الفصل الحادى عشر 11

دراسة لباقى أطراف المعالج من خلال

تطبيق: التحكم فى إشارة مرور

Control of a traffic light

obeikandl.com

11-1 مقدمة

إنه فى الكثير من الأحيان وعند ذكر التطبيقات التى تستخدم الميكروكومبيوتر للتحكم فى أى عملية صناعية يتبادر إلى ذهننا فورا الميكروكومبيوتر بصورته المركبة والمعقدة حيث الشاشة ولوحة المفاتيح والطابعة ووحدة التحكم المركزى cpu والكمية الهائلة من الذاكرة التى قد تصل إلى الكثير من الميجابايتات ، فى حين أن العملية من الممكن أن تكون أبسط من ذلك بكثير كما سنرى . سنحاول فى هذا الفصل أن نقدم عرضا مفصلا وشرحا دقيقا لدائرة المعالج على كارت اليكترونى واحد one board وهذا الكارت يمكن استخدامه فى الكثير من أغراض التحكم ومنها على سبيل المثال عملية التحكم فى متغيرات تنك تقطير المياه التى ذكرناها فى مقدمة الفصل العاشر وسوف نستخدم هذه الدائرة فى هذا الفصل فى عملية التحكم فى إشارة مرور رباعية كتطبيق على ذلك .

11-2 تركيب الدائرة

إن الميكروكومبيوتر بمعناه العام والشامل وكما ذكرنا فى الفصل الأول من هذا الكتاب يتركب من شاشة للعرض ولوحة مفاتيح ووحدة تحكم مركزى cpu وكمية من الذاكرة تقل أو تكثر على حسب متغيرات وأوضاع كثيرة . إن الكثير من التطبيقات وبالذات التى تستخدم المعالج فى أغراض التحكم لا تحتاج إلى كل هذه الإمكانيات ، فما فائدة شاشة العرض مثلا فى دائرة نريد تصميمها للتحكم فى سرعة محرك والحفاظ على ثباتها ؟ أو ما فائدة لوحة المفاتيح أو الكمية الهائلة من ال RAM فى دائرة نريد تصميمها للتحكم مثلا فى إشارة مرور فى تقاطع معين داخل مدينة ؟ من هنا كان السؤال عن ما هى أقل الإمكانيات التى نستطيع بها أن نصمم دائرة تكون سهلة البناء ، رخيصة التكاليف ، وتفى بمعظم التطبيقات التى تحتاج إلى المعالج كعنصر أساسى فى تطبيقات التحكم الآلى ؟

تتكون أى دائرة تستخدم المعالج فى أغراض التحكم العامة من الأجزاء التالية :

1. المعالج وقد تم تهيئة جميع مساراته لعملية المواجهة مع الأجهزة المحيطة .
2. شريحة ذاكرة EPROM تحتوى البرنامج الذى سيقوم بالعملية التى تستخدم من أجلها دائرة المعالج .
3. شريحة RAM قد تكون هناك الحاجة إليها من قبل البرنامج السابق ، وإذا لم تكن هناك حاجة إليها يمكن فى هذه الحالة الإستغناء عنها .

4. عدد من بوابات الإدخال والإخراج على حسب الحاجة والتطبيق الذى تستخدم من أجله الدائرة المذكورة .

لكى نفهم طريقة عمل هذه الدائرة فنحن نعلم أن المعالج عند إعطائه نبضة *RESET* أو عند بداية تشغيلة يبدأ التنفيذ من عنوان معين وهذا العنوان يختلف من معالج لآخر ، فإذا جعلنا هذا العنوان هو أول عنوان فى البرنامج الذى كتبناه لهذا الغرض (غرض التحكم فى أى عملية صناعية) والموجود فى شريحة ال EPROM والتى تم توصيلها بحيث تعمل مع هذا العنوان ، فإنه بمجرد تشغيل المعالج سينفذ البرنامج ويتعامل مع بوابات الإدخال والإخراج وعلى حسب ظروف العملية التى صمم من أجلها . المفروض طبعا أن هذا البرنامج سيكون برنامجا يدور فى حلقة لا نهائية تجعل المعالج فى حالة تنفيذ مستمرة للبرنامج .

11-2-1 لتثل ةهصيحو

كمثال على ذلك سنقوم فى هذا الفصل إن شاء الله بعمل نظام تحكم فى إشارة مرور فى تقاطع رباعى كالمبين فى شكل (11-1) بحيث يحتوى كرت التحكم على مفتاح (خط تحكم يدوى يستعمل بواسطة رجل المرور عند اللزوم) ، بحيث عندما يكون هذا المفتاح واحدا فإن الإشارات تعمل فى الوضع الطبيعى وعندما يكون هذا المفتاح صفرا فإن اللون الأصفر فى جميع الإتجاهات يضىء ويطفىء flashing بتردد معين وليكن نصف ثانية .

سنقوم هنا ببناء الدائرة مستخدمين المعالج 8085 بعد تجربتها والتأكد من صحتها معمليا ، وسنترك للقارىء حرية إعادة بناء الدائرة مستخدما إما المعالج Z80 أو أى معالج آخر إذا كان يفضل العمل بأحدها وسوف نشير إلى أى معلومات ضرورية لذلك فى حينها .

سيقوم المعالج لحل هذه المشكلة بادارة عدد 12 لمبة فى الأركان الأربعة من التقاطع منها أربعة باللون الأخضر Green, G ، واحدة فى كل ركن ، وأربعة باللون الأصفر Yellow, Y ، واحدة فى كل ركن ، وأربعة باللون الأحمر أيضا Red, R ، واحدة فى كل ركن كما هو مبين فى شكل (11-1) . لإنارة هذا العدد من اللمبات (12) سنحتاج إلى بوابتى إخراج سنستخدم منهما 12 بت للمبات والأربع بتات الباقية سنتركها بدون استخدام حاليا أو لما قد يجد فى المستقبل من إضافات بالنسبة للدائرة . بالنسبة لمفتاح التحكم الذى سيشغل الإشارة إما فى الوضع الطبيعى أو الوضع الترددى flashing فإننا سنحتاج لبوابة إدخال يقرأ منها المعالج قيمة هذا المفتاح باستمرار قبل البدء فى أى دورة من

دورات الإشارة كما سنرى . لذلك فإننا سنحتاج للشرائح التالية لكى نتم عملية بناء دائرة التحكم فى إشارة المرور :

- شريحة المعالج Intel8085 أو Z80 وقد وصلت جميع أطرافها إلى الجهد المناسب (سواء أرضى أو Vcc) وسنرى كيفية توصيل هذه الأطراف فى الجزء القادم .
- شريحتين 74374 لفصل buffer مسار العناوين وقد رأينا ذلك فى الفصل الثامن . أو 74244 فى حالة المعالج Z80 .
- شريحة 74245 لفصل buffer مسار البيانات كما فى الفصل الثامن أيضا .
- شريحة 74138 وشريحة 74125 للحصول على خطوط التحكم $\overline{MEMR}$ و $\overline{MEMW}$ و $\overline{IOR}$ و $\overline{IOW}$ وقد رأينا ذلك أيضا فى الفصل الثامن .
- ثلاث بوابات ، وهذه قد فضلنا أن نحصل عليها من الشريحة 8255A التى شرحناها فى الفصل العاشر .
- شريحة EPROM وهى الشريحة 2716 التى تحتوى على 2 كيلوبايت EPROM حيث سيتم حرق (كتابة) البرنامج عليها .
- بعض الشرائح البسيطة مثل 7408 وهى AND وشريحة عاكس التى نحتاجها لعملية التشفير المبسطة للبوابات وشريحة الذاكرة .
- القليل من المقاومات والمكثفات كما سنرى بعد قليل فى الدائرة الكاملة لهذا الكارت . سنطلق على هذه الدائرة إسم دائرة الميكروكومبيوتر ذى الكارت الواحد one board microcomputer .

يجب على مستخدمى المعالج Z80 والمعالجات الأخرى أن يراعوا استخدام الشرائح المناسبة كما رأينا فى الفصل الثامن عند فصل مسارات كل واحد من هذه المعالجات .

