

## دروس لغة الأسمبلي التابعة لموقع الفريق العربي للبرمجة <http://www.arabteam2000.com>

جميع الحقوق محفوظة للفريق العربي للبرمجة

يمنع منعاً باتاً مسح عنوان الموقع أو اسم المؤلف من هذه الدروس إلا بإذن صريح من إدارة موقع الفريق العربي للبرمجة

ملاحظة:

هذه الدروس هي بالأساس مقتبسة من منهاج السنة الثانية قسم هندسة الحاسبات بجامعة حلب

## الجزء الثاني تعليمات المعالج 8086

## مقدمة في لغة الأسمبلي

هذه اللغة مزودة لوصف كل من العمليات الأساسية التي يمكن إنجازها بواسطة المعالج المصغر، تُكتب تعليمات هذه اللغة باستعمال الرموز الهجائية أو ما يُدعى ALPHANUMERIC بدلاً من الأصفار والواحدات في شيفرة الآلة للمعالج. إن الصيغة العامة لكتابة الأمر (التعليمة) في لغة الأسمبلي هي:

تعليق ; تعليمة : لافتة

عادة فإن التعليقات أو الملاحظات التي تصف الأوامر توضع على الطرف الأيمن. وهذا النوع من التوثيق بين التعليمة و التعليق يجعل من السهل على المبرمج كتابة و قراءة و تصحيح الشيفرة. و نقصد بكلمة الشيفرة أن البرنامج مكتوب بلغة الآلة للمعالج و الذي يُعرف بشيفرة الهدف object code أما البرنامج المكتوب بلغة الأسمبلي فيدعى بشيفرة المصدر source code . هذا و إن كل تعليمة في برنامج المصدر تطابق أمراً واحداً في لغة الأسمبلي حيث أن الأمر يجب أن يحدّد أي عملية سيتم تنفيذها و ما هي متحولات المعطيات التي ستُعالج. لهذا السبب تُقسم التعليمة إلى قسمين منفصلين : رمز التعليمة opcode = operation و المتحولات operands . رمز العملية هو جزء من التعليمة و الذي يحدد العملية التي ستُنفذ فمثلاً نذكر بعض العمليات النموذجية كالجمع و الطرح و النقل.

في لغة الأسمبلي تستخدم الكلمات المختزلة mnemonic من أجل التعليمات فمثلاً بالنسبة للمعالج 8086 فالكلمات المختزلة في لغة الأسمبلي لعمليات الجمع و الطرح و النقل هي على الترتيب ADD و SUB و MOV . أما المتحولات فتحدد المعطيات التي ستُعالج من قبل المعالج بواسطة رمز العملية للتعليمة فمثلاً في التعليمة التي تضيف محتويات مسجل القاعدة إلى محتويات المراكم فإن AX و BX هي المتحولات و تُكتب التعليمة على الشكل التالي ADD AX,BX ففي هذا المثال تُضاف محتويات BX إلى AX و يوضع ناتج الجمع في AX و لذلك يُعتبر BX متحول المصدر و AX متحول الهدف.

## طاقم تعليمات المعالج 8086

يُزود المعالج 8086 بمجموعة تعليمات مؤلفة من 117 تعليمة أساسية و كذلك إن المجال الواسع للمتحويلات و أنظمة العنونة المسموحة للاستعمال مع هذه التعليمات يوسع مجموعة التعليمات إلى تعليمات أكثر، فمثلاً تعليمة Mov الأساسية تمتد إلى 28 تعليمة مختلفة و قابلة للتنفيذ على مستوى لغة الآلة.

### أولاً - تعليمات نقل المعطيات

يملك المعالج مجموعة تعليمات وظيفتها نقل المعطيات و ذلك إما بين مسجلات المعالج الداخلية أو بين مسجل داخلي و حجرة تخزين في الذاكرة و هي:

#### (1) تعليمة Mov

تستخدم هذه التعليمة لنقل بايت أو كلمة معطيات من متحول المصدر إلى متحول الهدف و لها الشكل التالي:

الأعلام المتأثرة	العملية	الصيغة	المعنى	الكلمة المختزلة
لا يوجد	S → D	MOV D,S	نقل	MOV

إن S,D لهذه التعليمة يمكن أن تكون مسجلات داخلية أو حجرات تخزين في الذاكرة و بين الجدول التالي مختلف أنواع متحويلات المصدر و الهدف مع مثال لكل منها :

المصدر Source	الهدف Destination
Acc	Mem
Mem	Acc
Reg	Reg
Mem	Reg
Reg	Mem
Reg	Imm
Imm	Reg
Imm	Mem
Reg16	Seg-reg
Mem16	Seg-reg
Seg-reg	Reg16
Seg-reg	Mem16

الرمز	المعنى
Acc	المراكم
Mem	حجرة ذاكرة
Reg	مسجل
Imm	متحول فوري
Seg-reg	متحول مقطع
Reg16	مسجل ذو 16 بت
Mem16	حجرتي ذاكرة

#### الحالات المستثناة من تعليمة MOV

- 1- لا تستطيع تعليمة MOV أن تنقل المعطيات بشكل مباشر بين حجرتي ذاكرة لذلك لا نرى في الجدول المجاور الحالة التالية :  
Mem → Mem و لحل هذه المشكلة فإن المعطيات المرغوب بنقلها يجب نقلها أولاً في مسجل داخلي بواسطة تعليمة MOV ، و من ثم تنقل محتويات هذا المسجل إلى حجرة جديدة في الذاكرة بواسطة تعليمة MOV أخرى.
- 2- لا يمكن وضع قيمة فورية في مسجل مقطع مباشرة. أي أن التعليمة التالية غير مسموح بها MOV DS,1000 و لحل هذا المشكلة نستخدم التعليمتين التاليتين :

```
MOV AX,1000
MOV DS,AX
```

- 3- لا يمكن نقل محتويات أحد مسجلات المقاطع إلى مسجل مقطع آخر مباشرة، أي أن التعليمة التالية غير مسموح بها  
MOV DS,ES و لحل هذه المشكلة نقوم بـ

```
MOV AX,ES
```

## MOV DS,AX

مثال عام : MOV AL,[SI] هذه التعليمة تعني نقل محتويات حجرة الذاكرة المشار إليها بواسطة المسجل SI إلى المسجل AL و إن نظام العنونة في هذه التعليمة هو عنونة غير مباشرة بالمسجل و متحول المصدر عنوانه الفيزيائي هو  $PA = DS \times 10h + SI$  أما متحول الهدف فهو AL .

## (2) تعليمة التبديل XCHG

تستخدم هذه التعليمة لاستبدال متحول المصدر بمتحول الهدف و لاستبدال متحول الهدف بمتحول المصدر.

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
XCHG	تبديل	XCHG D,S	S → D D → S	لا يوجد

المصدر S	الهدف D
Reg16	Acc
Reg	Mem
Reg	Reg

و يبين الجدول التالي مختلف أنواع متحويلات المصدر و الهدف لتعليمة XCHG .  
مثال:

## XCHG AX,BX

في هذا المثال يتم التبديل بين محتويات AX و BX .

## XCHG [SUM],BX

يتم التبديل بين محتوى الحجرة SUM في الذاكرة و بين المسجل BX .

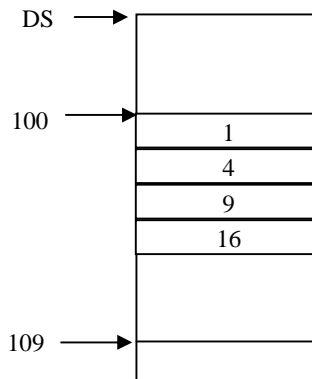
## (3) تعليمة XLAT

تعامل هذه التعليمة مع AL فقط ، إن تعامل هذه التعليمة يتم مع الجداول المخزنة في الذاكرة فلو وضعنا في BX إزاحة بداية الجدول نسبة إلى مقطع المعطيات DS و وضعنا في AL إزاحة العنصر نسبه إلى بداية الجدول، عندها تقوم تعليمة XLAT بجمع محتويات المسجل AL مع محتويات المسجل BX و تعتبر الناتج إزاحة بالنسبة إلى مقطع المعطيات، ثم تقوم بوضع قيمة الحجرة المعطى إزاحتها في AL .

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
XLAT	ترجمة	جدول المصدر XLAT	$(DS \times 10h) + [AL+BX]$ → AL	لا يوجد

مثال:

بفرض أنه لدينا جدول في الذاكرة يحتوي على مربعات الأعداد من 1 إلى 9 أي أنه في أول حجرة من الجدول يوضع مربع العدد 1 و في الحجرة الثانية يوضع مربع العدد 2 (أي 4) ، و هكذا ... ومن هذا نرى أن الجدول طوله تسع بايتات إزاحة بدايته عن بداية مقطع الـ DS هي 100 .



عندما يطلب منا الحصول على مربع أحد هذه الأعداد و ليكن العدد 4 أي أن المطلوب هو أن تصبح قيمة AL = 16 لذلك نقوم بما يلي:

(1) نضع AL = 3 و BX = 100 .

(2) نعطي التعليمة XLAT .

و بعد تنفيذها يصبح AL = 16 و هو المطلوب.

