

Chương 4: Tập lệnh máy tính

Nội dung

- 4.1 Khái niệm tập lệnh, thành phần lệnh máy (2,5t)
- 4.2 Chu trình thực hiện lệnh máy (1,5t)
- 4.3 Một số dạng lệnh thông dụng (2,5t)
- 4.4 Các dạng toán hạng (3t)
- 4.5 Các chế độ địa chỉ (2,5)

Nội dung

4.1 Khái niệm tập lệnh, thành phần lệnh máy

4.2 Chu trình thực hiện lệnh máy

4.3 Một số dạng lệnh thông dụng

4.4 Các dạng toán hạng

4.5 Các chế độ địa chỉ

4.1 Tập lệnh & các thành phần lệnh máy

Khái niệm 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 hợp ngữ

4.1 Tập lệnh & các thành phần lệnh máy

Lệnh máy: gồm có 2 thành phần:

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 của toán hạng: 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

Số lượng địa chỉ toán hạng trong lệnh máy (1/4)

- Ba địa chỉ toán hạng:
 - 2 toán hạng nguồn, 1 toán hạng đích
 - $c = a + b$
 - Từ lệnh dài vì phải mã hóa địa chỉ cho cả 3 toán hạng
 - Được sử dụng trên các bộ xử lý tiên tiến

Số lượng địa chỉ toán hạng trong lệnh máy (2/4)

- 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
 - Ví dụ: $a = a + b$
 - 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 máy (3/4)

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

Số lượng địa chỉ toán hạng trong lệnh máy (4/4)

- Không địa chỉ toán hạng:
 - Các toán hạng đều được ngầm định
 - Sử dụng stack
 - Ví dụ:

push A

push B

add

pop C

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 hơn
 - Chương trình có nhiều lệnh hơn
 - Nhận lệnh và thực hiện lệnh nhanh hơn

Các vấn đề của thiết kế tập lệnh

- 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

- Các thanh ghi
 - Số các 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 mode) (xét sau)

Nội dung

4.1 Khái niệm tập lệnh, thành phần lệnh máy

4.2 Chu trình thực hiện lệnh máy

4.3 Một số dạng lệnh thông dụng

4.4 Các dạng toán hạng

4.5 Các chế độ địa chỉ

4.2 Chu trình thực hiện lệnh máy

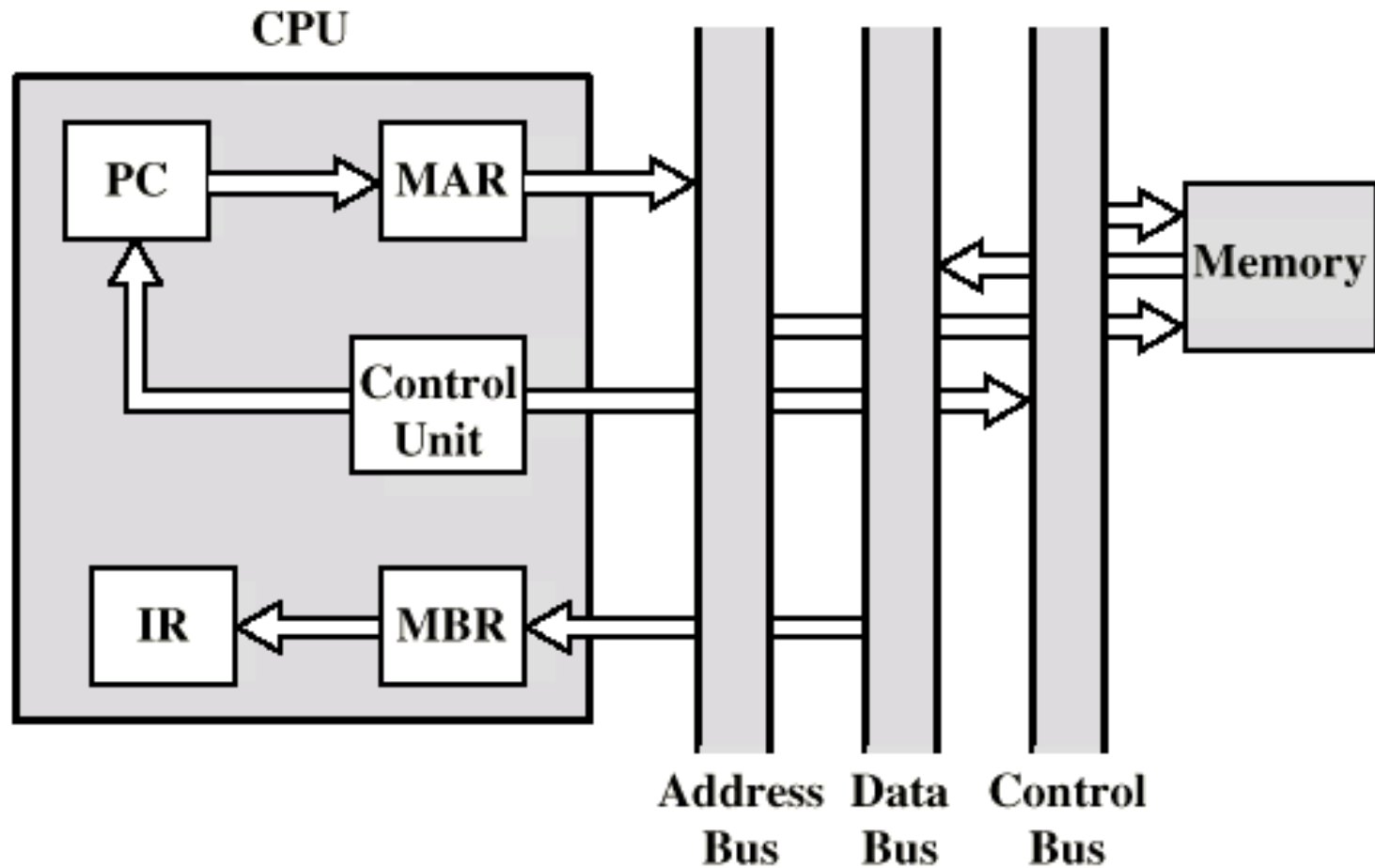
Bao gồm:

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

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.

Quá trình nhận lệnh



MBR = Memory buffer register
MAR = Memory address register
IR = Instruction register
PC = Program counter

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.

