

CHƯƠNG 5

Các chế độ đánh địa chỉ của 8051

CPC có thể truy cập dữ liệu theo nhiều cách khác nhau. Dữ liệu có thể ở trong một thanh ghi hoặc trong bộ nhớ hoặc được cho như một giá trị tức thời các cách truy cập dữ liệu khác nhau được gọi là các chế độ đánh địa chỉ. Chương này chúng ta bàn luận về các chế độ đánh địa chỉ của 8051 trong phạm vi một số ví dụ.

Các chế độ đánh địa chỉ khác nhau của bộ vi xử lý được xác định như nó được thiết kế và do vậy người lập trình không thể đánh địa chỉ khác nhau là:

1. tức thời
2. Theo thanh ghi
3. Trực tiếp
4. gián tiếp qua thanh ghi
5. Theo chỉ số

5.1 Các chế độ đánh địa chỉ tức thời và theo thanh ghi

5.1.1 Chế độ đánh địa chỉ tức thời

Trong chế độ đánh địa chỉ này toán hạng nguồn là một hằng số. Và như tên gọi của nó thì khi một lệnh được hợp dịch toán hạng đi tức thì ngay sau mã lệnh. Lưu ý rằng trước dữ liệu tức thời phải được đặt dấu (#) chế độ đánh địa chỉ này có thể được dùng để nạp thông tin vào bất kỳ thanh ghi nào kể cả thanh ghi con trỏ dữ liệu DPTR. Ví dụ:

MOV	A, # 25H	; Nạp giá trị 25H vào thanh ghi A
MOV	R4, #62	; Nạp giá trị 62 thập phân vào R4
MOV	B, #40H	; Nạp giá trị 40 H vào thanh ghi B
MOV	DPTR, #4521H	; Nạp 4512H vào con trỏ dữ liệu DPTR

Mặc dù thanh ghi DPTR là 16 bit nó cũng có thể được truy cập như 2 thanh ghi 8 bit DPH và DPL trong đó DPH là byte cao và DPL là byte thấp. Xét đoạn mã dưới đây:

MOV	DPTR, #2550H
MOV	A, #50H
MOV	DPH, #25H

Cũng lưu ý rằng lệnh dưới đây có thể tạo ra lỗi vì giá trị nạp vào DPTR lớn hơn 16 bit:

MOV	DPTR, # 68975	; Giá trị không hợp lệ > 65535 (FFFFH)
-----	---------------	--

Ta có thể dùng chỉ lệnh EQU để truy cập dữ liệu tức thời như sau

	COUNT	EQU	30	
	
	MOV	R4, #COUNT		; R4 = 1E (30 = 1EH)
	MOV	DPTR, #MYDATA		; DPTR = 200H
MYDATA:	ORG	200H		
	DB	"America"		

Lưu ý rằng ta cũng có thể sử dụng chế độ đánh được chỉ tức thời để gửi dữ liệu đến các cổng của 8051.

Ví dụ “MOV P1, #55H” là một lệnh hợp lệ.

5.1.2 chế độ đánh địa chỉ theo thanh ghi:

Chế độ đánh địa chỉ theo thanh ghi liên quan đến việc sử dụng các thanh ghi để dữ liệu cần được thao tác các ví dụ về đánh địa chỉ theo thanh ghi như sau:

MOV	A, R0	; Sao nội dung thanh ghi R0 vào thanh ghi A
MOV	R2, A	; Sao nội dung thanh ghi A vào thanh ghi R2
ADD	A, R5	; Cộng nội dung thanh ghi R5 vào thanh ghi A
ADD	A, R7	; Cộng nội dung thanh ghi R7 vào thanh ghi A
MOV	R6, A	; Lưu nội dung thanh ghi A vào thanh ghi R6

Cũng nên lưu ý rằng các thanh ghi nguồn và đích phải phù hợp về kích thước. Hay nói cách khác, nếu viết “MOV DPTR, A” sẽ cho một lỗi vì nguồn là thanh ghi 8 bit và đích lại là thanh ghi 16 bit. Xét đoạn mã sau:

```
MOV DPTR, #25F5H
MOV R7, DPL
MOV R6, DPH
```

Để ý rằng ta có thể chuyển dữ liệu giữa thanh ghi tích lũy A và thanh ghi Rn (n từ 0 đến 7) nhưng việc chuyển dữ liệu giữa các thanh ghi Rn thì không được phép. Ví dụ, lệnh “MOV R4, R7” là không hợp lệ.

