

آزمایشگاه ریزپردازنده

تمرین چهار

سینا عربی – سالار جهانشیری – امیرعلی وکیلی

# **سوالات تحلیلی:**

* **کاربرد تراشه 8255 ، سیگنالهای کنترلی و نحوه کار با آن را شرح دهید.**

**تراشه 8255**

این تراشه یک مدار مجتمع چند منظوره است که در سیستم های کامپیوتری اولیه برای برای ارائه پورت های ورودی/خروجی قابل برنامه ریزی استفاده میشد.

**کاربرد های تراشه 8255**

* جمع آوری داده های حسگر
* کنترل موتور ها و LED ها
* رابط با دستگاه های جانبی

**سیگنال های کنترلی تراشه 8255**

* Reset: این سیگنال برای ریست کردن 8255 به حالت پیش فرض استفاده میشود.
* A0,A1:این سیگنال ها برای انتخاب بین سه پورت استفاده میشود.
* :WR سیگنال برای نوشتن
* RD : سیگنال برای خواندن
* CS:سیگنال برای انتخاب ۸۲۵۵

**نحوه کار:**

ابتدا میبایست سیگنال های کنترلی مناسب را برای انتخاب پورت I/O را تنظیم کنیم.

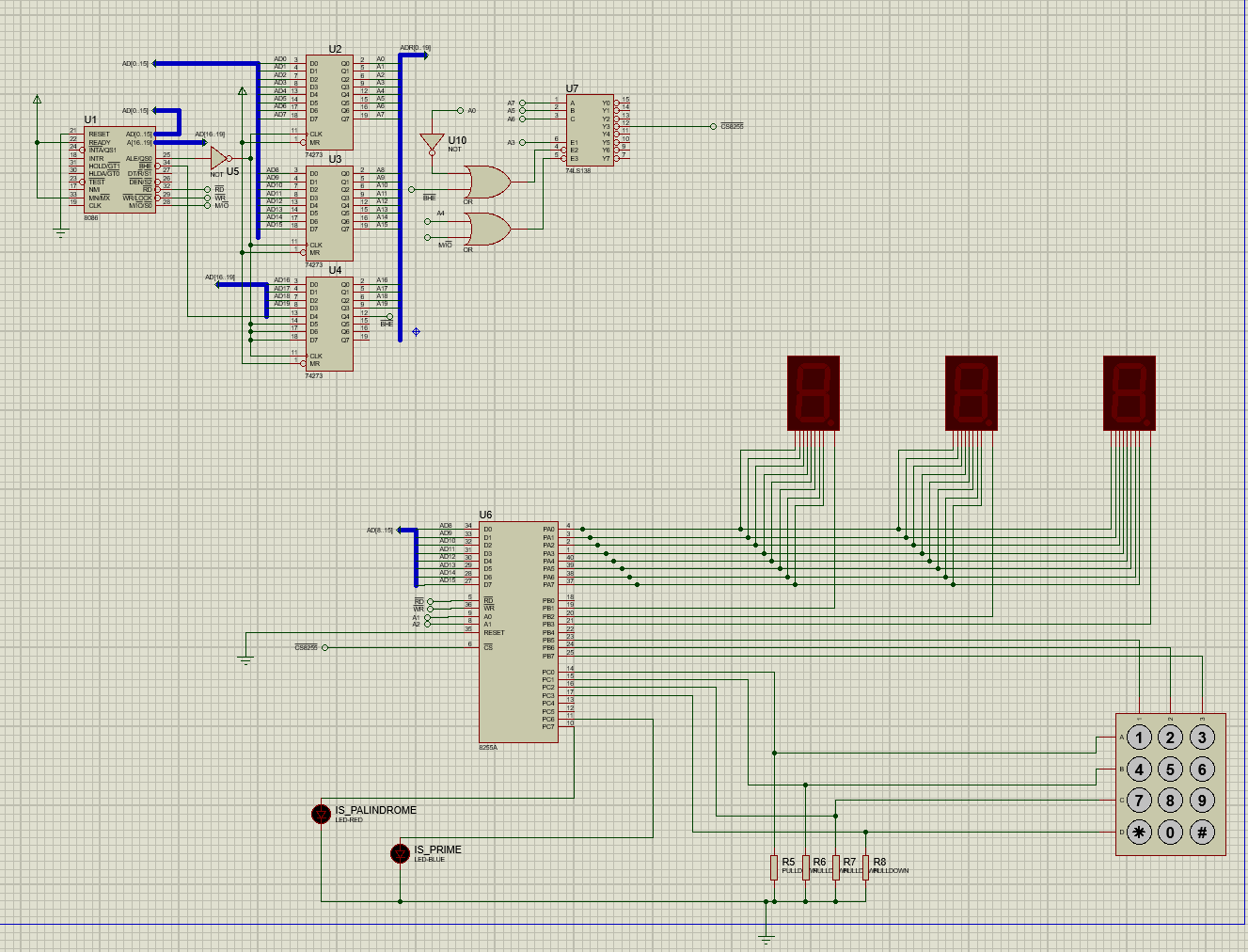
سپس CS را صفر میکنیم تا تراشه انتخاب شود.