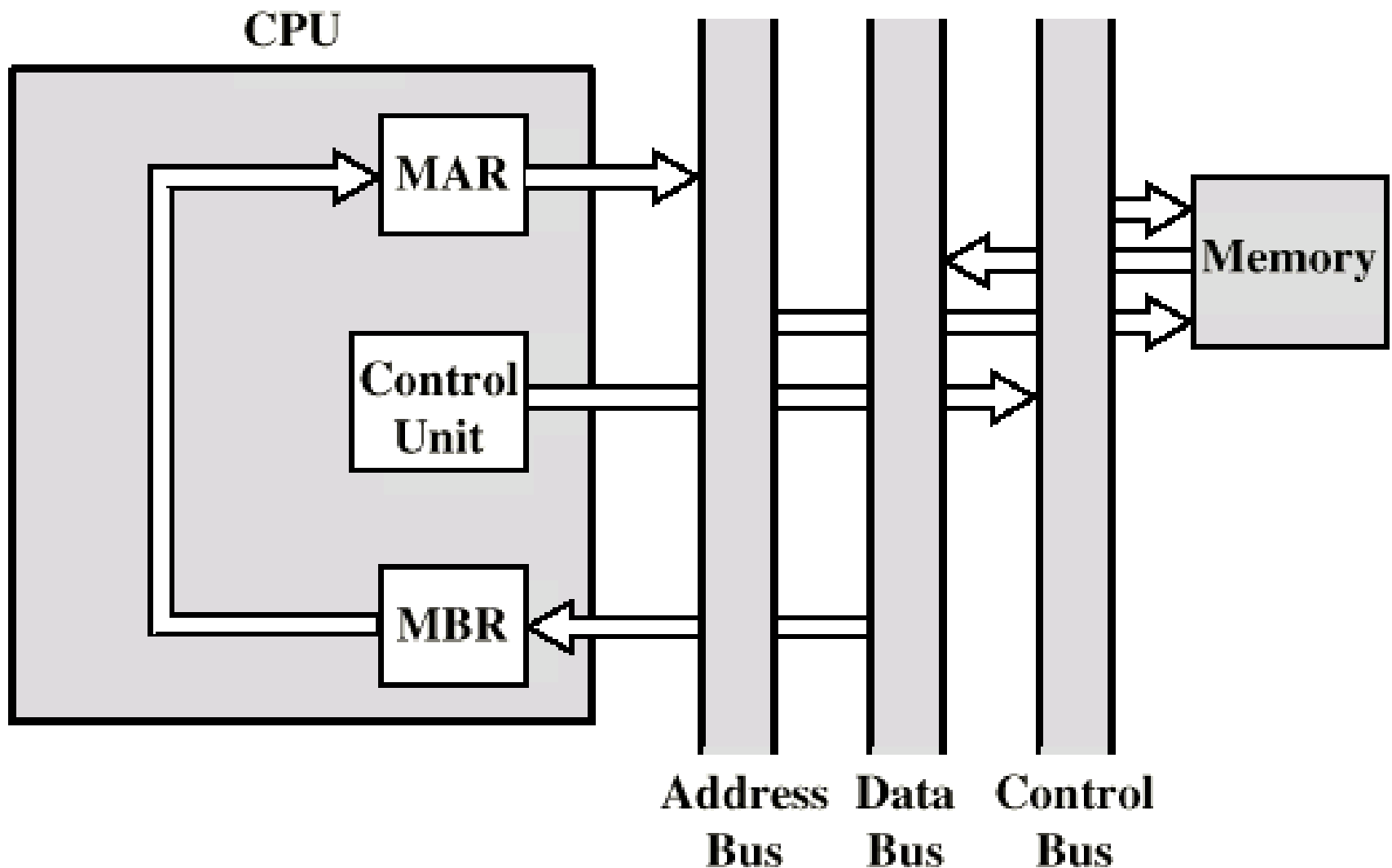
Nhận dữ liệu

- 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.

Nhận dữ liệu gián tiếp

- CPU đưa địa chỉ ra bus địa chỉ
- CPU phát tín hiệu điều khiển đọc
- Nội dung ngăn nhớ được đọc vào CPU, đó chính là địa chỉ của toán hạng
- Địa chỉ này được CPU phát ra bus địa chỉ để tìm ra toán hạng
- CPU phát tín hiệu điều khiển đọc
- Toán hạng được đọc vào CPU

Quá trình nhận dữ liệu gián tiếp



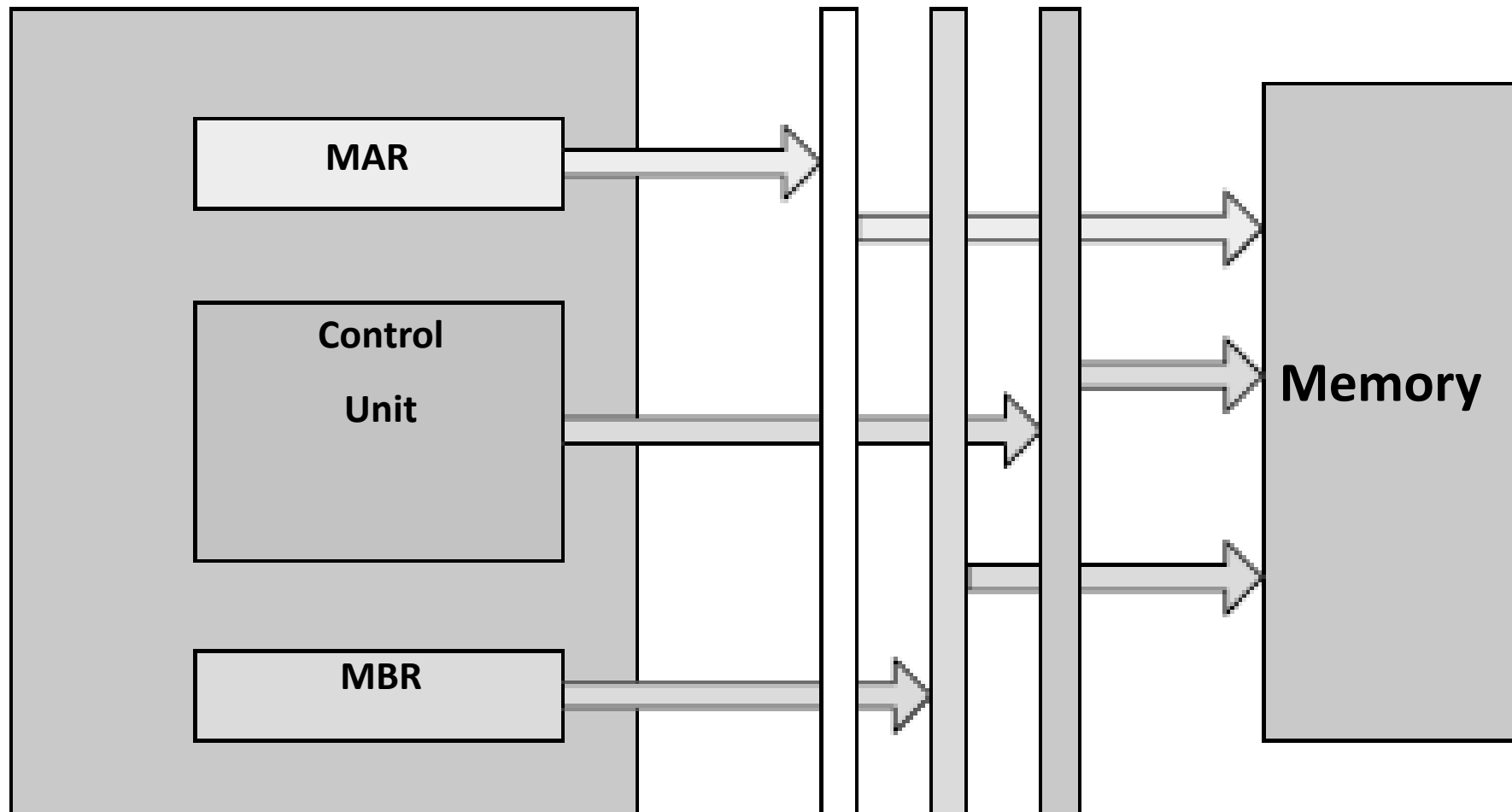
Thực hiện lệnh

- 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)
 - + ...

