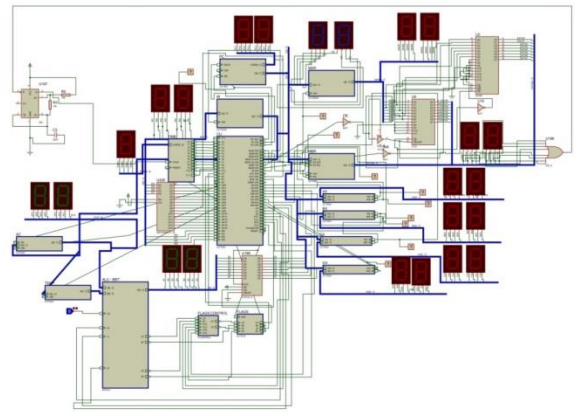


مشروع مادة المترجمات

مترجم لغة Assembly لمعالج 8 بت

```
1  MOV  ACC, 1H
2  MOV  [90H], ACC
3  MOV  RA, ACC ; Move
4  MOV  RB, ACC
5  ADD  ACC, RA
6  XOR  ACC, RB
7  JMP  0H ; Jump
8  HLT
```



إعداد الطلاب:

محمد يمان قلعه جي
عبد الكريم المكتبي
عبد الحميد سيد
محمد حمزة مشهدي

تحت إشراف:

م.أحمد قاضي

مقدمة عن المشروع:

يهدف المشروع إلى تصميم مترجم لغة Assembly خاصة بالمعالج الذي تم تصميمه في مادة بنية الحاسب. يتعرف المترجم على 27 تعليمة خاصة بالمعالج ، ويقوم بكتابة ملف لغة الآلة (Binary) لتنفيذه مباشرة. يتألف المترجم من الأجزاء الرئيسية التالية:

- 1- المحلل اللفظي Lexical Analyzer .
- 2- المحلل القواعدي Syntactical Analyzer .
- 3- المحلل المعنوي Semantic Analysis .
- 4- توليد شيفرة الآلة Binary Generating .

التعليمات التي ينفذها المترجم:

ينفذ المعالج 27 تعليمة أساسية :

الرقم	الترميز (بصيغة HEX)	التعليمة	عمل التعليمة
1	01 02 03 04 06 07 08 90 0B 0C 0D 0E	MOV RX , RX (4 مسجلات و 12 حالة)	نقل محتوى مسجل عام إلى مسجل عام آخر
2	10 11 12 13	MOV ACC , RX (4 حالات)	نقل محتوى مسجل عام إلى المراكم
3	20 24 28 2C	MOV RX , ACC (4 حالات)	نقل محتوى المراكم إلى مسجل عام
4	30	MOV ACC , Immediate	نقل قيمة عددية مباشرة إلى المراكم (يشترط أن تكون القيمة أصغر من 256)
*5	41	MOV ACC , [Memory]	نقل محتوى حجرة ذاكرية إلى المراكم (يشترط أن تكون القيمة أصغر من 251)
**6	42	MOV [Memory] , ACC	نقل محتوى المراكم إلى حجرة ذاكرية (يشترط أن تكون القيمة أصغر من 251 وأكبر من 127)
7	50 51 52 53	ADD ACC , RX (4 حالات)	جمع محتوى مسجل عام إلى المراكم
8	60 61 62 63	SUB ACC , RX (4 حالات)	طرح محتوى مسجل عام من المراكم
9	70 71 72 73	ADC ACC , RX (4 حالات)	جمع محتوى مسجل عام إلى المراكم مع الحمل
10	80 81 82 83	AND ACC , RX (4 حالات)	تطبيق عملية AND المنطقية بين مسجل عام و المراكم و وضعها في المراكم
11	90 91 92 93	OR ACC , RX (4 حالات)	تطبيق عملية OR المنطقية بين مسجل عام و المراكم و وضعها في المراكم
12	A0	NOT	تطبيق عملية NOT المنطقية على المراكم

