

CTT009

Thao tác dữ liệu

Lê Thị Nhân
lt Nhan@fit.hcmus.edu.vn



KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Nội dung

- ☐ Kiến trúc máy tính
- ☐ Ngôn ngữ máy



Hoạt động

- ☐ Tạo nhóm
- ☐ Câu hỏi
 - ☒ Bộ nhớ chính và thanh ghi khác và giống nhau như thế nào?
- ☐ Làm poster



Poster

- Là 1 ấn phẩm khổ lớn (A0 hoặc A1) có hình ảnh và văn bản
- Được trình bày như là thông báo, quảng cáo, trang trí...
- Để ở nơi công cộng
- Với mục đích truyền đạt tới người xem những thông tin thị giác về 1 sản phẩm, sự kiện, hay vấn đề





Các thành phần trong poster

- ☐ Tiêu đề
- ☐ Mảng lớn, mảng nhỏ
- ☐ Phần chính, phần phụ
- ☐ Hình ảnh, hình nền
- ☐ Màu sắc



TITLE

Names of all coauthors
Pomona College

Abstract

About 100 words or fewer. You need to give a comprehensive overall picture. Here and everywhere, use a font large enough to read from 3-4 feet.

Method

This should be brief. Just present the basic points. Consider using a bulleted list. Make sure the reader has a good general idea of what you did (Participants, Materials, Procedure).

Figures and tables

Make sure figures can be easily interpreted and are visible from a distance. Use a graphics program.

Do not try to pack too much information into a figure or table. It should be “readable”—it should make sense visually and present information in a manner that can be picked up quickly.

Results

Brief verbal description of what you found.

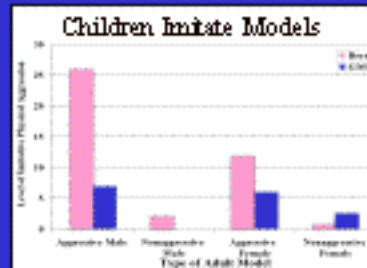
Discussion

What do your results mean? What theoretical ideas do they support or reject?

Introduction

Acquaint the reader with the problem and explain how your approach may answer some question. Refer to sources, using APA format.

Questionnaire or Stimulus Materials



Conclusions

Usually concise bulleted statements of 2 or 3 main points—the take-home message.

References

Just the main ones.

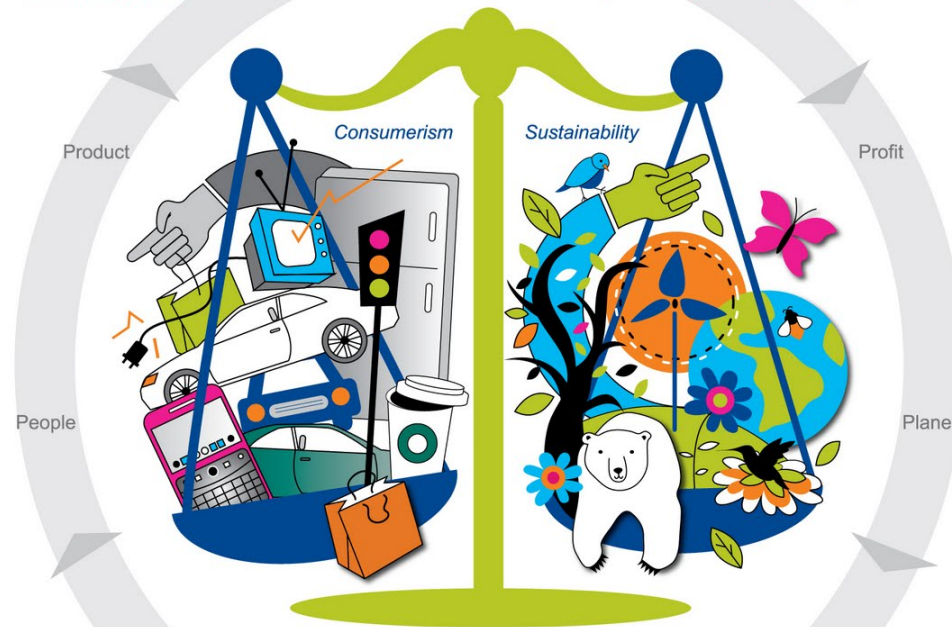
Global sourcing shouldn't cost us the Earth

1 INTRODUCTION

How can business achieve corporate sustainability in a world where global sourcing is essential for competitive advantage, long-term survival and cost management?

