

أسمبيلي للمبتدئين

أعداد المبرمج: مشتاق طالب العامري

لغة الأسمبيلي للمبتدئين

MUSHTAQ_TALIB58@YAHOO.COM

1/1/2009

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 و الواحدات في شيفرة الآلة للمعالج. إن الصيغة العامة لكتابة الأمر (التعليمة) في لغة الأسمبلي هي:

لافتة : تعليمة ؛ تعليق

عادة فإن التعليقات أو الملاحظات التي تصف الأوامر توضع على الطرف الأيمن. و هذا 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 code 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 ٨٠٨٦ 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 BX و AX 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 ٨٠٨٦

يُزود المعالج ٨٠٨٦ بمجموعة تعليمات مؤلفة من ١١٧ تعليمة أساسية و كذلك إن المجال الواسع للمتحولات و أنظمة العنونة المسموحة للاستعمال مع هذه التعليمات

يوسع مجموعة التعليمات إلى تعليمات أكثر، فمثلاً تعليمة Mov الأساسية تمتد إلى ٢٨ 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:

(١) 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 $S \rightarrow D$ MOV D,S نقل MOV

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

الحالات المستثناة من تعليمة MOV

١-لا تستطيع تعليمة MOV أن تنقل المعطيات بشكل مباشر بين حجرتي ذاكرة لذلك لا نرى في الجدول المجاور الحالة التالية : $Mem \rightarrow Mem$ 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.

٢-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

٣-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 العنوان في هذه التعليمة هو عنوان غير مباشرة بالمسجل و متحول المصدر عنوانه الفيزيائي هو $PA = DS \times 10h + SI$ أما متحول الهدف فهو AL .

٢) تعليمة التبديل XCHG

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

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

لا يوجد $S \rightarrow D$

$D \rightarrow S$ XCHG D,S تبديل XCHG

و يبين الجدول التالي مختلف أنواع متحولات 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 .

٣) تعليمة XLAT

تتعامل 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 .

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

لا يوجد $(AL + [BX]) \rightarrow AL$ XLAT DS x 10h جدول المصدر
ترجمة XLAT

مثال:

بفرض أنه لدينا جدول في الذاكرة يحتوي على مربعات الأعداد من ١ إلى ٩ أي أنه في أول حجرة من الجدول يوضع مربع العدد ١ وفي الحجرة الثانية يوضع مربع العدد ٢ (أي ٤) ، و هكذا ... ومن هذا نرى أن الجدول طوله تسع بايتات إزاحة بدايته عن بداية مقطع الـ DS هي ١٠٠ .

عندما يطلب منا الحصول على مربع أحد هذه الأعداد و ليكن العدد ٤ أي أن المطلوب هو أن تصبح قيمة $AL = 16$ لذلك نقوم بما يلي:

(١) نضع $AL = 3$ و $BX = 100$.

(٢) نعطي التعليمة XLAT .

و بعد تنفيذها يصبح $AL = 16$ و هو المطلوب.

(٤) التعليمات LEA, LES, LDS

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

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

لا يوجد $Mem16 \rightarrow reg16$ LEA reg16,mem16 تحميل
عنوان فعال LEA

لا يوجد $Mem32 \rightarrow reg16$

$DS \rightarrow Mem32+2$ LDS reg16,mem32 تحميل مسجل و المسجل DS
LDS

لا يوجد Mem32 → reg16

تحميل مسجل و المسجل ES LES reg16,mem32 Mem32+2 → ES
LES

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.

(١) تعليمات الجمع

و هي موصوفة بالجدول التالي:

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

أعلام الحالة $S + D \rightarrow D$

ADD جمع ADD D,S Carry → CF

أعلام الحالة $S + D + CF \rightarrow D$

ADC جمع مع أخذ الانزياح بعين الاعتبار ADC D,S Carry → CF

أعلام الحالة INC D $D+1 \rightarrow D$ الزيادة بمقدار واحد INC

AF,CF سيتم شرحها لاحقاً AAA تصحيح ناتج جمع عددين بشيفرة
الأسكي AAA

كل أعلام الحالة ماعدا OF سيتم شرحها لاحقاً DAA تصحيح ناتج جمع عددين
بشيفرة BCD DAA

