



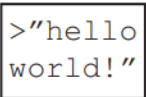


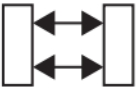
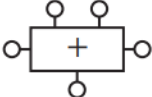

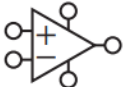


ПРЕДСТАВЛЕННЯ ПРОГРАМИ В ПАМ'ЯТІ КОМП'ЮТЕРА

Лекція 02
Основи інформатики, програмування та алгоритмічні
мови



План лекції

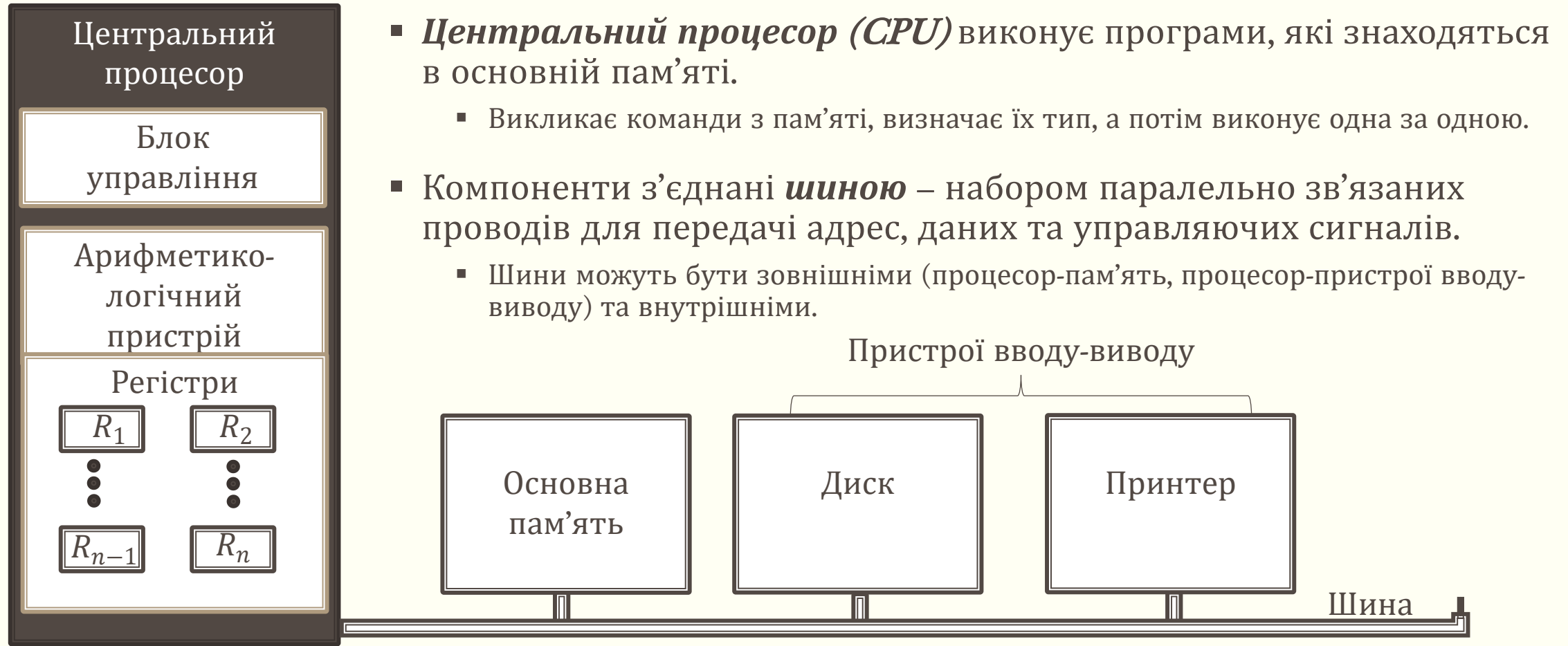
- **Організація комп'ютерних систем**
 - Цикл роботи процесора
 - Ієрархія пам'яті в комп'ютерних системах
- **Представлення інформації в пам'яті комп'ютера.**
 - Представлення даних в пам'яті
 - Типи даних
- **Представлення програми в пам'яті**

Application Software		Programs
Operating Systems		Device Drivers
Architecture		Instructions Registers
Micro-architecture		Datapaths Controllers
Logic		Adders Memories
Digital Circuits		AND Gates NOT Gates
Analog Circuits		Amplifiers Filters
Devices		Transistors Diodes
Physics		Electrons