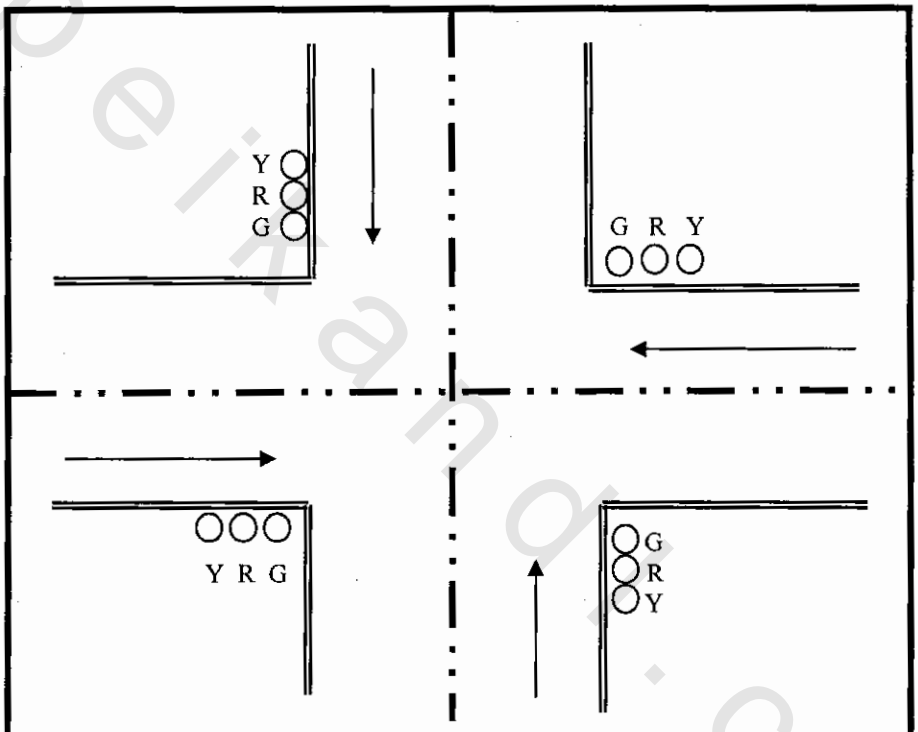
عند تنفيذ هذا الكارت ستواجهنا بعض الأطراف لشريحة المعالج التى لا نعرف وظيفتها بالضبط لأننا لم ندرسها حتى الآن ولذلك فإننا لا نعلم ماذا نفعل بهذه الأطراف ، هل نتركها مفتوحة floating أم هل نوصلها بالأرضى أم Vcc. الجزء القادم من هذا الفصل سنشرح فيه وظيفة كل طرف من هذه الأطراف وماذا نفعل به على الكارت وذلك لكل واحد من المعالجين 8085 و Z80 وذلك قبل أن ندخل فى تفاصيل حل مثال إشارات المرور .

3-11 الأطراف الأخرى للمعالج 8085

لقد درسنا فى الأجزاء السابقة وظيفة أطراف المسارات الثلاثة للشريحة 8085 (40 طرفا) وهى كالتالى :

1. أطراف مسارى العناوين والبيانات وعددها 16 خط .
2. أطراف التحكم وعددها 3 خطوط ($IO/\overline{M}$ و $\overline{RD}$ و $\overline{WR}$) .
3. الطرف ALE .
4. طرفان للقدرة Vcc والأرضى .

مجموع هذه الأطراف هو 22 طرفا ويتبقى 18 طرفا لم نعلم عنها شيئا حتى الآن . سنقوم فى هذا الجزء بشرح سريع لوظائف هذه الأطراف المتبقية . شكل (11-2) يبين إعادة لرسم أطراف الشريحة 8085 لتسهيل عملية المتابعة .

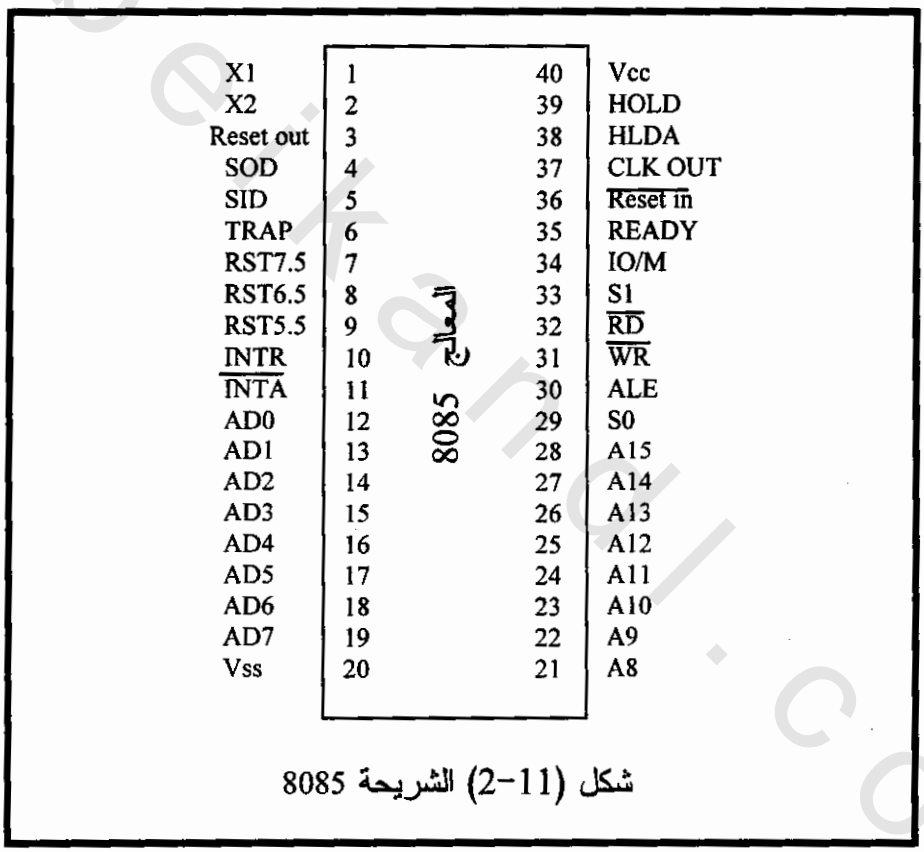


شكل (11-1) إشارات المرور فى تقاطع رباعى

11-3-1 إسئداة التذالم Clock signals

الشريحة 8085 تحتوى على مذبذب وهذا المذبذب يإخذ ترددانه من بللورة أو كريستال crystal توصل بين الطرفين 1 و 2 للشريحة . هذا المذبذب ينتج عنه موجة جيبية ذات تردد يساوى 4 ميجاهرتز . الشريحة 8085 تحتاج إلى نبضات تزامن مربعة وذات تردد يساوى 2 ميجاهرتز ولذلك فإنه وكما هو موضح فى شكل (11-3) فقد وصل خرج المذبذب الجيبى على مقارن من نوع شميت ليقوم

بتحويل الموجة الجيبية إلى موجة مربعة ثم بعد ذلك أدخلت هذه الموجة المربعة على قاسم ليقوم بقسمة تردد الموجة المربعة على 2 فنحصل عند خرج القاسم على موجة مربعة ذات تردد 2 ميجاهرتز حيث تستخدم هذه الموجة فى جميع أغراض التزامن والتشغيل داخل الشريحة 8085 . شكل (11-3) يبين كيفية الحصول على هذه النبضات . هناك الكثير من الشرائح المحيطة والمساعدة للشريحة 8085 والتى تحتاج إلى نفس نبضات التزامن التى تعمل عليها ، لذلك فقد تم إخراج نبضات التزامن على الطرف 37 لشريحة المعالج لكى تتمكن الشرائح والأجهزة المحيطة من الإستفادة منها .