تعليمتي ADD و ADC

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

و بالنسبة للمتحولات 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 للمتحولات من أجل توضيح حالة الأعلام

1 111 1111

$AX = 0100\ 1111\ 0011\ 1101\ b$

$BX = 1111\ 1101\ 1000\ 0001\ b$

+ $CF = 0001\ b$

$b\ 1111\ 1011\ 1100\ 0100$

و الآن أعلام الحالة هي:

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

$AF = 0$ لأنه لا يوجد انزياح من الخانة ٣ إلى الخانة ٤ في البايت 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

تعليمية التصحيح DAA

تستخدم هذه التعليمات لإنجاز عملية تصحيح ناتج 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 :

المنطقة الأولى أرقامها من ٠ إلى ٩ و فيها تكون نتيجة الجمع صحيحة و لا تحتوي على انزياح و ليست بحاجة إلى تصحيح مثلاً $9=2+7$ و هي أرقام واقعة ضمن نطاق المنطقة الأولى.

المنطقة الثانية أرقامها من ١٠ إلى ١٥ و فيها تكون نتيجة الجمع غير صحيحة و بحاجة إلى تصحيح بإضافة العدد ٦ فنحصل على رقم و حمل إلى العدد الثاني فمثلاً $E=5+9$ بإضافة ٦ إلى العدد E يكون الناتج $E=14+6$ و بذلك تكون النتيجة صحيحة.

المنطقة الثالثة أرقامها من ١٦ و حتى ١٨ 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 نمثل $AL = \text{Bit}7 \dots \text{Bit}0$

إن قاعدة التصحيح في هذه التعليمات هي :

(١) if Bit3 Bit2 Bit1 Bit0 of $AL > 9$ or $AF = 1$

then $AL = AL + 6$, $AF = 1$

(٢) if $AL > 9Fh$ or $CF = 1$

then $AL = AL + 60h$, $CF = 1$

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

ما هو ناتج تنفيذ ما يلي:

ADD AL,BL

DAA

الحل: إن نتيجة تنفيذ هاتين التعليمتين هي

$BCD = 0010 \ 1000$ b ٢٨

$$+ \quad \text{BCD} = 0110 \ 1000 \text{ b } ^{18}$$

$$\text{AL} \rightarrow \dots \ 1001$$

$$+ \quad \text{CF} = 0 \quad 0110$$

$$\text{AL} = 96 \text{ BCD} \leq 0110 \ 1001$$

$$\text{AF} = 1$$

تعليلة AAA

تستخدم هذه التعليلة لتصحيح ناتج جمع عددين بشيفرة آسكي (و هنا أيضاً يجب أن يكون ناتج الجمع في المسجل AL) و قاعدة التصحيح في هذه التعليلة 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: بفرض أن AL = 32h = 2 ASCII و BL = 34h = 4 ASCII ما هو ناتج تنفيذ التعليلتين التاليتين:

ADD AL,BL

AAA

الحل: إن ناتج تنفيذ هاتين التعليلتين هو كالتالي :

$$\text{AL} = 0011 \ 0010$$

$$+ \text{ BL} = 0011 \ 0100$$

$$\text{AL} = 66\text{h} \rightarrow \cdot 110 \cdot 110$$

$$\text{AL} = 06\text{h} , \text{AH} = 00$$

و هنا $\text{AF} = 0$ بسبب عدم وجود انزياح من الخانة ٣ إلى الخانة ٤ (حيث يبدأ الترقيم اعتباراً من الصفر)

