ایجاد آزمون پذیری
در سیستم ریز پردازندهٔ آموزشی هیت کیت

دکتر کریم فائز
استادیار دانشکده مهندسی برق دانشگاه صنعتی امیرکبیر
مهندس حسن طاهری
مربی دانشکده مهندسی برق دانشگاه صنعتی امیرکبیر

چکیده

ایجاد آزمون‌پذیری در سیستم‌های ریزپردازنده یکی از عوامل بسیار قوی جهت عیب‌یابی و رفع عیب این سیستم‌ها به شمار می‌رود. از آنجا که سیستم‌های مبتنی بر ریزپردازنده بخصوص انواع آموزشی آنها علیرغم مدارهای خاصی که آنها را در مقابل خطا تحمل‌پذیر می‌سازند، به علت این استفاده آنها توسط افرادی که دارای تجربیات زیادی نیستند، امکان ایجاد خرابی در آنها زیاد می‌باشد. و برای حل این مسأله بایستی امکانات خاص آزمون‌پذیری را در آنها ایجاد و یا پیش‌بینی نمود.

در این مقاله سعی شده است، مراحلی که به منظور ایجاد آزمون پذیری در سیستم آموزشی هیت کیت موجود در آزمایشگاه اجزاء کامپیوتر دانشکدهٔ برق طی شده است مورد بررسی قرار گیرد.

۱ـ مقدمه

سیستم‌های مبتنی بر ریزپردازنده‌ها بخصوص انواع آموزشی آنها غالبا" به علت مورد استفاده قرار گرفتن توسط افراد کم تجربه در معرض خرابی قرار دارند، لذا بایستی در این وسایل امکاناتی را پیش بینی و یا ایجاد کرد که در صورت بروز هرگونه خرابی بتوان به سرعت عیب را یافت و رفع نمود و سیستم را مجددا" آمادهٔ بهره‌برداری کرد. سیستم هیت‌کیت موجود درآزمایشگاه اجزاء کامپیوتر علیرغم مدارهای مختلفی که در آن طراحی شده‌اند، باعث می‌شوند در هنگام استفادهٔ غیرصحیح از آن به راحتی خراب شده و اغلب در معرض خرابی قرار می‌گیرد. با طرحی که در این مقاله ارائه می‌شود، امکانات عیب‌یابی و رفع عیب آن فراهم شده است.

در بخش ۲ قدمهائی را که جهت یک مرحله‌ای کردن این سیستم برداشته شده است خواهد آمد و آنگاه در فصل ۳ برنامه‌هائی که جهت عیب‌یابی سیستم نوشته شده‌اند شرح داده شده‌اند و کد ماشین مربوط به این برنامه‌ها در قسمت ضمیمهٔ ۱ـ شرح داده شده است. این برنامه‌ها شامل برنامهٔ آزمایش حافظه، آزمایش سون سگمنت‌ها و آزمایش جعبه کلید (Key Board) می‌باشند که به زبان اسمبلر مربوط به 8080 نوشته شده‌اند البته به کمک فلوچارت‌هائی که در هر قسمت آمده‌اند می‌توان برای سایر انواع CPU برنامه‌های مربوطه را نوشت.

۲ـ تک مرحله‌ای کردن CPU (Single Step) به منظور ایجاد آزمون پذیری

یکی از مراحلی که بایستی جهت ایجاد آزمون‌پذیری ریز ـ پردازنده‌ها طی کرد، مرحلهٔ تک مرحله‌ای کردن می‌باشد در اغلب سیستم‌های ریزپردازنده به منظور عیب یابی نرم افزارهای تحت توسعه، برنامه‌های خاصی وجود دارند که به کمک آنها می‌توان برنامهٔ تحت توسعه را به صورت تک مرحله‌ای اجراء نموده و عیب‌یابی کرد. ولی از آنجا که در هنگام خرابی سیستم عموما" این برنامه‌ها قابل استفاده نمی‌باشند، لذا بایستی امکانات تک مرحله‌ای را به صورت سخت افزار فراهم نمود، تا بتوان در هنگام ایجاد خطا به آزمایش بعضی از قسمتهای سخت افزاری اقدام کرد.

در سیستم هیت کیت از ریزپردازندهٔ 8080 استفاده شده است در این CPU ورودی‌ای به نام Ready وجود دارد که تحت کنترل آی سی ۸۲۲۴ قرار دارد. آی سی ۸۲۲۴ تحت کنترل سیگنال ورودی