#### 4) التعليمات LEA, LES, LDS

تستعمل هذه التعليمات من أجل عملية نقل المعطيات لتحميل مسجل مقطوع أو مسجل أغراض عامة بعنوان بشكل مباشر من الذاكرة. التعليمات LEA وظيفتها هي تحميل مسجل بعنوان فعال أما LDS فهي لتحميل مسجل ما و مسجل مقطوع المعطيات DS و تعليمات LES وظيفتها تحميل مسجل ما و مسجل مقطوع المعطيات الإضافي ES . وهذه التعليمات موصوفة كما في الجدول التالي:

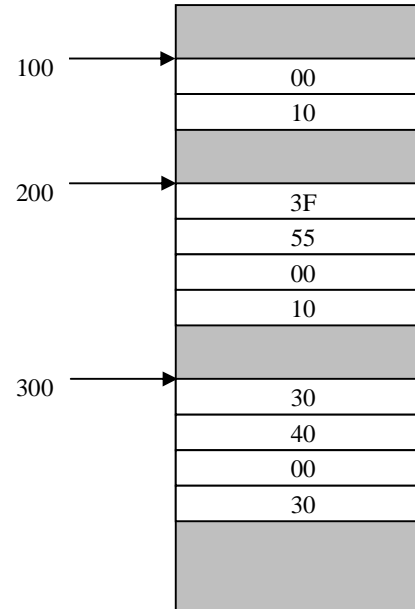
الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
LEA	تحميل عنوان فعال	LEA reg16,mem16	Mem16 → reg16	لا يوجد
LDS	تحميل مسجل و المسجل DS	LDS reg16,mem32	Mem32 → reg16 Mem32+2 → DS	لا يوجد
LES	تحميل مسجل و المسجل ES	LES reg16,mem32	Mem32 → reg16 Mem32+2 → ES	لا يوجد

أمثلة:

LEA SI,[100] => SI = 1000

LDS SI,[200] => SI = 553F  
DS = 1000

LES DI,[300] => DI = 4030  
ES = 3000



#### ثانياً - التعليمات الرياضية

و هي تشمل تعليمات من أجل عمليات الجمع، الطرح، الضرب و القسمة.

#### 1) تعليمات الجمع

و هي موصوفة بالجدول التالي:

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
ADD	جمع	ADD D,S	S + D → D Carry → CF	أعلام الحالة

ADC	جمع مع أخذ الانزياح بعين الاعتبار	ADC D,S	$S + D + CF \rightarrow D$ $Carry \rightarrow CF$	أعلام الحالة
INC	الزيادة بمقدار واحد	INC D	$D+1 \rightarrow D$	أعلام الحالة
AAA	تصحيح ناتج جمع عددين بشيفرة الأسكي	AAA	سيتم شرحها لاحقاً	AF,CF
DAA	تصحيح ناتج جمع عددين بشيفرة BCD	DAA	سيتم شرحها لاحقاً	كل أعلام الحالة ماعدا OF

### تعليمتي ADD و ADC

إن المتحولات المسموحة في تعليمات الجمع ADD, ADC مبينة في الجدول التالي:

المصدر S	الهدف D
Reg	Reg
Mem	Reg
Reg	Mem
Imm	Reg
Imm	Mem
Imm	Acc

الهدف D
Reg8
Reg16
Mem

و بالنسبة للمتحويلات المسموحة في تعليمة INC فهي :

مثال: بفرض  $AX = 4F3Dh$  و  $BX = FD81h$  و  $CF = 1$  فما هي نتيجة تنفيذ التعليمة ADC AX,BX ؟ مبيناً حالة أعلام الحالة بعد تنفيذ عملية الجمع هذه .

الحل: سنكتب الشيفرة الثنائية للمتحويلات من أجل توضيح حالة الأعلام

$$\begin{array}{r}
 \begin{array}{cccccccc}
 & 1 & 1 & 1 & 1 & & & \\
 & & & & & & & \\
 AX = & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & b \\
 BX = & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & b \\
 CF = & & & & & & & & & & & 0 & 0 & 0 & 1 & b & + \\
 \hline
 & 1 & & & & & & & & & & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & b
 \end{array}
 \end{array}$$

CF

و الآن أعلام الحالة هي:

$PF = 0$  لأن عدد الواحدات فردي في البايث الأول من ناتج الجمع

$AF = 0$  لأنه لا يوجد انزياح من الخانة 3 إلى الخانة 4 في البايث الأول من ناتج الجمع ( حيث يتم ترقيم الخانات بدءاً من الصفر )

$SF = 0$  و هي آخر خانة من نتيجة الجمع ( الناتج موجب )

$CF = 1$  بسبب وجود انزياح خارجي

OF = 0 لأنه يوجد إنزياح داخلي و إنزياح خارجي

ملاحظة: الانزياح الداخلي هو الداخل إلى الخانة ذات الأهمية العظمى MSB

ملاحظة: OF = 1 إذا وجد انزياح داخلي فقط أو وجد انزياح خارجي فقط

### تعليمة التصحيح DAA

تستخدم هذه التعليمة لإنجاز عملية تصحيح لنتائج جمع عددين بشيفرة BCD ( هذا و يجب أن يكون ناتج الجمع حتماً في AL أي في النصف السفلي من المراكم AX ) و الجدول التالي يبين الحالات الممكنة لجمع عددين بشيفرة BCD :

+	0	1	2	3	4	5	6	7	8	9
0										9
1									9	10
2								9	10	
3							9	10		
4						9	10			
5					9	10				
6				9	10					15
7			9	10					15	16
8		9	10					15	16	
9	9	10					15	16		18

المنطقة الأولى أرقامها من 0 إلى 9 و فيها تكون نتيجة الجمع صحيحة و لا تحتوي على انزياح و ليست بحاجة إلى تصحيح مثلاً  $7+2=9$  و هي أرقام واقعة ضمن نطاق المنطقة الأولى.

المنطقة الثانية أرقامها من 10 إلى 15 و فيها تكون نتيجة الجمع غير صحيحة و بحاجة إلى تصحيح بإضافة العدد 6 فنحصل على رقم و حمل إلى العدد الثاني مثلاً  $9+5=E$  بإضافة 6 إلى العدد E يكون الناتج  $6+E=14$  و بذلك تكون النتيجة صحيحة.

المنطقة الثالثة أرقامها من 16 و حتى 18 و فيها تكون نتيجة الجمع غير صحيحة و بحاجة إلى تصحيح و هنا تتكون النتيجة من حاصل جمع مع انزياح.

بما أن ناتج الجمع موجود في AL حيث تمثل Bit7 ... Bit0

إن قاعدة التصحيح في هذه التعليمة هي :

- 1) if Bit3 Bit2 Bit1 Bit0 of AL > 9 or AF = 1  
then AL = AL + 6 , AF = 1
- 2) if AL > 9Fh or CF = 1  
then AL = AL + 60h , CF = 1

مثال: بفرض أن BCD AL = 28 و BCD BL = 68

ما هو ناتج تنفيذ ما يلي:

ADD AL,BL

DAA

$$\begin{array}{r}
 \downarrow \\
 1 \\
 28 \text{ BCD} = 0010 \ 1000 \text{ b} \\
 68 \text{ BCD} = 0110 \ 1000 \text{ b} \quad + \\
 \hline
 1001 \ 0000 \rightarrow \text{AL} \\
 \text{CF} = 0 \quad \quad \quad 0110 \quad + \\
 \hline
 1001 \ 0110 \Rightarrow \text{AL} = 96 \text{ BCD} \\
 \text{AF} = 1
 \end{array}$$

الحل: إن نتيجة تنفيذ هاتين التعليمتين هي

حيث 0110 تمثل الرقم ستة

### تعليمية AAA

تستخدم هذه التعليمية لتصحيح ناتج جمع عددين بشيفرة آسكي ( و هنا أيضاً يجب أن يكون ناتج الجمع في المسجل AL ) و قاعدة التصحيح في هذه التعليمية هي:

if Bit3 Bit2 Bit1 Bit0 of AL > 9 or AF = 1  
 then AL = AL + 06  
     AL = AL and 0Fh  
     AH = AH + 1  
     AF = 1  
     CF = 1  
 Else AL = AL and 0Fh  
     AH = 00

مثال: يفرض أن AL = 32h = 2 ASCII و BL = 34h = 4 ASCII ما هو ناتج تنفيذ التعليمتين التاليتين:

ADD AL,BL

AAA

$$\begin{array}{r}
 \downarrow \times \\
 \text{AL} = 0011 \ 0010 \\
 \text{BL} = 0011 \ 0100 \quad + \\
 \hline
 0110 \ 0110 \rightarrow \text{AL} = 66\text{h} \\
 \text{AL} = 06\text{h} , \text{AH} = 00
 \end{array}$$

الحل: إن ناتج تنفيذ هاتين التعليمتين هو كالتالي :

و هنا AF = 0 بسبب عدم وجود انزياح من الخانة 3 إلى الخانة 4 (حيث يبدأ الترقيم اعتباراً من الصفر)

### (2) تعليمات الطرح

هناك مجموعة واسعة من تعليمات الطرح كما هو واضح من الجدول التالي:

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
SUB	طرح	SUB D,S	D - S → D borrow → CF	أعلام الحالة
SBB	الطرح مع الاستعارة	SBB D,S	D - S - CF → D Carry → CF	أعلام الحالة
DEC	الإنقاص بمقدار واحد	DEC D	D-1 → D	أعلام الحالة



NEG	المتمم الثنائي	NEG D	0 - D → D 1 → CF	أعلام الحالة
DAS	تصحيح ناتج طرح عددين بشيفرة BCD	DAS	سيتم شرحها لاحقاً	كل أعلام الحالة عدا OF
AAS	تصحيح ناتج جمع عددين بشيفرة الأسكي	AAS	سيتم شرحها لاحقاً	AF, CF