Trong hai chế độ đánh địa chỉ đầu tiên, các toán hạng có thể hoặc ở bên trong một trong các thanh ghi hoặc được gắn liền với lệnh. Trong hầu hết các chương trình dữ liệu cần được xử lý thường ở trong một số ngăn của bộ nhớ RAM hoặc trong không gian mà của ROM. Có rất nhiều cách để truy cập dữ liệu này mà phần tiếp theo sẽ xét đến.

5.2 Truy cập bộ nhớ sử dụng các chế độ đánh địa chỉ khác nhau.

5.2.1 Chế độ đánh địa chỉ trực tiếp.

Như đã nói ở chương 2 trong 8051 có 128 byte bộ nhớ RAM. Bộ nhớ RAM được gán các địa chỉ từ 00 đến FFH và được phân chia như sau:

1. Các ngăn nhớ từ 00 đến 1FH được gán cho các bảng thanh ghi và ngăn xếp.
2. Các ngăn nhớ từ 20H đến 2FH được dành cho không gian đánh địa chỉ theo bit để lưu các dữ liệu 1 bit.
3. Các ngăn nhớ từ 30H đến 7FH là không gian để lưu dữ liệu có kích thước 1byte.

Mặc dù toàn bộ byte của bộ nhớ RAM có thể được truy cập bằng chế độ đánh địa chỉ trực tiếp, nhưng chế độ này thường được sử dụng nhất để truy cập các ngăn nhớ RAM từ 30H đến 7FH. Đây là do một thực tế là các ngăn nhớ dành cho bảng ghi được truy cập bằng thanh ghi theo các tên gọi của chúng là R0 - R7 còn các ngăn nhớ khác của RAM thì không có tên như vậy. Trong chế độ đánh địa chỉ trực tiếp thì dữ liệu ở trong một ngăn nhớ RAM mà địa chỉ của nó được biết và địa chỉ này được cho như là một phần của lệnh. Khác với chế độ đánh địa chỉ tức thì mà toán hạng tự nó được cấp với lệnh. Dấu (#) là sự phân biệt giữa hai chế độ đánh địa chỉ. Xét các ví dụ dưới đây và lưu ý rằng các lệnh không có dấu (#):

```
MOV R0, 40H ; Lưu nội dung của ngăn nhớ 40H của RAM vào R0
```

MOV	56H, A	; Lưu nội dung thanh ghi A vào ngăn nhớ 56H của RAM
MOV	R4, 7FH	; Chuyển nội dung ngăn nhớ 7FH của RAM vào R4

Như đã nói ở trước thì các ngăn nhớ từ 0 đến 7 của RAM được cấp cho bằng 0 của các thanh ghi R0 - R7. Các thanh ghi này có thể được truy cập theo 2 cách như sau:

MOV	A, 4	; Hai lệnh này giống nhau đều sao nội dung thanh ghi R4 vào A
MOV	A, R4	
MOV	A, 7	; Hai lệnh này đều như nhau là sao nội dung R7 vào thanh ghi A
MOV	A, R7	

Để nhấn mạnh sự quan trọng của dấu (#) trong các lệnh của 8051. Xét các mã cho sau đây:

MOV	R2, #05	; Gán R2=05
MOV	A, 2	; Sao nội dung thanh ghi R2 vào A
MOV	B, 2	; Sao nội dung thanh ghi R2 vào B
MOC	7,2	; Sao nội dung thanh ghi R7 vì lệnh "MOV R7, R2" là không hợp lệ.

