64 процесорів.

В останні роки ХХ ст. століття і тепер визначилися контури й ведуться інтенсивні роботи зі створення нового, *п'ятого покоління комп'ютерів*. Значною мірою цьому сприяли публікації відомостей про проект («японський виклик») комп'ютера п'ятого покоління, що розробляється японськими фірмами і науковими організаціями, метою якого було завоювання японською промисловістю світового лідерства в галузі обчислювальної техніки. Згідно з цим проектом комп'ютери й обчислювальні системи п'ятого покоління, крім високої продуктивності і надійності, при нижчій вартості мають володіти такими якісно новими властивостями: можливістю взаємодії з комп'ютером за допомогою людської мови і графічних зображень; здатністю системи навчатися та самонавчатися, робити логічні судження, вести «розумне» спілкування з людиною у формі запитань і відповідей; здатністю системи «розуміти» вміст бази даних, яка при цьому перетворюється на базу знань, і використовувати ці «знання» при опрацюванні інформації (виконанні завдань).

1.3. Особливості архітектури найманівського комп'ютера

Великий внесок у розвиток обчислювальної техніки (побудови ЕОМ чи комп'ютерів) зробив відомий фізик і математик Джон фон Нейман. Він брав участь у розробці принципів програмування першого американського комп'ютера ЕНІАК, який був створений у 1945 р. Дж. фон Нейман

сформулював основні принципи побудови комп'ютерів, які надалі були названі його іменем.

Принципи фон Неймана:

1. Принципи програмного керування (управління) роботою комп'ютера. Програма складається з окремих кроків – команд. Послідовність команд є програмою.

2. Принцип умовного переходу. Можливість змінювати послідовність обчислення залежно від отриманих проміжних результатів.

3. Принцип зберігання програми. Команди подаються у числовій формі і зберігаються в тому самому ОЗП, що й дані для обчислень. Таким чином, команди можна подавати в арифметичний пристрій і перетворювати як звичайні числа. Це дає змогу створювати програми, здатні у процесі обчислень змінювати самі себе.

4. Використання двійкової системи числення для подання інформації.

5. Принцип ієрархічності запам'ятовувального пристрою. Дані, які найчастіше використовуються, зберігаються в найбільш швидкодіючому ЗП порівняно малої ємності, а дані, які використовуються дуже рідко, – в найбільшому повільному ЗП, але набагато більшої ємності.

Більшість комп'ютерів першого покоління будували на основі цих принципів, тому їх називають машинами найманівського типу. Важливо зазначити, що архітектура персональних комп'ютерів повністю базується на принципах фон Неймана, тому вони також є машинами найманівського типу. На даний час архітектура сучасних комп'ютерів недалеко відійшла від цих принципів. Як альтернативу цьому напрямку слід назвати комп'ютери, побудовані на основі напрямку розвитку трансп'ютерних мікропроцесорів, тобто комп'ютерів, побудованих на основі трансп'ютерів.

1.4. Основні типи комп'ютерів та сфери їхнього застосування

Великі успіхи в розвитку обчислювальної техніки, що базуються на досягненнях в галузі мікроелектротехніки, архітектури обчислювальних машин і програмного забезпечення, привели до значного розширення сфери застосування комп'ютерів.

Початково порівняно вузька сфера застосування комп'ютерів, переважно для наукових і технічних розрахунків, за короткий термін суттєво розширилась та охопила майже усі галузі науки, техніки, планування й управління технологічними процесами, всі сфери діяльності людини, пов'язані з обробкою великих обсягів інформації.

Різноманітність сфер і форм використання комп'ютерів зумовила виникнення спектра вимог до характеристик та особливостей організації машин і систем. У результаті на даний час відповідно до сфер застосування визначились основні типи комп'ютерів, які суттєво відрізняються не тільки за кількісними характеристиками, а й за архітектурою, електронно-технологічною базою і периферійними пристроями, які використовуються. Основні засоби сучасної обчислювальної техніки (сучасні комп'ютери) можна класифікувати таким чином: комп'ютерні системи (суперкомп'ютери), комп'ютери загального призначення, малі (міні-) комп'ютери, мікро- і персональні комп'ютери та мікропроцесори.

На рис. 1.2 типи комп'ютерів наведено відповідно до основних сфер застосування. Здійснений розподіл є доволі умовним через те, що межі між типами машин швидко змінюються під впливом досягнень у мікроелектроніці й архітектурі комп'ютерів.

Суперкомп'ютери

До надпродуктивних комп'ютерів (суперкомп'ютерів) на даний час належать комп'ютери («Cray», «Cray 1,2», «Cyber 205», «Эльбрус», «ПС-2000» і т. ін.), які при виконанні науково-технічних розрахунків досягають продуктивності від 10 до 10^3 Мфлоп/с (Мфлоп/с – мільйон операцій із плаваючою крапкою за секунду). Їхня колосальна продуктивність досягалася за рахунок унікальних структур апаратного і програмного забезпечення. Ці комп'ютери випускалися в незначних кількостях, як правило, під конкретне замовлення.

Подібні машини використовуються для розв'язування особливо складних задач науково-технічного характеру, задач опрацювання великих обсягів даних у реальному часі, пошуку оптимальних рішень у задачах економічного планування і при автоматизованому проектуванні.

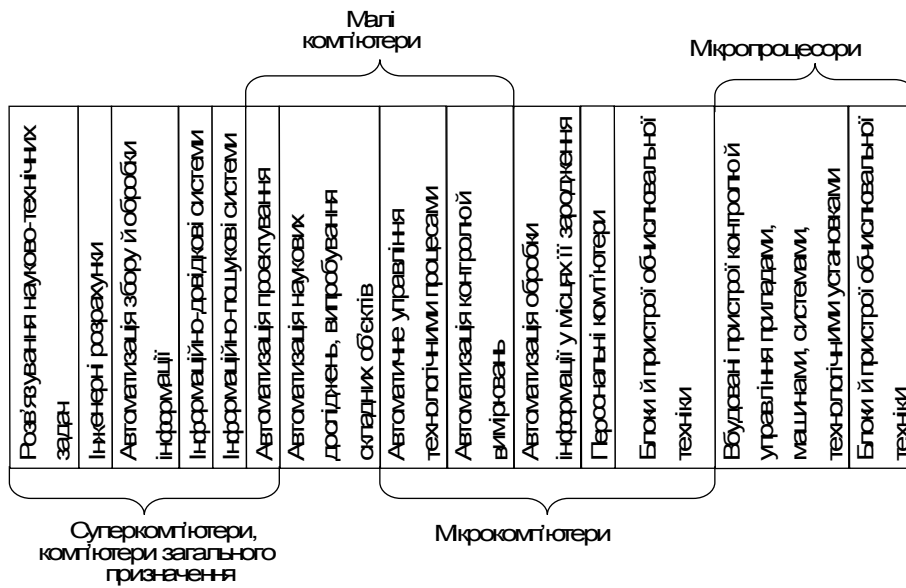


Рис. 1.2. Сфери застосування основних типів комп'ютерів

У табл. 1.1 подано класифікацію суперкомп'ютерів за типом архітектури.

Таблиця 1.1

Класифікація суперкомп'ютерів за типом архітектури

Архітектура	Приклади системи
Однопроцесорні системи з декількома функціональними пристроями і функціональними апаратними засобами вторинної обробки	«Alliant FX/1»; «IBM 3090»; «CDC 7600»; «FPS 164/264/364»; «Cray 1»; «Cray X-MP/1»; «Cyber 205»; «NEC SX-1»; «SX-2»; «SCS-40»
Процесорні матриці або приєднані процесори з архітектурою ОКМД	«Loral MPP»; «ICL/DAP»; «FPS 164/MAX»; «IBM-GF11»
Багатопроцесорні системи зі спільною пам'яттю	«Cray X-MP/2,4»; «Cray 2»; «Cray 3»; «IBM 3090/400»; «Univax 1184/ISP»
Багатомашинні системи з розподіленою пам'яттю	«Ipsc»; «Ametek 14»; «CDC Cyberplus»; «Cullen PSC»; «Warp»
Ієрархічні системи і системи, які реконфігуруються	«Cedar»; «ETA-10»; «IBM RP3»

Комп'ютери загального призначення

Комп'ютери вказаного типу належать до універсальних комп'ютерів (середні та великі комп'ютери) і відрізняються переважно показниками продуктивності. Сучасні комп'ютери загального призначення універсальні, вони можуть використовуватись як для розв'язування науково-технічних задач числовими методами, так і в режимі автономного опрацювання даних в АСУ. Такі машини мають високу швидкість, пам'ять великого обсягу, гнучку систему команд, широкий вибір периферійних пристроїв і способів кодування інформації, що враховує потреби опрацювання даних.

Для ефективного використання технічних засобів необхідна паралельна робота в часі процесора і периферійних пристроїв. Такий режим у машинах загального призначення організовується за допомогою спеціалізованих процесорів введення-виведення (каналів введення-виведення) інформації. Узагальнену структуру комп'ютера загального призначення вміщено на рис. 1.3.

На рисунку видно, що ряд функцій керування операціями введення-виведення виділено в окрему функціональну систему, реалізовану у вигляді сукупності каналів. Значне вивільнення процесора від керування введенням-виведенням є необхідною умовою для мультипрограмного режиму роботи (необхідною, але недостатньою). Потрібно забезпечити збалансованість пропускну здатності процесора, пам'яті і систем введення-виведення, причому в умовах, коли у процесорах нових моделей за рахунок ефективних структурних рішень здійснюється значне зростання швидкості та зменшується тривалість машинного такту. Це зумовлює необхідність підвищення пропускну здатності каналів і інтерфейсу введення-виведення та збільшення загального числа каналів.

Байт-мультиплексний канал може одночасно обслуговувати декілька порівняно повільно діючих периферійних пристроїв – друкувальних, введення інформації з перфострічок і перфокарт, виведення інформації на перфострічки й перфокарти, дисплеї і т. ін.

Селекторні і блок-мультиплексні канали зв'язують процесор і оперативну пам'ять з периферійними пристроями, що працюють з високою швидкістю обміну інформацією (магнітні, магнітооптичні й оптичні диски, магнітні стрічки тощо).

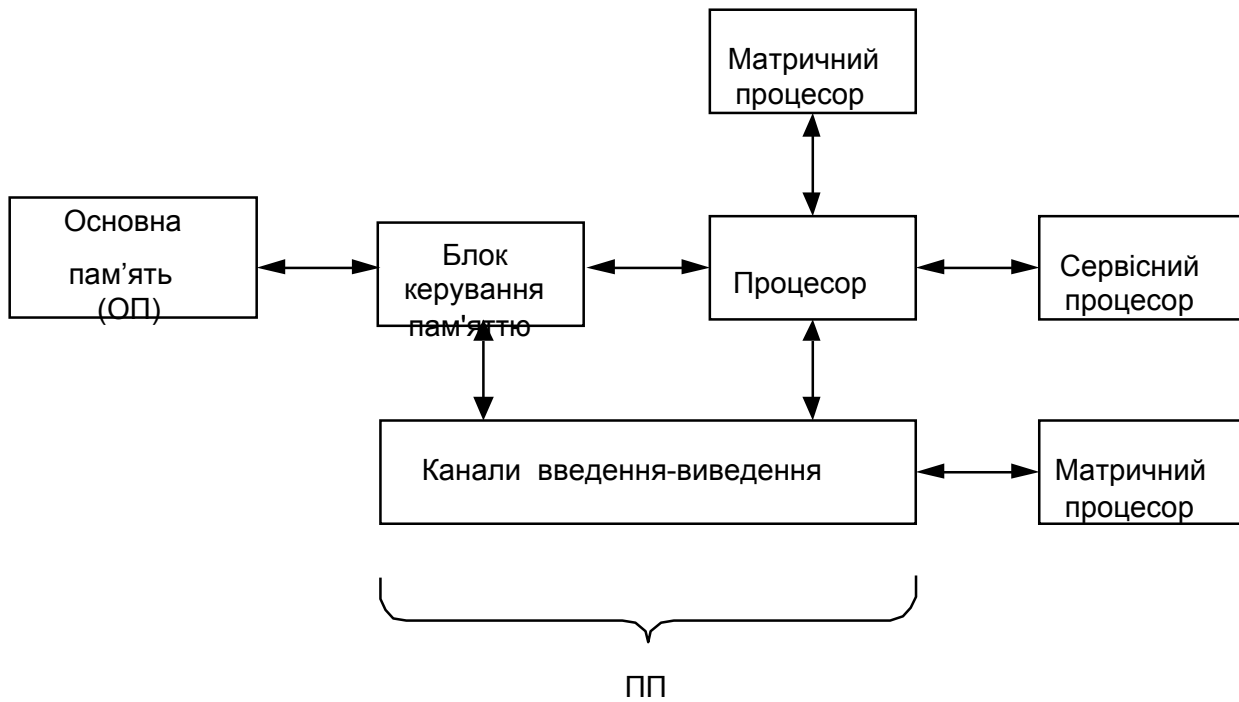


Рис. 1.3. Узагальнена структура комп'ютера загального призначення

Мікрокомп'ютери і мікропроцесори

Розвиток інтегральної технології привів до створення великих мікросхем (ВІС) різного призначення, що вміщують десятки й сотні тисяч простих елементів (транзисторів на кристалі). Процесор комп'ютера, реалізований на одній або декількох ВІС, отримав назву «мікропроцесор» (МП). МП за швидкодією і можливостями системи команд наближається до міні-комп'ютерів (малих комп'ютерів). Однак через обмежену кількість виводів корпусу ВІС (від 18 до 42) не вдається реалізувати інтерфейс МП із зовнішнім обладнанням з високими пропусковими можливостями.

Пристрій опрацювання даних, який має у своєму складі один або декілька МП, ВІС постійної й оперативної пам'яті, ВІС керування введенням-виведенням та ін., називається мікрокомп'ютером. Електронна апаратура мікрокомп'ютера може вміщувати декілька десятків ВІС, розміщених на одній або декількох друкованих платах. Мікрокомп'ютери можуть бути виконані й на одній ВІС (наприклад, однокристальні мікрокомп'ютери K1816BE48, K1816BE51 і т. ін.).

Здебільшого мікрокомп'ютери використовуються для створення проблемноорієнтованих і спеціалізованих обчислювальних систем. Один з напрямків розвитку мікрокомп'ютерів – це розробка персональних комп'ютерів. Другим важливим напрямком є використання МП та мікрокомп'ютерів як нової елементної бази автоматизованих систем,

машин, пристроїв. Мікропроцесори і мікрокомп'ютери відкривають принципово нові можливості для високоефективної автоматизації виробничих процесів, науково-дослідних та проектно-конструкторських робіт, опрацювання інформації при плануванні й управлінні виробництвом на підприємствах у всіх галузях народного господарства.

Міні-комп'ютери

Це широкий клас машин, які належать до стаціонарних. Порівняно з мікрокомп'ютерами, вони мають ширший набір команд, вищу точність обчислень, значний обсяг пам'яті. Як елементну базу сучасних міні-комп'ютерів також широко використовують ВІС. Характерною особливістю цього класу машин є розвинута система периферійних пристроїв.

Міні-комп'ютери переважно належать до проблемноорієнтованих обчислювальних машин. Основне їхнє призначення – це керування технологічним обладнанням, первинне опрацювання інформації, організація обміну по лініях зв'язку і т. ін. Швидкодія змінюється в широкому діапазоні від 10000 до декількох мільйонів операцій за секунду, залежно від функціонального призначення. Водночас міні-комп'ютери достатньо універсальні, щоб розв'язувати широке коло науково-технічних задач. На сьогодні можливості сучасних міні-машин, наприклад суперміні-комп'ютерів, перебивають можливості середніх і великих комп'ютерів.

1.5. Основні характеристики комп'ютерів

Важливими експлуатаційними характеристиками комп'ютера є його продуктивність P та загальний коефіцієнт ефективності E :

$$E = P / (C_k + C_{\text{екс}}), \quad (1.1)$$

який представляє собою відношення його продуктивності до суми вартості самого комп'ютера C_k , і затрати на його експлуатацію за певний період часу $C_{\text{екс}}$. Оскільки на практиці часто буває важко оцінити затрати на експлуатацію цієї моделі комп'ютера, то оцінюють ефективність машини за спрощеною формулою:

$$E^* = P / C_k. \quad (1.2)$$

Оцінювання й зіставлення продуктивності різних комп'ютерів є дуже складною проблемою, яка, власне, досі не отримала задовільного вирішення.

Продуктивність комп'ютера можна було би розуміти як кількість «обчислювальної роботи», або, інакше кажучи, кількість задач, які розв'язуються машиною за одиницю часу.

Однак на продуктивність комп'ютера, яка оцінюється за числом розв'язаних за одиницю часу задач, впливає дуже багато факторів, у тому числі й тип задач, стиль програмування та інші особливості програм, логічні можливості системи команд, особливості операційної системи, структура процедур, характеристики і організація оперативної й зовнішньої пам'яті, системи введення-виведення, склад та характер периферійних пристроїв.

У таких умовах оцінювання продуктивності за деяким набором задач здійснюється тільки тоді, коли обчислювальна установка (система) призначена для розв'язування доволі вузького кола задач.

У даний час продуктивність комп'ютерів загального призначення оцінюється спрощено за виконання деяких сумішей команд, що формується шляхом аналізу частоти виконання різних команд при виконанні програм розв'язування задач деякого доволі широкого класу, наприклад, науково-технічних задач чи задач обробки даних. На основі такого аналізу окремим видам команд (операцій) присвоюються певні вагові коефіцієнти. При використанні суміші команд продуктивність комп'ютерів визначається за формулою:

$$P = \frac{\sum_{s=1}^n k_s}{\sum_{s=1}^n k_s t_s}, \quad (1.3)$$

де k_s , t_s – відповідно вагові коефіцієнти і тривалість випадкових s -команд; n – число різних команд у суміші.

До основних характеристик комп'ютера належать:

- число розрядів у машинному слові (впливає на точність обчислень);
- швидкість виконання основних типів команд;
- обсяг оперативної пам'яті;
- максимальна швидкість передавання інформації між ядром комп'ютера (процесор і ОП) й зовнішніми периферійними пристроями.

Однак наведені вище характеристики є лише номінальними показниками. Вони реалізуються, якщо комп'ютер повністю працездатний, але існує ще один важливий показник – експлуатаційна надійність машини.

Надійність комп'ютера визначається частотою порушень його працездатності через відмови і збої й затратами часу на їхнє усунення.

Нехай K_n – комплексний коефіцієнт експлуатаційної надійності машини, який характеризує втрату продуктивності через порушення

надійного функціонування комп'ютера. Тоді формули (1.1) і (1.2) матимуть вигляд:

$$E = P K_n / (C_k + C_{екс}); \quad (1.4)$$

$$E^* = P K_n / C_k; \quad (1.5)$$

При створенні нових комп'ютерів має забезпечуватись значне зростання відношень продуктивність / вартість.

1.6. Вимірювання і визначення порівняльних характеристик продуктивності комп'ютерів

Для оцінювання продуктивності комп'ютерів, призначених для виконання наукових розрахунків, часто використовуються ядра, запропоновані Ліверморською лабораторією. Дж. Донгарра з Аргонської національної лабораторії провів порівняння характеристик продуктивності близько 100 різних типів комп'ютерів (від супер- до мікрокомп'ютерів), використовуючи для цього стандартну програму розв'язування систем лінійних рівнянь LINPACK у середовищі Фортрану. Оцінка продуктивності, отримана за допомогою контрольної задачі, може змінюватися при внесенні змін у програмні й апаратні засоби.

Продуктивність суперкомп'ютерів часто вимірюється у Мфлоп./с (мільйон операцій з плаваючою крапкою за секунду) або в млн. опер./с (мільйон операцій за секунду). Перша характеристика відображає можливості комп'ютерів щодо розв'язування числових задач великого обсягу, які часто пов'язані з опрацюванням 64- чи 32-розрядних операцій з плаваючою крапкою. Друга характеристика застосовується для оцінювання швидкодії комп'ютерів при виконанні як арифметичних, так і логічних команд та команд керування. Зв'язок між цими двома характеристиками продуктивності різний для машин різних класів і різних програм. Для типового суперкомп'ютера набір команд вміщує скалярні, векторні і керуючі команди, а швидкодія, що дорівнює 1 млн. опер./с, може відповідати продуктивності від 0,5 до 10 Мфлоп./с. Однак ці співвідношення ніяким чином не поширюються на всі комп'ютери.

Абсолютна верхня межа діапазону продуктивності визначається максимальною швидкодією машини.

У табл. 1.2 наведено характеристики продуктивності і вартості для комерційних суперкомп'ютерів трьох класів. Повні суперкомп'ютери утворюють клас дорогих моделей, до якого входять обчислювальні системи фірми «Gray», «ETA», «Fujitsu» й ін. До класу машин, які близькі за можливостями до суперкомп'ютерів, належать старші моделі

комп'ютерів, таких як IBM 3090/VF, Cyberplus фірми «CDC» й ін. Міні-суперкомп'ютери представляють собою недорогі паралельні чи векторні машини, прикладами яких можуть бути обчислювальні системи фірм «Alliant», «Convex» та ін.

Таблиця 1.2

Характеристика продуктивності і вартості суперкомп'ютерів

Клас	Максимальна продуктивність, Мфлоп./с, млн. опер./с	Вартість, млн. дол.	Приклад системи
Потужні суперкомп'ютери	200 – 2400;> 500	2 – 25	Gray 2, GrayX-MP, ETA-10, IBM GF-11
Близькі за можливостями до суперкомп'ютерів (старші моделі універсальних комп'ютерів)	50 – 500;> 50	1 – 4	IBM 3090/VF, Loral MPP, CDC Cyberplus
Міні-суперкомп'ютери	10 – 100;> 10	0,1 – 1,5	Alliant FX/8, SCS-40, iPSC, FPS 164 Max
Суперміні- (не належать до суперкомп'ютерів, характеристики наведено для порівняння)	0,5; 5	0,02 – 0,4	VAX 8600, IBM 4300, IBM 93700

Суперміні-комп'ютери не належать до класу суперкомп'ютерів. Відповідно до прийнятої класифікації машина входить до класу суперкомп'ютерів, якщо при виконанні науково-технічних розрахунків її продуктивність становить від 10 до 1000 Мфлоп./с.

При вимірюванні продуктивності суперкомп'ютерів дуже важливим є розуміння призначення конкретних задач. Ключова проблема полягає у використанні контрольних задач для отримання об'єктивних порівняльних характеристик продуктивності. Контрольні задачі дають змогу оцінити відносну продуктивність різних машин при однаковому робочому навантаженні.

За визначенням контрольна задача для комп'ютера представляє собою набір ключових програм або їхніх частин, які виконуються з метою хронометрування машинного часу.

Підмножини ключових задач, які називаються ядрами, часто є просто частинами великої програми, що потребують найбільших затрат машинного часу. Ядра представляють собою реальні і зручні в

користуванні програми, тому вони часто застосовуються як стандартні тестові програми.

Прикладом оцінювання продуктивності інших типів комп'ютерів може служити оцінювання продуктивності 16-розрядного мікропроцесора K1810BM86.

Оцінювання продуктивності 16-розрядного мікропроцесора K1810BM86

На етапі структурного аналізу при виборі елементної бази пристрою, який проектується, розробники стикаються з труднощами при оцінюванні продуктивності. Наприклад, при виборі мікропроцесора далеко не завжди можна оцінити його реальну обчислювальну потужність. Дані, які одержуються фірмами-виробниками компонентів, не дають повного уявлення про обчислювальну потужність.

Існуючі способи оцінювання обчислювальної потужності, які базуються на прогоні контрольних прикладів, потребують великих апаратних і часових затрат та характеризуються певним суб'єктивізмом. Найбільш доцільним способом оцінювання обчислювальної потужності мікропроцесора є статистичні методи, що набули значного поширення при оцінюванні продуктивності великих комп'ютерів.

Оцінювання обчислювальної потужності 16-розрядного мікропроцесора K1810BM86 базується на методиці визначення продуктивності комп'ютера щодо суміші команд ГБСОН-III. Оцінювання продуктивності комп'ютера доцільно проводити, враховуючи статистичні ваги відповідних операцій у типових програмах. При цьому вводиться поняття внутрішньої продуктивності комп'ютерів, яка задається виразом:

$$P_0 = \frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n a_i t_i + br}, \quad (1.6)$$

де t_i – час виконання i -ї команди; a_i – вага i -ї команди; b – середнє число одиниць інформації, які вводяться на n команд; r – час, затрачений процесором на організацію введення одиниці інформації.

Оскільки склад команд, порівнюваних комп'ютером, може відрізнятися доволі суттєво, доцільно звести їхній список до узагальнених команд, яких достатньо для реалізації більшості алгоритмів у колі задач, що розглядаються. Звичайно ці команди називають *сумішами команд*.

При визначенні продуктивності МП К1810ВМ86 використовується суміш (застосовується для комп'ютерів третього покоління) ГІБСОН III. Статичні ваги для цієї суміші, наведені у табл. 1.3, отримано при розв'язуванні великого числа науково-технічних задач. Для цього кола задач справджується допущення про мінімум процесора, затраченого на введення-виведення. Інакше кажучи, у формулі (1.6) можна припустити, що

$$br \ll \sum_{i=1}^n a_i b_i \quad (1.7)$$

і виключити br з розрахунків.

При складанні програм, які реалізують узагальнені команди (див. табл. 1.3), враховано, що слова науково-технічних задач мають довжину 32 біти (4 байти), тому вважалось, що всі операнди також дорівнювали 32 бітам.

Мікросхема К1810ВМ86 представляє собою однокристальний 16-розрядний мікропроцесор, який володіє властивостями як 8-, так і 16-розрядного процесора. Мікропроцесор здійснює операції над бітами, байтами, словами, а також виконує дії знакової і беззнакової, двійкової та десяткової арифметики, у тому числі множення і ділення.

Логічні й арифметичні операції мікропроцесор може виконувати над даними, які зберігаються не тільки в акумуляторі, а й в оперативній пам'яті або у внутрішніх регістрах. При написанні програм в усіх випадках, де це можливо і доцільно, для зберігання проміжних результатів використовувались внутрішні регістри процесора, що загалом скоротило час виконання команд. Крім чотирьох регістрів даних, які можна використовувати без обмежень у більшості арифметичних і логічних операцій, група регістрів, що охоплюють два вказівники (бази та стека) і два індексних регістри (джерела й приймача), також може брати участь у більшості арифметичних і логічних операцій.

Програми переміщення масиву даних (підпрограми 4 і 5 у табл. 1.3) застосовують стрічкові команди мікропроцесора. Процесорні інструкції 16-розрядного множення використовуються 32-розрядним множенням, а програми ділення написані за відомим алгоритмом ділення.

Як приклад розглянемо підпрограму 12 табл. 1.3.

Додавання двох чисел із заданих комірок із записом результату в пам'ять:

	Кількість робочих тактів
MOV BP, BASE1	4
MOV BX, BASE2	4
MOV AX, [BX]	10
ADD [BP], AX	14
MOV AX, 2[BX]	10
ADC [BP], AX	14
Загальна кількість тактів (0,2 мкс кожний)	56
Час виконання команди (мкс)	12,2

Аналогічні підрахунки були проведені щодо решти підпрограм, які реалізують загальні команди в табл. 1.3. В результаті отримано внутрішню продуктивність мікропроцесора K1810BM86:

$$P_b = \frac{\sum_{i=1}^{21} a_i}{\sum_{i=1}^{24} a_i t_i} \approx 15600 \text{ опер./с.}$$

Розроблений математичний процесор Intel-8087 призначено для виконання функцій сопроцесора (допоміжного процесора), безпосередньо пов'язаного з універсальним процесором K1810BM86. Математичний процесор Intel-8087 забезпечує в середньому 100-кратний вииграш щодо швидкодії при емуляції команд математичного процесора повністю стандартними підпрограмами на K1810BM86. Цей сопроцесор забезпечує виконання арифметичних та логічних операцій як з плаваючою, так і з фіксованою крапкою (з точністю 32 або 64 двійкових розряди), а також обчислення трансцендентних функцій.

Операції за підпрограмами 22, 23 і 24 визначають додатковий час для команд, що інтерпретуються, основними операціями в яких є звернення до оперативної пам'яті.

Математичний процесор підключається на системну шину K1810BM86 і працює з ним паралельно. В такій системі K1810BM86 та математичний процесор аналізують одну і ту саму послідовність інструкцій та виконують ті операції, для яких вони краще пристосовані.

Продуктивність пари мікропроцесорів K1810BM86 та Intel-8087 становить:

Таблиця 1.3

Інтерпретація суміші ГІБСОН-III і розрахунок внутрішньої
продуктивності
МП K1810BM86 та Intel-8087

№ підпрограми	Підпрограми	a _i	K1810BM86		Intel-8087	
			t _i (мкс)	a _i t _i	t _i (мкс)	a _i t _i
1.	Читання слова з комірки пам'яті у SM	7	6,2	43,4	–	–
2.	Те саме для чисел з плаваючою крапкою	7	6,2	43,4	–	–
3.	Передача вмісту суматора в задану комірку пам'яті	7	6,4	44,8	–	–
4.	Переміщення 500 слів послідовно адресованих комірок пам'яті в іншу область послідовно адресованих комірок	0,006	3400	20,4	–	–
5.	Переміщення 500 слів, які мають випадково розподілені задані адреси, в 500 послідовно адресованих комірок	0,004	6900	27,6	–	–
6.	Умовна передача керування в задану комірку (немає передачі керування)	6,5	0,8	5,2	–	–
7.	Умовна передача керування в задану комірку (є передача керування)	6,5	0,8	5,2	–	–
8.	Порівняння двох слів і встановлення ознаки, яка може бути перевірена	3	9,6	28,8	–	–
9.	Те саме для чисел з плаваючою крапкою	3	9,6	28,8	–	–
10.	Порівняння двох десяткових цифр для встановлення ознаки	1	4,6	4,6	–	–
11.	Безумовна передача керування в задану комірку	11,5	3	34,5	–	–
12.	Додавання двох чисел із заданих комірок і запис результату	7	11,2	78,4	–	–
13.	Те саме для операції «Відняти»	7	11,2	78,4	–	–
14.	Те саме для операції «Множити»	0,6	147	88,2	29,2	17,5
15.	Те саме для операції «Ділити»	0,2	1328	256,6	49,2	9,8
16.	Зсув вмісту регістра одинарної довжини або суматора вліво на 6 двійкових розрядів	4,6	29	133,4	–	–

Продовження таблиці 1.3

1	2	3	4	5	6	7
17.	Формування логічного І/АБО слова, розміщеного в суматорі або регістрі, та слова із заданої комірки пам'яті	1,7	8	13,6	–	–

18.	Додавання двох чисел з плаваючою крапкою	5,1	91,6	467,2	23	117,3
19.	Те саме для операції «Відняти»	5,1	91,6	3106	26	132,6
20.	Те саме для операції «Множити»	5,1	169,8	5856	25,6	130,6
21.	Те саме для операції «Ділити»	3,2	970,6	30,4	46	147,2
		92,11		5856		115 0,7
22-23.	Індексація	19	1,6	30,4		
24.	Непряма адресація	19	1	19		
				5905,4		1200,1

$$P_b = \frac{\sum_{i=1}^{21} a_i}{\sum_{i=1}^{24} a_i l_i} \approx 78000 \text{ опер./с.}$$

Для порівняння були взяті дані про внутрішню продуктивність комп'ютерів типу ЕС ЕОМ, які обчислено за сумішшю ГІБСОН-III. Порівняння показують, що за класом науково-технічних задач мікропроцесор К1810ВМ86 поступається ЕС 1030 за продуктивністю в два з половиною рази, а із сопроцесором Intel-8087 – переважає її в два рази.

2. СИСТЕМИ ЧИСЛЕННЯ

2.1. Системи числення та їхня класифікація

Систему числення розуміють як спосіб подання будь-кого числа за допомогою деякого алфавіту символів, що називаються цифрами. Від їхніх особливостей залежать наочність подання числа за допомогою цифр і складність використання арифметичних операцій. Вибір системи числення для комп'ютера впливає не тільки на алгоритми виконання операцій, а й на структуру вузлів, блоків і пристроїв комп'ютера. Історично склалося так, що найбільшого поширення в практиці обчислювальних робіт набула позиційна десяткова система числення. Однак ця система числення не є найбільш зручною для її реалізації в комп'ютері, де, як правило, використовують системи числення з не десятковою основою – двійкову чи двійково-кодовані (тобто такі, цифри яких закодовані двійковими символами).

Всі системи числення поділяються на **позиційні і непозиційні**

Непозиційною називається така система числення, в якій значення цифри не залежить від її положення (позиції) в ряді цифр, що зображають число.

Прикладами є римська система числення, в якій для позначення окремих чисел використовуються букви римського алфавіту, й одинична (унітарна) системи.

Цифри у римській системі числення позначаються різними символами: 1 – I; 3 – III; 5 – V; 10 – X; 50 – L; 100 – C; 500 – D; 1000 – M. Запис числа здійснюється за таким правилом: кожний менший знак, який поставлено справа від більшого, додається до його значення, а зліва – віднімається від нього: так, XC – 90, а CX – 110. Виконувати арифметичні дії в непозиційних системах дуже незручно, тому в наш час ці системи не використовуються для розрахунків.

Позиційною називається така система числення, в якій значення цифри залежить від її положення (позиції) в ряді цифр, що зображають дане число, тобто від ваги.

Позиційна система числення характеризується кількістю цифр, які використовуються для запису чисел. Максимальна кількість цифр q , які використовуються для запису чисел у даній позиційній системі числення, називається *основою системи* числення. Ці цифри означають q цілих чисел $0, 1, \dots, (q - 1)$. Наприклад, у десятковій системі використовується 10 цифр: $0, 1, 2, \dots, 9$, тобто в даній системі $q = 10$.

У загальному випадку в позиційній системі числення з основою q будь-яке число X можна подати у такому вигляді:

$$X_q = a_n q^n + a_{n-1} q^{n-1} + \dots + a_i q^i + \dots + a_1 q^1 + a_0 q^0 + a_{-1} q^{-1} + \dots + a_{-m} q^{-m}, \quad (2.1)$$

де a_i – коефіцієнти, для позначення яких застосовуються будь-які з q цифр, що використовуються в системі числення.

На основі формули (2.1), яку часто називають «розгорнутою формою», числа можна зображати у вигляді такої послідовності цифр:

$$X = a_n a_{n-1} \dots a_1 a_0, a_{-1} a_{-2} \dots a_{-m}. \quad (2.2)$$

У виразі (2.2) кома (або крапка, як це прийнято в обчислювальній техніці) відділяє цілу частину числа (коефіцієнти при додатних степенях, включаючи нуль) від дробової (коефіцієнти при від'ємних степенях). Кома (крапка) відсутня, якщо немає від'ємних степенів. Позиції цифр, які відліковуються від крапки (коми), називаються розрядами. В позиційній системі числення значення кожного розряду більше від значення сусіднього справа розряду в q разів.

Звичною для людини є десяткова система числення (DECIMAL NOTATION), але у комп'ютерах застосовуються позиційні системи числення з основою, відмінною від 10: шістнадцяткова, вісімкова, двійкова й ін.

У шістнадцятковій системі числення $q = 16$ використовуються: 10 цифр (0, 1, 2, ..., 9) – для зображення молодших 10 цифр; шість символів A, B, C, D, E, F – для зображення відповідно старших шести цифр 10, 11, 12, 13, 14, 15. Зобразимо, наприклад, за допомогою виразу (2.2) шістнадцяткове число

$$\mathbf{B2E.4}_{16}.$$

Дане шістнадцяткове число згідно з формулою (2.1) запишеться таким чином:

$$\mathbf{B2E.4}_{16} = 11 \times 16^2 + 2 \times 16^1 + 14 \times 16^0 + 4 \times 16^{-1}.$$

У вісімковій системі числення $q = 8$, тобто застосовуються цифри 0, 1, 2, ..., 7. Зобразимо, наприклад, за допомогою виразу (2.2) вісімкове число 703.04_8 , яке згідно з формулою (2.1) запишеться таким чином:

$$\mathbf{703.04}_8 = 7 \times 8^2 + 0 \times 8^1 + 3 \times 8^0 + 0 \times 8^{-1} + 4 \times 8^{-2}.$$

Двійкова система числення (BINARY NOTATION) знайшла найширше застосування у комп'ютерах. В даній системі $q = 2$, тобто використовуються дві цифри 0 і 1.

Двійкова система числення знайшла найширше застосування у комп'ютерах завдяки таким перевагам:

– зручність подання у комп'ютерах довільного розряду двійкового числа. При цьому може використовуватися будь-який простий елемент, що має всього два стійких стани 1 і 0;

– простота двійкової арифметики;

– спрощений процес синтезу схем комп'ютерів, оскільки позначення змінних і функцій у математичному апараті алгебри логіки збігається з двійковими цифрами.

Недоліком цієї системи є громіздкість запису двійкового числа.

2.2. Переведення чисел з однієї системи числення в іншу

Для переведення чисел з однієї системи числення в іншу використовують певні правила (способи чи варіанти). Розрізняють правила переведення цілих, дробових і змішаних чисел.

Розподілимо всі варіанти переведення чисел на три типових випадки:

1) переведення чисел з довільної системи числення в десяткову;

2) переведення чисел з десяткової в довільну систему числення;

3) переведення чисел з двійкової у вісімкову (шістнадцяткову) систему числення, і навпаки.

Переведення чисел у першому випадку здійснюється за допомогою формули (2.1).

Наприклад, використавши вищенаведені приклади, отримаємо:

$$10101101.101_2 = \dots = 173.625_{10};$$

$$703.04_8 = \dots = 451.0625_{10};$$

$$B2E.4_{16} = \dots = 2862.25_{10}.$$

Переведення чисел з десяткової системи числення в довільну можна розподілити на два варіанти:

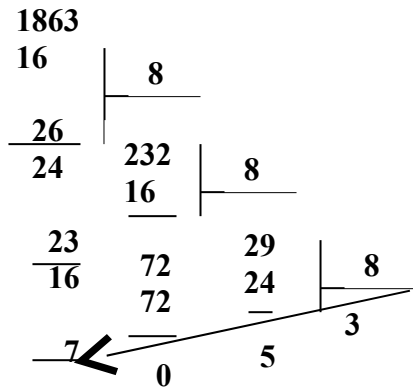
– переведення цілих чисел, яке здійснюється способом ділення;

– переведення дробових чисел, яке здійснюється способом множення.

Для переведення цілого числа з десяткової системи числення у будь-яку іншу необхідно розділити десяткове число на основу нової системи числення. Ділення проводиться доти, доки остача не буде меншою від основи нової системи числення.

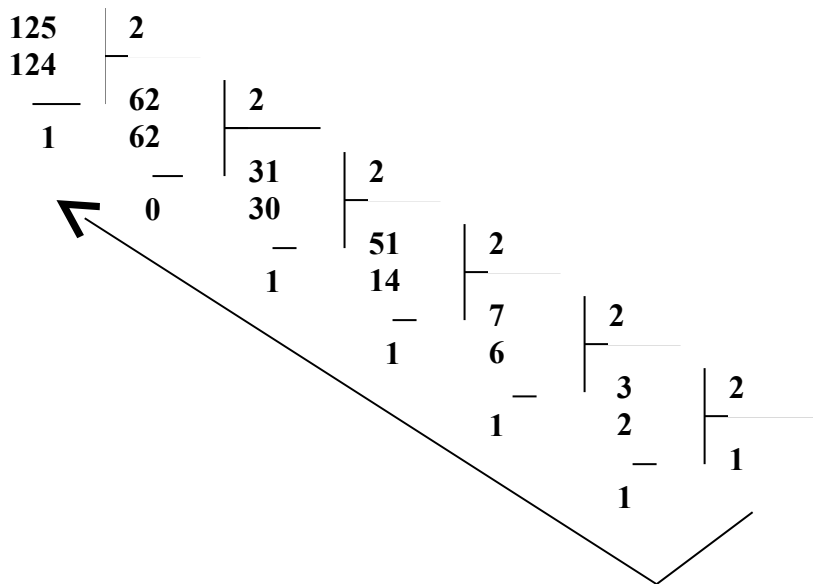
Наприклад:

$$1863_{10} \rightarrow X_8$$



Таким чином, $1863_{10} = 3507_8$.

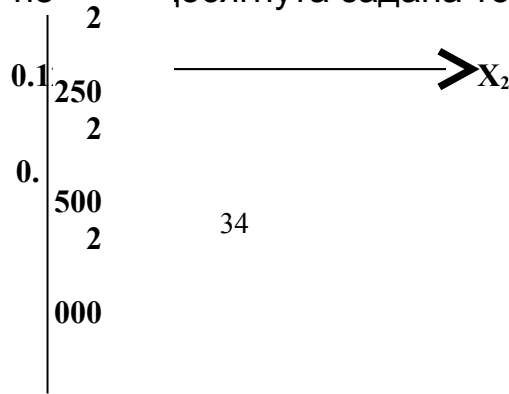
$$125_{10} \rightarrow X_2$$



Отже, $125_{10} = 1111101_2$.

Для переведення правильного десяткового дробу в іншу систему числення необхідно дробову частину десяткового числа послідовно множити на основу нової системи числення доти, доки у дробовій частині не залишаться нулі або не досягнута задана точність переведення.

Наприклад:



$$\begin{array}{r} \times \\ \hline 0 \\ \hline 0 \\ \hline 1 \end{array}$$

Таким чином, $0.125_{10} = 0.001_2$.

$$\begin{array}{r} 0.644_{10} \longrightarrow X_8 \\ \begin{array}{r} 0. \overline{644} \\ \times \quad 8 \\ \hline 5 \quad \overline{152} \\ \quad \quad 8 \\ \hline 1 \quad \overline{216} \\ \hline \cdot \\ \cdot \\ \cdot \\ \cdot \end{array} \longrightarrow \text{Похибка перетворення} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 0.644_{10} \approx 0.51_8 \end{array}$$

Неправильні дроби десяткової системи числення у будь-яку іншу систему переводяться у два прийоми: ціла частина переводиться за одним правилом, дробова – за іншим. Потім цілу і дробову частини числа записують разом, розділяючи їх крапкою.

Переведення чисел у десяткову систему числення з будь-якої іншої системи числення найзручніше здійснювати, подаючи ці числа в розгорнутій формі:

$$1101001_2 = 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 64 + 32 + 0 + 8 + 0 + 0 + 1 = 105_{10}.$$

Правила переведення вісімкових і шістнадцяткових чисел у двійкові, і навпаки дуже прості, оскільки основами вісімкової і шістнадцяткової систем числення є цілі степеневі числа ($8 = 2^3$ і $16 = 2^4$). Для переведення вісімкового (шістнадцяткового) числа у двійкову форму достатньо замінити кожну цифру цього числа трирозрядним (тріада) чи чотирирозрядним (тетрада) двійковим числом відповідно до табл. 2.1.

При переведенні числа з двійкової у вісімкову (шістнадцяткову) систему числення діють таким чином: рухаючись вправо і вліво від крапки, розбивають двійкове число на групи по три (чотири) розряди (тріади чи тетради), доповнюючи за необхідності нулями крайню ліву і крайню праву неповні групи. Потім кожну групу з трьох (чотирьох) розрядів заміняють відповідно вісімковою (шістнадцятковою) цифрою.

Вісімкова і шістнадцяткова системи числення безпосередньо не використовуються у комп'ютерах при реалізації операцій, а служать лише для полегшення запису комп'ютерних кодів та широко застосовуються у техніці програмування.

Очевидно, що найбільш зручною для людини є десяткова система числення, а для комп'ютерів – двійкова.

Таблиця 2.1

Відповідність цифр і символів у різних системах числення

X_{10}	X_2	X_8	X_{16}
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

$$2461.7_8 = 010\ 100\ 110\ 001.\ 111 = 10100110001.111_2$$

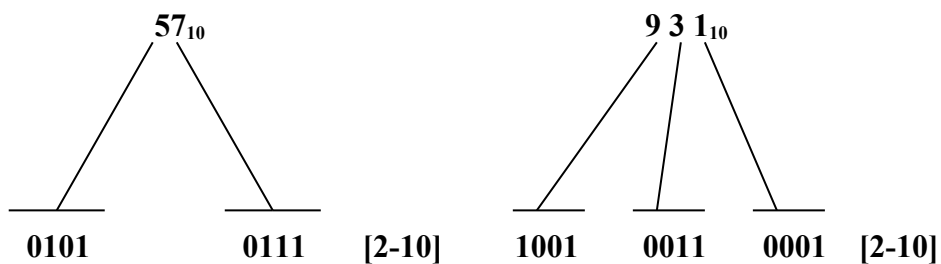
$$\begin{array}{cccccc} \text{—} & \text{—} & \text{—} & \text{—} & \text{—} & \\ 2 & 4 & 6 & 1 & 7 & \end{array}$$

$$8B4.E_{16} = 1000\ 1011\ 0100.\ 1110 = 100010110100.111_2$$

$$\begin{array}{cccc} \text{—} & \text{—} & \text{—} & \text{—} \\ 8 & B & 4 & E \end{array}$$

Розглянуті методи переведення чисел з однієї системи числення в іншу при їхній реалізації потребують суттєвих затрат (часових і апаратних). Для подолання цього користуються простим та оригінальним прийомом: попередньо у двійкову систему числення переводять не все число, а тільки його цифри – в результаті отримується деяка змішана *двійково-десяткова система*.

Двійково-десяткове число одержується таким чином: кожна цифра десяткового числа замінюється чотирирозрядним двійковим числом – тетрадою.



При зворотному перетворенні необхідно кожен тетраду замінити еквівалентною їй десятковою цифрою.

2.3. Арифметичні дії над числами в різних системах числення

Всі арифметичні операції над числами в різних системах числення проводяться за правилами арифметики. Як і в десятковій системі числення, користуються таблицею множення, а додавання виконують, використовуючи правила переносу в старший розряд. Наприклад, для вісімкової системи числення $7 + 1 = 10_8$.

Для двійкових і вісімкових систем числення наведено таблиці множення.

Таблиця 2.2

Таблиця множення для двійкових чисел

$0 \times 0 = 0$
$0 \times 1 = 0$
$1 \times 0 = 0$
$1 \times 1 = 1$

Таблиця 2.3

Таблиця множення для вісімкових чисел

×	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	27
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61

Приклад додавання двох двійкових чисел:

$$\begin{array}{r}
 1011011 \\
 + \quad 10111 \\
 \hline
 1110010
 \end{array}$$

Приклад віднімання двох двійкових чисел:

$$\begin{array}{r}
 1100001 \\
 - \quad 10 \\
 \hline
 1011111
 \end{array}$$

Приклади додавання двох вісімкових і шістнадцяткових чисел:

$$\begin{array}{r}
 157 \\
 + \quad 15 \\
 \hline
 174
 \end{array}$$

$$\begin{array}{r}
 A3B8.FE \\
 + \quad 242.A1 \\
 \hline
 A5FB.9F
 \end{array}$$

Приклади множення двох двійкових, вісімкових і шістнадцяткових чисел:

$$\begin{array}{r}
 10111.01 \\
 \times \quad 1011.1 \\
 \hline
 \end{array}$$

$$\begin{array}{r}
 15 \\
 \times \quad 12 \\
 \hline
 \end{array}$$

$$\begin{array}{r}
 AB1 \\
 \times \quad 1A \\
 \hline
 \end{array}$$

	32	6AEA
+ 1011101	+ 15	+ AB1
1011101	<hr style="width: 50%; margin: 0 auto;"/>	<hr style="width: 50%; margin: 0 auto;"/>
0000000	202	115FA
1011101		
<hr style="width: 50%; margin: 0 auto;"/>		
100001011.011		

Як видно з наведених прикладів, операція множення зводиться до операцій зсуву та додавання. Місце крапки (коми) визначається так само, як і при множенні десяткових чисел.

Двійково-десяткові системи числення широко використовуються у комп'ютерах як для подання чисел, так і для виконання арифметичних дій над ними. При цьому методика виконання арифметичних дій над двійково-десятковими числами, яка повністю базується на правилах двійкової арифметики, має свою специфіку. Необхідність цієї специфіки стає очевидною при спробі здійснити сумування двійково-десяткових чисел, обмежуючись правилами звичайного сумування двійкових кодів.

3. АЛГОРИТМИ ОБРОБКИ ІНФОРМАЦІЇ ТА ЇХНІ АРХІТЕКТУРНІ РІШЕННЯ

3.1. Загальні відомості про операції та їхній розподіл на групи

Всі арифметичні і логічні операції у комп'ютерах здійснюються в арифметико-логічних пристроях (АЛП), тобто арифметико-логічні пристрої використовуються для виконання арифметичних і логічних перетворень над словами, що називаються в даному разі операндами.

Операції, які виконуються в АЛП, можна поділити на такі групи:

- операції двійкової арифметики для чисел з фіксованою крапкою;
- операції двійкової (чи шістнадцяткової) арифметики для чисел з плаваючою крапкою;
- операції десяткової арифметики;
- операції індексної арифметики (при модифікації адрес команд);
- операції спеціальної арифметики;
- операції над логічними кодами (логічні операції);
- операції над алфавітно-цифровими полями.

Сучасні комп'ютери загального призначення, як правило, реалізують операції всіх наведених груп, а малі і мікрокомп'ютери, мікропроцесори та спеціалізовані комп'ютери часто не мають у своєму складі апаратури арифметики чисел плаваючою крапкою (за винятком старших моделей МП), десяткової арифметики й операцій над алфавітно-цифровими полями. В даному разі ці операції виконуються з використанням спеціальних мікропрограм.

До арифметичних операцій належать додавання, віднімання, віднімання модулів («короткі операції») і множення та ділення («довгі операції»). Групу логічних операцій складають такі операції: диз'юнкція (операція АБО) і кон'юнкція (логічне І) над багаторозрядними двійковими словами, а також порівняння кодів на рівність. До спеціальних арифметичних операцій входять нормалізація, арифметичний зсув (зсуваються тільки цифрові розряди, знаковий розряд залишається на місці) та логічний зсув (знаковий розряд зсувається разом з цифровими розрядами).

Можна навести таку класифікацію АЛП. За способом дії над операндами АЛП поділяються на послідовні і паралельні. В послідовних АЛП операнди подаються у послідовному коді, а операції проводяться послідовно в часі над їхніми окремими розрядами. У паралельних АЛП операнди подаються паралельним кодом і операції здійснюються паралельно в часі над всіма розрядами операндів.

За способом подання чисел розрізняють АЛП:

- для чисел з фіксованою крапкою;
- для чисел з плаваючою крапкою;
- для десяткових чисел.

За характером використання елементів і вузлів АЛП поділяються на блочні та багатофункціональні. У блочному АЛП операції над числами з фіксованою і плаваючою крапкою, десятковими числами й алфавітно-цифровими полями виконуються в окремих блоках, при цьому підвищується швидкість роботи, оскільки блоки можуть паралельно виконувати відповідні операції, але значно зростають затрати на обладнання. У багатофункціональних АЛП операції для всіх форм подання чисел виконуються одними і тими самими схемами, які комутуються відповідним чином залежно від потрібного режиму роботи.

3.2. Алгоритми виконання арифметичних операцій над двійковими числами та їхні архітектурні рішення

Як правило, в АЛП операція алгебраїчного додавання зводиться до арифметичного додавання кодів чисел шляхом застосування інверсних кодів – оберненого чи доповняльного для подання від'ємних чисел.

Алгоритми виконання в АЛП арифметичних операцій залежать від того, в якому вигляді зберігаються в пам'яті комп'ютера від'ємні числа – в прямому чи доповняльному (обернений код має два подання нуля (+0; -0), що ускладнює аналіз результату операції). В останньому випадку скорочується виконання операції за рахунок виключення операції перетворення отриманого в АЛП доповняльного коду від'ємного результату на прямий код, при цьому дещо ускладнюється операція множення.

Алгоритм алгебраїчного додавання двійкових чисел з фіксованою крапкою при використанні доповняльного коду для подання від'ємних чисел

При виконанні операції додавання додатні доданки подаються у прямому коді, а від'ємні – в доповняльному. Здійснюється додавання двійкових кодів, включаючи розряди знаків. Якщо при цьому виникає перенос зі знакового розряду суми за відсутності переносу в цей розряд чи перенос у знаковий розряд за відсутності переносу з розряду знака, то відбувається переповнення розрядної сітки відповідно при від'ємній і додатній сумах. Якщо немає переносів зі знакового розряду та у знаковий розряд суми чи є обидва переноси, то переповнення немає і при 0 у знаковому розряді сума додатна й подана у прямому коді, а при 1 у знаковому розряді сума від'ємна і подана в доповняльному коді.

На рис. 3.1 наведено спрощену структурну схему АЛП для операцій додавання і віднімання n -розрядних (нульовий розряд знаковий) двійко-

вих чисел з фіксованою крапкою. Припускається, що від'ємні числа зберігаються у пам'яті в доповняльному коді.

До складу АЛП входять n -розрядний паралельний комбінаційний суматор $См$, регістр суматора $РгСм$, вхідні регістри суматора $РгВ$ і $РгА$ вхідний регістр АЛП $Рг1$.

З оперативної пам'яті по вхідній інформаційній шині $ШІ Вх$ в АЛП надходять операнди: додатні числа у прямому коді, а від'ємні – в доповняльному. Операнди розміщуються в $РгВ$ (перший доданок чи зменшуване) і $Рг1$ (другий доданок або від'ємник); $Рг1$ зв'язаний з $РгА$ прямою й інверсною передачею коду.

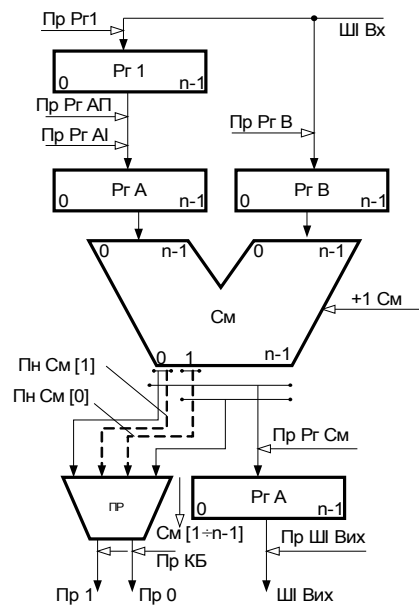


Рис. 3.1. Структурна схема АЛП для додавання і віднімання двійкових чисел з фіксованою крапкою

$Рг1$ зв'язаний з $РгА$ прямою й інверсною передачею коду. Пряма передача використовується при операції алгебраїчного сумування, а інверсна – при відніманні.

Результат операції видається з АЛП в оперативну пам'ять по вихідній операційній шині $ШІ Вих$.

При виконанні операції в АЛП, крім результату операції, формується 2-розрядний код ознаки результату $ПР$, який набуває значень, наведених у поданій нижче таблиці.

Результат операції	Ознака результату	
0	0	0
<0	0	1

>0	1	0
Переповнення	1	1

Припустимо, що код ознаки результату формується комбінаційною схемою ПР, на входи якої надходять сигнали, що відповідають значенням всіх розрядів суматора, а також сигнали переносів зі знакового розряду ПнСм[0] і у знаковий – зі старшого цифрового розряду ПнСм.

Умови вироблення ознак додатного і від’ємного результатів мають відповідно такий вигляд:

$$\overline{C_m[0]}(\overline{P_nC_m[0]}P_nC_m[1] \vee \overline{P_nC_m[0]}\overline{P_nC_m[1]});$$

$$C_m[0](\overline{P_nC_m[0]}P_nC_m[1] \vee \overline{P_nC_m[0]}\overline{P_nC_m[1]});$$

При виконанні алгебраїчного додавання коди операндів, які надійшли в АЛП, знаходяться у вхідних регістрах PгВ і PгА суматора. Код суми формується на виходах схеми См та фіксується в регістрі PгСм.

Операція алгебраїчного віднімання

$$Z=X-Y=X+(-Y) \quad (3.1)$$

може бути зведена до зміни знаку від’ємника Y й операції алгебраїчного додавання. Зміні знака відповідає така процедура (див. прямий, обернений і доповняльний коди): взятий в Pг1 код числа передається в PгА інверсно та при додаванні здійснюється додавання (підсумування) 1 у молодший розряд суматора.

Передавання інформації в регістрах АЛП проводиться окремими мікроопераціями, показаними на рис. 2.1 керуючими сигналами. Слово з Pг1 в PгА може бути передане у прямому (керуючий сигнал ПрPгАП) чи інверсному (керуючий сигнал ПрPгАІ) кодах.

Мікрооперація ПрКБ полягає у видачі в керуючий блок коду ознаки і у формуванні запиту переривання при переповненні розрядної сітки.

Множення чисел з фіксованою крапкою

Щодо двійкової системи числення найбільш відомі такі основні способи виконання операцій множення:

1. Множення, починаючи з молодших розрядів множника:

а) «ручний» метод:

$$\begin{array}{r}
 1010 \quad \text{– множене} \\
 \times 1001 \quad \text{– множник} \\
 \hline
 1010 \\
 + 0000 \\
 0000 \quad \Rightarrow \quad \text{часткові добутки} \\
 \hline
 1010 \\
 \hline
 1011010 \quad \text{– добуток}
 \end{array}$$

б) «машинний» метод:

$$1001 \quad \text{– множене}$$

$$\begin{array}{r}
 \times \underline{0101} \quad - \text{множник} \\
 1001 \\
 + 01001 \\
 \underline{101101} \\
 0101101 \quad - \text{добуток}
 \end{array}$$

2. Множення, починаючи зі старших розрядів множника:

а) «ручний» метод:

$$\begin{array}{r}
 1001 \quad - \text{множене} \\
 \times \underline{0101} \quad - \text{множник} \\
 0000 \\
 + 1001 \\
 \quad 0000 \\
 \quad \underline{1001} \\
 0101101 \quad - \text{добуток}
 \end{array}$$

б) «машинний» метод:

$$\begin{array}{r}
 1001 \quad - \text{множене} \\
 \times \underline{0101} \quad - \text{множник} \\
 0000 \\
 + 01001 \\
 \underline{010010} \\
 0101101 \quad - \text{добуток}
 \end{array}$$

При «ручному» методі в обох випадках (множення, починаючи з молодших і старших розрядів) множення зводиться до послідовного порозрядного множення множника на цифри множеного і сумування отриманих часткових сум. При цьому операціями множення можуть керувати розряди множника: якщо в i -му розряді множника міститься одиниця, то до суми часткових добутоків додається множене з відповідним зсувом на $i-1$ розряд (вправо або вліво залежно від прийнятого способу виконання операції множення; якщо в i -му розряді множника нуль, то множене не додається).

При «машинному» методі добуток формується у вигляді зростаючого (за модулем) значення часткової суми, яка дорівнює після множення на i -й розряд множника сумі перших i часткових добутоків.

Розглянуті приклади показують, що для отримання суми, крім додавання, необхідно виконувати операції зсуву чисел (множеного або часткової суми). Очевидно, що добутком двох n -розрядних чисел є число $2n$ -розрядне. Тому в разі обмеження поля цифр добутку n розрядами при множенні цілих чисел як результат беруться молодші n розрядів (у старших n розрядах мають бути нулі, тому що в іншому разі виробляється ознака переповнення), а при множенні правильних дробів як результат беруться старші n розрядів (молодші розряди відкидаються або використовуються для заокруглення коду старших n розрядів).

Схеми множення

Операції множення виконуються з використанням послідовного сумування і зсуву. Враховуючи, що у переважній більшості випадків точність отриманого добутку має відповідати точності співмножників, здебільшого будемо розглядати схеми множення з кількістю розрядів у добутку, яка дорівнює кількості розрядів множеного і множника. Розглядати будемо множення у прямих кодах.

Множення, починаючи з молодших розрядів

Схема множення з молодших розрядів множника набула найбільшого поширення в арифметичних пристроях комп'ютерів універсального типу. Для виконання операції множення множене завантажується в Rг1, а множник – в Rг2 (рис. 3.2).

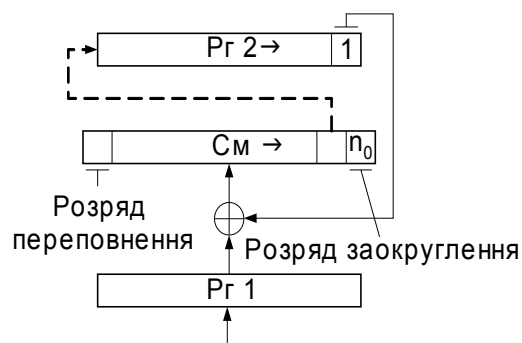


Рис. 3.2. Схема множення з молодших розрядів множення

Процес множення полягає у передачі з регістру Rг1 множеного в суматор у тому разі, коли в молодшому розряді множника міститься код одиниці з подальшим зсувом часткових добутків у суматорі. Одночасно відбувається зсув коду множника в Rг2. Процес множення триває доти, доки не буде виконано множення на всі розряди множника. Для отримання правильного результату (добутку) в суматорі необхідно передбачити розряд переповнення, в який може переходити одиниця, що з'являється при отриманні часткових добутків. При виконанні операції множення з числами, які менші від одиниці, положення крапки в добутку визначається автоматично, після виконання зсувів, число яких відповідає кількості розрядів у коді множника.

Для того, щоб похибка добутку не перевищувала $2^{-(n+1)}$, у більшості випадків в універсальних комп'ютерах передбачається заокруглення отриманого добутку (результату). Для цього в суматорі передбачено один розряд у правій частині розрядної сітки. При виконанні операції множення, починаючи з молодших розрядів множника, є можливість отримання $2n$ -значного добутку. Для цього необхідно передбачити зв'язок передачі

молодших розрядів часткових добутоків у старші розряди Rг2, які звільняються при зсуві коду множника. Результат виконання операції множення утримується в двох місцях: старші розряди добутку в суматорі, а молодші – в Rг2.

Множення, починаючи зі старших розрядів

Множення зі старших розрядів знаходить широке застосування в арифметичних пристроях спеціалізованих комп'ютерів. Це зумовлено тим, що використання цього методу (типу) множення дає змогу підвищити швидкість виконання операції і спростити логічну схему суматора.

Для виконання операції множення зі старших розрядів множене завантажується в Rг1, а множник – в Rг2 (рис. 3.3).

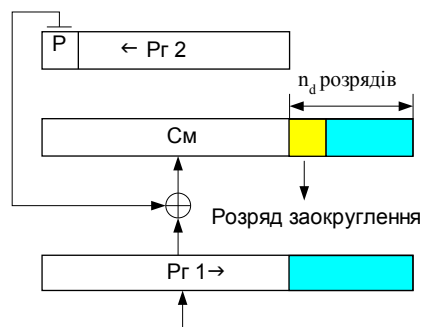


Рис. 3.3. Схема множення зі старших розрядів множника

Процес множення полягає у передачі на суматор множеного, якщо у старшому розряді множника є одиниця. Після подачі множеного проводиться зсув множеного вправо, а множника – вліво. Далі повторюється передача множеного у суматор за наявності одиниці у старшому розряді множника і т. ін. Цикл додавання та зсуву повторюється стільки разів, скільки розрядів є у множника. Добуток утримується у суматорі. Для реалізації цього процесу необхідно, щоб суматор і регістр множеного Rг1 мали подвоєну кількість розрядів.

3.3. Алгоритми і пристрої для виконання логічних операцій

До складу операцій, які реалізуються комп'ютером, зараховують такі порозрядні логічні операції: сумування за модулем 2, логічне множення І та логічне додавання АБО.

У результаті порозрядної логічної операції над словами X і Y формується слово Z, в якому

$$Z[i] = X[i] * Y[i], \tag{3.2}$$

де * – символ операції, яка виконується.

Для виконання логічних операцій можна скористатися пристроєм, структурну схему якого наведено на рис. 3.4. Вихідні операнди розміщуються в регістрах Rг1 і Rг3, звідки постійно можна переносити

їхній вміст відповідно в РгС і РгД. Для реалізації необхідних логічних операцій використовується *схема однобайтних логічних операцій СОЛО*, яка входить до складу АЛП і є комбінаційною схемою, що дає змогу реалізувати порозрядні операції логічного множення І, логічного додавання АБО і сумування за модулем 2 над двома однобайтними операндами. Результат обробки, тобто байт, фіксується у байтному регістрі Рг СОЛО, з якого результат можна переслати у вихідний регістр РгСм.

Крім вказаних логічних операцій, у СОЛО виконується операція порівняння двох байт, яка використовується, власне, при обробці порядків в операціях додавання і віднімання над числами з плаваючою крапкою.

3.4. Особливості виконання операцій десяткової арифметики

Арифметичні операції над десятковими числами (додавання, віднімання, множення, ділення) виконуються аналогічно до операцій над цілими двійковими числами.

Основою АЛП десяткової арифметики є *суматор двійково-десяткових кодів*. Такий суматор, як правило, будується на основі двійкового шляхом додавання деяких ланок.

Нехай необхідно додати модулі двійково-десяткових чисел X і Y . Перший доданок X перетворимо на код з надлишком 6 (позначимо X_6), який є результатом додавання до кожної цифри X двійкового числа 6. Перехід від X до X_6 змінює всі тетради X так, що в кожній тетраді X_6 міститься число 6–15.

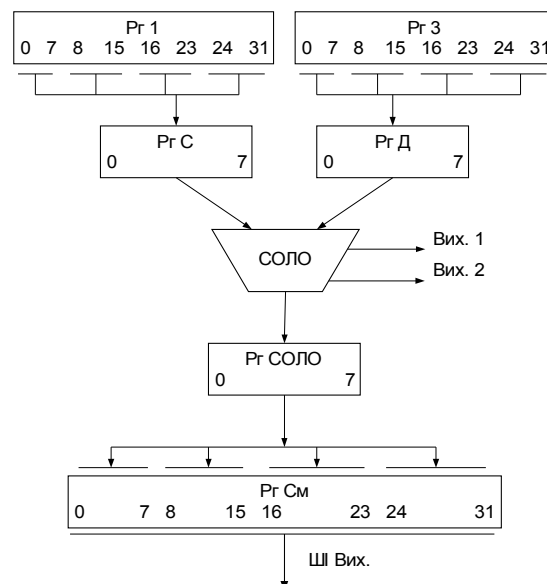


Рис. 3.4. Пристрій для виконання логічних операцій

Додаючи X_6 і Y за правилами двійкової арифметики, отримуємо результат Z' . У Z' одні тетради збігаються, а інші – не збігаються з тетрадами двійково-десятькової суми Z .

Якщо результат додавання в i -ому розряді $X[i]+Y[i]+P[i] \geq 10$, де $P[i]$ – десятковий перенос в i -й розряд, то i -а десяткова цифра $Z[i]=X[i]+Y[i]+P[i]-10$ і $P[i+1]=1$, де $P[i+1]$ – десятковий перенос в $(i+1)$ -й розряд.

Для Z' в даному разі отримаємо:

$$Z'[i] = X6[i] + Y[i] + P[i] - 16 = 6 + \underbrace{X[i] + Y[i] + P[i]}_{\geq 10} - 16 = Z[i].$$

При цьому виникає перенос в $(i+1)$ -у тетраду. Якщо i -а десяткова цифра $Z[i]$ має отримуватися з $X[i]+Y[i]+P[i] < 10$, то $Z[i]=X[i]+Y[i]+P[i]$ і $P[i+1]=0$.

Для $Z'[i]$ в даному разі отримаємо:

$$Z'[i] = X6[i] + Y[i] + P[i] = 6 + \underbrace{X[i] + Y[i] + P[i]}_{< 10} = Z[i] + 6.$$

Перенос в $(i+1)$ -у тетраду тут не виникає $P[i+1]=0$, оскільки $Z'[i] < 16$.

Таким чином, додаючи $X6$ і Y як двійкові числа, отримуємо Z' . Z' тетради, з яких виникав перенос, збігаються з тетрадами двійково-десятькового результату Z , а тетради, з яких не було переносу при додаванні, подані з надлишком 6. Для отримання суми Z необхідно відкоригувати Z' шляхом зменшення на 6 тих тетрад Z' , з яких не було переносу при додаванні $X6$ і Y . Віднімання 6 з тетрад, які потребують корекції, можна реалізувати шляхом підсумування 10 з одночасним ігноруванням переносу, що виникає при цьому з тетрад.

Приклад 3.1

Виконати додавання двох чисел $A = 57_{10}$ і $B = 34_{10}$.

Розв'язання. Двійково-десятьковий код 8421 для даних чисел буде мати такий вигляд: $A_{2-10} = 0101\ 0111$; $B_{2-10} = 0011\ 0100$.

Подамо число A в кодї з надлишком 6 ($6_{10} = 0110_2$), просумуємо $(A+6) + B$ і потім проведемо корекцію:

$$\begin{array}{r}
 A = 0101\ 0111 \\
 + \\
 \quad \underline{0110\ 0110} \\
 A_{\text{надл}} = 1011\ 1101 \\
 A_{\text{надл}} = 1011\ 1101 \\
 + \\
 B = \underline{0011\ 0100} \\
 \quad 1111\ 0001 \quad - \text{ попередня сума} \\
 + \\
 \quad \underline{0110} \quad - \text{ корекція} \\
 \quad 1001\ 0001 \quad - \text{ результат} \\
 \\
 1001\ 0001_{[2-10]} = 91_{10}.
 \end{array}$$

Приклад 3.2.

Виконати додавання двох чисел $X = 157_{10}$ і $Y = 134_{10}$.

Двійково-десятковий код 8421 для даних чисел буде мати такий вигляд:

$$X_{2-10} = 0001\ 0101\ 0111; Y_{2-10} = 0001\ 0011\ 0100.$$

подамо X в коді з надлишком 6, просумуємо X і Y та проведемо корекцію суми:

$$\begin{array}{r} X = 0001\ 0101\ 0111 \\ + \\ \quad \underline{0110\ 0110\ 0110} \\ X_{\text{надл}} = 0111\ 1011\ 1101 \\ X_{\text{надл}} = 0111\ 1011\ 1101 \\ + \\ Y = \underline{0001\ 0011\ 0100} \\ \quad 1000\ 1111\ 0001 \quad \text{– попередня сума} \\ + \underline{1010\ 1010} \quad \text{– корекція} \\ \quad 0010\ 1001\ 0001 \quad \text{– результат} \\ \quad 0010\ 1001\ 0001_{[2-10]} = 291_{10}. \end{array}$$

Віднімання двійково-десяткових модулів $X-Y$ здійснюється таким чином.

Усі розряди Y інвертуються, що дає доповнення кожної цифри Y до 15, при цьому отримується обернений код двійково-десятьового числа Y з надлишком 6, який позначимо $Y_{об6}$. Потім, додаючи $X+Y_{об6}$ і додаючи одиницю до молодшого розряду, отримуємо Z' .

Результат Z' є додатним числом, якщо з його старшої тетради виникає перенос, при цьому Z коригується за тими самими правилами, що й при додаванні модулів.

Якщо зі старшої тетради Z' немає переносу, то отримано від'ємний результат, який подано в доповняльному коді. В даному разі код Z' інвертується і до нього додається одиниця молодшого розряду. Нове Z' коригується, при цьому до тетрад, з яких виникав перенос при отриманні $(X+Y_{об6}+1)$, додається 10, а всі інші залишаються без змін.

Виконання додавання і віднімання чисел зі знаками зводиться до виконання додавання і віднімання модулів. Знак результату визначається окремо. Наприклад, при $X < 0$ і $Y < 0$ віднімання $X-Y$ замінюється відніманням $|Y|-|X|$. Знак результату змінюється на протилежний до знака $(|Y|-|X|)$.

3.5. Алгоритм додавання і віднімання чисел з плаваючою крапкою

Арифметичні операції над числами з плаваючою крапкою складніші, ніж операції над числами з фіксованою крапкою.

Додавання (віднімання) чисел з плаваючою крапкою виконується згідно з таким виразом за умови, що $|X| \geq |Y|$:

$$Z = X \pm Y = S^{P_x} q_x + S^{P_y} q_y (q_x \pm q_y / S^{(P_x - P_y)}) = S^{P_z} q_z \quad (3.5)$$

Алгоритм додавання і віднімання чисел з плаваючою крапкою

1. Проводиться вирівнювання порядків чисел. Порядок меншого (за модулем) числа береться рівним порядку більшого, а мантиса меншого числа зсувається вправо на кількість S -ричних розрядів, яка дорівнює різниці порядків чисел.

2. Проводиться додавання (віднімання) мантис, у результаті чого отримується мантиса суми (різниці).

3. Порядок результату береться рівним порядку більшого числа.

4. Отримана сума (різниця) нормалізується.

Вирівнювання порядків

Вирівнювання порядків починається з їхнього порівняння. Мантиса числа з меншим порядком при вирівнюванні зсувається вправо на число розрядів, яке дорівнює різниці порядків. Оскільки числа з плаваючою крапкою, які розглядаються, мають $S = 16$, зсув проводиться шістнадцятковими розрядами, тобто зсув проводиться на 4 двійкових розряди.

При порівнянні порядків можливі 5 випадків:

1) $P_x - P_y > m$ (m – число розрядів мантиси). Як результат сумування відразу можна взяти перший доданок, тому що при вирівнюванні порядків всі розряди мантиси другого доданка набувають нульового значення;

2) $P_y - P_x > m$. Як результат сумування може бути взятий другий доданок;

3) $P_x - P_y = 0$. Можна приступити до сумування мантис;

4) $P_x - P_y = k_1$ ($k_1 < m$). Мантиса другого доданка зсувається на k_1 розрядів вправо, а потім проводиться сумування мантис;

5) $P_y - P_x = k_2$ ($k_2 < m$). Перед сумуванням мантис проводиться зсув на k_2 розрядів вправо мантиси першого доданка.

За порядок результату при виконанні сумування береться більший з порядків операндів.

Додавання (віднімання) мантис проводиться за правилами додавання (віднімання) чисел з фіксованою крапкою.

Нормалізація суми (різниці) проводиться у разі невиконання умови $1 > q_z \geq 1/S$, при цьому якщо $|q_z| \geq 1$, P_z збільшується на одиницю, то мантиса q_z зсувається на один S -ічний розряд вправо, що дає $|q_z| < 1$. Якщо $|q_z| < 1/S$, то мантиса результату зсувається на розряд вліво при одночасному зменшенні порядку результату на одиницю. Ці операції

проводяться доти, доки не стане виконуватися умова $q_z \geq 1/S$ (при $q_z = 0$ нормалізація не виконується).

При отриманні порядку $+P_z$, який переповнює розрядну сітку, має формуватися сигнал переривання через переповнення порядку. При отриманні порядку $-P_z$, який переповнює розрядну сітку, відбувається формування нульового результату й ознаки втрати порядку.

В операціях з плаваючою крапкою, на відміну від операцій з фіксованою крапкою, додавання і віднімання виконуються наближено, тому що при вирівнюванні порядків здійснюється втрата молодших розрядів одного з доданків. У даному разі похибка завжди від'ємна і може доходити до одиниці молодшого розряду. Щоб зменшити похибку, застосовують заокруглення результату. Для цього може бути використаний додатковий розряд суматора, в який після виконання сумування додається 1.

Все сказане вище можна сформулювати й по-іншому.

У комп'ютерах застосовується визначений такий порядок виконання операції додавання чисел, які подано у формі з плаваючою крапкою:

1) віднімання порядків чисел, які додаються. Якщо різниця порядків дорівнює 0 (порядки рівні), то необхідно перейти до п. 3. Якщо порядки не рівні, то необхідно здійснити вирівнювання порядків (п. 2);

2) збільшення меншого з порядків до більшого. Мантиса числа з меншим порядком зсувається вправо на число розрядів, яке дорівнює різниці порядків;

3) додавання мантис. Якщо не відбулося переповнення розрядної сітки мантиси і сума отримана в нормалізованому вигляді, то обчислення потрібно закінчити. Сума мантис є мантисою суми чисел, а порядок суми чисел дорівнює порядку більшого числа (або спільному порядку). В іншому разі необхідно перейти до п. 4;

4) нормалізація отриманої суми вправо (при переповненні розрядної сітки мантиси) чи вліво (за наявності у мантисі нулів). При нормалізації вправо мантису суми потрібно зсунути вліво до першої значущої цифри, а порядок суми необхідно зменшити на таку саму кількість одиниць.

Приклад 3.3

$$X_1 = 0.1011001 \cdot 2^{011}; \quad X_2 = 0.1001101 \cdot 2^{011}; \quad X_1 + X_2 = ?$$

Розв'язання. Виконуємо п. 1:

$$P_1 = 0.011 \text{ і } P_2 = 0.011; \quad P_\Sigma = ?$$

код знака ← →

Замінімо різницю додатних чисел $P_1 - P_2$ сумою чисел з різними знаками $P_1 + (-P_2)$:

$$[P_1]_{об} = 0.011$$

+

$$[P_2]_{об} = 1.100$$

$$[P_1+P_2]_{об} = 1.111 \text{ – від'ємний нуль}$$

Оскільки $P_1 - P_2 = 0$, то $P_1 = P_2 = 0.011$. Тому переходимо до реалізації п. 3:
 $m_1 = 0.1011001$; $m_2 = 0.1001101$; $m_1 + m_2 = ?$

$$0.1011001$$

+

$$\underline{0.1001101}$$

$$1.0100110$$

При додаванні m_1 і m_2 відбулося переповнення розрядної сітки мантиси (денормалізація числа вліво), ознакою якого є наявність переносу в знаковий розряд і відсутність переносу зі знакового розряду. Отже, виконаємо п. 4.

Для нормалізації суми зсунемо мантису на 1 розряд вправо, і мантиса зменшиться на порядок (у два рази). Одночасно для збереження величини суми збільшимо на одиницю код порядку.

$$1.0100110 \rightarrow 0.1010011.$$

Після нормалізації отримаємо:

$$X_1 + X_2 = 0.1010011 \cdot 2^{10}$$

Приклад 3.4

$$X_1 = 0.1011001 \cdot 2^{011}; X_2 = -0.1001101 \cdot 2^{011}; X_1 + X_2 = ?$$

Розв'язання. Виконуємо п. 1, розв'язання якого аналогічне до прикладу 2.3.

Пункт 2 також не виконується, тому переходимо до виконання п. 3:

$$m_1 = 0.1011001; m_2 = -0.1001101; m_1 + m_2 = ?$$

$$0.1011001$$

+

$$\underline{1.0110010}$$

$$10.0001011 \Rightarrow 0.0001100$$

$$m_1 + m_2 = 0.0001100$$

Відбулася денормалізація числа вправо.

Для нормалізації числа отримана у п. 3 сума зсувається на три розряди вліво (збільшується у 2^3), а код порядку, сформований у п. 1, зменшується на три, тобто $P_\Sigma = 011 - 011 = 0$.

$$\text{Отже, } X_1 + X_2 = 0.1100000 \cdot 2^0.$$

Приклад 3.5

$$X_1 = 0.1101101 \cdot 2^{011}; X_2 = 0.1110001 \cdot 2^{001}; X_1 + X_2 = ?$$

Розв'язання. Пункт 1. $P_1 - P_2 = 010$.

Пункт 2. $P_2' = 010 + 1 = 011$; $m_2' = 0.001110001$

відкидається

При денормалізації числа вправо відбувається втрата точності його подання через обмеження розрядної сітки.

Пункт 3. $m_1 + m_2' = ?$

```
0.1101101
+
0.0011100
1.0001001
```

Відбулася денормалізація числа вліво.

Пункт 4. $m = 0.100100$; $P = 011 + 1 = 100$.

Після нормалізації

$$X_1 + X_2 = 0.100100 \cdot 2^{100}$$

Алгоритм множення чисел з плаваючою крапкою

Алгоритм множення чисел, які подано у формі з плаваючою крапкою, визначається таким співвідношенням:

$$(X_1 = m_1 \cdot 2^{P_1}) \cdot (X_2 = m_2 \cdot 2^{P_2}) = (m_1 \times m_2) \cdot 2^{P_1 + P_2}. \quad (3.6)$$

При реалізації операції множення над числами з плаваючою крапкою виділяють такі етапи:

1. Визначення знака добутку шляхом додавання за модулем 2 знаків мантис операндів

$$(0 \oplus 0 = 0; 0 \oplus 1 = 1; 1 \oplus 0 = 1; 1 \oplus 1 = 0).$$

2. Перемноження модулів мантис за правилами множення дробових чисел з фіксованою крапкою.

3. Визначення порядку добутку шляхом алгебраїчного сумування порядків співмножників (з використанням доповняльного, оберненого чи модифікованого коду).

4. Нормалізація результату (оскільки співмножники нормалізовані, то денормалізація можлива тільки на один розряд і тільки вправо) та заокруглення мантиси у разі необхідності.

4. ПОДАННЯ ІНФОРМАЦІЇ В КОМП'ЮТЕРАХ

4.1. Форми подання чисел і алфавітної інформації в комп'ютерах

Довільна інформація зображається в комп'ютерах у вигляді двійкових кодів (слів). Окремі компоненти двійкового коду, які можуть набувати значень 0 або 1, називаються розрядами або бітами. Отже, біт – це мінімальна одиниця інформації в комп'ютері, яка дорівнює двійковому розряду.

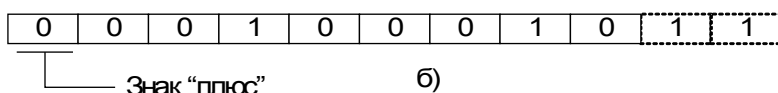
Слова в комп'ютерах розбиваються на байти (BYTE). Байт дорівнює 8 бітам.

У комп'ютерах застосовуються дві форми (режими): з *фіксованою і плаваючою крапками (комами)*. Ці форми називають також відповідно природною та нормальною.

При зображенні чисел з фіксованою крапкою положення останньої фіксується у строго визначеному місці відносно розрядів числа. На практиці поширені два випадки: крапка ставиться перед старшим або після молодшого розряду. В першому випадку можуть бути зображені лише числа, які за модулем менші від одиниці, а в другому – тільки цілі числа. Для чисел з фіксованою крапкою в комп'ютері необхідно у слові даних виділити *поле знака, поле крапки і поле цифр*. Для скорочення довжини розрядної сітки та спрощення опрацювання даних у конкретних типах комп'ютерів положення крапки фіксується схемотехнічно, тобто апаратними засобами. При цьому в словах даних зберігаються лише два структурних компоненти: поле знака і поле цифр (рис. 4.1, а). Очевидно, що обмеження довжини розрядної сітки (поля цифр) приводить до обмеження діапазону зміни чисел та втрати точності їхнього подання. Наприклад, якщо під поле цифр виділено 8 розрядів, то число 0.0010001011 буде сприйматися комп'ютером як 0.00100010 (рис. 4.1, б).



а)



б)

Рис. 4.1

Перші комп'ютери були машинами з фіксованою крапкою, причому крапка фіксувалася перед старшим розрядом числа. Зараз, як правило, форму з фіксованою крапкою використовують для подання цілих чисел (крапка фіксується після молодшого розряду).

На практиці, яка традиційно склалася в обчислювальній техніці, нумерація розрядів (біт) у комп'ютерах загального призначення розпочинається у напрямку зліва направо (див. рис. 4.1, а), а у малих, міні- та мікрокомп'ютерах і мікропроцесорах – справа наліво.

Для форматів з крапкою, що фіксується перед старшим розрядом (n), можуть бути подані з похибкою $2^{-(n-1)}$ числа (правильні дроби). Числа, для подання яких використовується прямий код, мають такий діапазон значень:

$$2^{-(n-1)} \leq |x| \leq 1 - 2^{-(n-1)}.$$

У даний час режим з фіксованою крапкою використовується (як основна форма подання чисел) лише у комп'ютерах, які мають обмежені обчислювальні можливості. Прикладами є комп'ютери у системах передавання даних, АСУ ТП, системах опрацювання вимірювальної інформації в реальному масштабі часу.

Слід зазначити, що *природна форма* потребує менше часу на опрацювання чисел порівняно з *нормальною формою*.

При *нормальній формі* (плаваюча крапка) запис одного числа може набувати різного вигляду залежно від обмежень, які накладаються на її форму. Наприклад, число 3552 можна записати таким чином: $3552 = 3.552 \cdot 10^3 = 0.3552 \cdot 10^4 = 35520 \cdot 10^{-1}$ і т. д. В загальному випадку нормальна форма запису числа може бути подана у такому вигляді:

$$X = \pm m \cdot q^{\pm p}, \quad (4.1)$$

де m – мантиса числа; q – основа системи числення; p – порядок числа.

Порядок (з урахуванням знака) показує, на скільки розрядів і в який бік зсунута крапка при зміні форми запису числа з природної на нормальну. Тому така форма називається *поданням чисел з плаваючою крапкою*.

Для подання числа в нормальній формі у слові даних виділяють чотири структурних компоненти (рис. 4.2): два поля знаків і два поля цифр.



Рис. 4.2

Нескладно переконатися, що в умовах обмеження розрядної сітки комп'ютера максимально можливою точністю подання чисел має нормальна форма запису числа, в якій старша цифра мантиси є значущою. Наприклад, $0.125 \cdot 10^3$; $0.1101 \cdot 10^2$. Подібні числа називаються нормалізованими. Число $X = mqr$ є нормалізованим, якщо задовольняється умова:

$$1 > |m| \geq 1/q. \quad (4.2)$$

Арифметичні дії над числами з плаваючою крапкою потребують виконання операцій над мантисами й операцій над порядками. Для спрощення операцій над порядками застосовують подання чисел з плаваючою крапкою зі зміщеним порядком.

4.2. Комп'ютерні коди: прямий, обернений і доповняльний



Код – це набір комбінацій, що складаються з різних елементарних сигналів, які називаються *елементами коду*.

Кодування – це процес однозначного відображення чисел (і символів) за допомогою кодових комбінацій.

У комп'ютерах числа зі знаком зображаються за допомогою *прямого, оберненого і доповняльного кодів*. Прямий код $[X]_{np}$ є найпростішим машинним кодом. Він формується кодуванням у числі X тільки знакової інформації, причому знак “+” кодується нулем, а знак “-” – одиницею.

Приклад 4.1

$$\begin{aligned} X_1 = -10011_2 &\rightarrow [X_1]_{np} = 1.10011 \\ X_2 = -0.11011_2 &\rightarrow [X_2]_{np} = 1.11011 \\ X_3 = +0.1101_2 &\rightarrow [X_3]_{np} = 0.1101 \end{aligned}$$

 код знака «-»
 код знака «+»

Слід наголосити, що якщо під поле цифр виділено більше розрядів, ніж це необхідно для подання числа X , то розряди (цифри) числа заносяться в розрядну сітку комп'ютера відповідно до своїх ваг.

Код знака числа практично у всіх комп'ютерах заноситься у старший розряд розрядної сітки.

Отже, при використанні, наприклад, 8-розрядної сітки наведені вище три числа у прямому коді будуть мати такий вигляд:

$$[X_1]_{np} = 1.0010011; [X_2]_{np} = 1.1101100; [X_3]_{np} = 0.11101000.$$

Прямий код числа широко використовується в комп'ютерах при зберіганні чисел у пам'яті і виконанні операцій множення. Алгебраїчне (з урахуванням знаків) додавання чисел у прямому коді ускладнене через необхідність виконання допоміжних дій (порівняння знаків доданків та за

модулем, додавання модулів і т. ін.). Для спрощення алгоритму алгебраїчного додавання використовують *обернений і доповняльний коди*.

Обернений код $[X]_{об}$ від'ємного числа формується за таким правилом: у знаковому розряді проставляється одиниця, а в усіх інших розрядах цифри замінюються на взаємно обернені.

$$[X]_{об} = 1.01100; [X_2]_{об} = 1.00100 \text{ – для 6-розрядної сітки.}$$

Доповняльний код $[X]_{доп}$ від'ємного числа утворюється з оберненого коду шляхом збільшення $[X]_{об}$ на одиницю молодшого розряду.

$$[X_1]_{доп} = 1.01101; [X_2]_{доп} = 1.00101.$$

Слід зауважити, що прямий, обернений і доповняльний коди додатних чисел збігаються.

Завдяки використанню оберненого і доповняльного кодів операція додавання від'ємних чисел зводиться у комп'ютерах до арифметичного додавання кодів цих чисел. Знакові розряди беруть участь в операції додавання нарівні з цифровими. При цьому якщо виконується операція додавання в обернених кодах, то одиниця переносу зі знакового розряду суми додається до її молодшого розряду (тобто виконується циклічний перенос). Якщо операція додавання виконується над числами, поданими в доповняльному коді, то одиниця переносу зі знакового розряду суми відкидається.

Перехід від оберненого і доповняльного кодів до прямого виконується аналогічно до переходу від прямого коду до оберненого і доповняльного.

Приклад 4.2

$$[X_1]_{об} = 0.001001; [X_2]_{об} = 1.001110; [X_1]_{об} + [X_2]_{об} = ?$$

$$\begin{array}{r} 0.001001 \\ +1.001110 \\ \hline 1.010111 \end{array}$$

$$\begin{aligned} [X_1]_{об} + [X_2]_{об} &= 1.010111 \\ X_1 + X_2 &= -0.101000 \end{aligned}$$

Результат додавання чисел, поданих в оберненому коді, теж є оберненим кодом (сума від'ємна).

Приклад 4.3

$$[X_1]_{об} = 0.110001; [X_2]_{об} = 1.110110; [X_1]_{об} + [X_2]_{об} = ?$$

$$\begin{array}{r} 0.110001 \\ +1.110110 \\ \hline 10.100111 \\ \swarrow + \quad 1 \\ \hline 0.101000 \end{array}$$

$$[X_1]_{об} + [X_2]_{об} = 0.101000$$

$$X_1 + X_2 = +0.101000$$

Сума додатна, тому результат відповідає прямому коду суми.

Приклад 4.4

$$[X_1]_{об} = 1.001110; [X_2]_{об} = 1.110110; [X_1]_{об} + [X_2]_{об} = ?$$

$$\begin{array}{r} 1.001110 \\ +1.110110 \\ \hline \overline{11.000100} \\ \leftarrow + \quad \quad \quad 1 \end{array}$$

$$1.000101$$

$$[X_1]_{об} + [X_2]_{об} = 1.000101$$

$$X_1 + X_2 = -0.111010$$

Приклад 4.5

$$[X_1]_{доп} = 0.001001; [X_2]_{доп} = 1.001111; [X_1]_{доп} + [X_2]_{доп} = ?$$

$$\begin{array}{r} 0.001001 \\ +1.001111 \\ \hline 1.011000 \end{array}$$

$$[X_1]_{доп} + [X_2]_{доп} = 1.011000$$

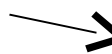
Для отримання результату суми необхідно провести перетворення на прямий код, тобто

$$X_1 + X_2 = -0.101000.$$

Приклад 4.6

$$[X_1]_{доп} = 0.110001; [X_2]_{доп} = 1.110111; [X_1]_{доп} + [X_2]_{доп} = ?$$


$$\begin{array}{r} 0.110001 \\ +1.110111 \\ \hline \underline{10.101000} \end{array}$$

 відкидається, отже, $X_1 + X_2 = 0.101000$.

Приклад 4.7

$$[X_1]_{доп} = 1.001111; [X_2]_{доп} = 1.110111; [X_1]_{доп} + [X_2]_{доп} = ?$$

$$\begin{array}{r} 1.001111 \\ +1.110111 \\ \hline \underline{11.000110} \end{array}$$

 відкидається, отже, $X_1 + X_2 = -0.111010$.

Як видно з наведених прикладів, додавання з використанням обернених кодів, як і в разі доповняльних кодів, виконується за звичайними правилами додавання двійкових чисел (правилами двійкової арифметики) (знаковий розряд розглядається нарівні з цифровими) з тією різницею, що в разі обернених кодів у процесі отримання суми реалізується циклічний перенос (за необхідності), тоді як при доповняльних кодах одиниця переносу зі знакового розряду відкидається. Відповідно, реалізація операції додавання в доповняльних кодах є простішою порівняно з реалізацією додавання в оберненому коді. Однак досягається це за рахунок ускладнення операції кодування чисел.

4.3. Кодування десяткових чисел і алфавітно-цифрової інформації

Комп'ютери перших поколінь обробляли тільки числа, і проблеми подання опрацювання алфавітно-цифрової інформації взагалі не було. Однак з появою алгоритмічних мов виникла необхідність подавати й опрацьовувати в комп'ютерах не тільки цифрову, а й символну інформацію.

Сукупність всіх символів, які використовують у комп'ютерах, утворює їхній алфавіт. Кожному символу відповідає машинна одиниця інформації – склад. Так називають групу двійкових розрядів, яка служить для подання символу в машині (двійковий код символу). Використовуються різні варіанти кодування символів, які застосовують коди різної довжини.

В даний час прийнято подавати один символ алфавітно-цифрової інформації у вигляді одного байта. За допомогою одного байта в загальному випадку можна закодувати $2^8 = 256$ символів. Історично склалося так, що різні виробники стали використовувати для подання символів усередині комп'ютерів різні коди. Це призвело до істотних технічних труднощів при стикуванні комп'ютерів різних типів. У даний час у світі найбільшого поширення набув стандартний американський код обміну інформацією ASCII, що має декілька модифікацій.

У базовому варіанті коду ASCII для кодування кожного символу використовується 7 біт, тобто можна закодувати $2^7 = 128$ символів, наприклад:

0 – 0011 0000	A – 0100 0001 – 41 ₁₆
1 – 0011 0001	B – 0100 0010 – 42 ₁₆
2 – 0011 0010	...
...	Z – 0101 1010 – 5A ₁₆
9 – 0011 1001	+ – 0010 1011 – 2B ₁₆
	= – 0011 1101 – 3D ₁₆

Код інформації і керуючих символів подається у вигляді дворозрядного шістнадцяткового числа.

Восьмий символ у байті може бути використаний для розширення набору символів, що відображається, або для перевірки правильності переданої кодової комбінації, наприклад, перевірки на парність.

У вітчизняних комп'ютерах також використовуються різні коди. Так, в ЕС ЕОМ застосовується двійковий код обміну інформації (ДКОІ), 8-розрядний код обміну інформацією (КОІ-8). Існують також 7-розрядні коди (КОІ-7).

Розглянемо приклади кодування алфавітно-цифрової інформації з використанням ДКОІ. Вираз «ТДЕУ 2005» буде зберігатись у пам'яті комп'ютера у вигляді слова із 8 байт:

Т	Д	Е	У	2	0	0	5
11100011	11000100	11000101	11101011	11110010	11110000	11110000	11110101

У кодї ДКОІ прийнято кодувати числа і знаки: “+” – 1110, “-” – 1101. Наприклад, число – 6,285 запишеться в ДКОІ у вигляді слова із 6 байт.

-	6	.	2	8	5
0110110	1111011	0100101	1111001	1111100	11110101
1	0	1	0	0	

З метою економії пам'яті та зручності виконання арифметичних операцій у комп'ютерах з байтним поданням інформації передбачені спеціальні формати для десяткових чисел: зонний («розпакований») і ущільнений («запакований»).

Число – 6,285 у зонному десятковому форматі (при роботі з ДКОІ) має такий вигляд:

6	2	8	-	5
1111011	1111001	1111100	110	0101
0	0	0	1	

Це ж число у запакованому десятковому форматі має вигляд:

0	6	2	8	5	-
000	011	001	100	010	1101
0	0	0	0	1	
байт		байт		байт	

У комп'ютерах, які використовують ці формати, є команди, що роблять перетворення чисел із зонного на запакований формат, і навпаки.

5. ПАМ'ЯТЬ КОМП'ЮТЕРІВ

5.1. Ієрархічний принцип побудови пам'яті

Пам'яттю комп'ютерів називається сукупність пристроїв, що служать для запам'ятовування, збереження і видачі інформації. Деякі пристрої, які входять до цієї сукупності, називають запам'ятовувальними пристроями або пам'яттю того чи іншого типу.

Обидва цих терміни в даний час стали майже синонімами. Однак термін «запам'ятовувальний пристрій» (ЗП), як правило, застосовують, коли йдеться про принципи побудови деякого пристрою пам'яті (наприклад, напівпровідникові ЗП, ЗП на магнітних дисках, магнітооптичні ЗП і т. ін.), а термін «пам'ять» – коли хочуть вказати на логічну функцію, яку виконує пристрій пам'яті, або місце розміщення у складі обладнання комп'ютера (наприклад, оперативна пам'ять, постійна пам'ять і т. ін.).

Основними операціями у пам'яті в загальному випадку є занесення інформації в пам'ять – запис і вибірка інформації з пам'яті – зчитування. Обидві ці операції називаються зверненням до пам'яті або зверненням при зчитуванні і зверненням при запису.

Важливими характеристиками окремих пристроїв пам'яті (запам'ятовувальних пристроїв) є ємність пам'яті, питома ємність та швидкодія.

Ємність пам'яті визначається максимальною кількістю даних, які можуть зберігатися. Ємність вимірюється у двійкових одиницях (бітах), машинних словах, але найчастіше в байтах (1 байт = 8 біт). При цьому доволі часто ємність пам'яті виражають у кілобайтах (1 Кбайт = 1024 байт), мегабайтах (1 Мбайт = 1024 К байт), гігабайтах і терабайтах.

Питома ємність є відношенням ємності ЗП до його фізичного об'єму.

Швидкодія пам'яті визначається тривалістю операції звернення, тобто часом, який затрачається на пошук необхідної одиниці інформації в пам'яті і на її зчитування, або часом на пошук місця в пам'яті для зберігання даної одиниці інформації і на її запис у пам'ять.

Тривалість звернення до пам'яті (час циклу пам'яті) при зчитуванні:

$$t_{зв}^{3ч} = t_{\partial}^{3ч} + t_{зч} + t_{рег}, \quad (5.1)$$

де $t_{\partial}^{3ч}$ – час доступу, який визначається проміжком часу між моментом початку операції звернення при зчитуванні, до моменту, коли є можливим доступ до даної одиниці інформації;

$t_{зч}$ – тривалість самого фізичного процесу зчитування;

$t_{рег}$ – час, який затрачається на регенерацію інформації, яка руйнується при зчитуванні.

Тривалість звернення до пам'яті (час циклу) при записі:

$$t_{зв}^{зан} = t_{\partial}^{зан} + t_{нідз} + t_{зан} \quad , \quad (5.2)$$

де $t_{\partial}^{зан}$ – час доступу при записі, тобто час від моменту початку звернення при записі, до моменту, коли є можливим доступ до запам'ятовувальних елементів, в які проводиться запис інформації;

$t_{нідз}$ – час підготовки, тобто час, який затрачається на приведення у вихідний стан запам'ятовувальних елементів або ділянок поверхні носія інформації для запису заданої одиниці інформації.

У більшості випадків $t_{\partial}^{зч} = t_{\partial}^{зан} =$.

За принципом дії пристрої пам'яті можна поділити на:

- електронні (напівпровідникові), де як запам'ятовувальні елементи використовуються напівпровідникові електронні елементи;
- магнітні з нерухомими магнітними запам'ятовувальними елементами;
- магнітомеханічні з рухомих магнітним носієм інформації;
- оптичні;
- магнітооптичні.

Залежно від операцій звернення, що реалізуються в пам'яті, розрізняють:

- пам'ять з довільним зверненням (можливе зчитування і запис у пам'ять);
- пам'ять тільки для зчитування інформації («постійна» або «одностороння» пам'ять). Занесення (запис) інформації в постійну пам'ять проводиться у процесі її виготовлення чи на стадії налагодження (в лабораторних умовах з використанням програматорів).

Найважливіші параметри ЗП суперечать одні одним. Так, наприклад, велика інформаційна ємність не поєднується з високою швидкістю, а водночас швидкість не поєднується з низькою вартістю. Тому системам пам'яті властива *багатоступінчаста ієрархічна структура*, і залежно від ролі того чи іншого ЗП його реалізація може бути суттєво різною.

У загальному випадку пам'ять комп'ютера (рис. 5.1) складається з надоперативної пам'яті (НОЗП) або місцевої пам'яті, оперативної пам'яті (ОП), пам'яті з прямим доступом на магнітних барабанах, магнітних та магнітооптичних дисках, пам'яті з послідовним доступом на магнітних стрічках і на оптичних дисках.

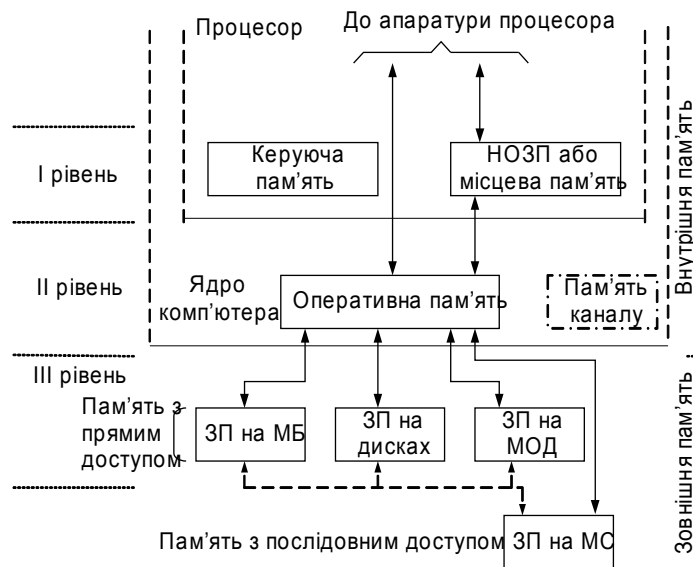


Рис. 5.1. Ієрархічна структура пам'яті комп'ютера

Кожен рівень ієрархії може вміщувати декілька екземплярів (модулів) відповідних пристроїв для отримання необхідної ємності цього рівня пам'яті.

Ієрархічна структура пам'яті дає змогу економічно ефективно поєднувати збереження великих обсягів інформації зі швидким доступом до інформації у процесі опрацювання.

За основу реалізації ієрархії пам'яті сучасних комп'ютерів взято два принципи: принцип локальності звернень і відношення вартість/продуктивність. Принцип локальності звернень свідчить про те, що більшість програм не виконує звернень до всіх своїх команд і даних рівноймовірно, а віддає перевагу деякій частині свого адресного простору.

Ієрархія пам'яті сучасних комп'ютерів базується на декількох рівнях, причому вищий рівень менший за обсягом, швидший і має більшу вартість у перерахунку на байт, ніж нижчий рівень. Рівні ієрархії взаємно зв'язані: всі дані на одному рівні можуть бути знайдені на нижчому рівні, а всі дані на цьому нижчому рівні можуть бути знайдені на наступному нижчому рівні і т. д., поки ми не досягнемо основи ієрархії.

Ієрархія пам'яті складається з декількох рівнів, але в кожному момент часу маємо справу тільки з двома найближчими рівнями. Мінімальна одиниця інформації, яка або наявна, або відсутня у дворівневій ієрархії, називається *блоком*. Розмір блока може бути або

фіксованим, або змінним. Якщо цей розмір зафіксований, то обсяг пам'яті є кратним розміру блока.

Успішне або неуспішне звернення до вищого рівня називають відповідно *влученням* (hit) або *промахом* (miss). *Влучення* – це звернення до об'єкта пам'яті, знайденого на вищому рівні, тоді як промах означає, що він (об'єкт) не знайдений на цьому рівні. Частка влучень (hit rate), або коефіцієнт влучень (hit ratio) є часткою звернень, не знайдених на вищому рівні. Частка промахів (miss rate) є часткою звернень, не знайдених на нижчому рівні.

Оскільки підвищення продуктивності – це головна причина появи ієрархії пам'яті, частота влучень і промахів стає важливою характеристикою. Час звернення при влученні (hit time) є часом звернення до вищого рівня ієрархії, який охоплює також час, необхідний для визначення того, чи є звернення влученням чи промахом. Втрати на промах (miss penalty) є часом для заміщення блока у вищому рівні на блок з нижчого рівня плюс час для пересилання цього блока в необхідний пристрій (наприклад, у процесор).

В ієрархічній структурі підсистем пам'яті ПК можна виділити такі рівні:

- регістрова пам'ять (реєстри мікропроцесора);
- буферна або кеш-пам'ять;
- оперативна пам'ять або ОЗП;
- зовнішня пам'ять або ЗЗП.

Регістрова пам'ять – це власні реєстри мікропроцесора, тому при роботі з ними у МП не виникає проблем, тим більше, що час доступу до реєстрів мінімальний.

Оперативна пам'ять має відносно малу швидкодію, що вже при тактовій МП у 33 МГц гальмує роботу комп'ютера. Це зумовило необхідність встановлення між ОЗП і мікропроцесором проміжної пам'яті (буферної або кеш-пам'яті), яка узгоджує швидкість роботи МП і ОЗП.

5.2. Види запам'ятовувальних пристроїв та їхні особливості

Пам'ять комп'ютера разом з основною (оперативною) пам'яттю високої швидкодії, але порівняно невеликої ємності вміщує зовнішню пам'ять, яка є набагато повільнішою, але може зберігати практично який завгодно великий обсяг необхідної для комп'ютера чи системи інформації (дані, програми).

Зовнішня пам'ять складається з декількох зовнішніх запам'ятовувальних пристроїв (ЗЗП), в ролі яких у сучасних комп'ютерах використовуються ЗП з носієм інформації у вигляді рухомої поверхні.

Зовнішні ЗП є пристроями з довільним зверненням, які допускають багатократне зчитування і запис нової інформації на місце записаної раніше.

Необхідний обсяг зовнішньої пам'яті досягається під'єднанням до комп'ютера відповідної кількості ЗЗП. Звернення до ЗЗП в загальному випадку передбачає послідовне виконання двох процесів:

- доступу до ЗЗП – підведення до головки (головок) ділянки носія, де міститься необхідна інформація або куди інформація має бути записана;

- зчитування і передача інформації із ЗЗП в ОП або передача інформації з ОП у ЗЗП і запис її на носій.

На носії, як правило, інформація розташовується впорядковано, тому доцільно, щоб при зверненні до ЗЗП відбувалися запис і зчитування не окремого слова або байта, а послідовно розміщеного на носії блока чи масиву даних.

Зовнішні ЗЗП поділяються на пристрої з прямим і послідовним доступом. У пристроях з прямим доступом час доступу практично мало залежить від положення носія відносно головки (головок) у момент ініціації звернення до ЗЗП, що досягається циклічним рухом носія з великою швидкістю відносно головки (головок). У пристроях з послідовним доступом (ЗЗП на магнітних стрічках) для пошуку необхідної ділянки носія необхідно послідовно проглянути записану на носії інформацію, що дуже зменшує швидкодію.

Накопичувачі на магнітних стрічках

Накопичувачі на магнітних стрічках (НМС) належать до ЗЗП послідовного доступу з рухомим магнітним запам'ятовувальним елементом, в ролі якого використовується магнітна стрічка шириною від 3,81 до 25,4 мм. НМС широко застосовуються для довгострокового зберігання великих масивів інформації і створення архівів з тієї причини, що мають низьку вартість зберігання одиниці інформації та високу надійність. Основний недолік НМС – великий час доступу.

В даний час у більшості систем НМС, як правило, під'єднуються до комп'ютера за допомогою шини SCSI. Дуже часто до цієї ж шини під'єднуються і дискові накопичувачі. Високий коефіцієнт використання шини SCSI практично всіма типами НМС стає критичним фактором при організації резервного копіювання і відновлення інформації, особливо в серверах з високим ступенем готовності. Очевидно, така висока завантаженість шини SCSI (до 20 – 65% пропускної здатності шини) при

роботі НМС накладає певні обмеження як на конфігурацію й тип НМС, так і на організацію самого резервного копіювання.

Найбільш вживаними на сьогодні є НМС з цифровою аудіострічкою (DAT) шириною 8 і 4 мм, які використовують технологію спірального сканування. На відміну від традиційних НМС зі стаціонарними головками й обмеженим числом доріжок, ці пристрої здійснюють зчитування і запис даних на магнітну стрічку, яка рухається повільно, за допомогою головок, розміщених на барабані, що дуже швидко обертається. При цьому доріжки перетинають стрічку з краю на край і розміщені під невеликим кутом у напрямі, перпендикулярному до напрямку руху стрічки. Іноді цю технологію називають «поперечним записом». На сьогодні подібні пристрої дають найвищу поверхневу густину запису. Наприклад, накопичувач EXB-8200 – близько 75 Мбіт на квадратний дюйм (2,54 см²). Пристрої DAT записують дані на стрічку шириною 4 мм з густиною 114 Мбіт на дюйм, що є близьким до теоретичної межі густини запису. Подальше її збільшення потребує зміни типу носія або використання технології *компресії* (стиснення) даних.

Найбільш популярними протягом багатьох років були 150–200 Мб картриджів QIC з шириною стрічки ¼ дюйма. На сьогодні існують 10 стандартів для картриджів конструктиву 5,25" і 9 стандартів міні-картриджів конструктиву 3.5". Картриджі найбільшої ємності дають змогу здійснювати запис до 13 Гбайт даних.

Одним з порівняно нових напрямків у сфері резервного копіювання є поява пристроїв стрічкових масивів (аналогічних до дискових масивів), які використовуються здебільшого у системах високої готовності. Прикладом такого пристрою може бути CKRiiON Series 4000 tape array компанії «Data General». Він може мати у своєму складі до п'яти накопичувачів з DAT шириною 4 мм загальною ємністю до 25 Гбайт.

Накопичувачі на оптичних дисках

Найбільш значні успіхи у сфері розробки масових запам'ятовувальних пристроїв досягнуті за останніх декілька років і стосуються переважно розробки пристроїв оптичної пам'яті.

Однією з причин значного поширення CD-ROM стало різке зростання обсягу програмних пакетів. Дистрибутиви обсягом три десятки дискет по 1,44 Мб стали звичними. Поширення такого програмного забезпечення на дискетах не лише не вигідне, а й доволі ненадійне: дискети не є найбільш надійним способом зберігання інформації.

Друга причина – це поява дешевих комп'ютерів, які оснащені засобами multimedia. Звук, а особливо відео потребують доволі великих

обсягів зовнішньої пам'яті, і найкраще для цього підходить CD-ROM: недорогий змінний носій ємністю 700 Мбайт.

Ще одна з причин полягає в тому, що CD-ROM – пристрій, інформація на якому не лише надійно захищена від випадкових збоїв при читанні, а й непідвладна випадковому знищенню чи перезапису при помилках користувача або «роботі» вірусів.

Накопичувачі на магнітооптичних дисках

Стрибок у розвитку комп'ютерної техніки і тенденції до постійного зростання потужних комп'ютерних інформаційних систем зумовили різке збільшення обсягів обробки інформації. Ця обставина спонукає вирішувати проблему зберігання такої інформації і типів пристроїв для систем резервного копіювання.

При вирішенні цієї проблеми широкого визнання набуває магнітооптична технологія, яка використовує магнітні й оптичні механізми запису і читання.

На сьогодні існує декілька стандартних типів магнітооптичних дисководів, серед яких найбільш поширеними є два з них. Це 3,5-дюймові і 5,25-дюймові накопичувачі. Стандартні ємності 3,5-дюймових дисків – 128, 230 і 640 Мб. У цих дисків одна робоча поверхня, їхній розмір відповідає розміру звичайної 3,5-дюймової дискети. МО диски розміром 5,25 дюйма мають стандартні ємності 600 і 650 Мб чи 1,2 і 1,3 Гбайт (диски подвійної густини запису).

Незважаючи на велику ємність МО дисків (на сьогодні існують 5,25-дюймові диски ємністю 4,6 Гбайт і більше), вони не можуть замінити НЖМД, передусім через низьку швидкодію. Швидкодія МО дисководів суттєво знижується при записі інформації на диск. Запис інформації на МО диск здійснюється за два проходи: при першому проході дані стираються з диска, а при другому – записуються на диск.

Магнітооптичні накопичувачі випускаються у двох варіантах: вмонтовані і зовнішні. Перевага зовнішніх накопичувачів полягає в тому, що нагрівання дисководу під час роботи лазера не підвищує температуру в корпусі самого комп'ютера. Вмонтовані накопичувачі можуть бути легко встановлені на місце звичайного флопі-дисководу.

5.3. Віртуальна пам'ять та її організація

Загальноприйнята в даний час концепція *віртуальної пам'яті* виникла достатньо давно. Вона дала змогу вирішувати цілий ряд актуальних питань організації обчислень. Перш за все, до таких питань належить забезпечення надійного функціонування мультипроцесорних систем.

Віртуальна пам'ять – це спосіб організації основної пам'яті великої ємності за допомогою зовнішньої пам'яті. Вона дає змогу при складанні програми розпоряджатися всім простором адрес, зарезервованих у процесорі.

У будь-який момент часу комп'ютер виконує множину процесів чи задач, кожна з яких має у своєму розпорядженні свій адресний простір. Було би дуже дорого віддавати всю фізичну пам'ять якій-небудь одній задачі, тим більше, що багато задач реально використовують тільки невелику частину адресного простору. Тому необхідним є механізм розподілу фізичної пам'яті між різними задачами. Віртуальна пам'ять – один зі способів реалізації такої можливості. Вона розбиває фізичну пам'ять на блоки і розподіляє їх між різними задачами. При цьому вона передбачає також деяку схему захисту, що обмежує задачу тими блоками, які їй належать.

Друге питання, тісно пов'язане з реалізацією концепції віртуальної пам'яті, стосується організації обчислень на комп'ютері задач дуже великого обсягу. Якщо програма є надто великою для фізичної пам'яті, то частину її необхідно було би зберігати в зовнішній пам'яті (на диску) і завдання щодо пристосування її для розв'язування на комп'ютері лягало на плечі програміста. Програмісти поділяли програми на частини і потім визначали ті з них, які можна було вирішувати незалежно, організовуючи при цьому так звані оверлейні структури, що завантажувалися в основну пам'ять і вивантажувалися з неї під керуванням програми користувача. Програміст повинен був стежити за тим, щоб програма не зверталася у не відведений їй простір фізичної пам'яті.

Віртуальна пам'ять звільнила програмістів від цього процесу. Вона автоматично керує двома рівнями ієрархії пам'яті: і основною пам'яттю, і зовнішньою (дисковою) пам'яттю.

Крім цього, віртуальна пам'ять спрощує також завантаження програм, забезпечуючи механізм автоматичного переміщення програм, який дає змогу виконувати одну і ту саму програму в довільному місці фізичної пам'яті.

Системи віртуальної пам'яті можна поділити на два класи: системи з фіксованим розміром блоків, які називаються сторінками, і системи зі змінним розміром блоків, які називаються сегментами.

Сторінкова організація пам'яті

У системах зі сторінковою організацією основна і зовнішня пам'ять (дисковий простір) поділяються на блоки або сторінки фіксованої довжини. Кожному користувачу надається деяка частина адресного

простору, яка може перевищувати основну пам'ять комп'ютера і яка обмежена тільки можливостями адресації, закладеними у системі команд. Ця частина адресного простору називається *віртуальною пам'яттю користувача*. Кожне слово у віртуальній пам'яті користувача визначається віртуальною адресою, яка складається з двох частин: старші розряди адреси розглядаються як номер сторінки, а молодші – як номер слова (байта) у середині сторінки.

Керування різними рівнями пам'яті здійснюється програмами ядра операційної системи, які стежать за розподілом сторінок і оптимізують обмін між цими рівнями. При сторінковій організації пам'яті суміжні віртуальні сторінки не обов'язково мають розміщуватися на суміжних сторінках основної фізичної пам'яті (рис. 5.2).

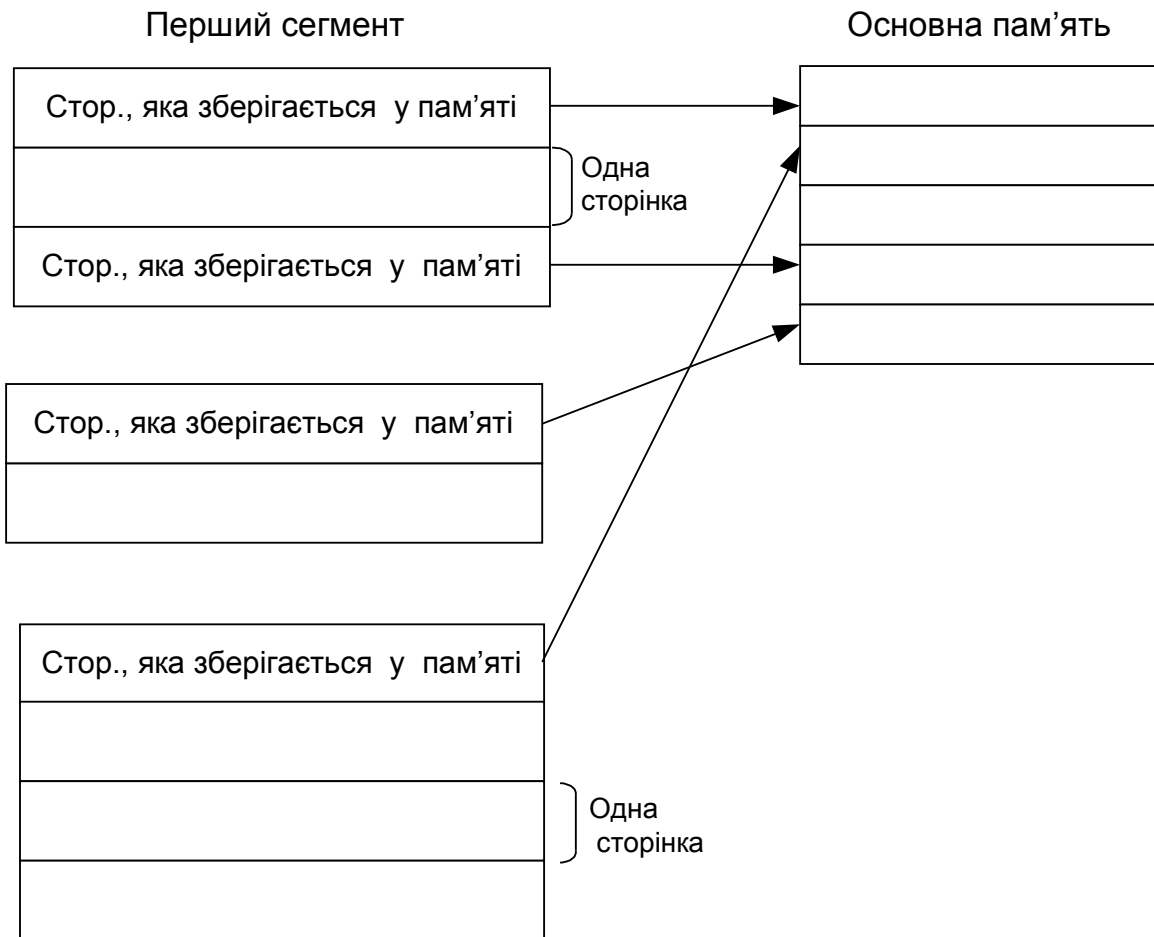


Рис. 5.2. Керування пам'яттю зі сторінковою організацією

Для встановлення відповідності між віртуальними сторінками і сторінками основної пам'яті операційна система має сформувати таблицю сторінок для кожної програми і розмістити її в основній пам'яті комп'ютера. При цьому кожній сторінці програми незалежно від того, міститься вона в основній пам'яті чи ні, ставиться у відповідність деякий

сегмент таблиці сторінок. Кожний сегмент таблиці сторінок вміщує номер фізичної сторінки основної пам'яті і спеціальний індикатор. Одиничний стан цього індикатора вказує на наявність цієї сторінки в основній пам'яті, а нульовий стан означає відсутність сторінки в основній пам'яті.

Пошук у таблицях сторінок, розміщених в основній пам'яті, здійснюється або програмним способом, або спеціальними апаратними засобами. В другому випадку для того, щоб запобігти можливості звернення програми користувача до таблиць сторінок, з якими вона не зв'язана, передбачені спеціальні заходи. З цією метою у процесорі передбачається (спеціальний) додатковий регістр захисту, який вміщує дескриптор таблиці сторінок (рис. 5.3) або базово-граничну пару. В полі керуючої інформації біт Р вказує, що сегмент міститься в основній пам'яті. Коли процесор звертається до пам'яті, пристрій керування пам'яттю досліджує вміст дескриптора цього сегмента, і якщо $P = 1$, то він здійснює перетворення віртуальної адреси на фізичну, в іншому разі ($P = 0$), цей пристрій передає у процесор сигнал про відсутність сегмента. При здійсненні запису в цей сегмент біт запису $W = 1$.

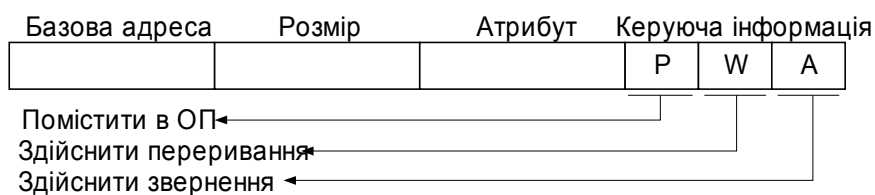


Рисунок 5.3 – Дискриптор сегмента віртуальної пам'яті

Сегментація пам'яті

Інший підхід до організації пам'яті базується на тому, що програми звичайно розподіляються на окремі ділянки – сегменти. Кожний сегмент є окремою логічною одиницею інформації, яка вміщує сукупність даних чи програм і розміщена в адресному просторі користувача. Сегменти створюються користувачами, які можуть звертатися до них за *символічним ім'ям*. У кожному сегменті встановлюється своя нумерація слів, починаючи з нуля.

Сегменти – це окремі логічні одиниці інформації, які необхідно захищати, і саме на цьому рівні вводяться різні режими доступу до сегментів. Можна виділити два основних типи сегментів: *програмні*

сегменти й сегменти даних. Із програмних сегментів допускаються тільки вибірка команд і зчитування констант. У системах із сегментацією пам'яті кожне слово в адресному просторі користувача визначається віртуальною адресою, яка складається з двох частин: старші розряди адреси розглядаються як номер сегмента, а молодші – як номер слова в середині сегмента.

Як і в разі зі сторінковою організацією, необхідно забезпечити перетворення віртуальної адреси на реальну фізичну адресу основної пам'яті. З цією метою для кожного користувача операційна система має сформувати таблицю сегментів. Таблиці сегментів різних користувачів операційна система зберігає в основній пам'яті. З метою визначення розміщення таблиці сегментів для програми, яка виконується, використовується спеціальний реєстр захисту, що завантажується операційною системою перед початком її виконання.

Для реалізації сегментації було запропоновано декілька схем, які відрізняються деталями реалізації, але за їхню основу взято одні й ті самі принципи.

5.4. Молекулярна і голографічна пам'ять та основні принципи їхньої побудови

Молекулярна пам'ять

Диски CD-ROM (CD-RW) на сьогодні дають змогу зберігати до 700 МВ інформації. Технології магнітного запису також розвиваються дуже швидко – за останній час місткість жорстких дисків у настільних комп'ютерах зросла до 400 ГБайт. Однак комп'ютерам у майбутньому доведеться обробляти сотні гігабайт і навіть терабайти інформації – набагато більше, ніж може вмістити будь-який з існуючих сьогодні дисків.

Обслуговування таких обсягів даних та переміщення їх для опрацювання надшвидкісними процесорами потребують радикально нових підходів при створенні пристроїв зберігання інформації.

Одним з радикально нових підходів у створенні пристроїв зберігання даних є так званий *молекулярний*.

Група дослідників центру W. M. Keck Center for Molecular Electronics (під керівництвом професора Роберта Р. Бірґа (Robert R. Birge)) отримала прототип підсистеми пам'яті, яка використовує для запам'ятовування цифрових бітів молекули. Це молекули протеїну, що називається бактеріородопсин. Цей мікроорганізм “проживає” у соляних болотах, де температура може досягати +150°C.

Р. Бірґ вибрав бактеріородипсин, тому що фотоцикл (послідовність структурних змін, що відбуваються з молекулою при реакції зі світлом)

робить цю молекулу ідеальним логічним елементом типу “AND” або типу перемикача з одного стану в інший (тригер). Як показали дослідження Бірга, bR-стан (значення біта «0») і Q-стан (значення біта «1») є проміжними станами молекули й можуть залишатися стабільними протягом багатьох років.

За оцінками Бірга, дані, записані на бактеріородопсоновому запам'ятовувальному пристрої, мають зберігатися приблизно п'ять років. Іншою важливою особливістю є те, що ці два стани мають спектри поглинання, які суттєво відрізняються. Це допомагає легко визначити поточний стан молекули за допомогою лазера, налагодженого на відповідну частоту.

Р. Бірг побудував прототип системи пам'яті, в якому бактеріородопсин запам'ятовує дані у тривимірній (3-D) матриці. Така матриця являє собою кювету (прозору посудину), заповнену поліакридним гелем, в який поміщено протеїн. Протеїн, що перебуває у bR-стані, фіксується у просторі при полімеризації гелю. Кювету оточують батарея лазерів і детекторна матриця, побудована на базі приладу, що використовує принцип зарядової інжекції, які застосовуються для запису і зчитування даних. При запису даних спочатку необхідно запалити жовтий «сторінковий» лазер для переведення молекул в Q-стан. Просторовий світловий модулятор (SLM), який являє собою LCD-матрицю, що створює маску на шляху променя, викликає виникнення активної площини в матеріалі всередині кювети. Ця енергоактивна площина є сторінкою даних, що може вміщувати масив $4096 * 4096$ біт.

Перед поверненням протеїну в стан спокою (у ньому він може перебувати тривалий час, зберігаючи інформацію) запалюється червоний, записуючий лазер, який, як правило, розміщений під прямим кутом відносно жовтого. Другий SLM відображає двійкові дані й таким чином створює на шляху променя відповідну маску, тому опроміненню піддаються тільки певні місця сторінки. Молекули в цих місцях перейдуть у Q-стан і будуть являти собою двійкову одиницю. Інша частина сторінки повернеться у початковий Q-стан та буде представляти двійкові нулі. Для того, щоб прочитати дані, потрібно знову запалити сторінковий лазер, який переводить сторінку читання у O-стан. Це робиться для того, щоб надалі за допомогою відмінності у спектрах поглинання ідентифікувати двійкові нулі й одиниці. Через 2 мс після цього сторінка «занурюється» в низькоінтенсивний світловий потік червоного лазера. Низька інтенсивність необхідна для того, щоб попередити «перестрибування» молекул у Q-стан. Молекули, які представляють двійковий нуль, поглинають червоне світло, а які представляють двійкову одиницю – пропускають промінь повз себе. Це

створює «шаховий» малюнок зі світлих і темних місць на CID-матриці, яка охоплює сторінку цифрової інформації.

Для стирання даних достатньо короткого імпульсу синього лазера, щоб повернути молекули з Q-стану у вихідний Q-стан. Синій колір не обов'язково має створюватися лазером: можна стерти всю кювету за допомогою ультрафіолетової лампи. При операціях читання/запису використовуються два додаткових біти парності для захисту від помилок. Сторінка даних може бути прочитана без зруйнування до 5000 разів. Кожна сторінка відстежується лічильником, і після 1024-го зчитування сторінка перезаписується (поновлюється) за допомогою нової операції запису.

Голографічна пам'ять

Широкі перспективи щодо побудови систем пам'яті відкриває технологія оптичного запису. Ця технологія відома як *голографія*: вона дає змогу забезпечити дуже високу густину запису при збереженні максимальної швидкості доступу до інформації. Це досягається за рахунок того, що голографічний образ (голограма) кодується в один великий блок даних, який записується всього за одне звернення. При зчитуванні інформація повністю зчитується з пам'яті.

Для зчитування чи запису блоків даних («сторінок»), які голографічно зберігаються на світлочутливому матеріалі, використовуються лазери. Теоретично тисячі таких цифрових сторінок, кожна з яких вміщує до мільйона біт, можна помістити у пристрій розміром у шматок цукру (кісточка). При цьому досягається густина даних в $1 \text{ TB}/\text{cm}^3$. Практично розробники прогнозують і очікують густини близько $10 \text{ GB}/\text{cm}^3$, а це вражає, якщо порівняти на сьогодні з магнітним способом – $100 \text{ KB}/\text{cm}^2$. При такій густині запису оптичний шар, що приблизно збігається за товщиною з колодою гральних карт, дасть змогу зберігати близько 1 TB інформації.

Як функціонує система *голографічної пам'яті*? Функціонування подібної пам'яті можна розглянути на прикладі установки, розробленої науковою групою з Almaden Research Center.

На початковому етапі у пристрої відбувається поділ променя синьо-зеленого аргонного лазера на дві складові – опорний і предметний промені (останній є носієм самих даних). Предметний промінь піддається розфокусуванню для того, щоб він міг повністю освітлювати просторовий світловий модулятор (Spatial Light Modulator – SLM). SLM представляє собою рідкокристалічну панель (LCD), на якій сторінка даних відображається у вигляді матриці, що складається зі світлих та темних пікселів (двійкові дані).

Обидва промені спрямовуються в середину світлочутливого кристала, де й відбувається їхня взаємодія. В результаті цієї взаємодії утворюється інтерференційна картина, яка є основою голограми і запам'ятовується у

вигляді набору варіацій показника заломлення чи коефіцієнта відбивання в середині цього кристала. При читанні даних кристал освітлюється опорним променем, який, взаємодіючи з кристалічною інтерференційною картиною, відтворює записану сторінку у вигляді образу «шахівниці» зі світлих і темних пікселів. Потім цей образ спрямовується в матричний детектор, що захоплює всю сторінку даних. При читанні даних опорний промінь має падати на кристал під аналогічним кутом. Змінюючи кут опорного променя чи його частоту, можна записати додаткові сторінки даних на тому самому кристалі.

Оскільки інтерференційні шаблони однорідно заповнюють весь матеріал, то це дає підстави зробити висновок про високу достовірність записаної інформації.

5.5. Адресація пам'яті комп'ютерів

Способи адресації

Спосіб адресації – спосіб обчислення адреси операнда на основі інформації, яка міститься в команді. У комп'ютерах першого покоління використовувався переважно тільки один спосіб адресації – пряма адресація.

Недоліки цього способу зумовили необхідність застосування поряд з прямою адресацією також інших способів.

У системі команд процесора, як правило, передбачається декілька способів адресації операндів для однієї і тієї самої операції. Тому в адресній частині команди є спеціальне поле – поле ознак операції, в якому кодується ознака адресації. Якщо такої ознаки немає, то спосіб адресації визначається за кодом операції. При використанні поля ознак адресації будь-яка операція може виконуватися будь-якими способами адресації, що спрощує програмування.

Неявна адресація

Цей спосіб адресації застосовується у всіх процесорах. Основне його призначення полягає у зменшенні довжини команд за рахунок виключення частини адрес.

При неявній адресації адреса обчислюється за будь-якою формулою з відомостей, що містяться в команді, або як операнд використовується вміст фіксованої комірки пам'яті чи регістра процесора. Перший спосіб застосовується, наприклад, для адресації наступної команди, яка обчислюється додаванням деякого числа I до адреси команди, що виконується (I – це довжина команди в мінімально адресованих елементах інформації в пам'яті). Довжина команди у більшості сучасних процесорів – число змінне, яке залежить від типу команди.

Аналіз наведеної програми показує, що часто необхідно як один з операндів застосовувати результат попередньої операції. Тому як адреса результату переважно використовується адреса одного з операндів і результат заміщує один з операндів.

Таблиця 5.1

КОП	Назва операції
1	Додавання
2	Віднімання
3	Множення
4	Ділення
5	Зупинка

Всі адреси, коди операцій та інші числа для зручності запишемо в десятковій системі числення. Нехай необхідно обчислити значення такого арифметичного виразу: $Y = (A+B)/C$. Вважатимемо, що вихідні дані на момент запуску програми були записані у пам'ять комп'ютера, а кодування операцій – задане табл. 5.1.

Приклад програми для гіпотетичного комп'ютера наведено у табл. 5.2. Припускається, що операнди є цілими числами.

Програма (табл. 5.2) складається всього з трьох команд, розміщених у послідовних адресах 00, 01, 02.

Безпосередня адресація

У програмах дуже часто потрібно виконувати операції, в яких один з операндів є константою, що не змінюється у процесі роботи програми. Можна записати таку константу в комірку пам'яті й адресувати її звичайним способом, але краще для цієї мети використати безпосередню адресацію (операнд при цьому записується безпосередньо у команді замість однієї з адрес). Такий спосіб адресації прискорює виконання команди, тому що не потрібен цикл звернення до пам'яті за операндом (він вибраний разом з командою), і водночас економить пам'ять.

Таблиця 5.2

Адресація комірок пам'яті		Вміст комірок				Коментар
		КОП	Адресація операндів			
			1	2	3	
Програма	00	1	03	04	06	$Y = A + B$
	01	4	06	05	06	$Y = Y/C$
	02	5	–	–	–	СТОП
Дані	03					A
	04					B
	05					C
	06					Y

Пряма адресація

Здійснюється вказування в команді прямої адреси операнда, тобто номера комірки пам'яті, в якій він записаний. Це простий універсальний спосіб адресації, який широко використовувався у комп'ютерах I-го і II-го поколінь. У даний час він застосовується обмежено через такі труднощі:

- а) переміщення програм у пам'ять;
- б) опрацювання масивів даних і організація спеціальних видів пам'яті;
- в) передавання фактичних параметрів у підпрограми.

У всіх цих випадках необхідна модифікація чи формування адресних частин команд, що ускладнює програмування й самі програми.

Непряма адресація.

Адреса операнда, який зберігається в деякій комірці пам'яті, називається вказівником.

Адресація операнда через ланцюг вказівників – *непряма адресація*. Кількість вказівників у ланцюзі – це кратність непрямої адресації. Найбільшого поширення набула однократна непряма адресація, схему якої наведено на рис. 5.4. Значення вказівника, тобто адреса операнда, називається непрямою адресою. Адреса вказівника залишається постійною, а непряма адреса може змінюватись у процесі виконання програми командами самої програми. Цим забезпечується можливість переадресації даних, тобто опрацювання масивів операндів чи складової структури даних за допомогою однієї і тієї самої програми чи ділянки програми.

Непряма адресація спрощує опрацювання масивів та спискових структур даних, а також передавання параметрів підпрограмам, але не забезпечує переміщення програм у пам'яті.

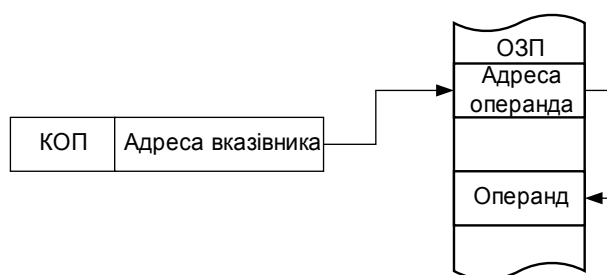


Рис. 5.4. Схема непрямої адресації

При багатократній непрямої адресації в команді вказується адреса першого вказівника і кратність – число вказівників, що послідовно вибираються. В останньому вказівнику розміщена адреса операнда. На практиці кратність, як правило, не перевищує 6–8. Однак при кратності більшій від 2 програмування значно ускладнюється.

Пряма адресація може розглядатися як частковий випадок непрямої адресації з кратністю 0.

Індексна адресація

Якщо як параметри використовуються змінні з індексами, то вони позначають елементи масивів, що є сукупністю однотипних значень. Найбільш зручною для роботи з масивами вважається індексна адресація, схему якої подано на рис. 5.5. При індексній адресації застосовуються індексні реєстри, розміщені у процесорі. Адреса операнда обчислюється як сума початкової адреси масиву операндів та індексу, записаного в одному з індексних реєстрів процесора. В команді відводиться додаткове поле, в якому вказується номер індексного реєстра X. В деяких процесорах передбачені реєстри для зберігання межі зміни індексу.

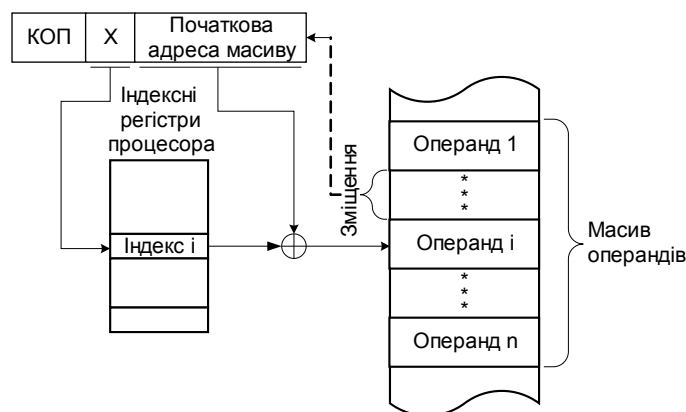


Рис. 5.5. Схема індексної адресації

Відносна адресація

Зміщенням називається програмний модуль, який однаково виконується процесором незалежно від адрес пам'яті, в яких він розміщений. Основна ідея відносної адресації полягає в тому, щоб у команді вказувати не абсолютну адресу операнда, а його зміщення відносно початку програмного модуля. Початкова адреса програмного модуля – базова адреса, що завантажується при вході в модуль у внутрішній реєстр процесора B – базовий реєстр. Схему відносної адресації наведено на рис. 5.6.

Для ефективного опрацювання масивів при відносній адресації застосовується комбінована індексно-відносна адресація, при якій адреса операнда A обчислюється як сума трьох величин:

$$A = (B)+(X)+D,$$

де (B), (X) – відповідно вміст базового й індексного реєстрів з номерами зміщення B, X, D.

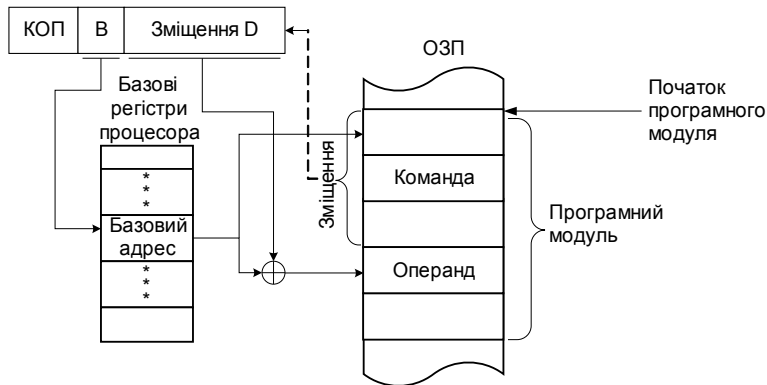


Рис. 5.6. Схема відносної адресації

У команді відводиться два поля для вказування номерів базового та індексного реєстрів і одне поле під зміщення.

Завантаження базової адреси у базовий реєстр, як правило, здійснюється із самого програмного модуля. Оскільки у процесорі вибірка команд відбувається за вмістом лічильника команд, то після отримання програмним модулем керування на лічильнику команд буде адреса першої команди модуля. Ця адреса й може бути використана як базова.

Як базовий реєстр можна застосувати також лічильник команд. У даному разі непотрібно відводити окремий реєстр для зберігання базової адреси і робити його завантаження.

Автоіндексна адресація

У міні- і мікрокомп'ютерах, де обладнання процесора обмежене, застосовується автоіндексна адресація, яка є розвитком непрямої адресації. При автоіндексній адресації непряма адреса збільшується або зменшується до чи після виконання операції з операндом. Цей спосіб адресації забезпечує високу ефективність обробки масивів при невеликих апаратних затратах на його реалізацію, особливо в разі зберігання непрямої адреси у внутрішньому реєстрі процесора. Якщо

непряма адреса збільшується, то адресація називається *автоінкрементною* або з *автозбільшенням*, якщо зменшується – то *автодекрементною* або з *автозменшенням*. Ці способи узгоджені так, щоб при послідовному застосуванні, наприклад автодекрементної й автоінкрементної адресації, використовувався один і той самий операнд. Це

потребує, щоб у випадку, який розглядається, при автодекрементній адресації спочатку відбувалося зменшення непрямої адреси, а потім операція; при автоінкрементній ж адресації, навпаки: спочатку здійснюється операція з операндом, а потім збільшення непрямої адреси.

Стекова пам'ять та її адресація

Стекова пам'ять належить до спеціальних видів пам'яті. При зверненні до неї адреса не використовується. Для читання/запису доступна тільки одна комірка, що називається *вершиною стека*. На рис. 5.7 показано стан стекової пам'яті при послідовному запису в стек значень *a* і *b* та читання їх зі стека. Як видно з рисунка, останній записаний елемент буде прочитаний першим. Подібний порядок запису/читання характерний для адрес повернення при вкладених викликах підпрограм, а також у багатьох інших випадках.

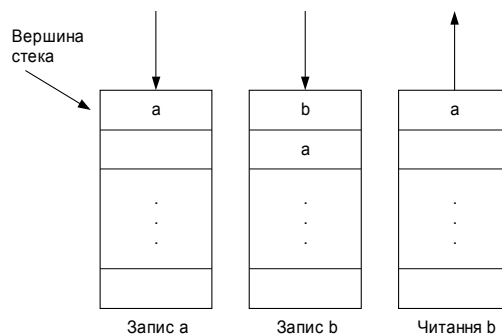


Рис. 5.7. Функціонування стекової пам'яті

Практично стекова пам'ять реалізується на основі звичайної пам'яті з використанням вказівника стека й автоіндексної адресації. Запис у стек здійснюється із застосуванням автодекрементної адресації, а читання – автоінкрементної.

Операції запису-читання зі стековою пам'яттю виконуються процесором неявно при викликах підпрограм і поверненнях із підпрограм, при обробці переривань та в інших випадках. У системі команд процесора, як правило, передбачаються операції пересилання значень зі стека і в стек та операції завантаження вказівника стека.

6. АРХІТЕКТУРА СУЧАСНИХ ПРОЦЕСОРІВ

6.1. Класифікація, архітектура і характеристика сучасних CISC-мікропроцесорів

Зміна елементної бази привела до зміни принципів попередніх поколінь проектування засобів обчислювальної техніки, таких як жорстка структура, послідовне центральне керування, лінійна організація пам'яті, відсутність можливості адаптації структури до особливостей завдання, яке виконується.

Альтернативою класичним принципам організації обчислювальних машин і систем, паралельної конвеєрної обробки інформації та використання табличних методів обробки даних і прийняття рішень є ідея створення адаптивних обчислювальних систем.

Успіхи напівпровідникової електроніки та поява завдяки їм ВІС значно підвищили ефективність цифрових систем – збільшили їхню надійність і продуктивність, зменшили габарити й масу та знизили вартість. При розробці пристроїв обчислювальної техніки почали широко використовувати стандартні універсальні ВІС, які отримали назву мікропроцесора (МП).

Мікропроцесор (МП) – програмнокерований пристрій, який проводить процес обробки інформації й керування ним, побудований переважно на одній або декількох ВІС.

Мікропроцесорна ВІС – інтегральна мікросхема, яка виконує функцію МП або його частини. Власне кажучи, це є ВІС з процесорною організацією, яку розроблено для побудови мікропроцесорних систем.

Першим мікропроцесором, який з'явився на ринку в 1971 р., був МП фірми «Intel-I4004». Розроблений передусім для кишенькових калькуляторів, він міг виконувати ряд інших функцій, наприклад керування світлофорами, і започаткував створення цілого ряду та широкої номенклатури МП. У 1980 р. випуск і широке розповсюдження мікропроцесора I8008 стали поштовхом для створення промислового стандарту в галузі мікропроцесорної техніки. На сьогодні понад 80% комп'ютерів та комп'ютерних систем базується на мікропроцесорах фірми «Intel».

Основними виробниками мікропроцесорів є компанії «Intel», «Motorola», «AMD», «Hewlett-Packard», «Cyrrix», «NexGeir», «IBM» й ін.

Новатором на ринку виробників МП є фірма «Intel», тому слід більш детально розглянути архітектурні особливості виробництва Intel-МП.

I80286	16-розрядна шина даних і 24-розрядна шина адреси.
I80386 DX	32-розрядна шина даних і адреси.
I80386 SX	16-розрядна шина даних, 32-розрядна внутрішня шина даних і 24-розрядна шина адреси.
I80486 DX	32-розрядні шини даних і адреси, вбудований сопроцесор (процесор плаваючої арифметики); вбудована кеш-пам'ять 8Кбайт для команд і даних, наявність властивості RISC-архітектури.
I80486 SX	аналогічний до 486 DX, але без вбудованого сопроцесора.
I80486 SL	аналогічний до 486 DX, але зі зниженою напругою живлення (3,3 В).
I80486 DX2	МП з подвоєнням внутрішньої тактової частоти.
I80486 DX4	МП з потроєнням внутрішньої тактової частоти, 16 КВ кеш-пам'ять.
Pentium (I80586)	64-розрядна шина даних і 32-розрядна шина адреси, два розділених блоки кеш-пам'яті зі зворотним записом ємністю по 8 КВ для команд і даних, суперскалярна архітектура. З метою покращення характеристик засобів мультимедіа фірма «Intel» розробила мікропроцесор Pentium MMX з тактовими частотами 200 МГц і 233 МГц. Технологія MMX була розроблена як засіб підтримки додатків, які активно використовують ресурси мікропроцесора для обробки графіки, відео- й аудіоінформації. Використання спеціалізованих MMX-додатків, які реалізують нові інструкції, дає змогу, за оцінками спеціалістів фірми, підвищити продуктивність на 60% порівняно зі звичайними МП Pentium з аналогічною тактовою частотою.
I80686	До цього покоління МП (1995–1997 рр.) належать МП Pentium Pro (перше покоління сімейства Р6) і Pentium II (представник другого покоління сімейства Р6) фірми «Intel», а також К6 фірми «AMD» і 6x86 фірми «Cyrix». Головна перевага й унікальність Р6, що міститься в одному корпусі з процесором, – це вторинна статична кеш-пам'ять ємністю 256 Кбайт, з'єднана з процесором спеціально виділеною шиною, яка, по-перше, полегшує проектування високопродуктивних систем на базі Р6, а по-друге, підвищує продуктивність. Кеш другого рівня зв'язаний з процесором спеціально шиною шириною 64 біти і працює на тій тактовій частоті, що й процесор. Мікропроцесор Pentium II доповнений MMX-командами, він містить у два рази більшу кеш-пам'ять (16 Кбайт) порівняно з Pentium Pro (8 Кбайт) для команд і даних, взаємодіє з кеш-пам'яттю L2 не на своїй повній тактовій частоті (як Pentium Pro), а на вдвічі меншій.
I80786	Нове покоління МП (1999–2000 рр.) компанії «Intel» під назвою Pentium III, а також К7 фірми «AMD» та Celeron. Ці мікропроцесори забезпечують високий рівень продуктивності, досягаючи надшвидкодії завдяки використанню технології, яка відома як динамічне виконання, з можливістю багаторазового прогнозування переходів і попередньої вибірки даних. Pentium III порівняно з МП Pentium II працює на високих частотах 450–1130 МГц, а також може звертатися до 64 Гб фізичної і 64 Тб віртуальної пам'яті. МП Pentium III – трипотоковий суперскалярний мікропроцесор, що містить багатofункціональний конвеєр, який дає змогу на більш ранніх етапах перетворювати команди на коди мікрокоманд.
I80886	2001–2002 рр. – покоління сучасних найбільш продуктивних мікропроцесорів фірми «Intel» під назвою Pentium 4. Pentium 4 порівняно з Pentium III працює на надзвичайно високих тактових частотах 1,2–2,83 ГГц і може звертатися до 128 Гб фізичної пам'яті, має вдвічі більший обсяг кеш-пам'яті L2, тобто 512 Кб.

До боротьби за нове покоління процесорів x86 приєдналися компанії, що раніше займалися виготовленням Intel-сумісних процесорів. Це, зокрема, компанії «Advanced Micro Devices (AMD)», «Cyrix Corp.» і

«NexGen». З точки зору мікроархітектури найближчий до Pentium процесор M1 компанії «Cyrix». Так, як і Pentium, він має два конвеєри та може виконувати до двох команд в одному такті. Проте у процесорі M1 число випадків, коли операції можуть виконуватися попарно, значно збільшено. Крім цього, в ньому застосовується методика обходів і прискорення пересилання даних, що дає змогу усунути припинення конвеєрів у тих, з якими не справляється Pentium. Процесор містить 32 фізичних реєстри (замість 8 логічних, передбачених архітектурою x86) та використовує методику перейменування реєстрів для усунення залежності за даними. Як і Pentium, процесор M1 для прогнозування напряму переходу застосовує буфер цільових адрес переходу місткістю 256 елементів, до того ж він підтримує спеціальний стек повернень, що відстежує виклик процедур та подальші повернення.

Процесори K5 компанії «AMD» і Nx586 компанії «NexGen» використовують зовсім інший підхід. Основою їхніх процесорів є дуже швидке RISC-ядро, що виконує високорегулярні операції у суперскалярному режимі. Внутрішні формати команд (ROP у компанії «AMD» і RISC86 у компанії «NexGen») відповідають традиційним системам команд RISC-процесорів. Всі команди мають однакову довжину і кодуються в регулярному форматі. Звернення до пам'яті виконуються спеціальними командами завантаження й запису. Як відомо, архітектура x86 має дуже складну для декодування систему команд. У процесорах K5 та Nx586 здійснюється апаратна трансляція команд x86 у команди внутрішнього формату, що дає кращі можливості для розпаралелювання обчислень. У процесорі K5 вміщено 40 фізичних реєстрів, а у процесорі Nx586 – 22 фізичних реєстри, які реалізують методику перейменування. У процесорі K5 інформація, необхідна для прогнозування напрямку переходу, записується прямо в кеш команд і зберігається разом з кожним рядком кеш-пам'яті. В процесорі Nx586 для цих цілей використовується кеш-пам'ять адрес переходів на 96 елементів.

Фірма «Intel» уклала стратегічну угоду з компанією «Hewlett-Packard» про розробку наступного покоління мікропроцесорів, в яких архітектура x86 буде поєднуватися з архітектурою дуже довгого командного слова (VLIW-архітектурою).

Прикладом вітчизняного МП може служити мікропроцесор K1810BM86.

МП – центральна частина будь-якої електронної системи керування і обробки даних, але не сама система. Окремий МП не може виконувати яких-небудь корисних функцій. Ця обставина визначається тим, що для

отримання реальної користі від функціонування МП у нього необхідно вводити вихідні дані про поточний стан якого-небудь об'єкта (процесу) та використовувати результати обробки за наперед відпрацьованим (заданим) алгоритмом. Таким чином, електронна система з мікропроцесором (або на основі мікропроцесора) обов'язково має складатися з таких функціональних частин – підсистем: введення-виведення й обробка даних, накопичення і зберігання даних про особливості внутрішньої структури та функціонування об'єкта чи процесу.

Створення МПК ВІС привело до виникнення принципово нових напрямків у застосуванні цифрової ОТ і дало змогу здійснити вбудоване керування простим обладнанням та приладами, розподілене керування складним обладнанням і агрегатами, розподілені обчислення у багатопроцесорних системах.

Мікропроцесорний комплект (МПК) – сукупність мікропроцесорних та інших інтегральних мікросхем, сумісних за конструктивно-технологічним виконанням і призначених для спільного застосування при побудові МП, мікрокомп'ютерів та інших засобів обчислювальної техніки.

Прикладом МПК може бути мікропроцесорний комплект І80486, який містить такі ВІС:

- 80486 – швидкодіючий 32-розрядний мікропроцесор;
- 82596СА – 32-розрядний сопроцесор LAN;
- 82320 – контролер магістралі Micro Channel;
- 82350 – контролер магістралі EISA;
- 82С508 – мікросхема програмованої логіки, яка мінімізує обсяг обладнання основної плати.

Класифікація мікропроцесорів

Для опису МП як функціональних пристроїв необхідно охарактеризувати формати команд і даних, що обробляються, кількість, тип та гнучкість команд, методи адресації даних, число внутрішніх регістрів загального призначення і регістрів результату, можливості організації й адресації стека, параметри віртуальної пам'яті та інформаційну місткість пам'яті, що адресується безпосередньо (прямо).

Серед вітчизняних ВІС є три класи мікропроцесорних ВІС, які відрізняються структурою, технічними характеристиками і функціональними можливостями: багатокристалльні секціоновані МП з нарощуванням розрядності й мікропрограмним керуванням; однокристалльні МП та однокристалльні мікрокомп'ютери з фіксованою розрядністю і системою команд.

Секціоновані МП – це комплекти (МПК), які допускають нарощування параметрів (перш за все, розрядності) та функціональних можливостей. Структура і система команд пристроїв та систем, які проектуються на їхній основі, визначаються при розробці відповідно до конкретного призначення. Секціоновані МПК орієнтовані переважно на застосування в універсальних і спеціалізованих комп'ютерах, контролерах та інших засобах обчислювальної техніки високої продуктивності.

Мікропроцесорні комплекти на основі однокристальних МП й однокристальних мікрокомп'ютерів, які мають меншу продуктивність, але гнучку систему команд та більші функціональні можливості, орієнтовані на широке застосування в різних галузях народного господарства.

За призначенням розрізняють універсальні і спеціалізовані МП.

Універсальні МП використовуються для розв'язання широкого кола різноманітних задач. При цьому їхня ефективна продуктивність слабо залежить від проблемної специфіки задач.

Серед спеціалізованих МП слід виділити: різні мікроконтролери, орієнтовані на виконання складних послідовностей логічних операцій; математичні МП, призначені для підвищення продуктивності при виконанні арифметичних операцій. За допомогою спеціалізованих МП можна ефективно вирішувати складні задачі паралельної обробки даних.

За видом вхідних сигналів, що обробляються, розрізняють *цифрові* й *аналогові* мікропроцесори. Самі мікропроцесори – це цифрові пристрої, але вони можуть мати вбудовані аналого-цифрові і цифро-аналогові перетворювачі. Тому вхідні аналогові сигнали передаються також МП через перетворювач у цифровій формі, обробляються й після зворотного перетворення в аналогову форму надходять на вихід. З архітектурної точки зору такі мікропроцесори представляють собою аналогові функціональні перетворювачі сигналів і називаються аналоговими мікропроцесорами.

За характером часової організації роботи мікропроцесори поділяють на синхронні й асинхронні.

Синхронні мікропроцесори – мікропроцесори, в яких початок і кінець виконання операцій задаються пристроєм керування.

Асинхронні мікропроцесори – мікропроцесори, які дають змогу початок виконання кожної наступної операції визначити за сигналом фактичного закінчення виконання попередньої операції. Для більш ефективного використання кожного пристрою МП-системи до складу асинхронно працюючих пристроїв входять електронні вітки, які забезпечують автономне функціонування пристроїв.

За кількістю програм, що виконуються, розрізняють одно- і багато-програмні мікропроцесори.

В однопрограмних мікропроцесорах виконується тільки одна програма. Перехід до виконання іншої програми відбувається після завершення поточної програми.

У багато- або мультипрограмних мікропроцесорах одночасно виконується декілька програм.

6.2. Мікропроцесори з RISC-архітектурою

Останнім часом одним з важливих напрямків розвитку структури й архітектури обчислювальних систем є створення систем, що базуються на використанні обмеженого набору команд, більша частина яких може виконуватися за один машинний цикл. Концепцію RISC, що диктується намаганням спростити проектування, скоротити число команд, форматів і режимів адресації, а також підвищити ефективність та швидкодію блока керування, почали розробляти у середині 70-х рр. XX ст., а перший комп'ютер IBM 801 на її основі був виготовлений у 1982 р.

Мікропроцесори з архітектурою RISC (Reduced Instruction Computers) використовують порівняно невеликий (скорочений) набір команд, який визначається в результаті статистичного аналізу великого числа програм для основних сфер застосування CISC-процесорів.

Всі команди працюють з операндами, розміщеними в регістрах процесора, і мають однаковий формат. Звернення до пам'яті виконується за допомогою спеціальних команд завантаження регістру та запису.

Простота структури і невеликий набір команд дають змогу повністю реалізувати їхнє апаратне виконання та ефективний конвеєр при порівняно невеликому обсязі обладнання. Тому RISC-процесори в 2–4 рази є більш швидкодіючими порівняно з CISC-процесорами при однаковій тактовій частоті.

Основними особливостями RISC-систем є необхідність великого числа регістрів (більше, ніж 100), максимальне використання цих регістрів і мінімальне число звернень до оперативної пам'яті й оптимізація конвеєрного режиму виконання команд.

Існуючі типи систем з RISC-архітектурою спеціалісти характеризують з допомогою восьми основних ознак:

- відносно невелике число команд (бажано < 100, але допускається 150 і більше);
- невелике число режимів адресації (бажано один-два, допускається 4);
- невелике число форматів команд (один-два, допускається 4);

- виконання всіх команд за один машинний цикл;
- доступ до ЗП тільки за командою завантаження-зберігання;
- відносно великий регістровий файл ЦП (>32 регістрів), виконання всіх операцій за типом регістр-регістр;
- жорстко комутований блок керування;
- програмне керування з використанням операцій мовою високого рівня.

Одним з факторів, які вплинули на розробку архітектури, був розвиток технології HBIC і можливість завдяки цьому значно спростити блок керування.

Починаючи з 1987 р. і до сьогодні на ринку RISC-МП з'явилася велика кількість моделей МП різних фірм виробників МП-техніки.

Серед відомих виробників слід назвати фірми «IBM», «Intel» (МП 80860) та «Motorola» (МП 88000).

RISC-мікропроцесор 80860 фірми «Intel»

Вперше RISC-МП був представлений у 1989 р. на Міжнародній конференції IEEE. Є три основних шляхи, де доцільно застосовувати RISC-МП i860:

а) декілька мікропроцесорів i860 можуть працювати паралельно за умови внутрішньо-машинної синхронізації. Така конфігурація буде представляти собою міні-суперком'ютер, який працює у середовищі операційної системи UNIX четвертої версії;

б) мікросхема i860 – це ідеальний процесор, який можна використувати як ядро високопродуктивної графічної робочої станції;

в) комбінація мікросхеми i860 й апаратних засобів ПК, побудованого на МП 80386, є настільним комп'ютером з продуктивністю потужного комп'ютера.

За останніх декілька років на ринку МП з'явилося декілька RISC-процесорів. Одним з найбільш популярних став набір мікросхем SPARC, який використовується в робочій станції Sun-4 і новій робочій станції SPARC Station фірми «Sun». Друге місце за популярністю займає MIPS-процесор, що є центральним процесором для персональної графічної станції фірми «IRIS». Третій конкурент – це мікропроцесор 88000 фірми «Motorola», який застосовується в робочих станціях фірми «Orux», «Everex» та «Data General». Четверту позицію посідає RISC-процесор, який є власною розробкою фірми «IBM» і використовується у персональних комп'ютерах IBM RT.

Архітектура процесора i860

Мікросхема i860 (рис. 6.1) вміщує на одному кристалі декілька компонентів. Її основними функціональними елементами є:

- центральний RISC-процесор;
- процесор з плаваючою крапкою (ППК);
- графічний процесор, призначений для побудови тривимірних зображень;
- кеш-пам'ять, а також блоки керування кеш-пам'яттю, системним ОЗП, шиною і т. ін.

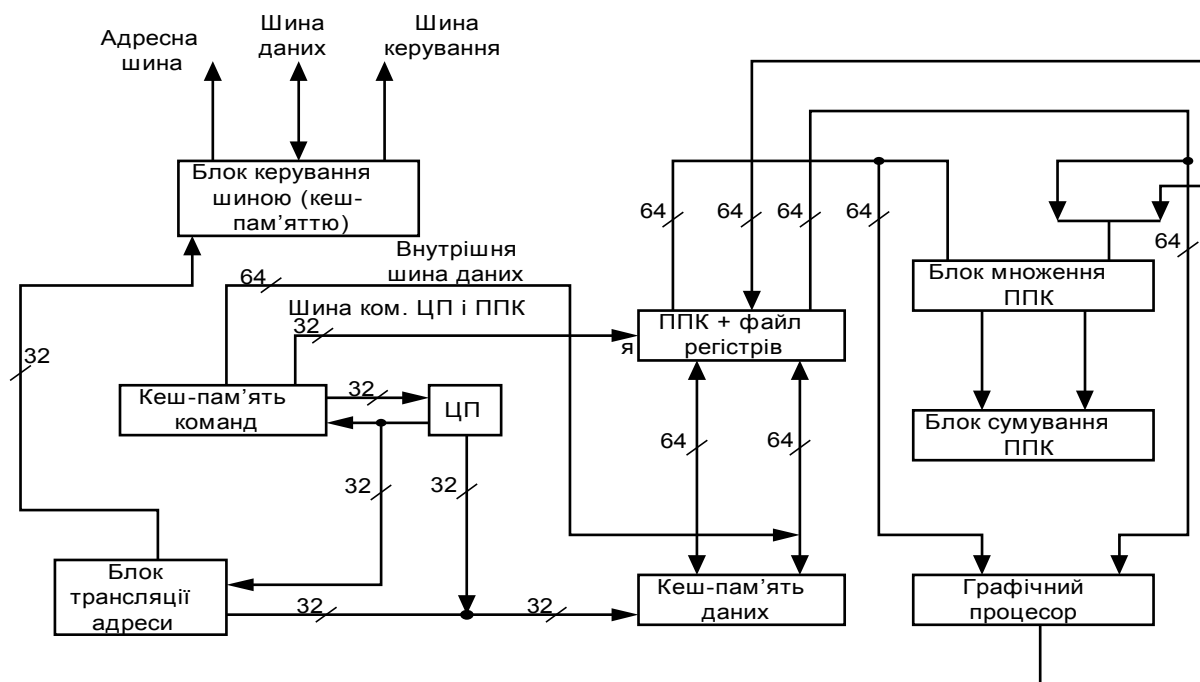


Рис. 6.1. Архітектура МП i860

Мікросхема має 64-розрядну шину даних, 32-розрядну адресну шину, яка дає змогу адресувати 4 Гб пам'яті. 32-розрядні логічні адреси, як і в мікропроцесорі I80386, транслюються у фізичні адреси блоком трансляції адреси.

Пам'ять має сторінкову організацію, причому розмір сторінки дорівнює 4 Кб. Блок керування кеш-пам'яттю контролює виконання трансляції адрес з метою забезпечення послідовного доступу до даних та команд. Розмір сторінок і методи трансляції сумісні з принципами організації віртуальної пам'яті у процесорах I80386, завдяки чому i860 може бути об'єднаний у мультипроцесорну систему з I80386/I80486 та спільно користуватися системним ОЗП.

Доступ до ОЗП відбувається завдяки двом вбудованим блокам кеш-пам'яті: один блок об'ємом 4 Кб призначений для команд, а інший блок об'ємом 8 Кб – для швидкодіючого оперативного зберігання команд.

За замовчуванням в i860 застосовується стратегія резервування кеш-пам'яті, при якій дані, що вже містяться у внутрішній кеш-пам'яті, не дублюються в основний ОЗП доти, доки нема потреби повторно використовувати блок внутрішньої кеш-пам'яті.

Дані, прийняті по внутрішній шині, можуть бути завантажені у два набори внутрішніх регістрів, які (за термінологією фірми «Intel») називаються «файлами». Файл регістрів цілих чисел є складовою частиною центрального RISC-процесора і складається з тридцяти двох 32-розрядних регістрів. Регістри мають нумерацію від R0 до R31, деякі з них зарезервовані для спеціального використання.

Файл регістрів з плаваючою крапкою вміщує тридцять два 32-розрядних регістри від F0 до F31. Доступ до файла може відбуватися двома способами: як до шістнадцяти 64-розрядних регістрів або як до восьми 128-розрядних регістрів.

ППК складається з трьох окремих блоків: графічного блока, блока сумування і блока множення. У кожному блоці на кожну команду припадає у середньому від трьох до чотирьох тактів. Блоки сумування й множення з плаваючою крапкою можуть бути використані для виконання векторних операцій.

Графічний процесор є спеціальним апаратним засобом тривимірної графіки та забезпечує виконання таких функцій, як z-буферизація або затінення Гуро і Фонга. В z-буферах зберігається інформація про третю координату кожної точки зображення, завдяки чому є можливість високошвидкісної побудови тривимірних зображень.

Структурну схему двопроцесорної системи, побудованої на основі RISC-МП i860, наведено на рис. 6.2.

У кінці 1991 р. фірми «Apple», «IBM» та «Motorola» вирішують спільно працювати над розробкою RISC-мікропроцесора, який вони назвали PowerPC. Це одна зі спроб використання RISC-процесорів у персональних комп'ютерах масового застосування. У 1987 р. англійська фірма «Acorn» виготовила комп'ютер Archimedes з RISC-процесором ARM (Acorn RISC Machine), розробленим спільно з фірмою «VLSI Technologies». Процесор був 32-розрядним і працював спільно з трьома сопроцесорами, які керували введенням-виведенням, пам'яттю та відео- й аудіоадаптерами. Тоді вперше RISC-технологія показала свою ефективність на ПК і дала змогу Archimedes з тактовою частотою 8 Гц випередити за швидкістю та продуктивністю найсильніші на той час ПК – Compaq Desk Pro 386, IBM PS/2 і Mac II незважаючи на те, що останні мали тактову частоту 16 МГц.

Архітектура PowerPC

Процесор PowerPC базується на архітектурі IBM POWER RISC, яка застосовувалась у робочих станціях PC/6000 фірми «IBM». В 1992 р. було виготовлено першу мікросхему RISC-МП PowerPC 601, структурну схему якого наведено на рис. 6.3. PowerPC – перше сімейство МП, здатне реально зменшити панування фірми «Intel» у сфері настільних ПК.

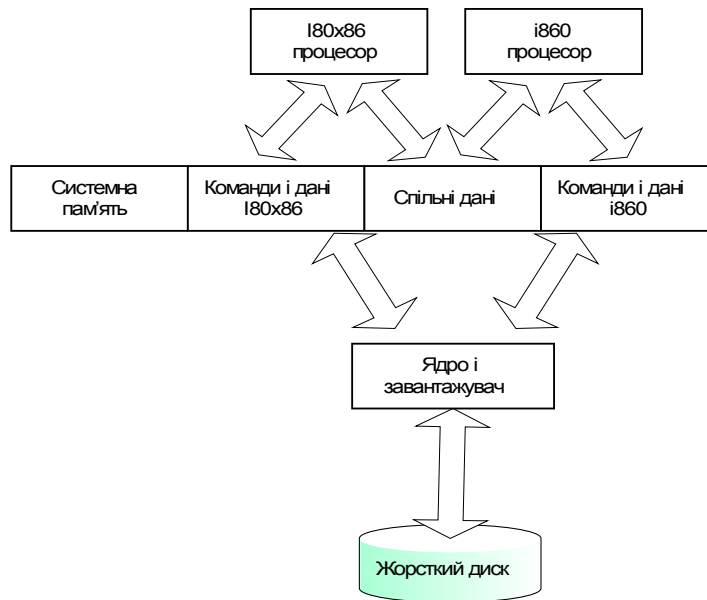


Рис. 6.2. Типова конфігурація складної двопроцесорної RISC-системи

RISC і CISC – дві найбільш популярні архітектури мікропроцесорів, які є багато в чому протилежними. RISC (Reduce Instruction Set Command) – архітектура зі зменшеним (скороченим) набором команд, CISC (Complex Instruction Set Command) – архітектура з великим (повним) набором складних команд. Характерною для RISC-процесора рисою є великий набір реєстрів, що дає змогу програмам зберігати в них велику кількість змінних. 32-розрядні процесори PPC 601, PPC 603 і PPC 604 (фірма «IBM») містять тридцять два 32-бітових реєстри загального призначення (РЗП, general-purpose registers) і тридцять два 64-бітових реєстри для операндів з плаваючою крапкою.

Мікропроцесор PowerPC (рис. 6.3) має близько 220 різних команд – цього достатньо (на перший погляд) для того, щоб зарахувати його швидше до CISC, ніж до RISC-процесорів. Особливо такі інструкції, як Branch and Link (виклик підпрограми), Load/Store Multiple і Load/Store String, дуже нагадують CISC-процесори великих комп'ютерів фірми «IBM». Проте такі невластиві RISC-процесорам команди існують через те, що одне з основних завдань розробників PowerPC полягало у зменшенні загальної

кількості команд у програмі, адже однією відносно складною командою можна виконати операцію, яка потребує серії простих команд.

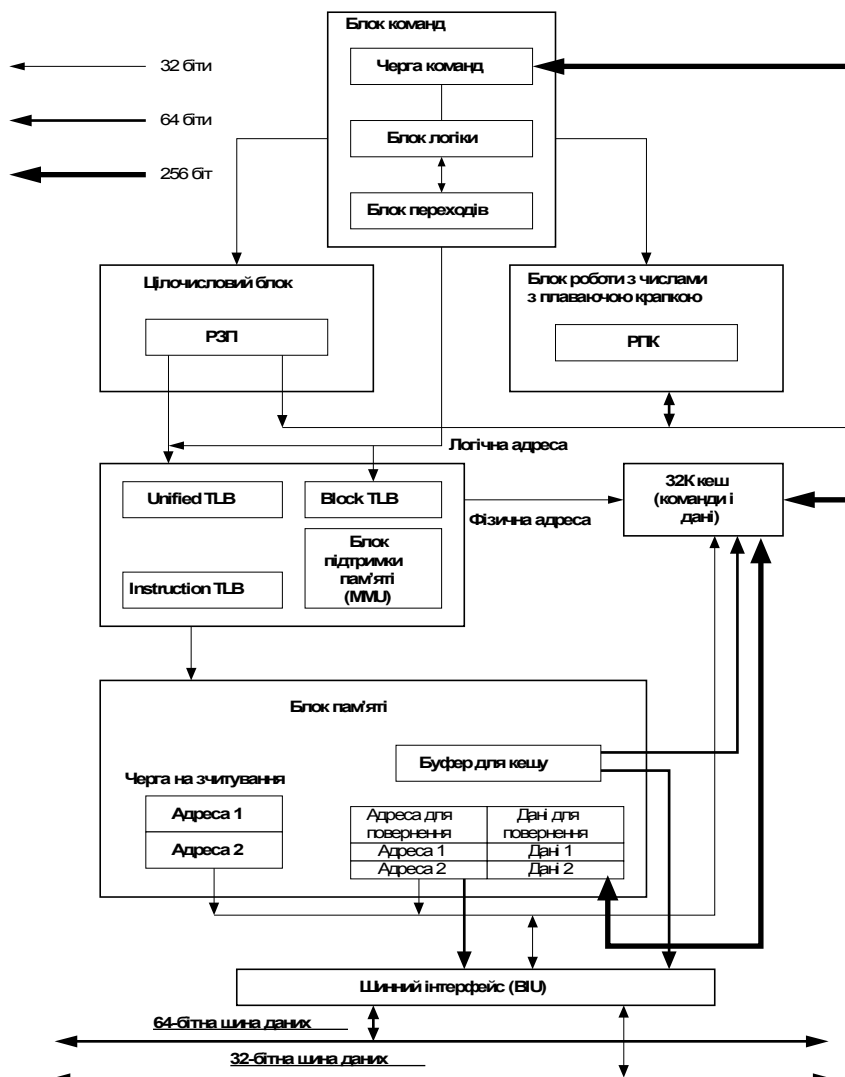


Рис. 6.3. Внутрішня організація PowerPC 601

У всіх інших відносинах PowerPC чітко дотримується традиційної RISC-філософії. Він має високу тактову частоту і затрачає малу кількість тактів на виконання однієї команди. Майже всі команди типу «load/store» працюють зі значеннями в регістрах. Доступ до пам'яті здійснюється окремими командами. Всі інструкції PowerPC є «зашитими» (hardwired), тобто реалізованими апаратними засобами. На противагу цьому багато CISC-процесорів обробляють складні команди в мікрокоді (невеликі програми низького рівня, що виконуються всередині процесора). Мікрокод зменшує продуктивність, потребує більшої логіки перетворень і збільшує розміри процесора.

Типовою рисою RISC-архітектури є застосування конвеєрів (pipeline): команда проходить кілька каскадів обробки. Як тільки над командою завершується один каскад, вона відразу переходить до наступного, а на цей каскад надходить команда, що йде за нею. Як правило, конвеєр містить у собі чотири каскади: витяг команди (instruction fetching), декодування команди (decoding), виконання команди (execution) і повернення її назад у потік команд (writeback).

Процесор з одним конвеєром називається скалярним, а з декількома – суперскалярним. Додаткові конвеєри збільшують продуктивність процесора.

Мікропроцесор PowerPC 601 може виконувати до трьох команд одночасно на трьох незалежних виконавчих блоках (див. рис. 6.3), у кожному з яких реалізовано конвеєр: цілочисловий блок (integer unit), блок переходів (branch unit) і блок роботи з числами з плаваючою крапкою (floating-point unit).

Окремий блок команд (instruction unit) керує надходженням інструкцій у виконавчі блоки. Блок команд витягує машинні інструкції з внутрішнього кешу, по 256-розрядній шині пересилає їх у чергу команд, а звідти спрямовує у відповідний виконавчий блок.

Для отримання максимального результату зі суперскалярної архітектури PPC 601 злегка перевпорядковує потік команд – техніка, що має назву «непослідовне виконання коду» (out-of-order execution). Черга команд процесора розрахована на 8 команд. Після того, як команди проходять каскад виконання у відповідному конвеєрі, вони повертаються в кеш у звичайному порядку.

Блок роботи з даними з плаваючою крапкою (БПК) підтримує IEEE-стандарт і працює з даними одинарної й подвійної точності. Через складність команд роботи з плаваючою крапкою конвеєр БПК має два каскади виконання, щоб полегшити обслуговування основної черги. Крім

цього, він містить дві додаткові внутрішні черги команд, щоб полегшити обслуговування основної черги.

PowerPC 601 здатний прогнозувати переходи, що дає змогу передбачати поведінку програми й уникати затримок, пов'язаних з великою кількістю переходів. Коли перехід прогнозований правильно, процесор не витрачає на нього ні одного такту, тобто виконує перехід практично миттєво. Однак неправильне передбачення розгалуження може викликати суттєву затримку: занесені в чергу і ті, які містяться у конвеєрі, команди мають бути вивантажені, а нові необхідно завантажити у перший каскад конвеєра. Прогнозування переходів – доволі ефективний спосіб збільшення продуктивності процесора. У більшості програм кожна сьома команда – перехід.

Безумовні переходи обробляються процесором технічно на належному рівні, тому що вони виконуються завжди. Блок переходів дає вказівку блоку команд починати завантаження у чергу команд з адреси, на яку має здійснитися перехід (у цей час ще виконуються команди з попередньої порції). Цей прийом відомий під назвою «згортка переходів». Власне інструкція переходу взагалі не заноситься у потік команд, таким чином не забираючи час для виконання.

Набагато складніше обробляються умовні переходи. У процесі виконання програми виявляються з'ясовані (resolved) і нез'ясовані (unresolved) переходи. Зі з'ясованими переходами (коли процесор заздалегідь «впевнений», що перехід здійсниться) відбувається те саме, що й з безумовними. Для нез'ясованих переходів PowerPC 601 застосовує статичну систему прогнозування переходів, яка ґрунтується на такому правилі: вважається, що якщо перехід має здійснитися назад, то він, швидше за все, відбудеться, а якщо вперед – то, швидше за все, не відбудеться.

Блок підтримки пам'яті (MMU) призначений для підтримки роботи операційної системи з віртуальною пам'яттю. Він виконує перетворення віртуальних адрес на реальні, забезпечує захист пам'яті й розподіл привілеїв доступу. PPC 601 може обслуговувати до 4 Мб фізичної пам'яті, а MMU – до 4000 Тб віртуальної пам'яті (Pentium – 64 Тб).

MMU виконує перетворення адрес на рівні сегмента, сторінки і блока. Останнє використовується при роботі з лінійними масивами даних великих розмірів (наприклад, з графічними зображеннями), тому що такий тип перетворень адрес значно знижує ефективність роботи системи з віртуальною пам'яттю через надмірну кількість запитів на підкачування сторінок.

PowerPC підтримує два види привілеїв доступу до пам'яті: рівень супервізора, що використовується операційною системою, і рівень користувача, який застосовується прикладними програмами.

Шинний інтерфейс містить 64-розрядну шину даних та 32-розрядну адресну шину. Всі пересилання відбуваються з перевіркою на парність. Можуть виконуватися два шинних цикли одночасно з використанням адресного конвеєра.

Дві нові версії процесора PowerPC 602e і PowerPC 602 (фірми «IBM Corp» та «Motorola Inc») створені на основі МП другого покоління PowerPC 603. Нові процесори демонструють гнучкість базової архітектури.

PowerPC 603 відомий як вищий рівень розвитку архітектури PowerPC – процесора PPC 601. Крім підвищеної тактової частоти 80 МГц, у процесорі PowerPC 603 реалізовано концепцію множинної диспетчеризації – множинного виконання інструкцій, що у поєднанні з пророкуванням розгалу-жень забезпечує виконання більшої кількості інструкцій за один такт і значно підвищує продуктивність процесора.

Однак як компенсацію за збільшення площі кристала й енергоспоживання суперскалярного ядра в моделі PowerPC603 був істотно зменшений обсяг первинної кеш-пам'яті – до 8 Кб окремо для інструкцій і даних. Цей компроміс виявився неприйнятним для «Apple», яка у своїй стратегії переведення користувачів із сімейства Macintosh на Power Mac дуже сильно залежить від швидкості виконання ПЗ емуляції мікропроцесорів 68000. Код емулятора не вмщався у маленькій кеш-пам'яті процесора PowerPC 603.

Центр спільної розробки IBM/Motorola Somerset Design Center вирішив цю проблему в PowerPC 603e. У мікропроцесорі PowerPC 603e об'єм кеш-пам'яті становить по 16 Кб для коду і для даних, а також у ньому використано складнішу організацію: на відміну від старої двопотокової кеш-пам'яті у процесорі PowerPC 603, нова кеш-пам'ять забезпечує асоціативне чотирипотокове кешування.

У PowerPC 603e використовується те саме ядро, що й у PowerPC 603, але його критично важливі елементи після доробки дають змогу працювати з тактовою частотою 100 МГц. Підвищена частота потребує більш гнучкого тактового генератора шини, тому зовнішня шина PowerPC 603e, що обслуговує вторинну кеш-пам'ять, а також введення-виведення і контролер ОЗП, може працювати з тактовою частотою, не кратною внутрішній частоті процесора.

На замовлення розробників комунікаційного обладнання та засобів відтворення мультимедіа був створений МП PowerPC 602 (модифікація

моделі PowerPC 603). Структурну схему RISC мікропроцесора PowerPC 602 подано на рис. 6.4.

У МП PowerPC 602 зменшено кеш-пам'ять до 4 Кб і спрощено логіку, а також залишено структуру двопотокової асоціативної кеш-пам'яті зі зворотним записом, що є характерним для мікропроцесора PowerPC 603.

Що стосується інших змін, то з модуля обчислень з плаваючою крапкою вилучено пристрій для обчислень з подвійною точністю та залишено тільки пристрій з одинарною точністю. Буфер транслятора скорочений з 64 до 32 інструкцій. Головна відмінність цюколівки нового процесора – це подвоєна 64-розрядна шина даних при 32-розрядній шині адреси, що дало змогу зменшити кількість ніжок з 240 до 144.

Модифіковано також пристрій керування пам'яттю, що дає змогу програмам працювати в захищеному режимі. Особливий інтерес викликає модифікація операційних блоків. В апаратних модулях цілочислових перемножувачів реалізовано режим ранньої зупинки рахування, що дає змогу із затримкою у два такти почати виконання кожного циклу побайтового множення. Модуль для операцій плаваючої арифметики, втративши блок подвійної точності, зберіг перемножувач-акумулятор, а блок завантаження-зберігання інструкцій був спрощений і прискорений.

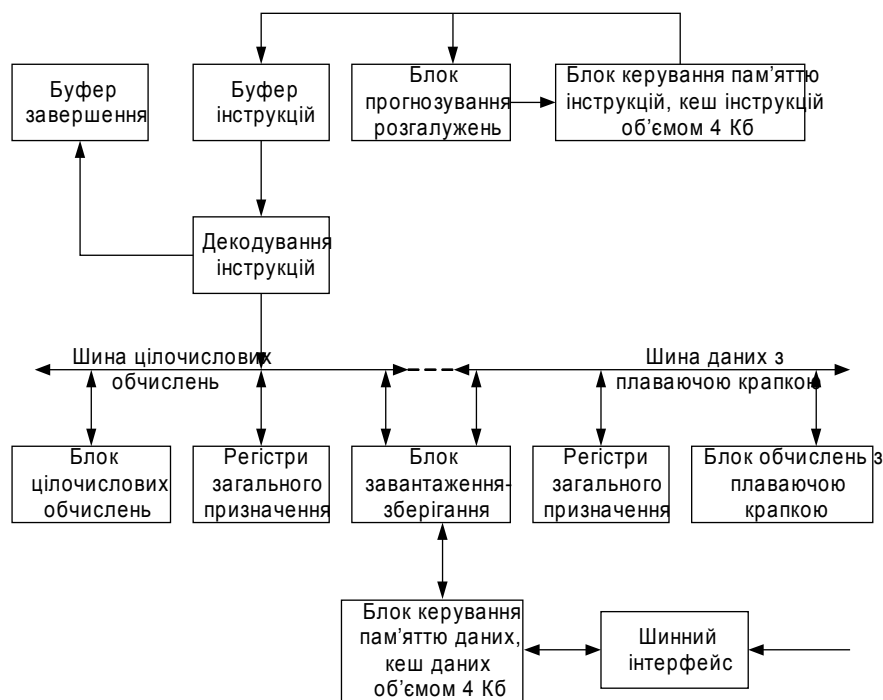


Рис. 6.4. Структура RISC – МП PowerPC 602

Суперскалярний процесор PowerPC 604 забезпечує одночасну видачу до чотирьох команд. При цьому паралельно в кожному такті може завершуватися виконання до шести команд. Процесор охоплює шість виконавчих пристроїв, які можуть працювати паралельно:

- пристрій плаваючої крапки (FPU);
- пристрій виконання переходів (BPU);
- пристрій завантаження-запису (LSU);
- три цілочислових пристрої (IU);
- два однотоктних цілочислових пристрої (SCIU);
- один багатотоктний цілочисловий пристрій (MCIU).

Така паралельна конструкція у поєднанні зі специфікацією команд PowerPC, що допускає реалізацію прискореного виконання команд, забезпечує високу ефективність і велику пропускну спроможність процесора. Вживані у процесорі PowerPC 604 буфери перейменування регістрів, буферні станції резервування, динамічне прогнозування напряму умовних переходів та пристрій завершення виконання команд істотно збільшують пропускну спроможність системи, гарантують завершення виконання команд у порядку, вказаному програмою, й забезпечують реалізацію моделі точного переривання.

У процесорі PowerPC 604 є окремі пристрої керування пам'яттю та окремі внутрішні кеші для команд і даних по 16 Кбайт. У ньому реалізовано два буфери перетворення віртуальних адрес та фізичні TLB (окремо для команд та для даних), що містять по 128 рядків. Обидва буфери є двоканальними множинно-асоціативними і забезпечують змінний розмір сторінок віртуальної пам'яті. Кеш-пам'яті й буфери TLB використовують для заміщення блоків алгоритму LRU.

Процесор PowerPC 604 має 64-бітову зовнішню шину даних та 32-бітову шину адреси. Інтерфейсний протокол процесора PowerPC 604 дає змогу декільком головним пристроям шини конкурувати щодо системних ресурсів за наявності централізованого зовнішнього арбітра. Крім цього, внутрішні логічні схеми спостереження за шиною підтримують когерентність кеш-пам'яті у мультипроцесорних конфігураціях. Процесор Power PC604 забезпечує як одиничні, так і групові пересилання даних при зверненні до основної пам'яті.

Архітектура PowerPC, що набула подальшого розвитку у вигляді нового процесора G4 або MPC7400 (мікропроцесор спроектований та виробляється компанією «Motorola»), цілком конкурентоздатна навіть у тих напрямках, де не використовуються переваги вбудованого

векторного модуля. RISC-процесор MPC7400 має 32-розрядну шину адреси і зовнішню 64-розрядну шину даних. До його складу входить кеш-пам'ять L1, яка є роздільною для інструкцій та для даних (кожна по 32 Кб), а також підтримується кеш L2 об'ємом до 2 Мб. MPC7400 має здатність функціонувати у складі симетричних мультипроцесорних систем. Для забезпечення зворотної сумісності з процесорами попереднього покоління цей МП може працювати з шиною 60х.

Векторний блок здатний оперувати з 128-розрядними операндами на основі архітектури SIMD, тобто виконувати паралельно до 16 операцій за один цикл. При цьому залежно від розрядності вектори можуть описуватися 4, 8 чи 16 координатами. Процесор містить 32 128-розрядних реєстри.

Блок-схема мікропроцесора MPC7400, яку подано на рис. 6.5, охоплює сім основних виконавчих модулів:

- два – для цілочислових обчислень;
- арифметики подвійної точності з плаваючою крапкою;
- векторний;
- завантаження і запису;
- системний;
- обробки розгалужень.

З їхньою допомогою суперскалярне ядро цього мікропроцесора типу PowerPC здатне виконувати три інструкції за такт одного переходу й два інші типи.

Мікропроцесори з архітектурою SPARC

Процесорна архітектура компанії «Sun Microsystems» (SPARC – Scalable Processor Architecture) є найбільш поширеною RISC-архітектурою, що відображає домінуюче положення компанії на ринку UNIX-робочих станцій і серверів.

Архітектуру SPARC було розроблено з метою спрощення реалізації 32-бітового процесора. Надалі відповідно до покращення технології виготовлення інтегральних схем вона поступово розвивалася та в даний час є 64-бітова версія цієї архітектури.

На відміну від більшості RISC-архітектур, SPARC використовує реєстрові вікна, які забезпечують зручний механізм передачі параметрів між програмами і повернення результатів. Архітектура SPARC стала першою комерційною розробкою, що реалізувала механізми відкладених переходів й ануляції команд. Це давало компілятору велику свободу для заповнення часу виконання команд

переходу командою, яка застосовується у разі виконання умов переходу та ігнорується у разі, якщо умова переходу не дотримується.

Перший процесор SPARC був виготовлений компанією «Fujitsu» на основі вентиляльної матриці, що працює на частоті 16,67 МГц. На основі цього процесора розроблено першу робочу станцію Sun-4 з продуктивністю 10 MIPS.

Мікропроцесор SuperSPARC

Подальшого збільшення продуктивності процесорів з архітектурою SPARC було досягнуто за рахунок реалізації у кристалах принципів суперскалярної обробки компаніями «Texas Instruments» і «Cypress». Процесор SuperSPARC компанії «Texas Instruments» став основою для серії робочих станцій та серверів SPARCstation/SPARCserver 10 і SPARCstation/SPARCserver 20. Є декілька версій цього процесора, що дає змогу залежно від суміші команд за один машинний такт обробляти до трьох команд, які відрізняються тактовою продуктивністю на операціях з фіксованою й плаваючою крапкою. Він має внутрішній кеш ємністю 36 Кб (20 Кб – кеш команд і 16 Кб – кеш даних), роздільні конвеєри цілочислової та плаваючої арифметики і при тактовій частоті 75 МГц забезпечує продуктивність близько 205 MIPS. Процесор SuperSPARC застосовується також у серверах SPARCserver 1000 і SPARCcenter 2000 компанії «Sun».

Конструктивно кристал вмонтовується на взаємозамінних процесорних модулях трьох типів, що відрізняються наявністю й об'ємом кеш-пам'яті другого рівня і тактовою частотою. Використання кеш-пам'яті дає змогу модулям CPU працювати з тактовою частотою, яка відрізняється від тактової частоти материнської плати; користувачі всіх моделей завдяки цьому можуть поліпшити продуктивність своїх систем заміною існуючих модулів CPU.

Компанія «Texas Instruments» розробила також 50 МГц процесор MicroSPARC із вбудованим кешем ємністю 6 Кб, який раніше широко використовувався в дешевих моделях робочих станцій SPARCclassic і SPARCstation LX, а в даний час застосовується лише в X-терміналах. «Sun» спільно з «Fujitsu» створили також нову версію кристала MicroSPARC II із вбудованим кешем ємністю 24 Кб. На його основі побудовані робочі станції та сервери SPARCstation/SPARCserver 4 і SPARCstation/SPARCserver 5, які працюють на частотах 70, 85 і 110 МГц.

Хоча архітектура SPARC залишається домінуючою на ринку процесорів RISC, особливо у секторі робочих станцій, підвищення

тактової частоти процесорів відбувалося більш повільними темпами порівняно з підвищенням тактової частоти конкуруючої архітектури процесорів. Щоб ліквідувати це відставання, а також у відповідь на появу на ринку 64-бітових процесорів компанія «Sun» розробила і впроваджує у життя програму модернізації. Відповідно до цієї програми «Sun» довела тактову частоту процесорів до 500 МГц (1998 р., процесори UltraSPARC-III).

Мікропроцесор hyperSPARC

Одним з головних завдань, які стояли перед розробниками мікропроцесора hyperSPARC, було підвищення продуктивності, особливо при виконанні операцій з плаваючою крапкою. Тому особливу увагу розробники приділяли створенню простих і збалансованих шестиступеневих конвеєрів цілочислової арифметики й плаваючої крапки. Логічні схеми цих конвеєрів ретельно розроблялися, кількість логічних рівнів вентилів між ступенями вирівнювалася, щоб спростити питання подальшого підвищення тактової частоти.

Продуктивність процесорів hyperSPARC може змінюватися незалежно від швидкості роботи зовнішньої шини (MBus). Набір кристалів hyperSPARC забезпечує як синхронні, так і асинхронні операції за допомогою спеціальної логіки кристала RT625. Відокремлення внутрішньої шини процесора від зовнішньої шини дає змогу збільшувати тактову частоту процесора незалежно від частоти роботи підсистем пам'яті і введення-виведення. Це забезпечує більш тривалий життєвий цикл, оскільки перехід на більш продуктивні модулі hyperSPARC не потребує переробки всієї системи.

Процесор hyperSPARC реалізовано у вигляді багатокристалльної мікрозбірки (рис. 6.6), до складу якої входять суперскалярна конвеєрна частина і тісно пов'язана з нею кеш-пам'ять другого рівня. До набору кристалів входять: RT620 (CPU) – центральний процесор, RT625 (CMTU) – контролер кеш-пам'яті, пристрій керування пам'яттю, пристрій тегів і чотири RT627 (CDU) кеш-пам'яті даних для реалізації кеш-пам'яті другого рівня ємністю 256 Кбайт. RT625 забезпечує також інтерфейс з шиною MBus.

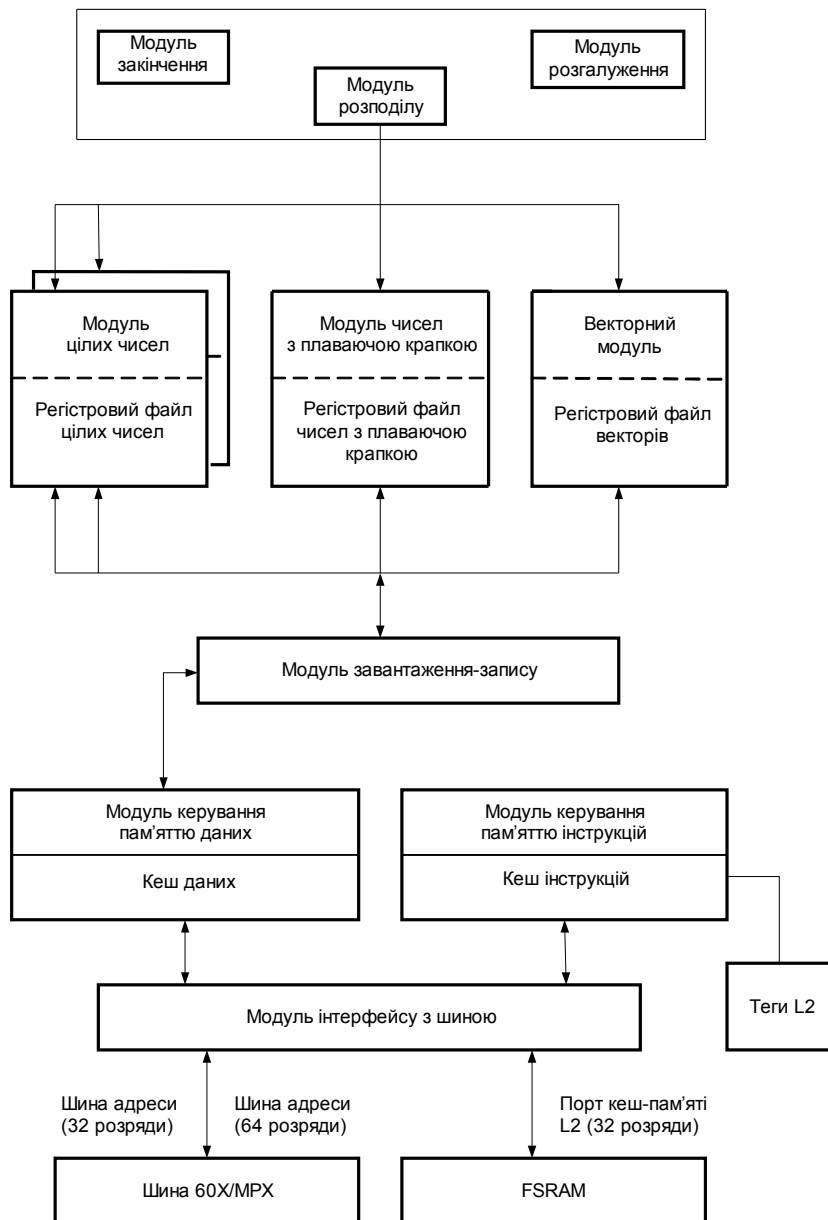


Рис. 6.5. Узагальнена структура процесора MPC7400

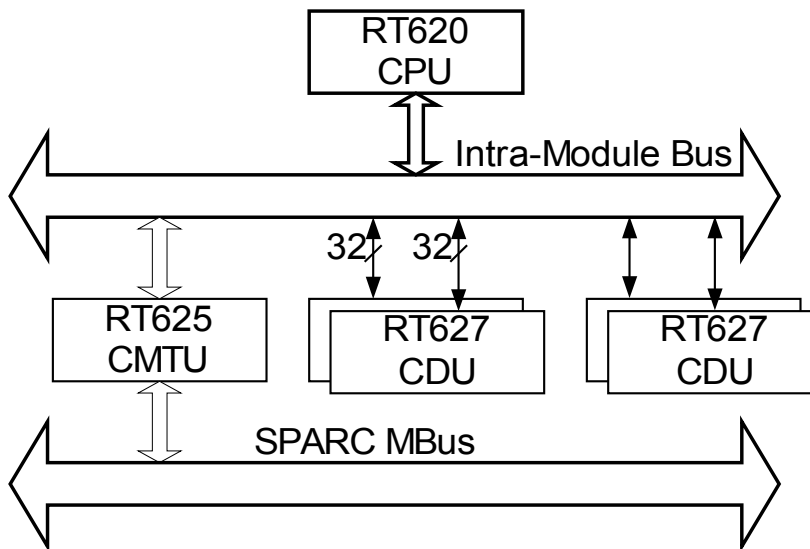


Рис. 6.6. Структура процесора hyperSPARC

Центральний процесор RT620 складається з цілочислового пристрою і пристрою з плаваючою крапкою, пристрою завантаження-запису, пристрою переходів та двоканальної множинно-асоціативної пам'яті команд ємністю 8 Кбайт. Цілочисловий пристрій вміщує АЛП і окремий тракт даних для операцій завантаження-запису, які є двома з чотирьох виконавчих пристроїв процесора. Пристрій переходів обробляє команди передачі керування, а пристрій плаваючої крапки реально складається з двох незалежних конвеєрів – сумування й множення чисел з плаваючою крапкою. Для збільшення пропускної спроможності процесора команди плаваючої крапки, проходячи через цілочисловий конвеєр, потрапляють у чергу, де вони чекають на запуск в одному з конвеєрів плаваючої крапки. В кожному такті вибираються дві команди. В загальному випадку доти, доки ці дві команди потребують для свого виконання різних виконавчих пристроїв за відсутності залежності за даними, вони можуть запускатися одночасно. RT620 містить два регістрових файли: 136 цілочислових регістрів, конфігурованих у вигляді восьми регістрових вікон, і 32 окремих регістри плаваючої крапки, розміщені у пристрої плаваючої крапки.

Кеш-пам'ять другого рівня у процесорі hyperSPARC будується на базі RT625 CMTU, який представляє собою комбінований кристал, що містить контролер кеш-пам'яті й пристрій керування пам'яттю, який підтримує зовнішню пам'ять, що розділяється, і симетричну

мультипроцесорну обробку. Контролер кеш-пам'яті підтримує кеш ємністю 256 Кбайт, що складається з чотирьох RT627 CDU. Кеш-пам'ять має пряме відображення і 4К тегів. Теги у кеш-пам'яті містять фізичні адреси, тому логічні схеми для дотримання когерентності кеш-пам'яті у багатопроцесорній системі, що є в RT625, можуть швидко визначити влучання або промахи при перегляді з боку зовнішньої шини без припинення звернень до кеш-пам'яті центрального процесора. Підтримується як режим наскрізного запису, так і режим зворотного копіювання.

Пристрій керування пам'яттю містить у своєму складі повністю асоціативну кеш-пам'ять перетворення віртуальних адрес на фізичні (TLB), що складається з 64 рядків, які підтримують 4096 контекстів. RT625 містить буфер читання ємністю 32 байти, який використовується для завантаження, і буфер запису ємністю 64 байти, що застосовується для розвантаження кеш-пам'яті другого рівня. Розмір рядка кеш-пам'яті становить 32 байти. Крім цього, в RT625 є логічні схеми синхронізації, які забезпечують інтерфейс між внутрішньою шиною процесора й SPARC MBus при виконанні асинхронних операцій.

RT627 представляє собою статичну пам'ять 16 Кбайт (2 Кбайт), спеціально розроблену для задоволення вимог hyperSPARC. Вона організована як чотириканальна статична пам'ять у вигляді чотирьох масивів з логікою побайтного запису і вхідними та вихідними регістрами-клямками. RT627 для ЦП є кеш-пам'яттю з нульовим станом очікування без втрат (тобто призупинок) на конвеєризацию для всіх операцій завантаження і запису, які потрапляють у кеш-пам'ять. RT627 був розроблений спеціально для процесора hyperSPARC, тому для з'єднання з RT620 й RT625 не потрібні ніякі додаткові схеми.

Набір кристалів дає змогу використовувати переваги щільного зв'язку процесора з кеш-пам'яттю. Конструкція RT620 допускає втрату одного такту в разі промаху в кеш-пам'яті першого рівня. Для доступу до кеш-пам'яті другого рівня в RT620 відведено спеціальний ступінь конвеєра. Якщо відбувається промах у кеш-пам'яті першого рівня, а у кеш-пам'яті другого рівня простежується влучання, то центральний процесор не зупиняється.

Команди завантаження і запису одночасно генерують два звернення: одне – до кеш-пам'яті команд першого рівня ємністю 8 Кбайт, а інше – до кеш-пам'яті другого рівня. Якщо адресу команди знайдено в кеш-пам'яті першого рівня, то звернення до кеш-пам'яті другого рівня відміняється і

команда стає доступною на стадії декодування конвеєра. Якщо ж у внутрішній кеш-пам'яті відбувся промах, а у кеш-пам'яті другого рівня знайдено влучання, то команда стане доступною з втратою одного такту, який вбудовано у конвеєр. Така можливість дає змогу конвеєру продовжувати безперервну роботу доти, доки простежуються влучання в кеш-пам'ять першого або другого рівня, які становлять відповідно 90% і 98% для типових прикладних задач робочої станції. З метою досягнення архітектурного балансу та спрощення обробки виняткових ситуацій цілочисловий конвеєр і конвеєр плаваючої крапки мають по п'ять стадій виконання операцій. Така конструкція допомагає RT620 забезпечити максимальну пропускну здатність, якої неможливо досягти іншим способом.

Мікропроцесори з архітектурою PA-RISC

Основою розробки сучасних виробів «Hewlett-Packard» є архітектура PA-RISC. Вона була розроблена компанією в 1986 р. і відтоді пройшла декілька стадій свого розвитку завдяки успіхам інтегральної технології від багатокристалного до однокристалного виконання. У вересні 1992 р. компанія «Hewlett-Packard» оголосила про створення свого суперскалярного процесора PA-7100, який відтоді став основою для побудови сімейства робочих станцій HP 9000 Series 700 і сімейства бізнес-серверів HP 9000 Series 800.

Мікропроцесор PA 7100

Особливістю архітектури PA-RISC є позакристална реалізація кешу, що дає змогу реалізувати різні обсяги кеш-пам'яті й оптимізувати конструкцію залежно від умов застосування (рис. 6.7). Зберігання команд і даних здійснюється в роздільних кешах, причому процесор з'єднується з ними за допомогою високошвидкісних 64-бітових шин. Кеш-пам'ять реалізується на високошвидкісних кристалах статичної пам'яті (SRAM), синхронізація яких здійснюється безпосередньо на тактовій частоті процесора. При тактовій частоті 100 МГц кожний кеш має смугу пропускання 800 Мбайт/с при виконанні операцій зчитування та 400 Мбайт/с при виконанні операцій запису. Мікропроцесор апаратно підтримує різний об'єм кеш-пам'яті: кеш команд може мати обсяг від 4 Кбайт до 1 Мбайт, кеш даних – від 4 Кбайт до 2 Мбайт. Щоб знизити коефіцієнт промахів, застосовується механізм кешування адреси. В обох кешах для підвищення надійності використовуються додаткові контрольні розряди, причому помилки кешу команд корегуються апаратними засобами.

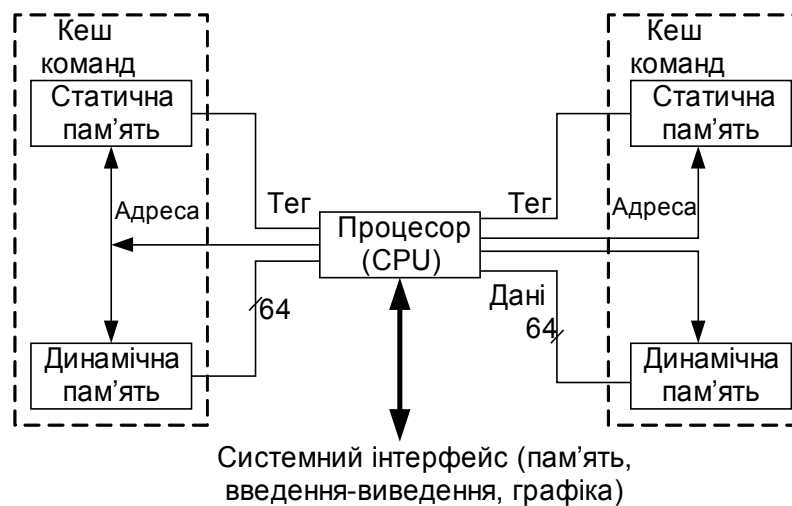


Рис. 6.7. Блок-схема процесора RA 7100

Процесор під'єднується до пам'яті і підсистеми введення-виведення за допомогою синхронної шини. Він може працювати з трьома різними відношеннями внутрішньої і зовнішньої тактової частоти залежно від частоти зовнішньої шини: 1:1, 3:2 і 2:1. Це дає змогу застосовувати у системах різні за швидкістю мікросхеми пам'яті.

Конструктивно на кристалі RA-7100 розміщено цілочисловий процесор, процесор для обробки чисел з плаваючою крапкою, пристрій керування кешем, уніфікований буфер TLB, пристрій керування і ряд інтерфейсних схем. Цілочисловий процесор вміщує АЛП, пристрій зсуву, суматор команд переходу, схеми перевірки кодів умов, схеми обходу, універсальний регістровий файл, регістри керування і регістри адресного конвеєра. Пристрій керування кеш-пам'яттю містить регістри, що забезпечують перезавантаження кеш-пам'яті при виникненні промахів та контроль когерентного стану пам'яті. Цей пристрій містить також адресні регістри сегментів, буфер перетворення адреси TLB й апаратуру хешування, що керує перезавантаженням TLB. До складу процесора плаваючої крапки входять: пристрій множення, арифметико-логічний пристрій, пристрій розподілу і добування квадратного кореня, регістровий файл та схеми «закорочування» результату. Інтерфейсні пристрої вміщують всі необхідні схеми для зв'язку з кеш-пам'яттю команд і даних, а також з шиною даних. Узагальнений буфер TLB містить 120 рядків асоціативної пам'яті фіксованого розміру і 16 рядків змінного розміру.

Пристрій плаваючої крапки (рис. 6.8) реалізує арифметику з одинарною й подвійною точністю у стандарті IEEE 754. Його пристрій множення використовується також для виконання операцій

цілочислового множення. Пристрої розподілу та обчислення квадратного кореня працюють з подвоєною частотою процесора. Арифметико-логічний пристрій виконує операції додавання, віднімання і перетворення форматів даних. Регістровий файл складається з двадцяти восьми 64-бітових регістрів, кожний з яких може використовуватися як два 32-бітових регістри для виконання операцій з плаваючою крапкою одинарної точності. Регістровий файл має п'ять портів зчитування і три порти запису, які забезпечують одночасне виконання операцій множення, додавання й завантаження-запису.

Більшість поліпшень продуктивності процесора пов'язана зі збільшенням тактової частоти до 100 МГц.

Конвеєр цілочислового пристрою охоплює шість ступенів: читання з кешу команд (IR), читання операндів (OR), виконання читання з кешу даних (DR), завершення читання кешу даних (DRC), запис у регістри (RW) і запис у кеш даних (DW). На ступені ID виконується вибірка команд. Реалізація механізму видачі двох команд потребує невеликого буфера попередньої вибірки, який забезпечує попередню вибірку команд за два такти до початку роботи ступеня IR. Під час виконання на ступені OR всі виконавчі пристрої декодують поля операндів у команді та починають обчислювати результат операції. На ступені DR цілочисловий пристрій завершує свою роботу. Крім цього, кеш-пам'ять даних виконує читання, але дані не надходять до моменту завершення роботи ступеня DRC. Результати операцій складання (ADD) і множення (MULTIPLY) також стають достовірними у кінці ступеня DRC. Запис в універсальні регістри й регістри плаваючої крапки виробляється на ступені RW. Запис у кеш даних командами запису (STORE) потребує двох тактів. Найбільш раннє двотактне вікно команди STORE виникає на ступенях RW і DW. Проте це вікно може зсуватися, оскільки записи у кеш даних відбуваються тільки тоді, коли з'являється наступна команда запису. Операції розподілу й обчислення квадратного кореня для чисел з плаваючою крапкою закінчуються на багато тактів пізніше за ступінь DW.

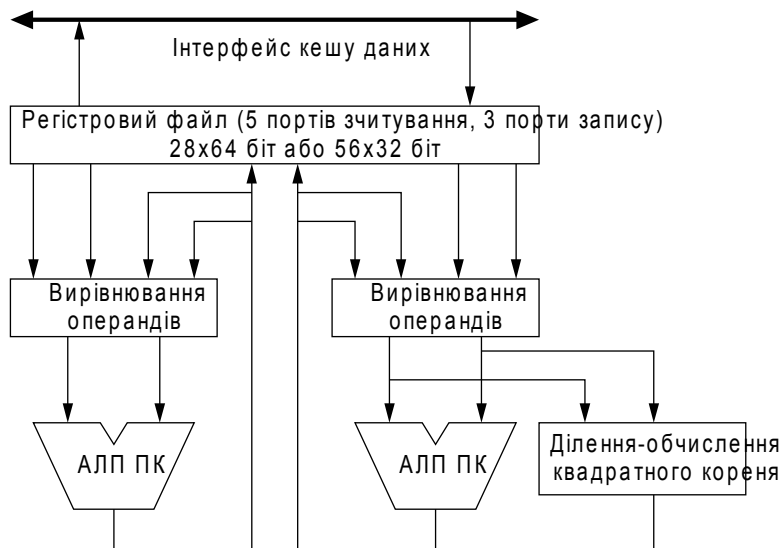


Рис. 6.8. Керування командами плаваючої крапки

Конвеєр проектувався з метою максимального збільшення часу, необхідного для виконання читання зовнішніх кристалів SRAM кеш-пам'яті даних. Це дає змогу максимізувати частоту процесора при заданій швидкості SRAM. Всі команди завантаження (LOAD) виконуються за один такт і потребують тільки одного такту смуги пропускання кеш-пам'яті даних. Оскільки кеші команд та даних розміщені на різних шинах, то в конвеєрі відсутні які-небудь втрати, пов'язані з конфліктами за звертаннями у кеш даних і кеш команд.

Процесор може в кожному такті видавати на виконання одну цілочислову команду й одну команду плаваючої крапки. Смуга пропускання кешу команд достатня для підтримки безперервної видачі двох команд у кожному такті. Відсутні які-небудь обмеження щодо вирівнювання або порядку проходження пари команд, що виконуються разом. Крім цього, відсутні втрати тактів, пов'язаних з переключенням при виконанні двох команд на виконання однієї команди.

Втрати, пов'язані із залежністю за даними й управлінням, в цьому конвеєрі мінімальні. Команди завантаження виконуються за один такт, за винятком випадку, коли подальша команда користується регістром-приймачем команди LOAD. Як правило, компілятор дає змогу обійти подібні втрати одного такту. Для зменшення втрат, пов'язаних з командами умовного переходу, в процесорі використовується алгоритм прогнозування напряму передачі управління. Для оптимізації продуктивності циклів і передачі управління вперед за програмою

прогнозуються як переходи, що не виконуються, а передачі управління назад за програмою – як переходи, що виконуються. Правильно спрогнозовані умовні переходи виконуються за один такт.

Кількість тактів, необхідних для запису слова або подвійного слова командою STORE, зменшено з трьох до двох тактів. Всі операції з плаваючою крапкою, за винятком команд розподілу й обчислення квадратного кореня, повністю конвеєризовані та мають двотактну затримку виконання як у режимі з одинарною, так і з подвійною точністю. Процесор може видавати на виконання незалежні команди з плаваючою крапкою в кожному такті за відсутності яких-небудь втрат. Послідовні операції із залежністю за регістрами призводять до втрати одного такту. Команди розподілу й обчислення квадратного кореня виконуються за 8 тактів при одинарній точності і за 15 тактів при подвійній точності.

Процесор може виконувати паралельно одну цілочислову команду та одну команду з плаваючою крапкою. При цьому цілочисловими командами вважаються команди і завантаження, і запису регістрів плаваючої крапки, а команди плаваючої крапки охоплюють команди FMPYADD та FMPYSUB. Ці останні команди об'єднують операцію множення з операціями додавання і віднімання, які виконуються паралельно.

Втрати для операцій плаваючої крапки, що використовують попереднє завантаження операнда командою LOAD, становлять один такт, якщо команди завантаження й плаваючої арифметики є суміжними, і два такти, якщо вони видаються для виконання одночасно. Для команди запису, що використовує результат операції з плаваючою крапкою, втрати відсутні, навіть якщо вони виконуються паралельно.

Втрати, що виникають при промахах у кеші даних, мінімізуються за допомогою застосування чотирьох різних методів: «влучання при промахи» для команд LOAD та STORE, потоковий режим роботи з кешем даних, спеціальне кодування команд запису, що допомагає уникнути копіювання рядка, в якому відбувся промах, і семафорні операції у кеш-пам'яті. Перша властивість дає змогу під час обробки промаху в кеші даних виконувати будь-які типи інших команд. Для промахів, що виникають при виконанні команди LOAD, обробка подальших команд може продовжуватися доти, доки регістр результату команди LOAD не буде потрібний як регістр операнда для іншої команди. Компілятор може використовувати цю властивість для попередньої вибірки у кеш необхідних даних задовго до того моменту, коли вони справді будуть потрібні. Для промахів, що виникають при виконанні команди STORE,

обробка подальших команд завантаження або операцій запису в частині одного слова продовжується доти, доки не виникають звернення до рядка, в якому відбувся промах. Компілятор може використовувати цю властивість для виконання команд на фоні запису результатів попередніх обчислень. Під час затримки, пов'язаної з обробкою промаху, інші команди LOAD і STORE, для яких відбувається влучання в кеш даних, можуть виконуватися так, як й інші команди цілочислової арифметики та плаваючої крапки. Протягом всього часу обробки промаху команди STORE інші команди запису в той самий рядок кеш-пам'яті можуть відбуватися без додаткових втрат часу. Для кожного слова в рядку кеш-пам'яті процесор має спеціальний індикаційний біт, що запобігає копіюванню з пам'яті тих слів рядка, які були записані командами STORE. Ця можливість застосовується до цілочислових і плаваючих операцій LOAD та STORE.

Виконання команд зупиняється, коли регістр-приймач команди LOAD, що виконується з промахом, стає потрібний як операнд іншої команди. Властивість «потокості» дає змогу продовжити виконання, як тільки потрібне або подвійне слово повертається з пам'яті. Таким чином, виконання команд може продовжуватися як під час затримки, пов'язаної з обробкою промаху, так і під час заповнення відповідного рядка при промаху.

При виконанні блокового копіювання даних у ряді випадків компілятор наперед знає, що запис має здійснюватися у повний рядок кеш-пам'яті. Для оптимізації обробки таких ситуацій архітектура PA-RISC 1.1 визначає спеціальне кодування команд запису («блокове копіювання»), який показує, що апаратурі не потрібно здійснювати вибірку з пам'яті рядка, при зверненні до якої може відбутися промах кеш-пам'яті. У цьому разі час звернення до кешу даних складається з часу, необхідного для копіювання у пам'ять старого рядка кеш-пам'яті за тією самою адресою у кеші (якщо він «брудний»), і часу, необхідного для запису нового тегу кешу. В процесорі PA 7100 таку можливість реалізовано як для привілейованих, так і для непривілейованих команд.

Останнє поліпшення управління кешем даних пов'язане з реалізацією семафорних операцій «завантаження з онуленням» безпосередньо у кеш-пам'яті. Якщо семафорна операція виконується у кеші, то втрати часу при її виконанні не перевищують втрат звичайних операцій запису. Це не тільки скорочує конвеєрні втрати, а й знижує трафік шини пам'яті. В архітектурі PA-RISC 1.1 передбачений також

інший тип спеціального кодування команд, який усуває вимогу синхронізації семафорних операцій з пристроями введення-виведення.

Керування кеш-пам'яттю команд дає змогу при промаху продовжити виконання команд відразу ж після надходження відсутньої у кеші команди з пам'яті. 64-бітова магістраль даних, які використовують для заповнення блоків кешу команд, відповідає максимальній смузі пропускання зовнішньої шини пам'яті 400 Мбайт/с при тактовій частоті 100 МГц.

У процесорі передбачений також ряд заходів з мінімізації втрат, пов'язаних з перетворенням віртуальних адрес на фізичні.

Конструкція процесора забезпечує реалізацію двох способів побудови багатопроцесорних систем. При першому способі кожен процесор під'єднується до інтерфейсного кристала, який спостерігає за всіма транзакціями на шині основної пам'яті. В такій системі всі функції щодо підтримки когерентного стану кеш-пам'яті покладені на інтерфейсний кристал, який посилає процесору відповідні транзакції. Кеш даних побудований на принципах відкладеного зворотного копіювання, і для кожного блока кеш-пам'яті підтримуються біти стану «приватний» (private), «брудний» (dirty) та «достовірний» (valid), значення яких змінюються відповідно до транзакцій, які видає або приймає процесор.

Другий спосіб організації багатопроцесорної системи дає змогу об'єднати два процесори і контролер пам'яті й введення-виведення на одній і тій самій локальній шині пам'яті. У такій конфігурації не виникає потреба в додаткових інтерфейсних кристалах, і вона сумісна з існуючою системою пам'яті. Когерентність кеш-пам'яті забезпечується спостереженням за локальною шиною пам'яті. Пересилання рядків між кешами виконуються без участі контролера пам'яті і введення-виведення. Така конфігурація забезпечує можливість побудови дуже дешевих високопродуктивних багатопроцесорних систем.

Процесор PA 7200

Процесор PA 7200 має ряд архітектурних удосконалень порівняно з PA 7100, головними з яких є додавання другого цілочислового конвеєра, побудова внутрішньокристального допоміжного кешу даних і реалізація нового 64-бітового інтерфейсу з шиною пам'яті.

Процесор PA 7200 забезпечує суперскалярний режим роботи з одночасною видачею до двох команд в одному такті. Всі команди процесора можна поділити на три групи: цілочислові операції, операції завантаження-запису й операції з плаваючою крапкою. PA 7200 здійснює одночасну видачу двох команд, які належать різним групам, або двох

цілочислових команд. Команди переходу виконуються у цілочисловому конвеєрі, причому ці переходи можуть утворювати пару для одночасної видачі на виконання тільки з попередньою командою.

Підвищення тактової частоти процесора потребує спрощення декодування команд на етапі видачі. З цією метою попереднє дешифрування потоку команд здійснюється ще на етапі завантаження кеш-пам'яті. Для кожного подвійного слова кеш-пам'ять команд охоплює 6 додаткових біт, які містять інформацію про наявність залежності за даними і конфліктів ресурсів, що істотно спрощує видачу команд у суперскалярному режимі.

Як і в PA 7100, у процесорі реалізовано інтерфейс із зовнішньою кеш-пам'яттю даних, що працює на тактовій частоті процесора з однотоктним часом очікування. Зовнішня кеш-пам'ять даних побудована за принципом прямого відображення. Крім цього, для підвищення ефективності на кристалі процесора реалізовано невеликий допоміжний кеш ємністю у 64 рядки. Формування, перетворення адреси і звернення до основної та допоміжної кеш-пам'яті даних виконуються на двох ступенях конвеєра. Максимальна затримка при виявленні влучання дорівнює одному такту.

Допоміжний внутрішній кеш містить 64 32-байтових рядки. При зверненні до кеш-пам'яті здійснюється перевірка 65 тегів: 64 тегів допоміжного кешу й одного тегу зовнішнього кешу даних. При виявленні збігу дані спрямовуються в необхідний функціональний пристрій.

За відсутності необхідного рядка у кеш-пам'яті виробляється її завантаження з основної пам'яті. При цьому рядок надходить у допоміжний кеш, що у деяких випадках допомагає скоротити кількість перезавантажень зовнішньої кеш-пам'яті, організованої за принципом прямого відображення. Архітектурою нового процесора для команд завантаження-запису передбачено кодування спеціальної ознаки локального розміщення даних (spatial locality only). При виконанні команд завантаження, позначених цією ознакою, відбувається звичайне заповнення рядка допоміжного кешу. Проте подальший запис рядка здійснюється безпосередньо в основну пам'ять, оминувши зовнішній кеш даних, що значно підвищує ефективність роботи з великими масивами даних, для яких розміру рядка кеш-пам'яті з прямим відображенням недостатньо.

Розширений набір команд процесора дає змогу реалізувати засоби автоіндексації для підвищення ефективності роботи з масивами, а також здійснювати попередню вибірку команд, які поміщаються в допоміжний внутрішній кеш. Цей допоміжний кеш забезпечує динамічне розширення

ступеня асоціативності основної кеш-пам'яті, побудованої на принципі прямого відображення, і є більш простим альтернативним рішенням порівняно з множинно-асоціативною організацією.

Процесор PA 7200 містить інтерфейс нової 64-бітової мультиплексної системної шини Runway, що реалізує розщеплення транзакцій і підтримку протоколу когерентності пам'яті. Цей інтерфейс охоплює буфери транзакцій, схеми арбітражу та схеми керування співвідношеннями зовнішніх і внутрішніх тактових частот.

Процесор MC88110 компанії «Motorola»

Процесор MC88110 належить до розряду суперскалярних RISC-процесорів. Головні особливості цього процесора пов'язані з використанням принципів суперскалярної обробки, 2 восьмипортних реєстрових файлів, 10 незалежних виконавчих пристроїв, великих за об'ємом внутрішніх кешів і широких магістралей даних.

Структурну схему процесора наведено на рис. 6.9. Центральною частиною цієї архітектури є шина операндів (у реалізації це шість 80-бітових шин), яка сполучає реєстрові файли й виконавчі пристрої.

Процесор має 10 виконавчих пристроїв, які працюють одночасно та незалежно, і 2 реєстрових файли. Файл реєстрів загального призначення має 32-бітову організацію. Розширені реєстри плаваючої крапки мають 80-бітову організацію. Ці реєстрові файли забезпечені 6 портами читання й 2 портами запису кожний.

Зовнішня шина процесора має окремі лінії даних (64 біт) і адреси (32 біт), що дає змогу реалізувати швидкі групові операції перезавантаження внутрішньої кеш-пам'яті. Зовнішня шина має також спеціальні сигнали керування, що забезпечують апаратну підтримку когерентності кеш-пам'яті у мультипроцесорних конфігураціях.

У процесорі є дві двоканальні множинно-асоціативні кеш-пам'яті об'ємом по 8 Кбайт (для команд і для даних). Всі операції з перезавантаження кеш-пам'яті виконуються в режимі групового пересилання даних, при цьому першим пересилається необхідне слово. Когерентність кешу даних забезпечується апаратним протоколом спостереження за шиною з чотирма станами (MESI). Для збільшення продуктивності у кеш-пам'яті даних застосовується стратегія затриманого зворотного копіювання.

Суперскалярна архітектура процесора базується на реалізації можливості завершення команд у порядку їхнього надходження для виконання, що дає змогу істотно збільшити продуктивність, але призводить до проблем організації точного переривання. Ця проблема

вирішується у процесорі MC88110 за допомогою так званого буфера історії, який збереже старі значення регістрів при виконанні й завершенні операцій не у вказаному порядку і дає змогу апаратно відновити необхідний стан у разі переривання.

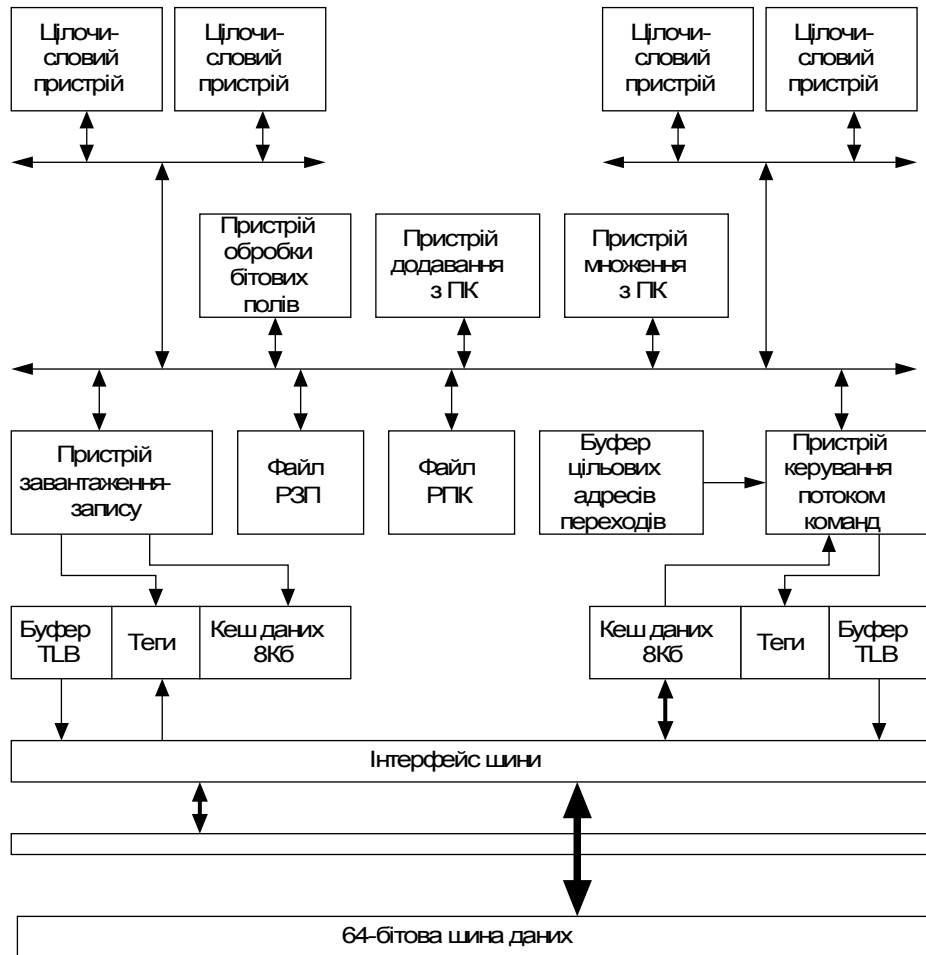


Рис. 6.9. Структура процесора MC 88110

У процесорі передбачено декілька способів прискорення обробки умовних переходів. Один з них – прогноз напрямку переходу – дає змогу компілятору повідомити процесору про пріоритетний напрям переходу. Для переходів, що виконуються, використовується буфер цільових адрес переходу ємністю у 32 рядки, який допомагає швидко вибрати дві команди за цільовою адресою переходу. Механізм прогнозу напрямку переходів дає змогу одночасно виконувати ці команди й оцінювати умову переходу. Для передбаченого напрямку переходу дозволено

спекулятивне (умовне) виконання команд. Якщо напрям переходу передбачено неправильно, то початковий стан процесора відновлюється за допомогою буфера історії. Виконання програми в цьому разі буде продовжено з «правильної» команди.

У кожному такті процесор може видавати на виконання дві команди. Здебільшого видача команд здійснюється у порядку, який вказано програмою. Команди запису й умовних переходів можуть посилатися на буферні станції резервування, з яких вони надалі будуть видані на виконання. Команди завантаження можуть нагромаджуватися у черзі. Таким чином ці команди не блокують видачу другої команди з пари. Велика кількість виконавчих пристроїв дає змогу здійснювати одночасну видачу двох команд у багатьох ситуаціях: 2 цілочислові команди, 2 команди з плаваючою крапкою, 2 графічні команди або будь-яка комбінація перерахованих команд.

У пристрої завантаження-запису реалізовано буфер завантаження FIFO на чотири рядки і три станції резервування операцій запису, що дає змогу мати у кожний момент часу до 4 відкладених команд завантаження та до 3 команд запису. Виконання цих команд усередині пристрою може перевпорядковуватися для забезпечення більшої ефективності.

При побудові мультипроцесорної системи всі процесори й основна пам'ять розміщуються на одній платі. Для забезпечення високої продуктивності системи кожний процесор у такій конфігурації забезпечується кеш-пам'яттю другого рівня ємністю 256 Кбайт. Протокол підтримки когерентного стану кеш-пам'яті (протокол спостереження) базується на методиці запису з ануляцією, що гарантує розміщення модифікованої копії рядка кеш-пам'яті тільки в одному з кешів системи. Протокол дає змогу декільком процесорам мати одну і ту саму копію рядка кеш-пам'яті. При цьому якщо один з процесорів виконує запис у пам'ять, то інші процесори повідомляються про те, що їхні копії є недійсними та їх необхідно анулювати.

6.3. Архітектура і характеристика 64-розрядних мікропроцесорів

У 90-х рр. ХХ ст. основні фірми-виробники мікропроцесорів розпочали розробку і випуск 64-розрядних універсальних мікропроцесорів, використовуючи різні архітектури. Так, фірма «Intel» працює над випуском 64-розрядних МП сімейства IA-64. Крім сімейства IA-64, ведеться розробка ще декількох універсальних МП за VLIW-подібною архітектурою.

IA-64 аббревіатура від «Intel» 64-bit Architecture – 64-розрядна архітектура «Intel», яка втілює концепцію EPIC (Explicitly Parallel Instruction Computing – обчислення з явним паралелізмом команд). В IA-64

використовується новий 64-розрядний набір команд, який розробили фірми «Intel» і «Hewlett-Packard».

Концепція EPIC, відповідно до «Intel», і «Hewlett – Packard», має переваги VLIW, але вона позбавлена недоліків останньої. Основними особливостями EPIC є:

- велика кількість реєстрів (128 64-розрядних реєстрів загального призначення (виконання операцій з фіксованою крапкою), 128 80-розрядних реєстрів плаваючої арифметики, 64 1-розрядних предикатних реєстри);

- масштабованість архітектури до великої кількості функціональних пристроїв;

- явний паралелізм у машинному коді. Пошук залежностей між командами робить не процесор, а компілятор;

- предикація (Predication). Команди належать до різних гілок умовного розгалуження;

- постачаються предикатними полями (полями умов) і запускаються паралельно;

- завантаження за припущенням (Speculative loading). Дані з повільної основної пам'яті завантажуються заздалегідь. Цей механізм призначений знизити простої процесора, пов'язані з чеканням виконання команд завантаження з повільної основної пам'яті.

Компілятор переміщує команди завантаження даних з пам'яті так, щоб вони виконувалися якомога раніше. Отже, коли дані з пам'яті знадобляться якійсь команді, то процесор не буде простоювати. Переміщені у такий спосіб команди називаються командами завантаження за припущенням і позначаються особливим чином. Безпосередньо перед командою, що використовує завантажені за припущенням дані, компілятор вставить команду перевірки припущення.

Предикація – засіб обробки умовних розгалужень. Суть цього засобу полягає в тому, що компілятор вказує, що обидві гілки виконуються на процесорі паралельно (процесори згідно з концепцією EPIC мають містити багато функціональних пристроїв).

Якщо у вихідній програмі трапляється умовне розгалуження (за статистикою, це буває через кожні 6 команд), то команди з різних гілок позначаються різними предикатними реєстрами (команди мають для цього предикатні поля), далі вони виконуються спільно, але їхні результати не записуються доти, доки значення предикатних реєстрів не визначені. Коли обчислюється умова розгалуження, предикатний реєстр, який відповідає «правильній гілці», встановлюється в «1», а інший – в «0». Перед записом результатів процесор буде перевіряти предикатне

поле і записувати результати тільки тих команд, предикатне поле яких містить предикатний регістр, встановлений в «1».

На сьогодні 64-розрядні мікропроцесори (Merced, MIPS, DEC Alpha, PowerPC 620, Sun UltraSPARC, HP PA-RISC 2.0) випускаються декількома фірмами серійно.

Архітектура Alpha компанії «DEC»

Процесор 21064

Мікросхема 21064 є повністю 64-розрядним суперскалярним RISC-процесором в однокристальному виконанні, до складу якого входять пристрої цілочислової і плаваючої арифметики, а також кеш-пам'ять об'ємом 16 Кб. Він проектувався з урахуванням реалізації передових методів збільшення продуктивності, зокрема конвеєрної організації всіх функціональних пристроїв, одночасної видачі декількох команд для виконання, а також засобів організації симетричної багатопроцесорної обробки.

У процесорі є два регістрових файли по 32 64-бітових регістри: один – для цілих чисел, інший – для чисел з плаваючою крапкою. З метою забезпечення сумісності з архітектурою MIPS і VAX архітектура Alpha підтримує арифметику з одинарною та подвійною точністю як відповідно до стандарту IEEE 754, так і відповідно до внутрішнього для компанії стандарту арифметики VAX.

Модернізована версія процесора – це модель 21064A, що має на кристалі кеш-пам'ять подвоєного об'єму й працює на більшій тактовій частоті. У пізніших моделях 21066 і 21068 передбачена реалізація на кристалі шини PCI. Це істотно спрощує та здешевлює як проектування, так і виробництво комп'ютерів. Відмінна особливість моделі 21068 – це низька споживана потужність (близько 8 ват). Основне призначення названих вище двох нових моделей – персональні й одноплатні комп'ютери.

На рис. 6.10 наведено блок-схему мікропроцесора 21066. Основними компонентами цього процесора є: кеш-пам'ять команд, цілочисловий пристрій, пристрій плаваючої крапки, пристрій виконання команд завантаження-запису, кеш-пам'ять даних, а також контролер пам'яті і контролер введення-виведення.

У процесорі є два регістрових файли по 32 64-бітових регістри: один – для цілих чисел, інший – для чисел з плаваючою крапкою. З метою забезпечення сумісності з архітектурою MIPS і VAX архітектура Alpha підтримує арифметику з одинарною та подвійною точністю як відповідно до стандарту IEEE 754, так і відповідно до внутрішнього для компанії стандарту арифметики VAX.

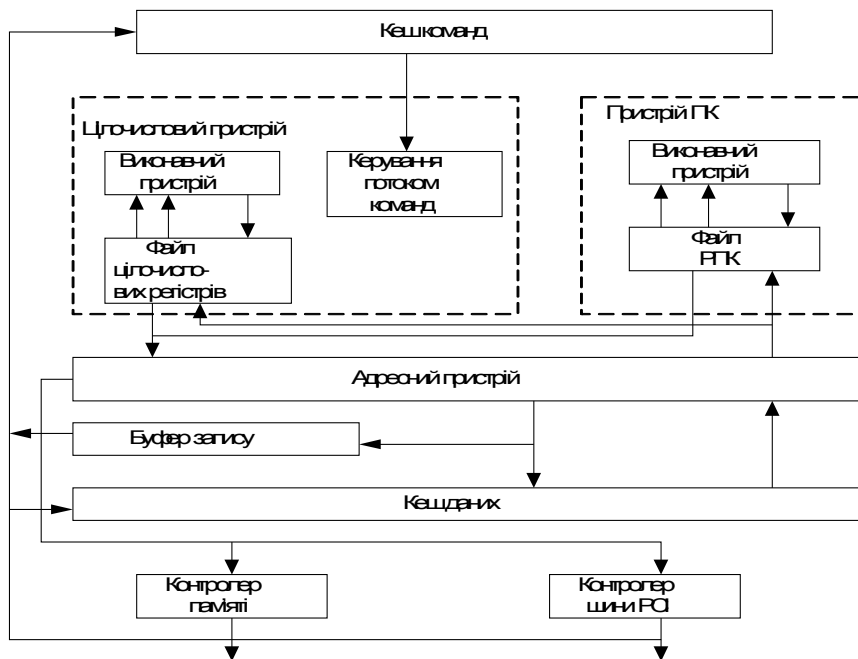


Рис. 6.10. Структурні компоненти процесора Alpha 21066

Процесор 21164

Мікропроцесор Alpha 21164 є другою повністю новою реалізацією архітектури Alpha.

Ключовими складовими для реалізації високої продуктивності вважають суперскалярний режим роботи процесора, що забезпечує видачу для виконання до чотирьох команд у кожному такті, високопродуктивну підсистему пам'яті з швидкодіюною кеш-пам'яттю першого рівня, велику, розміщену на кристалі кеш-пам'ять другого рівня і зменшену затримку виконання операцій у всіх функціональних пристроях.

На рис. 6.11 наведено структуру процесора, який містить п'ять функціональних пристроїв: пристрій керування потоком команд (IBOX), цілочисловий пристрій (EBOX), пристрій плаваючої крапки (FBOX), пристрій керування пам'яттю (MBOX) та пристрій керування кеш-пам'яттю й інтерфейсом шини (CBOX). На цьому рисунку також показано три розташованих на кристалі кеш-пам'яті. Кеш-пам'ять команд і кеш-пам'ять даних є первинними кешами, які реалізують пряме відображення. Множинно-асоціативна кеш-пам'ять другого рівня призначена для зберігання команд та даних.

Довжина конвеєрів процесора 21164 варіюється від 7 ступенів для виконання цілочислових команд і 9 ступенів для реалізації команд з плаваючою крапкою до 12 ступенів при виконанні команд звернення до

пам'яті в межах кристала й змінного числа ступенів при виконанні команд звернення до пам'яті за межами кристала.

Пристрій керування потоком команд здійснює вибірку і декодування команд з кешу команд та спрямовує їх для виконання у відповідні виконавчі пристрої. Пристрій керування містить 8 Кбайт кеш-команд, схеми попередньої вибірки команд і пов'язаний з ними буфер перезавантаження, схеми прогнозування напряму умовних переходів та буфер перетворення адрес команд (ІТВ).

Цілочисловий виконавчий пристрій виконує цілочислові команди, обчислює віртуальні адреси для всіх команд завантаження і запису, виконує цілочислові команди умовного переходу та всі інші команди керування. Він містить регістровий файл і декілька функціональних пристроїв, розташованих на чотирьох ступенях двох паралельних конвеєрів. Перший конвеєр вміщує суматор, пристрій логічних операцій, пристрій зсуву та помножувача, а другий – суматор, пристрій логічних операцій і пристрій виконання команд керування.

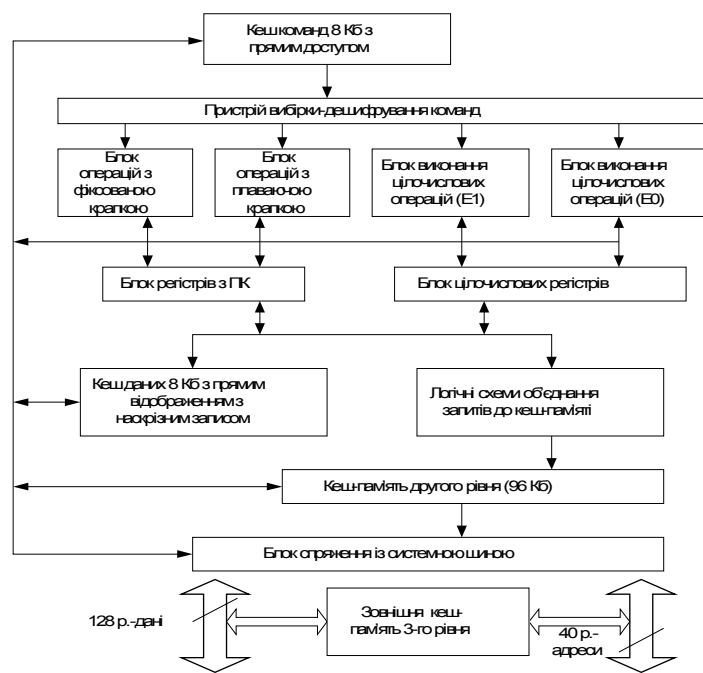


Рис. – 6.11. Блок-схема процесора Alpha 21164

Пристрій плаваючої крапки складається з двох конвеєрних виконавчих пристроїв: конвеєра сумування, який виконує всі команди плаваючої крапки, за винятком команд множення, та конвеєра множення, який виконує команди множення з плаваючою крапкою. Два спеціальних конвеєри завантаження й один конвеєр запису даних дають змогу командам завантаження-запису виконуватися паралельно з виконанням операцій з плаваючою крапкою. Апаратно підтримуються всі режими округлення, передбачені стандартами IEEE і VAX.

Пристрій керування пам'яттю виконує всі команди завантаження, запису й операції синхронізації. Він містить повністю асоціативний 64-рядковий буфер перетворення адрес (DTB), кеш-пам'ять даних 8 Кбайт з прямим відображенням, файл адрес промахів і буфер запису. Довжина рядка у кеші даних дорівнює 32 байт, він має два порти щодо читання та реалізований за принципом наскрізного запису. Він індексується розрядами фізичної адреси, і в тегах зберігаються фізичні адреси. У пристрій керування пам'яттю в кожному такті може надходити до двох віртуальних адрес цілочислового пристрою. DTB також має два порти, тому він може одночасно виконувати перетворення двох віртуальних адрес на фізичні. Команди завантаження звертаються до кешу даних і повертають результат у регістровий файл в разі влучання. При цьому затримка становить два такти. У разі промаху фізичні адреси спрямовуються у файл адрес промахів, де вони буферизуються та чекають на завершення звернення до кеш-пам'яті другого рівня. Команди запису записують дані у кеш даних в разі влучання і завжди поміщають дані у буфер запису, де вони чекають на звернення до кеш-пам'яті другого рівня.

Відмінною особливістю мікропроцесора 21164 є розміщення на кристалі повторного триканального множинно-асоціативного кешу ємністю 96 Кбайт. Повторний кеш різко знижує кількість звернень до зовнішньої шини мікропроцесора. Крім повторного кешу, на кристалі підтримується робота із зовнішнім кешем третього рівня.

Поєднання великої кількості обчислювальних пристроїв, швидшого виконання операцій з плаваючою крапкою (чотири такти замість шести) та швидшого доступу до первинного кешу (два такти замість трьох) за без-печують новому мікропроцесору рекордні параметри продуктивності.

Процесор 21264

Загальною особливістю процесора Alpha 21264 є наявність у його мікроархітектурі поняття кластера. У процесорі передбачено 2 кластери. Причина введення кластерів полягає в орієнтації на високі тактові

частоти (1ГГц і вище). У такому разі час поширення сигналу стає визначальною величиною та виникає потреба розміщати блоки процесора, які дуже часто обмінюються інформацією, поруч один з одним. Для реалізації цієї концепції розробники виділили по дві області (кластери), в середині яких логічні блоки процесора розташовані дуже близько один до одного. Кластери мікропроцесора містять по 1 копію файла цілочислових реєстрів і функціональних виконавчих пристроїв (ФП). У цілочислових реєстрах процесора Alpha 21264 є 8 портів читання (по 4 на кластер) та 6 портів запису.

В Alpha 21264 застосовується реалізоване у багатьох суперскалярних процесорах динамічне перейменування реєстрів.

Архітектура MIPS компанії «MIPS Technology»

Архітектура MIPS була однією з перших RISC-архітектур, які отримали визнання з боку промисловості. Спочатку це була повністю 32-бітова архітектура, яка охоплювала 32 реєстри загального призначення довжиною 32 біт, 16 реєстрів плаваючої крапки і спеціальну пару реєстрів для зберігання результатів виконання операцій цілочислового множення й розподілу. Розмір команд становив 32 біт, у ній підтримувався всього один метод адресації, а адресний простір також здійснювалося 32 бітами. Виконання арифметичних операцій визначався за стандартом IEEE 754. У комп'ютерній промисловості широкої популярності набули 32-бітові процесори R2000 і R3000, які протягом доволі тривалого періоду часу служили основою для побудови робочих станцій та серверів компаній «Silicon Graphics», «Digital», «Siemens Nixdorf» й ін. Процесори R3000/R3010 працювали на тактовій частоті 33 або 40 МГц.