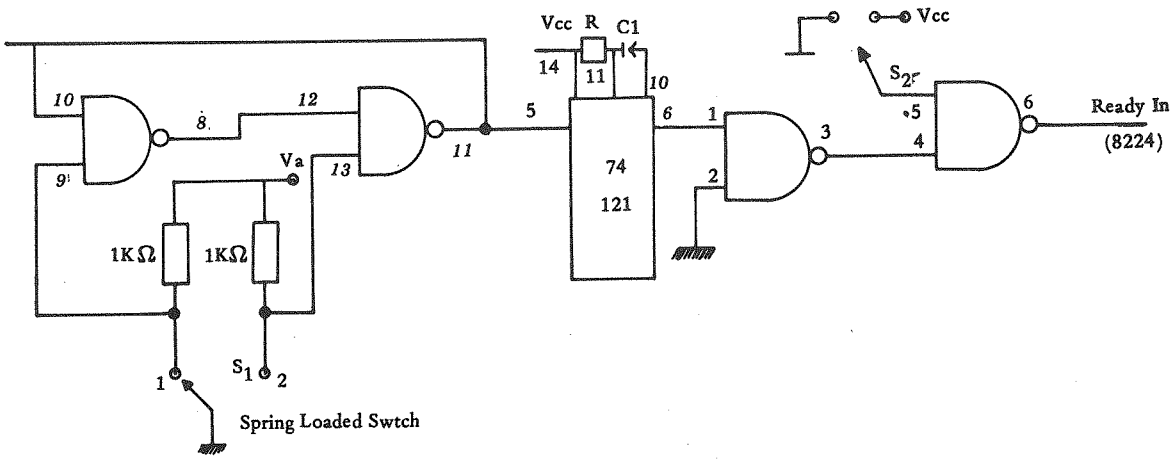
Ready In ، سیگنال Ready Out را تولید می‌کند که به ورودی Ready ۸۰۸۰ می‌رود. ورودی Ready In در زمان فعالیت نرمال سیستم به صورت high خواهد بود. حال اگر این ورودی را به low ببریم باعث می‌شود که در انتهای اولین سیکل حالت T_2 آتی، پردازنده به حالت Wait برود. و مادامی که این ورودی به صورت low باقی بماند ریزپردازنده در همان حالت Wait قرار خواهد داشت، تا آن که دوباره به حالت high برگردد. در این‌صورت ریز- پردازنده حالتهای T_3 و T_4 و . . . . . را ادامه داده و وارد سیکل ماشین بعدی می‌گردد. در طول سیکل حالت wait (T_W)، آدرس داده و سیگنالهای کنترل مربوط به به سیکل ماشین فعلی روی مسیرهای عمومی (BUS) نگاهداری می‌شوند. و تا هر زمان که لازم باشد می‌توان ریزپردازنده را در حالت Wait نگاهداشته و به کمک اسیلو- سکوپ سطوح ولتاژ را روی این سیگنالها بررسی نمود. پس از بررسی سیگنالهای مربوط به یک ماشین، ورودی Ready In را برای لحظه‌ای به high برده و سپس به low برمی‌گردانیم.

طول زمانی‌ای که سیگنال Ready In در حالت high قرار دارد بایستی طوری باشد که پردازنده بتواند از حالت Wait خارج شده و به حالت T_3 برود، لذا بایستی کمی از پریود پالس ساعت CPU بیشتر باشد از طرفی لازم است این زمان آن‌قدر طولانی نباشد که ریز پردازنده موفق به اجرای سیکل T_2 بعدی گردد، در غیر این‌صورت نمی‌توان پردازنده را درسیکل ماشین بعدی به حالت Wait برد و لذا موفق به بررسی سیگنالهای مربوطه نخواهیم شد. پس زمان فوق بایستی حداکثر از ۴ برابر پریود پالس ساعت CPU کمتر باشد بدین ترتیب می‌توان در هر سیکل ماشین حالت سیگنالهای موجود در روی مسیرهای عمومی را کنترل کرده و اجرای یک برنامه را به روش تک مرحله‌ای تعقیب نمود. در این روش از حالت Wait پردازنده و از ورودی Ready In آی سی ۸۲۲۴ جهت تک مرحله‌ای کردن کارCPU استفاده شده است.

مدار ساده شکل ۱ را می‌توان جهت تک مرحله‌ای کردن کار CPU استفاده کرد. در این مدار از دو سوئیچ S_1 و S_2 استفاده‌شده است. سوئیچ S_2 در کار نرمال ریزپردازنده به‌ولتاژ صفر وصل‌می‌باشد و باعث می‌شود که خروجی (9) مربوط به آی سی NAND در حالت high قرار داشته باشد و لذا پردازنده کار نرمال خود را ادامه می‌دهد.

