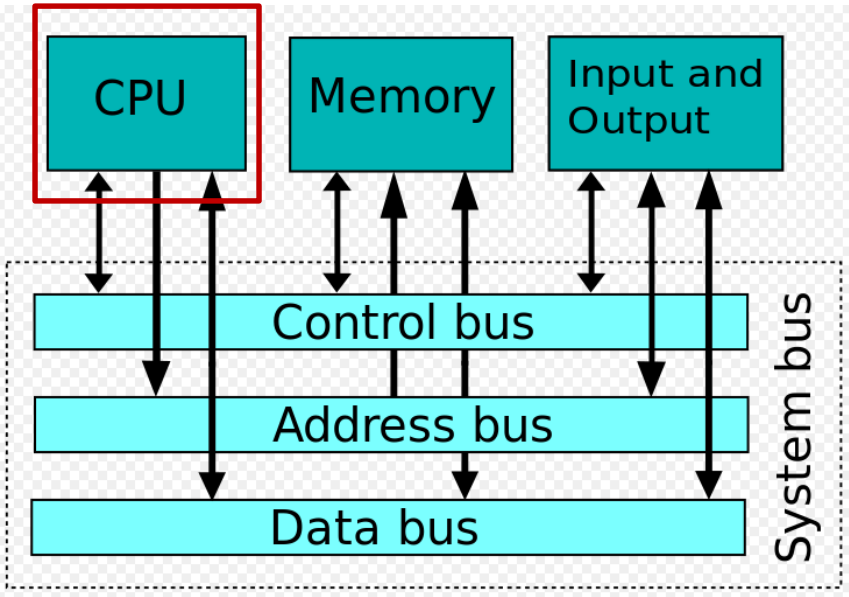
Chương 4

Bộ xử lý trung tâm (CPU)



Giảng viên: ThS. Phan Như Minh

4.3. Các phương pháp định địa chỉ

4.4. Hoạt động cơ bản của máy tính

4.5. Các kỹ thuật xử lý tiên tiến

INTEL PENTIUM 4

AMD ATHLON 64

INTEL PENTIUM M

4.3 Các phương pháp định địa chỉ

- Khái niệm về định địa chỉ (addressing)
  - Toán hạng của lệnh có thể là:
    - Một giá trị cụ thể nằm ngay trong lệnh
    - Nội dung của thanh ghi
    - Nội dung của ngăn nhớ hoặc cổng vào-ra
- Phương pháp định địa chỉ (addressing modes) là cách thức địa chỉ hóa trong trường địa chỉ của lệnh để xác định nơi chứa toán hạng

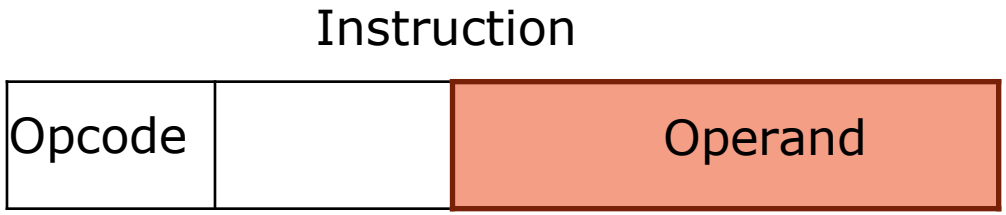
Các phương pháp định địa chỉ thông dụng

- Định địa chỉ tức thì (Immediate)
- Định địa chỉ thanh ghi
- Định địa chỉ trực tiếp
- Định địa chỉ gián tiếp qua thanh ghi
- Định địa chỉ gián tiếp qua ngăn nhớ
- Định địa chỉ dịch chuyển

Định địa chỉ tức thì

- Toán hạng nằm ngay trong Trường địa chỉ của lệnh
- Chỉ có thể là toán hạng nguồn
- Ví dụ:
  - ADD R1, 5 ; R1<- R1+5
- Không tham chiếu bộ nhớ
- Truy nhập toán hạng rất nhanh
- Dải giá trị của toán hạng bị hạn chế

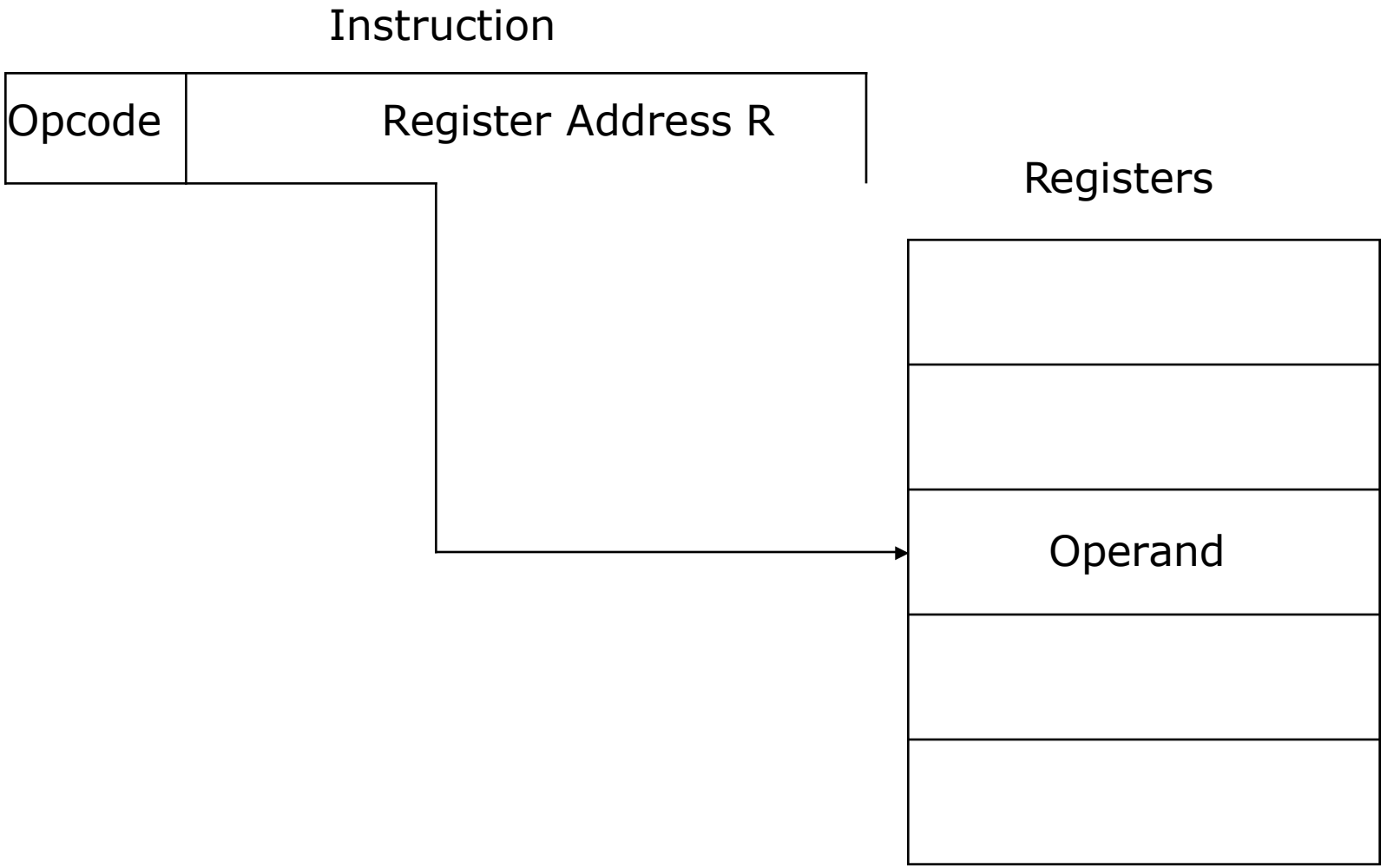
Sơ đồ định địa chỉ tức thì



Định địa chỉ thanh ghi (Register Addressing)

- Toán hạng được chứa trong thanh ghi có tên trong Trường địa chỉ
- Ví dụ:
  - ADD R1, R2 ; R1<- R1+R2
- Số lượng thanh ghi ít -> Trường địa chỉ chỉ cần ít bit
- Không tham chiếu bộ nhớ
- Truy nhập toán hạng nhanh
- Tăng số lượng thanh ghi => hiệu quả hơn

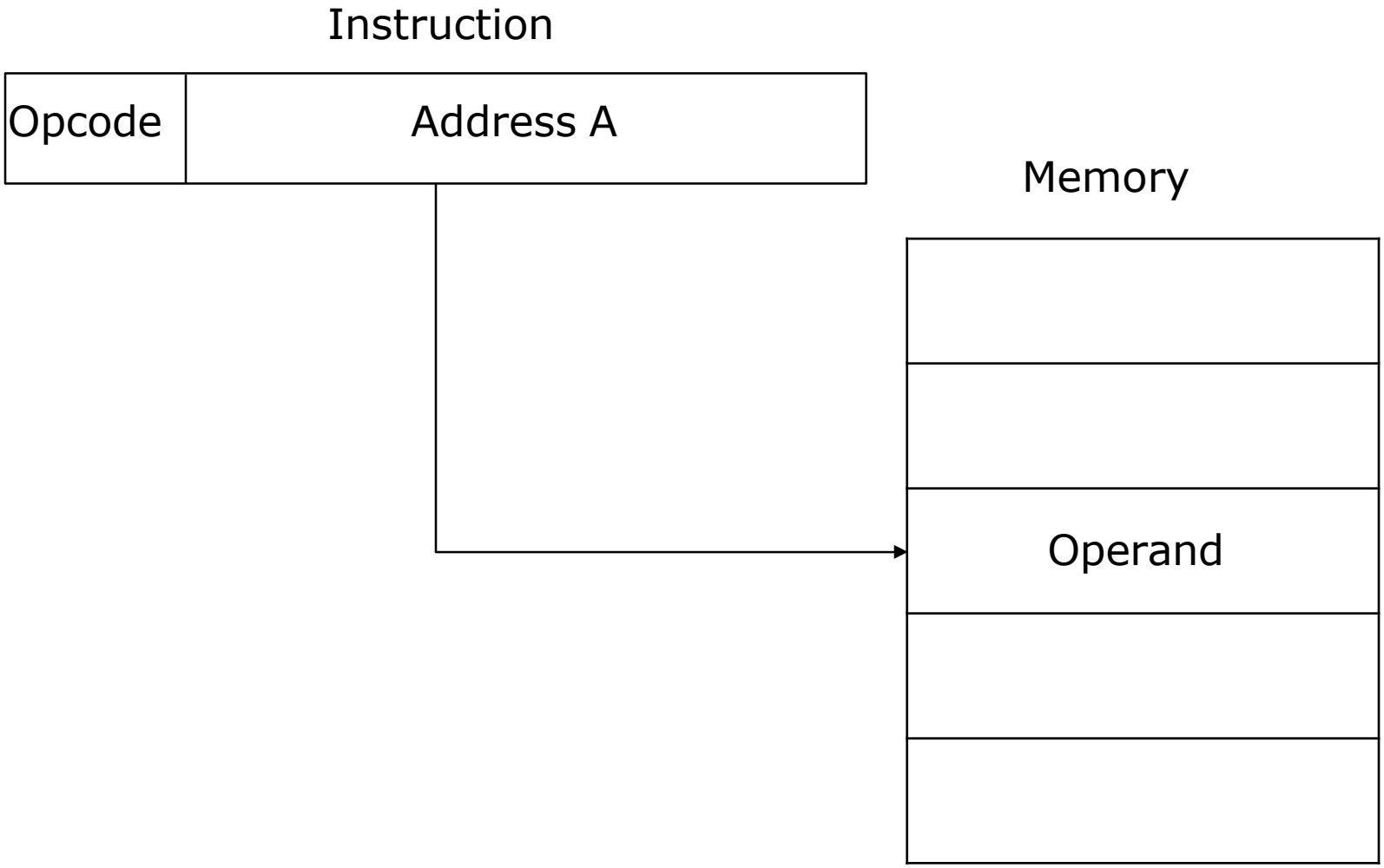
Sơ đồ định địa chỉ thanh ghi



Định địa chỉ trực tiếp (Direct Addressing)