سپس میتوان داده ها را با توجه به پورت انتخاب شده با توجه به سیکنال های WR,RD خواند یا نوشت.

# **نحوه کار با صفحه کلید ماتریسی و خواندن کلیدهای آن را شرح دهید.**

در این صفحه کلید ماتریسی سه ستون و چهار سطر داریم که اینها به پورت های IO وصل میشوند و وقتی یک کلید را میزنیم، فاصله خازن کم میشود و جریان برقرار میشود و یک اختلاف ولتاژی بدست میاید که یک طرف به زمین وصل است و طرف دیگر به باتری و یا power وصل می‌باشد،هرطلاقی در این مدار یک نقطه و یا یک کلید در صفحه را نشان میدهد که با فشار کلید و برقراری جریان و اختلاف ولتاژ ستون و سطر مورد نظر آن کلید فعال میشود و اگر این خروجی ها را به پورت های IO وصل کنیم میتوانیم با حالت بندی کلید ها تشخیص دهیم که چه کلیدی فشار داده شده است.

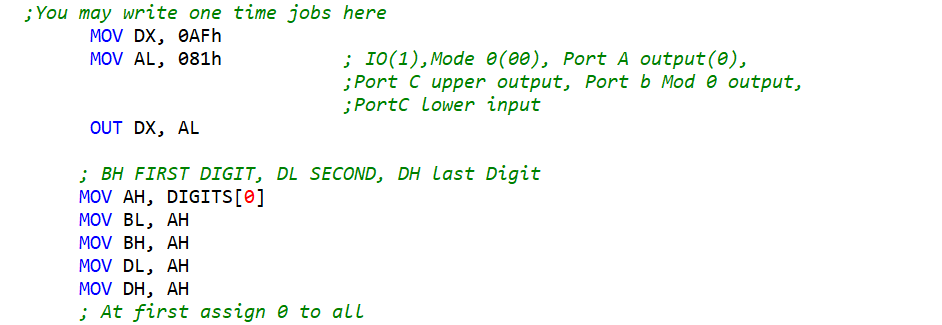
# **دستور کار:**



شماتیک کلی مدار(در هر بخش به صورت جزئی توضیح داده شده)

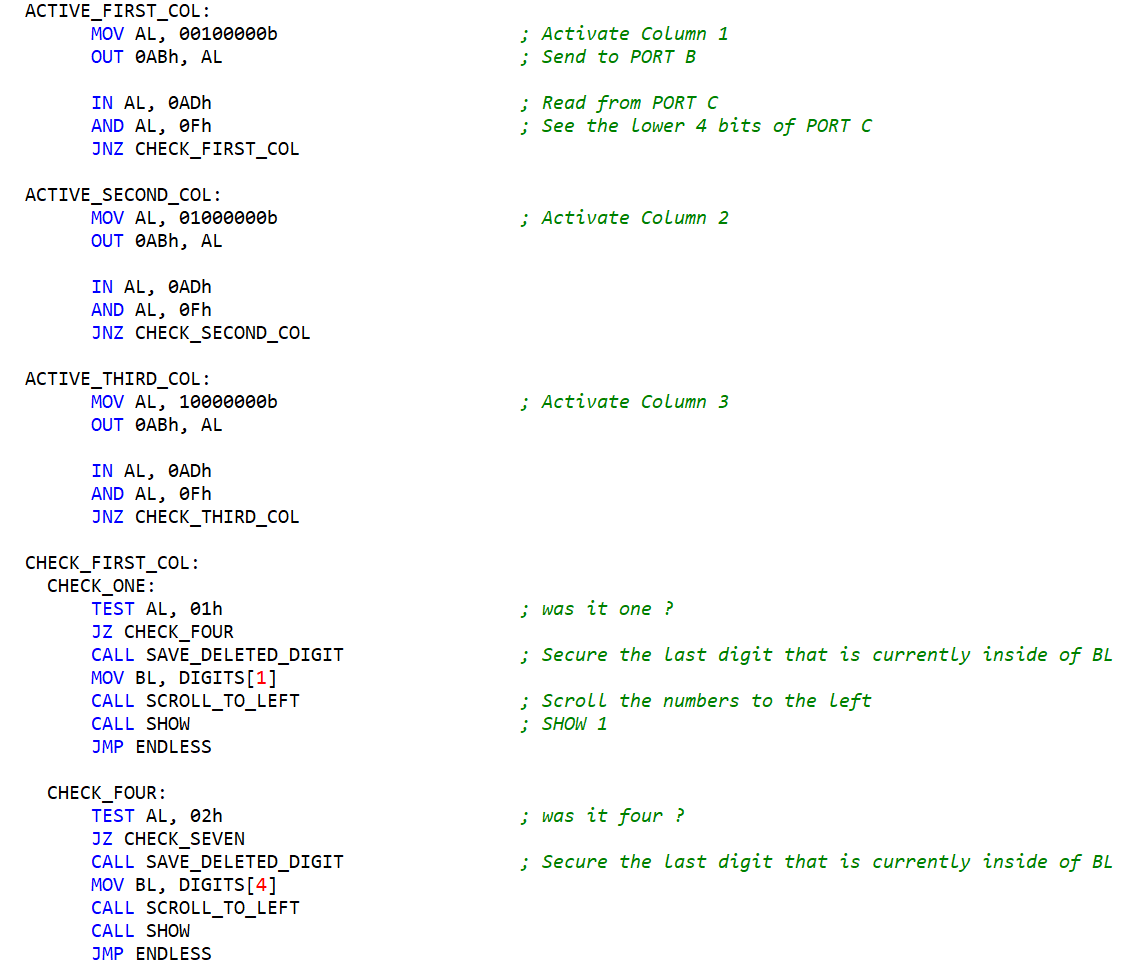
برای نمایش اعداد ورودی توسط کاربر از یک قطعه 7-segment با دیکودینگ مود آنود انتخاب کردیم، برای نمایش اعداد روی این نمایشگر عدد باینری معادل هر عدد که هر تکه از 7-seg را روشن می‌کند، بدست آورده و در آرایه Digits ذخیره می‌کنیم.

