



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)

دستور MOV

| | |
|--------------------|--------------|
| MOV A, direct | MOV A, 55H |
| MOV A, @Ri | MOV A, @R0 |
| MOV A, Rn | MOV A, R7 |
| MOV A, #data | MOV A, #35H |
| MOV direct, A | MOV 50H, A |
| MOV @Ri, A | MOV @R1, A |
| MOV Rn, A | MOV R4, A |
| MOV Rn, direct | MOV R2, 07H |
| MOV Rn, #data | MOV R0, #57H |
| MOV @Ri, direct | MOV @R0, P2 |
| MOV @Ri, #data | MOV @R1, #0 |
| MOV direct, Rn | MOV 30H, R5 |
| MOV direct, @Ri | MOV 45H, @R0 |
| MOV direct, #data | MOV P1, #FFH |
| MOV direct, direct | MOV P2, P1 |

• MOV dest, source

• برای حافظه داده داخلی (RAM)

• نکته :

• ثبات‌های کاربرد خاص (SFR)، با آدرس مستقیم یا نام سمبولیک آنها قابل دسترسی‌اند.

| تعداد سیکل ماشین | تعداد بایت | ترجمه دستور | دستور |
|---------------------|------------|------------------------------|--------------------|
| 1 | 2 | 11100101 aaaaaaaaa | MOV A, direct |
| 1 | 1 | 1110011i | MOV A, @Ri |
| 1 | 1 | 11101rrr | MOV A, Rn |
| 1 | 2 | 01110100 dddddddd | MOV A, #data |
| 1 | 2 | 11110101 aaaaaaaaa | MOV direct, A |
| 1 | 1 | 1111011i | MOV @Ri, A |
| 1 | 1 | 11111rrr | MOV Rn, A |
| 2 | 2 | 10101rrr aaaaaaaaa | MOV Rn, direct |
| 1 | 2 | 01111rrr dddddddd | MOV Rn, #data |
| 2 | 2 | 1010011i aaaaaaaaa | MOV @Ri, direct |
| 1 | 2 | 0111011i dddddddd | MOV @Ri, #data |
| 2 | 2 | 10001rrr aaaaaaaaa | MOV direct, Rn |
| 2 | 2 | 1000011i aaaaaaaaa | MOV direct, @Ri |
| 2 | 3 | 01110101 aaaaaaaaa dddddddd | MOV direct, #data |
| 3 | 3 | 10000101 aaaaaaaaa aaaaaaaaa | MOV direct, direct |

دستور ۱۶ بیتی MOV

MOV DPTR, #data16

- برای مقدار اولیه دادن به اشاره گر داده.
- ثبات اشاره گر داده برای دسترسی به حافظه داده خارجی (RAM) یا جداول حافظه (ROM) استفاده می شود.

- دستور ۳ بایتی است و در ۲ سیکل ماشین اجرا می شود.
- ترجمه دستور :

10010000 dddddddd dddddddd •

مثال :

MOV DPTR, #150AH •

دستور MOVX

• Move External

• میکروکنترلر 8051 دارای 128 بیت حافظه داده (RAM) است.

• می توان حافظه ی خارجی داده تا حداکثر 64K بیت به 8051 متصل کرد.

• این حافظه تنها با دستور MOVX قابل دسترسی است.

• حتما یکی از عملوندها A است.

• MOVX dest_byte, source_byte

• دستور MOVX یک بیت داده را بین حافظه خارجی داده و A انتقال می دهد.

دستور MOVX

• از این دستور به دو صورت استفاده می‌شود.

- با استفاده از آدرس غیرمستقیم ۸ بیتی
- از R0 و R1 برای مشخص کردن آدرس ۸ بیتی استفاده می‌شود.
- برای آدرس‌های یک حافظه RAM کوچک (نهایتاً 256 بایت) از طریق پورت P0

- با استفاده از آدرس غیرمستقیم ۱۶ بیتی
- از DPTR برای مشخص کردن آدرس ۱۶ بیتی استفاده می‌شود.
- برای اتصال به حافظه‌های بزرگ خارجی نهایتاً تا 64K بایت.
- ۸ بیت پرازش‌تر آدرس توسط ثبات DPH در پورت P2 قرار می‌گیرد و ۸ بیت کم‌ارزش‌تر آدرس توسط ثبات DPL در پورت P0 قرار می‌گیرد.
- البته در پورت P0 آدرس و داده با یکدیگر مالتی پلکس می‌شوند. یعنی ابتدا آدرس و سپس داده از طریق همان پورت منتقل می‌شود.