- Toán hạng là ngăn nhớ có địa chỉ được chỉ ra trực tiếp trong Trường địa chỉ của lệnh
- Ví dụ: ADD R1, A ;R1 <- R1 + (A)
  - Cộng nội dung thanh ghi R1 với nội dung của ngăn nhớ có địa chỉ là A
  - Tìm toán hạng trong bộ nhớ ở địa chỉ A
- CPU tham chiếu bộ nhớ một lần để truy nhập dữ liệu

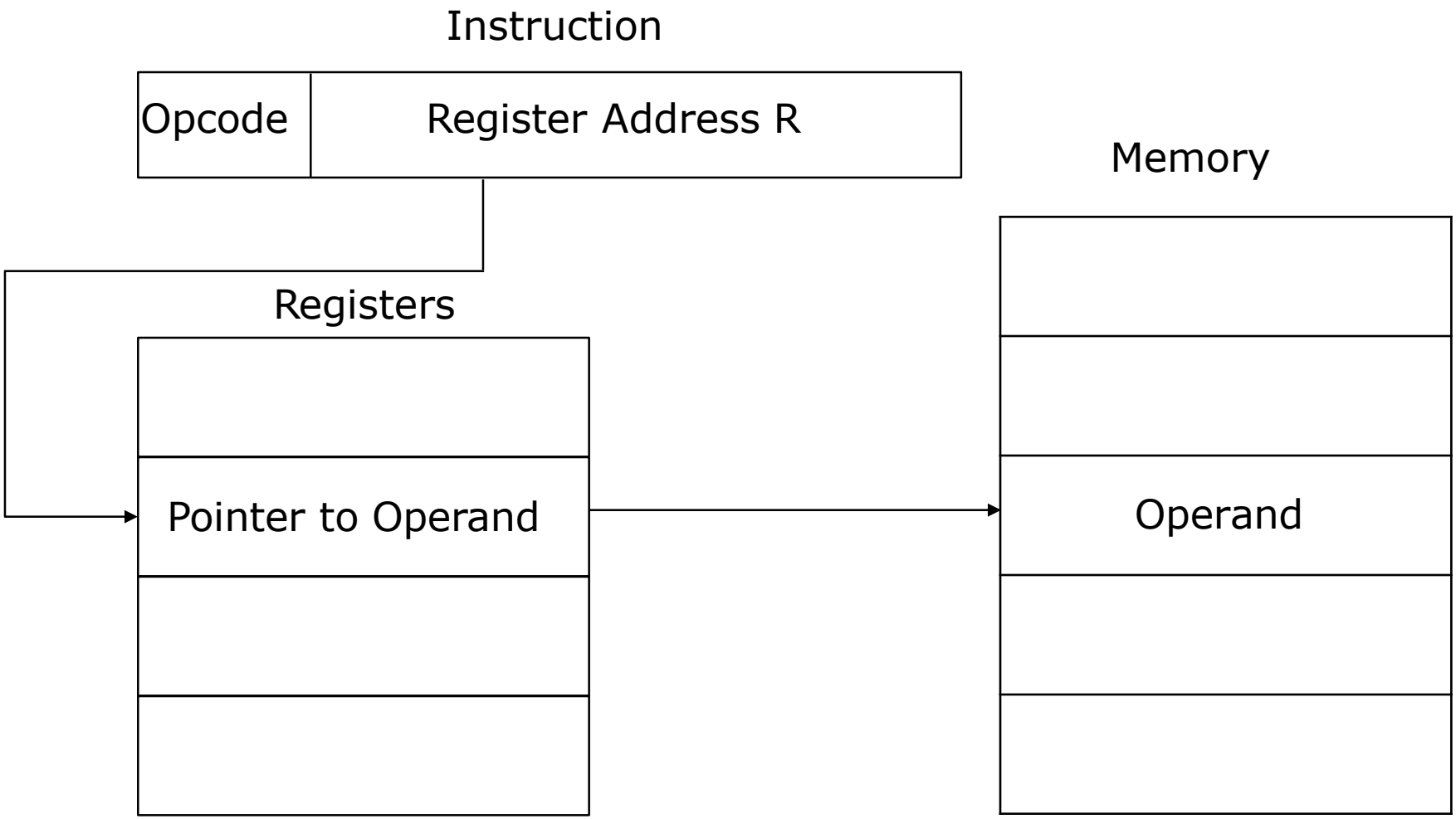
Sơ đồ định địa chỉ trực tiếp



Định địa chỉ gián tiếp qua thanh ghi

- Register Indirect Addressing
- Toán hạng là ngăn nhớ có địa chỉ nằm trong thanh ghi
- Trường địa chỉ cho biết tên thanh ghi đó
- Thanh ghi có thể là ngầm định
- Thanh ghi này được gọi là thanh ghi con trỏ
- Vùng nhớ có thể được tham chiếu là lớn (2^n), (với n là độ dài của thanh ghi)

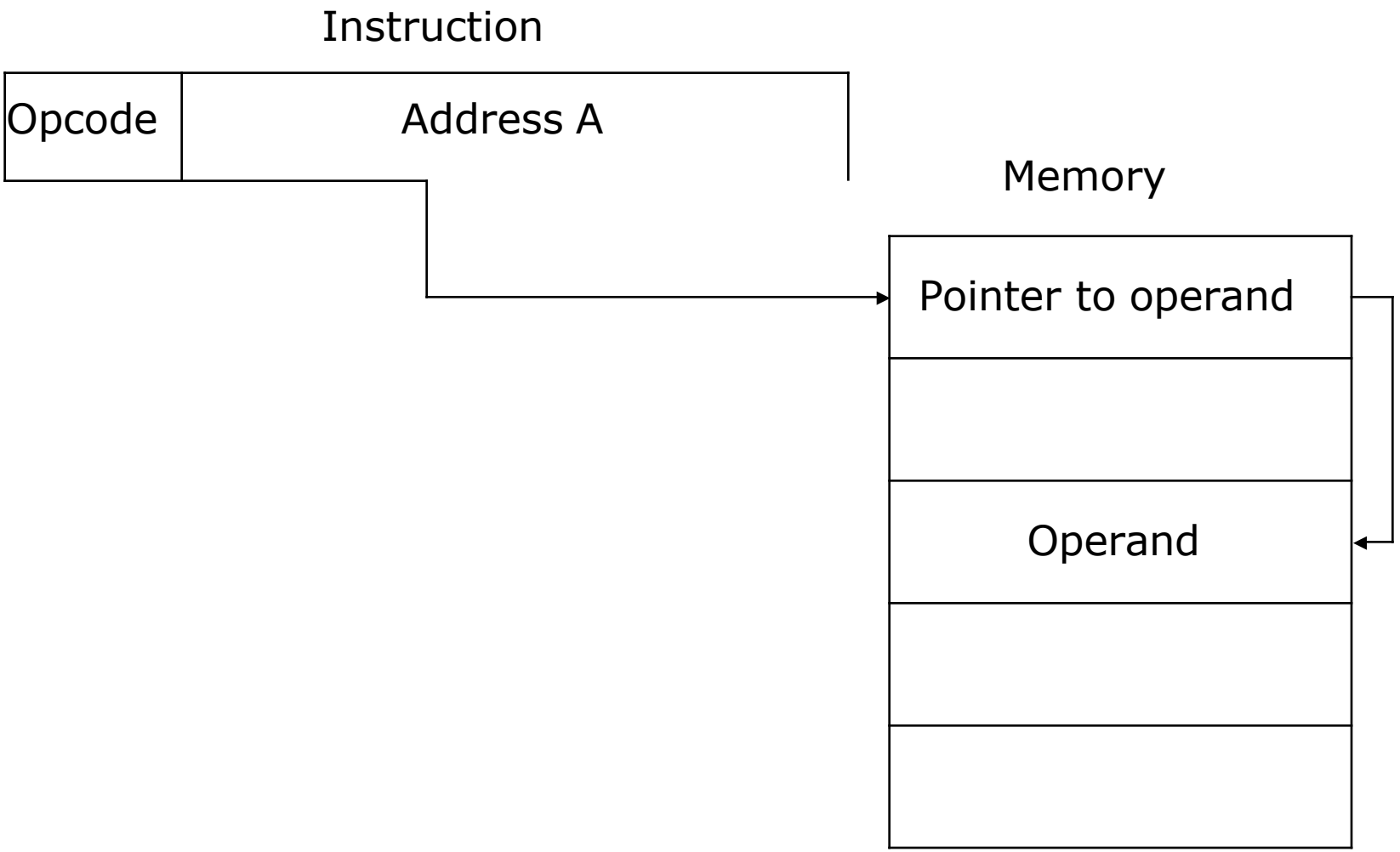
Sơ đồ định địa chỉ gián tiếp qua thanh ghi



Định địa chỉ gián tiếp qua ngăn nhớ

- Indirect Addressing
- Ngăn nhớ được trỏ bởi Trường địa chỉ của lệnh chứa địa chỉ của toán hạng
- Có thể gián tiếp nhiều lần
- Giống như khái niệm biến con trỏ và biến động trong lập trình
- CPU phải thực hiện tham chiếu bộ nhớ nhiều lần để tìm toán hạng => chậm
- Vùng nhớ có thể được tham chiếu là lớn