آدرس مودینگ قطعه 8255a برای نوشتن در پورت A و B ، 4 بیت کم‌ارزش تر C برای ورودی و 4 بیت پر ارزش تر برای نوشتن C استفاده می‌کنیم.

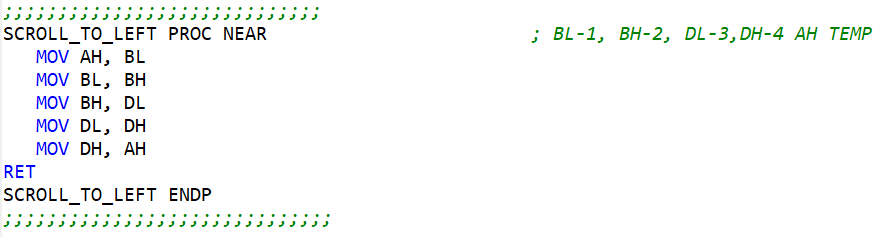


ابتدا تمامی رجیستر‌ها را با عدد دیکود شده 0 برای 7-seg مقدار دهی اولیه می‌کنیم.

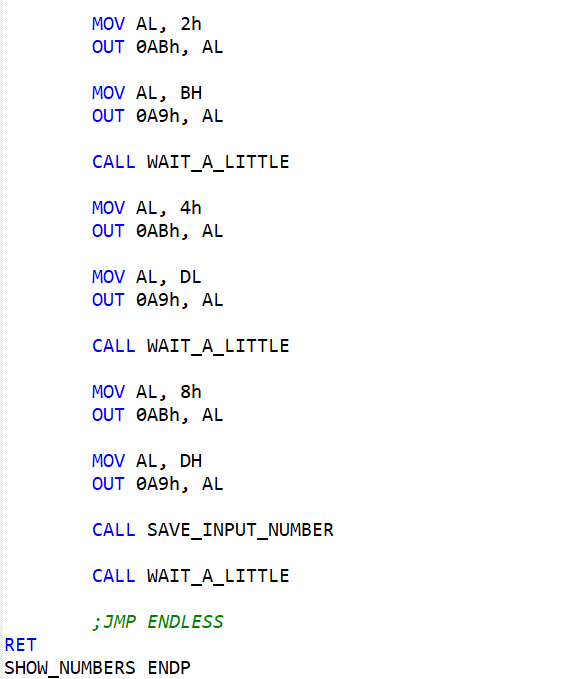
در ادامه در یک لوپ بی‌نهایت عملیات ورودی گرفتن از کاربر را تعریف می‌کنیم، به این صورت که با هر بار فشردن دکمه از صفحه کلید ماتریسی از طریق پورت‌ها بررسی می‌کنیم که در کدام سطر و ستون دکمه ای فشرده شده و سپس با توجه به آن عدد، حالت دیکود شده آن برای 7-seg را در رجیستر BL نگه می‌داریم.



