

شبیه سازی ماشین حساب با پردازنده 8086



دانشگاه شهید باهنر کرمان

زمستان 1402

دانشجویان: کاوه افشار – حمیدرضا بازاریار

استاد: دکتر مهدیه قزوینی

درس: ریزپردازنده و زبان اسمبلی

فهرست

1.	مقدمه	3
2.	معرفی ابزارها و قطعات	4
2.1.	صفحه نمایش	4
2.2.	صفحه کلید	5
2.3.	رمزگشای کلید	6
2.4.	بافر	6
2.5.	لچ	7
2.6.	فلیپ فلاپ	7
2.7.	تراشه 74LS245	8
2.8.	تراشه 8255	8
2.9.	میکروپردازنده 8086	9
3.	روش کار و پیاده سازی	11
3.1.	مدارهای میکروپردازنده و ایجاد باس مشترک آدرس و داده	11
3.2.	جداسازی آدرس و داده	12
3.3.	ایجاد سیگنال های فعال سازی و رمزگشایی آدرس پورت ها	14
3.4.	صفحه کلید و مدارهای خواندن از صفحه کلید	17
3.5.	صفحه نمایش و مدارهای نوشتن در صفحه نمایش	18
4.	عملکرد و راهنمای کاربر	20
	دسترسی به پروژه	22
	ارتباط با ما	23

1. مقدمه

یکی از ساده ترین و پرکاربرد ترین وسیله هایی که در زندگی روزمره مان برای انجام محاسبات و عملیات های ریاضی ساده استفاده می کنیم، ماشین حساب است. ماشین حساب ساده، یک قطعه الکترونیکی است که قابلیت انجام عملیات های مقدماتی ریاضی، شامل جمع، تفریق، ضرب و تقسیم را بین دو عدد را دارد. برای این منظور در نرم افزار پرتئوس، مدارها و رابط کاربری یک ماشین حساب ساده شبیه سازی شده است. این دستگاه شبیه سازی شده قابلیت انجام محاسبات و عملیات های نام برده را روی دو عدد 4 رقمی دارا می باشد.

نرم افزار پرتئوس، یکی از نرم افزارهای رایج برای شبیه سازی مدارها می باشد. در این نرم افزار کتابخانه های زیادی وجود دارد که تمامی سخت افزار های مورد نیاز را می توان یافت. همچنین به سادگی می توان کدهای اسمبلی نوشته شده را در میکروپردازنده برنامه ریزی (program) کرد.

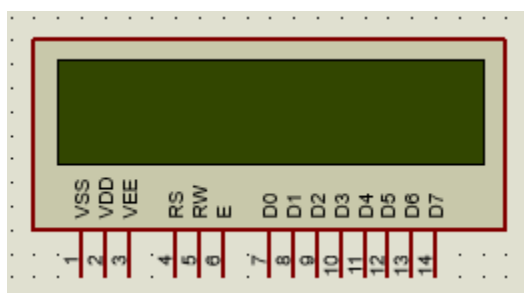
شبیه سازی ماشین حساب، علاوه بر ظاهر ساده آن، مدارهای زیادی برای رمزگشایی آدرس ها، جدا سازی آدرس و داده در میکروپردازنده 8086، تولید سیگنال های خاص، راه اندازی صفحه نمایش و راه اندازی صفحه کلید نیاز دارد. برای طراحی مدارهای ذکر شده، به قطعات الکترونیکی زیادی نیاز داریم؛ بنابراین در ابتدا به معرفی و تشریح مختصر طرز کار و استفاده از قطعات می پردازیم. در فصل بعد شیوه طراحی و پیاده سازی مدارها را بررسی می کنیم. در فصل چهارم تصاویر و نتایج و شیوه کار آن را آورده ایم.

2. معرفی ابزارها و قطعات

در ساخت ماشین حساب، از سخت افزار های زیادی برای طراحی مدارها استفاده کرده ایم. این سخت افزار ها شامل قطعاتی برای نمایش داده ها، خواندن داده ها، رمزگشایی، جدا سازی و یا نگهداری مقدار ها استفاده شده اند. در ادامه به معرفی تک تک این قطعات می پردازیم.

2.1. صفحه نمایش¹

یکی از کاربردی ترین اجزا در یک ماشین حساب، صفحه نمایش می باشد. از صفحه نمایش برای نمایش نتایج، نمایش اعداد وارد شده و همچنین نمایش عملیات انتخاب شده به استفاده می شود. یکی از رایج ترین صفحه نمایش مورد استفاده در پروژه ها، صفحه نمایش [LM016L](#) است. این صفحه نمایش دارای 8 پایه برای انتقال داده می باشد. همچنین این تراشه دارای پایه فعال ساز، پایه خواندن و نوشتن، پایه کنترل و 3 پایه برای تغذیه است. این صفحه نمایش توانایی نوشتن دو خط کاراکتر و در هر خط هشت کاراکتر می باشد.



شکل 2.1 : تصویری از تراشه LM016L در پرتئوس

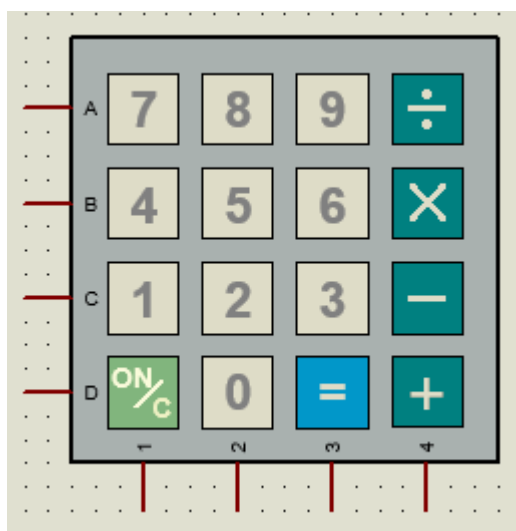
زمانی که پایه RS این پردازنده در سطح پایین (0 منطقی) قرار گیرد، پایه های داده دستور را به صفحه نمایش منتقل می کنند. برای مثال برای تعیین مکان نمایش گر، نوشتن در خط اول یا دوم باید به کمک دستورات برنامه نویسی مناسب داده مناسب را روی بیت های داده این تراشه قرار دهیم. زمانی که در سطح بالا (1 منطقی) قرار گیرد، مقادیر به عنوان داده در نظر گرفته می شود. پایه RW زمانی که در سطح پایین قرار گیرد، داده ها از کنترل کننده به صفحه نمایش منتقل می شوند اما اگر در سطح بالا باشد، داده ها از صفحه نمایش به کنترل کننده منتقل می شوند.

