

دستورات شیفت

دستورات شیفت یک رشته بیتی را به سمت راست یا چپ حرکت می دهند. توسط این دستورات می توان روی بیت های داده کار کرد؛ داده را ادغام یا جدا کرد و عملیات محاسباتی را انجام داد. ریزپردازنده ۸۰۸۶ سه دستورالعمل شیفت (shr, shl/sal) و sar) دارد. بخش زیر هر یک از این دستورالعمل ها را شرح می دهد.

[SHL/SAL](#)

[SHR](#)

[SAR](#)

[کاربردهای شیفت](#)

عمل شیفت بیت های داده را حرکت می دهد. حرکت بیت ها می تواند به سمت چپ (به سمت بیت های با ارزش) یا راست (به سمت بیت های کم ارزش) باشد. فلگ Carry معمولاً آخرین بیت شیفت داده شده که از عملوند خارج می شود را می گیرد.

دو نوع شیفت وجود دارد: شیفت منطقی و شیفت ریاضی. شیفت منطقی ساده ترین شیفت است که به طریق ساده ای بیت ها را شیفت می دهد. در شیفت ریاضی علامت عدد حفظ می شود.

مثال. یک عدد شیفت داده شده یک بایستی نشان داده شده است.

داده اصلی	1 1 1 0 1 0 1 0
بعد از یک شیفت به چپ	1 1 0 1 0 1 0 0
بعد از یک شیفت به راست	0 1 1 1 0 1 0 1

توجه کنید که بیت های جدیدی که وارد می شوند همیشه صفر هستند.

SHL/SAL

دستورالعمل shl (shift left) یا sal (shift arithmetic left) بیت های داده را به سمت چپ حرکت می دهد. فرم کلی آنها به صورت زیر است:

shl dest, count

sal dest, count

shl و sal معادل هستند و یک دستورالعمل را نشان می دهند یعنی کد یکسانی دارند. این دستورالعمل ها هر بیت عملوند مقصد را به سمت چپ عدد به تعداد عملوند count حرکت می دهند. از سمت راست عدد ۰ وارد عدد می شود و آخرین بیتی که از سمت چپ خارج می شود وارد فلگ carry می شود.



عملوند اول مقداری است که شیفت داده می شود و عملوند مقصد است count. تعداد شیفت ها را مشخص می کند و می تواند عدد 1 یا برای تعداد شیفت های بالاتر ثبات CL باشد. نوشتن تعداد شیفت بیشتر از 1 مستقیماً در دستور غیر مجاز است.

دستورالعمل shl/sal می تواند به صورت های زیر بکار رود:

shl register, 1

shl memory, 1

shl register, CL

shl memory, CL

دستورالعمل shl/sal به صورت زیر روی فلگ تاثیر می گذارد:

- اگر تعداد شیفت صفر باشد فلگ ها تغییری نمی کنند.
- فلگ carry آخرین بیت خارج شده از سمت چپ عملوند را نگه می دارد.
- فلگ overflow در یک بیت شیفت یک می شود اگر دو بیت آخر عملوند متفاوت باشند. به عبارت دیگر بعد از عمل شیفت بیت علامت عدد تغییر کند. برای شیفت های بیشتر از یکبار نامعین است.
- فلگ zero، sign و parity با توجه به نتیجه تغییر می کنند.
- فلگ Auxilury Carry نامعین است.

مثل.

```
mov AX, 4123h
shl AX, 1 ; shift 1 bit to left, ax = 8246H, CF = 0
```

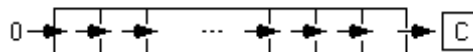
SHR

دستورالعمل shr (shift right) بیت های داده را به سمت راست حرکت می دهد. فرم کلی آنها به صورت زیر است :

```
shr dest, count
```

دستورالعمل shr کلیه بیت های عملوند مقصد را به تعداد count به سمت راست شیفت منطقی می دهد. از سمت چپ صفر وارد عملوند می شود و آخرین بیتی که از سمت راست خارج می شود وارد فلگ Carry می شود .

دستورالعمل shr مانند shl استفاده می شود؛ عملوند آن می تواند ثبات یا مکانی از حافظه باشد و تعداد شیفت ها می تواند عدد 1 یا ثبات CL باشد .



دستورالعمل shr فلگ ها را به صورت زیر تنظیم می کند:

- اگر تعداد شیفت صفر باشد فلگ ها تغییری نمی کنند.
- فلگ carry آخرین بیت خارج شده از سمت راست عملوند را نگه می دارد.
- در یک بیت شیفت فلگ overflow یک می شود اگر دو بیت آخر عملوند متفاوت باشند. به عبارت دیگر اگر بعد از عمل شیفت بیت علامت عدد تغییر کند. برای شیفت های بیشتر از یکبار نامعین است.
- فلگ zero، sign و parity با توجه به نتیجه تغییر می کنند.
- فلگ auxilury carry نامعین است.