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.

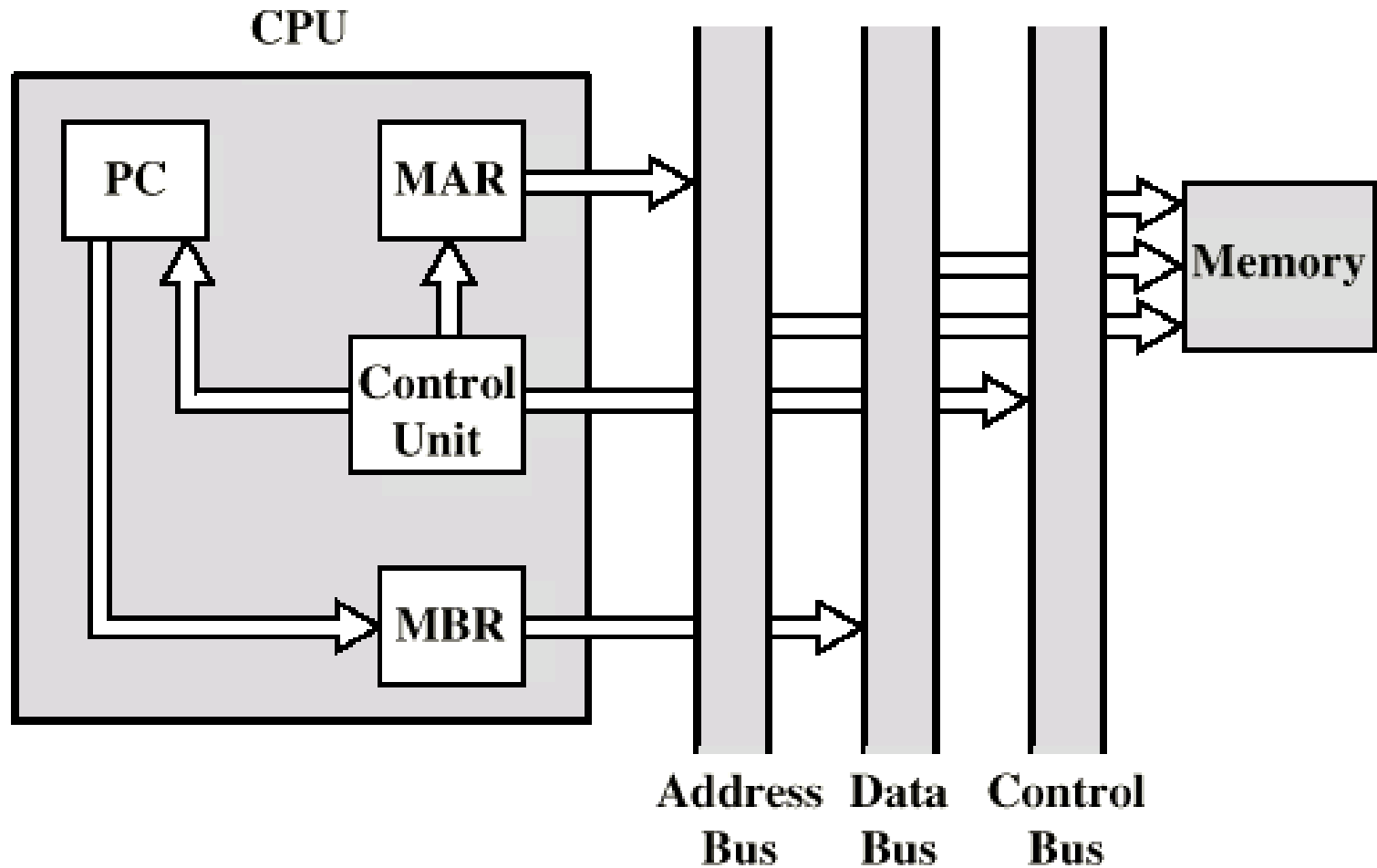
Quá trình ghi toán hạng



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

Mô tả chu trình ngắ



Nội dung

4.1 Khái niệm tập lệnh, thành phần lệnh máy

4.2 Chu trình thực hiện lệnh máy

4.3 Một số dạng lệnh thông dụng

4.4 Các dạng toán hạng

4.5 Các chế độ địa chỉ

4.3 Một số dạng lệnh thông dụng

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

Minh họa các lệnh AND, OR, XOR

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

(R1) = 1010 1010

(R2) = 0000 1111

- $R1 \leftarrow (R1) \text{ AND } (R2) = 0000 1010$

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

- $R1 \leftarrow (R1) \text{ OR } (R2) = 1010 1111$

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

- $R1 \leftarrow (R1) \text{ XOR } (R2) = 1010 0101$

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

Các lệnh vào/ra chuyên dụng

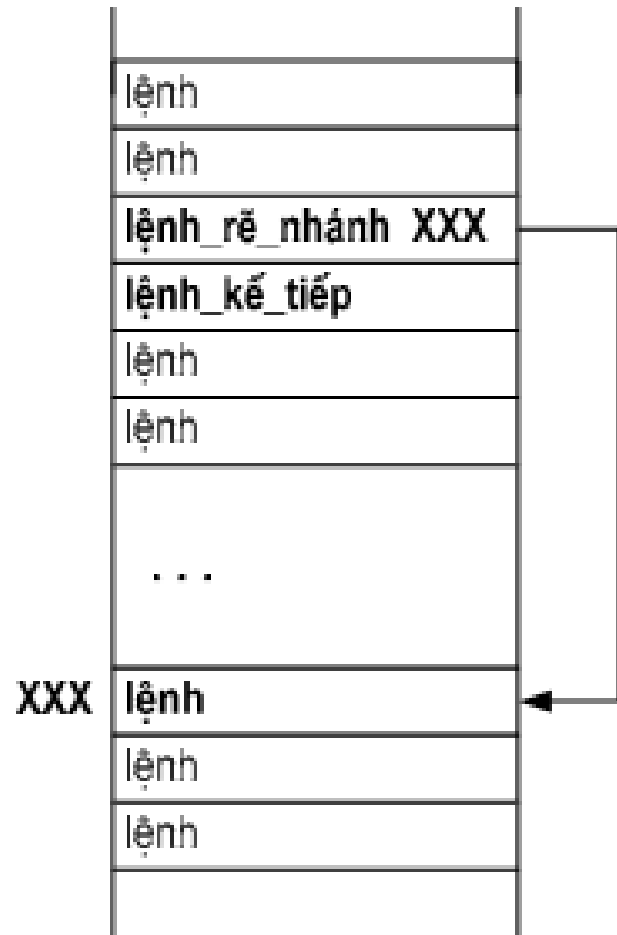
- **INPUT** Copy dữ liệu từ một cổng xác định đưa đến đích
- **OUTPUT** Copy dữ liệu từ nguồn đến một cổng xác định