2 RESEARCH AIMS

- Look at how businesses gain competitive advantage by rethinking the way we make, buy and sell things.
- Test the relationship between the 4Ps of corporate sustainability: Profit = planet + people + product



4 FINDINGS

Corporate sustainability score card for industry to use as a measure of what good looks like and to ensure the correct balance of the 4Ps is achieved.

Business cannot succeed in societies that fail.

– Bjorn Stigson, president of the world business council for sustainable development (WBCSD)

3 METHODS

The study will look at major manufacturing and retailing brands, taking the electronics industry as a primary case study and pilot group. Models will be used to test the hypothesis that business will achieve competitive advantage through sustainability. Models such as the Triple Bottom Line* and 3Ps will be used to test and demonstrate that responsibility to stakeholders, and not shareholders, leads to profitability in a world where making good products shouldn't cost us the Earth.

*Triple bottom line by John Elkington. A reporting framework for sustainability.

Source :
graphicdesign.stackexchange.com

Poster vs. Mindmap

☐ Poster

- ☐ Có thể không cần trình bày hay diễn giải bằng lời... người xem chỉ cần xem là hiểu

☐ Mindmap

- ☐ Tổ chức và sắp xếp thông tin nhằm hệ thống hóa nội dung, gợi nhớ
- ☐ Cần có thêm trình bày hay diễn giải bằng lời để người nghe hiểu rõ vấn đề

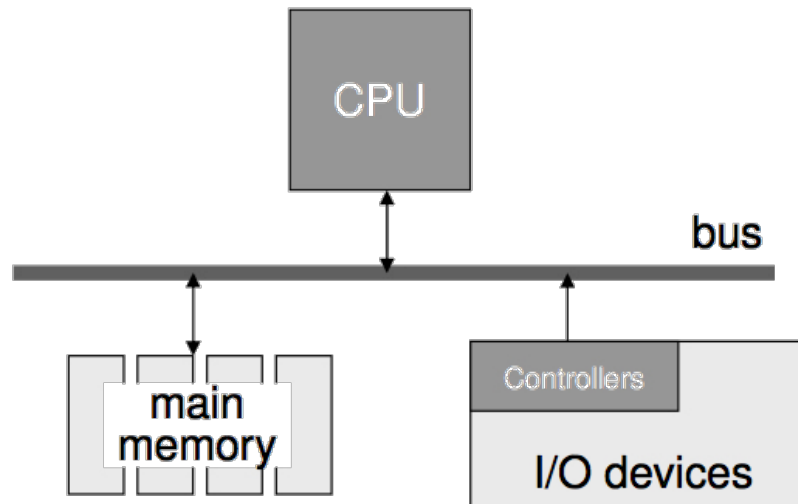


KIẾN TRÚC MÁY TÍNH



Tổng quan

- Central Processing Unit (CPU)
- Bộ nhớ chính (main memory)
- Đường truyền (bus)
- Thiết bị nhập/xuất (I/O devices)



Source: Chun-Jen Tsai, ics12, National Chiao Tung University

Tổng quan

- Central Processing Unit (CPU)
 - **Arithmetic/Logic Unit (ALU)**
 - **Control Unit**
 - **Thanh ghi (registers)**
 - Bộ nhớ đệm (cache memory)
 - Bộ nhớ chính (main memory)
 - Đường truyền (bus)
 - Thiết bị nhập/xuất (I/O devices)
- } processor core



Nhắc lại : mainboard

- Bo mạch chủ là 1 bản mạch chính chứa
 - Buses đường dẫn điện cho phép dữ liệu di chuyển giữa các thành phần với nhau
 - CPU, RAM
 - Khe cắm, bộ tản nhiệt/quạt làm mát, BIOS, các bộ chip, ổ cắm, kết nối trong và ngoài, nhiều cổng khác nhau và các dây nhúng