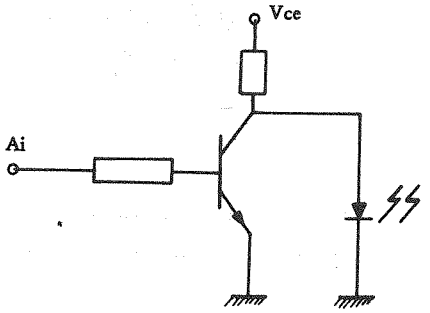
برای مثال ریزپردازنده به حالت تک مرحله‌ای می‌بایستی ورودی (۵) به کمک سوئیچ S_2 به حالت high برده شود. که باعث می‌گردد تا خروجی (۶) فوق‌الذکر تحت کنترل خروجی آی سی ۷۴۱۲۱ قرار گیرد. از آنجا که این آی سی یک فلیپ فلاپ منواستابل است که خروجی آن در حالت نرمال low می‌باشد لذا می‌توان به‌کمک سوئیچ S_1 که از نوع Spring loaded می‌باشد آن را تریگر کرد. به‌منظور حذف bouncing آن از دو عدد NAND طبق شکل ۱ استفاده‌شده است. پس از تریگر شدن آی سی ۷۴۱۲۱ خروجی آن برای فاصلهٔ زمانی‌ای که توسط خازن C_1 و مقاومت R_1 تعیین می‌شود به حالت high می‌رود و در پایان این زمان به حالت low بر می‌گردد. درمدار فوق این زمان برای خازن $C_1 = 100\ pf$ و مقاومت $R_1=10K\ \Omega$ برابر ۵۰۰ ns خواهد بود که این زمان مساوی پریود یک پالس ساعت با فرکانس 2MHz می‌باشد به طوری که اگر آن را به ورودی Ready In اعمال کنیم باعث می‌شود که ریزپردازندهٔ از حالت wait خارج شده پس از طی کردن سیکلهای حالت بعدی وارد سیکل wait مربوط به سیکل ماشین بعدی گردد.

بدین ترتیب با تغییر وضعیت سوئیچ S_1 به طور پشت سر هم می‌توان اجراء برنامه را به طور قدم به قدم به جلو برد و در هر مرحله سطح سیگنالها را روی مسیرهای عمومی آدرس و داده، کنترل کرد. در ضمن به‌منظور این که بتوان به‌سادگی آدرس دستورالعمل‌های



شکل ۱ــ مدار مربوط به تک مرحله‌ای کردن CPU ی ۸۰۸۰

مختلف را به دست آورد می‌توان از مداری که شامل ۱۶ عدد LED می‌باشد آدرس‌های موجود در روی مسیر عمومی آدرس را نمایش داد. این LED ها به کمک مدار زیر به مسیر عمومی آدرس وصل می‌شوند.



شکل ۲ـ مدار درایورLED

بدین ترتیب high یا low شدن بیتهای Ai باعث روشن یا خاموش شدن این دیودها می‌گردند. و به کمک این LED ها می‌توان آدرس‌های مختلف موجود در روی مسیر عمومی آدرس را نمایش داد.

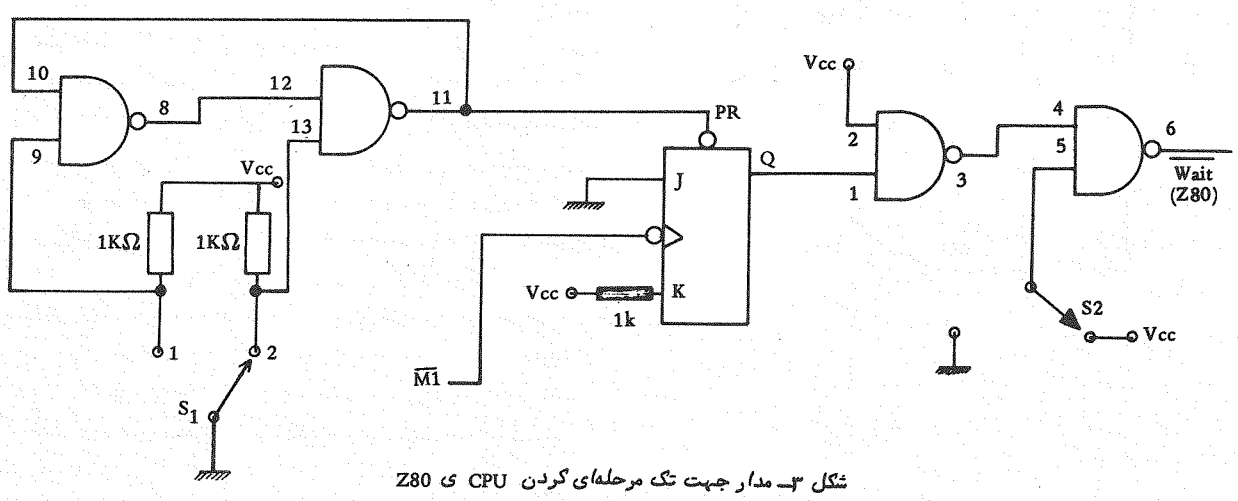
همان‌طور که ملاحظه شد سیستم فوق در هرسیکل ماشین یکبار در حالت wait متوقف می‌شود اما در حالت کلی لازم نیست که برای تعقیب کار پردازنده در هر سیکل ماشین این توقف صورت گیرد و عموماً" تنها توقف در سیکل اول ماشین مربوط به هر دستورالعمل که در آن عمل fetch صورت می‌گیرد برای عیب‌یابی برنامه کافی خواهد بود. البته درمورد PCU ی ۸۰۸۰ این عمل امکان پذیر نیست ولی درمورد CPU ی Z80 می‌توان با استفاده از خروجی $\overline{M1}$ (پین ۲۷) و ورودی $\overline{Wait}$ (پین ۲۴) این امکان را ایجاد کرد که این امکان بکمک مدار شکل ۳ فراهم می‌شود.