ملاحظة: إن المتحولات المستخدمة من أجل تعليمي الطرح SBB, SUB هي نفسها المتحولات المسموحة من أجل تعليمي الجمع ADD, ADC أما بالنسبة إلى المتحولات المستخدمة من أجل تعليمة DEC فهي نفسها المتحولات المسموحة من أجل تعليمة INC و بالنسبة لتعليمة NEG فالمتحولات المسموحة هي Reg, Reg16, Mem, Mem16 .  
مثال: بفرض أن SI = 0018h و DS = 2F00h و العنوان الفيزيائي المتولد عنهما هو 2F018h و بفرض كانت محتويات الحجرة التي يشير إليها العنوان الفيزيائي [2F018] = 0400h ، ما هو ناتج تنفيذ التعليمة SUB [SI], 03F8h .  
الحل:

تقوم هذه التعليمة بطرح محتويات متحول المصدر (متحول فوري هنا) من محتويات متحول الهدف (محتويات حجرة ذاكرة هنا) حيث أن تعليمة الطرح تتم بإيجاد المتمم الثنائي لمتحول المصدر و من ثم جمعه مع متحول الهدف.



$$\begin{array}{r} \text{Destination} = 0400h = 0000\ 0100\ 0000\ 0000\ b \\ \text{Source} = 03F8h = 1111\ 1100\ 0000\ 1000\ b \\ \hline \square\ 0000\ 0000\ 0000\ 1000\ b \end{array}$$

1

تذكرة بالمتمم الثنائي ( و الذي يشار إليه بوضع خطين فوق العدد الذي نريد إيجاد المتمم الثنائي له ) :  
إذا أردت الحصول على المتمم الثنائي للعدد 03F8h فاعمل ما يلي:  
1) تحويل هذا العدد إلى النظام الثنائي فيصبح 0000 0011 1111 1000  
2) أقلب الأصفار واحدات و الواحدات أصفاراً فينتج 1111 1100 0000 0111  
3) أضف واحد إلى الرقم الناتج فتحصل على المتمم الثنائي 1111 1100 0000 1000 - 03F8h

لاحظ أن : PF = 0 لأن عدد الواحدات فردي في البايث الأول من الناتج .

AF = 1 لأنه لا يوجد معنا حمل ( انزياح ) عند الانتقال من الخانة الثالثة إلى الخانة الرابعة (عكس حالة الجمع).

ZF = 0 لأن النتيجة ليست صفرية.

SF = 0 و هي قيمة آخر خانة من الناتج MSB.

CF = 0 لأن هناك انزياح خارجي (عكس حالة الجمع).

OF = 0 لحصول انزياح داخلي و انزياح خارجي بأن واحد.

تعليمة DAS

تستخدم هذه التعليمات لتصحيح ناتج طرح عددين بشيفرة BCD حيث يكمن ناتج طرح هذين العددين في المسجل AL و قاعدة التصحيح هي :

- 1) if Bit3 Bit2 Bit1 Bit0 of AL > 9 or AF = 1  
then AL = AL - 06 , AF = 1
- 2) if AL > 9Fh or CF=1  
then AL = AL - 60h , CF = 1

مثال: بفرض أن AL = 86 BCD و AH = 07 BCD ، بين نتيجة التعليمتين التاليتين:

SUB AL,AH  
DAS

الحل:

$$\begin{array}{r}
 \text{AL} = 1000\ 0110\ \text{b} \\
 \text{AH} = 1111\ 1001\ \text{b} \quad + \\
 \hline
 \boxed{1}0111\ 1111\ \text{b} \Rightarrow \text{AL} = 7\text{Fh}
 \end{array}$$

و الآن :

AF = 1 بسبب عدم وجود انزياح من الخانة الثالثة إلى الخانة الرابعة.

CF = 0 لوجود انزياح خارجي

و بتطبيق الشرط 1 من قاعدة التصحيح نجد أن AF = 1 , AL = 79h

**تعليمات AAS**

تستخدم هذه التعليمات لتصحيح ناتج طرح عددين بالشيفرة ASCII حيث يكمن ناتج الطرح في AL ، و قاعدة التصحيح هي:

- if Bit3 Bit2 Bit1 Bit0 of AL > 9 or AF = 1  
then AL = AL - 06h , AL = AL and 0Fh  
AH = AH - 01 , AF = 1 , CF = 1
- Else AL = AL and 0Fh , AH = 00

مثال:

بفرض أن AL = 38h = 8 ASCII و BL = 35h = 5 ASCII ، ما هو ناتج تنفيذ التعليمتين التاليتين:

SUB AL,BL  
AAS

الحل:

$$\begin{array}{r}
 \text{AL} = 0011\ 1000\ \text{b} \\
 \text{BL} = 1100\ 1011\ \text{b} \quad + \\
 \hline
 \boxed{1}0000\ 0011\ \text{b} \Rightarrow \text{AL} = 03\text{h}
 \end{array}$$

AF = 0 بسبب وجود انزياح من الخانة الثالثة إلى الخانة الرابعة

CF = 0 بسبب وجود انزياح خارجي

و بعد تطبيق قاعدة التصحيح نجد AH = 00 , AL = 03h

### 3) تعليمات الضرب و القسمة

يتم تطبيق هذه التعليمات على الأعداد الثنائية أو بالشيفرة BCD أي في معالجة الأعداد ذات الإشارة و الأعداد بدون إشارة. و

هذه التعليمات مبينة في الجدول التالي:

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
MUL	ضرب بدن إشارة	MUL S	AL.S8 → AX AX.S16 → DX,AX	أعلام الحالة
DIV	تقسيم بدون إشارة	DIV S	Q[AX/S8] → AL R[AX/S8] → AH Q[(DX,AX)/S16] → AX R[(DX,AX)/S16] → DX	أعلام الحالة

ببساطة : النقطة تعني عملية الضرب العادية، و الرمز S8 يعني متحول مصدر عبارة عن بايت أما الرمز R فيعني باقي القسمة و الرمز Q ما هو إلا حاصل قسمة.

ملاحظة: إذا كانت قيمة Q في الحالة الأولى ( حالة بايت ) مساوية لـ FF أو كانت قيمة Q في الحالة الثانية ( حالة كلمة ) مساوية إلى FFFFh فتحدث مقاطعة من النوع صفر، و تُعرف هذه المقاطعة بخطأ التقسيم.

ملاحظة: بالنسبة لتعليمات الضرب و التقسيم للأعداد ذات الإشارة فهي مشابهة تماماً للتعليمات السابقة و تُعرف كما يلي: IMUL هي تعليمة الضرب مع أخذ الإشارة بعين الاعتبار.

IDIV هي تعليمة التقسيم مع أخذ الإشارة بعين الاعتبار.

و تكون إشارة الناتج في كلتا التعليمتين آخر خانة منه أي خانة الـ MSB .

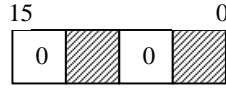
و بالإضافة إلى ذلك هناك التعليمات التالية (تابع لجدول الضرب و القسمة):

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
AAM	تصحيح الناتج في AL من ضرب عددين BCD أو عددين ثنائيين	AAM	Q[AL/10d] → AH R[AL/10d] → AL	أعلام الحالة
AAD	تصحيح AX من أجل القسمة حيث AX ليس ناتج القسمة و إنما هو متحول الهدف في عملية القسمة. لذلك نطبق هذه التعليمة قبل تعليمة القسمة على عكس باقي تعليمات التصحيح	AAD	AH.10d + AL → AL 00 → AH	SF, ZF, PF
CBW	تحويل بايت إلى كلمة	CBW	MSB of AL → All bits of AH	لا يوجد
CWD	تحويل كلمة إلى كلمة مضاعفة	CBW	MSB of AX → All bits of DX	لا يوجد

إن المتحولات المسموحة في تعليمات الضرب و القسمة هي بالنسبة للمصدر S :

Mem16, Mem8, Reg16, Reg8 و بالنسبة إلى للهدف D فالمتحول الوحيد المسموح هو المراكم دوماً.

ملاحظة: إن تعليمات القسمة يمكن استخدامها لتقسيم المقسوم بـ 8 بتات في AL على مقسوم عليه بـ 8 بتات أيضاً. و لإنجاز هذا يجب أولاً تمديد إشارة المقسوم لملء المسجل AX و هذا يعني ملء AH بأصفار إذا كان العدد موجباً أو بواحدات إذا كان العدد سالباً (أي حسب خانة الإشارة) و تتم هذه العملية بواسطة التعليمة CBW. و بشكل مشابه فإن تعليمات التقسيم 32 بت على 16 بت يمكن استخدامها لتقسيم مقسوم ذي 16 بت في AX على مقسوم عليه ذي 16 بت و ذلك بتحويل الكلمة إلى كلمة مضاعفة و يتم هذا بواسطة التعليمة CWD.



كما ذكرنا سابقاً فإن الأعداد غير المجمعة يتم حفظها كالتالي:  
القسم العلوي من البايث الذي يحتوي على العدد غير المجموع يجب أن تكون قيمته مساوية إلى الصفر.

