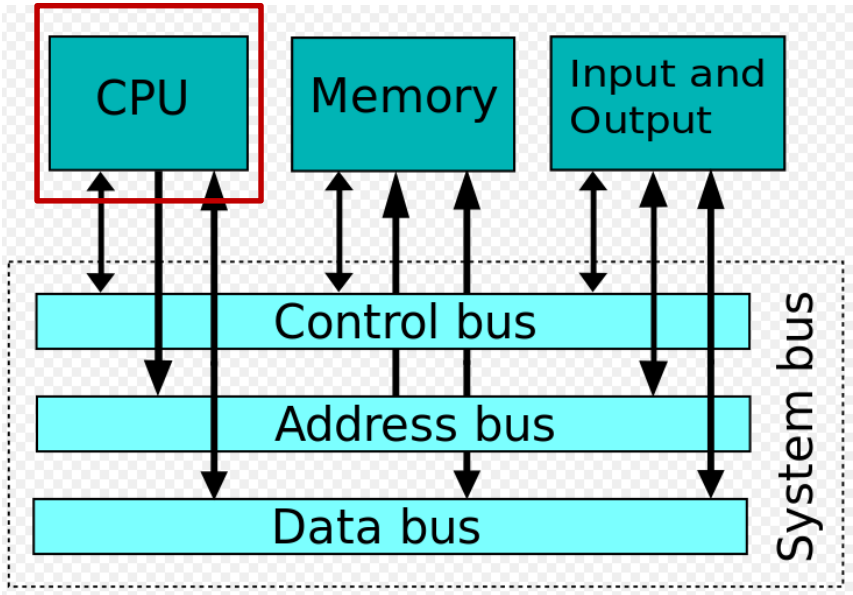
Chương 4

Bộ xử lý trung tâm (CPU)