Mặc dù sử dụng các tên R0 - R7 dễ hơn các địa chỉ bộ nhớ của chúng nhưng các ngăn nhớ 30H đến 7FH của RAM không thể được truy cập theo bất kỳ cách nào khác là theo địa chỉ của chúng vì chúng không có tên.

5.2.2 các thanh ghi SFSR và các địa chỉ của chúng.

Trong các thanh ghi được nói đến từ trước đến giờ ta thấy rằng các thanh ghi R0 - R7 là một phần trong 128 byte của bộ nhớ RAM. Vậy còn các thanh ghi A, B, PSW và DPTR là một bộ phận của nhóm các thanh ghi nhìn chung được gọi là các thanh ghi đặc biệt SFR (Special Function Register). Có rất nhiều thanh ghi với chức năng đặc biệt và chúng được sử dụng rất rộng rãi mà ta sẽ trình bày ở các chương sáu. Các thanh ghi FR có thể được truy cập theo tên của chúng (mà dễ hơn rất nhiều) hoặc theo các địa chỉ của chúng. Ví dụ địa chỉ của thanh ghi A là E0H và thanh ghi B là F0H như cho ở trong bảng 5.1. Hãy để ý đến những cặp lệnh có cùng ý nghĩa dưới đây:

MOV	0E0H, #55H	; Nạp 55H vào thanh ghi A(A=55H)
MOV	A, #55H	;
MOV	0F0H, #25H	; Nạp 25H vào thanh ghi B (B = 25)
MOV	3, #25H	;
MOV	0E0H	; Sao nội dung thanh ghi R2 vào A
MOV	A, R2	;
MOV	0F0	; Sao nội dung thanh ghi R0 vào B
MOV	B, R0	;

Bảng 5.1 dưới đây liệt kê các thanh ghi chức năng đặc biệt SFR của 8051 và các địa chỉ của chúng. Cần phải lưu ý đến hai điểm sau về các địa chỉ của SFR:

1. Các thanh ghi SFR có địa chỉ nằm giữa 80H và FFH các địa chỉ này ở trên 80H, vì các địa chỉ từ 00 đến 7FH là địa chỉ của bộ nhớ RAM bên trong 8051.

2. không phải tất cả mọi địa chỉ từ 80H đến FFH đều do SFH sử dụng, nhưng vị trí ngăn nhớ từ 80H đến FFH chưa dùng là để dự trữ và lập trình viên 8051 cũng không được sử dụng.

Bảng 5.1: Các địa chỉ của thanh ghi chức năng đặc biệt SFR

Lệnh	Tên	Địa chỉ
ACC*	Thanh ghi tích lũy (thanh ghi tổng) A	0E0H
B*	Thanh ghi B	0F0H
PSW*	Từ trạng thái chương trình	0D0H
SP	Con trỏ ngăn xếp	81H
DPTR	Con trỏ dữ liệu hai byte	
DPL	Byte thấp của DPTR	82H
DPH	Byte cao của DPTR	83H
P0*	Cổng 0	80H
P1*	Cổng 1	90H
P2*	Cổng 2	0A0H
P3*	Cổng 3	0B0H
IP*	Điều khiển ưu tiên ngắt	0B8H
IE*	Điều khiển cho phép ngắt	A08H
TMOD	Điều khiển chế độ bộ đếm/ Bộ định thời	89H
TCON*	Điều khiển bộ đếm/ Bộ định thời	88H
T2CON*	Điều khiển bộ đếm/ Bộ định thời 2	0C8H
T2MOD	Điều khiển chế độ bộ đếm/ Bộ định thời 2	0C9H
TH0	Byte cao của bộ đếm/ Bộ định thời 0	8CH
TL0	Byte thấp của bộ đếm/ Bộ định thời 0	8AH
TH1	Byte cao của bộ đếm/ Bộ định thời 1	8DH
TL1	Byte thấp của bộ đếm/ Bộ định thời 1	8BH
TH2	Byte cao của bộ đếm/ Bộ định thời 2	0CDH
TL2	Byte thấp của bộ đếm/ Bộ định thời 2	0CCH
RCAP2H	Byte cao của thanh ghi bộ đếm/ Bộ định thời 2	0CBH
RCAP2L	Byte thấp của thanh ghi bộ đếm/ Bộ định thời 2	0CAH
SCON*	Điều khiển nối tiếp	98H
SBUF	Bộ đệm dữ liệu nối tiếp	99H
PCON	Điều khiển công suất	87H