در این مدار موقعی که سوئیچ S_2 در وضعیت زمین باشدخروجی (۶) high بوده و ریز پردازنده‌ٔ Z80 به کار نرمال خود مشغول خواهد بود. زمانی که سوئیچ S_2 به حالت Vcc برود. فلیپ فلاپ JK وارد مدار می‌شود که این فلیپ فلاپ در حالت عادی به کمک پالس $\overline{M1}$ در حالت low قرار دارد. لذا با در مدار قرار گرفتن فلیپ فلاپ فوق، CPU ی Z80 بلافاصله به حالت Wait می‌رود و در همان حالت باقی ماند تا آن که این فلیپ فلاپ توسط سوئیچ Preset, S_1 شود. بدین صورت فلیپ فلاپ به حالت high رفته و تا آمدن پالس $\overline{M1}$ بعدی به حالت high باقی می‌ماند پس از high شدن (Q) ریز پردازنده از حالت Wait خارج شده و کار خود را ادامه می‌دهد پس از اجرای دستورالعمل مربوطه، به منظور اجرای دستورالعمل بعدی وارد سیکل ماشین M1 بعدی می‌گردد و پالس را روی پین $\overline{M1}$ ایجاد خواهد نمود این پالس باعث می‌شود که فلیپ فلاپ JKتریگر شده و به حالت low برود و بدین ترتیب پردازنده وارد حالت Wait خواهد شد و تا زمانی که به کمک سوئیچ S_1 آن را از حالت Wait خارج نمائیم در همان حالت باقی خواهد ماند، لذا می‌توان نظیرموردقبلی سیگنالهای روی مسیرهای عمومی را کنترل کرده و با اسیلوسکوپ مشاهده نمود و یا به کمک پروب منطقی بررسی کرد. و بدین ترتیب اجرای برنامه را به وسیلهٔ پردازندهٔ Z80 تعقیب کرد در موردسایر انواع ریز پردازنده نیز می‌توان از مدارهای مشابهی بهره برد.

با توجه به مطالب فوق، به سیستم هیت کیت آموزشی موجود در آزمایشگاه اجزاء کامپیوتر مدار شکل ۱ بطور دائم اضافه شده است ونیز مدار شکل ۲ نیز به صورت قابل نصب پیاده شده است که درهنگام عیب‌یابی می‌توان آن را به سیستم اضافه کرد. و بدین ترتیب در صورت بروز هرگونه خرابی‌ای می‌توان به طور سیستماتیک به رفع عیب اقدام نمود.

۳ـ ایجاد خودآزمائی در سیستم

در سیستم هیت کیت به منظور راه‌اندازی و کنترل کار ریز-



شکل ۳ـ مدار جهت تک مرحله‌ای کردن CPU ی Z80

پردازنده از یک ROM به ظرفیت 1K استفاده شده است در ابتدای کار سیستم به‌طور اتوماتیک Reset شده و رجیستر PC صفر می‌شود آنگاه ریز پردازنده اجرای برنامه را از آدرس صفر واقع در ROM آغاز می‌کند. به منظور ایجاد خودآزمائی در سیستم فوق با ایجاد تغییراتی در برد بجای ROM فوق‌الذکر یک آی سی EPROM (2532) به ظرفیت 4K نصب گردید که از این ظرفیت 2K اول آن برای Monitor سیستم و 2K دوم برای نوشتن برنامهٔ خودآزمائی سیستم استفاده شده است و به کمک یک سوئیچ که در مجاورت آن نصب شده است درهر لحظه فقط نصف حافظهٔ فوق در مدار قرار می‌گیرد. بدین ترتیب که پین A12 آی سی EPROM به این سوئیچ وصل شده است که اگر این سوئیچ به Vcc وصل باشد 2K دوم آن فعال خواهد بود و اگر این سوئیچ به زمین وصل باشد 2K اول آن فعال می‌گردد. لازم به تذکر است که در سیستم هیت کیت، پیش‌بینی‌های لازم جهت گسترش حافظهٔ ROM تا ظرفیت 8K به عمل آمده بود لذا با اضافه کردن حجم حافظهٔ ROM اشکالی در سیستم آدرس‌کردن بوجود نمی‌آید، بنابراین به کمک سوئیچ فوق می‌توان ریزپردازنده را از وجه نرمال به وجه آزمایش منتقل نمود و یا برعکس آن را به وجه نرمال برگرداند.

