



ریزپردازنده

دانشکده کامپیوتر دانشگاه یزد

نیم سال اول تحصیلی ۹۶-۹۷

ارائه دهنده : پریسا استواری



روش‌های آدرس دهی

دستورات 8051

- ۱۳۹ دستور یک بایتی
- ۹۲ دستور ۲ بایتی
- ۲۴ دستور ۳ بایتی

روش‌های آدرس‌دهی

- دستورات بر روی داده‌ها انجام می‌شوند. اما داده‌ها کجا قرار دارند؟
- آدرس‌دهی ثابت (Register Addressing)
- آدرس‌دهی مستقیم (Direct Addressing)
- آدرس‌دهی غیرمستقیم (Indirect Addressing)
- آدرس‌دهی بلافاصله (Immediate Addressing)
- آدرس‌دهی نسبی (Relative Addressing)
- آدرس‌دهی مطلق (Absolute Addressing)
- آدرس‌دهی بلند (Long Addressing)
- آدرس‌دهی اندیس‌دار (Index Addressing)

روش‌های آدرس‌دهی

• دستورات بر روی داده‌ها انجام می‌شوند. اما داده‌ها کجا قرار دارند؟

- آدرس‌دهی ثابت (Register Addressing)
- آدرس‌دهی مستقیم (Direct Addressing)
- آدرس‌دهی غیرمستقیم (Indirect Addressing)
- آدرس‌دهی بلافاصله (Immediate Addressing)
- آدرس‌دهی نسبی (Relative Addressing)
- آدرس‌دهی مطلق (Absolute Addressing)
- آدرس‌دهی بلند (Long Addressing)
- آدرس‌دهی اندیس‌دار (Index Addressing)

آدرس دهی ثبات (Register Addressing)

- میکروکنترلر به ۸ ثبات R0 تا R7 دسترسی دارد.

```
MOV A,R0      ;copy contents of R0 into A
MOV R2,A      ;copy contents of A into R2
ADD A,R5      ;add contents of R5 to A
ADD A,R7      ;add contents of R7 to A
MOV R6,A      ;save accumulator in R6
```

- ساینز ثبات‌های مبدا و مقصد باید مشابه باشد.
- MOV DPTR, A خطا می‌دهد.

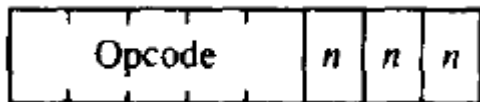
```
MOV DPTR,#25F5H
MOV R7,DPL
MOV R6,DPH
```

- انتقال بین ثبات‌های R0 تا R7 امکان پذیر نیست.
- MOV R7, R4 خطا می‌دهد.

آدرس دهی ثابت (Register Addressing)

- در دستورات آدرس دهی ثابت، سه بیت کم ارزش تر کد دستور آدرس ثابت را ذخیره می کند.
- کد اجرا و آدرس عملوند در یک بایت قرار دارند.

• مثال :



• ADD A, R7

• کد دستور برابر با $00101\underbrace{111}_B$ است.

- ۵ بیت پر ارزش تر نشان دهنده دستور جمع است.
- ۳ بیت کم ارزش تر شماره ۷ را مشخص کرده است.

• ADD A, R3

• کد دستور برابر با $00101\underbrace{011}_B$ است.

- برخی دستورات به خصوص دستورات ثابت های A, B, DPTR, PC و C آدرس آنها در دستور وجود ندارد. کد اجرایی نشان دهنده کل عملیات است.

دستور	کد اجرایی
DIV AB	10000100B
MUL AB	10100100B
INC DPTR	?

روش های آدرس دهی

- دستورات بر روی داده ها انجام می شوند. اما داده ها کجا قرار دارند؟
- آدرس دهی ثابت (Register Addressing)
- آدرس دهی مستقیم (Direct Addressing)
- آدرس دهی غیرمستقیم (Indirect Addressing)
- آدرس دهی بلافاصله (Immediate Addressing)
- آدرس دهی نسبی (Relative Addressing)
- آدرس دهی مطلق (Absolute Addressing)
- آدرس دهی بلند (Long Addressing)
- آدرس دهی اندیس دار (Index Addressing)

آدرس دهی مستقیم (Direct Addressing)

- می توان به هر یک از ثبات ها یا خانه ی حافظه دسترسی داشت.
- آدرس خانه ی حافظه به صورت یک بایت اضافه در دنباله ی بایت کد اجرا قرار می گیرد.



MOV A, 10H •

- مقداری که در خانه 10H حافظه RAM قرار دارد را در A می ریزد.

MOV A, P1 = MOV A, 90H •

- آدرس مستقیم هم می تواند آدرس حافظه ی RAM باشد (00-07 H) و یا آدرس حافظه ی کاربرد خاص
- کد اجرایی این دستور برابر با E5 90 H است.

MOV R0, 40H •

MOV 56H, A • کد اجرایی آن F5 56 است.

MOV A, 4 • دو بایتی →

MOV A, R4 • یک بایتی →

- این دو دستور مشابه اند.
- دستور اول آدرس دهی مستقیم. دستور دوم آدرس دهی ثباتی.
- دستور اولی دو بایتی است و دستور دوم یک بایتی.

آدرس دهی مستقیم (Direct Addressing)

• برای PUSH و POP در استک تنها می توان از آدرس دهی مستقیم استفاده نمود.

• × PUSH A

• ✓ PUSH 0E0H

```
PUSH 05      ;push R5 onto stack
PUSH 0E0H    ;push register A onto stack
POP  0F0H    ;pop top of stack into B
           ;now register B = register A
POP  02      ;pop top of stack into R2
           ;now R2=R6
```

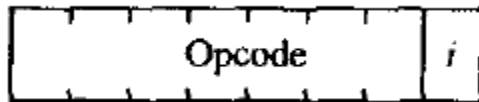
روش های آدرس دهی

• دستورات بر روی داده ها انجام می شوند. اما داده ها کجا قرار دارند؟

- آدرس دهی ثابت (Register Addressing)
- آدرس دهی مستقیم (Direct Addressing)
- آدرس دهی غیرمستقیم (Indirect Addressing)
- آدرس دهی بلافاصله (Immediate Addressing)
- آدرس دهی نسبی (Relative Addressing)
- آدرس دهی مطلق (Absolute Addressing)
- آدرس دهی بلند (Long Addressing)
- آدرس دهی اندیس دار (Index Addressing)

آدرس دهی غیرمستقیم (Indirect Addressing)

- می توان به خانه های پشت سر هم از حافظه اشاره کرد.
- می توان در زمان اجرا آدرس را عوض نمود.
- برای آدرس دهی غیرمستقیم از ثبات های R0 و R1 به عنوان اشاره گر استفاده می شود.
- محتوای آنها آدرس حافظه RAM را مشخص می کند.
- کم ارزش ترین بیت کد دستور مشخص کننده ثبات R0 یا R1 است.



