




KIẾN TRÚC MÁY TÍNH

Computer Architecture


Nguyễn Kim Khánh
 Bộ môn Kỹ thuật máy tính
 Viện Công nghệ thông tin và Truyền thông
 Trường Đại học Bách Khoa Hà Nội
 Department of Computer Engineering (DCE)
 School of Information and Communication Technology (SoICT)
 Hanoi University of Science and Technology (HUST)



Contact Information

- Address:
 - DCE's Office: 502-B1
 - SoICT's Office: 503-B1
 - CS Lab: 505-B1
- Mobile: 091-358-5533
- e-mail: khanhmk@soict.hut.edu.vn
khanh.nguyenkim@hust.edu.vn

14 January 2013 Kiến trúc máy tính 2



Mục tiêu học phần

- Sinh viên được trang bị các kiến thức cơ sở về kiến trúc tập lệnh và tổ chức của máy tính, cũng như những vấn đề cơ bản trong thiết kế máy tính.
- Sau khi học xong học phần này, sinh viên có khả năng:
 - Tìm hiểu kiến trúc tập lệnh của các bộ xử lý cụ thể
 - Lập trình hợp ngữ trên một số kiến trúc
 - Đánh giá hiệu năng của các họ máy tính
 - Khai thác và quản trị hiệu quả các hệ thống máy tính
 - Phân tích và thiết kế máy tính

14 January 2013 Kiến trúc máy tính 3



Tài liệu tham khảo chính






- [1] William Stallings - *Computer Organization and Architecture – Designing for Performance – 2009 (8th edition)*
- [2] David A. Patterson & John L. Hennessy - *Computer Organization and Design: The Hardware/Software Interface – 2009 (4th edition)*
- [3] David Money Harris and Sarah L. Harris, *Digital Design and Computer Architecture, 2007*
- [4] Behrooz Parhami - *Computer Architecture: From Microprocessors to Supercomputers - 2005*

14 January 2013 Kiến trúc máy tính 4

NKK-HUST

Nội dung học phần

- Chương 1. Giới thiệu chung
- Chương 2. Cơ bản về logic số
- Chương 3. Hệ thống máy tính
- Chương 4. Số học máy tính
- Chương 5. Kiến trúc tập lệnh
- Chương 6. Cấu trúc bộ xử lý
- Chương 7. Bộ nhớ
- Chương 8. Vào-ra
- Chương 9. Các kiến trúc song song

14 January 2013 Kiến trúc máy tính 5

NKK-HUST

Chú ý: Bài giảng mới nhất Jan 2013

<ftp://dce.hut.edu.vn/khanhmk/CA>

NKK-HUST

Kiến trúc máy tính

Chương 1
GIỚI THIỆU CHUNG

Nguyễn Kim Khánh
Trường Đại học Bách khoa Hà Nội

14 January 2013 Kiến trúc máy tính 7

NKK-HUST

Nội dung

- 1.1. Máy tính và phân loại
- 1.2. Kiến trúc máy tính
- 1.3. Lịch sử phát triển của máy tính
- 1.4. Hiệu năng máy tính

14 January 2013 Kiến trúc máy tính 8

NKK-HUST

1.1. Máy tính và phân loại máy tính

1. Máy tính

- **Máy tính (Computer)** là thiết bị điện tử thực hiện các công việc sau:
 - Nhận thông tin vào,
 - Xử lý thông tin theo dãy các lệnh được nhớ sẵn bên trong,
 - Đưa thông tin ra.
- Dãy các lệnh nằm trong bộ nhớ để yêu cầu máy tính thực hiện công việc cụ thể gọi là **chương trình (program)**
 → Máy tính hoạt động theo chương trình.

14 January 2013 Kiến trúc máy tính 9

NKK-HUST

Máy tính

```

graph LR
    A[Các thiết bị vào  
(Input Devices)] --> B
    subgraph B [ ]
        direction TB
        B1[Bộ xử lý trung tâm  
(Central Processing Unit)]
        B2[Bộ nhớ chính  
(Main Memory)]
        B1 <--> B2
    end
    B --> C[Các thiết bị ra  
(Output Devices)]
    
```

14 January 2013 Kiến trúc máy tính 10

NKK-HUST

2. Phân loại máy tính

- Phân loại truyền thống:
 - Máy vi tính (Microcomputers)
 - Máy tính nhỏ (Minicomputers)
 - Máy tính lớn (Mainframe Computers)
 - Siêu máy tính (Supercomputers)

14 January 2013 Kiến trúc máy tính 11

NKK-HUST

Phân loại máy tính hiện đại [H&P]

- Thiết bị di động cá nhân (Personal Mobile Device): Smartphone, Tablet,...
- Máy tính để bàn (Desktop Computers)
- Máy chủ (Server Computers)
 - Dùng trong mạng theo mô hình Client/Server
- Máy tính cụm/nhà kho (Clusters/Warehouse Scale Computers)
- Máy tính nhúng (Embedded Computers)
 - Đặt ẩn trong thiết bị khác
 - Thiết kế chuyên dụng

14 January 2013 Kiến trúc máy tính 12

NKK-HUST

1.2. Định nghĩa kiến trúc máy tính

- Định nghĩa trước đây về kiến trúc máy tính:
 - Cách nhìn logic của máy tính từ người lập trình (hardware/software interface)
 - Kiến trúc tập lệnh (Instruction Set Architecture – ISA)
- Là định nghĩa hẹp

14 January 2013 Kiến trúc máy tính 13

NKK-HUST

Định nghĩa của Hennessy/ Patterson

- Kiến trúc máy tính bao gồm:
 - Kiến trúc tập lệnh (Instruction Set Architecture): nghiên cứu máy tính theo cách nhìn của người lập trình (hardware/software interface).
 - Tổ chức máy tính (Computer Organization): nghiên cứu thiết kế máy tính ở mức cao, chẳng hạn như hệ thống nhớ, cấu trúc bus, thiết kế bên trong CPU.
 - Phần cứng (Hardware): nghiên cứu thiết kế logic chi tiết và công nghệ đóng gói của máy tính.
- Kiến trúc tập lệnh thay đổi chậm, tổ chức và phần cứng máy tính thay đổi rất nhanh.

14 January 2013 Kiến trúc máy tính 14

NKK-HUST

1. Kiến trúc tập lệnh

Kiến trúc tập lệnh của máy tính bao gồm:

- Tập lệnh: tập hợp các chuỗi số nhị phân mã hoá cho các thao tác mà máy tính có thể thực hiện
- Các kiểu dữ liệu: các kiểu dữ liệu mà máy tính có thể xử lý

14 January 2013 Kiến trúc máy tính 15

NKK-HUST

2. Cấu trúc cơ bản của máy tính

```

graph TD
    CPU[CPU] <--> Bus[Bus liên kết hệ thống]
    Mem[Bộ nhớ chính] <--> Bus
    Bus <--> IO[Hệ thống vào-ra]
    
```

14 January 2013 Kiến trúc máy tính 16

NKK-HUST

Các thành phần cơ bản của máy tính

- **Bộ xử lý trung tâm** (Central Processing Unit): Điều khiển hoạt động của máy tính và xử lý dữ liệu.
- **Bộ nhớ chính** (Main Memory): Chứa các chương trình và dữ liệu đang được sử dụng.
- **Vào ra** (Input/Output): Trao đổi thông tin giữa máy tính với bên ngoài.
- **Bus liên kết hệ thống** (System Interconnection Bus): Kết nối và vận chuyển thông tin giữa các thành phần với nhau.

14 January 2013

Kiến trúc máy tính

17

NKK-HUST

Mô hình phân lớp của máy tính

The diagram illustrates the layering of a computer system. On the left, a vertical stack of boxes represents the layers: **Phản ứng** (Application), **Phản ứng trung gian** (Intermediate), **Hệ điều hành** (OS), **Kiến trúc tập lệnh** (Instruction set architecture), **Vi kiến trúc** (Microarchitecture), **Logic-số** (Logic), and **Mạch điện tử** (Electronic circuitry). On the right, a flowchart shows the interaction between various components: **Người sử dụng** (User) interacts with **Các phần mềm ứng dụng** (Application software), which interacts with **Các phần mềm trung gian** (Intermediate software), which interacts with **Hệ điều hành** (OS), which interacts with **Phản ứng** (Application). Additionally, **Người lập trình** (Programmer) interacts with **Các phần mềm ứng dụng** and **Các phần mềm trung gian**. **Người thiết kế HĐH** (OS designer) interacts with **Hệ điều hành** and **Phản ứng**.

- **Phần cứng** (Hardware): hệ thống vật lý của máy tính.
- **Phần mềm** (Software): các chương trình và dữ liệu.

14 January 2013

Kiến trúc máy tính

18

NKK-HUST

1.3. Lịch sử phát triển của của máy tính

Các thế hệ máy tính

- Thế hệ thứ nhất: Máy tính dùng **đèn điện tử** chân không (1950s)
- Thế hệ thứ hai: Máy tính dùng **transistor** (1960s)
- Thế hệ thứ ba: Máy tính dùng **vi mạch SSI, MSI và LSI** (1970s)
- Thế hệ thứ tư: Máy tính dùng **vi mạch VLSI** (1980s)
- Thế hệ thứ năm: Máy tính dùng **vi mạch ULSI, SoC** (1990s)

14 January 2013

Kiến trúc máy tính

19

NKK-HUST

Máy tính dùng đèn điện tử

- **ENIAC**- Máy tính điện tử đầu tiên
 - Electronic Numerical Interpolator And Computer
 - Dự án của Bộ Quốc phòng Mỹ
 - Do John Mauchly và John Presper Eckert ở Đại học Pennsylvania thiết kế.
 - Bắt đầu từ 1943, hoàn thành 1946
 - Nặng 30 tấn
 - 18000 đèn điện tử và 1500 rơle
 - 5000 phép cộng/giây
 - Xử lý theo số thập phân
 - Bộ nhớ chỉ lưu trữ dữ liệu
 - Lập trình bằng cách thiết lập vị trí của các chuyển mạch và các cáp nối.

14 January 2013

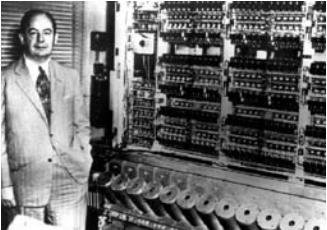
Kiến trúc máy tính

20

NKK-HUST

Máy tính von Neumann

- Đó là máy tính IAS: Princeton Institute for Advanced Studies
- Được bắt đầu từ 1947, hoàn thành 1952
- Do John von Neumann thiết kế
- Được xây dựng theo ý tưởng “chương trình được lưu trữ” (*stored-program concept*) của von Neumann/Turing (1945)



14 January 2013 Kiến trúc máy tính 21

NKK-HUST

Đặc điểm chính của máy tính IAS

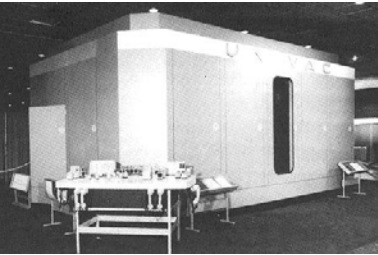

- Bao gồm các thành phần: đơn vị điều khiển, đơn vị số học và logic (ALU), bộ nhớ chính và các thiết bị vào-ra.
- Bộ nhớ chính chứa chương trình và dữ liệu
- Bộ nhớ chính được đánh địa chỉ theo từng ngăn nhớ, không phụ thuộc vào nội dung của nó.
- ALU thực hiện các phép toán với số nhị phân
- Đơn vị điều khiển nhận lệnh từ bộ nhớ, giải mã và thực hiện lệnh một cách tuần tự.
- Đơn vị điều khiển điều khiển hoạt động của các thiết bị vào-ra
- Trở thành mô hình cơ bản của máy tính

14 January 2013 Kiến trúc máy tính 22

NKK-HUST

Các máy tính thương mại

- 1947 - Eckert-Mauchly Computer Corporation
- UNIVAC I (Universal Automatic Computer)
- 1950s - UNIVAC II
 - Nhanh hơn
 - Bộ nhớ lớn hơn





14 January 2013 Kiến trúc máy tính 23

NKK-HUST

Hãng IBM

- IBM - International Business Machine
- 1953 - IBM 701
 - Máy tính lưu trữ chương trình đầu tiên của IBM
 - Sử dụng cho tính toán khoa học
- 1955 - IBM 702
 - Các ứng dụng thương mại




14 January 2013 Kiến trúc máy tính 24

NKK-HUST

Máy tính dùng transistor

- Máy tính PDP-1 của DEC (Digital Equipment Corporation): *máy tính mini* đầu tiên
- IBM 7000
- Hàng trăm nghìn phép cộng trong một giây.
- Các *ngôn ngữ lập trình bậc cao* ra đời.



14 January 2013 Kiến trúc máy tính 25

NKK-HUST

Máy tính dùng vi mạch SSI, MSI và LSI

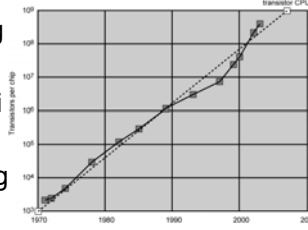
- Vi mạch (Integrated Circuit - IC): nhiều transistor và các phần tử khác được tích hợp trên một chip bán dẫn.
 - SSI (Small Scale Integration)
 - MSI (Medium Scale Integration)
 - LSI (Large Scale Integration)
 - VLSI (Very Large Scale Integration) (thế hệ thứ tư)
 - ULSI (Ultra Large Scale Integration) (thế hệ thứ năm)
 - SoC (System on Chip)
- Siêu máy tính xuất hiện: CRAY-1, VAX
- Bộ vi xử lý (microprocessor) ra đời
 - Bộ vi xử lý đầu tiên → Intel 4004 (1971).

14 January 2013 Kiến trúc máy tính 26

NKK-HUST

Luật Moore


- Gordon Moore – người đồng sáng lập Intel
- Số transistors trên chip sẽ gấp đôi sau 18 tháng
- Giá thành của chip hầu như không thay đổi
- Mật độ cao hơn, do vậy đường dẫn ngắn hơn
- Kích thước nhỏ hơn dẫn tới độ phức tạp tăng lên
- Điện năng tiêu thụ ít hơn
- Hệ thống có ít các chip liên kết với nhau, do đó tăng độ tin cậy



14 January 2013 Kiến trúc máy tính 27

NKK-HUST

IBM 360 Family và PDP-11 (1973)



14 January 2013 Kiến trúc máy tính

NKK-HUST

VAX-11, Micro VAX, CRAY-1



1

Kiến trúc máy tính

29

NKK-HUST

Máy tính dùng vi mạch VLSI/ULSI

Các sản phẩm chính của công nghệ VLSI/ULSI:

- **Bộ vi xử lý (Microprocessor):** CPU được chế tạo trên một chip.
- **Vi mạch điều khiển tổng hợp (Chipset):** một hoặc một vài vi mạch thực hiện được nhiều chức năng điều khiển và nối ghép.
- **Bộ nhớ bán dẫn (Semiconductor Memory):** ROM, RAM, Flash
- **Các bộ vi điều khiển (Microcontroller):** máy tính chuyên dụng được chế tạo trên 1 chip.

14 January 2013

Kiến trúc máy tính

30

NKK-HUST

Sự phát triển của bộ vi xử lý

- 1971: bộ vi xử lý 4-bit Intel 4004
- 1972-1977: các bộ xử lý 8-bit
- 1978-1984: các bộ xử lý 16-bit
- Khoảng từ 1985: các bộ xử lý 32-bit
- Khoảng từ 2000: các bộ xử lý 64-bit
- Từ 2006: các bộ xử lý đa lõi (multicores)

14 January 2013

Kiến trúc máy tính

31

NKK-HUST

Máy tính ngày nay



14 January 2013

Kiến trúc máy tính

32

NKK-HUST

1.4. Hiệu năng máy tính

- Định nghĩa hiệu năng P(Performance):

$$P = 1/t$$
 trong đó: t là thời gian thực hiện
- “Máy tính A nhanh hơn máy B n lần”

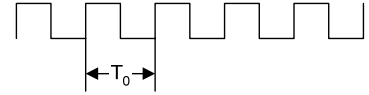
$$P_A / P_B = t_B / t_A = n$$
- Ví dụ: Thời gian chạy chương trình:
 - 10s trên máy A, 15s trên máy B
 - $t_B / t_A = 15s / 10s = 1.5$
 - Vậy máy A nhanh hơn máy B 1.5 lần

14 January 2013 Kiến trúc máy tính 33

NKK-HUST

Xung nhịp của CPU

- Hoạt động của CPU được điều khiển bởi xung nhịp có tần số xác định



- Chu kỳ xung nhịp T_0 (Clock period): thời gian của một chu kỳ
- Tần số xung nhịp f_0 (Clock rate): số chu kỳ trong 1 giây.
 - $f_0 = 1/T_0$
 - VD: Bộ xử lý có $f_0 = 4GHz = 4000MHz = 4 \times 10^9 Hz$
 $T_0 = 1/(4 \times 10^9) = 0.25 \times 10^{-9}s = 0.25ns$

14 January 2013 Kiến trúc máy tính 34

NKK-HUST

Thời gian CPU (t_{CPU})

$$t_{CPU} = n \times T_0 = \frac{n}{f_0}$$

- trong đó: n là số chu kỳ xung nhịp
- Hiệu năng được tăng lên bằng cách:
 - Giảm số chu kỳ xung nhịp n
 - Tăng tần số xung nhịp f_0

14 January 2013 Kiến trúc máy tính 35

NKK-HUST

Ví dụ

- Máy tính A:
 - Tần số xung nhịp: $f_A = 2GHz$
 - Thời gian của CPU: $t_A = 10s$
- Máy tính B
 - Thời gian của CPU: $t_B = 6s$
 - Số chu kỳ xung nhịp của B = 1.2 x Số chu kỳ xung nhịp của A
- Xác định tần số xung nhịp của máy B (f_B)?
- Giải:


$$f_B = \frac{n_B}{t_B} = \frac{1.2 \times n_A}{6s}$$

$$n_A = t_A \times f_A = 10s \times 2GHz = 20 \times 10^9$$

$$f_B = \frac{1.2 \times 20 \times 10^9}{6s} = \frac{24 \times 10^9}{6s} = 4GHz$$

14 January 2013 Kiến trúc máy tính 36

NKK-HUST



Số lệnh và số chu kỳ trên một lệnh

- Số chu kỳ = Số lệnh x Số chu kỳ trên một lệnh
$$n = IC \times CPI$$


n - số chu kỳ, IC - số lệnh (Instruction Count), CPI - số chu kỳ trên một lệnh (Cycles per Instruction)
- Thời gian thực hiện của CPU:
$$t_{CPU} = IC \times CPI \times T_0 = \frac{IC \times CPI}{f_0}$$
- Trong trường hợp các lệnh khác nhau có CPI khác nhau, cần tính CPI trung bình

14 January 2013

Kiến trúc máy tính

37

NKK-HUST



Ví dụ

- Máy tính A: $T_A = 250\text{ps}$, $CPI_A = 2.0$
- Máy tính B: $T_B = 500\text{ps}$, $CPI_B = 1.2$
- Cùng kiến trúc tập lệnh (ISA)
- Máy nào nhanh hơn và nhanh hơn bao nhiêu ?

$$t_A = IC \times CPI_A \times T_A = IC \times 2.0 \times 250\text{ps} = IC \times 500\text{ps}$$
$$t_B = IC \times CPI_B \times T_B = IC \times 1.2 \times 500\text{ps} = IC \times 600\text{ps}$$
$$\frac{t_B}{t_A} = \frac{IC \times 600\text{ps}}{IC \times 500\text{ps}} = 1.2$$


Vậy:
A nhanh hơn B 1.2 lần

14 January 2013

Kiến trúc máy tính

38

NKK-HUST



Chi tiết hơn về CPI


- Nếu loại lệnh khác nhau có số chu kỳ khác nhau, ta có tổng số chu kỳ:
$$n = \sum_{i=1}^K (CPI_i \times IC_i)$$
- CPI trung bình:
$$CPI_{TB} = \frac{n}{IC} = \sum_{i=1}^K \left(CPI_i \times \frac{IC_i}{IC} \right)$$

14 January 2013

Kiến trúc máy tính

39

NKK-HUST



Ví dụ

- Cho bảng chỉ ra các dãy lệnh sử dụng các lệnh thuộc các loại A, B, C. Tính CPI trung bình?

Loại lệnh	A	B	C
CPI theo loại lệnh	1	2	3
IC trong dãy lệnh 1	2	1	2
IC trong dãy lệnh 2	4	1	1

- Dãy lệnh 1: IC = 5
 - Số chu kỳ = $2 \times 1 + 1 \times 2 + 2 \times 3 = 10$
 - $CPI_{TB} = 10/5 = 2.0$
- Dãy lệnh 2: IC = 6
 - Số chu kỳ = $4 \times 1 + 1 \times 2 + 1 \times 3 = 9$
 - $CPI_{TB} = 9/6 = 1.5$

14 January 2013

Kiến trúc máy tính

40

NKK-HUST

Tóm tắt về Hiệu năng

CPU Time = $\frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}}$

$$t_{CPU} = IC \times CPI \times T_0 = \frac{IC \times CPI}{f_0}$$

■ Hiệu năng phụ thuộc vào:

■ Thuật toán: ảnh hưởng tới IC

■ Ngôn ngữ lập trình: ảnh hưởng tới IC, CPI

■ Chương trình dịch: ảnh hưởng tới IC, CPI

■ Kiến trúc tập lệnh: ảnh hưởng tới IC, CPI

14 January 2013

Kiến trúc máy tính

41

NKK-HUST

MIPS như là thước đo hiệu năng

■ MIPS: Millions of Instructions Per Second
(Số triệu lệnh trên 1 giây)

$$\text{MIPS} = \frac{\text{Instruction count}}{\text{Execution time} \times 10^6} = \frac{\text{Instruction count}}{\frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}} \times 10^6} = \frac{\text{Clock rate}}{\text{CPI} \times 10^6}$$

$$\text{MIPS} = \frac{f_0}{\text{CPI} \times 10^6} \qquad \text{CPI} = \frac{f_0}{\text{MIPS} \times 10^6}$$

14 January 2013

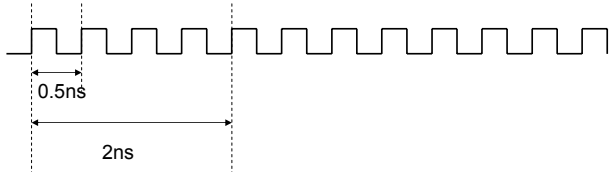
Kiến trúc máy tính

42

NKK-HUST

Ví dụ

Tính MIPS của bộ xử lý với:
clock rate = 2GHz và CPI = 4



1 chu kỳ = $1/(2 \times 10^9) = 0,5\text{ns}$

CPI = 4 → 1 lệnh = $4 \times 0,5\text{ns} = 2\text{ns}$

Vậy bộ xử lý thực hiện được 500MIPS

14 January 2013

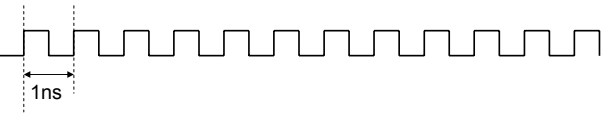
Kiến trúc máy tính

43

NKK-HUST

Ví dụ

Tính CPI của bộ xử lý với:
clock rate = 1GHz và 400 MIPS?



4×10^8 lệnh thực hiện trong 1s

→ 1 lệnh thực hiện trong $1/(4 \times 10^8)\text{s} = 2,5\text{ns}$

→ CPI = 2,5

14 January 2013

Kiến trúc máy tính

44

Nguyễn Kim Khánh - ĐHBKHN

NKK-HUST

MFLOPS

Millions of Floating Point Operations per Second
(Số triệu phép toán số dấu phẩy động trên một giây)

$$\text{MFLOPS} = \frac{\text{Executed floating point operations}}{\text{Execution time} \times 10^6}$$

GFLOPS (10^9)

TFLOPS (10^{12})

14 January 2013 Kiến trúc máy tính 45

NKK-HUST

Hết chương 1

14 January 2013 Kiến trúc máy tính 46

NKK-HUST

Kiến trúc máy tính

Chương 2

CƠ BẢN VỀ LOGIC SỐ

Nguyễn Kim Khánh
Trường Đại học Bách khoa Hà Nội

14 January 2013 Kiến trúc máy tính 47

NKK-HUST

Nội dung học phần

- Chương 1. Giới thiệu chung
- **Chương 2. Cơ bản về logic số**
- Chương 3. Hệ thống máy tính
- Chương 4. Số học máy tính
- Chương 5. Kiến trúc tập lệnh
- Chương 6. Cấu trúc bộ xử lý
- Chương 7. Bộ nhớ
- Chương 8. Vào-ra
- Chương 9. Các kiến trúc song song

14 January 2013 Kiến trúc máy tính 48

NKK-HUST

Nội dung của chương 2

- 2.1. Các hệ đếm cơ bản
- 2.2. Đại số Boole
- 2.3. Các cổng logic
- 2.4. Mạch tổ hợp
- 2.5. Mạch dãy

14 January 2013 Kiến trúc máy tính 49

NKK-HUST

2.1. Các hệ đếm cơ bản

- Hệ thập phân (Decimal System)
→ con người sử dụng
- Hệ nhị phân (Binary System)
→ máy tính sử dụng
- Hệ mười sáu (Hexadecimal System)
→ dùng để viết gọn cho số nhị phân

14 January 2013 Kiến trúc máy tính 50

NKK-HUST

1. Hệ thập phân

- Cơ số 10
- 10 chữ số: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Dùng n chữ số thập phân có thể biểu diễn được 10^n giá trị khác nhau:
 - 00...000 = 0
 - 99...999 = $10^n - 1$

14 January 2013 Kiến trúc máy tính 51

NKK-HUST

Dạng tổng quát của số thập phân

$$A = a_n a_{n-1} \dots a_1 a_0, a_{-1} \dots a_{-m}$$

Giá trị của A được hiểu như sau:

$$A = a_n 10^n + a_{n-1} 10^{n-1} + \dots + a_1 10^1 + a_0 10^0 + a_{-1} 10^{-1} + \dots + a_{-m} 10^{-m}$$

$$A = \sum_{i=-m}^n a_i 10^i$$

14 January 2013 Kiến trúc máy tính 52

NKK-HUST

Ví dụ số thập phân

$472.38 = 4 \times 10^2 + 7 \times 10^1 + 2 \times 10^0 + 3 \times 10^{-1} + 8 \times 10^{-2}$

- Các chữ số của phần nguyên:
 - $472 : 10 = 47 \text{ dư } 2$
 - $47 : 10 = 4 \text{ dư } 7$
 - $4 : 10 = 0 \text{ dư } 4$
- Các chữ số của phần lẻ:
 - $0.38 \times 10 = 3.8 \text{ phần nguyên} = 3$
 - $0.8 \times 10 = 8.0 \text{ phần nguyên} = 8$

14 January 2013

Kiến trúc máy tính

53

NKK-HUST

2. Hệ nhị phân

- Cơ số 2
- 2 chữ số nhị phân: 0 và 1
- chữ số nhị phân gọi là **bit** (binary digit)
- Bit là đơn vị thông tin nhỏ nhất
- Dùng n bit có thể biểu diễn được 2^n giá trị khác nhau:
 - $00...000 = 0$
 - $11...111 = 2^n - 1$

14 January 2013

Kiến trúc máy tính

54

NKK-HUST

Bits, Bytes, Nibbles...

- Bits

10010110

most significant bit

least significant bit
- Bytes & Nibbles

10010110

byte

nibble
- Bytes

CEBF9AD7

most significant byte

least significant byte

14 January 2013

Kiến trúc máy tính

55

NKK-HUST

Lũy thừa hai

- $2^{10} = 1 \text{ kilo} \approx 1000 \text{ (1024)}$
- $2^{20} = 1 \text{ mega} \approx 1 \text{ triệu (1,048,576)}$
- $2^{30} = 1 \text{ giga} \approx 1 \text{ tỷ (1,073,741,824)}$
- $2^{40} = 1 \text{ tera} \approx 1000 \text{ tỷ}$
- $2^{50} = 1 \text{ peta} \approx 1 \text{ triệu tỷ}$

14 January 2013

Kiến trúc máy tính

56

57

58

59

60

NKK-HUST

Phương pháp phân tích thành tổng của các 2^i

■ Ví dụ 1: chuyển đổi $105_{(10)}$

■ $105 = 64 + 32 + 8 + 1 = 2^6 + 2^5 + 2^3 + 2^0$

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
0	1	1	0	1	0	0	1

■ Kết quả: $105_{(10)} = 0110\ 1001_{(2)}$

■ Ví dụ 2: $17000_{(10)} = 16384 + 512 + 64 + 32 + 8$
 $= 2^{14} + 2^9 + 2^6 + 2^5 + 2^3$
 $17000_{(10)} = 0100\ 0010\ 0110\ 1000_{(2)}$

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

14 January 2013

Kiến trúc máy tính

61

NKK-HUST

Chuyển đổi số lẻ thập phân sang nhị phân

■ Ví dụ 1: chuyển đổi $0.6875_{(10)}$

■ $0.6875 \times 2 = 1.375$ phần nguyên = 1

■ $0.375 \times 2 = 0.75$ phần nguyên = 0

■ $0.75 \times 2 = 1.5$ phần nguyên = 1

■ $0.5 \times 2 = 1.0$ phần nguyên = 1

■ Kết quả : $0.6875_{(10)} = 0.1011_{(2)}$

14 January 2013

Kiến trúc máy tính

62

NKK-HUST

Chuyển đổi số lẻ thập phân sang nhị phân (tiếp)

■ Ví dụ 2: chuyển đổi $0.81_{(10)}$

■ $0.81 \times 2 = 1.62$ phần nguyên = 1

■ $0.62 \times 2 = 1.24$ phần nguyên = 1

■ $0.24 \times 2 = 0.48$ phần nguyên = 0

■ $0.48 \times 2 = 0.96$ phần nguyên = 0

■ $0.96 \times 2 = 1.92$ phần nguyên = 1

■ $0.92 \times 2 = 1.84$ phần nguyên = 1

■ $0.84 \times 2 = 1.68$ phần nguyên = 1

■ $0.81_{(10)} \approx 0.1100111_{(2)}$

14 January 2013

Kiến trúc máy tính

63

NKK-HUST

Chuyển đổi số lẻ thập phân sang nhị phân (tiếp)

■ Ví dụ 3: chuyển đổi $0.2_{(10)}$

■ $0.2 \times 2 = 0.4$ phần nguyên = 0

■ $0.4 \times 2 = 0.8$ phần nguyên = 0

■ $0.8 \times 2 = 1.6$ phần nguyên = 1

■ $0.6 \times 2 = 1.2$ phần nguyên = 1

■ $0.2 \times 2 = 0.4$ phần nguyên = 0

■ $0.4 \times 2 = 0.8$ phần nguyên = 0

■ $0.8 \times 2 = 1.6$ phần nguyên = 1

■ $0.6 \times 2 = 1.2$ phần nguyên = 1

■ $0.2_{(10)} \approx 0.00110011_{(2)}$

14 January 2013

Kiến trúc máy tính

64

NKK-HUST

3. Hệ mười sáu (Hexa)

- Cơ số 16
- 16 chữ số: 0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F
- Dùng để viết gọn cho số nhị phân: cứ một nhóm 4-bit sẽ được thay bằng một chữ số Hexa

14 January 2013 Kiến trúc máy tính 65

NKK-HUST

Quan hệ giữa số nhị phân và số Hexa

4-bit	Chữ số Hexa
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Ví dụ chuyển đổi số nhị phân → số Hexa:

- $1011\ 0011_2 = B3_{16}$
- $0000\ 0000_2 = 00_{16}$
- $0010\ 1101\ 1001\ 1010_2 = 2D9A_{16}$
- $1111\ 1111\ 1111\ 1111_2 = FFFF_{16}$

14 January 2013 Kiến trúc máy tính 66

NKK-HUST

2.2. Đại số Boole

- Đại số Boole sử dụng các biến logic và phép toán logic
- Biến logic có thể nhận giá trị 1 (TRUE) hoặc 0 (FALSE)
- Phép toán logic cơ bản là AND, OR và NOT với ký hiệu như sau:
 - A AND B : $A \cdot B$
 - A OR B : $A + B$
 - NOT A : \overline{A}
- Thứ tự ưu tiên: NOT > AND > OR

14 January 2013 Kiến trúc máy tính 67

NKK-HUST

Các phép toán logic (tiếp)

- Các phép toán NAND, NOR, XOR:
 - A NAND B : $\overline{A \cdot B}$
 - A NOR B : $\overline{A + B}$
 - A XOR B : $A \oplus B = A \cdot \overline{B} + \overline{A} \cdot B$

14 January 2013 Kiến trúc máy tính 68

NKK-HUST

Phép toán đại số Boole

P	Q	\overline{P}	P AND Q $P \cdot Q$	P OR Q $P + Q$	P NAND Q $\overline{P \cdot Q}$	P NOR Q $\overline{P + Q}$	P XOR Q $P \oplus Q$
0	0	1	0	0	1	1	0
0	1	1	0	1	1	0	1
1	0	0	0	1	1	0	1
1	1	0	1	1	0	0	0

14 January 2013

Kiến trúc máy tính

69

NKK-HUST

Các đồng nhất thức của đại số Boole

$A \cdot B = B \cdot A$ $A \cdot (B + C) = (A \cdot B) + (A \cdot C)$ $1 \cdot A = A$ $A \cdot \overline{A} = 0$	$A + B = B + A$ $A + (B \cdot C) = (A + B) \cdot (A + C)$ $0 + A = A$ $A + \overline{A} = 1$
$0 \cdot A = 0$ $A \cdot A = A$ $A \cdot (B \cdot C) = (A \cdot B) \cdot C$ $\overline{A \cdot B} = \overline{A} + \overline{B}$ (Định lý De Morgan)	$1 + A = 1$ $A + A = A$ $A + (B + C) = (A + B) + C$ $\overline{A + B} = \overline{A} \cdot \overline{B}$ (Định lý De Morgan)

14 January 2013

Kiến trúc máy tính

70

NKK-HUST

2.3. Các cổng logic (Logic Gates)

- Thực hiện các hàm logic:
 - NOT, AND, OR, NAND, NOR, etc.
- Cổng logic một đầu vào:
 - Cổng NOT
- Cổng hai đầu vào:
 - AND, OR, XOR, NAND, NOR, XNOR
- Cổng nhiều đầu vào

14 January 2013

Kiến trúc máy tính

71

NKK-HUST

Các cổng logic

Name	Graphical Symbol	Algebraic Function	Truth Table															
AND		$F = A \cdot B$ or $F = AB$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	0	1	0	0	1	1	1
A	B	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$F = A + B$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	1
A	B	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NOT		$F = \overline{A}$ or $F = A'$	<table><tr><th>A</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	F	0	1	1	0									
A	F																	
0	1																	
1	0																	
NAND		$F = \overline{AB}$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	1	0	1	1	1	0	1	1	1	0
A	B	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$F = \overline{A + B}$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	1	0	1	0	1	0	0	1	1	0
A	B	F																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
XOR		$F = A \oplus B$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	0
A	B	F																
0	0	0																
0	1	1																
1	0	1																
1	1	0																

14 January 2013

Kiến trúc máy tính

72

NKK-HUST

Tập đầy đủ

- Là tập các cổng có thể thực hiện được bất kỳ hàm logic nào từ các cổng của tập đó.
- Một số ví dụ về tập đầy đủ:
 - {AND, OR, NOT}
 - {AND, NOT}
 - {OR, NOT}
 - {NAND}
 - {NOR}

14 January 2013

Kiến trúc máy tính

73

NKK-HUST

Sử dụng cổng NAND

The diagrams illustrate the implementation of basic logic gates using only NAND gates:

- NOT gate:** A single input A is connected to a NAND gate, with the other input also connected to A. The output is \bar{A} .
- AND gate:** Two inputs A and B are connected to a NAND gate. The output is $\overline{A \cdot B}$. This output is then connected to another NAND gate with both inputs tied together, resulting in the final output $A \cdot B$.
- OR gate:** Two inputs A and B are each connected to a separate NAND gate with both inputs tied together, producing \bar{A} and \bar{B} . These two outputs are then connected to a third NAND gate, resulting in the final output $A + B$.

