



ریزپردازنده

دانشکده کامپیوتر دانشگاه یزد

نیم سال دوم تحصیلی ۹۶-۹۷

ارائه دهنده : پریسا استواری



مجموعه دستورات 8051

انواع دستورات

- دستورات محاسباتی (Arithmetic)
- دستورات منطقی (Logic)
- دستورات انتقال اطلاعات (Data Transfer)
- دستورات بر روی بیت یا متغیرهای بولین (Boolean Variables)
- دستورات کنترل یا انشعاب برنامه (Program Branching)

انواع دستورات

- دستورات محاسباتی (Arithmetic)
- دستورات منطقی (Logic)
- دستورات انتقال اطلاعات (Data Transfer)
- دستورات بر روی بیت یا متغیرهای بولین (Boolean Variables)
- دستورات کنترل یا انشعاب برنامه (Program Branching)

دستور ADD

$\text{ADD A, source} \rightarrow A = A + \text{source}$

- عملوند مقصد همیشه Accumulator (A) است. حاصل جمع همیشه در A ریخته می شود.
- عملوند منبع می تواند تنها به چهار حالت زیر باشد.

ADD A, 7FH	آدرس دهی مستقیم
ADD A, @R0	آدرس دهی غیرمستقیم
ADD A, R7	آدرس دهی ثبات
ADD A, #35H	آدرس دهی بلافاصله

- در زبان اسمبلی 8051، عملیات محاسباتی حافظه به حافظه امکان پذیر نیست.

ADD 44H, 54H ×	MOV A, 44H ✓
	ADD A, 54H
	MOV 44H, A

دستور ADD

- اگر از بیت ۷ ام بیت نقلی خارج شود $C=1$ می شود. (شمارش بیت ها از صفر شروع می شود).
- اگر از بیت ۳ ام بیت نقلی خارج شود $AC=1$ می شود.
- زمانی که دو عدد بدون علامت با یکدیگر جمع می شوند، بیت پرچم نقلی (C) نشاندهندهی سرریز است.
- زمانی که دو عدد علامتدار با هم جمع می شوند، بیت پرچم سرریز نشان دهندهی این است که یا حاصل جمع دو عدد مثبت، یک عدد منفی شده است یا اینکه حاصل جمع دو عدد منفی، یک عدد مثبت شده است.
- بیت پرچم سرریز (OV) موقعی یک می شود که :
 - بیت نقلی از بیت ۶ اکومولیتور خارج شود و از بیت ۷ ام خارج نشود.
 - یا زمانی که بیت نقلی از بیت ۷ اکومولیتور خارج شود و از بیت ۶ ام خارج نشود.

مثال

- نشان دهید ثبات‌های پرچم بعد از اجرای دستور فوق چگونه تغییر می‌کنند.

MOV A, # 0C3H

ADD A, #0AAH

1100 0011 (C3)

+ 1010 1010 (AA)

1 0110 1101 (6D)

C = 1

AC = 0

OV = 1

P = 1

دستور ADD

دستور	نوع آدرس دهی	ترجمه دستور	تعداد بایت	تعداد سیکل ماشین
ADD A, direct	آدرس دهی مستقیم	00100101 aaaaaaaaa	2	1
ADD A, @Ri	آدرس دهی غیرمستقیم	0010011i	1	1
ADD A, Rn	آدرس دهی ثبات	00101rrr	1	1
ADD A, #data	آدرس دهی بلافاصله	00100100 dddddddd	2	1

مثال

- فرض کنید خانه‌های 40H تا 44H دارای مقادیر زیر باشد. برنامه‌ای بنویسید که مجموع این اعداد را محاسبه نماید. در انتها باید A شامل بایت کوچکتر حاصل جمع و R7 شامل بایت بزرگتر باشد.