ОРГАНІЗАЦІЯ КОМП'ЮТЕРНИХ СИСТЕМ

Схема комп'ютера з одним центральним процесором та двома пристроями вводу-виводу

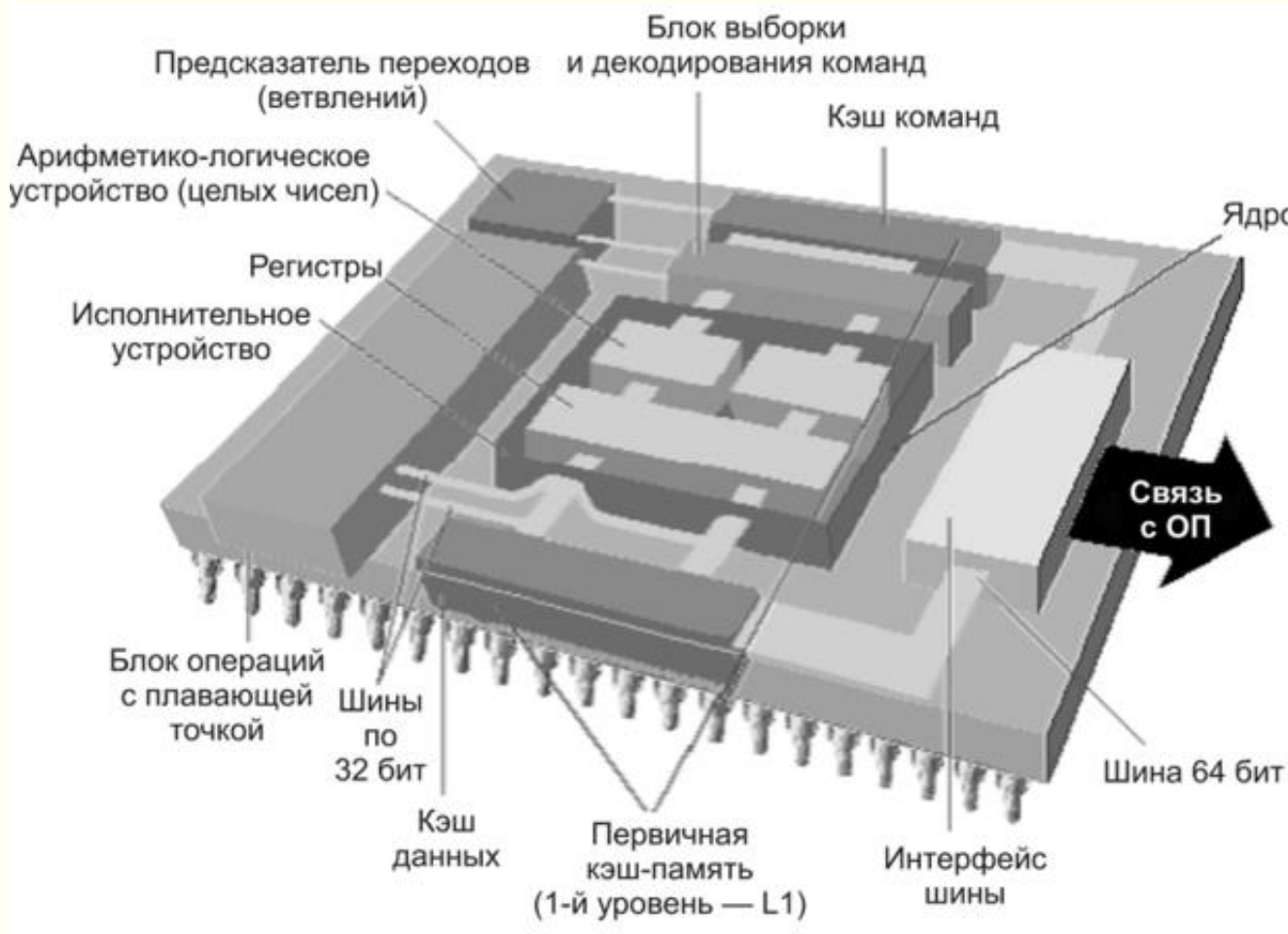


Архітектура фон Неймана (принстонська архітектура)



- У 1941 р. Джон фон Нейман описав принципи роботи та обґрунтував принципову схему комп'ютера з класичною однопроцесорною архітектурою:
 - **Використання двійкової системи числення в обчислювальних машинах.** Спрощує пристрої, арифметичні та логічні операції.
 - **Програмне управління ЕОМ.** Робота ЕОМ контролюється програмою, що складається з набору послідовно виконуваних команд.
 - **Пам'ять комп'ютера використовується для збереження не тільки даних, а й програм.** Команди програми і дані кодуються в двійковій системі числення, тому в деяких випадках над командами можна виконувати ті ж дії, що й над даними.
 - **Комірки пам'яті ЕОМ мають адреси, які послідовно пронумеровані.** У будь-який момент можна звернутись до будь-якої комірки пам'яті за її адресою. Даний принцип дозволив використовувати змінні в програмуванні.
 - **Можливість умовного переходу під час виконання програми.** У програмах можна реалізувати можливість переходу до будь-якої частини коду.

Основні компоненти процесора Pentium



- **control unit (CU)**: виконує команди та має
 - реєстр команд (*Instruction Register*), який містить наступну інструкцію для обробки
 - *Program Counter*, що зберігає адресу наступної інструкції в пам'яті
- **Арифметико-логічний пристрій (ALU)**: виконує арифметичні операції
 - Містить реєстри, в яких зберігаються дані та результат операцій
- **Блок декодування команд (instruction decoding unit)**: декодує команду після її зчитування з пам'яті
- **Блок вибірки (prefetch unit)**: виконує запит на зчитування команди з пам'яті
- **Інтерфейс шини (bus interface unit)**: управляє взаємодією з блоком пам'яті (окремим від CPU)

Структура машинних інструкцій

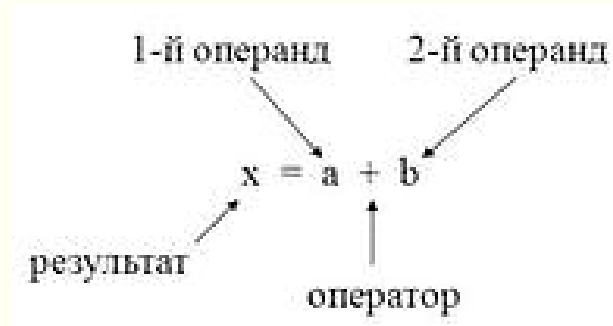
- **Команда (машинна інструкція)** – це інструкція машині на виконання елементарної операції.
 - Набір операцій, які може виконувати комп'ютер, і правил їх запису утворюють *машинну мову*.