(٢) تعليمات الطرح

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

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

أعلام الحالة $\text{D} - \text{S} \rightarrow \text{D}$

$\text{SUB D,S} \rightarrow \text{CF}$ طرَح SUB

أعلام الحالة $\text{D} - \text{S} - \text{CF} \rightarrow \text{D}$

$\text{SBB D,S} \rightarrow \text{CF}$ الطرح مع الاستعارة SBB

أعلام الحالة $\text{D} - 1 \rightarrow \text{D}$ DEC D الإنقاص بمقدار واحد DEC

أعلام الحالة $\text{D} - 0 \rightarrow \text{D}$

$\text{NEG D} \rightarrow \text{CF}$ المتمم الثنائي NEG

كل أعلام الحالة عدا OF سيتم شرحها لاحقاً DAS تصحيح ناتج طرَح

عددين بشيفرة BCD DAS

AF, CF سيتم شرحها لاحقاً AAS تصحيح ناتج جمع عددين بشيفرة

الأسكي AAS

ملاحظة: إن المتحولات المستخدمة من أجل تعليمتي الطرح SUB, SBB هي نفسها

المتحولات المسموحة من أجل تعليمتي 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 $F018h^2$ و بفرض كانت محتويات الحجرة التي يشير إليها العنوان الفيزيائي $[F018] = 0400h^2$ ، 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.

$$\text{Destination} = 0400h = 0000\ 0100\ 0000\ 0000\ b$$

$$+ \text{Source} = 03F8h = 1111\ 1100\ 0000\ 1000\ b$$

$$b\ 1000\ 0000\ 0000\ 0000\ 1$$

تذكرة بالمتمم الثنائي (و الذي يشار إليه بوضع خطين فوق العدد الذي نريد إيجاد المتمم الثنائي له) :

إذا أردت الحصول على المتمم الثنائي للعدد $F8h^{03}$ فاعمل ما يلي:

(١) تحويل هذا العدد إلى النظام الثنائي فيصبح $1000\ 1111\ 0011\ 0000$

(٢) أقلب الأصفار واحداث و الواحدات أصفاراً فينتج $1111\ 0011\ 1100\ 1111$

(٣) أضف واحد إلى الرقم الناتج فتحصل على المتمم الثنائي $-03 = F8h$
 $1111\ 1100\ 0000\ 1000$

لاحظ أن : $PF = 0$ لأن عدد الواحدات فردي في البايت 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.

تعليلة DAS

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

(١) if Bit3 Bit2 Bit1 Bit0 of AL > 9 or AF = 1

then AL = AL - 06 , AF = 1

(٢) if AL > 9Fh or CF=1

then AL = AL - 60h , CF = 1

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

SUB AL,AH

DAS

الحل:

AL = 1000 0110 b

+ AH = 1111 1001 b

b => AL = 7Fh ١١١١ ٠١١١ ١

و الآن :

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

CF = 0 لوجود انزياح خارجي

و بتطبيق الشرط ١ من قاعدة التصحيح نجد أن AF = 1 , AL = 79h

تعليلة AAS

تستخدم هذه التعليمات لتصحيح ناتج طرح عددين بالشفيرة 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:

AL = 0011 1000 b

+ BL = 1100 1011 b

b => AL = 03h ٠٠١١ ٠٠٠٠ ١

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

CF = 0 بسبب وجود انزياح خارجي

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

٣) تعليمات الضرب و القسمة

يتم تطبيق هذه التعليمات على الأعداد الثنائية أو بالشفيرة BCD أي في معالجة الأعداد ذات الإشارة و الأعداد بدون إشارة. و هذه التعليمات مبينة في الجدول التالي:

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

أعلام الحالة $AL.S8 \rightarrow AX$

MUL ضرب بدن إشارة $MUL SAX.S16 \rightarrow DX,AX$

أعلام الحالة

$Q[AX/S8] \rightarrow AL$

$R[AX/S8] \rightarrow AH$

$Q[(DX,AX)/S16] \rightarrow AX$

DIV تقسيم بدون إشارة $DIV S R[(DX,AX)/S16] \rightarrow DX$

ببساطة : النقطة تعنى 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 لجداول الضرب و القسمة):

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

أعلام الحالة $Q[AL/10d] \rightarrow AH$

$R[AL/10d] \rightarrow AL$

AAM تصحيح الناتج في AL من ضرب عددين BCD أو عددين ثنائيين

AAM

AH.10d + AL → AL SF, ZF, PF

•• AH → AAD تصحيح AX من أجل القسمة حيث AX ليس ناتج القسمة و إنما هو متحول الهدف في عملية القسمة. لذلك نطبق هذه التعليمات قبل تعليمات القسمة على عكس باقي تعليمات التصحيح AAD

