المسجلات Registers

يملك المعالج 8086 أربعة مجموعات من المسجلات ذات 16 بت يستطيع المبرمج الوصول إليها و هي:

1) مؤشر التعليمة IP
2) أربعة مسجلات معطيات AX,BX,CX,DX .
3) أربعة مسجلات تأشير و فهرسة SI,DI,BP,SP .
4) أربعة مسجلات مقاطع CS,DS,SS,ES .

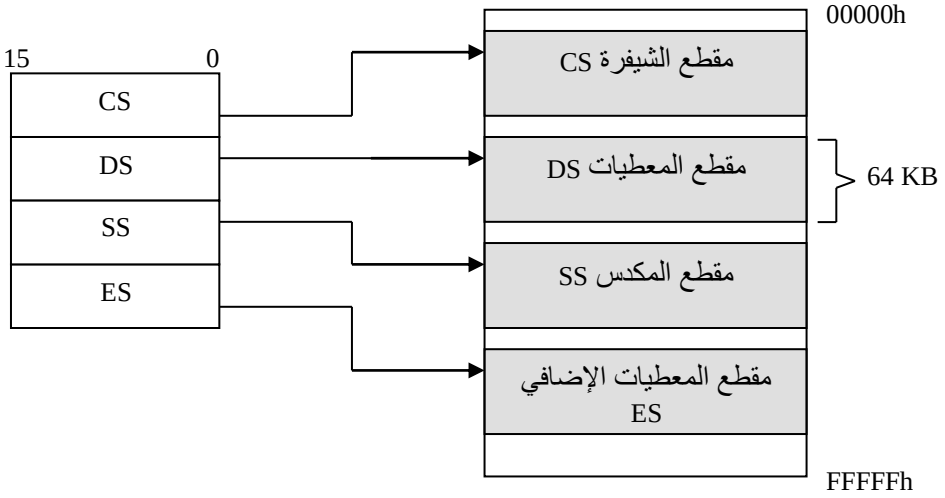
بالإضافة إلى ذلك يوجد مسجل آخر هو مسجل الأعلام و يدعى أيضاً مسجل الحالة و هو مسجل ذو 16 بت و لكن نستخدم منه 9 خانات فقط.

سنشرح كلٍ من هذه المسجلات بالتفصيل :

المجموعة الأولى : مسجلات المقاطع

و هي عبارة عن أربعة مسجلات طول كل منها 16 بت أي 2 بايت و هي :

1) مسجل مقطع الشيفرة CS : يحتوي على عنوان أول حجرة في مقطع شيفرة البرنامج في الذاكرة، أي أنه يشير إلى بداية مقطع الشيفرة.
2) مسجل مقطع المعطيات DS : يحتوي على عنوان أول حجرة في مقطع المعطيات في الذاكرة، أي أنه يشير إلى بداية مقطع المعطيات.
3) مسجل مقطع المكدس SS : يحتوي على عنوان أول حجرة في مقطع المكدس في الذاكرة، أي أنه يشير إلى بداية مقطع المكدس.
4) مسجل مقطع المعطيات الإضافي ES : يحتوي على عنوان أول حجرة في مقطع المعطيات الإضافي في الذاكرة، أي أنه يشير إلى بداية مقطع المعطيات الإضافي.



المجموعة الثانية: مسجلات الفهرسة و التأشير

و هي عبارة عن أربعة مسجلات مساعدة تساعد في إيجاد العنوان الفيزيائي بالتعاون مع مسجلات المقاطع، و طول هذه المسجلات 16 بت أي 2 بايت، و هي :

1) مسجل دليل المصدر Source Index SI : يخزن فيه عنوان يدل على الإزاحة ضمن مقطع المعطيات DS و بمعنى آخر يستعمل في إمساك العناوين الفعالة من أجل التعليمات التي تتناول المعطيات المخزنة في مقطع المعطيات في الذاكرة.

2) مسجل دليل الهدف Destination Index DI : يخزن فيه عنوان يدل على الإزاحة ضمن مقطع المعطيات الإضافي ES ، و بمعنى آخر يستعمل مسجل دليل الهدف DI من أجل استنتاج العنوان الفيزيائي الذي يحدد حجرة متحول الهدف.
3) مسجل مؤشر المكدس Stack Pointer SP : يسمح مؤشر المكدس بوصول سهل للحجرات في مقطع المكدس الموجود في الذاكرة حيث أن القيمة في SP تمثل العنوان الفعال لحجرة المكدس التالية التي يمكن الوصول إليها نسبة إلى العنوان الحالي الموجود في مسجل مقطع المكدس SS و يحتفظ SP دوماً بقيمة تدل على قمة المكدس ، هذا و إن قيمة هذا المسجل تتعدل تلقائياً عند وضع أو سحب معلومة بالمكدس.
4) مسجل مؤشر القاعدة Base Pointer BP : يحوي قيمة تدل على الإزاحة بالنسبة لمقطع المكدس SS و هو يستخدم لقراءة المعطيات ضمن مقطع المكدس بدون إزالتها من المكدس.