¹ LCD

2.2. صفحه کلید^۲

صفحه کلید، مهمترین جز برای ارتباط کاربر با ماشین حساب می باشد. از طریق صفحه کلید اعداد به دستگاه اعلام می شوند و همچنین عملیات مورد نظر انتخاب می شود. برای این منظور از یک صفحه کلید ماتریسی 4x4 استفاده می کنیم. این صفحه کلید شامل ارقام 0 تا 9 و عملگر های جمع، تفریق، ضرب، تقسیم و مساوی و همچنین یک کلید برای راه اندازی مجدد می باشد.

در نرم افزار پرتئوس^۳، در دسته ابزار های صفحه کلید، صفحه کلیدی تحت عنوان keypad-smallcalc وجود دارد. این ابزار 16 کلید اشاره شده را دارا است. همچنین این ابزار دارای 8 پایه ی خروجی می باشد. پایه های 1 و 2 و 3 و 4 که در پایین آن قرار دارند و همچنین پایه های A و B و C و D که در کنار آن قرار دارند، برای تعیین سطر و ستون کلید فشار داده شده به کار می روند. برای سادگی تعیین کلید فشار داده شده، تراشه خاصی برای تعیین کلید به کار می رود. در ادامه به معرفی این تراشه می پردازیم.



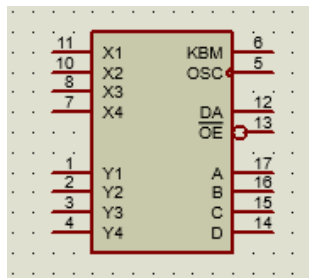
شکل 2.2: تصویری از keypad small calc در پرتئوس

^۲ Keypad

^۳ proteus

2.3. رمز گشای کلید

رمز گشای کلید یا همان تراشه [MM74C922](#)، یک رمز گشا⁴ برای یک صفحه کلید 4x4 است. این تراشه دارای 4 ورودی با نام های X1,X2,X3,X4 و 4 ورودی دیگر با نام های Y1,Y2,Y3,Y4 می باشد. ورودی های نام برده به سطر و ستون صفحه کلید متصل می شوند. همچنین این رمز گشا، به وسیله 4 خروجی A,B,C,D یک عدد باینری 4 بیتی (معادل 0 تا 15) که بیانگر کلید فشرده شده است را در اختیار ما قرار می دهد. همچنین این تراشه دارای دو سیگنال کنترلی OE⁵ و DA⁶ است. سیگنال اول برای تعیین اینکه خروجی ها فعال شوند به کار می رود. این سیگنال در سطح پایین فعال می شود. سیگنال دوم توسط تراشه تعیین می شود. این سیگنال در زمانی که داده ای توسط صفحه کلید به ورودی ها اعمال می شود در مدت زمان کوتاهی فعال و سپس غیرفعال می شود.



شکل 2.3 : تصویری از تراشه MM74C922 در پرتئوس

2.4. بافر⁷

بافر سه حالت (به اختصار بافر) برای تعیین اینکه یک سیم به سیم دیگر متصل باشد به کار می رود. زمانی که سیگنال کنترلی بافر فعال باشد، هر مقداری در ورودی قرار بگیرد در خروجی ظاهر می شود اما وقتی سیگنال کنترلی بافر غیرفعال باشد، اتصال بین ورودی و خروجی قطع می شود و خروجی هیچ مقداری ندارد.

زمانی که قرار است یک مقدار را روی دیتاباس قرار دهیم، برای اینکه همزمان چند دستگاه و سیم روی دیتاباس مقدار قرار ندهند، بین سیم و دیتاباس بافر قرار می دهیم.

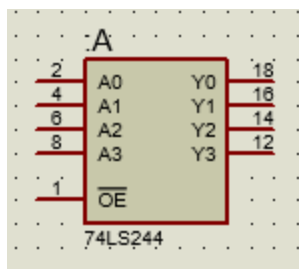
تراشه [LS24474](#) دارای 4 بافر می باشد. در نتیجه این تراشه 4 ورودی و 4 خروجی است. همچنین سیگنال کنترلی این بافرها توسط پایه ی OE تعیین می شود. این سیگنال در سطح پایین فعال می شود.

⁴ Encoder

⁵ Output Enable

⁶ Data Available

⁷ 3 state buffer

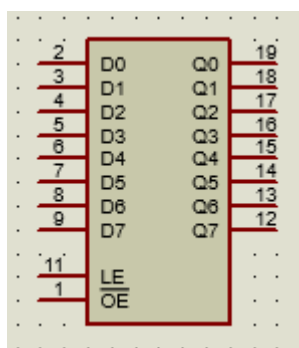


شکل 2.4: شماتیک تراشه 74LS244 در پرتئوس

2.5. لچ⁸

این قطعه الکترونیکی، یک نوع حافظه موقت است. این قطعه مقدار ورودی را زمانی که فعالساز آن فعال باشد در خروجی نمایش می دهد. همچنین زمانی که فعالساز آن قطع شود، خروجی تغییری نمی کند تا زمانی که دوباره فعال ساز فعال شود. همچنین این قطعه دارای یک سیگنال کنترلی به اسم OE می باشد. این سیگنال تعیین می کند که خروجی های طعه فعال باشند یا خیر.

تراشه [74HC573](#) یک لچ با 8 ورودی و 8 خروجی می باشد. سیگنال OE این تراشه در سطح پایین و فعالساز این تراشه (LE) در سطح 1 فعال می شود.



