Design Flash Memory Programmer Device

Dhuha Basheer Abdullah **Reyath Z .Mahmood**

Prof.dhuha_basheer@uomosul.edu.iq

College of Computer Sciences and Mathematics
University of Mosul, Iraq

Received on: 06/04/2005 **Accepted on: 09/08/2005**

ABSTRACT

The purpose of this research is to design a ***flash memory programmer device*** consisting of a set of Integrated Circuits (ICs) to meet the needs. These circuits have been connected according to operations (defining, reading, programming and erasing) run on the memory chip.

The research also deals with building a program which programs two types of Flash Memory (***Multi-purpose and Bulk-Erase***) by using the above mentioned hardware design. The program provides the possibility reading these two types of Flash Memory, Programming and Erasing them. The design of a ***Flash Memory Programmer Device*** and the following algorithms particular for its programming have led to a complete system regarding the Hardware and Software.

Keywords: Flash Memory, Integrated Circuits, Bulk-Erase

تصميم جهاز مبرمج الذاكرة الومضية

ضحى بشير عبد الله **رياض زغلول محمود**

كلية علوم الحاسوب والرياضيات، جامعة الموصل

تاريخ استلام البحث: 2005/04/06 **تاريخ قبول البحث: 2005/08/09**

الملخص

إن الغرض من هذا البحث هو تصميم جهاز مبرمج الذاكرة الومضية اذ يتكون الجهاز من مجموعة من الدوائر المتكاملة (ICs) لكي تفي بالغرض . وتم ربط هذه الدوائر حسب خارطة تم وضعها خصيصا لهذا الجهاز مما يخدم جميع العمليات (تعريف ، قراءة ، برمجة ، مسح) التي تجرى على شريحة الذاكرة .

كما تناول البحث بناء برنامج يقوم ببرمجة نوعين من الذاكرة الومضية (متعددة الأغراض ، ذات المسح الكتلي) باستخدام تصميم الجهاز المادي المذكور آنفاً . يقوم البرنامج بتوفير إمكان قراءة هذين النوعين من الذاكرة الومضية وبرمجتهما ومسحهما. إن تصميم جهاز مبرمج الذاكرة الومضية واتباع الخوارزميات الخاصة ببرمجته أدى إلى الوصول إلى نظام متكامل من ناحية المكونات المادية (Hardware) والبرمجية (Software) .

الكلمات المفتاحية: الذاكرة الومضية، الدوائر المتكاملة، المسح الكتلي.

1- المقدمة

إن الذاكرة الومضية (Flash Memory) هي ذاكرة قراءة فقط من عائلة (CMOS) تخزن فيها معلومات (برامج/بيانات) يحتاج إليها نظام الحاسوب دوما عند بداية تشغيل الحاسوب و أثناء تنفيذ أي برنامج بعد التشغيل ذلك إنها تحتوي على برامج الفحص الذاتي لعتاد الحاسوب (POST) وكذلك برامج خدمة المقاطعات (ISRs) الضرورية التي تعد واجهة بين نظام التشغيل وعتاد الحاسوب .

تعد الذاكرة الومضية نوعا من ذاكرات الـ EEPROM غير ان معلوماتها يمكن ان تمسح وتكتب على هيئة قوالب بدلا من البايتات المفردة . مما يجعلها أسرع من الـ EEPROM النظامية . كذلك فأنها تتطلب فولتية اقل لإجراء هذه العمليات ، لذا فان الـ Flash Rom يعد واسع الاستخدام الآن في حفظ معلومات الـBIOS الخاصة بالحواسيب الشخصية[2]. وهذا يمنح مصنعي الـ BIOS الفرصة لتوفير ما يستجد عن طريق الانترنيت ويوفر للمستخدمين إمكان الوصول إلى ميزات جديدة لم تدعم أصلا من الحاسبات الخاصة بهم . ومن المهم ان نحصل على برمجة الذاكرة الومضية لكل BIOS معين. فلا تستخدم أية برامج يستحصل عليها من صديق او من عدد من الشبكات غير المعروفة. ولذلك يجب الذهاب إلى شبكة المصنع نفسه وحيازة الـ BIOS الخاص بالحاسوب الشخصي ، والإصدار ، واللوحة الأم، ورقم الموديل والرقم التسلسلي، وتاريخ الصنع. يجب ان تطبق التعليمات على ورقة و تتبع عن كثب . كما يجب التأكد من أية معلومات بشأن دعم وخزن الـ BIOS السابق الخاص بالحاسوب الشخصي إذا ما تأكد أن شيئا ما قد اخطأ في مساره [17,14,4].

إن تقانة هذا النوع من الذاكرات مهمة في دراسة أنظمة ألحوا سيب الدقيقة ؛ فأجهزة الذاكرة الومضية مشابهة لذاكرات EPROM's بنواحٍ عديدة ، إلا انها تختلف عنها في نواح أخرى مهمة جداً. إن الذاكرات الومضية في الحقيقة تعمل مثل ذاكرات EPROM's تماماً، فهي غير متطايرة وتقرأ مثل الـ EPROM تماما وتبرمج بخوارزمية مشابهة للخوارزمية الخاصة ببرمجة الـ EPROM .[5,4]

أن الاختلاف الرئيس بين الذاكرة الومضية وذاكرة الـ EPROM هو ان خلايا ذاكرتها تمسح كهربائيا ، بدلا من تعريضها لضوء الأشعة فوق البنفسجية . أي ان مصفوفة الخزن للذاكرة الومضية يمكن أن تمسح كهربائيا وتعاد برمجتها ببيانات جديدة . وبخلاف ذاكرة الوصول العشوائي RAM ، فهي تعد ذاكرة غير قابلة للمسح وغير قابلة للكتابة ، فعندما يتم انجاز عملية المسح على

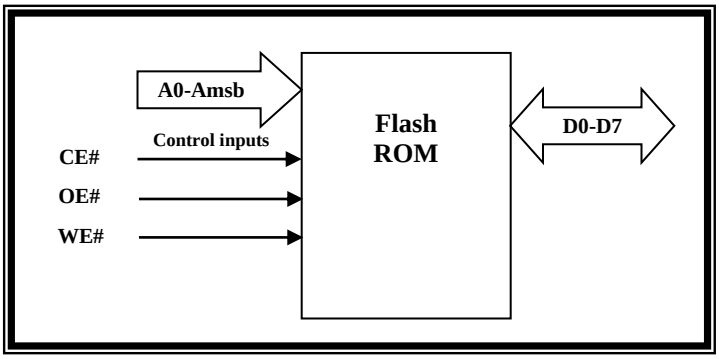
الذاكرة الومضية يتم مسح مصفوفة الذاكرة كليا أو لمقطع كبير من مواقع الخزن و ليس مسح بايت واحد فقط . وفضلا عن ذلك تعد عملية المسح معقدة ويمكن ان تستغرق عدة ثوان . ويمكن أن تتبع عملية المسح هذه عملية كتابة ، وهي دورة برمجة حقيقية ، تحمل البيانات الجديدة إلى مواقع الخزن ، كما تستغرق عملية الكتابة هذه وقتا طويلا عندما تقارن بعدد دورات الكتابة لذاكرة الوصول العشوائي (RAM) [17].

وحتى لو كانت الذاكرة الومضية قابلة للكتابة ، فهي تستخدم غالبا في أنظمة ألحوا سيب الدقيقة لخزن الـ (Firmware) تماما مثل الذاكرات من النوع EPROM's . إلا أن قابليتها المحدودة على إعادة الكتابة والمسح تمكنها من الاستخدام في عدد من التطبيقات المهمة الأخرى إذ تبتغي إعادة كتابة البيانات، ولكن ليس على نحو متكرر[9,3].

إن هناك عددا من الأمثلة لتمثيل جدول غير متطاير (Non-Volatile Lookup Table) وقابل للكتابة فضلا عن أنظمة البرمجيات الداخلية لتحديث الشفرات وسواقات الأجهزة الصلبة (Solid State Drives) [9,8,7].

إن الذاكرات الومضية تعمل بطريقة مشابهة لذاكرة الـ EPROM[6] . ويظهر الشكل (1) مخططاً صندوقياً لجهاز الذاكرة الومضية الذي يشبه تماما المخطط الصندوقي لذاكرة الوصول العشوائي المستقرة (SRAM) . ان خطوط العنوان A_0-A_{msb} وطرف تمكين الشريحة (#CE) وتمكين المخرجات (#OE) تخدم الوظيفة نفسها تماما لكلا الجهازين . أي ان العنوان يلتقط موقع الخزن الذي سيتم الوصول اليه . ويمكن الطرف #CE الجهاز من التشغيل كما يمكن #OE وصول البيانات إلى المخرجات خلال دورات القراءة .