- اگر محتوای خانه R1 برابر با 40H باشد و محتوی خانه 40 حافظه برابر با 87H باشد
- `MOV A, @R1`
- مقدار A برابر با 87H می شود.

آدرس دهی غیرمستقیم (Indirect Addressing)

• مثال :

```
MOV A,@R0    ;move contents of RAM whose
               ;address is held by R0 into A
MOV @R1,B    ;move contents of B into RAM
               ;whose address is held by R1
```

• مثال : دستورات زیر محتوای خانه‌های حافظه از آدرس 60H تا 7FH را برابر با 5 می‌کنند.

```
MOV R0, #60H
LOOP:  MOV @R0, #5
        INC R0
        CJNE R0, #80H, LOOP
```

Compare Jump Not Equal

مثال

• برنامه‌ای بنویسید که مقدار 55H را در خانه‌های 40H تا 41H حافظه RAM کپی کند.

• الف) با آدرس‌دهی مستقیم

• ب) با آدرس‌دهی غیرمستقیم بدون حلقه

• ج) با آدرس‌دهی غیرمستقیم با حلقه

Solution:

(a)

```
MOV A, #55H    ;load A with value 55H
MOV 40H, A     ;copy A to RAM location 40H
MOV 41H, A     ;copy A to RAM location 41H
```

(b)

```
MOV A, #55H    ;load A with value 55H
MOV R0, #40H   ;load the pointer. R0=40H
MOV @R0, A     ;copy A to RAM R0 points to
INC R0         ;increment pointer. Now R0=41h
MOV @R0, A     ;copy A to RAM R0 points to
```

(c)

```
MOV A, #55H    ;A=55H
MOV R0, #40H   ;load pointer. R0=40H,
MOV R2, #02    ;load counter, R2=3
AGAIN: MOV @R0, A ;copy 55 to RAM R0 points to
INC R0         ;increment R0 pointer
DJNZ R2, AGAIN ;loop until counter = zero
```

Decrement Jump Not Zero

روش های آدرس دهی

- دستورات بر روی داده ها انجام می شوند. اما داده ها کجا قرار دارند؟
- آدرس دهی ثابت (Register Addressing)
- آدرس دهی مستقیم (Direct Addressing)
- آدرس دهی غیرمستقیم (Indirect Addressing)
- آدرس دهی بلافاصله (Immediate Addressing)
- آدرس دهی نسبی (Relative Addressing)
- آدرس دهی مطلق (Absolute Addressing)
- آدرس دهی بلند (Long Addressing)
- آدرس دهی اندیس دار (Index Addressing)

آدرس دهی بلافاصله (Immediate Addressing)

- عملوند یک عدد ثابت است که بلافاصله بعد از دستور می آید.
- بایت اول کد دستور
- بایت دوم (یا دوم و سوم) عملوند ثابت



- مقدار داده باید با علامت # مشخص شود.

MOV A, #12

- کد دستور 74 0C H است.

- دستور دو بایت است.

- تنها برای انتقال عدد ثابت به DPTR که رجیستر ۱۶ بیتی است، برای عملوند دو بایت در نظر گرفته می شود و دستور سه بایتی خواهد بود.

MOV DPTR, #8899H

- کد دستور برابر 90 88 99 H است.

آدرس دهی بلافاصله (Immediate Addressing)

- می توان با دستور EQU برای آدرس دهی بلافاصله استفاده نمود.

Equal

```

Count EQU 30
...
MOV R4, #COUNT ; R4=1EH
MOV DPTR, #MYDATA ; DPTR=200H

ORG 200H
MYDATA: DB "America"
  
```

- همچنین از آدرس دهی بلافاصله برای مقداردهی به پورت ها نیز می توان استفاده نمود.

MOV P1, #55H

مثال

• دستورات زیر دارای کدامیک از مودهای آدرس دهی اند و چند بایتی می باشند؟

MOV 23H, R4	○	
MOV R4, #0FFH	○	
ADD A, R6	○	○ آدرس دهی ثباتی
MOV @R1, A	○	○ آدرس دهی مستقیم
MOV 44H, #20H	○	○ آدرس دهی غیرمستقیم
MOV A, #11101011B	○	○ آدرس دهی بلافاصله
MOV 19H, @R0	○	
MOV A, #14	○	

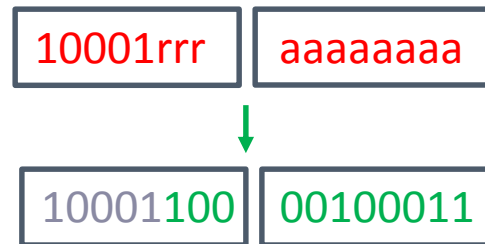
مثال

• دستورات زیر دارای کدامیک از مودهای آدرس دهی اند و چند بایتی می باشند؟

MOV 23H, R4
 MOV R4, #0FFH
 ADD A, R6
 MOV @R1, A
 MOV 44H, #20H
 MOV A, #11101011B
 MOV 19H, @R0
 MOV A, #14



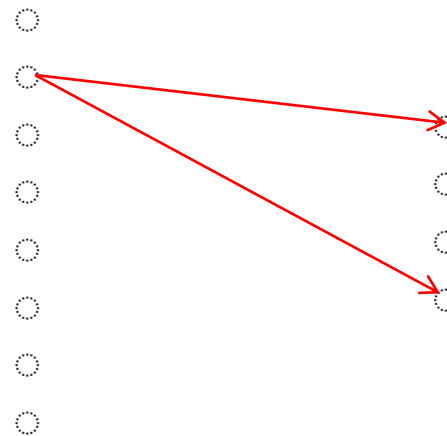
آدرس دهی ثباتی
 آدرس دهی مستقیم
 آدرس دهی غیرمستقیم
 آدرس دهی بلافاصله



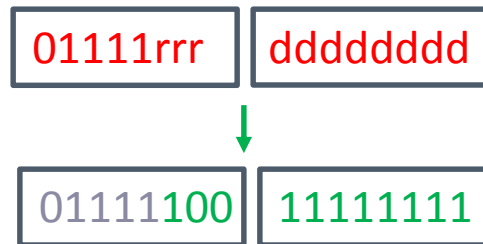
مثال

• دستورات زیر دارای کدامیک از موده‌های آدرس‌دهی اند و چند بایتی می‌باشند؟

```
MOV 23H, R4
MOV R4, #0FFH
ADD A, R6
MOV @R1, A
MOV 44H, #20H
MOV A, #11101011B
MOV 19H, @R0
MOV A, #14
```