المجموعة الثالثة: مسجلات المعطيات

تستخدم هذه المسجلات من أجل التخزين المؤقت للنتائج المرحلية أثناء تنفيذ البرنامج حيث أن تخزين المعطيات في هذه المسجلات يمكننا من الولوج إلى تلك المعطيات بشكل أسرع مما لو كانت في الذاكرة، و تقسم المسجلات إلى :

1) مسجل المراكم Accumulator و يرمز له بالرمز A .
2) مسجل القاعدة Base و يرمز له بالرمز B .
3) مسجل العد Count و يرمز له بالرمز C .
4) مسجل المعطيات Data و يرمز له بالرمز D .

و كل مسجل من المسجلات السابقة يمكن استعماله إما ككلمة 16 بت و يدل على ذلك بكتابة الحرف X بعد اسم المسجل أو يمكن استعماله كبايتين كل منهما 8 بت و يدل على ذلك باستخدام الحرفين H,L حيث :

L للبايت ذو العنوان الأصغر ، مثال AL .
H للبايت ذو العنوان الأكبر ، مثال BH .

هكذا و إن كلاً من هذه المسجلات يمكن استخدامه من أجل التعليمات الرياضية أو المنطقية في لغة الأسمبلي مثل And, Add .

و من أجل بعض التعليمات مثل البرامج التي تحتوي على تعليمات سلاسل فإنها تستعمل مسجلات معينة مثل استعمال المسجل C لتخزين العدد الذي يمثل عدد البايتات التي ستنفذ عليها تعليمات السلاسل (عدد مرات تكرار تعليمة السلسلة)

مسجل مؤشر التعليمة Instruction Pointer IP

هذا المسجل يحدد موقع التعليمة التالية التي ستنفذ في مقطع الشيفرة و بعد جلب شيفرة التعليمة من الذاكرة فإن BIU تعدل قيمة IP بحيث تشير إلى التعليمة التالية في الذاكرة (التعديل يتم آلياً).

مسجل الأعلام Flags Register

هو مسجل ذو 16 بت موجود في وحدة التنفيذ كما هو واضح بالشكل :

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

و كما نلاحظ من الشكل السابق أنه يوجد ستة أعلام للحالة هي CF, PF, AF, ZF, SF, OF ، و كذلك يوجد ثلاثة أعلام للتحكم DF, IF, TF .

أ) أعلام الحالة

تشير إلى الحالات الناتجة كنتيجة لتنفيذ تعليمة منطقية أو رياضية حيث تكون إما في حالة واحد منطقي Set أو تكون في حالة صفر منطقي Reset ، و سنلخص فيما يلي عمل كلٍ منها:

أولاً: علم الإنزياح Carry Flag

يكون في حالة الواحد المنطقي إذا وجد انزياح خارجي (حمل) أو استعارة من أجل الخانة الأخيرة (البت الأخير) و ذلك أثناء تنفيذ التعليمات الرياضية.
و يكون في حالة الصفر المنطقي إذا لم يوجد حمل أو استعارة من أجل البت الأخير.
أمثلة:

أولاً: حالة الإنزياح

7	6	5	4	3	2	1	0	
1	1	0	0	0	1	1	0	
1	1	0	0	0	1	1	1	+
1	0	0	0	1	1	0	1	

CF=1

لاحظ بأن النتيجة لم تتسع في ثمانية بتات و إنما تحتاج إلى تسع بتات و نعبر عن ذلك بثمانية بتات و CF=1 أي أنه لدينا في اليد واحد.
ببساطة: فمهما كَبُرَ العددان فإن تسعة بتات يمكن أن تستوعبها.

7	6	5	4	3	2	1	0	
1	0	1	0	1	1	0	1	
	1	1	1	1	1	0	0	0	-
	0	1	1	0	0	0	1	1

1

لاحظ بأن العدد الأول الممثل ثنائياً أصغر من العدد الثاني الممثل ثنائياً أيضاً ، لذلك فعند إجراء عملية الطرح و في مثالنا هذا تخيلنا بت تاسع فيه القيمة واحد (استعرنا) و بالتالي فإن CF=1 أي لدينا استعارة من أجل البت الأعلى رتبة.

لام من أجل 2 بايت و لكن الإنزياح الخارج و الاستعارة تكون من أجل البت الخامس عشر (الأخير).

ثانياً: علم الازدواجية Parity Flag PF

يصبح في حالة واحد منطقي إذا كانت نتيجة آخر تعليمة تحوي على عدداً زوجياً من الخانات الواحدية (بعد التحويل إلى النظام الثنائي طبعاً) و إلا يكون في حالة الصفر المنطقي.
نلاحظ أن علم PF يفحص البايت السفلي فقط حتى لو كنا نتعامل مع كلمة (2 بايت) ، أما عندما نتعامل مع بايت واحد فقط فإنه يفحصه كله.