كذلك نرى انهما يختلفان في ناحيتين: الأولى ، تعرف خطوط البيانات في الذاكرة الومضية على أنها ثنائية الاتجاه . والسبب في هذا إن الذاكرة الومضية يمكن ان تستخدم في أي تطبيق إذ تتم الكتابة اليها والقراءة منها . ثانيا ، يتم في الذاكرة الومضية توفير طرف سيطرة أخر ، وهو طرف تمكين الكتابة (#WE) وهذه الإشارة يجب أن تكون في المستوى المنطقي الواطئ ('0') خلال جميع عمليات الكتابة على الشريحة [9,4].



الشكل (1) : المخطط الصندوقي لذاكرة (Flash ROM)

تتوافر أجهزة الذاكرة الومضية مع معمارية المصفوفة وهي مختلفة وعديدة ؛ اذ ترتبط هذه المعماريات بكيفية تنظيم الجهاز لغرض المسح . وقد أشرنا فيما سبق إلى حصول مسح لجميع محتويات الذاكرة او لقالب كبير من مواقع خزن الذاكرة عندما يتم إنجاز عملية المسح على الذاكرة الومضية. هنالك عدة أصناف من الذاكرة الومضية مثل (ذاكرة المسح الكتلي Bulk-Erase وذاكرة قالب الإقلاع Boot-Block وذاكرة الملف الومضي Flash- File). يتم في جهاز ذاكرة المسح الكتلي تنظيم مصفوفة الخزن الكاملة على أنها قالب منفرد. وقتما يتم إنجاز عملية المسح ، فان محتويات جميع مواقع الخزن يتم تنظيفها . وهذه المعمارية مستخدمة في تصميم أجهزة الذاكرة الومضية القديمة [17,1]. تعد ذاكرة قالب الإقلاع وذاكرة الملف الومضي من أجهزة الذاكرة الومضية الأكثر حداثة فهي تضيف (Granularity) إلى عملية البرمجة . ولا ينبغي لمصفوفة الذاكرة أن تمسح بالكامل ، بل يمسح كل من القوالب المستقلة لمواقع الخزن على نحو منفصل. من الملاحظ ان القوالب الخاصة بجهاز قالب الإقلاع غير متناسقة في الحجم، إذ ثمة قالب صغير يعرف بقالب الإقلاع. يهدف إلى خزن شفرة الإقلاع للنظام، يتبعه قالبان صغيران يسميان (Parameter Block) واستعمالهما المقصود هو خزن معايير النظام المعينة ، مثل جدول تهيئة النظام او زمن بحث[13,12] . أخيرا ثمة عدد من قوالب الذاكرة وهي اكبر بكثير من السابقة تعرف بالقوالب الرئيسة تخزن فيها شفرة الـ (firmware). تستخدم أجهزة قالب الإقلاع في مجموعة من التطبيقات التي تتطلب سعة ذاكرة اصغر او تستفيد من القوالب غير المتناسقة ، ويعرف احد هذه التطبيقات ببرامج النظام الداخلي , ففي نوع كهذا من

التطبيق يتم تثبيت شفرة الصندوق المستخدمة لبدء الحاسوب الدقيق في جزء قالب الإقلاع للذاكرة الومضية [17,5]. أما أجهزة الذاكرة الومضية لمعمارية الملف الومضي فتختلف عن أجهزة قالب الإقلاع في كون مصفوفة الذاكرة مقسمة إلى قوالب ذات أحجام متساوية ، ولهذا السبب فانها مرتبة على نحو متناسق . ويستخدم هذا النوع من المصفوفة على نحو رئيس في تصميم اجهزة ذات كثافة عالية . اذ تستخدم اجهزة الذاكرة الومضية ذات الكثافة العالية في تطبيقات تتطلب قدراً كبيراً من الشفرات والبيانات التي سيتم خزنها فيها مثل الـ Flash Disk أو Flash Memory . وتعد الذاكرة الومضية متعددة الأغراض (Multi-Purpose Flash) من نوع ذاكرة الملف الومضي [17,16,11,10].

2- الوصف المادي لجهاز مبرمج الذاكرة الومضية

تم تصميم جهاز مبرمج يقوم ببرمجة مختلف أنواع ذاكرات القراءة فقط وفي مقدمتها الذاكرة الومضية (Flash Memory) ، وقد تم اختيار عدد من الدوائر المتكاملة المستخدمة بكثرة وفي اغلب التطبيقات الكهربائية لكي تفي بالغرض . ان جميع الدوائر في أعلاه هي دوائر إلكترونية ومنطقية معول عليها في التصميم الخاص بموضوع البحث، ومصادر هذه الدوائر شركات مشهورة مصنعة لهذه الدوائر . وتستخدم في كثير من التطبيقات الخاصة بدوائر الملاءمة التي تمكن إضافتها إلى جهاز الحاسوب الشخصي عن طريق الممرات التوسعية المختلفة . والشكل (2) يوضح مخطط الربط لجهاز مبرمج الذاكرة الومضية.

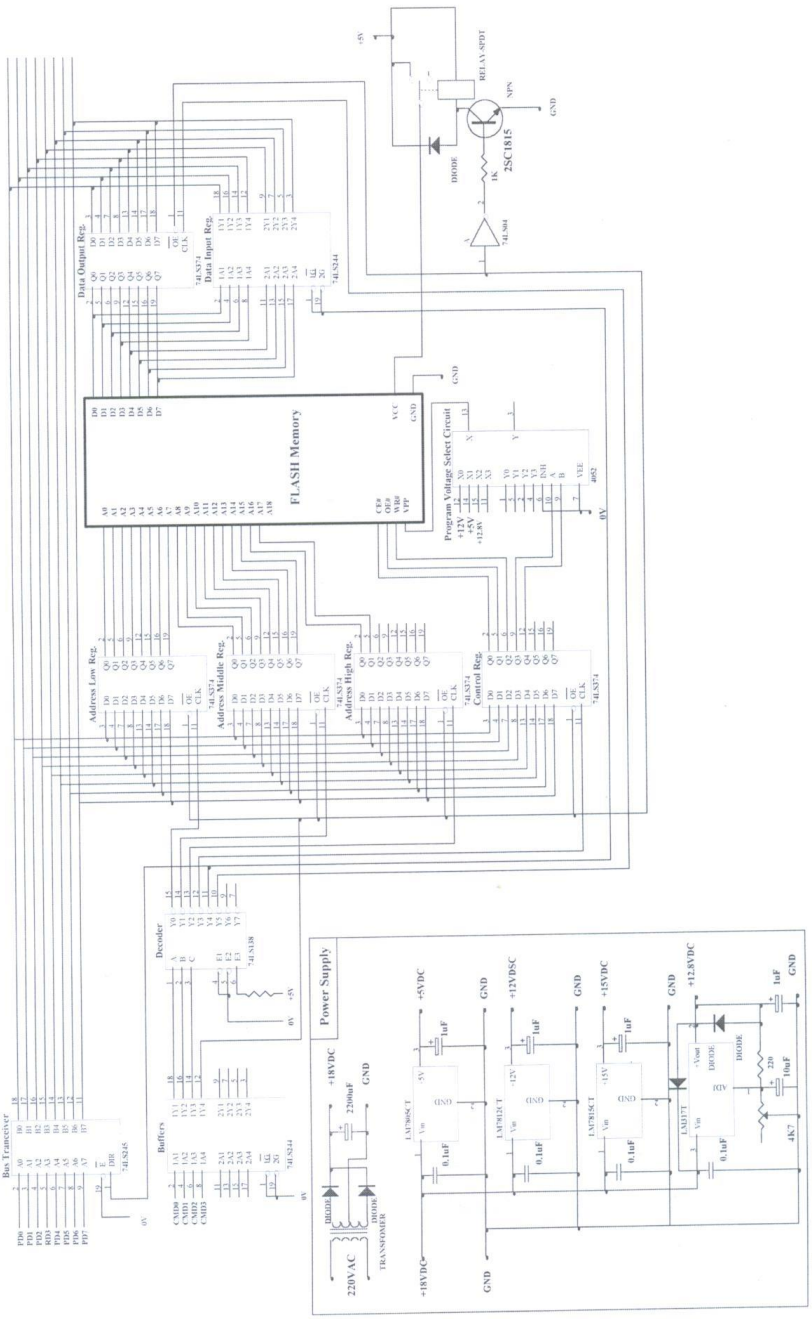
2-1استخدام منفذ الطابعة المتوازي

تم استخدام منفذ الطابعة المتوازي لكي يربط بالجهاز المبرمج عن طريق كابل البيانات الخاص لهذا الغرض . يتكون منفذ الطابعة المتوازي من ثلاثة منافذ فرعية ، الاول يسمى منفذ البيانات (Data Port) ويكون باتجاهين (Bi-direction) ، ويتم عن طريقه اخراج البيانات بعرض ثمانية أرقام ثنائية بين المحيط الداخلي للحاسوب وأجهزة الإخراج (مثل الطابعة او أي تطبيق آخر). كما يتم عن طريقه إدخال بيانات بالعرض نفسه من أي جهاز خارجي إلى داخل الحاسب الشخصي .