*Các thanh ghi có thể đánh địa chỉ theo bit.

Xét theo chế độ đánh địa chỉ trực tiếp thì cần phải lưu ý rằng giá trị địa chỉ được giới hạn đến 1byte, 00 - FFH. Điều này có nghĩa là việc sử dụng của chế độ đánh địa chỉ này bị giới hạn bởi việc truy cập các vị trí ngăn nhớ của RAM và các thanh ghi với địa chỉ được cho bên trong 8051.

Ví dụ 5.1:

Viết chương trình để gửi 55H đến cổng P1 và P2 sử dụng hoặc

- Tên các cổng
- Hoặc địa chỉ các cổng

Lời giải:

- MOV A, #55H ; A = 55H

MOV	P1, A	; P1 = 55H
MOV	P2, A	; P2 = 55H

b) Từ bảng 5.1 ta lấy đại chỉ cổng P1 là 80H và P2 là A0H

MOV	A, #55H	; A = 55H
MOV	80H, A	; P1 = 55H
MOV	0A0H, A	; P2 = 55H

5.2.3 Ngăn xếp và chế độ đánh địa chỉ trực tiếp.

Một công dụng chính khác của chế độ đánh địa chỉ trực tiếp là ngăn xếp. Trong họ 8051 chỉ có chế độ đánh địa chỉ trực tiếp là được phép đẩy vào ngăn xếp. Do vậy, một lệnh như “PVSH A” là không hợp lệ. Việc đẩy thanh ghi A vào ngăn xếp phải được viết dưới dạng “PVAH 0E0H” với 0E0H là địa chỉ của thanh ghi A. Tương tự như vậy để đẩy thanh ghi R3 rãnh 0 vào ngăn xếp ta phải viết là “PVSH 03”. Chế độ đánh địa chỉ trực tiếp phải được sử dụng cho cả lệnh POP. Ví dụ “POP 04” sẽ kéo đỉnh của ngăn xếp vào thanh ghi R4 rãnh 0.

Ví dụ 5.2:

Trình bày mã để đẩy thanh ghi R5, R6 và A vào ngăn xếp và sau đó kéo chùng ngược trở lại R2, R3 và B tương ứng.

Lời giải:

PUSH	05	; Đẩy R5 vào ngăn xếp
PUSH	06	; Đẩy R6 vào ngăn xếp
PUSH	0E0H	; Đẩy thanh ghi A vào ngăn xếp
POP	0F0H	; Kéo đỉnh ngăn xếp cho vào thanh ghi B
		; Bây giờ B = A
POP	02	; Kéo đỉnh ngăn xếp cho vào thanh ghi R2
		; Bây giờ R2 = R6
POP	03	; Kéo đỉnh ngăn xếp cho vào thanh ghi
		; Bây giờ R3 = R5

5.2.4 chế độ đánh địa chỉ gián tiếp thanh ghi.

Trong chế độ này, một thanh ghi được sử dụng như một con trỏ đến dữ liệu. Nếu dữ liệu ở bên trong CPU thì chỉ các thanh ghi R0 và R1 được sử dụng cho mục đích này. Hay nói cách khác các thanh ghi R2 - R7 không có thể dùng được để giữ địa chỉ của toán hạng nằm trong RAM khi sử dụng chế độ đánh địa chỉ này khi R0 và R1 được dùng như các con trỏ, nghĩa là khi chúng giữ các địa chỉ của các ngăn nhớ RAM thì trước chúng phải đặt dấu (@) như chỉ ra dưới đây.