شکل 2.5: شماتیک تراشه لچ 8 ورودی در پرتئوس

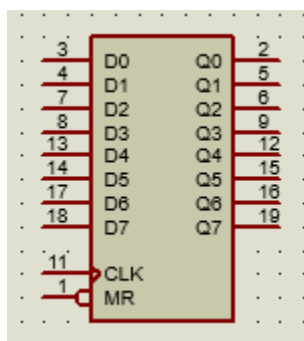
2.6. فلیپ فلاپ⁹

فلیپ فلاپ، همانن لچ ها یک نوع حافظه موقت هستند. فلیپ فلاپ ها انواع گوناگونی دارند که از دی فلیپ فلاپ (D-Flip Flop) استفاده می کنیم. این قطعه هنگامی که کلاک بخورد، مقدار ورودی اش در خروجی قرار

⁸ Latch

⁹ Flip Flop

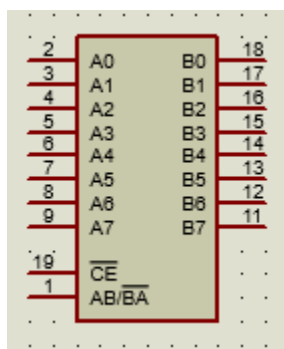
می گیرد تا زمانی که دوباره کلاک بخورد. تراشه [74273](#) نمونه ای از فلیپ فلاپ 8 ورودی می باشد. این تراشه سیگنالی تحت عنوان MR را دارا می باشد. این سیگنال تعیین می کند که خروجی روشن یا خاموش باشد.



شکل 2.6 : شماتیک تراشه 74273 یا همان دی فلیپ فلاپ در پرتئوس

2.7. تراشه 74LS245

تراشه [74LS245](#) عملکردی همانند بافر دارد. این تراشه برای اتصال یک دستگاه یا تراشه ی دیگر به دیتا باس که ارتباط دو طرفه با باس (هم روی بای داده می گذارند و هم از روی باس داده بر می دارند) دارند استفاده می شود. برای کنترل اینکه چه زمانی قرار است داده از باس خوانده شود و چه زمانی داده روی باس نوشته شود، سیگنال AB در نظر گرفته می شود.

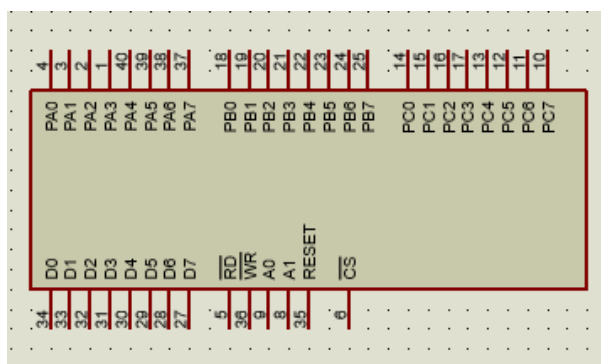


شکل 2.7 : شماتیک تراشه 74LS245 در پرتئوس

2.8. تراشه 8255

رابط قابل برنامه ریزی 8255 یک تراشه بسیار متداول است که امروزه کاربردهای زیادی یافته است. این تراشه دارای 24 پایه برای I/O است که در گروههای 12 پایه ای قابل برنامه ریزی هستند. هر گروهی می تواند در به حالت مجزا کار کند I/O : ساده، IO استروب شده و I/O دو طرفه، 8255 قادر است هر وسیله I/O موازی سازگار با TTL را به آسانی به ریز پردازنده ارتباط دهد. برای ایجاد برخی از سیگنال های کنترلی

دستگاه ها، قرار دادن داده ها در خروجی از این تراشه قابل برنامه ریزی استفاده می کنیم. این تراشه 3 گروه 8 بیتی پورت انتقال داده دارد که در یک لحظه تنها یکی از آن ها می تواند فعال باشد. همچنین این تراشه دارای 8 بیت برای دریافت یا قرار دادن داده روی باس دارد. به کمک دو پایه ی A_1, A_0 می توان انتخاب کرد که کدام پورت باید انتخاب شود. همچنین حالت قابل انتخاب دیگر حالت کنترل تراشه است. در این حالت داده های ورودی به رجیستر کنترلی داخل تراشه منتقل می شوند و تنظیمات کنترلی تراشه بر آن اساس انجام می شود. دو سیگنال RD, WR که هر دو در سطح پایین فعال می شوند، برای تعیین نوع عملیات (نوشتن یا خواندن) به کار می روند. دو پایه ی دیگر این تراشه، پایه های انتخاب تراشه (CS) و پایه راه اندازی مجدد ($RESET$) می باشد.

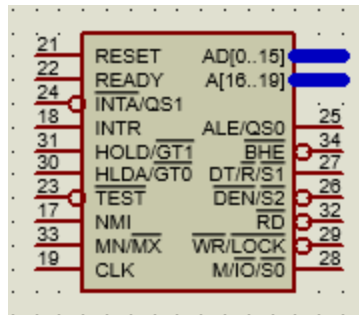


شکل 2.8 : تراشه 8255 در پرتئوس

2.9. میکروپردازنده 8086¹⁰

در هر پروژه ای، بخشی از پروژه وظیفه کنترل و مدیریت سایر اجزا را به عهده دارد. این کار توسط پردازنده و در پروژه های کوچک تر توسط میکروپردازنده انجام می شود. میکروپردازنده 8086، یک میکروپردازنده 40 پایه با دیتاباس 16 بیتی و آدرس باس 20 بیتی است. از آن رو که پایه های این پردازنده در زمان های متفاوت عملکرد متفاوتی دارند، باید در هنگام اتصال دستگاه ها به باس ها، از بافر برای قرار دادن داده و از لچ برای نگهداری داده و جلوگیری از هدر رفتن آن استفاده کنیم. برای تعیین این که چه عملی قرار هست انجام شود یا چه نوع داده یا سیگنالی روی پایه ها قرار گرفته اند، از 16 پایه این پردازنده برای سیگنال های کنترلی استفاده شده است.