Nhắc lại : CPU

- ☐ Là bộ não của máy tính
- ☐ Bộ vi xử lý (processor)
- ☐ CPU thực thi **chương trình** vốn là 1 chuỗi các **câu lệnh** được lưu trữ sẵn
 - ☐ Reduced Instruction Set Computer (RISC)
 - ☐ Complex Instruction Set Computer (CISC)

Nhắc lại : CPU

- Số lượng dữ liệu mà CPU có thể xử lý tại 1 thời điểm tùy thuộc vào kích thước của đường bus
- Tốc độ của CPU được đo đạc bởi ***cycles per second***
 - ▣ Megahertz – MHz hoặc Gigahertz – GHz
- Để tăng cường hiệu năng, các CPU thường kết hợp với ***hyperthreading*** hoặc ***hypertransport***

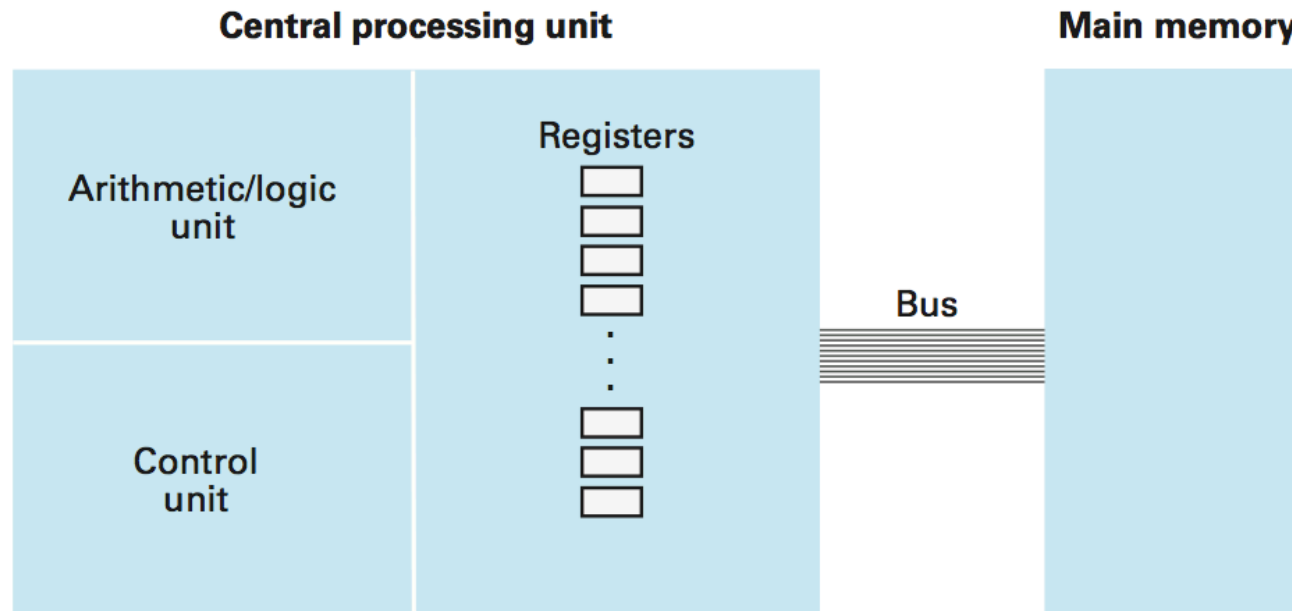


Nhắc lại : CPU

- Hiện nay các nhà sản xuất CPU tích hợp nhiều hơn 1 lõi CPU (CPU core) vào 1 con chip
 - Dual Core CPU
 - Triple Core CPU
 - Quad Core CPU
 - Hexa-Core CPU
 - Octa-Core CPU



