

تعليمات الضرب والقسمة:

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

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

إن النقطة تعني عملية الضرب العادية، و الرمز S8 يعني متحول مصدر عبارة عن بايت أما الرمز R فيعني باقي القسمة والرمز 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 والتقسيم للأعداد ذات الإشارة فهي مشابهة تماماً للتعليمات السابقة وتُعرف 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 .

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.

مثال 1:

أوجد خرج البرنامج التالي:

Mov Al,08Ch

Mov Bl,04h

Mul Bl

لضرب العددين 08Ch و 04h يمكننا إما تحويلهما إلى النظام العشري ومن ثم ضربهما ببعض وإيجاد الناتج أو يمكننا ضربهما بالنظام الثنائي كما يلي:

$$\begin{array}{r}
 1100\ 1000 \\
 \underline{100*} \\
 00000000 \\
 00000000 \\
 \underline{11001000} \ + \\
 1100100000
 \end{array}$$

بتحويل الناتج إلى النظام الست عشري نجد أنه 320h بالتالي يتم تخزين النتيجة في المسجل AX كما يلي : Ax=320h

من جهة أخرى إن العدد 08Ch إذا قمنا بتحويله إلى النظام العشري نجد أنه 200 d بالتالي فإن ناتج ضرب 4*200 يساوي 800 وهو يساوي 320h.

التطبيق العملي:

ننشئ مشروع جديد ضمن برنامج Crossware Embedded Development Studio ونقوم بكتابة الكود البرمجي ضمنه عند العنوان 0050:0100h كما يلي:

```

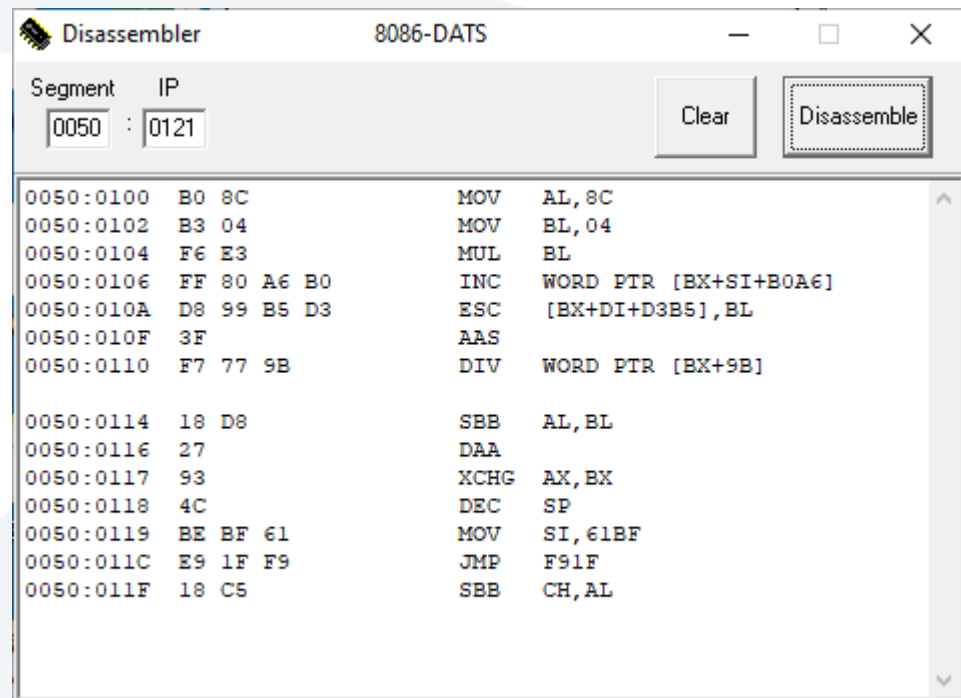
; Initial assembler source file for 8086-DATS

assume cs:CODE, ds:CODE, es:CODE
org     0050h:0100h
; add your source code here

mov al,08ch
mov bl, 04h
mul bl

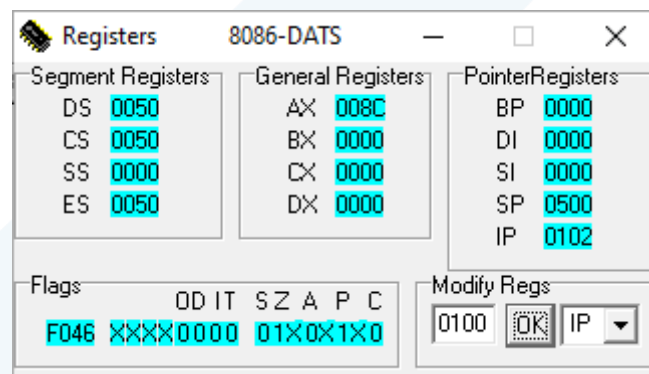
```