ثالثاً: علم الإنزياح المساعد Auxiliary Flag AF

يكون في حالة الواحد المنطقي إذا وجد إنزياح من النصف السفلي إلى النصف العلوي أو استعارة من النصف العلوي إلى النصف السفلي و ذلك من أجل البايت السفلي من الكلمة (2 بايت) و بمعنى آخر أنه إذا كان لدينا إنزياح من الخانة 3 إلى الخانة 4 فإن AF=1 و ذلك في حال كانت المعطيات بايت واحد أو بايتين (كلمة)، و فيما عدا ذلك يكون AF=0 .
مثال:

7	6	5	4	3	2	1	0	
1	1	0	0	0	1	1	0	
1	1	0	0	0	1	1	1	+
1	0	0	0	1	1	0	1	

الخانة الخانة

في هذه الحالة يكون AF=0 لأنه لم يكن معنا باليد واحد عند الانتقال من الخانة الثالثة إلى الخانة الرابعة في الناتج

رابعاً: علم الصفر Zero Flag ZF
يصبح في حالة واحد منطقي عندما يكون ناتج آخر عملية حسابية أو منطقية يساوي الصفر.
يصبح في حالة صفر منطقي عندما يكون ناتج آخر عملية حسابية أو منطقية لا يساوي الصفر.
خامساً: علم الإشارة Sign Flag SF
يكون علم SF في حالة واحد منطقي Set إذا كانت نتيجة آخر عملية حسابية عدداً سالباً.
يكون علم SF في حالة صفر منطقي Reset إذا كانت نتيجة آخر عملية حسابية عدداً موجباً.
مصطلح: من إحدى طرق تمثيل الأعداد السالبة في الكمبيوتر هي اعتبار الخانة الأخيرة مخصصة للإشارة و بما أن البايت مكون من ثمانية خانات فسيتم اقتطاع الخانة الأخيرة منه من أجل الإشارة فإن احتوت على القيمة واحد فإن الخانات السبعة الباقية هي عدد ثنائي سالب أما إذا احتوت على القيمة صفر فإن الخانات السبعة المتبقية ما هي إلا عدد موجب.
و بذلك يكون SF هو نسخة عن الخانة الأخيرة في الناتج عند اعتماد هذا النظام لتمثيل الأعداد السالبة.
لاحظ أنه انطلاقاً من هذا المبدأ في التمثيل يمكننا تمثيل المجالات التالية من الأعداد:
من أجل بايت واحد من –128 إلى +127
من أجل بايتين من –32768 إلى +32767
سادساً: علم الطفحان Overflow Flag OF
يكون في حالة واحد منطقي عندما لا تتسع النتيجة في المكان المخصص لتخزينها أي تتجاوز القدرة التخزينية، أما إذا لم تكن النتيجة خارج المجال المحدد فإن OF يبقى في حالة الصفر المنطقي.
يحدث الطفحان في الحالات التالية:
1) جمع أعداد موجبة كبيرة.
2) جمع أعداد سالبة كبيرة.
3) طرح عدد موجب كبير من عدد سالب كبير.
4) طرح عدد سالب كبير من عدد موجب كبير.
ملاحظة: جميع الأعلام السابقة ما عدا CF تُقرأ فقط أي لا نستطيع تغيير محتواها لذلك يمكن قراءتها فقط و لا يمكن تغيير محتوياتها بواسطة تعليمات برمجية مباشرة.
المعالج مزود بتعليمات تستطيع اختبار حالة هذه الأعلام لتغيير تتابع تنفيذ البرنامج فمثلاً يمكن اختبار علم ZF=1 كشرط من أجل القفز إلى جزء آخر من البرنامج.
و فيما يلي سنشرح أعلام التحكم:
أولاً: علم الخطوة الوحيدة Trap Flag TF
يوضع بالحالة واحد منطقي عندما نرغب بتنفيذ البرنامج خطوة خطوة و هو مفيد عندما نريد تصحيح برنامجنا و استكشاف مواقع الأخطاء.
ثانياً: علم المقاطعة Interrupt Flag IF
يستخدم من أجل التعبير عن إمكانية أو عدم إمكانية تنفيذ المقاطعة، فيوضع بالحالة واحد منطقي عندما لا نرغب بتنفيذ أي مقاطعة (المقاطعة محجوبة) أما عند وضعه في حالة الصفر المنطقي فإن المقاطعة مسموح بها.
ملاحظة: المقاطعة هي عبارة عن خدمة تؤدي إلى عمل معين فمثلاً المقاطعة 21 و التي من أحد خدماتها العودة إلى نظام التشغيل.
ثالثاً: علم الاتجاه Direction Flag DF
يدل على اتجاه سير العمليات التسلسلية.
عندما يكون في حالة واحد منطقي فإن السلسلة تكون من العنوان الأعلى إلى العنوان الأدنى.

عندما يكون في حالة صفر منطقي فإن السلسلة تكون من العنوان الأدنى إلى العنوان الأعلى.