لا يوجد MSB of AL →

CBW All bits of AH تحويل بايت إلى كلمة CBW

لا يوجد MSB of AX →

CWD CBW All bits of DX تحويل كلمة إلى كلمة مضاعفة CWD

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 ب ٨ بتات في AL على مقسوم عليه ب ٨ بتات أيضاً. و لإنجاز هذا يجب أولاً تمديد إشارة المقسوم لملاء 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 التعليمات CBW .
و بشكل مشابه فإن تعليمات التقسيم ٣٢ بت على ١٦ بت يمكن استخدامها لتقسيم مقسوم ذي ١٦ بت في AX على مقسوم عليه ذي ١٦ بت و ذلك بتحويل الكلمة إلى كلمة مضاعفة و يتم هذا بواسطة التعليمات CWD.

كما ذكرنا سابقاً فإن الأعداد غير المجمعة يتم حفظها كالتالي:

القسم العلوي من البايت الذي يحتوي على العدد غير المجمع يجب أن تكون قيمته مساوية إلى الصفر.

إن التعليمات 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 نصحها بواسطة التعليمات 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:

$$AL = 06 \times 10d + 04h = 64d = 40h$$

$$AH = 00h$$

ثالثاً - التعليمات المنطقية

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

الأعلام المتأثرة	العملية الصيغة	المعنى الكلمة المختزلة
أعلام الحالة	$S.D \rightarrow D$ AND D,S	AND المنطقي
أعلام الحالة	$S + D \rightarrow D$ OR D,S	OR المنطقي
أعلام الحالة		

XOR المنطقي XOR XOR D,S S + D → D

لا يوجد

NOT المنطقي NOT NOT D D → D

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

رابعاً - تعليمات الإزاحة

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

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

OF,CF إزاحة رياضية/إزاحة منطقية و SAL/SHL D,count

كلاهما نحو اليسار SAL/SHL

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 التعليمة على علم OF : إذا تبدلت خانة الإشارة نتيجة الإزاحة فإن $OF = 1$.

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

OF,CF

SHR إزاحة منطقية نحو اليمين SHR D,count

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 التعليمة على العلم OF : إذا تبدلت خانة الإشارة نتيجة الإزاحة فإن $OF = 1$

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

أعلام الحالة

SAR D,count إزاحة رياضية نحو اليمين SAR

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 SHL, SAL : إذا طبقنا هاتين التعليمتين من أجل الإزاحة بعدد من الخانات $\text{count} = N$ فهذا يعني ضرب متحول الهدف بـ 2 و الذي هو مضاعفات العدد 2 .

ملاحظة: إن التعليمية SHR تعني تقسيم متحول الهدف على العدد تحت كون LSB = 0 كل مرة و في حالة $\text{LSB} = 1$ فعندها يكون لدينا باقي موضوع في العلم CF .

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

$$AX) + 7.(BX) \rightarrow DX).^3$$

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

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 => يجب أن نكتب :

MOV CL,count

SAL AX,CL

هذا و إن قيمة count محددة بالمجال [FF,١] و الأقواس المحيطة ليس لها علاقة بمفهوم الإزاحة طبعاً.

خامساً - تعليمات التدوير

و هي مبينة في الجدول التالي:

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

OF,CF ROL D,count تدوير نحو اليمين ROL

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.

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

OF,CF ROR D,count تدوير نحو اليسار ROR

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.

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

OF,CF

RCL D,count تدوير نحو اليسار عبر الـ CF RCL

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.

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

OF,CF

RCR D,count تدوير نحو اليمين عبر الـ CF RCR

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 :

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

لا يوجد AH → Flags

النصف الأول من مسجل الأعلام يوضع في AH LAHF تحميل AH من مسجل الأعلام LAHF

أعلام الحالة عدا OF → AH

يوضع AH في النصف الأول من مسجل الأعلام SAHF تخزين قيمة AH في مسجل الأعلام SAHF

CF → 0 CF CLC تنظيف الـ CF CLC

CF → 1 CF STC توضيع الـ CF STC

CF

CF → CF CMC متمم أحادي لـ CF CMC

IF → 0 IF CLI تنظيف IF CLI

IF → 1 IF STI توضيع الـ IF STI

سابعاً - تعليمات المقارنة

تسمح تعليمة المقارنة CMP بمقارنة عددين ب ٨ بت أو ١٦ بت و هي مشروحة بالجدول التالي:

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

أعلام الحالة D - S

تتأثر الأعلام CMP D,S مقارنة عددين CMP

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:

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، و إلا يُتابع التنفيذ بالتعليمة التي تلي تعليمة القفز في البرنامج.