40 = (7D)

41 = (EB)

42 = (C5)

43 = (5B)

44 = (30)

Solution:

```

MOV R0, #40H    ;load pointer
MOV R2, #5      ;load counter
CLR A           ;A=0
MOV R7, A       ;clear R7
AGAIN: ADD A, @R0 ;add the byte ptr to by R0
Jump Not Carry → JNC NEXT ;if CY=0 don't add carry
INC R7          ;keep track of carry
NEXT: INC R0     ;increment pointer
DJNZ R2, AGAIN ;repeat until R2 is zero

```

دستور ADDC

ADDC A, source

Add with Carry •

• محتوای A را با مقدار source و محتوای C جمع می کند و نتیجه را در A می ریزد.

$A = A + \text{source} + C$ •

• بر روی پرچم های C، AC، OV و P تاثیر می گذارد.

• همانند دستور ADD

• مثال : فرض شود مقدار بیت پرچم نقلی برابر با یک است.

MOV A, #0C3H

ADDC A, #0AAH

```

1100 0011 (C3)
+ 1010 1010 (AA)
+           1 (C)
-----

```

1 0110 1110 (6E)

C = 1

AC = 0

OV = 1

P = 1

دستور ADDC

- مثال : برنامه‌ای بنویسید که دو عدد ۱۶ بیتی زیر را جمع کرده و نتیجه را در R7 و R6 قرار دهد. R6 بایت کم ارزشتر حاصل جمع است.

$$\begin{array}{r}
 1 \\
 3C \ E7 \\
 + \quad 3B \ 8D \\
 \hline
 78 \ 74
 \end{array}$$

```

CLR    C           ;make CY=0
MOV    A, #0E7H    ;load the low byte now A=E7H
ADD    A, #8DH     ;add the low byte
MOV    R6, A       ;save the low byte sum in R6
MOV    A, #3CH     ;load the high byte
→ ADDC A, #3BH     ;add with the carry
MOV    R7, A       ;save the high byte sum

```

دستور ADDC

تعداد سیکل ماشین	تعداد بایت	ترجمه دستور	نوع آدرس دهی	دستور
1	2	00110101 aaaaaaaaa	آدرس دهی مستقیم	ADDC A, direct
1	1	0011011i	آدرس دهی غیرمستقیم	ADDC A, @Ri
1	1	00110rrr	آدرس دهی ثبات	ADDC A, Rn
1	2	00110100 dddddddd	آدرس دهی بلافاصله	ADDC A, #data

ADDC A, 7FH	آدرس دهی مستقیم
ADDC A, @R0	آدرس دهی غیرمستقیم
ADDC A, R7	آدرس دهی ثبات
ADDC A, #35H	آدرس دهی بلافاصله

سیستم اعداد BCD

• BCD (Binary Coded Decimal)

- نمایش باینری اعداد ۰ تا ۹

• Unpacked BCD :

- ۴ بیت کم ارزش تر عدد BCD را نشان می دهد و ۴ بیت پر ارزش تر صفر است.
- مثال :

0000 0101 5

0000 1001 9

• Packed BCD :

- در یک بایت دو عدد BCD نمایش داده می شود. یکی در ۴ بیت پر ارزش تر و یکی در ۴ بیت کم ارزش تر.
- مثال :

1001 0101 95H

سیستم اعداد BCD

- جمع دو عدد BCD باید عدد BCD باشد.
- مثال :

MOV A, #17H

ADD A, #28H

```

    0001 0111 (17H)
+   0010 1000 (28H)
-----
    0011 1111 (3FH) ✗
+   0000 0110 (06)
-----
    0100 0101 (45H) ✓
  
```

- جواب باید $17+28=45$ باشد اما 3FH است.
- باید عدد 6 به بایت کم ارزش تر اضافه شود. (تصحیح شود)
- $3F + 06 = 45H$

دستور DAA

• Decimal Adjust for Addition

• دستور DA برای تصحیح جمع BCD است.