¹⁰ Microprocessor



شکل 2.9 : تصویری از میکروپردازنده 8086 در پرتئوس

3. روش کار و پیاده سازی

در این فصل به کمک سخت افزار و تراشه های معرفی شده و همچنین گیت های منطقی، چگونگی طراحی مدارهای مورد نیاز را شرح می دهیم. در ابتدا در یک تقسیم بندی کلی، می توانیم سخت افزار پروژه مان را به 5 بخش تقسیم کنیم:

1. میکروپردازنده و ایجاد باس مشترک آدرس و داده
2. جداسازی آدرس و داده
3. ایجاد سیگنال های فعال سازی و رمزگشایی آدرس پورت ها
4. صفحه کلید و مدارهای خواندن از صفحه کلید
5. صفحه نمایش و مدارهای نوشتن در صفحه نمایش

در ادامه به تشریح هر یک از این بخش ها و شیوه عملکرد و ساخت آن هارا شرح می دهیم.

3.1. مدار های میکروپردازنده و ایجاد باس مشترک آدرس و داده

میکروپردازنده 8086 را از قسمت ابزار ها انتخاب و آن را به پروژه اضافه کنید. در قسمت تنظیمات تراشه فرکانس داخلی را انتخاب کرده و مقدار آن را روی 1MHz قرار دهید. با این کار فرکانس کاری پردازنده را روی این مقدار تنظیم کرده ایم. در این فرکانس پردازنده بهترین عملکرد را برای این پروژه خواهد داشت. در گام بعدی باید برخی پایه ها را مقدار دهی کنید. پایه ی راه اندازی مجدد¹¹ را به زمین اتصال دهید. همان طور که می دانید، در طول اجرا پروژه ماشین حساب نیازی به استفاده از این پایه و راه اندازی مجدد پردازنده نداریم، پس مقدار این پایه را با اتصال به زمین برابر 0 منطقی قرار می دهیم تا از راه اندازی مجدد پردازنده جلوگیری کنیم.

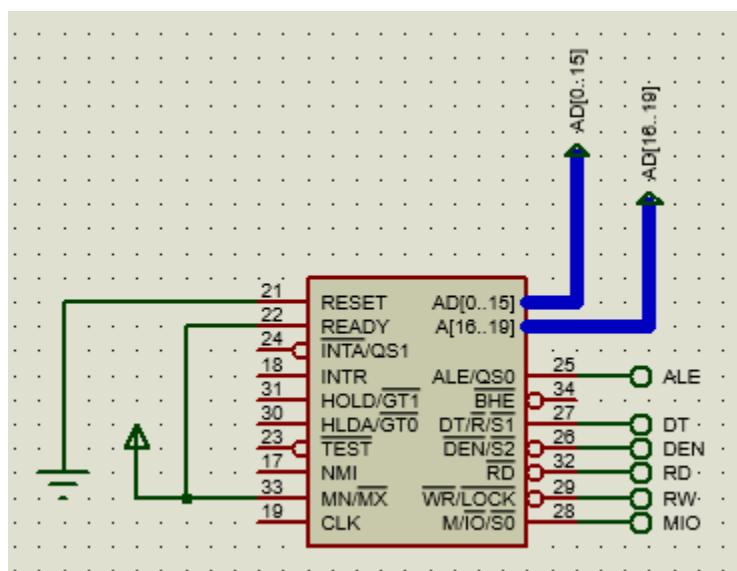
در گام بعدی باید تعیین کنید که پردازنده در حالت بیشینه یا کمینه فعالیت می کند. برای این منظور از آن رو که پردازنده در حالت کمینه و به صورت مستقل مورد استفاده قرار می گیرد، پایه ی 33 پردازنده (پایه ی mn/mx) را به منبع تغذیه اتصال دهید. با این کار 1 منطقی را روی این پایه قرار گرفته و سیگنال حالت کمینه فعال می شود.

¹¹ RESET

در مرحله بعد پایه ی آمادگی^{۱۲} پردازنده را مقدار دهی کنید. از آن رو که از حافظه یا سخت افزار خاصی که تاخیری برای قرار دادن داده روی باس ها داشته باشد استفاده نمی شود، این پایه را نیز به 1 منطقی یا همان منبع تغذیه اتصال دهید.

برای سادگی در دسترسی به پایه های داده و آدرس، آن ها را در یک باس قرار دهید و نامی متناسب برای آن انتخاب کنید. در اینجا برای پایه های مشترک بین داده و آدرس، باس $AD[0..15]$ و پایه های مشترک بین آدرس و وضعیت باس $A[16..19]$ در نظر گرفته شده است.

از مقدار های پایه های ALE در مدارهای لچ و جداسازی آدرس، از پایه های RD و WR برای ایجاد برخی سیگنال های فعال سازی برخی تراشه ها و همچنین تعیین وضعیت برخی مدارها مانند مدار نوشتن در صفحه نمایش استفاده می شود. همچنین پایه های M/IO و DT و DEN نیز برای تولید برخی سیگنال ها به کار می روند. برای سادگی و جلوگیری از پیچیده شدن اتصالات، آن ها را نشانه گذاری کنید و روی آن اتصال، برچسب مناسبی قرار دهید.



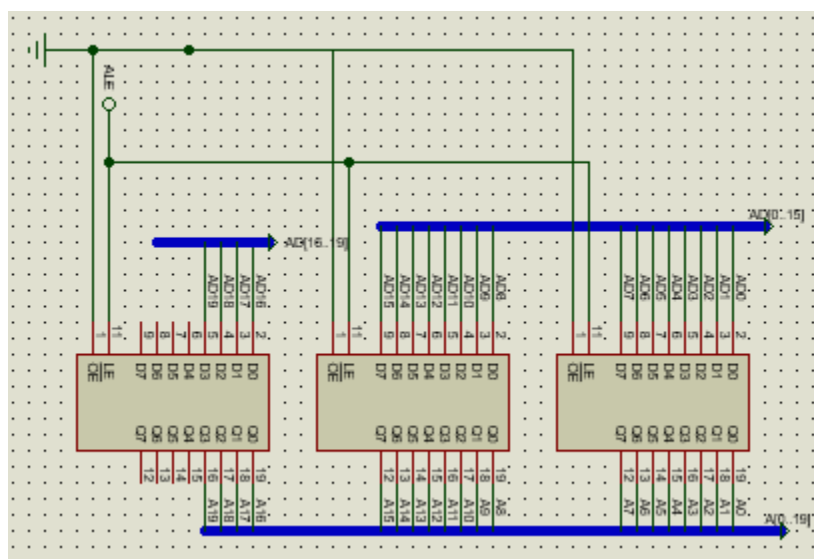
شکل 3.1 : نمایی از پردازنده 8086 به همراه اتصالات مورد نیاز