أما المنفذ الثاني فيسمى منفذ الأوامر (Command Port) ويستخدم عادة لإرسال الأوامر الخاصة بالطباعة إلى جهاز الطابعة المتصل بالحاسوب الشخصي ، وقد تم استخدامه في التصميم الخاص بجهاز المبرمج لإرسال إشارات السيطرة الخاصة باختيار السجلات المراد الكتابة فيها أو

القراءة منها ، والمبنية ضمن اللوحة الخاصة بالجهاز المبرمج وقد تم استخدام جميع إخراجات هذا المنفذ وعددها أربعة أرقام ثنائية لهذا الغرض .

والمنفذ الثالث الذي يسمى منفذ حالة الطابعة(Status Port) ليتم عن طريقه تعرف الحاسوب الشخصي على حالات الطابعة في أثناء عمليات الطباعة ، ولم يتم استخدام هذا المنفذ مطلقا في الجهاز المصمم وذلك لعدم الحاجة إليه . إن إدخال أي نوع من البيانات من الجهاز المبرمج إلى داخل الحاسوب الشخصي قد تم من خلال منفذ البيانات (Data Port) الخاص بمنفذ الطابعة المتوازي [15].



الشكل (2): مخطط الربط لجهاز مبرمج الذاكرة الومضية

2-2 المكونات المادية لجهاز مبرمج الذاكرة الومضية

2-2-1 مجهز القدرة

يتكون مجهز القدرة من محولة خافضة تقوم بتخفيض الفولتية المتناوبة من 220V AC إلى 18V AC ، وبما أن كل جهاز إلكتروني يجب تغذيته بفولتية مستمرة ، لذا يجب تحويل الفولتية الخارجة من المحولة من الفولتية المتناوبة إلى الفولتية المستمرة ، وذلك باستخدام موحدات التيار وقد اخترنا موحداً كامل الموجة بسبب وجود طرف وسطي في إخراج المحولة المستخدمة . لقد تم ربط أطراف المحولة الثلاثة بهذا الموحد المتكون من ثنائيين بلوريين إذ يمرر الأول الفولتية الموجبة عند أحد أطراف إخراج المحولة في أول 180^o من الطور ، أما الثاني فيقوم بتمرير الفولتية الموجبة في الجزء الثاني من الطور . بعد ذلك نحصل على فولتية مستمرة قدرها 18V DC ولكنها غير مرشحة لذا يجب ربط إخراج الموحد بمتسعة الكتروليتية ذات سعة عالية نسبيا مثلا (2200μF) حتى نحصل على فولتية مستمرة قريبة جدا من فولتية البطارية . ويمكن الإشارة هنا إلى أنه كلما زادت سعة المتسعة أقترب التيار الخارج من المجهز من تيار البطارية ، أي ، كلما كان التيار المحسوب من المجهز كبيرا يجب زيادة سعة المتسعة المربوطة في مجهز القدرة .

2-2-2 منظمات الفولتية

تم استخدام عدة منظمات للفولتية لتجهيز مكونات الجهاز المبرمج بالفولتيات المناسبة لكل جزء .

*** منظم الفولتية 7805**

تم استخدام دوائر متكاملة من عائلة TTL (Transistor-Transistor-Logic) وهذه العائلة يجب تغذيتها بفولتية 5V + نسبة سماح 5% ، لذا تم استخدام منظم الفولتية الثابت ذي الرقم (7805) الذي يقوم بتنظيم الفولتية الداخلة اليه وتحويلها الى فولتية 5V+ منظمة على نحو ممتاز . وبما ان التيار الذي يتم سحبه من الدائرة الالكترونية يكون في حدود 1A لذا استخدم هذا النوع من المنظمات الذي يجهز فولتية قدرها 5V+ وبتيار اخراج قدره 1.5A او اكثر بقليل .

*** منظم الفولتية 7812**

في عملية برمجة او مسح الذاكرة الومضية التي تحتاج الى فولتية برمجة عند الطرف V_{PP} قدرها 12V ، نحتاج الى مجهز لهذه الفولتية المستمرة ؛ لذا تم استخدام منظم الفولتية ذي الرقم (7812) الذي يقوم بتحويل الفولتية المستمرة التي تدخل اليه الى فولتية مستمرة منظمة قدرها 12V لتغذية دبوس V_{PP} الخاص بفولتية البرمجة لهذه الذاكرة .

*** منظم الفولتية 7815**

عندما يراد برمجة او مسح الذاكرة الومضية التي تحتاج الى هذا الدبوس يجب ان يجهز بفولتية قدرها 12V عند عمليات البرمجة / المسح ، أما في حالة وضع الذاكرة في طور القراءة فيجب ان تكون فولتية V_{PP} تساوي 5V ، لذا يجب استخدام دائرة المركز لاختيار فولتية V_{PP} ، وهذا النوع من المركزات يكون تماثليا إذ له القدرة على تركيز الفولتيات المستمرة عند إخراج واحد ، وبما انه يقوم بتركيز إحدى فولتيتي V_{PP} 12V عند طور البرمجة / المسح و 5V عند طور القراءة لذا يجب تجهيز هذا المركز MUX بفولتية قدرها 15V ، ومن ثمَّ تم استخدام المنظم ذي الرقم 7815 للقيام بهذه المهمة .

*** منظم الفولتية LM 317**

لغرض تطوير جهاز المبرمج برمجة ذاكرات من نوع PROM أو EPROM فأن فولتية البرمجةV_{PP} لهذا النوع من الذاكرات عند عملية البرمجة يجب أن تكون 12.9V $\rightarrow$ 12.7 ، المثالي (12.8V) ، وبما ان منظم الفولتية (7812)لا يجهز فولتية اكثر من 12V فقد تم استخدام منظم الفولتية القابل للمعايرة ذي الرقم LM317 ، ومعايرته عن طريق المقاومة المتغيرة المتصلة بين أحد أطراف هذا المنظم الثلاثة المسمى (Adj) والأرضي ، حتى تساوي فولتية إخراجه 12.8V+ تقريبا وهي الفوليتة اللازمة لعملية برمجة PROM و EPROM .

ومن الجدير بالذكر أن فولتية إدخال كل منظم يجب ان تكون اكبر من فولتية الإخراج (فولتية التنظيم) بـ 2V ولا تتعدى فولتية الإخراج Vout+16، لذا فان المحولة المختارة لفولتية إخراج 18V تفي لهذا الغرض.

3-2-2 دائرة اختيار فولتية البرمجة

تـم اسـتخدام الـدائرة المتكاملـة ذات الـرقم CD4052 وهـي مـن عائلـة CMOS (Complementary of Metal Oxide Semiconductor) والتي تمثل دائرة اختيار لأربع قنوات تماثلية لكي تتصل إحداها بإخراج الدائرة . وقد تم اختيار إحدى دوائر الاختيار الأربع قنوات بسبب الحاجة إلى فولتيات Vpp وعددها ثلاث وهي 5V ، 12V و 12.8V أي ثلاث فولتيات.أما الجزء الثاني من الدائرة المتكاملة فهو غير مستخدم لعدم الحاجة اليه .

4-2-2 سجلات العنوان

تم استخدام ثلاث دوائر متكاملة ذوات الرقم 74LS374 إذ أن كل واحدة منها تحتوي على ثماني مراجيح من نوع D-type Flip Flop والتي تستخدم لتجهيز خطوط العنوان الخاصة بالذاكرة المراد القراءة منها او الكتابة فيها او مسحها بالعنوان المطلوب ، والسبب في استخدام ثلاث دوائر متكاملة ذوات الرقم 74LS374 هو ان جهاز المبرمج المصنع يقوم ببرمجة الذاكرات ذوات الحجم 1M-bit او 2M-bit (128k-byte و 256k-byte) وهذان الحجمان يمكن الوصول إليهما عن طريق خطوط عنونة تقدر بـ (17-bit و 18-bit) وحيث أن كل دائرة متكاملة تتكون من 8-Bit Register لذا استخدمنا ثلاث منها . كما ويمكن تطوير الجهاز لكي يقوم ببرمجة أحجام اكبر بسبب وجود عدد من الـ Bits غير مستخدمة (Free) في السجل الثالث.