14 January 2013

Kiến trúc máy tính

74

NKK-HUST

Sử dụng cổng NOR

The diagrams illustrate the implementation of basic logic gates using only NOR gates:

- NOT gate:** A single input A is connected to a NOR gate, with the other input also connected to A. The output is \bar{A} .
- OR gate:** Two inputs A and B are connected to a NOR gate. The output is $\overline{A + B}$. This output is then connected to another NOR gate with both inputs tied together, resulting in the final output $A + B$.
- AND gate:** Two inputs A and B are each connected to a separate NOR gate with both inputs tied together, producing \bar{A} and \bar{B} . These two outputs are then connected to a third NOR gate, resulting in the final output $A \cdot B$.

14 January 2013

Kiến trúc máy tính

75

NKK-HUST

Một số ví dụ vi mạch logic

This section displays eight specific logic circuit diagrams for different integrated circuits:

- 7400: Quad 2-input NAND gates.
- 7401: Quad 2-input NOR gates.
- 7402: Quad 2-input inverters (NOT gates).
- 7403: Hex 2-input inverters (NOT gates).
- 7404: Hex 2-input inverters (NOT gates).
- 7405: Hex 2-input inverters (NOT gates).
- 7406: Hex 2-input inverters (NOT gates).
- 7407: Hex 2-input inverters (NOT gates).

14 January 2013

76

NKK-HUST

2.4. Mạch tổ hợp

- Mạch logic là mạch bao gồm:
 - Các đầu vào (Inputs)
 - Các đầu ra (Outputs)
 - Đặc tả chức năng (Functional specification)
 - Đặc tả thời gian (Timing specification)
- Các kiểu mạch logic:
 - Mạch logic tổ hợp (Combinational Logic)
 - Mạch không nhớ
 - Đầu ra được xác định bởi các giá trị hiện tại của đầu vào
 - Mạch logic dãy (Sequential Logic)
 - Mạch có nhớ
 - Đầu ra được xác định bởi các giá trị trước đó và giá trị hiện tại của đầu vào

14 January 2013

Kiến trúc máy tính

77

NKK-HUST

Mạch tổ hợp

- Mạch tổ hợp là mạch logic trong đó đầu ra chỉ phụ thuộc đầu vào ở thời điểm hiện tại.
- Là mạch không nhớ và được thực hiện bằng các cổng logic cơ bản
- Mạch tổ hợp có thể được định nghĩa theo ba cách:
 - Bảng thật
 - Dạng sơ đồ
 - Phương trình Boole

14 January 2013

Kiến trúc máy tính

78

NKK-HUST

Ví dụ

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

$$F = \overline{A}B\overline{C} + \overline{A}BC + ABC$$

14 January 2013

Kiến trúc máy tính

79

NKK-HUST

Bộ dồn kênh (Multiplexer-MUX)

- 2^n đầu vào dữ liệu
- n đầu vào chọn
- 1 đầu ra
- Đầu vào chọn (S) xác định đầu vào nào (D) sẽ được nối với đầu ra (F).

S2	S1	F
0	0	D0
0	1	D1
1	0	D2
1	1	D3

14 January 2013

Kiến trúc máy tính

80

NKK-HUST

Thực hiện MUX bốn đầu vào

14 January 2013

Kiến trúc máy tính

81

NKK-HUST

Bộ giải mã (Decoder)

- N đầu vào, 2^N đầu ra
- Chỉ có một đầu ra tích cực (được chọn) tương ứng với một tổ hợp của N đầu vào.

2:4 Decoder

A_1
 A_0

11
10
01
00

Y_3
 Y_2
 Y_1
 Y_0

A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

14 January 2013

Kiến trúc máy tính

82

NKK-HUST

Thực hiện bộ giải mã 3 ra 8

14 January 2013

Kiến trúc máy tính

83

NKK-HUST

Bộ cộng (Adder)

- Bộ bán tổng (Half-adder)
 - Cộng hai bit tạo ra bit tổng và bit nhớ
- Bộ toàn tổng (Full-adder)
 - Cộng 3 bit
 - Cho phép xây dựng bộ cộng N-bit

14 January 2013

Kiến trúc máy tính

84

NKK-HUST

Bộ cộng (tiếp)

A	B	Sum	C _{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

(a) Half-adder truth table and implementation

A	B	C _{in}	Sum	C _{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

14 January 2013

Kiến trúc máy tính

85

NKK-HUST

Bộ cộng 4-bit

14 January 2013

Kiến trúc máy tính

86

NKK-HUST

2.5. Mạch dãy

- Mạch dãy là mạch logic trong đó đầu ra phụ thuộc giá trị đầu vào ở thời điểm hiện tại và quá khứ
- Là mạch có nhớ, được thực hiện bằng phần tử nhớ (Latch, Flip-Flop) và có thể kết hợp với các cổng logic cơ bản
- Mạch dãy bao gồm:
 - Mạch tổ hợp
 - Mạch hồi tiếp

14 January 2013

Kiến trúc máy tính

87

NKK-HUST

Các thành phần chính của mạch dãy

14 January 2013

Kiến trúc máy tính

88

NKK-HUST

Các Flip-Flop cơ bản

Name	Graphical Symbol	Truth Table															
S-R		<table><thead><tr><th>S</th><th>R</th><th>Q_{n+1}</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>Q_n</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>—</td></tr></tbody></table>	S	R	Q_{n+1}	0	0	Q_n	0	1	0	1	0	1	1	1	—
S	R	Q_{n+1}															
0	0	Q_n															
0	1	0															
1	0	1															
1	1	—															
J-K		<table><thead><tr><th>J</th><th>K</th><th>Q_{n+1}</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>Q_n</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>$\overline{Q_n}$</td></tr></tbody></table>	J	K	Q_{n+1}	0	0	Q_n	0	1	0	1	0	1	1	1	$\overline{Q_n}$
J	K	Q_{n+1}															
0	0	Q_n															
0	1	0															
1	0	1															
1	1	$\overline{Q_n}$															
D		<table><thead><tr><th>D</th><th>Q_{n+1}</th></tr></thead><tbody><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></tbody></table>	D	Q_{n+1}	0	0	1	1									
D	Q_{n+1}																
0	0																
1	1																

14 January 2013

Kiến trúc máy tính

89

NKK-HUST

R-S Latch và các Flip-Flop

R-S Latch

R-S Flip Flop

D Flip Flop

J-K Flip Flop

14 January 2013

Kiến trúc máy tính

90

NKK-HUST

Thanh ghi 8-bit song song

14 January 2013

Kiến trúc máy tính

91

NKK-HUST

Thanh ghi dịch 5-bit

14 January 2013

Kiến trúc máy tính

92

NKK-HUST

Bộ đếm 4-bit

(a) Sequential circuit

(b) Timing diagram

14 January 2013 Kiến trúc máy tính 93

NKK-HUST

Hết chương 2

14 January 2013 Kiến trúc máy tính 94

NKK-HUST

Kiến trúc máy tính

Chương 3

HỆ THỐNG MÁY TÍNH

Nguyễn Kim Khánh
Trường Đại học Bách khoa Hà Nội

Kiến trúc máy tính 95

NKK-HUST

Nội dung học phần

- Chương 1. Giới thiệu chung
- Chương 2. Cơ bản về logic số
- Chương 3. Hệ thống máy tính**
- Chương 4. Số học máy tính
- Chương 5. Kiến trúc tập lệnh
- Chương 6. Cấu trúc bộ xử lý
- Chương 7. Bộ nhớ
- Chương 8. Vào-ra
- Chương 9. Các kiến trúc song song

14 January 2013 Kiến trúc máy tính 96

NKK-HUST

Nội dung của chương 3

- 3.1. Các thành phần cơ bản của máy tính
- 3.2. Hoạt động cơ bản của máy tính
- 3.3. Bus máy tính

14 January 2013 Kiến trúc máy tính 97

NKK-HUST

3.1. Các thành phần cơ bản của máy tính

- Bộ xử lý trung tâm (CPU)
- Bộ nhớ (Memory)
- Vào ra (Input/Output)
- Bus liên kết hệ thống (System Interconnection Bus)

14 January 2013 Kiến trúc máy tính 98

NKK-HUST

1. Bộ xử lý trung tâm (CPU)

- Chức năng:
 - điều khiển hoạt động của máy tính
 - xử lý dữ liệu
- Nguyên tắc hoạt động cơ bản:
CPU hoạt động theo chương trình nằm trong bộ nhớ chính.

14 January 2013 Kiến trúc máy tính 99

NKK-HUST

Cấu trúc cơ bản của CPU

14 January 2013 Kiến trúc máy tính 100

NKK-HUST

Các thành phần cơ bản của CPU

- **Đơn vị điều khiển** (*Control Unit - CU*): điều khiển hoạt động của máy tính theo chương trình đã định sẵn.
- **Đơn vị số học và logic** (*Arithmetic and Logic Unit - ALU*): thực hiện các phép toán số học và phép toán logic.
- **Tập thanh ghi** (*Register File - RF*): lưu giữ các thông tin tạm thời phục vụ cho hoạt động của CPU.
- **Đơn vị nối ghép bus** (*Bus Interface Unit - BIU*) kết nối và trao đổi thông tin giữa bus bên trong (*internal bus*) và bus bên ngoài (*external bus*).

14 January 2013 Kiến trúc máy tính 101

NKK-HUST

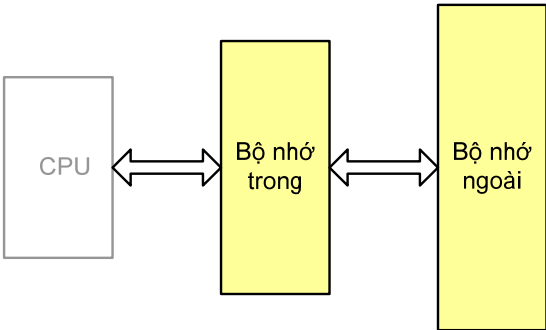
2. Bộ nhớ máy tính

- Chức năng: lưu trữ chương trình và dữ liệu.
- Các thao tác cơ bản với bộ nhớ:
 - Thao tác ghi (Write)
 - Thao tác đọc (Read)
- Các thành phần chính:
 - Bộ nhớ trong (Internal Memory)
 - Bộ nhớ ngoài (External Memory)

14 January 2013 Kiến trúc máy tính 102

NKK-HUST

Các thành phần của bộ nhớ máy tính



```

graph LR
    CPU[CPU] <--> BM[Bộ nhớ trong]
    BM <--> BMN[Bộ nhớ ngoài]
    
```

14 January 2013 Kiến trúc máy tính 103

NKK-HUST

Bộ nhớ trong

- Chức năng và đặc điểm:
 - Chứa các thông tin mà CPU có thể trao đổi trực tiếp
 - Tốc độ rất nhanh
 - Dung lượng không lớn
 - Sử dụng bộ nhớ bán dẫn: ROM và RAM
- Các loại bộ nhớ trong:
 - Bộ nhớ chính
 - Bộ nhớ cache (bộ nhớ đệm)

14 January 2013 Kiến trúc máy tính 104

NKK-HUST

Bộ nhớ chính (Main Memory)

- Chứa các chương trình và dữ liệu đang được CPU sử dụng.
- Tổ chức thành các ngăn nhớ được đánh địa chỉ.
- Ngăn nhớ thường được tổ chức theo byte.
- Nội dung của ngăn nhớ có thể thay đổi, song địa chỉ vật lý của ngăn nhớ luôn cố định.

Nội dung	Địa chỉ
1011 0010	0000
1110 0010	0001
0001 1111	0010
1010 1011	0011
0000 1000	0100
1111 1111	0101
0011 1100	0110
1000 1111	0111
1111 0001	1000
0011 1101	1001
1000 1111	1010
0011 0011	1011
1100 1101	1100
0101 1010	1101
1000 1101	1110
1111 0000	1111

14 January 2013

Kiến trúc máy tính

105

NKK-HUST

Bộ nhớ cache

- Bộ nhớ có tốc độ nhanh được đặt đệm giữa CPU và bộ nhớ chính nhằm tăng tốc độ CPU truy cập bộ nhớ
- Dung lượng nhỏ hơn bộ nhớ chính
- Tốc độ nhanh hơn
- Cache thường được chia thành một số mức
- Cache có thể được tích hợp trên cùng chip bộ xử lý.
- Cache có thể có hoặc không

14 January 2013

Kiến trúc máy tính

106

NKK-HUST

Bộ nhớ ngoài (External Memory)

- Chức năng và đặc điểm
 - Lưu giữ tài nguyên phần mềm của máy tính
 - Được kết nối với hệ thống dưới dạng các thiết bị vào-ra
 - Dung lượng lớn
 - Tốc độ chậm
- Các loại bộ nhớ ngoài
 - Bộ nhớ từ: đĩa cứng, đĩa mềm
 - Bộ nhớ quang: đĩa CD, DVD
 - Bộ nhớ bán dẫn: Flash storage, memory card

14 January 2013

Kiến trúc máy tính

107

NKK-HUST

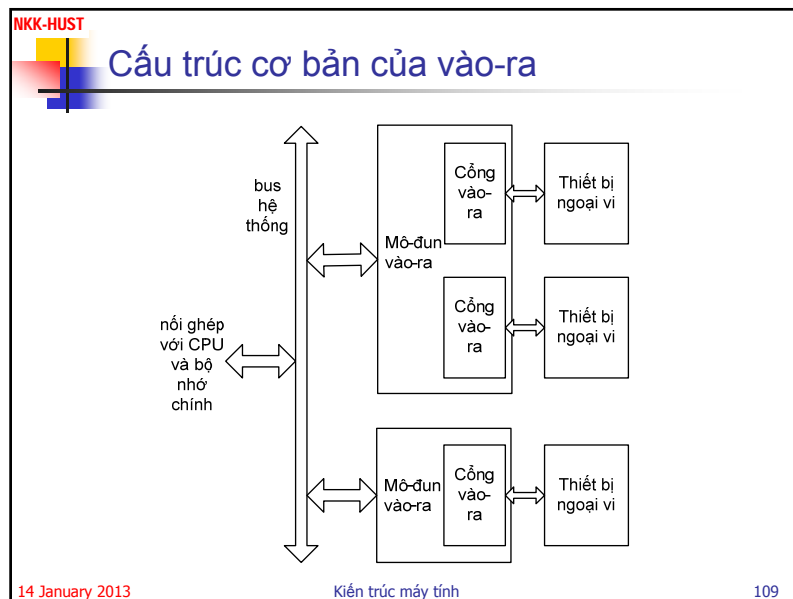
3. Vào-ra (Input-Output)

- Chức năng: Trao đổi thông tin giữa máy tính với thế giới bên ngoài.
- Các thao tác cơ bản:
 - Vào dữ liệu (Input)
 - Ra dữ liệu (Output)
- Các thành phần chính:
 - Các thiết bị ngoại vi (Peripheral Devices)
 - Các mô-đun vào-ra (IO Modules)

14 January 2013

Kiến trúc máy tính

108



- NKK-HUST**
- ### Các thiết bị ngoại vi
- Chức năng: chuyển đổi dữ liệu giữa bên trong và bên ngoài máy tính
 - Các loại thiết bị ngoại vi cơ bản
 - Thiết bị vào: bàn phím, chuột, máy quét ...
 - Thiết bị ra: màn hình, máy in ...
 - Thiết bị nhớ: các ổ đĩa ...
 - Thiết bị truyền thông: MODEM ...
- 14 January 2013
- Kiến trúc máy tính
- 110

- NKK-HUST**
- ### Mô-đun vào-ra
- Chức năng: nối ghép các thiết bị ngoại vi với máy tính
 - Mỗi mô-đun vào-ra có một hoặc một vài cổng vào-ra (I/O Port).
 - Mỗi cổng vào-ra được đánh một địa chỉ xác định.
 - Các thiết bị ngoại vi được kết nối và trao đổi dữ liệu với máy tính thông qua các cổng vào-ra.
- 14 January 2013
- Kiến trúc máy tính
- 111

- NKK-HUST**
- ### 3.2. Hoạt động cơ bản của máy tính
- Thực hiện chương trình
 - Hoạt động ngắt
 - Hoạt động vào-ra
- 14 January 2013
- Kiến trúc máy tính
- 112

NKK-HUST

1. Thực hiện chương trình

- Là hoạt động cơ bản của máy tính
- Máy tính lặp đi lặp lại hai bước:
 - Nhận lệnh
 - Thực hiện lệnh
- Thực hiện chương trình bị dừng nếu thực hiện lệnh bị lỗi hoặc gặp lệnh dừng.

14 January 2013

Kiến trúc máy tính

113

NKK-HUST

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.

14 January 2013

Kiến trúc máy tính

114

NKK-HUST

Minh họa quá trình nhận lệnh

The diagram illustrates the instruction reception process in two states:

- Trước khi nhận lệnh i (Before receiving instruction i):** The CPU's PC register contains the value 302. It points to the memory location 302, which contains 'lệnh i'. The IR register is empty.
- Sau khi nhận lệnh i (After receiving instruction i):** The CPU's PC register has incremented to 303. It now points to the memory location 303, which contains 'lệnh i+1'. The IR register now contains 'lệnh i'.

14 January 2013

Kiến trúc máy tính

115

NKK-HUST

Thực hiện lệnh

- Bộ xử lý giải mã lệnh đã được nhận và phát tín hiệu điều khiển thực hiện thao tác mà lệnh yêu cầu.
- Các kiểu thao tác của lệnh:
 - Trao đổi dữ liệu giữa CPU và bộ nhớ chính
 - Trao đổi dữ liệu giữa CPU và mô-đun vào-ra
 - 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.

14 January 2013

Kiến trúc máy tính

116

NKK-HUST

2. Hoạt động ngắt (Interrupt)

- **Khái niệm chung về ngắt:** Ngắt là cơ chế cho phép CPU tạm dừng chương trình đang thực hiện để chuyển sang thực hiện một chương trình khác, gọi là *chương trình con phục vụ ngắt*.
- Các loại ngắt:
 - Ngắt do lỗi khi thực hiện chương trình, ví dụ: tràn số, chia cho 0.
 - Ngắt do lỗi phần cứng, ví dụ lỗi bộ nhớ RAM.
 - Ngắt do mô-đun vào-ra phát tín hiệu ngắt đến CPU yêu cầu trao đổi dữ liệu.
 - Ngắt do bộ định thời trong chế độ đa chương trình

14 January 2013

Kiến trúc máy tính

117

NKK-HUST

Hoạt động ngắt (tiếp)

- Sau khi hoàn thành mỗi một lệnh, bộ xử lý kiểm tra tín hiệu ngắt
- Nếu không có ngắt → bộ xử lý nhận lệnh tiếp theo của chương trình hiện tại
- Nếu có tín hiệu ngắt:
 - Tạm dừng chương trình đang thực hiện
 - Cắt ngữ cảnh (các thông tin liên quan đến chương trình bị ngắt)
 - Thiết lập PC trỏ đến chương trình con phục vụ ngắt
 - Chuyển sang thực hiện chương trình con phục vụ ngắt
 - Cuối chương trình con phục vụ ngắt, khôi phục ngữ cảnh và tiếp tục chương trình đang bị tạm dừng

14 January 2013

Kiến trúc máy tính

118

NKK-HUST

Hoạt động ngắt (tiếp)

Chương trình đang thực hiện

lệnh

lệnh

lệnh

lệnh i

lệnh i+1

lệnh

...

Ngắt ở đây

Chương trình con phục vụ ngắt

lệnh

lệnh

lệnh

...

RETURN

14 January 2013

Kiến trúc máy tính

119

NKK-HUST

Xử lý với nhiều tín hiệu yêu cầu ngắt

- **Xử lý ngắt tuần tự**
 - Khi một ngắt đang được thực hiện, các ngắt khác sẽ bị cấm.
 - Bộ xử lý sẽ bỏ qua các ngắt tiếp theo trong khi đang xử lý một ngắt
 - Các yêu cầu ngắt vẫn đang đợi và được kiểm tra sau khi ngắt đầu tiên được xử lý xong
 - Các ngắt được thực hiện tuần tự

User Program

Interrupt Handler X

Interrupt Handler Y

14 January 2013

Kiến trúc máy tính

120

NKK-HUST

Xử lý với nhiều tín hiệu yêu cầu ngắt...

- Xử lý ngắt ưu tiên
 - Các ngắt được định nghĩa mức ưu tiên khác nhau
 - Ngắt có mức ưu tiên thấp hơn có thể bị ngắt bởi ngắt ưu tiên cao hơn
 - Xảy ra ngắt lồng nhau



14 January 2013 Kiến trúc máy tính 121

NKK-HUST

3. Hoạt động vào-ra

- Hoạt động vào-ra: là hoạt động trao đổi dữ liệu giữa mô-đun vào-ra với bên trong máy tính.
- Các kiểu hoạt động vào-ra:
 - CPU trao đổi dữ liệu với mô-đun vào-ra
 - Mô-đun vào-ra trao đổi dữ liệu trực tiếp với bộ nhớ chính (DMA- Direct Memory Access).

14 January 2013 Kiến trúc máy tính 122

NKK-HUST

3.3. Bus máy tính

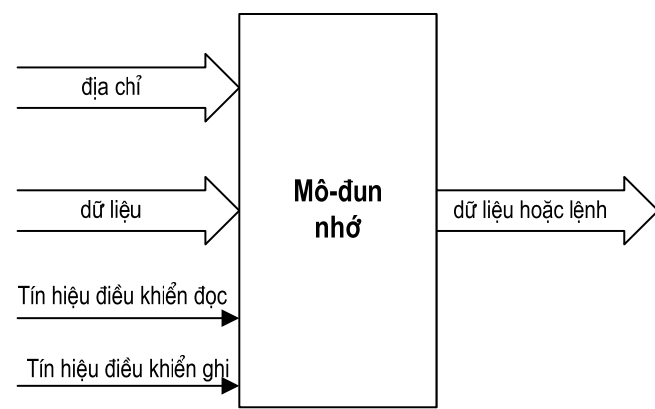
1. Luồng thông tin trong máy tính

- Các mô-đun trong máy tính:
 - CPU
 - Mô-đun nhớ
 - Mô-đun vào-ra
- cần được kết nối với nhau

14 January 2013 Kiến trúc máy tính 123


NKK-HUST

Kết nối mô-đun nhớ



14 January 2013 Kiến trúc máy tính 124

NKK-HUST



Kết nối mô-đun nhớ (tiếp)


- Địa chỉ đưa đến để xác định ngăn nhớ
- Dữ liệu được đưa đến khi ghi
- Dữ liệu hoặc lệnh được đưa ra khi đọc (lưu ý: bộ nhớ không phân biệt lệnh và dữ liệu)
- Nhận các tín hiệu điều khiển:
 - Điều khiển đọc (Read)
 - Điều khiển ghi (Write)

14 January 2013

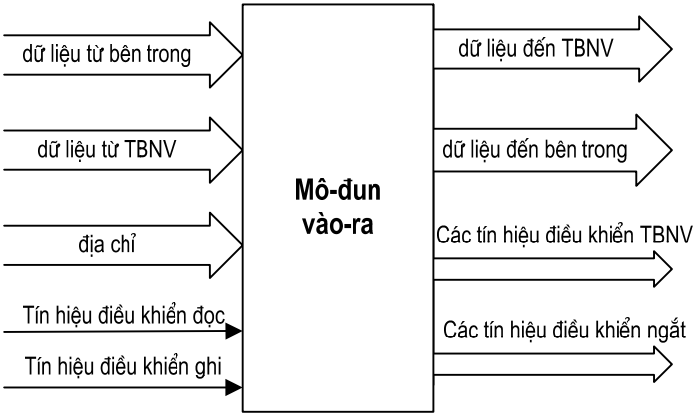
Kiến trúc máy tính

125

NKK-HUST



Kết nối mô-đun vào-ra




The diagram shows a central box labeled 'Mô-đun vào-ra'. On the left, four arrows point into the box: 'dữ liệu từ bên trong', 'dữ liệu từ TBNV', 'địa chỉ', and 'Tín hiệu điều khiển đọc'. On the right, four arrows point out of the box: 'dữ liệu đến TBNV', 'dữ liệu đến bên trong', 'Các tín hiệu điều khiển TBNV', and 'Các tín hiệu điều khiển ngắt'.

14 January 2013

Kiến trúc máy tính

126

NKK-HUST



Kết nối mô-đun vào-ra (tiếp)


- Địa chỉ đưa đến để xác định cổng vào-ra
- Ra dữ liệu (Output)
 - Nhận dữ liệu từ CPU hoặc bộ nhớ chính
 - Đưa dữ liệu ra thiết bị ngoại vi
- Vào dữ liệu (Input)
 - Nhận dữ liệu từ thiết bị ngoại vi
 - Đưa dữ liệu vào CPU hoặc bộ nhớ chính
- Nhận các tín hiệu điều khiển từ CPU
- Phát các tín hiệu điều khiển đến thiết bị ngoại vi
- Phát các tín hiệu ngắt đến CPU

14 January 2013

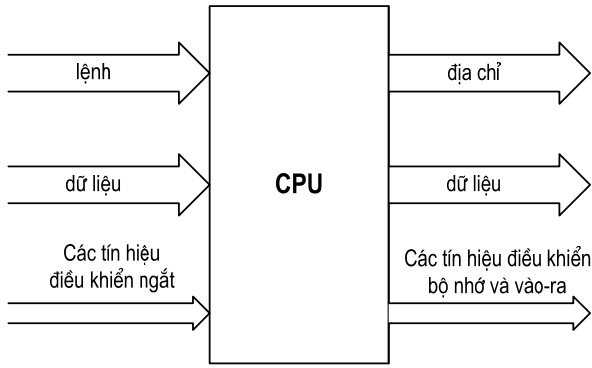
Kiến trúc máy tính

127

NKK-HUST



Kết nối CPU



The diagram shows a central box labeled 'CPU'. On the left, three arrows point into the box: 'lệnh', 'dữ liệu', and 'Các tín hiệu điều khiển ngắt'. On the right, three arrows point out of the box: 'địa chỉ', 'dữ liệu', and 'Các tín hiệu điều khiển bộ nhớ và vào-ra'.

14 January 2013

Kiến trúc máy tính

128

NKK-HUST

Kết nối CPU (tiếp)

- Phát địa chỉ đến các mô-đun nhớ hay các mô-đun vào-ra
- Đọc lệnh và dữ liệu
- Đưa dữ liệu ra (sau khi xử lý)
- Phát tín hiệu điều khiển đến các mô-đun nhớ và các mô-đun vào-ra
- Nhận các tín hiệu ngắt

14 January 2013 Kiến trúc máy tính 129

NKK-HUST

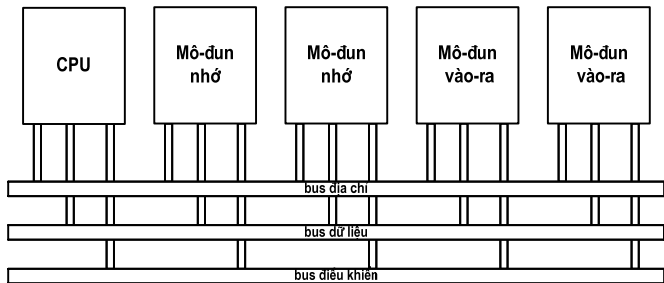
2. Cấu trúc bus cơ bản

- **Bus**: tập hợp các đường kết nối dùng để vận chuyển thông tin giữa các mô-đun của máy tính với nhau.
- **Các bus chức năng**:
 - Bus địa chỉ
 - Bus dữ liệu
 - Bus điều khiển
- **Độ rộng bus**: là số đường dây của bus có thể truyền các bit thông tin đồng thời (chỉ dùng cho bus địa chỉ và bus dữ liệu)

14 January 2013 Kiến trúc máy tính 130

NKK-HUST

Sơ đồ cấu trúc bus cơ bản



The diagram illustrates a basic bus structure. At the top, five modules are shown: CPU, Mô-đun nhớ (Memory Module), Mô-đun nhớ (Memory Module), Mô-đun vào-ra (I/O Module), and Mô-đun vào-ra (I/O Module). Each module is connected to three horizontal bus lines below it. The top line is labeled 'bus địa chỉ' (Address Bus), the middle line is labeled 'bus dữ liệu' (Data Bus), and the bottom line is labeled 'bus điều khiển' (Control Bus). The connections show that all modules share these three common buses.

14 January 2013 Kiến trúc máy tính 131

NKK-HUST

Bus địa chỉ

- **Chức năng**: vận chuyển địa chỉ để xác định ngăn nhớ hay cổng vào-ra
- **Độ rộng bus địa chỉ**: cho biết số lượng ngăn nhớ tối đa được đánh địa chỉ.
 - N bit: $A_{N-1}, A_{N-2}, \dots, A_2, A_1, A_0$
 - có thể đánh địa chỉ tối đa cho 2^N ngăn nhớ (không gian địa chỉ bộ nhớ)
- **Ví dụ**:
 - Bộ xử lý Pentium có bus địa chỉ 32 bit
 - có khả năng đánh địa chỉ cho 2^{32} bytes nhớ (4GBytes) (ngăn nhớ tổ chức theo byte)

14 January 2013 Kiến trúc máy tính 132

NKK-HUST

Bus dữ liệu

- **Chức năng:**
 - vận chuyển lệnh từ bộ nhớ đến CPU
 - vận chuyển dữ liệu giữa CPU, mô đun nhớ, mô đun vào-ra với nhau
- **Độ rộng bus dữ liệu:** Xác định số bit dữ liệu có thể được trao đổi đồng thời.
 - M bit: $D_{M-1}, D_{M-2}, \dots, D_2, D_1, D_0$
 - M thường là 8, 16, 32, 64, 128 bit.
- **Ví dụ:** Các bộ xử lý Pentium có bus dữ liệu 64 bit

14 January 2013 Kiến trúc máy tính 133

NKK-HUST

Bus điều khiển

- **Chức năng:** vận chuyển các tín hiệu điều khiển
- **Các loại tín hiệu điều khiển:**
 - Các tín hiệu điều khiển đọc/ghi
 - Các tín hiệu điều khiển ngắt
 - Các tín hiệu điều khiển bus

14 January 2013 Kiến trúc máy tính 134

NKK-HUST

Một số tín hiệu điều khiển điển hình

- Các tín hiệu (phát ra từ CPU) điều khiển đọc-ghi:
 - *Memory Read* (MEMR): điều khiển đọc dữ liệu từ một ngăn nhớ có địa chỉ xác định lên bus dữ liệu.
 - *Memory Write* (MEMW): điều khiển ghi dữ liệu có sẵn trên bus dữ liệu đến một ngăn nhớ có địa chỉ xác định.
 - *I/O Read* (IOR): điều khiển đọc dữ liệu từ một cổng vào-ra có địa chỉ xác định lên bus dữ liệu.
 - *I/O Write* (IOW): điều khiển ghi dữ liệu có sẵn trên bus dữ liệu ra một cổng có địa chỉ xác định.

14 January 2013 Kiến trúc máy tính 135

NKK-HUST

Một số tín hiệu điều khiển điển hình (tiếp)

- Các tín hiệu điều khiển ngắt:
 - *Interrupt Request* (INTR): Tín hiệu từ bộ điều khiển vào-ra gửi đến yêu cầu ngắt CPU để trao đổi vào-ra. Tín hiệu INTR có thể bị che.
 - *Interrupt Acknowledge* (INTA): Tín hiệu phát ra từ CPU báo cho bộ điều khiển vào-ra biết CPU chấp nhận ngắt để trao đổi vào-ra.
 - *Non Maskable Interrupt* (NMI): tín hiệu ngắt không che được gửi đến ngắt CPU.
 - *Reset*: Tín hiệu từ bên ngoài gửi đến CPU và các thành phần khác để khởi động lại máy tính.

14 January 2013 Kiến trúc máy tính 136

NKK-HUST

Một số tín hiệu điều khiển điển hình (tiếp)

- Các tín hiệu điều khiển bus:
 - *Bus Request* (BRQ) hay là *Hold*: Tín hiệu từ mô-đun điều khiển vào-ra gửi đến yêu cầu CPU chuyển nhượng quyền sử dụng bus.
 - *Bus Grant* (BGT) hay là *Hold Acknowledge* (HLDA): Tín hiệu phát ra từ CPU chấp nhận chuyển nhượng quyền sử dụng bus.
 - *Lock/ Unlock*: Tín hiệu *cấm/cho-phép* xin chuyển nhượng bus

14 January 2013 Kiến trúc máy tính 137

NKK-HUST

Đặc điểm của cấu trúc đơn bus

- Bus hệ thống chỉ phục vụ được một yêu cầu trao đổi dữ liệu tại một thời điểm
- Bus hệ thống phải có tốc độ bằng tốc độ bus của mô-đun nhanh nhất trong hệ thống
- Bus hệ thống phụ thuộc vào cấu trúc bus (các tín hiệu) của bộ xử lý → các mô-đun nhớ và các mô-đun vào-ra cũng phụ thuộc vào bộ xử lý.
- Khắc phục: phân cấp bus → cấu trúc đa bus

14 January 2013 Kiến trúc máy tính 138

NKK-HUST

3. Phân cấp bus trong máy tính

- Tổ chức thành nhiều bus trong hệ thống máy tính
 - Cho các thành phần khác nhau:
 - Bus của bộ xử lý
 - Bus của bộ nhớ chính
 - Các bus vào-ra
 - Các bus khác nhau về tốc độ
- Bus bộ nhớ chính và các bus vào-ra không phụ thuộc vào bộ xử lý cụ thể.