13	B0 B1 B2 B3	XOR ACC , RX (4 حالات)	تطبيق عملية XOR المنطقية بين مسجل عام و المراكم و وضعها في المراكم
14	C1	STC	تغيير قيمة Carry Flag إلى واحد منطقي
15	C2	CLC	تغيير قيمة Carry Flag إلى صفر منطقي
16	C4	STZ	تغيير قيمة Zero Flag إلى واحد منطقي
17	C8	CLZ	تغيير قيمة Zero Flag إلى صفر منطقي
*18	D0	JMP Memory	القفز إلى حجرة تعليمة في الذاكرة (يشترط أن تكون القيمة أصغر من 251)
*19	EC	JS Memory	القفز إلى حجرة تعليمة في الذاكرة في حال كان العدد الناتج سالب (يشترط أن تكون القيمة أصغر من 251)
*20	E4	JNS Memory	القفز إلى حجرة تعليمة في الذاكرة في حال كان العدد الناتج موجب (يشترط أن تكون القيمة أصغر من 251)
*21	EA	JZ Memory	القفز إلى حجرة تعليمة في الذاكرة في حال كان العدد الناتج يساوي الصفر (يشترط أن تكون القيمة أصغر من 251)
*22	E2	JNZ Memory	القفز إلى حجرة تعليمة في الذاكرة في حال كان العدد الناتج لا يساوي الصفر (يشترط أن تكون القيمة أصغر من 251)
*23	E9	JC Memory	القفز إلى حجرة تعليمة في الذاكرة في حال وجود حمل (يشترط أن تكون القيمة أصغر من 251)
*24	E1	JNC Memory	القفز إلى حجرة تعليمة في الذاكرة في حال عدم وجود حمل (يشترط أن تكون القيمة أصغر من 251)
*25	F1 FB	CALL Memory	استدعاء تابع فرعي في الذاكرة (يشترط أن تكون القيمة أصغر من 251)
26	F2 FB	RET	العودة إلى مكان الاستدعاء المحفوظ في الحجرة FB
27	FF	HLT	نهاية البرنامج

* يجب أن تكون القيمة أصغر من 251 لأن الحجرات من 252 إلى 256 مستخدمة داخلياً في المعالج.

** يجب أن تكون القيمة أكبر من 127 لأن التصميم الداخلي للمعالج لن يسمح بتفعيل الكتابة في الذاكرة إلا عند القيمة 128 فما أكبر.

***** ملاحظة تشمل كافة القيم :**

يجب أن تكون كافة القيم لاتساوي إلى القيمة 10 أو 26 ، لأن المقابل الست عشري لها هو 1A , 0A وهذه المحارف محجوزة من قبل النظام Windows لتعبر عن نهاية الملف المخزن في القرص.

التوابع المستخدمة في المحلل اللفظي:

1- التابع `substring` :

يقوم بتقطيع السلسلة النصية عبر تحديد مكان البداية والنهاية وإعادة السلسلة الجديدة.

الكود المصدري للمحلل اللفظي LEX :

التوابع المستخدمة في المحلل القواعدي:

1- التابع `writetofile`:

دخله عدد ممثل ست عشرياً بشكل مباشر (ذو الشكل 0x00) ، يقوم هذا التابع بكتابة القيمة مباشرة إلى ملف البرنامج `program.bin` بشكل ثنائي.

2- التابع `strtol` :

تابع معرف في لغة ال C ، ويتم استخدامه عبر تضمين الملف الرأس `stdlib.h` ، دخله هو سلسلة نصية مرمزة حسب نظام عددي ما (ست عشري في حالتنا) وخرجه هو عدد من النوع `Long` ، يتم تحويله عبر مفهوم ال Casting الى عدد صحيح `Integer` ليتم استخدامه في المحلل المعنوي.

3- التابع `stringtohex` :

دخله هو سلسلة نصية تعبر عن قيمة ست عشرية، وخرجه هو عدد ممثل ست عشرياً بشكل مباشر (ذو الشكل 0x00) ، ويتم إرسال هذا العدد إلى التابع `writetofile` الذي يقوم بكتابته إلى ملف البرنامج.

4- التابع `main` :

التابع الأساسي الذي تتم عملية الترجمة فيه.

5- التابع `yyerror` :

التابع المسؤول عن عرض الخطأ الحاصل أثناء عملية الترجمة وتفاصيله.

الكود المصدري للمحلل القواعدي YACC :

الملفات المرفقة على القرص :

- 1- الأكواد المصدريّة Lex و Yacc .
- 2- برنامج Hex Editor Neo لمعاينة ملف الخرج المكتوب بالشكل الثنائي.
- 3- ملف تصميم المعالج basic.pdsprj على برنامج Proteus الإصدار 8.0
- 4- برنامج EditPlus .
- 5- ملف شرح المشروع Word + PDF .
- 6- ملفات الإكمال التلقائي على برنامج EditPlus وملفات تلوين السياق.
- 7- الملفات الدفعية (bat) وملفات سكربت Visual Basic (vbs) المستخدمة لتشغيل وضع المحاكاة.