4.1. Tổ chức của CPU

4.2. Tập lệnh



Giảng viên: ThS. Phan Như Minh

Minh họa

INTEL PENTIUM 4

AMD ATHLON 64

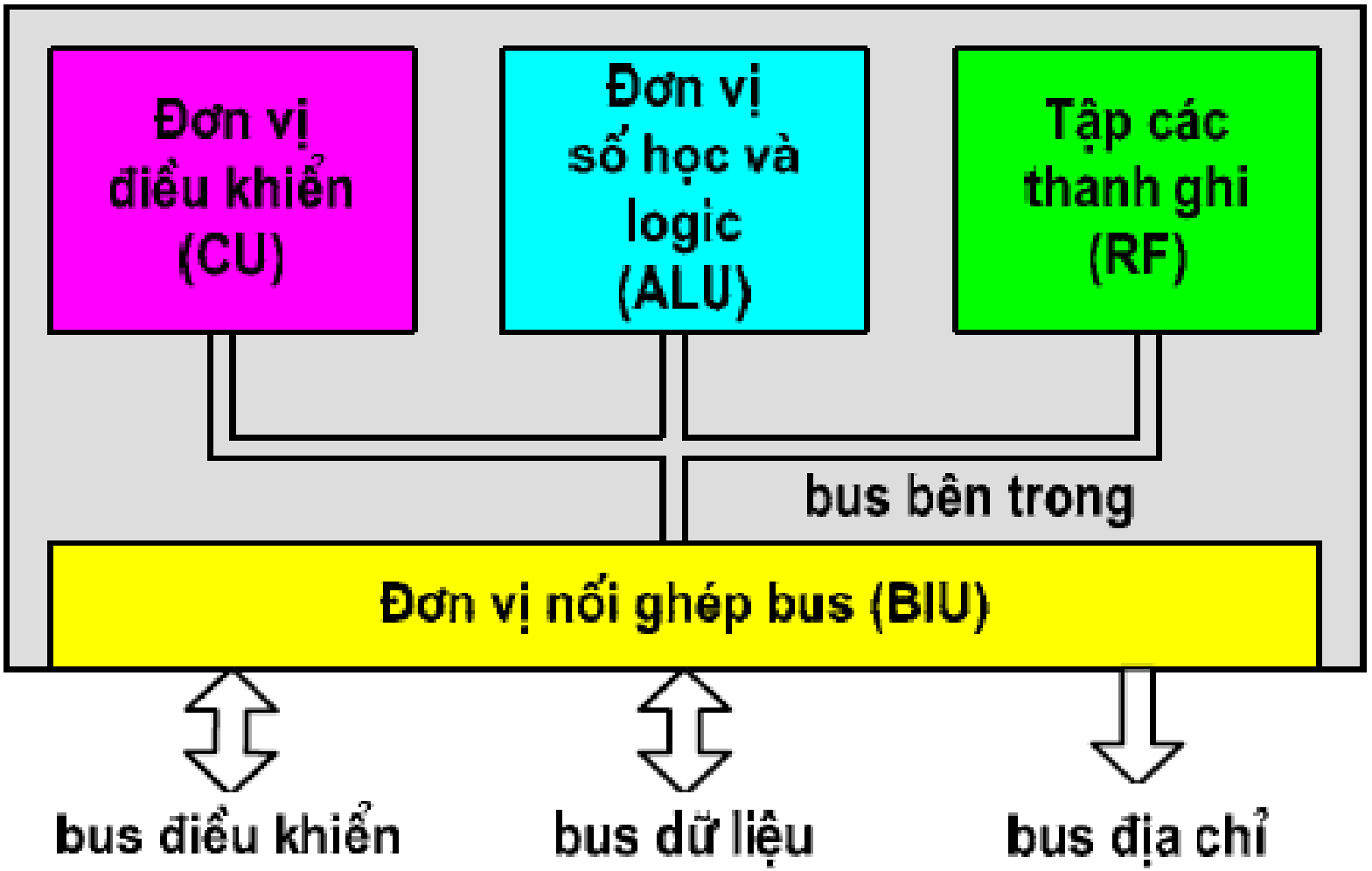
INTEL PENTIUM M

4.1. Tổ chức 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

Sơ đồ cấu trúc cơ bản của CPU



Các thành phần cơ bản của CPU

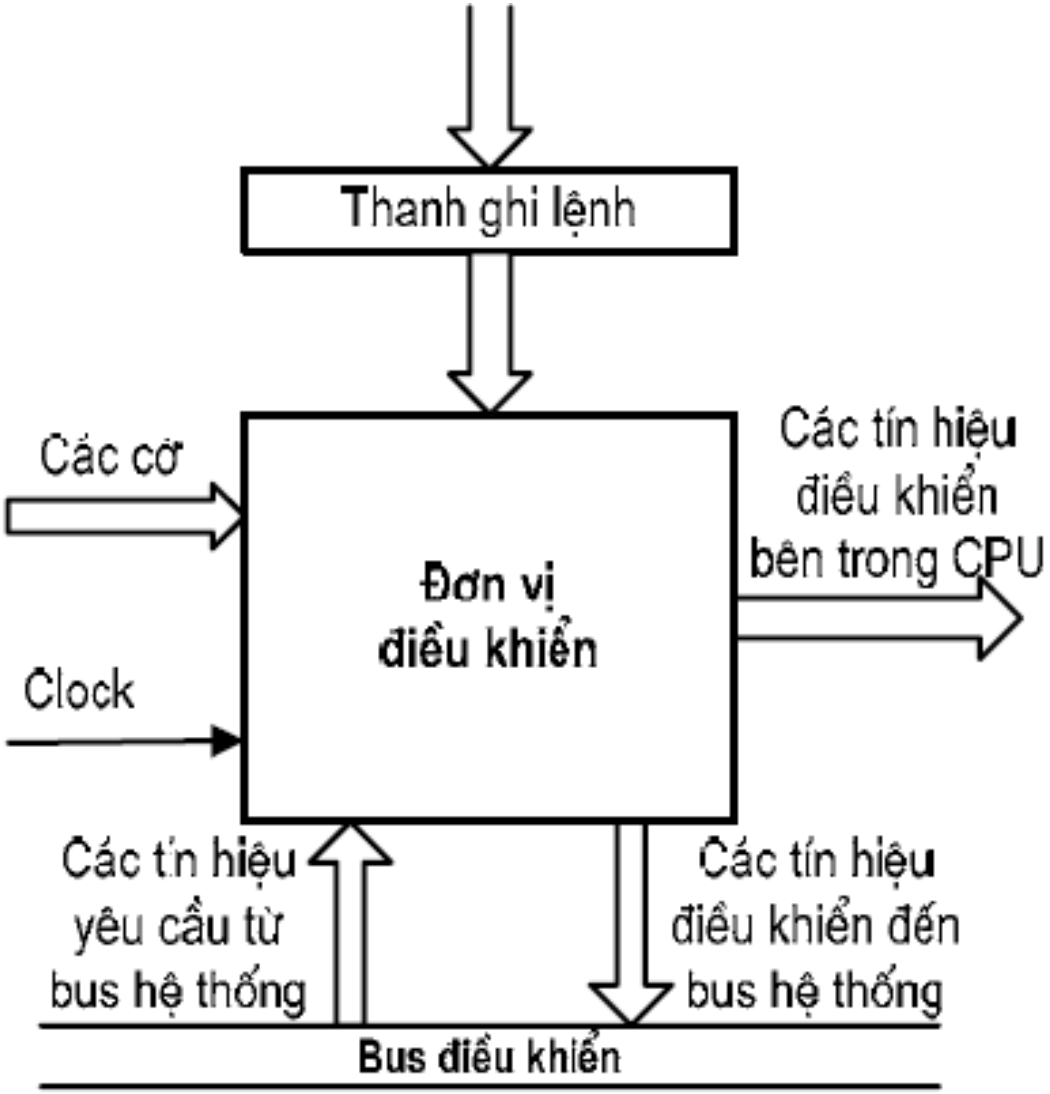
- Đơn vị điều khiển (Control Unit - CU)
- Đơn vị số học và logic (Arithmetic and Logic Unit - ALU)
- Tập thanh ghi (Register File - RF)
- Đơn vị nối ghép bus (Bus Interface Unit - BIU)
- Bus bên trong (Internal Bus)

Đơn vị điều khiển

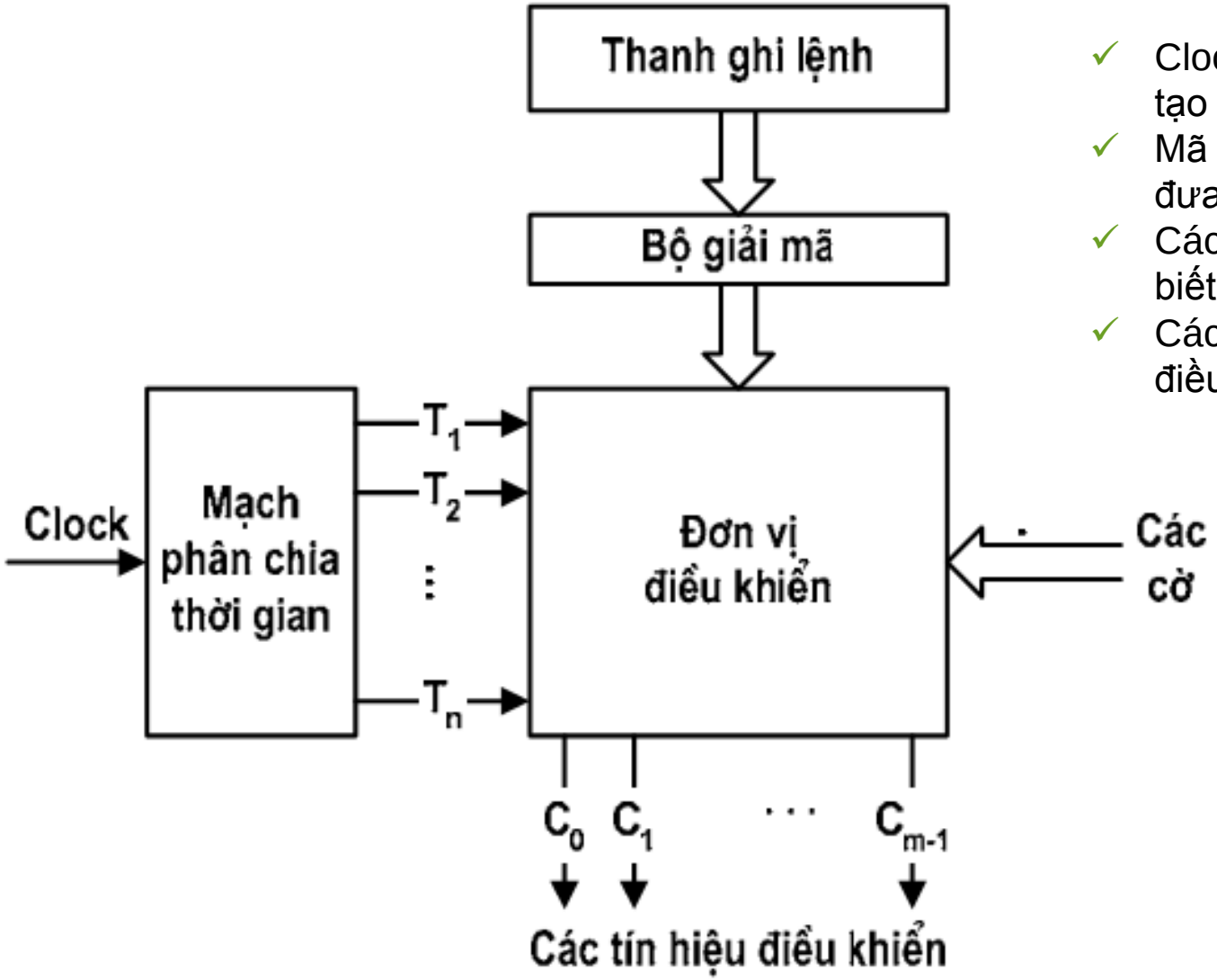
Chức năng

- Điều khiển nhận lệnh từ bộ nhớ đưa vào thanh ghi lệnh
- Tăng nội dung của PC để trỏ 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.