آدرس‌دهی ثباتی
آدرس‌دهی مستقیم
آدرس‌دهی غیرمستقیم
آدرس‌دهی بلافاصله



مثال

• دستورات زیر دارای کدامیک از مودهای آدرس دهی اند و چند بایتی می باشند؟

MOV 23H, R4

MOV R4, #0FFH

ADD A, R6

MOV @R1, A

MOV 44H, #20H

MOV A, #11101011B

MOV 19H, @R0

MOV A, #14

○

○

○

○

○

○

○

○

○

○

○

○

آدرس دهی ثباتی

آدرس دهی مستقیم

آدرس دهی غیرمستقیم

آدرس دهی بلافاصله

11111rrr



11111110

مثال

• دستورات زیر دارای کدامیک از مودهای آدرس دهی اند و چند بایتی می باشند؟

MOV 23H, R4	○	
MOV R4, #0FFH	○	
ADD A, R6	○	○ آدرس دهی ثباتی
MOV @R1, A	○	○ آدرس دهی مستقیم
MOV 44H, #20H	○	○ آدرس دهی غیرمستقیم
MOV A, #11101011B	○	○ آدرس دهی بلافاصله
MOV 19H, @R0	○	
MOV A, #14	○	

1111011i



11110111

مثال

• دستورات زیر دارای کدامیک از مودهای آدرس دهی اند و چند بایتی می باشند؟

MOV 23H, R4

MOV R4, #0FFH

ADD A, R6

MOV @R1, A

MOV 44H, #20H

MOV A, #11101011B

MOV 19H, @R0

MOV A, #14

○

○

○

○

○

○

○

○

○

○

○

○

آدرس دهی ثباتی

آدرس دهی مستقیم

آدرس دهی غیرمستقیم

آدرس دهی بلافاصله

01110101

aaaaaaaa

dddddddd



01110101

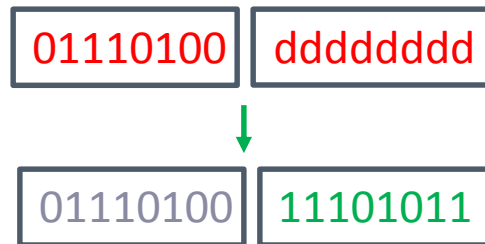
01000100

00100000

مثال

• دستورات زیر دارای کدامیک از مودهای آدرس دهی اند و چند بایتی می باشند؟

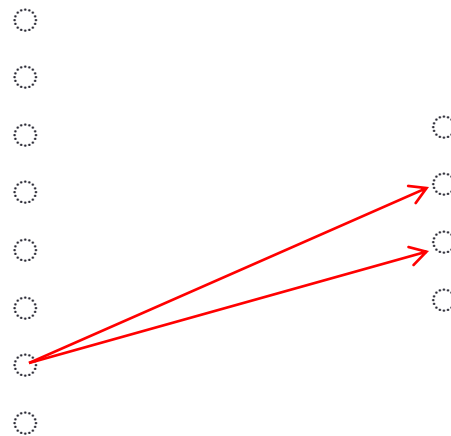
MOV 23H, R4	○	
MOV R4, #0FFH	○	
ADD A, R6	○	○ آدرس دهی ثباتی
MOV @R1, A	○	○ آدرس دهی مستقیم
MOV 44H, #20H	○	○ آدرس دهی غیرمستقیم
MOV A, #11101011B	○	○ آدرس دهی بلافاصله
MOV 19H, @R0	○	
MOV A, #14	○	



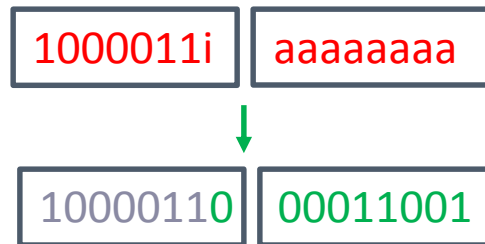
مثال

• دستورات زیر دارای کدامیک از مودهای آدرس دهی اند و چند بایتی می باشند؟

```
MOV 23H, R4
MOV R4, #0FFH
ADD A, R6
MOV @R1, A
MOV 44H, #20H
MOV A, #11101011B
MOV 19H, @R0
MOV A, #14
```



آدرس دهی ثباتی
آدرس دهی مستقیم
آدرس دهی غیرمستقیم
آدرس دهی بلافاصله



مثال

• دستورات زیر دارای کدامیک از مودهای آدرس دهی اند و چند بایتی می باشند؟

MOV 23H, R4

MOV R4, #0FFH

ADD A, R6

MOV @R1, A

MOV 44H, #20H

MOV A, #11101011B

MOV 19H, @R0

MOV A, #14

○

○

○

○

○

○

○

○

○

○

○

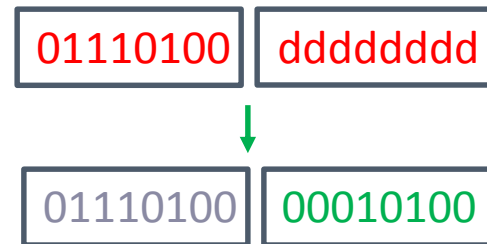
○

آدرس دهی ثباتی

آدرس دهی مستقیم

آدرس دهی غیرمستقیم

آدرس دهی بلافاصله



روش های آدرس دهی

- دستورات بر روی داده ها انجام می شوند. اما داده ها کجا قرار دارند؟
- آدرس دهی ثابت (Register Addressing)
- آدرس دهی مستقیم (Direct Addressing)
- آدرس دهی غیرمستقیم (Indirect Addressing)
- آدرس دهی بلافاصله (Immediate Addressing)
- آدرس دهی نسبی (Relative Addressing)
- آدرس دهی مطلق (Absolute Addressing)
- آدرس دهی بلند (Long Addressing)
- آدرس دهی اندیس دار (Index Addressing)