Thành phần chính của CPU



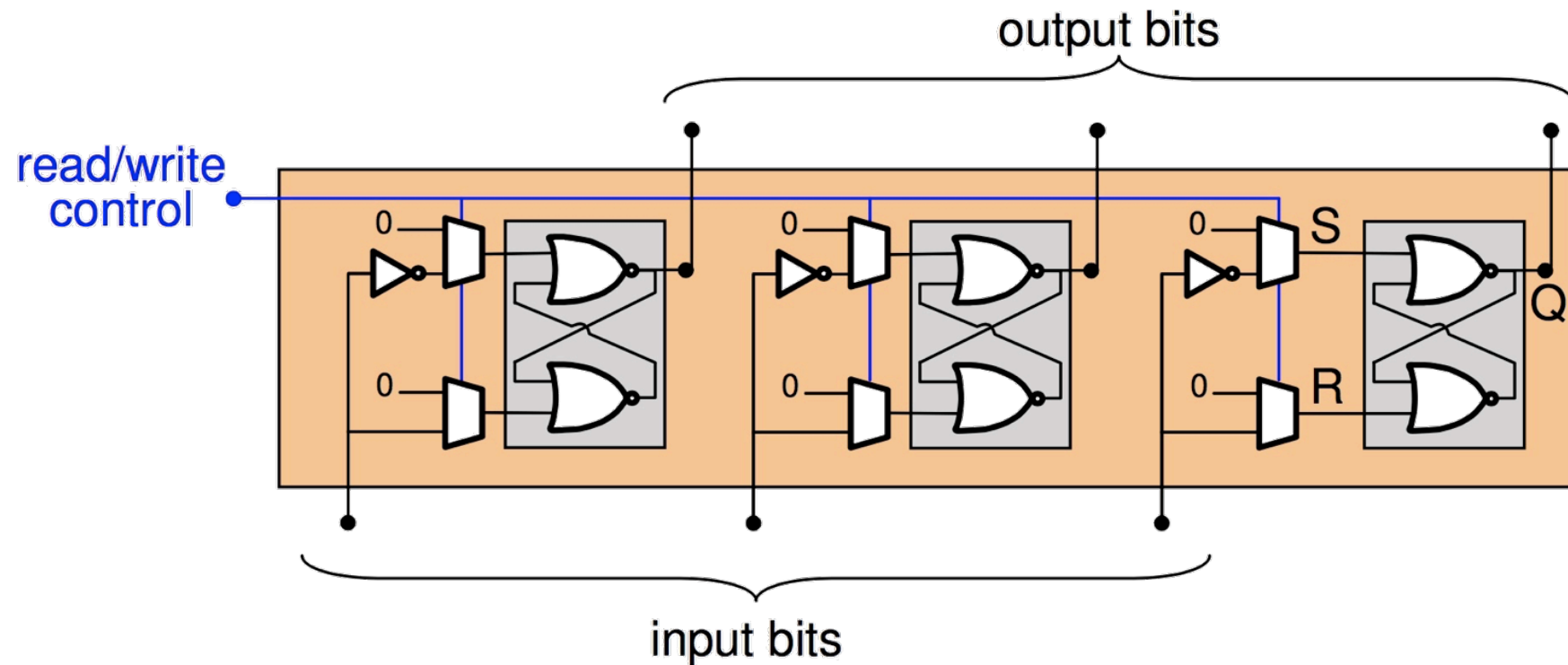
Nguồn: Computer Science - An Overview, 12e


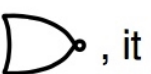

Thanh ghi

- Thanh ghi là 1 ô nhớ trong CPU có thể lưu dữ liệu *n-bit* (n : 8, 16, 32, và 64)
- Gồm 2 loại
 - Mục đích chung (general-purpose registers)
 - Mục đích đặc biệt (special-purpose registers)
- Thanh ghi dùng để lưu trữ kết quả tính toán trung gian



Ví dụ : thanh ghi 3-bit



Note:  is denoted as , it is called a NOR gate.  is a 2-to-1 multiplexor.

Source: Chun-Jen Tsai, ics12, National Chiao Tung University

Arithmetic/logic unit

- Bộ số học và logic
 - Là 1 bộ phận trong CPU thực hiện các phép tính số học và logic cơ bản trên dữ liệu
 - Cộng, trừ, so sánh...
 - AND, OR, NOT, XOR...
 - Sau đó trả lại kết quả cho thanh ghi hoặc bộ nhớ
- Có các thanh ghi 16, 32 hoặc 64-bit

Control unit

□ Bộ điều khiển

- Có nhiệm vụ thông dịch các lệnh của chương trình và điều khiển hoạt động xử lý
- Được điều tiết chính xác bởi xung nhịp đồng hồ hệ thống
- Cấu tạo từ
 - Mạch logic so sánh
 - Linh kiện bán dẫn (transistor...)