Mô hình kết nối đơn vị điều khiển



Đơn vị điều khiển nối kết cứng



- ✓ 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

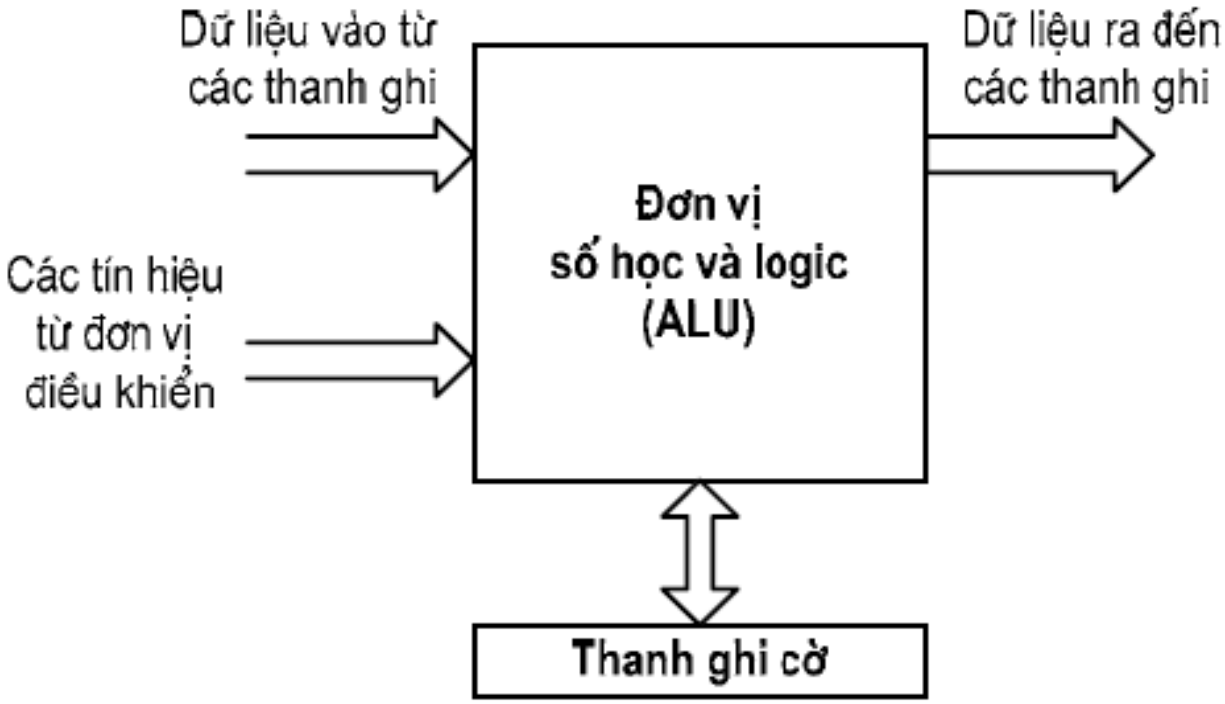
Các tín hiệu đưa đến đơn vị điều khiển

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

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.



Tập thanh ghi

Chức năng và đặc điểm:

- Chứa các thông tin tạm thời phục vụ cho hoạt động ở thời điểm hiện tại của CPU
- 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

Phân loại thanh ghi theo chức năng

- Thanh ghi địa chỉ: quản lý địa chỉ của ngăn nhớ hay cổng vào-ra.
- Thanh ghi dữ liệu: chứa tạm thời các dữ liệu.
- Thanh ghi đa năng: có thể chứa địa chỉ hoặc dữ liệu.
- Thanh ghi điều khiển/trạng thái: chứa các thông tin điều khiển và trạng thái của CPU.
- Thanh ghi lệnh: chứa lệnh đang được thực hiện.

Một số thanh ghi điển hình

- Các thanh ghi địa chỉ
  - 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

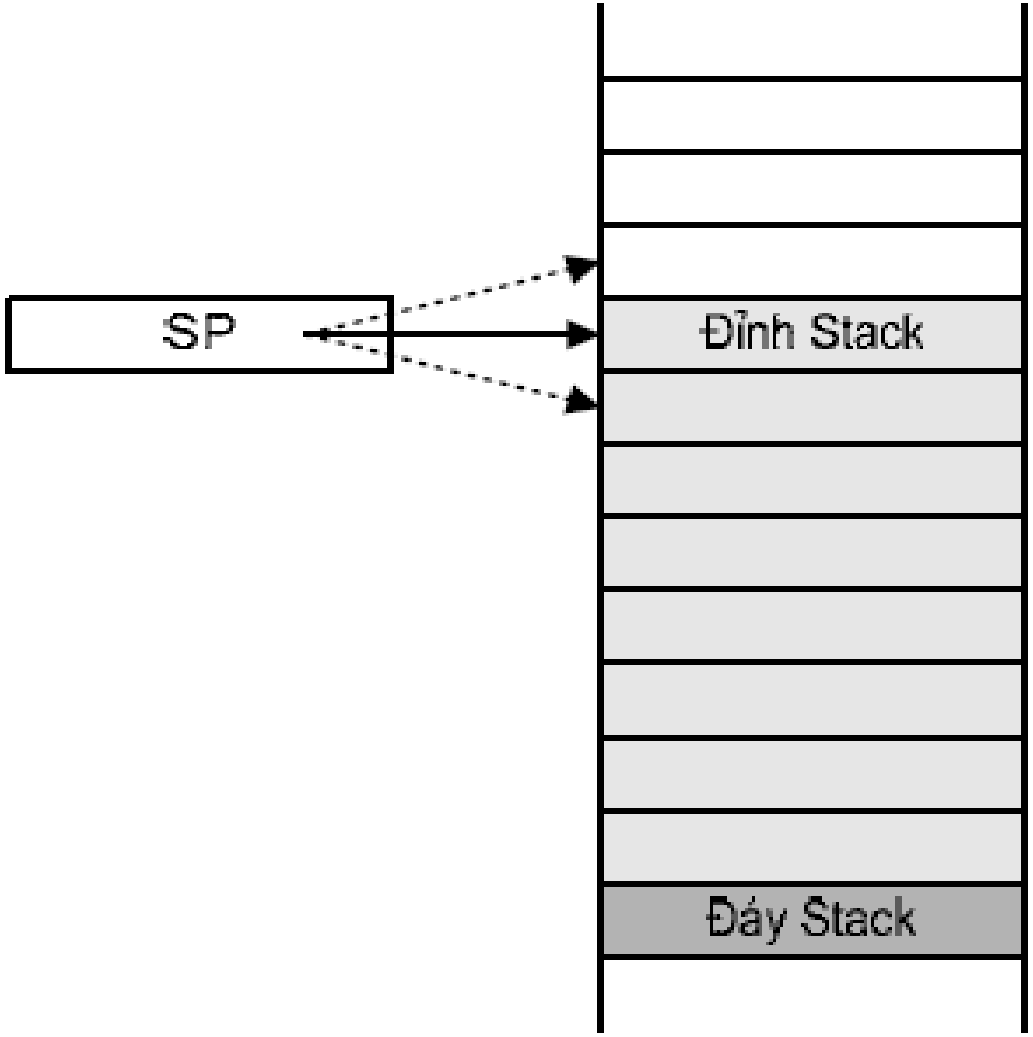
Ngăn xếp (Stack)