نحمل البرنامج ضمن بيئة DATS ونقوم باستعراضه للتأكد



Segment	IP	Address	Hex	Instruction
0050	0121	0050:0100	B0 8C	MOV AL, 8C
0050	0121	0050:0102	B3 04	MOV BL, 04
0050	0121	0050:0104	F6 E3	MUL BL
0050	0121	0050:0106	FF 80 A6 B0	INC WORD PTR [BX+SI+B0A6]
0050	0121	0050:010A	D8 99 B5 D3	ESC [BX+DI+D3B5], BL
0050	0121	0050:010F	3F	AAS
0050	0121	0050:0110	F7 77 9B	DIV WORD PTR [BX+9B]
0050	0121	0050:0114	18 D8	SBB AL, BL
0050	0121	0050:0116	27	DAA
0050	0121	0050:0117	93	XCHG AX, BX
0050	0121	0050:0118	4C	DEC SP
0050	0121	0050:0119	BE BF 61	MOV SI, 61BF
0050	0121	0050:011C	E9 1F F9	JMP F91F
0050	0121	0050:011F	18 C5	SBB CH, AL

نغير قيمة IP إلى 100 ومن ثم ننفذ البرنامج تعليمة تلو الأخرى فنجد ناتج التنفيذ كما يلي:

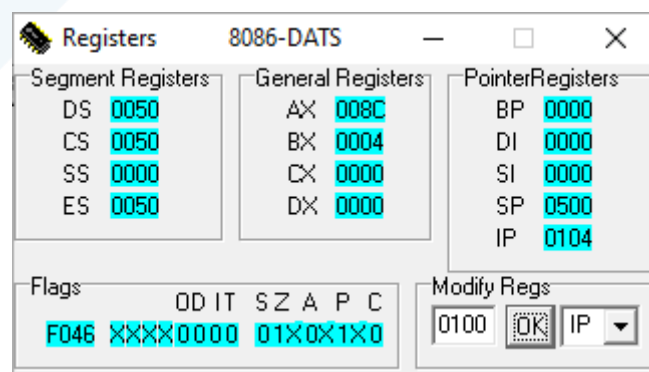


Segment Registers	General Registers	Pointer Registers
DS: 0050	AX: 008C	BP: 0000
CS: 0050	BX: 0000	DI: 0000
SS: 0000	CX: 0000	SI: 0000
ES: 0050	DX: 0000	SP: 0500
		IP: 0102

Flags: 0D IT SZ A P C
F046 XXXX0000 01X0X1X0

Modify Regs: 0100 OK IP

تم تحميل المسجل AX بالقيمة 008Ch بعد تنفيذ أول تعليمة

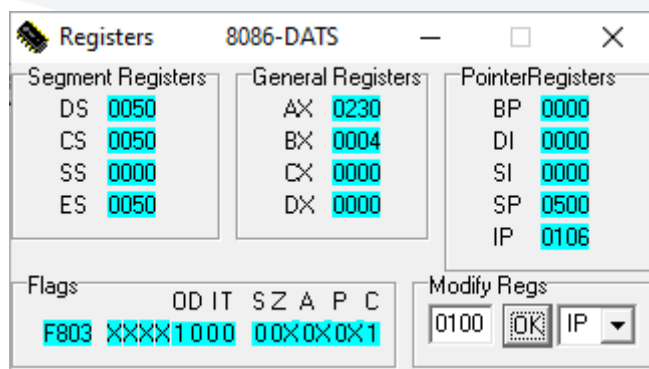


Segment Registers	General Registers	Pointer Registers
DS: 0050	AX: 008C	BP: 0000
CS: 0050	BX: 0004	DI: 0000
SS: 0000	CX: 0000	SI: 0000
ES: 0050	DX: 0000	SP: 0500
		IP: 0104

Flags: 0D IT SZ A P C
F046 XXXX0000 01X0X1X0

Modify Regs: 0100 OK IP

تم تحميل المسجل BL بالقيمة 4 بعد تنفيذ التعليمة الثانية



نتيجة عملية الضرب تم تخزينها في المسجل AX

مثال 2: أوجد ناتج تنفيذ البرنامج التالي:

Mov Ax,ff35h

Mov Bl,04h

IDiv Bl

إن العدد $(FF35)h = (1111111100110101)b$ بالتالي نلاحظ من خانة MSB أن العدد سالب بالتالي نقوم بأخذ المتمم الثنائي له فيكون $(00CB)h = (0000000011001011)b$

نقوم الآن بقسمة 11001011 على العدد 4 أي على 100 فنجد أن الناتج هو 110010 فإذا كتبناه كبايت كامل يكون 00000011 والباقي 00110010