(١) 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 لتعليم القفز غير المشروط هي :

القفز ضمن المقطع الجزئي

أ (إن متحولات اللافتة القصيرة و اللافتة القريبة تحدد القفز النسبي لعنوان تعليم القفز نفسها فمثلاً في تعليم القفز باللافتة القصيرة يتم تشفير العدد ذي ٨ بت كمتحول فوري لتحديد الإزاحة (Disp) ذات الإشارة التي تشير إلى التعليم التالية التي سيتم تنفيذها من حجرة تعليم القفز، و عندما تنفذ تعليم القفز يعاد شحن الIP بقيمة جديدة موضحة كما يلي:

قيمة IP الجديدة = [قيمة IP + طول شيفرة تعليم القفز (+ مقدار الإزاحة ذات الإشارة بعد تمديدها بجعل متحول ٨ بتات بالشكل ١٦ بت]

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

مثال:

ليكن لدينا

IP = 0112h

JMP disp ; disp = 0F2h

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

address العنوان المنطقي = IP + 2 + disp (بعد تمديد إشارتها) = ٠١١٢ + ٢
FFF2 = 0106h + ٢ (أهملنا خانة الحمل)

بما أن العنوان الناتج أصغر من عنوان تعليم القفز فهذا يعني أننا نقفز إلى تعليم تسبق تعليم القفز أي القفز نحو الـ ٠١٠٦ وراء ٠١١٢ .

مثال آخر:

IP = 0112h

JMP 04

$$\text{Address} = 0112 + 2 + 0004 = 0118h$$

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

و للحصول على العنوان الفيزيائي يجب إضافة مقدار الـ CS لقيمة address .

ملاحظة: بما أن متحول اللافتة القصيرة ذو ٨ بتات فهو يسمح بالقفز في المجال من ١٢٦- إلى ١٢٩+ و سبب ذلك أنه إذا أضفنا طول شيفرة تعليمة القفز و هو ٢ بايت إلى المجال التالي من ١٢٨- إلى ١٢٧+ سنحصل على المجال السابق. أما متحول اللافتة القريبة فهو متحول فوري ذو ١٦ بت و لذلك يسمح بالقفز ضمن مجال يساوي KB^{32} 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 ١٦ بت) إلى قيمة الـ IP 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 ١٦ بت أو متحول مؤشر مسجلي ١٦ بت و هنا أيضاً يتم القفز ضمن مجال .

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 :

$$BX = 0200h$$

$$CS = 0100h$$

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

ملاحظة : يمكن استخدام مختلف أنواع أنظمة العنونة لتحديد المتحول المستعمل كمؤشر ذاكري فمثلاً [SI] [JMP] ففي هذه التعليمات 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 للتعليمات 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) تستعمل اللافتة البعيدة متحولاً فورياً ذا ٣٢ بت لتحديد القفز إلى عنوان ما. حيث يتم تحميل الـ ١٦ بت الأولى من هذا المتحول في IP و تكون هي العنوان الفعال نسبة لمحتويات المسجل CS أما الـ ١٦ بت الثانية فيتم تحميلها في المسجل CS و التي تحدد مقطع الشيفرة الجديد.

مثال:

JMP farlabel

حيث farlabel هو متحول بـ ٣٢ بت (الكلمة الأول تشحن في IP و الكلمة الثانية تشحن في الـ CS).