- Ngăn xếp là vùng nhớ có cấu trúc LIFO (Last In - First Out)
- 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

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 tự động 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 tự động tăng
- Khi ngăn xếp rỗng, SP trỏ vào đáy

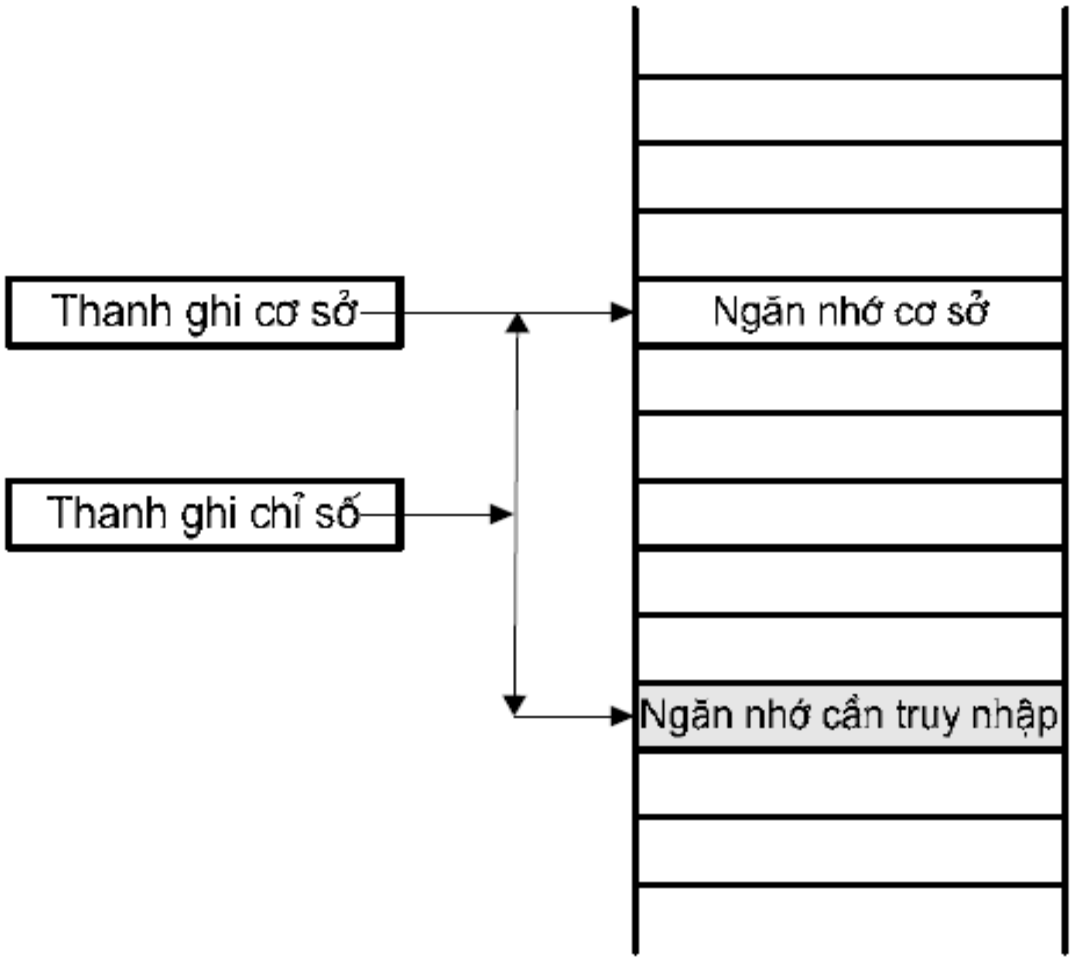
Minh họa con trỏ ngăn xếp SP



Thanh ghi cơ sở và thanh ghi chỉ số

- Thanh ghi cơ sở: chứa địa chỉ của ngăn nhớ cơ sở (địa chỉ cơ sở)
- Thanh ghi chỉ số: chứa độ lệch địa chỉ giữa ngăn nhớ mà CPU cần truy nhập so với ngăn nhớ cơ sở (chỉ số)
- Địa chỉ của ngăn nhớ cần truy nhập = địa chỉ cơ sở + chỉ số

Minh họa thanh ghi cơ sở và thanh ghi chỉ số



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

Thanh ghi trạng thái (Status Register)

- 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

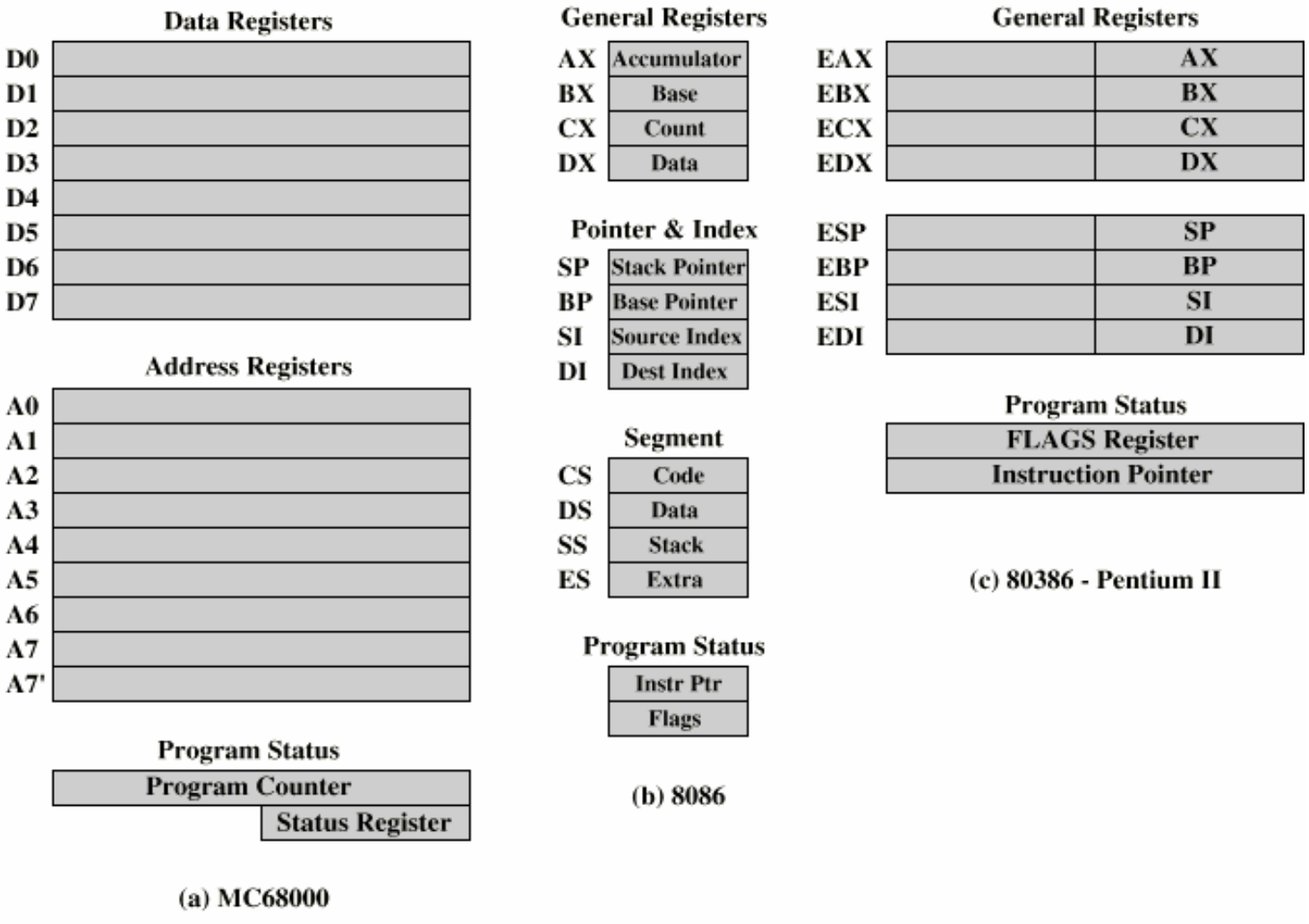
Ví dụ cờ phép toán

- Cờ Zero (cờ rỗng): được thiết lập lên 1 khi kết quả của phép toán bằng 0.
- Cờ Sign (cờ dấu): được thiết lập lên 1 khi kết quả phép toán nhỏ hơn 0
- Cờ Carry (cờ nhớ): được thiết lập lên 1 nếu phép toán có nhớ ra ngoài bit cao nhất => cờ báo tràn với số không dấu.
- Cờ Overflow (cờ tràn): được thiết lập lên 1 nếu cộng hai số nguyên cùng dấu mà kết quả có dấu ngược lại => cờ báo tràn với số có dấu .

Ví dụ cờ điều khiển

- Cờ Interrupt (Cờ cho phép ngắt):
  - Nếu IF = 1=> CPU ở trạng thái cho phép ngắt với tín hiệu yêu cầu ngắt từ bên ngoài gửi tới
  - Nếu IF = 0=> CPU ở trạng thái cấm ngắt với tín hiệu yêu cầu ngắt từ bên ngoài gửi tới

Tập thanh ghi của một số bộ xử lý



4.2. Tập lệnh

- 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 gợi nhớ => chính là các lệnh của hợp ngữ

Các thành phần của lệnh máy

Mã thao tác	Địa chỉ toán hạng

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

Mô tả lệnh

- Lệnh máy là mã nhị phân
- Để dễ hiểu và dễ nhớ đối với con người, người ta mô tả lệnh bằng các ký hiệu gợi nhớ
  - Ví dụ: ADD, SUB, LOAD
- Toán hạng có thể được mô tả như sau:
  - ADD A,B

Các kiểu lệnh

- Xử lý dữ liệu
- Lưu trữ dữ liệu (bộ nhớ chính)
- Vận chuyển dữ liệu (vào-ra)
- Điều khiển luồng dữ liệu

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

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

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

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

Đánh giá về số địa chỉ toán hạng

- Nhiều địa chỉ toán hạng
  - Các lệnh phức tạp hơn
  - Cần nhiều thanh ghi
  - Chương trình có ít lệnh hơn
  - Nhận lệnh và thực hiện lệnh chậm hơn
- Ít địa chỉ toán hạng
  - Các lệnh đơn giản hơn
  - Cần ít thanh ghi
  - Chương trình có nhiều lệnh hơn
  - Nhận lệnh và thực hiện lệnh nhanh hơn

Các vấn đề của thiết kế tập lệnh (1)

- Về thao tác
  - Bao nhiêu thao tác ?
  - Các thao tác nào ?
  - Mức độ phức tạp của các thao tác ?
- Các kiểu dữ liệu
- Các khuôn dạng lệnh
  - Độ dài của trường mã thao tác
  - Số lượng địa chỉ toán hạng

Các vấn đề của thiết kế tập lệnh (2)

- Các thanh ghi
  - Số thanh ghi của CPU được sử dụng
  - Các thao tác nào được thực hiện trên các thanh ghi ?
- Các phương pháp định địa chỉ (addressing modes)
- RISC hay CISC
  - Reduced Instruction Set Computing
  - Complex Instruction Set Computing

Các kiểu toán hạng

- Địa chỉ
- Số
  - Số nguyên
  - Số dấu phẩy động
- Ký tự
  - Ví dụ: mã ASCII
- Dữ liệu logic
  - Các bit hoặc các cờ

Các thao tác của lệnh

- Chuyển dữ liệu
- Xử lý số học với số nguyên
- Xử lý logic
- Điều khiển vào-ra
- Chuyển điều khiển (rẽ nhánh)
- Điều khiển hệ thống

Các lệnh chuyển dữ liệu

- MOVE Copy dữ liệu từ nguồn đến đích
- LOAD Nạp dữ liệu từ bộ nhớ đến bộ xử lý
- STORE Cất dữ liệu từ bộ xử lý đến bộ nhớ
- EXCHANGE Trao đổi nội dung của nguồn và đích
- CLEAR Chuyển các bit 0 vào toán hạng đích
- SET Chuyển các bit 1 vào toán hạng đích
- PUSH Cất nội dung toán hạng nguồn vào ngăn xếp
- POP Lấy nội dung đỉnh ngăn xếp đưa đến toán hạng đích

Các lệnh số học

- ADD Cộng hai toán hạng
- SUBTRACT Trừ hai toán hạng
- MULTIPLY Nhân hai toán hạng
- DIVIDE Chia hai toán hạng
- ABSOLUTE Lấy trị tuyệt đối toán hạng
- NEGATE Đổi dấu toán hạng (lấy bù 2)
- INCREMENT Tăng toán hạng thêm 1
- DECREMENT Giảm toán hạng đi 1
- COMPARE Trừ hai toán hạng để lập cờ

Các lệnh logic

- AND Thực hiện phép AND hai toán hạng
- OR Thực hiện phép OR hai toán hạng
- XOR Thực hiện phép XOR hai toán hạng
- NOT Đảo bit của toán hạng (lấy bù 1)
- TEST Thực hiện phép AND hai toán hạng để lập cờ

Minh hoạ các lệnh AND, OR, XOR

- Giả sử có hai thanh ghi chứa dữ liệu như sau:

  (R1) = 1010 1010

  (R2) = 0000 1111

- R1 <- (R1) AND (R2) = 0000 1010

  Phép toán AND dùng để xoá một số bit và giữ nguyên một số bit còn lại của toán hạng.