• دستور DA در صورت لزوم عدد ۶ را با قسمت پایینی یا بالایی بایت یا هر دو جمع می‌کند.

```

      1001 0101 (95H)
+   1001 0110 (96H)
-----
1 0010 1011 (2BH) ×
+ 0000 0110 (06)
-----
      0011 0001 (31H) ×
+ 0110 0000 (06)
-----
      1001 0001 (91H) ✓

```

• مثال : $95+96=191$

دستور DAA

- دستور DA تنها بر روی A کار می کند.
- مثال :

MOV A, #47H

MOV B, #25H

ADD A, B

← A = 6CH

DA A

← A = 72H

- این دستور تنها بعد از دستور ADD و ADDC کار می کند. بعد از دستور INC کار نمی کند.

خلاصه دستور DAA

- بعد از دستور ADD و ADDC
- اگر ۴ بیت کم ارزش تر بزرگ تر از ۹ بشود، یا اگر $AC=1$ بشود :
- عدد 0110 را با ۴ بیت کم ارزش تر جمع می کند.
- اگر ۴ بیت پر ارزش تر بزرگ تر از ۹ بشود، یا اگر $C=1$ بشود :
- عدد 0110 را با ۴ بیت پر ارزش تر جمع می کند.
- بسته به مقدار A و PSW، یکی از مقادیر 00H، 06H، 60H، 66H را به A اضافه می کند.
- اگر $C=1$ شود به این معناست که عدد بزرگتر از 99 شده است.
- این دستور بر OV تاثیری ندارد.
- طول این دستور یک بایت است و در یک سیکل ماشین انجام می شود.
- این دستور بعد از عملیات تفریق عمل نمی کند.

مثال

Example:

	HEX		BCD	
	29		0010 1001	
+	<u>18</u>	+	<u>0001 1000</u>	
	41		0100 0001	AC=1
+	<u>6</u>	+	<u> 0110</u>	
	47		0100 0111	

از آنجایی که AC=1 است، دستور DA عدد ۶ را به قسمت کم ارزش تر عدد اضافه می کند.

مثال

40=(71)

41=(11)

42=(65)

43=(59)

44=(37)

- فرض کنید ۵ عدد BCD در خانه‌های حافظه از خانه 40H قرار دارند. برنامه‌ای بنویسید که حاصل جمع آنها را حساب کند. نتیجه باید به صورت BCD باشد.

Solution:

```

MOV    R0, #40H    ;Load pointer
MOV    R2, #5      ;Load counter
CLR    A           ;A=0
MOV    R7, A       ;Clear R7
AGAIN: ADD    A, @R0 ;add the byte pointer
                        ;to by R0
        DA     A     ;adjust for BCD
        JNC    NEXT  ;if CY=0 don't
                        ;accumulate carry
        INC    R7     ;keep track of carries
NEXT:   INC    R0     ;increment pointer
        DJNZ   R2, AGAIN ;repeat until R2 is 0

```

مثال

- برنامه‌ای بنویسید که دو عدد ۲ بایتی BCD را با هم جمع نماید. اولین عدد در خانه‌های 40H و 41H قرار دارد و عدد دوم در خانه‌های 42H و 43H قرار دارد. پرارزش‌ترین بایت هر دو عدد در خانه‌های 40H و 42H قرار دارد. حاصل جمع را در خانه‌های 44H و 45H بریزید.

```
MOV A, 43H
```

```
ADD A, 41H
```

```
DA A
```

```
MOV 45H, A
```

```
MOV A, 42H
```

```
ADDC A, 40H
```

```
DA A
```

```
MOV 44H, A
```

دستور SUBB

SUBB A, source

$$A = A - \text{source} - C$$

Subtract with Borrow

- در بسیاری از میکروکنترلرها دو دستور SUB و SUBB وجود دارد.
- در 8051 تنها دستور SUBB وجود دارد.
- برای داشتن دستور SUB باید ابتدا مقدار C را صفر کرد (CLR C) و سپس دستور SUBB را اجرا کرد.
- اگر بیت قرضی برای بیت ۷ اکومولیتور نیاز باشد، این دستور بیت پرچم نقلی را ۱ می‌کند. در غیر اینصورت این بیت ۰ است.
- اگر بیت قرضی برای بیت ۳م لازم باشد، پرچم AC برابر با ۱ می‌شود.