14 January 2013 Kiến trúc máy tính 139

NKK-HUST

Một số bus điển hình trong máy để bàn

- Bus của bộ xử lý: có tốc độ nhanh nhất
- Bus của bộ nhớ chính (nối ghép với các mô-đun RAM)
- PCI Express bus (Peripheral Component Interconnect): nối ghép với các thiết bị ngoại vi có tốc độ trao đổi dữ liệu nhanh.
- SATA (Serial Advanced Technology Attachment): Bus kết nối với ổ đĩa cứng hoặc ổ đĩa CD/DVD
- USB (Universal Serial Bus): Bus nối tiếp đa năng

14 January 2013 Kiến trúc máy tính 140

141

142

143

144

NKK-HUST

Nội dung chương 4

- 4.1. Biểu diễn số nguyên
- 4.2. Phép cộng và phép trừ số nguyên
- 4.3. Phép nhân và phép chia số nguyên
- 4.4. Số dấu phẩy động

14 January 2013 Kiến trúc máy tính 145

NKK-HUST

4.1. Biểu diễn số nguyên

- Số nguyên không dấu (Unsigned Integer)
- Số nguyên có dấu (Signed Integer)

14 January 2013 Kiến trúc máy tính 146

NKK-HUST

1. Biểu diễn số nguyên không dấu

- Nguyên tắc tổng quát: Dùng n bit biểu diễn số nguyên không dấu A:

$$a_{n-1}a_{n-2} \dots a_2a_1a_0$$

Giá trị của A được tính như sau:

$$A = \sum_{i=0}^{n-1} a_i 2^i$$

Dài biểu diễn của A: từ 0 đến $2^n - 1$

14 January 2013 Kiến trúc máy tính 147

NKK-HUST

Các ví dụ

- Ví dụ 1. Biểu diễn các số nguyên không dấu sau đây bằng 8-bit:

$A = 41$; $B = 150$

Giải:

$A = 41 = 32 + 8 + 1 = 2^5 + 2^3 + 2^0$
 $41 = 0010\ 1001$

$B = 150 = 128 + 16 + 4 + 2 = 2^7 + 2^4 + 2^2 + 2^1$
 $150 = 1001\ 0110$

14 January 2013 Kiến trúc máy tính 148

NKK-HUST

Các ví dụ (tiếp)

■ Ví dụ 2. Cho các số nguyên không dấu M, N được biểu diễn bằng 8-bit như sau:

■ M = 0001 0010

■ N = 1011 1001

Xác định giá trị của chúng ?

Giải:

■ M = 0001 0010 = 2⁴ + 2¹ = 16 + 2 = 18

■ N = 1011 1001 = 2⁷ + 2⁵ + 2⁴ + 2³ + 2⁰
= 128 + 32 + 16 + 8 + 1 = 185

14 January 2013

Kiến trúc máy tính

149

NKK-HUST

Với n = 8 bit

Biểu diễn được các giá trị từ 0 đến 255

0000 0000 = 0

0000 0001 = 1

0000 0010 = 2

0000 0011 = 3

...

1111 1111 = 255

Chú ý:

1111 1111

+ 0000 0001

1 0000 0000

Vậy: 255 + 1 = 0 ?

→ do tràn nhớ ra ngoài

14 January 2013

Kiến trúc máy tính

150

NKK-HUST

Trục số học với n = 8 bit

Trục số học:

Trục số học máy tính:

14 January 2013

Kiến trúc máy tính

151

NKK-HUST

Với n = 16 bit, 32 bit, 64 bit

■ n = 16 bit: dải biểu diễn từ 0 đến 65535 (2¹⁶ – 1)

■ 0000 0000 0000 0000 = 0

...

■ 0000 0000 1111 1111 = 255

■ 0000 0001 0000 0000 = 256

...

■ 1111 1111 1111 1111 = 65535

■ n = 32 bit: dải biểu diễn từ 0 đến 2³² - 1

■ n = 64 bit: dải biểu diễn từ 0 đến 2⁶⁴ - 1

14 January 2013

Kiến trúc máy tính

152

NKK-HUST

2. Biểu diễn số nguyên có dấu

Số bù chín và Số bù mười

■ Cho một số thập phân A được biểu diễn bằng n chữ số thập phân, ta có:

■ Số bù chín của $A = (10^n - 1) - A$

■ Số bù mười của $A = 10^n - A$

■ Số bù mười của $A = (\text{Số bù chín của } A) + 1$

14 January 2013

Kiến trúc máy tính

153

NKK-HUST

Số bù chín và Số bù mười (tiếp)

■ Ví dụ: với $n=4$, cho $A = 3265$

■ Số bù chín của A:

9999 $(10^4 - 1)$

- 3265 (A)

6734

■ Số bù mười của A:

10000 (10^4)

- 3265 (A)

6735

14 January 2013

Kiến trúc máy tính

154

NKK-HUST

Số bù một và Số bù hai

■ Định nghĩa: Cho một số nhị phân A được biểu diễn bằng n bit, ta có:

■ Số bù một của $A = (2^n - 1) - A$

■ Số bù hai của $A = 2^n - A$

■ Số bù hai của $A = (\text{Số bù một của } A) + 1$

14 January 2013

Kiến trúc máy tính

155

NKK-HUST

Số bù một và Số bù hai (tiếp)

■ Ví dụ: với $n = 8$ bit, cho $A = 0010\ 0101$

■ Số bù một của A được tính như sau:

1111 1111 $(2^8 - 1)$

- 0010 0101 (A)

1101 1010

→ đảo các bit của A

■ Số bù hai của A được tính như sau:

1 0000 0000 (2^8)

- 0010 0101 (A)

1101 1011

→ thực hiện khó khăn

14 January 2013

Kiến trúc máy tính

156

Nguyễn Kim Khánh - ĐHBKHN

NKK-HUST

Quy tắc tìm Số bù một và Số bù hai

- Số bù một của A = đảo giá trị các bit của A
- (Số bù hai của A) = (Số bù một của A) + 1
- Ví dụ:
 - Cho A = 0010 0101
 - Số bù một = 1101 1010
 - Số bù hai = 1101 1011
- Nhận xét:

A	=	0010 0101
Số bù hai	=	+ 1101 1011
		+ 1
		1 0000 0000 = 0

(bỏ qua bit nhớ ra ngoài)

→ Số bù hai của A = -A

14 January 2013 Kiến trúc máy tính 157

NKK-HUST

Biểu diễn số nguyên có dấu bằng mã bù hai

Nguyên tắc tổng quát: Dùng n bit biểu diễn số nguyên có dấu A:

$$a_{n-1}a_{n-2} \dots a_2a_1a_0$$

- Với A là số dương: bit $a_{n-1} = 0$, các bit còn lại biểu diễn độ lớn như số không dấu
- Với A là số âm: được biểu diễn bằng số bù hai của số dương tương ứng, vì vậy bit $a_{n-1} = 1$

14 January 2013 Kiến trúc máy tính 158

NKK-HUST

Biểu diễn số dương

- Dạng tổng quát của số dương A:

$$0a_{n-2} \dots a_2a_1a_0$$
- Giá trị của số dương A:

$$A = \sum_{i=0}^{n-2} a_i 2^i$$
- Dải biểu diễn cho số dương: 0 đến $2^{n-1}-1$

14 January 2013 Kiến trúc máy tính 159

NKK-HUST

Biểu diễn số âm

- Dạng tổng quát của số âm A:

$$1a_{n-2} \dots a_2a_1a_0$$
- Giá trị của số âm A:

$$A = -2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i$$
- Dải biểu diễn cho số âm: -1 đến -2^{n-1}

14 January 2013 Kiến trúc máy tính 160

NKK-HUST

Biểu diễn tổng quát cho số nguyên có dấu

- Dạng tổng quát của số nguyên A:
$$a_{n-1}a_{n-2} \dots a_2a_1a_0$$
- Giá trị của A được xác định như sau:
$$A = -a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} a_i2^i$$
- Dải biểu diễn: từ $-(2^{n-1})$ đến $+(2^{n-1}-1)$

14 January 2013

Kiến trúc máy tính

161

NKK-HUST

Các ví dụ

- Ví dụ 1. Biểu diễn các số nguyên có dấu sau đây bằng 8-bit:
$$A = +58 ; \quad B = -80$$

Giải:

$$A = +58 = 0011 \ 1010$$
$$B = -80$$

Ta có: $+80 = 0101 \ 0000$
Số bù một $= 1010 \ 1111$
$$+ \quad \quad \quad 1$$

Số bù hai $= 1011 \ 0000$

Vậy: $B = -80 = 1011 \ 0000$

14 January 2013

Kiến trúc máy tính

162

NKK-HUST

Các ví dụ

- Ví dụ 2. Hãy xác định giá trị của các số nguyên có dấu được biểu diễn dưới đây:
 - $P = 0110 \ 0010$
 - $Q = 1101 \ 1011$

Giải:

- $P = 0110 \ 0010 = 64+32+2 = +98$
- $Q = 1101 \ 1011 = -128+64+16+8+2+1 = -37$

14 January 2013

Kiến trúc máy tính

163

NKK-HUST

Với $n = 8$ bit

Biểu diễn được các giá trị từ -128 đến +127

0000 0000	=	0
0000 0001	=	+1
0000 0010	=	+2
0000 0011	=	+3
...		
0111 1111	=	+127
1000 0000	=	- 128
1000 0001	=	- 127
...		
1111 1110	=	-2
1111 1111	=	-1

Chú ý:
 $+127 + 1 = -128$
 $-128 - 1 = +127$
→ do tràn xảy ra

14 January 2013

Kiến trúc máy tính

164

NKK-HUST

Trục số học số nguyên có dấu với $n = 8$ bit

■ Trục số học:

■ Trục số học máy tính:

14 January 2013

Kiến trúc máy tính

165

NKK-HUST

Với $n = 16$ bit, 32 bit, 64 bit

■ Với $n=16$ bit:

biểu diễn từ -32768 đến +32767

■ 0000 0000 0000 0000 = 0

■ 0000 0000 0000 0001 = +1

■ ...

■ 0111 1111 1111 1111 = +32767

■ 1000 0000 0000 0000 = -32768

■ ...

■ 1111 1111 1111 1111 = -1

■ Với $n=32$ bit:

biểu diễn từ -2^{31} đến $2^{31}-1$

■ Với $n=64$ bit:

biểu diễn từ -2^{63} đến $2^{63}-1$

14 January 2013

Kiến trúc máy tính

166

NKK-HUST

Chuyển đổi từ 8 bit thành 16 bit

■ Đối với số dương:

+19 = 0001 0011 (8bit)

+19 = 0000 0000 0001 0011 (16bit)

→ thêm 8 bit 0 bên trái

■ Đối với số âm:

- 19 = 1110 1101 (8bit)

- 19 = 1111 1111 1110 1101 (16bit)

→ thêm 8 bit 1 bên trái

14 January 2013

Kiến trúc máy tính

167

NKK-HUST

4.2. Thực hiện phép cộng/trừ với số nguyên

1. Phép cộng số nguyên không dấu

Bộ cộng n-bit

14 January 2013

Kiến trúc máy tính

168

NKK-HUST

Nguyên tắc cộng số nguyên không dấu

Khi cộng hai số nguyên không dấu n-bit, kết quả nhận được là n-bit:

- Nếu $C_{out}=0 \rightarrow$ nhận được **kết quả đúng**.
- Nếu $C_{out}=1 \rightarrow$ nhận được **kết quả sai**, do **tràn nhớ ra ngoài (Carry Out)**.
- Tràn nhớ ra ngoài khi: **tổng $> (2^n - 1)$**

14 January 2013 Kiến trúc máy tính 169

NKK-HUST

Ví dụ cộng số nguyên không dấu

- $$\begin{array}{r} 57 \\ + 34 \\ \hline 91 \end{array} = \begin{array}{r} 0011\ 1001 \\ + 0010\ 0010 \\ \hline 0101\ 1011 \end{array} = 64+16+8+2+1=91 \rightarrow \text{đúng}$$
- $$\begin{array}{r} 209 \\ + 73 \\ \hline 282 \end{array} = \begin{array}{r} 1101\ 0001 \\ + 0100\ 1001 \\ \hline 0001\ 1010 \end{array}$$

$$0001\ 1010 = 16+8+2=26 \rightarrow \text{sai}$$

\rightarrow có tràn nhớ ra ngoài ($C_{out}=1$)

Để có kết quả đúng ta thực hiện cộng theo 16-bit:

$$\begin{array}{r} 209 = 0000\ 0000\ 1101\ 0001 \\ + 73 = 0000\ 0000\ 0100\ 1001 \\ \hline 0000\ 0001\ 0001\ 1010 = 256+16+8+2 = 282 \end{array}$$

14 January 2013 Kiến trúc máy tính 170

NKK-HUST

2. Phép đảo dấu

- Ta có:

$$\begin{array}{r} + 37 = 0010\ 0101 \\ \text{bù một} = 1101\ 1010 \\ + \underline{\quad 1 \quad} \\ \text{bù hai} = 1101\ 1011 = -37 \end{array}$$
- Lấy bù hai của số âm:

$$\begin{array}{r} - 37 = 1101\ 1011 \\ \text{bù một} = 0010\ 0100 \\ + \underline{\quad 1 \quad} \\ \text{bù hai} = 0010\ 0101 = +37 \end{array}$$
- Kết luận: Phép đảo dấu số nguyên trong máy tính thực chất là lấy bù hai

14 January 2013 Kiến trúc máy tính 171

NKK-HUST

3. Cộng số nguyên có dấu

Khi cộng hai số nguyên có dấu n-bit, kết quả nhận được là n-bit và **không cần quan tâm đến bit C_{out}**

- Cộng hai số khác dấu: **kết quả luôn luôn đúng**.
- Cộng hai số cùng dấu:
 - nếu dấu kết quả cùng dấu với các số hạng thì **kết quả là đúng**.
 - nếu kết quả có dấu ngược lại, khi đó có **tràn xảy ra (Overflow)** và **kết quả bị sai**.
- Tràn xảy ra khi tổng nằm ngoài dải biểu diễn:

$$[-(2^{n-1}), +(2^{n-1}-1)]$$

14 January 2013 Kiến trúc máy tính 172

NKK-HUST

Ví dụ cộng số nguyên có dấu không tràn

■

(+ 70)

=

0100 0110

+

(+ 42)

=

0010 1010

+ 112

0111 0000

= +112

■

(+ 97)

=

0110 0001

+

(- 52)

=

1100 1100

+ 45

1 0010 1101

= +45

(+52=0011 0100)

■

(- 90)

=

1010 0110

+

(+36)

=

0010 0100

- 54

1100 1010

= - 54

(+90=0101 1010)

■

(- 74)

=

1011 0110

+

(- 30)

=

1110 0010

-104

1 1001 1000

= -104

(+74=0100 1010)

(+30=0001 1110)

14 January 2013

Kiến trúc máy tính

173

NKK-HUST

Ví dụ cộng số nguyên có dấu bị tràn

■

(+ 75)

=

0100 1011

+

(+ 82)

=

0101 0010

+157

1001 1101

= - 128+16+8+4+1= -99 → sai

■

(- 104)

=

1001 1000

+

(- 43)

=

1101 0101

- 147

1 0110 1101

= 64+32+8+4+1= +109 → sai

(+104=0110 1000)

(+ 43 =0010 1011)

■

Cả hai ví dụ đều tràn vì tổng nằm ngoài dải biểu diễn [-128, +127]

14 January 2013

Kiến trúc máy tính

174

NKK-HUST

4. Nguyên tắc thực hiện phép trừ

■

Phép trừ hai số nguyên: $X - Y = X + (-Y)$

■

Nguyên tắc: Lấy bù hai của Y để được $-Y$, rồi cộng với X

n-bit

Y

Bù hai

n-bit

X

Bộ cộng n-bit

n-bit

S = X - Y

14 January 2013

Kiến trúc máy tính

175

NKK-HUST

4.3. Phép nhân và phép chia số nguyên

1. Nhân số nguyên không dấu

1011

x 1101

1011

0000

1011

1011

10001111

Số bị nhân (11)

Số nhân (13)

Các tích riêng phần

Tích (143)

14 January 2013

Kiến trúc máy tính

176

Nguyễn Kim Khánh - ĐHBKHN

NKK-HUST

Nhân số nguyên không dấu (tiếp)

- Các **tích riêng phần** được xác định như sau:
 - Nếu bit của số nhân bằng 0 → tích riêng phần bằng 0.
 - Nếu bit của số nhân bằng 1 → tích riêng phần bằng số bị nhân.
 - Tích riêng phần tiếp theo được dịch trái một bit so với tích riêng phần trước đó.
- Tích bằng tổng các **tích riêng phần**
- Nhân hai số nguyên n-bit, tích có độ dài 2n bit (không bao giờ tràn).

14 January 2013

Kiến trúc máy tính

177

NKK-HUST

Bộ nhân số nguyên không dấu

14 January 2013

Kiến trúc máy tính

178

NKK-HUST

Lưu đồ nhân số nguyên không dấu

14 January 2013

Kiến trúc máy tính

179

NKK-HUST

Ví dụ nhân số nguyên không dấu

- Số bị nhân $M = 1011$ (11)
- Số nhân $Q = 1101$ (13)
- Tích $= 1000\ 1111$ (143)

	C	A	Q	
	0	0000	1101	Các giá trị khởi đầu
		+ 1011		
	0	1011	1101	$A \leftarrow A + M$
	0	0101	1110	Dịch phải
	0	0010	1111	Dịch phải
		+ 1011		
	0	1101	1111	$A \leftarrow A + M$
	0	0110	1111	Dịch phải
		+ 1011		
	1	0001	1111	$A \leftarrow A + M$
	0	1000	1111	Dịch phải

14 January 2013

Kiến trúc máy tính

180

NKK-HUST

Ví dụ nhân số nguyên không dấu

- Số bị nhân M = 0010 (2)
- Số nhân Q = 0011 (3)
- Tích = 0000 0110 (6)

	C	A	Q	
■	0	0000	0011	Các giá trị khởi đầu
		+ 0010		
	0	0010	0011	A ← A + M
■	0	0001	0001	Dịch phải
		+ 0010		
	0	0011	0001	
■	0	0001	1000	Dịch phải
■	0	0000	1100	Dịch phải
■	0	0000	0110	Dịch phải

14 January 2013

Kiến trúc máy tính

181

NKK-HUST

2. Nhân số nguyên có dấu

- Sử dụng thuật giải nhân không dấu
- Sử dụng thuật giải Booth

14 January 2013

Kiến trúc máy tính

182

NKK-HUST

Sử dụng thuật giải nhân không dấu

- Bước 1. Chuyển đổi số bị nhân và số nhân thành số dương tương ứng
- Bước 2. Nhân hai số dương bằng thuật giải nhân số nguyên không dấu, được tích của hai số dương.
- Bước 3. Hiệu chỉnh dấu của tích:
 - Nếu hai thừa số ban đầu cùng dấu thì giữ nguyên kết quả ở bước 2.
 - Nếu hai thừa số ban đầu là khác dấu thì đảo dấu kết quả của bước 2 (lấy bù hai).

14 January 2013

Kiến trúc máy tính

183

NKK-HUST

Thuật giải Booth (xem sách COA-[1])

14 January 2013

Kiến trúc máy tính

184

NKK-HUST

3. Chia số nguyên không dấu

Số bị chia

10010011

1011

001110

1011

001111

1011

100

1011

00001101

Số chia

Thương

Phần dư

14 January 2013

Kiến trúc máy tính

185

NKK-HUST

Bộ chia số nguyên không dấu

Số chia M

M_{n-1}

M_{n-2}

...

M₁

M₀

Bộ cộng/trừ n-bit

Điều khiển cộng/trừ

Bộ logic điều khiển cộng/trừ và dịch

Điều khiển dịch trái

A_{n-1}

A_{n-2}

...

A₁

A₀

Q_{n-1}

Q_{n-2}

...

Q₁

Q₀

Số bị chia Q

14 January 2013

Kiến trúc máy tính

186

NKK-HUST

Lưu đồ chia số nguyên không dấu

BẮT ĐẦU

A ← 0
M ← Số chia
Q ← Số bị chia
Bộ đếm ← n

Dịch trái A, Q

A ← A - M

No

Yes

A < 0 ?

Q₀ ← 1

Q₀ ← 0
A ← A + M

Bộ đếm ← Bộ đếm - 1

No

Yes

Bộ đếm = 0 ?

KẾT THÚC

Thương ở Q
Số dư ở A

14 January 2013

Kiến trúc máy tính

187

NKK-HUST

4. Chia số nguyên có dấu

Bước 1. Chuyển đổi số bị chia và số chia về thành số dương tương ứng.

Bước 2. Sử dụng thuật giải chia số nguyên không dấu để chia hai số dương, kết quả nhận được là thương Q và phần dư R đều là dương

Bước 3. Hiệu chỉnh dấu của kết quả như sau:
(Lưu ý: phép đảo dấu thực chất là thực hiện phép lấy bù hai)

Số bị chia	Số chia	Thương	Số dư
dương	dương	giữ nguyên	giữ nguyên
dương	âm	đảo dấu	giữ nguyên
âm	dương	đảo dấu	đảo dấu
âm	âm	giữ nguyên	đảo dấu

14 January 2013

Kiến trúc máy tính

188

NKK-HUST

4.4. Số dấu phẩy động

1. Nguyên tắc chung

- Floating Point Number → biểu diễn cho số thực
- Tổng quát: một số thực X được biểu diễn theo kiểu số dấu phẩy động như sau:
$$X = M * R^E$$
 - M là phần định trị (Mantissa),
 - R là cơ số (Radix),
 - E là phần mũ (Exponent).

14 January 2013

Kiến trúc máy tính

189

NKK-HUST

2. Chuẩn IEEE754/85

- Cơ số R = 2
- Các dạng:
 - Dạng 32-bit
 - Dạng 44-bit
 - Dạng 64-bit
 - Dạng 80-bit

14 January 2013

Kiến trúc máy tính

190

NKK-HUST

Các dạng biểu diễn chính

31 30 23 22 0

S	e	m
---	---	---

63 62 52 51 0

S	e	m
---	---	---

79 78 64 63 0

S	e	m
---	---	---

14 January 2013

Kiến trúc máy tính

191

NKK-HUST

Dạng 32 bit

31 30 23 22 0

S	e	m
---	---	---

- S là bit dấu:
 - S = 0 → số dương
 - S = 1 → số âm
- e (8 bit) là mã *excess-127* của phần mũ E:
 - $e = E + 127 \rightarrow E = e - 127$
 - giá trị 127 gọi là độ lệch (bias)
- m (23 bit) là phần lẻ của phần định trị M:
 - $M = 1.m$
- Công thức xác định giá trị của số thực:
$$X = (-1)^S * 1.m * 2^{e-127}$$

14 January 2013

Kiến trúc máy tính

192

NKK-HUST

Ví dụ 1

Xác định giá trị của số thực được biểu diễn bằng 32-bit như sau:

- 1100 0001 0101 0110 0000 0000 0000 0000
 - S = 1 → số âm
 - e = 1000 0010₂ = 130 → E = 130 - 127 = 3

Vậy

$$X = -1.10101100 \cdot 2^3 = -1101.011 = -13.375$$

- 0011 1111 1000 0000 0000 0000 0000 0000 = ?
= +1.0

14 January 2013 Kiến trúc máy tính 193

NKK-HUST

Ví dụ 2

Biểu diễn số thực X = 83.75 về dạng số dấu phẩy động IEEE754 32-bit

Giải:

- X = 83.75₍₁₀₎ = 1010011.11₍₂₎ = 1.01001111 × 2⁶
- Ta có:
 - S = 0 vì đây là số dương
 - E = e - 127 = 6 → e = 127 + 6 = 133₍₁₀₎ = 1000 0101₍₂₎
- Vậy:

$$X = 0100 0010 1010 0111 1000 0000 0000 0000$$

14 January 2013 Kiến trúc máy tính 194

NKK-HUST

Ví dụ 3

Biểu diễn số thực X = -0,2 về dạng số dấu phẩy động IEEE754 32-bit

Giải:

- X = -0,2₍₁₀₎ = -0.00110011...0011...₍₂₎ = -1.100110011...0011... × 2⁻³
- Ta có:
 - S = 1 vì đây là số âm
 - E = e - 127 = -3 → e = 127 - 3 = 124₍₁₀₎ = 0111 1100₍₂₎
- Vậy:

$$X = 1011 1110 0100 1100 1100 1100 1100 1100$$

14 January 2013 Kiến trúc máy tính 195

NKK-HUST

Bài tập

Biểu diễn các số thực sau đây về dạng số dấu phẩy động IEEE754 32-bit:

$$X = -27.0625; \quad Y = 1/32$$

14 January 2013 Kiến trúc máy tính 196

NKK-HUST

Các quy ước đặc biệt

- Các bit của e bằng 0, các bit của m bằng 0, thì $X = \pm 0$
 $\times 000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000 \rightarrow X = \pm 0$
- Các bit của e bằng 1, các bit của m bằng 0, thì $X = \pm \infty$
 $\times 111\ 1111\ 1000\ 0000\ 0000\ 0000\ 0000 \rightarrow X = \pm \infty$
- Các bit của e bằng 1, còn m có ít nhất một bit bằng 1, thì nó không biểu diễn cho số nào cả (NaN - not a number)

14 January 2013 Kiến trúc máy tính 197

NKK-HUST

Dải giá trị biểu diễn

- 2^{-127} đến 2^{+127}
- 10^{-38} đến 10^{+38}

14 January 2013 Kiến trúc máy tính 198

NKK-HUST

Dạng 64-bit

- S là bit dấu
- e (11 bit): mã *excess-1023* của phần mũ E $\rightarrow E = e - 1023$
- m (52 bit): phần lẻ của phần định trị M
- Giá trị số thực:

$$X = (-1)^S \cdot 1.m \cdot 2^{e-1023}$$
- Dải giá trị biểu diễn: 10^{-308} đến 10^{+308}

14 January 2013 Kiến trúc máy tính 199

NKK-HUST

Dạng 80-bit

- S là bit dấu
- e (15 bit): mã *excess-16383* của phần mũ E $\rightarrow E = e - 16383$
- m (64 bit): phần lẻ của phần định trị M
- Giá trị số thực:

$$X = (-1)^S \cdot 1.m \cdot 2^{e-16383}$$
- Dải giá trị biểu diễn: 10^{-4932} đến 10^{+4932}

14 January 2013 Kiến trúc máy tính 200

201

202

203

204

NKK-HUST

Thuật toán nhân số dấu phẩy động

```
graph TD
    Start([NHÂN]) --> X0{X = 0?}
    X0 -- Y --> Z0[Z ← 0]
    Z0 --> End([TRỞ VỀ])
    X0 -- N --> Y0{Y = 0?}
    Y0 -- Y --> AddExp[Cộng phần mũ]
    Y0 -- N --> SubBias[Trừ cho độ lệch]
    AddExp --> SubBias
    SubBias --> Overflow{Tràn trên phần mũ?}
    Overflow -- Y --> ReportOverflow[Thông báo tràn trên]
    ReportOverflow --> End
    Overflow -- N --> Underflow{Tràn dưới phần mũ?}
    Underflow -- Y --> ReportUnderflow[Thông báo tràn dưới]
    ReportUnderflow --> End
    Underflow -- N --> MultMant[Nhân phần định trị]
    MultMant --> Norm[Chuẩn hoá]
    Norm --> Round[Làm tròn]
    Round --> End
```

14 January 2013

Kiến trúc máy tính

205

NKK-HUST

Thuật toán chia số dấu phẩy động

```
graph TD
    Start([CHIA]) --> X0{X = 0?}
    X0 -- Y --> Z0[Z ← 0]
    Z0 --> End([TRỞ VỀ])
    X0 -- N --> Y0{Y = 0?}
    Y0 -- Y --> SubExp[Trừ phần mũ]
    Y0 -- N --> ZInf[Z ← ∞]
    SubExp --> AddBias[Cộng thêm độ lệch]
    AddBias --> Overflow{Tràn trên phần mũ?}
    Overflow -- Y --> ReportOverflow[Thông báo tràn trên]
    ReportOverflow --> End
    Overflow -- N --> Underflow{Tràn dưới phần mũ?}
    Underflow -- Y --> ReportUnderflow[Thông báo tràn dưới]
    ReportUnderflow --> End
    Underflow -- N --> DivMant[Chia phần định trị]
    DivMant --> Norm[Chuẩn hoá]
    Norm --> Round[Làm tròn]
    Round --> End
```

14 January 2013

Kiến trúc máy tính

206

NKK-HUST

Hết chương 4

14 January 2013

Kiến trúc máy tính

207

NKK-HUST

Kiến trúc máy tính

Chương 5

KIẾN TRÚC TẬP LỆNH

(Instruction Set Architecture)

Nguyễn Kim Khánh

Trường Đại học Bách khoa Hà Nội

14 January 2013

Kiến trúc máy tính

208

NKK-HUST

Nội dung học phần

- Chương 1. Giới thiệu chung
- Chương 2. Cơ bản về logic số
- Chương 3. Hệ thống máy tính
- Chương 4. Số học máy tính
- Chương 5. Kiến trúc tập lệnh**
- Chương 6. Cấu trúc bộ xử lý
- Chương 7. Bộ nhớ
- Chương 8. Vào-ra
- Chương 9. Các kiến trúc song song

14 January 2013

Kiến trúc máy tính

209

NKK-HUST

Nội dung của chương 5

5.1. Giới thiệu chung về kiến trúc tập lệnh

5.2. Kiến trúc tập lệnh MIPS

5.3. Kiến trúc tập lệnh Intel x86 *

14 January 2013

Kiến trúc máy tính

210

NKK-HUST

5.1. Giới thiệu chung về kiến trúc tập lệnh

1. Mô hình lập trình của máy tính

The diagram illustrates the computer architecture model. On the left, the CPU is shown with components: PC (Program Counter), IR (Instruction Register), Đơn vị điều khiển (Control Unit), ALU (Arithmetic Logic Unit), and Tập thanh ghi (Registers). In the center, the Bộ nhớ chính (Main Memory) is shown with a stack of cells, some labeled 'lệnh' (instructions) and others 'dữ liệu' (data). On the right, the Vào-ra (I/O) component is shown. Arrows indicate the flow of information: from Main Memory to CPU, from CPU to Main Memory, and from CPU to I/O.

PC: Program Counter
IR: Instruction Register

14 January 2013

Kiến trúc máy tính

211

NKK-HUST

Tập thanh ghi

- Chứa các thông tin (dữ liệu, địa chỉ, trạng thái) cho hoạt động điều khiển và xử lý dữ liệu của CPU ở thời điểm hiện tại
- Được coi là mức đầu tiên của hệ thống nhớ
- Số lượng thanh ghi nhiều → tăng hiệu năng của CPU
- Có hai loại thanh ghi:
 - Các thanh ghi lập trình được
 - Các thanh ghi không lập trình được

14 January 2013

Kiến trúc máy tính

212

NKK-HUST

Một số thanh ghi điển hình

- Bộ đếm chương trình PC (Program Counter)
- Con trỏ dữ liệu DP (Data Pointer)
- Con trỏ ngăn xếp SP (Stack Pointer)
- Thanh ghi cơ sở và Thanh ghi chỉ số (Base Register & Index Register)
- Các thanh ghi dữ liệu
- Thanh ghi trạng thái

14 January 2013

Kiến trúc máy tính

213

NKK-HUST

Bộ đếm chương trình PC

- Còn được gọi là con trỏ lệnh IP (Instruction Pointer)
- Giữ địa chỉ của lệnh tiếp theo sẽ được nhận vào.
- Sau khi một lệnh được nhận vào, nội dung PC tự động tăng để trở sang lệnh kế tiếp.
- PC tăng bao nhiêu?

The diagram illustrates the flow of instructions. A box labeled 'PC' has an arrow pointing to the 'lệnh kế tiếp' (next instruction) in a vertical stack of instructions. The stack contains: 'lệnh', 'lệnh', 'lệnh', 'lệnh sẽ được nhận', 'lệnh kế tiếp', 'lệnh', and 'lệnh'.

14 January 2013

Kiến trúc máy tính

214

NKK-HUST

Thanh ghi con trỏ dữ liệu

- Chứa địa chỉ của ngăn nhớ dữ liệu mà CPU muốn truy nhập

The diagram shows a box labeled 'DP' with an arrow pointing to the 'dữ liệu cần đọc/ghi' (data to be read/written) in a vertical stack of memory locations. The stack contains: 'dữ liệu', 'dữ liệu', 'dữ liệu', 'dữ liệu cần đọc/ghi', 'dữ liệu', 'dữ liệu', and 'dữ liệu'.

