به نام خدا

گزارشکار پروژه درس ساختارهای کامپیوتر و میکرو پروسسور

نام و نام خانوادگی : امیرحسین دهقانپور شماره دانشجویی : 400101186

بخش 1:

دستگاهی که قصد طراحی آن را داریم دستگاهی است که وظیفه اجرای دستورات برنامه (به عنوان ورودی) به صورتtime – Real و ذخیره تاثیرات آن روی حافظه برای استفاده کننده از آن را فراهم میسازد. با توجه به نظریه ماشین های اتومات طراحی چنین ماشین محاسبه ای در حالت کلی یک 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 شکلی که سیستم اصلی با کد اصلی برنامه رفتار میکند را داشته باشد (یعنی 3 مرحله Fetch, Decode, Executeولی این بار به صورت نرم افزاری) مطابق عملکرد نسخه اول همین محصول در دنیای واقعی عملکرد کلی 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 برمیگردد و با وارد کردن خانه مورد نظر حافظه توسط کاربر، اطلاعات خانه مشخص شده نمایش داده شود، ضمنا بهتر است زمانی که تعداد ارقام وارد شده توسط کاربر از تعداد مشخصی بیشتر شد، ارقام نشان دهنده آدرس از سمت دیگر خارج شوند، همچنین بهتر است برای جلو گیری از اختلاط کد اصلی و حافظه مورد نیاز برنامه اصلی با برنامه کاربر ، آدرس را از مکان مشخصی مثلا

(H1000)آغاز کنیم .(برای حالتی که کد برنامه را در ROM میریزیم 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 Data میتوانیم داده مورد نظر را در مکان آدرس مورد نظر بنویسیم (این داده میتواند OpCode مربوط به دستورات یا Operandمربوط به دستورات باشد) و نیز برای ذخیره شدن آن از دکمه های + یا – استفاده بکنیم که یک خانه به بالا یا پایین برویم و محتویات آن خانه ها را عوض کنیم.

*نمای اولیه طراحی این پروژه:

چالش 1 :

یکی از معایب پروتئوس این است که کیبورد مناسب به اندازه ۸ در ۸ ندارد و به جای آن از ۴ keypad در ۴ استفاده کردیم و با decomposeکردن مدل و عوض کردن نام کلیدها آنان را به دلخواه خود در آوردیم، اما مشکل اینجاست که برای وارد کردن ورودی ها 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 باقی مانده نمایش داد و در آخر باقی

مانده تقسیم عدد ورودی توسط کاربر را بر ۲۵۶ (bit8) به عنوان عدد ورودی مورد استفاده قرار میگیرد . اجزای اصلی مدار مطابق خواست سوال طراحی شده اند ۲ دیکودر برای تبدیل اعداد به معادل سون سگمنت های ۷ سون سگمنت و دیکودر برای مشخص کردن اینکه الآن کدام یک از ۷ سونسگمنت ها میبایست روشن شود مورد استفاده قرار میگیرد، این کار باعث میشود که بتوان با تنها استفاده از یکی از پورت های آی سی ۸۰۵۱ کل فرآیند نمایش را کنترل کرد که می دانیم بسیار ارزشمند است.

چالش 2:

احتمالا به زودی با مشکل کمبود پورت مواجه خواهیم شد.فرآیند تشخیص کلید زده شده هم مانند آزمایش ۷ ام ازمایشگاه با کلیک شدن حداقل یکی از کلید ها Interrupt مربوط به کلید فعال میشود و CPU مشغول دریافت ورودی از کاربر میشود . بعد از اتمام این مراحل بهتر است فضایی را برای دیتایی که باید برای نمایش به روی ۷ سون سگمنت ها استفاده بشوند را در محلی ذخیره کنیم (هر رقم معادل یک بایت است و با در نظر گرفتن ۸ بیت برای نمایش این اعداد میتوان پیوسته این اعداد را نمایش داد و از طریق برنامه تنها محتویات این خانه ها را عوض کنیم که این موضوع باعث عوض شدن مقادیری نمایشی توسط سون سگمنت ها نیز می شود)

چالش 3:

از آنجایی که عملیات نمایش تنها با ۱ سری پورت انجام میشود کافی است ۸ خانه حافظه را در نظر بگیریم و در nible کم ارزش آن عدد مورد نظر برای نمایش روی segment – 7 وnible بالایی مربوط به کنترل اندیس segment – 7 مورد نظر است. مثلا فرض کنیم میخواهیم اعداد 12345678 را روی این مجموعه نمایش دهیم برای این کار کافی است ۸ خانه حافظه را مطابق زیر مقداردهی کنیم:

Data(Adress) : 00000001(2000H) / 100010010(2001H) / 00100011(2002H) /00110100(2003H) / 01000101(2004H) / 01010110 (2005H) /01100111(2006H) /01111000(2007H)

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

بخش 2 :