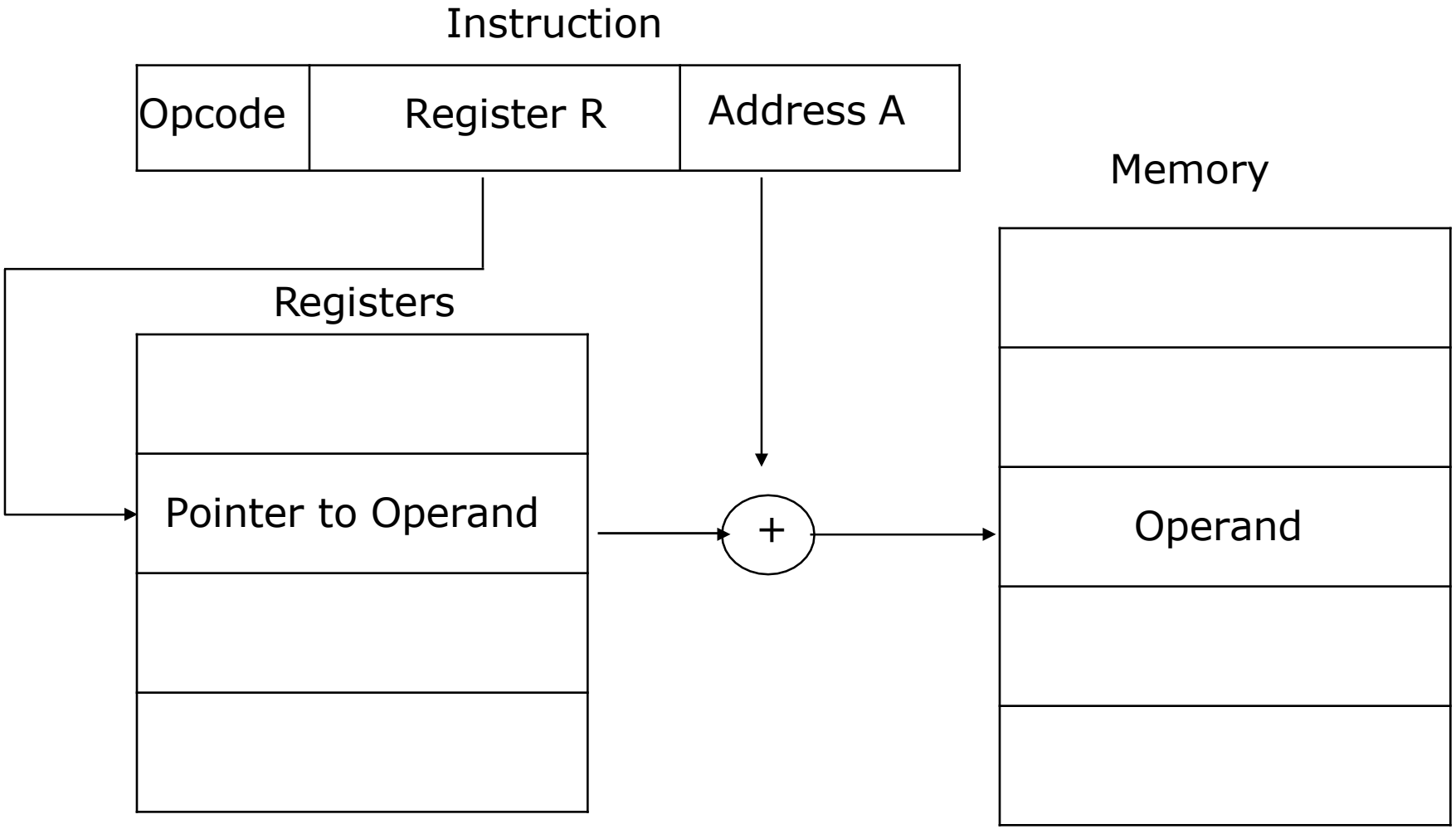
Sơ đồ định địa chỉ gián tiếp qua ngăn nhớ



Định địa chỉ dịch chuyển

- Displacement Addressing
- Để xác định toán hạng, Trường địa chỉ chứa hai thành phần:
  - Tên thanh ghi
  - Hằng số
- Địa chỉ của toán hạng = nội dung thanh ghi + hằng số
- Thanh ghi có thể được ngầm định

Sơ đồ định địa chỉ dịch chuyển



Các dạng của định địa chỉ dịch chuyển

Địa chỉ hoá tương đối với PC

- Thanh ghi là Bộ đếm chương trình PC
- Toán hạng có địa chỉ cách ngăn nhớ được trỏ bởi PC một độ lệch xác định

Định địa chỉ cơ sở

- Thanh ghi chứa địa chỉ cơ sở
- Hằng số là chỉ số

Định địa chỉ chỉ số

- Hằng số là địa chỉ cơ sở
- Thanh ghi chứa chỉ số

4.4. Hoạt động cơ bản của máy tính

1. Thực hiện chương trình

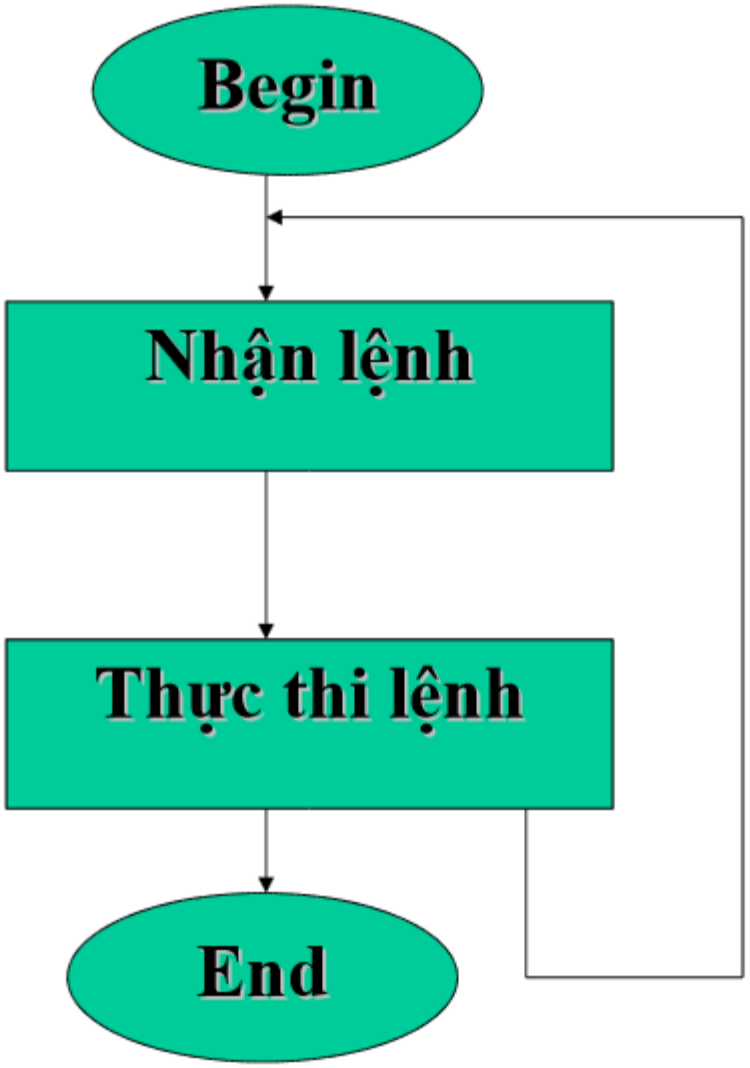
Là hoạt động cơ bản của Máy tính. Máy tính lặp đi lặp lại quá trình thực hiện lệnh gồm hai bước cơ bản:

- ✓ Nhận lệnh (Fetch)
- ✓ Thực hiện lệnh (Execute)

Thực hiện chương trình dừng khi:

- ✓ Mất nguồn
- ✓ Gặp lệnh dừng
- ✓ Gặp tình huống không giải quyết được(lỗi)

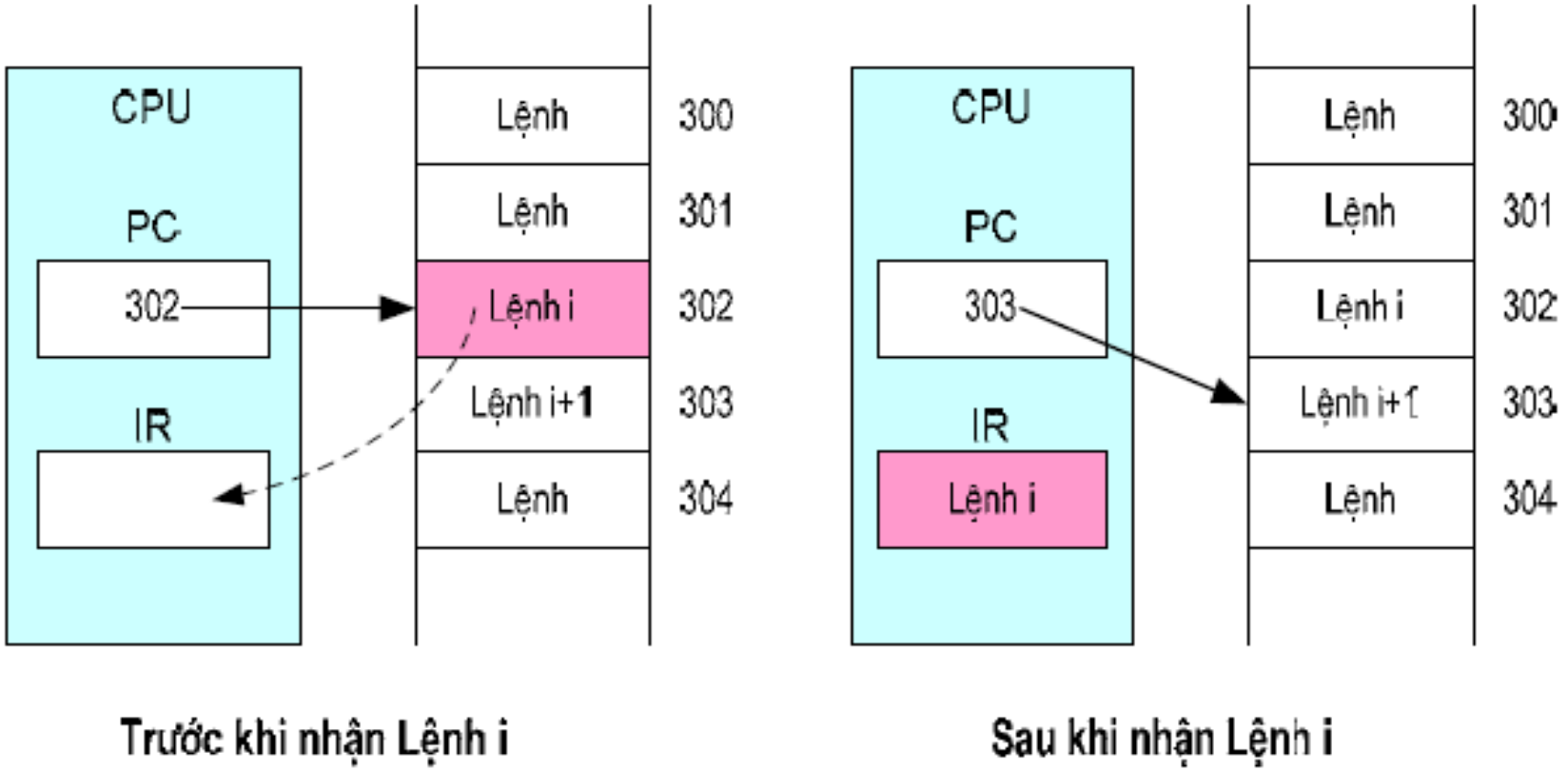
Chu trình lệnh



Nhận lệnh

- Bắt đầu mỗi chu trình lệnh, CPU nhận lệnh từ bộ nhớ chính.
- Bộ đếm chương trình PC (Program Counter) của CPU giữ địa chỉ của lệnh sẽ được nhận.
- CPU nhận lệnh từ ngăn nhớ được trỏ bởi PC.
- Lệnh được nạp vào thanh ghi lệnh IR (Instruction Register).
- Sau khi lệnh được nhận vào, nội dung PC tự động tăng để trỏ sang lệnh kế tiếp.