Các lệnh chuyển quyền 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

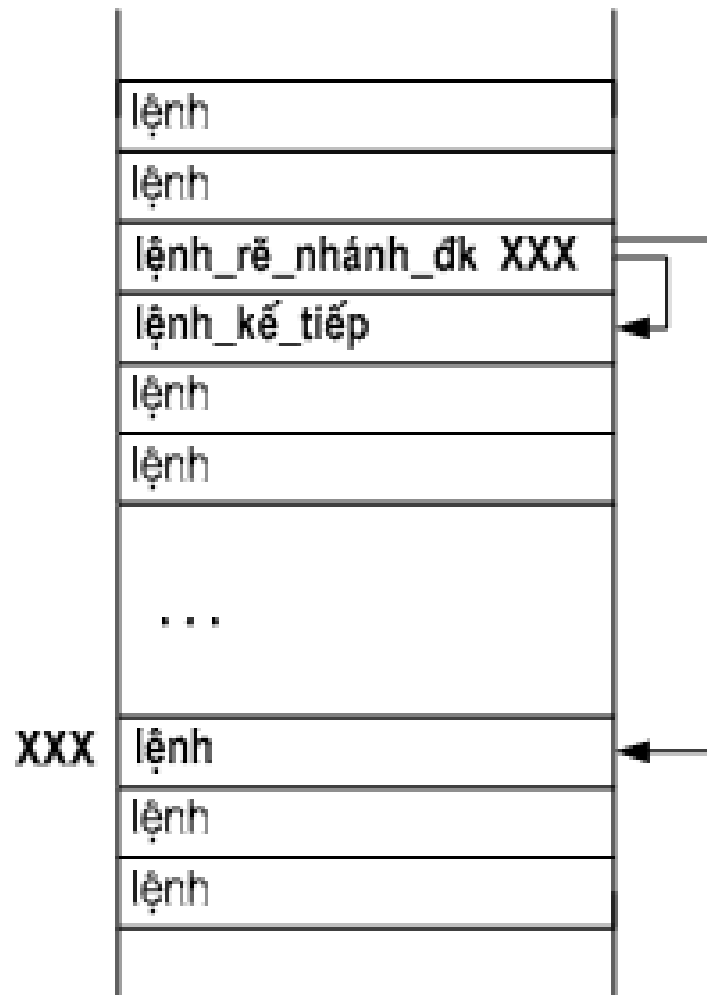
- Chuyển tới thực hiện lệnh ở vị trí có địa chỉ XXX:
 $PC \leftarrow XXX$



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
 $PC \leftarrow XXX$
 - 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