مشکل دیگری که در این طراحی وجود دارد مشکل بعضی دستورات خاص که روند خطا برنامه را مختل میکنند است دستوراتی مثل CALLو یا دستورات شرطی که در آن ها مقدارPC به گونه ای غیرقابل بازگشت عوض میشود، در این حالات(با فرض بر اینکه میتوانیم حافظه داده و کد مشترک داشته باشیم) میتوانیم از بانک های

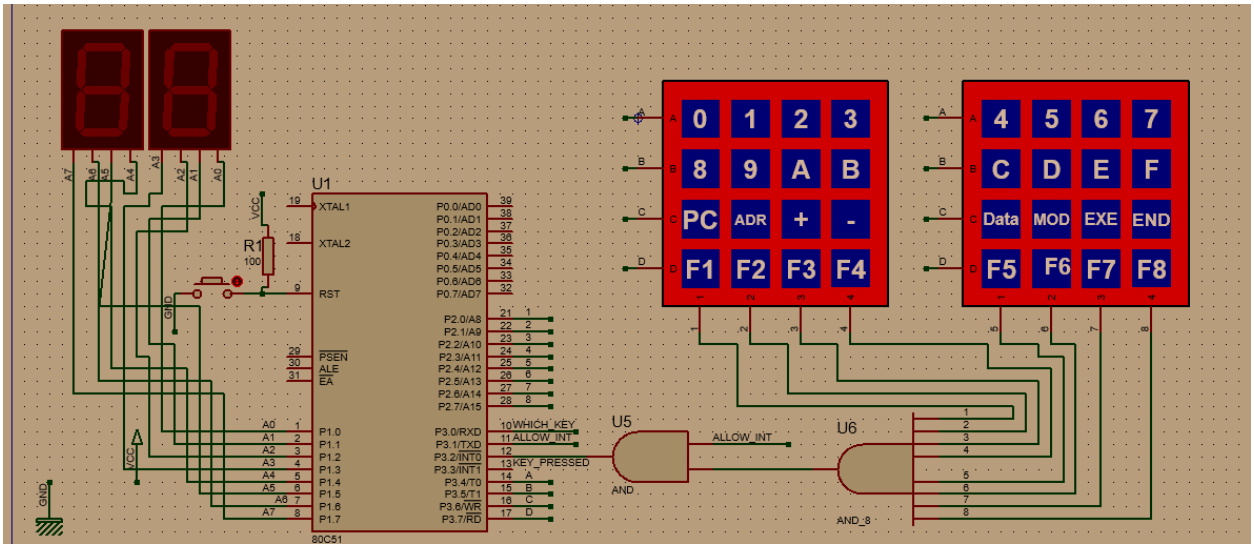
رجیستری و یا Stak برای ذخیره کردن PC قبل از اجرای دستورات برنامه کاربر استفاده کنیم و بعد از اجرای آن ها به روند اصلی برنامه بازگردیم اما مشکل اصلی اینجا است که طراحی ۸۰۵۱ به ما این اجازه را نمیدهد که حافظه کد و داده یکسان داشته باشیم و به صورت معمولی حافظه کد را ROM داخل خود و حافظه برنامه را RAM داخل خود در نظر میگیرد و این ارتباط هم در حالت کلی غیر قابل شکستن است، تنها کاری که میتوان انجام داد این است که یک bootloader ROM در نوشته شود که حافظه RAM را به عنوان حافظه برنامه تشخیص دهد ولی در این حالت این حافظه غیر قابل تغییر است . پس کاری که میتوان انجام داد این است که همان حافظه ROMرا در نظر بگیریم به عنوان حافظه برنامه و برنامه کاربر را در RAMذخیره کنیم و تمام مراحل شبیه سازی کد را انجام دهیم یکی از راه حل های احتمالی برای حل مشکل ورودی ها (چون ممکن است دردسر ساز شوند)

استفاده از ۲ ماژول کلید و متصل کردن آن ها به یک دیکودر است. به این صورت که ردیف های این دو صفحه کلید را به هم متصل میکنیم و ستون ها 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 را میدهد: مشکلی که در این حالت به وجود می آید این است که در صورت نیاز به استفاده از

حافظه خارجی با کمبود پورت روبه رو هستیم که برای رفع آن میتوان ساختار کلید ها را به استفاده از BDL ۲ کنترل کرد که در نتیجه تنها نیاز به استفاده از ۸ پورت بود.

بخش 3 :

برای طراحی سیستم نیاز به BCD 7 _ SEG / CUSTOMIZE KEYPAD / 100 OHM RES/BUTTON/AND GATE/8051 است که در نهایت شماتیک شبیه سازی بدین شکل است:



همچنین میکروپروسسوری که طراحی کرده ایم قابلیت انجام دستورات زیر را دارد :

- MOV
- CALL
- AJUMP
- RET
- ADD
- ADDC
- SUB
- INC
- DEC

چند نمونه از خروجی های میکروپروسسور نیز به صورت زیر میباشد: