

13.4 Моделі обчислювальних систем

Проектування і використання різноманітних обчислювальних систем ґрунтується на аналізі і синтезі їх моделей.

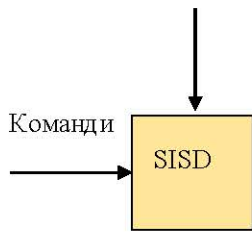
13.4.1 Види обчислювальних систем

Існують різні архітектури обчислювальних систем, і однією зі схем їх класифікації є таксономія Флінна, заснована на описі роботи системи з потоками команд і даних (рис. 13.23).

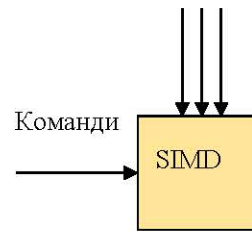
Взаємодія даних і команд може бути зображена інформаційним графом. Інформаційний граф описує послідовність виконання операцій і взаємну залежність між різними макроопераціями чи блоками операцій. Вузлами інформаційного графу є макрооперації, а односпрямованими дугами – канали обміну даними (рис. 13.24)

Кожен вузол можна охарактеризувати парою (n, s) , де n – ім'я вузла, s – його розмір. Розмір вузла визначається кількістю елементарних операцій, що входять до його складу. Дуга характеризується парою (v, d) , де v – дані, що пересилаються, а d – час, що витрачається на їх пересилання. Інформаційний граф містить лінійні ділянки, розгалуження і цикли.

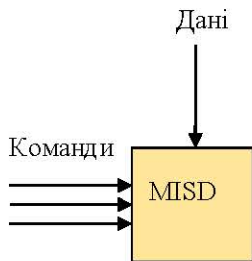
Архітектура фон Неймана – широко відома модель обчислювальної системи зі спільним зберіганням програм і даних у пам'яті комп'ютера. Обчислювальні системи такого роду часто позначають терміном “машина фон Неймана”, а основні принципи її побудови – “принципами фон Неймана”.



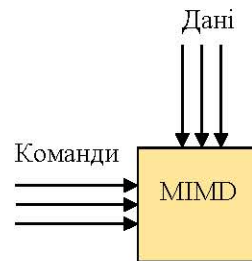
а) Архітектура SISD(Single Instruction Stream - Single Data Stream) – один потік команд і один потік даних



б) Архітектура SIMD(Single Instruction Stream - Multiple Data Stream) – один потік команд і декілька потоків даних

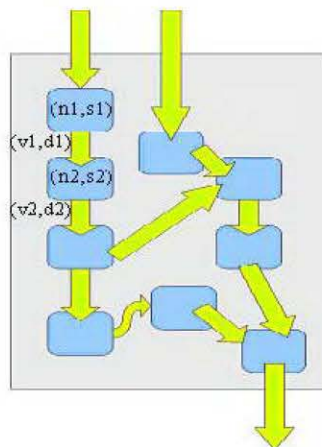


в) Архітектура MISD(Multiple Instruction Stream - Single Data Stream) – декілька потоків команд і один потік даних



г) Архітектура MIMD(Instruction Stream - Multiple Data Stream) – декілька потоків команд і декілька потоків даних

Рисунок 13.23 – Архітектура обчислювальних систем



Принципи фон Неймана

1. *Принцип двоїстості.* Для запису даних і команд використовується двійкова система числення.

2. *Принцип програмного управління.* Програма складається з набору команд, які виконуються процесором одна за одною у певній послідовності.

3. *Принцип однорідності пам'яті.* Як програми (команди), так і дані зберігаються в одній і тій же пам'яті (і кодуються в одній і тій же системі числення). Над командами можна виконувати такі ж дії, як і над даними.

4. *Принцип адресації пам'яті.* Структурно основна пам'ять складається з пронумерованих комірок; процесору в довільний момент часу доступна будь-яка комірка.

5. *Принцип послідовного програмного управління.* Всі команди розташовуються в пам'яті і виконуються послідовно, одна після завершення іншої.

6. *Принцип умовного переходу.* Команди програми не завжди виконуються одна за одною. Можлива присутність у програмі команд умовного переходу, які змінюють послідовність виконання команд залежно від значень даних.

На відміну від архітектури фон Неймана, у рамках паралельної моделі існують різні підходи, орієнтовані на різні архітектури високопродуктивних обчислювальних систем і різні інструментальні засоби:

- модель передачі повідомлень;
- модель паралелізму даних;
- модель загальної пам'яті.

У комп'ютерній архітектурі *систолічна модель* розглядає масив процесорних елементів (рис. 13.25), які називаються комірками. Це спеціалізована форма паралельних обчислень, де комірки (тобто процесори), здійснюють обчислення і зберігання даних незалежно один від одного.

Систолічний масив складається з рядків елементарних комірок, подібно до матриць. Комірка DPU аналогічна процесору (CPU). Кожна комірка обмінюється інформацією зі своїми сусідами відразу ж після обробки. Систолічні масиви звичайно прямокутні, потоки даних через масив між сусідніми DPU взаємодіють з іншими даними, що рухаються у різних напрямках.

13.4.2 Математичні моделі обчислень

Системи виконання обчислень можуть розглядатися на мікрорівні (рівні процесора і його регістрів) і на макрорівні (рівні комп'ютера і обчислювальної системи).

На мікрорівні базовою моделлю обчислень є абстрактний автомат. Абстрактний автомат – модель дискретного пристрою, що має один вхід, один вихід і в кожний момент часу знаходиться в одному стані з множини можливих. На вхід

цього пристрою надходять символи однієї мови (множини), на виході він видає символи в загальному випадку іншої мови.

Формально абстрактний автомат визначається як п'ятірка

$$A = (S, X, Y, \delta, \lambda), \quad (13.41)$$

де S – множина станів автомата, X, Y – вхідний і вихідний алфавіти відповідно, $\{\delta : S \times X \rightarrow S\}$ – функція переходів, $\{\lambda : S \times X \rightarrow Y\}$ – функція виходів.

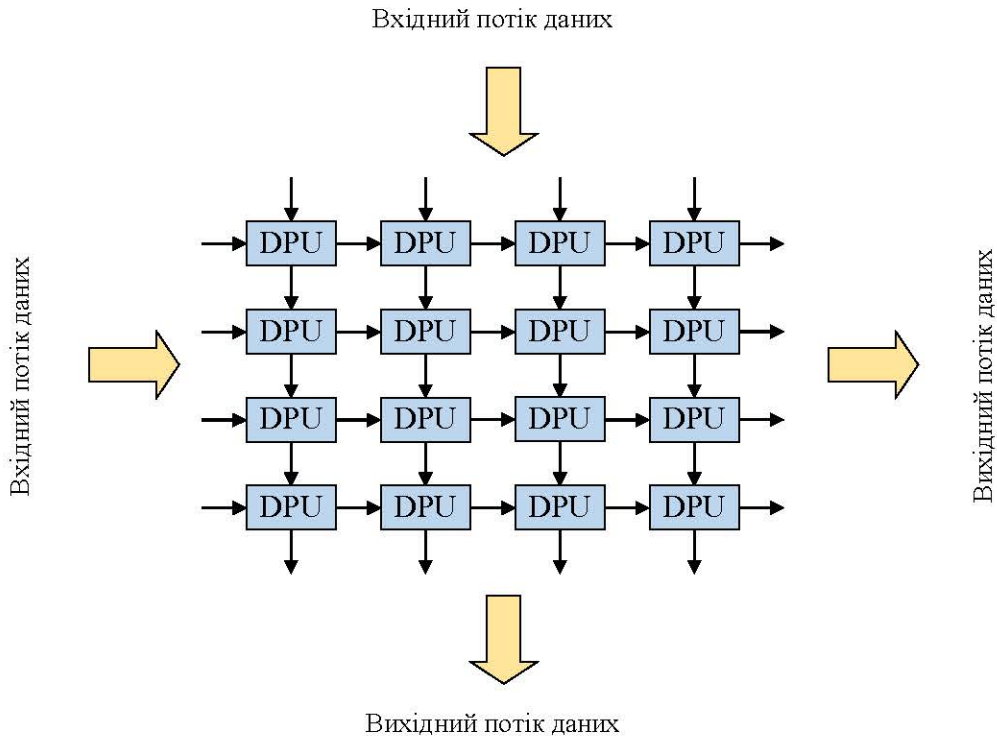


Рисунок 13.25 – Систолічний масив процесорних елементів

Якщо функції переходів і виходів однозначно визначені для кожної пари (s, x) , то автомат називають детермінованим. В іншому випадку автомат називають недетермінованим або частково визначеним. Якщо функція переходів або функція виходів є випадковою, то автомат називають ймовірнісним.

Функціонування автомата полягає в породженні двох послідовностей: послідовності станів автомата $\{s_1, s_2, s_3 \dots\}$ і послідовності вихідних символів $\{y_1, y_2, y_3 \dots\}$, які для послідовності символів $\{x_1, x_2, x_3 \dots\}$ розгортаються в моменти дискретного часу $\{t = 1, 2, 3, \dots\}$. Моменти дискретного часу отримали назву тактів.

Функціонування автомата в дискретні моменти часу t може бути описано системою рекурентних співвідношень, яка є аналогом моделі дискретної системи управління у просторі станів (13.30):

$$\begin{cases} s(t+1) = \delta[s(t), x(t)] \\ y(t) = \lambda[s(t), x(t)] \end{cases} \quad (13.42)$$

Кінцевий автомат – модель автомата, за умови, що загальна можлива кількість станів обмежена. Якщо кількість можливих станів невелика (наприклад, кінцевий автомат “магнітний замок” може знаходитися у трьох станах: вимкнений, увімкнений і закритий, увімкнений і відкритий), то кінцевий автомат може задаватися формальним описом усіх його станів.

Діаграма станів (або іноді *граф переходів*) – графічне зображення множини станів і функції переходів, є зваженим орієнтованим графом, вершини якого – стани автомата, ребра – переходи з одного стану в інший, а ваги – символи, при яких здійснюється даний перехід.

Таблиця переходів – табличне подання функції переходів δ . Зазвичай у такій таблиці кожному рядку відповідає один стан, а стовпцю – один допустимий вхідний символ. У комірці на перетині рядка і стовпця записується дія, яку повинен виконати автомат, якщо в ситуації, коли він знаходився в даному стані, на вході він отримав цей знак.

Детермінованим кінцевим автоматом (ДКА) називається такий автомат, в якому для кожної послідовності вхідних символів існує лише один стан, до якого автомат може перейти з поточного.

Недетермінований кінцевий автомат (НКА) є узагальненням детермінованого. Недетермінованість автоматів досягається двома способами:

- існують переходи, помічені порожнім ланцюжком ε ;
- з одного стану виходить кілька переходів, помічених одним і тим же символом.

Кінцевий автомат з пам'яттю – математична модель пристрою, поведінка якого залежить як від вхідних умов, так і від попереднього стану.

Для опису кінцевого автомата з пам'яттю використовуються мови операторних схем, регулярних виразів алгебри подій, а також матриці і графи переходів.

Секвенціальна логіка – це логіка пам'яті цифрових пристроїв. Секвенціальна логіка відрізняється від комбінаційної логіки тим, що моделює цифрові пристрої з урахуванням передісторії їхнього функціонування.

Залежно від способу функціонування цифрові пристрої підрозділяються на синхронні та асинхронні. Відповідно, їхня поведінка підкоряється або синхронній, або асинхронній логіці.

Синхронна секвенціальна логіка. При логічному моделюванні пристроїв з пам'яттю особлива роль приділяється фактору часу, який у синхронних схемах враховується тактами кінцевого автомата. Такти визначають моменти зміни ста-

нів автомата, тобто, синхронізують відповідну функцію. Математичний апарат синхронної логіки задають автоматні моделі Мілі та Мура.

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

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

Секвенція – це послідовність елементів, яка надається впорядкованою множиною, наприклад, $\langle x \rangle = \langle x_1 x_2 \dots x_n \rangle$, де $x_i \in \{0,1\}$.

За допомогою секвенції реалізується двійкова функція $z = \varphi(\langle x \rangle)$, така, що має місце тільки у разі $\forall x_i = 1$, причому $(x_i = 1) \prec (x_j = 1)$ для всіх $i < j$. Символ \prec задає відношення випередження. Таким чином, *секвенціальна функція дорівнює одиниці при одиничних значеннях аргументів, встановлення яких здійснюється по чергово, починаючи з x_1 і закінчуючи x_n . В усіх інших випадках $z=0$.*

Вен'юнкція – це асиметрична логіко-динамічна операція \prec , згідно з якою зв'язка $x \prec y$ приймає одиничне значення тільки у випадку $x \wedge y = 1$ за умови, що в момент встановлення $x = 1$ рівність $y = 1$ вже мала місце.

Вен'юнкція і мінімальна (двоелементна) секвенція функціонально ідентичні: $x \prec y = \langle yx \rangle$.

Вен'юнктор є основним операційним елементом пам'яті секвенціальної логіки. Секвентор будується на основі композиції вен'юнкторів.

13.4.3 Моделі промислових комп'ютерних мереж

З масовим поширенням дешевих контролерів і «інтелектуальних» елементів (з вбудованими програмованими блоками обробки даних) основним принципом організації систем управління є інтеграція елементів системи за допомогою промислових мереж.

Промислова мережа – мережа передачі даних, що зв'язує сенсори, виконавчі механізми, промислові контролери. Термін описується стандартом ІЕС 61158.

Основні функції промислових мереж:

- передавання даних між сенсорами, контролерами і виконавчими механізмами;
- діагностика та дистанційне конфігурування сенсорів і виконавчих механізмів;
- калібрування сенсорів;
- живлення сенсорів і виконавчих механізмів;
- зв'язки між контролерами та системами людино-машинного інтерфейсу (SCADA).