3.2. جداسازی آدرس و داده

همان طور که اشاره کردیم، پایه های داده و آدرس در پردازنده 8086 مشترک هستند. برای جدا سازی آدرس ها از داده، از لچ استفاده می کنیم. در پردازنده 8086، 20 خط آدرس وجود دارد. 16 خط کم ارزش این آدرس ها مشترک با داده می باشند و در یک باس، و 4 بیت پر ارزش آن مشترک با وضعیت و در یک باس دیگر

¹² READY

قرار دارند. لچ های موجود، 8 بیتی می باشند. برای جداسازی کامل 20 خط آدرس از سه لچ 8 بیتی استفاده کنید. ورودی های لچ های اول را به باس مشترک بین داده و آدرس، یعنی باس AD[0..15] متصل کنید. پس از اتصال ورودی ها، روی هر ورودی برچسب مناسبی قرار دهید. این برچسب نشان می دهد که این سیم به کدام بیت از باس متصل شده است. 4 بیت کم ارزش از لچ سوم را به باس مشترک بین وضعیت و آدرس یا همان باس A[16..19] اتصال دهید و برچسب مناسب را روی هر اتصال قرار دهید. سیگنال کنترلی فعال ساز لچ ها باید مشترک باشد. این سیگنال باید زمانی فعال شود که پردازنده قصد قرار دادن آدرس را روی باس دارد. خود پردازنده پایه و سیگنالی تحت عنوان ALE دارد. این پایه را که از پیش برچسب گذاری کرده بودیم، اکنون با استفاده از یک برچسب، به طور مشترک به هر سه لچ اتصال می دهیم. سیگنال کنترلی دیگر لچ ها که برای روشن یا خاموش بودن لچ ها به کار می رود، سیگنال OE است. این سیگنال در سطح پایین فعال می شود. از آن رو که همیشه نیاز داریم که لچ های ما روشن باشند، آن را به صفر منطقی یا زمین اتصال دهید. اکنون نوبت به اتصال خروجی لچ ها می رسد. برای سادگی در دسترسی و جلوگیری از شلوغی، یک باس برای خروجی طراحی کنید. این باس مقدار آدرس را روی خود دارد، پس برچسب مناسب با آن A[0..19] است. خروجی سه لچ را به این باس اتصال می دهیم و روی هر اتصال برچسب مناسب با آن اتصال قرار می دهیم. اکنون یک آدرس داریم که هر جا نیازمند استفاده و رمزگشایی آدرس باشد، می توان از آن استفاده کرد.

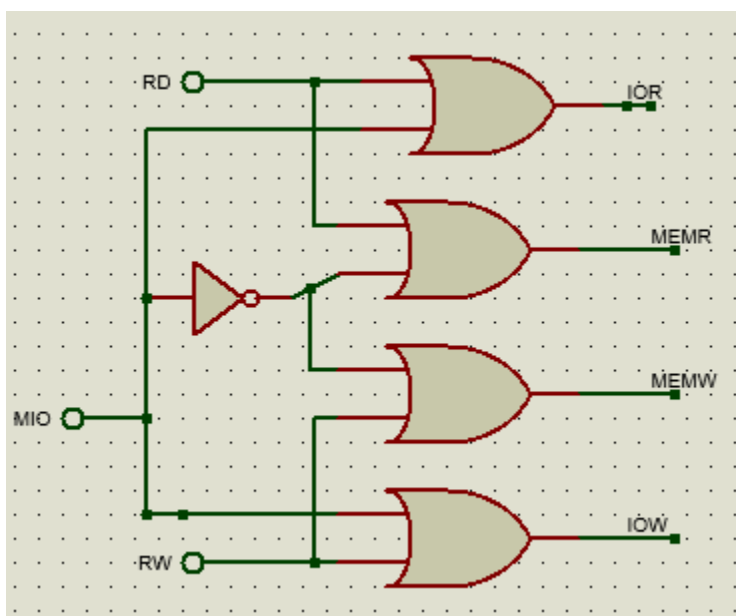


شکل 3.2 : شیوه اتصال لچ ها به باس ها و جداسازی آدرس

3.3. ایجاد سیگنال های فعال سازی و رمزگشایی آدرس پورت ها

برای خواندن از یک دستگاه و یا نوشتن در یک دستگاه، نیازمند طراحی پورت متناسب با آن در گاه هستیم. هر پورت یک آدرس دارد. همچنین هر پورت نیازمند یک سری سیگنال ها مانند $IOWR$, $IORD$ می باشد. به کمک باس آدرس و سیگنال های کنترلی که پردازنده ایجاد می کند می توانیم سیگنال های مورد نیاز مان را با استفاده از گیت های منطقی ایجاد کنیم.

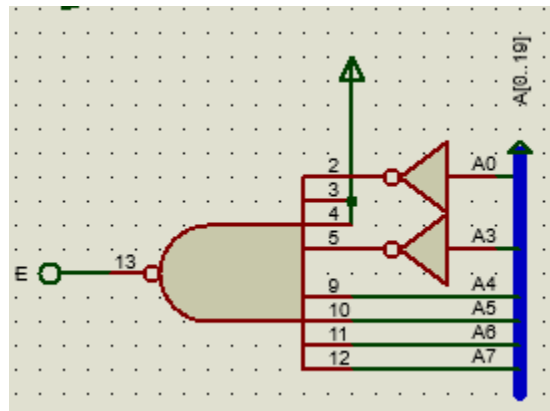
سیگنال های IOR , IOW , $MEMR$, $MEMW$ چهار سیگنال اساسی و پایه برای ایجاد سایر سیگنال ها می باشند. این سیگنال ها به ترتیب سیگنال های نوشتن در حافظه، خواندن از حافظه، نوشتن در دستگاه جانبی و خواندن در دستگاه جانبی می باشند. با انجام عملیات های منطقی روی سیگنال های MIO , RD , WR که از پردازنده خارج می شوند این سیگنال ها را ایجاد می کنیم. ساختار ایجاد این سیگنال ها را در تصویر زیر آورده ایم.