5-2-2 سجل إخراج البيانات

يستخدم هذا السجل لتجهيز الذاكرة المراد برمجتها أو مسحها بالبيانات المطلوبة ، حيث تم استخدام الدائرة المتكاملة 74LS374 أيضا لهذا الغرض ، نخزن البيانات المرسلة من منفذ الطابعة المتوازي عن طريق منفذ البيانات وتثبيتها عند خطوط البيانات الخاصة بالذاكرة المراد برمجتها او مسحها لحين إجراء إحدى هاتين العمليتين (البرمجة/المسح) عليها .

6-2-2 سجل إدخال البيانات

لغرض قراءة البيانات من الذاكرة المراد القراءة منها ، تم استخدام سجل إدخال والمتمثل بالدائرة المتكاملة ذات الرقم 74LS244 التي تحوي في داخلها ثماني بوابات ثلاثية الحالة والتي تستخدم عادة بوصفها سجل إدخال في اكثر التطبيقات ، كما أن هناك فائدة كبيرة لهذا السجل في هذا التصميم تتمثل بمنع حدوث النزاع على خطوط البيانات بين الخطوط الخاصة بالذاكرة المراد برمجتها وخطوط البيانات الخاصة بمنفذ الطابعة المتوازي.

2-2-7 سجل السيطرة

تم استخدام الدائرة المتكاملة ذات الرقم 74LS374 كذلك بوصفها سجل إخراج من مهامه إخراج قيم إشارات السيطرة المتمثلة بـ CE#, OE#, WE# للذاكرة المراد برمجتها او القراءة منها ، هذه الإشارات تخرج عن طريق b_2,b_1,b_0 من هذا السجل أما b_3 , b_4 فيستخدمان بوصفهما مدخلات اختيار فولتية Vpp الخاصة بالدائرة المتكاملة CD4052 .

2-2-8 مرسل/مستقبل الخطوط

بما أن أحد محاور تصميم جهاز المبرمج هو ربط طريق منفذ الطابعة المتوازي عبر كابل بيانات لمسافة بعيدة نسبيا ، فقد تم استخدام الدائرة المتكاملة ذات الرقم 74LS245 لكي تقوم بمهمة عزل خطوط البيانات الخاصة بالجهاز المصمم وبيانات منفذ البيانات الخاص بمنفذ الطابعة المتوازي ، أما الفائدة الأخرى من الدائرة المتكاملة في أعلاه فهي تقوية إشارات الخطوط وزيادة Fan-Out لمنع حدوث اي خطأ في أثناء عملية النقل بين منفذ الطابعة المتوازي وجهاز المبرمج.

2-2-9 دائرة عزل خطوط السيطرة

تم استخدام الدائرة المتكاملة ذات الرقم 74LS244 لعزل خطوط السيطرة الخاصة بمنفذ السيطرة الخاص بمنفذ الطابعة المتوازي والخطوط الخاصة بجهاز المبرمج ، كما أن لها فائدة تقوية الإشارات الخاصة بالسيطرة وزيادة Fan-Out .

2-2-10 محلل الشفرة

تم استخدام الدائرة المتكاملة المعروفة والمشهورة ذات الرقم 74LS138 التي تمثل 8-3 الموزع/محلل الشفرة وتقوم بعملية حل الشفرات لثلاثة أرقام ثنائية الوزن كما تم استخدامها لوظيفة التوزيع للسيطرة على عملية تمكين إشارات السيطرة الخاصة بجميع سجلات الإدخال والإخراج فضلا عن سجل السيطرة .

2-3عملية القراءة من الذاكرة الومضية

تتطلب عملية القراءة من الذاكرة عدة مراحل إذ يجب أولا إرسال عنوان الموقع المراد القراءة منه إلى جهاز المبرمج، وبما أن تصميم الجهاز قد تم ليقوم ببرمجة ذاكرات بحجم (128K-

(128K-byte,256K-byte) وهذه الأحجام تتطلب خطوط عنوان (A_0-A_{16}, A_0-A_{17}) على الترتيب ، فيجب تجزئة العنوان إلى ثلاثة أجزاء ، يتكون الجزء الأول من (A_0-A_7) والجزء الثاني من (A_8-A_{15}) أما الجزء الثالث فيتكون من (A_{16}) إذا كان حجم الذاكرة (128K-byte) ، و(A_{16}-A_{17}) إذا كان حجم الذاكرة (256K-byte) . وقد تمت تجزئة العنوان إلى هذه الأجزاء بسبب أن منفذ البيانات الخاص بمنفذ الطابعة المتوازي الذي سوف يتم إرسال أجزاء العنوان من خلاله إلى جهاز المبرمج يكون بعرض ثمانية أرقام ثنائية .

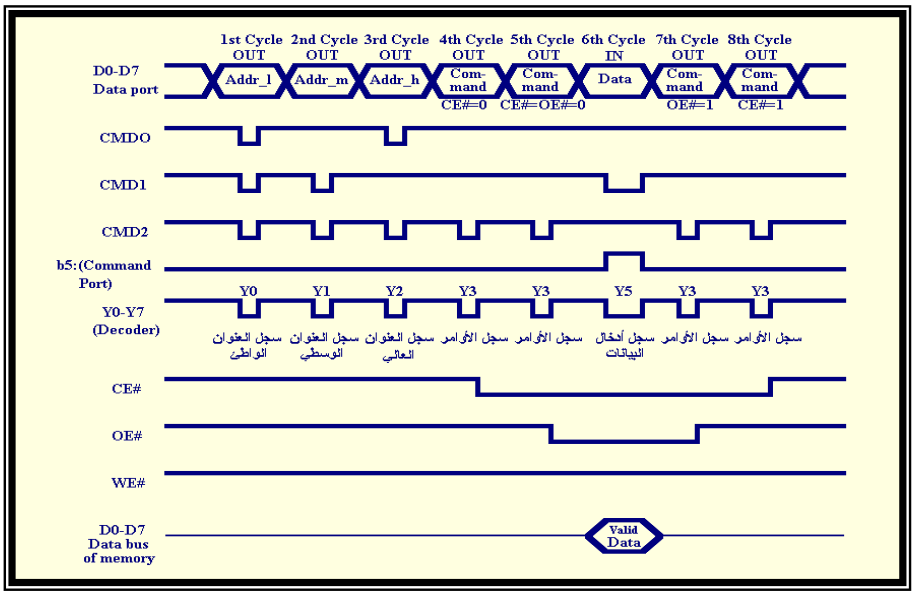
يجب إرسال الأجزاء الثلاثة من العنوان إلى سجلات العنوان الخاصة بالذاكرة المراد برمجتها ويتم خزن الجزء الواطئ منه في السجل الأول الذي يحدد عنوانه عن طريق أول ثلاثة أرقام ثنائية من منفذ الأوامر الخاص بمنفذ الطابعة المتوازي والمتمثلة بـ (b_2,b_1,b_0) كما يتم خزن الجزء الوسطي في السجل الثاني ، أما الجزء العالي فيخزن في السجل الثالث. والجدول (1) يوضح عناوين كل من السجلات الخاصة بجهاز المبرمج .

بعد إتمام عملية إرسال العنوان يتم تحديد مسار البيانات لمنفذ البيانات الخاص بمنفذ الطابعة المتوازي عن طريق تثبيت قيمة (b_5=1) في سجل الأوامر، ثم يتم إرسال إشارة السيطرة #CE عن طريق سجل الأوامر الخاص بالجهاز المصمم لكي تطبق على الشريحة المراد القراءة منها ثم ترسل إشارة #OE إلى الشريحة نفسها لتمكين خطوط البيانات الخاصة بالذاكرة من إخراج البيانات إلى المحيط الخارجي وبعد ذلك يتم تمكين إخراج سجل إدخال البيانات بإرسال عنوانه من خلال محلل الشفرة. تنساب البيانات الخارجة بعد ذلك في خطوط البيانات الخاصة بجهاز المبرمج ومن الجدير بالذكر أن من الضروري بعد جميع العمليات أعلاه أن يتم إرجاع قيم إشارات السيطرة الخاصة بالذاكرة المراد القراءة منها إلى القيمة غير الفعالة. والشكل (3) يمثل المخطط الزمني لعملية القراءة من الذاكرة الومضية والخاص بجهاز المبرمج.