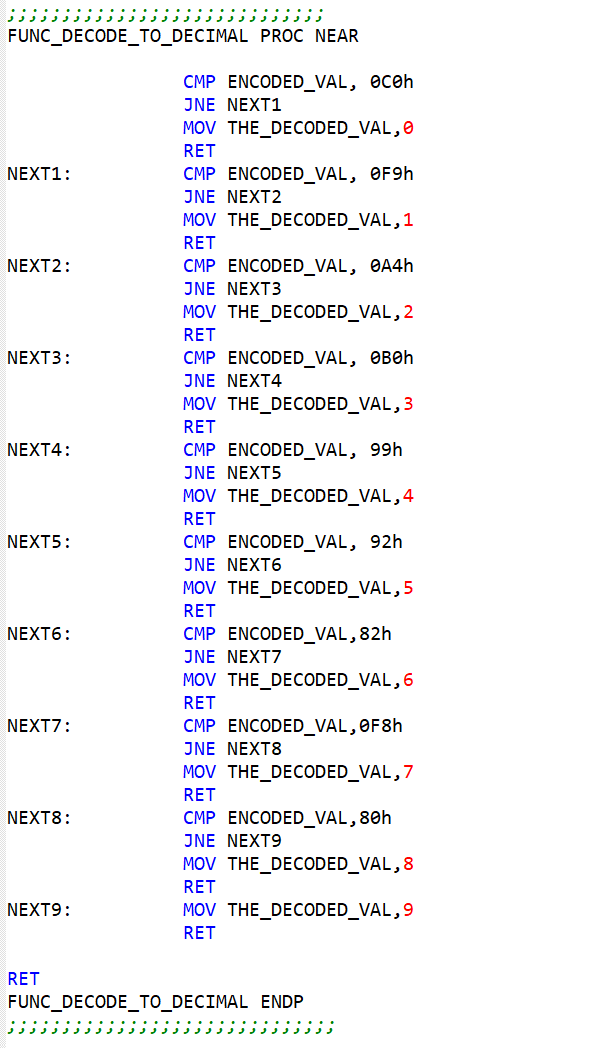
در ادامه با فشرده شدن عدد بعدی هر عدد را به رجیستر بعدی منتقل می‌کنیم(طبق قرار رقم اول در BH، رقم دوم در DL و رقم سوم را در DH نگه ‌می‌داریم)

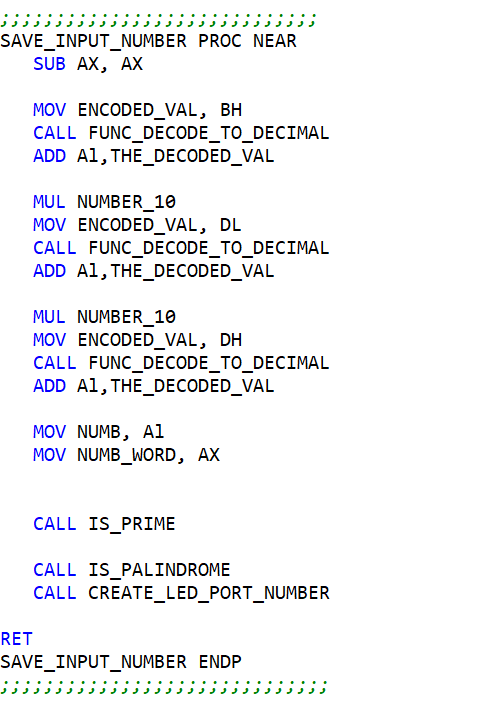


در مرحله بعد برای نمایش اعداد روی 7-seg رجیستر‌های شامل اعداد دیکود شده را در پورت متصل به نمایشگر می‌نویسیم. بین هر نوشتن یک wait پیاده‌سازی شده تا اعداد به درستی از روی پورت در نمایشگر نشان داده شوند.



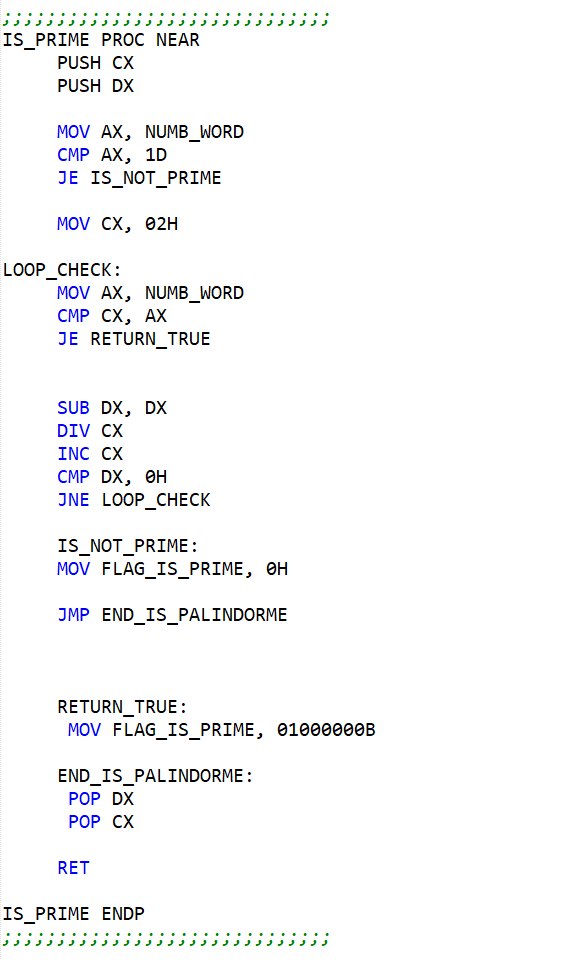
سپس برای بررسی اول یا پالیندرومی بودن عدد به سراغ متد Save\_Input\_Number می‌رویم، برای پیاده سازی این بخش اعدادی که برای 7-seg دیکود شده بود را دوباره به عدد دسیمال انکود کرده و عدد سه رقمی را تولید می‌کنیم.