КОП	A1	A2	A3	A4
Код операції	Адреса 1-го операнда	Адреса 2-го операнда	Адреса поміщення результату	Адреса наступної команди

Приклад команди:

MOV R5,#25H

;завантажити 25H в R5



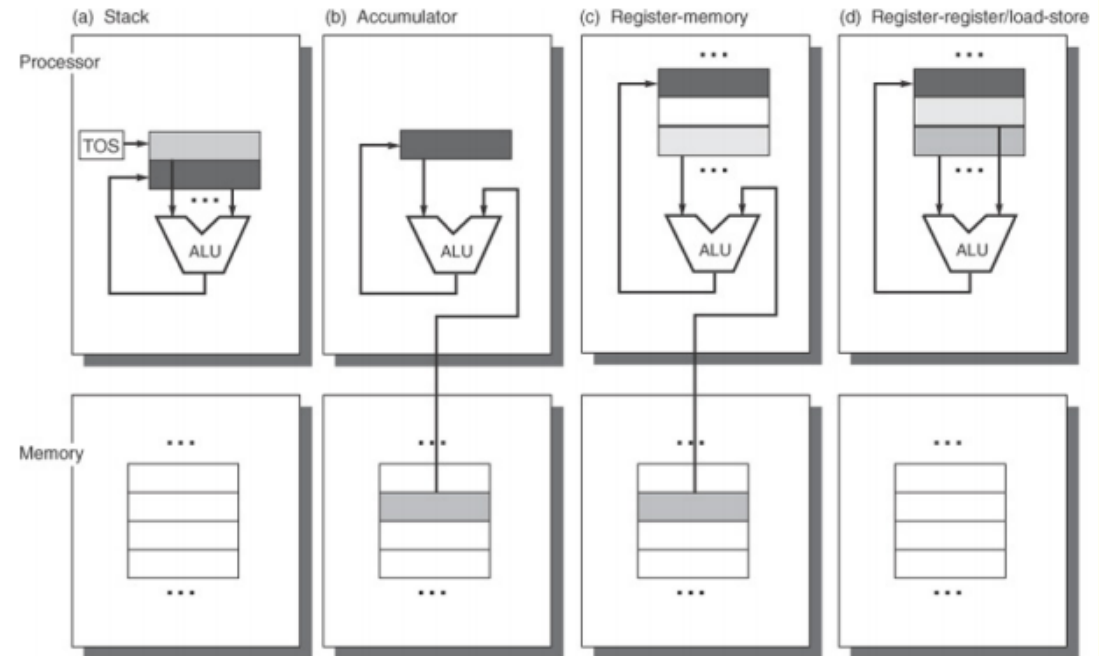
C=A+B у різних схемах зберігання (storage schemes)

1. *Accumulator*

```
load A
add B
Store C
```

2. *Stack*

```
push A
push B
add
pop C
```



© 2007 Elsevier, Inc. All rights reserved.

3. *Memory-Memory*

```
add C, A, B
```

4. *Register-Memory*

```
load R1, A
add R2, R1, B
store R2, C
```

5. *Register-Register*

```
load R1, A
load R2, B
add R3, R2, R1
store R3, C
```


КОП	<i>нульадресная команда</i>		
КОП	операнд	<i>одноадресная команда</i>	
КОП	регистр	операнд	<i>полуторадресная команда</i>
КОП	регистр	регистр	операнд <i>полуторадресная команда</i>
КОП	операнд	операнд	<i>двухадресная команда</i>
КОП	операнд	операнд	результат <i>трехадресная команда</i>
КОП	операнд	операнд	результат адрес следующей команды

Форматы команд

Одноадресні команди

```
LOAD X
MULT Y
STORE TEMP
LOAD W
MULT U
ADD TEMP
STORE Z
```

```
MULT R1,X,Y
MULT R2,W,U
ADD Z,R1,R2
```

Двоадресні команди

```
LOAD R1,X
MULT R1,Y
LOAD R2,W
MULT R2,U
ADD R1,R2
STORE Z,R1
```

Z = (X × Y) + (W × U)



Z = X Y × W U × +



Набір інструкцій процесора Intel 4004

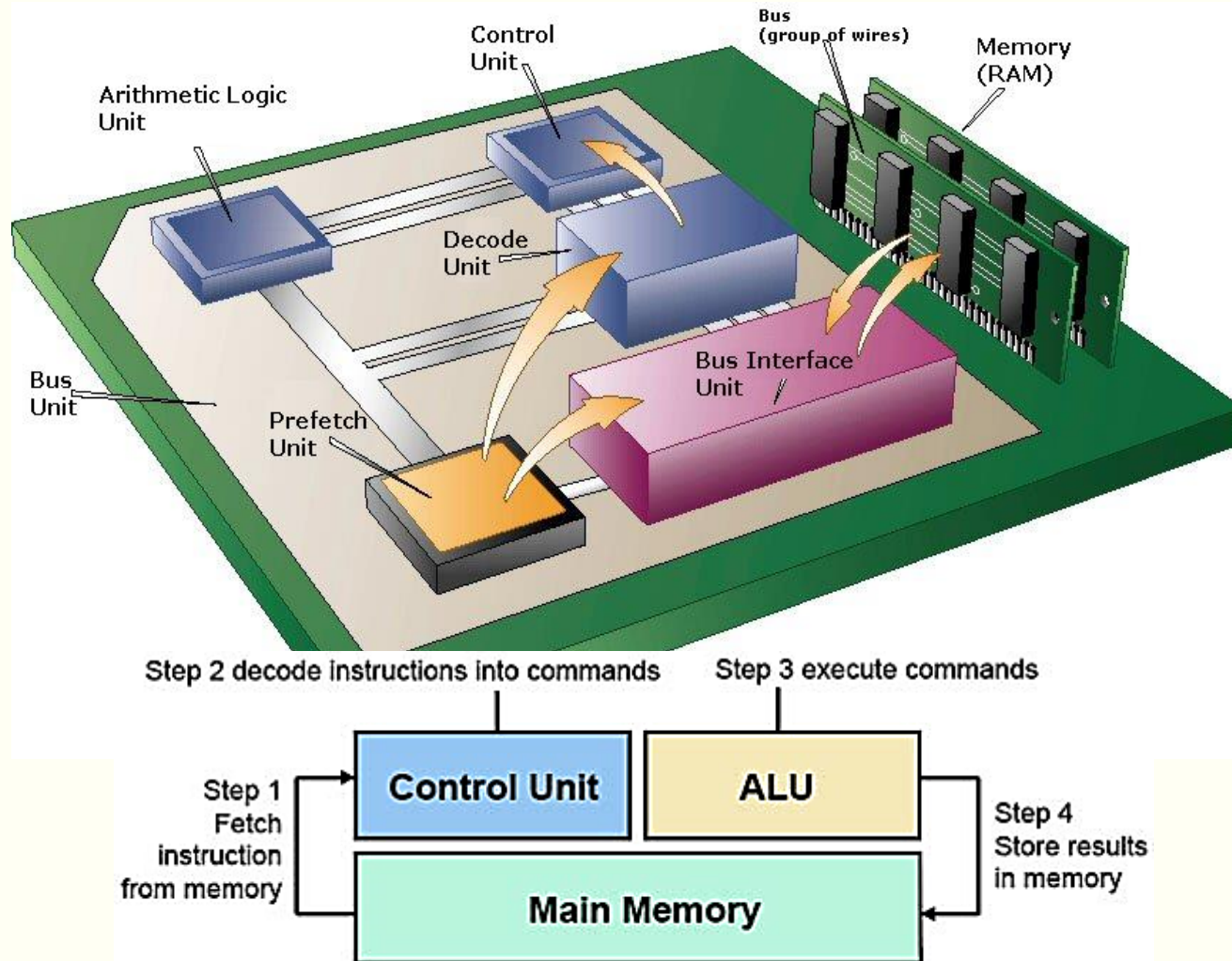
■ Має 46 інструкцій:

- Інструкції з переміщення даних (Data moving instructions).
- Арифметичні інструкції (add, subtract, increment, decrement).
- Логічні – rotate (циклічний бітовий зсув).
- Інструкції передачі управління (Control transfer) – умовний перехід, безумовний перехід, виклик підпрограми (subroutine) та повернення (return) з підпрограми.
- Інструкції вводу-виводу.
- Інші - carry flag operations, decimal adjust тощо.

■ Час виконання:

- 8-бітна інструкція виконується 10.8 нс.
- 16-бітна інструкція виконується 21.6 нс.

ЦП виконує кожну команду за кілька кроків



Етапи машинного циклу (*machine cycle, instruction cycle*):

- **Отримання (Fetch)** команди: блок вибірки виконує запит команди з пам'яті. Команда передається по шині в блок декодування.
- **Декодування (Decode)** команди: блок декодування дешифрує операцію та операнди, активує блок управління (control unit)
- **Виконання (Execute)** команди: блок управління надсилає значення до ALU, щоб додати, відняти його та ін.
- **Зберігання (Store)** результатів: результати операції можуть зберігатись у пам'яті

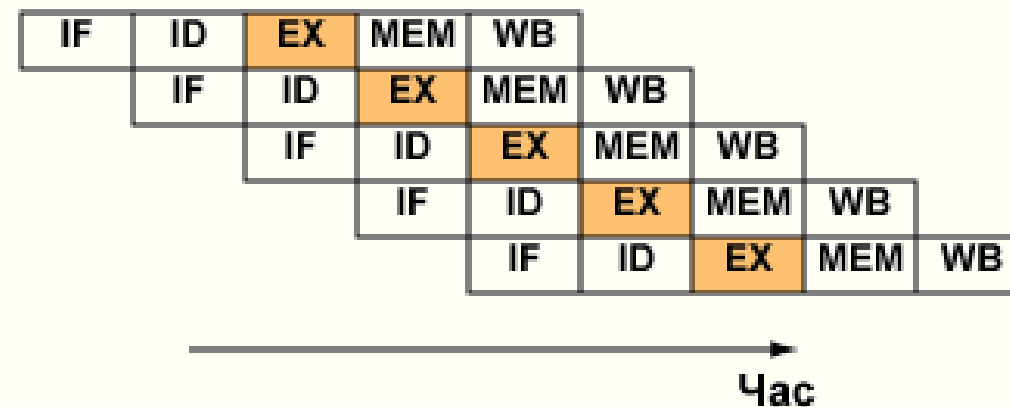
Повторюється мільйони разів за секунду.

