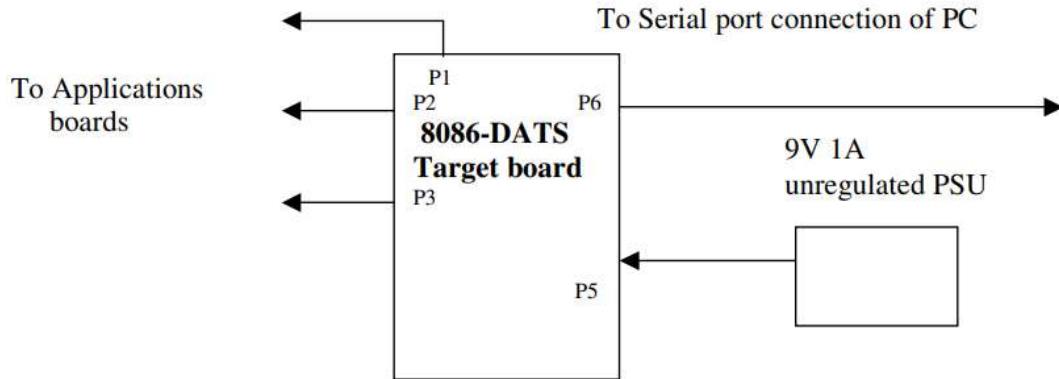


المحاضرة الاولى

مقدمة في استخدام البرمجيات

الهدف من التجربة:

تقوم اللوحة بتبسيط العمل على تدريب المعالج 8086 وطرفياتها حيث يتم وصلها مع الحاسب من خلال المنفذ التسلسلي حيث يتم تحويل الإجراء البرمجي إلى لغة الآلة وترجمته من خلال البرنامج المرفق ومن ثم تحميله إلى ذاكرة الـ RAM الخاصة باللوحة ومن ثم يمكن توصيل اللوحة إلى طرفيات أخرى.



فور تحميل الأجراء يمكن تنفيذه واختباره في النظام حيث يمكن الدخول إليه من قبل المستخدم. يتم تخزين التطبيق على ذاكرة EPROM والقرص على جهاز الحاسب.

خصائص اللوحة:

- المعالج 8086 يعمل بالنمط الأدنى.
- قيادة مولد نبضات الساعة 8284A وهي كريستالة من الكوارتز بتردد 14.7456 ميغاهرتز وساعة المعالج CLK بتردد 4.9152 ميغاهرتز وساعة الطرفية PCLK بتردد 2.4576 ميغاهرتز.

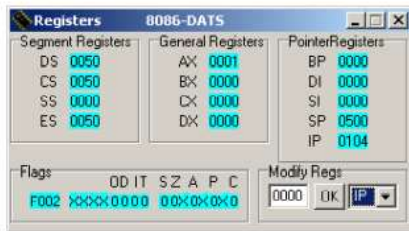
الذاكرة: من نمط EPROM بحجم 64 كيلوبايت و RAM بحجم 16 كيلوبايت.

الطرفيات على اللوحة:

- وحدتي واجهة الربط الطرفية القابلة للبرمجة PPI من النوع 8255A تتمتع بأربعة من البوابة التفرعية ذات الطول 8 بت بخاصية المصافحة.
- المؤقت الزمني القابل للبرمجة PIT من النوع 8253 ويؤمن ثلاثة قنوات عدادات- مؤقتات بطول 16 بت.
- واجهة اتصال قابلة للبرمجة USART من النوع 8251A متزامنة أو غير متزامنة كاملة الاتجاه تعتمد على النمط RS232 (رباعي) من أجل إزاحة المستوى ويتم تحديد معدل البود باستخدام الوحدة 8253.
- متحكم بالمقاطعات PIC من النوع 8259A يؤمن ثمانية مستويات من الأولوية للأجهزة السابقة.

اختبار الإجراء البرمجي:

بعد توصيل اللوحة إلى جهاز الحاسب وتشغيل الطاقة يتم تشغيل تطبيق 8086-DATS حيث سيقوم التطبيق باكتشاف اللوحة:



يقوم البرنامج بالتأكد من توصيل اللوحة والبحث عن جميع الإعدادات وفي حال كانت التوصيلات صحيحة ستظهر النافذة التالية:



في حال عدم اكتشاف اللوحة يمكن إجراء Reset لها ومن ثم اختيار Detect من القائمة .comms. يمكن فحص الوصلات بأي وقت من خلال Check monitor وفي حال عم التطبيق واللوحة بشكل صحيح سيعطينا الرسالة monitor ok.