الجدول (1): عناوين السجلات الخاصة بجهاز المبرمج

السجل	سجل الأوامر لمنفذ الطابعة المتوازي		
	CMD2 B_2	**CMD1 B_1**	**CMD0 B_0**

سجل جزء العنوان الواطئ (A_0-A_7)	0	0	0
سجل جزء العنوان الوسطي (A_8-A_{15})	0	0	1
سجل جزء العنوان الوسطي (A_{16} : 010) ، (A_{16}-A_{17}:020)	0	1	0
سجل الأوامر	0	1	1
سجل إخراج البيانات	1	0	0
سجل إدخال البيانات	1	0	1
غير مستخدم	1	1	0
غير مستخدم	1	1	1



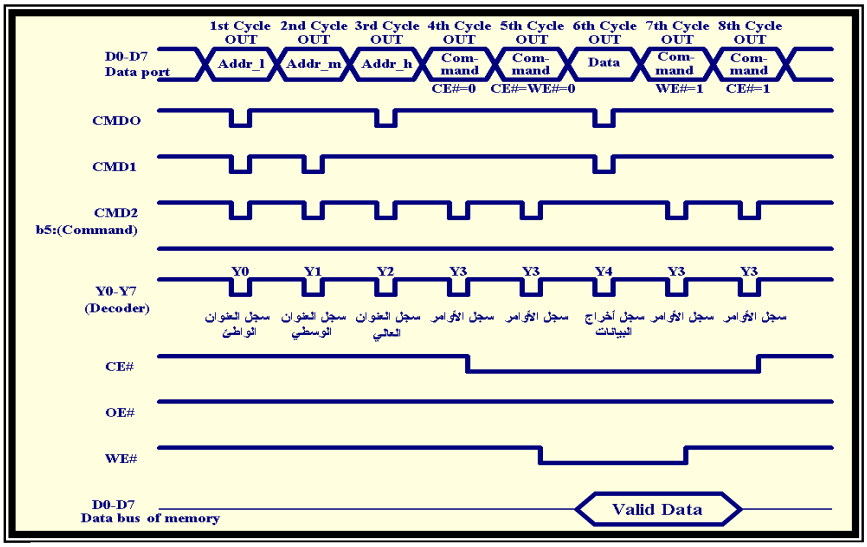
الشكل (3): المخطط الزمني لعملية القراءة من الذاكرة الومضية الخاص بجهاز المبرمج

4-2 عملية الكتابة في الذاكرة الومضية

يتم إرسال العنوان الخاص بالموقع المراد الكتابة فيه بالطريقة نفسها تماما والمتبعة في عملية القراءة من الذاكرة ، بعد ذلك يتم تحديد فولتية البرمجة (Vpp) المناسبة للذاكرة المراد الكتابة فيها عن طريق (b_3,b_2) من سجل الأوامر الخاص بجهاز المبرمج إذ تدخل هذه القيمة إلى طرفي أختيار المركز ، أما الخطوة التالية فتتمثل بإرسال البيانات المراد كتابتها في الذاكرة إلى سجل إخراج البيانات بعد جعل قيمة (b_5=0) في منفذ الأوامر الخاص بمنفذ الطابعة المتوازي لكي يكون اتجاه البيانات إلى الخارج. عند خروج هذه البيانات وانسيابها في خطوط البيانات التابعة لجهاز المبرمج يتم خزنها في سجل إخراج البيانات المتصلة إدخالاته بخطوط البيانات الخاصة بالجهاز ، وتكون

إخراجات هذا السجل متصلة بخطوط البيانات الخاصة بالذاكرة المراد كتابتها فيها ، يلي هذه العمليات إرسال إشارة تمكين الشريحة #CE عبر سجل الأوامر الخاص بجهاز المبرمج ليتم بعد ذلك إرسال إشارة تمكين الكتابة #WE عبر السجل نفسه لكي تتم الكتابة في الذاكرة .

ومن الجدير بالذكر أن عملية الكتابة هذه تشمل كتابة الأوامر الخاصة بالذاكرة المراد برمجتها أو البيانات التي تتم برمجتها في الذاكرة . وقد تم إرجاع جميع قيم إشارات السيطرة الخاصة بالذاكرة ألى القيمة غير الفعالة بعد الانتهاء من هذه العملية. ويمكن الإشارة هنا إلى استخدام (b_3) من منفذ الأوامر الخاص بمنفذ الطابعة المتوازي لجعل جميع الخطوط الخاصة بإخراجات السجلات الخاصة بجهاز المبرمج والمتصلة بالذاكرة المراد برمجتها ومن ضمنها طرفي التغذية فعالة أو خاملة لكي نستطيع فصل الذاكرة عن المأخذ الخاص بها بأمان عند تبديلها بأخرى .وبعبارة أخرى: إنه من الممكن جعل الذاكرة عائمة في أي وقت عن طريق (b_3) السابق ذكره . والشكل (4) يمثل المخطط الزمني لعملية الكتابة في الذاكرة الومضية والخاص بجهاز المبرمج.



الشكل (4): المخطط الزمني لعملية الكتابة في الذاكرة الومضية والخاص بجهاز

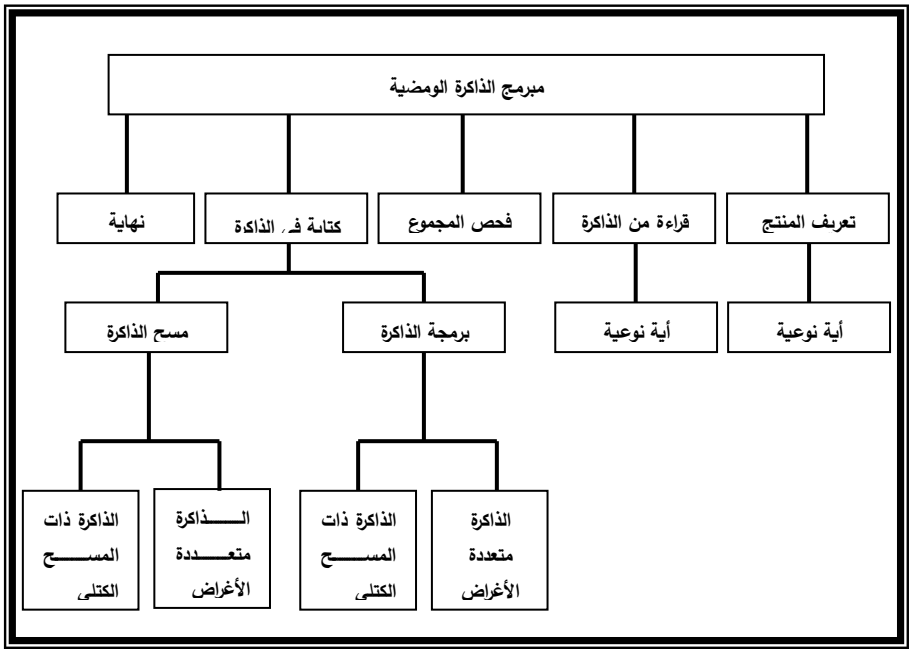
3- الوصف البرمجي لجهاز المبرمج

تم استخدام لغة (C++) لبناء البرنامج والدوال الخاصة به لما تتمتع به هذه اللغة من مرونة عالية في التعامل مع البيانات بالنظام السداسي عشر وكذلك سهولة التعامل مع موانئ

الإدخال والإخراج وباستخدام دوال مكتبية بسيطة وغير معقدة فضلا عن توافر إمكانات عالية في حل أغلب المشاكل التي قد تظهر في اللغات الأخرى.

1-3 الهيكل العام للبرنامج

أن الهيكل العام للبرنامج يتكون من عدة أجزاء وكما موضح في الشكل (5).



الشكل (5) المخطط العام لبرنامج مبرمج الذاكرة الومضية

1-1-3 تعريف المنتج Product ID:

*** الدالة ( ) product_id:**

تستخدم هذه الدالة للكشف عن الذاكرة الومضية متعددة الاغراض MPF والتي تحمل الرقم 39SF020 .

الخوارزمية :

1.استدعاء الدالة Write-byte لكتابة القيمة (AAH) في الموقع (5555H).

2.استدعاء الدالة Write-byte لكتابة القيمة (55H) في الموقع (2AAAH).

3. استدعاء الدالة Write-byte لكتابة القيمة (90H) في الموقع (5555H).

4.استدعاء الدالة read-byte لقراءة البايت الأول الذي يمثل رمز الشركة المصنعة من العنوان (0000H).

5.استدعاء الدالة read-byte لقراءة البايت الاول الذي يمثل رمز الشركة المصنعة من العنوان (0001H).

6.استدعاء الدالة write-byte لكتابة القيمة (AAH) في الموقع (5555H).

7. استدعاء الدالة write-byte لكتابة القيمة (55H) في الموقع (2AAAH).

8. استدعاء الدالة write-byte لكتابة القيمة (F0H) في الموقع (5555H).

ملاحظة: * الخطوات 1 , 2 , 3 تستخدم للدخول إلى طور تعريف المنتج .

* الخطوات 6 , 7 , 8 تستخدم للخروج من طور تعريف المنتج .

9. إرجاع المتغير return_status الذي يحمل نجاح العملية عندما تكون قيمته True أو فشلها عندما تكون قيمته false .

*** الدالة write-byte (a-h, a-m, a-l, dat) :**

تستخدم هذه الدالة لكتابة بايت واحد في العنوان المحدد بأجزاء العنوان الثلاثة المدخلة إلى هذه الدالة .