14 January 2013

Kiến trúc máy tính

215

NKK-HUST

Ngăn xếp (Stack)

- Ngăn xếp là vùng nhớ có cấu trúc LIFO (Last In - First Out → vào sau – ra trước)
- Ngăn xếp thường dùng để phục vụ cho chương trình con
- Đáy ngăn xếp là một ngăn nhớ xác định
- Đỉnh ngăn xếp là thông tin nằm ở vị trí trên cùng trong ngăn xếp
- Đỉnh ngăn xếp có thể bị thay đổi

14 January 2013

Kiến trúc máy tính

216

NKK-HUST

Con trỏ ngăn xếp SP (Stack Pointer)

- Chứa địa chỉ của ngăn nhớ đỉnh ngăn xếp
- Khi cất một thông tin vào ngăn xếp:
 - Nội dung của SP giảm
 - Thông tin được cất vào ngăn nhớ được trỏ bởi SP
- Khi lấy một thông tin ra khỏi ngăn xếp:
 - Thông tin được đọc từ ngăn nhớ được trỏ bởi SP
 - Nội dung của SP tăng
- Khi ngăn xếp rỗng, SP trở vào đáy

14 January 2013

Kiến trúc máy tính

217

NKK-HUST

Thanh ghi cơ sở và thanh ghi chỉ số

- Để truy nhập một ngăn nhớ có thể sử dụng hai tham số:
 - Địa chỉ cơ sở (base address)
 - Phần dịch chuyển địa chỉ (offset)
- Địa chỉ của ngăn nhớ cần truy nhập = địa chỉ cơ sở + offset
- Có thể sử dụng các thanh ghi để quản lý các tham số này:
 - Thanh ghi cơ sở: chứa địa chỉ cơ sở
 - Thanh ghi chỉ số: chứa phần dịch chuyển địa chỉ

14 January 2013

Kiến trúc máy tính

218

NKK-HUST

Các thanh ghi dữ liệu

- Chứa các dữ liệu tạm thời hoặc các kết quả trung gian
- Cần có nhiều thanh ghi dữ liệu
- Các thanh ghi số nguyên: 8, 16, 32, 64 bit
- Các thanh ghi số dấu phẩy động

14 January 2013

Kiến trúc máy tính

219

NKK-HUST

Thanh ghi trạng thái (Status Register)


- Được sử dụng trên một số kiến trúc cụ thể
- Còn gọi là thanh ghi cờ (Flag Register)
- Chứa các thông tin trạng thái của CPU
 - Các cờ phép toán: báo hiệu trạng thái của kết quả phép toán
 - Các cờ điều khiển: biểu thị trạng thái điều khiển của CPU

14 January 2013

Kiến trúc máy tính

220

NKK-HUST



2. Thứ tự lưu trữ các byte trong bộ nhớ chính


- Bộ nhớ chính thường đánh địa chỉ theo byte
- Hai cách lưu trữ thông tin nhiều byte:
 - **Đầu nhỏ** (*Little-endian*): Byte có ý nghĩa thấp được lưu trữ ở ngăn nhớ có địa chỉ nhỏ, byte có ý nghĩa cao được lưu trữ ở ngăn nhớ có địa chỉ lớn.
 - **Đầu to** (*Big-endian*): Byte có ý nghĩa cao được lưu trữ ở ngăn nhớ có địa chỉ nhỏ, byte có ý nghĩa thấp được lưu trữ ở ngăn nhớ có địa chỉ lớn.

14 January 2013

Kiến trúc máy tính

221

NKK-HUST



Ví dụ lưu trữ dữ liệu 32-bit

0001 1010 0010 1011 0011 1100 0100 1101

1A	2B	3C	4D
----	----	----	----

	4D	3000
	3C	3001
	2B	3002
	1A	3003

little-endian

1A	3000
2B	3001
3C	3002
4D	3003


big-endian

14 January 2013

Kiến trúc máy tính

222

NKK-HUST



Lưu trữ của các bộ xử lý điển hình


- Intel x86: little-endian
- Motorola 680x0, MIPS, SunSPARC: big-endian
- Power PC, Itanium: bi-endian

14 January 2013

Kiến trúc máy tính

223

NKK-HUST



3. Giới thiệu chung về tập lệnh

- Mỗi bộ xử lý có một tập lệnh xác định
- Tập lệnh thường có hàng chục đến hàng trăm lệnh
- Mỗi lệnh là một chuỗi số nhị phân mà bộ xử lý hiểu được để thực hiện một thao tác xác định.
- Các lệnh được mô tả bằng các ký hiệu gọi nhớ dạng text → chính là các lệnh của hợp ngữ (assembly language)

14 January 2013

Kiến trúc máy tính

224

NKK-HUST

Các thành phần của lệnh máy

Mã thao tác	Địa chỉ của các toán hạng
-------------	---------------------------

- Mã thao tác (operation code → opcode): mã hóa cho thao tác mà bộ xử lý phải thực hiện
- Địa chỉ toán hạng: chỉ ra nơi chứa các toán hạng mà thao tác sẽ tác động
 - Toán hạng nguồn (source operand): dữ liệu vào của thao tác
 - Toán hạng đích (destination operand): dữ liệu ra của thao tác

14 January 2013 Kiến trúc máy tính 225

NKK-HUST

Các kiểu thao tác thông dụng của tập lệnh

- Các lệnh chuyển dữ liệu
- Các lệnh xử lý số học
- Các lệnh xử lý logic
- Các lệnh chuyển điều khiển (rẽ nhánh, nhảy)

14 January 2013 Kiến trúc máy tính 226

NKK-HUST

Định địa chỉ toán hạng

- 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

14 January 2013 Kiến trúc máy tính 227

NKK-HUST

Các phương pháp định địa chỉ thông dụng

- Định địa chỉ tức thì
- Định địa chỉ thanh ghi
- Định địa chỉ trực tiếp
- Định địa chỉ gián tiếp qua thanh ghi
- Định địa chỉ dịch chuyển

14 January 2013 Kiến trúc máy tính 228

229

230

231

232

NKK-HUST

Định địa chỉ dịch chuyển

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

14 January 2013 Kiến trúc máy tính 233

NKK-HUST

Số lượng địa chỉ toán hạng trong lệnh (1)

- Ba địa chỉ toán hạng:
 - 2 toán hạng nguồn, 1 toán hạng đích
 - $c = a + b$
 - Từ lệnh dài vì phải mã hoá địa chỉ cho cả ba toán hạng
 - Được sử dụng trên các bộ xử lý tiên tiến

14 January 2013 Kiến trúc máy tính 234

NKK-HUST

Số lượng địa chỉ toán hạng trong lệnh (2)

- Hai địa chỉ toán hạng:
 - Một toán hạng vừa là toán hạng nguồn vừa là toán hạng đích; toán hạng còn lại là toán hạng nguồn
 - $a = a + b$
 - Giá trị cũ của 1 toán hạng nguồn bị mất vì phải chứa kết quả
 - Rút gọn độ dài từ lệnh
 - Phổ biến

14 January 2013 Kiến trúc máy tính 235

NKK-HUST

Số lượng địa chỉ toán hạng trong lệnh (3)

- Một địa chỉ toán hạng:
 - Một toán hạng được chỉ ra trong lệnh
 - Một toán hạng là ngầm định → thường là thanh ghi (thanh chứa –accumulator)
 - Được sử dụng trên các máy ở các thế hệ trước

14 January 2013 Kiến trúc máy tính 236

NKK-HUST

Số lượng địa chỉ toán hạng trong lệnh (4)

- 0 địa chỉ toán hạng:
 - Các toán hạng đều được ngầm định
 - Sử dụng Stack
 - Ví dụ:
 - push a
 - push b
 - add
 - pop c
 - có nghĩa là : $c = a + b$
 - không thông dụng

14 January 2013 Kiến trúc máy tính 237

NKK-HUST

4. CISC và RISC

- CISC: Complex Instruction Set Computer:
 - Máy tính với tập lệnh phức tạp
 - Các bộ xử lý truyền thống: Intel x86, Motorola 680x0
- RISC: Reduced Instruction Set Computer:
 - Máy tính với tập lệnh thu gọn
 - SunSPARC, Power PC, MIPS, ARM ...
 - RISC đối nghịch với CISC
 - Kiến trúc tập lệnh tiên tiến

14 January 2013 Kiến trúc máy tính 238

NKK-HUST

Các đặc trưng của RISC

- Số lượng lệnh ít
- Hầu hết các lệnh truy nhập toán hạng ở các thanh ghi
- Truy nhập bộ nhớ bằng các lệnh LOAD/STORE
- Thời gian thực hiện lệnh là một chu kỳ máy
- Các lệnh có độ dài cố định (32 bit)
- Số lượng dạng lệnh ít (≤ 4)
- CPU có tập thanh ghi lớn
- Có ít phương pháp định địa chỉ toán hạng (≤ 4)
- Hỗ trợ các thao tác của ngôn ngữ bậc cao

14 January 2013 Kiến trúc máy tính 239

NKK-HUST

5.2. Kiến trúc tập lệnh MIPS

14 January 2013 Kiến trúc máy tính 240

NKK-HUST

1. Giới thiệu chung

- MIPS- Microprocessor without Interlocked Pipeline Stages
- Được phát triển ở đại học Stanford, sau đó được thương mại hóa bởi Công ty MIPS Technologies (www.mips.com)
- Kiến trúc RISC
- Chiếm thị phần lớn trong các sản phẩm nhúng
- Điển hình cho nhiều kiến trúc tập lệnh hiện đại

14 January 2013 Kiến trúc máy tính 241

NKK-HUST

2. Phép toán số học và các toán hạng

- Cộng và trừ: 3 toán hạng
 - Hai toán hạng nguồn và một toán hạng đích
$$\text{add } a, b, c \quad \# \quad a \leftarrow b + c$$
- Tất cả các lệnh số học có dạng trên
- Toán hạng có thể là:
 - Nội dung thanh ghi
 - Nội dung ngăn nhớ
 - Hằng số

14 January 2013 Kiến trúc máy tính 242

NKK-HUST

Toán hạng thanh ghi

- Các lệnh số học sử dụng toán hạng thanh ghi
- MIPS có tập 32 thanh ghi 32-bit
 - Được sử dụng thường xuyên
 - Được đánh số từ 0 đến 31 (dùng 5 bit)
 - Dữ liệu 32-bit được gọi là “word”
- Chương trình dịch Assembler đặt tên:
 - \$t0, \$t1, ..., \$t9 chứa các giá trị tạm thời
 - \$s0, \$s1, ..., \$s7 cất các biến

14 January 2013 Kiến trúc máy tính 243

NKK-HUST

Tập thanh ghi của MIPS

Tên thanh ghi	Số hiệu thanh ghi	Công dụng
\$zero	0	the constant value 0
\$at	1	assembler temporary
\$v0-\$v1	2-3	procedure return values
\$a0-\$a3	4-7	procedure arguments
\$t0-\$t7	8-15	temporaries
\$s0-\$s7	16-23	saved variables
\$t8-\$t9	24-25	more temporaries
\$k0-\$k1	26-27	OS temporaries
\$gp	28	global pointer
\$sp	29	stack pointer
\$fp	30	frame pointer
\$ra	31	procedure return address

14 January 2013 Kiến trúc máy tính 244

245

246

247

248

NKK-HUST

Thanh ghi với Bộ nhớ

- Truy nhập thanh ghi nhanh hơn bộ nhớ
- Thao tác dữ liệu trên bộ nhớ yêu cầu nạp (load) và lưu (store).
 - Cần thực hiện nhiều lệnh hơn
- Chương trình dịch sử dụng các thanh ghi cho các biến nhiều nhất có thể
 - Chỉ sử dụng bộ nhớ cho các biến ít được sử dụng
 - Cần tối ưu hóa sử dụng thanh ghi

14 January 2013 Kiến trúc máy tính 249

NKK-HUST

Toán hạng tức thì (immediate)

- Dữ liệu hằng số được xác định ngay trong lệnh


```
addi $s3, $s3, 4    # $s3 ← $s3+4
```
- Không có lệnh trừ (subi) với giá trị tức thì
 - Sử dụng hằng số âm để thực hiện phép trừ


```
addi $s2, $s1, -1    # $s2 ← $s1-1
```

14 January 2013 Kiến trúc máy tính 250

NKK-HUST

Hằng số Zero

- Thanh ghi 0 của MIPS (\$zero hay \$0) luôn chứa hằng số 0
 - Không thể thay đổi giá trị
- Hữu ích cho một số thao tác thông dụng
 - Chẳng hạn, chuyển dữ liệu giữa các thanh ghi


```
add $t2, $s1, $zero # $t2 ← $s1
```

14 January 2013 Kiến trúc máy tính 251

NKK-HUST

3. Mã máy

- Các lệnh được mã hóa dưới dạng nhị phân được gọi là mã máy
- Các lệnh của MIPS:
 - Được mã hóa bằng các từ lệnh 32-bit
 - Có ít dạng lệnh
- Số hiệu thanh ghi
 - \$t0 – \$t7 là các thanh ghi 8 – 15
 - \$t8 – \$t9 là các thanh ghi 24 – 25
 - \$s0 – \$s7 là các thanh ghi 16 – 23

14 January 2013 Kiến trúc máy tính 252

NKK-HUST

Các dạng lệnh của MIPS

R

opcode	rs	rt	rd	shamt	funct
31	26 25	21 20	16 15	11 10	6 5
0					

I

opcode	rs	rt	immediate
31	26 25	21 20	16 15
0			

J

opcode	address
31	26 25
0	

14 January 2013

Kiến trúc máy tính

253

NKK-HUST

Lệnh dạng R (Register)

op	rs	rt	rd	shamt	funct
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

■ Các trường của lệnh

■ op: operation code (opcode): mã thao tác

■ rs: số hiệu thanh ghi nguồn thứ nhất

■ rt: số hiệu thanh ghi nguồn thứ hai

■ rd: số hiệu thanh ghi đích

■ shamt (shift amount): số bit được dịch

■ funct: function code (extends opcode): mã hàm (mã thao tác mở rộng)

14 January 2013

Kiến trúc máy tính

254

NKK-HUST

Ví dụ dạng lệnh R

Hợp ngữ

add \$s0, \$s1, \$s2

sub \$t0, \$t3, \$t5

Giá trị các trường

op	rs	rt	rd	shamt	funct
0	17	18	16	0	32
0	11	13	8	0	34
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

Mã máy

op	rs	rt	rd	shamt	funct
000000	10001	10010	10000	00000	100000
000000	01011	01101	01000	00000	100010
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

(0x02328020)

(0x016D4022)

14 January 2013

Kiến trúc máy tính

255

NKK-HUST

Lệnh dạng I (Immediate)

op	rs	rt	constant or address
6 bits	5 bits	5 bits	16 bits

■ Dùng cho các lệnh số học với toán hạng tức thì và các lệnh load/store (nạp/lưu)

■ rt: số hiệu thanh ghi đích hoặc thanh ghi nguồn

■ Hằng số: từ -2^{15} đến $+2^{15} - 1$

■ Địa chỉ: offset cộng với địa chỉ cơ sở nằm ở rs

■ addi rt, rs, imm # (rt) \leftarrow (rs)+imm

■ lw rt, imm(rs) # (rt) \leftarrow mem[(rs)+imm]

■ sw rt, imm(rs) # (rt) \rightarrow mem[(rs)+imm]

14 January 2013

Kiến trúc máy tính

256

Nguyễn Kim Khánh - ĐHBKHN

Lệnh lui

lui \$s0, 61 # Giá trị tức thì 61 được nạp vào
nửa cao của \$s0 với 16 bit thấp
được thiết lập về 0

Instruction format breakdown:


- op: 00111
- rs: 0000
- rt: 1000
- operand / offset: 0000000000001111

Labels for instruction fields:

- lui = 15 (points to op)
- Unused (points to rs)
- Destination (points to rt)
- Immediate operand (points to operand / offset)

Content of \$s0 after the instruction is executed: 0000000000001111

NKK-HUST



Các lệnh logic

■ Các lệnh logic để thao tác trên các bit

Operation	C	Java	MIPS
Shift left	<<	<<	sll
Shift right	>>	>>>	srl
Bitwise AND	&	&	and, andi
Bitwise OR			or, ori
Bitwise NOT	~	~	nor

14 January 2013

Kiến trúc máy tính

260

NKK-HUST

Ví dụ lệnh logic

Source Registers

\$s1

1111

1111

1111

1111

0000

0000

0000

0000

\$s2

0100

0110

1010

0001

1111

0000

1011

0111

Assembly Code

and \$s3, \$s1, \$s2

or \$s4, \$s1, \$s2

xor \$s5, \$s1, \$s2

nor \$s6, \$s1, \$s2

Result

\$s3

\$s4

\$s5

\$s6

14 January 2013

Kiến trúc máy tính

261

NKK-HUST

Ví dụ lệnh logic

Source Registers

\$s1

1111

1111

1111

1111

0000

0000

0000

0000

\$s2

0100

0110

1010

0001

1111

0000

1011

0111

Assembly Code

and \$s3, \$s1, \$s2

or \$s4, \$s1, \$s2

xor \$s5, \$s1, \$s2

nor \$s6, \$s1, \$s2

Result

\$s3

0100

0110

1010

0001

0000

0000

0000

0000

\$s4

1111

1111

1111

1111

1111

0000

1011

0111

\$s5

1011

1001

0101

1110

1111

0000

1011

0111

\$s6

0000

0000

0000

0000

0000

1111

0100

1000

14 January 2013

Kiến trúc máy tính

262

NKK-HUST

Ví dụ lệnh logic

Source Values

\$s1

0000

0000

0000

0000

0000

0000

1111

1111

imm

0000

0000

0000

0000

1111

1010

0011

0100

zero-extended

Assembly Code

andi \$s2, \$s1, 0xFA34

ori \$s3, \$s1, 0xFA34

xori \$s4, \$s1, 0xFA34

Result

\$s2

\$s3

\$s4

14 January 2013

Kiến trúc máy tính

263

NKK-HUST

Ví dụ lệnh logic

Source Values

\$s1

0000

0000

0000

0000

0000

0000

1111

1111

imm

0000

0000

0000

0000

1111

1010

0011

0100

zero-extended

Assembly Code

andi \$s2, \$s1, 0xFA34

ori \$s3, \$s1, 0xFA34

xori \$s4, \$s1, 0xFA34

Result

\$s2

0000

0000

0000

0000

0000

0000

0011

0100

\$s3

0000

0000

0000

0000

1111

1010

1111

1111

\$s4

0000

0000

0000

0000

1111

1010

1100

1011

14 January 2013

Kiến trúc máy tính

264

NKK-HUST

Ý nghĩa của các phép toán logic

- Phép AND dùng để giữ nguyên một số bit trong word, xóa các bit khác về 0
- Phép OR dùng để giữ nguyên một số bit trong word, thiết lập các bit còn lại lên 1
- Phép XOR dùng để giữ nguyên một số bit trong word, đảo giá trị các bit còn lại
- Phép NOT dùng để đảo các bit trong word
 - Đổi 0 thành 1, và đổi 1 thành 0
 - MIPS không có lệnh NOT, nhưng có lệnh NOR với 3 toán hạng
 - $a \text{ NOR } b == \text{NOT} (a \text{ OR } b)$

nor \$t0, \$t1, \$zero

14 January 2013 Kiến trúc máy tính 265

NKK-HUST

Thao tác dịch bit

op	rs	rt	rd	shamt	funct
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

- shamt: dịch bao nhiêu vị trí
- Dịch trái logic (shift left logical)
 - Dịch trái và điền các bit 0 vào bên phải
 - sl l với i bits là nhân với 2^i
- Dịch phải logic (shift right logical)
 - Dịch phải và điền các bit 0 vào bên trái
 - srl với i bits là chia cho 2^i (chỉ với số nguyên không dấu)

14 January 2013 Kiến trúc máy tính 266

NKK-HUST

Lệnh dịch

- sl l : shift left logical
 - sl l \$t0, \$t1, 5 # \$t0 <= \$t1 << 5
- srl : shift right logical
 - srl \$t0, \$t1, 5 # \$t0 <= \$t1 >> 5
- sra: shift right arithmetic
 - sra \$t0, \$t1, 5 # \$t0 <= \$t1 >>> 5

Variable shift instructions:

- sl l v: shift left logical variable
 - sl l v \$t0, \$t1, \$t2 # \$t0 <= \$t1 << \$t2
- srl v: shift right logical variable
 - srl v \$t0, \$t1, \$t2 # \$t0 <= \$t1 >> \$t2
- srav: shift right arithmetic variable
 - srav \$t0, \$t1, \$t2 # \$t0 <= \$t1 >>> \$t2

14 January 2013 Kiến trúc máy tính 267

NKK-HUST

Ví dụ các lệnh dịch

Assembly Code

```

sll $t0, $s1, 2
srl $s2, $s1, 2
sra $s3, $s1, 2
                    
```

Field Values

op	rs	rt	rd	shamt	funct
0	0	17	8	2	0
0	0	17	18	2	2
0	0	17	19	2	3

6 bits 5 bits 5 bits 5 bits 5 bits 6 bits

Machine Code

op	rs	rt	rd	shamt	funct	
000000	00000	10001	01000	00010	000000	(0x00114080)
000000	00000	10001	10010	00010	000010	(0x00119082)
000000	00000	10001	10011	00010	000011	(0x00119883)

6 bits 5 bits 5 bits 5 bits 5 bits 6 bits

14 January 2013 Kiến trúc máy tính 268

NKK-HUST

Nạp hằng số vào thanh ghi

- Trường hợp hằng số 16-bit → sử dụng lệnh **addi**:
 - Ví dụ: nạp hằng số 0x4f3c vào thanh ghi \$s0:
`addi $s0, $0, 0x4f3c`
- Trong trường hợp hằng số 32-bit → sử dụng lệnh **lui** và lệnh **ori**:
`lui rt, constant_hi 16bit`
 - Copy 16 bit cao của hằng số vào 16 bit trái của rt
 - Xóa 16 bits bên phải của rt về 0`ori rt, rt, constant_low 16bit`
 - Đưa 16 bit thấp của hằng số 32 bit vào thanh ghi rt

14 January 2013 Kiến trúc máy tính 269

NKK-HUST

Ví dụ khởi tạo thanh ghi 32-bit

- Nạp vào các thanh ghi \$s0 và \$s1 các giá trị 32-bit sau:
 0010 0001 0001 0000 0000 0000 0011 1101
 1111 1111 1111 1111 1111 1111 1111 1111
- 0010 0001 0001 0000 0000 0000 0011 1101 = 0x2110003d
`lui $s0, 0x2110`

0010 0001 0001 0000 0000 0000 0000 0000
- `ori $s0, $s0, 0x003d`

0010 0001 0001 0000 0000 0000 0011 1101
- 1111 1111 1111 1111 1111 1111 1111 1111 = 0xffffffff
 Có thể làm tương tự như trên với giá trị này, tuy nhiên có thể thực hiện đơn giản hơn:
`nor $s1, $zero, $zero`

14 January 2013 Kiến trúc máy tính 270

NKK-HUST

4. Tạo các cấu trúc điều khiển

- Câu lệnh **If**
- Câu lệnh **If/else**
- Câu lệnh lặp **While**
- Câu lệnh lặp **For**

14 January 2013 Kiến trúc máy tính 271

NKK-HUST

Các lệnh rẽ nhánh và lệnh nhảy

- Rẽ nhánh đến lệnh được đánh nhãn nếu điều kiện là đúng, ngược lại, thực hiện tuần tự
- `bltz rs, L1`
 - branch on less than zero
 - nếu (rs < 0) rẽ nhánh đến lệnh ở nhãn L1;
- `beq rs, rt, L1`
 - branch on equal
 - nếu (rs == rt) rẽ nhánh đến lệnh ở nhãn L1;
- `bne rs, rt, L1`
 - branch on not equal
 - nếu (rs != rt) rẽ nhánh đến lệnh ở nhãn L1;
- `j L1`
 - nhảy (jump) không điều kiện đến lệnh ở nhãn L1

14 January 2013 Kiến trúc máy tính 272

NKK-HUST

Dịch câu lệnh If

- Mã C:


```
if (i == j)
    f = g+h;
    f = f-i;
```

 - f, g, h, i, j ở \$s0, \$s1, \$s2, \$s3, \$s4
- Mã MIPS:


```
# $s0 = f, $s1 = g, $s2 = h
# $s3 = i, $s4 = j
    bne $s3, $s4, L1
    add $s0, $s1, $s2
L1: sub $s0, $s0, $s3
```

14 January 2013 Kiến trúc máy tính 273

NKK-HUST

Dịch câu lệnh If/else

- Mã C:


```
if (i == j) f = g+h;
else f = g-h;
```

 - f, g, h, i, j ở \$s0, \$s1, \$s2, \$s3, \$s4
- Mã MIPS:


```
    bne $s3, $s4, Else
    add $s0, $s1, $s2
    j Exit
Else: sub $s0, $s1, $s2
Exit: ...
```

Assembler calculates addresses

```

graph TD
    Cond{i == j?} -- true --> TrueBox[f = g + h]
    Cond -- false --> FalseBox[f = g - h]
    TrueBox --> Exit[Exit:]
    FalseBox --> Exit
    
```

14 January 2013 Kiến trúc máy tính 274

NKK-HUST

Dịch câu lệnh vòng lặp While

- Mã C:


```
while (save[i] == k) i += 1;
```

 - i ở \$s3, k ở \$s5, địa chỉ của mảng save ở \$s6
- Mã MIPS được dịch:


```
Loop: slt $t1, $s3, 2      # $t1 = 4*i
        add $t1, $t1, $s6  # $t1 trở tới save[i]
        lw $t0, 0($t1)     # $t0 ← save[i]
        bne $t0, $s5, Exit # nếu save[i] = k
        addi $s3, $s3, 1    # thì i = i+1
        j Loop             # quay lại
Exit: ...                 # nếu save[i] <> k, thoát
```

14 January 2013 Kiến trúc máy tính 275

NKK-HUST

Dịch câu lệnh vòng lặp For

- Mã C:


```
// add the numbers from 0 to 9
int sum = 0;
int i;
for (i=0; i!=10; i = i+1) {
    sum = sum + i;
}
```
- Mã MIPS được dịch:

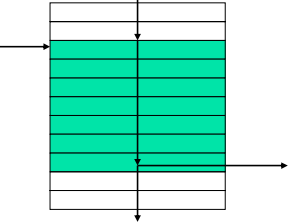

```
# $s0 = i, $s1 = sum
    addi $s1, $0, 0      # sum = 0
    add $s0, $0, $0      # i = 0
    addi $t0, $0, 10     # $t0 = 10
for: beq $s0, $t0, done  # Nếu i = 10, thoát
    add $s1, $s1, $s0    # sum = sum + i
    addi $s0, $s0, 1     # i = i+1
    j for                # quay lại
done:
```

14 January 2013 Kiến trúc máy tính 276

NKK-HUST

Khối lệnh cơ sở

- Khối lệnh cơ sở là dãy các lệnh với
 - Không có lệnh rẽ nhánh nhúng trong đó (ngoại trừ ở cuối)
 - Không có đích rẽ nhánh tới (ngoại trừ ở vị trí đầu tiên)



- Chương trình dịch xác định khối cơ sở để tối ưu hóa
- Các bộ xử lý tiên tiến có thể tăng tốc độ thực hiện khối cơ sở

14 January 2013

Kiến trúc máy tính

277

NKK-HUST

Thêm các thao tác điều kiện

- Thiết lập kết quả = 1 nếu điều kiện là đúng, trái lại kết quả = 0
- `slt rd, rs, rt`
 - set on less than
 - if (rs < rt) rd = 1; else rd = 0;
- `slti rt, rs, constant`
 - if (rs < constant) rt = 1; else rt = 0;
- Sử dụng kết hợp với các lệnh `beq, bne`

`slt $t0, $s1, $s2 # if ($s1 < $s2)`
`bne $t0, $zero, L # branch to L`

14 January 2013

Kiến trúc máy tính

278

NKK-HUST

Ví dụ mã lệnh `slt` và `slti`

- `slt $s1,$s2,$s3` # nếu $\$s2 < (\$s3)$, $\$s1 \leftarrow 1$
- # ngược lại $\$s1 \leftarrow 0$;
- # thường theo sau là `beq/bne`
- `slti $s1,$s2,61` # nếu $\$s2 < 61$, $\$s1 \leftarrow 1$
- # ngược lại $\$s1 \leftarrow 0$

R

31	op	25	rs	20	rt	15	rd	10	sh	5	fn	0
0	0	0	0	0	1	0	0	1	1	0	0	0
ALU instruction		Source 1 register			Source 2 register		Destination		Unused		slt = 42	

I

31	op	25	rs	20	rt	operand / offset														
0	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	1	1	1	0	1
slti = 10		Source			Destination		Immediate operand													

14 January 2013

Kiến trúc máy tính

279

NKK-HUST

So sánh số có dấu và không dấu

- So sánh số có dấu: `slt`, `slti`
- So sánh số không dấu: `sltu`, `sltiu`
- Ví dụ
 - $\$s0 = 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111$
 - $\$s1 = 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001$
 - `slt $t0, $s0, $s1` # signed
 - $-1 < +1 \Rightarrow \$t0 = 1$
 - `sltu $t0, $s0, $s1` # unsigned
 - $+4,294,967,295 > +1 \Rightarrow \$t0 = 0$

14 January 2013

Kiến trúc máy tính

280

Nguyễn Kim Khánh - ĐHBKHN

NKK-HUST

5. Gọi thủ tục (Procedure Calling)

- Các bước yêu cầu:
 - Đặt các tham số vào các thanh ghi
 - Chuyển điều khiển đến thủ tục
 - Thực hiện các thao tác của thủ tục
 - Đặt kết quả vào thanh ghi cho chương trình đã gọi thủ tục
 - Trở về vị trí đã gọi

14 January 2013 Kiến trúc máy tính 281

NKK-HUST

Sử dụng các thanh ghi

- \$a0 – \$a3: các tham số (các thanh ghi 4 – 7)
- \$v0, \$v1: giá trị kết quả (các thanh ghi 2 và 3)
- \$t0 – \$t9: các giá trị tạm thời
 - Có thể được ghi lại bởi thủ tục được gọi
- \$s0 – \$s7: cất giữ các biến
 - Cần phải cất/khôi phục bởi thủ tục được gọi
- \$gp: global pointer - con trỏ toàn cục cho dữ liệu tĩnh (thanh ghi 28)
- \$sp: stack pointer - con trỏ ngăn xếp (thanh ghi 29)
- \$fp: frame pointer – con trỏ khung (thanh ghi 30)
- \$ra: return address – địa chỉ trở về (thanh ghi 31)

14 January 2013 Kiến trúc máy tính 282

NKK-HUST

Các lệnh gọi thủ tục

- Gọi thủ tục: jump and link

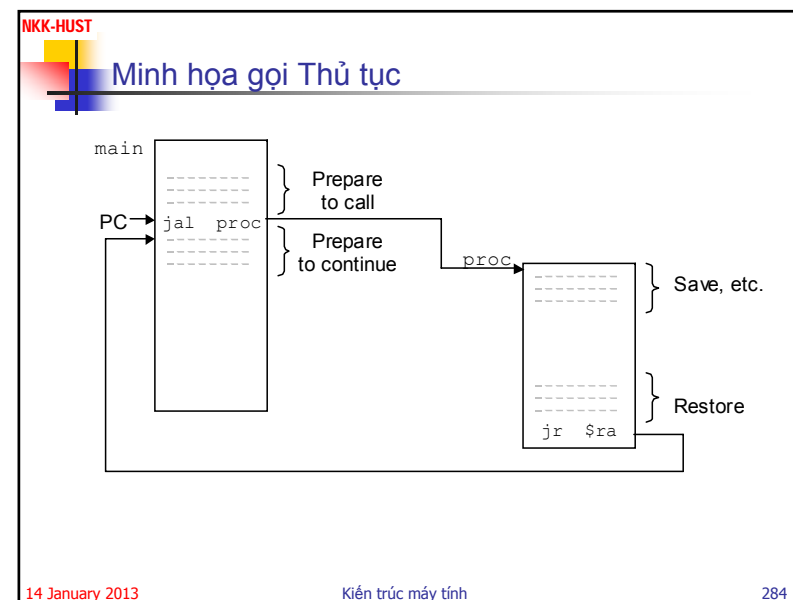

```
jal ProcedureLabel
```

 - Địa chỉ của lệnh kế tiếp được cất ở \$ra
 - Nhảy đến nhãn đích
- Trở về từ thủ tục: jump register


```
jr $ra
```

 - Copy \$ra vào bộ đếm chương trình PC

14 January 2013 Kiến trúc máy tính 283



NKK-HUST

Gọi thủ tục lồng nhau

14 January 2013

Kiến trúc máy tính

285

NKK-HUST

Ví dụ Thủ tục lá

- Thủ tục lá là thủ tục không có lời gọi thủ tục khác
- Mã C:

```
int leaf_example (int g, h, i, j)
{
    int f;
    f = (g + h) - (i + j);
    return f;
}
```

 - Các tham số g, h, i, j ở \$a0, \$a1, \$a2, \$a3
 - f ở \$s0 (do đó, cần cất \$s0 ra ngăn xếp)
 - Kết quả ở \$v0

14 January 2013

Kiến trúc máy tính

286

NKK-HUST

Ví dụ Thủ tục lá

- Mã MIPS:

leaf_example:	
addi \$sp, \$sp, -4	Cất \$s0 ra stack
sw \$s0, 0(\$sp)	
add \$t0, \$a0, \$a1	Thân thủ tục
add \$t1, \$a2, \$a3	
sub \$s0, \$t0, \$t1	
add \$v0, \$s0, \$zero	Kết quả
lw \$s0, 0(\$sp)	Khôi phục \$s0
addi \$sp, \$sp, 4	
jr \$ra	Trở về

14 January 2013

Kiến trúc máy tính

287

NKK-HUST

Ví dụ Thủ tục cành

- Là thủ tục có gọi thủ tục khác
- C code:

```
int fact (int n)
{
    if (n < 1) return (1);
    else return n * fact(n - 1);
}
```

 - Tham số n ở \$a0
 - Kết quả ở \$v0

14 January 2013

Kiến trúc máy tính

288

NKK-HUST

Ví dụ Thủ tục càn (tiếp)

■ Mã MIPS:

fact:		
addi	\$sp, \$sp, -8	# dành stack cho 2 mục
sw	\$ra, 4(\$sp)	# cất địa chỉ trở về
sw	\$a0, 0(\$sp)	# cất tham số n
slti	\$t0, \$a0, 1	# kiểm tra n < 1
beq	\$t0, \$zero, L1	
addi	\$v0, \$zero, 1	# nếu đúng, kết quả là 1
addi	\$sp, \$sp, 8	# lấy 2 mục từ stack
jr	\$ra	# và trở về
L1:	addi	\$a0, \$a0, -1
	jal	fact
		# gọi đệ qui
	lw	\$a0, 0(\$sp)
		# khôi phục n ban đầu
	lw	\$ra, 4(\$sp)
		# và địa chỉ trở về
	addi	\$sp, \$sp, 8
		# lấy 2 mục từ stack
	mul	\$v0, \$a0, \$v0
		# nhân để nhận kết quả
	jr	\$ra
		# và trở về

14 January 2013

Kiến trúc máy tính

289

NKK-HUST

Sử dụng Stack khi gọi thủ tục

Before calling

After calling

14 January 2013

Kiến trúc máy tính

290

NKK-HUST

Ví dụ sử dụng stack

■ Cất \$fp, \$ra và \$s0 ra stack và khôi phục chúng cuối thủ tục

```
proc: sw    $fp, -4($sp)    # cất giá trị cũ của frame pointer
      addi  $fp, $sp, 0     # cất ($sp) sang $fp
      addi  $sp, $sp, -12   # chuyển đỉnh stack 3 vị trí
      sw    $ra, -8($fp)   # cất ($ra) ra stack
      sw    $s0, -12($fp)  # cất ($s0) ra stack
      .
      .
      .

      lw    $s0, -12($fp)  # lấy phần tử đỉnh stack đưa vào $s0
      lw    $ra, -8($fp)   # lấy phần tử tiếp theo ở stack đưa vào $ra
      addi  $sp, $fp, 0    # khôi phục $sp
      lw    $fp, -4($sp)   # khôi phục $fp
      jr    $ra            # trở về từ thủ tục
```

14 January 2013

Kiến trúc máy tính

291

NKK-HUST

Dữ liệu ký tự

■ Các tập ký tự được mã hóa theo byte

■ ASCII: 128 ký tự

■ 95 ký thị hiển thị , 33 mã điều khiển

■ Latin-1: 256 ký tự

■ ASCII và các ký tự mở rộng

■ Unicode: Tập ký tự 32-bit

■ Được sử dụng trong Java, C++, ...