إن التعليمة AAM تستخدم لتصحيح ناتج ضرب عددين غير مجموعين لأنه عند ضرب عددين غير مجموعين نحصل على نتيجة مجمعة و النتيجة يجب أن تكون غير مجمعة، لذلك نصححها بواسطة التعليمة AAM. مثال: بفرض أن  $BL = 09$  و  $AL = 07$  فما هي نتيجة تنفيذ التعليمات التالية:

MUL BL  
AAM

الحل:

```

AX = 00 07
BX = 00 09
-----
MUL  00 3F  AX
AAM  06 03  AX
    
```

قاعدة التصحيح في تعليمة AAD هي :

إن التقسيم بالنسبة إلى الأعداد غير المجمعة يؤدي إلى الحصول على نتائج خاطئة و لذلك يجب تجميع الأعداد قبل قسمتها. و بفرض أن  $AX = 0604h$  ( و هي أعداد غير مجمعة ) فنتيجة تطبيق تعليمة التصحيح AAD ( و التي يتم تطبيقها قبل عملية التقسيم ) هي:

$$\left. \begin{array}{l} AL = 06 \times 10d + 04h = 64d = 40h \\ AH = 00h \end{array} \right\} \Rightarrow AX = 0040h$$

### ثالثاً - التعليمات المنطقية

تنجز عملياتها المنطقية خانة بخانة على متحولاتها. و الجدول التالي يبين التعليمات المنطقية:

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
AND	المنطقي AND	AND D,S	$S \cdot D \rightarrow D$	أعلام الحالة
OR	المنطقي OR	OR D,S	$S + D \rightarrow D$	أعلام الحالة
XOR	المنطقي XOR	XOR D,S	$S \oplus D \rightarrow D$	أعلام الحالة
NOT	المنطقي NOT	NOT D	$\overline{D} \rightarrow D$	لا يوجد

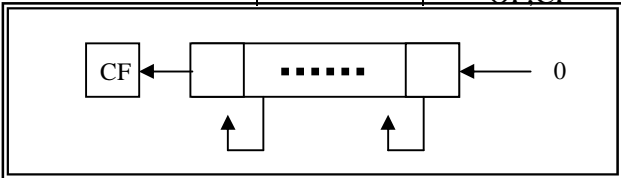
إن المتحولات المسموحة من أجل تعليمة AND, OR, XOR مبيّنة في الجدول جانباً:

D	S
Reg	Reg
Reg	Mem
Mem	Reg
Reg	Imm
Mem	Imm
AX	Imm

### رابعاً - تعليمات الإزاحة

هناك نوعان من تعليمات الإزاحة هما الإزاحة المنطقية و الإزاحة الرياضية كما هو واضح في الجدول التالي:

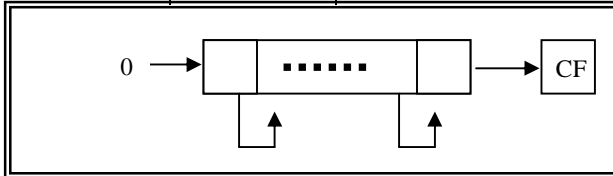
الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
SAL/SHL	إزاحة رياضية/إزاحة منطقية و كلاهما نحو اليسار	SAL/SHL D,count		OF,CF



العملية هنا هي إزاحة محتويات D نحو اليسار باتجاه CF عدداً من الخانات مساوياً لقيمة count و ملء جميع الخانات اليمنى المفرغة بأصفار.

و بالنسبة لتأثير هذه التعليمات على علم OF : إذا تبذلت خانة الإشارة نتيجة الإزاحة فإن  $OF = 1$ .

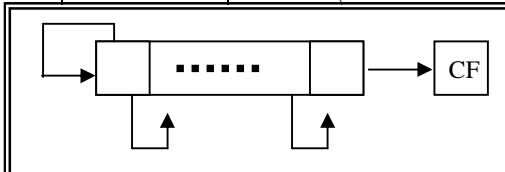
الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
SHR	إزاحة منطقية نحو اليمين	SHR D,count		OF,CF



العملية هنا هي إزاحة محتويات D نحو اليمين باتجاه CF عدداً من الخانات مساوياً لقيمة count و ملء جميع الخانات اليسرى المفرغة بأصفار.

و بالنسبة لتأثير هذه التعليمات على العلم OF : إذا تبذلت خانة الإشارة نتيجة الإزاحة فإن  $OF = 1$ .

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
SAR	إزاحة رياضية نحو اليمين	SAR D,count		أعلام الحالة



العملية هنا هي إزاحة محتويات D نحو اليمين باتجاه CF عدداً من المرات مساوياً لقيمة count و ملء الخانات جميع الخانات اليسرى بقيمة الخانة MSB (خانة الإشارة أو آخر خانة).

ملاحظة: بالنسبة للتعليمتين SAL, SHL: إذا طبقنا هاتين التعليمتين من أجل الإزاحة بعدد من الخانات  $count = N$  فهذا يعني ضرب متحول الهدف بـ  $2^N$  والذي هو مضاعفات العدد 2.

ملاحظة: إن التعليم SHR يعني تقسيم متحول الهدف على العدد  $2^{count}$  تحت كون  $LSB = 0$  كل مرة و في حالة  $LSB = 1$  فعندها يكون لدينا باقي موضوع في العلم CF.

مثال: اكتب برنامجاً يقوم بحساب العلاقة الرياضية التالية مستخدماً تعليمات الإزاحة و التعليمات الرياضية:

```

3.(AX) + 7.(BX) → DX
MOV SI,AX      ; copy AX into SI
SAL SI,1       ; 2 AX
ADD SI,AX      ; 3 AX
MOV DX,BX     ; copy BX into DX
MOV CL,03H    ; load shift count
SAL DX,CL     ; 8 BX
SUB DX,BX     ; 7 BX
ADD DX,SI     ; result
    
```

D	Count
Reg	1
Reg	CL
Mem	1
Mem	CL

إن المتحولات المسموحة بالنسبة لتعليمات الإزاحة هي:

أي عندما Count لا يساوي الواحد فعندئذ يجب تحميل قيمة count في المسجل CL ثم كتابة تعليمات الإزاحة أي: عندما count يساوي الواحد فيمكن أن نكتب:

```
SAL AX,1
```

و عندما  $count \neq 1$  يجب أن نكتب:

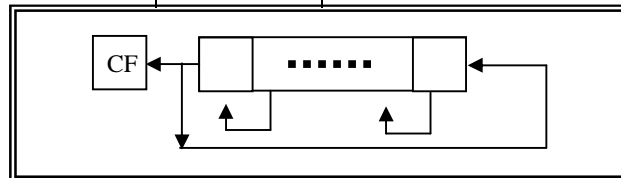
```
MOV CL,count
SAL AX,CL
```

هذا و إن قيمة count محددة بالمجال [1,FF] و الأقواس المحيطية ليس لها علاقة بمفهوم الإزاحة طبعاً.

خامساً - تعليمات التدوير

و هي مبينة في الجدول التالي:

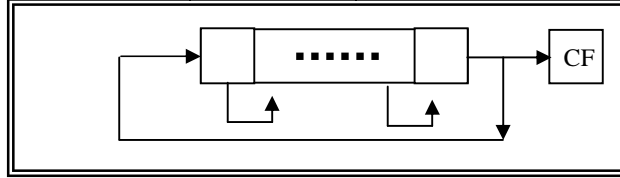
الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
ROL	تدوير نحو اليمين	ROL D,count		OF,CF



العملية هنا هي تدوير محتويات D نحو اليسار عدداً من المرات مساوياً لقيمة count . و كل خانة تُزاح خارج الـ MSB توضع في الخانة LSB و في CF .

و بالنسبة لتأثير هذه التعليمة على العلم OF فهو نفس المناقشة في التعليمات السابقة.

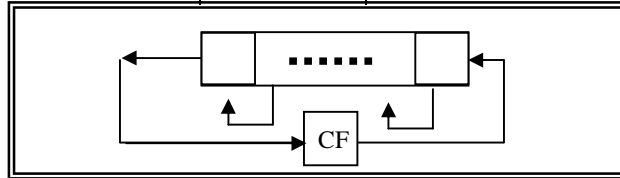
الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
ROR	تدوير نحو اليسار	ROR D,count		OF,CF



العملية هنا هي تدوير محتويات D نحو اليمين عدداً من المرات مساوياً لقيمة count . و كل خانة تُزاح خارج الـ LSB توضع في الخانة MSB و في CF .

و بالنسبة لـ OF فهو نفس المناقشة في التعليمات السابقة.

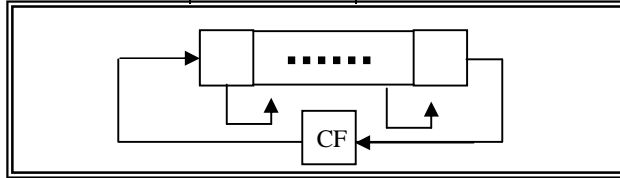
الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
RCL	تدوير نحو اليسار عبر الـ CF	RCL D,count		OF,CF



العملية هنا مشابهة لتعليمة ROL ما عدا أن المحتوى الأصلي لـ CF يوضع في الخانة LSB أما الخانة المزاحة خارج الـ MSB فتوضع في CF .

و بالنسبة لـ OF نفس المناقشة السابقة.

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
RCR	تدوير نحو اليمين عبر الـ CF	RCR D,count		OF,CF



العملية هنا مشابهة لتعليمة ROR ما عدا أن المحتوى الأصلي لـ CF يوضع في الخانة MSB أما الخانة المزاحة خارج الـ LSB فتوضع في CF .

و بالنسبة لـ OF نفس المناقشة السابقة.