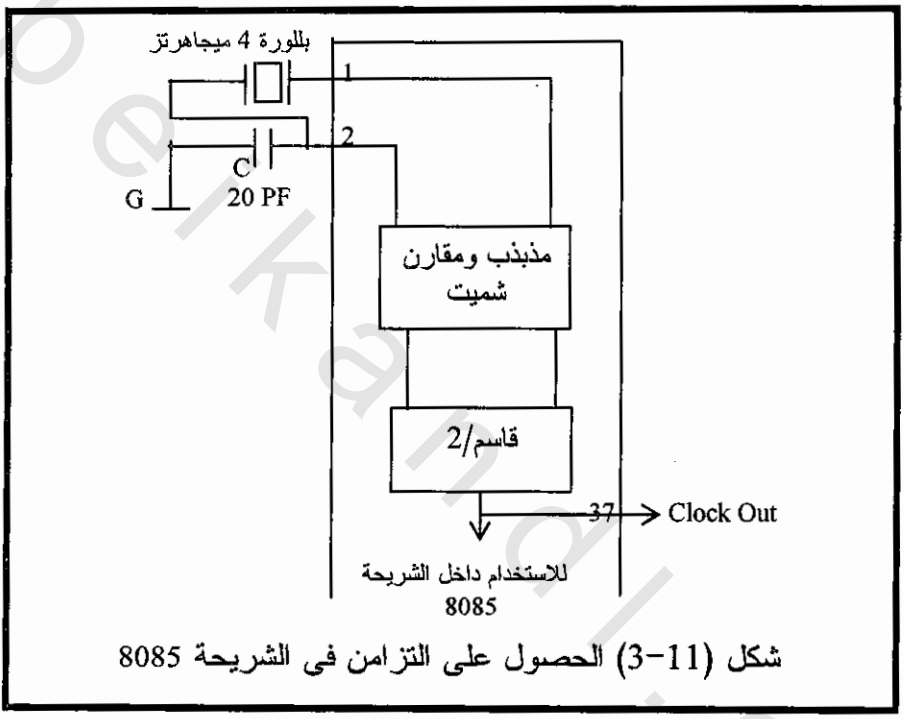
شكل (11-2) الشريحة 8085

11-3-2 بدأ وإعادة تشغيل المعالج

Starting and reseting the processor

عند وضع صفر على الطرف 36 للشريحة 8085 وهو الطرف $\overline{RESET\ IN}$ فإن عداد البرنامج داخل الشريحة يصبح 0000H (ستعشرى) وعند عودة هذا الطرف

إلى الواحد فإن المعالج يبدأ تنفيذ البرنامج الخاص بإعادة التشغيل والموجود عند المكان 0000H فى الذاكرة . شكل (11-4) يبين دائرة يمكن استخدامها فى عملية بدأ وإعادة التشغيل . عند بدء التشغيل فإن المكثف C يشحن من خلال المقاومة R1 إلى أن يصل جهده إلى القيمة high أو الواحد الثنائى عندها تبدأ الشريحة فى العمل من العنوان 0000H كما ذكرنا . إن هذا التأخير الناتج من شحن المكثف يعتبر ضروريا حتى نضمن عدم تشغيل الشريحة قبل أن يستقر الجهد وحتى نتلافى المشاكل اللحظية Transients التى تحدث عند بدأ التشغيل .

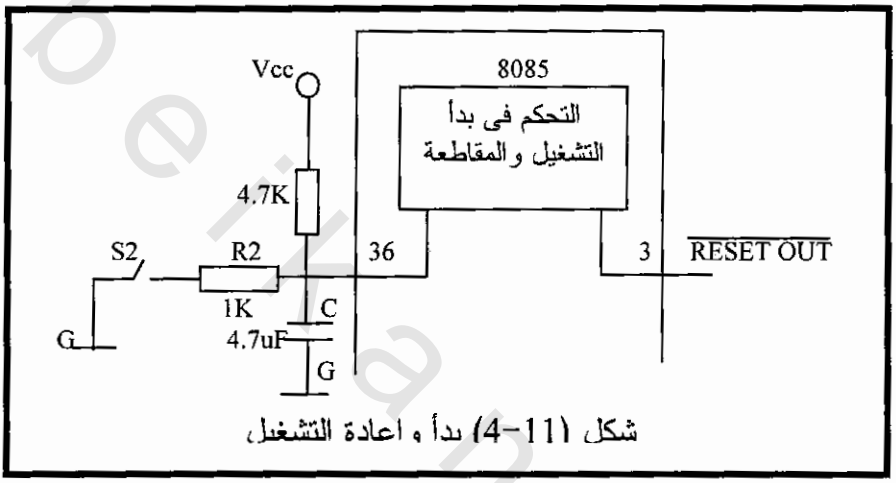


شكل (11-3) الحصول على التزامن فى الشريحة 8085

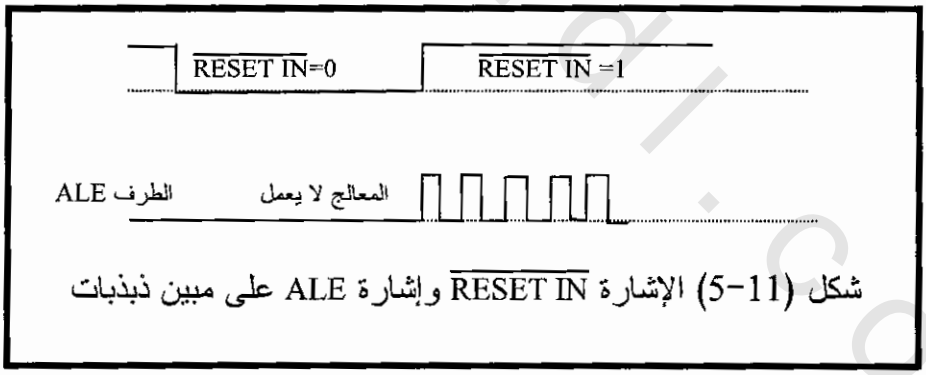
يمكن فى أى وقت أثناء عمل الشريحة 8085 إرسال إشارة إعادة تشغيل إليها من خلال الطرف 36 عن طريق قفل المفتاح S2 حيث عندها يبدأ المكثف C فى التفريغ من خلال المقاومة R2 وطالما أن المفتاح S2 مغلق فإن الشريحة 8085 لن تعمل . عند ترك المفتاح S2 يفتح يبدأ المكثف C فى عملية الشحن من خلال المقاومة R1 حيث تبدأ خطوات إعادة التشغيل التى شرحناها سابقا . يمكن رؤية عملية بدء وإعادة التشغيل على مبين الذبذبات كما فى شكل (11-5) .

كما نعلم فإن الطرف ALE فى أثناء عمل الشريحة 8085 يجب أن تكون عليه موجة مربعة تعكس حالة الإشارة الموجودة على أطراف المسارات AD0-AD7 ، فإذا كان هذا الخط low فإن الإشارة تكون بيانات ، أما إذا كان هذا الخط high

فإن الإشارة تكون عناوين وكما نعلم فإن المعالج أثناء عمله لابد وأن يكون فى حالة تعامل مع بيانات أو عناوين لذلك فإننا يجب أن نرى موجة مربعة على الخط ALE وهذه يمكن استخدامها للدلالة على أن المعالج يعمل . شكل (11-5) يبين الإشارة الموجودة على الطرف ALE مع الإشارة الموجودة على الطرف $\overline{RESET\ IN}$ ، لاحظ فى هذا الشكل أنه طالما أن طرف بدء التشغيل low فإنه ليس هناك أى إشارة على الطرف ALE وعند رجوع طرف بدء التشغيل إلى ال high تظهر الموجة المربعة على الطرف ALE دلالة على أن المعالج بدأ فى العمل .



شكل (11-4) بدأ و اعادة التشغيل



شكل (11-5) الإشارة $\overline{RESET\ IN}$ وإشارة ALE على مبين ذبذبات

11-3-3 الطرفان HOLD و HLDA

لكى يتم اتصال بين أى جهاز خارجى والذاكرة فإن ذلك يكون عادة عن طريق المعالج ، حيث يأخذ المعالج المعلومة من الجهاز الخارجى من خلال مسار البيانات ويقوم بإرسالها إلى المكان المحدد فى الذاكرة من خلال مسار البيانات وبمساعدة مسارى العناوين والتحكم . هذه الطريقة من الإتصال تسمى بطريقة الإتصال غير المباشر مع الذاكرة حيث يكون الإتصال عن طريق المعالج .