■ Hầu hết các ký tự của các ngôn ngữ trên thế giới và các ký hiệu

14 January 2013

Kiến trúc máy tính

292

NKK-HUST

Các thao tác với Byte/Halfword

- Có thể sử dụng các phép toán logic
- Nạp/Lưu byte/halfword trong MIPS
 - `lb rt, offset(rs)` `lh rt, offset(rs)`
 - Mở rộng dấu thành 32 bits trong `rt`
 - `lbu rt, offset(rs)` `lhu rt, offset(rs)`
 - Mở rộng zero thành 32 bits trong `rt`
 - `sb rt, offset(rs)` `sh rt, offset(rs)`
 - Chỉ lưu byte/halfword bên phải

14 January 2013

Kiến trúc máy tính

293

NKK-HUST

Ví dụ copy String

- Mã C:

```
void strcpy (char x[], char y[])
{
    int i;
    i = 0;
    while ((x[i]=y[i])!='\0')
        i += 1;
}
```

 - Các địa chỉ của `x`, `y` ở `$a0`, `$a1`
 - `i` ở `$s0`

14 January 2013

Kiến trúc máy tính

294

NKK-HUST

Ví dụ Copy String

- MIPS code:

strcpy:		
<code>addi \$sp, \$sp, -4</code>	<code># adjust stack for 1 item</code>	
<code>sw \$s0, 0(\$sp)</code>	<code># save \$s0</code>	
<code>add \$s0, \$zero, \$zero</code>	<code># i = 0</code>	
L1: <code>add \$t1, \$s0, \$a1</code>	<code># addr of y[i] in \$t1</code>	
<code>lbu \$t2, 0(\$t1)</code>	<code># \$t2 = y[i]</code>	
<code>add \$t3, \$s0, \$a0</code>	<code># addr of x[i] in \$t3</code>	
<code>sb \$t2, 0(\$t3)</code>	<code># x[i] = y[i]</code>	
<code>beq \$t2, \$zero, L2</code>	<code># exit loop if y[i] == 0</code>	
<code>addi \$s0, \$s0, 1</code>	<code># i = i + 1</code>	
<code>j L1</code>	<code># next iteration of loop</code>	
L2: <code>lw \$s0, 0(\$sp)</code>	<code># restore saved \$s0</code>	
<code>addi \$sp, \$sp, 4</code>	<code># pop 1 item from stack</code>	
<code>jr \$ra</code>	<code># and return</code>	

14 January 2013

Kiến trúc máy tính

295

NKK-HUST

Địa chỉ hóa cho các lệnh Branch

- Các lệnh **Branch** chỉ ra:
 - Mã thao tác, hai thành ghi, offset
- Hầu hết các đích rẽ nhánh là rẽ nhánh gần
 - Rẽ xuôi hoặc rẽ ngược

op	rs	rt	constant
6 bits	5 bits	5 bits	16 bits

- Định địa chỉ tương đối với PC
 - PC-relative addressing
 - Địa chỉ đích = $PC + \text{hằng số} \times 4$
 - Chú ý: trước đó PC đã được tăng lên

14 January 2013

Kiến trúc máy tính

296

NKK-HUST

Rẽ nhánh xa

- Nếu đích rẽ nhánh là quá xa để mã hóa với offset 16-bit, assembler sẽ viết lại code
- Ví dụ

```
beq $s0, $s1, L1
      ↓
bne $s0, $s1, L2
j    L1
L2: ...
```

14 January 2013

Kiến trúc máy tính

301

NKK-HUST

6. Tóm tắt về các phương pháp định địa chỉ

1. Immediate addressing

op rs rt Immediate

1. Định địa chỉ tức thì

2. Register addressing

op rs rt rd ... Immediate

2. Định địa chỉ thanh ghi

3. Base addressing

op rs rt Address

3. Định địa chỉ cơ sở

4. PC-relative addressing

op rs rt Address

4. Định địa chỉ tương đối với PC

5. Pseudo-direct addressing

op Address

5. Định địa chỉ giả trực tiếp

14 January 2013

Kiến trúc máy tính

302

NKK-HUST

7. Thêm một số lệnh

Lệnh nhân và lệnh chia:

```
mult $s0, $s1 # set Hi,Lo to ($s0)×($s1)
div  $s0, $s1 # set Hi to ($s0)mod($s1)
                # and Lo to ($s0)/($s1)
```

Lệnh copy nội dung các thanh ghi Hi và Lo

Reg file

Mul/Div unit

Hi Lo

Lệnh copy nội dung các thanh ghi Hi và Lo

```
mfhi $t0 # set $t0 to (Hi)
mflo $t0 # set $t0 to (Lo)
```

R

31 op 25 rs 20 rt 15 rd 10 sh 5 fn 0

0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 x 0

ALU instruction Source register 1 Source register 2 Unused Unused mult = 24 div = 26

R

31 op 25 rs 20 rt 15 rd 10 sh 5 fn 0

0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 x 0

ALU instruction Unused Unused Destination register Unused mfi = 16 mlo = 16

14 January 2013

Kiến trúc máy tính

303

NKK-HUST

Các lệnh số học số nguyên không dấu

addu \$t0, \$s0, \$s1 # set \$t0 to (\$s0)+(\$s1)

subu \$t0, \$s0, \$s1 # set \$t0 to (\$s0)-(\$s1)

mul tu \$s0, \$s1 # set Hi , Lo to (\$s0)×(\$s1)

div vu \$s0, \$s1 # set Hi to (\$s0)mod(\$s1)

and Lo to (\$s0)/(\$s1)

addi u \$t0, \$s0, 61 # set \$t0 to (\$s0)+61;

the immediate operand is

sign extended

14 January 2013

Kiến trúc máy tính

304

Nguyễn Kim Khánh - ĐHBKHN

NKK-HUST

Các lệnh với số dấu phẩy động

- Các thanh ghi số dấu phẩy động
 - 32 thanh ghi 32-bit (single-precision): \$f0, \$f1, ... \$f31
 - Cặp đôi để chứa dữ liệu dạng 64-bit (double-precision): \$f0/\$f1, \$f2/\$f3, ...
 - (Release 2 of MIPS ISA supports 32 × 64-bit FP reg's)
- Các lệnh số dấu phẩy động chỉ thực hiện trên các thanh ghi số dấu phẩy động
- FP load and store instructions
 - lwc1, ldc1, swc1, sdc1
 - e.g., ldc1 \$f8, 32(\$sp)

14 January 2013 305

NKK-HUST

Các lệnh với số dấu phẩy động

- Single-precision arithmetic
 - add.s, sub.s, mul.s, div.s
 - e.g., add.s \$f0, \$f1, \$f6
- Double-precision arithmetic
 - add.d, sub.d, mul.d, div.d
 - e.g., mul.d \$f4, \$f4, \$f6
- Single- and double-precision comparison
 - c.xx.s, c.xx.d (xx is eq, lt, le, ...)
 - Sets or clears FP condition-code bit
 - e.g., c.lt.s \$f3, \$f4
- Branch on FP condition code true or false
 - bc1t, bc1f
 - e.g., bc1t TargetLabel

14 January 2013 306

NKK-HUST

8. Thực hành lập trình hợp ngữ MIPS

- Phần mềm lập trình: MARS
- Lập trình các ví dụ
- Chạy các chương trình có sẵn và phân tích
- MIPS Reference Data

14 January 2013 Kiến trúc máy tính 307

NKK-HUST

Dịch và chạy ứng dụng

```

graph TD
    A[High Level Code] --> B[Compiler]
    B --> C[Assembly Code]
    C --> D[Assembler]
    D --> E[Object File]
    E --> F[Linker]
    G[Object Files] --> F
    H[Library Files] --> F
    F --> I[Executable]
    I --> J[Loader]
    J --> K[Memory]
    
```

14 January 2013 Kiến trúc máy tính 308

NKK-HUST

Chương trình trong bộ nhớ

- Các lệnh (instructions)
- Dữ liệu
 - Toàn cục/tĩnh: được cấp phát trước khi chương trình bắt đầu thực hiện
 - Động: được cấp phát trong khi chương trình thực hiện
- Bộ nhớ:
 - $2^{32} = 4$ gigabytes (4 GB)
 - Từ địa chỉ 0x00000000 đến 0xFFFFFFFF

14 January 2013

Kiến trúc máy tính

309

NKK-HUST

Bản đồ bộ nhớ của MIPS

Address	Segment
0xFFFFFC	Reserved
0x80000000	Stack
0x7FFFFFFC	Dynamic Data
0x10010000	Heap
0x1000FFFC	Static Data
0x10000000	Text
0x0FFFFFFC	Reserved
0x00400000	
0x003FFFFC	
0x00000000	

14 January 2013

Kiến trúc máy tính

310

NKK-HUST

Ví dụ: Mã C

```
int f, g, y; // global
variables

int main(void)
{
    f = 2;
    g = 3;
    y = sum(f, g);
    return y;
}

int sum(int a, int b) {
    return (a + b);
}
```

14 January 2013

Kiến trúc máy tính

311

NKK-HUST

Ví dụ chương trình hợp ngữ

```
.data
f:
g:
y:
.text
main:
    addi $sp, $sp, -4 # stack frame
    sw   $ra, 0($sp) # store $ra
    addi $a0, $0, 2  # $a0 = 2
    sw   $a0, f      # f = 2
    addi $a1, $0, 3  # $a1 = 3
    sw   $a1, g      # g = 3
    jal  sum         # call sum
    sw   $v0, y      # y = sum()
    lw   $ra, 0($sp) # restore $ra
    addi $sp, $sp, 4 # restore $sp
    jr   $ra         # return to OS
sum:
    add  $v0, $a0, $a1 # $v0 = a + b
    jr   $ra          # return
```

14 January 2013

Kiến trúc máy tính

312

NKK-HUST

Bảng ký hiệu

Ký hiệu	Địa chỉ
f	0x10000000
g	0x10000004
y	0x10000008
main	0x00400000
sum	0x0040002C

14 January 2013

Kiến trúc máy tính

313

NKK-HUST

Chương trình thực thi

Executable file header	Text Size	Data Size
	0x34 (52 bytes)	0xC (12 bytes)

Text segment	Address	Instruction
	0x00400000	0x23BDFEFC
	0x00400004	0xAFBF0000
	0x00400008	0x20040002
	0x0040000C	0xAF848000
	0x00400010	0x20050003
	0x00400014	0xAF858004
	0x00400018	0x0C10000B
	0x0040001C	0xAF828008
	0x00400020	0x8FBF0000
	0x00400024	0x23BD0004
	0x00400028	0x03E00008
	0x0040002C	0x00851020
	0x00400030	0x03E00008

Data segment	Address	Data
	0x10000000	f
	0x10000004	g
	0x10000008	y

```
addi $sp, $sp, -4
sw   $ra, 0($sp)
addi $a0, $0, 2
sw   $a0, 0x8000($gp)
addi $a1, $0, 3
sw   $a1, 0x8004($gp)
jal  0x0040002C
sw   $v0, 0x8008($gp)
lw   $ra, 0($sp)
addi $sp, $sp, -4
jr   $ra
add  $v0, $a0, $a1
jr   $ra
```

14 January 2013

Kiến trúc máy tính

314

NKK-HUST

Chương trình trong bộ nhớ

Address

Memory

0x7FFFFFFC

Reserved

Stack

↓

0x10010000

↑

Heap

↓

↓

↓

↓

y

g

f

↓

↓

0x03E00008

0x00851020

0x03E00008

0x23BD0004

0x8FBF0000

0xAF828008

0x0C10000B

0xAF858004

0x20050003

0xAF848000

0x20040002

0xAFBF0000

0x23BDFEFC

0x00400000

Reserved

← \$sp = 0x7FFFFFFC

← \$gp = 0x10008000

← PC = 0x00400000

14 January 2013

Kiến trúc máy tính

315

NKK-HUST

Ví dụ lệnh giả (Pseudoinstruction)

Pseudoinstruction	MIPS Instructions
li \$s0, 0x1234AA77	lui \$s0, 0x1234 ori \$s0, 0xAA77
mul \$s0, \$s1, \$s2	mult \$s1, \$s2 mflo \$s0
clear \$t0	add \$t0, \$0, \$0
move \$s1, \$s2	add \$s2, \$s1, \$0
nop	sll \$0, \$0, 0

14 January 2013

Kiến trúc máy tính

316

NKK-HUST

5.3. Kiến trúc tập lệnh Intel x86(*)

- Sự tiến hóa của các bộ xử lý Intel
 - 8080 (1974): 8-bit microprocessor
 - Accumulator, plus 3 index-register pairs
 - 8086 (1978): 16-bit extension to 8080
 - Complex instruction set (CISC)
 - 8087 (1980): floating-point coprocessor
 - Adds FP instructions and register stack
 - 80286 (1982): 24-bit addresses, MMU
 - Segmented memory mapping and protection
 - 80386 (1985): 32-bit extension (now IA-32)
 - Additional addressing modes and operations
 - Paged memory mapping as well as segments

14 January 2013

Kiến trúc máy tính

317

NKK-HUST

Kiến trúc tập lệnh Intel x86

- i486 (1989): pipelined, on-chip caches and FPU
 - Compatible competitors: AMD, Cyrix, ...
- Pentium (1993): superscalar, 64-bit datapath
 - Later versions added MMX (Multi-Media eXtension) instructions
 - The infamous FDIV bug
- Pentium Pro (1995), Pentium II (1997)
 - New microarchitecture
- Pentium III (1999)
 - Added SSE (Streaming SIMD Extensions) and associated registers
- Pentium 4 (2001)
 - New microarchitecture
 - Added SSE2 instructions
- Intel Core (2006)
 - Added SSE4 instructions, virtual machine support

14 January 2013

Kiến trúc máy tính

318

NKK-HUST

Các thanh ghi cơ bản của x86

Name	Bit	Use
EAX	31	General Purpose Register 0
ECX		General Purpose Register 1
EDX		General Purpose Register 2
EBX		General Purpose Register 3
ESP		General Purpose Register 4
EBP		General Purpose Register 5
ESI		General Purpose Register 6
EDI		General Purpose Register 7
CR0		Control segment pointer
CR2		Stack segment pointer (top of stack)
CR3		Data segment pointer 0
CR4		Data segment pointer 1
CR5		Data segment pointer 2
CR6		Data segment pointer 3
EIP		Instruction pointer (PC)
EFLAGS		Condition codes

14 January 2013

Kiến trúc máy tính

319

NKK-HUST

Các phương pháp định địa chỉ cơ bản

- Hai toán hạng của lệnh


Source/dest operand	Second source operand
Register	Register
Register	Immediate
Register	Memory
Memory	Register
Memory	Immediate
- Các phương pháp định địa chỉ
 - Address in register
 - Address = $R_{base} + \text{displacement}$
 - Address = $R_{base} + 2^{\text{scale}} \times R_{index}$ (scale = 0, 1, 2, or 3)
 - Address = $R_{base} + 2^{\text{scale}} \times R_{index} + \text{displacement}$

14 January 2013

Kiến trúc máy tính

320

NKK-HUST



Mã hóa lệnh x86

a. JE $EF + displacement$

4	4	8
JE	Condition	Displacement

b. CALL

8	32
CALL	Offset

c. MOV $EBX, [EDI + 45]$

8	1	1	8
MOV	d	w	Imm Postbyte
Displacement			

d. PUSH ESI

5	8
PUSH	Reg

e. ADD $EAX, \#6766$

4	3	1	32
ADD	Reg	w	Immediate

f. TEST $EBX, 0x2$


7	1	8	32
TEST	w	Postbyte	Immediate

14 January 2013

Kiến trúc máy tính

321

NKK-HUST



Hết chương 5

14 January 2013

Kiến trúc máy tính

322

NKK-HUST



Kiến trúc máy tính

Chương 6

CẤU TRÚC BỘ XỬ LÝ

(Processor)

Nguyễn Kim Khánh


Trường Đại học Bách khoa Hà Nội

14 January 2013

Kiến trúc máy tính

323

NKK-HUST



Nội dung học phần

- Chương 1. Giới thiệu chung
- Chương 2. Cơ bản về logic số
- Chương 3. Hệ thống máy tính
- Chương 4. Số học máy tính
- Chương 5. Kiến trúc tập lệnh
- Chương 6. Cấu trúc bộ xử lý
- Chương 7. Bộ nhớ
- Chương 8. Vào-ra
- Chương 9. Các kiến trúc song song

14 January 2013

Kiến trúc máy tính

324

NKK-HUST

Nội dung

- 6.1. Tổ chức của bộ xử lý
- 6.2. Thiết kế đơn vị điều khiển
- 6.3. Kỹ thuật đường ống lệnh
- 6.4. Ví dụ thiết kế bộ xử lý theo kiến trúc MIPS*

14 January 2013 Kiến trúc máy tính 325

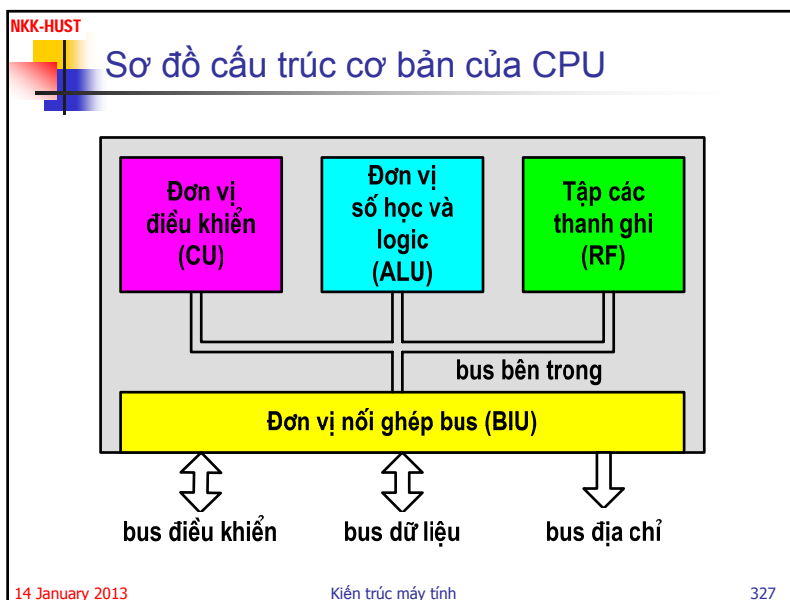
NKK-HUST

6.1. Tổ chức của CPU

1. Cấu trúc cơ bản của CPU

- Nhiệm vụ của CPU:
 - Nhận lệnh (Fetch Instruction): CPU đọc lệnh từ bộ nhớ.
 - Giải mã lệnh (Decode Instruction): xác định thao tác mà lệnh yêu cầu.
 - Nhận dữ liệu (Fetch Data): nhận dữ liệu từ bộ nhớ hoặc các cổng vào-ra.
 - Xử lý dữ liệu (Process Data): thực hiện phép toán số học hay phép toán logic với các dữ liệu.
 - Ghi dữ liệu (Write Data): ghi dữ liệu ra bộ nhớ hay cổng vào-ra

14 January 2013 Kiến trúc máy tính 326



NKK-HUST

Các thành phần cơ bản của CPU

- Đơn vị điều khiển (Control Unit - CU)
- Đơn vị số học và logic (Arithmetic and Logic Unit - ALU)
- Tập thanh ghi (Register File - RF)
- Đơn vị nối ghép bus (Bus Interface Unit - BIU)
- Bus bên trong (Internal Bus)

14 January 2013 Kiến trúc máy tính 328

NKK-HUST

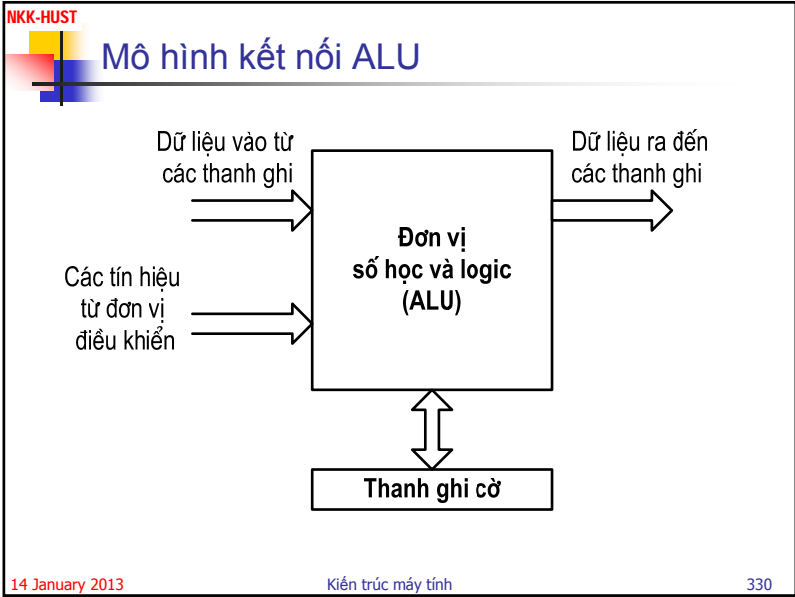
2. Đơn vị số học và logic

- **Chức năng:** Thực hiện các phép toán số học và phép toán logic:
 - Số học: cộng, trừ, nhân, chia, tăng, giảm, đảo dấu
 - Logic: AND, OR, XOR, NOT, phép dịch bit.

14 January 2013

Kiến trúc máy tính

329



NKK-HUST

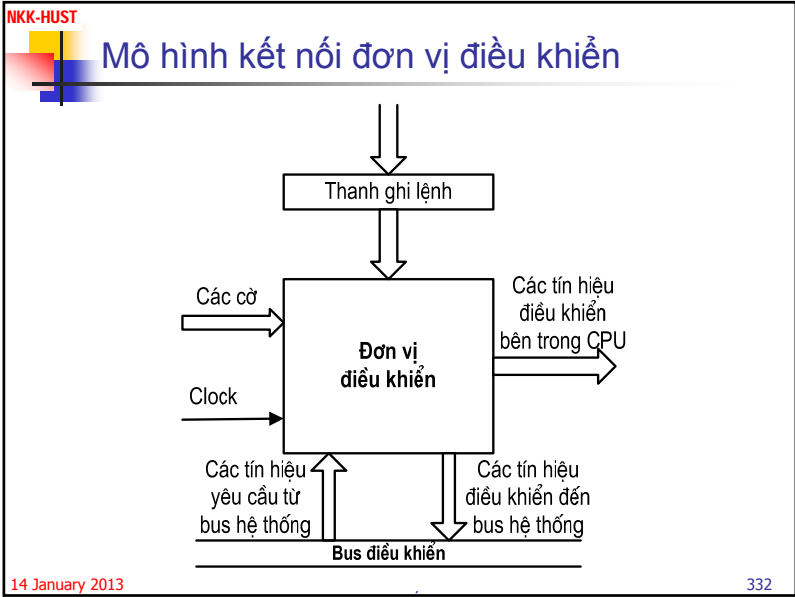
3. Đơn vị điều khiển

- **Chức năng**
 - Điều khiển nhận lệnh từ bộ nhớ đưa vào thanh ghi lệnh
 - Tăng nội dung của PC để trở sang lệnh kế tiếp
 - Giải mã lệnh đã được nhận để xác định thao tác mà lệnh yêu cầu
 - Phát ra các tín hiệu điều khiển thực hiện lệnh
 - Nhận các tín hiệu yêu cầu từ bus hệ thống và đáp ứng với các yêu cầu đó.

14 January 2013

Kiến trúc máy tính

331



NKK-HUST

Các tín hiệu đưa đến đơn vị điều khiển

- Clock: tín hiệu nhịp từ mạch tạo dao động bên ngoài.
- Mã lệnh từ thanh ghi lệnh đưa đến để giải mã.
- Các cờ từ thanh ghi cờ cho biết trạng thái của CPU.
- Các tín hiệu yêu cầu từ bus điều khiển

14 January 2013

Kiến trúc máy tính

333

NKK-HUST

Các tín hiệu phát ra từ đơn vị điều khiển

- Các tín hiệu điều khiển bên trong CPU:
 - Điều khiển các thanh ghi
 - Điều khiển ALU
- Các tín hiệu điều khiển bên ngoài CPU:
 - Điều khiển bộ nhớ
 - Điều khiển các mô-đun vào-ra

14 January 2013

Kiến trúc máy tính

334

NKK-HUST

4. Hoạt động của chu trình lệnh

Chu trình lệnh

- Nhận lệnh
- Giải mã lệnh
- Nhận toán hạng
- Thực hiện lệnh
- Cát toán hạng
- Ngắt

14 January 2013

Kiến trúc máy tính

335

NKK-HUST

Giản đồ trạng thái chu trình lệnh

```
graph TD; N1((Nhận lệnh)) --> T1((Tinh địa chỉ của lệnh)); T1 --> G1((Giải mã thao tác lệnh)); G1 --> T2((Tinh địa chỉ toán hạng)); T2 --> T3((Thao tác dữ liệu)); T3 --> T4((Tinh địa chỉ toán hạng)); T4 --> K1((Kiểm tra ngắt)); K1 -- "Có Ngắt" --> N2((Ngắt)); K1 -- "Không Ngắt" --> L1((Lệnh hoàn thành, nhận lệnh tiếp theo)); L1 --> T1; T3 -- "Nhiều toán hạng" --> T4; K1 -- "Nhiều toán hạng" --> T4;
```

14 January 2013

Kiến trúc máy tính

336

NKK-HUST

Nhận lệnh

- CPU đưa địa chỉ của lệnh cần nhận từ bộ đếm chương trình PC ra bus địa chỉ
- CPU phát tín hiệu điều khiển đọc bộ nhớ
- Lệnh từ bộ nhớ được đặt lên bus dữ liệu và được CPU copy vào thanh ghi lệnh IR
- CPU tăng nội dung PC để trở sang lệnh kế tiếp

14 January 2013 Kiến trúc máy tính 337

NKK-HUST

Sơ đồ mô tả quá trình nhận lệnh

PC: Bộ đếm chương trình
IR: Thanh ghi lệnh

Bus địa chỉ Bus dữ liệu Bus điều khiển

14 January 2013 Kiến trúc máy tính 338

NKK-HUST

Giải mã lệnh

- Lệnh từ thanh ghi lệnh IR được đưa đến đơn vị điều khiển
- Đơn vị điều khiển tiến hành giải mã lệnh để xác định thao tác phải thực hiện
- Giải mã lệnh xảy ra bên trong CPU

14 January 2013 Kiến trúc máy tính 339

NKK-HUST

Nhận dữ liệu từ bộ nhớ

- CPU đưa địa chỉ của toán hạng ra bus địa chỉ
- CPU phát tín hiệu điều khiển đọc
- Toán hạng được đọc vào CPU
- Tương tự như nhận lệnh

14 January 2013 Kiến trúc máy tính 340

NKK-HUST

Sơ đồ mô tả nhận dữ liệu từ bộ nhớ

MAR: Thanh ghi địa chỉ bộ nhớ
MBR: Thanh ghi đệm bộ nhớ

Bus địa chỉ Bus dữ liệu Bus điều khiển

14 January 2013

Kiến trúc máy tính

341

NKK-HUST

Thực hiện lệnh

- Có nhiều dạng tùy thuộc vào lệnh
- Có thể là:
 - Đọc/Ghi bộ nhớ
 - Vào/Ra
 - Chuyển giữa các thanh ghi
 - Thao tác số học/logic
 - Chuyển điều khiển (rẽ nhánh)
 - ...

14 January 2013

Kiến trúc máy tính

342

NKK-HUST

Ghi toán hạng

- CPU đưa địa chỉ ra bus địa chỉ
- CPU đưa dữ liệu cần ghi ra bus dữ liệu
- CPU phát tín hiệu điều khiển ghi
- Dữ liệu trên bus dữ liệu được copy đến vị trí xác định

14 January 2013

Kiến trúc máy tính

343

NKK-HUST

Sơ đồ mô tả quá trình ghi toán hạng

MAR: Thanh ghi địa chỉ bộ nhớ
MBR: Thanh ghi đệm bộ nhớ

Bus địa chỉ Bus dữ liệu Bus điều khiển

14 January 2013

Kiến trúc máy tính

344

NKK-HUST

Ngắt

- Nội dung của bộ đếm chương trình PC (địa chỉ trở về sau khi ngắt) được đưa ra bus dữ liệu
- CPU đưa địa chỉ (thường được lấy từ con trỏ ngăn xếp SP) ra bus địa chỉ
- CPU phát tín hiệu điều khiển ghi bộ nhớ
- Địa chỉ trở về trên bus dữ liệu được ghi ra vị trí xác định (ở ngăn xếp)
- Địa chỉ lệnh đầu tiên của chương trình con điều khiển ngắt được nạp vào PC

14 January 2013

Kiến trúc máy tính

345

NKK-HUST

Sơ đồ mô tả chu trình ngắt

MAR: Thanh ghi địa chỉ bộ nhớ
MBR: Thanh ghi đệm bộ nhớ
PC: Bộ đếm chương trình
SP: Con trỏ ngăn xếp

Bus địa chỉ
Bus dữ liệu
Bus điều khiển

14 January 2013

Kiến trúc máy tính

346

NKK-HUST

6.2. Các phương pháp thiết kế đơn vị điều khiển

- Đơn vị điều khiển vi chương trình (Microprogrammed Control Unit)
- Đơn vị điều khiển nối kết cứng (Hardwired Control Unit)

14 January 2013

Kiến trúc máy tính

347

NKK-HUST

1. Đơn vị điều khiển vi chương trình

- Bộ nhớ vi chương trình (ROM) lưu trữ các vi chương trình (microprogram)
- Một vi chương trình bao gồm các vi lệnh (microinstruction)
- Mỗi vi lệnh mã hoá cho một vi thao tác (microoperation)
- Để hoàn thành một lệnh cần thực hiện một hoặc một vài vi chương trình
- Tốc độ chậm

Các tín hiệu điều khiển từ bus hệ thống
Clock
Các cơ
Vi lệnh tiếp theo
Các tín hiệu điều khiển bên trong CPU
Các tín hiệu điều khiển đến bus hệ thống

14 January 2013

Kiến trúc máy tính

348

NKK-HUST

2. Đơn vị điều khiển nối kết cứng

- Sử dụng mạch cứng để giải mã và tạo các tín hiệu điều khiển thực hiện lệnh
- Tốc độ nhanh
- Đơn vị điều khiển phức tạp

Clock

Mạch phân chia thời gian

T₁

T₂

⋮

T_n

Thanh ghi lệnh

Bộ giải mã

Đơn vị điều khiển

Các tín hiệu điều khiển

C₀

C₁

⋯

C_{m-1}

Các cổng

14 January 2013

Kiến trúc máy tính

349

NKK-HUST

6.3. Kỹ thuật đường ống lệnh và song song mức lệnh

- Kỹ thuật đường ống lệnh (Instruction Pipelining): Chia chu trình lệnh thành các công đoạn và cho phép thực hiện gối lên nhau (như dây chuyền lắp ráp)
- Chẳng hạn có 6 công đoạn:
 - Nhận lệnh (Fetch Instruction - FI)
 - Giải mã lệnh (Decode Instruction - DI)
 - Tính địa chỉ toán hạng (Calculate Operand Address-CO)
 - Nhận toán hạng (Fetch Operands - FO)
 - Thực hiện lệnh (Execute Instruction - EI)
 - Ghi toán hạng (Write Operands - WO)

14 January 2013

Kiến trúc máy tính

350

NKK-HUST

Biểu đồ thời gian của đường ống lệnh

	1	2	3	4	5	6	7	8	9	10	11	12	13	t
lệnh 1	FI	DI	CO	FO	EI	WO								
lệnh 2		FI	DI	CO	FO	EI	WO							
lệnh 3			FI	DI	CO	FO	EI	WO						
lệnh 4				FI	DI	CO	FO	EI	WO					
lệnh 5					FI	DI	CO	FO	EI	WO				
lệnh 6						FI	DI	CO	FO	EI	WO			
lệnh 7							FI	DI	CO	FO	EI	WO		
lệnh 8								FI	DI	CO	FO	EI	WO	

14 January 2013

Kiến trúc máy tính

351

NKK-HUST

Các Hazard (trở ngại) của đường ống lệnh

- Hazard: Tình huống ngăn cản bắt đầu của lệnh tiếp theo ở chu kỳ tiếp theo.
 - Hazard cấu trúc: do tài nguyên được yêu cầu đang bận
 - Hazard dữ liệu: cần phải đợi để lệnh trước hoàn thành việc đọc/ghi dữ liệu
 - Hazard điều khiển: do rẽ nhánh gây ra

14 January 2013

Kiến trúc máy tính

352

Nguyễn Kim Khánh - ĐHBKHN

NKK-HUST

Hazard về cấu trúc

- Khắc phục:
 - nhân tài nguyên để tránh xung đột
 - Làm trễ

Ví dụ:

- Bus dữ liệu: truyền lệnh và dữ liệu

→ Bus lệnh riêng, bus dữ liệu riêng (cache lệnh và cache dữ liệu)

14 January 2013

Kiến trúc máy tính

353

NKK-HUST

Ví dụ Hazard về cấu trúc

conflict on arithmetic unit

3 clocks necessary for multiplication

	I	D	F	E	E	E	W			
MULT A,B,C										
MULT D,E,F		I	D	F	●	●	E	E	E	W

cache miss

stall

TLB miss

14 January 2013

Kiến trúc máy tính

354

NKK-HUST

Hazard về dữ liệu

- Các dạng:
 - RAW (Read After Write)
 - WAR (Write After Read)
 - WAW (Write After Write)

14 January 2013

Kiến trúc máy tính

355

NKK-HUST

Sự phụ thuộc về dữ liệu

RAW	ADD A,B,C	Write-A must be earlier than
	ADD E,A,D	Read-A

WAR	ADD A,B,C	Read-B must be earlier than
	ADD B,D,E	Write-B

WAW	ADD A,B,C	First Write-A must be earlier
	ADD A,D,E	Than second Write-A

14 January 2013

Kiến trúc máy tính

356

NKK-HUST

WAR and WAW

WAR

ADD A,B,C
ADD B,D,E

Read-B is earlier than Write-B

WAW

ADD A,B,C
ADD A,D,E

first Write-A is earlier than second Write-A

no conflict at in-order pipeline

conflict at out-of-order pipeline

14 January 2013

Kiến trúc máy tính

357

NKK-HUST

RAW

ADD A,B,C
ADD E,A,D

Write-A must be earlier
Than Read-A

stall

14 January 2013

Kiến trúc máy tính

358

NKK-HUST

Hazard điều khiển

Wait for branch

BEQ A, B, Label
LOAD C, X
....