На зміну мікропроцесорам сімейства R3000 прийшли нові 64-бітові мікропроцесори R4000 і R4400 («MIPS» стала першою у комп'ютерній промисловості компанією, яка випустила процесори з 64-бітовою архітектурою). Набір команд цих процесорів (специфікація MIPS II) був розширений командами завантаження й запису 64-розрядних чисел з плаваючою крапкою, командами обчислення квадратного кореня з одинарною та подвійною точністю, командами умовних переривань, а також атомарними операціями, необхідними для підтримки мультипроцесорних конфігурацій. У процесорах R4000 і R4400 реалізовано 64-бітові шини даних та 64-бітові реєстри, а також використано метод подвоєння внутрішньої тактової частоти.

Процесори R2000 і R3000 мали стандартні п'ятиступінчасті конвеєри команд. У процесорах R4000 і R4400 застосовуються довші

конвеєри (іноді їх називають суперконвеєрами). Кількість ступенів у процесорах R4000 і R4400 збільшилася до восьми, що пояснюється, перш за все, збільшенням тактової частоти й необхідністю розподілу логіки для забезпечення заданої пропускнуєї спроможності конвеєра. Процесор R4000 може працювати з тактовою частотою 50/100 МГц.

Процесори R4000 мали внутрішню кеш-пам'ять ємністю 16 Кбайт, розділену на 8-Кб кеш команд і 8-Кб кеш даних. З точки зору реалізації кеш-пам'яті процесор R4400 має більш розвинуті можливості. Він випускається у трьох модифікаціях: PC (Primary Cash) має внутрішні кеші команд та даних ємністю по 16 Кбайт. Процесор у такій конфігурації призначений переважно для дешевих моделей робочих станцій. SC (Secondary Cash) містить логіку управління кеш-пам'яттю другого рівня. MC (Multiprocessor Cash) використовує спеціальні алгоритми забезпечення когерентності й злагодженого стану пам'яті для багатопроцесорних конфігурацій.

Суперскалярний процесор R10000 (рис. 6.12) забезпечує пікову продуктивність у 800 MIPS при роботі з внутрішньою тактовою частотою 200 МГц за рахунок забезпечення видачі для виконання чотирьох команд в одному такті синхронізації. При цьому він забезпечує обмін даними з кеш-пам'яттю другого рівня зі швидкістю 3,2 Гбайт/с.

Щоб забезпечити такий високий рівень продуктивності у процесорі R10000, реалізовано деякі нові досягнення у сфері технології й архітектури процесорів.

Кеш-пам'ять даних першого рівня процесора R10000 має ємність 32 Кбайт і організована у вигляді двох однакових банків розміром по 16 Кбайт, що забезпечує двократне розшарування при виконанні звернень до цієї кеш-пам'яті. Кожний банк є двоканальною множинно-асоціативною кеш-пам'яттю з розміром рядка (блока) 32 байти. Кеш даних індексується за допомогою віртуальної адреси і зберігає теги фізичних адрес пам'яті. Такий метод індексації дає змогу вибрати підмножину кеш-пам'яті у тому самому такті, в якому формується віртуальна адреса. Проте для того, щоб підтримувати когерентність з кеш-пам'яттю другого рівня, у кеші першого рівня зберігаються теги фізичних адрес пам'яті.

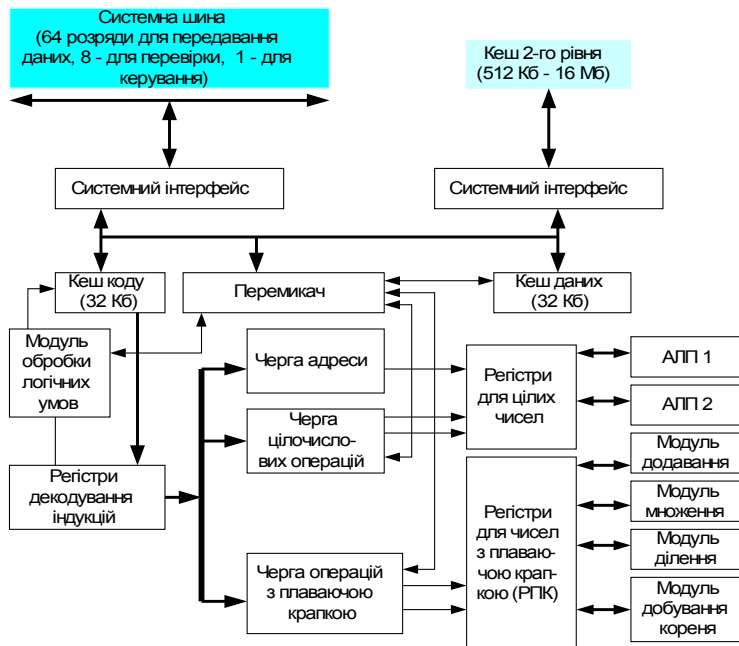


Рис. 6.12. Узагальнена структура процесора MIPS R10000

Інтерфейс кеш-пам'яті другого рівня процесора R10000 підтримує 128-бітову магістраль даних, яка може працювати з тактовою частотою 200 МГц, забезпечуючи швидкість обміну 3,2 Гбайт/с. Всі стандартні синхронні сигнали управління статичною пам'яттю виробляються всередині процесора. Мінімальний об'єм кеш-пам'яті другого рівня становить 512 Кбайт, а максимальний об'єм – 16 Мбайт. Розмір рядка цієї кеш-пам'яті програмується і може дорівнювати 64 або 128 байт.

Об'єм внутрішньої двоканальної множинно-асоціативної кеш-пам'яті команд становить 32 Кбайт. Команди частково декодуються до їхнього розміщення у кеші команд. При цьому до кожної команди додаються 4 додаткових біт, які вказують на виконавчий пристрій, в якому вона буде виконуватися. Розмір рядка кеш-пам'яті команд дорівнює 64 байти.

Пристрій переходів процесора може декодувати й виконувати одну команду переходу в кожному такті. Оскільки за кожною командою переходу йде слот затримки, максимально можна вибрати одночасно дві команди переходу, але тільки одна більш рання команда переходу може декодуватися в даний момент часу. Біт ознаки переходу додається до кожної команди під час декодування команд. Ці біти використовуються для позначення команд переходу в конвеєрі вибірки команд. Напрямо умовного переходу прогнозується за допомогою спеціальної пам'яті, яка

зберігає історію виконання переходів у минулому. Двобітовий код у цій пам'яті обновлюється кожного разу, коли прийнято остаточне рішення про напрям переходу. Всі команди, вибрані вслід за командою умовного переходу, вважаються умовними (спекулятивними). Це означає, що у момент їхньої вибірки наперед невідомо, чи буде завершено їхнє виконання. Процесор допускає попереднє прогнозування напряму чотирьох команд умовного переходу, які можуть дозволятися в довільному порядку. Спеціальний стек переходів містить рядок про те, що виконується кожна спекулятивна команда умовного переходу. Кожний рядок цього стека містить інформацію, необхідну для відновлення стану процесора, якщо спекулятивні команди переходу були передбачені неправильно. Стек переходів дає змогу швидко й ефективно відновити конвеєр, якщо прогноз напряму переходу виявився неправильним.

Процесор R10000 містить три черги (буфери) команд (черга цілочислових команд, черга команд плаваючої крапки і черга адресних команд). Ці три черги здійснюють видачу команд в динаміці у відповідні виконавчі пристрої. З кожною командою в черзі зберігається тег команди, який переміщається разом з командою по ступенях конвеєра. Кожна черга здійснює динамічне планування потоку команд і може визначити моменти часу, коли стають доступними операнди, необхідні для кожної команди. Крім цього, черга визначає порядок виконання команд на основі аналізу стану відповідних виконавчих пристроїв. Як тільки ресурс виявляється вільним, черга видає команду у відповідний виконавчий пристрій.

Залежність між командами може призвести до деградації продуктивності процесора. Щоб цього уникнути, застосовується спеціальна методика, яка називається методикою перейменування регістрів. Її основне завдання – визначення залежності між командами і забезпечення точної адреси переривання програми. У процесі перейменування регістрів кожний логічний регістр, вказаний у команді, замінюється фізичним регістром на основі таблиці розподілу регістрів. Таке перейменування відбувається для кожного регістра результату команди. Тому коли команда записує в логічний регістр нове значення, цей логічний регістр перейменовується й буде використовувати ім'я нового фізичного регістра. Проте його попереднє значення виявляється збереженим у старому фізичному регістрі. Збереження значень старого регістра дає змогу обробляти точні переривання. Тоді, коли всі команди перейменовуються, логічні номери їхніх регістрів порівнюються для визначення залежності між чотирма командами, декодованими в одному і тому самому такті.

У процесорі R10000 є п'ять повністю незалежних виконавчих пристроїв: два цілочислових АЛП, два основних пристрої плаваючої крапки й два повторних пристрої плаваючої крапки, які працюють з довгими операціями, такими як розподіл та обчислення квадратного кореня.

Пристрій завантаження-запису містить чергу адрес, пристрій обчислення адреси, пристрій перетворення віртуальних адрес на фізичні (TLB), стек адрес, буфер запису і кеш-пам'ять даних першого рівня. Пристрій завантаження-запису виконує команди завантаження, запису, попередньої вибірки та команди роботи з кеш-пам'яттю.

Виконання всіх команд завантаження і запису починається з тритактової послідовності, під час якої здійснюються видача команди, обчислення віртуальної адреси й перетворення віртуальної адреси на фізичну. Перетворення адреси здійснюється тільки одного дня під час виконання команди. Виробляється звернення до кеш-пам'яті даних, і пересилання необхідних даних завершується за наявності даних у кеш-пам'яті першого рівня. У разі промаху або у разі зайнятості необхідного порту реєстрового файла, що поділяється, звернення до кешу даних та до тегу відбувається повторно після отримання даних з кеш-пам'яті другого рівня або з основної пам'яті.

TLB містить 64 рядки і виконує перетворення віртуальної адреси на фізичну. Віртуальна адреса для перетворення надходить з пристрою обчислення адреси або з лічильника команд.

Зовнішня кеш-пам'ять другого рівня управляється за допомогою внутрішнього контролера, який має спеціальний порт для під'єднання кеш-пам'яті. Спеціальна магістраль даних довжиною в 128 біт здійснює пересилання даних на тактовій частоті процесора 200 МГц. У процесорі є також 64-бітова шина даних системного інтерфейсу. Кеш-пам'ять другого рівня має двоканальну множинно-асоціативну організацію. Її максимальний об'єм – 16 Мбайт, а мінімальний об'єм – 512 Кбайт. Пересилання здійснюються 128-бітовими порціями (4 32-бітових слова). Для пересилання великих блоків даних використовуються послідовні цикли шини:

- чотирисловне звертання (128 біт), що застосовується для команд кеш-пам'яті;
- восьмислівне звертання (256 біт), що використовується для перезавантаження первинного кешу даних;
- шістнадцятислівне звертання (512 біт), що застосовується для перезавантаження первинного кешу команд;

– тридцятидвослівне звертання (1024 біт), що використовується для перезавантаження кеш-пам'яті другого рівня.

Системний інтерфейс процесора R10000 працює як шлюз між самим процесором і пов'язаним з ним кешем другого рівня й іншою системою. Системний інтерфейс працює з тактовою частотою зовнішньої синхронізації. Можливим є програмування роботи системного інтерфейсу на тактовій частоті 200, 133, 100, 80, 67, 57 та 50 МГц.

Процесор підтримує протокол розщеплювання транзакцій, що дає змогу здійснювати видачу чергових запитів процесором або зовнішнім абонентом шини без очікування на відповідь на попередній запит. Максимально підтримується до чотирьох одночасних транзакцій на шині.

Процесор R10000 допускає два способи організації багатопроцесорної системи. Один зі способів пов'язаний зі створенням спеціального зовнішнього інтерфейсу для кожного процесора системи. Цей інтерфейс звичайно реалізується за допомогою замовленої інтегральної схеми, яка організовує шлюз до основної пам'яті й підсистеми введення-виведення. При такому типі з'єднань процесори не пов'язані один з одним безпосередньо, а взаємодіють через цей спеціальний інтерфейс. Хоча така реалізація загальноприйнята, її вартість доволі висока.

Другий спосіб призначений для досягнення максимальної продуктивності при мінімальних затратах. Він передбачає використання від двох до чотирьох процесорів, з'єднаних шиною Cluster Bus. У цьому разі необхідний тільки один зовнішній інтерфейс для взаємодії з іншими ресурсами системи.

6.4. Мікропроцесори з довгим командним словом (VLIW)

Архітектура машин з дуже довгим командним словом (VLIW – Very Long Instruction Word) базується на множині незалежних функціональних пристроїв. Замість того, щоб намагатися паралельно видавати в ці пристрої незалежні команди, у таких машинах декілька операцій запаковуються в одну дуже довгу команду. При цьому відповідальність за вибір і паралельне виконання операцій покладена повністю на компілятор, а апаратні засоби, необхідні для реалізації суперскалярної обробки, просто будуть відсутні.

VLIW – це команда, що охоплює, наприклад, дві цілочислові операції, дві операції з плаваючою крапкою, дві операції звернення до пам'яті й операцію переходу.

З точки зору архітектурних рішень машину з дуже довгим командним словом можна розглядати як розширення RISC-архітектури.

Як і в RISC-архітектурі, апаратні ресурси VLIW-машини надані компілятору. До цих ресурсів належать конвеєрні функціональні пристрої, шини й банки пам'яті. Для підтримки високої пропускної здатності між функціональними пристроями і регістрами необхідно використати декілька наборів регістрів. Апаратний дозвіл конфліктів вимикається, й перевага віддається простій логіці керування. На відміну від традиційних комп'ютерів, регістри та шини не резервуються, а їхнє використання повністю визначається під час компіляції.

Один із сучасних підходів до збільшення пропускної здатності пам'яті – використання розшарування пам'яті. Однак у системі з розшарованою пам'яттю виникає конфлікт банку, якщо банк зайнятий попереднім зверненням. У звичайних комп'ютерах стан зайнятості банків пам'яті відстежується апаратно і перевіряється, коли видається команда, виконання якої пов'язане зі зверненням до пам'яті. В комп'ютері типу VLIW цю функцію передано програмним засобам. Можливі конфлікти банків визначає спеціальний модуль компілятора – модуль уникнення конфліктів.

Найбільш повно VLIW-концепцію використовує суперкомп'ютер «Ельбрус-3», а також мікропроцесор E2K («Ельбрус-2000») та ін.

Архітектура процесора E2K

У класичному варіанті VLIW довжина команди фіксована. Це призводить до значних додаткових затрат пам'яті під коди програми і, як наслідок, до неефективного використання програмного кешу. До того ж це є обмеженням на «масштабність» мікропроцесора. Для цієї концепції масштабність слід розуміти як можливість нарощувати число функціональних виконавчих пристроїв (ФП) без зміни програмних кодів.

В IA-64 по 3 команди об'єднуються у так звані зв'язки (bundle) довжиною 128 розрядів. У формат команди вводяться спеціальні розряди маски, які вказують на залежності між командами. Наявність такої залежності зменшує можливість паралельного виконання відповідних операцій. У RISC-процесорах подібні взаємозалежності визначаються апаратним шляхом, або апаратно.

У мікропроцесорі E2K використовується підхід, що базується на застосуванні команд змінної довжини. Загальний формат команд МП E2K наведено на рис. 6.13.

Заголовок	Склад 1	...	Склад N
-----------	---------	-----	---------

Рис. 6.13. Структура команди МП E2K

Команда МП Е2К складається зі складів довжиною 32 розряди кожний. Число цих складів може змінюватися від 2 до 16, причому цю структуру команди можна розширити до 32 складів.

Будь-яка команда завжди вміщує 1 склад заголовка і від 1 до 15 складів, які вказують на операції, що можуть виконуватися паралельно. Склад заголовка містить інформацію про структуру команди та її довжину, що полегшує команди змінної довжини.

Таблиця 6.1

Типи складів команди Е2К

Зміст складу	Число складів
Заголовок	1
Операції АЛП	6
Керування підготовкою переходу	3
Додаткові операції АЛП при зачепленні	2
Завантаження з буфера попередньої вибірки масивів у реєстр	4
Літеральні константи для ФП	4
Логічні операції з предикатами	3
Предикати і маски для керування ФП	3

Якщо не брати до уваги заголовок, то в архітектурі Е2К використовується 7 типів складів (табл. 6.1). У команді може бути подано відразу декілька складів одного типу, максимальне число яких вказано в останньому стовпці цієї таблиці. Кількість і типи складів у команді МП Е2К закладаються у заголовку.

Застосування заголовка дає змогу не проводити попереднього декодування команд перед їхнім переміщенням у кеш команд. Негативним аспектом введення поля заголовка є збільшення довжини команди.

Спрощену структурну схему МП Е2К наведено на рис. 6.14.

В архітектурі МП Е2К подано надвеликий файл реєстрів та 256 реєстрів по 64 розряди кожний. Усі реєстри є універсальними й можуть зберігати як цілочислові дані, так і числа з плаваючою крапкою. Це різко виділяє Е2К серед практично всіх сучасних МП, в яких реєстри загального призначення відокремлені від реєстрів плаваючої крапки.

До архітектурних особливостей МП Е2К слід зарахувати наявність в їхній мікроархітектурі 2 кластерів – кожний кластер містить по 256 реєстрів. Всього у МП є 30 реєстрових портів: 20 портів читання (по 10 портів на кластер) та 10 портів запису.

Кеш даних другого рівня (D-кеш L2) має ємність 256 Кбайт при часі доступу у 8 тактів. Він є двоканальним частково-асоціативним і

має 4 банки, тобто забезпечує 4-кратне розшарування кеш-пам'яті. На відміну від D-кеш L1, у D-кеш L2 застосовується алгоритм зворотного запису.

Якщо вхідні буфери кеш-пам'яті другого рівня будуть переповнені, то МП зупиняється. При невлучанні у кеш другого рівня буде відбуватися звертання до зовнішнього інтерфейсу оперативної пам'яті.

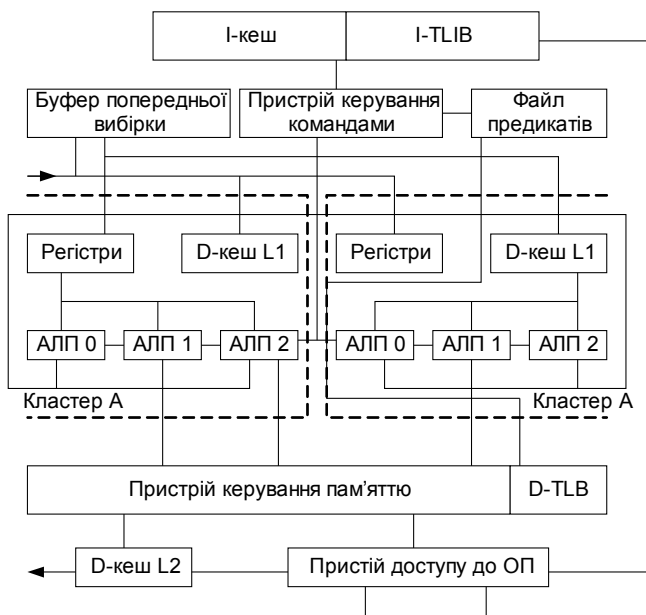


Рис. 6.14. Спрощена структурна схема МП Е2К

Як і кеш-пам'ять, буфер швидкої переадресації (TLB) у МП Е2К має дворівневу ієрархію: 16 рядків повністю асоціативної пам'яті плюс 512 рядків TLB другого рівня (останній є 4-канальним з доступом у 2 такти). До того ж підтримується до 4 одночасних перетворень адрес при звертанні до кешу другого рівня чи до оперативної пам'яті.

При невлучанні у TLB першого рівня процесор зупиняється на 4 такти. В разі невлучання у TLB другого рівня час чекання процесора становить від 10 до приблизно 200 тактів залежно від числа необхідних переглядів у таблиці сторінок і ситуації з влучанням-невлучанням у кеші другого й третього рівнів. Ємність TLB другого рівня є доволі великою, тому більш реальна небезпека полягає в невлучанні у TLB першого рівня. При роботі з кешем першого рівня TLB взагалі непотрібний, тому що цей кеш індексується і тегується віртуально.

Кеш команд першого рівня має ємність 64 Кбайт при довжині рядка 256 байт. Цей кеш є 4-канальним та частково-асоціативним. Ширина тракту, по якому команди з кешу команд надходять у пристрій керування командами, становить 266 байт, але максимальна швидкість заповнення кешу команд дорівнює 32 байти за такт, що може створити вузьке місце в мікроархітектурі МП, якщо коди не будуть майже лінійними. Буфер TLB для команд має ємність 64 рядки і є повністю асоціативним.

Функціональні пристрої МП рознесено по двох кластерах, у кожному з яких вміщено по три однакових цілочислових конвеєри АЛП (один з кластерів має також ФП розподілу – з цілочисловою та плаваючою крапками). Кожний кластер може одночасно виконувати до 2 операцій завантаження регістрів чи одну операцію запису в оперативну пам'ять, відповідно 2 кластери – до 4 операцій завантаження регістрів чи до 2 записів в оперативну пам'ять. Можливий також змішаний випадок: два завантаження плюс один запис.

Буфер попереднього підкачування (є частиною пристрою доступу до масивів та задіюється тільки при роботі з масивами у циклах), як і кластурні адресні суматори, на рис. 6.14 не наведено.

Серед інших особливостей МП Е2К можна виділити сегментно-сторінкову організацію і підтримку мультипрограмування у стилі x86. У поєднанні з розробленими засобами двійкової компіляції й спеціальних апаратних засобів її підтримки дозволяють x86-коди. Мікропроцесор Е2К є однокристальною реалізацією суперкомп'ютера «Эльбрус-3».

7. АРХІТЕКТУРА ТРАНСП'ЮТЕРІВ І ТРАНСП'ЮТЕРНИХ СИСТЕМ

7.1. Основні поняття і концепції архітектури трансп'ютерів

Поряд з традиційними мікропроцесорами, що мають важливе значення для архітектури персональних комп'ютерів, міні- і суперміні-комп'ютерів, дедалі важливішого значення в галузі мікропроцесорної техніки набуває так званий трансп'ютер – особливий тип мікропроцесора, розроблений компанією «Inmos International plc.».

Трансп'ютер – це незалежний обчислювальний елемент (перший мікропроцесор), який має власний набір команд й орієнтований на архітектуру систем паралельної обробки даних. Головна мета створення трансп'ютера – надати в розпорядження розробників систем стандартний блок, який можна було би використовувати у великих кількостях як базовий елемент при побудові винятково високопродуктивних комп'ютерів та систем. Його особливістю є те, що кожний окремих трансп'ютер має чотири елементи зв'язку, через які він може спрягатися з іншими трансп'ютерами (рис. 7.1). Отже, трансп'ютер можна умовно представити блоком з чотирма зовнішніми зв'язками. Таким чином можливо з'єднати один з одним декілька трансп'ютерів у вигляді мережі. Використовуючи такий вид зв'язку, можна об'єднати велику кількість трансп'ютерів в єдину системну архітектуру для проведення паралельних обчислень. При цьому продуктивність такої системи прирівнюється до продуктивності великих комп'ютерів.

Реалізація паралелізму відбувається трьома способами. Перший з них – SIMD – визначає роботу з одним потоком команд і множиною потоків даних. Тут за єдиною програмою всі процесори обробляють власні масиви даних під керуванням провідного процесора. Цей спосіб широко застосовується при обробці сигналів та зображень у реальному часі, тобто у тих випадках, коли фрагменти задачі обробляються за подібними алгоритмами. Пропонований спосіб є відносно простим, але він не дає змоги скористатися всіма можливостями паралельної обробки.

Другий спосіб – MIMD – визначає наявність множини потоків як команд, так і даних. Тут процесори працюють за своїми програмами незалежно один від одного, лише епізодично з'єднуючись один з одним. Цей спосіб використовується особливо широко, наприклад, при симетричній мультипроцесорній обробці.

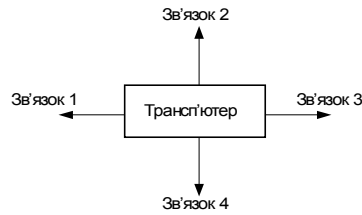


Рис. 7.1. Умовне зображення трансп'ютера

Третій спосіб – MISD – є, власне кажучи, теоретичною побудовою. У нього декілька потоків команд виконують різні операції над одним потоком даних.

Особливість трансп'ютерних систем полягає у можливості їхньої адаптації для розв'язання специфічних задач на основі оптимального вибору топології трансп'ютерних мереж. Завдяки цьому ще до етапу програмування можна досягти високого ступеня розпаралелювання задач.

7.2. Типи трансп'ютерів та їхня організація

Перший 32-розрядний трансп'ютер T414 характеризується часом циклу 50 нс і швидкодією близько 10 млн. операцій/с. Він вміщує мікропроцесор з RISC-архітектурою (МП зі скороченим набором команд), а також апаратні засоби, які забезпечують паралельні обчислення. За повідомленнями представників фірми «Inmos», трансп'ютер T414 виконує програми мовою високого рівня швидше, ніж МП 68020 компанії «Motorola», 80286 фірми «Intel» і 32032 корпорації «National Semiconductor» відповідно в 4, 9 і 12 разів. При виконанні операцій нормальної точності над числами з плаваючою крапкою швидкодія трансп'ютера T414 в 15 разів більша, ніж у МП 68020 і лише на 20% нижча, ніж у МП 68020 і сопроцесора при їхній спільній роботі. Структурну схему трансп'ютера T414 наведено на рис. 7.2.

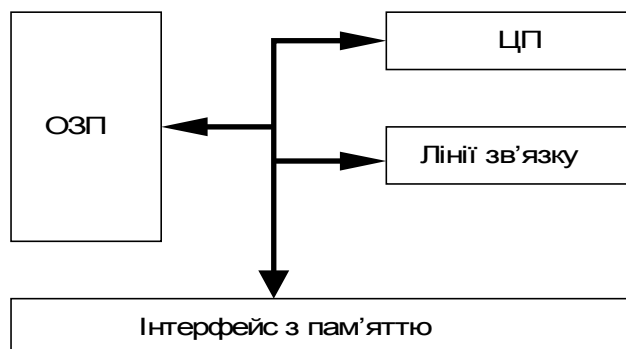


Рис. 7.2. Внутрішня архітектура трансп'ютера T414

Радикальною відмінністю трансп'ютера від усіх інших МП є наявність на кристалі пам'яті ємністю 2 або 4 Кбайт, причому існує спеціальний інтерфейс, який дає змогу під'єднати додатковий зовнішній ОЗП. Обмін даними між трансп'ютером і зовнішнім ОЗП може проводитися зі швидкістю до 26 Мбайт/с. Забезпечення кожного МП-трансп'ютера власною локальною пам'яттю виключає вузьке місце «процесор – пам'ять», характерне для традиційних централізованих систем. Крім цього, до складу трансп'ютера входять чотири високошвидкісних асинхронних канали зв'язку, які використовуються для обміну повідомленнями з іншими трансп'ютерами. Ці канали працюють паралельно з процесорами, забезпечуючи сумарну швидкість передачі даних більше, ніж 5 Мбайт/с. За повідомленнями представників фірми «Inmos», така мережа каналів передачі даних, яка замінює шину колективного користування, підвищує швидкодію системи до 1 млрд. операцій за секунду.

Трансп'ютер Т800

На відміну від трансп'ютера Т414, трансп'ютер Т800 має арифметичний блок для роботи з плаваючою крапкою, в 10–15 разів вищу швидкодію складних обчислень, в 2 рази вищу швидкість зовнішнього зв'язку і в 2 рази більшу внутрішню пам'ять (4Кбайт). Трансп'ютер може звертатися до зовнішньої пам'яті ємністю до 32 Мбайт. Структуру трансп'ютера Т800 наведено на рис. 7.3.

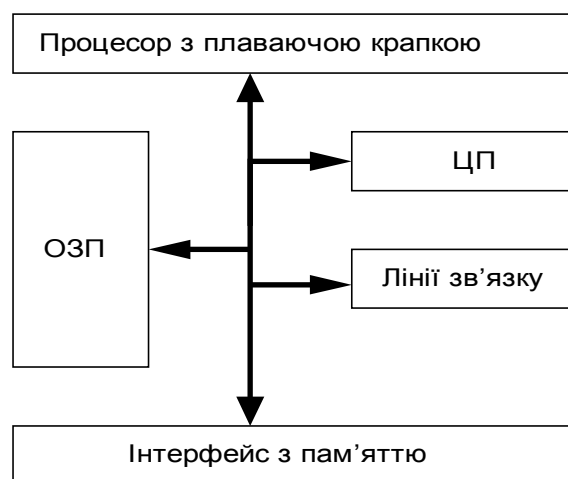


Рис. 7.3. Внутрішня архітектура трансп'ютера Т800

Трансп'ютер має вбудований планувальник, що допомагає планувати процеси до виконання. Планувальник, який є традиційною

частиною операційної системи, розміщений безпосередньо в мікропроцесорному чіпі. Крім вказаних типів трансп'ютерів, на сьогодні розроблено і використовуються для побудови систем також інші типи.

7.3. Трансп'ютерна організація паралельних обчислень

Прикладом паралельних обчислень з використанням трансп'ютерів, об'єднаних у мережу, є перемноження двох матриць А і В (рис. 7.4). Кожний окремий трансп'ютер виконує операцію:

$$r_{ij} = \sum_{k=1}^n a_{ik} \cdot b_{kj} \quad (7.1)$$

Однак не кожна проблема так легко піддається вирішенню за допомогою трансп'ютерної мережі. Для побудови трансп'ютерної мережі використовують два пройоми. Один з них ґрунтується на аналізі *критичних шляхів на графі задач*, а інший – називають *методом базових елементів*.

Для першого методу (розв'язання квадратного рівняння) побудуємо граф для процесу, в межах якого визначаються корені квадратного рівняння $ax^2 + bx + c = 0$. Розв'язанням, як відомо, буде вираз:

$$X_{1,2} = (-b \pm \sqrt{b^2 - 4ac}) / 2a.$$

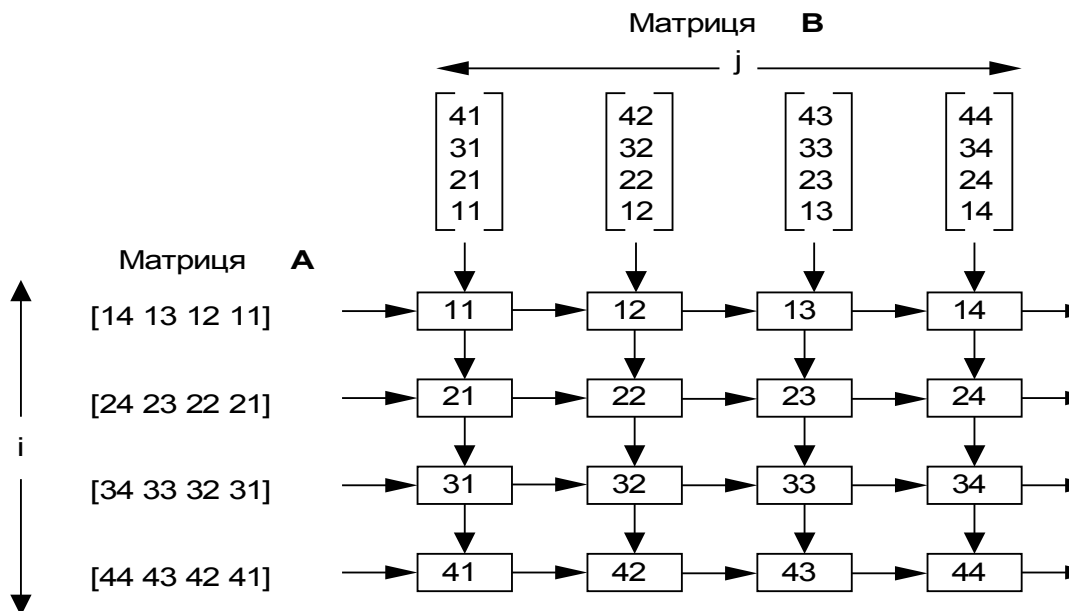


Рис. 7.4. Трансп'ютерна мережа для перемноження матриць

Нехай відносна тривалість базових обчислювальних операцій визначається деяким набором вагових коефіцієнтів, наприклад:

- введення-виведення – 20;
- добування квадратного кореня – 90;
- додавання-віднімання – 40;
- множення – 70;
- ділення – 80.

На основі цих коефіцієнтів можна підрахувати відносний час виконання окремих інструкцій програми обчислення коренів (*root*) квадратного рівняння. Складену програму у вигляді набору інструкцій, що передбачає виконання відповідних задач, наведено в табл. 7.1.

Потік даних, які використовуються у процесі обчислень, не зовсім відповідає послідовності інструкцій програми. Цей потік, що визначає обчислювальну структуру алгоритму, можна описати графом задачі, в якому окремі задачі відповідають вершинам (рис. 7.5). Надалі граф задачі слід відобразити на трансп'ютерну мережу. Ця процедура неоднозначна, оскільки різним може бути число трансп'ютерів у мережі й існують різні варіанти перерозподілу задач по трансп'ютерах. Мережу з трьох трансп'ютерів для виконання згаданої програми показано на рис. 7.6.

Таким чином, граф задач формує деякі вихідні позиції для планування обчислювального процесу з урахуванням мінімального значення холостих втрат часу. Надалі цей план можна використати для розробки програми мовою OCCAM. Ефективність слід оцінити за прискоренням обчислень у межах паралельного процесу порівняно з послідовним процесом.

Таблиця 7.1

Інструкції для визначення коренів квадратного рівняння

Інструкція	Номер задачі	Відносна тривалість
starttask	1	0
input a,b,c	2	20
d:= b • b	3	70
e:= 4ac	4	140
f:= sgrt(d–e)	5	130
g:= 2a	6	70
root 1:= (–b–f)/g	7	160
root 2:= (–b+f)/g	8	160
output root 1, root 2	9	20
endtask	10	0

Метод базових елементів належить до другого способу організації паралелізму. Згідно з цим способом виділяють елементарні процеси і будують на їхній основі цілісну систему. Внутрішній планувальник трансп'ютера підтримує паралельну роботу елементарних процесів.

7.4. Мультипроцесорні трансп'ютерні системи

У системі мінімальної конфігурації всі процеси можуть виконуватися на одному трансп'ютері. Однак для підвищення продуктивності використовують декілька трансп'ютерів, виконуючи кожний процес на окремому трансп'ютері. Розробники фірми «Meiko» перетворили цю загальну архітектурну концепцію на обчислювальну систему Computing Surface, яка будується з набору, що містить чотири базові стандартні плати:

- плату комп'ютера;
- плату пам'яті;
- графічну плату;
- плату введення-виведення і розкручування (плата ВВ0).

На платі комп'ютера розміщені чотири трансп'ютери разом з ОЗП (пам'ять з довільною вибіркою) ємністю 48Кбайт. На платі пам'яті знаходяться 120 нс ДОЗП ємністю 2 Мбайт і один трансп'ютер-контролер. На графічній платі розміщені два трансп'ютери кожний зі своїм ОЗП ємністю 16 Кбайт, чотири графічних контролери 7220 фірми «NEC» та кристал процесора кольору G170 компанії «Inmos», який може безпосередньо керувати растровою відеосистемою. На платі введення-виведення і розкручування (ВВ) знаходяться ОЗП ємністю 16 Кбайт та ППЗП ємністю 32 Кбайт. На цій платі розміщені два порти RS-232C і один інтерфейс IEEE-488. Передбачається вводити в систему й інші плати, наприклад, МП з плаваючою крапкою. Подвійне шасі-каркас стандарту Eurocard дає змогу встановлювати до 40 схемних плат.

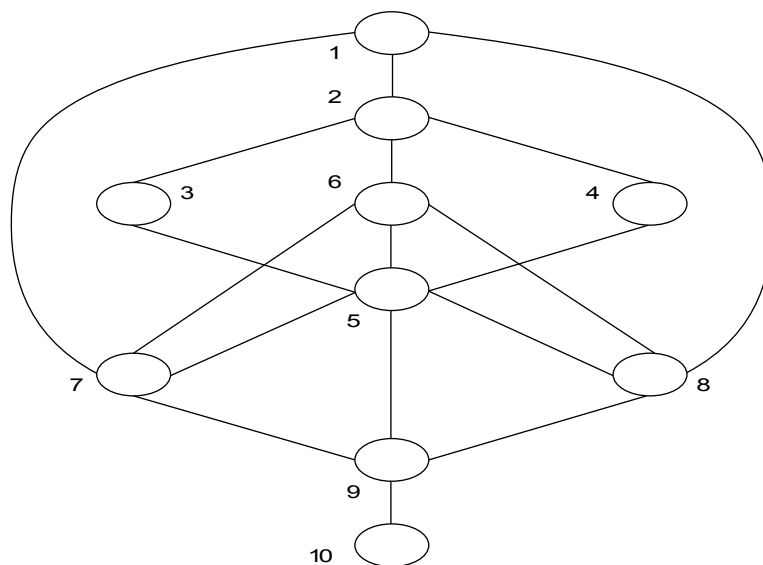


Рис. 7.5. Граф задач для процесу визначення коренів квадратного рівняння

У постійній пам'яті плати ВВ міститься програма початкового завантаження, яка приводить процесор у робочий стан, а також засоби контролю його роботи. У процесі розкручування проводяться завантаження коду із системи проектування програм і передача цього коду по мережі трансп'ютерних каналів на відповідні процесори. Плата ВВ обслуговує також контрольну шину, перевіряючи, який тип плати знаходиться в кожному гнізді, а також збирає статистичні дані про помилки в апаратурі. Вона забезпечує локалізацію помилок з точністю до плати, процесора і навіть конкретної адреси.

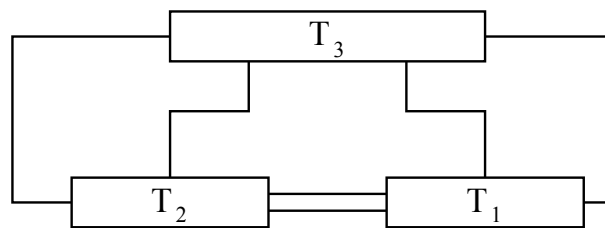


Рис. 7.6. Мережа трансп'ютерів для графа задач на рис. 7.5

У ході реалізації одного з перших в Європі суперкомп'ютерних проектів фірма «Parsytek GmbH» створила комп'ютер Megaframe Supercluster на базі 32-розрядних трансп'ютерів T800 компанії «Imnos Ltd», які об'єднані у групи (кластери) та взаємодіють через виділені їм канали зв'язку.

У комп'ютері Supercluster внутрішні процесорні лінії зв'язку використовуються для розділення обчислювальної задачі на множину паралельних підзадач обміну даними й управління інформаційними потоками по багатьох каналах порівняно з шинною архітектурою. Цей підхід відрізняється значно більшою пропускнуою здатністю. Компанія «Parstek» представила на ринок дві моделі комп'ютера: 64-процесорну модель 64 і 256-процесорну модель 256.

Комп'ютер Supercluster виконаний на базі трансп'ютера T800 з чотирма каналами зв'язку. Група із 16 трансп'ютерів об'єднується в кластер, який обмінюється даними через пристрій управління конфігурацією мережі (ПУКМ). Кожен кластер має дві 16-канальні лінії зв'язку. У базовому 64-процесорному комп'ютері кожен чотири кластери об'єднуються у блоки двома ПУКМ. Кожен блок має інтерфейси АРМ і кластер системного обслуговування, що вміщує дискову файлову систему, засоби головного процесора і деякі спеціалізовані модулі. Більші комп'ютери цього типу можна отримати, сполучаючи два або

декілька базових блоків каналами зв'язку, які утворюються двома ПУКМ кожного кластера.

Система із 64 процесорів базового комп'ютера Supercluster функціонує зі швидкістю 640 млн. команд/с, а при виконанні скалярних обчислень швидкодія досягає 96 млн. операцій з плаваючою крапкою за секунду. Модель 256, яка об'єднує чотири блоки, має продуктивність 2560 млн. команд/с, або 386 млн. операцій з плаваючою крапкою за секунду.

Незважаючи на практично необмежені можливості для розширення, комп'ютер Supercluster відрізняється вбудованими засобами головного процесора, які забезпечують йому незалежність від вхідних буферних комп'ютерів. Крім цього, через високошвидкісні канали до комп'ютера Supercluster можуть під'єднуватись стандартні АРМ, прикладні програми яких отримують доступ до заданих користувачем областей системи.

Віденською фірмою «IMPULS» розроблена мультитрансп'ютерна система (МТС) IMPULS 2400. Вона складається із 64 + 2 трансп'ютерів Т800 з тактовою частотою 17 МГц і ОЗП ємністю до 2 Мбайт. Трансп'ютери під'єднані не послідовно один за одним, а через три так звані комутатори каналів зв'язку (Link-Switches), причому через ці комутатори можуть бути встановлені програмнокеровані зв'язки. Для зберігання даних система IMPULS 2400 має у своєму складі два дискових накопичувачі ємністю 640 Мбайт кожний і накопичувач на магнітній стрічці ємністю 2,3 Гбайт.

Керування МТС відбувається при допомозі двох зовнішніх комп'ютерів: ATARI, що підтримує за допомогою двох додаткових трансп'ютерів файлову систему, через яку можна звернутися до двох жорстких дисків, та IBM PC/XT з диском ємністю 80 Мбайт і трансп'ютерною платою для ініціалізації системи. Для під'єднання до мережі передбачений мережевий адаптер ETHERNET.

Трансп'ютер був спроектований як блок для побудови паралельних систем. Це дає змогу (проектувальникам систем – системним проектувальникам) створювати мережі процесорів, які задовольняють вимоги щодо продуктивності в деяких сферах застосування засобів ОТ. Трансп'ютерна мережа є набором трансп'ютерів, кожний з яких вважається процесором з локальною пам'яттю і зв'язаний один з одним за допомогою ліній зв'язку трансп'ютера. Як правило, в такій мережі немає спільної пам'яті у процесорів; дані передаються між процесорами тільки через лінії зв'язку.

Багато з існуючих алгоритмів легко реалізуються в конвеєрах чи декількох конвеєрах, і оскільки програмісти ознайомлені з такою організацією обчислювального процесу, то це та мережа, яка найбільш часто використовується. N-кроковий конвеєр має $2 \times N$ вільних ліній зв'язку, тому в ньому неоптимально використовуються засоби зв'язку трансп'ютера. При цьому конвеєру властиві всі недоліки машин класу МКОД. Деревоподібна топологія підходить для ієрархічних процесів, таких як редукція даних з передаванням даних вгору по дереву і сортування даних з передаванням даних вниз по дереву.

Двовимірний масив використовується для даних, що мають структуру масиву, як наприклад, образи чи матриці й застосовуються для опрацювання зображень на низькому рівні. Топологія двовимірного масиву еквівалентна топології матричних процесорів у машинах класу ОКМД, і в них повністю використовуються чотири лінії зв'язку трансп'ютера.

Мережа може бути неоднорідною, тобто в неї можуть входити різні елементи сімейства трансп'ютерів, як наприклад, такі процесори:

IMS T222 – 16-бітовий процесор;

IMS M212 – 16-бітовий процесор з дисковим інтерфейсом;

IMS T414 – 32-бітовий процесор;

IMS T800 – 32-бітовий процесор з вбудованим у мікросхему модулем, що працює з числами з плаваючою крапкою.

Лінії зв'язку в трансп'ютерній мережі можуть бути фіксованими або з переключенням. Крім цього, топологію мережі можна змінювати за допомогою схем комутації ліній зв'язку, які настраюються на рівні програмного забезпечення до запуску програми в мережі. Фірма «INMOS» розробила чіп (VIC) переключення ліній зв'язку – IMS C004, який представляє собою матричний перемикач між 32 вхідними та 32 вихідними лініями зв'язку і є основним компонентом для побудови трансп'ютерних плат з лініями зв'язку, які переключаються.

Програмне забезпечення

Для програмування трансп'ютерних машин і систем використовують пакет TDS (Transputer Development System), який застосовується для підтримки зовнішніх ПК, а також для завантаження і перевірки всієї системи. Програмне забезпечення розглянемо на прикладі програмного забезпечення мультитрансп'ютерної системи IMPULS 2400. Із систем програмування ця МТС має три компілятори: OCCAM, паралельний FORTRAN і паралельний C. FORTRAN та C для програмістів є знайомим середовищем, а мова OCCAM – новинка. З її

допомогою відбувається взаємодія з трансп'ютерною структурою системи і засобами комунікацій. OCCAM робить можливим програмування на нижньому рівні, як наприклад, занесення змінних у спеціальні адреси пам'яті, доступ до таймерів або вставка машинного коду. Щоб запустити яку-небудь трансп'ютерну програму, необхідно розподілити окремі її частини по відповідних трансп'ютерах. В OCCAM цей розподіл проводиться за допомогою так званого опису конфігурацій. Опис відбувається одним єдиним конфігуратором. Конфігуратор видає необхідну інформацію про початкове завантаження, щоб запустити трансп'ютерну систему у відповідній мережі, а також записує цю інформацію разом з компільованим кодом у певну частину програми. Крім цього, є ряд пакетів підпрограм з прикладними математичними функціями, наприклад, підпрограми для керування комутаторами каналів. За допомогою цих комутаторів вся трансп'ютерна мережа з моменту введення програми може змінювати топологію. Залежно від задачі, для різних прикладних програм можуть бути програмно встановлені різні варіанти топології.

Мова OCCAM

Мова програмування OCCAM дає змогу описувати прикладну задачу як набір паралельно працюючих процесів, що взаємодіють по каналах. У такому поданні кожен OCCAM-процес описує поведінку одного компонента прикладної задачі, а кожний канал – взаємодію між компонентами.

Операційні системи

ОС EXPRESS. Операційна система EXPRESS складається з ядра, побудованого на основі методу надсилання повідомлень, і набору бібліотечних викликів мови С, які забезпечують інтерфейс. Цим забезпечується середовище паралельного програмування, де реалізовані деякі елементарні функції загального призначення. Вона забезпечує введення та виведення на термінал з будь-якої точки мережі процесорів без явного задавання топології мережі.

ОС LINDA. Операційна система LINDA є другою системою, спроектованою для спрощення процесу написання паралельних програм. Її можна описати як мову програмування, хоч у дійсності вона складається з набору викликів, внесених в існуючі мови програмування, такі як С чи ФОРТРАН, для виконання деяких функцій операційної системи.

ОС TROLLIUS. Ця операційна система призначена для виконання на комп'ютерах з паралельною архітектурою, де кожний комп'ютер

складається з ряду обчислювачів на базі трансп'ютерів, з'єднаних з деякою машиною традиційної архітектури. ОС Trolius складається з ядра, що завантажується в мережу трансп'ютерів і підтримує набір бібліотечних викликів для програм, які виконуються під його керуванням. Підтримуються також дві мови програмування С та Фортран, а також декілька систем розробки, таких як Асемблер, Компоновщик і Відладчик.

ОС HELIOS. Ця операційна система була спеціально спроектована для виконання на трансп'ютерах, і з самого початку проектування передбачалось, що вона буде працювати на багато процесорних системах. ОС Helios побудована на моделі «клієнт–сервер» для операційних систем; ця методика широко застосовується у багатьох існуючих ОС. Процес-клієнт, який потребує доступу до системного ресурсу, як наприклад, для відкриття файлу, надсилає повідомлення обслуговуючому процесу-серверу для того, щоб останній виконав для нього необхідну дію. Сервер надсилає повідомлення-відповідь, яке вказує на успішне чи неуспішне виконання цієї дії. Потім процес-клієнт може виконувати операції зчитування і запису в цей файл шляхом надсилання додаткових повідомлень серверу.

До потенційних замовників трансп'ютерів належать виробники обладнання, яким необхідно вбудовувати у свої системи засоби великої обчислювальної потужності, але малої вартості.

В електронній промисловості трансп'ютерні системи використовуються у високопродуктивних АРМ-системах автоматизованого проектування при моделюванні схем, а також у складі спеціальних процесорів-акселераторів. Для розробників механічних пристроїв ця система може створювати тривимірні моделі нових виробів, що дуже схожі на реальні об'єкти. Якщо вести мову про інші сфери застосування, то це: автоматизація виробничих процесів реального часу, високошвидкісний аналіз даних, розпізнавання сигналів і образів, застосування в кінематографії та телебаченні (генерація рухомих відеозображень), у тренажерах, для навчання пілотів літальних апаратів і операторів різних промислових установок. До того ж трансп'ютерні системи мають достатню обчислювальну потужність, необхідну для майбутнього застосування в системах штучного інтелекту.

8. ІНТЕРФЕЙСИ ТА ШИНИ КОМП'ЮТЕРІВ

8.1. Загальна характеристика інтерфейсів

Створення сучасних засобів обчислювальної техніки пов'язане із завданням об'єднання в єдиний комплекс різних видів блоків комп'ютерів, пристроїв зберігання й відображення інформації, вимірювальних приладів, пристроїв для зв'язку з об'єктами апаратури передачі даних і т. ін. Це завдання покладається на уніфіковані системи спряження – інтерфейси.

Інтерфейс в обчислювальній техніці – це сукупність правил, які визначають електричні, програмні та конструктивні засоби для реалізації взаємодії різних пристроїв. Згідно з ГОСТ 15971-74, стандартний інтерфейс слід розуміти як сукупність уніфікованих апаратних, програмних і конструктивних засобів, необхідних для реалізації взаємодії різних функціональних елементів в автоматичних системах збору й обробки інформації за умов, регламентованих стандартом (для забезпечення інформаційної, електричної і конструктивної сумісності вказаних елементів).

Структурну схему інтерфейсу наведено на рис. 8.1.

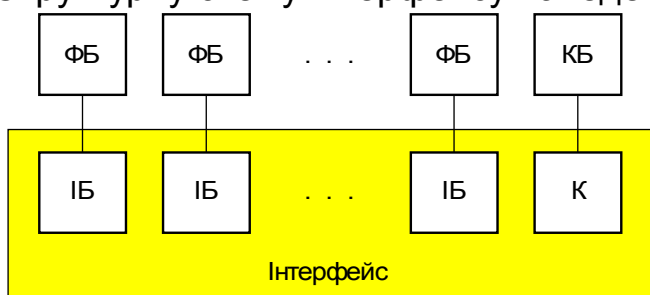


Рис. 8.1. Структурна схема інтерфейсу:

ФБ – функціональний блок; ІБ – інтерфейсний блок; К – контролер; КБ – керуючий блок

Інтерфейс може бути як стандартним, так і оригінальним для конкретних пристроїв. Розробка й впровадження стандартних інтерфейсів спрямовані на забезпечення конструктивної, електричної й програмної сумісності між різними пристроями комп'ютерів і комп'ютерних систем. Різні інтерфейси характеризуються сумісністю.

Конструктивна сумісність – це узгодженість конструктивних елементів інтерфейсу, яка забезпечує під'єднання різних пристроїв до комп'ютера чи системи.

Умови конструктивної сумісності у більшості стандартів визначено не повністю. Тому в багатьох випадках використовують різні типи з'єднувачів. Розподіл у них сигналів добирають за домовленістю між виробниками різних пристроїв.

Електрична сумісність – це узгодженість структури й складу ліній зв'язку і сигналів між пристроями, величини, форми та інших параметрів електричних сигналів, рекомендації щодо параметрів ліній зв'язку, обмеження на відстань поширення сигналів й ін.

Програмна сумісність – це узгодженість взаємодії між різними компонентами згідно з логічними умовами режимів роботи. Логічні умови визначають формати даних, команд, адрес, станів пристроїв, набори та часові характеристики процедур взаємодії між пристроями.

Основні елементи інтерфейсів

Лінії інтерфейсу – це електричні зв'язки, через які передаються електричні сигнали. Лінії є фізичними елементами зв'язку інтерфейсу. Магістраль – це сукупність всіх ліній інтерфейсу.

Шина інтерфейсу – це сукупність ліній, які групуються відповідно до свого функціонального призначення. Розрізняють два типи шин: *інформаційні* та *керування*. Інформаційні шини теж поділяються на *адресні шини* і *шини даних*.

Шина керування призначена для передавання сигналів керування взаємодією між різними пристроями.

Шина даних призначена для передавання інформації, яку використовують в обчислювальному процесі. Як правило, інформація передається бінарним (двійковим кодом). Найчастіше шина даних містить 8, 16, 24, 32, 64 лінії інтерфейсу, тобто кількість ліній кратна байту.

Шина адреси призначена для передавання кодів адрес пристроїв, складових частин пристроїв, комірок пам'яті у процесі організації взаємодії між пристроями.

8.2. Класифікація інтерфейсів периферійних пристроїв

На даний час не існує загальноприйнятого підходу до класифікації інтерфейсів. Найпоширенішими слід вважати два підходи. Перший підхід ґрунтується на використанні класифікаційних ознак, а другий – на функціональному призначенні інтерфейсів. Певним узагальненням класифікації є стандарт на класифікаційні ознаки інтерфейсів (ГОСТ 26.016-81), який вміщує чотири основні ознаки:

- спосіб з'єднання компонентів схеми (магістральний, радіальний, ланцюговий, змішаний або ланцюговий);
- спосіб передавання інформації (паралельний, послідовний, паралельно-послідовний);
- принципи обміну інформацією (синхронний, асинхронний);
- режим передавання інформації (двобічний одночасний – дуплексний, двобічний почерговий – напівдуплексний, одnobічний – симплексний).

Названі ознаки дають змогу характеризувати тільки певні аспекти організації інтерфейсів. Здійснити більш повну характеристику і систематизацію інтерфейсів можна за умови класифікації за кількома сукупностями ознак, а саме: сфера поширення (функціональне призначення); логічна та функціональна організація; фізична реалізація.

Якщо використовувати перший підхід, то певний інтерфейс можна класифікувати таким чином:

- магістральний, паралельний, синхронний, дуплексний інтерфейс;
- радіальний, послідовний, синхронний, симплексний інтерфейс.

Згідно з функціональним призначенням, інтерфейси введення-виведення чи ПП об'єднують в окремих клас. Проте їх неможливо розглядати ізольовано без урахування особливостей комп'ютера, систем і комплексів, де використовуються ПП. В одних випадках інтерфейси орієнтовані лише на ПП, в інших – на більшість пристроїв комп'ютера. З огляду на це широкий клас інтерфейсів ПП поділяють на підкласи:

- введення-виведення універсальних комп'ютерів;
- системні малих комп'ютерів;
- універсальні загального призначення;
- спеціалізовані.

Одним з найпоширеніших інтерфейсів першого підкласу можна вважати інтерфейс введення-виведення комп'ютерів типу IBM 360/370 чи ЄС ЕОМ.

Системні інтерфейси малих комп'ютерів використовують, як правило, магістральний спосіб з'єднання компонентів. Найпоширенішими інтерфейсами цього підкласу є системні інтерфейси комп'ютерів фірми «DEC» і комп'ютерів на базі мікропроцесорів фірми «Intel».

Універсальні інтерфейси ПП загального призначення – це найпоширеніший підклас. Вони використовують різні способи з'єднання компонентів, паралельний та послідовний способи передавання інформації, синхронний і асинхронний принципи обміну інформацією й різні режими функціонування.

Спеціалізовані інтерфейси орієнтовані на конкретний тип ПП і насамперед на такі зовнішні запам'ятовувальні пристрої, як накопичувачі на магнітних, магнітооптичних та оптичних дисках і магнітних стрічках.

Основними характеристиками інтерфейсів введення-виведення є:

- місткість, що визначає максимально можливу кількість абонентів, яких можна одночасно під'єднати до контролера інтерфейсу;
- пропускну здатність (швидкість передавання), яка визначається допустимою швидкістю передавання даних, тривалістю виконання операції

встановлення і роз'єднання зв'язку й ступенем синхронності процесів передавання даних;

– максимальна довжина ліній зв'язку, яка визначає максимальну віддаленість джерел і приймачів інформації. Для більшості інтерфейсів введення-виведення довжина ліній перебуває в межах від 15 до 65 м;

– розрядність (ширина) інтерфейсу – максимальний розмір інформаційного елемента (в бітах), який може бути через нього переданий за один цикл.

8.3. Канали введення-виведення та їхні функції

Розв'язання економічних і статистичних задач на комп'ютері характеризується великим обсягом вхідної й вихідної інформації при порівняно нескладних обчисленнях. Використання великої кількості периферійних пристроїв (ПП) з відносно невеликою швидкістю приводить до того, що основний час роботи процесора зайнятий опрацюванням операцій введення-виведення й очікуванням готовності зовнішніх пристроїв до обміну інформацією. Це суттєво знижує швидкість комп'ютера. Ефективність роботи комп'ютера можна підвищити шляхом розподілу функції керування між центральним процесором і периферійними пристроями. Для цього до складу комп'ютера вносяться додаткові пристрої – канали введення-виведення, завданням яких є забезпечення взаємодії ЦП з ПП (рис. 8.2).

Канал – це пристрій або комплекс апаратури, який спеціально використовується для виконання операцій введення і виведення інформації й забезпечує всі необхідні двосторонні зв'язки між ОП, ЦП та множиною зовнішніх пристроїв при стандартних формах всіх даних, що передаються.

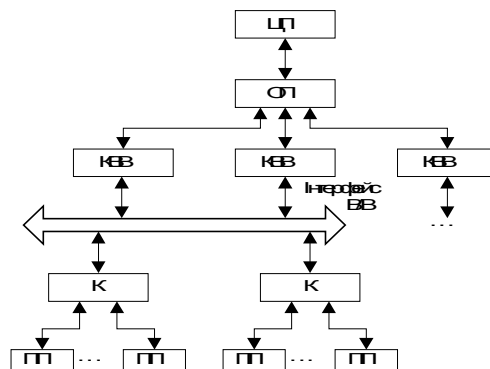


Рис. 8.2. Схема комп'ютера з каналами введення-виведення

Характерна особливість КВВ в тому, що канал працює за програмою, яка зберігається в пам'яті, тобто так само, як і центральний процесор. Звідси випливає, що КВВ є, власне кажучи, спеціалізованим

процесором введення-виведення. В результаті ЦП повністю звільняється від обслуговування операцій обміну ПП і пам'яті.

До складу каналу входять запам'ятовувальні вузли (регістри) й логічні схеми. Вони виконують пакування, розпакування і підрахунок даних, а також модифікацію адреси даних та передавання інформації з оперативної пам'яті у ПП і в зворотному порядку.

Канал працює під управлінням каналної програми, що зберігається у спеціально відведеній області основної пам'яті комп'ютера й складається з послідовності підкоманд, які називаються в ЄС ЕОМ керуючими словами каналу (КСК). Кожне КСК каналної програми визначає операцію обміну з пристроями введення чи виведення, область використаної оперативної пам'яті й дії, які необхідно провести після виконання цієї операції.

Залежно від швидкості роботи периферійних пристроїв і порядку обміну застосовуються різні типи каналів: *селекторні, байт-мультиплексні й блок-мультиплексні*.

Основними функціями каналу введення-виведення є: організація логічного зв'язку між ПП і центральними пристроями на час передавання інформації; організація керування автономною роботою ПП; буферизація даних та перетворення форматів у процесі обміну; контроль інформації, що передається; визначення поточних адрес ОЗП, в які має записатися чи з яких має зчитатися інформація, що буде передаватися; вироблення послідовності синхронізуючих і керуючих сигналів; організація завершення введення-виведення та від'єднання ПП.

8.4. Мультиплексний і селекторний канали, їхня структура й організація роботи

Можливість організації паралельної роботи ПП за допомогою каналів визначає режими обміну даними в каналах. Канали поділяються на *селекторні (СК)* і *мультиплексні (МК)*.

Селекторний канал забезпечує привілейоване обслуговування одного швидкодіючого периферійного пристрою. Засоби каналу, необхідні для обслуговування одного ПП, утворюють підканал. У разі роботи з СК периферійний пристрій після запуску залишається зв'язаний до закінчення ланцюга операцій. Запити на обслуговування від інших ПП, як і нові команди запуску операцій введення-виведення від процесора в цей час, каналом не сприймаються. Таким чином, всі засоби селекторного каналу монополізуються на час операції одним периферійним пристроєм, тобто можна вважати, що селекторний канал має тільки один підканал

(рис. 8.3). Селекторний канал забезпечує передавання даних, тобто зв'язок з високошвидкісними ПП, передусім з НМД і НМС. Характеристики СК залежать від параметрів моделі комп'ютера.

Мультимплексний канал (рис. 8.4) дає змогу одночасно обслуговувати кілька ПП, які паралельно працюють. Кожен з цих пристроїв зв'язується з каналом на короткі періоди часу тільки після того, як ПП підготовлений до приймання або видачі чергової порції інформації (байта, слова, групи слів). Період часу, протягом якого відбувається передавання інформації між каналом і підготовленим ПП, можна назвати сеансом зв'язку.

Якщо кілька ПП підготувались до сеансу зв'язку і запитують дозвіл на обслуговування з боку каналу, то канал вибирає один з них відповідно до прийнятих для цієї системи пріоритетних правил, наприклад, відповідно до порядку під'єднання пристроїв до вхідної магістралі каналу. Всі інші пристрої, що готові до сеансу зв'язку, мають чекати, коли настане їхня черга на обслуговування. Максимальне число ПП, які можуть бути під'єднані до МК, визначається кількістю підканалів.

Мультимплексний канал можна (умовно) поділити на *байт-мультимплексний* і *блок-мультимплексний* канали.

Байт-мультимплексний канал організовує паралельну роботу кількох ПП, які мають порівняно малу швидкодію.

Блок-мультимплексний (БлМК) канал використовується для організації паралельної роботи кількох швидкодійних ПП, дані (інформація) з яких містяться у вигляді блоків, наприклад, секторні НМД та НМЛ. БлМК має кілька підканалів і поєднує високу швидкодію селекторного каналу й можливості байт-мультимплексного каналу щодо швидкого обслуговування великої кількості периферійних пристроїв. Це досягається за рахунок того, що операції, не пов'язані з передаванням даних (пошук доріжок та записів на МД, переміщення магнітних головок і т. ін.), виконуються в мультимплексному режимі, а передавання блока інформації відбувається в монопольному (селекторному) режимі.

Кількісно робота систем введення-виведення характеризується пропускнуою здатністю КВВ – сумарною продуктивністю всіх каналів, що входять до складу комп'ютера. Ця характеристика особливо важлива при розв'язуванні економічних задач, які потребують великої кількості швидкодійних ПП.

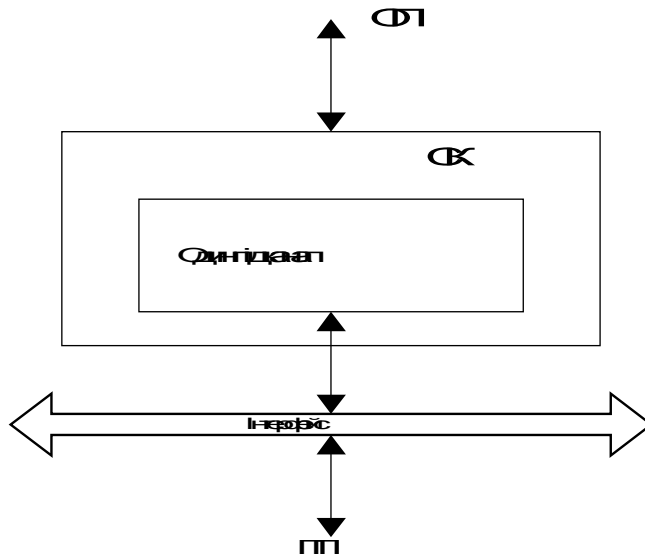


Рис. 8.3. Узагальнена структура селекторного каналу

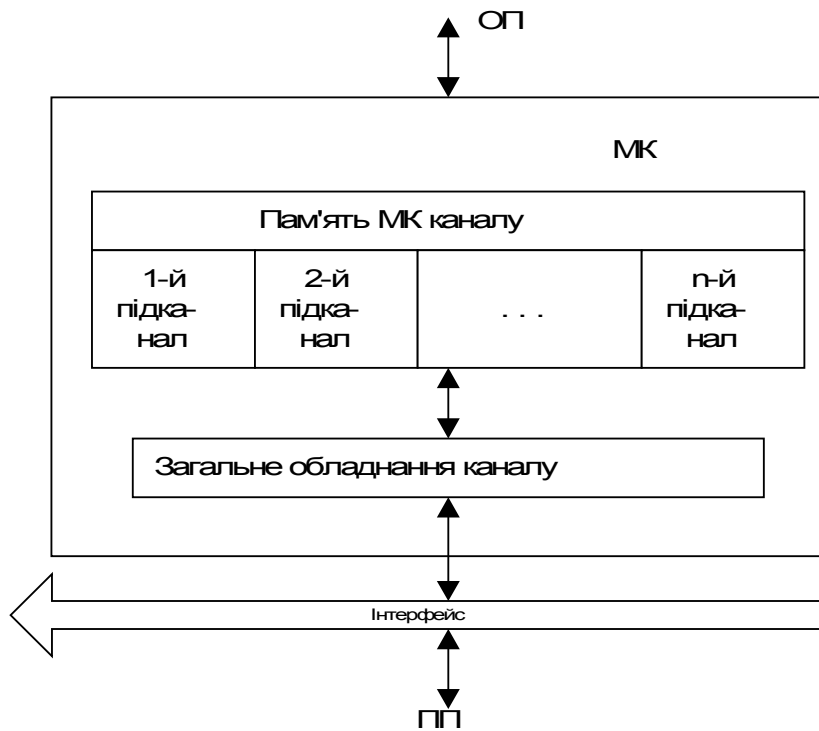


Рис. 8.4. Узагальнена структура мультиплексного каналу

8.5. Шини міні-, мікро- і персональних комп'ютерів, їхня організація й характеристика

На системній платі, а також на платах адаптерів між компонентами існують групи різних зв'язків, що називаються *шинами*, по яких для обміну даними здійснюється передавання адресних і керуючих сигналів.

Якщо інформація – це вся сукупність сигналів, що циркулюють у системі, то дані – це ті корисні відомості, які є предметом опрацювання у процесорі й контролерах і якими обмінюються у процесі виконання програм процесор, ОЗП та ПВВ.

Існують апаратні й програмні інтерфейси, що передбачають наявність фізичних і логічних ліній контролю та керування, які об'єднують різні пристрої чи програмні додатки і компоненти системи.

Елементами інтерфейсу є шина й специфікації, тобто відомості про те, як налаштована електрична частина інтерфейсу, що може знадобитися при розгляді параметрів і характеристик.

Шини можуть бути системними (system bus) та периферійними, що вказує на їхню належність до системної плати чи ПВВ. Часто шину розуміють як цілі розділи комп'ютерних технологій, наприклад, шина EISA, шина PCI, що розширює уявлення про явища і принципи, зокрема й відомості про апаратне забезпечення, наприклад, про роз'єми, лінії зв'язку, сигнали, які передаються по цих лініях, та пристрої, що посилають ці сигнали в лінії зв'язку.

Незалежно від системної шини чи архітектури введення-виведення на системній платі найчастіше трапляються такі шини:

1. *Процесорна шина*, або основна шина системи (комп'ютера), на яку виходять процесор (кілька процесорів), контролер кеш-пам'яті і сама кеш-пам'ять. Ця шина (рис. 8.5) характеризується найбільшою пропускною здатністю.



Рис. 8.5. Процесорна шина

2. *Шина пам'яті* (memory bus) призначена для передавання інформації між ОЗП та процесором, ОЗП і системною шиною (рис. 8.6). На цю шину виходять контролер шини, ОЗП та буфери для зберігання даних.

На цій шині може бути організована двопортова пам'ять. Цей термін означає, що контролер пам'яті оптимально налаштований для реакції на запити використання пам'яті від процесора і від ПВВ, які представляють два порти, апаратно організовані на двонаправлених буферних регістрах даних.

Як правило, в одній групі регістрів накопичуються дані для доступу до одного банку за парними адресами, а в іншій – для непарного банку, але можуть бути передбачені й інші ступені для доступу, наприклад, до чотирьох, а не до двох банків ОЗП. В такому разі обидва порти застосовуються однією пам'яттю, використовуючи один і той самий протокол доступу до неї. Такий підхід називається слабкою взаємодією.

3. *Системна шина* визначається системною архітектурою. Всі сучасні комп'ютери володіють комбінованими системними шинами, наприклад, ISA і PCI або EISA і PCI. Одна із шин називається первинною системною шиною (EISA, ISA), а інша (PCI) – вторинною системною шиною. Обидві системні шини визначають тактові частоти й швидкодію системи (комп'ютера), режими роботи компонентів системи, характеристики і параметри обміну даними, тип та кількість слотів розширення і встановлені в них периферійні пристрої. В такому разі системна шина поділена на дві секції з різною швидкодією, які електрично об'єднують спеціальний пристрій – електронний міст. На шині PCI можуть працювати такі пристрої, як НЖМД EIDE, графічна й аудіопідсистеми, а на шині ISA – послідовний, паралельний і комунікаційний порти.

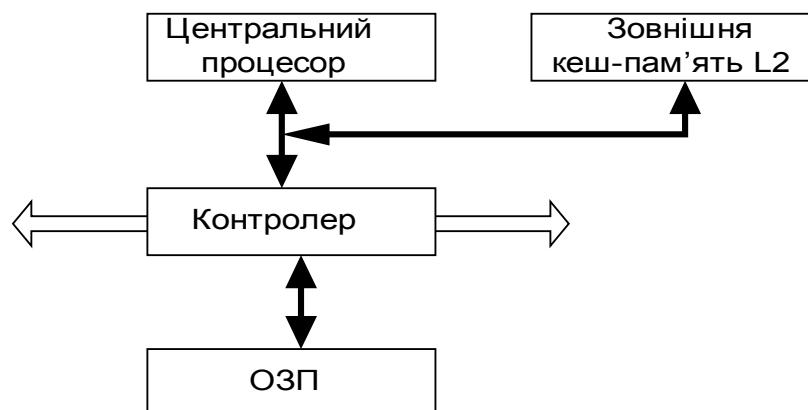


Рис. 8.6. Шина пам'яті

Шини введення-виведення

Шина введення-виведення дає змогу процесору через роз'єми для розширення його можливостей взаємодіяти з периферійними пристроями. Останнім часом було розроблено багато варіантів шин введення-виведення, які спрямовувались переважно на підвищення їхньої швидкодії та продуктивності. В сучасних комп'ютерах використовуються принципово інші шини введення-виведення порівняно з комп'ютерами більш ранніх моделей, постійно змінюється також їхня структура.

Кожний тип шин введення-виведення має свої архітектурні особливості. Основними шинами введення-виведення є:

- ISA (Industry Standart Architecture);
- MCA (Micro Channel Architecture);
- EISA (Extended Industry Standard Architecture);
- локальна шина PCI (Peripheral Component Interconnect);
- VESA (VL-Bus або VLB-VESA Local Bus);
- AGP (Accelerated Graphics Port);
- PC Card;
- Fire Wire (IEEE-1394);
- USB (Universal Serial Bus).

Головні відмінності між цими шинами полягають в обсязі даних, які можуть ними передаватися за одиницю часу, а це пов'язано з розрядністю кожної з шин і швидкістю передавання даних (швидкодією). Шина будується за допомогою спеціальних мікросхем (шинних формувачів або контролерів), що під'єднуються до шини мікропроцесора.

Локальні шини (ISA, MCA і EISA) мають порівняно низьку швидкодію. Для підвищення швидкості обміну інформацією в комп'ютері частина операцій з даними стала здійснюватися не через роз'єми шини введення-виведення, а через додаткові швидкодіючі роз'єми, які в такому разі слід розміщувати на швидкодіючій шині мікропроцесора. Таке під'єднання нагадує під'єднання зовнішньої кеш-пам'яті. Ця конфігурація одержала назву *локальної шини*, де зовнішні пристрої мають доступ до шини мікропроцесора.

Шина ISA (Industry Standart Architecture – промислова стандартна архітектура). Найбільш ранні версії шин, з яких виникла ця, одна з найпоширеніших шин (PC-bus, XT-bus), призначалась для комп'ютерів IBM PC і IBM PC XT. Це були 8-розрядні шини з тактовою частотою від 4,77 МГц до 10 МГц для процесора I8088 та 16 МГц для NEC V20. Ця шина досі широко використовується в нових моделях ПК, в яких для під'єднання периферійних пристроїв (клавіатура, модеми, ручні сканери, FDD і т. ін.) використовується стандарт ISA.

Шина MCA. Ця шина була розроблена фірмою «IBM» і призначалась для комп'ютерів класу PS/2. В ній було підвищено пропускну здатність до 20 Мб/с за рахунок збільшення тактової частоти до 10 МГц і розрядності до 32 біт. Відпала необхідність вручну конфігурувати зовнішні пристрої, встановлені у слоти розширення MCA. Однак шина MCA не набула поширення. Причиною цього стала повна її несумісність із шиною ISA і необхідність заміни материнської плати й карт розширення.

Шина EISA. Необхідність підвищення продуктивності комп'ютерів і комп'ютерних систем та забезпечення сумісності їхніх компонентів привели до подальшого розвитку і модернізації шини ISA. Завдяки провідним фірмам-виробникам апаратного забезпечення («Epson», «Hewlett-Packard», «NEC», «Compaq», «Wyse») впроваджена розширена версія шини ISA-EISA.

Шина EISA володіє такими перевагами:

- слот EISA повністю сумісний зі слотом ISA. В зв'язку з цим у слоти шини EISA можна встановлювати карти ISA, завдяки чому при переході на нову шину відпадає необхідність змінювати всі карти розширення;

- шина EISA є 32-розрядною, тобто всі 32 розряди шини даних ЦП виведені на слот, що робить можливим використання відповідних карт (мережних, графічних, жорсткого диска), які більше не гальмуються шиною ISA. Максимальна швидкість передавання даних по шині дорівнює 33 Мбайт/с;

- як і MCA, шина EISA є інтелектуальною шиною, тобто конфігурація карт розширення здійснюється не апаратними засобами, а програмно.

Шина EISA не набула значного поширення через високу вартість і відсутність у достатній кількості карт розширення EISA. Крім цього, її пропускну здатність нижча порівняно з пропускну здатністю локальної шини VESA.

Один з можливих варіантів організації шини EISA показано на рис. 8.7.

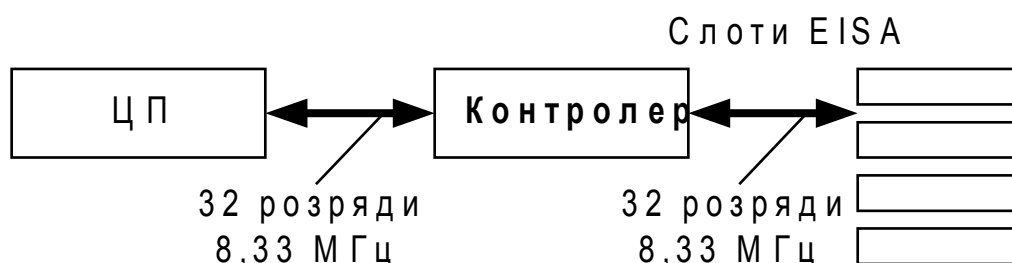


Рис. 8.7. Шина EISA

Шина VESA. Ця шина була розроблена для підвищення швидкості обміну з відеосистемою й адаптерами IDE жорстких дисків, тобто для зв'язку процесора зі швидкодіючими периферійними пристроями. Перша версія шини VL-Bus 1.0 мала ряд суттєвих обмежень:

- залежність від зовнішньої частоти синхронізації, наприклад, пристрої VL-Bus, призначені для роботи на частоті синхронізації 25 МГц, не можуть працювати на частоті 33 МГц;

- залежність від частоти роботи процесора, а отже, від його типу;

- обмежена навантажувальна здатність шини та її залежність від тактової частоти;

- обмежена підтримка високошвидкісних периферійних пристроїв;

- відсутність на шині арбітра і конфлікти, що породжуються при одночасному запиті на обмін даними.

Більш нова версія VL-Bus 2.0 усуває недоліки першої модифікації. Вона має слоти з напругою живлення 3,3 В, підтримує 64-розрядний обмін даними на частотах синхронізації 33 або 50 МГц. Теоретично швидкість передавання даних на такій шині досягає 400 Мбайт/с.

Шина PCI. Шина PCI була розроблена фірмою «Intel» в 1993 р. як альтернатива шині VL-Bus для свого нового високопродуктивного процесора Pentium.

У сучасних материнських платах тактова частота шини PCI задається як половина тактової частоти системної шини, тобто при тактовій частоті системної шини 66 МГц шина PCI буде працювати на частоті 33 МГц, а при частоті системної шини 75 МГц – на частоті 37,5 МГц. Основним принципом, взятим за основу шини PCI, є застосування так званих *мостів*, що здійснюють зв'язок між шиною PCI й іншими шинами (наприклад, PCI to ISA Bridge).

Важлива особливість шини PCI виявляється у тому, що в ній реалізовано принцип Bus Mastering, який передбачає здатність зовнішнього пристрою при пересиланні даних керувати шиною (без участі мікропроцесора). Під час передавання інформації пристрій, який підтримує Bus Mastering, захоплює шину і стає головним. При такому підході центральний процесор звільняється для виконання інших завдань на час, коли здійснюється передавання даних.

На сьогодні шина PCI стала стандартом для шин введення-виведення. Архітектуру шини PCI наведено на рис. 8.8. Основними перевагами цієї шини є:

1. У шині PCI використовується зовсім відмінний від шини ISA спосіб передавання даних. Цей спосіб полягає в тому, що у системі визначаються два пристрої: *передавальний* і *приймальний*. Коли передавальний пристрій готовий до передавання, він виставляє дані на лінії даних та супроводжує їх

відповідним сигналом, при цьому приймальний пристрій записує дані у свої регістри і видає сигнал Target Ready, підтверджуючи запис даних та готовність до приймання наступних даних.

2. Відносна незалежність окремих компонентів схеми. Відповідно до концепції PCI, передачею пакета даних керує не процесор, а включений між ним і шиною PCI міст. Процесор може продовжувати роботу й тоді, коли здійснюється занесення даних в ОЗП (чи їхнє зчитування) або при обміні даними між двома будь-якими компонентами системи.

3. Відповідно до специфікації PCI 1.0 шина PCI – 32 розрядна, а PCI 2.0 – 64-розрядна. Таким чином, смуга пропускання шини дорівнює відповідно $33 \text{ МГц} \times (32 \text{ біт} : 8) = 132 \text{ Мбайт/с}$ і $33 \text{ МГц} \times (64 \text{ біт} : 8) = 264 \text{ Мбайт/с}$.

4. Шина PCI є універсальною. Оскільки системна шина і шина PCI з'єднані за допомогою *головного моста*, то остання буде самостійним пристроєм та може застосовуватися незалежно від типу процесора.

5. Система PCI використовує принцип мультиплексування, тобто коли для передавання даних і адрес використовуються одні й ті самі лінії.

6. Важливою ознакою шини PCI є її інтелектуальність, тобто здатність розпізнавати апаратні засоби й аналізувати конфігурації системи відповідно до технології Plug&Play, яку розробила фірма «Intel».

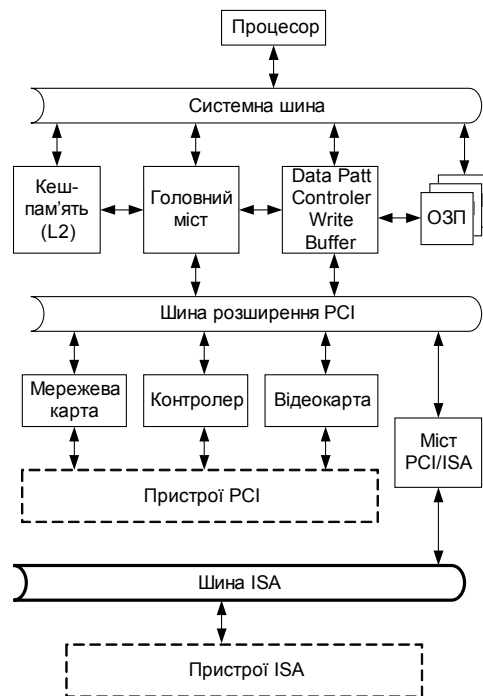


Рис. 8.8. Архітектура шини PCI

9. АРХІТЕКТУРА СУПЕРКОМП'ЮТЕРІВ

9.1. Поняття про суперкомп'ютери та їхню організацію

З початку ери комп'ютерної техніки проблема зменшення різниці між продуктивністю, необхідною для розв'язання конкретної прикладної задачі, й продуктивністю, яка забезпечується комп'ютером, створила множину альтернативних моделей комп'ютерів. Деякі з них довели свою ефективність на практиці. Однак на сучасному етапі інформаційного вибуху з порядку денного не знято питання постійного подальшого підвищення продуктивності. Наприклад, сучасні проекти потребують продуктивності близько 10^9 операцій за секунду і більше (тобто команда має виконуватися за 1 нс), що наближається до граничних можливостей тих різновидів техніки, для яких границею є швидкість поширення світла. Термін «суперкомп'ютер» вживається на позначення комп'ютера з деякими додатковими властивостями, в якому на основі досягнень сучасної науки й техніки за певних умов може досягатися максимально можлива продуктивність, і більшою чи меншою мірою стосується комп'ютерів із суперпродуктивністю, яка необхідна для розв'язання великих числових задач. Передусім до них належать задачі моделювання складних систем та явищ, що потребують великого обсягу матричних обчислень (прогнозування погоди, ядерні дослідження і т. ін.).

Для розвитку архітектури сучасних суперкомп'ютерів характерною є тенденція до збільшення апаратних і програмних функцій, які вбудовуються у комп'ютер чи систему. За основу більшості сучасних суперкомп'ютерів взято методи попереднього перегляду, функціональний паралелізм, конвеєрну обробку на різних рівнях, векторну і паралельну обробку в режимах ОКМД (один потік команд та множина потоків даних) і МКМД (множина потоків команд та множина потоків даних). У більшості суперкомп'ютерів паралельна скалярна й векторна обробка здійснюється завдяки використанню великої кількості функціональних пристроїв, що входять в однопроцесорну чи багатопроцесорну систему.

Паралелізм слід розуміти як одночасну обробку завдань, кроків завдань, програм, підпрограм, циклів чи операторів (рис. 9.1).

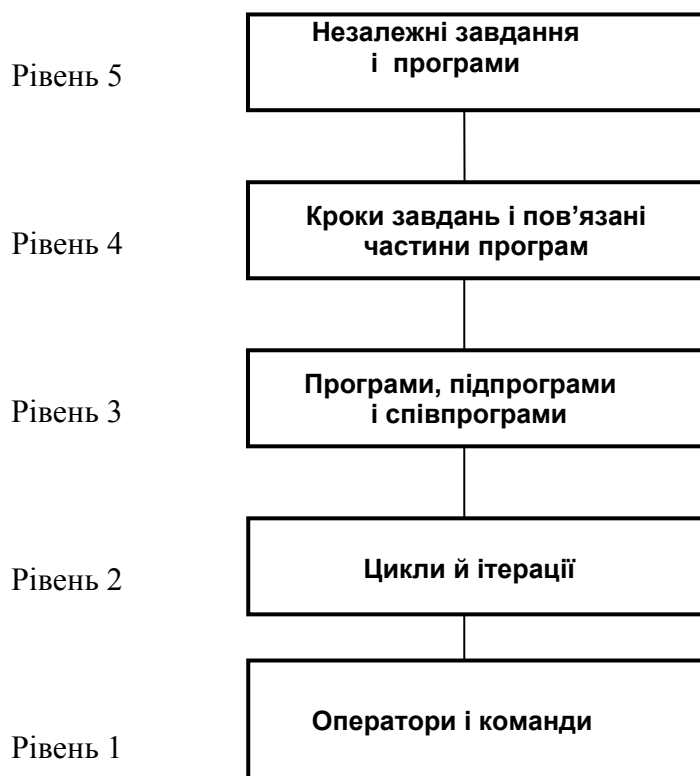


Рис. 9.1. П'ять рівнів паралелізму при виконанні програм

Чим нижчий рівень, тим дрібнішим є дроблення програмних процесів. У загальному випадку паралельна обробка полягає в реалізації паралелізму на будь-якому з цих рівнів або в їхньому поєднанні.

Векторна обробка полягає у паралельній обробці ітерацій циклів на рівні 2. У багатьох комп'ютерах, за основу яких взято метод попереднього перегляду і які містять велику кількість функціональних пристроїв, здійснюється паралельне виконання незалежних скалярних операторів на рівні 1. Більшість сучасних комп'ютерів допускає *мультипрограмування*, яке передбачає розподіл ресурсів процесора між кількома незалежними процесорами. Подібна процедура відбувається і в однопроцесорних системах, де використовується суміщення роботи ЦП та пристроїв введення-виведення.

Багатопроцесорна обробка – це спосіб паралельної обробки, при якому забезпечується мультипрограмний режим роботи двох і більше процесорів, які взаємодіють між собою. При незалежній однопроцесорній обробці знаходить застосування паралелізм на рівні 1, який реалізується шляхом використання кількох режимів типу ОКОД (один потік команд і один потік даних).

Багатозадачний режим – це спеціальний випадок багатопроцесорної обробки, коли програмним процесом (задачею) є крок завдання чи підпрограма на рівнях 3 і 4.

9.2. Класифікація суперкомп'ютерів

Паралельні комп'ютери часто поділяють, згідно з класифікацією Флінна, на комп'ютери типу ОКМД і типу МКМД. Однак така класифікація не є досконалою: існують суперкомп'ютери, які не входять до цієї класифікації; наявні важливі ознаки, які у цій класифікації не враховано.

У дослідженнях деяких вчених, які працюють у галузі паралельних обчислень, запропоновано також інші системи класифікації суперкомп'ютерів. У науковій праці (Головкин Б. А. Параллельные вычислительные системы. – М.: Наука, 1980. – 520 с.) подано деталізацію класифікації Флінна, яка поділяє основні класи за кількома рівнями: порозрядна і послідовна обробка; обробка з низькою та високою зв'язністю; однорідні і неоднорідні системи; системи зі зв'язком канал-канал, пам'ять-пам'ять та процесор-процесор; системи зі зв'язком «спільна шина», «множина шин» і з перехресними зв'язками з комутатором. Існують також інші варіанти щодо деталізації класифікації Флінна, але вони, як правило, не йдуть далі від уточнення чотирьох основних класів: ОКОД, ОКМД, МКОД та МКМД.

Інша запропонована класифікація дає змогу поділити всю множину систем на три групи: із зосередженою, розподіленою і зосереджено-розподіленою пам'яттю.

Ще однією класифікаційною ознакою служить ступінь жорсткості архітектури. За цією ознакою виділяють системи з жорсткою і програмованою архітектурою. Програмована архітектура передбачає можливість налагодження багатопроцесорних, у тому числі мультипроцесорних, систем на підклас задач конкретного користувача.

Архітектура сучасних суперкомп'ютерів може бути класифікована за п'ятьма групами залежно від способу організації міжпроцесорних зв'язків і прийнятою структурою схеми (табл. 9.1).

Таблиця 9.1

Класифікація суперкомп'ютерів за типом архітектури

Архітектура	Приклади систем
Однопроцесорні системи з кількома функціональними пристроями і факультативними апаратними засобами векторної обробки	IBM 3090; CDC 7600; Convex C-1; FPS 164/264/364; Crey 1; Crey X-MP/1; Cyber 205; Fujitsu VP-50, 100,200,400; NEC SX-1, SX-2, SCS-40
Процесорні матриці або приєднані процесори з архітектурою ОКМД	Loral MPP; ICL/DAP; FPS 164/MAX; Connection Machine; IBM-GF11
Багатопроцесорні системи зі спільною пам'яттю	Grey X-MP/2,4; Grey 2; Alliant FX /8; Elsxi 6400; Stquent 8000; IBM 3090/400 VF
Багатомашинні системи з розподіленою пам'яттю	iPSC; Ametek 14; BBN Butterfly; Culler PSC; Warp
Ієрархічні і реконфігуровані системи	Cedar; ETA-10; IBM RP3; Remps

1. Багатоконвеєрні однопроцесорні системи

Більшість векторних суперкомп'ютерів у найпростішому випадку є конвеєрними однопроцесорними системами, які вміщують кілька функціональних пристроїв. Архітектура типу „пам'ять-пам'ять” потребує великої пропускної здатності від пам'яті і команд великої довжини, що дає вигоду лише при обробці довгих векторів. Для коротких векторів чи скалярних величин продуктивність такої системи може бути надто низькою. І навпаки, архітектура типу «регістр-регістр» є більш продуктивною при обробці суміші скалярних та векторних даних.

2. Процесорні матриці з архітектурою типу ОКМД

Ці матриці є процесорами паралельного типу, які працюють синхронно під керуванням спільного пристрою керування. Фізично процесорні елементи утворюють процесорну матрицю. Оскільки комп'ютери цього типу часто використовуються для обробки масивів даних великої розмірності, вони також називаються матричними процесорами. Більшість матричних процесорів з архітектурою типу ОКМД є спеціалізованими комп'ютерами, які застосовуються переважно для обробки сигналів і зображень.

3. Багатопроцесорні системи зі спільною пам'яттю

Ці системи є сильнопов'язаними комп'ютерами з архітектурою МКМД, в яких кілька процесорів спільно використовують одну пам'ять. Архітектура в цих системах поділяється на два основних класи: зв'язок через шину і безпосередній зв'язок. У більшості міні-суперкомп'ютерів кілька процесорів, пристрої пам'яті паралельного типу, мережні інтерфейси та контролери пристроїв під'єднані до однієї з'єднувальної шини. Різновидами архітектури з безпосередніми зв'язками є матричні комутатори, частково пов'язані графи і багатокаскадні мережі. Подібну архітектуру зв'язку має більшість дорогих моделей суперкомп'ютерів.

4. Багатомашинні системи з розподіленою пам'яттю

Цей клас суперкомп'ютерів охоплює слабопов'язані системи з архітектурою МКМД і з розподіленими локальними ЗП, під'єднаними до множини процесорних вузлів. До поширених топологій зв'язку зараховують гіперкуб, кільце, комутатор типу «метелик», гіпердерева і гіпермережі. Основний спосіб зв'язку між обчислювальними вузлами у багатокомп'ютерній системі – це передача повідомлень. Ефективність зв'язку й зв'язність апаратних засобів вважаються основними факторами, які враховуються при виборі архітектури системи за критерієм економічної ефективності.

5. Ієрархічні та реконфігуровані суперкомп'ютери

Ці комп'ютери мають комбіновану архітектуру, яка містить пам'ять, що використовується спільно, і засоби передачі повідомлень для між процесорного зв'язку. До цієї групи належать кілька багатопроцесорних систем, наприклад, Cedar і ETA-10. До складу останньої входить ієрархічна система пам'яті. Прийнятий підхід полягає у використанні макропотоків даних на рівні груп процесорів та власного керуючого потоку в кожному процесорі. Таким чином, паралелізм простежується на кількох рівнях, у тому числі й з різними обчислювальними моделями.

9.3. Типові архітектури суперкомп'ютерів

Архітектура суперкомп'ютерів зі спільною пам'яттю

Архітектура комп'ютерів зі спільною пам'яттю передбачає організацію доступу до одних і тих самих даних спільної пам'яті з боку кількох процесорів (рис. 9.2). Перевага такого підходу полягає в тому, що він дає змогу кожному процесору бачити задачу, яка вирішується, загалом, а не тільки ті її частини, над якими він безпосередньо працює. Оскільки існує тільки одна шина, яка пов'язує процесори з пам'яттю, виникає серйозна проблема пропускної здатності цього каналу обміну з пам'яттю.

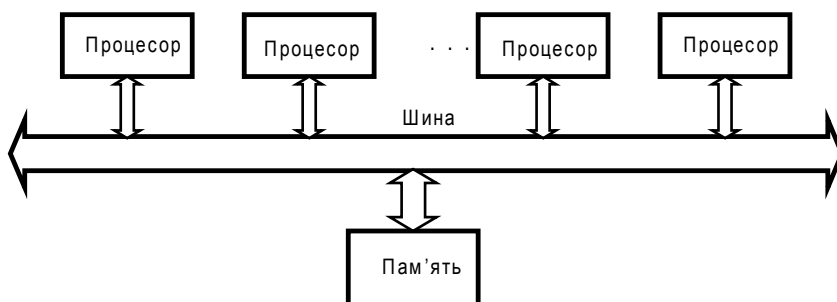


Рис. 9.2. Архітектура суперкомп'ютера зі спільною пам'яттю

Просте застосування більш швидкої пам'яті не вирішує проблеми. При цьому проблема залишається, а вартість системи збільшується.

Один зі способів вирішення подібної проблеми – використання архітектури з кеш-пам'яттю (рис. 9.3). За наявності багатьох кеш-пам'ятей і центральної (основної) пам'яті виникає проблема когерентності, тобто відповідності місткостей кеш-пам'ятей та основної пам'яті.

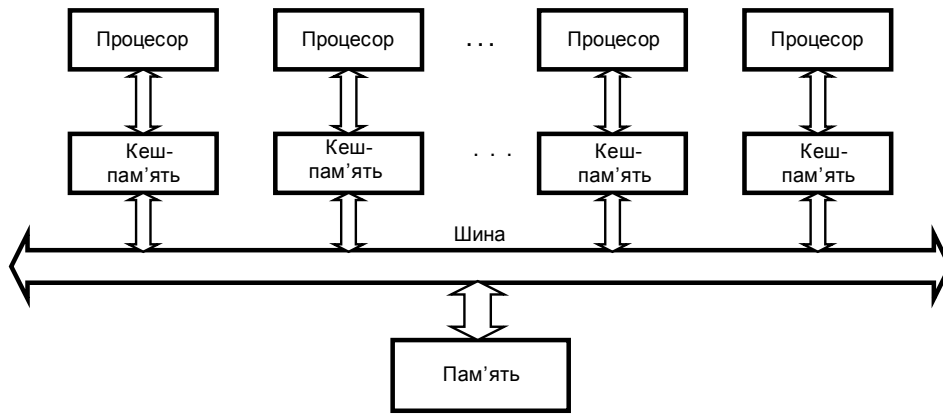
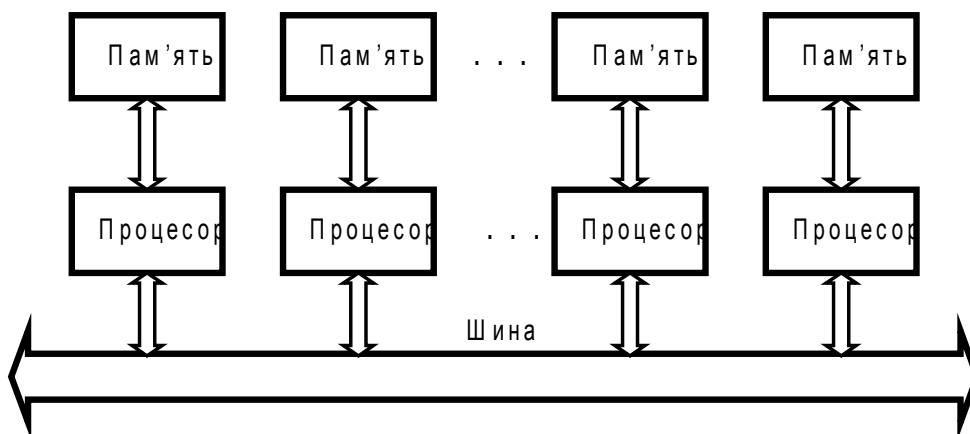


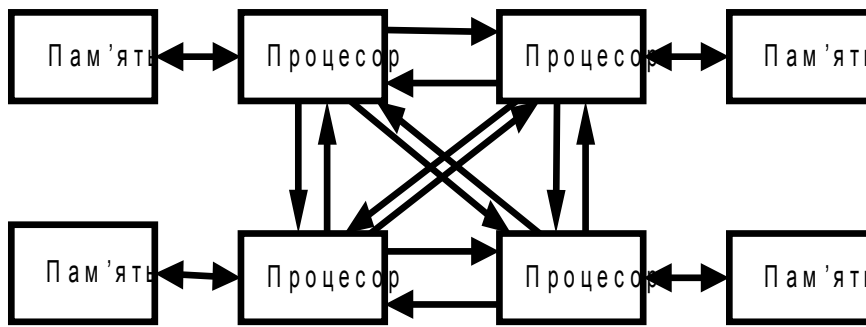
Рис. 9.3. Архітектура суперкомп'ютерів з використанням кеш-пам'яті

Будь-які зміни у місткості своїх кеш-пам'ятей іншими процесорами мають відобразитися в кожній кеш-пам'яті і в основній пам'яті.