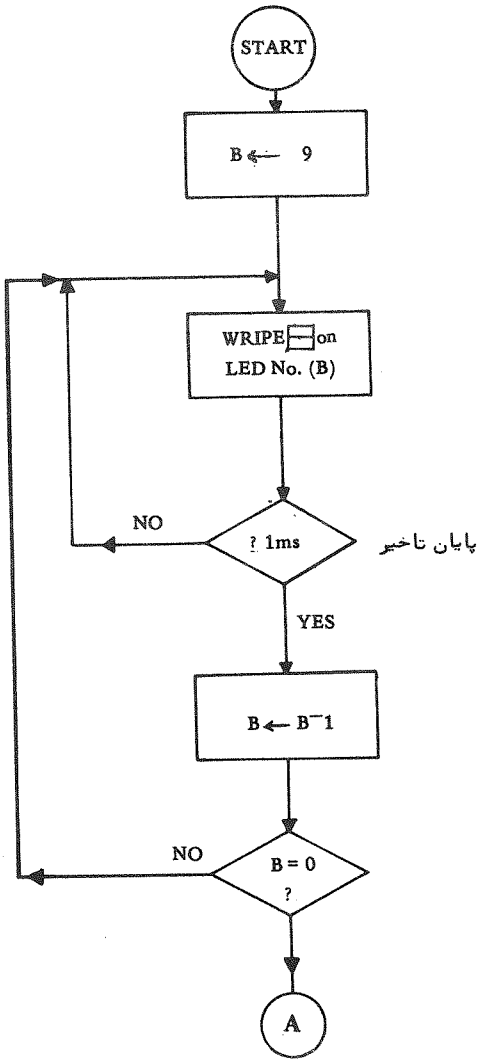
۱ـ۳ آزمایش قسمت نمایش دهنده (Display)

به منظور نمایش حالت سیستم و آگاه شدن از اطلاعات ورودی از جعبه کلید به‌سیستم هیت‌کیت از ۹ عدد سون سگمنت استفاده می‌شود که این سگمنت‌ها به روش مالتی پلکس کردن روشن می‌شوند در این سیستم برای مالتی پلکس‌کردن سون سگمنت‌ها از دروازهٔ FØ و برای نوشتن داخل آنها از دروازهٔ F1 استفاده می‌شود. در ضمن برای خواندن جعبه کلید (Key Board) نیز از همان دروازهٔ FØ استفاده خواهد شد.

اولین قدمی که بایستی در وجه آزمایش پیمود عبارت است از آزمایش‌کردن تمام سون سگمنت‌ها تا مطمئن شویم تمام سون سگمنت‌ها سالم بوده و خاموش بودن آنها در وجه نرمال به خاطر خرابی سگمنت‌های مختلف و یا مدارهای مربوطه نمی‌باشد. چون کنترل سون سگمنت‌ها تحت کنترل برنامه‌ای می‌باشد که توسط وقفه فعال می‌گردد. لذا در ابتدای برنامهٔ آزمایش، پس از غیرفعال کردن وقفه (Interrupt) و تحت کنترل برنامهٔ خودآزمائی، شروع به روشن کردن اولین سون سگمنت می‌نمائیم و پس از گذشت مدت زمانی که از نظر چشم قابل حس باشد آن سون سگمنت،را خاموش نموده و به سراغ سون سگمنت بعدی می‌رویم و همین آزمایش را برای آن تکرار می‌کنیم و پس از آزمایش تمام سون سگمنت‌ها مرحلهٔ بعدی آزمایش را آغاز می‌کنیم.

برای غیرفعال کردن وقفه دو راه وجود دارد یکی استفاده از فرمان DI می‌باشد که علاوه بر غیرفعال کردن وقفه‌ها باعث به کار انداختن بلندگوی کوچکی می‌شود که در داخل سیستم نصب شده است تا به هنگام بروز خطا در کار نرمال سیستم، این بلندگو به صدا درآمده و با فرکانس ۵۰۰ Hz شروع به کار نماید. البته این مکانیزم بیشتر در مقابل خطاهای نرم افزاری حساس می‌باشد و در ضمن فرمان DI نیز که در حالت عادی باعث ایجاد وقفه در سیستم نمایش سیستم می‌گردد باعث به کار افتادن بلندگو خواهد شد.

روش دیگر شامل از کار انداختن سیگنالهای INT10 و INT20, می‌باشد که این سیگنالها جهت ایجاد وقفه در سیستم به منظور مالتی پلکس کردن سون سگمنت‌ها و نیز کنترل برنامهٔ نرم افزاری تک مرحله‌ای به کار رفته‌اند با نوشتن عدد ۱x۰۱ به داخل آی سی IC106 که شامل ۴ عدد Latch نوع D می‌باشد می‌توان هم سیگنالهای فوق را قطع کرد و هم بلندگو را غیرفعال نگهداشت بدین ترتیب ریز پردازنده هیچگونه سیگنالی را که باعث ایجاد وقفه در آن شود دریافت نمی‌نماید البته می‌توان هر دو روش فوق را در ابتدای برنامه به کار برد.



