

به نام خدا

حسین حیاتی

گزارش کار پروژه

دستگاهی که در ابتدا طراحی آن هستیم دستگاهی است که وظیفه اجرای دستورات برنامه (به عنوان ورودی) به صورت ریل- Real Time و ذخیره تأثیرات

آن روی حافظه برای استفاده کننده از آن را فراهم می سازد، با توجه به نظریه ماشین های اتوماتا طراحی چنین ماشین محاسباتی در حالت کلی مسئله تصمیم ناپذیر ولی تشخیص پذیر است به این منظور قطعه کدی که روح

اصلی برنامه را اجرا خواهد کرد باید توانایی رفتار با خانه های حافظه به همان شکل که سیستم اصلی با کد اصلی برنامه رفتار میکند را داشته باشد یعنی ۳

مرحله Execute, Decode, etch F ولی این بار به صورت نرم افزاری = مطابق عمل در نسخه

اول همین محصول در دنیای واقعی عمل در کل این دستگاه به این شکل است که ابتدا با فشردن دکمه ADDRESS توسط کاربر

دستگاه در وضعیت نمایش اطلاعات خانه های حافظه را به خود بگیرد و با وارد کردن خانه مورد نظر حافظه توسط کاربر، اطلاعات خانه مشخص

شده نمایش داده شود، ضمناً بهتر است زمانیکه تعداد ارقام وارد شده توسط کاربر از تعداد مشخصی

بیشتر شد، ارقام نشان دهنده آدرس از سمت دیگر خارج شوند، همچنین بهتر است برای جلوگیری از اختلاط کد اصلی و حافظه مورد نیاز برنامه اصلی با

برنامه کاربر، آدرس را از مکان مشخص، مثال H1000 آغاز کنیم (برای حالتی که کد برنامه را رد

ROM می ریزیم و کد دیتا در RAM است این ملاحظه نیاز نیست).

در گام بعدی، بعد از مشخص شدن مکانی که باید در آن داده را بنویسیم، با فشردن دکمه Data میتوانیم داده مورد نظر را در مکان آدرس مورد نظر

بنویسیم (این داده میتواند OpCode مربوط به دستورات یا Operand مربوط به دستورات باشد (و نیز برای ذخیره شدن آن از دکمه های + یا - استفاده

کنیم که یک خانه به بال یا پایین برویم و محتویات آن خانه را عوض کنیم

نمای اولیه طراحی شده برای این پروژه به صورت زیر است :

یکی از معایب پروتئوس این است که کیبورد مناسب به اندازه ۸ در ۸ ندارد و به جای آن از keypad ۴ در ۴ استفاده کردیم و با decompose کردن مدل

و عوض کردن نام کلیدها آنان را به دلخواه خود در آوردیم، اما مشکل اینجاست که برای وارد کردن ورودیها که در مبنای ۱۶ است تعداد

ارقام است. این مشکل در قسمت آدرس با دکمه های + و - قابل حل است ولی برای

قسمت داده میتوان هم محدودیت ورودی اعداد ایجاد کرد و هم میتوان عدد ورودی توسط کاربر را به صورت در مبنای ۱۰ تعریف کرد و

بدین صورت میتوان از ۰ تا ۲۵۵ را در ۴ سون سگمنت باقیمانده نمایش داد و در آخر باقیمانده تقسیم عدد

ورودی توسط کاربر را بر ۲۵۶ به عنوان عدد ورودی مورد استفاده قرار میگیرد..

جزئیات اصلی مدار مطابق خواست سوال طراحی شدهاند: دو دیکودر یکی برای تبدیل اعداد به معادل سگمنتهای ۷ سگمنت و دیگری برای مشخص

کردن اینکه آلن کدام یک از ۷ سگمنتهای باید روشن شود، مورد استفاده قرار میگیرند. این کار باعث میشود که بتوان با تنها استفاده از یکی از پورت های آی سی