هناك طريقة الاتصال المباشر Direct Memory Access, DMA والتى يقوم فيها الجهاز الخارجى بإرسال معلوماته مباشرة إلى الذاكرة دون اللجوء إلى المعالج . من أمثلة ذلك نسخ المعلومات من ذاكرة الحاسب إلى الإسطوانة اللينة Flopy disk والعكس . فى حالة الإتصال المباشر بالذاكرة فإن المعالج يجب عليه فى هذه الحالة أن ينعزل أو يترك أو ينفصل عن المسارات جميعها ليستخدمها الجهاز الخارجى فى عملية الإتصال بالذاكرة وحتى لا يحدث أى تصادم فى المعلومات على هذه المسارات . عندما يحتاج الجهاز الخارجى للمسارات ليقوم من خلالها بالإتصال المباشر بالذاكرة فإنه يخبر المعالج بذلك عن طريق اعطاء إشارة High على الخط HOLD . كلمة Hold تعنى امسك أو قف أو تجمد على وضعك الحالى وهذا هو ما يحدث فعلا للمعالج . عندما يتبين المعالج وجود high على الطرف HOLD وهو الطرف رقم 39 فإنه يقوم بإنهاء دورة الماكينة Machine cycle الحالية والتى يقوم بتنفيذها ثم يوقف تنفيذ البرنامج ويضع جميع خطوط المسارات فى حالة المقاومة العالية وهى حالة العزل أو الانفصال ويتجمد على هذا الوضع إلى أن يعود الخط HOLD إلى الصفر مرة أخرى . بذلك يكون المعالج قد انفصل عن المسارات ، وعند ذلك فإنه يعطى إشارة للدلالة على أنه انفصل عن المسارات عن طريق وضع الخط رقم 38 وهو HLDA فى وضع ال high أى واحد . HLDA تعنى الاعتراف بحالة التجمد Hold Acknowledge . المفروض على الجهاز الخارجى بعد أن يرسل الإشارة high إلى المعالج على خط ال HOLD أن يتتبع الإشارة الموجودة على الخط HLDA ، فعندما يصبح هذا الخط واحدا فإن ذلك يعنى أن المعالج قد انفصل عن المسارات وعندها فقط يستطيع الجهاز الخارجى أن يستخدم المسارات . يمكن استخدام الخط HLDA أيضا بحيث عندما يكون واحدا فإنه يضع جميع الشرائح المستخدمة فى عملية فصل المسارات الثلاثة فى الحالة الثالثة وهى حالة المقاومة العالية . فى دائرة الميكروكومبيوتر التى سنبنيها نريد أن يكون المعالج فى حالة عمل مستمر ولن نقاطعه أبدا أو نطلب منه مساراته ، ولذلك سنوصل خط الدخل HOLD بالحالة low أى غير فعال ، وخط الخرج HLDA سنتركه مفتوحا ولن نوصله بأى شىء لأنه يمثل إشارة خرج من المعالج وتركه مفتوحا لن يؤثر على عمل المعالج .

11-3-4 الطرف READY

إن جميع أجهزة الميكروكومبيتر تكون بها إمكانية تنفيذ البرامج بنظام الخطوة بخطوة Step by step execution فى حالة تعاملها بلغة الأسمبلى . فى هذا النظام فإن المعالج ينفذ خطوة أو أمرا واحدا من البرنامج بعد إعطائه الأمر بذلك ثم ينتظر أمرا آخر لكى ينفذ الخطوة التالية وهكذا . إن المعالج بعد تنفيذ أى أمر

يدخل فى حالة انتظار Waiting إلى أن يجيئه الأمر بتنفيذ الخطوة التالية . فى أثناء حالة الانتظار تبقى آخر إشارة وضعت على المسارات كما هى ولا تتغير حتى أنه بعد الإنتهاء من حالة الانتظار يستأنف المعالج عملية التنفيذ من نفس المكان الذى توقف عنده . السؤال الآن كيف نستطيع إدخال المعالج فى حالة الانتظار هذه؟ إن ذلك يتم عن طريق الطرف رقم 35 من الشريحة 8085 وهو الطرف READY بمعنى جاهز أو مستعد . إن المعالج قبل تنفيذ أى أمر يقوم باختبار الطرف READY فإن كان هذا الطرف high يتم تنفيذ الأمر وإن كان الطرف READY فى حالة low فإن المعالج لن يتم تنفيذ الأمر وسيدخل فى حالة انتظار كما شرحنا . يجب هنا أن نفرق بين حالة الانتظار الناتجة من صفر يوضع على الطرف READY وحالة HOLD التى رأيناها فى الجزء السابق . فى حالة HOLD ينفصل المعالج تماما عن المسارات وتكون جميع المسارات فى وضع المقاومة العالية . أما فى حالة الانتظار التى نحن بصددها هنا فلا ينفصل المعالج عن المسارات ولكن تبقى المسارات حاملة لآخر إشارة تم وضعها على المسارات قبل أن يكون الخط READY صفرا . فى دائرة الميكروكومبيوتر التى سنبنيها نريد المعالج أن يكون فى حالة عمل مستمر ولذلك فسوف نضع الخط READY فى حالة HIGH باستمرار أى غير فعال .

11-3-5 طرفى الحالة S1, S0

الإشارة الموجودة على هذين الخطين تبين حالة المعالج عند أى لحظة من اللحظات حيث أن المعالج لابد وأن يكون فى حالة من الأربع حالات الآتية :

1. حالة انتظار Waiting وهذه كما رأينا تكون عندما نضع صفرا أو low على الخط READY .
2. حالة كتابة سواء كانت كتابة فى ذاكرة أو بوابة إخراج .
3. حالة قراءة أمر من الذاكرة وتكون هذه فى أثناء دورة إحضار الأمر .
4. حالة قراءة معلومة سواء كانت المعلومة فى الذاكرة أو فى بوابة إدخال .

شكل (11-6) يبين الشفرة الموجودة على هذين الخطين فى مقابل كل حالة من الحالات الأربع . لاحظ أن الإشارة الموجودة على هذين الخطين تكون خارجة من المعالج ولذلك فإننا فى دائرة الميكروكومبيوتر التى سنبنيها سنترك هذين الخطين مفتوحين ولن نوصلهما بأى توصيلات خارجية .

11-3-6 أطراف المقاطعة INTR, TRAP, RST7.5, RST6.5, RST5.5

يمكن استخدام خطوط المقاطعة لإيقاف المعالج من تنفيذ البرنامج الذى يقوم بتنفيذه الآن وجعله يذهب لتنفيذ برنامج آخر يسمى برنامج خدمة المقاطعة وبعد الانتهاء من تنفيذ برنامج خدمة المقاطعة يعود المعالج إلى البرنامج الأصلى

حيث يستأنف تنفيذه من نفس المكان الذى حدثت عنده المقاطعة . لقد أفردنا فصلا كاملا للمقاطعة لمن يريد تفاصيل وأمثلة عن هذا الموضوع ، وما يهمنا الآن هو ماذا سنفعل بهذه الخطوط الآن ؟ إذا كنا نريد استخدام المقاطعة فى البرنامج الذى سنستخدمه مع الميكروكومبيوتر الذى ننوى بناءه فى هذا الفصل فعلينا أن نرجىء عملية بناء الدائرة لحين مراجعة الفصل الخاص بالمقاطعة ، أما إذا كنا لن نستخدم المقاطعة (كما هى الحال فى المثال الذى نحن بصدده) فإن جميع هذه الخطوط يجب أن توضع فى حالة عدم الفعالية وهذه الحالة تكون عند توصيل هذه الخطوط بالأرضى أى low كما هو مبين فى الدائرة الكاملة للميكروكومبيوتر فى آخر هذا الفصل .

حالة المعالج	S1	S0
انتظار Waiting	0	0
كتابة Write	0	1
قراءة Read	1	0
قرائة أمر Fetching	1	1

شكل (11-6) الحالات المختلفة للشريحة 8085

11-3-7 الطرفان SOD, SID

الشريحة 8085 يمكنك أن تستقبل منها أو ترسل إليها بيانات تتابعية Serial أى بت بعد بت وليس بنظام البايت كما عرفنا سابقا . لن نتكلم عن عملية إدخال وإخراج البيانات تتابعيا فى هذا الكتاب ولكن الذى يهمنا هنا هو أن نعرف أن البيانات التتابعية تخرج من المعالج على الطرف SOD الذى يعنى Serial Output Data بعد وضعها فى مسجل التراكم وأما البيانات المراد إدخالها تتابعيا فإنها تدخل على الطرف SID الذى يعنى Serial Input Data ومنه إلى مسجل التراكم . هذان الطرفان لن يكون لهما أى فعالية فى دائرة الميكروكومبيوتر ذى الكارت الواحد ولذلك فإننا سنتركهما مفتوحان حيث أنهما لن يؤثرا على تشغيل المعالج بأى حال . لاحظ أن جميع الخطوط التى تحمل إشارة خرج من المعالج والتى لن يتم استخدامها يجب أن نتركها مفتوحة ولا يتم توصيلها بأى شىء (الأرضى أو ال ccV) لأن ذلك يمكن أن يؤثر على عمل المعالج بما لا تحمد عقباه .