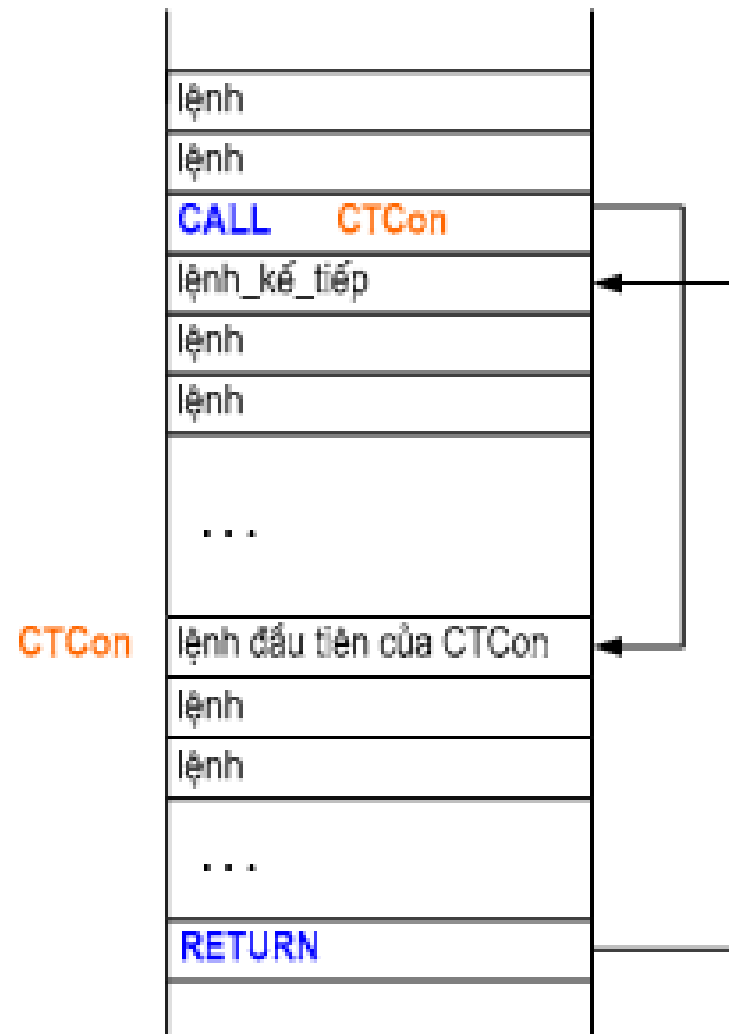
Rẽ nhánh có điều kiện



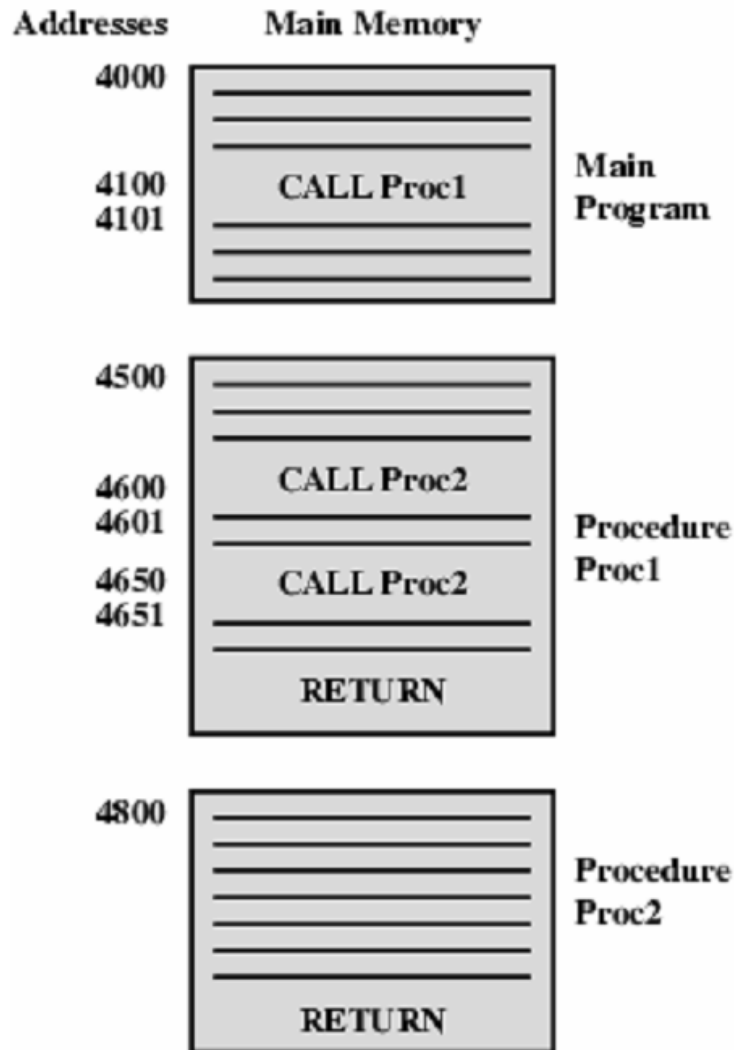
Lệnh call và return

- 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

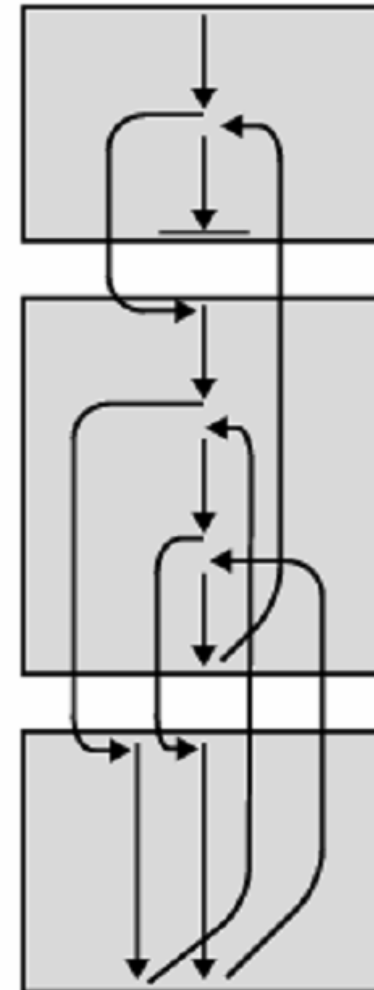
Lệnh call và return



Các thủ tục lồng nhau

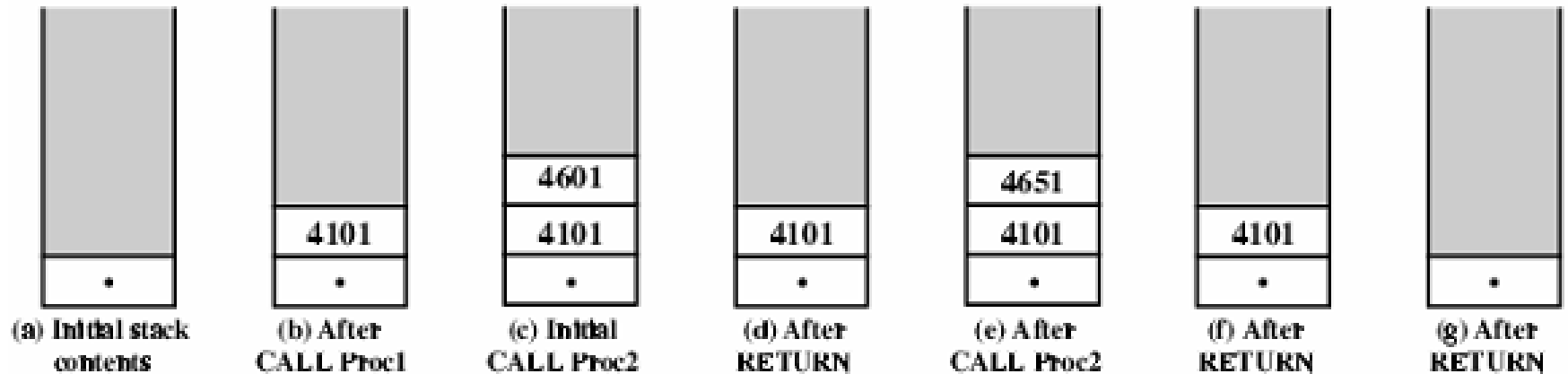


(a) Calls and returns



(b) Execution sequence

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

Nội dung

4.1 Khái niệm tập lệnh, thành phần lệnh máy

4.2 Chu trình thực hiện lệnh máy

4.3 Một số dạng lệnh thông dụng

4.4 Các dạng toán hạng

4.5 Các chế độ địa chỉ

4.4 Các dạng toán hạng

Các dạng toán hạng của lệnh:

- Toán hạng dạng 3 địa chỉ:
 - Dạng: **opcode addr1, addr2, addr3**
 - Mỗi địa chỉ addr1, addr2, addr3 tham chiếu đến một ô nhớ hoặc một thanh ghi
- Toán hạng dạng 2 địa chỉ:
 - Dạng: **opcode addr1, addr2**
 - Mỗi địa chỉ addr1, addr2, tham chiếu đến một ô nhớ hoặc một thanh ghi

4.4 Các dạng toán hạng

- Toán hạng dạng 1 địa chỉ:
 - Dạng: **opcode addr**
 - Địa chỉ addr tham chiếu đến một ô nhớ hoặc một thanh ghi.
- Toán hạng 0 địa chỉ:
 - Thường được sử dụng trong các lệnh thao tác với ngăn xếp: **PUSH** và **POP**

Các ví dụ phân tích trong tập lệnh

Nội dung

4.1 Khái niệm tập lệnh, thành phần lệnh máy

4.2 Chu trình thực hiện lệnh máy

4.3 Một số dạng lệnh thông dụng

4.4 Các dạng toán hạng

4.5 Các chế độ địa chỉ

4.5 Các chế độ địa chỉ

Khái niệm về định địa chỉ (addressing)

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

Các phương pháp định địa chỉ thông dụng

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

Định địa chỉ tức thì

- Toán hạng nằm ngay trong Trường địa chỉ của lệnh
- Chỉ có thể là toán hạng nguồn
- Ví dụ:

ADD R1, 5 ; $R1 \leftarrow R1 + 5$

- Không tham chiếu bộ nhớ
- Truy nhập toán hạng rất nhanh
- Dải giá trị của toán hạng bị hạn chế

Mã thao tác		Toán hạng
-------------	--	-----------

Đánh địa chỉ tức thì

MOV BL, 44 ; Copy số thập phân 44 vào thanh ghi BL

MOV AX, 44H ; Copy 0044H vào thanh ghi AX

MOV AL, 'A' ; Copy mã ASCII của A vào thanh ghi AL

MOV DS, 0FF0H ; không hợp lệ

MOV AX, 0FF0H ;