الخوارزمية :

1.استدعاء الدالة ( ) send-address .

2.استدعاء الدالة ( ) write-data .

*** الدالة send_address (addr_h,addr_m,addr-l):**

تستخدم هذه الدالة لإرسال العناوين الخاصة بالذاكرة المراد القراءة منها أو الكتابة فيها.يتم إرسال هذه العناوين على ثلاث مراحل ، يتم في كل مرحلة منها خزن جزء من العنوان في أحد سجلات العنوان للجهاز المبرمج .

الخوارزمية :

1. إرسال القيمة (08H) إلى منفذ الأوامر ، لإلغاء تفعيل جميع إخراجات محلل الشفرة الخاص بعملية الكتابة في السجلات الخاصة بجهاز المبرمج.
2. إرسال جزء من العنوان الواطئ (addr-l) الى منفذ البيانات لكي يتم خزنه في السجل الخاص به.

3. إرسال القيمة (0FH) الى منفذ الأوامر لتفعيل اشارة strobe الخاصة بالسجل الاول .
4. إعادة الخطوة الأولى (1) .
5. إرسال جزء العنوان الوسطي (addr-m) الى منفذ البيانات لكي يتم خزنه في السجل الخاص به .
6. إرسال القيمة (0EH) الى منفذ الأوامر لتفعيل اشارة strobe الخاص بالسجل الثاني .
7. إعادة الخطوة الأولى (1) .
8. إرسال جزء من العنوان العالي (addr-h) الى منفذ البيانات لكي يتم خزنه في سجله الخاص.
9. إرسال القيمة (0DH) الى منفذ الأوامرلتفعيل اشارة strobe الخاص بالسجل الثالث.
10. إعادة الخطوة الأولى (1) .

*** الدالة write-data (dat):**

تستخدم هذه الدالة لعملية الكتابة في الموقع المحدد سابقا من قبل الدالة send_address.

الخوارزمية :

1. ارسال القيمة (08H) إلى منفذ الأوامر لتأكيد إلغاء تفعيل إخراجات محلل الشفرة .
2. إرسال البيانات (dat) إلى منفذ البيانات المراد كتابتها في الذاكرة .
3. إرسال القيمة (0AH) إلى منفذ الأوامر لخزن هذه البيانات في سجل اخراج البيانات لجهاز المبرمج .
4. إعادة الخطوة الأولى (1) .
5. إرسال القيمة (0FH) إلى منفذ البيانات لكي تخزن في سجل الأوامر الخاص بجهاز المبرمج ، جعل قيمة #CE و #OE و #WE عند الحالة المنطقية العالية .
6. استدعاء الدالة clk() لتوليد نبضة الكتابة في سجل الأوامر الخاص بجهاز المبرمج .
7. إرسال القيمة (0EH) إلى منفذ البيانات ، (تمكين إشارة #CE) الخاصة بالذاكرة .
8. إعادة الخطوة السادسة .
9. إرسال القيمة (0AH) (تمكين إشارة #WE)الخاصة بالذاكرة .
10. إرسال القيمة (0EH) إلى منفذ البيانات لإلغاء تمكين إشارة #WE (الحصول على الحافة الصاعدة اللازمة لعملية الكتابة في الذاكرة) .
11. إعادة الخطوة السادسة (6) .

12. إرسال القيمة (0FH) إلى منفذ البيانات لإلغاء تمكين إشارة #CE .
13. إعادة الخطوة السادسة (6) .
14. إعادة الخطوة الأولى (1) .

*** الدالة ()clk:**

تستخدم هذه الدالة لإرسال نبضة للحصول على حافة صاعدة يحتاج اليها سجل الاوامر الخاص بالجهاز المبرمج لخزن إشارات السيطرة الخاصة بالذاكرة المراد القراءة منها أو برمجتها .

الخوارزمية :

1. إرسال القيمة (08H) إلى منفذ الأوامر (الحصول القيمة المنطقية العالية).
2. إرسال القيمة (0CH) إلى منفذ الأوامر (للحصول على القيمة المنطقية الواطئة).
3. إرسال القيمة (08H) مرة أخرى إلى منفذ الأوامر(للحصول على الحافة الصاعدة).

3-1-2 القراءة من الذاكرة الومضية Read FLASH:

*** الدالـة ( )read_flash**

تستخدم هذه الدالة لقراءة محتويات اي نوع من انواع الذاكرات وخزن البيانات المقروءة في ملف محدد .

الخوارزميـة :

1. قراءة اسم الملف المراد قراءة البيانات من الذاكرة وخزنها فيه .
2. تكـرار عمليـة القـراءة مـن الـذاكرة بعـدد مـن المـرات مسـاو لحجمهـا باسـتخدام الدالـة read-byte.

*** الدالة ([f-n[10]) f-name:**

تستخدم هذه الدالـة لتحديد وقراءة اسم الملف الذي سوف تتم القراءة منـه لغرض خزن محتوياته في الذاكرة المراد برمجتها أو كتابة البيانات في الملف نفسه التي تتم قراءتها من الذاكرة.

الخوارزمية :

1. قراءة اسم الملف من لوحة المفاتيح ، يتكون اسم الملف من 11 حرفاً على الأكثر، وثمانية أحرف لاسم الملف وثلاثة أحرف للامتداد.
2. إرجاع اسم الملف إلى الدالة التي قامت باستدعاء هذه الدالة.

*** الدالة read-byte (a-b, a-m, a-l) :**

تستخدم هذه الدالة لقراءة بايت واحد من العنوان المحدد بأجزاء العنوان الثلاثة المدخلة إلى هذه الدالة .

الخوارزمية:

1. استدعاء الدالة send-address( ) .
2. استدعاء الدالة read-data( ) .
3. إرجاع قيمة dat إلى الدالة المستدعية .

*** الدالة read-data ( ):**

تستخدم هذه الدالة لعملية القراءة من أي نوع من أنواع الذاكرات التي يراد القراءة منها، ذلك إن عمليات القراءة من هذه الأنواع متطابقة تماما .

الخوارزمية :

1. إرسال القيمة (08H) إلى منفذ الأوامر لتأكيد إلغاء تمكين إخراجات محلل الشفرة.
2. إرسال القيمة (8FH) إلى منفذ البيانات لكي تخزن في سجل الأوامر الخاص بالجهاز المبرمج، (جعل قيمة #CE و #OE و #WE عند الحالة المنطقية العالية) ، كذلك جعل b_7=1 من هذا السجل لتحديد اتجاه البيانات ، (جعل إخراجات سجل إخراج البيانات الخاص بالجهاز المبرمج سائبا).
3. استدعاء الدالة clk لتوليد نبضة الكتابة في سجل الأوامر الخاص بالجهاز المبرمج .
4. إرسال القيمة (8FH) إلى منفذ البيانات لكي تخزن في سجل الأوامر (تمكين إشارة #CE) الخاص بالذاكرة .
5. إعادة الخطوة الثالثة (3) .
6. إرسال القيمة (0CH) إلى منفذ البيانات لكي تخزن في سجل الأوامر (تمكين إشارة القراءة #OE) .
7. إعادة الخطوة الثالثة (3) .

8. إرسال القيمة (28H) إلى منفذ الأوامر لمنفذ الطابعة (إلغاء تمكين إخراجات الشفرات) مع تحديد اتجاه البيانات الخاص بمنفذ البيانات (جعل حركة البيانات داخلة) .
9. إرسال القيمة (2BH) إلى منفذ الأوامر لكي يتم تمكين إخراج سجل إدخال البيانات الخاص بالجهاز المبرمج .
10. بعد انسياب البيانات في خطوط البيانات الخاصة بالجهاز المبرمج تتم قراءتها عبر منفذ البيانات باستخدام الأمر dat= inportb(0X378) .
11. إرسال القيمة (28H) الى منفذ الأوامر لالغاء تمكين اخراجات محلل الشفرة.
12. إرسال القيمة (8EH) إلى منفذ البيانات لكي تخزن في سجل الأوامر (جعل قيمة OE#=1) .
13. إعادة الخطوة الثالثة (3).
14. إرسال القيمة (8FH) إلى منفذ البيانات لكي تخزن في سجل الاوامر (لجعل قيمة CE#=1).
15. إعادة الخطوة الثالثة (3).
16. إرسال القيمة (0FH) إلى منفذ البيانات لتمكين مخرجات سجل اخراج البيانات.
17. إعادة لخطوة الثالثة (3).
18. إرسال القيمة (08H) إلى منفذ الأوامر لإلغاء تمكين إخراجات محلل الشفرة وكذلك إعادة اتجاه حركة البيانات من خلال منفذ البيانات.

