

دستورات منطقی

دستورات منطقی ۸۰۸۶ عملیات منطقی and ، or ، xor و not را به صورت بیت به بیت روی عملوندها انجام می دهند .

[AND](#)

[OR](#)

[XOR](#)

[NOT](#)

[TEST](#)

[دستکاری بیت ها](#)

AND

دستورالعمل and عمل and منطقی را روی دو عملوند خود انجام می دهد و نتیجه را در عملوند اول ذخیره می کند. فرم کلی آن به صورت زیر است :

and dest, src

طبق جدول زیر عمل and روی بیت های متناظر دو عملوند انجام می شود. هر بیت نتیجه در صورتی یک است که بیت های متناظر هر دو عملوند یک باشند در غیر اینصورت صفر می شود .

A	B	A and B
1	1	1
1	0	0
0	1	0
0	0	0

دستورالعمل and به شکل های زیر می تواند بکار برود:

and register, register
and memory, register
and register, memory
and register, immediate data
and memory, immediate data
and AX/AL, immediate data

دستورالعمل and روی فلگ های زیر تاثیر می گذارد :

- فلگ carry و overflow را صفر می کند.
- فلگ های zero، sign و parity با توجه به نتیجه تاثیر می پذیرند.
- فلگ zero وقتی دو عملوند در هیچ مکانی بیت مشابه نداشته باشند یک می شود.

مثل

```
mov AX, C123h
and AX, 82F6h
```

```
  1100 0001 0010 0011
and 1000 0010 1111 0110
-----
  1000 0000 0010 0010
```

AX=8022h

دستورالعمل and برای محاسبه سریع باقیمانده یک عدد بر توانی از ۲ می تواند استفاده شود. برای پیدا کردن باقیمانده عملوندی بر مقدار 2^n کافیست and عملوند با مقدار $2^n - 1$ محاسبه شود.

مثال. دستور زیر باقیمانده AX بر عدد ۸ را محاسبه می کند.

and AX, 7

OR

دستورالعمل or عمل or منطقی را روی عملوندهای خود انجام می دهد. فرم کلی آن به صورت زیر است :

or dest, src

عمل or ، طبق جدول زیر، روی عملوندها بیت به بیت انجام می گیرد. هر بیت نتیجه دستور or در صورتی صفر است که بیت های متناظر هر دو عملوند صفر باشند در غیر اینصورت یک می شود .

A	B	A or B
1	1	1
1	0	1
0	1	1
0	0	0

دستورالعمل or مشابه عمل and روی فلگ های Carry ، Zero ، Sign ، Overflow و Parity تاثیر می گذارد. فلگ zero وقتی یک می شود که هر دو عملوند صفر باشند .

عملوندهای دستورالعمل or مشابه دستورالعمل and می تواند ثبات، حافظه یا داده فوری باشد .

مثل.

mov AX, C123h

or AX, E831h

```

1100 0001 0010 0011
or 1110 1000 0011 0001
-----
1110 1001 0011 0011

```

AX=E933h

XOR

دستورالعمل xor عمل xor منطقی را روی بیت های دو عملوند خود انجام می دهد و نتیجه را در عملوند اول ذخیره می کند. فرم کلی آن به صورت زیر است :

xor dest, src

هر بیت نتیجه دستورالعمل xor مطابق جدول زیر تنظیم می شد. بیت نتیجه زمانی صفر است که هر دو بیت عملوند مشابه هم باشند در غیر اینصورت یک می شود .

A	B	A xor B
1	1	0
1	0	1
0	1	1
0	0	0

دستورالعمل xor مشابه عمل and روی فلگ ها تاثیر می گذارد. اگر هر دو عملوند دستورالعمل مساوی باشند نتیجه صفر می شود و در نتیجه فلگ zero یک می شود.

برای صفر کردن ثبات ها می توان از دستورالعمل xor register, register استفاده کرد که کوتاهتر از دستورالعمل mov register, 0 است. مثل.

```
mov AX, C123h
xor AX, E831h
```

```

  1100 0001 0010 0011
xor 1110 1000 0011 0001
-----
  0010 1001 0001 0010
AX=2912h
```

NOT

دستورالعمل not مکمل یک عملوند خود را محاسبه و در آن ذخیره می کند. فرم کلی آن به صورت زیر است:

```
not dest
```

دستورالعمل not بیت های عملوند را عکس می کند؛ صفرها را به یک و یک ها را به صفر تبدیل می نماید.

دستورالعمل not به یکی از دو حالت زیر می تواند استفاده شود:

```
not register
not memory
```

دستورالعمل not روی هیچکدام از فلگ ها تاثیر ندارد.

مثل.

```
mov AX, C123h
not AX
```

```

not 1100 0001 0010 0011
-----
  0011 1110 1101 1100
AX=3EDCh
```

TEST

دستورالعمل test مشابه عمل and است و عمل and منطقی را روی دو عملوند خود انجام می دهد با این تفاوت که نتیجه را جایی ذخیره نمی کند.

```
test dest, src
```

فلگ های Carry، Zero، Sign، Overflow و Parity مشابه دستورالعمل and تاثیر می پذیرند.

دستورالعمل test برای بررسی یک بودن بیتی می تواند استفاده شود.

مثال. دستور زیر مقدار ۱ را با ثبات AL به طور منطقی and می کند. اگر بیت شماره ۰ ثبات AL صفر باشد نتیجه دستور صفر شده و فلگ zero برابر با یک می شود در غیر اینصورت فلگ zero صفر می شود. بررسی فلگ zero بعد از این اجرای دستور نشان می دهد که بیت صفر ثبات AL یک بوده است یا خیر.

```
test AL, 1
```

مثال. دستور زیر برای بررسی بیت های ۰، ۲ و ۸ ثبات DX می تواند استفاده شود. اگر همگی صفر باشند فلگ zero یک می شود.

```
test DX, 105h
```

دستکاری بیت ها

از دستورات منطقی برای دستکاری بیت های معینی از عملوند مقصد بدون تاثیر روی بیت های دیگر آن می استفاده می شود. برای این منظور یک ماسک ساخته می شود. یک ماسک مقدار است که بیت های مشخصی از یک عملوند را صفر یا یک می کند بدون اینکه تغییری روی بقیه بیت های آن داشته باشد.

صفر کردن بیت ها با دستور العمل AND

با دقت در جدول and می توان مشاهده کرد که نتیجه and عدد صفر با هر بیتی برابر با صفر می شود. اگر عدد یک با بیتی and شود نتیجه همان بیت می شود. از این خاصیت برای صفر کردن بیت های انتخابی یک مقدار بدون تاثیر روی بقیه بیت ها می توان استفاده کرد.

مثل در مثال زیر بیت شماره ۵ ثبات AX صفر می شود و بقیه بیت ها بدون تغییر باقی می مانند.

```
mov AX, C123h
and AX, FFDFh
```

یک کردن بیت ها با دستور العمل OR

دستور العمل or را می توانید برای یک کردن بیت های انتخابی یک عدد بکار ببرید.

مثل در مثال زیر بیت شماره ۳ ثبات AX یک می شود و بقیه بیت ها بدون تغییر باقی می مانند.

```
mov AX, C123h
or AX, 8
```