- R1 <-(R1) OR (R2) = 1010 1111

  Phép toán OR dùng để thiết lập một số bit và giữ nguyên một số bit còn lại của toán hạng.

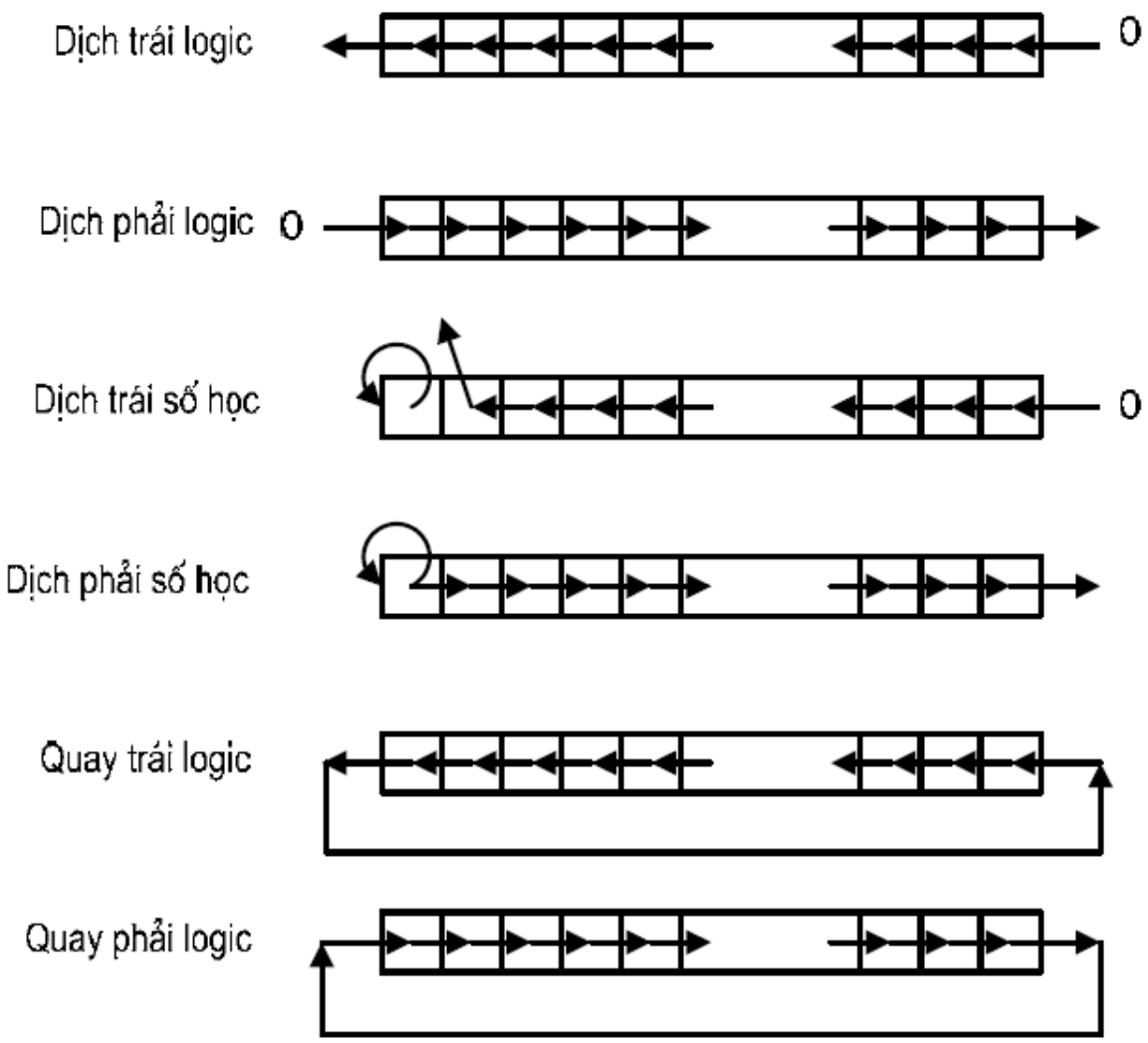
- R1<- (R1) XOR (R2) = 1010 0101

  Phép toán XOR dùng để đảo một số bit và giữ nguyên một số bit còn lại của toán hạng.

Các lệnh logic (tiếp)

- SHIFT Dịch trái (phải) toán hạng
- ROTATE Quay trái (phải) toán hạng

Các thao tác SHIFT và ROTATE



Các lệnh vào ra chuyên dụng

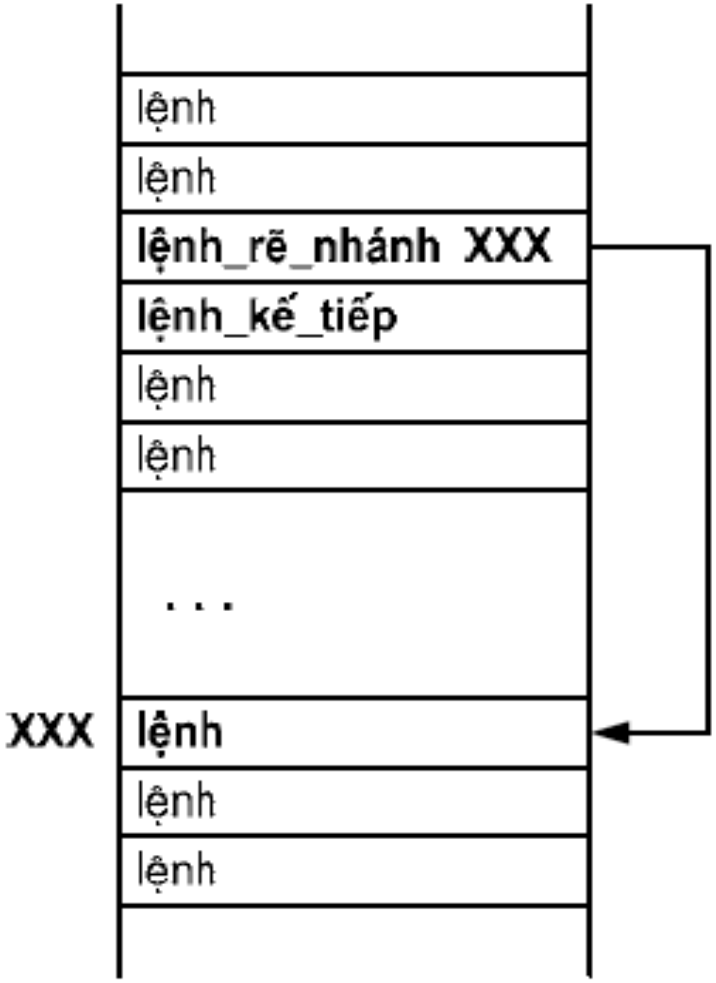
- INPUT Copy dữ liệu từ một cổng xác định đưa đến đích
- OUTPUT Copy dữ liệu từ nguồn đến một cổng xác định