NGÔN NGỮ MÁY

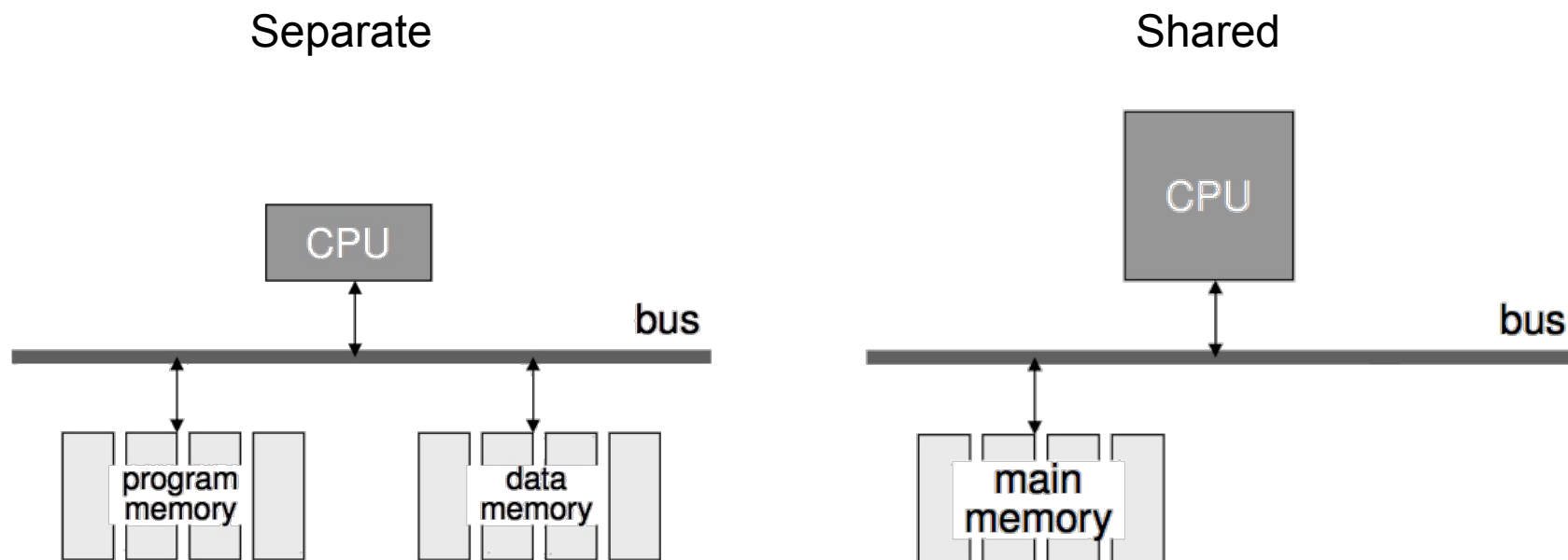


Khái niệm

- Chương trình lưu trữ (stored program)
 - Có thể được mã hóa dưới dạng chuỗi bits và được ***lưu trong bộ nhớ***
 - Là ***chuỗi các câu lệnh*** thực hiện 1 thuật toán
 - Từ đó, CPU có thể trích ra các câu lệnh và đem đi thực thi
 - Là một dạng đặc biệt của dữ liệu

Khái niệm

□ Bộ nhớ chương trình và bộ nhớ dữ liệu



Source: Chun-Jen Tsai, ics12, National Chiao Tung University

Câu lệnh là gì?

- Khi yêu cầu người khác thực hiện điều gì đó, ta dùng *ngôn ngữ của con người*
 - Rườm rà và mơ hồ đối với máy tính để có thể giải mã (decode) và thực thi (execute)
- ***Ngôn ngữ máy***
 - Ngắn gọn (concise) : dễ dàng giải mã
 - Rõ ràng (precise) : 1 câu lệnh chỉ được thực hiện bởi 1 cách

Thuật ngữ liên quan

- Mã lệnh (machine instruction)
 - Là lệnh (command hay instruction) được mã hóa dưới dạng chuỗi bits có thể nhận biết bởi CPU
- Ngôn ngữ máy (machine language)
 - Là một tập lệnh được nhận biết bởi 1 máy