MOV	A, @ R0	; Chuyển nội dung của ngăn nhớ RAM có địa chỉ trong R0 vào A
MOV	@ R1, B	; Chuyển nội dung của B vào ngăn nhớ RAM có địa chỉ ở R1

Lưu ý rằng R0 cũng như R1 luôn có dấu “@” đứng trước. Khi không có dấu này thì đó là lệnh chuyển nội dung các thanh ghi R0 và R1 chứ không phải dữ liệu ngăn nhớ mà địa chỉ có trong R0 và R1.

Ví dụ 5.3:

Viết chương trình để sao chép giá trị 55H vào ngăn nhớ RAM tại địa chỉ 40H đến 44H sử dụng:

- a) Chế độ đánh địa chỉ trực tiếp
- b) Chế độ đánh địa chỉ gián tiếp thanh ghi không dùng vòng lặp
- c) Chế độ b có dùng vòng lặp

Lời giải:

	MOV	A, #55H	; Nạp A giá trị 55H
	MOV	40H, A	; Sao chép A vào ngăn nhớ RAM 40H
	MOV	41H, A	; Sao chép A vào ngăn nhớ RAM 41H
	MOV	42H, A	; Sao chép A vào ngăn nhớ RAM 42H
	MOV	43H, A	; Sao chép A vào ngăn nhớ RAM 43H
	MOV	44H, A	; Sao chép A vào ngăn nhớ RAM 44H
b)			
	MOV	A, # 55H	; Nạp vào A giá trị 55H
	MOV	R0, #40H	; Nạp con trỏ R0 = 40 H
	MOV	@R0, A	; Sao chép A vào vị trí ngăn nhớ RAM do R0 chỉ đến
	INC	R0	; Tăng con trỏ. Bây giờ R0 = 41H
	MOV	@R0, A	; Sao chép A vào vị trí ngăn nhớ RAM do R0 chỉ
	INC	R0	; Tăng con trỏ. Bây giờ R0 = 42H
	MOV	@R0, A	; Sao chép A vào vị trí ngăn nhớ RAM do R0 chỉ
	INC	R0	; Tăng con trỏ. Bây giờ R0 = 43H
	MOV	@R0, A	; Sao chép A vào vị trí ngăn nhớ RAM do R0 chỉ
	MOV	@R0, A	; Tăng con trỏ. Bây giờ R0 = 44H
	MOV	@R0, A	
c)			
	MOV	A, # 55H	; Nạp vào A giá trị 55H
	MOV	R0, #40H	; Nạp con trỏ địa chỉ ngăn nhớ RAM R0 = 40H
	MOV	R2, #05	; Nạp bộ đếm R2 = 5
AGAIN:	MOV	@R0, A	; Sao chép A vào vị trí ngăn nhớ RAM do R0 chỉ đến
	INC		; Tăng con trỏ R0
	DJNZ	R2, AGAIN	; Lặp lại cho đến khi bộ đếm = 0.

5.2.5 Ưu điểm của chế độ đánh địa chỉ gián tiếp thanh ghi.

Một trong những ưu điểm của chế độ đánh địa chỉ gián tiếp thanh ghi là nó làm cho việc truy cập dữ liệu năng động hơn so với chế độ đánh địa chỉ trực tiếp.

Ví dụ 5.3 trình bày trường hợp sao chép giá trị 55H vào các vị trí ngăn nhớ của RAM từ 40H đến 44H .

Lưu ý rằng lời giải b) có hai lệnh được lặp lại với một số lần. Ta có thể tạo ra vòng lặp với hai lệnh này như ở lời giải c). Lời giải c) là hiệu quả nhất và chỉ có thể khi sử dụng chế độ đánh địa chỉ gián tiếp qua thanh ghi. Vòng lặp là không thể trong chế độ đánh địa chỉ trực tiếp. Đây là sự khác nhau chủ yếu giữa đánh địa chỉ trực tiếp và gián tiếp.