شکل 3.3: پیاده سازی سیگنال به کمک گیت منطقی OR و ورودی های RD, WR, MIO

سیگنال دیگری که مورد استفاده قرار گرفته است و باید آن را ایجاد کنیم، سیگنال مربوط به CS یا همان فعال ساز مربوط به 8255 می باشد. این سیگنال را E نام گذاری کرده ایم. زمانی که آدرس های $11110xx0$ باید فعال ساز مربوط به 8255 فعال شود. برای رمزگشایی این سیگنال از یک گیت NAND با 8 ورودی استفاده می کنیم. ورودی های این گیت 6 بیت از آدرس مربوطه می باشد. برای دو ورودی آزاد این تراشه مقدار یک

منطقی را در نظر می گیریم تا در نتیجه اثری نداشته باشد. تصویر زیر شیوه اتصال برای رمزگشایی و ایجاد این سیگنال را نشان می دهد.



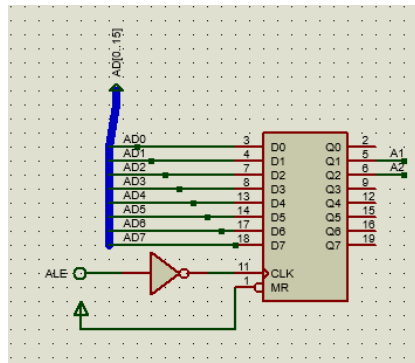
شکل 3.4 : شیوه اتصال و ایجاد سیگنال CS مربوط به 8255

از دیگر سیگنال های مورد استفاده، سیگنال های $A0, A1$ برای ورودی های مربوطه در 8255 می باشد. این ورودی ها نشان می دهند که کدام پورت از این تراشه باید انتخاب شود. جدول درستی این تراشه به صورت زیر می باشد:

A1	A0	Result
0	0	Select port A
0	1	Select port B
1	0	Select port C
0	1	Select control register

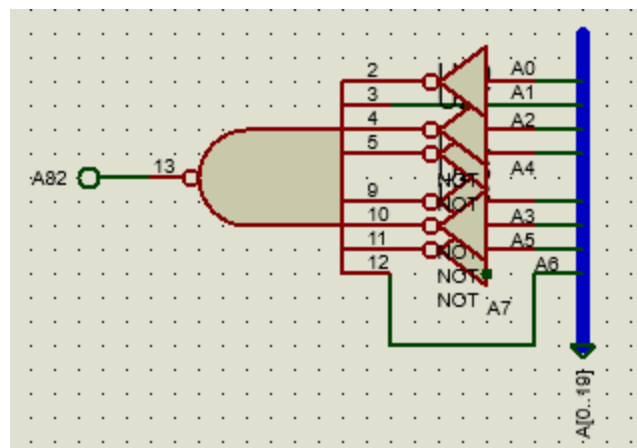
جدول 3.1 : جدول درستی 8255

مقدار دو سیگنال را به کمک یک فلیپ فلاپ و خطوط آدرس $A1, A2$ ایجاد می کنیم. کلاک مربوط به تراشه را به وسیله ی یک گیت not به سیگنال ALE از پردازنده متصل می کنیم. در آخر مدار مربوطه به صورت زیر می شود:

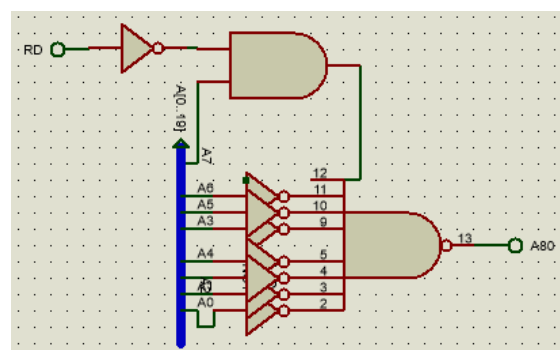


شکل 3.5: شیوه ایجاد سیگنال های A0, A1

برای خواندن و نوشتن از گذرگاه های طراحی شده از دو آدرس 80 و 82 هگز در نظر گرفته شده است. بنابراین نیاز داریم که آدرس های ذکر شده را رمز گشایی کنیم. بنابراین با استفاده از گیت های not و nand آدرس ها را به صورت شرح داده شده در شکل های 3.6 و 3.7 رمز گشایی می کنیم.



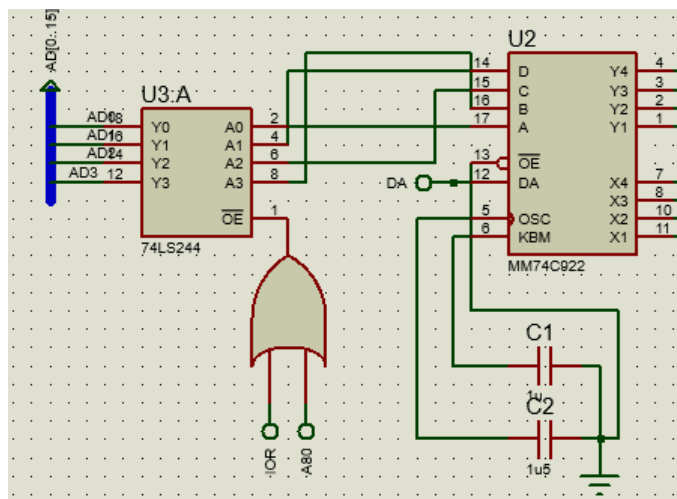
شکل 3.6: رمز گشایی آدرس 82 هگز



شکل 3.7: رمز گشایی آدرس 80 هگز

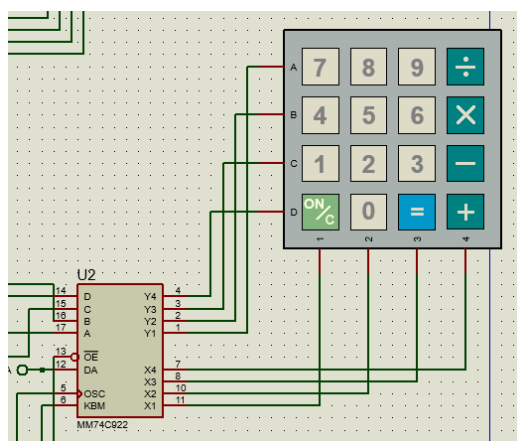
3.4. صفحه کلید و مدارهای خواندن از صفحه کلید

برای خواندن از صفحه کلید، پورت 80 هگز در نظر گرفته شده است. آدرس 80 هگز را که رمز گشایی کرده بودیم به همراه سیگنال IOR را به یک گیت OR وصل می کنیم و خروجی آن را به OE یک بافر وصل می کنیم. خروجی های بافر به دیتاباس اتصال دارند. این بافر ورودی های خود را از رمزگشای صفحه کلید دریافت می کند. تصویر 3.8 شیوه اتصال رمزگشای صفحه کلید به بافر و اتصال بافر به باس را نمایش می دهد.