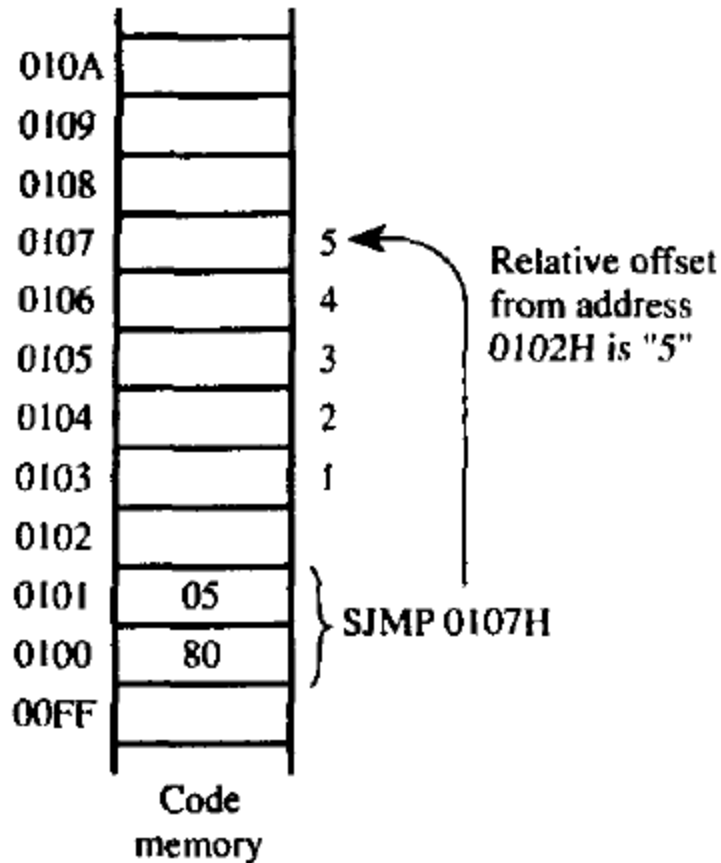
آدرس دهی نسبی (Relative Addressing)

- تنها برای برخی از دستورات پرش در کد برنامه استفاده می شود.
- آدرس نسبی یا **offset** یک عدد ۸ بیتی علامت دار است که به عنوان بایت دوم در انتهای کد دستور قرار می گیرد.
- این **offset** به مقدار PC اضافه می گردد تا آدرس دستور بعدی که باید اجرا شود مشخص شود.
- دقت شود این پرش ها در حافظه ROM صورت می گیرد.

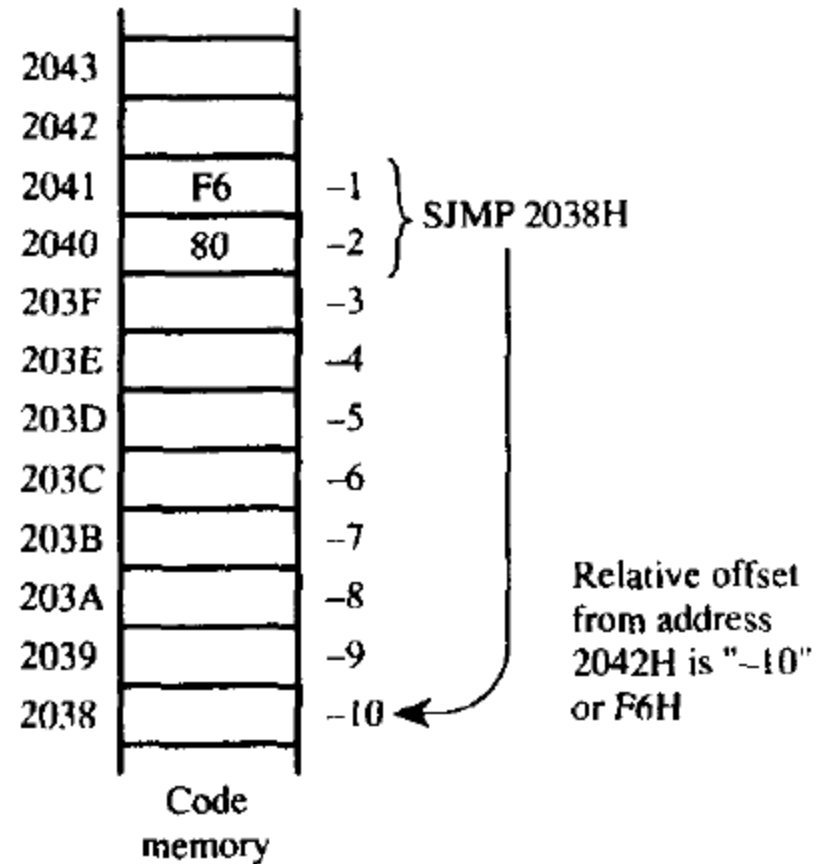


- دستور SJMP (Short Jump) از این آدرس دهی استفاده می نماید.
- بدین دلیل که **offset** یک بایت (عدد علامت دار ۸ بیتی) است تنها می توان به $+127$ خانه حافظه ی بعد یا -128 خانه حافظه قبل پرش کرد.
- باید دقت کرد چون PC همیشه آدرس دستور بعدی را نشان می دهد، آدرس پرش نسبت به دستور بعد محاسبه می شود نه دستور فعلی

آدرس دهی نسبی (Relative Addressing)



(a) Short jump ahead in memory



(b) Short jump back in memory

آدرس دهی نسبی (Relative Addressing)

- جزئیات اطلاع از این پرش ها در برنامه نویسی مهم نیست.
- محل پرش با یک label مشخص می شود.

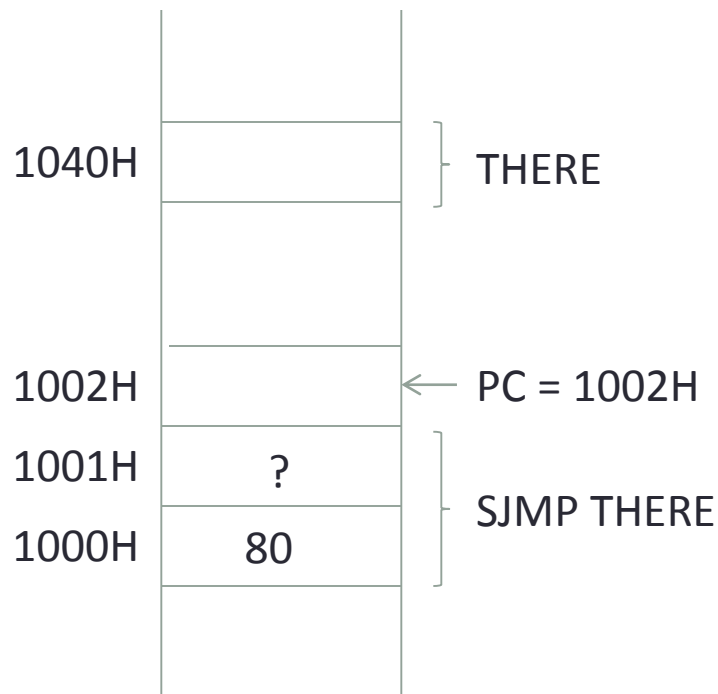
آدرس مبدا - آدرس مقصد = آفست

- مثال : دستور پرش در آدرس 1000H و 1001H قرار دارد.