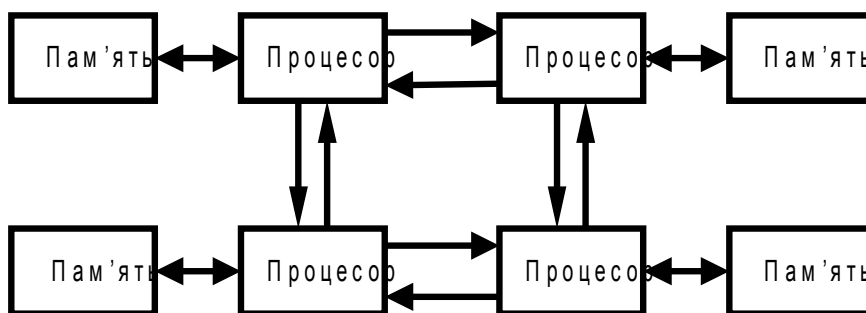
Інший спосіб вирішити проблему ширини смуги пропускання каналу обміну з пам'яттю – це взагалі відмовитися від центральної основної пам'яті. В паралельних системах класу так званих багатомашинних обчислювальних комплексів кожен процесор має власну пам'ять. Процесори взаємодіють різними способами за допомогою шини або по каналах зв'язку з усіма іншими процесорами чи тільки з деякими з інших процесорів (рис. 9.4).



а)



б)



в)

Рис. 9.4. Архітектура суперкомп'ютерів зі слабпов'язаними зв'язками: а) зі зв'язками через шину; б) з повними перехресними зв'язками; в) з фіксованою топологією

В ідеалі було би добре, якби кожен процесор мав прямі канали зв'язку з кожним іншим процесором, але це практично неможливо здійснити (в крайньому разі у найближчий час) з огляду на велику вартість такого рішення, оскільки для створення мережі з N процесорів необхідно $N-1$ каналів зв'язку.

Певним компромісом є мережа з фіксованою топологією (рис. 9.4, в), в якій кожен процесор з'єднується з деякою підмножиною своїх сусідів-процесорів. Якщо двом процесорам, які не мають прямого каналу зв'язку, необхідно взаємодіяти, то вони передають повідомлення через допоміжні процесори. Перевага такого підходу полягає в тому, що він не накладає явного обмеження на кількість додаткових процесорів, які можна під'єднати до системи. Його ж недолік виявляється в тому, що цей підхід потребує оптимізації коду прикладних програм, щоб забезпечити

виконання процесів, яким необхідні активні взаємодії, на сусідніх процесорах. Одним з можливих методів побудови таких комбінованих архітектур є конфігурація з комутацією, коли кожен процесор має свою локальну пам'ять, а з'єднуються процесори за допомогою комутатора (рис. 9.5).

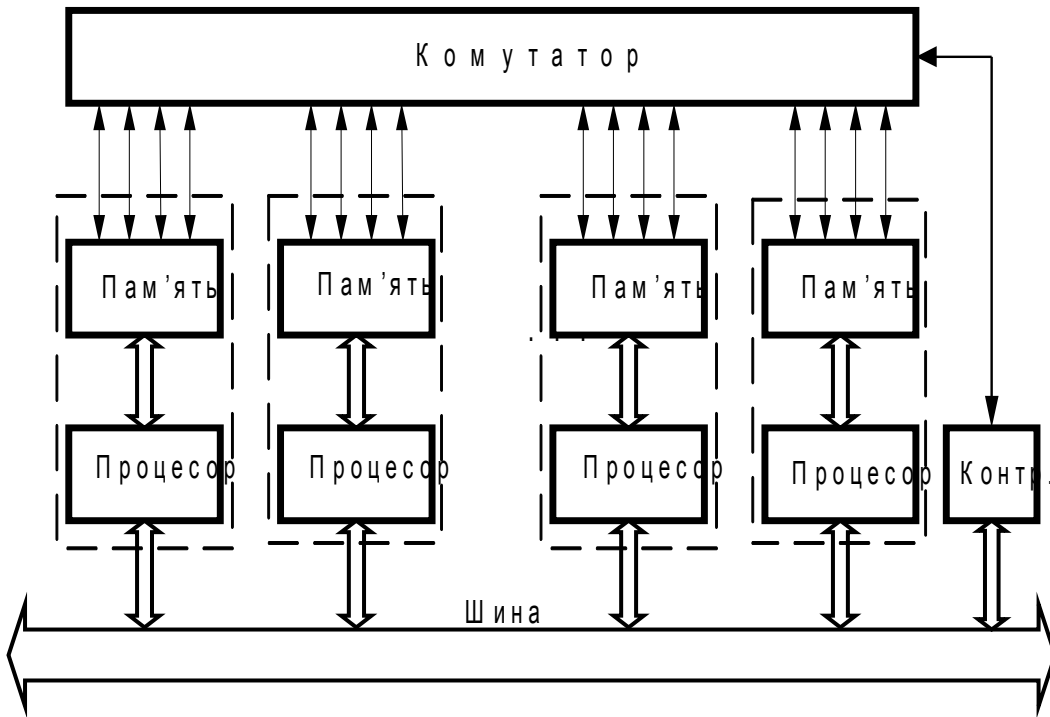


Рис. 9.5. Архітектура суперкомп'ютерів з комутаторами

10. АРХІТЕКТУРА НЕНЕЙМАНІВСЬКИХ КОМП'ЮТЕРІВ

10.1. Рекурсивна архітектура

Поряд з традиційними найманівськими машинами існують альтернативні підходи до розробки архітектур комп'ютерів. Однією з таких архітектур є рекурсивна архітектура побудови комп'ютерів.

Рекурсивна архітектура комп'ютера дає змогу по-новому організувати узгоджену роботу процесорів у мультипроцесорному комп'ютері з використанням принципу макроконвеєра. Вперше у світі ці ідеї були сформульовані і представлені радянськими вченими на конгресі у Стокгольмі в 1974 р. У цьому класі машин доведено можливість побудови такої обчислювальної системи, яка забезпечує зростання продуктивності прямо пропорційно до кількості об'єднаних у макроконвеєр процесорів. Структурна і програмна організація рекурсивних обчислювальних машин (РОМ) визначається такими основними принципами:

а) внутрішня мова РОМ охоплює рекурсивно визначені програмні елементи як узагальнення машинних команд і елементи даних як узагальнення машинних слів – операндів, які можуть мати довільну складність;

б) в РОМ здійснюється рекурсивно-паралельне керування виконанням програми, причому порядок виконання програмних елементів може задаватися неявно за допомогою функціональних (у загальному випадку – рекурсивних) відношень і визначається у процесі виконання програми;

в) внутрішня пам'ять РОМ складається з блоків, що можуть містити комірки, чи з блоків з програмними зв'язками між ними, які можуть перенастроюватися;

г) зовнішня фізична структура однозначно визначається за допомогою кінцевого числа рекурсивних відношень;

ґ) внутрішня структура, що визначає зв'язки між обчислювальними процесами в РОМ, є гнучкою, програмно перебудовується і динамічно відображає структуру задач, які розв'язуються.

Ці принципи вважають загальними, й вони дають змогу будувати різні комп'ютери.

В одному з варіантів структура РОМ подається у формі обчислювальної мережі. У вузлах мережі розміщені процесори різних типів, термінали, зовнішні пристрої і рекурсивні машини нижчих рівнів ієрархії. Внутрішня мова РОМ орієнтована на паралельну реалізацію обчислень з децентралізованим керуванням; розпаралелювання в РОМ здійснюється не зовнішніми засобами за допомогою складних операційних систем, а шляхом інтерпретації внутрішньої мови. Порядок виконання операторів у РОМ визначається динамічно, залежно від готовності операторів до виконання. Водночас готовність операторів до виконання визначається за наявністю всіх операндів

операторів, які розглядаються. До складу ROM входять дві основні частини: комутаційне й операційне поля. Перше поле складається з однакових чи різних комутаційних процесорів, з'єднаних між собою в деяку рекурсивну структуру. Раціональна багаторівнева структура подається у вигляді піраміди з деревоподібними зв'язками між рівнями і матричними зв'язками між елементами кожного рівня. Операційне поле складається з керуючих та виконавчих процесорів і зовнішніх пристроїв. Елементи операційного поля, як правило, не мають безпосередніх зв'язків між собою й підключаються до елементів комутаційного поля нижнього рівня. Можливе підключення також до елементів комутаційного поля вищих рівнів. З'єднання між елементами операційного поля реалізується динамічно через елементи комутаційного поля за допомогою налагодження і переналагодження зв'язків між ними. Внутрішня пам'ять ROM розподілена між процесорами, але будь-який операційний процесор може використовувати пам'ять інших операційних процесорів. Програми оперують з віртуальною пам'яттю. Адресація даних та програмних елементів здійснюється тільки за допомогою індексів й імен (реалізована так звана багаторівнева лексична адресація). Перехід до фізичних адрес відбувається тільки в межах пам'яті свого процесора у процесі інтерпретації програми. Операційна система ROM має таку саму структуру внутрішньою мовою машини, як й інші програми. Програми користувачів розглядаються як внутрішні блоки операційної системи. Багато стандартних функцій операційної системи належать до функцій інтерпретаторів елементів внутрішньої мови машини завдяки її високому рівню і рекурсивно-паралельному способу керування.

Можливості переналагодження структури ROM та контролю комутаційними процесорами суміжних з ними операційних і комутаційних процесорів визначають високу живучість; глибока структурованість програм внутрішньою мовою ROM забезпечує локалізацію програмних помилок та підвищує надійність програмного забезпечення. У великих ROM до виконавчих процесорів можуть належати матричні чи магістральні процесори. Важливо, що при нарощуванні однієї ROM чи при об'єднанні двох ROM непотрібно змінювати структуру машин. При цьому програма використовує стандартну версію внутрішньої мови і не залежить від структури та складу конкретної ROM.

10.2. Машина Айліфа: її особливості і практичне застосування

Машина Айліфа визначена тільки на функціональному рівні. Її відмінності від нейманівської машини є такими:

1. Найманівський принцип зберігання в пам'яті програми передбачає, що коди слів інформації не залежать від типу інформації; програміст повинен стежити за тим, щоб опрацюванню даних певного типу відповідали команди певного типу (наприклад, у наявності є різні види команд для опрацювання цілих і дійсних чисел); якщо це не враховувати, то виникають курйозні помилки. Айліф пропонує тип інформації відображати в кодах даних. Це приводить до скорочення списку машинних операцій та зменшення кількості програмних помилок.

2. Пам'ять найманівської машини суто лінійна: це послідовність комірок пам'яті з адресами 0, 1, ... F. При будь-яких зв'язках між даними програміст повинен відобразити ці дані на лінійну послідовність адрес. Потім при складанні програми доводиться знаходити спосіб виділення адрес, що відповідають окремим структурним одиницям даних. Айліф пропонує вносити опис структури інформації безпосередньо у пам'ять машини. Це забезпечує автоматичне виявлення адрес окремих елементів у процесі роботи машини. Виграш полягає у спрощенні не тільки процесу програмування, а й самих програм. Це зменшує кількість програмних помилок.

Додаткові можливості машини Айліфа закладаються шляхом ускладнення апаратури, але при цьому знижуються затрати на програмування.

На основі ідей Айліфа працюють машини «ЭЛЬБРУС» і «Burroughs» (США). Особливостями цих машин є такі:

а) рівень внутрішньої мови машини наближається до рівня вхідної алгоритмічної мови, що спрощує не тільки трансляцію програм, а й сам транслятор (розробка транслятора може бути доручена менш кваліфікованому програмісту);

б) суто магазинна, стекова організація. Принципу стекової організації підпорядковані вся структура ЦП і система програмування. Через стек відбувається обмін параметрами між процедурами; у стеку запам'ятовується стан тієї процедури, яка звернулася до іншої; стек використовується для реалізації вкладених переривань і відновлень;

в) кожне слово пам'яті має групу додаткових розрядів (тегів), що ідентифікують інформацію в комірці. Теги вказують: чи дане слово є операндом чи змінною, яке значення зберігається в даному слові (ціле, дійсне, подвійної точності і т. ін.), і нарешті, теги вказують на дозвіл (заборону) звернення до даного слова;

г) вводиться новий об'єкт – дескриптор (описувач). Дескриптор є подальшим розширенням поняття непрямой адресації. В ньому поміщається не тільки посилання на програмний об'єкт, а й опис цього

об'єкта (команда чи дані). Формат дескриптора машини Айліфа наведено на рис. 10.1.

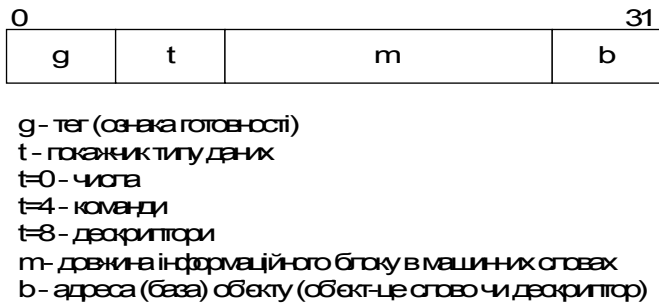


Рис. 10.1. Формат дескриптора машини Айліфа

Слід зазначити, що при готовності даних у машині Айліфа простежується одна з двох ситуацій: дані впізнані машиною; дані не впізнані машиною (або даних немає в пам'яті). У другому випадку вимушено включається апарат переривань.

Схему вибірки інформаційних елементів із пам'яті машини Айліфа подано на рис. 10.2.

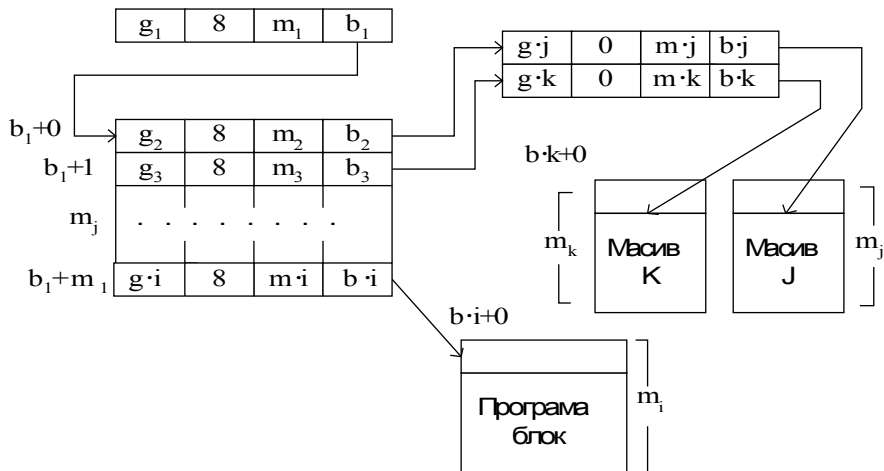


Рис. 10.2. Організація інформаційних елементів у пам'яті машини Айліфа

10.3. РЕФАЛ: мова і машина

Основою при створенні у СРСР (1968 р.) нової алгоритмічної системи РЕФАЛ послужили нормальні алгоритми Маркова. Скорочення РЕФАЛ означає «рекурсивних функцій алгоритмічна мова». Відома реалізація мови РЕФАЛ на машині ЄС ЕОМ.

РЕФАЛ-мова:

а) алфавіт – сукупність об'єктних символів / А, Б, В, ..., Я, А, В, С, D, ..., Z, α , β , ..., ω , 0, 1, 2, ..., 9, .../;

б) алгоритм – сукупність об'єктних символів і власних символів РЕФАЛу;

в) власні символи:

§ – початок пропозиції (підстановки – за Марковим);

k – конкретизаційна дужка;

⊥ – конкретизаційна крапка;

' – утворення складових символів (тобто ієрогліфів), наприклад: 'ЯКЩО', 'IF', 'D/DX';

() – структурні дужки;

~ – розділювач лівої і правої частин речення;

ets – рядкові символи, що називаються власними змінними.

Порівняно з іншими мовами номенклатура власних символів вважається невеликою;

г) операція конкретизації є єдиною і полягає у переході від імені до значення. Приклади конкретизації наведено на рис. 10.3;

Ім'я	Значення	Примітка
$kx\perp$ $k7 + 2\perp$	$X=25$ 9	
$k(a + b)(a - b)\perp$	$a^2 - b^2$	a і b – невідомі
$k(a^2 - b^2)\perp$	12	a і b – відомі
$k(a^2 - b^2)\perp$	$a^2 - 4$	a – невідоме b – відоме

Рис. 10.3. Приклади конкретизації РЕФАЛу

д) структура даних:

$k\ ssss\ \dots\ ss\ \perp$
дані

Дані є не просто послідовністю, оскільки вони мають виражену структуру. Структуру даних наведено на рис. 10.4;

е) вираз – це або пустий вираз, або терм. Терм – це символ: А, В, /, але не АВ чи А/В;

ж) речення (за Марковим – підстановка):

§ 2.1.8	$k(a + b)(a - b)\perp$	$\sim k(a^2 - b^2)\perp$
Коментар	типовий вираз	вираз, що підставляється найчастіше
	↓	
номер пропозиції	P →	Q – в термах Маркова

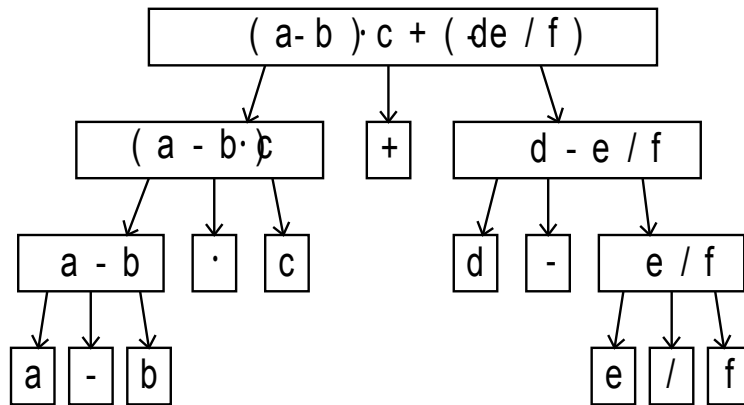


Рис. 10.4. Структура даних

Структуру РЕФАЛ-машини подано на рис. 10.5.

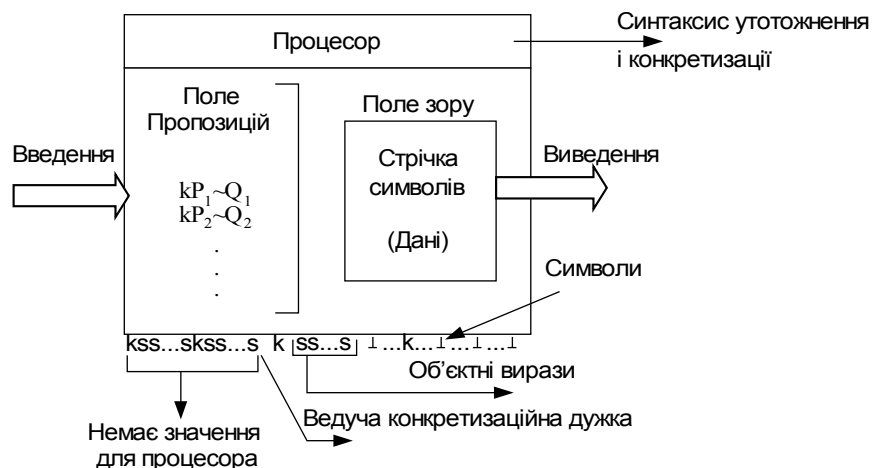


Рис. 10.5. Структура РЕФАЛ-машини

- Порядок роботи РЕФАЛ-машини може бути, наприклад, таким:
- завантажується алгоритм у формі пропозицій і дані;
 - процесор відшукує ведучу конкретизаційну дужку;

– типові вирази P_1, P_2, \dots утотожуються з об'єктним виразом. При збігу об'єктний вираз замінюється на Q_i . Після цього відбувається перехід на пункт другий;

– якщо немає конкретизуючої дужки або утотоження неможливе, то проводиться виведення даних з поля зору.

10.4. Асоціативні паралельні процесори

Асоціативні паралельні процесори (АПП) належать до класу ненеіманівських машин. Відмінності між АПП і ненеіманівськими машинами найсильніше виявляються при розгляді пам'яті.

Найширше використовуються три способи організації пам'яті:

– лінійно адресована пам'ять, повністю ізольована від пристрою опрацювання інформації у тому значенні, що така пам'ять самостійно не проводить перетворення інформаційних структурних одиниць;

– пам'ять з магазинною адресацією (стек), де допускається суміщення функцій зберігання й обробки інформаційних структурних одиниць. Однак таке суміщення має штучний характер у тому значенні, що у вершину стека вбудовується пристрій, який опрацьовує інформацію. Важливо, що стек реалізує нуль-адресну систему команд, яка за необхідності впливає з дисципліни обслуговування стека (FIFO – перший прийшов, перший вийшов чи LIFO – останній прийшов, перший вийшов);

– асоціативна пам'ять, в якій кожна комірка окремо і одночасно всі комірки разом не тільки зберігають інформаційні структурні одиниці, а й опрацьовують інформацію, яка зберігається. Взагалі немає необхідності доповнювати асоціативну пам'ять пристроєм опрацювання інформації, тобто в пам'яті цього типу чітко реалізується принцип розподіленого опрацювання через злиття функцій зберігання і опрацювання інформації.

Пам'ять з лінійною адресацією використовується, наприклад, у машинах сімейства ЕС ЕОМ. Пам'ять з магазинною адресацією реалізована в машинах СМ ЕОМ та є основним способом організації пам'яті у багатопроесорних обчислювальних системах «ЭЛЬБРУС-1» і «ЭЛЬБРУС-2».

Відоме використання асоціативної пам'яті для прискорення перетворення віртуальних адрес на фізичні в машинах сімейства ЕС ЕОМ-2. У цьому типі комп'ютерів асоціативна пам'ять використовується як кеш-пам'ять (швидкодіюча буферна пам'ять великої ємності) для різкого скорочення числа звернень з боку процесора до основної пам'яті. При цьому значно підвищується швидкодія машини загалом.

Одну з можливих структур асоціативного паралельного процесора наведено на рис. 10.6.

Інформація і команди передаються від центрального пристрою керування до кожної комірки пам'яті. Передача відбувається паралельно. Наприклад, центральний пристрій керування повідомляє у комірки, щоб вони порівняли власну інформацію з компарандом. Компаранд – це операнд в операції порівняння. Кожна комірка зв'язана з центральним пристроєм керування теговим розрядом T_i .

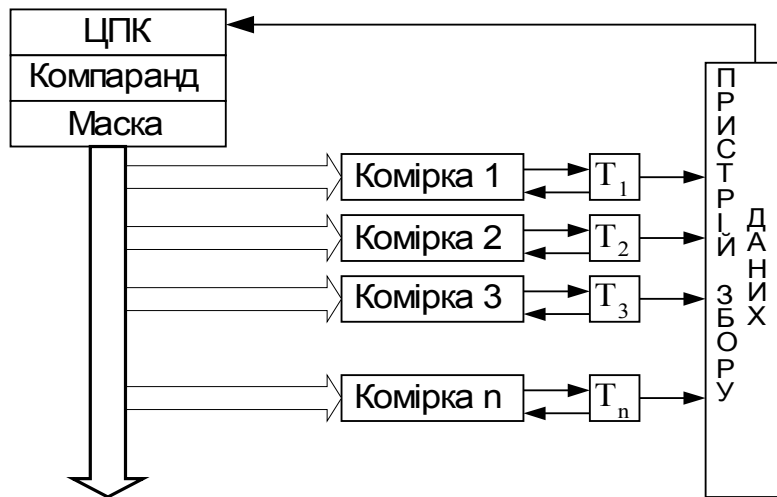


Рис. 10.6. Структура простого асоціативного процесора

Сукупність тегових розрядів називається пам'яттю реакції. В наявності є команда SET, що може бути видана центральним пристроєм керування для встановлення всіх тегових розрядів в одиницю. Коли центральний пристрій керування видає команду COMPARE (порівняти), будь-яка комірка, що містить слово, яке не збігається з компарандом, буде формувати сигнал. Цей сигнал скидає теговий розряд. Після команди COMPARE тільки ті комірки, місткість яких дорівнює компаранду, будуть залишатися з теговими розрядами, встановленими в одиницю. До складу центрального пристрою керування входять два регістри: компаранда і маски. У разі, коли непотрібен точний збіг елемента зі всіма розрядами компаранда, використовується регістр маски. Регістр маски має одиницю у тих розрядах, в яких здійснюється порівняння компаранда з місткістю комірок. В інших регістрах маски записуються нулі.

11. АРХІТЕКТУРА НЕЙРОКОМП'ЮТЕРІВ

11.1. Визначення поняття «нейрокомп'ютер» і передумови його виникнення

У середині 80-х рр. ХХ ст. у США, а потім в Японії та країнах ЄС були розгорнуті широкомасштабні національні і міжнародні програми досліджень та розробок, спрямовані на створення нейрокомп'ютерів – комп'ютерів на основі штучних нейронних мереж, які володіють розвиненим інтелектом і програмуються шляхом навчання на прикладах розв'язання задач.

У 1995 р. було завершено розробку першого нейрокомп'ютера на стандартній мікропроцесорній елементній базі.

Нейрокібернетика – це широке поле наук і технологій, пов'язаних з розумінням пристрою нейронних систем та застосуванням отриманих знань у техніці й різних галузях науки і виробництва.

У науковому розумінні *нейрокібернетика* – це наука, що вивчає механізми управління, які відбуваються в мозку живих організмів. Своїм народженням передусім вона зобов'язана розвитку фізіології і науки про мозок. (В 1911 р. Р. Казал висунув ідею про нейрон як цеглинку побудови мозку).

На сьогодні відомо, що людський мозок містить близько ста мільярдів нейронів, кожен з яких має від тисячі до десятиох тисяч зв'язків. В останні роки у результаті динамічного розвитку обчислювальної техніки, теорії хаосу і теорії самоорганізації, а також на підставі досягнень синергетики й теорії дисипативних структур (структур, фазовий обсяг яких зменшується з часом) спостерігається якісний бум у розвитку *нейрокомп'ютерних технологій*.

У світі існує кілька десятків спеціалізованих фірм, що випускають продукцію в галузі нейроінформатики. До того ж багато спеціалізованих комп'ютерних фірм («IBM», «Siemens Nocsdorff», «Mitsubisi») ведуть дослідження і мають власні розробки в цій галузі. Фірма «Siemens» останнім часом випускає спеціальні *нейрочіпи*. Ці пристрої складаються з великої кількості нейропроцесорів, здатних на відміну від звичайних процесорів робити послідовно-паралельні обчислення. Така схема обчислень пов'язана з особливістю роботи головного мозку людини, аналогом якого і є нейрочіпи.

За основу побудови нейрокомп'ютерів взято штучний нейрон (*перцептрон*). Кожний нейрон отримує сигнали від сусідніх нейронів за допомогою спеціальних нервових волокон. Ці сигнали можуть бути збудливими або гальмівними. Їхня сума утворює електричний потенціал у середині тіла нейрона. Коли потенціал перевищує деякий поріг, нейрон переходить у збуджений стан і посилає сигнал по вихідному нервовому

волокну. Окремі штучні нейрони з'єднуються один з одним шляхом використання різних методів. Це дає змогу створювати різноманітні нейронні мережі з різною архітектурою, правилами навчання і можливостями.

Нейрокомп'ютер – це обчислювальна система з архітектурою MSIMD, в якій реалізовано два принципових технічних рішення: спрощено до рівня нейрона процесорний елемент однорідної структури і різко ускладнено зв'язки між елементами; програмування обчислювальної структури перенесено на зміну вагових зв'язків між процесорними елементами.

Загальне визначення нейрокомп'ютера можна подати таким чином.

Нейрокомп'ютер – це обчислювальна система з архітектурою апаратного і програмного забезпечення, адекватною до виконання алгоритмів, поданих у нейромережевому логічному базисі.

11.2. Архітектурні особливості й апаратне забезпечення нейрокомп'ютерів

Ідею побудови автомата на основі порогових елементів, подібних до нейронів (нервових клітин), здатних виконувати логічні функції, сформулювали більш як півстоліття тому Мак-Каллок і Піттс. Однак завдання проектування систем на основі порогових елементів викликало великі труднощі і спосіб його виконання було знайдено лише 20 років по тому. Це було настільки складно, що практично відкидало можливість синтезу автоматів, які складалися більш як з десятків нейронів.

Системи на основі порогових елементів отримали назву *штучних нейронних мереж (ШНМ)*. Перші працездатні ШНМ – перцептрон Ф. Розенблатта й система «Альфа» А. Г. Іваненка – були створені вже в кінці 50-х рр. XX ст. Перші великі перцептрони на основі аналогової та цифрової техніки («Адам-А» і «Адам-Д») за межами США були створені протягом 1969–1971 рр. в одному з київських НДІ.

Схему найпростішого перцептрона Розенблатта подано на рис. 10.1. Він вміщує три шари порогових елементів. Вхідні сигнали (стимули), діючи на рецептори (S-елементи), переводять їх у збуджений стан. S-елементи випадково зв'язані із сукупністю асоціативних нейронів (A-елементів). Вихід A-елемента відрізняється від нуля тільки тоді, коли збуджено достатньо велику кількість зв'язаних з ним рецепторів. Реакції A-елементів надходять на входи ефекторів (R-елементів) через зв'язки, ваги яких змінюються при навчанні. В ефекторах обчислюється постсинаптичний потенціал – врівноважена сума сигналів, які надійшли. Як правило, у перцептроні для кожного запам'ятовуваного образу

виділяється один ефектор, і рішення приймається за максимальним значенням постсинаптичного потенціалу.

Властива перцептрону Розенблатта неоднорідність структури (розділення на S-, A- і R-елементи) у більш пізніх моделях ШНС втрачається.

На рис. 11.2 наведено модель штучного нейрона. Штучний нейрон імітує в першому наближенні властивості біологічного нейрона. На вхід штучного нейрона надходить деяка множина сигналів, кожний з яких є виходом іншого нейрона. Кожний вхід збільшується на відповідну вагу, аналогічну до синаптичної сили, і всі добутки сумуються, визначаючи рівень активації нейрона. Хоча мережні нейропарадигми (табл. 10.1) дуже різні, за основу майже всіх їх взято цю конфігурацію.

Таблиця 11.1

Навідоміші нейропарадигми

Назва нейропарадигми	Рік розробки	Основні сфери застосування
Перцептрон	1957	Розпізнавання друкованого шрифту
Адалін, Мадалін	1960	Адаптивні модеми і еквалайзери в телефонії
Метод адаптивного резонансу	1968	Розпізнавання образів, особливо незвичних для людини (сигнали радара, сонара)
Неокогнітрон	1978	Розпізнавання рукописного тексту, ієрогліфів
Карта Кохонена	1980	Відображення однієї області на іншу
Мережа Хопфілда	1982	Пошук і відновлення даних за їхніми фрагментами
Двостороння асоціативна пам'ять	1985	Контекстно-адресована асоціативна пам'ять
Машина Больцмана	1985	Розпізнавання зображень, сигналів радара, сонара
Зустрічного розпізнавання	1986	Стиснення даних, оцінювання ефективності капіталовкладень

Починаючи з перцептрона, основна увага розробників ШНС приділялась розробці і вдосконаленню методів їхнього навчання. Через відсутність надійної теорії навчання ці розробки мали здебільшого евристичний характер та отримали назву «нейропарадигм».

Множина вхідних сигналів x_1, x_2, \dots, x_n надходить на штучний нейрон. Ці вхідні сигнали, у сукупності позначені вектором X , відповідають сигналам, що надходять у синапси біологічного нейрона. Кожний сигнал збільшується на відповідну вагу w_i ($i = 1, 2, \dots, n$) і надходить у підсумовуючий блок СУМ (адаптивний суматор). Кожна вага

відповідає «силі» одного синаптичного біологічного зв'язку. Множина ваг у сукупності утворює вектор ваг W .

Підсумовуючий блок, що відповідає тілу біологічного елемента, сумує зважені входи алгебраїчно, утворюючи вихід NET. У векторних позначеннях це може бути виражено таким чином: $NET = XW$. Надалі сигнал NET, як правило, перетворюється активізаційною функцією F і дає вихідний нейронний сигнал OUT. Активізаційна функція може бути звичайною лінійною функцією:

$$OUT = K(NET),$$

де K – постійна граничної функції

$OUT = 1$, якщо $Ne > T$;

$OUT = 0$ в інших випадках,

де T – деяка постійна гранична величина.

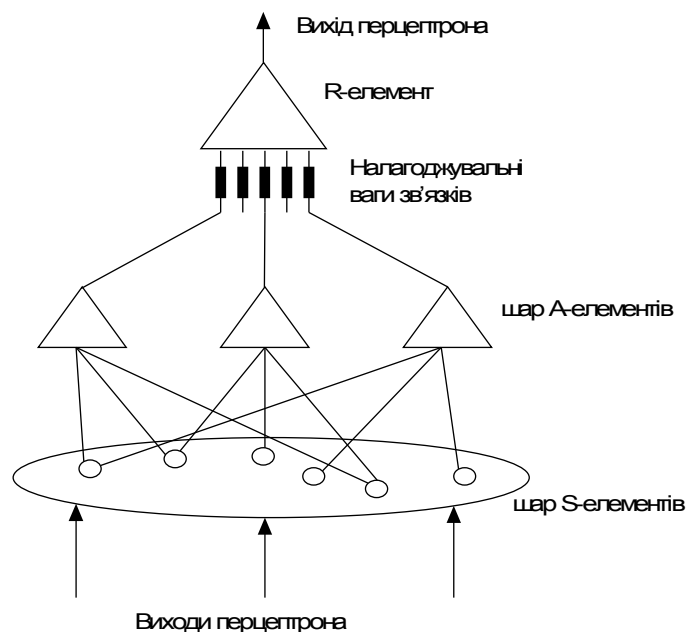


Рис. 11.1. Перцептрон Розенблатта

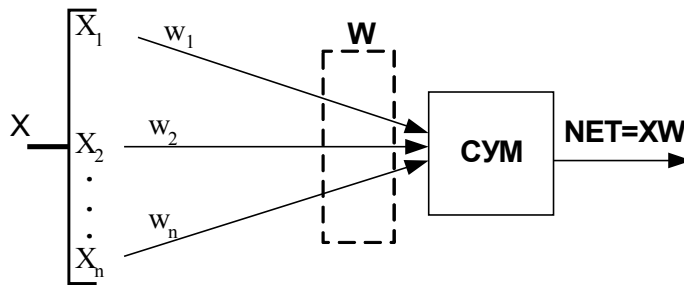


Рис. 11.2. Модель штучного нейрона

На рис. 11.3 подано структуру штучного нейрона з активізаційною функцією. Блок F приймає сигнал NET і видає сигнал OUT. Якщо блок F звужує діапазон зміни величини NET так, що при будь-яких значеннях NET значення OUT належать деякому кінцевому інтервалу, то F називається стискаючою функцією. Як стискаюча функція часто використовується *логістична чи сигмоїдальна* (S-подібна) функція, що математично виражається таким чином:

$$F(x) = 1/(1 + e^{-x}).$$

Отже,

$$OUT = 1/(1 + e^{-NET}).$$

За аналогією з електронними системами активаційну функцію можна вважати нелінійною підсилювальною характеристикою штучного нейрона. Коефіцієнт підсилення обчислюється як відношення збільшення величини OUT до його невеликого збільшення, що викликало появу величин NET.

Розглянута модель штучного нейрона ігнорує багато властивостей свого біологічного аналога. Наприклад, вона не бере до уваги затримки в часі, що впливають на динаміку системи. Вхідні сигнали відразу ж породжують вихідний сигнал.

На сьогодні кількість проданих у світі нейрокомп'ютерів обчислюється десятками, а можливо, й сотнями тисяч. Переважно це *нейрокомп'ютерні* програми для персональних комп'ютерів, призначені для розв'язання задач апроксимації і прогнозування числових даних. Близько 5% нейрокомп'ютерів належать до пристроїв професійного рівня, орієнтованих на застосування потужних робочих станцій і апаратних *нейроакселераторів*. Програмне забезпечення таких систем, як правило, вміщує *бібліотеки нейропарадигм*, що дає змогу при розв'язуванні задач використовувати різні типи нейронних мереж. Типовим прикладом може бути система *Brain Maker* фірми «CSS» (США). Вона може працювати на будь-якому комп'ютері, де встановлено Windows. Базова версія орієнтована на широке коло користувачів. Її застосування не потребує спеціальних знань. Налаштування мережі

обмежене встановленням кількох параметрів, головним серед яких є допустима похибка відповіді. Основна робота полягає у підготовці даних для навчання, які можна імпортувати з файлів формату *.DBF, *.TXT, Excel, Lotus 1-2-3.

Для розширення можливостей системи служить набір додаткових програм Toolkit Option, які дають змогу прискорити процес навчання і покращити подання графічних даних.

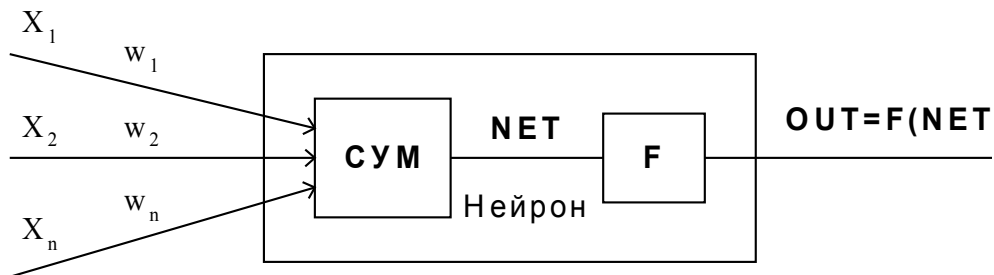


Рис. 11.3. Структура штучного нейрона з активаційною функцією

11.3. Нейрокомп'ютерні мережі

Штучна нейронна мережа може розглядатися як направлений граф зі зваженими зв'язками, в якому штучні нейрони є вузлами. За архітектурою зв'язків ШНМ можуть бути згруповані у два класи (рис. 11.4): мережі прямого поширення, в яких графи не мають петель, і рекурсивні мережі, або мережі зі зворотними зв'язками. У найбільш поширеному сімействі мереж першого класу та багат шарових парцептронів нейрони розташовані шарами і мають односпрямовані зв'язки між шарами.

Мережі прямого поширення є статичними в тому значенні, що на заданий вхід вони виробляють одну сукупність вихідних значень, які не залежать від попереднього стану мережі.

Рекурентні мережі – це динамічні мережі в тому значенні, що з огляду на зворотні зв'язки у них модифікуються входи нейронів, а це приводить до зміни стану мережі.

Навчання штучних нейронних мереж

Здатність до навчання є фундаментальною властивістю мозку людини. У контексті ШНМ процес навчання може розглядатися як настроювання архітектури мережі і ваг зв'язків для ефективного виконання спеціального завдання. Нейронна мережа має налагодити ваги зв'язків за наявною навчальною вибіркою. Функціонування мережі поліпшується відповідно до ітеративного настроювання вагових коефіцієнтів. Властивість мережі навчатися на прикладах робить їх більш привабливими. Алгоритм

навчання означає процедуру, в якій використовуються правила навчання для настроювання ваг.

Існують три парадигми навчання:

- з «учителем»;
- без «учителя» (самонавчання);
- змішана (комбінована).

У першому випадку нейронна мережа має у своєму розпорядженні правильні відповіді на кожний вхідний приклад. Ваги налагоджуються так, щоб мережа робила найближчі відповіді до відомих правильних відповідей. Посилений варіант навчання з «учителем» припускає, що відомо тільки критичну оцінку правильності виходу нейронної мережі, але не самі правильні значення виходу.

Навчання без «учителя» не потребує знання правильних відповідей на кожний приклад навчальної вибірки. В даному разі розкривається внутрішня структура даних чи кореляції між зв'язками у системі даних, що дає змогу розподілити зв'язки за категоріями.

При змішаному навчанні частина ваг визначається за допомогою навчання з «учителем», тоді як інша – за допомогою самонавчання.

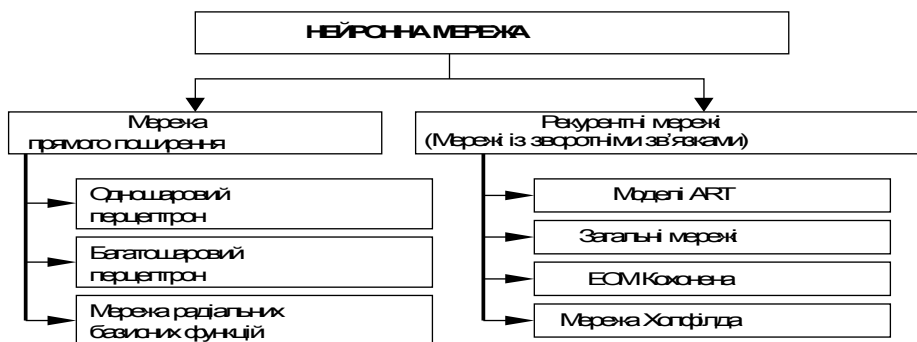


Рис. 11.4. Базові архітектури нейронних мереж

Відомі 4 основних типи правил навчання: корекція помилково, машина Больцмана, правило Хебба і навчання за методом змагання.

Правило корекції помилково

При навчанні з «учителем» для кожного вхідного прикладу заданий бажаний вихід d . Реальний вихід мережі y може не збігатися з бажаним. Принцип корекції помилково при навчанні полягає у використанні сигналу $(d-y)$ для модифікації ваг, що забезпечує поступове зменшення помилки. Навчання відбувається тільки у тому разі, коли перцептрон помиляється.

Навчання Больцмана

Це стохастичне правило навчання, що впливає з інформаційних теоретичних і термодинамічних принципів. Метою навчання Больцмана є таке настроювання вагових коефіцієнтів, при якому стани видимих нейронів

задовольняють бажаний розподіл ймовірностей. Навчання Больцмана може розглядатися як спеціальний випадок корекції помилково, в якому помилка розуміється як розбіжність кореляції станів у двох режимах.

Правило Хебба

Найдавнішим навчальним правилом є постулат Хебба. Хебб спирався на такі нейрофізіологічні спостереження: якщо нейрони по обидва боки синапсу активізуються одночасно і регулярно, то сила синаптичного зв'язку зростає. Важлива особливість цього правила полягає в тому, що зміна синаптичної ваги залежить тільки від активності нейронів, зв'язаних цим синапсом. Це істотно спрощує ланцюг навчання в реалізації VLSI.

Метод змагання

При навчанні за методом змагання вихідні нейрони змагаються між собою за активізацію. Це явище відоме як правило «переможець бере все». Навчання за допомогою змагання дає змогу кластеризувати вхідні дані: подібні приклади групуються мережею відповідно до кореляцій і подаються одним елементом. При навчанні модифікуються тільки ваги нейрона-«переможця». Ефект цього правила досягається за рахунок такої зміни збереженого в мережі зразка (вектора ваг зв'язків, який переміг нейрон), при якому він стає трохи ближче до вхідного прикладу.

Порівняно зі звичайними комп'ютерами нейрокомп'ютери мають ряд переваг:

- висока швидкодія пов'язана з тим, що алгоритми нейроінформатики мають високий ступінь паралельності;
- нейросистеми стійкі до перешкод;
- стійкі і надійні нейросистеми можуть створюватися з ненадійних елементів, які мають значний розкид параметрів.

Крім переваг, нейросистеми мають також ряд недоліків:

- вони створюються спеціально для розв'язання конкретних задач, пов'язаних з нелінійною логікою і теорією самоорганізації. Розв'язання подібних задач на звичайних комп'ютерах можливе тільки з допомогою числових методів;

- через свою унікальність ці пристрої достатньо дорогі.

Незважаючи на недоліки, нейрокомп'ютери можуть успішно використовуватися в різних галузях народного господарства:

а) керування в режимі реального часу (літаками, ракетами, технологічними процесами беззупинного виробництва);

б) розпізнавання об'єктів (букв та ієрогліфів, сигналів радара і сонара, відбитків пальців у криміналістиці, захворювань за симптомами у медицині, місцевостей при пошуку корисних копалин і т. ін.);

в) прогнози: погоди, курсу акцій (та інших фінансових показників), політичних подій (зокрема результатів виборів), поведінки супротивників у військових конфліктах і в умовах економічної конкуренції;

г) оптимізація й пошук найкращих варіантів: при конструюванні технічних пристроїв, виборі економічної стратегії й т. ін.

ЛІТЕРАТУРА

1. Амамия М., Танака Ю. Архитектура ЭВМ и искусственный интеллект: / Пер. с яп.– М.: Мир, 1993. – 453 с.
2. Березко Л. А., Пуйда В. Я., Троценко В. В. Архитектура вычислительных машин и систем: Уч. пособ. – К.: УМК ВО, 1988. – 84 с.
3. Головкин Б. А. Вычислительные системы с большим числом процессоров. – М.: Радио и связь, 1995. – 520 с.
4. Добуш Ю. Д., Козевич О. П. Оценка вычислительной мощности 16-розрядного микропроцессора K1810BM86 // Управляющие системы и машины. – 1986. – № 6. – С. 35–37.
5. Каган Б. М. Электронные вычислительные машины и системы. – М.: Энергоатомиздат, 1991. – 552 с.
6. Киселев А., Корнеев В. Современные микропроцессоры. – СПб.: БХВ – Санкт-Петербург, 2003. – 448 с.
7. Локазюк В. М. Мікропроцесори та мікро-ЕОМ у виробничих системах. – К.: Академія, 2002. – 368 с.
8. Малиновский Б. Н. Очерки по истории компьютерной науки и техники в Украине.– К.: Феникс, 1998. – 452 с.
9. Малые ЭВМ высокой производительности: архитектура и программирование / Г. П. Васильев и др. – М.: Радио и связь, 1990.
10. Майерс Г. Архитектура современных ЭВМ: В 2 кн. – М.: Мир, 1985. – 364 с.
11. Сергеев Н. П., Вашкевич Н. П. Основы вычислительной техники. – М.: Высшая школа, 1988.– 311 с.
12. Смирнов А. Д. Архитектура вычислительных систем: Уч. пособ. для вузов. – М.: Наука, 1990. – 320 с.
13. Степаненко О. С. Досье пользователя: компьютер внутри. – К.: Логос, 1998. – 128 с.
14. Супер ЭВМ. Аппаратная и программная организация: Пер. с англ. / Под ред. С. Фернбаха. – М.: Радио и связь, 1991. – 320 с.
15. Таненбаум Э. Архитектура компьютера. – СПб.: Питер, 2002. – 704 с.
16. Транспьютеры. Архитектура и программное обеспечение: Пер. с англ. / Под ред. Г. Харпа. – М.: Радио и связь, 1993.– 304 с.
17. Уоссермен Ф. Нейрокомпьютерная техника: теория и практика. – М.: Мир, 1992.– 240 с.

18. Орлов С. А., Цилькер Б. Я. Организация ЭВМ и систем. – СПб.: Питер, 2004.– 667 с.

НАУКОВЕ ВИДАННЯ

А. Карачка, О. Дудко

АРХІТЕКТУРА КОМП'ЮТЕРІВ

За редакцією доктора технічних наук, професора А. О. Саченка

Редактор *Інна Буняк*
Комп'ютерна верстка *Лариси Яцюк*
Дизайн обкладинки *Марії Одобецької*

Підписано до друку 21.04.2010 р.
Формат 60x84 ¹/₁₆. Гарнітура Times.
Папір офсетний. Друк офсетний.
Обл.-вид. арк. 17,6. Умов. друк. арк. 15,8.
Зам. № У232-09. Тираж 200 прим.

Свідоцтво про внесення суб'єкта видавничої справи
до Державного реєстру видавців ДК № 3467 від 23.04.2009 р.

Віддруковано в ТНЕУ
(видавництво «Економічна думка»)
46020 Тернопіль, вул. Львівська, 11
тел. (0352) 47-58-72
E-mail: edition@tneu.edu.ua