شکل ۴ ــ فلوچارت آزمایش سون سگمنت‌ها

۳.۲ـ خودآزمائی قسمت حافظه

در حال حاضر در این سیستم دو عدد برد حافظهٔ RAM وجود دارد که اولی در آدرس‌های 2ØØØH تا 3FFFH و دومی در آدرس‌های 4ØØØH الی 5FFFH قرار دارد و هربرد 8K ظرفیت دارد از آنجا که سیستم قادر است باحداقل یک برد حافظهٔ RAM به کار بپردازد لذا برنامهٔ آزمایش فقط به آزمایش برد اول می‌پردازد و درصورتی که لازم باشد با تغییر آدرس مکانهای تحت آزمایش می‌توان هر دو برد را تحت آزمایش قرارداد .

برای آزمایش مکانهای مختلف داخل RAM ، ابتدا در تک تک کلمات آن عدد 01010101 را نوشته و سپس محتوای آنها را می‌خوانیم و درصورت آشکار شدن خطا حرف E را روی سگمنت دوم

(از سمت راست) خواهیم نوشت در هر دو حالت برنامه درصورت بروز خطا وارد یک loop شده و دائما" در مکان تحت آزمایش عدد مورد آزمایش را می‌نویسد و در غیر این‌صورت آزمایش مکانهای بعدی را آغاز می‌کند بدین ترتیب می‌توان با آزمایش مکانهای حافظه و خواندن آدرس روی مسیر عمومی آدرس در هنگام وجود خطا ، آی سی معیوب را یافت .

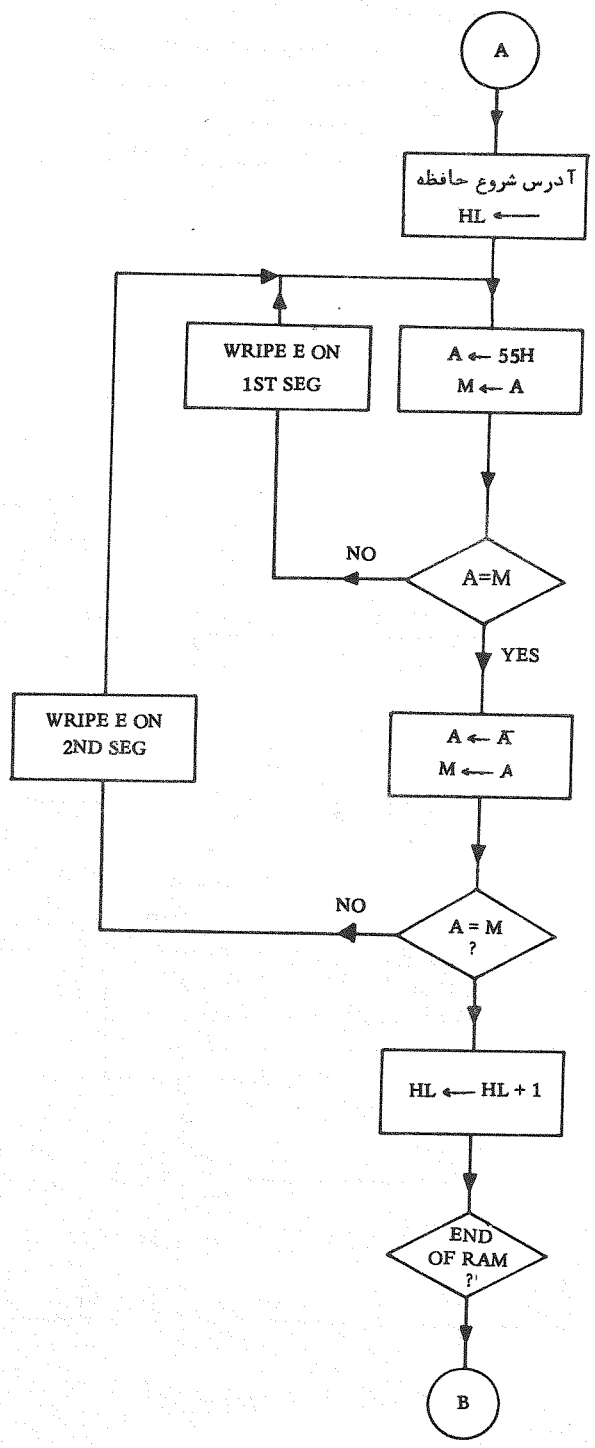
این برنامه جهت آشکار سازی خطاهای ایستائی (خطاهای ایستا روی صفر و ایستا روی یک) بر روی بیت‌های مختلف حافظه به کار می‌رود . فلوچارت این برنامه در شکل ۵ آمده است .