دستور SUBB هنگامی که $C=1$

- این دستور برای تفریق اعداد چند بایتی به کار می‌رود.

```

CLR      C
MOV      A, #62H    ; A=62H
SUBB     A, #96H    ; 62H-96H=CCH with CY=1
MOV      R7, A      ; save the result
MOV      A, #27H    ; A=27H
SUBB     A, #12H    ; 27H-12H-1=14H
MOV      R6, A      ; save the result

```

$A = 62H - 96H - 0 = CCH$
 $CY = 1$

$A = 27H - 12H - 1 = 14H$
 $CY = 0$

Solution:

We have $2762H - 1296H = 14CCH$.

دستور SUBB

• عملوند منبع می تواند تنها به چهار حالت زیر باشد.

تعداد سیکل ماشین	تعداد بایت	ترجمه دستور	نوع آدرس دهی	دستور
1	2	10010101 aaaaaaaaa	آدرس دهی مستقیم	SUBB A, direct
1	1	1001011i	آدرس دهی غیرمستقیم	SUBB A, @Ri
1	1	10011rrr	آدرس دهی ثبات	SUBB A, Rn
1	2	10010100 dddddddd	آدرس دهی بلافاصله	SUBB A, #data

دستور MUL AB

- در 8051 تنها می توان ضرب بایت در بایت را انجام داد.
- فرض می شود این بایت حاوی داده ی بدون علامت است.
- بایت پر ارزش تر نتیجه در ثبات B و بایت کم ارزش تر نتیجه در ثبات A قرار می گیرد.

MOV	A, #25H	;load 25H to reg. A
MOV	B, #65H	;load 65H to reg. B
MUL	AB	;25H * 65H = E99 where ;B = 0EH and A = 99H

- طول این دستور برابر با یک بایت است که در ۴ سیکل ماشین انجام می شود.

دستور DIV AB

- در 8051 تنها می توان تقسیم بایت در بایت را انجام داد.
- فرض می شود این بایت حاوی داده ی بدون علامت است.

- محتوای A بر محتوای B تقسیم می شود.
- خارج قسمت در ثبات A و باقیمانده در ثبات B قرار می گیرد.

MOV	A, #95	;load 95 to reg. A
MOV	B, #10	;load 10 to reg. B
DIV	AB	;A = 09(quotient) and ;B = 05(remainder)

- بعد از عمل تقسیم C همیشه 1 است.
- اگر $B \neq 0$ باشد، $OV=0$ است.
- اگر $B=0$ باشد، $OV=1$ می شود که نشان دهنده ی خطا است.
- در این حالت مقدار A و B نامشخص خواهند بود.

- طول این دستور برابر با یک بایت است که در ۴ سیکل ماشین انجام می شود.

مثال

- برنامه‌ای بنویسید که داده‌ی هگزادسیمال از 00H تا FFH را از پورت ۱ بگیرد و آن را به دسیمال تبدیل کرده و در R5، R6 و R7 ذخیره کند.
- ابتدا فرض می‌کنیم در پورت ۱ عدد FDH قرار دارد :

	Q	R
FD/0A =	19	3 (low digit)
19/0A =	2	5 (middle digit)
		2 (high digit)

FDH=253.