1-2-3 الكتابة في الذاكرة الومضية Write FLASH:

إذا كان نوع الذاكرة MPF: 39SF02 فيتم استدعاء الدالة .()MPF_program لبرمجة الذاكرة بالكامل .أما إذا كان نوع الذاكرة ذات المسح الكتلي BE:28F020 فيتم استدعاء الدالة BE-program(er).

*** الكتابة في الذاكرة الومضية المتعددة الأغراض (MPF:39SF020):**

تتم الكتابة في هذا النوع من الذاكرات باستخدام الدالة ()MPF-Program التي تقوم ببرمجة الشريحة ، (Sector) بعد الآخر.

*** الدالة ()MPF-Program:**

تستخدم هذه الدالة لبرمجة الذاكرة الومضية متعددة الأغراض بأكملها (Sector) بعد الآخر.

الخوارزمية:

1. استدعاء الدالة f- name لقراءة اسم الملف المراد القراءة منه من لوحة المفاتيح .
2. تكرار عملية استدعاء الدالة program_one_sector لـ 64 مرة لملء الذاكرة بأجمعها بالبيانات المراد برمجتها .

*** الدالة program_one_sector(sector,f1)**

تستخدم هذه الدالة لبرمجة كتلة واحدة محددة بالعنوان (sector) المدخل الى هذه الدالة. يجب ان تمسح الكتلة بالكامل قبل اجراء عملية البرمجة .

الخوارزمية:

1. استدعاء الدالة erase_one_sector لمسح الكتلة ابتداء بالعنوان (sector) .
2. تكرار استدعاء الدالة program_one_byte لـ (4096) مرة لملء الذاكرة بالبيانات المراد برمجتها.

*** الدالة erase-one-sector(sect):**

تستخدم هذه الدالة لمسح كتلة متكونة من 4k-byte من الذاكرة ابتداء بالموقع sect .

الخوارزمية:

1. استدعاء الدالة write-byte لكتابة الامر (AAH) في الموقع (5555H) .
2. استدعاء الدالة write-byte لكتابة الامر (55H) في الموقع (2AAAH) .
3. استدعاء الدالة write-byte لكتابة الامر (80H) في الموقع (5555H) .
4. استدعاء الدالة write-byte لكتابة الامر (AAH) في الموقع (5555H) .
5. استدعاء الدالة write-byte لكتابة الامر (55H) في الموقع (2AAAH) .
6. استدعاء الدالة write-byte لكتابة الامر (30H) في الموقع (sect) .
7. استدعاء الدالة c-t-r للكشف عن اكتمال عملية مسح المقطع الداخلية للشريحة .

*** الدالة program-one-byte(adh,adm,adl,dt) :**

تستخدم هذه الدالة لبرمجة بايت واحد (dt) في العنوان المحدد باجزاء العنوان الثلاثة المدخلة الى هذه الدالة .

الخوارزمية:

1. استدعاء الدالة write-byte لكتابة الأمر (AAH) في الموقع (555H).
2. استدعاء الدالة write-byte لكتابة الأمر (55H) في الموقع (2AAAH).
3. استدعاء الدالة write-byte لكتابة الأمر (A0H) في الموقع (5555H).
4. استدعاء الدالة write-byte لكتابة البايت (dt) في الموقع المحدد بأجزاء العنوان الثلاثة المدخلة إلى هذه الدالة .
5. استدعاء الدالة c-t-r للكشف عن اكتمال عملية البرمجة الداخلية للبايت.

*** الدالة c-t-r (a-h, a-m, a-l):**

وظيفة هذه الدالة هي نفسها في الدالة السابقة .

الخوارزمية:

1. استدعاء الدالة read-byte() لقراءة البايت الذي تمت كتابته لاحقا من العنوان نفسه الذي كتبت فيه .
2. حجب جميع الـ bits الخاص بالبايت عدا bit-6 .
3. تكرار عملية القراءة من الموقع نفسه الى ان نحصل على الـ bit نفسه في المرة الاولى . عند ذاك تكون عملية الكتابة قد اكتملت داخليا .

- **الكتابة في الذاكرة الومضية ذات المسح الكتلي (BE:28F020):**

تتم الكتابة في هذا النوع من الذاكرات باستخدام الدالة BE-program() التي تقوم ببرمجة الذاكرة بالكامل.

*** الدالة BE-Program(er):**

تستخدم هذه الدالة لبرمجة الذاكرة الومضية ذات المسح الكلي . اعتمد في هذه الدالة على الخوارزمية الخاصة ببرمجة هذه الذاكرة من الشركة المصنفة .

الخوارزمية:

1. استدعاء الدالة f-name لقراءة اسم الملف المراد القراءة منه في لوحة المفاتيح .
2. تهيئة العنوان (address=0) وتهيئة فولتية البرمجة (Vpp=12V) .
3. تهيئة عداد النبضات (plscnt=0) .
4. قراءة byte من الملف المحدد في الخطوة (1) .
5. استدعاء الدالة write-byte لكتابة القيمة (40H) في أي موقع .
6. استدعاء الدالة write-byte لكتابة البايت الذي تمت قراءته من الملف في العنوان المحدد.
7. اعطاء زمن تأخير قدره 10 مايكروثانية.
8. استدعاء الدالة write-byte لكتابة القيمة (C0H) في أي موقع .
9. إعطاء زمن تأخير قدره 6 مايكروثانية ومن ثم استدعاء الدالة read-byte لقراءة بايت من العنوان نفسه الذي تمت كتابة البايت فيه في النقطة (6) .
10. زيادة عداد النبضات بـ(1) .
11. عند عدم مطابقة البايت الذي تمت برمجته في الخطوة (6) للبايت الذي تمت قراءته في الخطوة (9) وكانت قيمة عداد النبضات (plscnt) أقل من (25) يتم الانتقال إلى الخطوة (5).
12. إذا كان العنوان اقل من 40000H اعد الخطوات السابقة ابتداء من الخطوة (3) .
13. استدعاء الدالة write-byte لكتابة القيمة (00H) في أي موقع .
14. إرجاع قيمة Vpp=5V .
15. يتم إرجاع القيمة (1) عند نجاح عملية البرمجة أو القيمة (1) عند فشلها .

4-1-3 مسح الذاكرة الومضية Erase-FLASH:

إذا كان نوع الذاكرة MPF:39F020 فيتم استدعاء الدالة erase-entitre-chip() . أما إذا كانت الذاكرة من نوع BE:28F020 فيتم استدعاء الدالة BE-erase().

- **مسح الذاكرة الومضية المتعددة الأغراض (MPF:39SF020)**

*** الدالة erase-entire-chip() :**

الخوارزمية:

تستخدم هذه الدالة لمسح محتويات مصفوفة الذاكرة بأكملها .

1. استدعاء الدالة write-byte لكتابة القيمة (AAH) في العنوان (5555H).
2. استدعاء الدالة write-byte لكتابة القيمة (55H) في العنوان (2AAAH).
3. استدعاء الدالة write-byte لكتابة القيمة (80H) في العنوان (5555H).
4. استدعاء الدالة write-byte لكتابة القيمة (AAH) في العنوان (5555H).
5. استدعاء الدالة write-byte لكتابة القيمة (55H) في العنوان (2AAAH).
6. استدعاء الدالة write-byte لكتابة القيمة (10H) في العنوان (5555H)
7. استدعاء الدالة d-p أو c-t-r للكشف عن اكتمال عملية المسح الداخلي للشريحة بأكملها.

- **مسح الذاكرة الومضية ذات المسح الكتلي (BE:28F020)**

*** الدالة BE-erase():**

الخوارزمية:

1. تهيئة فولتية البرمجة (Vpp=12V) .
2. استدعاء الدالة be-program لخزن القيمة (0) في جميع المواقع الخاصة بالذاكرة واستلام المتغير suc الذي يمثل نجاح عملية الكتابة أو فشلها .
3. إذا لم يتم إنجاز الخطوة (2) بنجاح يتم الخروج من الدالة بفشل العملية . (عملية المسح) .
4. تهيئة العنوان (address=0) .
5. تهيئة عداد النبضات (plscnt=0) . والمتغير (stat=0) الذي يمثل حالة التسلسل لعمليات الكتابة .
6. إذا كانت قيمة (stat=1) فيتم الانتقال إلى الخطوة (10) .
7. استدعاء الدالة write-byte لكتابة القيمة (20H) في أي موقع .
8. استدعاء الدالة write-byte لكتابة القيمة (20H) في أي موقع ايضا .
9. إعطاء زمن تأخير قدره 10 ms.
10. استدعاء الدالة write-byte لكتابة القيمة (A0H) في أي موقع .
11. إعطاء زمن تأخير قدره 6 µ second .
12. قراءة محتويات الموقع الذي تمت عملية مسحه .