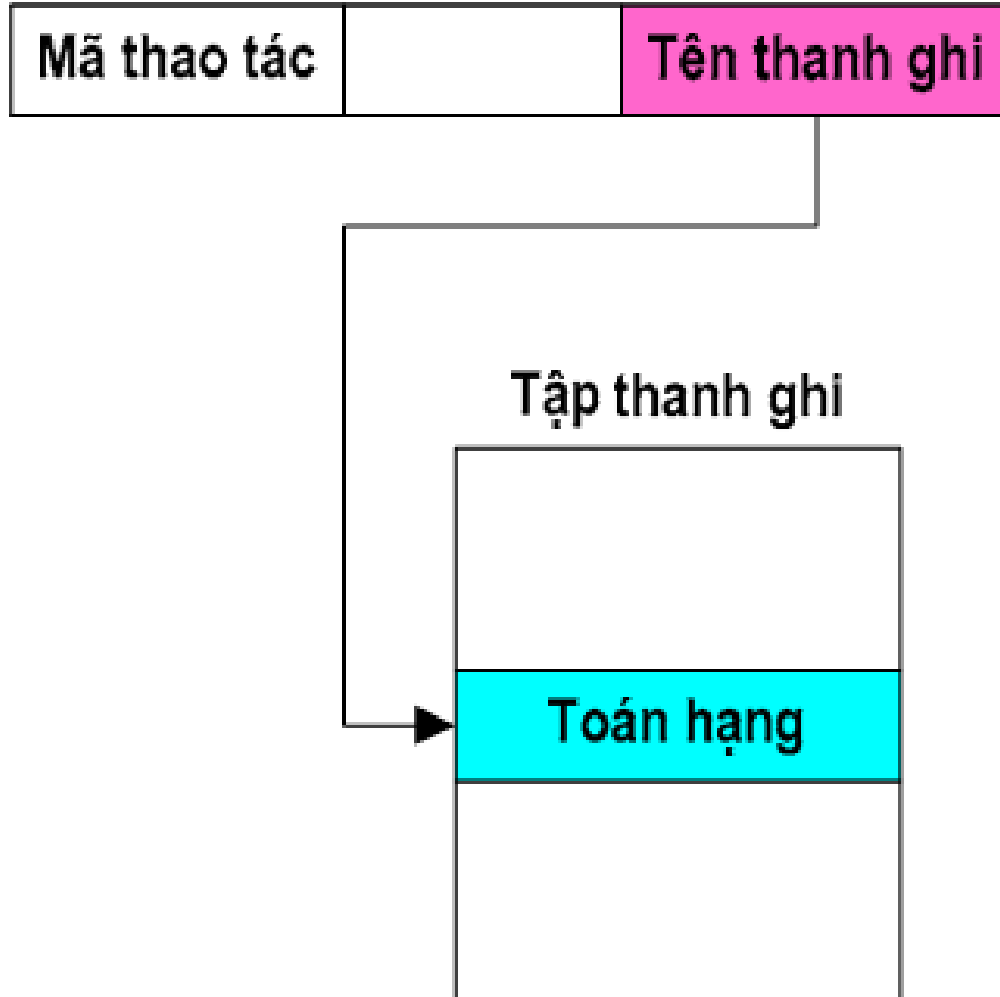
MOV DS, AX ;

MOV [BX], 10 ; copy số thập phân 10 vào ô nhớ
DS:BX

Định địa chỉ thanh ghi

- Toán hạng được chứa trong thanh ghi có tên trong Trường địa chỉ
- Ví dụ:
$$\text{ADD } R1, R2 \quad ; R1 \leftarrow R1 + R2$$
- Số lượng thanh ghi ít \rightarrow Trường địa chỉ chỉ cần ít bit
- Không tham chiếu bộ nhớ
- Truy nhập toán hạng nhanh
- Tăng số lượng thanh ghi \rightarrow hiệu quả hơn

Định địa chỉ thanh ghi



MOV BX, DX

MOV AL, BL

~~MOV AL, BX;~~

~~MOV ES, DS;~~ (segment to segment)

~~MOV CS, AX;~~ không hợp lệ vì CS không được dùng làm thanh ghi đích

ADD AL, DL

Định địa chỉ trực tiếp

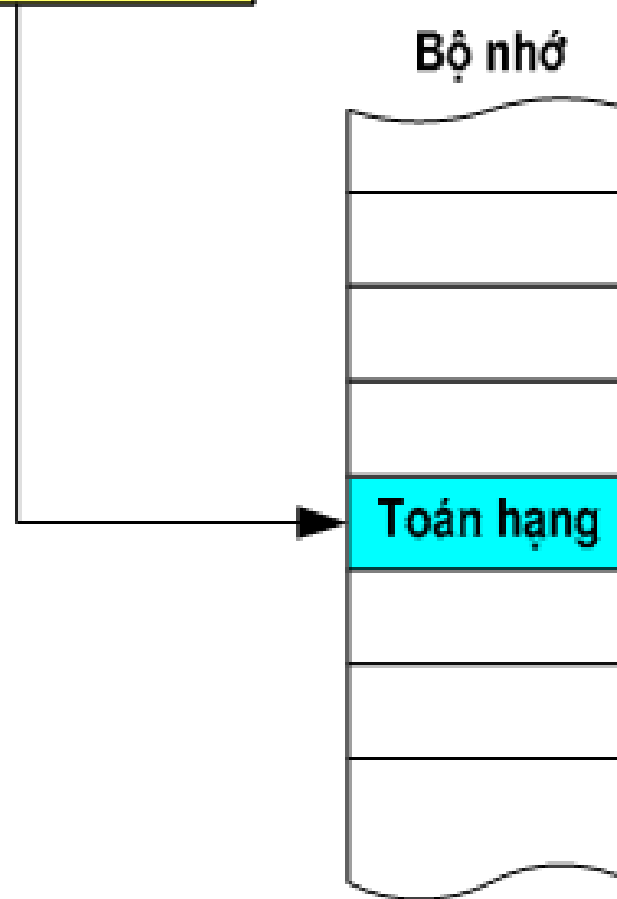
- Toán hạng là ngăn nhớ có địa chỉ được chỉ ra trực tiếp trong Trường địa chỉ của lệnh
- Ví dụ: `ADD R1, A` ; $R1 \leftarrow R1 + (A)$
 - Cộng nội dung thanh ghi R1 với nội dung của ngăn nhớ có địa chỉ là A
 - Tìm toán hạng trong bộ nhớ ở địa chỉ A
- CPU tham chiếu bộ nhớ một lần để truy nhập dữ liệu