SJMP THERE

- و برچسب THERE در آدرس 1040H قرار دارد.

- مترجم اسمبلر مقدار offset را حساب می نماید.
- مقدار offset برابر با 3E است.



آدرس دهی نسبی (Relative Addressing)

- جزئیات اطلاع از این پرش ها در برنامه نویسی مهم نیست.
- محل پرش با یک label مشخص می شود.

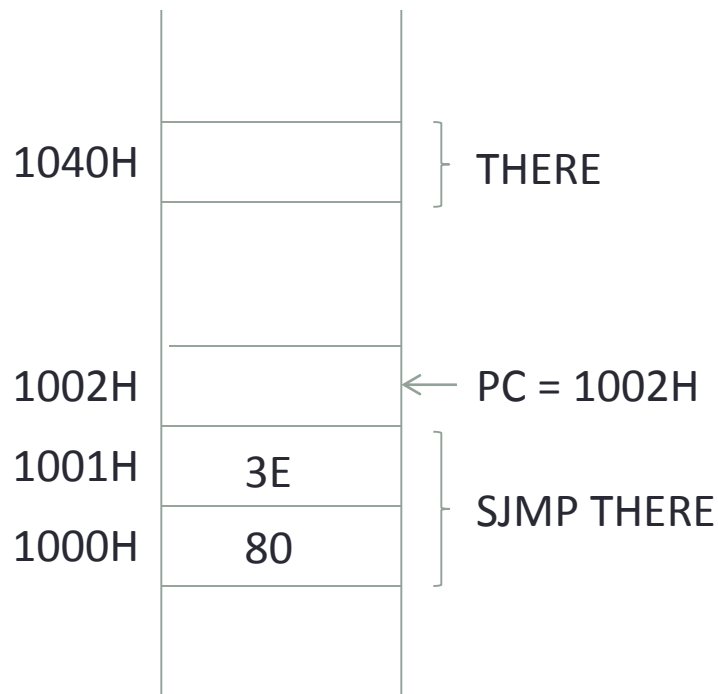
آدرس منبع - آدرس مقصد = آفست

- مثال : دستور پرش در آدرس 1000H و 1001H قرار دارد.

SJMP THERE

- و برچسب THERE در آدرس 1040H قرار دارد.

- مترجم اسمبلر مقدار offset را حساب می نماید.
- مقدار offset برابر با 3E است.



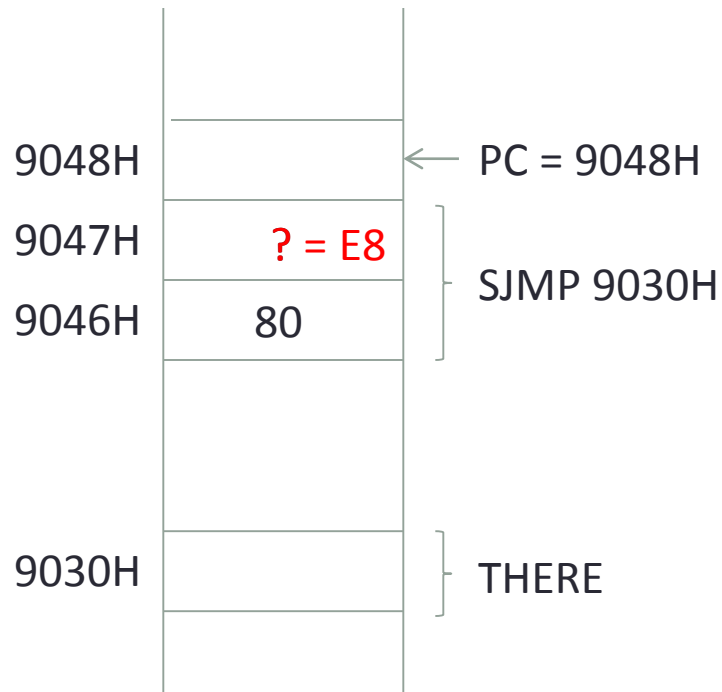
$$\begin{array}{r}
 1040 \\
 -1002 \\
 \hline
 003E
 \end{array}$$

آدرس دهی نسبی (Relative Addressing)

- مثال : دستور SJMP THERE در آدرس 9046H و 9047H قرار دارد. و THERE در آدرس 9030H قرار دارد.