Các lệnh chuyển điều khiển

- JUMP (BRANCH) Lệnh nhảy không điều kiện:
  - nạp vào PC một địa chỉ xác định
- JUMP CONDITIONAL Lệnh nhảy có điều kiện:
  - điều kiện đúng =>nạp vào PC một địa chỉ xác định
  - điều kiện sai =>không làm gì cả
- CALL Lệnh gọi chương trình con:
  - Cất nội dung của PC (địa chỉ trở về) ra một vị trí xác định (thường ở Stack)
  - Nạp vào PC địa chỉ của lệnh đầu tiên của chương trình con
- RETURN Lệnh trở về từ chương trình con:
  - Khôi phục địa chỉ trở về trả lại cho PC để trở về chương trình chính

Lệnh rẽ nhánh không điều kiện

- Chuyển tới thực hiện lệnh ở vị trí có địa chỉ XXX:
- PC <- XXX

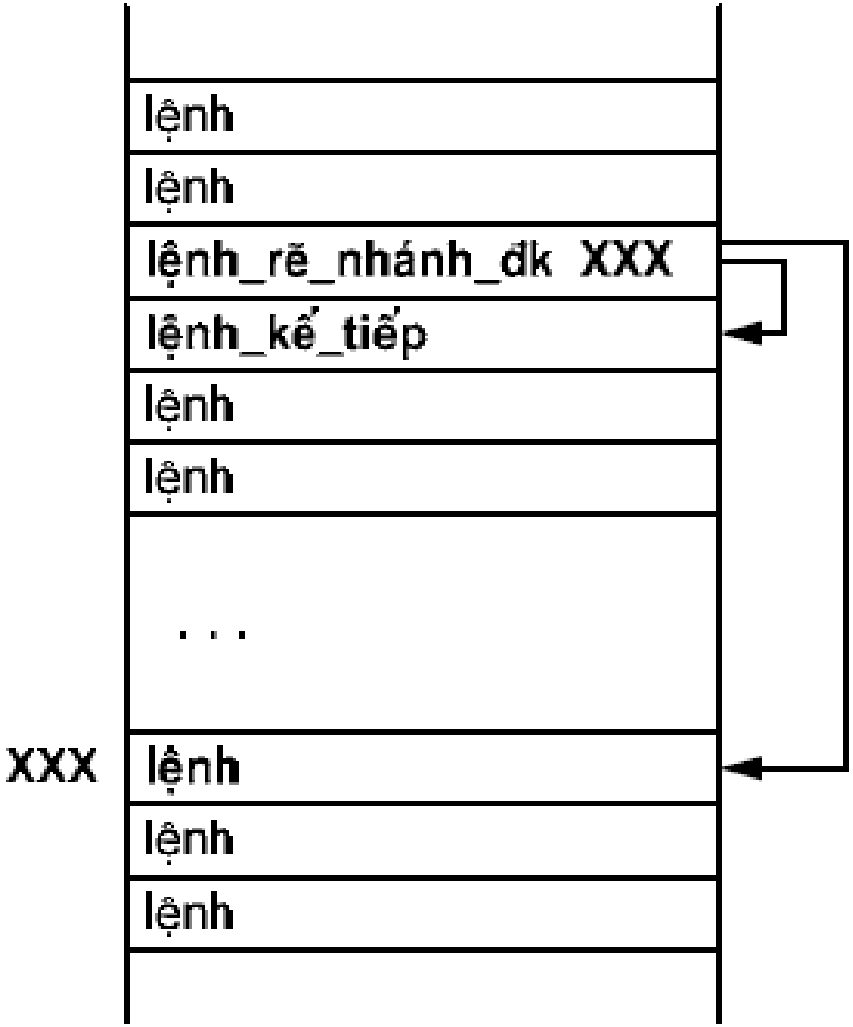


Lệnh rẽ nhánh có điều kiện

- Trong lệnh có kèm theo điều kiện
- Kiểm tra điều kiện trong lệnh:
  - Nếu điều kiện đúng => chuyển tới thực hiện lệnh ở vị trí có địa chỉ XXX :

    PC <- XXX
  - Nếu điều kiện sai => chuyển sang thực hiện **lệnh_kế_tiếp**
- Điều kiện thường được kiểm tra thông qua các cờ
- Có nhiều lệnh rẽ nhánh có điều kiện