۸۰۵۱ کل فرآیند نمایش را کنترل کرد که میدانیم بسیار ارزشمند

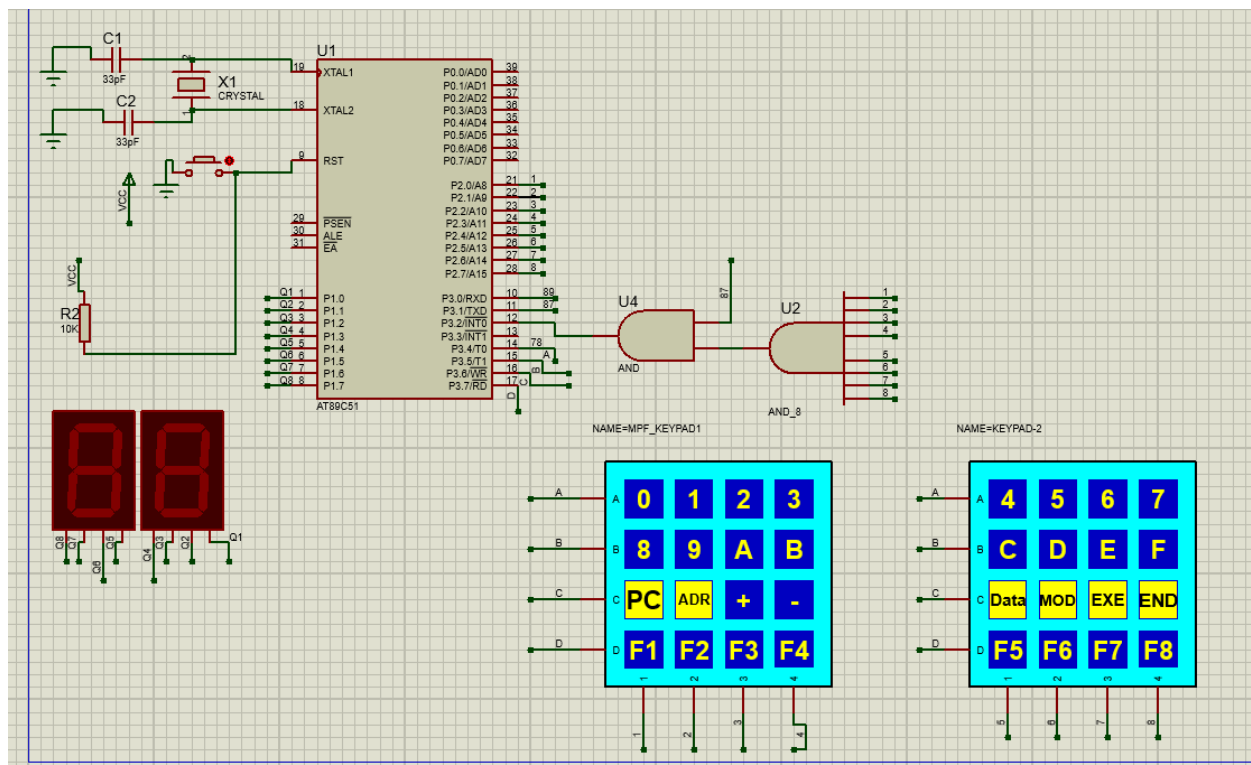
است، زیرا احتمالاً به زودی با مشکل کمبود پورت مواجه خواهیم شد. فرآیند تشخیص کلید زده شده هم مانند آزمایش ۷ام مشغول دریافت ورودی CPU آزمایشگاه این درس با کلیک شدن حداقل یکی از کلیدها، اینترت مربوط به کلید فعال میشود و از کاربر میشود

بعد از اتمام این مراحل، بهتر است فضایی را برای دیتایی که باید برای نمایش روی ۷ سگمنتها استفاده شوند در محلی ذخیره کنیم. هر رقم معادل یک بایت است و با در نظر گرفتن ۸ بایت برای نمایش این اعداد میتوان پیوسته این اعداد را نمایش داد و از طریق برنامه تنها محتویات این خانهها را عوض کنیم که این موضوع باعث عوض شدن مقادیری که توسط ۷ سگمنتها نمایش داده میشوند نیز میشود. ضمناً از آنجایی که عملیات نمایش تنها با یک سری پورت انجام میشود، کافی است ۸ خانه حافظه را در نظر بگیریم و در نیمبایت کم ارزش آن عدد مورد نظر برای نمایش روی ۷ سگمنت و نیمبایت بالایی مربوط به کنترل اندیس ۷ سگمنت مورد نظر است. مثلاً فرض کنیم میخواهیم اعداد 12345678 را روی این مجموعه نمایش دهیم، برای این کار کافی است ۸ خانه حافظه را مطابق زیر مقداردهی کنیم

برای زمانی است که در حالت CP روی این دستگاه برای تعویض مد کاری احتمالی دستگاه است و نیز دکمه Mod دکمه کلیک شود و در نتیجه آن، برنامه کاربر از همان آدرس شروع به اجرا شود DAT به جای ADDRESS