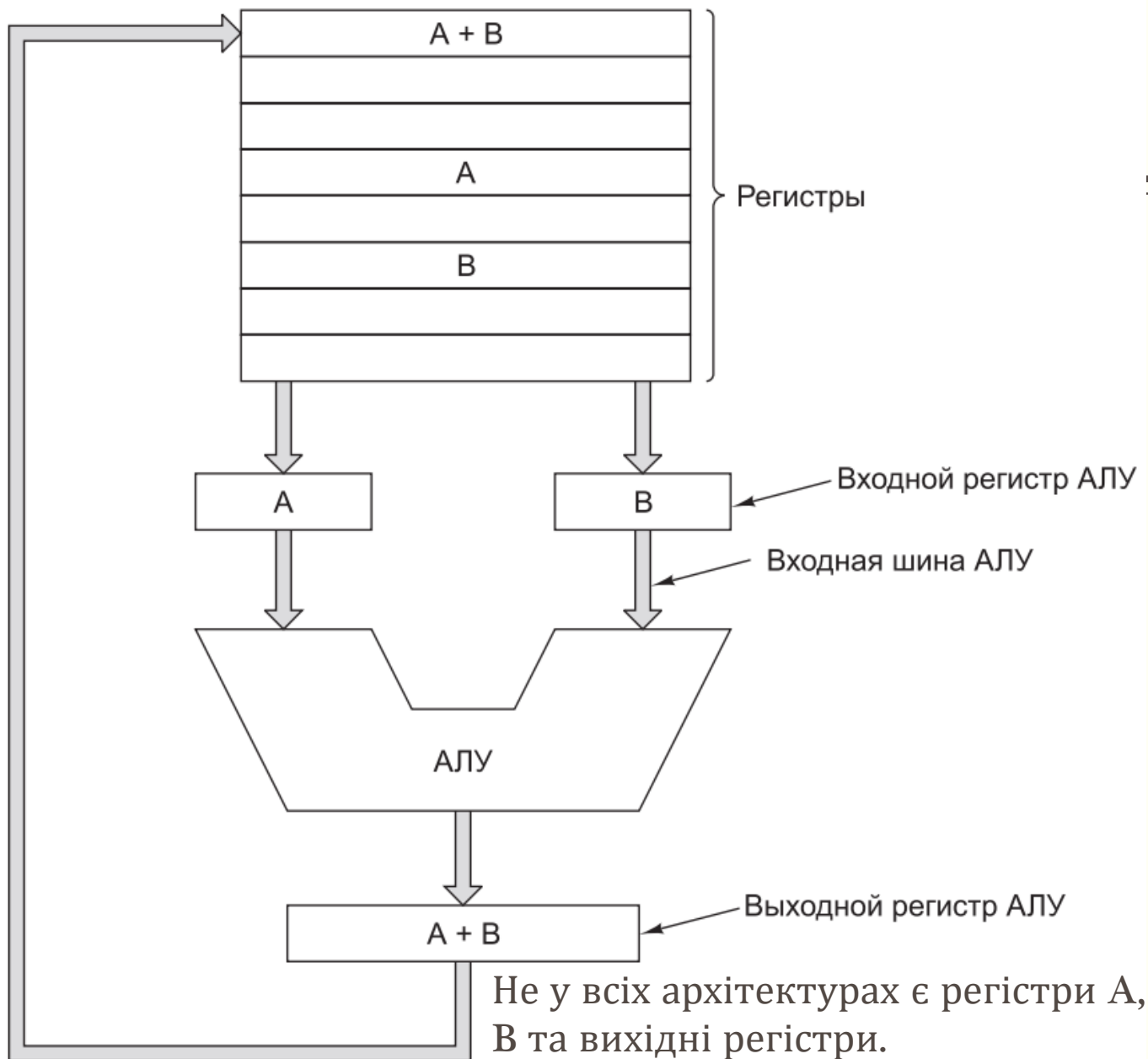
- Управляється тактовим генератором (*clock*), який формує періодичні імпульси – такти.
- Деякі інструкції потребують більше тактів процесора, ніж інші.

Найпростіший конвеєр команд ЦП

- Загальноприйнятою в теорії конвеєрних структур є така послідовність етапів:
 - **Вибірка** (instruction fetch, IF) — завантаження нової команди з пам'яті
 - **Декодування** (instruction decode, ID) — інтерпретація та відправка команди у відповідний операційний пристрій в залежності від різновиду операції
 - **Виконання** (execution, EX) — виконання команд та обчислення ефективної адреси пам'яті для результату або операндів, які необхідно завантажити
 - **Звертання до пам'яті** (memory, MEM) — виконання операцій з пам'яттю (для команд завантаження/збереження)
 - **Збереження результату** (writeback, WB) — збереження результату обчислень в регістрі

- Найпростіший конвеєр:



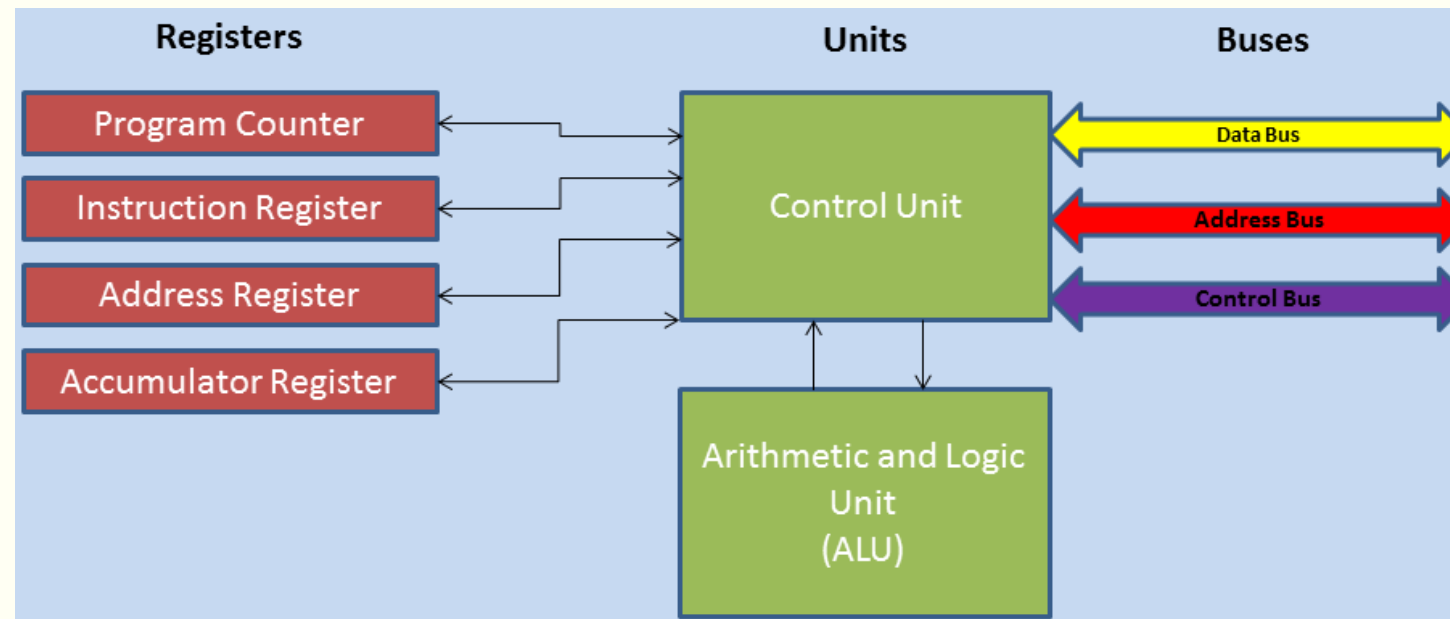


Влаштування центрального процесора

- Тракт даних складається з регістрів, арифметико-логічного пристрою (АЛУ) та кількох з'єднувальних шин.
- Вміст регістрів надходить на входні регістри АЛУ,
 - поки АЛУ виконує обчислення.
- АЛУ виконує додавання, віднімання та інші прості операції над входніми даними та поміщає результат у вихідний регістр.
 - Вміст вихідного регістру може записуватись назад в один з регістрів або за потреби зберігатись у пам'яті.