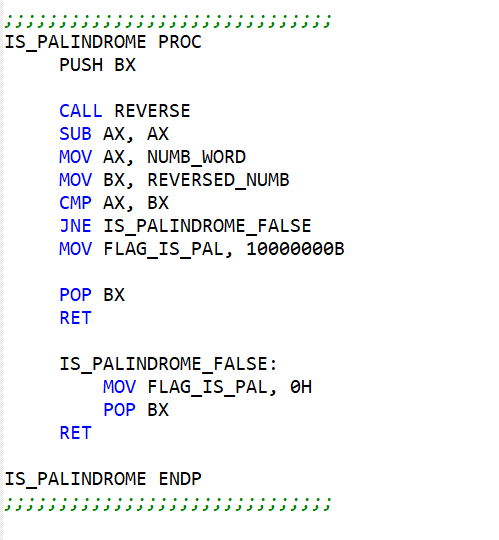


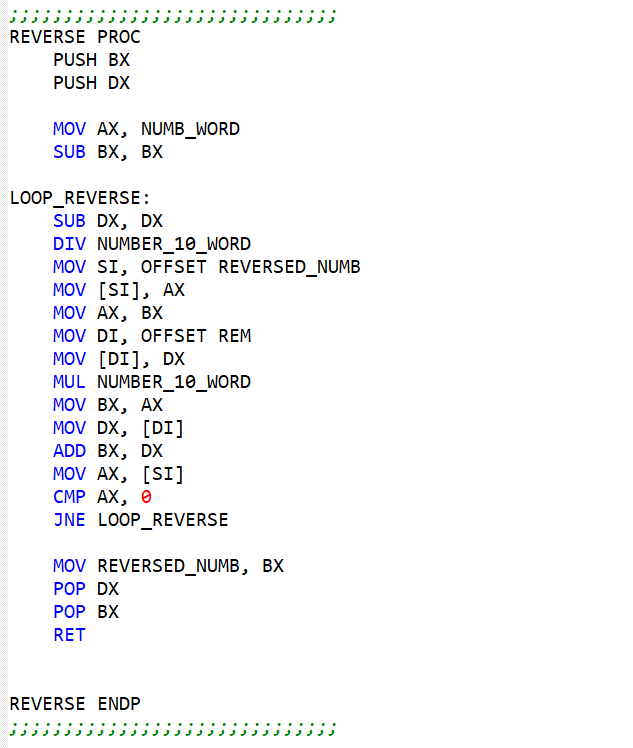
در ادامه متد‌های IS\_PRIME و IS\_PALINDROME برای بررسی اول بودن و پالیندرومی بودن عدد صدا زده می‌شوند و درصورتی که تایید شود نتیجه در فلگ‌های FLAG\_IS\_PRIME و

FLAG\_IS\_PAL بصورت اعداد باینری بیتی که خروجی را روی پورت C به LED های مربوطه وصل شده می‌نویسیم.



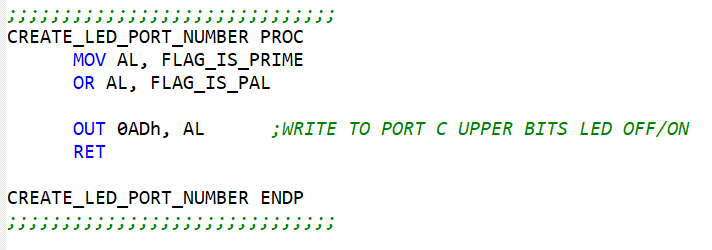
کد بررسی اول بودن عدد N با بررسی بخش‌پذیری بر اعداد کوچکتر از N.