۳.۳ـ آزمایش جعبه کلید (Key Board) و مدارهای وابسته

به منظور آزمایش جعبه کلید می‌بایستی پس از فشار دادن هر کلید ، خروجی‌های بافرهای مربوطه را بررسی و کدهای بدست آمده را با کدهای مجاز مقایسه کرد . درصورتی که کد بدست آمده با یکی از کدهای مجاز تطبیق کرد کد مشخص کنندهٔ آن را برروی سون سگمنت اول (از سمت راست) نوشت و چون کد قرائت شده با کد نمایش داده شده برای هر کلید متفاوت می‌باشند قبل از نمایش می‌بایستی عمل تبدیل کد انجام شود . همزمان با فشار داده شدن هر کلید ، پردازنده کد مربوطه را قرائت و پس از تبدیل ، کد نمایشی مربوطه را برروی سون سگمنت قرار می‌دهد و تافشارداده‌شدن کلید بعدی همین که کد روی سون سگمنت باقی می‌ماند (حتی با برداشتن انگشت از روی کلید ، کد مربوطه نمایش داده می‌شود) . کدهای قرائت شده برای هرکلید بهمراه کدهای نمایش دهندهٔ آن در خانه‌های EPROM به آدرس‌های ØFØØ الی OFIE ذخیره شده‌اند .

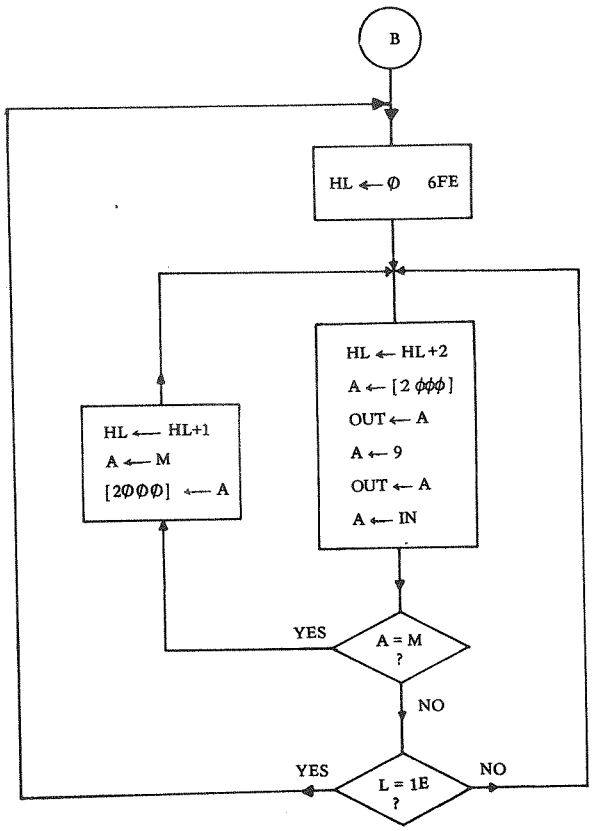
۴ـ نتیجه

در این طرح با توجه به امکانات موجود در سیستم هیت کیت ، حافظهٔ ROM آن که به ظرفیت 1K بود با یکEPROM به ظرفیت 4K بایت (2532) جانشین شد که به کمک کلیدی که در کنار آن نصب شده است همواره 2K اول آن و یا 2K دوم آن در مدار خواهد بود



شکل ۵ ـ فلوچارت برای آزمایش حافظهٔ

محتوای حافظه قبلی که به عنوان Monitor روی سیستم نصب شده بود به این آی سی منتقل شده است و چون Monitor فقط از 1K بایت حافظهٔ ROM استفاده کرده است لذا 1K بایت از نیمهٔ اول خالی می‌ماند که می‌توان در آینده آن را به کار برد در نیمه‌دوم که برنامهٔ خودآزمائی نامیده شده است برنامهٔ آزمایش سیستم جهت رفع خرابی آن گنجانده شده است که با تغییر وضعیت سوئیچ ذکر شده می‌توان نیمه دوم و در نتیجه برنامه‌های موجود در آن را فعال کرد. لازم به تذکر است که گرچه در سیستم اصلی ظرفیتROM فقط 1K بایت بود ولی این ظرفیت قابل گسترش تا 8K بایت پیش‌بینی شده بود که در این طرح تا 2K گسترش یافت.



شکل ۶ــ فلوچارت آزمایش جعبه گلید

ضمیمهٔ ۱ــ برنامهٔ خودآزمائی سیستم

در این برنامه ابتدا با نوشتن عدد باینری 1Ø1Ø1Ø1Ø و 1Ø1Ø1Ø1Ø روی مکانهای حافظهٔ RAM که در آدرسهای 2ØØØH الی 3FFFH قرار گرفته است آنها را آزمایش نموده و پس از اطمینان از سالم بودن حافظهٔ RAM به آزمایش سون سگمنت‌ها می‌پردازیم برای آزمایش سون سگمنت‌ها عدد را روی تک‌تک‌سون سگمنت‌ها و از سمت راست به چپ می‌نویسم.

