۳-۳) دستور حلقه LOOP :

گاهی یک حلقه می بایست به تعداد مشخصی تکرار شود و یا اگر به شرط مشخصی رسید متوقف شود.

الگو:

LOOP آدرس هدف یا ابتدای حلقه

با اجرای این دستور یک واحد از ثبات CX کم شده و در صورت صفر نبودن این ثبات تکرار حلقه از برچسب (آدرس)ادامه می یابد.

معمولاً تعداد دفعات تکرار عملاً بایستی از قبل در ثبات CX ذخیره شود.

نکته : دستور LOOP معادل دستورات زیر است:

```
DEC  CX
JNZ  آدرس
```

مثال کتاب: در مثال زیر ۳ بار دستورات داخل حلقه اجرا می شود.

```
        PAGE 100,110
TITLE   'loop.asm' loop  program
;-----------------------------------------------
        .MODEL SMALL
        .STACK  64          ;Define stack
;-----------------------------------------------
        .CODE               ;Define code segment
MAIN    PROC FAR
        MOV AX,@data        ;1- Set data segment
        MOV DS,AX           ;2- address
;
        MOV AL,1            ;3-AL=1
        MOV BL,2            ;4-BL=2
        MOV CX,3            ;5-CX=3
        MOV DL,4            ;6-DL=4
AGAIN:  ADD AL,BL           ;7-AL=AL+BL
        MUL DL              ;8-AX=AL*DL
        LOOP AGAIN          ;9-Loop to AGAIN until
;                           ;CX=0
;
        MOV AX,4C00H        ;10-End of
        INT 21H             ;11- processing
MAIN    ENDP                ; End of procedure
        END MAIN            ; End of program
```

شکل (۱۱-۳) نمونه‌ای از برنامه اسمبلی با دستور LOOP

دستور **LOOP** به روشهای دیگری هم به شرح زیر بکارمی رود.

دستور	اسم دیگر	شرط تکرار
LOOPZ	LOOPE	CX<>0 & ZF=1
LOOPNZ	LOOPNE	CX<>0 & ZF=0
JCXZ	-------	CX=0

نکته : دستور LOOPZ و LOOPE هر دو یکی هستند ولی برای خوانایی برنامه زمانی که بخواهیم مساوی بودن یک نتیجه را
بررسی کنیم از LOOPE و زمانی که بخواهیم صفر بودن نتیجه ای را بدانیم از LOOPZ استفاده می کنیم.

تمرین : چگونه می توان با دستور LOOP در برنامه تاخیرهای مختلف ایجاد کرد ؟ کاربرد تاخیر را بنویسید.

مثال کتاب:
جدولی از آفست آدرس ARRAY به طول ۱۰۰بایت در حافظه کامپیوتر وجود دارد , بررسی نمایید که آیا محتوای تمام خانه های حافظه مذکور دارای مقدار FFH هستند یا نه, اولین خانه حافظه ای که محتوای آن برابر با FFH نبود , برنامه به سیستم عامل برگردد.

```
MOV BX, OFFSET ARRAY
DEC  BX
MOV  CX,100
NEXT: INC   BX
CMP   [BX],0FFH

LOOPE  NEXT

MOV   AX,4C00H
INT   21H
```

چند مثال ساده از آنچه تاکنون گفته ایم:

1 - برنامه ای بنویسید که بتواند دو عدد 20 و 31 را بایکدیگر جمع نموده و نتیجه را در ثبات DL ذخیره نماید.

جواب:

روش اول

```
MOV   AL,20
MOV   BL,31
ADD   AL,BL
MOV   DL,AL
```

روش دوم

```
MOV   AL,20
MOV   DL,31
ADD   DL,AL
```

روش سوم

```
MOV    DL,20
ADD    DL,31
```

۲- برنامه ای بنویسید که محتویات آدرسهای 300H الی 304H حافظه را با یکدیگر جمع نموده و با فرض اینکه نتیجه این عمل جمع
از هشت بیت بیشتر نمی شود مجموع را در ثبات AL ذخیره نماید.

```
MOV AL,[0300H]
ADD AL,[0301H]
ADD AL,[0302H]
ADD AL,[0303H]
ADD AL,[0304H]
```

۳ -برنامه ای بنویسید که بتواند دو عدد ۸ و ۱۵ را در هم ضرب کرده و قسمت پر ارزش نتیجه ضرب را از ۱۰ کم کند و نتیجه کل
در ثبات DH ذخیره نماید.

```
MOV  AL,8
MOV  BL,15
MUL  BL
SUB   AH,10
MOV  DH,AH
```

تمرین : برنامه ای بنویسید که ابتدا در ثبات AL عدد ۲۱ را قرار دهد . سپس ثبات BL را با عدد ۷ جمع و نتیجه را در BL قرار دهد . سپس محتوای AL را در BL ضرب و نتیجه را در DX ذخیره کند و سرانجام یک واحد از DX کم کند.

۱– دستورات منطقی :

این دسنورات عبارتند از TEST,XOR,OR,AND,NOT و ....که توط آنها عملیات منطقی بر روی هریک از بیتهای ثباتها یا خانه های حافظه انجام می شود.در ضمن رو بیتهای پرچم ZF, CF,OF,SF,PF, اثر می گذارند.