Nhận lệnh



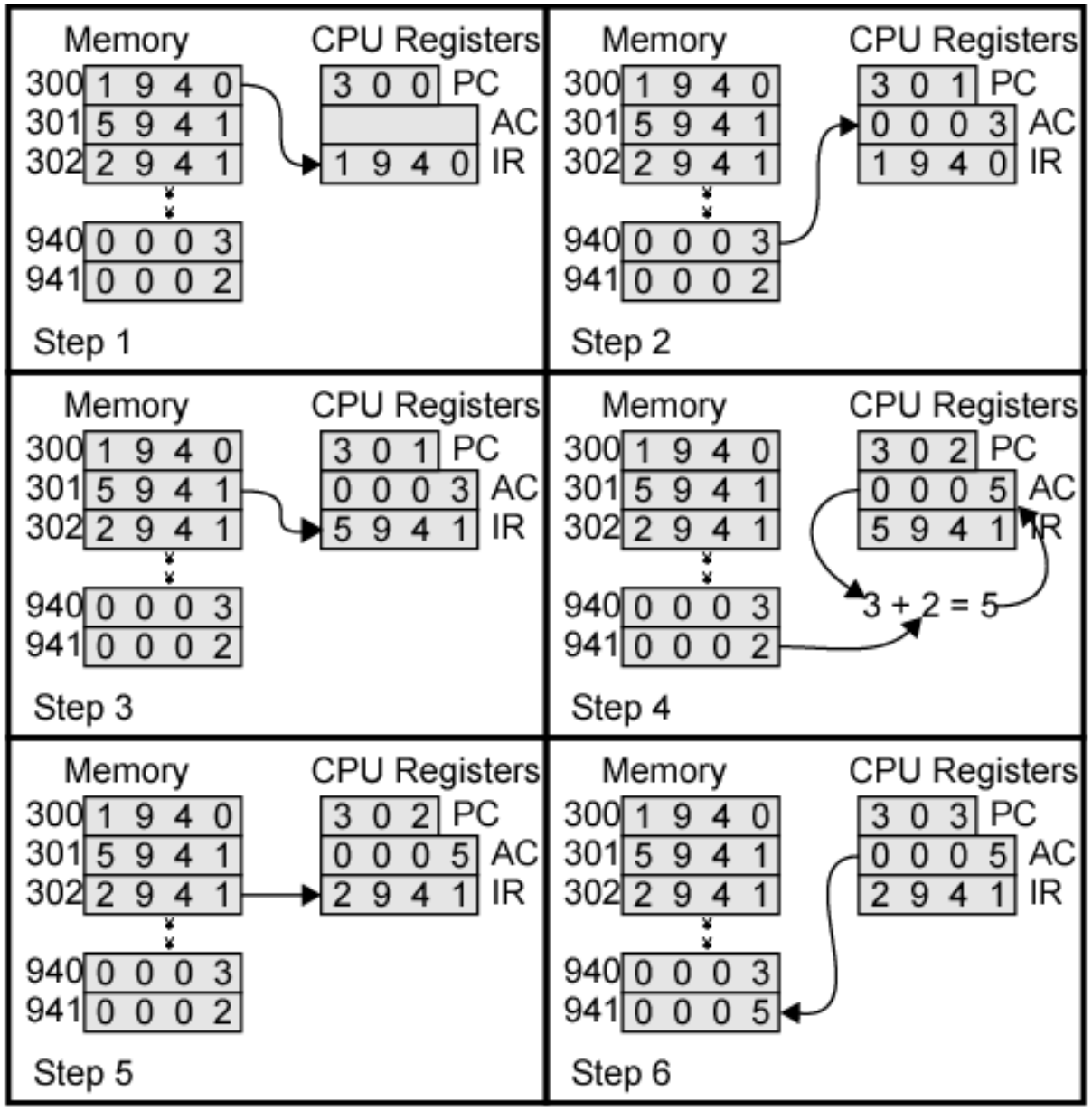
Trước khi nhận Lệnh i

Sau khi nhận Lệnh i

Thực hiện lệnh

- Bộ xử lý giải mã chỉ thị đã được nhận và phát tín hiệu điều khiển thực hiện **thao tác** mà chỉ thị yêu cầu.
- Các kiểu **thao tác** của chỉ thị:
  - Trao đổi dữ liệu giữa CPU và bộ nhớ chính
  - Trao đổi dữ liệu giữa CPU và IO Module
  - Xử lý dữ liệu: thực hiện các phép toán số học hoặc phép toán logic với các dữ liệu
  - Điều khiển rẽ nhánh
  - Kết hợp các thao tác trên

Minh họa 1



Program Counter (PC)
Instruction Register (IR)
Accumulator (AC)

0001 = Load AC from Memory
0010 = Store AC to Memory
0101 = Add to AC from Memory

Minh họa 2

Evaluated **d = a + b × c,** when **a** = 5, **b** = 3 and **c** = 11

Program Pseudocode:	Assembly Code:	Meaning:
1. Multiply **b** with **c**	MOV R1, [1058H]	R1 ← [1058H]
	MOV R2, [1059H]	R2 ← [1059H]
	MUL R1, R2	R1 ← R1 * R2
2. Add the result of multiplication to **a**	MOV R3, [1057H]	R3 ← [1057H]
	ADD R1, R3	R1 ← R1 + R3
3. Save the result as **d**	STR R1, [105AH]	R1 → [105AH]