سادساً - تعليمات مسجلات الأعلام

و هي مبينة في الجدول التالي :

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
LAHF	تحميل AH من مسجل الأعلام	LAHF	Flags → AH النصف الأول من مسجل الأعلام يوضع في AH	لا يوجد
SAHF	تخزين قيمة AH في مسجل الأعلام	SAHF	AH → Flags يوضع AH في النصف الأول من مسجل الأعلام	أعلام الحالة OF
CLC	تنظيف الـ CF	CLC	0 → CF	CF
STC	توضيع الـ CF	STC	1 → CF	CF
CMC	متمم أحادي لـ CF	CMC	$\overline{CF} \rightarrow CF$	CF
CLI	تنظيف IF	CLI	0 → IF	IF
STI	توضيع الـ IF	STI	1 → IF	IF

### سابعاً - تعليمات المقارنة

تسمح تعليمة المقارنة CMP بمقارنة عددين بـ 8 بت أو 16 بت و هي مشروحة بالجدول التالي:

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
CMP	مقارنة عددين	CMP D,S	D - S تتأثر الأعلام	أعلام الحالة

تجري عملية الطرح ضمناً دون تخزين نتيجتها في متحول الهدف D (أي تبقى كلاً من محتويات المصدر S و محتويات الهدف D على حالها) و تستعمل هذه التعليمة لجعل أعلام الحالة تأخذ قيمة واحد منطقي أو صفر منطقي. إن المتحولات المسموحة لهذه التعليمة مبينة في الجدول التالي:

D	S
Reg	Reg
Reg	Mem
Mem	Reg
Reg	Imm
Mem	Imm
Acc	Imm

### ثامناً - تعليمات القفز

الغاية من تعليمة القفز هي تعديل طريق تنفيذ التعليمات في البرنامج. و هناك نوعان من تعليمات القفز، وهي: القفز المشروط و القفز غير المشروط. في القفز غير المشروط لا يوجد أي شروط من أجل حدوث القفز أما في القفز المشروط فإن الحالات الشرطية الموجودة في لحظة تنفيذ تعليمة القفز تتخذ القرار فيما إذا سيحدث القفز أم لا، ففي حال تحقق الحالات الشرطية فإنه يتم القفز، وإلا يُتابع التنفيذ بالتعليمة التي تلي تعليمة القفز في البرنامج.

#### 1) تعليمة القفز غير المشروط

و هي مشروحة في الجدول التالي:



الأعلام المتأثرة	العملية	الصيغة	المعنى	الكلمة المختزلة
لا يوجد	القفز إلى العنوان المحدد بواسطة المتحول operand	JMP operand	قفز غير مشروط	JMP

هناك نوعان أساسيان من القفز غير المشروط الأول يدعى بالقفز ضمن المقطع الجزئي، و الثاني هو القفز بين المقاطع الجزئية أي يُمكننا من القفز من أحد مقاطع الشيفرة إلى مقطع آخر و إن تحقيق هذا النوع من القفز يتطلب منا تعديل محتويات كل من مقطع الCS و مسجل مؤشر التعليم IP ، أما القفز ضمن المقطع الجزئي فإنه يتطلب منا تعديل قيمة الIP فقط. إن المتحولات المسموحة لتعليمه القفز غير المشروط هي :

Operand	
Short_Label	للقفز ضمن المقطع الجزئي
Near_Label	
Memptr16	
Regptr16	للقفز بين المقاطع الجزئية
Far_Label	
Memptr32	

منحول الالفة القصيرة  
منحول الالفة القريبة  
منحول مؤشر ذاكري 16 بت  
منحول مؤشر مسجلي 16 بت  
منحول الالفة البعيدة  
منحول مؤشر ذاكري 32 بت

### القفز ضمن المقطع الجزئي

أ ) إن متحولات الالفة القصيرة و الالفة القريبة تحدد القفز النسبي لعنوان تعليمه القفز نفسها فمثلاً في تعليمه القفز بالالفة القصيرة يتم تشفير العدد ذي 8 بت كمتحول فوري لتحديد الإزاحة (Disp) ذات الإشارة التي تشير إلى التعليمه التالية التي سيتم تنفيذها من حجرة تعليمه القفز، و عندما تنفذ تعليمه القفز يعاد شحن الIP بقيمة جديدة موضحه كما يلي:  
قيمة IP الجديدة = ( قيمة IP + طول شيفرة تعليمه القفز ) + مقدار الإزاحة ذات الإشارة بعد تمديدها بجعل متحول 8 بتات بالشكل 16 بت ]

إن القيمة الجديدة لـ IP مع قيمة CS الحالية تعطي العنوان الفيزيائي للتعليمه التالية التي ستجلب و تنفذ.  
مثال:

ليكن لدينا

IP = 0112h

JMP disp ; disp = 0F2h

إن عنوان تعليمه القفز ( موجود تحت العنوان المخزن في IP ) ، إذن سيتم القفز إلى التعليمه ذات العنوان التالي:

(أهملنا خانة الحمل)  $0106h = 0112 + 2 + FFF2 = 0112 + 2 + disp =$  العنوان المنطقي address

بما أن العنوان الناتج أصغر من عنوان تعليمه القفز فهذا يعني أننا نقفز إلى تعليمه تسبق تعليمه القفز أي القفز نحو السوراء  
 $0106 < 0112$  .

مثال آخر:

IP = 0112h

JMP 04

Address =  $0112 + 2 + 0004 = 0118h$

نلاحظ أن  $0118 > 0112$  فهذا يعني أن القفز نحو الأمام.

و للحصول على العنوان الفيزيائي يجب إضافة مقدار الCS لقيمة address .  
ملاحظة: بما أن متحول الالافنة القصيرة ذو 8 بتات فهو يسمح بالقفز في المجال من 126- إلى 129+ و سبب ذلك أنه إذا أضفنا طول شيفرة تعليمة القفز و هو 2 بايت إلى المجال التالي من 128- إلى 127+ سنحصل على المجال السابق. أما متحول الالافنة القريبة فهو متحول فوري ذو 16 بت و لذلك يسمح بالقفز ضمن مجال يساوي 32KB نحو الخلف أو نحو الأمام من عنوان تعليمة القفز.  
مثال:

#### JMP label

هذا يعني القفز إلى نقطة في البرنامج مقابلة للمتحول label حيث تتم إضافة هذا المتحول (الإزاحة 16 بت) إلى قيمة الIP و القيمة الجديدة لـ IP و القيمة الحالية في CS تعطي العنوان الفيزيائي للتعليمة التي ستنفذ .  
ب) يمكن تحديد القفز إلى عنوان بشكل غير مباشر بواسطة محتويات حجرة ذاكرة أو محتويات مسجل أي باستخدام متحول مؤشر ذاكري 16 بت أو متحول مؤشر مسجلي 16 بت و هنا أيضاً يتم القفز ضمن مجال  $\pm 32KB$  .  
مثال:

#### JMP BX

في هذه التعليمة يُستعمل مضمون المسجل BX من أجل الإزاحة و هذا يعني أن قيمة BX يتم تحميلها في IP ثم يحسب العنوان الفيزيائي للتعليمة التي سيتم القفز إليها باستعمال المحتويات الحالية لـ CS و القيمة الجديدة لـ IP .  
بفرض أن :

$$\left. \begin{array}{l} BX = 0200h \\ CS = 0100h \end{array} \right\} \text{العنوان الفيزيائي للتعليمة التي سيتم القفز إليها}$$

$$PA = (CS \times 10h) + BX = 01000 + 0200 = 01200h$$

ملاحظة: يمكن استخدام مختلف أنواع أنظمة العنونة لتحديد المتحول المستعمل كمؤشر ذاكري فمثلاً [SI] JMP ففي هذه التعليمة تستعمل محتويات SI كعنوان حجرة الذاكرة التي تحتوي على العنوان الفعال، هذا العنوان يتم تحميله في IP و الذي يُستعمل مع محتويات CS الحالية لحساب العنوان الفيزيائي للتعليمة التي سيتم القفز إليها و عادة في هذه الحالة تستخدم المسجلات التالية: DI, SI, BX .

#### القفز بين المقاطع الجزئية أو القفز خارج المقطع الجزئية

أ) تُستعمل الالافنة البعيدة متحولاً فورياً ذا 32 بت لتحديد القفز إلى عنوان ما. حيث يتم تحميل الـ 16 بت الأولى من هذا المتحول في IP و تكون هي العنوان الفعال نسبة لمحتويات المسجل CS أما الـ 16 بت الثانية فيتم تحميلها في المسجل CS و التي تحدد مقطع الشيفرة الجديد.  
مثال:

#### JMP farlabel

حيث farlabel هو متحول بـ 32 بت (الكلمة الأولى تشحن في IP و الكلمة الثانية تشحن في الCS).  
ب) إن الطريقة غير المباشرة لتحديد العنوان الفعال و عنوان مقطع الشيفرة من أجل القفز بين المقاطع الجزئية هي باستخدام متحول مؤشر ذاكري بـ 32 بت. و في هذه الحالة فإن أربع بايتات من الذاكرة متتابعة اعتباراً من العنوان المحدد تحتوي على العنوان الفعال و عنوان مقطع الشيفرة الجديد على الترتيب. و هنا أيضاً يمكن استخدام أي نوع من أنواع أنظمة العنونة المختلفة،  
مثال:

JMP farseg [DI] ففي هذه التعليلة تُستعمل محتويات DI, DS لحساب عنوان حجرة الذاكرة التي تتضمن الكلمة الأولى للمؤشر الذي يُعرّف الحجرة التي سيتم القفز إليها، فإذا كان :

$$PA = DS \times 10h + DI = 01000 + 0200 = 01200h$$

$$\begin{cases} DI = 0200h \\ DS = 0100h \end{cases}$$

و لكن محتويات هذه الحجرة و الحجرات التي تليها كما هو واضح في الشكل التالي:

Address ( h )	Content
01200	10
01201	30
01202	00
01203	04

قيمة IP الجديدة هي IP = 3010h

قيمة CS الجديدة هي CS = 0400h

إذن العنوان الفيزيائي للتعليلة التي سيتم القفز إليها هو :

$$PA = CS \times 10h + IP = 07010h$$

## (2) تعليلة القفز المشروط

و هي مشروحة في الجدول التالي:

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
Jcc	قفز مشروط	متحول Jcc	إذا تحقق الشرط CC فإنه يتم القفز إلى العنوان المحدد بواسطة المتحول و إلا فيتم تنفيذ التعليلة التالية لتعليلة القفز	لا يوجد

هناك 18 من تعليمات القفز المشروط و هي مشروحة في الجدول التالي:

الكلمة المختزلة	المعنى
JC	القفز إذا كان CF = 1
JNC	القفز إذا كان CF = 0
JO	القفز إذا كان OF = 1
JNO	القفز إذا كان OF = 0
JS	القفز إذا كان SF = 1
JNS	القفز إذا كان SF = 0
JCXZ	القفز إذا كان CX = 0000
JE/JZ	القفز في حالة التساوي/أو إذا كان الناتج يساوي الصفر
JGE/JNL	القفز إذا كان أكبر أو يساوي/القفز إذا لم يكن أصغر
JA/JNBE	القفز إذا كان فوق/القفز إذا لم يكن تحت أو يساوي
JAE/JNB	القفز إذا كان فوق أو يساوي/القفز إذا لم يكن تحت
JB/JNAE	القفز إذا كان تحت/القفز إذا لم يكن فوق أو يساوي
JBE/JNA	القفز إذا كان تحت أو يساوي/القفز إذا لم يكن فوق
JG/JNLE	القفز إذا كان أكبر/القفز إذا لم يكن أصغر أو يساوي

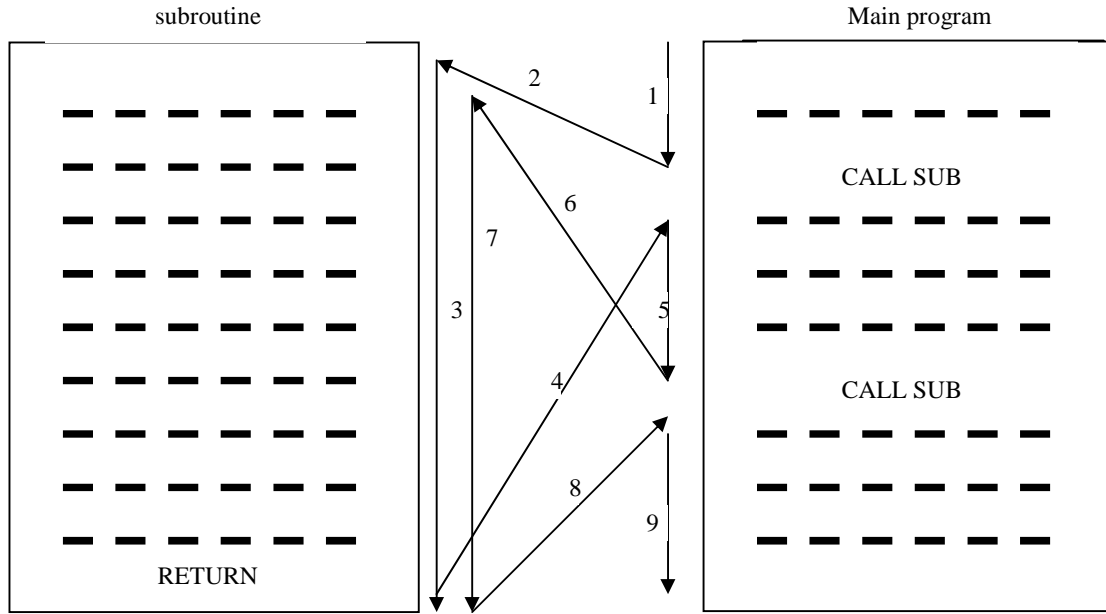
JLE/JNG	القفز إذا كان أصغر أو يساوي/القفز إذا لم يكن أكبر
JNE/JNZ	القفز إذا لم يكن يساوي/القفز إذا كان الناتج يساوي قيمة غير صفرية
JNB/JBO	القفز إذا كانت خانة Parity غير موجودة/القفز إذا كان $PF = 0$
JP/JPE	القفز في حالة وجود خانة Parity/القفز إذا كان $PF = 1$

ملاحظة:

للتمييز بين مقارنة الأعداد ذات الإشارة و الأعداد بدون إشارة فإن هناك اسمين مختلفين يبدو أنهما نفس الشيء في تعليمات القفز و هما فوق ( A ) و تحت ( B ) من أجل مقارنة الأعداد بدون إشارة، و أصغر ( L ) و أكبر ( G ) من أجل مقارنة الأعداد ذات الإشارة. فمثلاً العدد ABCDh هو فوق العدد 1234h إذا اعتبرناهما عددين بدون إشارة. أما إذا اعتبرناهما بإشارة فإن ABCDh هو عدد سالب و 1234h هو عدد موجب و لذلك ABCDh هو أصغر من 1234h.

## البرامج الفرعية SUBROUTINES

هي إجراءات مكتوبة بشكل مستقل عن البرنامج الرئيسي. متى وجب على البرنامج الرئيسي أن ينجز الوظيفة المحددة بواسطة البرنامج الفرعي فإنه يستدعي البرنامج الفرعي إلى العمل و من أجل هذا يجب أن يتحول التحكم من البرنامج الرئيسي إلى نقطة البداية في البرنامج الفرعي، حيث يستمر تنفيذ البرنامج الفرعي، و عند اكتمال التنفيذ يعود التحكم إلى البرنامج الرئيسي بالتعليمة التالية لتعليمة مناداة البرنامج الفرعي:



ملاحظة:

إن الفرق بين العمل لمناداة البرنامج الفرعي و القفز هو أن مناداة البرنامج الفرعي لا تنتج قفراً فقط إلى العنوان المناسب في ذاكرة تخزين البرنامج و لكنها أيضاً تملك تقنية من أجل حفظ المعلومات مثل IP و CS التي تكون مطلوبة للعودة إلى البرنامج الرئيسي.

### تعليمات المناداة و العودة

كلاً هاتين التعليمتين معاً تُزودان تقنية من أجل استدعاء البرنامج الفرعي إلى العمل و إعادة التحكم إلى البرنامج الأساسي لمتابعة تنفيذه. إن تعليمة المناداة مشروحة في الجدول:

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
CALL	مناداة برنامج فرعي	CALL operand	يُتابع التنفيذ في البرنامج الفرعي من العنوان المحدد بواسطة المتحول operand الموجود في تعليمة المناداة. و المعلومات المطلوبة من أجل العودة مثل IP و CS تُحفظ في المكس	لا يوجد

هناك 5 أنواع للمتحولات المسموح باستخدامها مع تعليمة المناداة و هي:

OPERAND	}	للمناداة ضمن المقطع الجزئي
Near_pro		
Memptr16	}	للمناداة خارج المقطع الجزئي
Regptr16		
Far_proc		
Memptr32		

إن المتحولات الثلاثة الأولى مخصصة للمناداة ضمن المقطع الجزئي للبرنامج الفرعي ( أي البرنامج الرئيسي و البرنامج الفرعي يقعان في نفس مقطع الشيفرة ) حيث أن تنفيذ تعليمة المناداة يسبب حفظ محتويات IP في المكس لأنه سوف يتم تعديل قيمة IP ألياً لتلائم مع البرنامج الفرعي. و عندئذ ينقص مؤشر المكس بمقدار 2 ، إن القيمة المحفوظة في IP ضمن المكس هي عنوان التعليمة التي تلي تعليمة المناداة.

بعد وضع قيمة IP في المكس ( أي حفظ العنوان الذي سنعود إليه بعد تنفيذ البرنامج الفرعي ) يتم شحن IP بعنوان و بقيمة جديدة ذات 16 بت هذه القيمة تشير إلى عنوان التعليمة الأولى من تعليمات البرنامج الفرعي المخزنة في الذاكرة، و يمكن ذكر تعليمة المناداة ضمن المقطع الجزئي على الشكل التالي كأمثلة على متحوات الجدول السابق و على الترتيب:

CALL near\_proc  
CALL [SI]  
CALL BX

أما النوع الآخر لتعليمة المناداة ( المناداة خارج المقطع الجزئي ) فهو يسمح للبرنامج الفرعي بأن يكمن في مقطع شيفرة آخر، و في هذه الحالة تستخدم المتحولات التالية Far\_pro ، Memptr32 ، كما هو واضح في الجدول السابق. تحدد هذه المتحولات كلاً من العنوان الجديد لـ IP و عنوان المقطع الجديد لـ CS . في كلتا الحالتين فإن تنفيذ تعليمة المناداة يسبب حفظ محتويات المسجلات CS ثم IP في المكس و من ثم تحميل القيم الجديدة المحددة بالمتحول operand في IP و CS . إن القيم المخترنة لـ CS و IP في المكس تسمح بالعودة إلى البرنامج الرئيسي من مقطع شيفرة آخر. إن المتحول Far\_proc يمثل متحولاً فورياً بـ 32 بت و الذي يكون مخزناً في البايتات الأربعة التي تلي رمز التعليمة ( opcode ) لتعليمة المناداة في ذاكرة البرنامج.

مثال :

CALL 01234321 حيث أن هاتان الكلمتان يتم تحميلهما مباشرة من ذاكرة تخزين البرنامج في IP و CS حيث CS هو مقطع الشيفرة للبرنامج الفرعي. إن عنوان التعليمة الأولى في البرنامج الفرعي يكون محدداً بالكلمة الأولى بعد تعليمة CALL أي يحزن ضمن IP . أما بالنسبة لمتحول المؤشر من نوع ذاكري بـ 32 بت فإن المؤشر للبرنامج الفرعي يكون مخزناً كأربعة بايتات في ذاكرة المعطيات، و الحجرة الأولى للمؤشر يمكن تحديدها بشكل مباشر بواسطة أحد المسجلات ( المثال هنا هو نفس مثال القفز [DI] JMP farseg السابق ).