دستور MOVX

| | | |
|---------------|-------------|--|
| MOVX A, @Ri | MOVX A, @R1 | اگر مقدار R1 برابر با 60H باشد، مقدار خانه 60H حافظه RAM خارجی به A منتقل می‌شود. |
| MOVX @Ri, A | MOVX @R0, A | اگر مقدار R0 برابر با FFH باشد، محتوای A در آخرین خانه حافظه RAM خارجی ذخیره می‌شود. |
| MOVX A, @DPTR | | اگر مقدار DPTR برابر با 1200H باشد، مقدار خانه 1200H حافظه RAM خارجی در A قرار می‌گیرد. |
| MOVX @DPTR, A | | اگر مقدار DPTR برابر با FFFFH باشد، محتوای A در آخرین خانه حافظه RAM خارجی قرار می‌گیرد. |

دستور MOVX

| دستور | ترجمه دستور | تعداد بایت | تعداد سیکل ماشین |
|---------------|-------------|------------|------------------|
| MOVX A, @Ri | 1110001i | 1 | 2 |
| MOVX @Ri, A | 1111001i | 1 | 2 |
| MOVX A, @DPTR | 11100000 | 1 | 2 |
| MOVX @DPTR, A | 11110000 | 1 | 2 |

دستور MOV C

• Move Code

- برای خواندن از حافظه کد است.
- دقت شود در حافظه کد (ROM) تنها می توان اطلاعات را خواند و نمی توان نوشت.

• این دستور برای جستجو در جداول استفاده می شود.

• از این دستور به دو صورت استفاده می شود.

• دستور MOV C با اشاره گر داده (DPTR)

MOV C A, @A+DPTR

• دستور MOV C با کنتور برنامه (PC)

MOV C A, @A+PC

دستور MOV C

MOV C A, @A+DPTR

- مثال : برنامه‌ای بنویسید که یک رشته string که داخل حافظه ROM در آدرس 100H قرار دارد را به پورت P1 بفرستد.

```

MOV DPTR, #100H
LOOP1 : CLR A
        MOV C A, @A+DPTR
        MOV P1, A
        JZ EXIT ← Jump If Zero
        INC DPTR
        SJMP LOOP1
EXIT :   .....
        ORG 100H
        DB "TEHRAN" 0 ← Define Byte

```

دستور MOV C

MOV C A, @A+PC

- در این دستور PC به عنوان ثبات پایه استفاده می‌شود و A به عنوان آفست.
- آدرس ابتدای جدول با PC مشخص می‌شود و A شماره عنصر جدول است.
- برای دسترسی به عناصر جدول معمولاً از subroutine استفاده می‌شود. و سابروتین با دستور CALL فراخوانی می‌شود.

```
MOV A, #ENTRY_NUMBER
CALL LOOK_UP
```

.....

```
LOOK_UP : INC A
```

```
MOV C A, @A+PC
```

یک بایت دستور

```
RET
```

```
TABLE : DB data0, data1, data2,....
```

اگر بین دستور MOV C و ابتدای جدول، چندین بایت فاصله باشد، باید قبل از دستور MOV C تعداد بایت‌ها به A اضافه گردد.

هنگامی که این دستور اجرا می‌شود، PC آدرس دستور RET را دارد و چون به A یک واحد افزوده شده می‌توان به خانه‌های جدول دسترسی پیدا کرد.

دستور MOV C

| دستور | ترجمه دستور | تعداد بایت | تعداد سیکل ماشین |
|------------------|-------------|------------|---------------------|
| MOV C A, @A+DPTR | 10010011 | 1 | 2 |
| MOV C A, @A+PC | 10000011 | 1 | 2 |

مثال

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

- الف) با استفاده از سابروتین و جدول.
- ب) با استفاده از سابروتین بدون جدول.