Label: LOAD C, Y

next instruction to a branch instruction cannot be fetched until branch condition defined and PC updated

address calculation

set PC

stall

14 January 2013

Kiến trúc máy tính

359

NKK-HUST

Các kiến trúc song song mức lệnh

- Siêu đường ống (Superpipeline & Hyperpipeline)
- Siêu vô hướng (Superscalar)

14 January 2013

Kiến trúc máy tính

360

NKK-HUST

Superpipeline

Lệnh 1

Lệnh 2

Lệnh 3

Lệnh 4

FI	DI	CO	FO	EI	WO
FI	DI	CO	FO	EI	WO
FI	DI	CO	FO	EI	WO
FI	DI	CO	FO	EI	WO

14 January 2013

Kiến trúc máy tính

361

NKK-HUST

Superscalar

Lệnh 1

Lệnh 2

Lệnh 3

Lệnh 4

Lệnh 5

Lệnh 6

FI	DI	CO	FO	EI	WO
FI	DI	CO	FO	EI	WO
FI	DI	CO	FO	EI	WO
FI	DI	CO	FO	EI	WO
FI	DI	CO	FO	EI	WO
FI	DI	CO	FO	EI	WO

14 January 2013

Kiến trúc máy tính

362

NKK-HUST

6.4. Thiết kế bộ xử lý theo kiến trúc MIPS*

Bộ xử lý MIPS – Chapter 4 – [2]

14 January 2013

Kiến trúc máy tính

363

NKK-HUST

Hết chương 6

14 January 2013

Kiến trúc máy tính

364

NKK-HUST

Kiến trúc máy tính

Chương 7

BỘ NHỚ MÁY TÍNH

Nguyễn Kim Khánh
Trường Đại học Bách khoa Hà Nội

Kiến trúc máy tính

NKK-HUST

Nội dung học phần

- Chương 1. Giới thiệu chung
- Chương 2. Cơ bản về logic số
- Chương 3. Hệ thống máy tính
- Chương 4. Số học máy tính
- Chương 5. Kiến trúc tập lệnh
- Chương 6. Cấu trúc bộ xử lý
- Chương 7. Bộ nhớ**
- Chương 8. Vào-ra
- Chương 9. Các kiến trúc song song

14 January 2013

Kiến trúc máy tính

366

NKK-HUST

Nội dung của chương 7

- 7.1. Phân cấp bộ nhớ
- 7.2. Bộ nhớ chính
- 7.3. Bộ nhớ cache
- 7.4. Bộ nhớ ngoài
- 7.5. Bộ nhớ ảo

14 January 2013

Kiến trúc máy tính

367

NKK-HUST

7.1. Phân cấp bộ nhớ máy tính

1. Các đặc trưng của bộ nhớ máy tính

- Vị trí
 - Bên trong CPU:
 - tập thanh ghi
 - Bộ nhớ trong:
 - bộ nhớ chính
 - bộ nhớ cache
 - Bộ nhớ ngoài: các thiết bị nhớ
- Dung lượng
 - Độ dài từ nhớ (tính bằng bit)
 - Số lượng từ nhớ

14 January 2013

Kiến trúc máy tính

368

NKK-HUST

Các đặc trưng của hệ thống nhớ (tiếp)

- Đơn vị truyền
 - Từ nhớ
 - Khối nhớ
- Phương pháp truy nhập
 - Truy nhập tuần tự (băng từ)
 - Truy nhập trực tiếp (các loại đĩa)
 - Truy nhập ngẫu nhiên (bộ nhớ bán dẫn)
 - Truy nhập liên kết (cache)

14 January 2013 Kiến trúc máy tính 369

NKK-HUST

Các đặc trưng của hệ thống nhớ (tiếp)

- Hiệu năng (performance)
 - Thời gian truy nhập
 - Chu kỳ nhớ
 - Tốc độ truyền
- Kiểu vật lý
 - Bộ nhớ bán dẫn
 - Bộ nhớ từ
 - Bộ nhớ quang

14 January 2013 Kiến trúc máy tính 370

NKK-HUST

Các đặc trưng của hệ thống nhớ (tiếp)

- Các đặc tính vật lý
 - Khả biến / Không khả biến (volatile / nonvolatile)
 - Xoá được / không xoá được
- Tổ chức

14 January 2013 Kiến trúc máy tính 371

NKK-HUST

2. Phân cấp bộ nhớ

14 January 2013 Kiến trúc máy tính 372

NKK-HUST

Công nghệ bộ nhớ

- Static RAM (SRAM)
 - 0.5ns – 2.5ns, \$2000 – \$5000 per GB
- Dynamic RAM (DRAM)
 - 50ns – 70ns, \$20 – \$75 per GB
- Ổ đĩa từ
 - 5ms – 20ms, \$0.20 – \$2 per GB
- Bộ nhớ lý tưởng
 - Thời gian truy nhập như SRAM
 - Dung lượng và giá thành như ổ đĩa cứng

14 January 2013

Kiến trúc máy tính

373

NKK-HUST

Ví dụ hệ thống nhớ thông dụng

The diagram illustrates a memory hierarchy. On the left, a box labeled 'Bộ vi xử lý' (Microprocessor) contains a 'CPU' box. Inside the CPU box is a 'Tập thanh ghi' (Register set). To the right of the CPU box is 'Bộ nhớ Cache L1' (L1 Cache), followed by 'Bộ nhớ Cache L2' (L2 Cache). Arrows indicate bidirectional flow between the CPU and L1 cache, and between L1 and L2. To the right of the L2 cache is 'Bộ nhớ chính' (Main memory), and to its right is 'Bộ nhớ ngoài' (External memory). An arrow points from the external memory to a stack of documents labeled 'Bộ nhớ mạng' (Network memory). Below the diagram, text states: 'Từ trái sang phải: dung lượng tăng dần, tốc độ giảm dần, giá thành/1bit giảm dần' (From left to right: capacity increases, speed decreases, cost/bit decreases).

14 January 2013

Kiến trúc máy tính

374

NKK-HUST

Nguyên lý cục bộ hoá tham chiếu bộ nhớ

- Trong một khoảng thời gian đủ nhỏ CPU thường chỉ tham chiếu các thông tin trong một khối nhớ cục bộ
- Ví dụ:
 - Cấu trúc chương trình tuần tự
 - Vòng lặp có thân nhỏ
 - Cấu trúc dữ liệu mảng

14 January 2013

Kiến trúc máy tính

375

NKK-HUST

7.2. Bộ nhớ chính

1. Bộ nhớ bán dẫn

Kiểu bộ nhớ	Tiêu chuẩn	Khả năng xoá	Cơ chế ghi	Tính khả biến	
Read Only Memory (ROM)	Bộ nhớ chỉ đọc	Không xoá được	Mặt nạ	Không khả biến	
Programmable ROM (PROM)					
Erasable PROM (EPROM)	Bộ nhớ hầu như chỉ đọc	bằng tia cực tím, cả chip	Bảng điện		
Electrically Erasable PROM (EEPROM)		bằng điện, mức từng byte			
Flash memory	Bộ nhớ đọc-ghi	bằng điện, từng khối	Bảng điện	Khả biến	
Random Access Memory (RAM)		bằng điện, mức từng byte			

14 January 2013

Kiến trúc máy tính

376

NKK-HUST

ROM (Read Only Memory)

- Bộ nhớ không khả biến
- Lưu trữ các thông tin sau:
 - Thư viện các chương trình con
 - Các chương trình điều khiển hệ thống (BIOS)
 - Các bảng chức năng
 - Vi chương trình

14 January 2013 Kiến trúc máy tính 377

NKK-HUST

Các kiểu ROM

- ROM mặt nạ:
 - thông tin được ghi khi sản xuất
 - rất đắt
- PROM (Programmable ROM)
 - Cần thiết bị chuyên dụng để ghi bằng chương trình → chỉ ghi được một lần
- EPROM (Erasable PROM)
 - Cần thiết bị chuyên dụng để ghi bằng chương trình → ghi được nhiều lần
 - Trước khi ghi lại, xóa bằng tia cực tím

14 January 2013 Kiến trúc máy tính 378

NKK-HUST

Các kiểu ROM (tiếp)

- EEPROM (Electrically Erasable PROM)
 - Có thể ghi theo từng byte
 - Xóa bằng điện
- Flash memory (Bộ nhớ cực nhanh)
 - Ghi theo khối
 - Xóa bằng điện

14 January 2013 Kiến trúc máy tính 379

NKK-HUST

RAM (Random Access Memory)

- Bộ nhớ đọc-ghi (Read/Write Memory)
- Khả biến
- Lưu trữ thông tin tạm thời
- Có hai loại: SRAM và DRAM (Static and Dynamic)

14 January 2013 Kiến trúc máy tính 380

NKK-HUST

SRAM (Static) – RAM tĩnh

- Các bit được lưu trữ bằng các Flip-Flop
→ thông tin ổn định
- Cấu trúc phức tạp
- Dung lượng chip nhỏ
- Tốc độ nhanh
- Đắt tiền
- Dùng làm bộ nhớ cache

14 January 2013 Kiến trúc máy tính 381

NKK-HUST

DRAM (Dynamic) – RAM động

- Các bit được lưu trữ trên tụ điện
→ cần phải có mạch làm tươi
- Cấu trúc đơn giản
- Dung lượng lớn
- Tốc độ chậm hơn
- Rẻ tiền hơn
- Dùng làm bộ nhớ chính

14 January 2013 Kiến trúc máy tính 382

NKK-HUST

Các DRAM tiên tiến

- Enhanced DRAM
- Cache DRAM
- Synchronous DRAM (SDRAM): làm việc được đồng bộ bởi xung clock
- DDR-SDRAM (Double Data Rate SDRAM)
- Rambus DRAM (RDRAM)

14 January 2013 Kiến trúc máy tính 383

NKK-HUST

Tổ chức của chip nhớ

- Sơ đồ cơ bản của chip nhớ

14 January 2013 Kiến trúc máy tính 384

NKK-HUST

Các tín hiệu của chip nhớ

- Các đường địa chỉ: $A_{n-1} \div A_0 \rightarrow$ có 2^n từ nhớ
- Các đường dữ liệu: $D_{m-1} \div D_0 \rightarrow$ độ dài từ nhớ = m bit
- Dung lượng chip nhớ = $2^n \times m$ bit
- Các đường điều khiển:
 - Tín hiệu chọn chip CS (Chip Select)
 - Tín hiệu điều khiển đọc OE (Output Enable)
 - Tín hiệu điều khiển ghi WE (Write Enable)(Các tín hiệu điều khiển thường tích cực với mức 0)

14 January 2013

Kiến trúc máy tính

385

NKK-HUST

Tổ chức bộ nhớ một chiều

The diagram illustrates a 1D memory organization. It features a central memory array labeled 'Ma trận nhớ $2^n \times m$ bit'. To the left, a 'Bộ giải mã địa chỉ' (Address decoder) receives address lines $A_0, A_1, A_2, \dots, A_{n-1}$ and outputs to the memory array. To the right, a 'Bộ đệm dữ liệu' (Data buffer) is connected to the memory array and data lines D_0, D_1, \dots, D_{m-1} . At the bottom, a control block 'Điều khiển và định thời gian' (Control and timing) receives CS, WE, and OE signals and provides control to the memory array.

14 January 2013

Kiến trúc máy tính

386

NKK-HUST

Tổ chức bộ nhớ hai chiều

The diagram illustrates a 2D memory organization. It features a central memory array labeled 'Ma trận nhớ $2^{n1} \text{ hàng} \times (2^{n2} \text{ cột} \times m \text{ bit})$ '. To the left, a 'Bộ giải mã địa chỉ hàng' (Row address decoder) receives address lines $A_0, A_1, A_2, \dots, A_{n1-1}$ and outputs to the memory array. To the right, a 'Bộ giải mã địa chỉ cột' (Column address decoder) receives address lines $A_{n1}, A_{n1+1}, A_{n1+2}, \dots, A_{n-1}$ and outputs to the memory array. A 'Bộ đệm dữ liệu' (Data buffer) is connected to the memory array and data lines D_0, D_1, \dots, D_{m-1} . At the bottom, a control block 'Điều khiển và định thời gian' (Control and timing) receives CE, WE, and OE signals and provides control to the memory array.

14 January 2013

Kiến trúc máy tính

387

NKK-HUST

Tổ chức bộ nhớ hai chiều

- Có n đường địa chỉ: $n = n_1 + n_2$
 - 2^{n1} hàng,
 - mỗi hàng có 2^{n2} từ nhớ,
- Có m đường dữ liệu:
 - mỗi từ nhớ có độ dài m-bit.
- Dung lượng của chip nhớ:
 $[2^{n1} \times (2^{n2} \times m)] \text{ bit} = (2^{n1+n2} \times m) \text{ bit} = (2^n \times m) \text{ bit}.$
- Hoạt động giải mã địa chỉ:
 - Bước 1: bộ giải mã hàng chọn 1 trong 2^{n1} hàng.
 - Bước 2: bộ giải mã cột chọn 1 trong 2^{n2} từ nhớ (cột) của hàng đã được chọn.

14 January 2013

Kiến trúc máy tính

388

NKK-HUST

Tổ chức của DRAM

The diagram illustrates the internal structure of a DRAM. It features a central memory array labeled 'Ma trận nhớ' with dimensions 2^n hàng x (2^m cột x m bit). This array is connected to a 'Bộ giải mã địa chỉ hàng' (row decoder) and a 'Bộ giải mã địa chỉ cột' (column decoder). The row decoder is controlled by 'Chốt địa chỉ hàng' (row address strobe) and 'RAS'. The column decoder is controlled by 'Chốt địa chỉ cột' (column address strobe) and 'CAS'. A 'Bộ đệm dữ liệu' (data buffer) is connected to the array and provides an output of m bits. The entire system is managed by a 'Điều khiển và định thời gian' (control and timing) block, which receives external signals: CE (Chip Enable), WE (Write Enable), and OE (Output Enable). The address bus consists of n lines for the row address.

14 January 2013

Kiến trúc máy tính

389

NKK-HUST

Tổ chức của DRAM

- Dùng n đường địa chỉ dòng kênh \rightarrow cho phép truyền $2n$ bit địa chỉ
- Tín hiệu chọn địa chỉ hàng RAS (Row Address Select)
- Tín hiệu chọn địa chỉ cột CAS (Column Address Select)
- Dung lượng của DRAM = $2^{2n} \times m$ bit

14 January 2013

Kiến trúc máy tính

390

NKK-HUST

Chip nhớ

Two pin diagrams are shown. Diagram (a) is for an 8 Mbit EPROM, a 32-pin DIP package with a 0.6-inch pitch. It shows address lines A0-A19, data lines D0-D7, and control lines Vcc, Vss, Vpp, CE, RAS, and WE. Diagram (b) is for a 16 Mbit DRAM, a 24-pin package. It shows address lines A0-A10, data lines D0-D7, and control lines Vcc, Vss, CAS, RAS, and WE.

(a) 8 Mbit EPROM

(b) 16 Mbit DRAM

14 January 2013

Kiến trúc máy tính

391

NKK-HUST

Thiết kế mô-đun nhớ bán dẫn

- Dung lượng chip nhớ $2^n \times m$ bit
- Cần thiết kế để tăng dung lượng:
 - Thiết kế tăng độ dài từ nhớ
 - Thiết kế tăng số lượng từ nhớ
 - Thiết kế kết hợp

14 January 2013

Kiến trúc máy tính

392

NKK-HUST

Tăng độ dài từ nhớ

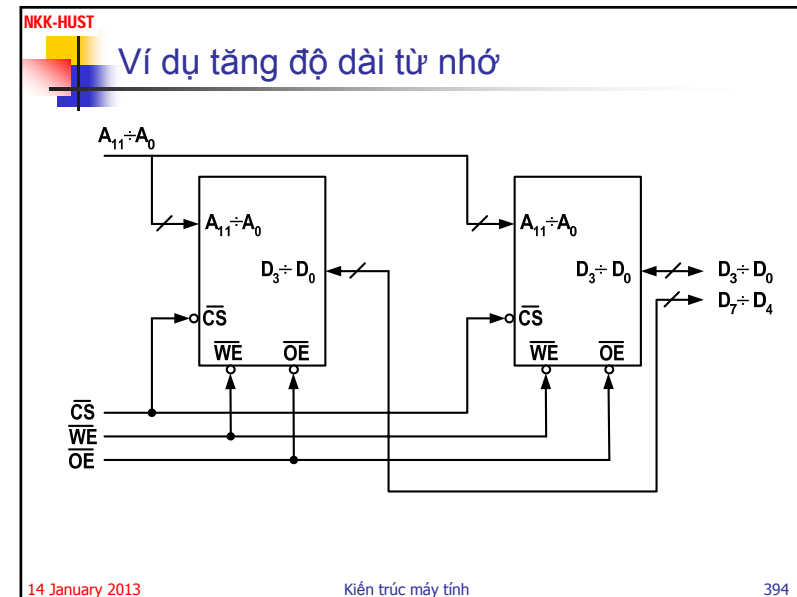
VD1:

- Cho chip nhớ SRAM 4K x 4 bit
- Thiết kế mô-đun nhớ 4K x 8 bit

Giải:

- Dung lượng chip nhớ = $2^{12} \times 4$ bit
- chip nhớ có:
 - 12 chân địa chỉ
 - 4 chân dữ liệu
- mô-đun nhớ cần có:
 - 12 chân địa chỉ
 - 8 chân dữ liệu

14 January 2013 Kiến trúc máy tính 393



NKK-HUST

Bài toán tăng độ dài từ nhớ tổng quát

- Cho chip nhớ $2^n \times \text{mbit}$
- Thiết kế mô-đun nhớ $2^n \times (\text{k.m})$ bit
- Dùng k chip nhớ

14 January 2013 Kiến trúc máy tính 395

NKK-HUST

Tăng số lượng từ nhớ

VD2:

- Cho chip nhớ SRAM 4K x 8 bit
- Thiết kế mô-đun nhớ 8K x 8 bit

Giải:

- Dung lượng chip nhớ = $2^{12} \times 8$ bit
- chip nhớ có:
 - 12 chân địa chỉ
 - 8 chân dữ liệu
- Dung lượng mô-đun nhớ = $2^{13} \times 8$ bit
 - 13 chân địa chỉ
 - 8 chân dữ liệu

14 January 2013 Kiến trúc máy tính 396

NKK-HUST

Tăng số lượng từ nhớ

\overline{G}	A	$\overline{Y0}$	$\overline{Y1}$
0	0	0	1
0	1	1	0
1	x	1	1

14 January 2013

Kiến trúc máy tính

397

NKK-HUST

Bài tập

1. Tăng số lượng từ gấp 4 lần:

- Cho chip nhớ SRAM 4K x 8 bit
- Thiết kế mô-đun nhớ 16K x 8 bit

2. Tăng số lượng từ gấp 8 lần:

- Cho chip nhớ SRAM 4K x 8 bit
- Thiết kế mô-đun nhớ 32K x 8 bit

3. Thiết kế kết hợp:

- Cho chip nhớ SRAM 4K x 4 bit
- Thiết kế mô-đun nhớ 8K x 8 bit

14 January 2013

Kiến trúc máy tính

398

NKK-HUST

Bộ giải mã 2→4

\overline{G}	B	A	$\overline{Y0}$	$\overline{Y1}$	$\overline{Y2}$	$\overline{Y3}$
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0
1	x	x	1	1	1	1

14 January 2013

Kiến trúc máy tính

399

NKK-HUST

Bài tập

- Tăng số lượng từ nhớ gấp 3 lần
- Tăng số lượng từ nhớ gấp 5, 6, 7 lần

14 January 2013

Kiến trúc máy tính

NKK-HUST

2. Các đặc trưng cơ bản của bộ nhớ chính

- Chứa các chương trình đang thực hiện và các dữ liệu đang được sử dụng
- Tồn tại trên mọi hệ thống máy tính
- Bao gồm các ngăn nhớ được đánh địa chỉ trực tiếp bởi CPU
- Dung lượng của bộ nhớ chính nhỏ hơn không gian địa chỉ bộ nhớ mà CPU quản lý.
- Việc quản lý logic bộ nhớ chính tùy thuộc vào hệ điều hành

14 January 2013

Kiến trúc máy tính

401

NKK-HUST

Tổ chức bộ nhớ đan xen (interleaved memory)

- Độ rộng của bus dữ liệu để trao đổi với bộ nhớ: $m = 8, 16, 32, 64, 128 \dots$ bit
- Các ngăn nhớ được tổ chức theo byte
→ tổ chức bộ nhớ vật lý khác nhau

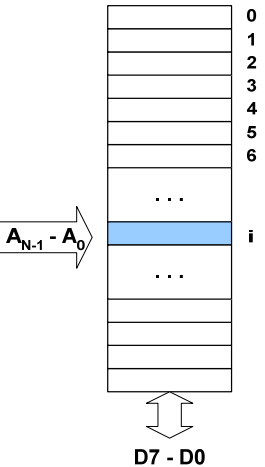
14 January 2013

Kiến trúc máy tính

402

NKK-HUST

$m=8\text{bit} \rightarrow$ một băng nhớ tuyến tính



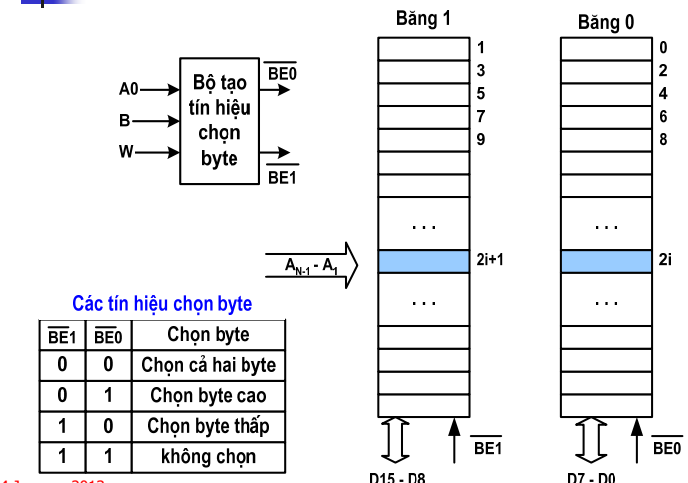
14 January 2013

Kiến trúc máy tính

403

NKK-HUST

$m = 16\text{bit} \rightarrow$ hai băng nhớ đan xen



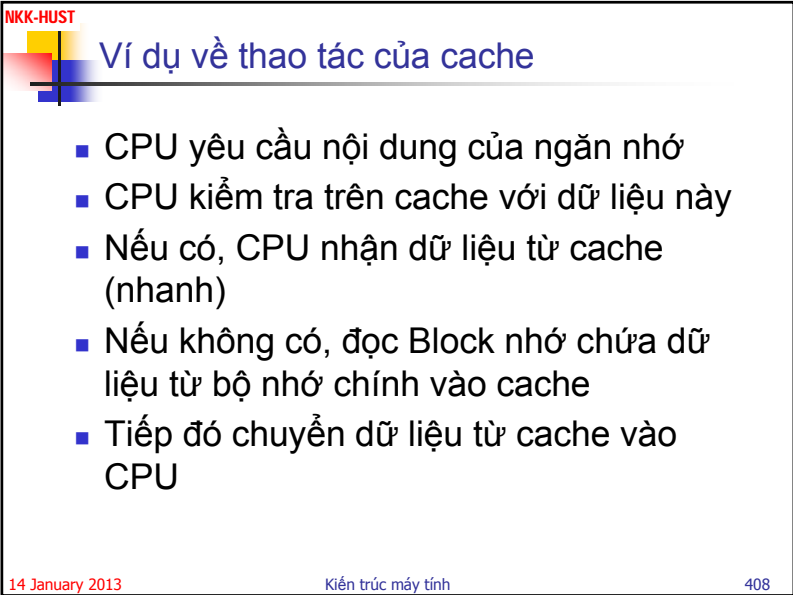
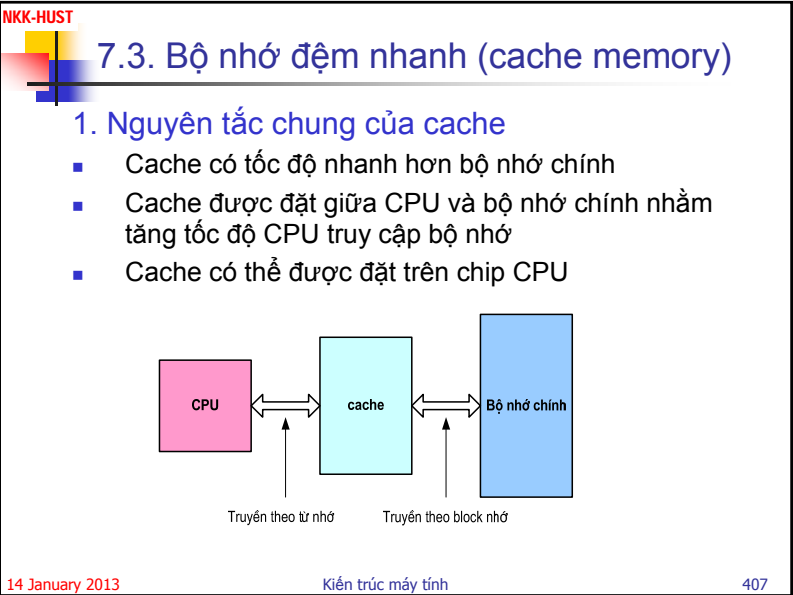
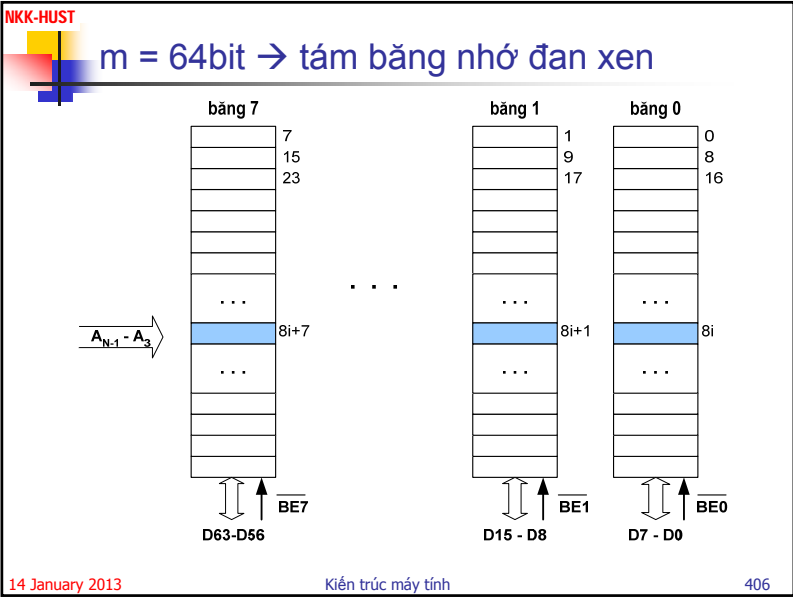
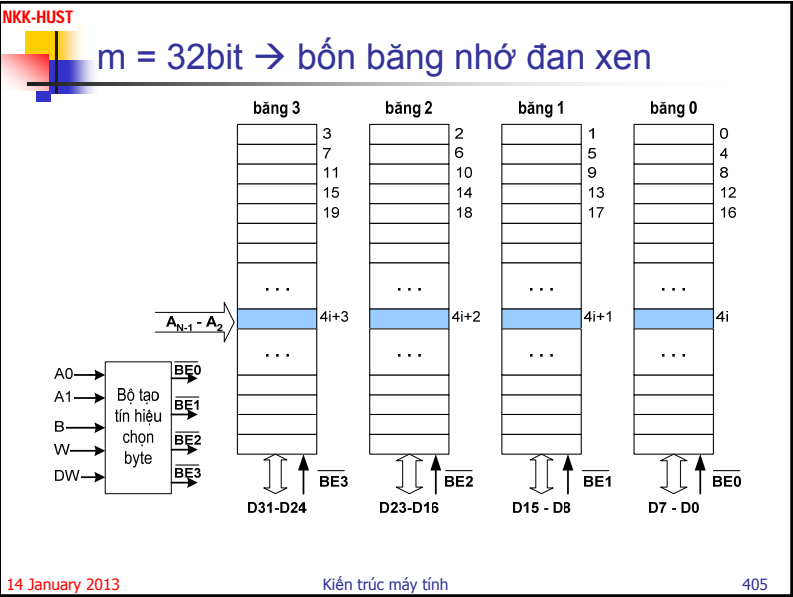
Các tín hiệu chọn byte

BE1	BE0	Chọn byte
0	0	Chọn cả hai byte
0	1	Chọn byte cao
1	0	Chọn byte thấp
1	1	không chọn

14 January 2013

Kiến trúc máy tính

404



NKK-HUST

Cấu trúc chung của cache / bộ nhớ chính

CPU

Tag

Cache

L₀

L₁

L₂

L₃

...

L_i

...

L_{m-1}

Bộ nhớ chính

B₀

B₁

B₂

B₃

...

B_j

...

B_{p-1}

14 January 2013

Kiến trúc máy tính

409

NKK-HUST

Cấu trúc chung của cache / bộ nhớ chính (tiếp)

- Bộ nhớ chính có 2^N byte nhớ
- Bộ nhớ chính và cache được chia thành các khối có kích thước bằng nhau
 - Bộ nhớ chính: $B_0, B_1, B_2, \dots, B_{p-1}$ (p Blocks)
 - Bộ nhớ cache: $L_0, L_1, L_2, \dots, L_{m-1}$ (m Lines)
 - Kích thước của Block = 8,16,32,64,128 byte

14 January 2013

Kiến trúc máy tính

410

NKK-HUST

Cấu trúc chung của cache / bộ nhớ chính (tiếp)

- Một số Block của bộ nhớ chính được nạp vào các Line của cache.
- Nội dung Tag (thẻ nhớ) cho biết Block nào của bộ nhớ chính hiện đang được chứa ở Line đó.
- Khi CPU truy nhập (đọc/ghi) một từ nhớ, có hai khả năng xảy ra:
 - Từ nhớ đó có trong cache (cache hit)
 - Từ nhớ đó không có trong cache (cache miss).

14 January 2013

Kiến trúc máy tính

411

NKK-HUST

2. Các phương pháp ánh xạ

(Chính là các phương pháp tổ chức bộ nhớ cache)

- Ánh xạ trực tiếp
(Direct mapping)
- Ánh xạ liên kết toàn phần
(Fully associative mapping)
- Ánh xạ liên kết tập hợp
(Set associative mapping)

14 January 2013

Kiến trúc máy tính

412

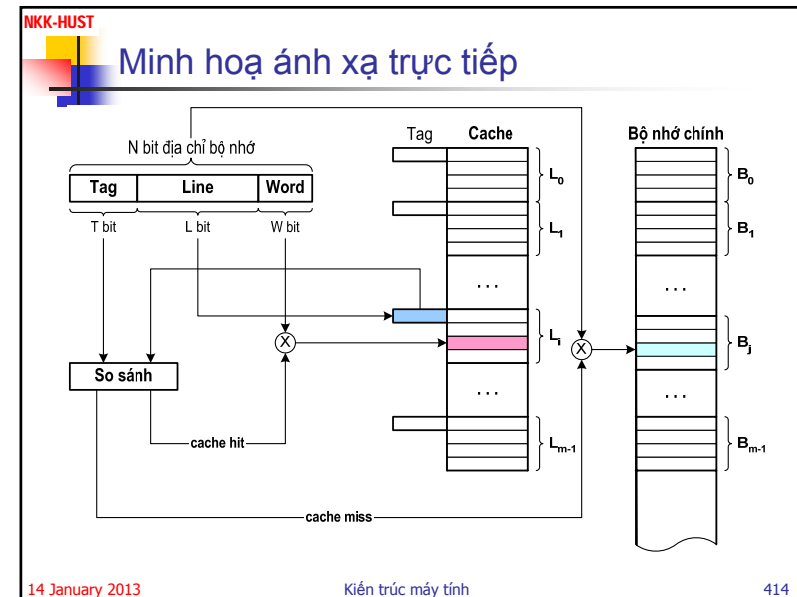
Nguyễn Kim Khánh - ĐHBKHN

NKK-HUST

Ánh xạ trực tiếp

- Mỗi Block của bộ nhớ chính chỉ có thể được nạp vào một Line của cache:
 - $B_0 \rightarrow L_0$
 - $B_1 \rightarrow L_1$
 -
 - $B_{m-1} \rightarrow L_{m-1}$
 - $B_m \rightarrow L_0$
 - $B_{m+1} \rightarrow L_1$
 -
- Tổng quát
 - B_j chỉ có thể nạp vào $L_{j \bmod m}$
 - m là số Line của cache.

14 January 2013 Kiến trúc máy tính 413



NKK-HUST

Đặc điểm của ánh xạ trực tiếp

- Mỗi một địa chỉ N bit của bộ nhớ chính gồm ba trường:
 - Trường Word** gồm W bit xác định một từ nhớ trong Block hay Line:
 $2^W = \text{kích thước của Block hay Line}$
 - Trường Line** gồm L bit xác định một trong số các Line trong cache:
 $2^L = \text{số Line trong cache} = m$
 - Trường Tag** gồm T bit:
 $T = N - (W+L)$
- Bộ so sánh đơn giản
- Xác suất cache hit thấp

14 January 2013 Kiến trúc máy tính 415

NKK-HUST

Ánh xạ liên kết toàn phần

- Mỗi Block có thể nạp vào bất kỳ Line nào của cache.
- Địa chỉ của bộ nhớ chính bao gồm hai trường:
 - Trường Word giống như trường hợp ở trên.
 - Trường Tag dùng để xác định Block của bộ nhớ chính.
- Tag xác định Block đang nằm ở Line đó

14 January 2013 Kiến trúc máy tính 416

NKK-HUST

Minh hoạ ánh xạ liên kết toàn phần

The diagram illustrates full-associative mapping. An N-bit address is split into a Tag (T bit) and Word (W bit). The Tag is compared against all tags in the cache (L₀ to L_{m-1}). If a match is found, the corresponding word is retrieved from the main memory (B₀ to B_{m-1}). If no match is found, it's a cache miss.