Định địa chỉ trực tiếp



MOV AL, [1234H] ; Copy
nội dung ô nhớ có địa
chỉ DS:1234 vào AL

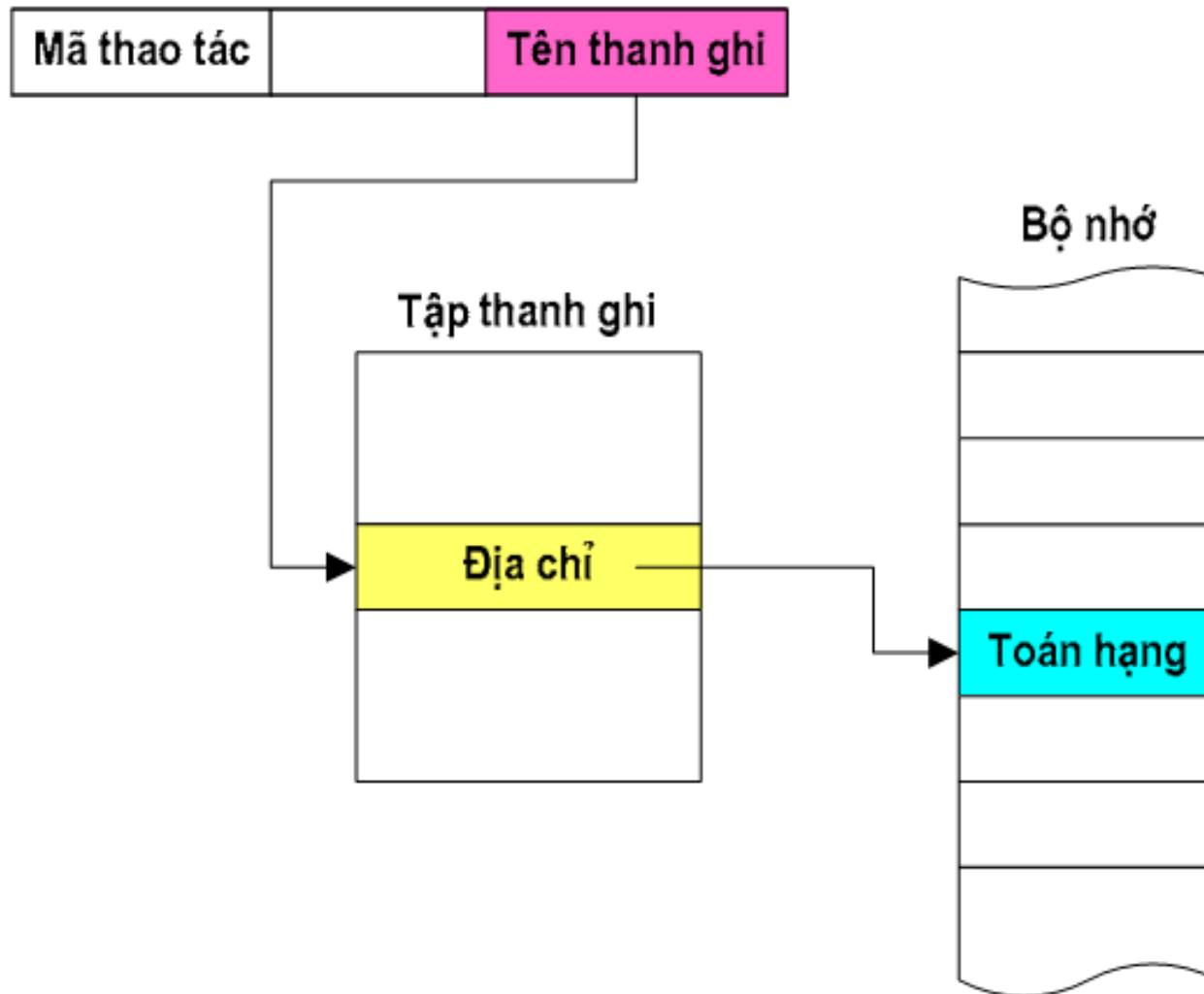
MOV [4320H], CX ;
Copy nội dung của CX
vào 2 ô nhớ liên tiếp
DS:4320 và DS:4321



Định địa chỉ gián tiếp qua thanh ghi

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

Đánh địa chỉ gián tiếp qua thanh ghi



Đánh địa chỉ gián tiếp qua thanh ghi

MOV AL, [BX] ; Copy nội dung ô nhớ có địa chỉ DS:BX vào AL

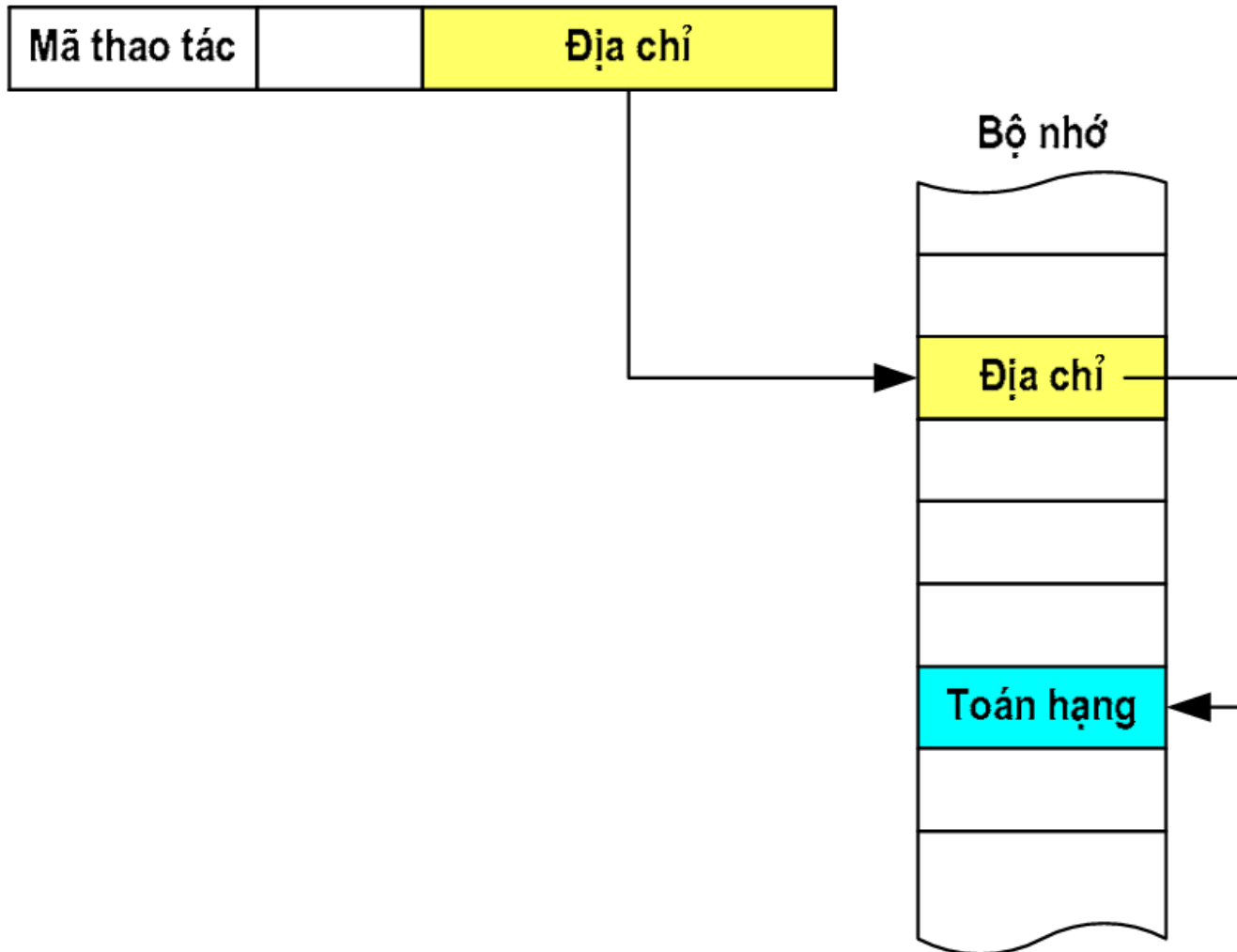
MOV [SI], CL ; Copy nội dung của CL vào ô nhớ có địa chỉ DS:SI

MOV [DI], AX ; copy nội dung của AX vào 2 ô nhớ liên tiếp DS: DI và DS: (DI +1)