نأخذ المتمم الثنائي للناتج فيكون 11001110 بالتالي هو CEh ويخزن في المسجل AL

نأخذ المتمم الثنائي للباقي فيكون FDh ويخزن في المسجل AH

ملاحظة للتأكد:

$$(ff35)h = (-203)d$$

$$(CE)h = (-50)d$$

$$(FD)h = (-3)d$$

$$-50 = -203/4 \text{ والباقي } -3$$

التطبيق العملي :

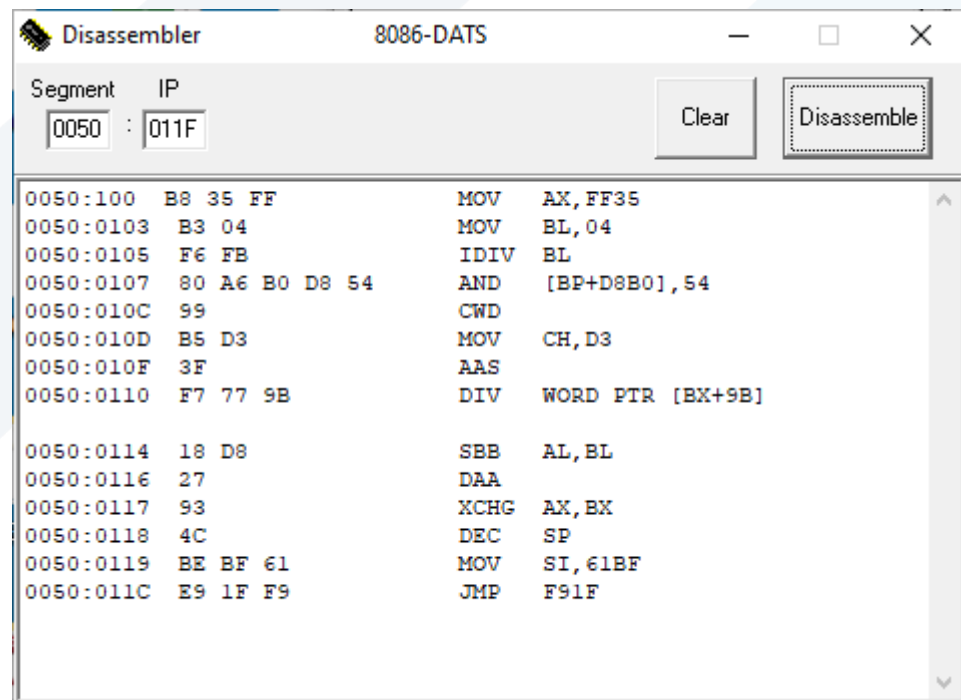
سنقوم في الخطوات التالية بكتابة البرنامج وتنفيذه تعليمة تلو الأخرى كما في المثال السابق

; Initial assembler source file for 8086-DATS

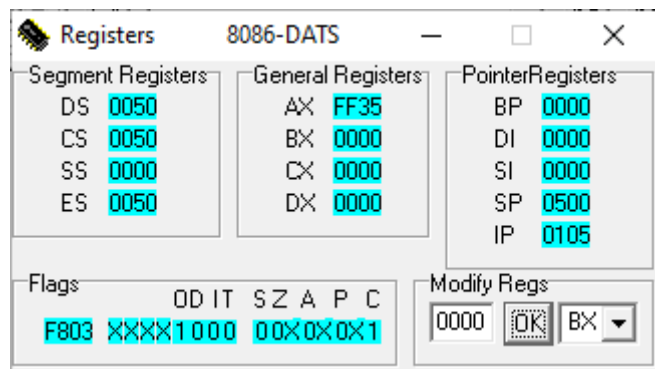
```
assume cs:CODE, ds:CODE, es:CODE
org     0050h:0100h
; add your source code here

mov ax,0ff35h
mov bl, 04h
idiv bl
```

تحميل البرنامج ضمن بيئة DATS



ناتج تنفيذ التعليمات بشكل متسلسل:



Registers		8086-DATS	
Segment Registers		General Registers	
DS	0050	AX	FF35
CS	0050	BX	0004
SS	0000	CX	0000
ES	0050	DX	0000
		Pointer Registers	
		BP	0000
		DI	0000
		SI	0000
		SP	0500
		IP	0105
Flags			
OD IT SZ A P C			
F803 XXXX1000 00X0X0X1			
Modify Regs			
0100 OK IP			

Registers		8086-DATS	
Segment Registers		General Registers	
DS	0050	AX	FDCE
CS	0050	BX	0004
SS	0000	CX	0000
ES	0050	DX	0000
		Pointer Registers	
		BP	0000
		DI	0000
		SI	0000
		SP	0500
		IP	0107
Flags			
OD IT SZ A P C			
F096 XXXX0000 10X1X1X0			
Modify Regs			
0100 OK IP			