13. مقارنة البايت المقروء في الخطوة (12) بالقيمة (FFH). يتم الانتقال إلى الخطوة (7) عند عدم المطابقة بعد زيادة عداد النبضات (plscnt) بـ (1) ومادامت قيمة عداد النبضات اقل من 3000 .وإلا يتم الخروج من الدالة بفشل العملية . (عملية المسح) .
14. زيادة العنوان (eaddress) بـ (1).
15. جعل قيمة المتغير (stat=1).
16. الانتقال إلى الخطوة (10) حتى يتم مسح الشريحة بأكملها (eaddress=40000H) .
17. استدعاء الدالة write-byte لكتابة القيمة (00H) في أي موقع.
18. إعادة قيمة (Vpp=5V) .
19. يتم إرجاع القيمة (1) عند نجاح عملية المسح او القيمة (1) عند فشلها .

3-1-5 فحص المجموع CHECHSUM:

*** الدالة check_sum (c-s):**

تستخدم هذه الدالة لجمع محتويات الشريحة بأكملها لاكتشاف أية حالة خطأ في عملية نقل البيانات أو عملية البرمجة .

الخوارزمية :

1. طباعة ناتج جمع محتويات الذاكرة على شاشة العرض.
2. انتظار الضغط على أي مفتاح للرجوع الى الدالة المستدعية لهذه الدالة.

3-1-6 النهاية End:

يتم استدعاء الدالة ()off لفصل دبابيس الذاكرة عن جهاز المبرمج لرفعها من المقبس الخاص بها والخروج من البرنامج .

* الدالة : ()off

تستخدم لفصل دبابيس الذاكرة كهربائيا من أي مصدر إشارة سواء أكانت إشارات منطقية أم تغذية.

الخوارزمية :

1. إرسال القيمة (00H) إلى منفذ الأوامر.

2. الانتظار إلى حين الضغط على أي مفتاح ضمن لوحة المفاتيح الخاصة بالحاسوب لكي يتسنى للمستخدم رفع او تبديل الشريحة.

4- الاستنتاجات

تم تصميم جهاز مبرمج الذاكرة الومضية عن طريق وضع خارطة لربط الدوائر المتكاملة . كما تم بناء برنامج لبرمجة نوعين من الذاكرة الومضية (متعددة الأغراض وذات المسح الكتلي) ، إذ وفر البرنامج إمكان قراءة وبرمجة ومسح هذين النوعين من الذاكرة .

وتم تطبيق هذا النظام في عملية استنساخ وبرمجة هذين الصنفين من الذاكرة الومضية والتي تستخدم في اكثر اللوحات الرئيسة للحاسوب . وفيما يلي بعض من هذه الاستنتاجات:

1. قبل أية عملية كتابة في الذاكرة الومضية يجب فك نظام الحماية الموجود داخل الذاكرة الومضية الذي يختلف أسلوبه من نوع إلى أخر وذلك بكتابة عدد من الأوامر الخاصة بهذا الغرض ذلك إن هنالك نظام حماية خاصاً داخل الذاكرة الومضية وضعته الشركة المصنعة لضمان عدم حدوث أية عملية كتابة عرضية في أثناء رفع او خفض القدرة عند إيقاف تشغيل الحاسوب ، ومن دون نظام الحماية المذكور سابقاً قد يتوقف الحاسوب عن العمل ولا يمكن إعادة تشغيله على نحو صحيح بأي شكل من الأشكال إلا بعد رفع هذه الذاكرة من الجهاز و إعادة استنساخها بنسخة مطابقة للنسخة الأصيلة مأخوذة من جهاز من النوع نفسه ، وهذه العملية صعبة جداً .
2. لا يمكن الكتابة في أي موقع ضمن مصفوفة الذاكرة الومضية إلا بعد أن يتم مسح محتويات هذا الموقع او مسح المقطع الذي يضم هذا الموقع او مصفوفة الذاكرة بأكملها.
وتكون عملية المسح في الذاكرة الومضية أسرع من ذاكرة (EEPROM) ذلك أن في الأولى يمكن مسح (Block) كامل او الشريحة بأكملها في وقت واحد في حين لا يمكن في الثانية مسح اكثر من موقع في وقت واحد .
3. إن عددا من الفيروسات التي تتسبب في توقف الحاسبة عن العمل مثل فيروس (جيرنوبل) ليس إلا برنامجاً سهلاً يمكنه الوصول الى أي موقع من مواقع الذاكرة الومضية و تغيير محتوياته بمجرد فك الحماية عن عملية الكتابة في الذاكرة الومضية المخزون فيها النظام (BIOS) بعد معرفة عنوان بداية الذاكرة ضمن خارطة الذاكرة (Memory Map) .

من الجدير بالذكر أن الحاسبات القديمة التي هي من جيل (80486) فما دون لا يمكن إصابتها بفيروس (جيرنوبل) بسبب استخدامها لذاكرات (ROM) القياسية التي لا

يمكن الكتابة فيها وهي متصلة بالجهاز المضيف إذ أنها تحتاج إلى فولتيات عالية في أثناء عملية البرمجة لا تتوافر في مجهز القدرة الخاص بالحاسوب .

4. إن الـ(Flash Disk) هو نوع من أنواع الذاكرة الومضية التي لا تحتاج إلى فولتية برمجة عالية ذلك أنه يستخدم فولتية (+5V) حسب لعملية المسح و البرمجة ، و هذه الإمكانية جعلت هذا الـ(Flash Disk) ممكن وإيصاله بالحاسبة عن طريق منفذ(USB) الذي لا يجهز أكثر من (+5V)للمحيط الخارجي للحاسبة.

5- التوصيات

نوصي مستقبلا بتطوير جهاز المبرمج لغرض برمجة أنواع من ذاكرات القراءة فقط (ROM) ومن أمثلتها (PROM,EPROM,EEPROM) كما يمكن تطوير الجهاز للقيام ببرمجة هذه الأنواع بأحجامها المختلفة ابتداء من اقل حجم إلى أكبر حجم. وكذلك يمكن تطويره لبرمجة نوع أخر من الذاكرات الومضية هي الذاكرة الومضية المتوالية (Serial Flash) ولأن هذا التطوير يتطلب مكونات مادية إضافية كبيرة وتصميم دائرة معقدة للغاية ومركزات وموزعات للإشارات وعوازل إضافية للممرات فقد حالت هذه المتطلبات جميعا فضلا عن ضعف الإمكانات التقنية التي تحتاج إليها دون تطويره في الوقت الحاضر .

المصادر

[1] Advanced Micro Device, 1995, "Am28F020-2Megabit (262, 144×8-Bit) CMOS 12.0 Volt, Bulk Erase Flash Memory", Data-Net 2.0.

[2] Advanced Micro Device,2004, "28F020-256K CMOS Flash Memory", www.futurlec.com/Memory/28F020pr.shtml.

[3] Barr,M., 2001, "Introduction to Memory Types", www.netrino. com /publications/Glossary/Memory Types.htm.

[4] Barr, M., 2001, "Memory Types", Embedded System Programming, Vol. 10; No.1, pp.103-104.

[5] Campbell, B., 2002, "Understanding Types of Memory: Different Types of Memory & Their Uses", www.Buymemory.com/memtypes.htm.

[6] Catalyst Semiconductor, Inc., 1992, "2 Megabit CMOS FLASH MEMORY", Data-Net 2.0.

[7] Douglas, V. H., 1988, "Microprocessors and Interfacing-Programming and Hardware" .McGraw-Hill.

[8] HARRIS Semiconductor, 1992, "CD4052MS CMOS Analog Multiplexer/Demultiplexer", Data-Net 2.0.

[9] IBM Personal computer Company, 1997, "Types of Memory", www.computekcomputing.com/pdf/xbmemtype.pdf.

[10] Intel Corporation, 1994, "Flash Memory", www.Citeseer. ist. psu. edu /context/1257710.

[11] Intel, 1994, "28F020-2048k(256k*8) CMOS FLASH MEMORY", Data-Net 2.0.

[12] Intel, 1995, "28F020 2048k (256x8) FLASH MEMORY", www. lyring.com/DocumentosLyR/EXT/ICs/Memory/2MbitFlashEPROM. PDF.

[13] Jeff R.,2000, "SST39SF020 flash used for BIOS", www. iopener. home.fastolfe.net/ihack/0048.htm.

[14] Morris, M. M., Charles R. Kime, 2001, "Logic and computer design Fundamentals", Second Edition Updated, Printice Hall.

[15] Slone, A., 1999, "Computer Communication", McGraw-Hill.

[16] SST, 1998, "2Megabit (256x8) Multipurpose Flash", www. katalogi. iele. polsl.gliwice.p1/upload/39SF020.pdf.

[17] Tyson, J.,2004, "How Rom works", www. howstuffworks. com/ com /.htm.