ب) إن الطريقة غير المباشرة لتحديد العنوان الفعال و عنوان مقطع الشيفرة من أجل القفز بين المقاطع الجزئية هي باستعمال متحول مؤشر ذاكري بـ ٣٢ 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:

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

DI = 0200h إن العنوان الفيزيائي للمؤشر هو :

$$DS = 0100h$$

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

و لتكن محتويات هذه الحجرة و الحبرات التي تليها كما هو واضح في الشكل التالي:

قيمة IP الجديدة هي $IP = 3010h$ (Address (h Content

قيمة CS الجديدة هي $CS = 0400h$ ١٠ ٠١٢٠٠

إذن العنوان الفيزيائي للتعليلة التي سيتم القفز إليها هو: ٣٠ ٠١٢٠١

$PA = CS \times 10h + IP = 07010h$ ٠٠ ٠١٢٠٢

٠٤ ٠١٢٠٣

(٢) تعليلة القفز المشروط

و هي مشروحة في الجدول التالي:

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

لا يوجد إذا تحقق الشرط CC فإنه يتم القفز إلى العنوان المحدد بواسطة المتحول و إلا فيتم تنفيذ التعليلة التالية لتعليلة القفز JCC متحول قفز مشروط JCC

هناك ١٨ من تعليلات القفز المشروط و هي مشروحة في الجدول التالي:

المعنى الكلمة المختزلة

القفز إذا كان $JC CF = 1$

القفز إذا كان $JNC CF = 0$

القفز إذا كان $JOOF = 1$

القفز إذا كان $JNOOF = 0$

القفز إذا كان $JS SF = 1$

القفز إذا كان $JNS SF = 0$

القفز إذا كان $JCXZ CX = 0000$

القفز في حالة التساوي/أو إذا كان الناتج يساوي الصفر JE/JZ

القفز إذا كان أكبر أو يساوي/القفز إذا لم يكن أصغر JGE/JNL

القفز إذا كان فوق/القفز إذا لم يكن تحت أو يساوي JA/JNBE

القفز إذا كان فوق أو يساوي/القفز إذا لم يكن تحت JAE/JNB

القفز إذا كان تحت/القفز إذا لم يكن فوق أو يساوي JB/JNAE

القفز إذا كان تحت أو يساوي/القفز إذا لم يكن فوق JBE/JNA

القفز إذا كان أكبر/القفز إذا لم يكن أصغر أو يساوي JG/JNLE

القفز إذا كان أصغر أو يساوي/القفز إذا لم يكن أكبر JLE/JNG

القفز إذا لم يكن يساوي/القفز إذا كان الناتج يساوي قيمة غير صفرية JNE/JNZ

القفز إذا كانت خانة Parity غير موجودة/القفز إذا كان $PF = 0$ JNB/JBO

القفز في حالة وجود خانة Parity/القفز إذا كان $PF = 1$ JP/JPE

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 ١٢٣٤h إذا اعتبرناهما عددين بدون إشارة. أما إذا اعتبرناهما بإشارة فإن ABCDh هو عدد سالب و ١٢٣٤h هو عدد موجب و لذلك ABCDh هو أصغر من ١٢٣٤h.

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:

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

لا يوجد 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 CALL operand مناداة برنامج فرعي CALL
هناك ٥ أنواع للمتحولات المسموح باستخدامها مع تعليمة المناداة و هي:

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 آلياً لتلائم مع البرنامج الفرعي. و عندئذ ينقص مؤشر المكس بمقدار ٢ ، 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 ١٦ 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، Mempt32 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 المختزنة لـ CS و IP في المكس تسمح بالعودة إلى البرنامج الرئيسي من مقطع شيفرة آخر. إن المتحول Far_proc يمثل متحولاً فورياً بـ ٣٢ 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 يخزن ضمن IP. أما بالنسبة لمتحول المؤشر من نوع ذاكري بـ ٣٢ 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 [JMP farseg [DI 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:

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

لا يوجد

RET/RET operand العودة إلى البرنامج المُستدعي RET

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 ب ١٦ 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:

PUSH xx

PUSH yy لحفظ المسجلات و البارامترات في المكس

PUSH zz

----- جسم البرنامج الفرعي

POP zz

نلاحظ أنه يتم الاسترجاع هنا بشكل POP yy لاسترجاع المسجلات و البارامترات من المكس

عكسي و ذلك لأن مبدأ عمل المكس

POP xx

هو LIFO RET للعودة إلى البرنامج المستدعي

ملاحظة: يتعامل المكس مع كلمات و ليس مع بايتات.

تعليمات PUSH, POP

و هي مشروحة في الجدول التالي:

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

المكس، مسجل مقطع المكس SS، مؤشر المكس SP

أثناء عمليات المقاطعة ومناداة البرنامج الفرعي يتم دفع محتويات المسجلات الداخلية المعيّنة 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 وتوضع 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 تمّ دفع المعطيات إليها. إن القيمة في مؤشر المكس تبدأ ب 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 ٢ بايت، كما أنه من الضروري أن تكون جميع حجرات المكس في حدود الكلمات الزوجية و ذلك لإنقاص 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 ٢ 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 h٠٠٠٠ و عندما تسحب القيمة من قمة المكس فإن العكس لهذا التسلسل يحدث. إن العنوان الفيزيائي المعروف بواسطة SS و SP دائماً يشير إلى حجرة القيمة الأخيرة المدفوعة إلى المكس حيث أن محتوياتها تسحب أولاً من

المكدس إلى المسجل المعني ضمن المعالج ثم يزداد SP بمقدار ٢ . إن قمة المكدس الجديدة تطابق القيمة السابقة المدفوعة إلى المكدس.

مثال: تبين الأشكال الثلاثة التالية حالات المكدس:

نلاحظ أن مسجل مقطع المكدس يحوي على $h0105$ و كما أشرنا سابقاً فإن قاعدة المكدس تكمن في العنوان الفيزيائي المشتق من SS مع العنوان الفعال FFFFh و هذا يعطي عنوان قاعدة المكدس BOS :

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

بالإضافة إلى ذلك فإن مؤشر المكدس الذي يمثل العنوان الفعال من قاعدة المكدس إلى قمته يساوي $h0008$ لذلك فالقمة الحالية للمكدس هي في العنوان الفيزيائي:

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

إن العناوين ذات القيم الأعلى من قمة المكدس $h01058$ تحتوي على معطيات حقيقية للمكدس بينما المعطيات ذات العناوين الأدنى من قمة المكدس ليست معطيات حقيقية للمكدس (بالتعريف : المكدس هو القيم المحصورة بين القاعدة و القمة) . نلاحظ أن القيمة الأخيرة المدفوعة إلى المكدس 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 $h1234$ و أن تنفيذ تعليمة PUSH يسبب إنقاص محتويات SP بمقدار ٢ و لكنها لا تؤثر على محتويات مسجل مقطع المكدس SS لذلك فإن الحجرة التالية التي يتم الوصول إليها في المكدس تقابل العنوان $h01056$. إلى هذه الحجرة يتم دفع القيمة المخزنة في AX إلى المكدس. نلاحظ أن البايت العلوي من المسجل AX (و الذي قيمته تساوي $h12$) يكمن الآن في البايت السفلي للكلمة في المكدس و كذلك فالبايت السفلي من المسجل AX (و الذي قيمته تساوي $h34$) يكمن الآن في البايت العلوي للكلمة في المكدس.

يبين الشكل الثالث ما الذي يحدث عندما تُسحب المعطيات من المكدس إلى المسجل الذي دُفعت المعطيات منه إلى المكدس و ذلك بعد تنفيذ التعليمة POP AX ثم POP BX على الترتيب. نفس المناقشة بالنسبة إلى دفع قيمة فورية إلى المكدس.

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:

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

لا يوجد

LOOP short_label حلقة

إنقاص CX بمقدار واحد دون التأثير على الأعلام ثم القفز إلى الحجرة المعرفة بواسطة اللافتة القصيرة إذا كان CX لا يساوي الصفر و إلا يتم تنفيذ التعليمة التالية لتعليمة الحلقة. و هنا يكون $IP = IP + disp$ حيث أخذناها بعد تمديد إشارتها (أي جعلها ب ١٦ بت).

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

لا يوجد

/LOOPE

LOOPZ حلقة طالما يساوي/ أو طالما صفر LOOPE/ LOOPZ

إنقاص CX بمقدار واحد دون التأثير على الأعلام ثم القفز إلى الحجرة المعرفة بواسطة اللافتة 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.

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

لا يوجد

/LOOPNE

LOOPNZ حلقة طالما لا يساوي/ أو طالما ليس صفراً LOOPNE/ LOOPNZ

إنقاص CX بمقدار واحد ثم القفز إلى الحجرة المحددة بواسطة اللافتة 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 :
٧,٥,٤,٩,٨ و العنصر المراد إيجاده هو ٤ . هنا $CX = 5$ و هو عدد العناصر. و
يكون جسم الحلقة كالتالي:

MOV CX, 5

----- :Nxt

LOOPNE Nxt

١١ - 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 كنس 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:

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

لا يوجد	MOVS operand	نقل عنصر من سلسلة MOVS
---------	--------------	------------------------

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

لا يوجد	نفس العملية السابقة و مقدار التزايد هو ١	MOVSB	نقل عنصر
---------	--	-------	----------

بايت من سلسلة	MOVSB
---------------	-------

لا يوجد نفس العملية السابقة و مقدار التزايد هو ٢ MOVSW نقل عنصر
كلمة من السلسلة MOVSW

أعلام الحالة CMPS operand مقارنة عنصر سلسلة CMPS

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

أعلام الحالة SCAS operand مسح عنصر سلسلة SCAS(B or W

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

لا يوجد

LODS operand تحميل عنصر سلسلة LODS (B or W

الأعلام المتأثرة العملية الصيغة المعنى الكلمة المختزلة

لا يوجد

STOS operand تخزين عنصر سلسلة STOS(B or W

١٢ - 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 التعليمات الأساسية للسلسلة التي سوف تُكرر. هذا و إن أنواع تعليمات التكرار مبينة في الجدول التالي:

الاستخدام المعنى الكلمة المختزلة

MOV, STOS التكرار طالما لم نصل إلى نهاية السلسلة أي

REP CX ≠ 0

CMPS, SCAS التكرار طالما لم نصل إلى نهاية السلسلة و السلسلتان

متساويتان أي REPE/REPZ=1, CX≠0

CMPS, SCAS التكرار طالما لم نصل إلى نهاية السلسلة و السلسلتان غير

متساويتان أي REPNE/REPNZ=0, CX≠0

مثال:

بفرض أن :

DS = 0200h

SI = 0100h

ES = 0400h

DI = 0110h

فإن نتيجة تنفيذ التعليمتين التاليتين :

MOV CX,20h

REP MOVSB

هي أن التعليمية الأولى تقوم بتحميل المسجل CX بالقيمة $h = 32d_{20}$ أما التعليمية الثانية فتنتقل ٣٢ بايت من حجرات ذاكرة المصدر المحددة بواسطة DS و SI إلى بلوك حجرات ذاكرة الهدف المحددة بواسطة ES و DI .

١٣ - تعليمتا مسح و توضيع 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

DF ٠ $DF \rightarrow$ CLD تنظيف DF CLD

DF ١ $DF \rightarrow$ STD توضيع DE STD

١٤ - تعليمتا IN و OUT

العملية الصيغة المعنى الكلمة المختزلة

$port) \rightarrow Acc$ IN Acc,port تعليمية دخل مباشرة IN

$DX)) \rightarrow Acc))$ IN Acc,DX تعليمية دخل غير مباشرة IN

$Acc \rightarrow (port)$ OUT port,Acc تعليمية خرج مباشرة OUT

$Acc \rightarrow ((DX))$ OUT DX,Acc تعليمية خرج غير مباشرة OUT

حيث في التعليمات المباشرة يكون طول الـ port بايتاً واحداً و في التعليمات غير المباشرة يكون DX محتويًا على عنوان نافذة.

مثال:

بفرض أن نافذتي دخل بحجم بايت في العناوين AAh, A9h على الترتيب ستُقرأ و من ثم سيتم إخراج محتوياتها إلى نافذة خرج بحجم كلمة في العنوان B000h المطلوب كتابة التعليمات اللازمة لإنجاز هذا العمل.

الحل:

أسمبلي للمبتدئين

أعداد المبرمج: مشتاق طالب العامري

لغة الأسمبلي للمبتدئين

[IN AL,[0AAh

MOV AH,AL

[IN AL,[0A9h

MOV DX,0B000h

OUT DX,AX

MUSHTAQ_TALIB58@YAHOO.COM

1/1/2009