يتضمن البرنامج الواجهات التالية:



تسمح القائمة EDIT للمستخدم بكتابة الكود البرمجي على ملف نصي يمكن استخدامه مع تطبيقات Cross-Assemblers والتي يتم اعدادها في الملف 86DATs.INI . في حين تسمح القائمة Assemble بفتح نافذة لاختيار ملف assembler المطلوب بناءه . تسمح القائمة Link بعمليات الربط حيث يتم إدخال ملف من نمط object اليها وإخراج ملف من نمط Intel Hex File في حال الرغبة بتحميله للوحة. تسمح اللائحة Download بتحميل ملف Intel Hex File إلى اللوحة البرمجية.

قائمة Debug :

تسمح هذه القائمة بالتحكم بعرض الواجهات التي تعرض حالة الذاكرة والسجلات والوصلات المختلفة:



جَامِعَة
الْمَنَارَة
MANARA UNIVERSITY



تسمح الواجهة Memory للمستخدم بتفحص أو تحديث محتويات الذاكرة حيث عند النقر على هذا الخيار تظهر النافذة التالية:



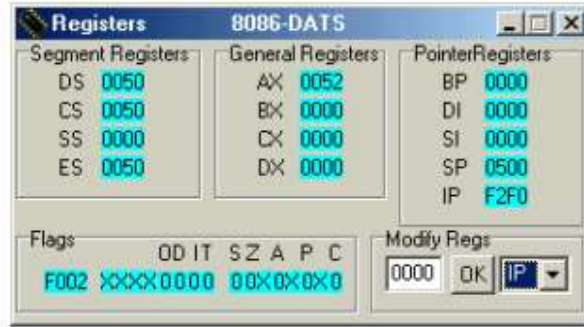
تعرض عناوين البيانات في أقسام المقطع والإزاحة 'Segment' and 'Offset' في حين تخزن البيانات في قسم 'Data'. يتم وضع البيانات في هذه الصناديق بالقيمة السداسية عشر ومن ثم نقر ok. يمكن التنقل بين السجلات ببساطة بنقر الأزرار Next, Previous.

بنفس الطريقة تعرض قائمة السجلات حالة السجلات كما في الشكل التالي:

MANARA UNIVERSITY



جامعة
المنارة
MANARA UNIVERSITY



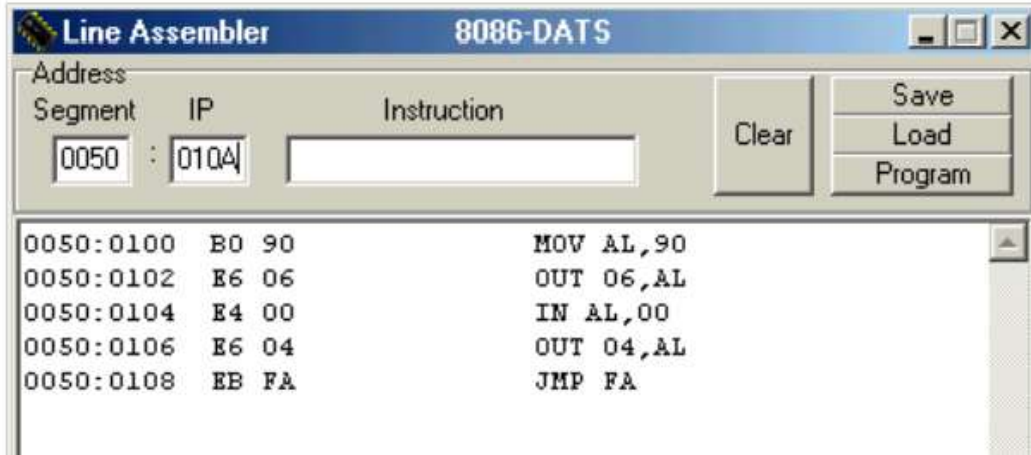
تعرض السجلات في ثلاث اقسام مستقلة, Segment registers, General registers and Pointer registers في حين تعرض الاعلام في القسم السفلي بصورة مستقلة. يمكن التعديل على كل هذه السجلات باستثناء الاعلام من خلال القسم 'Modify Regs' حيث يتم اختيار السجل وإدخال القيمة السداسية عشر ومن ثم نقر Ok.

تسمح قائمة Watches بإدخال لائحة من مواقع الذاكرة والتي نريد ادارتها خلال عمليات debug حيث يتم النقر في صناديق segment and offset وإدخال قيمة سداسي عشرية مقبولة واختيار Add وتستمر العملية لكل العناوين المرغوبة.