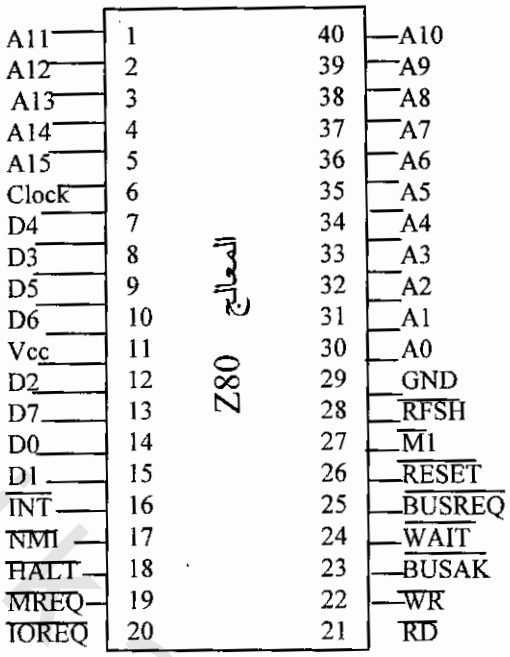
11-4 الأطراف الأخرى للالمعالج Z80

11-4-1 أطراف المقاطعة RESET, BUSRQ, INT, NMI

شكل (11-7) يبين أطراف المعالج Z80 مرة أخرى ليسهل متابعتها فى هذا الفصل . جميع هذه الأطراف تعتبر مداخل للمعالج يمكن مقاطعته من عليها وجميعها فعالة عند الصفر active low ، ولذلك فإنه طالما أن دائرة الميكروكومبيوتر ذى الكارت الواحد لن تستخدم المقاطعة أصلا فإننا سنضع جميع هذه الخطوط فى حالة خمول بتوصيلها إلى Vcc أى high مع العلم أن فصل المقاطعة فى هذا الكتاب خاص بعمليات المقاطعة لمن يريد الإستزادة فى هذا الموضوع . الذى يهمنا معرفته هنا هو العنوان الذى يقفز إليه المعالج عند عمل RESET له أو عند إعادة القدرة إليه ؟ عند إعطاء نبضة على الطرف $\overline{RESET}$ يقفز المعالج Z80 إلى العنوان 0000H حيث يبدأ تنفيذ البرنامج الموجود فى هذا المكان ، لذلك يمكن توصيل هذا الطرف بدائرة كالمبينة فى شكل (11-4) والتى استخدمناها لعمل RESET للمعالج 8085 . الخطان $\overline{BUSRQ}$ و $\overline{BUSACK}$ يكافئان الخطان HOLD و HLDA فى حالة المعالج 8085 حيث يمكن لأى جهاز خارجى بجعل الطرف $\overline{BUSRQ}$ فعالا low أن يطلب من المعالج التخلى عن المسارات ووضعها فى الحالة المنطقية الثالثة حتى يتسنى للجهاز الخارجى إستعمالها فى التعامل المباشر مع الذاكرة . $\overline{BUSRQ}$ تعنى طلب المسارات Requist Bus . عندما يتخلى المعالج عن المسارات يقوم بوضع صفر low على الخط $\overline{BUSACK}$ والذى منه يعرف الجهاز الخارجى أن المعالج قد تخلى فعلا عن المسارات وتركها وكلمة $\overline{BUSACK}$ تعنى اعترافا بطلب التخلى عن المسارات Bus Acknowledge . فى حالة استخدام المعالج Z80 فى دائرة الميكروكومبيوتر المقترحة يجب أن يوضع الخط $\overline{BUSRQ}$ فى حالة خمول أو عدم فعالية بتوصيلة على Vcc أى high . أما الخط $\overline{BUSACK}$ فطالما أنه خط خرج فيجب أن يترك مفتوحا open ولا يوصل بشىء .

11-4-2 الطرف RFSH

يستخدم هذا الخط فى عملية تجديد محتويات الذاكرة الديناميكية dynamic memory كل فترة زمنية محددة ، وطالما أن هذا الخط يعتبر خط خرج أى يحمل إشارات خارجة من المعالج فإننا سنتركه مفتوحا طالما أننا لن نستخدم هذا النوع من الذاكرة فى الدائرة المقترحة ولا ننوى الدخول فى تفاصيل عملية المواجهة مع الذاكرة الديناميكية هنا .



شكل (11-7) أطراف المعالج Z80

11-4-3 الطرف HALT

الخط HALT خط خرج ويكون فعالا عندما يكون المعالج فى حالة HALT أى توقف عن العمل بعد تنفيذ الأمر HALT ولا يخرج المعالج من هذه الحالة إلا عن طريق مقاطعة وطالما أنه خط خرج فسنتركه مفتوحا open .

11-4-4 الطرف *WAIT*

يستخدم هذا الخط فى إدخال المعالج فى حلقة إنتظار حيث يتوقف فيها المعالج عن تنفيذ أى أمر ولا ينفصل عن المسارات ويمكن استخدام هذا الطرف فى تنفيذ البرامج بنظام الخطوة خطوة مثل الطرف READY فى المعالج 8085 وأيضا عندما تكون الأجهزة المحيطة غير مستعدة للتعامل مع المعالج بسبب فارق السرعة مثلا . هذا الخط يكون فعالا عندما يكون صفرا أى أنه active low ولذلك فإننا يجب أن نوصله على Vcc فى دائرة الميكروكومبيوتر المقترحة لنجعله غير فعال .

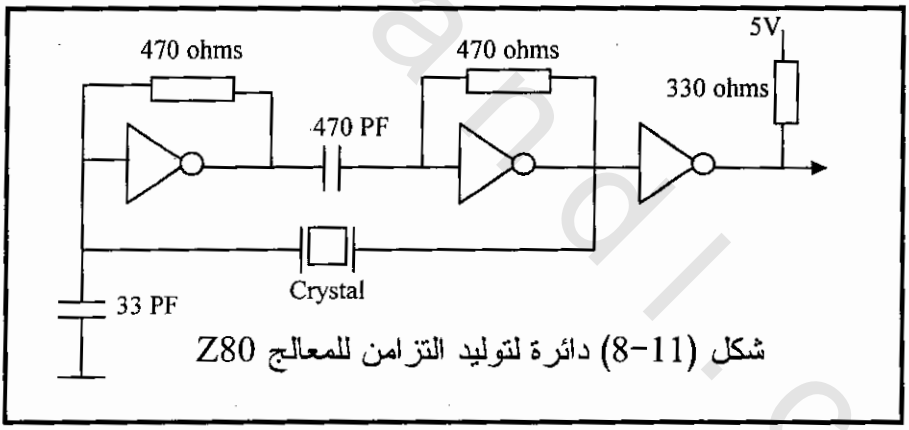
11-4-5 الطرف $\overline{M1}$

هذا الخط هو خط خرج يبين حالة المعالج حيث يكون فعالا (0) عندما يكون المعالج فى حالة إحضار الأوامر من الذاكرة وهذا الخط لن نستخدمه فى دائرتنا ولذلك سنتركه مفتوحا open .

11-4-6 الطرف CLK

يستخدم هذا الخط لإدخال نبضات التزامن الخاصة بالمعالج وهى نبضات TTL يتراوح ترددها بين 2.5MH أو 4MH أو 6MH وذلك على حسب نوع المعالج المستخدم . شكل (11-8) يبين دائرة مقترحة لتوليد نبضات التزامن اللازمة للمعالج Z80 .

بذلك نكون قد انتهينا من التعرف على جميع الأطراف الأخرى للمعالج Z80 وعرفنا فكرة موجزة عنها ونستطيع أن نلخص القول فى أن جميع خطوط الخرج الغير مستخدمة يجب أن تترك مفتوحة open وجميع خطوط الدخل الغير مستخدمة توضع فى وضع خمول أى عدم فعالية .