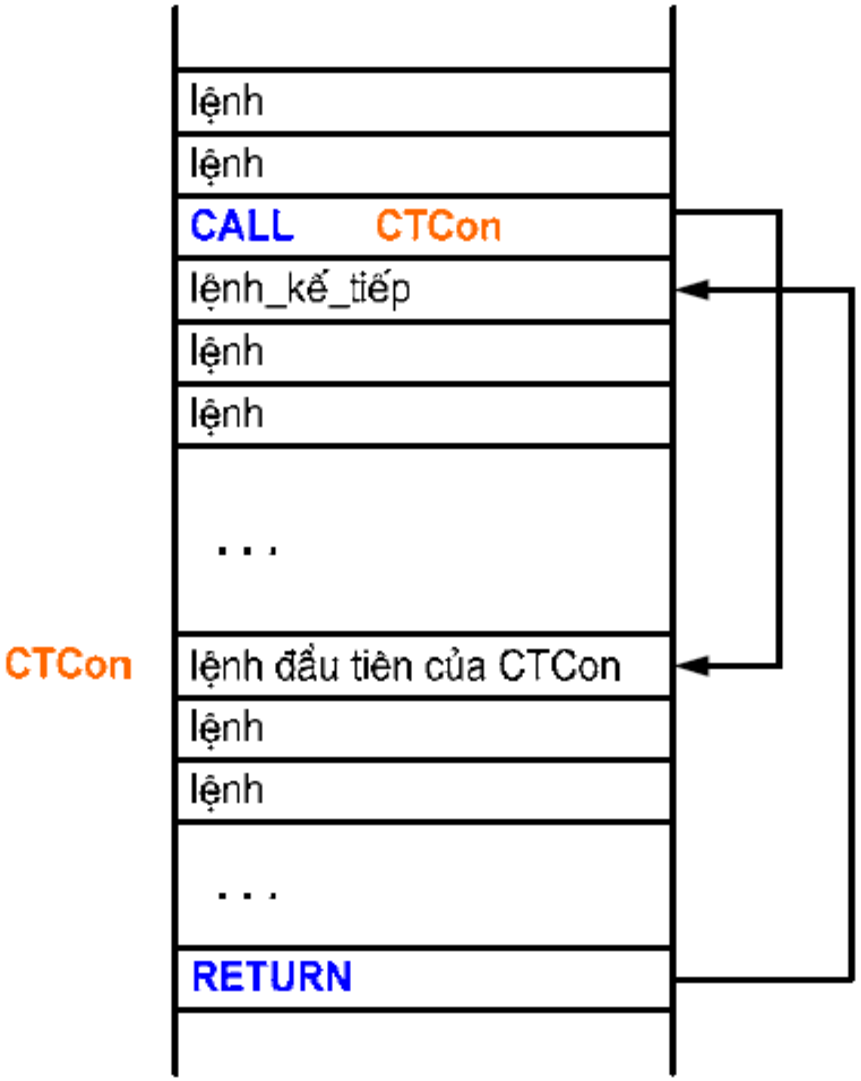
Minh hoạ lệnh rẽ nhánh có điều kiện



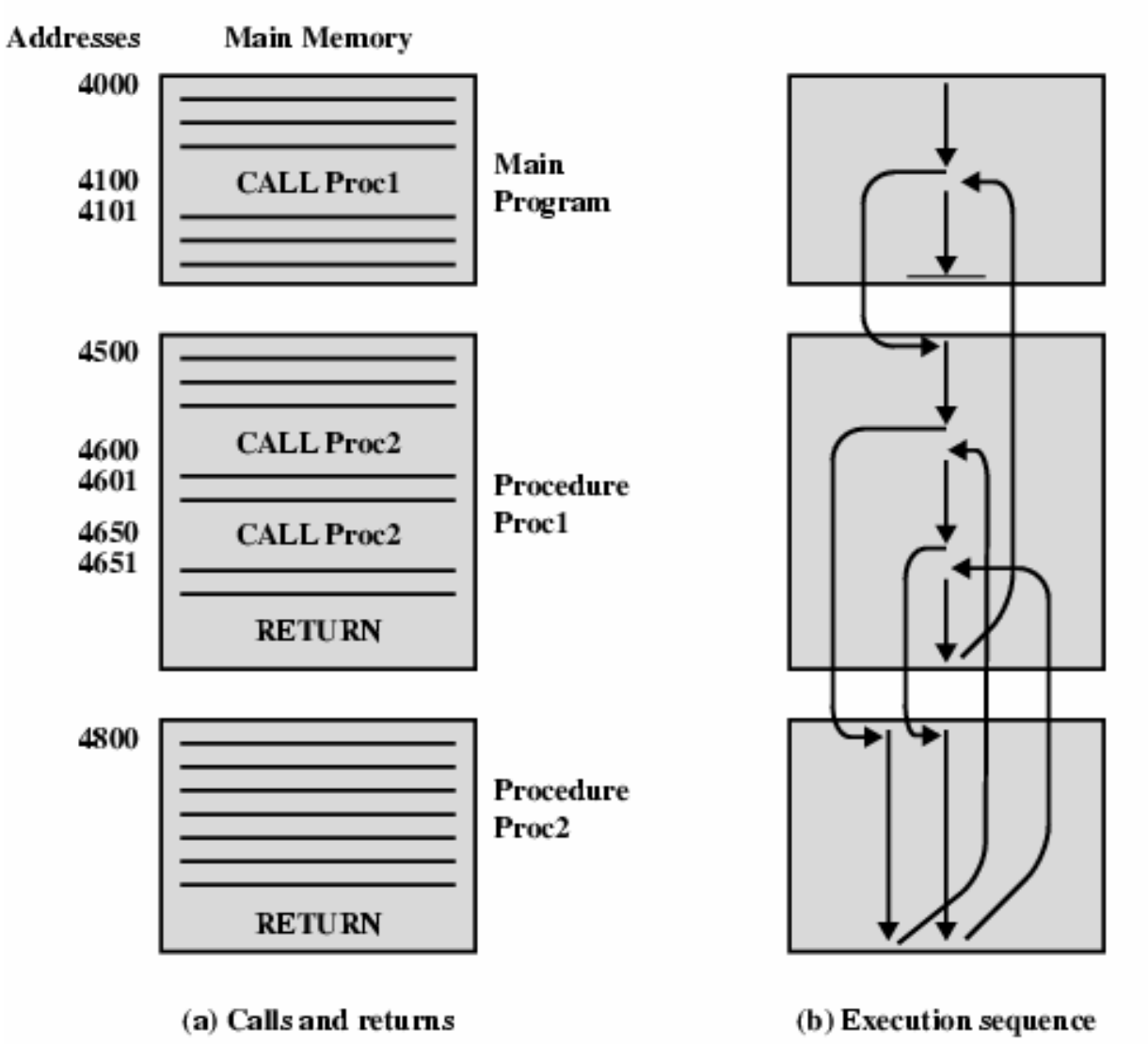
Lệnh CALL và RETURN

- Lệnh gọi chương trình con: lệnh CALL
  - Cất nội dung PC (chứa địa chỉ của lệnh_kế_tiếp) ra Stack
  - Nạp vào PC địa chỉ của lệnh đầu tiên của chương trình con được gọi
  - => Bộ xử lý được chuyển sang thực hiện chương trình con tương ứng
- Lệnh trở về từ chương trình con: lệnh RETURN
  - Lấy địa chỉ của lệnh_kế_tiếp được cất ở Stack nạp trả lại cho PC => Bộ xử lý được điều khiển quay trở về thực hiện tiếp lệnh nằm sau lệnh CALL

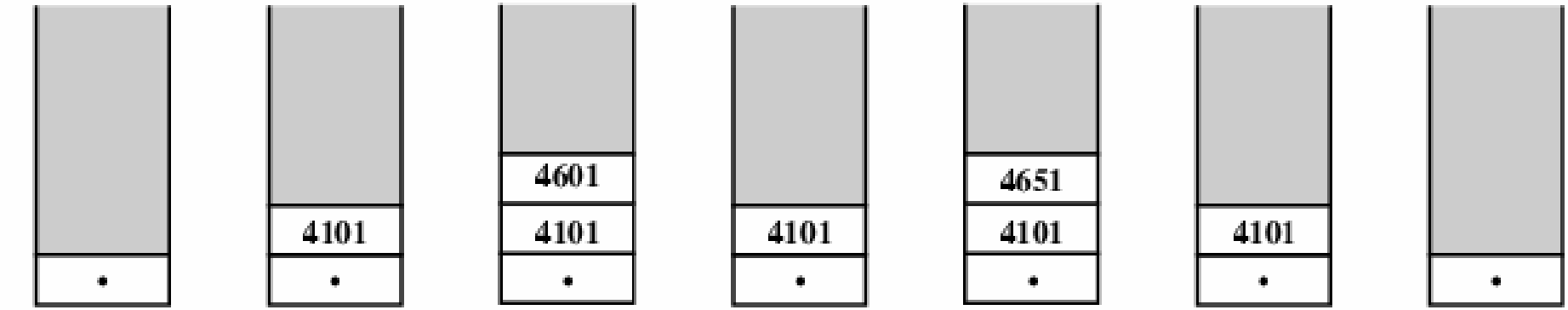
Minh hoạ lệnh CALL và RETURN



Gọi các thủ tục lồng nhau



Sử dụng Stack



Các lệnh điều khiển hệ thống

- HALT Dừng thực hiện chương trình
- WAIT Tạm dừng thực hiện chương trình, lặp kiểm tra điều kiện cho đến khi thoả mãn thì tiếp tục thực hiện
- NO OPERATION Không thực hiện gì cả
- LOCK Cấm không cho xin chuyển nhượng bus
- UNLOCK Cho phép xin chuyển nhượng bus