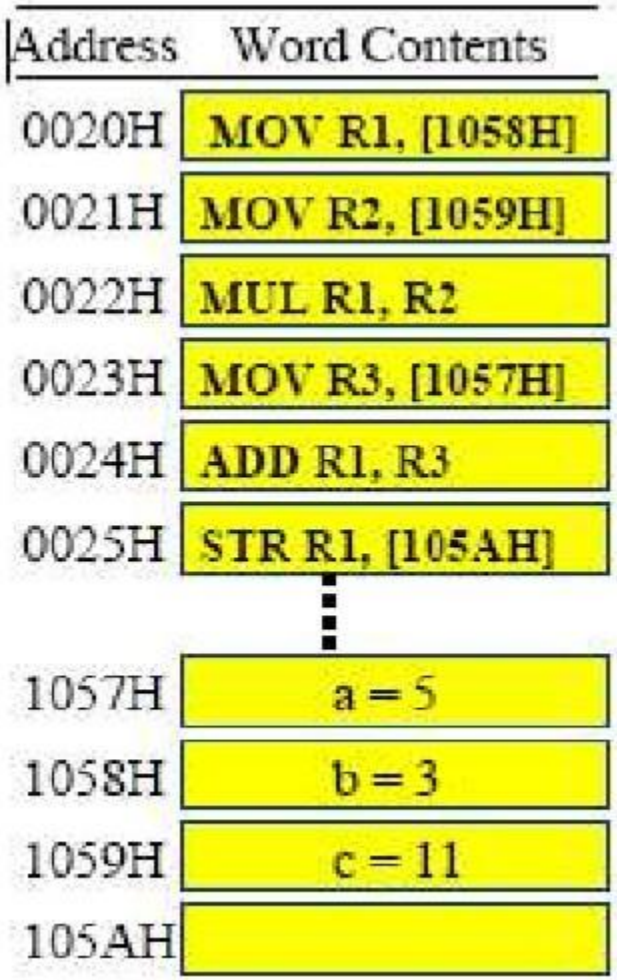
Minh họa 3

Following arithmetic expression to be evaluated for d,

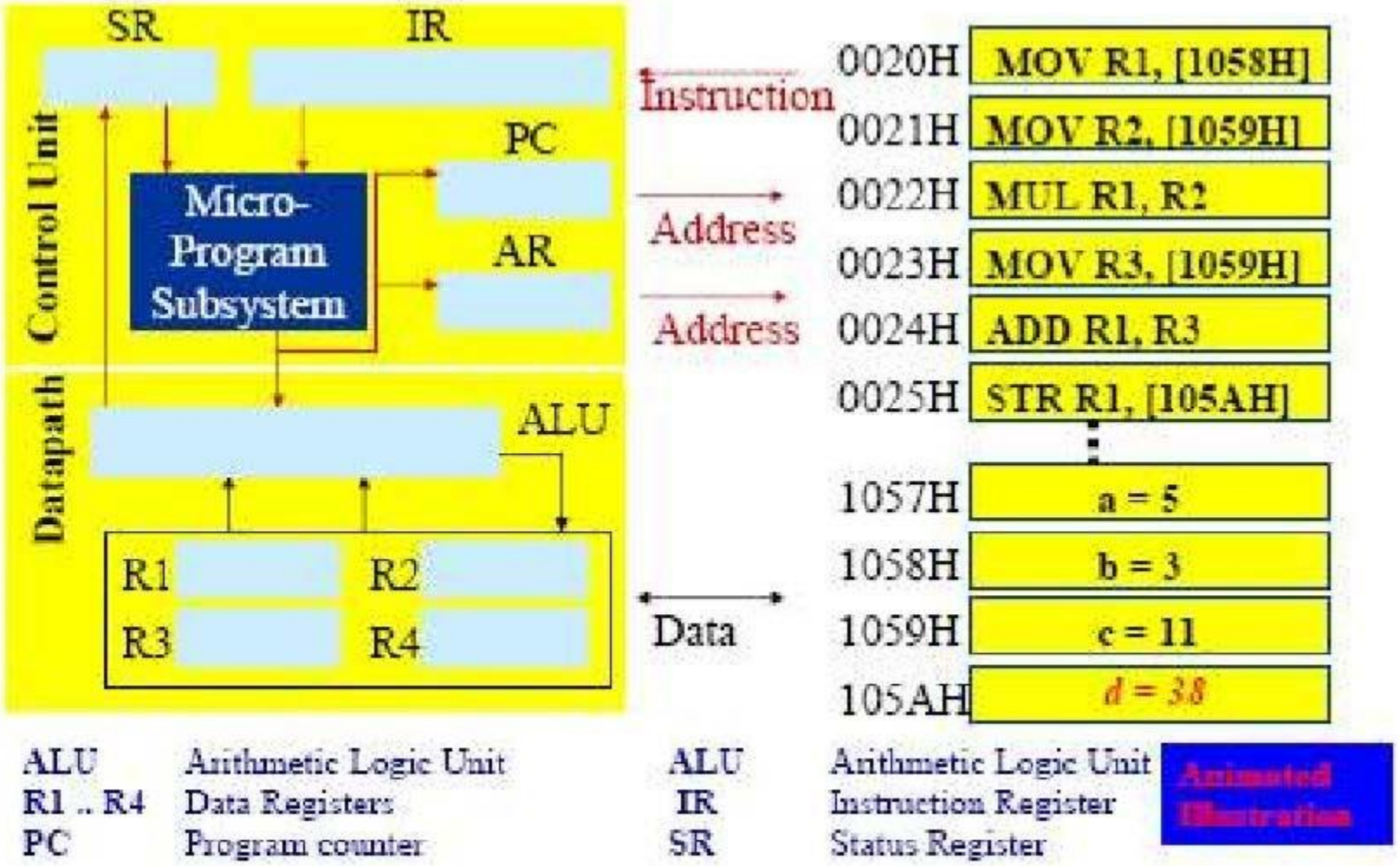
$$d = a + b * c$$

when a = 5, b = 3 and c = 11

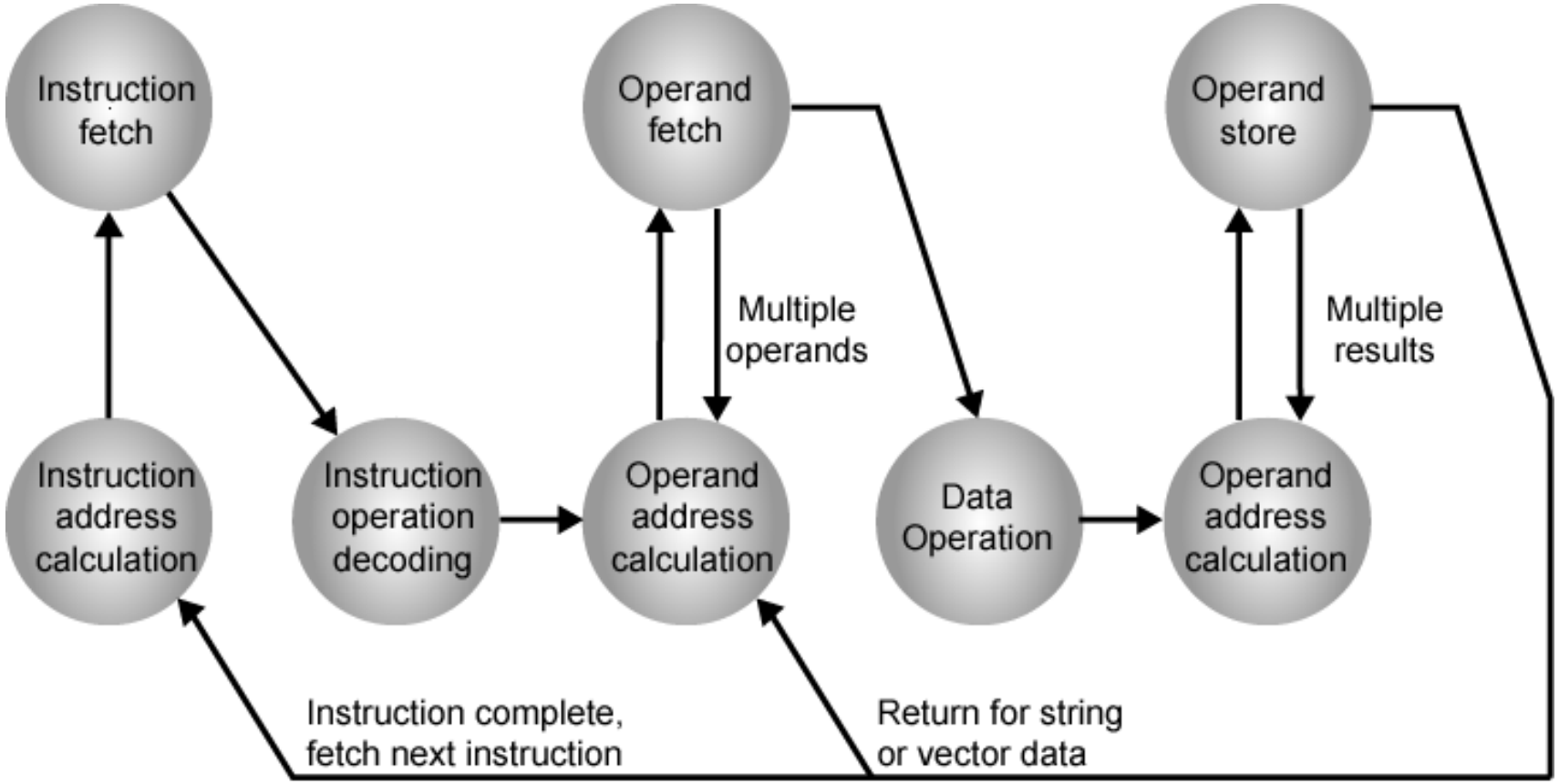
Memory map shows where the program and input data are loaded



Minh họa 3


SR
IR
Control Unit
PC
Micro-
Program
Subsystem
AR
Datapath
ALU
R1
R2
R3
R4
Instruction
Address
Address
Data
0020H
MOV R1, [1058H]
0021H
MOV R2, [1059H]
0022H
MUL R1, R2
0023H
MOV R3, [1059H]
0024H
ADD R1, R3
0025H
STR R1, [105AH]
1057H
a = 5
1058H
b = 3
1059H
c = 11
105AH
d = 3.8
ALU Arithmetic Logic Unit
R1 .. R4 Data Registers
PC Program counter
ALU Arithmetic Logic Unit
IR Instruction Register
SR Status Register
Animated Illustration

Sơ đồ các bước thực hiện lệnh



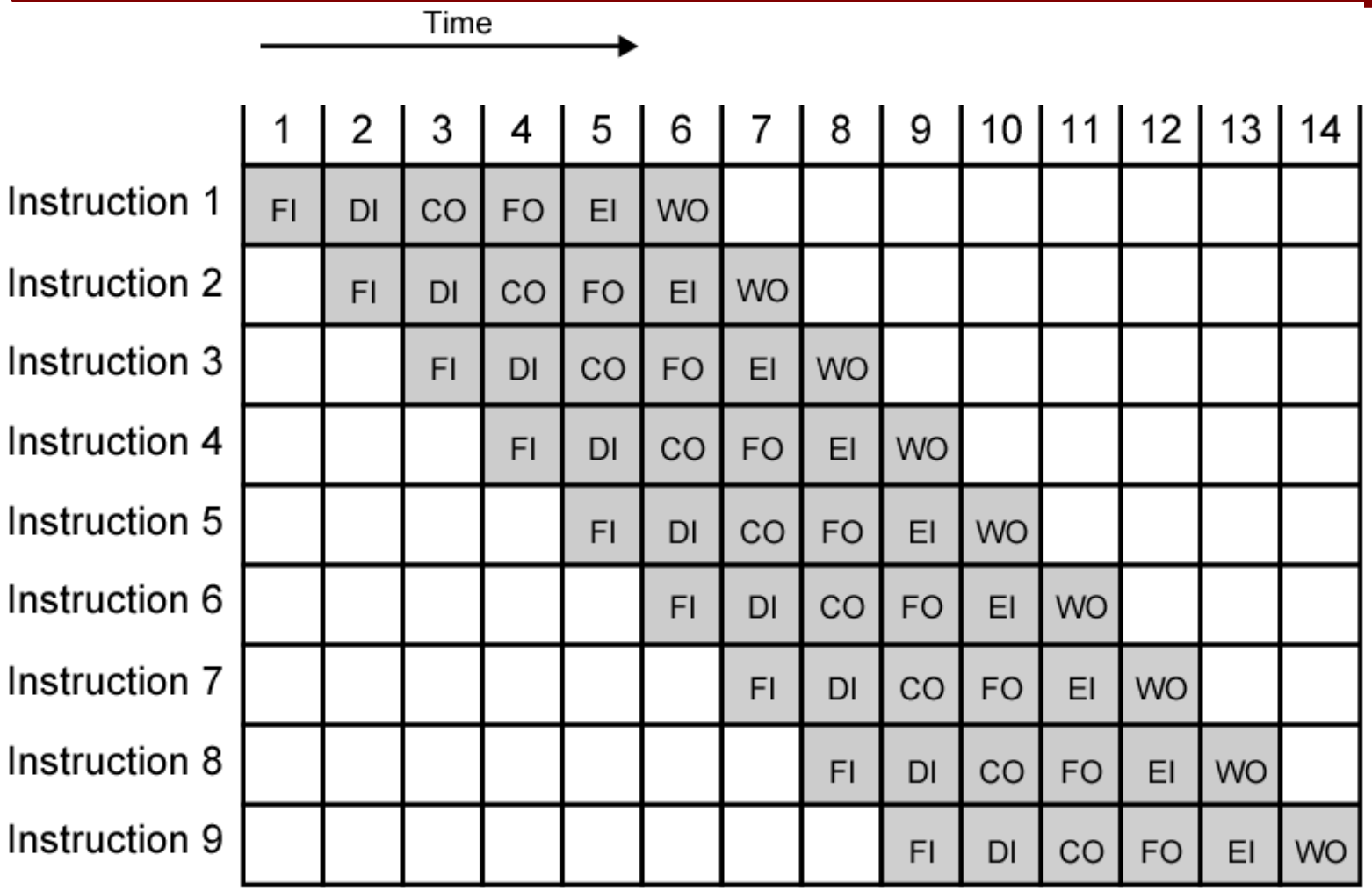
4.5 Các kỹ thuật xử lý tiên tiến

- Kỹ thuật xử lý đường ống (pipline)
- Kỹ thuât xử lý đa phân luồng (multistream)
- Công nghệ siêu phân luồng (Hyper-Threading)
- Sư dụng bộ xử lý đa lõi (Multicore)

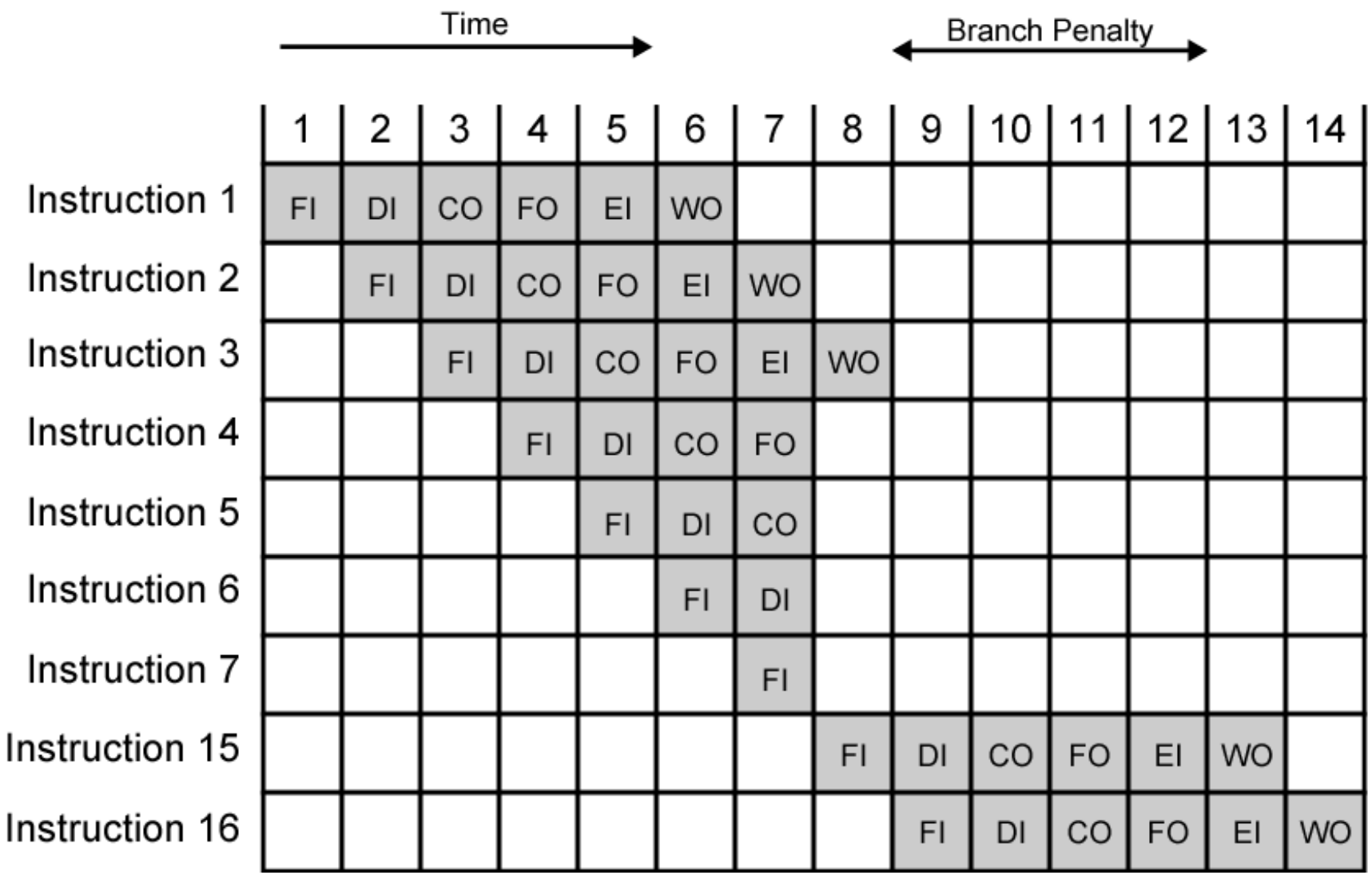
1. Kỹ thuật đường ống lệnh (Instruction Pipelining)

- Chia chu trình lệnh thành các công đoạn và cho phép thực hiện gối lên nhau (tương tự dây chuyền lắp ráp)
- Chẳng hạn, các giai đoạn thực hiện một lệnh là:
  - Tải lệnh (IF: Instruction Fetch)
  - Giải mã (ID: Instruction Decode)
  - Tính địa chỉ toán hạng (CO: Calculate Operand Address)
  - Nhận toán hạng (FO: Fetch Operands)
  - Thực hiện lệnh (EI: Execute Instruction)
  - Ghi toán hạng (WO: Write Operands)