شكل (11-8) دائرة لتوليد التزامن للمعالج Z80

11-5 إشارات المرور

الآن بعد أن عرفنا وظيفة كل طرف من أطراف المعالج وماذا سنفعل بكل طرف من الأطراف الغير مستخدمه ، ماذا عن دائرة المعالج التى ستتحكم فى إشارة المرور كما ذكرنا فى المثال التوضيحى ؟ قبل أن ندخل فى تفاصيل بناء أو حل هذه المسألة سننفق بعض الوقت فى التفكير فى خطة الحل . إن حل هذه المسألة كباقى المسائل التى تستخدم المعالج يتكون من جزء بناء الدائرة

hardware وجزء برمجة software . بالنسبة لجزء البناء وكما أشرنا فى بداية الفصل فإننا سنحتاج لما يلى :

1. المعالج وقد تم فصل جميع مساراته ، مسار البيانات D0 إلى D7 ومسار العناوين A0 إلى A15 ومسار التحكم بالإضافة إلى خط إعادة الوضع . سنستخدم كما ذكرنا المعالج 8085 كمثال ومن يريد استخدام أى معالج آخر فذلك أصبح سهلا جدا بعد قراءة هذا الفصل . ولقد رأينا فى الفصل الثامن كيفية فصل المسارات الثلاثة ويمكن مراجعة ذلك وبالذات للمعالج 8085 الذى سنستعمله فى هذا المثال . شكل (11-9) يبين المعالج والشرائح المستخدمه فى عملية الفصل .
2. إشارة المرور الرباعية كما رأينا بها 12 لمبة (ثلاثة فى كل ركن من أركان التقاطع ، أحمر وأخضر وأصفر) ومطلوب إدارة هذه اللمبات بتتابع معين وأزمنة محسوبة كما سنرى بعد قليل لذلك سنحتاج إلى بوابتى إخراج بمجموع 16 بت سنستخدم منهم 12 لإدارة ال 12 لمبة ويتبقى 4 بتات من ال 16 لن تستخدم وستترك للاستخدام المستقبلى . كما ذكرنا سابقا فى المثال التوضيحى فإن هناك مفتاح سيقوم المعالج بقراءته دائما وإذا كان هذا المفتاح واحدا (ON) فإن الإشارة تعمل فى الحالة العادية ، وأما إذا كان المفتاح صفرا (OFF) فإن الإشارة ستعمل فى الحالة الترددية للون الأصفر فقط flashing . لذلك فإننا سنحتاج إلى بوابة إدخال سنستعمل منها بت واحدة لقراءة حالة هذا المفتاح والباقى لن يستخدم وسيترك لأى استخدامات مستقبلية . خلاصة القول أننا سنحتاج إلى بوابتى إخراج وبوابة إدخال . بعد أن قررنا أننا سنحتاج إلى بوابتى إخراج وبوابة إدخال يجب أن نقرر بأى طريقة من الطرق التى درسناها فى الفصل العاشر سنبنى هذه البوابات . لقد قررنا نحن أن نستخدم البوابات القابلة للبرمجة أى الشريحة 8255A وذلك لسهولة برمجتها وبنائها وكونها شريحة واحدة تحتوى الثلاث بوابات التى نحتاج إليها . شكل (11-10) يبين طريقة توصيل هذه الشريحة على المسارات الثلاثة القادمة من المعالج .
3. يبين شكل (11-10) أيضا كيفية توصيل شريحة الذاكرة EPROM وهى الشريحة 2716 التى سنكتب عليها برنامج التحكم فى كل العملية . لقد درسنا فى الفصل التاسع طريقة توصيل الذاكرة على المعالج والعملية هنا أبسط بكثير مما شرحنا فى الفصل التاسع ، فإنه طالما أن المعالج سيكون متصلا بشريحة ذاكرة واحدة فإن عملية العنونة والتشفير من الممكن أن تكون بسيطة جدا إذا وصلنا خط تنشيط الشريحة 2716 وهو الخط $\overline{CS}$ على خط التحكم $\overline{MEMR}$ وخط تنشيط خرج الشريحة $\overline{OE}$ بالأرضى مباشرة . فى هذه

الحالة فإن الشريحة ستعمل وتخرج خرجها مع أى أمر قراءة من الذاكرة ولا حرج فى ذلك حيث أنها هى شريحة الذاكرة الوحيدة الموصلة مع المعالج . لاحظ أنه طالما أن ال EPROM ستعمل مع أى عنوان يخرجه المعالج فإنه يجب أن نتوقع أنها ستعمل بمجرد توصيل القدرة للدائرة لأن المعالج 8085 كما رأينا سابقا عند توصيل القدرة إليه أو إعادة وضعه RESET فإنه يبدأ التنفيذ من العنوان 0000H حيث سيشغل ال EPROM .

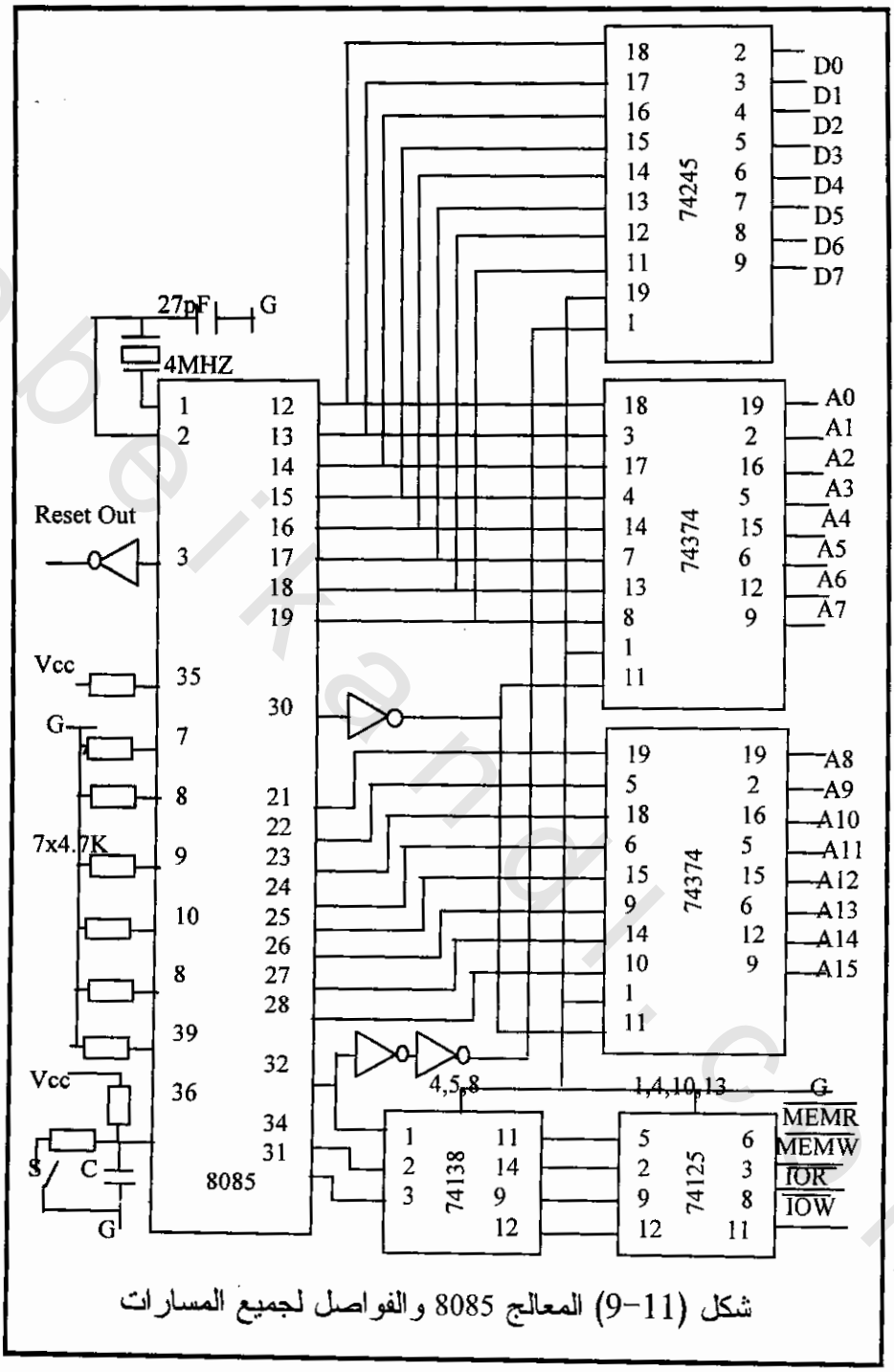
4. يمكن توصيل كل بت من بتات الخرج الخارجة من الشريحة 8255 على قاعدة ترانزستور ليعمل كفاصل buffer لإدارة هذه الدايودات حتى تعطى كمية إضاءة أحسن . شكل (11-10) يبين ذلك أيضا .
5. يمكن توصيل خرج البوابات على لمبات 220 فولت للحصول على إنارة قوية ودائرة واقعية عن طريق استخدام عوازل ضوئية opto_couplers .

بالنسبة لجزء البرمجة software فإننا سنفكر فيه بالطريقة التالية :

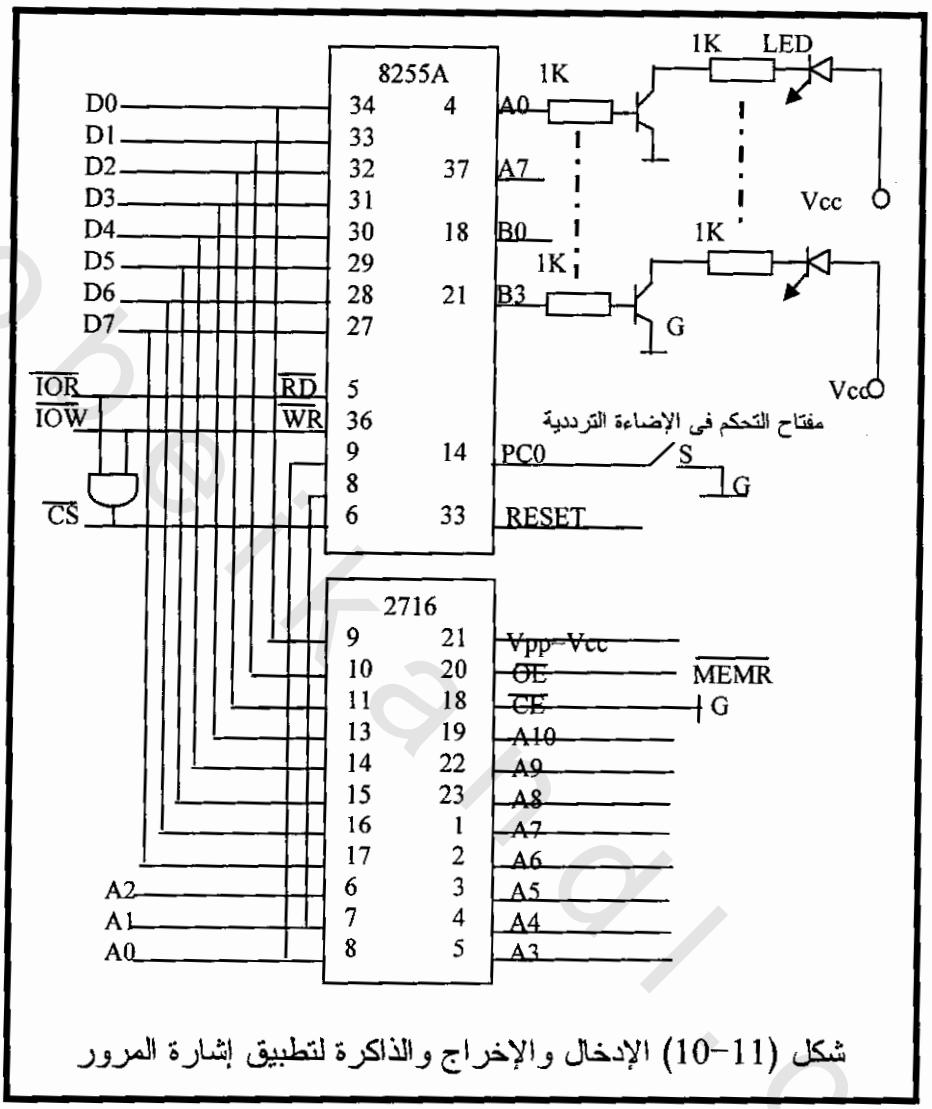
1. يحتوى شكل (11-11) على جدول به جميع الحالات الممكنة لجميع اللمبات وزمن كل حالة والشفرة أو الرقم المطلوب إخراجه على بوابات الإخراج للحصول على هذه الحالة . من شكل (11-11) نلاحظ مثلا أنه فى الحالة الأولى يكون الأخضر الأول مضيئا والأحمر فى جميع الاتجاهات الأخرى مضيئا أيضا والأصفر فى جميع الاتجاهات مطفأ وسوف يستمر هذا الوضع لمدة 50 ثانية . لاحظ أن المضيء فى الجدول يساوى واحدا والمطفأ يساوى صفرا ، لذلك فإن هذا الوضع الأول سيكافىء الشفرة 4C على بوابة الإخراج الأولى والشفرة 02 على بوابة الإخراج الثانية . بعد ذلك يضيء الأصفر الأول مع الأخضر الأول مع نفس الوضع السابق لجميع اللمبات الأخرى وذلك لمدة 10 ثوان وذلك كزمن تحذير بالوقوف بعد الضوء الأخضر . بعد ذلك ستضيء جميع اللمبات الحمراء فى جميع الاتجاهات لمدة 5 ثوان كزمن أمان خوفا من السيارات المسرعة التى لا تستطيع التوقف فى خلال الضوء الأصفر . بعد ذلك تكرر هذه الأزمنة لجميع الاتجاهات الأخرى .

لاحظ أن هذه الأزمنة يمكن التحكم فيها بالزيادة والنقصان على حسب الرغبة ، ثم إن تتابع الضوء الأخضر فى الاتجاهات المختلفة بمعنى أن أى اتجاه سيسمح له بالمرور وأى اتجاه سيسمح له بعده ، هذه أيضا يمكن التحكم فيها . إن فكرة البرنامج التى نقترحها هنا (بالطبع فإن كل قارىء ستكون لديه فكرة مختلفة وربما أفضل) تعتمد على ما يلى :

1. سنخزن فى مكان ما فى الذاكرة الشفرات المطلوب إخراجها على بوابات الإخراج لكل حالة وكذلك زمن كل حالة بالتتابع من الجدول المبين فى شكل (11-11) وليكن ذلك كما يلى :



شكل (11–9) المعالج 8085 والفواصل لجميع المسارات



شكل (11-10) الإدخال والإخراج والذاكرة لتطبيق إشارة المرور

E100 4C , 4C ستخرج على البوابة الأولى
E101 02 , 02 ستخرج على البوابة الثانية
E102 50 , 50 زمن تأخير للحالة الأولى
E103 4E , 4E تخرج على البوابة الأولى
E104 02 , 02 تخرج على البوابة الثانية
E105 10 , 10 هى زمن تأخير للحالة الثانية

وهكذا سيحتوى هذا الجدول على 12 بايت .

2. سيكون البرنامج عبارة عن قراءة للشفرتين المقابلتين لكل حالة وإخراجهما على بوابتى الإخراج رقم 00 ورقم 01 ثم الدخول فى زمن التأخير المقابل لهذه الحالة .
3. قبل الدخول فى أى حالة جديدة لابد أن يقرأ المعالج بوابة الإدخال رقم 02 ليعرف إذا كان مفتاح التحكم المقابل للبت رقم صفر يساوى واحد أم صفر ، فإذا كانت هذه البت تساوى واحد فإن البرنامج يسير فى سيره الطبيعى كما فى الجدول المبين فى شكل (11-11) ، وإذا كانت هذه البت تساوى صفرا سيقفز إلى برنامج آخر ينفذ عملية التردد على اللون الأصفر .