Đánh địa chỉ gián tiếp qua ô nhớ

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

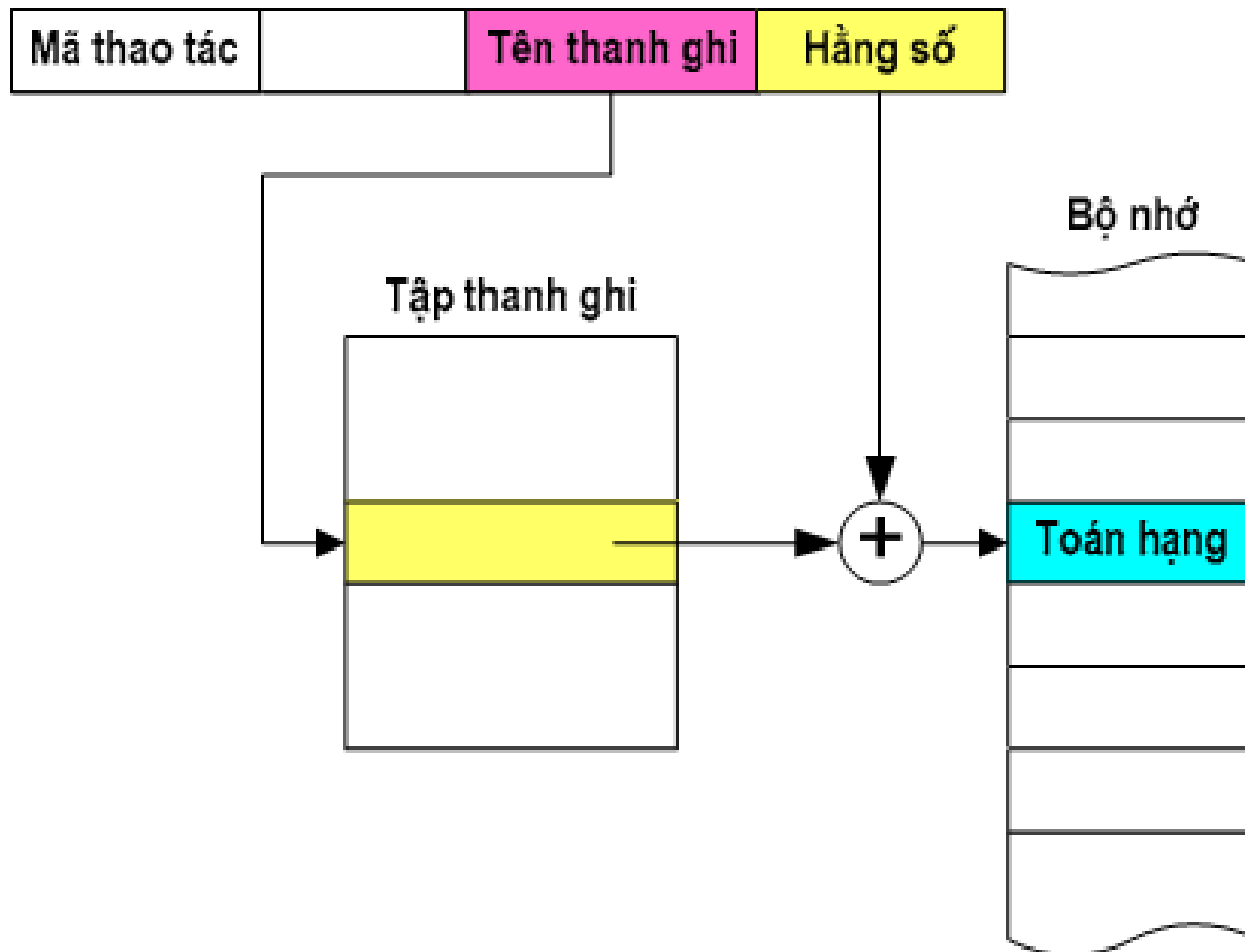
Đánh địa chỉ gián tiếp qua ô nhớ



Định địa chỉ dịch chuyển (tương đối)

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

Đánh địa chỉ tương đối



Các dạng của đánh địa chỉ tương đối

- Địa chỉ hoá tương đối với PC
 - Thanh ghi là Bộ đếm chương trình PC
 - Toán hạng có địa chỉ cách ngăn nhớ được trở bởi PC một độ lệch xác định
- Định địa chỉ cơ sở
 - Thanh ghi chứa địa chỉ cơ sở
 - Hằng số là chỉ số
- Định địa chỉ chỉ số
 - Hằng số là địa chỉ cơ sở
 - Thanh ghi chứa chỉ số

Chế độ địa chỉ tương đối cơ sở

- Một toán hạng là thanh ghi cơ sở BX, BP và các hằng số biểu diễn giá trị dịch chuyển
- Toán hạng kia chỉ có thể là thanh ghi
- Ví dụ:
 - `MOV CX, [BX]+10` ; Copy nội dung 2 ô nhớ liên tiếp có địa chỉ `DS:BX+10` và `DS:BX+11` vào CX
 - `MOV CX, [BX+10]` ; Cách viết khác của lệnh trên
 - `MOV AL, [BP]+5` ; copy nội dung của ô nhớ `SS:BP+5` vào thanh ghi AL

Chế độ địa chỉ tương đối chỉ số

- Một toán hạng là thanh ghi chỉ số SI, DI và các hằng số biểu diễn giá trị dịch chuyển
- Toán hạng kia chỉ có thể là thanh ghi
- Ví dụ:
 - `MOV AX, [SI]+10` ; Copy nội dung 2 ô nhớ liên tiếp có địa chỉ `DS:SI+10` và `DS:SI+11` vào AX
 - `MOV AX, [SI+10]` ; Cách viết khác của lệnh trên
 - `MOV AL, [DI]+5` ; copy nội dung của ô nhớ `DS:DI+5` vào thanh ghi AL

Chế độ địa chỉ tương đối chỉ số cơ sở

- Ví dụ:
 - `MOV AX, [BX] [SI]+8` ; Copy nội dung 2 ô nhớ liên tiếp có địa chỉ `DS:BX+SI+8` và `DS:BX+SI+9` vào `AX`
 - `MOV AX, [BX+SI+8]` ; Cách viết khác của lệnh trên
 - `MOV CL, [BP+DI+5]` ; copy nội dung của ô nhớ `SS:BP+DI+5` vào thanh ghi `CL`

Tổng kết các chế độ địa chỉ của 8086

Chế độ địa chỉ	Toán hạng	Thanh ghi đoạn ngầm định
Thanh ghi	Thanh ghi	
Tức thì	Dữ liệu	
Trực tiếp	[offset]	DS
Gián tiếp qua thanh ghi	[BX], [SI], [DI]	DS,DS,DS
Tương đối cơ sở	[BX] + dịch chuyển [BP] + dịch chuyển	DS SS
Tương đối chỉ số	[DI] + dịch chuyển [SI] + dịch chuyển	DS DS
Tương đối chỉ số cơ sở	[BX] + [DI] + dịch chuyển [BX] + [SI] + dịch chuyển [BP] + [DI] + dịch chuyển [BP] + [SI] + dịch chuyển	DS DS SS SS