Ví dụ 5.4:

Hãy viết chương trình để xóa 16 vị trí ngăn nhớ RAM bắt đầu tại địa chỉ 60H.

Lời giải:

CLR	A	; Xóa A=0
MOV	R1, #60H	; Nạp con trỏ. R1= 60H
MOV	R7, #16H	; Nạp bộ đếm, R7 = 16 (10 H dạng hex)

```

AGAIN:    MOV    @R1, A        ; Xoá vị trí ngăn nhớ RAM do R1 chỉ đến
          INC     R1           ; Tăng R1
          DJNZ    R7, AGAIN    ; Lặp lại cho đến khi bộ đếm = 0

```

Một ví dụ về cách sử dụng cả R0 và R1 trong chế độ đánh địa chỉ gián tiếp thanh ghi khi truyền khối được cho trong ví dụ 5.5.

Ví dụ 5.5:

Hãy viết chương trình để sao chép một khối 10 byte dữ liệu từ vị trí ngăn nhớ RAM bắt đầu từ 35H vào các vị trí ngăn nhớ RAM bắt đầu từ 60H

Lời giải:

```

          MOV     R0, # 35H    ; Con trỏ nguồn
          MOV     R1, # 60H    ; Con trỏ đích
          MOV     R3, # 10     ; Bộ đếm
BACK:     MOV     A, @R0       ; Lấy 1byte từ nguồn
          MOV     @R1, A       ; Sao chép nó đến đích
          INC     R0           ; Tăng con trỏ nguồn
          INC     R1           ; Tăng con trỏ đích
          DJNZ    R3, BACK     ; Lặp lại cho đến khi sao chép hết 10 byte

```

5.2.6 Hạn chế của chế độ đánh địa chỉ gián tiếp thanh ghi trong 8051.

Như đã nói ở phần trước rằng R0 và R1 là các thanh ghi duy nhất có thể được dùng để làm các con trỏ trong chế độ đánh địa chỉ gián tiếp thanh ghi. Vì R0 và R1 là các thanh ghi 8 bit, nên việc sử dụng của chúng bị hạn chế ở việc truy cập mọi thông tin trong các ngăn nhớ RAM bên trong (các ngăn nhớ từ 30H đến 7FH và các thanh ghi SFR). Tuy nhiên, nhiều khi ta cần truy cập dữ liệu được cất trong RAM ngoài hoặc trong không gian mã lệnh của ROM trên chip. Hoặc là truy cập bộ nhớ RAM ngoài hoặc ROM trên chip thì ta cần sử dụng thanh ghi 16 bit đó là DPTR.

5.2.7 Chế độ đánh địa chỉ theo chỉ số và truy cập bộ nhớ ROM trên chip.

Chế độ đánh địa chỉ theo chỉ số được sử dụng rộng rãi trong việc truy cập các phân tử dữ liệu của bảng trong không gian ROM chương trình của 8051. Lệnh được dùng cho mục đích này là “Move A, @ A + DPTR”. Thanh ghi 16 bit DPTR là thanh ghi A được dùng để tạo ra địa chỉ của phân tử dữ liệu được lưu cất trong ROM trên chip. Do các phân tử dữ liệu được cất trong không gian mã (chương trình) của ROM trên chip của 8051, nó phải dùng lệnh Move thay cho lệnh Mov (chữ C ở cuối lệnh là chỉ mã lệnh Code). Trong lệnh này thì nội dung của A được bổ xung vào thanh ghi 16 bit DPTR để tạo ra địa chỉ 16 bit của dữ liệu cần thiết. Xét ví dụ 5.6.

Ví dụ 5.6:

Giả sử từ “VSA” được lưu trong ROM có địa chỉ bắt đầu từ 200H và chương trình được ghi vào ROM bắt đầu từ địa chỉ 0. Hãy phân tích cách chương trình hoạt động và hãy phát biểu xem từ “VSA” sau chương trình này được cất vào đâu?