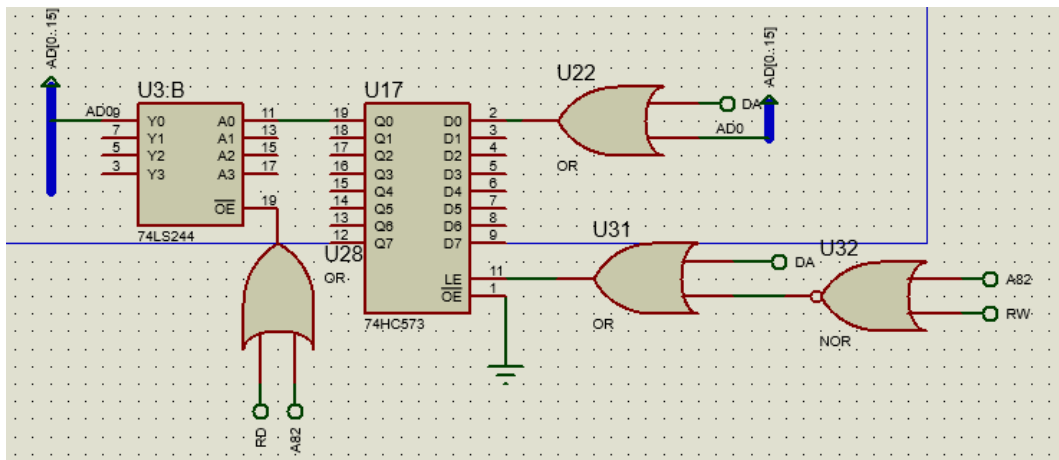
شکل 3.8: اتصال رمزگشای صفحه کلید به بافر

اکنون باید اتصالات لازم بین صفحه کلید و رمزگشا را نیز انجام دهیم. شکل 3.9 شیوه این اتصال را نمایش می دهد.



شکل 3.9: اتصال صفحه کلید به رمزگشای آن

برای خواندن از صفحه کلید، ابتدا باید بررسی کنیم که آیا کلیدی فشار داده شده و مقدار جدیدی روی رمزگشا قرار گرفته است یا خیر. برای این منظور یک مدار طراحی می‌کنیم که زمانی که کلیدی فشار داده شد و سیگنال DA از رمزگشا فعال شد، این مدار مقدار 1 را روی لچ و ورودی یک بافر قرار دهد. همچنین زمانی که مقداری را خواندیم، مقدار این لچ را 0 کنیم. برای این منظور در آدرس پورت 82 هگز، خروجی‌های یک لچ را به ورودی‌های یک بافر و خروجی‌های بافر را به دیتاباس اتصال می‌دهیم. ورودی این بافر توسط یک گیت OR که ورودی‌های آن مقدار D0 از دیتاباس و DA از رمزگشای صفحه کلید است مقدار می‌گیرد. شکل 3.10 اتصالات این مدار را نشان می‌دهد.

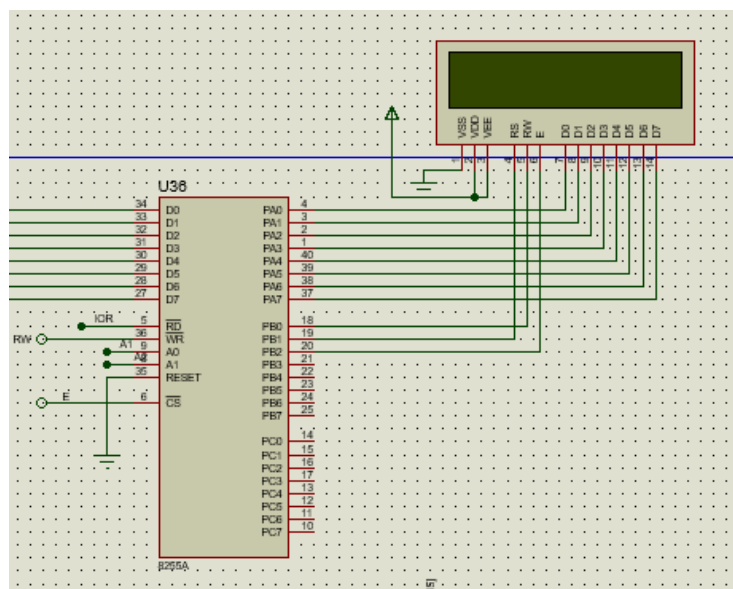


شکل 3.10: مدار طراحی شده برای بررسی اینکه کلیدی در صفحه کلید فشار داده شده است یا خیر و اتصال این مدار به دیتاباس