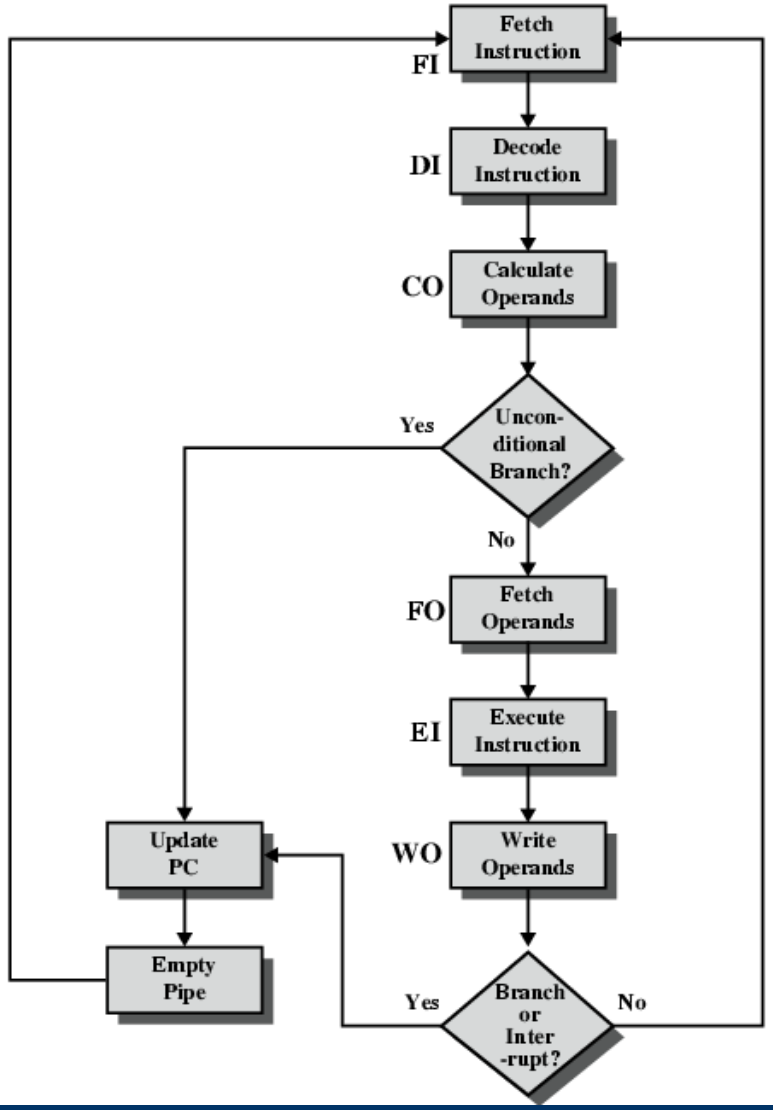
Biểu đồ thời gian của đường ống lệnh



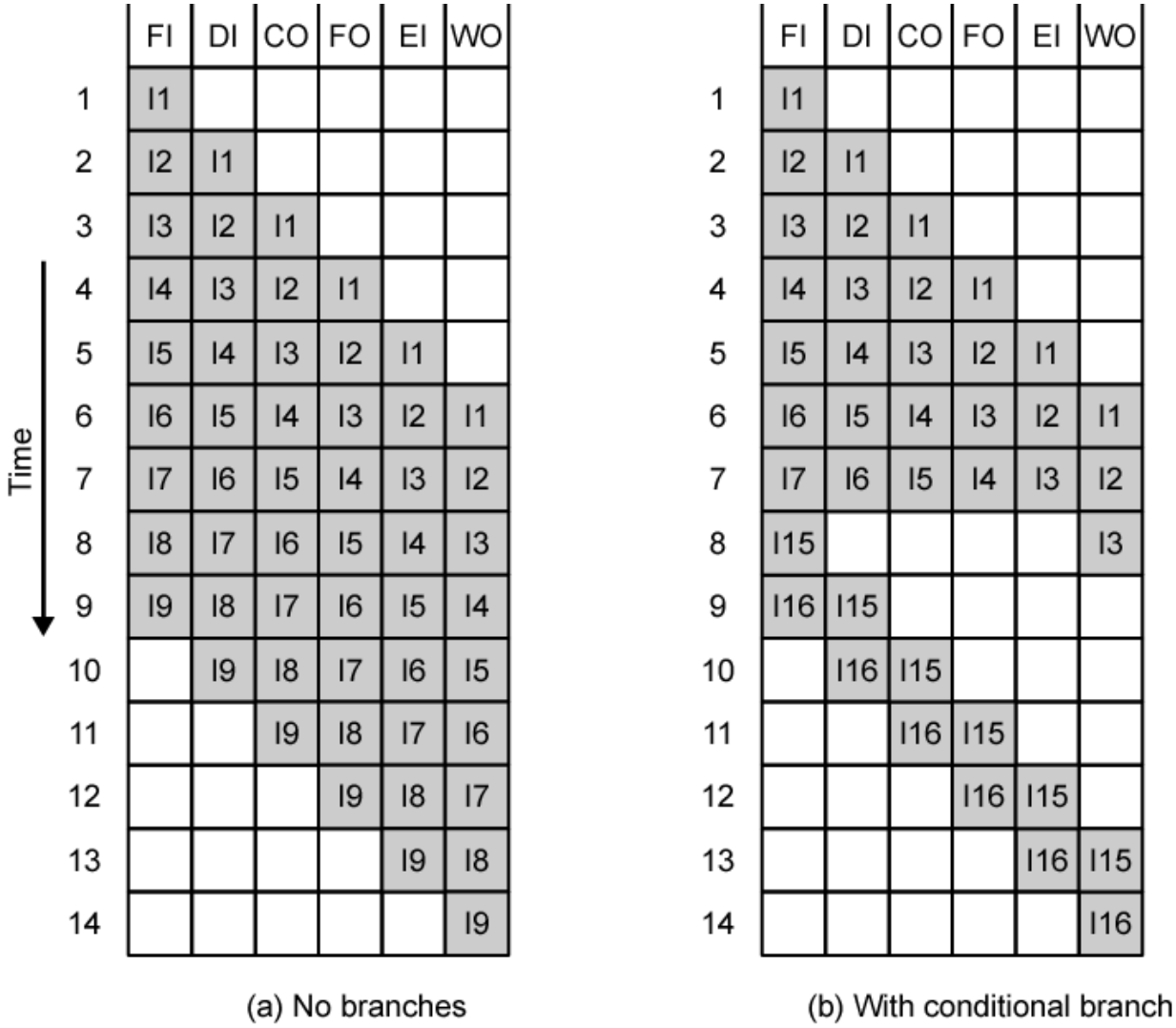
Trở ngại do lệnh rẽ nhánh



Six Stage Instruction Pipeline



Dạng mô tả khác của pipelining



(a) No branches

(b) With conditional branch

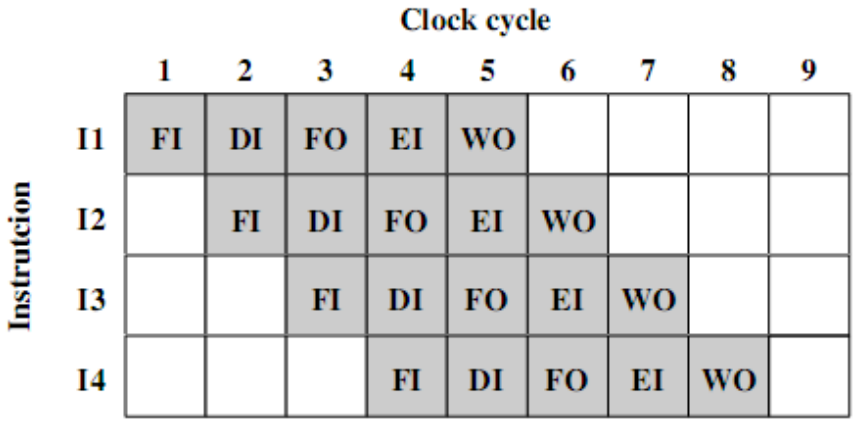
Các trở ngại của cơ chế ống dẫn

- Trở ngại về cấu trúc: do nhiều công đoạn dùng chung một tài nguyên
- Trở ngại về dữ liệu: lệnh sau sử dụng dữ liệu kết quả của lệnh trước
- Trở ngại về điều khiển: do rẽ nhánh gây ra

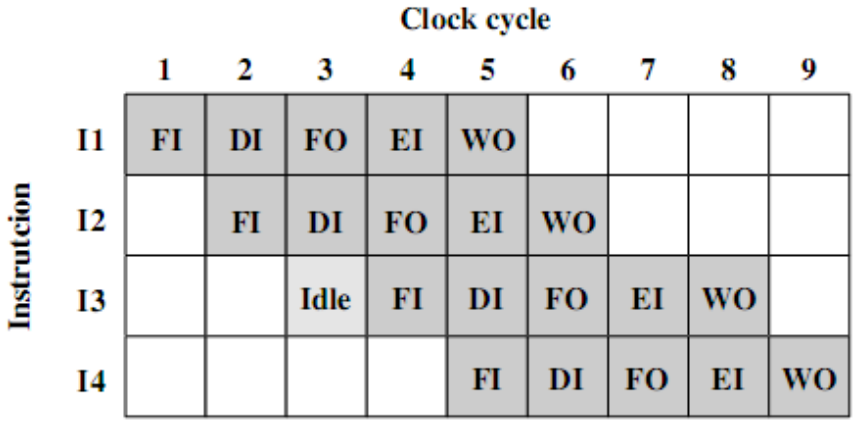
Trở ngại về cấu trúc

- Hai hay nhiều lệnh sẵn sàng trong ống dẫn tranh chấp cùng một tài nguyên => các lệnh này sẽ được thực hiện tuần tự thay vì được thực hiện song song.