مثل.

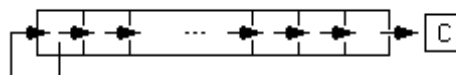
```
mov AX, C1A5h
mov CL,3
shr AX, CL ; shift 3 bit to right, ax = 1834h, CF = 1
```

SAR

دستورالعمل sar (shift arithmetic right) مانند دستورالعمل shr است با این تفاوت که علامت عملوند تغییری را نمی دهد. فرم کلی آن به صورت زیر است :

```
sar dest, count
```

این شیفت برای اعداد علامتدار طراحی شده است و بیت های عملوند مقصد را به سمت راست شیفت ریاضی می دهد و بیت علامت را در خودش کپی می کند .



دستورالعمل sar مشابه دستورالعمل shr بکار می رود و به همان صورت روی فلگ ها تاثیر می گذارد.

مثل.

```
mov AX, C1A5h
sar AX, 1 ; shift 1 bit to right, ax = E0D2h, CF = 1
```

کاربردهای شیفت

مثال. فرض کنید دو نیبل پائینی ثبات های AL و AH را با هم به صورت زیر ترکیب کنید. کد زیر این کار را انجام می دهد .



```
mov CL, 4
shl AH, CL
and AL, 0Fh
or AL, AH
```

مثال. فرض کنید دو نیبل ثبات AL را از هم جدا کرده و نیمه سمت چپ را در ثبات AH و نیمه سمت راست را در ثبات AL به صورت زیر قرار دهید. کدهای زیر این عمل را انجام می دهد .



```
mov AH, AL
mov CL, 4
shr AH, CL
and AL, 0Fh
```

مثال. هر شیفت به چپ باعث دو برابر شدن عملوند می شود که سرعت بیشتری نسبت به عمل mul دارد. دستورالعمل های shl/sal برای ضرب مقادیر علامت دار یا بدون علامت در توان های ۲ استفاده می شود. دستور زیر مقدار ثبات AX را در عدد ۴ ضرب می کند .

```
mov CL, 2
shl AX, CL
```

مثال. برای محاسبه $10 \times AX$ می توانید به روش زیر از دستورالعمل شیفت چپ استفاده کنید (با توجه به اینکه $10 \times AX = 8 \times AX + 2 \times AX$).

```
shl AX, 1
mov BX, AX
shl AX, 1
shl AX, 1
add AX, BX
```

مثال. کدهای زیر حاصل ضرب $AX \times 7$ را محاسبه می کنند (با توجه به اینکه $ax \times 7 = (ax \times 8) - ax$).

```
mov BX, AX
shl AX, 1
shl AX, 1
shl AX, 1
sub AX, BX
```

مثال. چون یک شیفت منطقی به سمت راست مقدار یک عدد صحیح بدون علامت را نصف می کند می توان برای تقسیم بر توان های ۲ از آن استفاده کرد. دستورات زیر خارج قسمت مقدار ثبات AX بر ۸ را محاسبه می کنند .

```
mov CL,3
shr AX, CL
```

مثال. برای انجام تقسیم علامتدار بر توان های ۲ از شیفت ریاضی راست استفاده می شود. دستور زیر مقدار ثبات AX را بر عدد ۳۲ تقسیم می کند .

```
mov CL,5
sar AX, CL
```

مثال. توجه کنید اگر عملوند منفی باشد نتیجه دو دستور sar و idiv متفاوت می شود. به دستورات زیر دقت کنید .

```
mov ax, -15
cwd
mov bx, 2
idiv      ; خارج قسمت ۷- می شود;
```

```
mov ax, -15
sar ax, 1 ; خارج قسمت ۸- می شود;
```

مثال. از شیفت ریاضی راست می توانید برای گسترش ریاضی یک عدد علامتدار استفاده کنید. به کدهای زیر دقت کنید .

```
; معادل دستور CBW
mov AH, AL
mov CL, 7
sar AH, CL
```

```
; معادل دستور CWD
mov DX, AX
mov CL, 15
sar DX, CL
```

البته وقتی یک دستور cbw یا cwd برای گسترش وجود دارد کسی از دو دستور استفاده نمی کند، ولی دستور شیفت اجازه می دهد که مقدار یک ثبات را در هر ثبات دیگر هم اندازه ای به طور ریاضی گسترش دهید :

```
; گسترش ریاضی BX به DX:BX
mov DX, BX
mov CL, 15
sar DX, CL
```