:Line Assembler

يسمح هذا القسم بكتابة القسم البرمجي للوحة على شكل قيم سطرية, يجب إدخال العنوان المطلوب تخزين البرنامج له في القسم Segment and Instruction Pointer , في حال كان البرنامج المدخل صحيحاً عندها يتم تحديث العرض لعرض عنوان البدء ويتم ترجمة الإجراء إلى النط hex إلى اللوحة.

المنارة
MANARA UNIVERSITY



The screenshot shows the Line Assembler 8086-DATS window. The title bar reads "Line Assembler 8086-DATS". Below the title bar, there are fields for "Address", "Segment", "IP", and "Instruction". The "Address" field is set to "0050", "Segment" to "010A", and "IP" to "010A". There are buttons for "Clear", "Save", "Load", and "Program". Below these fields is a list of assembly instructions:

```

0050:0100  B0 90      MOV AL,90
0050:0102  B6 06      OUT 06,AL
0050:0104  E4 00      IN AL,00
0050:0106  E6 04      OUT 04,AL
0050:0108  EB FA      JMP FA

```

يجب الانتباه إلى حجم القيم المدخلة للبرنامج وفقاً للشروط المختلفة.

عند ادخال MOV [1000],02 هذا يعني تحميل العنوان 1000 بالقيمة 02 أما التعليمة MOV [1000],0002 فهي تعني تحميل العناوين 1000 و 1001 بالقيمة 0002.

من أجل تحديد نمط المعطيات المستخدم يمكن لنا استخدام التعليمة:

BYTE PTR for byte And WORD PTR for word

على سبيل المثال:

INC BYTE PTR [1000]

INC WORD PTR [1000]

يسمح Disassembler بقراءة الإجراء البرمجي عند العنوان المحدد ومن ثم عرض التعليمات المتعلقة، تسمح القائمة comms بتحديد البوابة التي سيتم الاتصال معها من com1-4 .

تسمح Go User بتشغيل البرنامج المخزن مسبقاً على اللوحة في حين يسمح خيار Single Step بتشغيل الإجراء خطوة تلو الأخرى. تسمح Read Registers بقراءة السجلات على اللوحة وبالتحديد التأكد من تفسير السجلات على اللوحة.

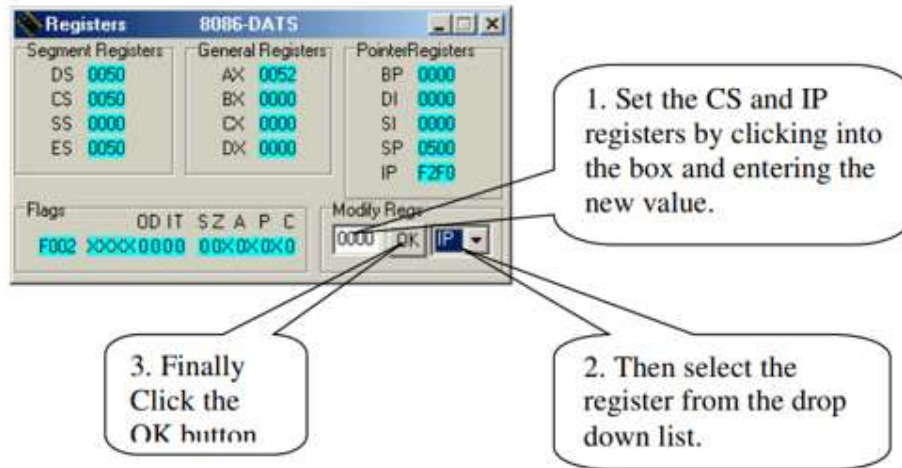
خطوات العمل:

1- بعد توصيل اللوحة نقوم بالنقر على Check Monitor والتأكد من إعطاء الخرج .Monitor OK

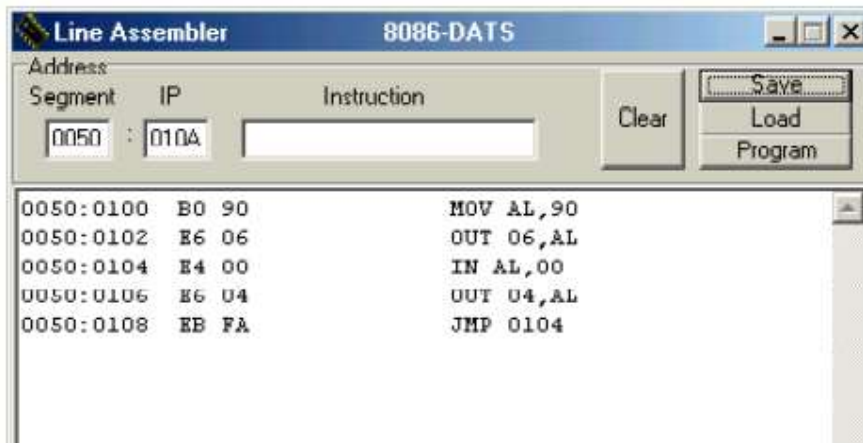
2- نقوم بفتح خيار .Debug من قائمة Line Assembler.

