

פעולות כפל וחילוק

מטרה 1:

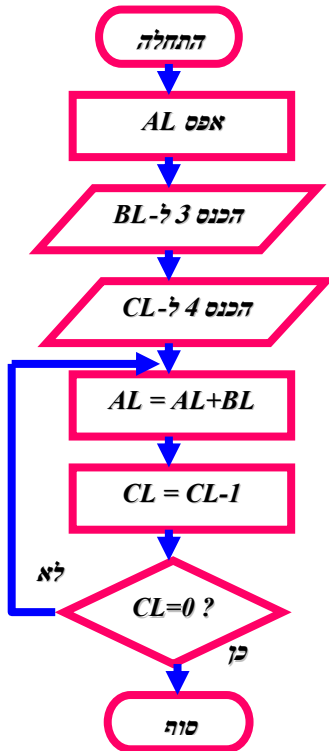
– לבצע כפל בין 2 נתונים על ידי פעולות בסיסיות של חיבור, חיסור והשוואות בלבד.

הדרג:

1. אפס תא סיכומים AL
2. הכנס נתון ראשון ב- BL
3. הכנס נתון שני במונה CL
4. סכם AL עם BL לתוך תא סיכומים AL
5. הקטן באחד המונה CL
6. האם המונה שווה לאפס?
-אם לא אז חזור ל- סעיף 4
-אחרת עבור לפקודה הבאה.
7. סיים את התוכנית.

```

0000 Mov AL , 0
0001 Mov BL , 3
0002 Mov CL , 4
0003 Add AL ,
0004 Sub CL , 1
0005 JNZ 0003
0006 Nop
    
```



מטרה 2:

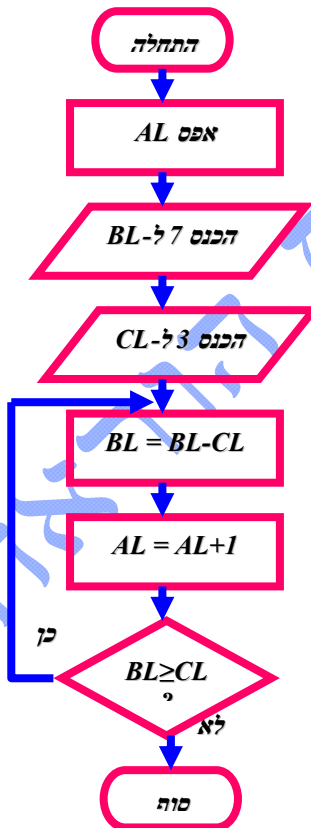
– לבצע חילוק בין 2 נתונים על ידי פעולות בסיסיות של חיבור, חיסור והשוואות בלבד.

הדרג:

1. אפס תא תשובה AL
2. הכנס נתון ראשון ב- BL
3. הכנס נתון שני ב- CL
4. החסר CL מ- BL
5. קדם תא תשובה AL
6. השווה בין BL לבין CL
7. האם ניתן לבצע חיסור נוסף?
אם $BL \geq CL$ אז חזור ל- סעיף 4
אחרת עבור לפקודה הבאה.
8. סיים את התוכנית.

```

0000 Mov AL , 0
0001 Mov BL , 7
0002 Mov CL , 3
0003 Sub BL ,
0004 Add AL , 1
0005 Cmp BL ,
0006 JNB 0003
0007 Nop
    
```

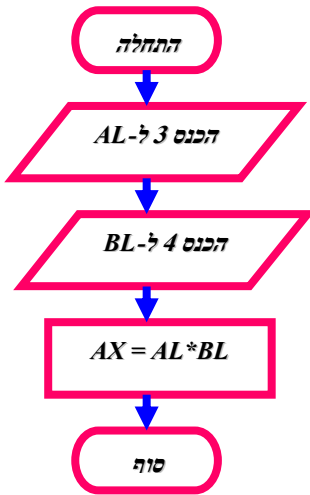


$ALU = BL - CL$
 (BL, CL no changes)
 If $BL < CL$
 Neg flag = "1"
 Else Neg flag = "0"

JNB=Jump if Not Below
 (Test Neg flag)

מטרה 3 :

– לבצע כפל בין 2 נתונים על ידי פעולת כפל בלבד.



```
0000 Mov AL , 3
0001 Mov BL , 4
0002 Mul BL
0003 Nop
```

הדר:

1. הכנס נתון ראשון ב- AL בלבד
2. הכנס נתון שני במונה BL
3. בצע פעולת כפל (MUL)
4. התשובה באוגר AX בלבד.
5. סיים את התוכנית.