Lời giải:

```

          ORG     0000H        ; Bắt đầu đốt ROM tại địa chỉ 00H
          MOV     DPTR, #200H  ; Địa chỉ bảng trình bày DPTR = 200H
          CLA     A            ; Xoá thanh ghi A (A = 0)
          MOVC    A, @A + DPTR ; Lấy ký tự từ không gian nhớ chương trình
          MOV     R0, A        ; Cất nó vào trong R0

```

```

INC      DPTR      ; DPTR = 201, chỉ đến ký tự kế tiếp
CLR      A          ; Xoá thanh ghi A
MOVC     A, @A + DPTR ; Lấy ký tự kế tiếp
MOV      R1, A      ; Cất nó vào trong R1
INC      DPTR      ; DPTR = 202 con trở chỉ đến ký tự sau đó
CLA      A          ; Xoá thanh ghi A
MOVC     A, @A + DPTR ; Nhận ký tự kế tiếp
MOV      R2, A      ; Cất nó vào R2
HERE:    SJMP      HERE ; Dừng lại ở đây.
; Dữ liệu được đót trong không gian mã lệnh tại địa chỉ 200H
ORG 200H
MYDATA:  DB "VSA"
END      ; Kết thúc chương trình

```

Ở trong chương trình nói trên thì các vị trí ngăn nhớ ROM chương trình 200H - 2002H có các nội dung sau:

200 = ('U'); 201 = ('S') và 202 = ('A').

Chúng ta bắt đầu với DPTR = 200H và A = 0. Lệnh "MOVC A, @ A + DPTR" chuyển nội dung của vị trí nhớ 200H trong ROM (200H + 0 = 200H) vào A.

Thanh ghi A chứa giá trị 55H là giá trị mà ASC của ký tự "U". ký tự này được cất vào R0. Kế đó, DPTR được tăng lên tạo thành DPTR = 201H. A lại được xoá về 0 để lấy nội dung của vị trí nhớ kế tiếp trong ROM là 201H chứa ký tự "S". Sau khi chương trình này chạy ta có R0 = 55H, R1 = 53H và R2 = 41H là các mã ASCII của các ký tự "U", "S" và "A".

Ví dụ 5.7:

Giả sử không gian ROM bắt đầu từ địa chỉ 250H có chứa "America", hãy viết chương trình để truyền các byte vào các vị trí ngăn nhớ RAM bắt đầu từ địa chỉ 40H.

Lời giải

; (a) Phương pháp này sử dụng một bộ đếm

```

ORG      000
MOV      DPTR, #MYDATA ; Nạp con trỏ ROM
MOV      R0, #40H      ; Nạp con trỏ RAM
MOV      R2, #7        ; Nạp bộ đếm
BACK:    CLR      A      ; Xoá thanh ghi A
MOVC     A, @A + DPTR   ; Chuyển dữ liệu từ không gian mã
MOV      R0, A          ; Cất nó vào ngăn nhớ RAM
INC      DPTR           ; Tăng con trỏ ROM
INC      R0             ; Tăng con trỏ RAM
DJNZ     R2, BACK      ; Lặp lại cho đến khi bộ đếm = 0
HERE:    SJMP      HERE
;----- -- không gian mã của ROM trên chip dùng để cất dữ liệu ORG 250H

```

```

MYDATA:  DB      "AMER1CA"
END

```

; (b) phương pháp này sử dụng ký tự null để kết thúc chuỗi

```

ORG      000
MOV      DPTR, #MYDATA ; Nạp con trỏ ROM
MOV      R0, #40H      ; Nạp con trỏ RAM
BACK:    CLR      A      ; Xoá thanh ghi A (A=0)