Регістри процесора

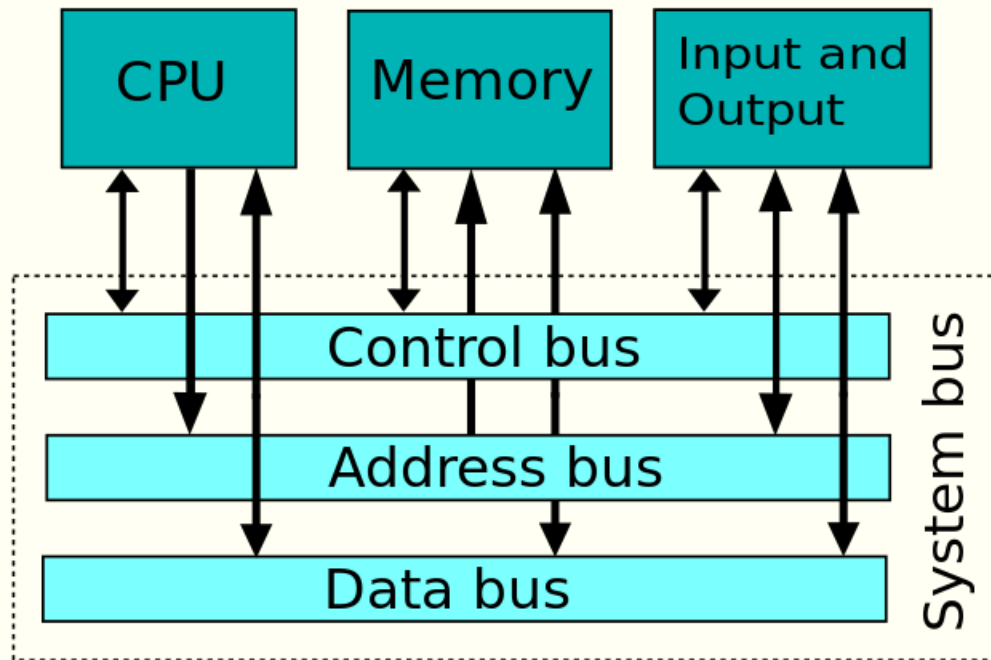
- Дуже швидка пам'ять, що використовується для прискорення виконання програм, надаючи швидкий доступ до значень, які часто використовуються.
 - Зазвичай це значення, що використовуються в процесі обчислень.
 - Зчитування/запис даних відбувається за 1 такт процесора.
 - Результат операції ALU зберігається тут і може повторно використовуватись у подальшій операції або зберігатись у пам'яті.



Чим характеризується ЦП?

- **Тактова частота процесора (*clock speed*)** — кількість тактів (елементарних операцій) за одиницю часу.
 - 1 Гц = 1 такт / 1 секунду
 - Процесор з вищою тактовою частотою може працювати повільніше за процесор з нижчою тактовою частотою.
 - Деякі команди вимагають більше тактів, ніж інші
 - Якщо додаток використовує більше складних команд (*complex instructions*), він працюватиме повільніше, ніж інший додаток.
- ЦП-чип може містити багато **ядер**.
 - Кожне ядро є процесором, всі вони можуть працювати одночасно.
 - Кожне ядро зазвичай має власну кеш-пам'ять, проте деякі кеші можуть бути спільними.
 - Всі ядра мають спільний модуль пам'яті RAM
- **Кешування** – це використання додаткової швидкодіючої пам'яті для зберігання копій блоків інформації з основної (оперативної) пам'яті, ймовірність звернення до яких у найближчий час є високою.

Системна магістраль (шина)

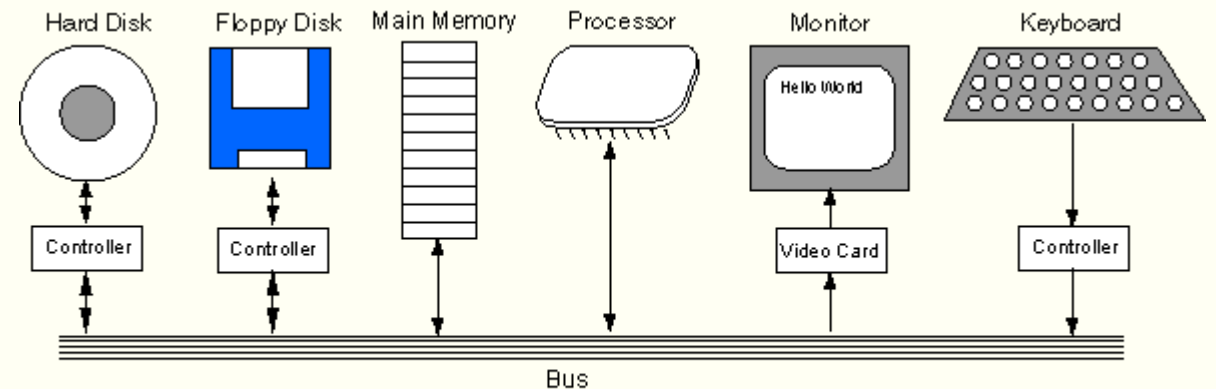


■ Front Side Bus (FSB)

- Для процесора Pentium 4 з частотою 2.4ГГц має власну частоту всього 400МГц.

■ Intel QuickPath Interconnect (QPI)

- Представлена в листопаді 2008р. для процесорів Intel Core i7-9xx та чипсету X58.
- Працює на частотах 2.4ГГц, 2.93ГГц, 3.2ГГц, 4.0ГГц або 4,8ГГц.



Main Components of a Computer System

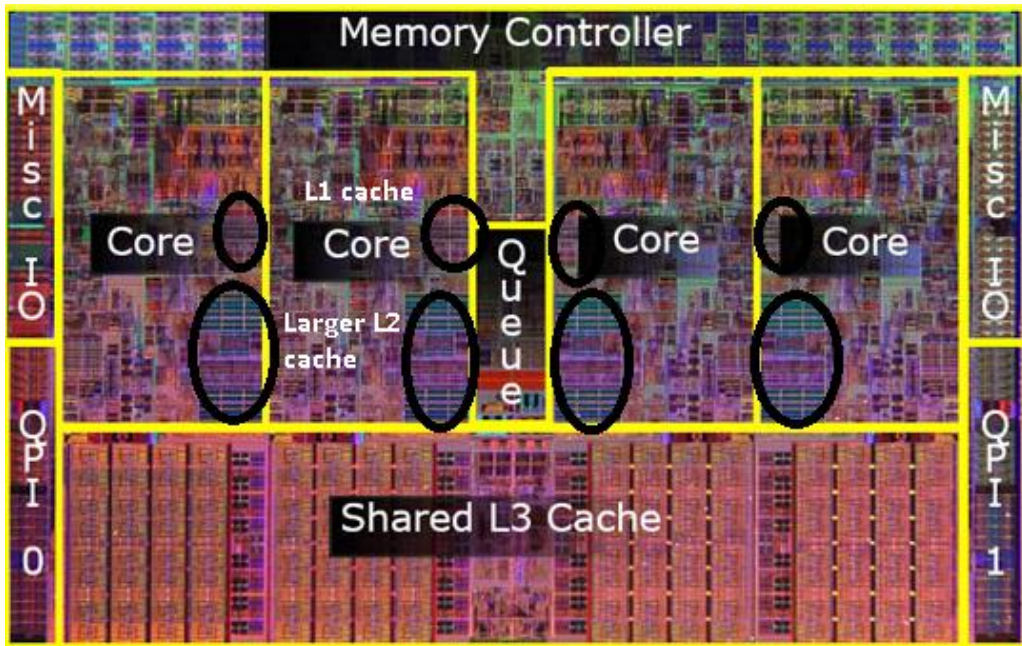
Статична (SRAM) та динамічна (DRAM) пам'ять з довільним доступом

- Кеш є статичною пам'яттю з довільним доступом.
 - Нещодавно та часто використовувані інструкції та дані кешуються.
 - Контролер пам'яті отримує дані з ОЗП (RAM) та передає їх у кеш.
 - Для кешу теж характерна ієрархія пам'яті.



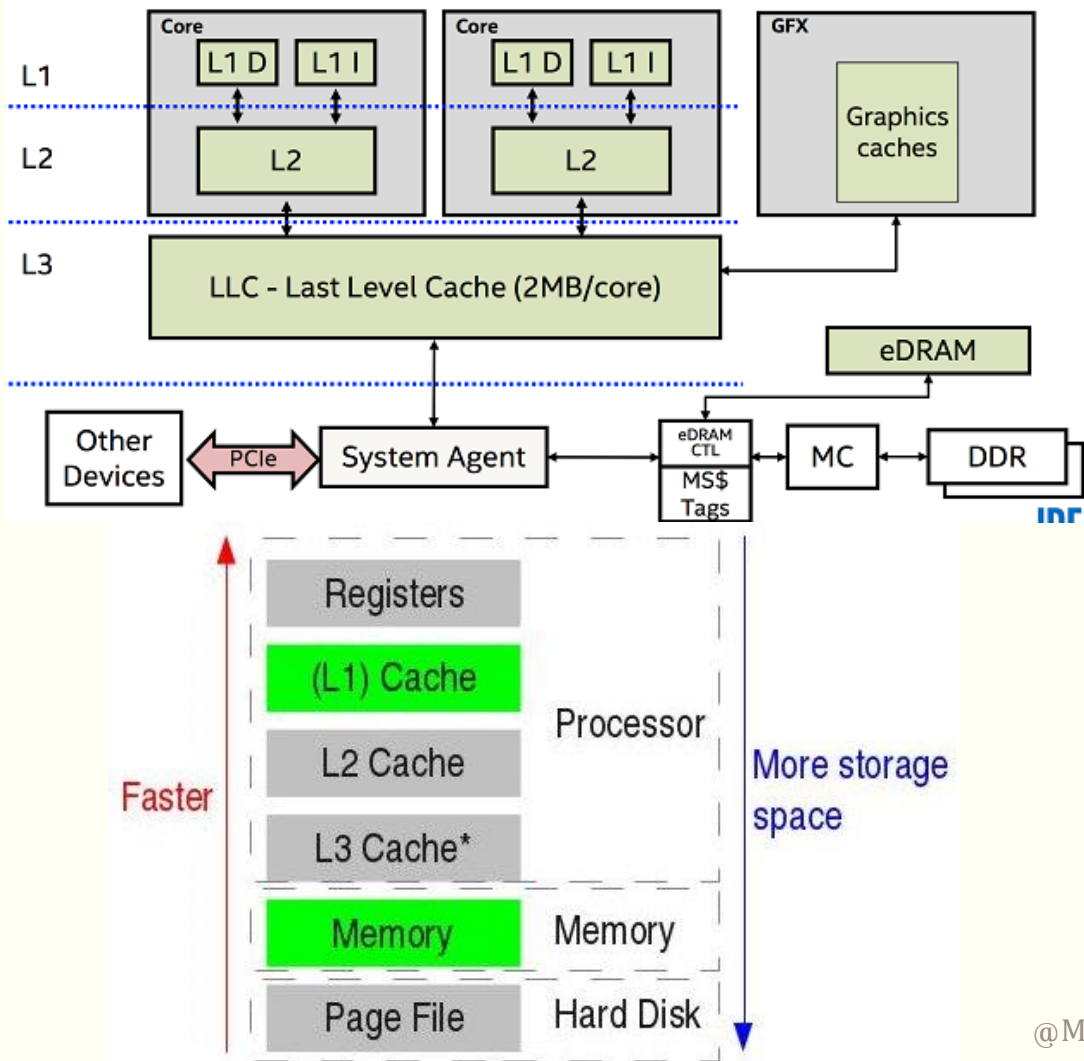
Статична пам'ять	Динамічна пам'ять
Використовує транзистор для зберігання одного біту даних	Використовує окремий конденсатор для зберігання кожного біту даних
Не потребує періодичного оновлення (refreshment) для підтримки даних	Потребує періодичного оновлення для підтримки заряду в конденсаторах для даних
Має складнішу структуру, ніж DRAM	Має простішу структуру, ніж SRAM
Дорожча за DRAM	Дешевша за SRAM
Швидша за DRAM	Повільніша за SRAM
Використовується в кеш-пам'яті	Використовується в основній пам'яті

Кеш процесора



- Кожне ядро використовує 2 рівні кешу:
 - Від 2 до 64 Кб кешу першого рівня (L1) з дуже високою швидкістю.
 - ~256 Кб кешу другого рівня (L2) із середньою швидкістю.
 - Всі ядра мають спільний кеш третього рівня (L3 – близько 8 Мб).
- Латентність пам'яті – затримки на виконання операцій зчитування, запису та копіювання.
 - Для Intel Coffee Lake (серії 8xxx, 9xxx):
 - L1 – 4 цикли (проста адреса), 5 циклів (складена (complex) адреса) – 32Кб.
 - L2 – 12 циклів – 256Кб
 - L3 – 42 цикли
 - Основна пам'ять – 42 цикли + 51нс
- Кеш-пам'яттю керує контролер, який аналізує виконувану програму та намагається передбачити, які дані та команди найімовірніше знадобляться ЦП у найближчий час.
 - При цьому можливі «влучання» (hit) та «промахи» (miss)

Формування ієрархії кешів



- Кеш L1 з'явився наприкінці 1980-х рр. як компроміс між збільшенням латентності підсистеми пам'яті та значним зниженням латентності запитів даних та інструкцій від процесора.
 - Тоді це була зовнішня SRAM-пам'ять, яка знаходилась близько до процесора на материнській платі та допомагала тримати його завантаженням роботою.
 - З часом L1-кеш розбився на дві області: для часто використовуваних даних та для популярних інструкцій.
- Кеш L2 почав додаватись (спочатку ззовні від ЦП) у відповідь на різкий ріст частот ЦП у 1990-х – на початку 2000-х рр.
 - Потім кеш інтегрувався в ЦП, як і L1 до того.
- Кеш L3 з'явився у відповідь на появу багатоядерних систем та розповсюдження багатоканальної пам'яті після 2005р.
 - Вищий рівень кешу вимагає більшого об'єму (нині L2 – кілька Мб, L3 – десятки Мб), що здорожчує ЦП.

Теми доповідей



Порівняння архітектур процесорів CISC, RISC, MISC



Порівняння гарвардської та принстонської архітектур



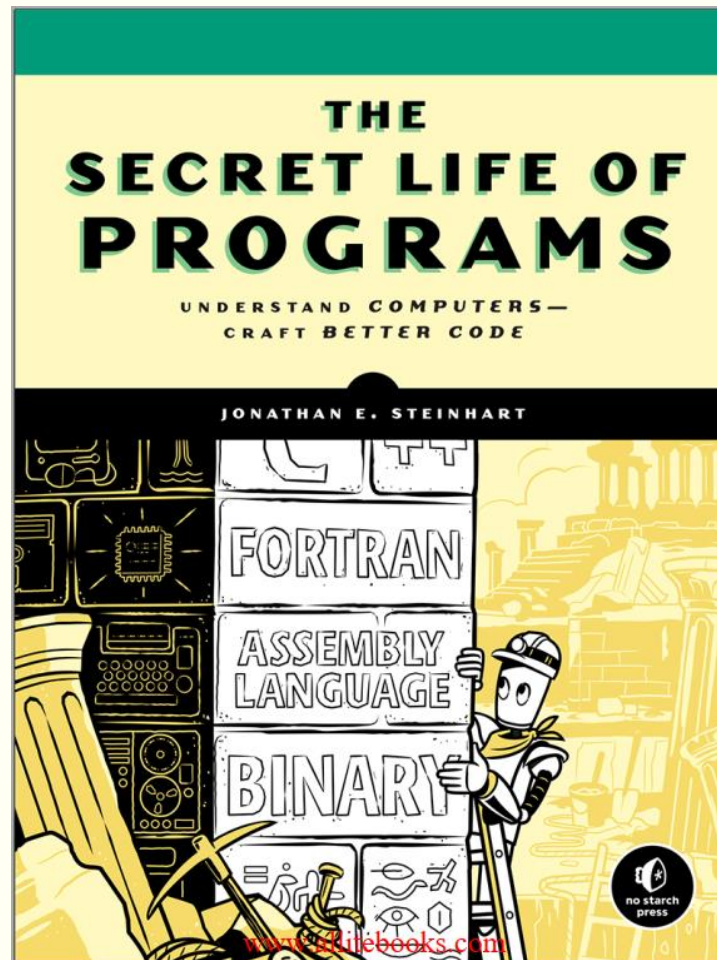
Конвеєрні структури центральних процесорів



Типи шин у сучасних комп'ютерних системах

- Більше зможете дізнатись у предметі «Архітектура ЕОМ»

Де почитати та подивитись?



coursera

Контрольні запитання?

- Як пропонує організовувати обчислювальні пристрої архітектура фон Неймана?
- Що таке машинний цикл та як він виконується?
- Опишіть ієрархію пам'яті для сучасних комп'ютерних систем.



ДЯКУЮ ЗА УВАГУ!

Наступне запитання: Представлення інформації в пам'яті комп'ютера. Типи даних.