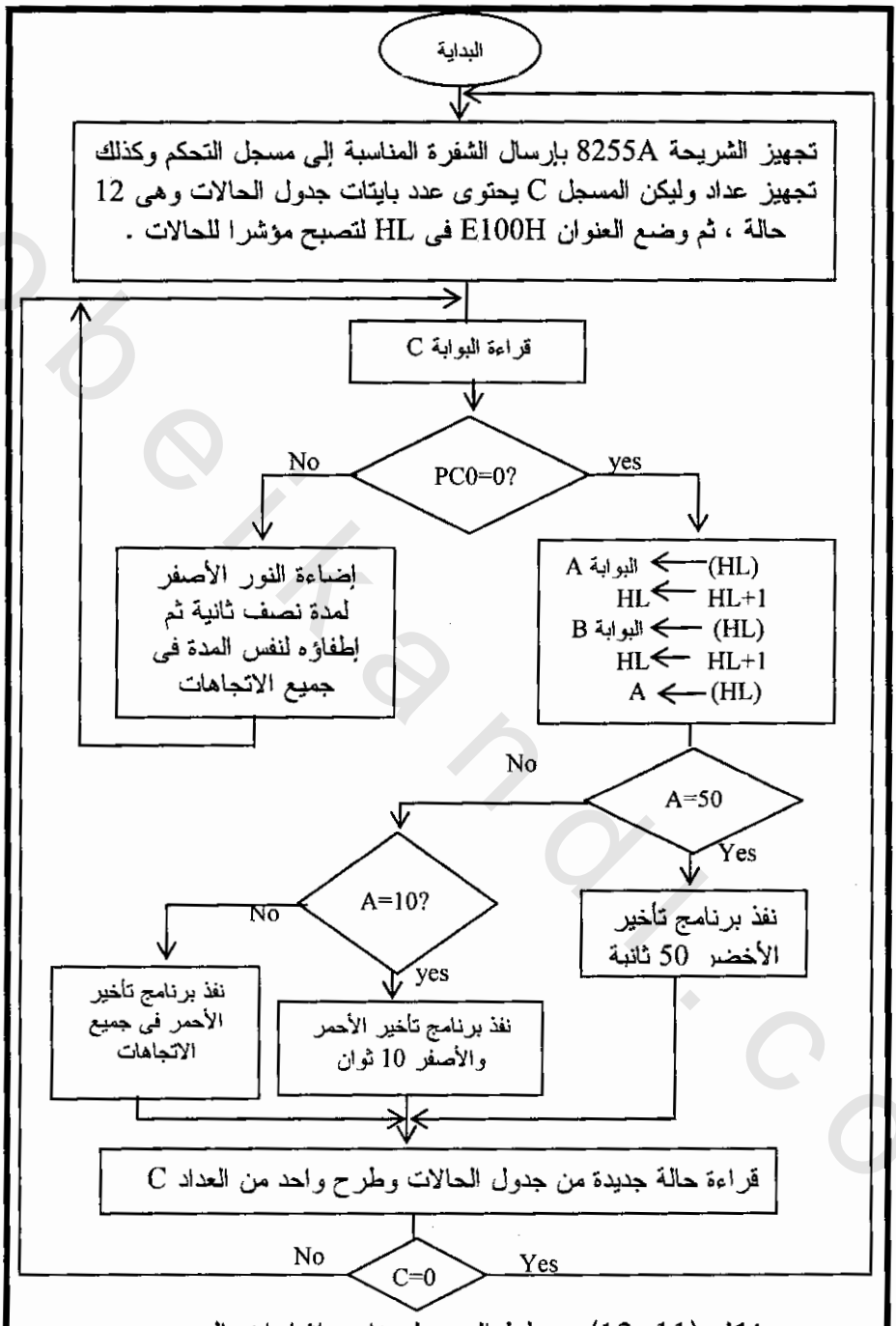
شكل (11-12) يبين خريطة التدفق ، والبرنامج مكتوبا بلغة التجميع للمعالج 8085 سيكون كما يلى : (يلاحظ من هذا البرنامج خلوه من البرامج الفرعية subroutines بالرغم من أفضلية استخدامها هنا فى مثل هذه التطبيقات ولكننا تجنبنا ذلك لسببين أولهما أننا مازلنا لم نشرح البرامج الفرعية حتى الآن فى هذا الكتاب وثانيهما أن الدائرة التى بنيناها ليس بها ذاكرة RAM تستخدم كمكدسة stack فى حالة استخدام البرامج الفرعية كما سنرى) .

رقم الحالة	G1Y1R1	G2Y2R2	G3Y3R3	G4Y4R4	البوابة 00 01	الزمن بالثانية
1	1 0 0	0 0 1	0 0 1	0 0 1	02 4C	50
2	1 1 0	0 0 1	0 0 1	0 0 1	02 4E	10
3	0 0 1	0 0 1	0 0 1	0 0 1	02 49	5
4	0 0 1	1 0 0	0 0 1	0 0 1	02 61	50
5	0 0 1	1 1 0	0 0 1	0 0 1	02 71	10
6	0 0 1	0 0 1	0 0 1	0 0 1	02 49	5
7	0 0 1	0 0 1	1 0 0	0 0 1	03 09	50
8	0 0 1	0 0 1	1 1 0	0 0 1	03 89	10
9	0 0 1	0 0 1	0 0 1	0 0 1	02 49	5
10	0 0 1	0 0 1	0 0 1	1 0 0	08 49	50
11	0 0 1	0 0 1	0 0 1	1 1 0	0C 49	10
12	0 0 1	0 0 1	0 0 1	0 0 1	02 49	5
1	من هنا يبدأ تكرار هذه الحالات				02 49	5

شكل (11-11) جدول الحالات المختلفة لإشارة المرور

البوابة A أخراج و B أخراج و C إدخال والجميع فى الحالة صفر

```
MVI A,89H ;
OUT 03H   ;          إرسال إلى مسجل التحكم فى 8255A
```



شكل (11-12) مخطط السير لبرنامج إشارات المرور

```
START: MVI C,0CH    ;          مسجل C عداد يحتوى 12 بايت
       LXI H,E100   ;          HL مؤشر للذاكرة ابتداء من E100
READ:  IN 02H       ;          قراءة البوابة C
       ANI 01H      ;          حجب لجميع البتات ما عدا الأولى
       JZ FLASH     ;          قفز عندما PC0=0 لتردد اللون الأصفر
       MOV A,M      ;          إحضار الشفرة الأولى من الذاكرة
       OUT 00       ;          إخراج هذه الشفرة على البوابة A
       INX H        ;
       MOV A,M      ;          إحضار الشفرة الثانية من الذاكرة
       OUT 01       ;          إخراج هذه الشفرة على البوابة B
       INX H        ;
       MOV A,M      ;          إحضار زمن تأخير
       CPI 50H      ;          هل هذا الزمن =50
       JNZ TEN      ;          قفز إذا لم يكن الزمن=50
       MVI B,6CH    ;          بداية برنامج تأخير 50 ثانية
LOOP1: MVI D,FFH    ;
LOOP2: MVI A,FF     ;
LOOP3: DCR A        ;
       JNZ LOOP3    ;
       DCR D        ;
       JNZ LOOP2    ;
       DCR B        ;
       JNZ LOOP1    ;
       JMP CONTIN   ;
TEN:   CPI 10H      ;          هنا التأخير لا يساوى 50 فهل =10
       JNZ FIVE     ;          إقفز إذا لم يكن التأخير = 10
       MVI B,16H    ;          بداية برنامج تأخير 10 ثوان
LOOP4: MVI D,FFH    ;
LOOP5: MVI A,FFH    ;
LOOP6: DCR A        ;
       JNZ LOOP6    ;
       DCR D        ;
       JNZ LOOP5    ;
       DCR B        ;
       JNZ LOOP4    ;
       JMP CONTIN   ;
FIVE:  MVI B,0BH    ;          هنا التأخير لا يساوى 10 ولكن يساوى 5 ثوان
LOOP7: MVI D,FFH    ;          بداية برنامج تأخير 5 ثوان
LOOP8: MVI A,FFH
```

```
LOOP9:  DCR A
        JNZ LOOP9    ;
        DCR D        ;
        JNZ LOOP8    ;
        DCR B        ;
        JNZ LOOP7    ;
CONTIN: DCR C
        JZ START     ;
        INX H        ;
        JMP READ     ;
FLASH:  MVI A,92H    ;   بداية برنامج تردد الضوء الأصفر
        OUT 00H      ;   الرقمان 92H و 04H يضيئان الأصفر
        MVI A,04H    ;   فى جميع الأتجاهات
        OUT 02H      ;
        MVI B,D8H    ;   بداية تأخير نصف ثانية
LOOPA:  MVI D,FFH    ;
LOOPB:  DCR D        ;
        NOP          ;
        JNZ LOOPB    ;
        DCR B        ;
        JNZ LOOPA    ;
        MVI A,00H    ;   إطفاء النور الأصفر فى جميع الأتجاهات
        OUT 00H      ;
        OUT 01H      ;
        MVI B,D8H    ;   بداية تأخير نصف ثانية
LOOPC:  MVI D,FFH    ;
LOOPD:  DCR D        ;
        NOP          ;
        JNZ LOOPD    ;
        DCR B        ;
        JNZ LOOPC    ;
        JMP READ     ;   قفز لقراءة بوابة الإدخال PC
```

- سنرى فى فصل البرامج الفرعية كيفية الحصول على أزمنة التأخير باستخدام البرامج الفرعية مما سيؤدى إلى اختصار خطوات مثل هذا البرنامج بدرجة كبيرة ، ويمكن إعادة كتابة البرنامج بهذا الوضع كتمرين على البرامج الفرعية .

obeikandi.com

6-11 تمارين

1. أعد تصميم كارت الميكروكومبيوتر ، وكذلك البرنامج ، الخاصين بالتحكم فى إشارة المرور ولكن مستخدما هذه المرة المعالج Z80 بدلا من المعالج 8085 كما كان فى هذا الفصل .

obeikandl.com