```



```

MOV C    A, @A + DPTR    ; Chuyển dữ liệu từ không gian mã
JZ       HERE            ; Thoát ra nếu có ký tự Null
MOV      DPTR, #MYDATA   ; Gán nó vào ngădn nhớ của RAM
INC      @R0, A           ; Tăng con trỏ ROM
INC      R0              ; Tăng con trỏ RAM
SJM      BACK            ; Lặp lại
HERE:    SJMP             HERE
;----- không gian mã của ROM trên chip dùng để cất dữ liệu ORG 250H
MYADTA:  DB "AMER1CA", 0 ; Ký tự Null để kết thúc chuỗi END

```

Lưu ý đến cách ta sử dụng lệnh JZ để phát hiện ký tự NOLL khi kết thúc chuỗi

5.2.8 Bảng sắp xếp và sử dụng chế độ đánh địa chỉ theo chỉ số.

Bảng sắp xếp là khái niệm được sử dụng rất rộng rãi trong lập trình các bộ vi xử lý. Nó cho phép truy cập các phần tử của một bảng thường xuyên được sử dụng với thao tác cực tiểu. Như một ví dụ, hãy giả thiết rằng đối với một ứng dụng nhất định ta cần x^2 giá trị trong phạm vi 0 đến 9. Ta có thể sử dụng một bảng sắp xếp thay cho việc tính toán nó. Điều này được chỉ ra trong ví dụ 5.8.

Ví dụ 5.8

Hãy viết một chương trình để lấy x giá trị cổng P1 và gửi giá trị x^2 tới cổng P2 liên tục.

Lời giải:

```

ORG      000
MOV      DPTR, #300H    ; Nạp địa chỉ bảng sắp xếp
MOV      A, #0FFH       ; Nạp A giá trị FFH
MOV      P1, A           ; Đặt cổng P1 là đầu vào
BACK:    MOV      A, P1   ; Lấy giá trị X từ P1
         MOVC     A, @A + DPTR ; Lấy giá trị X từ bảng XSDQ-TABLE
         MOV      P2, A   ; Xuất nó ra cổng P2
         SJMP     BACK    ; Lặp lại

ORG      300H
XSQR - TABLE:
DB       0, 1, 4, 9, 16, 25, 36, 49, 64, 81
END

```

Lưu ý bảng lệnh đầu tiên có thể thay bằng "MOV DPTR, #XSQR - TABLE".

Ví dụ 5.9:

Trả lời các câu hỏi sau cho ví dụ 5.8.

- Hãy chỉ ra nội dung các vị trí 300 - 309H của ROM
- Tại vị trí nào của ROM có giá trị 6 và giá trị bao nhiêu
- Giả sử P1 có giá trị là 9 thì giá trị P2 là bao nhiêu (ở dạng nhị phân)?

Lời giải:

- Các giá trị trong các ngăn nhớ 300H - 309H của ROM là:

```

300 = (00)   301 = (01)   302 = (04)   303 = (09)
304 = (10)   4 × 4 = 16 = 10 in hex
305 = (19)   5 × 5 = 25 = 19 in hex
306 = (24)   6 × 6 = 36 = 24H
307 = (31)   308 = (40)           309 = (51)

```

b) vị trí chứa giá trị 306H và giá trị là 24H

c) 01010001B là giá trị nhị phân của 51H và 81 ($9^2 = 81$)

Ngoài việc sử dụng DPTR để truy cập không gian bộ nhớ ROM chương trình thì nó còn có thể được sử dụng để truy cập bộ nhớ ngoài nối với 8051 (chương 14).

Một thanh ghi khác nữa được dùng trong chế độ đánh địa chỉ theo chỉ số là bộ đếm chương trình (AppendixA).

Trong nhiều ví dụ trên đây thì lệnh MOV đã được sử dụng để đảm bảo đúng đắn, mặc dù ta có thể sử dụng bất kỳ lệnh nào khác chừng nào nó hỗ trợ cho chế độ đánh địa chỉ. Ví dụ lệnh “ADD A, @R0” sẽ cộng nội dung ngăn nhớ cho RO chỉ đến vào nội dung của thanh ghi A.