إن كل برنامج فرعي يجب أن ينتهي بتنفيذ التعليمة التي تعيد التحكم إلى البرنامج الرئيسي و هذه التعليمة هي تعليمة العودة RET و هي مشروحة بالجدول التالي:

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
RET	العودة إلى البرنامج المُستدعي	RET/RET operand		لا يوجد

العودة إلى البرنامج المستدعي عن طريق إعادة تخزين قيم IP فقط أو IP و CS معاً (حسب نوع تعليمة المناداة أي ضمن المقطع الجزئي أو خارجه) من أجل المتحول Far\_pro . و إذا كان المتحول (operand) موجوداً في تعليمة العودة RET فيجب إضافته إلى محتويات SP . هذا و إن المتحول إذا وجد في تعليمة العودة فهو عبارة عن متحول إزاحة بـ 16 بت .

### تاسعاً - تعليمات الدفع و السحب

إن التعليمة المستخدمة لحفظ البارامترات في المكس هي تعليمة الدفع PUSH و التعليمة المستخدمة لاسترجاعها هي تعليمة POP . بعد سياق التحويل إلى البرنامج الفرعي نجد أنه من الضروري عادة حفظ محتويات المسجلات الرئيسية أو بعض بارامترات البرنامج الرئيسي هذه القيم يتم حفظها بواسطة دفعها إلى المكس . و بهذه الطريقة يتم حفظ المحتويات سليمة في مقطع المكس للذاكرة أثناء تنفيذ البرنامج الفرعي، و قبل العودة إلى البرنامج الرئيسي فإن المسجلات المحفوظة و بارامترات البرنامج الرئيسي يُعاد تخزينها بواسطة سحب القيم المحفوظة من المكس . لذلك فإن البنية النموذجية للبرنامج الفرعي تكون كالتالي:



ملاحظة: يتعامل المكس مع كلمات و ليس مع بايتات .

### تعليمات PUSH, POP

و هي مشروحة في الجدول التالي:

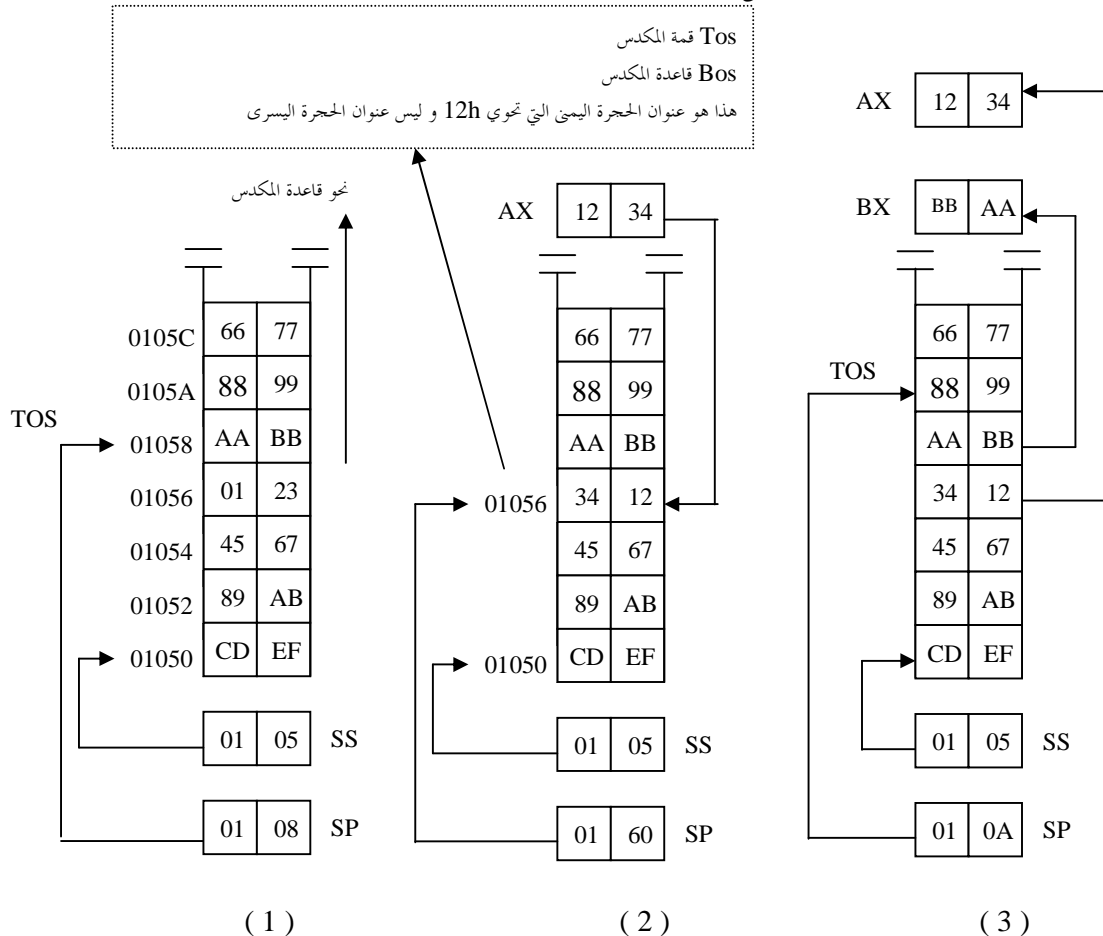
الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
PUSH	دفع كلمة إلى المكس	PUSH S	$S \rightarrow ((SP))$	لا يوجد
POP	سحب كلمة من المكس	POP D	$((SP)) \rightarrow D$	لا يوجد

### المكس، مسجل مقطع المكس SS، مؤشر المكس SP

أثناء عمليات المقاطعة و مناداة البرنامج الفرعي يتم دفع محتويات المسجلات الداخلية المعينة بالمعالج إلى قسم من الذاكرة يدعى بالمكس حيث تبقى هذه المحتويات هناك بشكل مؤقت . وعند إكمال روتين خدمة المقاطعة أو البرنامج الفرعي يتم سحب هذه القيم من المكس و توضع في نفس المسجل الداخلي حيث كان يحتوها أصلاً . فمثلاً عندما تحدث المقاطعة فإن المعالج و بشكل

أوتوماتيكي يدفع بمسجل الأعلام، القيمة الحالية في CS ، و القيمة الحالية في IP إلى المكس. يمكن الحصول على مقطع مكس جديد ببساطة بعنوان SS برمجياً من جديد. و إن مؤشر المكس SP يحتوي على العنوان الفعال نسبة للقيمة في SS . و العنوان المشتق من محتويات SS و SP هو العنوان الفيزيائي لحجرة التخزين الأخيرة في المكس (قمة المكس) التي تم دفع المعطيات إليها. إن القيمة في مؤشر المكس تبدأ بـ FFFFh عند بدء تشغيل المعالج. و إن جمع هذه القيمة مع القيمة الحالية الموجودة في SS يعطي الحجرة ذات العنوان العلوي في المكس (قاعدة المكس). بما أن المعطيات المنقولة من و إلى المكس عادة هي كلمات فإننا نتصور المكس على شكل حجرات ذات 2 بايت، كما أنه من الضروري أن تكون جميع حجرات المكس في حدود الكلمات الزوجية و ذلك لإنفاص عدد دورات الذاكرة المطلوبة لدفع أو سحب المعطيات من المكس. يقوم المعالج بدفع المعطيات و العناوين إلى المكس كلمة في كل مرة، و في كل مرة يتم دفع قيمة مسجل ما إلى قمة المكس فإن القيمة في مؤشر المكس أولاً تنقص بمقدار 2 و من ثم تُكتب محتويات ذلك المسجل في ذاكرة المكس. بهذه الطريقة فإن المكس ينمو نحو الأسفل في الذاكرة انطلاقاً من قاعدة المكس التي تطابق العنوان الفيزيائي المشتق من SS و القيمة FFFFh إلى نهاية (قمة) المكس و التي تطابق العنوان الفيزيائي المشتق من SS و العنوان الفعال 0000h و عندما تسحب القيمة من قمة المكس فإن العكس لهذا التسلسل يحدث. إن العنوان الفيزيائي المعرف بواسطة SS و SP دائماً يشير إلى حجرة القيمة الأخيرة المدفوعة إلى المكس حيث أن محتوياتها تسحب أولاً من المكس إلى المسجل المعني ضمن المعالج ثم يزداد SP بمقدار 2 . إن قمة المكس الجديدة تطابق القيمة السابقة المدفوعة إلى المكس.

مثال: تبين الأشكال الثلاثة التالية حالات المكس:





نلاحظ أن مسجل مقطع المكس بجوي على 0105h و كما أشرنا سابقاً فإن قاعدة المكس تكمن في العنوان الفيزيائي المشتق من SS مع العنوان الفعال FFFFh و هذا يعطي عنوان قاعدة المكس BOS :

$$A (bos) = 0105h + FFFF = 1104Fh$$

بالإضافة إلى ذلك فإن مؤشر المكس الذي يمثل العنوان الفعال من قاعدة المكس إلى قيمته يساوي 0008h لذلك فالقيمة الحالية للمكس هي في العنوان الفيزيائي:

$$A (tos) = 01050 + 0008 = 01058h$$

إن العناوين ذات القيم الأعلى من قيمة المكس 01058h تحتوي على معطيات حقيقية للمكس بينما المعطيات ذات العناوين الأدنى من قيمة المكس ليست معطيات حقيقية للمكس ( بالتعريف : المكس هو القيم المحصورة بين القاعدة و القمة ) . نلاحظ أن القيمة الأخيرة المدفوعة إلى المكس في الشكل الأول من الشكل السابق هي BBAAh . و يبين الشكل الثاني ما الذي يحدث عند تنفيذ تعليمة PUSH AX . هنا نجد أن محتويات AX هي 1234h و أن تنفيذ تعليمة PUSH يسبب إنقاص محتويات SP بمقدار 2 و لكنها لا تؤثر على محتويات مسجل مقطع المكس SS لذلك فإن الحجره التالية التي يتم الوصول إليها في المكس تقابل العنوان 01056h . إلى هذه الحجره يتم دفع القيمة المخزنة في AX إلى المكس . نلاحظ أن البايث العلوي من المسجل AX ( و الذي قيمته تساوي 12h ) يكمن الآن في البايث السفلي للكلمة في المكس و كذلك فالبايث السفلي من المسجل AX ( و الذي قيمته تساوي 34h ) يكمن الآن في البايث العلوي للكلمة في المكس . يبين الشكل الثالث ما الذي يحدث عندما تُسحب المعطيات من المكس إلى المسجل الذي دُفعت المعطيات منه إلى المكس و ذلك بعد تنفيذ التعليمة POP AX ثم POP BX على الترتيب . نفس المناقشة بالنسبة إلى دفع قيمة فورية إلى المكس .