Ngôn ngữ máy

☐ Reduced Instruction Set Computing (RISC)

- ☐ Ít lệnh, đơn giản, hiệu quả và nhanh

- ☐ Ví dụ:

- PowerPC của Apple/IBM/Motorola
 - ARM

☐ Complex Instruction Set Computing (CISC)

- ☐ Nhiều lệnh, thuận lợi, mạnh mẽ

- ☐ Ví dụ: Intel

Loại mã lệnh

☐ Data transfer

- ☐ Sao chép dữ liệu từ 1 địa điểm tới 1 địa điểm khác

☐ Arithmetic/Logic

- ☐ Sử dụng các chuỗi bits có sẵn để tính toán ra chuỗi bits mới

☐ Control

- ☐ Chỉ đạo thực hiện chương trình

Ví dụ : cộng

- Step 1.** Get one of the values to be added from memory and place it in a register.
- Step 2.** Get the other value to be added from memory and place it in another register.
- Step 3.** Activate the addition circuitry with the registers used in Steps 1 and 2 as inputs and another register designated to hold the result.
- Step 4.** Store the result in memory.
- Step 5.** Stop.

Ví dụ : chia

Step 1. LOAD a register with a value from memory.

Step 2. LOAD another register with another value from memory.

Step 3. If this second value is zero, JUMP to Step 6.

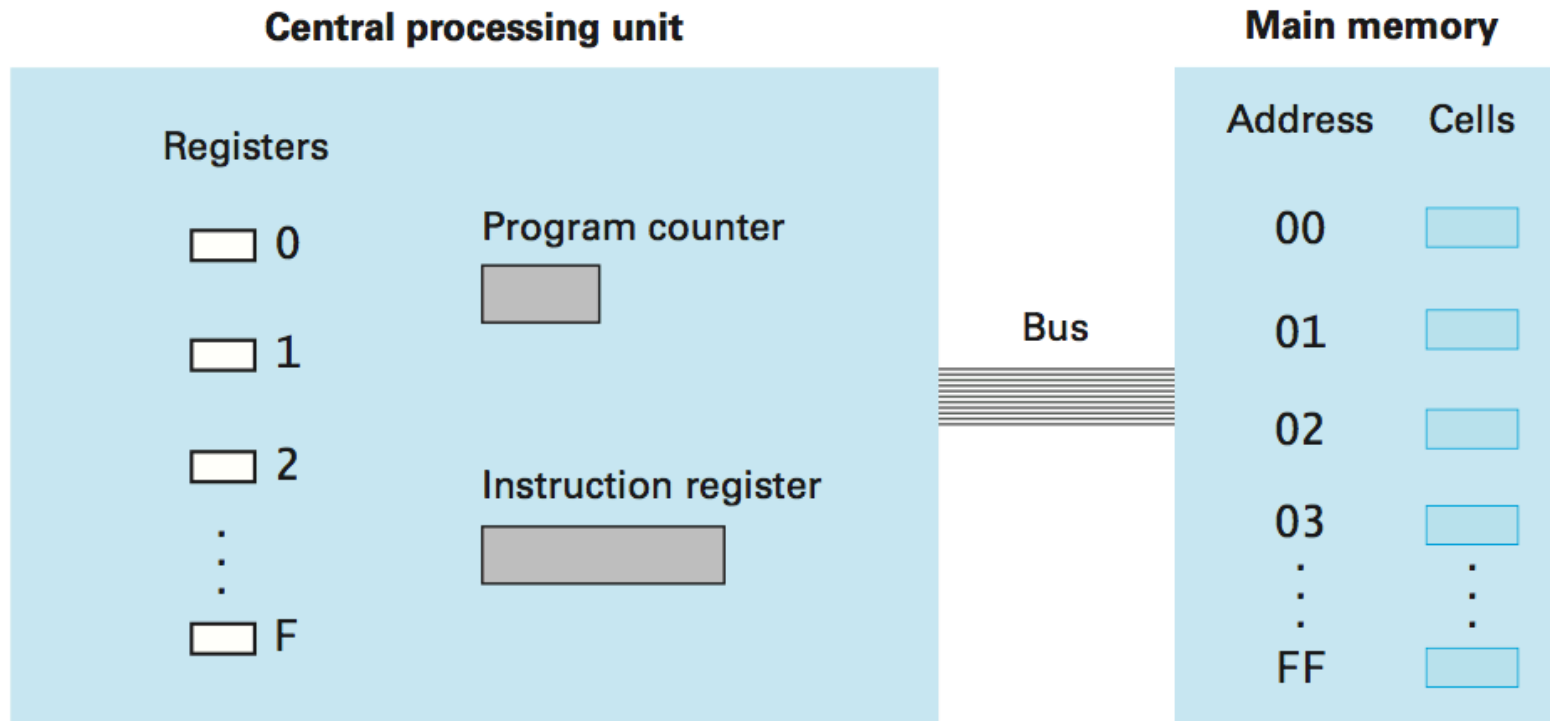
Step 4. Divide the contents of the first register by the second register and leave the result in a third register.