14 January 2013

Kiến trúc máy tính

417

NKK-HUST

Đặc điểm của ánh xạ liên kết toàn phần

- So sánh đồng thời với tất cả các Tag → mất nhiều thời gian
- Xác suất *cache hit* cao.
- Bộ so sánh phức tạp.

14 January 2013

Kiến trúc máy tính

418

NKK-HUST

Ánh xạ liên kết tập hợp

- Cache được chia thành các Tập (Set)
- Mỗi một Set chứa một số Line
- Ví dụ:
 - 4 Line/Set → 4-way associative mapping
- Ánh xạ theo nguyên tắc sau:
 - B₀ → S₀
 - B₁ → S₁
 - B₂ → S₂
 -

14 January 2013

Kiến trúc máy tính

419

NKK-HUST

Minh hoạ ánh xạ liên kết tập hợp


The diagram illustrates set-associative mapping. An N-bit address is split into Tag (T bit), Set (S bit), and Word (W bit). The Tag is compared against all tags in the cache (L₀ to L_{m-1}). If a match is found, the corresponding word is retrieved from the main memory (B₀ to B_{m-1}). If no match is found, it's a cache miss.

14 January 2013

Kiến trúc máy tính

420

NKK-HUST



Đặc điểm của ánh xạ liên kết tập hợp


- Kích thước $Block = 2^W$ Word
- Trường Set có S bit dùng để xác định một trong số $V = 2^S$ Set
- Trường Tag có T bit: $T = N - (W+S)$
- Tổng quát cho cả hai phương pháp trên
- Thông thường 2,4,8,16Lines/Set

14 January 2013

Kiến trúc máy tính

421

NKK-HUST



Ví dụ về ánh xạ địa chỉ


- Không gian địa chỉ bộ nhớ chính = 4GB
- Dung lượng bộ nhớ $cache$ là 256KB
- Kích thước $Line$ ($Block$) = 32byte.
- Xác định số bit của các trường địa chỉ cho ba trường hợp tổ chức:
 - Ánh xạ trực tiếp
 - Ánh xạ liên kết toàn phần
 - Ánh xạ liên kết tập hợp 4 đường

14 January 2013

Kiến trúc máy tính

422

NKK-HUST



Với ánh xạ trực tiếp

- Bộ nhớ chính = 4GB = 2^{32} byte $\rightarrow N = 32$ bit
- $Cache = 256$ KB = 2^{18} byte.
- $Line = 32$ byte = 2^5 byte $\rightarrow W = 5$ bit
- Số $Line$ trong $cache = 2^{18} / 2^5 = 2^{13}$ $Line \rightarrow L = 13$ bit
- $T = 32 - (13 + 5) = 14$ bit


Tag	Line	Word
14 bit	13 bit	5 bit

14 January 2013

Kiến trúc máy tính

423

NKK-HUST



Với ánh xạ liên kết toàn phần

- Bộ nhớ chính = 4GB = 2^{32} byte $\rightarrow N = 32$ bit
- $Line = 32$ byte = 2^5 byte $\rightarrow W = 5$ bit
- Số bit của trường Tag sẽ là: $T = 32 - 5 = 27$ bit

Tag	Word
27 bit	5 bit

14 January 2013

Kiến trúc máy tính

424

NKK-HUST

Với ánh xạ liên kết tập hợp 4 đường

- Bộ nhớ chính = 4GB = 2^{32} byte $\rightarrow N = 32$ bit
- Line = 32 byte = 2^5 byte $\rightarrow W = 5$ bit
- Số Line trong cache = $2^{18} / 2^5 = 2^{13}$ Line
- Một Set có 4 Line = 2^2 Line
- \rightarrow số Set trong cache = $2^{13} / 2^2 = 2^{11}$ Set $\rightarrow S = 11$ bit
- Số bit của trường Tag sẽ là: $T = 32 - (11 + 5) = 16$ bit

Tag	Set	Word
16 bit	11 bit	5 bit

14 January 2013 Kiến trúc máy tính 425

NKK-HUST

Bài tập

Giả thiết rằng máy tính có 128KB cache tổ chức theo kiểu ánh xạ liên kết tập hợp 4-line. Cache có tất cả là 1024 Set từ S0 đến S1023. Địa chỉ bộ nhớ chính là 32-bit và đánh địa chỉ cho từng byte.

- Tính số bit cho các trường địa chỉ khi truy nhập cache ?
- Xác định byte nhớ có địa chỉ $003D02AF_{(16)}$ được ánh xạ vào Set nào của cache ?

14 January 2013 Kiến trúc máy tính 426

NKK-HUST

3. Thuật giải thay thế (1): Ánh xạ trực tiếp

- Không phải lựa chọn
- Mỗi Block chỉ ánh xạ vào một Line xác định
- Thay thế Block ở Line đó

14 January 2013 Kiến trúc máy tính 427

NKK-HUST

Thuật giải thay thế (2): Ánh xạ liên kết

- Được thực hiện bằng phần cứng (nhanh)
- Random**: Thay thế ngẫu nhiên
- FIFO** (First In First Out): Thay thế Block nào nằm lâu nhất ở trong Set đó
- LFU** (Least Frequently Used): Thay thế Block nào trong Set có số lần truy nhập ít nhất trong cùng một khoảng thời gian
- LRU** (Least Recently Used): Thay thế Block ở trong Set tương ứng có thời gian lâu nhất không được tham chiếu tới.
- Tối ưu nhất: **LRU**

14 January 2013 Kiến trúc máy tính 428

NKK-HUST

4. Phương pháp ghi dữ liệu khi cache hit

- Ghi xuyên qua (Write-through):
 - ghi cả cache và cả bộ nhớ chính
 - tốc độ chậm
- Ghi trả sau (Write-back):
 - chỉ ghi ra cache
 - tốc độ nhanh
 - khi Block trong cache bị thay thế cần phải ghi trả cả Block về bộ nhớ chính

14 January 2013 Kiến trúc máy tính 429

NKK-HUST

7.4. Bộ nhớ ngoài

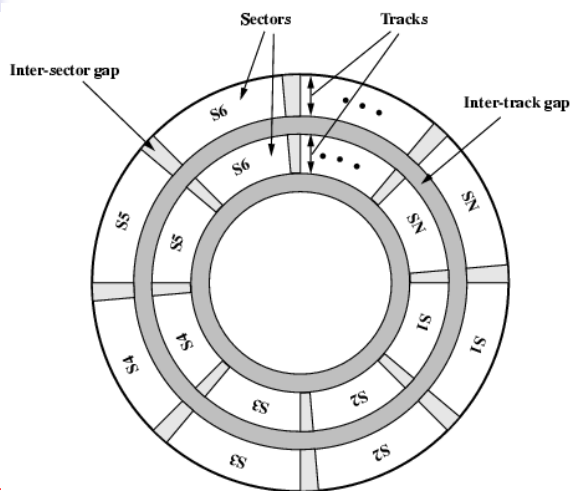
Các kiểu bộ nhớ ngoài

- Băng từ
- Đĩa từ
- Đĩa quang
- Bộ nhớ Flash

14 January 2013 Kiến trúc máy tính 430

NKK-HUST

Đĩa từ



The diagram illustrates the structure of a magnetic disk. It shows concentric circles representing tracks, labeled from S1 to SN. Each track is divided into sectors, labeled from S1 to S6. The gaps between sectors are labeled 'Inter-sector gap', and the gaps between tracks are labeled 'Inter-track gap'. Arrows point to these specific features.

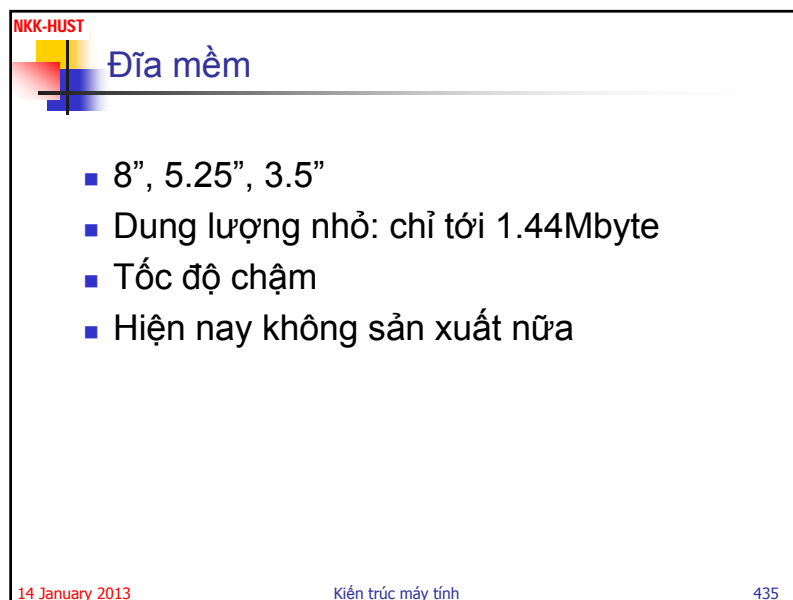
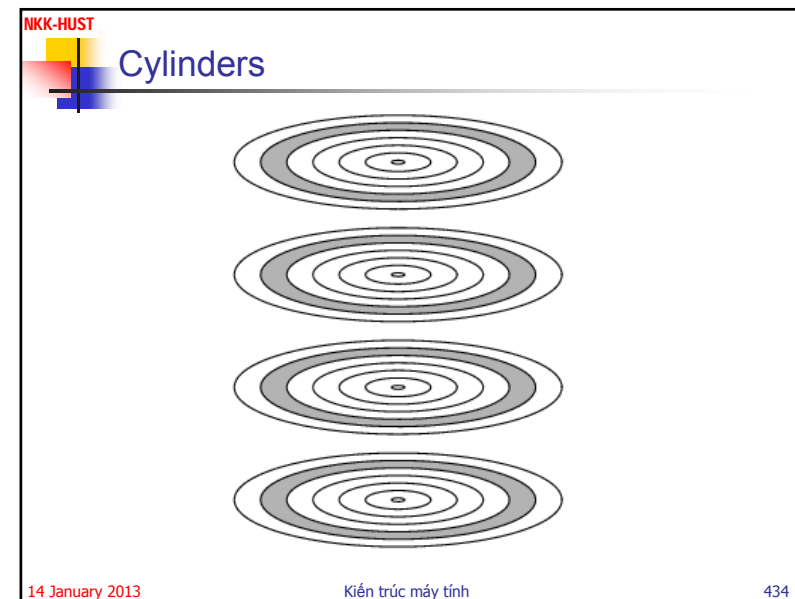
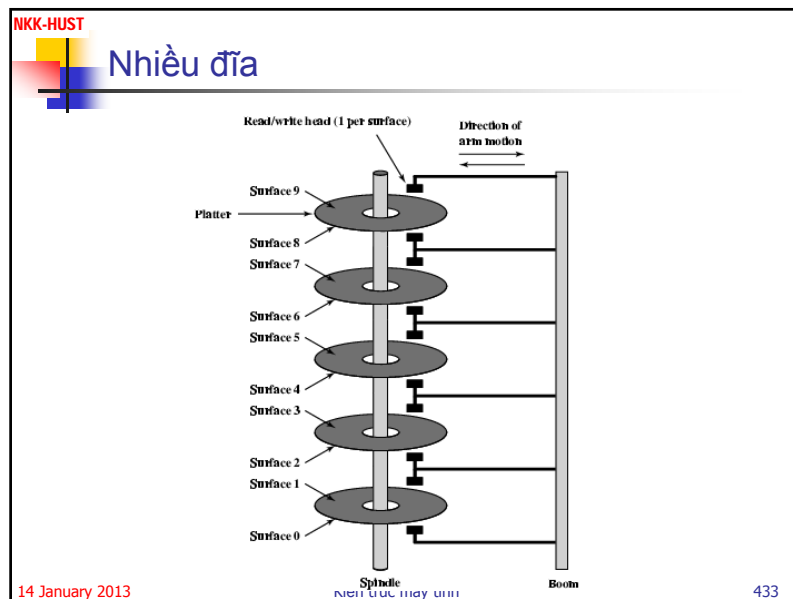
14 January 2013 Kiến trúc máy tính 431

NKK-HUST

Các đặc tính đĩa từ

- Đầu từ cố định hay đầu từ di động
- Đĩa cố định hay thay đổi
- Một mặt hay hai mặt
- Một đĩa hay nhiều đĩa
- Cơ chế đầu từ
 - Tiếp xúc (đĩa mềm)
 - Không tiếp xúc

14 January 2013 Kiến trúc máy tính 432



NKK-HUST

RAID

- Redundant Array of Inexpensive Disks
- Redundant Array of Independent Disks
- Hệ thống nhớ dung lượng lớn

14 January 2013

Kiến trúc máy tính

437

NKK-HUST

Đặc điểm của RAID

- Tập các đĩa cứng vật lý được OS coi như một ổ logic duy nhất → **dung lượng lớn**
- Dữ liệu được lưu trữ phân tán trên các ổ đĩa vật lý → **truy cập song song (nhanh)**
- Có thể sử dụng dung lượng dư thừa để lưu trữ các thông tin kiểm tra chẵn lẻ, cho phép khôi phục lại thông tin trong trường hợp đĩa bị hỏng → **an toàn thông tin**
- 7 loại phổ biến (RAID 0 – 6)

14 January 2013

Kiến trúc máy tính

438

NKK-HUST

RAID 0, 1, 2

strip 0

strip 4

strip 8

strip 12

strip 1

strip 5

strip 9

strip 13

strip 2

strip 6

strip 10

strip 14

strip 3

strip 7

strip 11

strip 15

(a) RAID 0 (non-redundant)

strip 0

strip 4

strip 8

strip 12

strip 1

strip 5

strip 9

strip 13

strip 2

strip 6

strip 10

strip 14

strip 3

strip 7

strip 11

strip 15

strip 4

strip 8

strip 12

strip 16

strip 5

strip 9

strip 13

strip 17

strip 6

strip 10

strip 14

strip 18

strip 7

strip 11

strip 15

strip 19

(b) RAID 1 (mirrored)

b₀

b₁

b₂

b₃

f₀(b)

f₁(b)

f₂(b)

(c) RAID 2 (redundancy through Hamming code)

14 January 2013

Kiến trúc máy tính

439

NKK-HUST

RAID 3 & 4

b₀

b₁

b₂

b₃

P(b)

(d) RAID 3 (bit-interleaved parity)

block 0

block 4

block 8

block 12

block 1

block 5

block 9

block 13

block 2

block 6

block 10

block 14

block 3

block 7

block 11

block 15

P(0-3)

P(4-7)

P(8-11)

P(12-15)

(e) RAID 4 (block-level parity)

14 January 2013

Kiến trúc máy tính

440

Nguyễn Kim Khánh - ĐHBKHN

NKK-HUST

RAID 5 & 6

(f) RAID 5 (block-level distributed parity)

(g) RAID 6 (dual redundancy)

14 January 2013

Kiến trúc máy tính

441

NKK-HUST

Ảnh xạ dữ liệu của RAID 0

14 January 2013

Kiến trúc máy tính

442

NKK-HUST

Đĩa quang

- CD-ROM (Compact Disk ROM)
- CD-R (Recordable CD)
- CD-RW (Rewriteable CD)
- Dung lượng thông dụng 650MB
- Ổ đĩa CD:
 - Ổ CD-ROM
 - Ổ CD-Writer: Ghi một phiên hoặc ghi nhiều phiên
 - Ổ CD-RW
- Tốc độ đọc cơ sở 150KByte/s.
- Tốc độ bội, ví dụ: 48x, 52x,...

14 January 2013

Kiến trúc máy tính

443

NKK-HUST

Đĩa quang (tiếp)

- DVD
 - Digital Video Disk: chỉ dùng trên ổ đĩa xem video
 - Digital Versatile Disk: ổ trên máy tính
 - Ghi một hoặc hai mặt
 - Một hoặc hai lớp trên một mặt
 - Thông dụng: 4,7GB/lớp


14 January 2013

Kiến trúc máy tính

444

Bộ nhớ flash

- Bộ nhớ bán dẫn (flash memory)
- Tốc độ nhanh
- Các dạng:
 - Ổ nhớ kết nối qua cổng USB
 - Thẻ nhớ
 - Ổ SSD (Solid State Drive): kết nối nhiều chip nhớ flash và cho phép truy cập song song



14 January 2013 Kiến trúc máy tính 445

7.5. Bộ nhớ ảo (Virtual Memory)

- Khái niệm bộ nhớ ảo: gồm bộ nhớ chính và bộ nhớ ngoài mà được CPU coi như là một bộ nhớ duy nhất (bộ nhớ chính).
- Các kỹ thuật thực hiện bộ nhớ ảo:
 - Kỹ thuật phân trang: Chia không gian địa chỉ bộ nhớ thành các trang nhớ có kích thước bằng nhau và nằm liền kề nhau
Thông dụng: kích thước trang = 4KBytes
 - Kỹ thuật phân đoạn: Chia không gian nhớ thành các đoạn nhớ có kích thước thay đổi, các đoạn nhớ có thể gộp lên nhau.

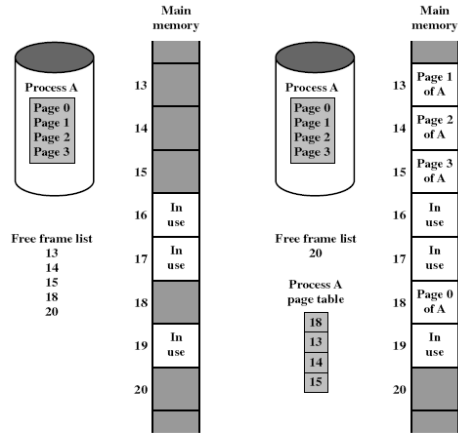
14 January 2013 Kiến trúc máy tính 446

Phân trang

- Phân chia bộ nhớ thành các phần có kích thước bằng nhau gọi là các khung trang
- Chia chương trình (tiến trình) thành các trang
- Cấp phát số hiệu khung trang yêu cầu cho tiến trình
- HĐH duy trì danh sách các khung trang nhớ trống
- Tiến trình không yêu cầu các khung trang liên tiếp
- Sử dụng bảng trang để quản lý

14 January 2013 Kiến trúc máy tính 447

Cấp phát các khung trang



(a) Before (b) After

14 January 2013 Kiến trúc máy tính 448

NKK-HUST

Địa chỉ logic và địa chỉ vật lý của phân trang

The diagram shows the mapping of logical addresses to physical addresses. On the left, a 'Logical address' is shown with a 'Page number' of 1 and a 'Relative address within page' of 30. This points to the 'Process A page table', which contains entries for pages 13, 14, and 15. The entry for page 13 points to a 'Physical address' with a 'Frame number' of 13 and a 'Relative address within frame' of 30. This physical address then points to 'Main memory', specifically to 'Page 1 of A' at address 13. The main memory is shown as a vertical stack of pages: Page 1 of A (13), Page 2 of A (14), Page 3 of A (15), and Page 0 of A (18).

14 January 2013

Kiến trúc máy tính

449

NKK-HUST

Nguyên tắc làm việc của bộ nhớ ảo phân trang

- Phân trang theo yêu cầu
 - Không yêu cầu tất cả các trang của tiến trình nằm trong bộ nhớ
 - Chỉ nạp vào bộ nhớ những trang được yêu cầu
- Lỗi trang
 - Trang được yêu cầu không có trong bộ nhớ
 - HĐH cần hoán đổi trang yêu cầu vào
 - Có thể cần hoán đổi một trang nào đó ra để lấy chỗ
 - Cần chọn trang để đưa ra

14 January 2013

Kiến trúc máy tính

450

NKK-HUST

Thất bại

- Quá nhiều tiến trình trong bộ nhớ quá nhỏ
- HĐH tiêu tốn toàn bộ thời gian cho việc hoán đổi
- Có ít hoặc không có công việc nào được thực hiện
- Đĩa luôn luôn sáng
- Giải pháp:
 - Thuật toán thay trang
 - Giảm bớt số tiến trình đang chạy
 - Thêm bộ nhớ

14 January 2013

Kiến trúc máy tính

451

NKK-HUST

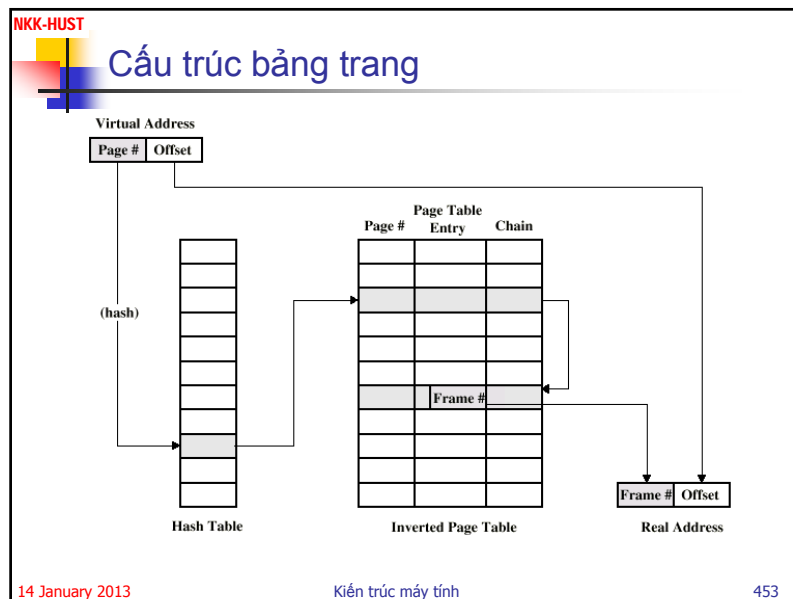
Lợi ích

- Không cần toàn bộ tiến trình nằm trong bộ nhớ để chạy
- Có thể hoán đổi trang được yêu cầu
- Như vậy có thể chạy những tiến trình lớn hơn tổng bộ nhớ sẵn dùng
- Bộ nhớ chính được gọi là bộ nhớ thực
- Người dùng cảm giác bộ nhớ lớn hơn bộ nhớ thực

14 January 2013

Kiến trúc máy tính

452



- NKK-HUST**
- ### Minh họa: bộ nhớ trên PC hiện nay
- Bộ nhớ cache: tích hợp trên chip vi xử lý:
 - L1: cache lệnh và cache dữ liệu
 - L2, L3
 - Bộ nhớ chính: Tồn tại dưới dạng các mô-đun nhớ RAM
 - SIMM – Single Inline Memory Module
 - DIMM – Dual Inline Memory Module
- 14 January 2013 Kiến trúc máy tính 454

- NKK-HUST**
- ### Hệ thống nhớ trên PC (tiếp)
- ROM BIOS chứa các chương trình sau:
 - Chương trình POST (Power On Self Test)
 - Chương trình CMOS Setup
 - Chương trình Bootstrap loader
 - Các trình điều khiển vào-ra cơ bản (BIOS)
 - CMOS RAM:
 - Chứa thông tin cấu hình hệ thống
 - Đồng hồ hệ thống
 - Có pin nuôi riêng
 - Video RAM: quản lý thông tin của màn hình
 - Các loại bộ nhớ ngoài
- 14 January 2013 Kiến trúc máy tính 455

NKK-HUST

Hết chương 7

14 January 2013 Kiến trúc máy tính 456

NKK-HUST

Kiến trúc máy tính

Chương 8

VÀO-RA

Nguyễn Kim Khánh
Trường Đại học Bách khoa Hà Nội

14 January 2013 Kiến trúc máy tính 457

NKK-HUST

Nội dung học phần

- Chương 1. Giới thiệu chung
- Chương 2. Cơ bản về logic số
- Chương 3. Hệ thống máy tính
- Chương 4. Số học máy tính
- Chương 5. Kiến trúc tập lệnh
- Chương 6. Cấu trúc bộ xử lý
- Chương 7. Bộ nhớ
- **Chương 8. Vào-ra**
- Chương 9. Các kiến trúc song song

14 January 2013 Kiến trúc máy tính 458

NKK-HUST

Nội dung của chương 8

- 8.1. Tổng quan về vào-ra
- 8.2. Các phương pháp điều khiển vào-ra
- 8.3. Nối ghép thiết bị ngoại vi
- 8.4. Các cổng vào-ra thông dụng trên PC

14 January 2013 Kiến trúc máy tính 459

NKK-HUST

8.1. Tổng quan về vào-ra

1. Giới thiệu chung

- Chức năng của vào-ra: Trao đổi thông tin giữa máy tính với thế giới bên ngoài
- Các thao tác cơ bản:
 - Vào dữ liệu (Input)
 - Ra dữ liệu (Output)
- Các thành phần chính:
 - Các thiết bị ngoại vi
 - Các mô-đun vào-ra

14 January 2013 Kiến trúc máy tính 460

NKK-HUST

Cấu trúc cơ bản của vào-ra

14 January 2013

Kiến trúc máy tính

461

NKK-HUST

Đặc điểm của vào-ra

- Tồn tại đa dạng các thiết bị ngoại vi khác nhau về:
 - Nguyên tắc hoạt động
 - Tốc độ
 - Khuôn dạng dữ liệu
- Tất cả các thiết bị ngoại vi đều chậm hơn CPU và RAM

→ Cần có các mô-đun vào-ra để nối ghép các thiết bị ngoại vi với CPU và bộ nhớ chính

14 January 2013

Kiến trúc máy tính

462

NKK-HUST

2. Các thiết bị ngoại vi

- Chức năng: chuyển đổi dữ liệu giữa bên trong và bên ngoài máy tính
- Phân loại:
 - Thiết bị ngoại vi giao tiếp người-máy: Bàn phím, Màn hình, Máy in,...
 - Thiết bị ngoại vi giao tiếp máy-máy: gồm các thiết bị theo dõi và kiểm tra
 - Thiết bị ngoại vi truyền thông: Modem, Network Interface Card (NIC)

14 January 2013

Kiến trúc máy tính

463

NKK-HUST

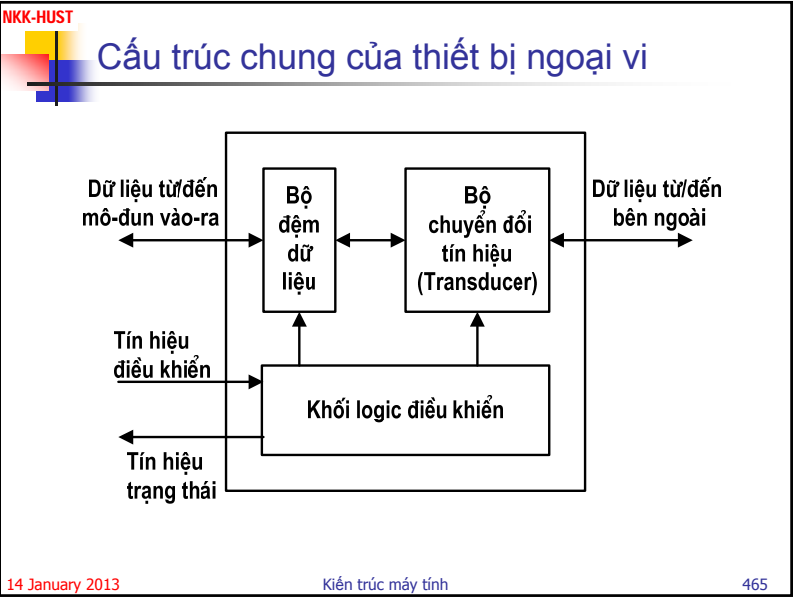
Một số thiết bị ngoại vi

Input type	Prime examples	Other examples	Data rate (b/s)	Main uses
Symbol	Keyboard, keypad	Music note, OCR	10s	Ubiquitous
Position	Mouse, touchpad	Stick, wheel, glove	100s	Ubiquitous
Identity	Barcode reader	Badge, fingerprint	100s	Sales, security
Sensory	Touch, motion, light	Scent, brain signal	100s	Control, security
Audio	Microphone	Phone, radio, tape	1000s	Ubiquitous
Image	Scanner, camera	Graphic tablet	1000s-10 ⁶ s	Photos, publishing
Video	Camcorder, DVD	VCR, TV cable	1000s-10 ⁹ s	Entertainment
Output type	Prime examples	Other examples	Data rate (b/s)	Main uses
Symbol	LCD line segments	LED, status light	10s	Ubiquitous
Position	Stepper motor	Robotic motion	100s	Ubiquitous
Warning	Buzzer, bell, siren	Flashing light	A few	Safety, security
Sensory	Braille text	Scent, brain stimulus	100s	Personal assistance
Audio	Speaker, audiotape	Voice synthesizer	1000s	Ubiquitous
Image	Monitor, printer	Plotter, microfilm	1000s	Ubiquitous
Video	Monitor, TV screen	Film/video recorder	1000s-10 ⁹ s	Entertainment
Two-way I/O	Prime examples	Other examples	Data rate (b/s)	Main uses
Mass storage	Hard/floppy disk	CD, tape, archive	10 ⁶ s	Ubiquitous
Network	Modem, fax, LAN	Cable, DSL, ATM	1000s-10 ⁹ s	Ubiquitous

14 January 2013

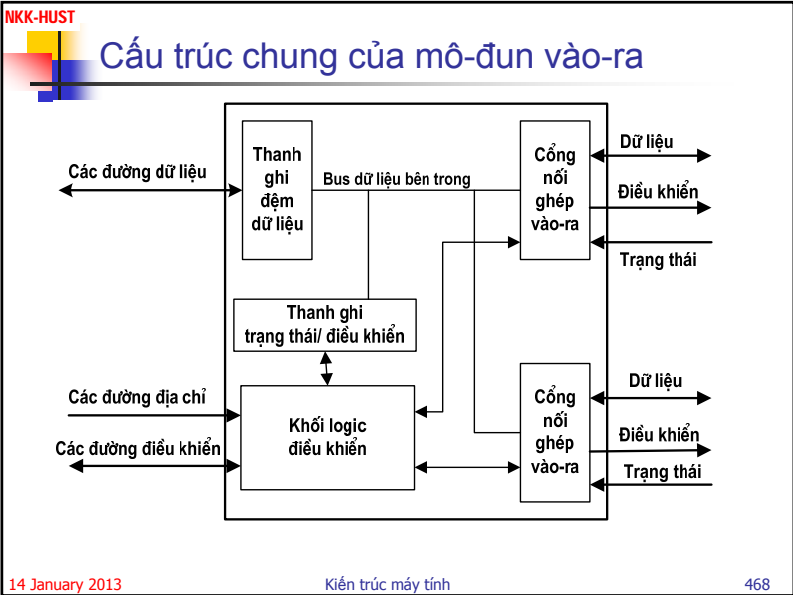
Kiến trúc máy tính

464



- Các thành phần của thiết bị ngoại vi**
- Bộ chuyển đổi tín hiệu: chuyển đổi dữ liệu giữa bên ngoài và bên trong máy tính
 - Bộ đệm dữ liệu: đệm dữ liệu khi truyền giữa mô-đun vào-ra và thiết bị ngoại vi
 - Khối logic điều khiển: điều khiển hoạt động của thiết bị ngoại vi đáp ứng theo yêu cầu từ mô-đun vào-ra

- 3. Mô-đun vào-ra**
- Chức năng của mô-đun vào-ra:
 - Điều khiển và định thời
 - Trao đổi thông tin với CPU hoặc bộ nhớ chính
 - Trao đổi thông tin với thiết bị ngoại vi
 - Đệm giữa bên trong máy tính với thiết bị ngoại vi
 - Phát hiện lỗi của thiết bị ngoại vi



469

470

471

472

NKK-HUST

Các phương pháp địa chỉ hoá cổng vào-ra

- Vào-ra riêng biệt
(Isolated IO hay IO mapped IO)
- Vào-ra theo bản đồ bộ nhớ
(Memory mapped IO)

14 January 2013 Kiến trúc máy tính 473

NKK-HUST

Vào-ra riêng biệt

- Cổng vào-ra được đánh địa chỉ theo không gian địa chỉ vào-ra
- CPU trao đổi dữ liệu với cổng vào-ra thông qua các lệnh vào-ra chuyên dụng (IN, OUT)
- Chỉ có thể thực hiện trên các hệ thống có quản lý không gian địa chỉ vào-ra riêng biệt

14 January 2013 Kiến trúc máy tính 474

NKK-HUST

Vào-ra theo bản đồ bộ nhớ

- Cổng vào-ra được đánh địa chỉ theo không gian địa chỉ bộ nhớ
- Vào-ra giống như đọc/ghi bộ nhớ
- CPU trao đổi dữ liệu với cổng vào-ra thông qua các lệnh truy nhập dữ liệu bộ nhớ
- Có thể thực hiện trên mọi hệ thống

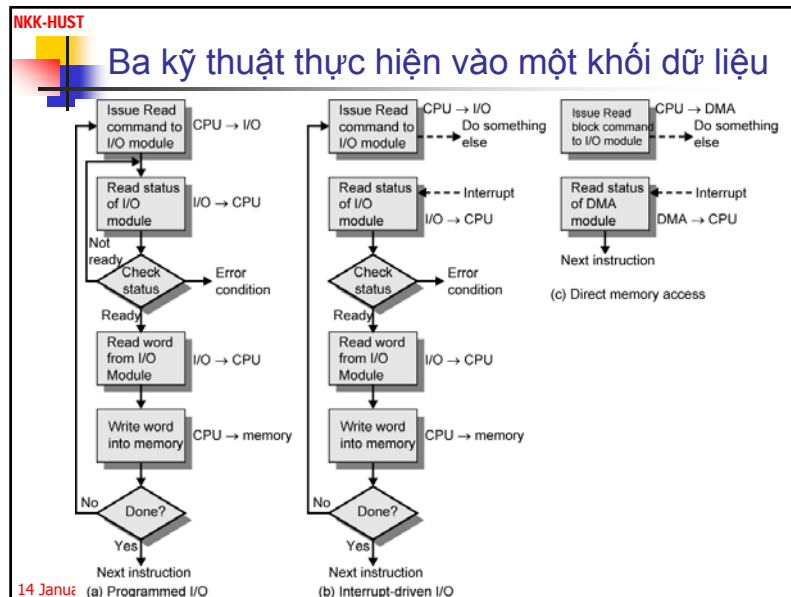
14 January 2013 Kiến trúc máy tính 475

NKK-HUST

8.2. Các phương pháp điều khiển vào-ra

- Vào-ra bằng chương trình
(Programmed IO)
- Vào-ra điều khiển bằng ngắt
(Interrupt Driven IO)
- Truy nhập bộ nhớ trực tiếp - DMA
(Direct Memory Access)

14 January 2013 Kiến trúc máy tính 476



NKK-HUST

1. Vào-ra bằng chương trình

- Nguyên tắc chung: CPU điều khiển trực tiếp vào-ra bằng chương trình → cần phải lập trình vào-ra.