### عاشراً - تعليمات الحلقات

هناك ثلاث تعليمات مصممة بشكل خاص لتحقيق عملية الحلقة . و هذه التعليمات يمكن استعمالها بدلاً من تعليمات القفز الشرطي . و هي مبينة في الجدول التالي:

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
LOOP	حلقة	LOOP short_label		لا يوجد

إنقاص CX بمقدار واحد دون التأثير على الأعلام ثم القفز إلى الحجره المعرّفة بواسطة اللافنة القصيرة إذا كان CX لا يساوي الصفر و إلا يتم تنفيذ التعليمة التالية لتعليمة الحلقة . و هنا يكون  $IP = IP + disp$  حيث  $disp$  أخذناها بعد تمديد إشارتها ( أي جعلها بـ 16 بت ) .

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
LOOPE/ LOOPZ	حلقة طالما يساوي/ أو طالما صفر	LOOPE/ LOOPZ } لافنة قصيرة		لا يوجد

إنقاص CX بمقدار واحد دون التأثير على الأعلام ثم القفز إلى الحجره المعرّفة بواسطة اللافنة القصيرة إذا كان CX لا يساوي الصفر و ZF يساوي الصفر و إلا يتم تنفيذ التعليمة التالية لتعليمة الحلقة . و هنا جسم الحلقة فقط هو الذي يؤثر على الأعلام .

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
LOOPNE/ LOOPNZ	حلقة طالما لا يساوي/ أو طالما ليس صفراً	LOOPNE/ LOOPNZ } لائنة قصيرة		لا يوجد

إنقاص CX بمقدار واحد ثم القفز إلى الحجرة المحددة بواسطة اللائنة القصيرة إذا كان CX لا يساوي الصفر و ZF يساوي الصفر و إلا يتم تنفيذ التعليمة التالية لتعليمة الحلقة.

و هنا أيضاً جسم الحلقة فقط هو الذي يؤثر على الأعلام.

مثال:

نريد البحث عن عنصر ضمن متجهة من العناصر ( مصفوفة أحادية البعد ) مثلاً : 8,9,4,5,7 و العنصر المراد إيجاد هو 4 . هنا CX = 5 و هو عدد العناصر. و يكون جسم الحلقة كالتالي:

```
MOV CX, 5
Nxt: -----
      -----
      -----
      -----
      LOOPNE Nxt
```

} جسم الحلقة

هنا يتم الخروج من الحلقة بالرغم من أن  $CX \neq 0$  لأنه حصل تطابق و أصبحت  $ZF = 1$ .

## 11 - تعليمات السلسلة

نقصد بكلمة السلسلة أن بايتات أو كلمات معطيات تكمن في حجرات متعاقبة للذاكرة. إن تعليمات السلسلة تسمح للمبرمج بتنفيذ عمليات مثل نقل المعطيات من بلوك ذاكرة إلى بلوك آخر في الذاكرة، مسح أو كس SCAN سلسلة من عناصر المعطيات المخزنة في الذاكرة و البحث عن قيمة معينة، مقارنة عناصر سلسلتين لتحديد فيما إذا كانا متطابقتين أو مختلفتين. و تعليمات السلسلة الأساسية هي:

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
MOVS	نقل عنصر من سلسلة	MOVS operand		لا يوجد

العملية: العنصر المحدد بواسطة DS:SI يتم نقله إلى الحجرة المحددة بواسطة القيمة ES:DI ثم:  
 $SI \pm 1 \text{ or } 2 \rightarrow SI$   
 $DI \pm 1 \text{ or } 2 \rightarrow DI$

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
-----------------	--------	--------	---------	------------------

MOVSB	نقل عنصر بايت من سلسلة	MOVSB	نفس العملية السابقة و مقدار التزايد هو 1	لا يوجد
MOVSW	نقل عنصر كلمة من السلسلة	MOVSW	نفس العملية السابقة و مقدار التزايد هو 2	لا يوجد
CMPS	مقارنة عنصر سلسلة	CMPS operand		أعلام الحالة

العملية: يتم طرح متحول الهدف من متحول المصدر و لا تُخزن النتيجة إنما تُعدل أعلام الحالة فقط، أي:

أعلام الحالة  $\rightarrow ((DS \times 10h) + SI) - ((ES \times 10h) + DI)$

$SI \pm 1 \text{ or } 2 \rightarrow SI$

$DI \pm 1 \text{ or } 2 \rightarrow DI$

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
SCAS(B or W)	مسح عنصر سلسلة	SCAS operand		أعلام الحالة

العملية :

أعلام الحالة  $\rightarrow (AL \text{ or } AX) - ((ES \times 10h) + DI)$

$DI \pm 1 \text{ or } 2 \rightarrow DI$

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
LODS (B or W)	تحميل عنصر سلسلة	LODS operand		لا يوجد

العملية :

$((DS \times 10h) + SI) \rightarrow (AL \text{ or } AX)$

$SI \pm 1 \text{ or } 2 \rightarrow SI$

الكلمة المختزلة	المعنى	الصيغة	العملية	الأعلام المتأثرة
STOS(B or W)	تخزين عنصر سلسلة	STOS operand		لا يوجد

العملية :

$(AL \text{ or } AX) \rightarrow ((ES \times 10h) + DI)$

$DI \pm 1 \text{ or } 2 \rightarrow DI$

## 12 - تعليمات تكرار السلسلة

في معظم التطبيقات يجب تكرار العمليات الأساسية للسلسلة من أجل معالجة جميع عناصرها. و يتم إنجاز هذا العمل بواسطة إدخال تعليمات التكرار قبل التعليمة الأساسية للسلسلة التي سوف تُكرر. هذا وإن أنواع تعليمات التكرار مبيّنة في الجدول التالي:

الكلمة المختزلة	المعنى	الاستخدام
REP	التكرار طالما لم يصل إلى نهاية السلسلة أي $CX \neq 0$	MOVS, STOS
REPE/REPZ	التكرار طالما لم يصل إلى نهاية السلسلة و السلسلتان متساويتان أي $ZF=1, CX \neq 0$	CMPS, SCAS
REPNE/REPZ	التكرار طالما لم يصل إلى نهاية السلسلة و السلسلتان غير متساويتان أي $ZF=0, CX \neq 0$	CMPS, SCAS

مثال:

بفرض أن :

SI = 0100h      DS = 0200h  
DI = 0110h      ES = 0400h

فإن نتيجة تنفيذ التعليمتين التاليتين :

MOV CX,20h  
REP MOVSB

هي أن التعليمة الأولى تقوم بتحميل المسجل CX بالقيمة  $20h = 32d$  أما التعليمة الثانية فتنتقل 32 بايت من حجرات ذاكرة المصدر المحددة بواسطة DS و SI إلى بلوك حجرات ذاكرة الهدف المحددة بواسطة ES و DI .

## 13 - تعليمتا مسح و توضيح علم الاتجاه

ذكرنا أنه يتم زيادة أو إنقاص قيم SI و DI بشكل أوتوماتيكي أثناء تنفيذ تعليمات السلسلة و أنه يتم تقرير الزيادة أو الإنقاص اعتماداً على قيمة علم الاتجاه DF حيث عندما  $DF = 0$  تحدث الزيادة الأوتوماتيكية و العكس بالعكس. و يتم التحكم بعلم الاتجاه بواسطة التعليمتين التاليتين:

الأعلام المتأثرة	العملية	الصيغة	المعنى	الكلمة المختزلة
------------------	---------	--------	--------	-----------------

CLD	تنظيف DF	CLD	0 → DF	DF
STD	توضيع DE	STD	1 → DF	DF

## 14 - تعليمتا IN و OUT

الكلمة المختزلة	المعنى	الصيغة	العملية
IN	تعليمه دخل مباشرة	IN Acc,port	(port) → Acc
IN	تعليمه دخل غير مباشرة	IN Acc,DX	((DX)) → Acc
OUT	تعليمه خرج مباشرة	OUT port,Acc	Acc → (port)
OUT	تعليمه خرج غير مباشرة	OUT DX,Acc	Acc → ((DX))

حيث في التعليمه المباشره يكون طول الـ port بايتاً واحداً و في التعليمه غير المباشره يكون DX محتويماً على عنوان نافذه.  
مثال:

بفرض أن نافذتي دخل بحجم بايت في العناوين A9h, AAh على الترتيب ستقرأ و من ثم سيتم إخراج محتوياتها إلى نافذه خرج بحجم كلمه في العنوان B000h المطلوب كتابة التعليمات اللازمه لإنجاز هذا العمل.  
الحل:

```
IN AL,[0AAh]
MOV AH,AL
IN AL,[0A9h]
MOV DX,0B000h
OUT DX,AX
```

تمت بحمد الله