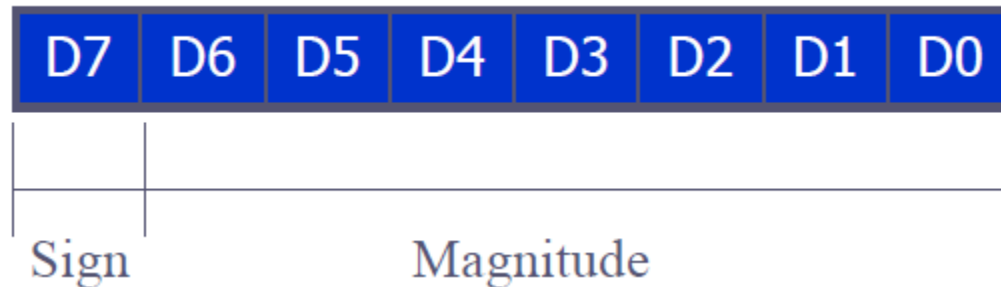
```

MOV  A, #0FFH
MOV  P1, A           ;make P1 an input port
MOV  A, P1           ;read data from P1
MOV  B, #10          ;B=0A hex
DIV  AB              ;divide by 10
MOV  R7, B           ;save lower digit
MOV  B, #10
DIV  AB              ;divide by 10 once more
MOV  R6, B           ;save the next digit
MOV  R5, A           ;save the last digit

```

اعداد علامت‌دار

- بیت D7 علامت است و بیت‌های D6-D0 مقدارند.
- اگر D7 برابر با صفر باشد عدد مثبت است.
- اگر D7 برابر با یک باشد عدد منفی است.



- اعداد مثبت از 0 تا +127 هستند.
- اعداد منفی از -1 تا -128 هستند که به صورت مکمل ۲ نمایش داده می‌شوند.
- مقدار عدد منفی را بنویس.
- آن را مکمل یک کن. (مقدار هر بیت را معکوس کن)
- به آن عدد یک را اضافه کن.

اعداد علامت دار

• مثال : عدد -34H

1. 0011 0100 34H given in binary
2. 1100 1011 invert each bit
3. 1100 1100 add 1 (which is CC in hex)

• مثال های دیگر :

Decimal	Binary	Hex
-128	1000 0000	80
-127	1000 0001	81
-126	1000 0010	82
...
-2	1111 1110	FE
-1	1111 1111	FF
0	0000 0000	00
+1	0000 0001	01
+2	0000 0010	02
...
+127	0111 1111	7F

مشکل سرریز در اعداد علامت‌دار

- اگر دو عدد مثبت جمع شوند و حاصل منفی شود.
- یا دو عدد منفی جمع شوند و حاصل مثبت شود.
- برنامه‌نویس باید به وقوع سرریز توجه نماید.

```
MOV    A, #+96          ; A=0110 0000 (A=60H)
MOV    R1, #+70         ; R1=0100 0110 (R1=46H)
ADD    A, R1            ; A=1010 0110
                        ; A=A6H=-90, INVALID
```

Solution:

	+96	0110 0000	
+	<u>+70</u>	<u>0100 0110</u>	
+	166	1010 0110	and OV =1

از نظر میکروکنترلر حاصل
عدد -90 است.

مشکل سرریز در اعداد علامت‌دار

```
MOV A, #-128 ;A=1000 0000 (A=80H)
MOV R4, #-2 ;R4=1111 1110 (R4=FEH)
ADD A, R4 ;A=0111 1110 (A=7EH=+126, INVALID)

-128      1000 0000
+   -2      1111 1110
-----
-130      0111 1110 and OV=1
```

OV = 1

The result +126 is wrong

```
MOV A, #-2 ;A=1111 1110 (A=FEH)
MOV R1, #-5 ;R1=1111 1011 (R1=FBH)
ADD A, R1 ;A=1111 1001 (A=F9H=-7,
;Correct, OV=0)

-2      1111 1110
+   -5      1111 1011
-----
-7      1111 1001 and OV=0
```

OV = 0

The result -7 is correct

پرچم OV

- در کار با اعداد بدون علامت، برنامه‌نویس باید پرچم نقلی (C) را چک کند.
- از دستورات زیر استفاده شود.

JC label

JNC label

- در کار با اعداد علامت‌دار، برنامه‌نویس باید پرچم سرریز (OV) را چک کند.
- از دستورات زیر استفاده شود.

JB OV, label

JNB OV, label

مکمل دو

• اگر خواستیم مکمل ۲ عددی را حساب کنیم :

CPL	A	;1's complement (invert)
ADD	A, #1	;add 1 to make 2's comp.