14 January 2013 Kiến trúc máy tính 478

NKK-HUST

Các tín hiệu điều khiển vào-ra

- Tín hiệu **điều khiển** (*Control*): kích hoạt thiết bị ngoại vi
- Tín hiệu **kiểm tra** (*Test*): kiểm tra trạng thái của mô-đun vào-ra và thiết bị ngoại vi
- Tín hiệu điều khiển **đọc** (*Read*): yêu cầu mô-đun vào-ra nhận dữ liệu từ thiết bị ngoại vi và đưa vào thanh ghi đệm dữ liệu, rồi CPU nhận dữ liệu đó
- Tín hiệu điều khiển **ghi** (*Write*): yêu cầu mô-đun vào-ra lấy dữ liệu trên bus dữ liệu đưa đến thanh ghi đệm dữ liệu rồi chuyển ra thiết bị ngoại vi

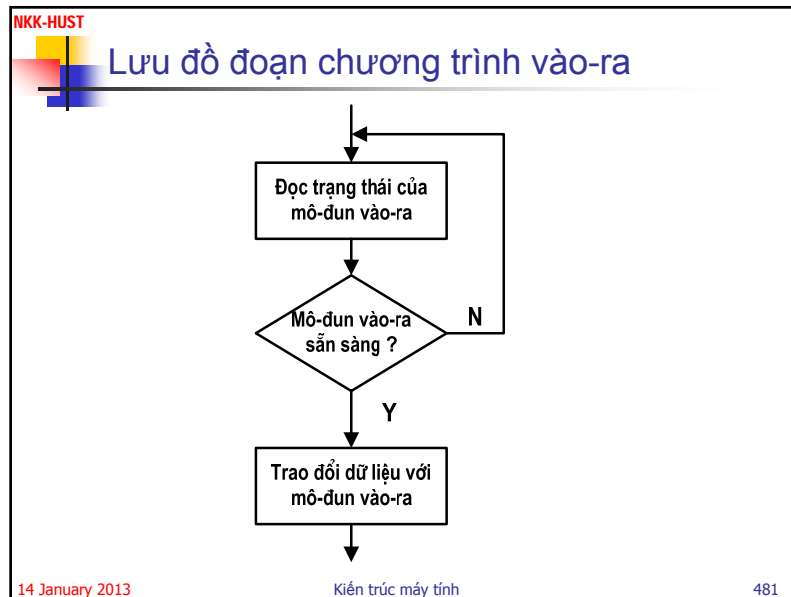
14 January 2013 Kiến trúc máy tính 479

NKK-HUST

Các lệnh vào-ra

- Với vào-ra riêng biệt: sử dụng các lệnh vào-ra chuyên dụng (IN, OUT).
- Với vào-ra theo bản đồ bộ nhớ: sử dụng các lệnh trao đổi dữ liệu với bộ nhớ để trao đổi dữ liệu với cổng vào-ra.

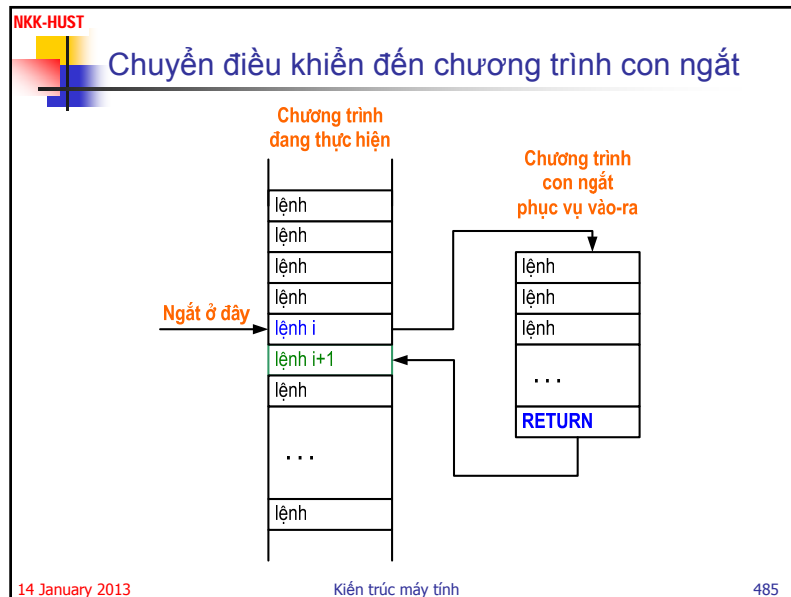
14 January 2013 Kiến trúc máy tính 480



- NKK-HUST
- ### Hoạt động của vào-ra bằng chương trình
- CPU yêu cầu thao tác vào-ra
 - Mô-đun vào-ra thực hiện thao tác
 - Mô-đun vào-ra thiết lập các bit trạng thái
 - CPU kiểm tra các bit trạng thái:
 - Nếu chưa sẵn sàng thì quay lại kiểm tra
 - Nếu sẵn sàng thì chuyển sang trao đổi dữ liệu với mô-đun vào-ra
- 14 January 2013
- Kiến trúc máy tính
- 482

- NKK-HUST
- ### Đặc điểm
- Vào-ra do ý muốn của người lập trình
 - CPU trực tiếp điều khiển vào-ra
 - CPU đợi mô-đun vào-ra → tiêu tốn thời gian của CPU
- 14 January 2013
- Kiến trúc máy tính
- 483

- NKK-HUST
- ### 2. Vào-ra điều khiển bằng ngắt
- Nguyên tắc chung:
 - CPU không phải đợi trạng thái sẵn sàng của mô-đun vào-ra, CPU thực hiện một chương trình nào đó
 - Khi mô-đun vào-ra sẵn sàng thì nó phát tín hiệu ngắt CPU
 - CPU thực hiện chương trình con vào-ra tương ứng để trao đổi dữ liệu
 - CPU trở lại tiếp tục thực hiện chương trình đang bị ngắt
- 14 January 2013
- Kiến trúc máy tính
- 484



- NKK-HUST
- ### Hoạt động vào dữ liệu: nhìn từ mô-đun vào-ra
- Mô-đun vào-ra nhận tín hiệu điều khiển **đọc** từ CPU
 - Mô-đun vào-ra nhận dữ liệu từ thiết bị ngoại vi, trong khi đó CPU làm việc khác
 - Khi đã có dữ liệu → mô-đun vào-ra phát tín hiệu ngắt CPU
 - CPU yêu cầu dữ liệu
 - Mô-đun vào-ra chuyển dữ liệu đến CPU
- 14 January 2013
- Kiến trúc máy tính
- 486

- NKK-HUST
- ### Hoạt động vào dữ liệu: nhìn từ CPU
- Phát tín hiệu điều khiển **đọc**
 - Làm việc khác
 - Cuối mỗi chu trình lệnh, kiểm tra tín hiệu ngắt
 - Nếu bị ngắt:
 - Cắt ngữ cảnh (nội dung các thanh ghi)
 - Thực hiện chương trình con ngắt để vào dữ liệu
 - Khôi phục ngữ cảnh của chương trình đang thực hiện
- 14 January 2013
- Kiến trúc máy tính
- 487

- NKK-HUST
- ### Các vấn đề nảy sinh khi thiết kế
- Làm thế nào để xác định được mô-đun vào-ra nào phát tín hiệu ngắt ?
 - CPU làm như thế nào khi có nhiều yêu cầu ngắt cùng xảy ra ?
- 14 January 2013
- Kiến trúc máy tính
- 488

NKK-HUST

Các phương pháp nối ghép ngắt

- Sử dụng nhiều đường yêu cầu ngắt
- Hỏi vòng bằng phần mềm (Software Poll)
- Hỏi vòng bằng phần cứng (Daisy Chain or Hardware Poll)
- Sử dụng bộ điều khiển ngắt (PIC)

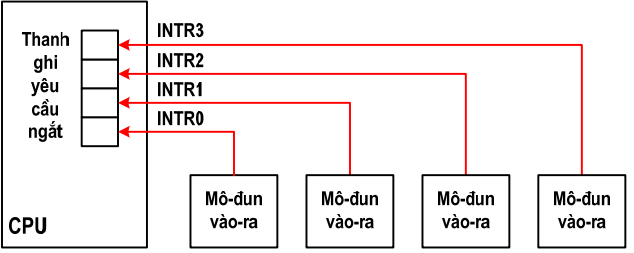
14 January 2013

Kiến trúc máy tính

489

NKK-HUST

Nhiều đường yêu cầu ngắt



The diagram illustrates a CPU with four dedicated interrupt request lines labeled INTR0, INTR1, INTR2, and INTR3. These lines are connected to four separate input/output modules, each labeled 'Mô-đun vào-ra'. The CPU side of the lines is grouped under the label 'Thanh ghi yêu cầu ngắt' (Interrupt Request Register).

- Mỗi mô-đun vào-ra được nối với một đường yêu cầu ngắt
- CPU phải có nhiều đường tín hiệu yêu cầu ngắt
- Hạn chế số lượng mô-đun vào-ra
- Các đường ngắt được qui định mức ưu tiên

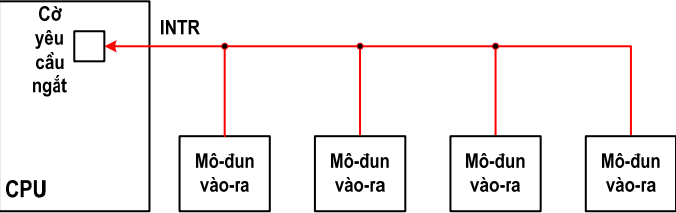
14 January 2013

Kiến trúc máy tính

490

NKK-HUST

Hỏi vòng bằng phần mềm



The diagram shows a CPU with a single interrupt request line labeled INTR. This line is connected to four input/output modules, each labeled 'Mô-đun vào-ra'. On the CPU side, there is a component labeled 'Cờ yêu cầu ngắt' (Interrupt Flag).

- CPU thực hiện phần mềm hỏi lần lượt từng mô-đun vào-ra
- Chậm
- Thứ tự các mô-đun được hỏi vòng chính là thứ tự ưu tiên

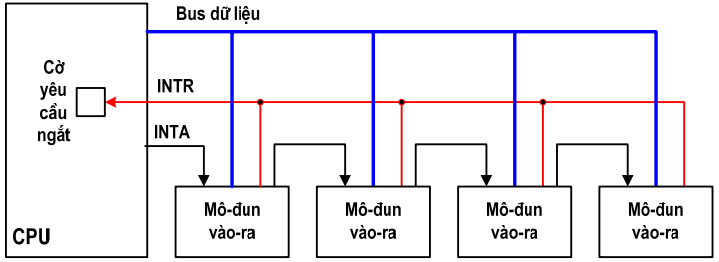
14 January 2013

Kiến trúc máy tính

491

NKK-HUST

Hỏi vòng bằng phần cứng



The diagram illustrates a hardware polling mechanism. A CPU is connected to four input/output modules via a single interrupt request line (INTR) and an interrupt acknowledge line (INTA). A data bus, labeled 'Bus dữ liệu', is also shown connecting the CPU to the modules.

14 January 2013

Kiến trúc máy tính

492

NKK-HUST

Kiểm tra vòng bằng phần cứng (tiếp)

- CPU phát tín hiệu chấp nhận ngắt (INTA) đến mô-đun vào-ra đầu tiên
- Nếu mô-đun vào-ra đó không gây ra ngắt thì nó gửi tín hiệu đến mô-đun kế tiếp cho đến khi xác định được mô-đun gây ngắt
- Thứ tự các mô-đun vào-ra kết nối trong chuỗi xác định thứ tự ưu tiên

14 January 2013 Kiến trúc máy tính 493

NKK-HUST

Bộ điều khiển ngắt lập trình được

- PIC – Programmable Interrupt Controller
- PIC có nhiều đường vào yêu cầu ngắt có qui định mức ưu tiên
- PIC chọn một yêu cầu ngắt không bị cấm có mức ưu tiên cao nhất gửi tới CPU

14 January 2013 Kiến trúc máy tính 494

NKK-HUST

Đặc điểm của vào-ra điều khiển bằng ngắt

- Có sự kết hợp giữa phần cứng và phần mềm
 - Phần cứng: gây ngắt CPU
 - Phần mềm: trao đổi dữ liệu
- CPU trực tiếp điều khiển vào-ra
- CPU không phải đợi mô-đun vào-ra → hiệu quả sử dụng CPU tốt hơn

14 January 2013 Kiến trúc máy tính 495

NKK-HUST

Ngắt của 80x86

- Tổ chức kiểu vector ngắt
- Số hiệu ngắt: n (00-FF)
- Bảng vector ngắt: $256 \times 4 \text{ byte} = 1024 \text{ bytes}$ 00000 – 003FF
- Lệnh INT n

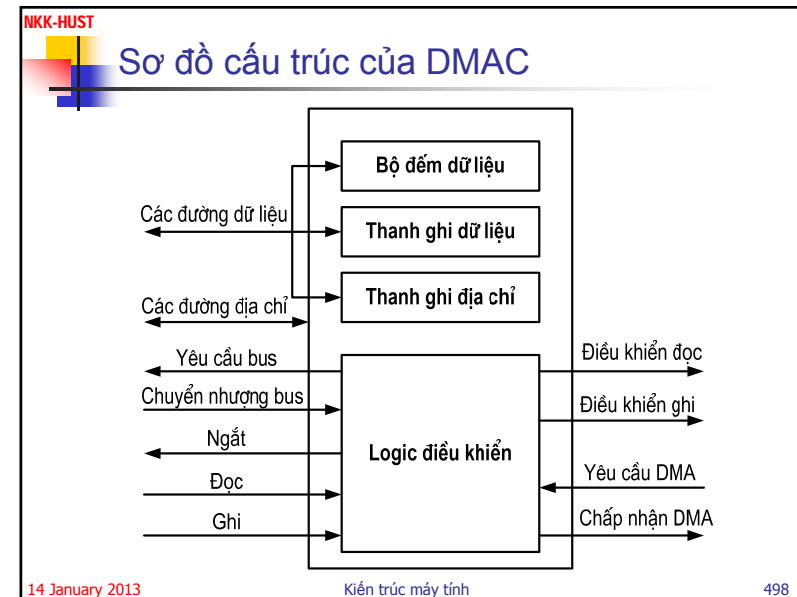
14 January 2013 Kiến trúc máy tính 496

NKK-HUST

3. DMA (Direct Memory Access)

- Vào-ra bằng chương trình và bằng ngắt do CPU trực tiếp điều khiển:
 - Chiếm thời gian của CPU
 - Tốc độ truyền bị hạn chế vì phải chuyển qua CPU
- Để khắc phục dùng DMA
 - Thêm mô-đun phần cứng trên bus → DMAC (Controller)
 - DMAC điều khiển trao đổi dữ liệu giữa mô-đun vào-ra với bộ nhớ chính

14 January 2013 Kiến trúc máy tính 497



NKK-HUST

Các thành phần của DMAC

- Thanh ghi dữ liệu: chứa dữ liệu trao đổi
- Thanh ghi địa chỉ: chứa địa chỉ ngăn nhớ dữ liệu
- Bộ đếm dữ liệu: chứa số từ dữ liệu cần trao đổi
- Logic điều khiển: điều khiển hoạt động của DMAC

14 January 2013 Kiến trúc máy tính 499

NKK-HUST

Hoạt động DMA

- CPU “nói” cho DMAC
 - Vào hay Ra dữ liệu
 - Địa chỉ thiết bị vào-ra (cổng vào-ra tương ứng)
 - Địa chỉ đầu của mảng nhớ chứa dữ liệu → nạp vào thanh ghi địa chỉ
 - Số từ dữ liệu cần truyền → nạp vào bộ đếm dữ liệu
- CPU làm việc khác
- DMAC điều khiển trao đổi dữ liệu
- Sau khi truyền được một từ dữ liệu thì:
 - nội dung thanh ghi địa chỉ tăng
 - nội dung bộ đếm dữ liệu giảm
- Khi bộ đếm dữ liệu = 0, DMAC gửi tín hiệu ngắt CPU để báo kết thúc DMA

14 January 2013 Kiến trúc máy tính 500

NKK-HUST

Các kiểu thực hiện DMA

- DMA truyền theo khối (Block-transfer DMA): DMAC sử dụng bus để truyền xong cả khối dữ liệu
- DMA lấy chu kỳ (Cycle Stealing DMA): DMAC cưỡng bức CPU treo tạm thời từng chu kỳ bus, DMAC chiếm bus thực hiện truyền một từ dữ liệu.
- DMA trong suốt (Transparent DMA): DMAC nhận biết những chu kỳ nào CPU không sử dụng bus thì chiếm bus để trao đổi một từ dữ liệu.

14 January 2013

Kiến trúc máy tính

501

NKK-HUST

Cấu hình DMA (1)

The diagram shows a horizontal line representing the System Bus. Below this line, five boxes are connected: CPU, DMAC, I/O Module, I/O Module, and Memory. There are ellipses between the two I/O Module boxes, indicating more than two modules.

- Mỗi lần trao đổi một dữ liệu, DMAC sử dụng bus hai lần
 - Giữa mô-đun vào-ra với DMAC
 - Giữa DMAC với bộ nhớ

14 January 2013

Kiến trúc máy tính

502

NKK-HUST

Cấu hình DMA (2)

The diagram shows a horizontal line representing the System Bus. Below this line, four boxes are connected: CPU, DMAC, DMAC, and Memory. The second DMAC box is connected to two I/O Module boxes below it. There are ellipses between the two I/O Module boxes, indicating more than two modules.

- DMAC điều khiển một hoặc vài mô-đun vào-ra
- Mỗi lần trao đổi một dữ liệu, DMAC sử dụng bus một lần
 - Giữa DMAC với bộ nhớ

14 January 2013

Kiến trúc máy tính

503

NKK-HUST

Cấu hình DMA (3)

The diagram shows a horizontal line representing the System Bus. Below this line, three boxes are connected: CPU, DMAC, and Memory. Below the DMAC box, there is another horizontal line representing the IO Bus. Below this IO Bus, three boxes are connected: I/O Module, I/O Module, and I/O Module. There are ellipses between the second and third I/O Module boxes, indicating more than three modules.

- Bus vào-ra tách rời hỗ trợ tất cả các thiết bị cho phép DMA
- Mỗi lần trao đổi một dữ liệu, DMAC sử dụng bus một lần
 - Giữa DMAC với bộ nhớ

14 January 2013

Kiến trúc máy tính

504

NKK-HUST

Đặc điểm của DMA

- CPU không tham gia trong quá trình trao đổi dữ liệu
- DMAC điều khiển trao đổi dữ liệu giữa bộ nhớ chính với mô-đun vào-ra (hoàn toàn bằng phần cứng) → tốc độ nhanh
- Phù hợp với các yêu cầu trao đổi mảng dữ liệu có kích thước lớn

14 January 2013 Kiến trúc máy tính 505

NKK-HUST

4. Kênh vào-ra hay là bộ xử lý vào-ra

- Việc điều khiển vào-ra được thực hiện bởi một bộ xử lý vào-ra chuyên dụng
- Bộ xử lý vào-ra hoạt động theo chương trình của riêng nó
- Chương trình của bộ xử lý vào-ra có thể nằm trong bộ nhớ chính hoặc nằm trong một bộ nhớ riêng
- Hoạt động theo kiến trúc đa xử lý

14 January 2013 Kiến trúc máy tính 506

NKK-HUST

8.3. Nối ghép thiết bị ngoại vi

1. Các kiểu nối ghép vào-ra

- Nối ghép song song
- Nối ghép nối tiếp

14 January 2013 Kiến trúc máy tính 507

NKK-HUST

Nối ghép song song

```

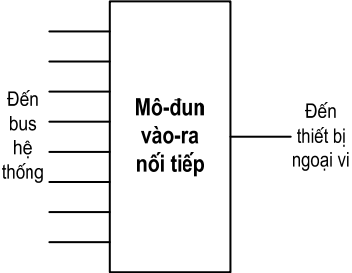
graph LR
    Bus[Đến bus hệ thống] --- Module[Mô-đun vào-ra song song]
    Module --- Device[Đến thiết bị ngoại vi]
    
```

- Truyền nhiều bit song song
- Tốc độ nhanh
- Cần nhiều đường truyền dữ liệu

14 January 2013 Kiến trúc máy tính 508

NKK-HUST

Nối ghép nối tiếp



- Truyền lần lượt từng bit
- Cần có bộ chuyển đổi từ dữ liệu song song sang nối tiếp hoặc/và ngược lại
- Tốc độ chậm hơn
- Cần ít đường truyền dữ liệu

14 January 2013

Kiến trúc máy tính

509

NKK-HUST

2. Các cấu hình nối ghép

- Điểm tới điểm (Point to Point)
 - Thông qua một cổng vào-ra nối ghép với một thiết bị ngoại vi
- Điểm tới đa điểm (Point to Multipoint)
 - Thông qua một cổng vào-ra cho phép nối ghép được với nhiều thiết bị ngoại vi
 - Ví dụ:
 - SCSI (Small Computer System Interface): 7 hoặc 15 thiết bị
 - USB (Universal Serial Bus): 127 thiết bị
 - IEEE 1394 (FireWire): 63 thiết bị

14 January 2013

Kiến trúc máy tính

510

NKK-HUST

Hết chương 8

14 January 2013

Kiến trúc máy tính

511

NKK-HUST

Kiến trúc máy tính

Chương 9

CÁC KIẾN TRÚC SONG SONG

Nguyễn Kim Khánh

Trường Đại học Bách khoa Hà Nội

14 January 2013

Kiến trúc máy tính

512

NKK-HUST

Nội dung học phần

- Chương 1. Giới thiệu chung
- Chương 2. Cơ bản về logic số
- Chương 3. Hệ thống máy tính
- Chương 4. Số học máy tính
- Chương 5. Kiến trúc tập lệnh
- Chương 6. Cấu trúc bộ xử lý
- Chương 7. Bộ nhớ
- Chương 8. Vào-ra
- Chương 9. Các kiến trúc song song**

14 January 2013 Kiến trúc máy tính 513

NKK-HUST

Nội dung của chương 9

- 9.1. Phân loại kiến trúc máy tính
- 9.2. Đa xử lý bộ nhớ dùng chung
- 9.3. Đa xử lý bộ nhớ phân tán
- 9.4. Bộ xử lý đa lỗi
- 9.5. Bộ xử lý đồ họa*

14 January 2013 Kiến trúc máy tính 514

NKK-HUST

9.1. Phân loại kiến trúc máy tính

- Phân loại của Michael Flynn (1966)
 - SISD - Single Instruction Stream, Single Data Stream
 - SIMD - Single Instruction Stream, Multiple Data Stream
 - MISD - Multiple Instruction Stream, Single Data Stream
 - MIMD - Multiple Instruction Stream, Multiple Data Stream

14 January 2013 Kiến trúc máy tính 515

NKK-HUST

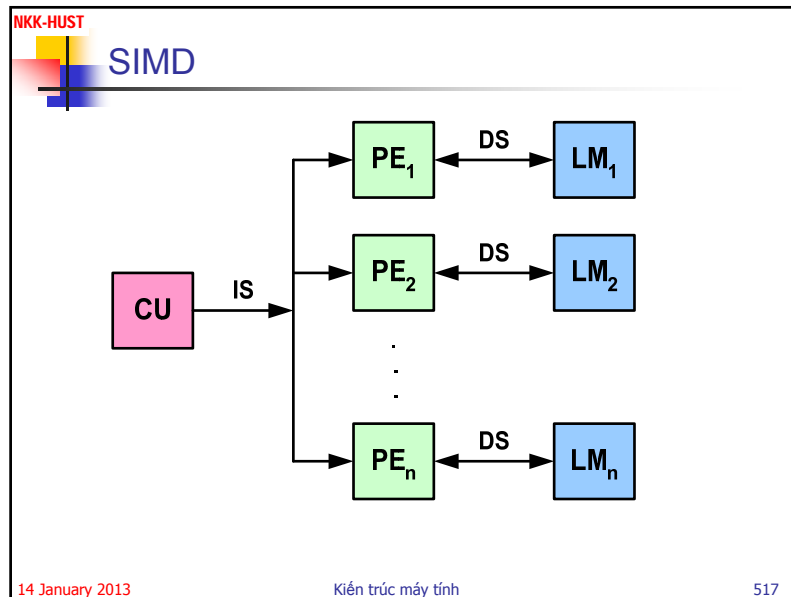
SISD

```

graph LR
    CU[CU] -- IS --> PU[PU]
    PU <--> |DS| MU[MU]
    
```

- CU: Control Unit
- PU: Processing Unit
- MU: Memory Unit
- Một bộ xử lý
- Đơn dòng lệnh
- Dữ liệu được lưu trữ trong một bộ nhớ
- Chính là Kiến trúc von Neumann

14 January 2013 Kiến trúc máy tính 516



NKK-HUST

SIMD (tiếp)

- Đơn dòng lệnh điều khiển đồng thời các phần tử xử lý PE (processing elements)
- Mỗi phần tử xử lý có một bộ nhớ dữ liệu riêng LM (local memory)
- Mỗi lệnh được thực hiện trên một tập các dữ liệu khác nhau
- Các mô hình SIMD
 - Vector Computer
 - Array processor

14 January 2013

Kiến trúc máy tính

518

NKK-HUST

MISD

- Một luồng dữ liệu cùng được truyền đến một tập các bộ xử lý
- Mỗi bộ xử lý thực hiện một dãy lệnh khác nhau.
- Chưa tồn tại máy tính thực tế
- Có thể có trong tương lai

14 January 2013

Kiến trúc máy tính

519

NKK-HUST

MIMD

- Tập các bộ xử lý
- Các bộ xử lý đồng thời thực hiện các dãy lệnh khác nhau trên các dữ liệu khác nhau
- Các mô hình MIMD
 - Multiprocessors (Shared Memory)
 - Multicomputers (Distributed Memory)

14 January 2013

Kiến trúc máy tính

520

NKK-HUST

MIMD - Shared Memory

14 January 2013Kiến trúc máy tính521

NKK-HUST

MIMD - Distributed Memory

14 January 2013Kiến trúc máy tính522

NKK-HUST

9.2. Đa xử lý bộ nhớ dùng chung

SMP- Symmetric Multiprocessors

14 January 2013Kiến trúc máy tính523

NKK-HUST

SMP (tiếp)

- Một máy tính có $n \geq 2$ bộ xử lý giống nhau
- Các bộ xử lý dùng chung bộ nhớ và hệ thống vào-ra
- Thời gian truy cập bộ nhớ là bằng nhau với các bộ xử lý
- Tất cả các bộ xử lý chia sẻ truy nhập vào-ra
- Các bộ xử lý có thể thực hiện chức năng giống nhau
- Hệ thống được điều khiển bởi một hệ điều hành phân tán

14 January 2013Kiến trúc máy tính524

NKK-HUST

Ưu điểm của SMP

- Hiệu năng
 - Các công việc có thể thực hiện song song
- Tính sẵn dùng
 - Các bộ xử lý có thể thực hiện các chức năng giống nhau, vì vậy lỗi của một bộ xử lý sẽ không làm dừng hệ thống
- Khả năng mở rộng
 - Người sử dụng có thể tăng hiệu năng bằng cách thêm bộ xử lý

14 January 2013 Kiến trúc máy tính 525

NKK-HUST

9.3. Đa xử lý bộ nhớ phân tán

Clusters

- Nhiều máy tính được kết nối với nhau bằng mạng liên kết tốc độ cao (~ Gbps)
- Mỗi máy tính có thể làm việc độc lập
- Mỗi máy tính được gọi là một node
- Các máy tính có thể được quản lý làm việc song song theo nhóm (cluster)
- Toàn bộ hệ thống có thể coi như là một máy tính song song

14 January 2013 Kiến trúc máy tính 526

NKK-HUST

Cluster (tiếp)

- Dễ dàng xây dựng và mở rộng
- Tính sẵn sàng cao
- Khả năng chịu lỗi
- Giá thành rẻ với hiệu năng cao

14 January 2013 Kiến trúc máy tính 527

NKK-HUST

Cluster of PCs

```

graph TD
    PC1[PC] --- S1[Switch tốc độ cao]
    PC2[PC] --- S1
    PC3[PC] --- S1
    PC4[PC] --- S1
    PC5[PC] --- S2[Switch tốc độ cao]
    PC6[PC] --- S2
    PC7[PC] --- S2
    PC8[PC] --- S2
    PC9[PC] --- S3[Switch tốc độ cao]
    PC10[PC] --- S3
    PC11[PC] --- S3
    PC12[PC] --- S3
    PC13[PC] --- S4[Switch tốc độ cao]
    PC14[PC] --- S4
    PC15[PC] --- S4
    PC16[PC] --- S4
    S1 --- S5[Switch tốc độ cao]
    S2 --- S5
    S3 --- S5
    S4 --- S5
    
```

14 January 2013 Kiến trúc máy tính 528

NKK-HUST

Cluster of SMPs

14 January 2013

Kiến trúc máy tính

529

NKK-HUST

Ví dụ: Hệ thống máy chủ Google (12/ 2000)

- Cluster of PCs
- Hơn 6.000 bộ xử lý
- Hệ thống lưu trữ dùng RAID: có 12.000 đĩa cứng ~ 1petabyte (1triệu GB)
- 2 site ở Silicon Valley, 1site ở Virginia
- Mỗi site được kết nối với Internet qua OC48 (2488Mbps)

14 January 2013

Kiến trúc máy tính

530

NKK-HUST

9.4. Bộ xử lý đa lõi (multicores)

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

14 January 2013

Kiến trúc máy tính

531

NKK-HUST

Các dạng tổ chức bộ xử lý đa lõi

14 January 2013

Kiến trúc máy tính

532

NKK-HUST

Intel - Core Duo

- 2006
- Two x86 superscalar, shared L2 cache
- Dedicated L1 cache per core
 - 32KB instruction and 32KB data
- 2MB shared L2 cache

14 January 2013

Kiến trúc máy tính

533

NKK-HUST

Intel Core Duo (3)

The diagram illustrates the internal architecture of the Intel Core Duo. At the top, two identical core blocks are shown side-by-side. Each core contains 32 KB L1 Caches (split into instruction and data), Execution Resources, and Arch. state. Below the cores are Thermal Control and APIC blocks. A central Power Management Logic block is connected to both cores. Below this is a 2 MB L2 Shared Cache, which is connected to a Bus Interface at the bottom. The Bus Interface is connected to a Front-Side Bus, represented by a double-headed arrow.

14 January 2013

534

NKK-HUST

Intel x86 Multicore Organization - Core i7

- November 2008
- Four x86 SMT processors
- Dedicated L2, shared L3 cache
- Speculative pre-fetch for caches
- On chip DDR3 memory controller
 - Three 8 byte channels (192 bits) giving 32GB/s
 - No front side bus
- QuickPath Interconnection
 - Cache coherent point-to-point link
 - High speed communications between processor chips
 - 6.4G transfers per second, 16 bits per transfer
 - Dedicated bi-directional pairs
 - Total bandwidth 25.6GB/s

14 January 2013

Kiến trúc máy tính

535

NKK-HUST

Intel Core i7

The diagram shows the architecture of the Intel Core i7. It features four cores, labeled Core 0 through Core 3, arranged in a row. Each core has its own 32 KB I&D L1 Caches and a 256 KB L2 Cache. These are connected to a shared 8 MB L3 Cache. Below the L3 cache are two main interconnect blocks: DDR3 Memory Controllers and the QuickPath Interconnect. The DDR3 Memory Controllers are connected to the L3 cache and provide 3 x 8B @ 1.33 GT/s. The QuickPath Interconnect is connected to the L3 cache and provides 4 x 20b @ 6.4 GT/s.

14 January 2013

Kiến trúc máy tính

536

NKK-HUST

9.5. Bộ xử lý đồ họa (GPU)

History of GPUs

- Early video cards
 - Frame buffer memory with address generation for video output
- 3D graphics processing
 - Originally high-end computers (e.g., SGI)
 - Moore's Law \Rightarrow lower cost, higher density
 - 3D graphics cards for PCs and game consoles
- Graphics Processing Units
 - Processors oriented to 3D graphics tasks
 - Vertex/pixel processing, shading, texture mapping, rasterization

14 January 2013

Kiến trúc máy tính

537

NKK-HUST

Graphics in the System

14 January 2013

Kiến trúc máy tính

538

NKK-HUST

GPU Architectures

- Processing is highly data-parallel
 - GPUs are highly multithreaded
 - Use thread switching to hide memory latency
 - Less reliance on multi-level caches
 - Graphics memory is wide and high-bandwidth
- Trend toward general purpose GPUs
 - Heterogeneous CPU/GPU systems
 - CPU for sequential code, GPU for parallel code
- Programming languages/APIs
 - DirectX, OpenGL
 - C for Graphics (Cg), High Level Shader Language (HLSL)
 - Compute Unified Device Architecture (CUDA)

14 January 2013

Kiến trúc máy tính

539

NKK-HUST

Example: NVIDIA Tesla

14 January 2013

Kiến trúc máy tính

540

NKK-HUST

Example: NVIDIA Tesla

- Streaming Processors
 - Single-precision FP and integer units
 - Each SP is fine-grained multithreaded
- Warp: group of 32 threads
 - Executed in parallel, SIMD style
 - 8 SPs × 4 clock cycles
 - Hardware contexts for 24 warps
 - Registers, PCs, ...

The diagram illustrates the architecture of a Tesla Multiprocessor. It shows a grid of 24 warps, each containing 32 threads. The threads are organized into groups of 8 single-precision (SP) units, which are executed in parallel in a SIMD style. The hardware contexts for these warps are shown as a separate grid, indicating that the multiprocessor can support 24 warps simultaneously. The diagram also shows the hardware supported threads and the hardware supported warps.

14 January 2013

Kiến trúc máy tính

541

NKK-HUST

Classifying GPUs

- Don't fit nicely into SIMD/MIMD model
 - Conditional execution in a thread allows an illusion of MIMD
 - But with performance degradation
 - Need to write general purpose code with care

	Static: Discovered at Compile Time	Dynamic: Discovered at Runtime
Instruction-Level Parallelism	VLIW	Superscalar
Data-Level Parallelism	SIMD or Vector	Tesla Multiprocessor

14 January 2013

Kiến trúc máy tính

542

NKK-HUST

Hết

14 January 2013

Kiến trúc máy tính

543