Ví dụ



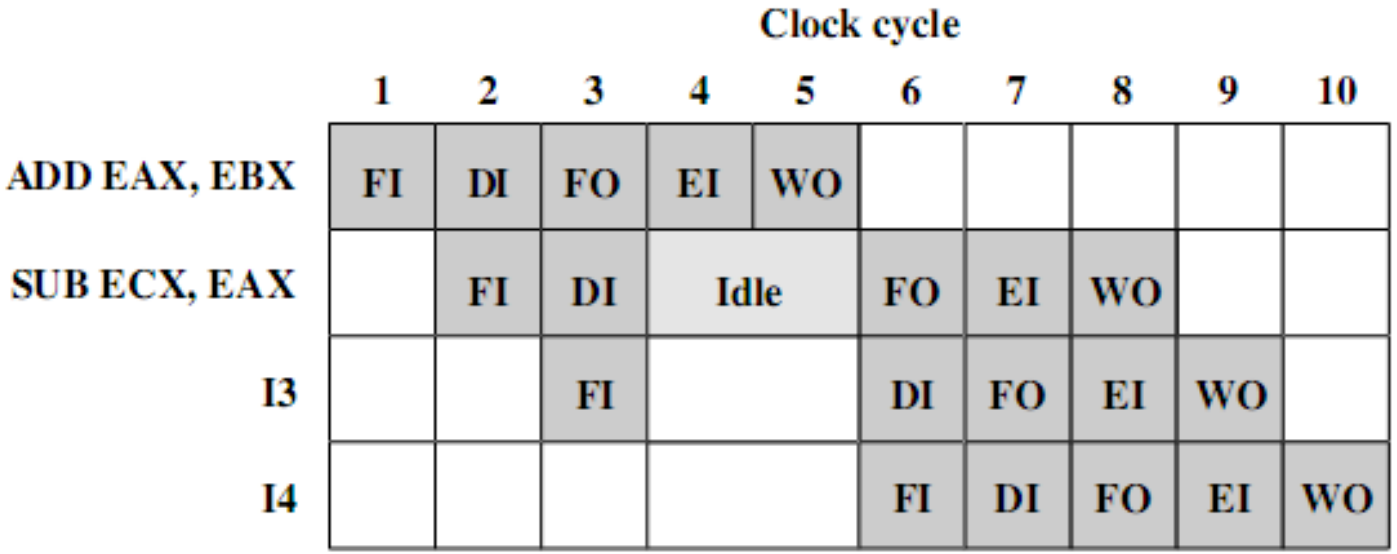
(a) Five-stage pipeline, ideal case



(b) I1 source operand in memory

Trở ngại về dữ liệu

- Ví dụ:
  - ADD EAX, EBX /* EAX = EAX + EBX
  - SUB ECX, EAX /* ECX = ECX – EAX
- Cách giải quyết



Một số dạng trở ngại về dữ liệu

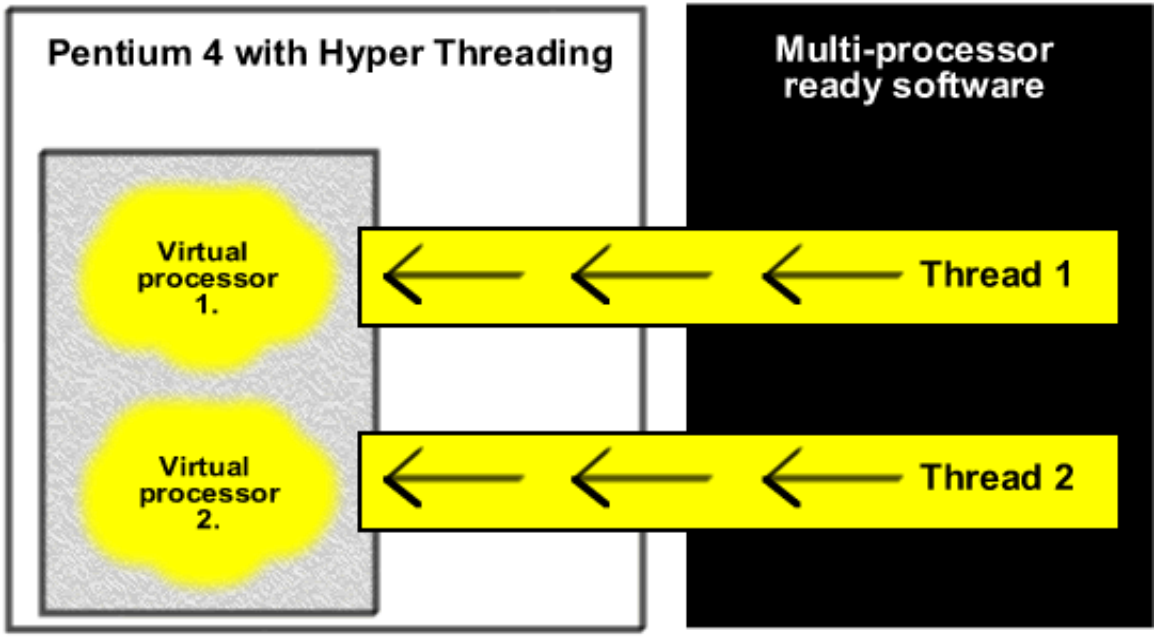
- Read after write (RAW)
  - Một lệnh ghi vào thanh ghi (ô nhớ) và lệnh liền sau đó đọc thanh ghi (ô nhớ) đó.
  - Nếu việc đọc được thực hiện trước việc ghi?
- Write after read (WAR):
  - Một lệnh đọc thanh ghi (ô nhớ) và lệnh liền sau đó ghi vào thanh ghi (ô nhớ) đó.
  - Nếu việc ghi được thực hiện trước việc đọc?
- Write after write (WAW):
  - Hai lệnh cùng ghi vào một thanh ghi (ô nhớ).
  - Nếu hành động ghi được thực hiện sai thứ tự?

Trở ngại về điều khiển

- Do lệnh rẽ nhánh (branch) gây ra.
- Một số cách tiếp cận về giải quyết trở ngai do điều khiển:
  - Multiple Streams
  - Prefetch Branch Target
  - Loop buffer
  - Branch prediction
  - Delayed branching

2. Công nghệ siêu phân luồng (Hyper-Threading)

- Cho phép nhiều tuyến đoạn trong một chương trình được thực hiện đồng thời
- Xuất hiện trong bộ VXL Pentium 4 (Prescott)
- Bộ VXL P4 có thể hoạt động như 2 bộ xử lý trong một máy tính

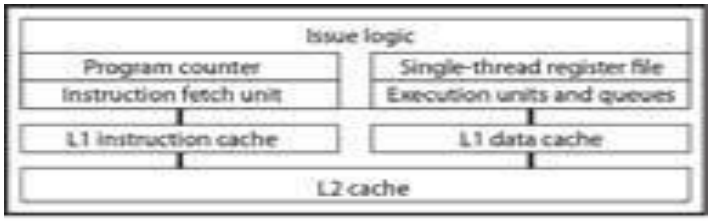


3. Bộ xử lý đa lõi (Multicores)

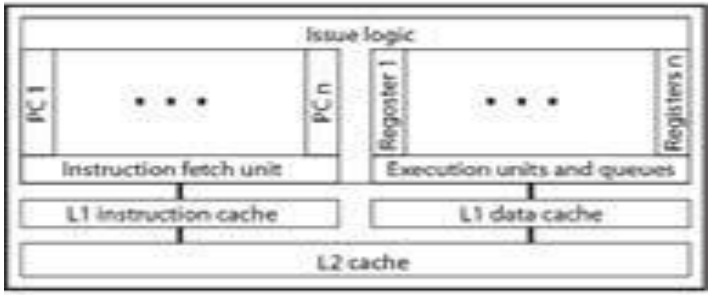
- Do tần số đồng hồ giới hạn ở 4GHz
- Tạo ra nhiều CPU trên cùng một chip để tăng hiệu năng của máy tính (tăng khả năng phân luồng bằng phần cứng)
- AMD (Opteron) và Intel P4 (Smithfield) hay Pentium D

Bộ xử lý đa lõi (multicores)

- Thay đổi của bộ xử lý:
  - Tuần tự
  - Pipeline
  - Siêu vô hướng
  - Đa luồng
  - Đa lõi



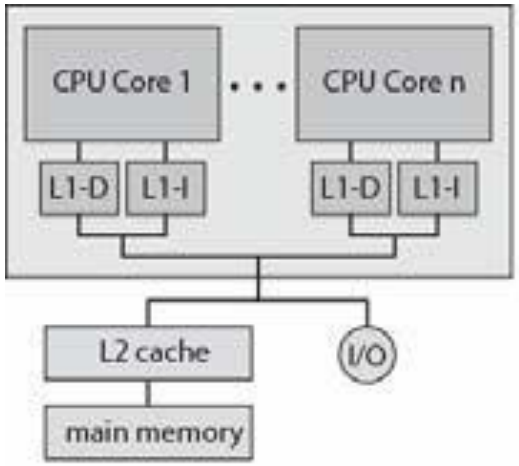
(a) Superscalar



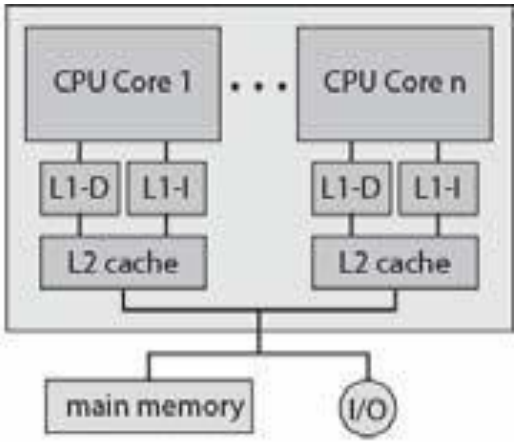
(b) Simultaneous multithreading

(c) Multicore

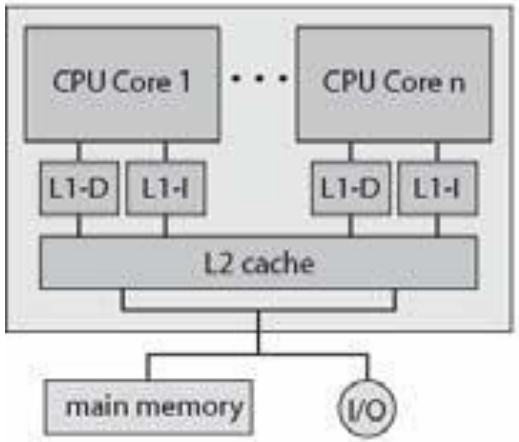
Một số dạng bộ xử lý đa lõi



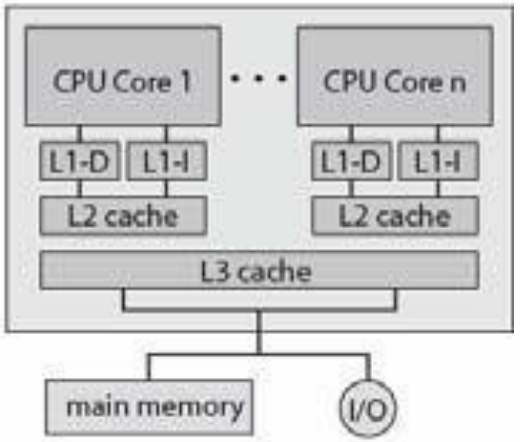
(a) Dedicated L1 cache



(b) Dedicated L2 cache



(c) Shared L2 cache



(d) Shared L3 cache

Chip đa lõi của Intel

- Pentium Extreme Edition và Pentium D
  - 4/2005
  - Phát triển từ Intel P4 Prescott

- ✓ Pentium D
  - Tốc độ: 2,8 GHz - 3,2GHz
  - FSB: 800MHz
  - EM64T 64-bit
  - Công nghệ 90 nm
  - Cache L2: 2MB
  - Socket T (LGA775)