مشکل دیگری که در این طراحی وجود دارد، مشکل بعضی دستورات خاص که روند خطی برنامه را مختل میکنند است. به گونهای غیرقابل بازگشت عوض میشود. در این حالت PC و یا دستورات شرطی که در آنها مقدار CALL دستورات مانند برای ذخیره Stack) با فرض بر اینکه میتوانیم حافظه داده و کد مشترک داشته باشیم) میتوانیم از بانکهای رجیستری و یا قبل از اجرای دستورات برنامه کاربر استفاده کنیم و بعد از اجرای آنها به روند اصلی برنامه بازگردیم. اما مشکل PC کردن اصلی اینجا است که طراحی ۸۰۵۱ به ما این اجازه را نمیدهد که حافظه کد و داده یکسان داشته باشیم و به صورت معمولی داخل خود در نظر بگیرد. و این ارتباط هم در حالت کلی غیر قابل RAM داخل خود و حافظه برنامه را ROM حافظه کد را را به RAM نوشته شود که حافظه ROM در bootloader شکستن است. تنها کاری که میتوان انجام داد این است که یک عنوان حافظه برنامه تشخیص دهد ولی در این حالت این حافظه غیر قابل تغییر است. پس کاری که میتوان انجام داد این است که ذخیره کنیم و تمام مراحل شبیهسازی RAM را در نظر بگیریم به عنوان حافظه برنامه و برنامه کاربر را در ROM همان حافظه کد را انجام دهیم

یکی از راهحلهای احتمالی برای حل مشکل ورودیها (چون ممکن است در درس ساز شوند) استفاده از دو ماژول کلید و متصل کردن آنها به یکدیگر است. به این صورت که ردیفهای این دو صفحه کلید را به هم متصل میکنیم و ستونها را از هم مجزا میکنیم. به این ترتیب یک صفحه کلید بزرگتر با ابعاد ۴ در ۸ در اختیار داریم که به ما ۳۲ کلید میدهد. مشکلی که در این حالت به وجود میآید این است که در صورت نیاز به استفاده از حافظه خارجی با کمبود پورت روبهرو هستیم که برای رفع آن میتوان کنترل کرد که در نتیجه تنها نیاز به استفاده از ۸ پورت بود.



کار با میکروپروفسور:

❑ حتماً برای شروع کار باید PC زده شود (هر کلید دیگری نادیده گرفته میشود).

❑ دو عدد زده شود که همان آدرس شروع هستند (پس از زدن عدد دوم به مدت چند ثانیه آدرس شروع کد روی دو سون سگمنت نشان داده میشود) (هر کلیدی غیر از اعداد نادیده گرفته میشود).

❑ پس از نشان دادن آدرس شروع حتماً باید EXE زده شود (هر کلید دیگری نادیده گرفته میشود).

❑ دو عدد زده شود که آپکود اولین دستور کد هستند (پس از زدن عدد دوم به مدت چند ثانیه آدرس شروع کد روی دو سون سگمنت نشان داده میشود) (هر کلیدی غیر از اعداد نادیده گرفته میشود).

❑ پس از نشان دادن آپکود حتماً باید EXE زده شود (هر کلید دیگری نادیده گرفته میشود).

❑ برای انتخاب آدرس بعدی میتوان چند کار انجام داد:

- ❑ کلید + زده شود که بالفاصله پس از آن چند ثانیه آدرس بعدی روی سون سگمنت نشان داده میشود.
- ❑ کلید - زده شود که بالفاصله پس از آن چند ثانیه آدرس قبلی روی سون سگمنت نشان داده میشود.
- ❑ در دو حالت بال نیاز به زدن EXE پس از نشان دادن آدرس نیست.
- ❑ میتوان دو عدد وارد کرد و پس از نمایش آن روی سون سگمنت EXE را زد.
- ❑ میتوان کلید END را زد که یعنی وارد کردن کد به اتمام رسیده و کدهای وارد شده اجرا خواهند شد (در این مرحله قرارداد میکنیم که کاربر باید آخرین آپکود خود قبل از زدن END را 00 وارد کند).

❑ بعد از زدن END عددی روی سون سگمنت نشان داده میشود که آن عدد نتیجه محاسبات کد نوشته شده است و اگر 88 نشان داده شد یعنی یکی از آپکودهای وارد شده اشتباه بوده است.

❑ بعد از نمایش نتیجه روی سون سگمنت، برای کد نوشتن مجدد باید شبیهسازی متوقف شود و دوباره آن را اجرا کرد.