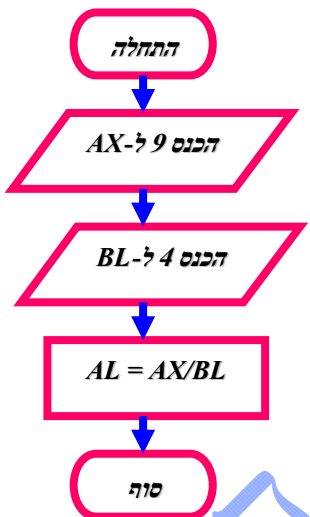
סיכום:

אם הנתונים הם של 8 סיביות:
 - נתון אחד חייב להיכנס בתוך AL, השני בכל אוגר או תא זיכרון של 8 סיביות.
 - תוצאת הכפל תהיה בעלת 16 סיביות בתוך AX.
 - AH=High 8 bit – AL=Low 8 bit , יש לזכור ש-D0 הוא LSB ו-D15 הוא MSB
 דוגמה: $35 = 5 * 7$ - $0023H = 05H * 07H$
 $0000\ 0000\ 0010\ 0011B = 0000\ 0101B * 0000\ 0111B$

אם הנתונים הם של 16 סיביות:
 - נתון אחד חייב להיכנס בתוך AX, השני בכל אוגר או תא זיכרון של 16 סיביות.
 - תוצאת הכפל תהיה בעלת 32 סיביות בתוך צמד האוגרים המורכב מצמד האוגרים DX:AX.
 - DX=High 16 bit – AX=Low 16 bit , יש לזכור ש-D0 הוא (Low) LSB ו-D31 הוא (High) MSB.

מטרה 4 :

– לבצע חילוק בין 2 נתונים על ידי פעולת חילוק בלבד.



```
0000 Mov AX , 9
0001 Mov BL , 4
0002 Div BL
0003 Nop
```

1. הכנס נתון המונה ב- Ax בלבד
2. הכנס נתון שני המחלק (מכנה) ב-BL
3. בצע פעולת החילוק (DIV)
4. התשובה באוגר AL בלבד.
5. השארית באוגר AH בלבד.
6. סיים את התוכנית.

בדוגמה, בסיום החילוק AH=1 , AL=2

סיכום:

אם הנתונים הם של 8 סיביות:
 - נתון המונה חייב להיכנס בתוך AX, המכנה בכל אוגר או תא זיכרון של 8 סיביות.
 - תוצאת החילוק תהיה בעלת 8 סיביות בתוך AL והשארית בתוך AH.

אם הנתונים הם של 16 סיביות:
 - נתון המונה חייב להיכנס בתוך זוג האוגרים DX:AX, המכנה בכל אוגר או תא זיכרון של 16 סיביות.
 - תוצאת החילוק תהיה בעלת 16 סיביות בתוך האוגר AX והשארית ב-DX.

דוגמה:

```
MOV AX, 203 ; AX = 00CBh
MOV BL, 4
DIV BL ; AL = 50 (32h), AH = 3
RET
```

פקודת OR

חיבור לוגי בין כל הסיביות של שני האופרנדים, התשובה נשמרת באופרנד הראשון.
הכללים הם:

1 OR 1 = 1
1 OR 0 = 1
0 OR 1 = 1
0 OR 0 = 0

אנו מקבלים "1" כאשר לפחות סיבית אחת הוא "1".

דוגמה:

```
MOV AL, 'A' ; AL = 01000001b
OR AL, 00100000b ; AL = 01100001b ('a')
RET
```

פקודת AND

כפל לוגי בין כל הסיביות של שני האופרנדים, התשובה נשמרת באופרנד הראשון.
הכללים הם:

1 AND 1 = 1
1 AND 0 = 0
0 AND 1 = 0
0 AND 0 = 0

אנחנו מקבלים "1" כאשר שתי הסיביות הן "1".

דוגמה:

```
MOV AL, 'a' ; AL = 01100001b
AND AL, 11011111b ; AL = 01000001b ('A')
RET
```

פקודת XOR

חיבור לוגי מיוחד בין כל הסיביות של שני האופרנדים, התשובה נשמרת באופרנד הראשון.
הכללים הם:

1 XOR 1 = 0
1 XOR 0 = 1
0 XOR 1 = 1
0 XOR 0 = 0

אנחנו מקבלים "1" כאשר שתי הסיביות שונות ביניהם.

דוגמה:

```
MOV AL, 00000111b
XOR AL, 00000101b ; AL = 0000101b
RET
```

פקודת NOT

הפוך כל סיבית של האופרנד. זה עובד על פי האלגוריתם הבא:
if bit is 1 turn it to 0.
if bit is 0 turn it to 1.

דוגמה:

```
MOV AL, 00011011b
NOT AL ; AL = 11100100b
RET
```