Step 5. STORE the contents of the third register in memory.

Step 6. STOP.

Nguồn: Computer Science - An Overview, 12e

Ví dụ : kiến trúc máy



Thành phần của mã lệnh

□ Chuỗi bits của mã lệnh gồm 2 phần

□ Op-code field

- Chỉ định thao tác nào được thực thi

□ Operand field

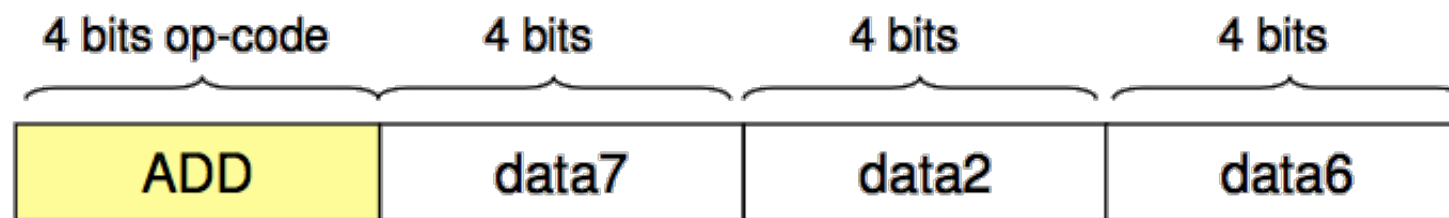
- Cung cấp thêm thông tin chi tiết về thao tác