Промислові мережі можуть взаємодіяти зі звичайними комп'ютерними мережами, зокрема використовувати глобальну мережу Internet.

У порівнянні з підключенням периферійного обладнання до контролера окремими проводами промислова мережа має такі переваги:

- у кілька разів знижуються витрати на кабель і його прокладання;
- збільшується допустима відстань до сенсорів і виконавчих пристроїв;
- спрощується управління мережею комплексом обладнання АСУ ТП;
- спрощується модифікація системи при заміні обладнання і програмного забезпечення.

Наразі поширення набули декілька стандартів промислових мереж:

– Modbus/RS-485 – найпростіша, найдешевша і широко розповсюджена промислова мережа, заснована на стандарті RS-485, що використовує захищену виту пару;

– Промислові модифікації Ethernet: Profinet; FOUNDATION Fieldbus HSE (High Speed Ethernet, FF H2); EtherCAT; Ethernet Powerlink; Ether /IP; SERCOS III;

– ProfiBus – міжнародний стандарт, створений за активної участі фірми Siemens AG: ProfiBus DP, ProfiBus FMS, ProfiBus PA; FF H1 – польовий протокол, аналог ProfiBus PA;

– HART – мережа для аналогових сенсорів;

– AS-Interface – дешева і заводостійка мережа для дискретних приладів малої продуктивності;

– CAN – промислова мережа для автоматизації транспорту та машинобудування: CANbus, CANopen, DeviceNet, SDS, J1939

– Промислові мережі для автоматизації будівель: LonWorks; BACnet; EIB; MBus

– Промислові мережі автомобілів: LIN; FlexRay.

Усі правила організації комп'ютерних мереж називають протоколами.

Опис системи таких протоколів є моделлю мережі.

Модель OSI (базова еталонна модель взаємодії відкритих систем, англ. Open Systems Interconnection Basic Reference Model, 1978 р.) – модель для комунікацій і розробки мережних протоколів. Пропонує багаторівневу модель комп'ютерної мережі. Кожен рівень моделює свою частину процесу взаємодії.

В даний час основною використовуваною системою протоколів є TCP/IP, розробка яких була здійснена до прийняття моделі OSI, проте вони частково відповідають цій моделі.

Модель OSI складається з семи рівнів. Рівні взаємодіють один з одним (по «вертикалі») за допомогою інтерфейсів і можуть взаємодіяти з паралельним рівнем іншої системи (по «горизонталі») за допомогою протоколів. Кожен рівень може взаємодіяти тільки зі своїми сусідами й виконувати відведені тільки йому функції, як показано на рис. 13.26.

Модель OSI		
Тип даних	Рівень	Функції
Дані	7. Прикладний рівень	Доступ до мережевих служб
	6. Рівень подання	Подання і кодування даних
	5. Сеансовий рівень	Управління сеансом зв'язку
Сегменти	4. Транспортний	Прямий зв'язок між кінцевими пунктами і надійність
Пакети	3. Мережевий	Маршрутизація і логічна адресація
Кадри	2. Канальний	Фізична адресація
Біти	1. Фізичний рівень	Робота з середовищем передавання, сигналами і фізичним поданням бітів

Рисунок 13.26 – Рівні протоколів у моделі OSI

Сімейство TCP / IP розвивалося паралельно з протоколами OSI, тому не зовсім відповідає цій моделі. Сімейство TCP / IP має три транспортних протоколи: TCP, повністю відповідний OSI, що забезпечує перевірку отримання даних; UDP, що відповідає транспортному рівню тільки наявністю порту, що забезпечує обмін датаграмами між додатками, але не гарантує отримання даних; SCTP, розроблений для усунення деяких недоліків TCP, в який додані деякі нововведення. У сімействі TCP / IP є ще близько двохсот протоколів.

У сімействі *IPX/SPX* порти (так звані “сокети” або “гнізда”) з'являються в протоколі мережевого рівня IPX, забезпечуючи обмін датаграмами між додатками (операційна система резервує частину сокетів для себе). Протокол SPX, у свою чергу, доповнює IPX всіма іншими можливостями транспортного рівня в повній відповідності з OSI. Як адресу хоста IPX використовує ідентифікатор, утворений з чотирибайтових номерів мережі (що призначаються маршрутизаторами) і MAC-адреси мережевого адаптера.

Взаємодія елементів мережі на основі системи протоколів є дуже складною, і в процесі такої взаємодії можуть виникати різноманітні колізії. **NS-2 (Network Simulator)** – система моделювання дискретних подій в мережі. NS-2 широко використовуються в моделюванні маршрутизації та протоколи передачі даних. NS-2 підтримує цілий ряд популярних мережевих протоколів.