- کد دستور SJMP چه خواهد بود؟

آدرس مبدا - آدرس مقصد = آفست



9048
-9030

0018 (neg) -> E8

9030
-9048

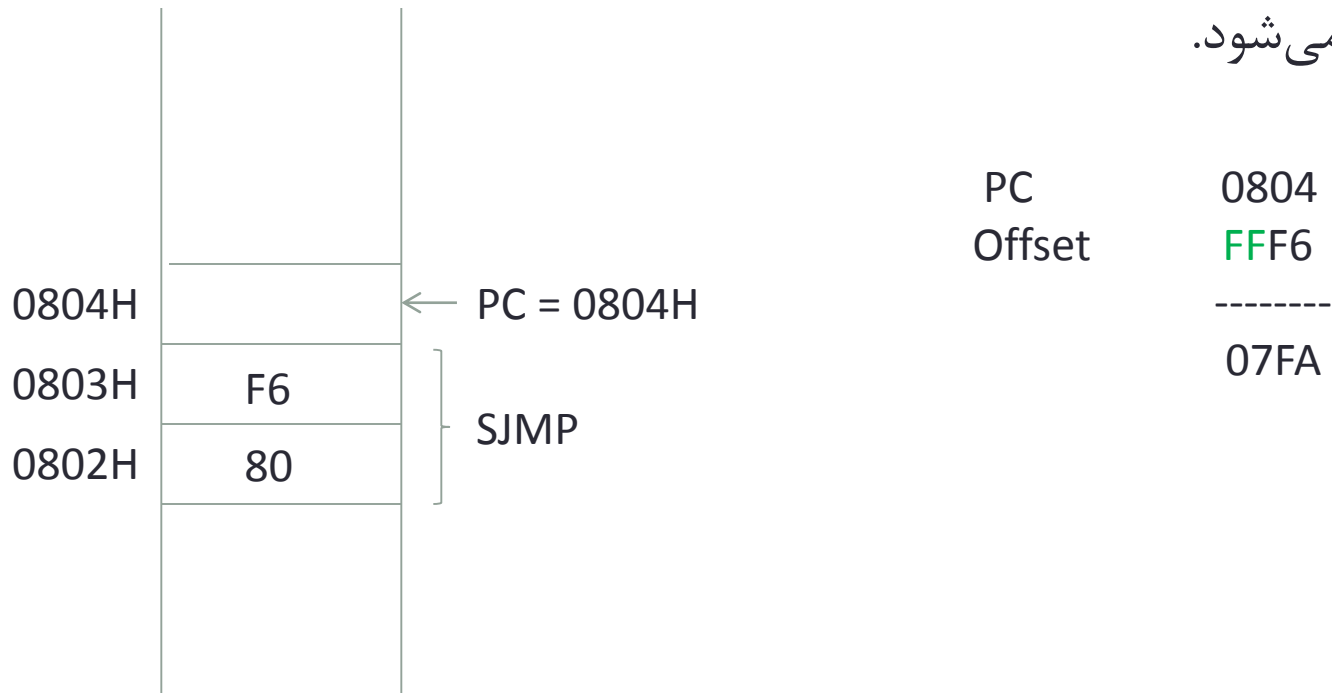
FFFE8 -> E8

آدرس دهی نسبی (Relative Addressing)

- مثال : دستور پرش SJMP دارای کد دستور 80 F6 H است. و در خانه های 0802H و 0803H قرار دارد.

آفست + آدرس مبدا = آدرس مقصد

- به چه آدرسی پرش می شود.

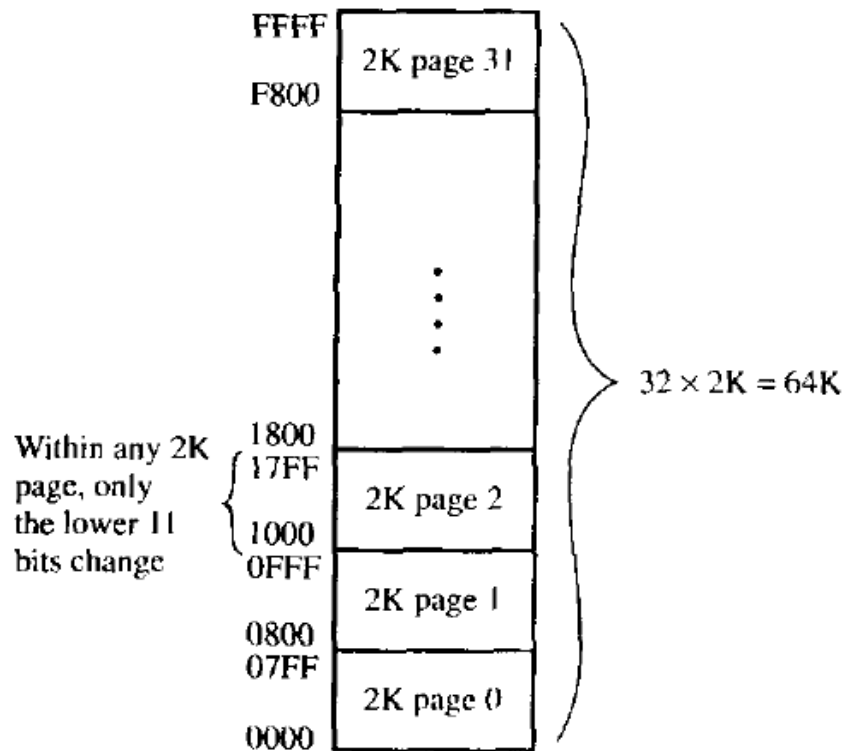


روش های آدرس دهی

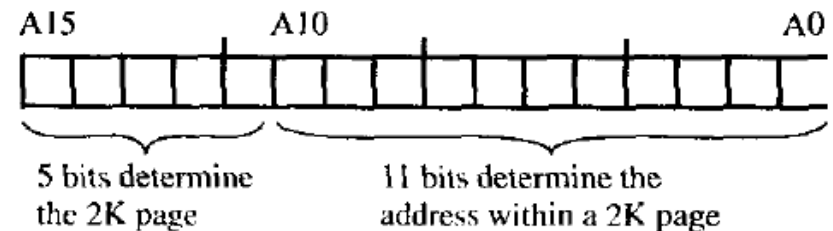
- دستورات بر روی داده ها انجام می شوند. اما داده ها کجا قرار دارند؟
- آدرس دهی ثابت (Register Addressing)
- آدرس دهی مستقیم (Direct Addressing)
- آدرس دهی غیرمستقیم (Indirect Addressing)
- آدرس دهی بلافاصله (Immediate Addressing)
- آدرس دهی نسبی (Relative Addressing)
- آدرس دهی مطلق (Absolute Addressing)
- آدرس دهی بلند (Long Addressing)
- آدرس دهی اندیس دار (Index Addressing)

آدرس دهی مطلق (Absolute Addressing)

- فقط برای دستورات ACALL (Absolute Call) و AJMP (Absolute Jump) استفاده می شود.
- با این دستورات می توان تا مقدار 2K بایت در داخل یک صفحه حافظه کد (ROM) پرش نمود.



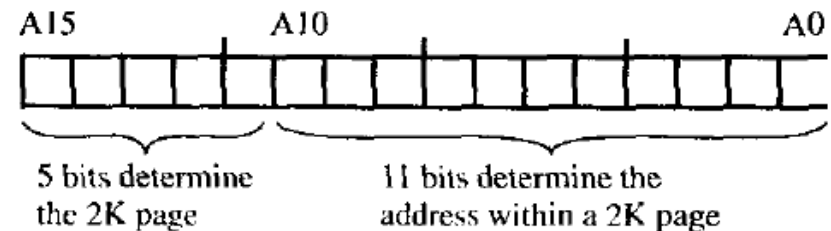
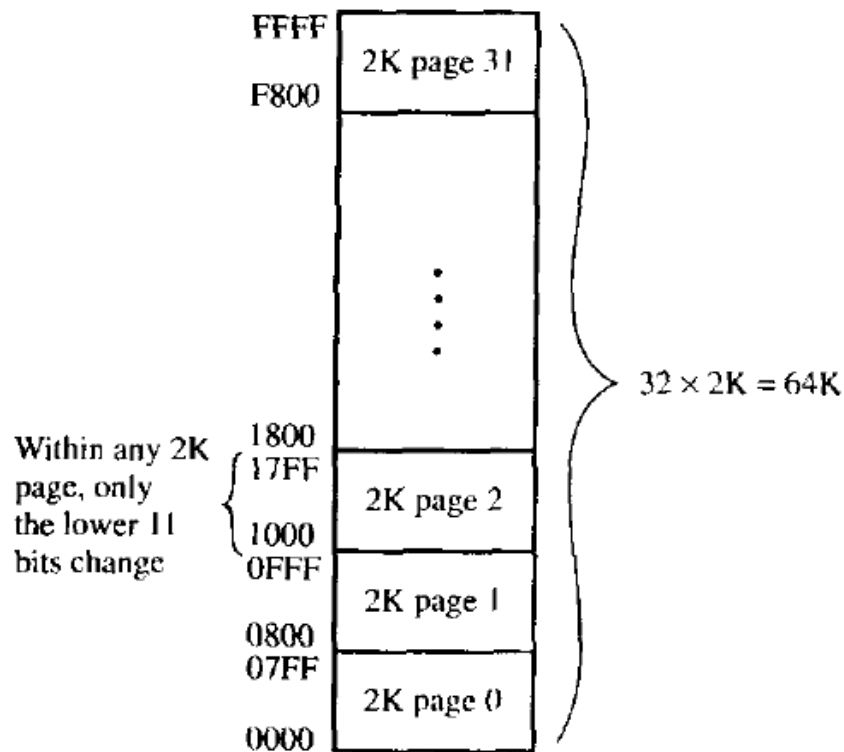
(a) 64K memory map divided into 32 2K pages



- (b) The upper 5 bits in the program counter remain the same. The lower bits are replaced by the bits supplied in the instruction.

آدرس دهی مطلق (Absolute Addressing)

- در داخل صفحه با طول 2K فقط ۱۱ بیت کم ارزش تر آدرس تغییر می کند.
- ۵ بیت پر ارزش تر PC نشان دهنده ی شماره صفحه و ۱۱ بیت کم ارزش تر آن نشان - دهنده آدرس داخل آن صفحه است.



- (b) The upper 5 bits in the program counter remain the same. The lower bits are replaced by the bits supplied in the instruction.

(a) 64K memory map divided into 32 2K pages

آدرس دهی مطلق (Absolute Addressing)

- برای محاسبه‌ی آدرس مقصد پرش، پنج بیت پر ارزش‌تر PC در پنج بیت پر ارزش‌تر آدرس مقصد پرش کپی می‌شود.
- ۱۱ بیت کم ارزش‌تر آدرس مقصد پرش از دستور می‌آید.
- ۸ بیت کم ارزش‌تر آدرس پرش (A0-A7) در بایت دوم دستور و ۳ بیت پر ارزش‌تر آدرس پرش (A8-A10) در بایت اول دستور قرار دارد.



آدرس دهی مطلق (Absolute Addressing)

- مثال : اگر دستور AJMP THERE در آدرس 0900H و 0901H قرار داشته باشد و برچسب THERE مکان 0F46H حافظه را نشان دهد.

- آیا هر دستور و مقصد پرش در یک صفحه قرار دارند؟

PC 0902H = 0000 1001 0000 0010
 0F46H = 0000 1111 0100 0110

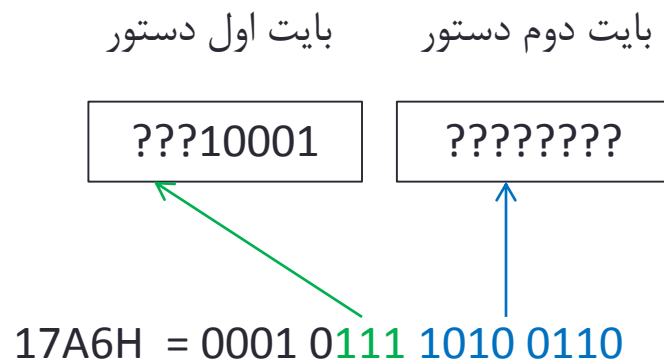
- باید به ۵ بیت پر ارزش تر آدرس ها نگاه کنیم.
- در یک صفحه اند.

کد دستور AJMP THERE : 11110001 01000110

- برای محاسبه ی آدرس مقصد، ۵ بیت پر ارزش تر PC تغییر نمی کند.
- چون در دستور AJMP دستور مقصد و مبدا در یک صفحه 2K بایتی قرار دارند.
- سپس ۱۱ بیت رنگی در کد دستور، در ۱۱ بیت کم ارزش تر PC کپی می شود.

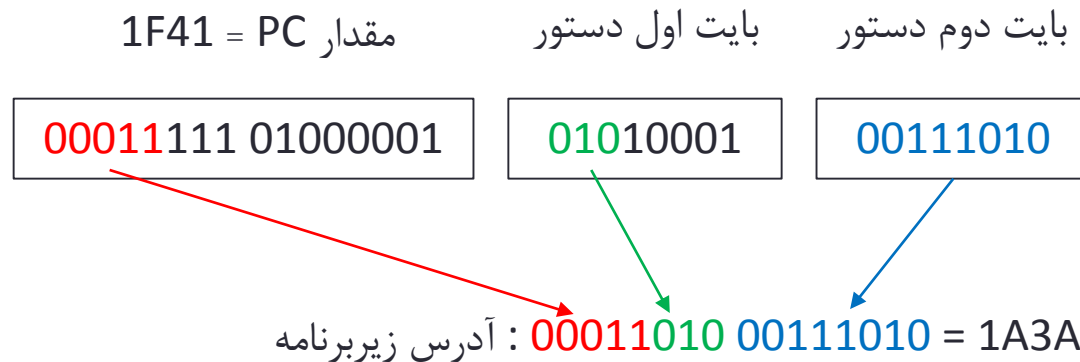
آدرس دهی مطلق (Absolute Addressing)

- مثال : اگر دستور فراخوانی زیربرنامه ACALL در آدرس 1024H و 1025H قرار داشته باشد و آدرس زیربرنامه از آدرس 17A6H شروع شود، ترجمه دستور ACALL چه خواهد بود؟



آدرس دهی مطلق (Absolute Addressing)

- مثال : اگر دستور فراخوانی زیربرنامه ACALL در آدرس 1F3FH و 1F40H قرار داشته باشد و ترجمه دستور ACALL برابر با 513A باشد، آدرس زیربرنامه چه خواهد بود؟

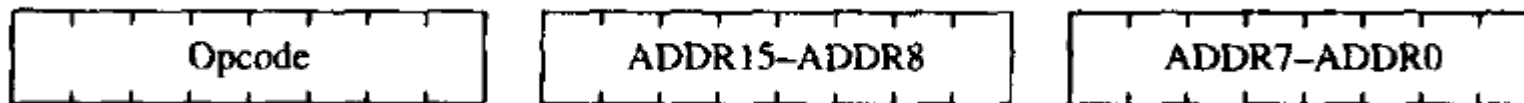


روش های آدرس دهی

- دستورات بر روی داده ها انجام می شوند. اما داده ها کجا قرار دارند؟
- آدرس دهی ثابت (Register Addressing)
- آدرس دهی مستقیم (Direct Addressing)
- آدرس دهی غیرمستقیم (Indirect Addressing)
- آدرس دهی بلافاصله (Immediate Addressing)
- آدرس دهی نسبی (Relative Addressing)
- آدرس دهی مطلق (Absolute Addressing)
- آدرس دهی بلند (Long Addressing)
- آدرس دهی اندیس دار (Index Addressing)

آدرس دهی بلند (Long Addressing)

- تنها با دستورات LCALL (Long Call) و LJMP (Long Jump) به کار برده می-شود.
- این دستورات سه بایتی هستند.
- در بایت دوم و سوم آنها آدرس ۱۶ بیتی مقصد ذخیره می شود.



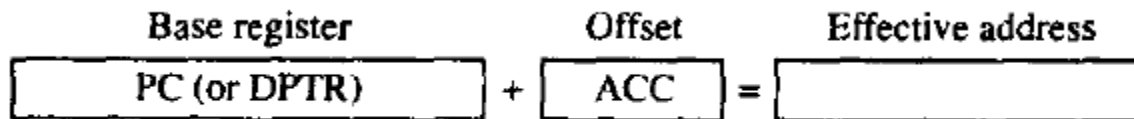
- با این دستورات می توان به تمام نقاط حافظه پرش کرد.
- اشکال آنها این است که سه بایت طول دارند.
- مثال : دستور LJMP THERE که در آن THERE در آدرس 2040H قرار دارد به صورت 02 20 40 H ترجمه می شود.

روش های آدرس دهی

- دستورات بر روی داده ها انجام می شوند. اما داده ها کجا قرار دارند؟
- آدرس دهی ثابت (Register Addressing)
- آدرس دهی مستقیم (Direct Addressing)
- آدرس دهی غیرمستقیم (Indirect Addressing)
- آدرس دهی بلافاصله (Immediate Addressing)
- آدرس دهی نسبی (Relative Addressing)
- آدرس دهی مطلق (Absolute Addressing)
- آدرس دهی بلند (Long Addressing)
- آدرس دهی اندیس دار (Index Addressing)

آدرس دهی اندیس دار (Index Addressing)

- در دستورات پرش (JMP) و انتقال (MOVC) استفاده می شود.
- محتوای ثبات پایه PC یا DPTR را با محتوای A جمع می کند تا آدرس مقصد تولید شود.



(h) Indexed addressing (e.g., MOVC A, @A + PC)

Move Code

• **MOVC A, @A+Base Register**

- در اینجا مقداری که در یکی از خانه ها حافظه ROM قرار دارد به A منتقل می شود.
- آدرس این خانه از حافظه برابر است با مقدار فعلی A به اضافه ی مقدار فعلی ثبات پایه که PC یا DPTR است.

- از این دستور برای جستجو در جداول استفاده می شود.

• **JMP @A+DPTR**

- در اینجا آدرس خانه حافظه ROM که قرار است به آن پرش شود برابر است با مقدار فعلی A به اضافه ی مقدار فعلی ثبات DPTR است.

```

ORG 0
MOV DPTR, #MYDATA
CLR A
MOVC A, @A+DPTR
MOV R7, A
CLR A
INC DPTR
MOVC A, @A+DPTR
MOV R7, A
CLR A
INC DPTR
MOVC A, @A+DPTR
MOV R7, A
CLR A
INC DPTR
MOVC A, @A+DPTR
MOV R7, A

```

```
ORG 60H
```

```
MYDATA : DB "IRAN"
```

آدرس دهی اندیس دار

- مثال : فرض کنید کلمه IRAN در آدرس 60H حافظه ROM باشد.

مثال

- برنامه‌ای بنویسید که ۱۶ خانه RAM را از خانه ی 60H پاک کند.

Solution:

```
CLR A           ; A=0
MOV R1, #?
MOV R7, #?
AGAIN: MOV @R1,A ; clear RAM R1 points to
      INC R1     ; increment R1 pointer
      DJNZ R7, AGAIN ; loop until counter=zero
```

مثال

- برنامه‌ای بنویسید که ۱۶ خانه RAM را از خانه ی 60H پاک کند.

Solution:

```
CLR A           ;A=0
MOV R1,#60H     ;load pointer. R1=60H
MOV R7,#16      ;load counter, R7=16
AGAIN: MOV @R1,A ;clear RAM R1 points to
INC R1          ;increment R1 pointer
DJNZ R7,AGAIN  ;loop until counter=zero
```

مثال

- برنامه‌ای بنویسید که ۱۰ بایت داده را از آدرس 35H به آدرس 60H کپی کند.

Solution:

```
MOV R0,?  
MOV R1,?  
MOV R3,?  
BACK: MOV A,@R0      ;get a byte from source  
      MOV @R1,A      ;copy it to destination  
      ?  
      ?  
      DJNZ R3,BACK ;keep doing for ten bytes
```


مثال

- برنامه‌ای بنویسید که ۱۰ بایت داده را از آدرس 35H به آدرس 60H کپی کند.

Solution:

```
MOV R0,#35H    ;source pointer
MOV R1,#60H    ;destination pointer
MOV R3,#10     ;counter
BACK: MOV A,@R0 ;get a byte from source
      MOV @R1,A ;copy it to destination
      ?
      ?
      DJNZ R3,BACK ;keep doing for ten bytes
```

مثال

- برنامه‌ای بنویسید که ۱۰ بایت داده را از آدرس 35H به آدرس 60H کپی کند.

Solution:

```
MOV R0,#35H    ;source pointer
MOV R1,#60H    ;destination pointer
MOV R3,#10     ;counter
BACK: MOV A,@R0 ;get a byte from source
      MOV @R1,A ;copy it to destination
      INC R0    ;increment source pointer
      INC R1    ;increment destination pointer
      DJNZ R3,BACK ;keep doing for ten bytes
```

مثال

- برنامه‌ای بنویسید که مقدار x را از پورت P1 بگیرد و مقدار x^2 را در پورت P2 بنویسد.

Solution:

```

    ORG    0
    MOV    DPTR, #300H    ;LOAD TABLE ADDRESS
    MOV    A, #0FFH      ;A=FF
    MOV    P1, A          ;CONFIGURE P1 INPUT PORT
BACK: MOV    A, P1        ;GET X
    MOVC   A, @A+DPTR     ;GET X SQAURE FROM TABLE
    MOV    P2, A          ;ISSUE IT TO P2
    SJMP   BACK          ;KEEP DOING IT

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

```