۴-۱-دستور NOT

این دستور مکمل ۱ عملوند را محاسبه و در عملوند قرار می دهد. (در واقع بیت به بیت معکوس می گردند)

الگو:

NOT عملوند

مثال : نتیجه ثبات DL در قطعه برنامه زیر چیست ؟

```
MOV DL,8AH
NOT DL
```

جواب : در خط دوم از 8AH مکمل ۱ گرفته می شود و نتیجه در ثبات DL ذخیره می شود . برای بدست آوردن نتیجه ابتداء بایستی 8AHرا به مبناء ۲ تبدیل و سپس مکمل ۱ بگیریم:

$8A^H \longrightarrow 1000\ 1010^B \xrightarrow{\text{مکمل ۱}} 0111\ 0101 \longrightarrow 75^H$

$DL=75^H$

۴-۲- دستور AND:

در این دستور اگر هر دو بیت نظیر دو عملوند یک باشد بیت نظیر نتیجه هم یک می شود , اما اگر هریک از دو بیت دو عملوند صفر باشد بیت نظیر نتیجه آنها صفر خواهد شد.

الگو:

AND عملوند ۱ , عملوند ۲

جدول درستی آن به شرح زیر است:

X	Y	X . Y
0	0	0
0	1	0
1	0	0
1	1	1

مثال : نتیجه ثبات BL در قطعه برنامه زیر چیست ؟

```
MOV  BL,35H
AND  BL,0FH
```

جواب : در این برنامه بین 35H و 0FH عمل AND منطقی انجام و نتیجه در ثبات BL ذخیره می شود.

```
0011 0101
0000 1111
---------
```

0000 0101 ⟶ 05H ⟶ BL=05H

نکته: یکی از کارهای دستور and عمل reset کردن یا صفر کردن است. مثلا در دستور زیر ثبات ax مقدارش صفر می شود.

```
And   ax,0
```

۴-۳- دستور OR:

در این دستور اگر هریک از دو بیت نظیر دو عملوند یک باشد بیت نظیر نتیجه هم یک می شود , اگر هردو بیت دو عملوند صفر باشد بیت نظیر نتیجه آنها صفر خواهد شد.

الگو:

OR عملوند ۱ , عملوند ۲

جدول درستی آن به شرح زیر است:

X	Y	X + Y
0	0	0
0	1	1
1	0	1
1	1	1

مثال : نتیجه ثبات AL در قطعه برنامه زیر چیست ؟

```
MOV BL,0A5H
MOV AL,2AH
OR AL,BL
```

جواب:

```
1010 0101
0010 1010
```

1010 1111 → AFH → AL=AFH

نکته : از دستور OR منطقی جهت تست صفر بودن یک ثبات استفاده می شود . مثلاً در دستور OR BL,0 اگر BLصفر باشد
مطمئناً ZF=1 می شود بنابراین با کنترل پرچم ZF می توان صفر بودن یک ثبات را تست نمود.

۴-۴- دستور XOR:

در این دستور اگر فقط یکی از دو بیت اطلاعات یک باشد نتیجه هم یک می شود ولی اگر هردو بیت صفر باشد یا هر دو بیت یک باشند نتیجه آنها صفر خواهد شد.

الگو:

XOR عملوند ۱ , عملوند ۲

جدول درستی آن به شرح زیر است:

X	Y	XOR
0	0	0
0	1	1
1	0	1
1	1	0

مثال : نتیجه ثبات AL در قطعه برنامه زیر چیست ؟

```
MOV  CL,2DH
MOV  AL,0C2H
```

XOR AL,CL

جواب:

0010 1101
1100 0010

1110 1111 ⟶ EFH ⟶ AL=EFH

نکته۱ : جهت پاک کردن یک ثبات از XOR استفاده می شود (هر ثباتی با خودش XOR شود صفر می شود)
XOR AX,AX

نکته۲: کد اسکی حروف بزرگ انگلیسی از 65 الی 90 یا از 41H الی 5AH می باشد . حروف کوچک نیز دارای کد اسکی بین 97 الی 122 و یا از 61H الی 7AH می باشند . تنها تفاوت بین حروف بزرگ و کوچک در بیت d5 می باشد که این بیت درحروف کوچک برابر 1 و در حروف بزرگ برابر 0 است . بنابراین جهت تبدیل حروف کوچک به بزرگ و بلعکس از دستور زیراستفاده می شود:

XOR حرف بزرگ یا کوچک, 0010 0000B

مثال : کد اسکی حرف A برابر است با 01000001 و حالا اگر آن را با عدد 00100000 عمل XOR انجام بدهیم نتیجه 011000001 میشود که معادل کد اسکی حرف a می باشد و بالعکس.

۴-۵- دستور TEST:

این دستور مانند دستور AND منطقی عمل می کند بدون ذخیره نتیجه و فقط بیتهای ثبات پرچم را تغییر می دهد.

الگو:

TEST عملوند ۱ , عملوند ۲

مثال۱:
قطعه برنامه زیر مشخص می کند که آیا بیتهای 0 و 2 و 5 و 7 ثبات AL برابر با یک است یا خیر؟ در صورت یک بودن پرش انجام می شود.

TEST AL,10100101B
JZ NEXT

مثال۲: در دستور روبرو اگر مقدار CX برابر با صفر باشد پرش انجام می شود.

TEST CX,0FFH
JZ NEXT