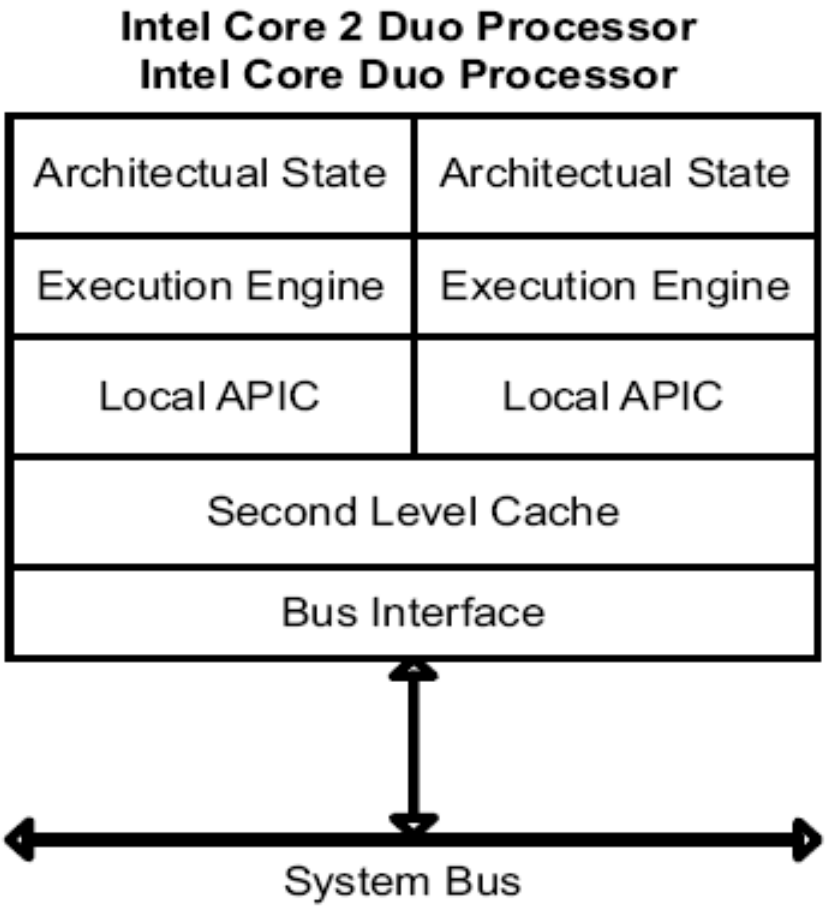
- ✓ Pentium Extreme Edition
  - ✓ Giống Pentium D
  - ✓ Hỗ trợ siêu phân luồng (HT Technology)
  - ✓ Cho phép thay đổi số nhân

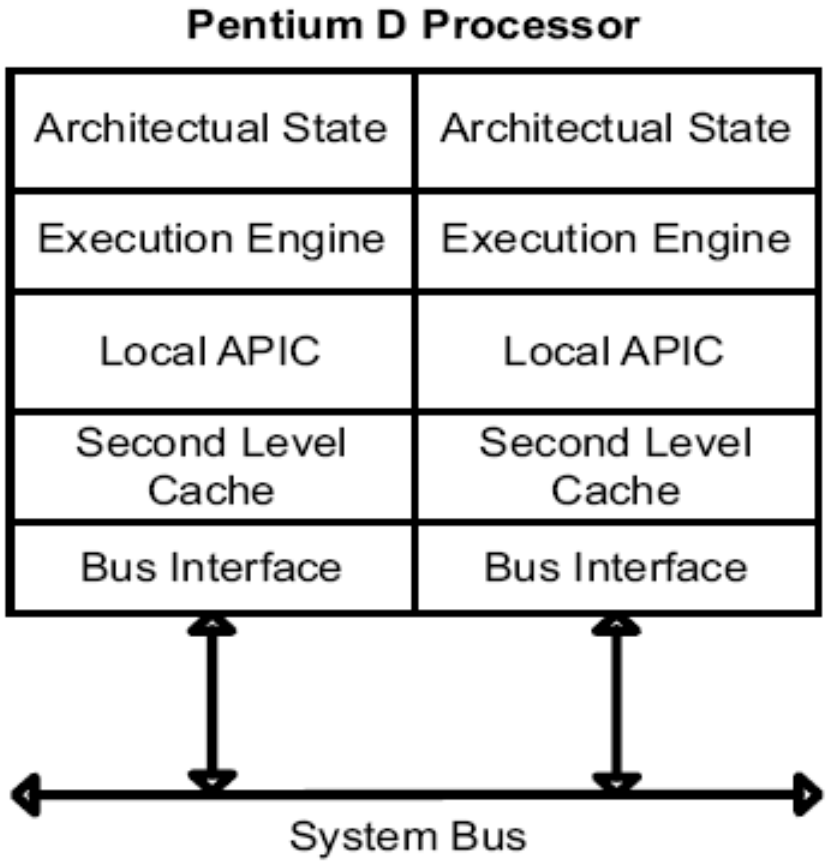
Core Duo và Core 2 Duo

Đặc điểm chính của họ Core 2

- Vi cấu trúc Core
- Bộ nhớ Cache L1 cho lệnh 32 KB và dữ liệu 32KB cho mỗi lõi .
- Công nghệ Dual-core .
- Sản xuất dựa trên xử lí 65 nm .
- Socket 775.
- Tốc độ Bus 800 MHz (200 MHz x 4) hoặc 1,066 MHz (266 MHz x 4).
- Bộ nhớ Cache L2 hợp nhất 2 MB hoặc 4 MB .
- Hỗ trợ công nghệ Intel Virtualization (trừ Core 2 Duo E4300)
- Hỗ trợ côngnghệ Intel EM64T .
- Tập lệnh SSE3 .
- Hỗ trợ Execute Disable Bit.
- Khả năng quản lí nguồn thông minh - Intelligent Power Capability.
- Hỗ trợ công nghệ Enhanced SpeedStep

Chip đa lõi của Intel

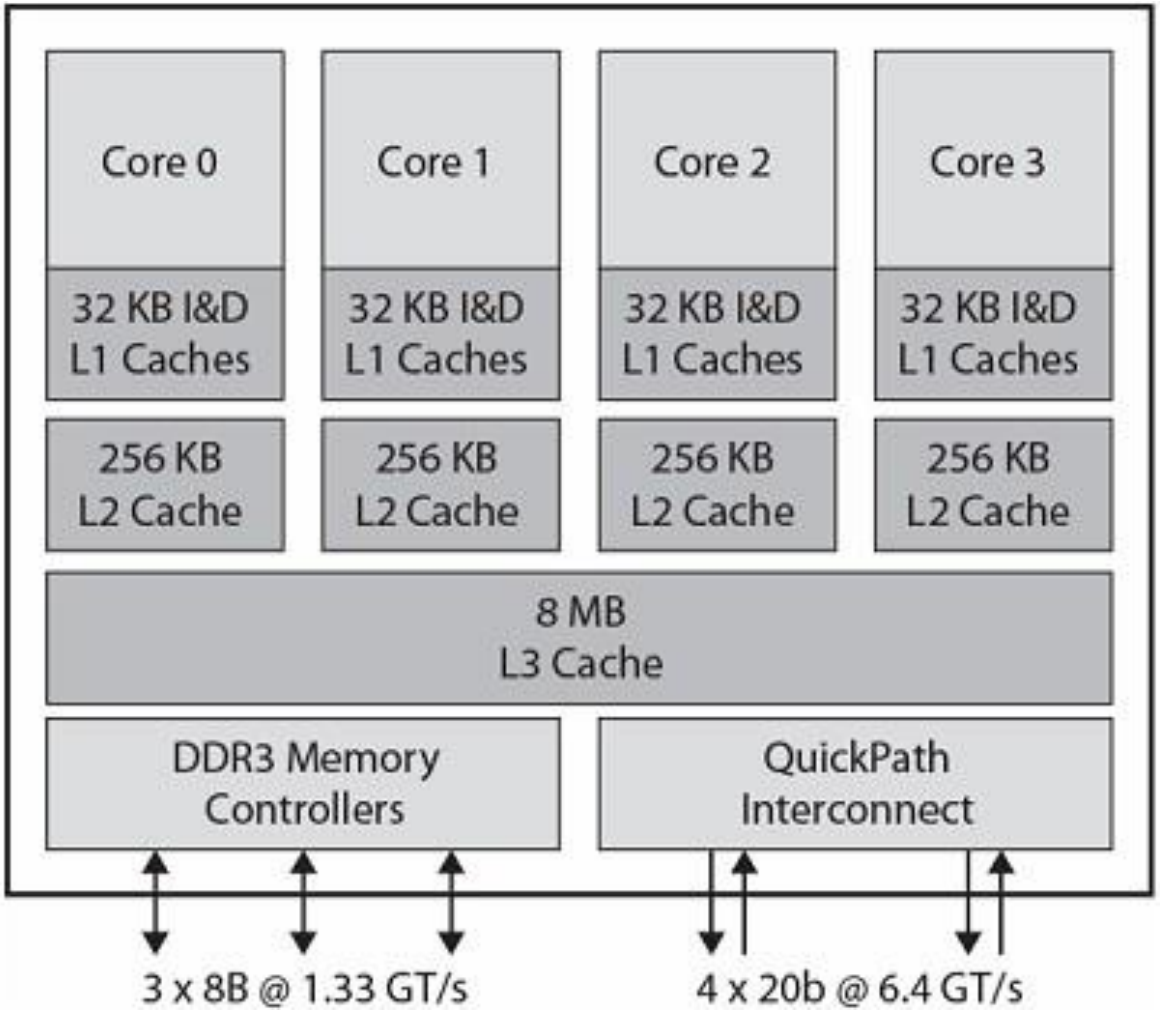




Core i7

- 11/ 2008
- Bốn bộ xử lý
- Dành riêng cache L2, chi sẻ cache L3 giữa các nhân
- Speculative pre-fetch for caches
- DDR3 memory controller đặt trên chip CPU
  - 3 kênh 8 bytes (192 bits) -> băng thông 32GB/s
  - Không có FSB
- QuickPath Interconnection (phương thức kết nối thay thế FSB)
  - Cache coherent point-to-point link
  - Tốc độ trao đổi cao giữa các chips
  - 6.4G transfers per second (6.4 GT/s), 16 bits per transfer
  - Dedicated bi-directional pairs
  - Total bandwidth 25.6GB/s

Intel Core i7



BÀI TẬP

Câu 6: Mối quan hệ giữa các thuộc tính dung lượng, giá thành và thời gian truy cập của bộ nhớ là gì?

Câu 7: Thanh ghi là gì? Nêu chức năng một số thanh ghi có bên trong CPU?

Câu 8: Hãy cho biết các thành phần chính có trong CPU và chức năng chính của chúng?

Câu 9: Viết sơ đồ thuật toán để mô tả cách thức đọc một địa chỉ từ bộ nhớ vào CPU để xử lý

Câu 10. Thanh ghi của vi xử lý là gì? Nêu chức năng và đặc điểm của thanh ghi tích luỹ A

Câu 11. Nêu sơ đồ và đặc điểm của hai dạng kiến trúc cache: Look Aside và Look Through. Trong hai dạng kiến trúc trên, dạng nào được sử dụng nhiều hơn trong thực tế hiện nay? Tại sao?

Câu 12: Trình bày khái niệm Data path khi đề cập đến tổ chức của bộ xử lý.

Câu 13: Giải thích quá trình thực thi lệnh của bộ xử lý với việc áp dụng kỹ thuật đường ống lệnh (pipelining).

Câu 14. Cơ chế xử lý xen kẽ dòng lệnh (ống lệnh – pipeline) là gì ? Nêu các đặc điểm của cơ chế ống lệnh.

Câu 15. Hãy giải thích 5 bước trong việc xử lý một lệnh của CPU: instruction fetch, instruction decode, operand fetch, instruction execute, write-back.

Anh/chị hiểu thế nào về cơ chế pipeline ? Một lệnh của CPU xử lý bằng cơ chế pipeline có nhanh hơn một lệnh CPU không xử lý bằng cơ chế pipline không? Tại sao?

Câu 16. Trình bày cấu trúc cơ bản về Ngôn ngữ Assembler

Câu 1. Trình bày chức năng và cấu trúc của một bộ vi xử lý?

Câu 2. Trình bày phương pháp tổ chức của CPU:

a) Nhiệm vụ của CPU?
b) Sơ đồ cấu trúc cơ bản của CPU?
c) Các thành phần cơ bản của CPU?
d) Đơn vị điều khiển có chức năng gì?

Câu 3. Trình bày chức năng và nhiệm vụ:

a) Các thanh ghi đoạn?
b) Các thanh ghi đa năng?
c) Các thanh ghi con trỏ và chỉ số?
d) Đơn vị số học và logic (ALU)?
e) Các thanh ghi cờ FR?
f) Đơn vị điều khiển?