- Dữ liệu và địa chỉ

- Số lượng operands tùy thuộc vào op-code

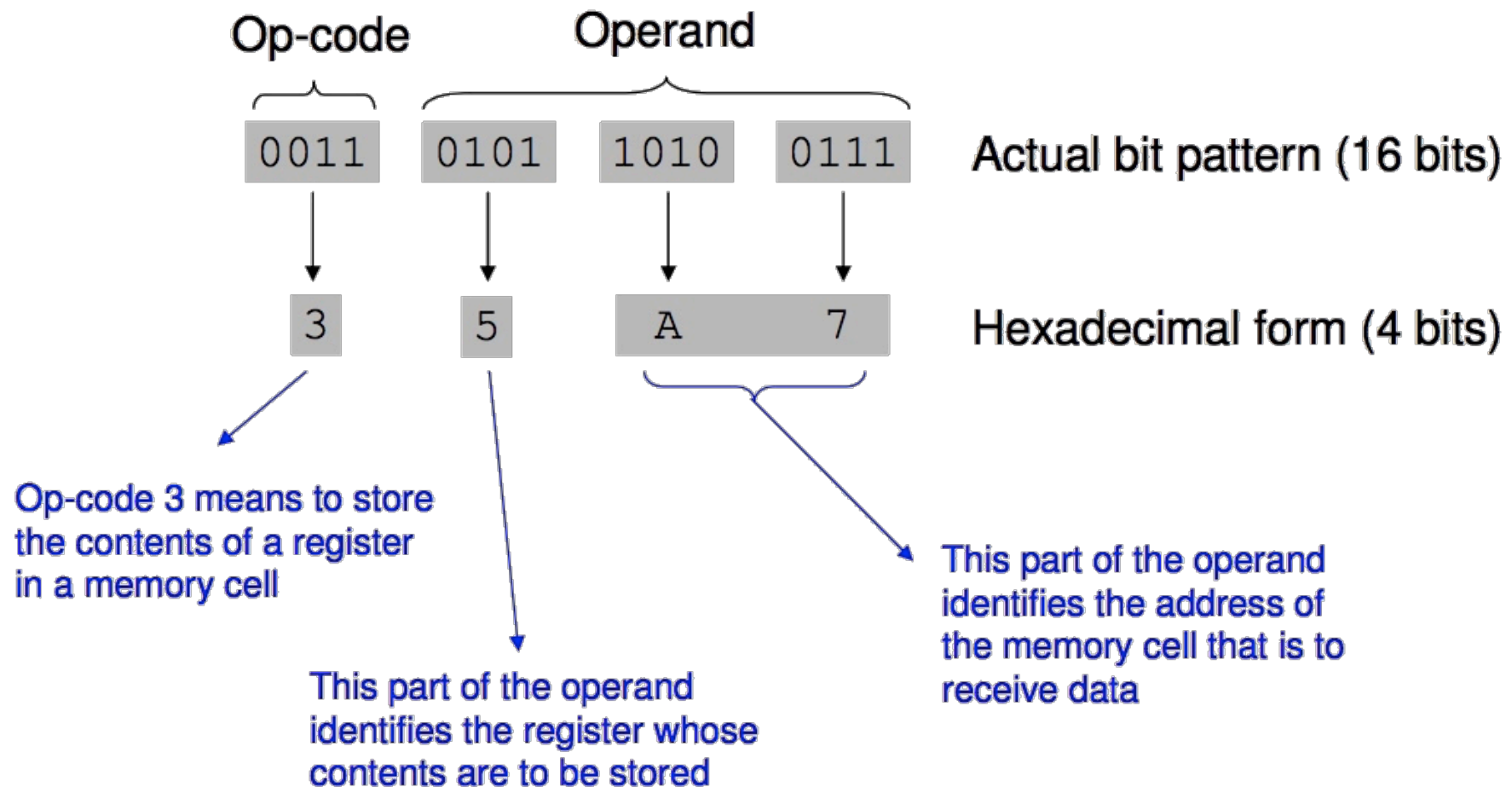
Ví dụ

- Câu lệnh yêu cầu CPU
 - Cộng hai ô nhớ data2 và data6
 - Sau đó lưu kết quả vào ô nhớ data7
- Yêu cầu này được mã hóa bởi 16 bits



Source: Chun-Jen Tsai, ics12, National Chiao Tung University

Ví dụ cụ thể



Source: Chun-Jen Tsai, ics12, National Chiao Tung University

Ví dụ cộng ở mã lệnh

Encoded instructions	Translation
156C	Load register 5 with the bit pattern found in the memory cell at address 6C.
166D	Load register 6 with the bit pattern found in the memory cell at address 6D.
5056	Add the contents of register 5 and 6 as though they were two's complement representation and leave the result in register 0.
306E	Store the contents of register 0 in the memory cell at address 6E.
C000	Halt.

Nguồn: Computer Science - An Overview, 12e

Tập mã lệnh đơn giản

Op-code	Operand	Description
1	RXY	LOAD reg. R from cell XY
2	RXY	LOAD reg. R with XY
3	RXY	STORE reg. R at XY
4	0RS	MOVE R to S
5	RST	ADD S and T into R (2's comp.)
6	RST	ADD S and T into R (floating pt.)
7	RST	OR S and T into R
8	RST	AND S and T into R
9	RST	XOR S and T into R
A	R0X	ROTATE reg. R X times
B	RXY	JUMP to XY if R = reg. 0
C	0	HALT

Ví dụ

☐ 14A3

☐ 20A3

☐ 35B1

☐ 40A4

☐ 5726

☐ 634E

☐ 7CB4

☐ 8045

☐ 95F3

☐ A403

☐ B43C

☐ C000



TÓM TẮT



Bài giảng hôm nay

- ☐ CPU

- ☐ Thanh ghi

- ☐ Mã lệnh (mã máy)

- ☐ Op-code

- ☐ Operand



Bài giảng lần tới

- Thao tác dữ liệu (chapter 2)
 - Thực thi chương trình
 - Lệnh arithmetic/logic
 - Giao tiếp với thiết bị khác
 - Các kiến trúc máy tính khác