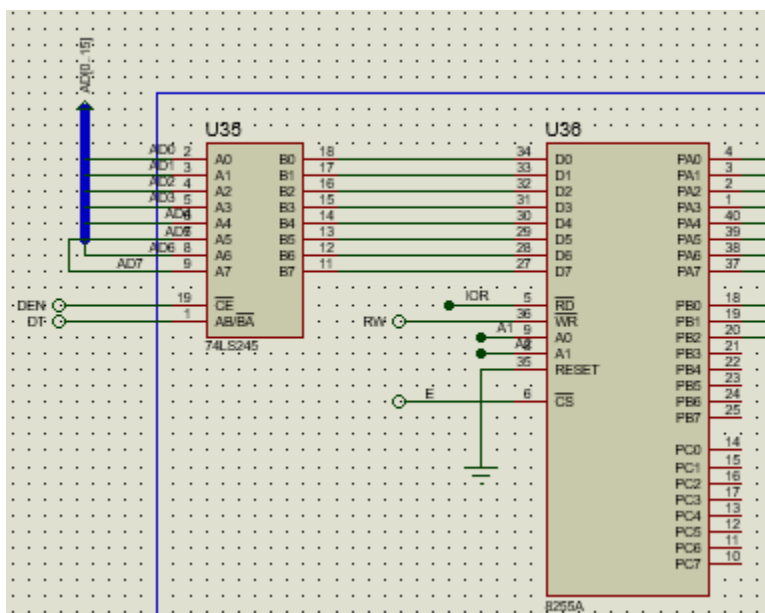
3.5. صفحه نمایش و مدارهای نوشتن در صفحه نمایش

در آخرین گام از سخت افزار و مدارمان، باید اطلاعات مورد نیاز را به صفحه نمایش ارسال کنیم. برای ارسال داده‌ها به صفحه نمایش، باید اتصالات درستی بین صفحه نمایش و تراشه 8255 برقرار نماییم. همچنین تراشه 8255 یک تراشه ورودی/خروجی است. بنابراین باید با استفاده از تراشه 74LS245 ورودی‌های این تراشه را کنترل کنیم و از بروز خطا روی سیم‌های دیتاباس جلوگیری کنیم. همچنین برای نمایش اطلاعات در صفحه نمایش، پورت خروجی با آدرس 11110xx1 در نظر گرفته شده است. این آدرس و مقدار را در هنگام ایجاد سیگنال CS یا همان E رمزگشایی کرده ایم. پایه‌های داده صفحه نمایش را به پورت A از تراشه 8255 و پایه‌های فعال ساز، خواندن و نوشتن و کنترل آن را به پایه‌های پورت B از تراشه 8255 اتصال می‌دهیم. همچنین سیگنال‌های انتخاب A1, A2 را که از پیش آن‌ها را ایجاد کرده بودیم، به پایه‌های 8255 و صب می‌کنیم. هر زمان که این آدرس فعال باشد و همچنین پردازنده قصد ارسال داده را داشته باشد، تراشه 8255 فعال شده و

عملیات مورد انتخاب را انجام می دهد. در شکل 3.11 و 3.12 شیوه اتصال صفحه نمایش به 8255 و 8255 به تراشه 74LS245 و همچنین سایر اتصالات لازم آمده است.



شکل 3.11: اتصالات بین صفحه نمایش و 8255



شکل 3.12: اتصالات بین تراشه 8255 و تراشه 74LS245 و دیتاباس

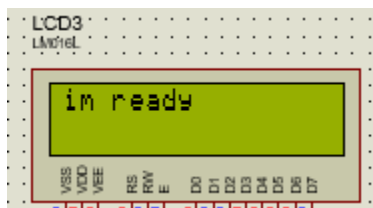
4. عملکرد و راهنمای کاربر

در این فصل عملکرد ماشین حساب ساخته شده را با بررسی می کنیم. در ابتدا با اجرای پروژه صفحه خوش آمدگویی فعال و متن انتخاب شده روی صفحه نمایش داده می شود.



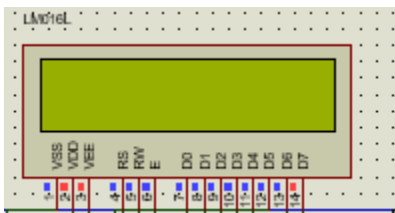
شکل 4.1: صفحه خوش آمدگویی

در گام بعد، صفحه اعلام وضعیت نمایش داده می شود.



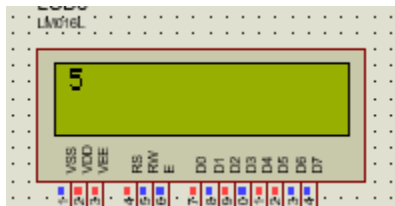
شکل 4.2: صفحه اعلام وضعیت

در گام بعدی صفحه نمایش خالی شده و آماده ی دریافت اولین رقم از سمت کاربر می ماند.



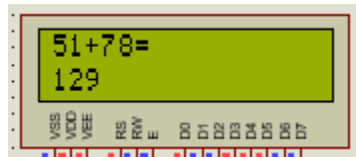
شکل 4.3: صفحه آماده ی دریافت رقم بعدی

پس از وارد کردن اولین رقم، رقم وارد شده در صفحه نمایش ظاهر می شود.



شکل 4.4: صفحه نمایش پس از ورود اولین رقم

به همین ترتیب رقم های بعدی وارد شده و عملیات مدنظر انتخاب می شود. پس از انتخاب عملیات و ورود عدد دوم، ماشین حساب منتظر می ماند تا کلید مساوی فشار داده شود. پس از فشردن کلید مساوی، نتیجه در کنار اعداد و عملیات فشار داده شده نمایش داده می شود.



شکل 4.5: وضعیت صفحه نمایش بعد از انجام عملیات

برای استفاده ی دوباره از ماشین حساب، باید کلید CLEAR را فشار دهیم تا صفحه نمایش و حافظه ماشین حساب خالی شود. پس از فشردن این دکمه، در صفحه نمایش اعلام می شود که ماشین حساب در حال خالی کردن حافظه خود می باشد.



شکل 4.6: صفحه اعلام وضعیت در هنگام پاک کردن حافظه

دسترسی به پروژه

فایل های پروژه شامل, فایل پرتئوس, کدهای اسمبلی, مستندات فایل و سایر پیوست ها در گیت هاب قرار دارد.

برای دسترسی به آن ها می توانید روی این [لینک](https://github.com/hamidbz/COCO) کلیک کنید. یا در گیت هاب آدرس <https://github.com/hamidbz/COCO> را جست و جو کنید. همچنین می توانید با اسکن کردن کد زیر, وارد پوشه ی گیت هاب این پروژه شوید.



شکل 5.1 : QR کد دسترسی به پروژه

ارتباط با ما

حمیدرضا بازیار



@hamidbz



h.r.bz1381@gmail.com

کاوه افشار



@kavehafshar81



kaveh.afshar.1381@gmail.com