3- العنوان الافتراضي لبدء البرنامج هو 0050:0100 ويتم إدخال العناوين للبرنامج بصورة متسلسلة.

4- يجب التأكد من توافق قيمة IP الموجودة في Line Assembler مع مقابلتها في واجهة .Registers

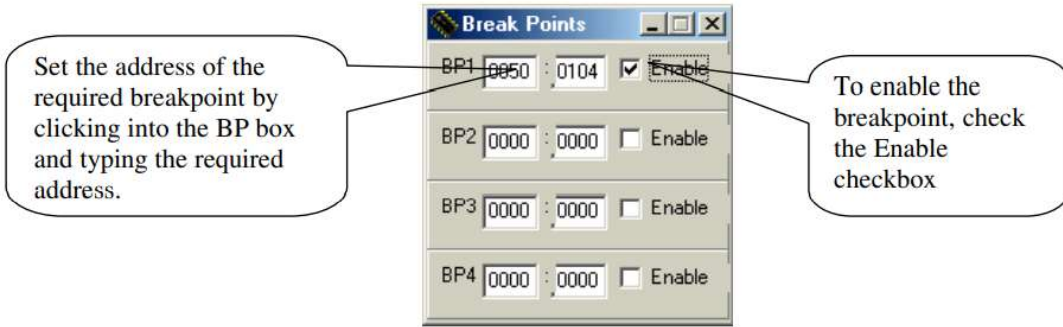


5- نضيف البرنامج التالي:

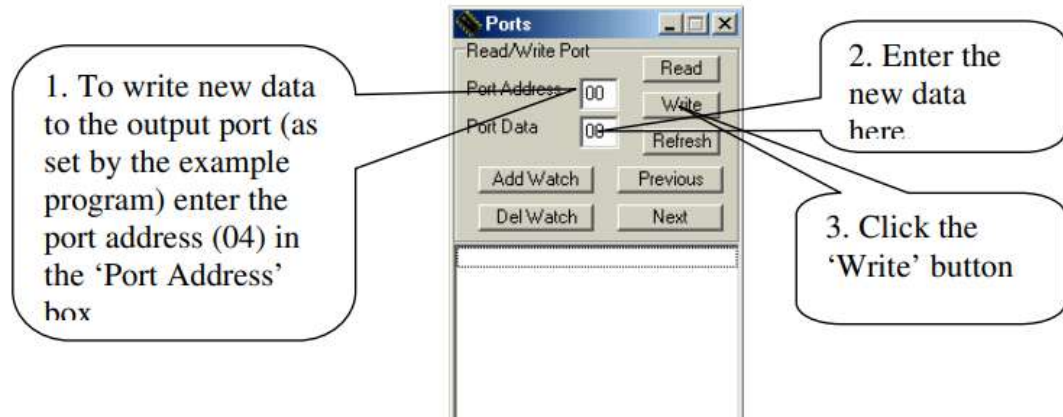


6- نقوم بالنقر على زر الحفظ حيث تظهر نافذة عرض الحفظ مع تحديد تنسيق الملف من النمط .asm. وللتحميل ننقر على زر load. ومن ثم يتم النقر على زر Program.

- 7- يقوم هذا البرنامج بتحديد PPI بحيث تكون البوابة A كدخل والبوابة C كخرج ومن ثم تقوم بصورة مستمرة بقراءة البوابة الأولى وإخراجها على الثانية.
- 8- من أجل إيقاف البرنامج في مراحل محددة يمكن اختيار Breakpoint من القائمة debug كما في النافذة التالية:



- يمكن تحديد حتى أربعة نقاط توقف لملاحظة طريقة سير البرنامج.
- 9- من أجل كتابة بيانات إلى احد المنافذ نقوم بفتح النافذة Ports من القائمة debug كما يلي:



بعد ذلك يتم النقر على خيار single step من أجل تنفيذ البرنامج على شكل خطوات متسلسلة.