Complement



دستور INC

INC source

• محتوای source را یک واحد زیاد می کند.

• Increment

دستور	نوع آدرس دهی	ترجمه دستور	تعداد بایت	تعداد سیکل ماشین
INC A	اکومولاتور	00000100	1	1
INC direct	آدرس دهی مستقیم	00000101 aaaaaaaaa	2	1
INC @Ri	آدرس دهی غیرمستقیم	0000011i	1	1
INC Rn	آدرس دهی ثبات	00001rrr	1	1
INC DPTR	دیتا پوینتر	10100011	1	2

• اگر مقدار اولیه برابر با FFH باشد، مقدار جدید بعد از اجرای دستور برابر با 00H می شود.

• این دستور بر بیت های پرچم اثری ندارد.

دستور DEC

DEC source

• محتوای source را یک واحد کم می کند.

Decrement •

تعداد سیکل ماشین	تعداد بایت	ترجمه دستور	نوع آدرس دهی	دستور
1	1	00010100	اکومولاتور	DEC A
1	2	00010101 aaaaaaaaa	آدرس دهی مستقیم	DEC direct
1	1	0001011i	آدرس دهی غیرمستقیم	DEC @Ri
1	1	00011rrr	آدرس دهی ثبات	DEC Rn

• اگر مقدار اولیه برابر با 00H باشد، مقدار جدید بعد از اجرای دستور برابر با FFH می شود.

• این دستور بر بیت های پرچم اثری ندارد.

مثال

- اگر محتوای ثبات R0 برابر با 7FH باشد و محتوای خانه‌های 7EH و 7FH حافظه به ترتیب برابر با 00H و 40H باشد، دستورات زیر چه عملی انجام می‌دهند.

DEC @R0

DEC R0

DEC @R0

- ابتدا محتوای خانه 7FH که برابر با 40H است یک واحد کم می‌شود. یعنی برابر با 3FH می‌شود.
- سپس مقدار R0 یک واحد کم شده و برابر با 7EH می‌شود.
- سپس محتوای خانه 7EH که برابر با 00H است یک واحد کم شده و مقدار آن FFH می‌شود.

مرجع سریع دستورات محاسباتی

ADD	A, source
ADD	A, #data
ADDC	A, source
ADDC	A, #data
SUBB	A, source
SUBB	A, #data
INC	A
INC	source
DEC	A
DEC	source
INC	DPTR
MUL	AB
DIV	AB
DA	A

راهنما

Rn	آدرس‌دهی ثبات R0 تا R7
direct	آدرس ۸ بیتی حافظه داده (RAM) داخلی (00H-FFH)
@Ri	آدرس‌دهی غیرمستقیم با استفاده از ثبات‌های R0 یا R1
source	بایت منبع که هر یک از ثبات‌های Rn، آدرس مستقیم (direct)، یا آدرس غیرمستقیم @Ri می‌تواند باشد.
dest	بایت مقصد که هر یک از ثبات‌های Rn، آدرس مستقیم (direct)، یا آدرس غیرمستقیم @Ri می‌تواند باشد.
#data	عدد ۸ بیتی در دستور
#data16	عدد ۱۶ بیتی در دستور
bit	آدرس ۸ بیتی یک بیت اطلاعات
rel	آدرس نسبی یا آفست ۸ بیتی علامت‌دار
addr11	آدرس ۱۱ بیتی برای صفحه 2k بایتی حافظه
addr16	

زمان اجرا دستورات محاسباتی

- تمام دستورات محاسباتی در یک سیکل ماشین اجرا می‌شوند.
DA, DEC, INC, SUBB, ADDC, ADD
- دستور INC DPTR در دو سیکل ماشین اجرا می‌شود.
- دستورات MUL AB و DIV AB هر یک در ۴ سیکل ماشین اجرا می‌شوند.
- اگر پالس ساعت در میکروکنترلر 8051 برابر با 12MHz باشد، مدت زمان اجرای این دستور چقدر است؟
- ۴ میکروثانیه