ADD	MACH CODE	PROGRAM	
ØØØØ	F3	DI	
ØØØ1	21 ØØ2 Ø	LXI H, 2 ØØØH	*BEG OF RAM
ØØØ4	3 E 55	MVI A, 55 H	* TO TEST RAM
ØØØ6	77	MOV M, A	
ØØØ7	BE	CMP M	
ØØØ8	CA 16 ØØ	JZ ØØ16 H	* IF AN ERROR
ØØØB	3 E ØC	MVI A, ØC H	* WRITE E ON SEG
ØØØD	D3 F1	OUT F1	* THE 1ST SEVEN SEG
ØØØF	3E Ø1	MVI A, Ø1 N	
ØØ11	D3 FØ	OUT FØ	* TURN ON 1ST 7 SEG
ØØ13	C3 Ø4 ØØ	JMP ØØØ4 H	* STAY IN LOOP
ØØ16	2 F	CMA	* IF NO ERROR
ØØ17	77	MOV M, A	* TEST IT AGAIN
ØØ18	BE	CMP M	
ØØ19	CA 27 ØØ	JZ ØØ27	* GO TO NEXT LOC
ØØ1C	3 E ØC	MVI A, ØC	* WRITE E ON
ØØ1E	D3 F1	OUT F1	* THE 2ND 7 SEGM
ØØ2Ø	3E Ø2	MVI A, Ø2	
ØØ22	D3 FØ	OUT F Ø	* TURN ON 2ND 7 SEGM
ØØ24	C3 Ø4 ØØ	JMP ØØØ4	
ØØ27	23	INX	* GO TO NEXT RAM LOC
ØØ28	7C	MOV A, H	
ØØ29	FE 4Ø	CPI 4 Ø	* END OF 8K RAM
ØØ2B	C2 Ø4 ØØ	JNZ ØØØ4	* CONTINUE
ØØ2 E	Ø6 Ø9	MVI B, Ø9 H	
ØØ3 Ø	21 ØØØØ	LXI H, ØØØØh	
ØØ33	3 E ØØ	MVI A, ØØH	* TO WRITE ON
ØØ35	D3 F1	OUT F1	* THE NEXT 7 SEG
ØØ37	78	MOV A, B	* PUT REG B IN A
ØØ38	F6 9Ø	ORI 9 ØH	* TURN OFF SPEAKER
ØØ3A	D3 F Ø	OUT F Ø	* TO TURN ON 7 SEGM
ØØ3C	23	INX H	* DELAY ?
ØØ3D	7C	MOV A, H	
ØØ3E	FE 2 Ø	CPI 2 ØH	
ØØ4 Ø	C2 33 ØØ	JNZ ØØ33	* STAY IN LOOP
ØØ43	Ø5	DCR B	* DID YOU TEST ALL
ØØ44	C2 3 ØØØ	JNZ ØØ 3 ØH	* IF YES! GO TO TEST
ØØ47	ØØ	NOP	* KEYBOARD
ØØ48	ØØ	NOP	
ØØ4F	ØØ	NOP	

ADD	MACH CODE	PROGRAM	
ØØ5Ø	21 FE Ø6	LXI H, Ø6 FEH	
ØØ53	23	INX H	
ØØ54	23	INX H	
ØØ55	3 A ØØ2 Ø	LDA 2 ØØØ	* LOAD LAST KEY IN A
ØØ58	D3 F1	OUT F1	* DISPAY LAST KEY
ØØ5A	3E 99	MVI A, 99 H	
ØØ5 C	D3 F1	OUT FØ	
ØØ5E	DBF Ø	IN FØ	READ NEXT KEY
ØØ6 Ø	BE	CMP M	SKIP IF THE SAME KEY
ØØ61	C2 6C ØØ	JNZ	
ØØ64	23	INX H	
ØØ65	7 E	MOV A, M	DECODE NEW KEY
ØØ66	32 ØØ2 Ø	STA 2 ØØØ	* STORE THE KEY IN RAM
ØØ69	C 35 ØØØ	JMP ØØ 5 Ø	
ØØ6 C	3E 1E	MVI A, 1 E H	* DID YOU LOOK ALL
ØØ6 E	BD	CMP L	* TABLE ?
ØØ6 F	C2 53 ØØ	JNZ ØØ53 H	* NO CHECK NEXT KEY
ØØ72	C35 ØØØ	JMP ØØØ5 ØH	* READ NEXT KEY
ØØ75	ØØ	NOP	
ØØ76	ØØ	NOP	

KEY	READ	WRITE	DATA
Ø	FE	Ø1	ØF ØØ, ØF Ø1
1	FC	F3	ØF Ø2, ØFØ3
2	FA	48	ØF Ø4, ØF Ø5
3	F8	6 Ø	ØF Ø6, ØF Ø7
4	F6	32	ØF Ø8, ØF Ø9
5	F4	24	ØF ØA, ØF ØB
6	F2	Ø4	ØF ØC, ØF ØD
7	F Ø	F1	ØF ØE, ØF ØF
8	EF	ØØ	ØF 1 Ø, ØF 11
9	CF	3 Ø	ØF 12, ØF 13
+	AF	1 Ø	ØF 14, ØF 15
−	8F	Ø6	ØF 16, ØF 17
0	6F	CE	ØF 18, ØF 19
/	4F	42	ØF 1 A, ØF 1 B
≠	2F	ØC	ØF 1 C, ØF 1 D
0	ØF	1 C	ØF 1 E, ØF1F