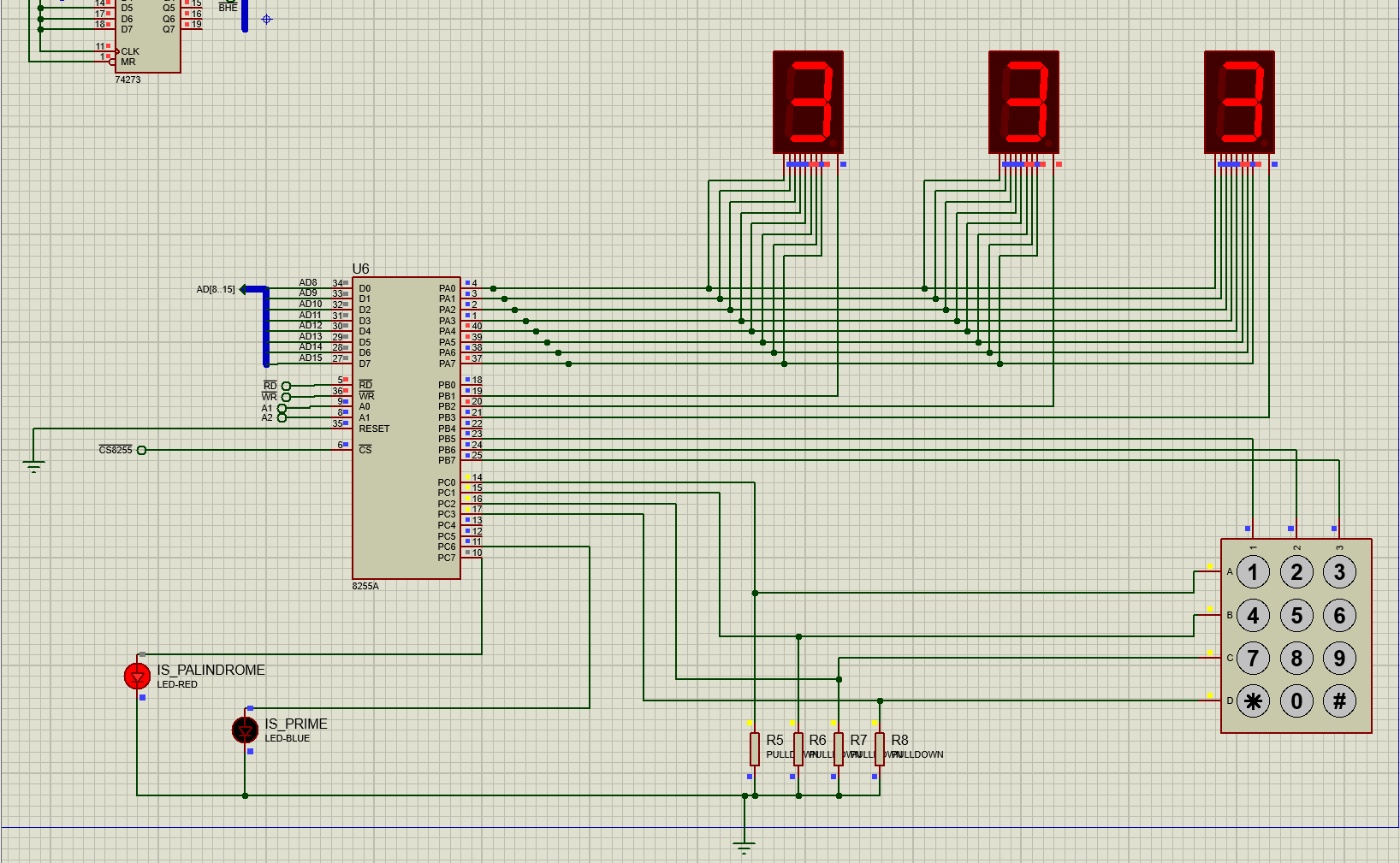


بررسی پالیندرومی بودن عدد abc با بررسی برابری آن با عدد cba.

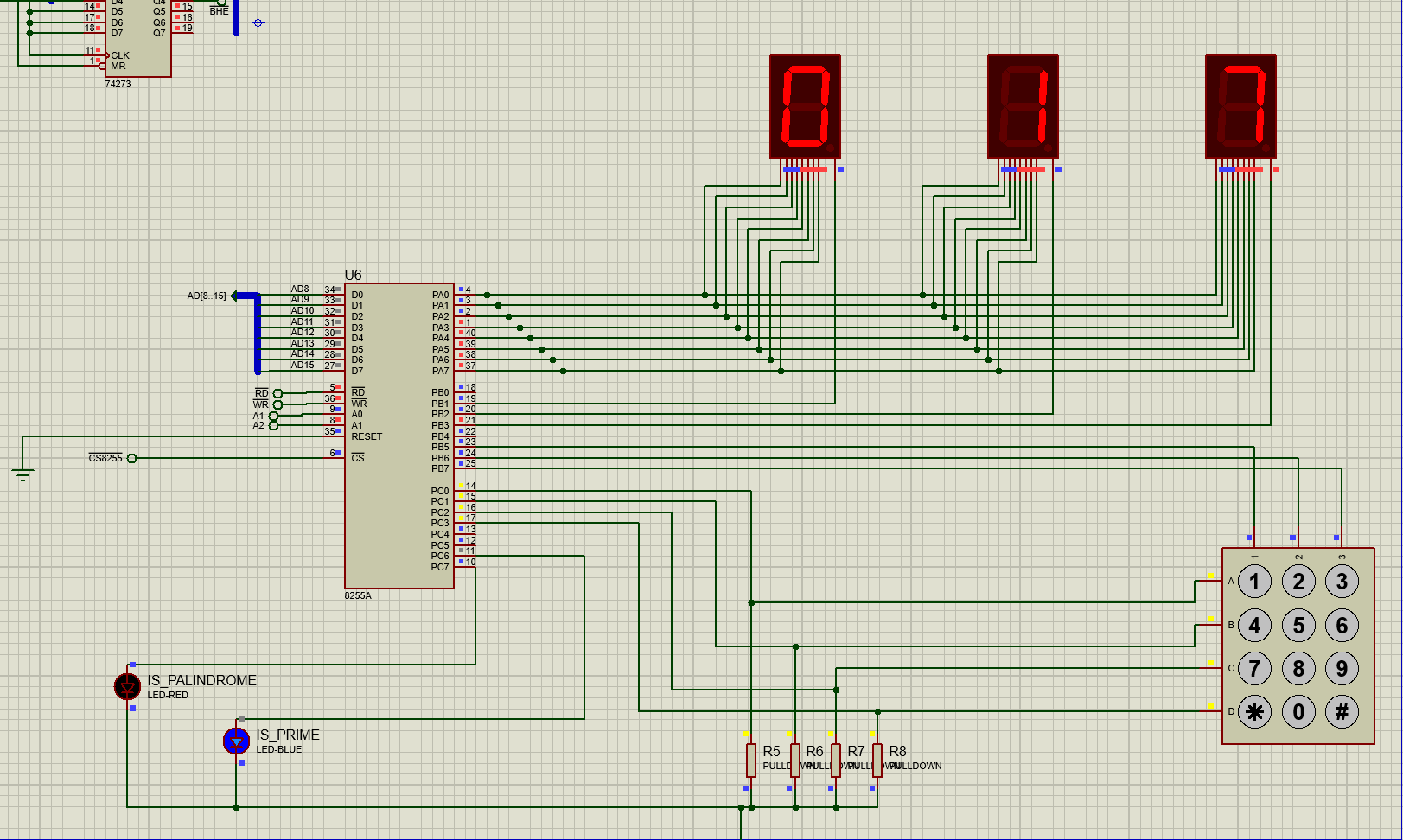
در نهایت با استفاده از فلگ‌های ذخیره شده خروجی روی پورت C را تعیین می‌کنیم تا LED های مربوط به هر بخش روشن یا خاموش شوند.