الف

```
MOV A, #0FFH
MOV P1, A
BACK : MOV A, P1
CALL SQUARE
MOV P2, A
SJMP BACK
ORG 300H
SQUARE : INC A
MOVC A, @A+PC
RET
TABLE : DB 0,1,4,9,16,25,36,49,64,81
```

ب

```
MOV A, #0FFH
MOV P1, A
BACK : MOV A, P1
CALL SQUARE
MOV P2, A
SJMP BACK
ORG 300H
SQUARE : PUSH 0F0H
MOV 0F0H, A
MUL AB
POP 0F0H
RET
```

بهتر است موقع استفاده از سابروتین، در ابتدای سابروتین مقدار ثبات‌ها را در پشته ذخیره و در انتهای سابروتین محتوای آنها بازیابی شود تا محتوای ثبات‌ها در حین اجرای سابروتین تغییر نکند.

دستور XCH

XCH A, source

• Exchange

• این دستور محتوای A را با محتوای source جابجا می کند.

| تعداد سیکل ماشین | تعداد بایت | ترجمه دستور | مثال | دستور |
|---------------------|------------|--------------------|------------|---------------|
| 1 | 1 | 11001rrr | XCH A, R4 | XCH A, Rn |
| 1 | 2 | 11000101 aaaaaaaaa | XCH A, 40H | XCH A, direct |
| 1 | 1 | 1100011i | XCH A, @R1 | XCH A, @Ri |

دستور XCHD

XCHD A, @Ri

Exchange •

- این دستور ۴ بیت کم ارزش تر خانه حافظه‌ای که Ri به آن اشاره می‌کند را با ۴ بیت کم ارزش تر A جابجا می‌کند.

| تعداد سیکل ماشین | تعداد بایت | ترجمه دستور | مثال | دستور |
|---------------------|------------|-------------|------------|-------------|
| 1 | 1 | 1101011i | XCH A, @R0 | XCHD A, @Ri |

- مثال : محتوای ثبات‌های A و B بعد از اجرای این دستورات چیست؟

MOV 0F0H, #12H

MOV R0, #0F0H

MOV A, #34H

XCH A, 0F0H

XCHD A, @R0

A = 14H

B = 32H

دستور PUSH و POP

PUSH direct •

- در این دستور ابتدا اشاره گر پشته (SP) یک واحد افزایش می یابد. سپس محتوای خانه ی حافظه ای که آدرس آن در direct است، در خانه حافظه ای که آدرس آن را SP مشخص کرده است ریخته می شود.
- این دستور بر بیت های پرچم اثری ندارد.

POP direct •

- در این دستور ابتدا محتوای خانه ای که آدرس آن را SP مشخص کرده است خوانده شده و در خانه ی حافظه ای که آدرس آن در direct است ریخته می شود. سپس از اشاره گر پشته یک واحد کم می شود.
- این دستور بر بیت های پرچم اثری ندارد.

| تعداد سیکل ماشین | تعداد بایت | ترجمه دستور | دستور |
|------------------|------------|--------------------|-------------|
| 2 | 2 | 11000000 aaaaaaaaa | PUSH direct |
| 2 | 2 | 11010000 aaaaaaaaa | POP direct |

مرجع سریع دستورات انتقال اطلاعات

| | |
|------|---------------|
| MOV | A, source |
| MOV | A, #data |
| MOV | dest, A |
| MOV | dest, source |
| MOV | dest, #data |
| MOV | DPTR, #data16 |
| MOVC | A, @A+DPTR |
| MOVC | A, @A+PC |

| | |
|------|-----------|
| MOVX | A, @Ri |
| MOVX | A, @DPTR |
| MOVX | @Ri, A |
| MOVX | @DPTR, A |
| PUSH | direct |
| POP | direct |
| XCH | A, source |
| XCHD | A, @Ri |

راهنما

| | |
|---------|---|
| Rn | آدرس دهی ثبات R0 تا R7 |
| direct | آدرس ۸ بیتی حافظه داده (RAM) داخلی (00H-FFH) |
| @Ri | آدرس دهی غیر مستقیم 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 | |