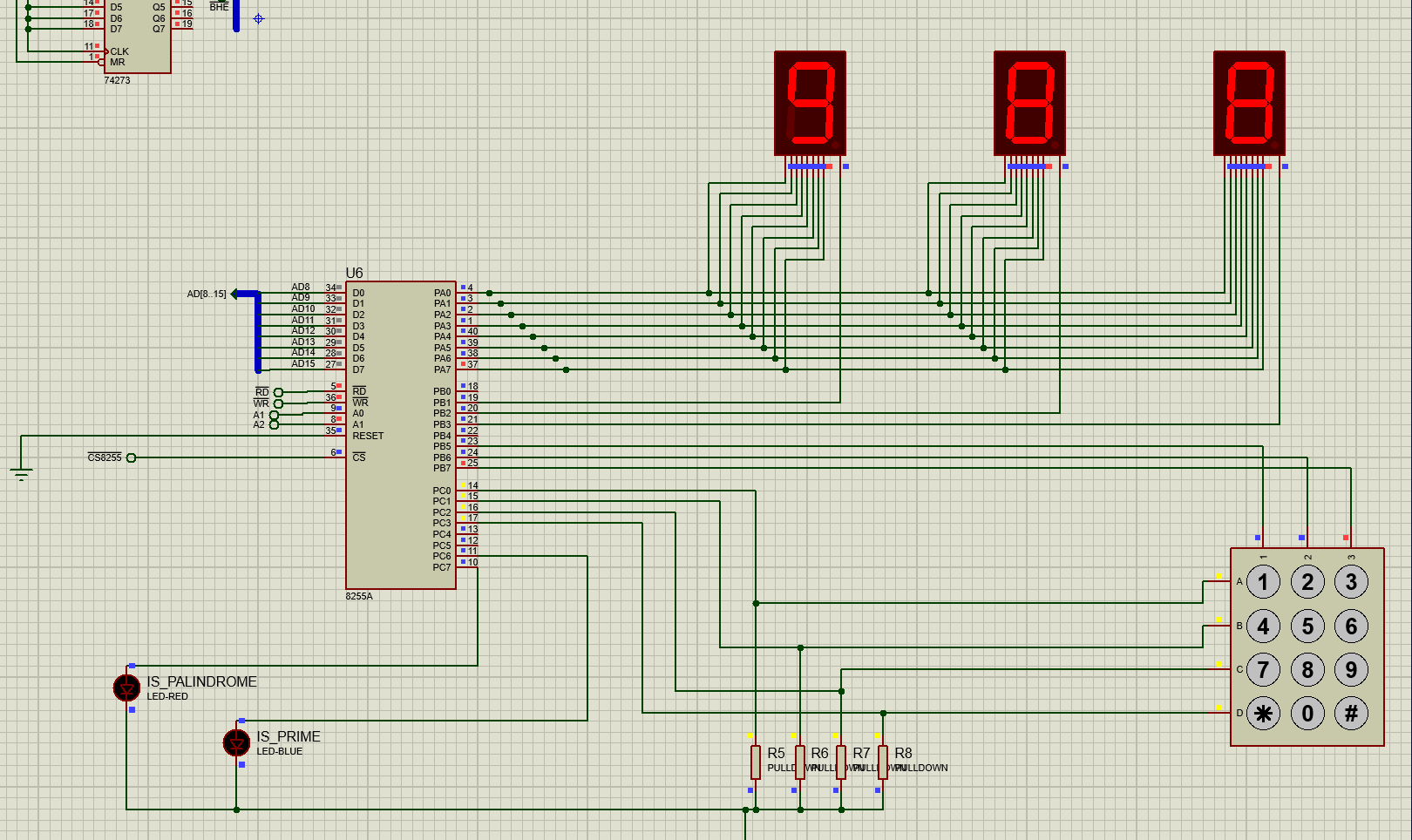
نمونه‌های ورودی و خروجی‌:‌



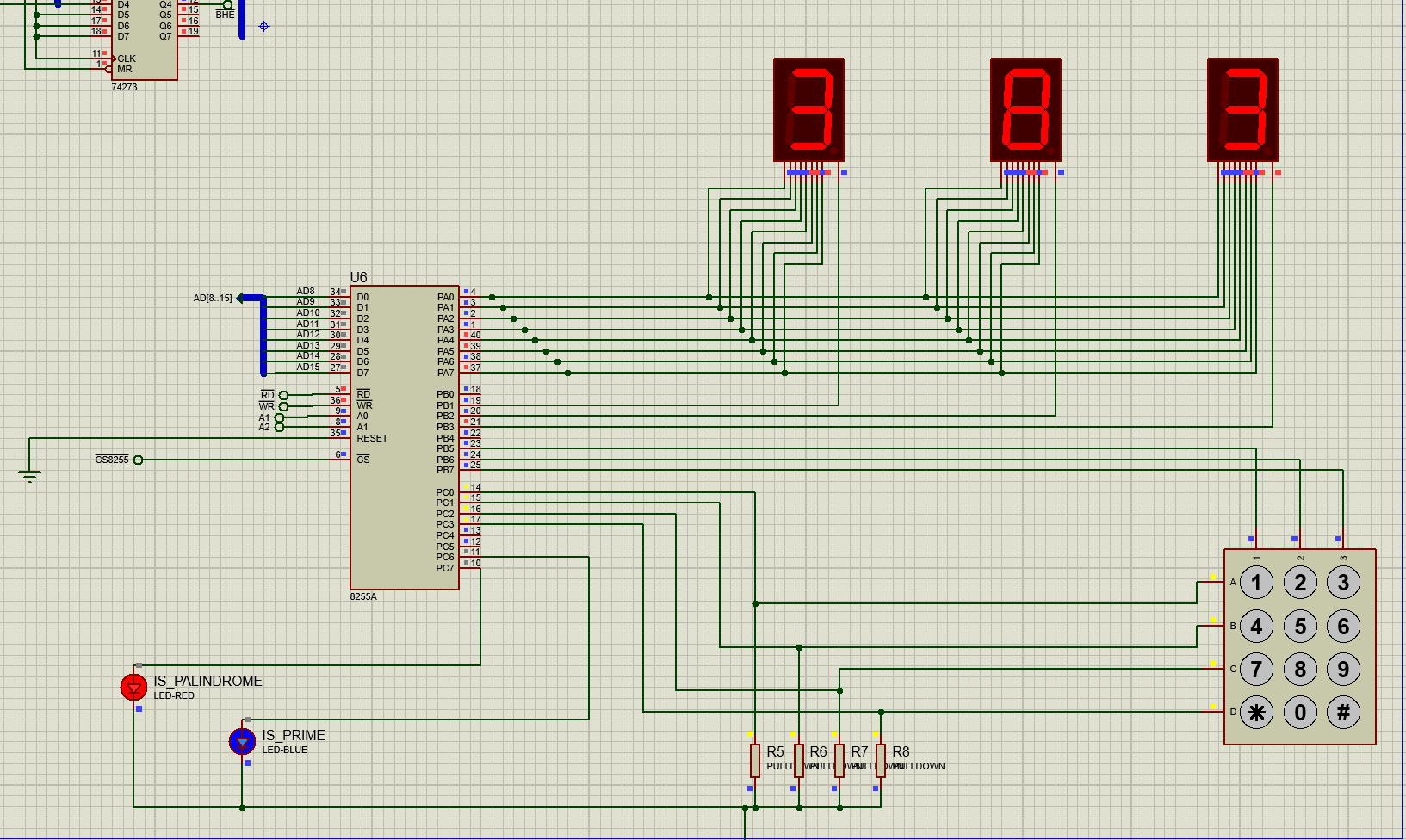
* 333 یک عدد پالیندرومی و غیر اول می‌باشد.



* 17 یک عدد اول و غیر پالیندرومی می‌باشد.



* 988 یک عدد غیر اول و غیر پالیندرومی می‌باشد.



* 383 یک عدد اول و پالیندرومی می‌باشد.