

به نام خدا



دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

آزمایشگاه معماری کامپیوتر

آزمایش چهارم:

مبدل BCD به باینری

اطلاعات تیم	
نام اعضا	شماره دانشجویی
متین داغیانی	۹۸۱۰۶۴۵۶
بردیا محمدی	۹۸۱۷۱۱۰۴
محمدجواد هزاره	۹۸۱۰۱۰۷۴

پاییز ۱۴۰۰

فهرست مطالب

۲	۱	هدف آزمایش
۳	۲	مراحل طراحی و پیاده‌سازی مدار
۳	۱.۲	ورودی و خروجی مدار
۳	۲.۲	ماژول‌های SUB_A و SUB_B
۴	۳.۲	ماژول‌های MUX_A و MUX_B
۵	۴.۲	ماژول REG
۷	۵.۲	ماژول‌های SHIFT_A و SHIFT_B
۸	۶.۲	ماژول Counter
۱۰	۳	تست مدار

۱ هدف آزمایش

در این آزمایش هدف پیاده‌سازی مداری برای تبدیل اعداد دهدهی به باینری بود. عدد ورودی به صورت BCD است و خروجی تبدیل شده‌ی این عدد در مبنای ۲ خواهد بود.

به منظور تبدیل خواسته شده، از الگوریتم ارائه شده در گزارش کار استفاده شد که مراحل انجام آن به صورت مختصر در زیر آمده است:

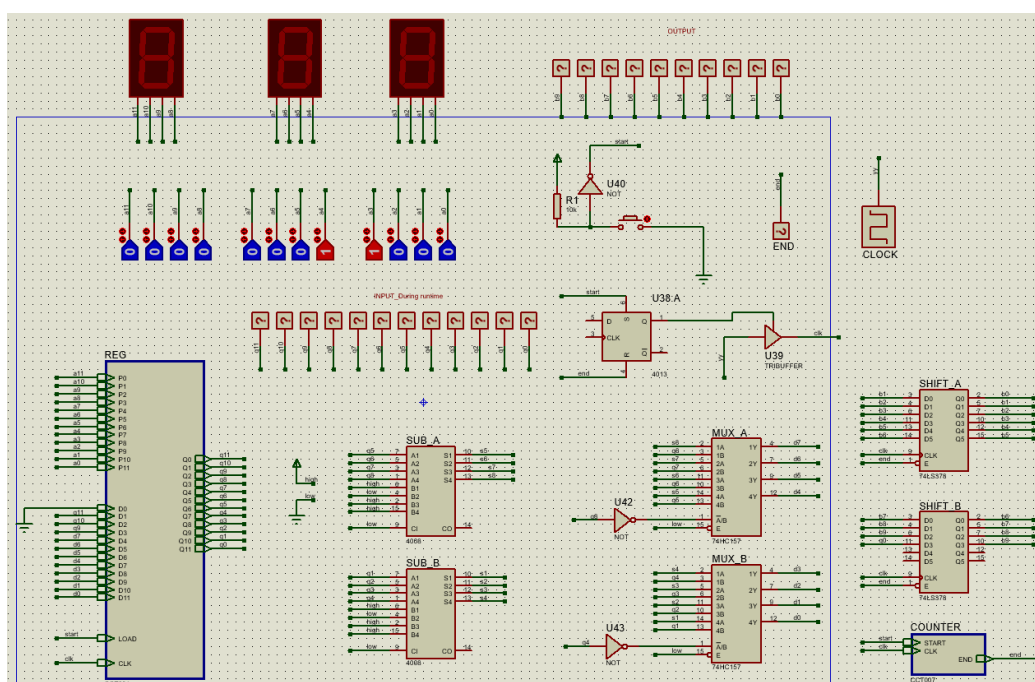
۱. عدد BCD یک واحد به سمت راست شیفت داده شود.

۲. اگر بیت چهارم هر یک از ارقام ۱ است، آن رقم منهای ۳ شود.

۳. به مرحله ۱ رفته و مراحل تا جایی ادامه داده شود که تمام بیت‌های عدد BCD برابر صفر شوند.

در پایان این الگوریتم، بیت‌هایی که از سمت راست عدد خارج شده‌اند، نمایش باینری عدد مورد نظر را تشکیل خواهند داد.

همانطور که از مراحل الگوریتم پیداست، برای طراحی مدار به یک تفریق‌کننده و رجیسترهایی برای نگهداری مقدار شیفت‌داده شده‌ی عدد ورودی نیاز است. به جای چک کردن صفر شدن تمام بیت‌ها، می‌توان از یک شمارنده استفاده کرد. در این آزمایش چون تعداد رقم‌های عدد دهدهی ورودی مشخص و برابر ۳ است، حداکثر مراحل الگوریتم برابر ۱۰ خواهد بود و در نتیجه می‌توان با شمارش تا ۱۰، به اتمام کار مدار پی برد. در ادامه به طراحی مدار داخلی ماژول‌های ذکر شده خواهیم پرداخت. نمای کلی مدار را نیز می‌توان در شکل ۱ مشاهده کرد.



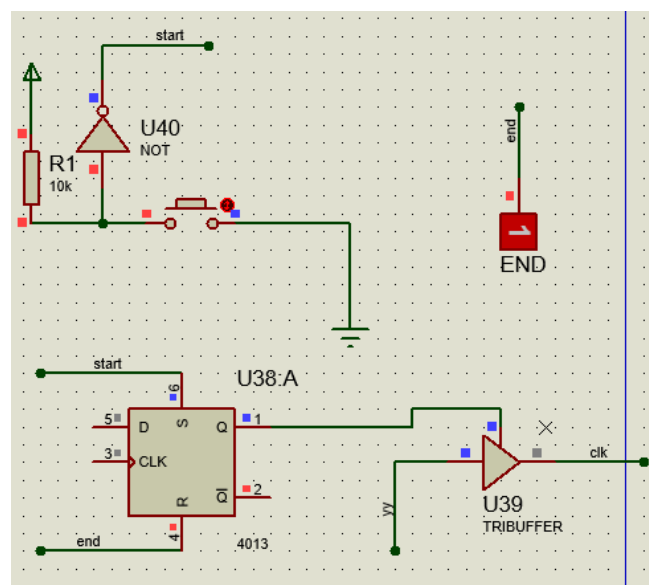
شکل ۱: نمای کلی مدار

۲ مراحل طراحی و پیاده‌سازی مدار

۱.۲ ورودی و خروجی مدار

مطابق شکل ۱ که در بخش قبل آمده است، ورودی مدار که عدد ده‌دهی مورد نظر است با بیت‌های a_{11} تا a_0 مشخص شده است و خروجی مدار که عدد دودویی ۱۰ رقمی است با بیت‌های b_0 تا b_9 . سیگنال‌های q_0 تا q_{11} نیز همان عدد ده‌دهی ورودی را در طول مراحل الگوریتم نشان می‌دهند که به سمت راست شیفت داده می‌شود.

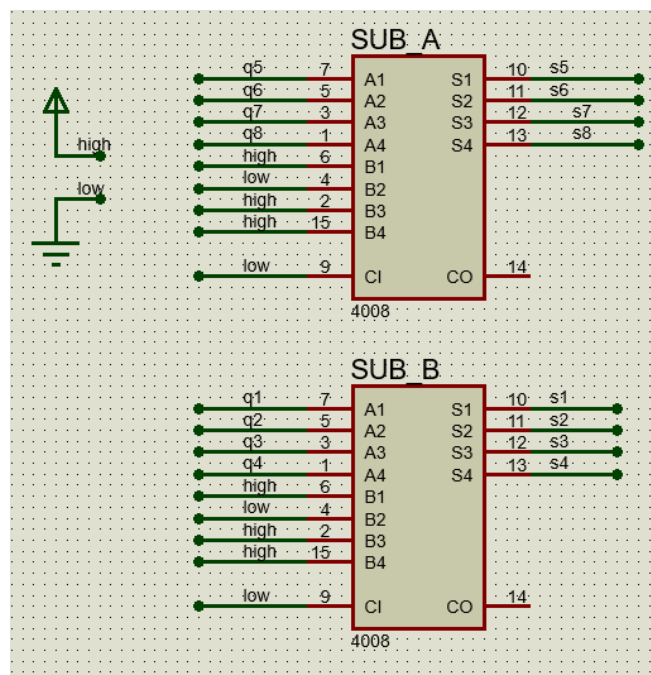
سیگنال‌های $start$ و end نیز به ترتیب برای فعال شدن مدار و مشخص کردن اتمام کار مدار استفاده می‌شوند. مطابق شکل ۲، برای سیگنال $start$ از یک پوش باتن استفاده شده که با فشردن این دکمه، فلیپ‌فلاپی که در تصویر وجود دارد ست شده و باعث فعال شدن کلاک مدار می‌شود. همچنین با یک شدن سیگنال end ، این فلیپ‌فلاپ ری‌ست شده و کلاک مدار از کار می‌افتد و کار آن به پایان می‌رسد.



شکل ۲: عملکرد سیگنال‌های $start$ و end

۲.۲ ماژول‌های SUB_A و SUB_B

وظیفه این ماژول کم کردن ۳ واحد از عدد داده شده به آن است. رابط آن را در شکل ۳ می‌توان دید و برای آن از ماژول‌های آماده‌ی پروتئوس استفاده شده است.

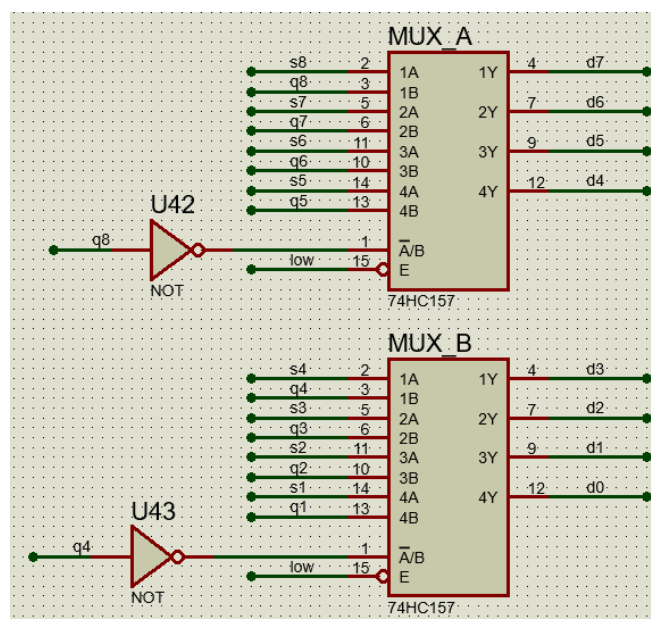


شکل ۳: رابط ورودی و خروجی ماژول تفریق‌کننده

با توجه به الگوریتم، اگر بیت باارزش هریک از ارقام ۱ باشد، باید سه واحد از آن کم کنیم. در این مدار چون ابعاد ورودی ثابت و برابر ۳ رقم است، کافیست فقط رقم‌های اول و دوم را بررسی کنیم؛ چرا که ابتدا شیفت به راست داده و سپس بیت باارزش را بررسی می‌کنیم، در نتیجه بیت باارزش رقم آخر همواره برابر صفر خواهد بود. به این منظور نخست رقم‌های اول و دوم را سه واحد کم کرده و سپس با استفاده از یک مالتی‌پلکسر و براساس بیت باارزش رقم مورد نظر تصمیم می‌گیریم از کدام عدد (عدد اولیه یا عددی که سه واحد از آن کم شده است) استفاده کنیم. همانطور که در شکل ۳ دیده می‌شود، ورودی دوم این ماژول متمم دو عدد ۳ است و ورودی اول آن در یک اینستنس از ماژول بیت‌های رقم اول و در دیگری بیت‌های رقم دوم است. خروجی آن نیز رقم اول منهای سه و رقم دوم منهای سه خواهد بود.

۳.۲ ماژول‌های MUX_A و MUX_B

همانطور که در بخش (ماژول‌های SUB_A و SUB_B) اشاره شد، نیاز داریم تصمیم بگیریم از بین رقم اولیه یا رقم منهای سه شده، یکی را انتخاب کنیم. این تصمیم براساس بیت باارزش آن رقم خواهد بود. وظیفه این ماژول‌ها انجام همین کار است. برای آن از ماژول‌های آماده‌ی پروتئوس استفاده شده است. ورودی و خروجی‌های آن‌ها را در شکل ۴ می‌توان دید.

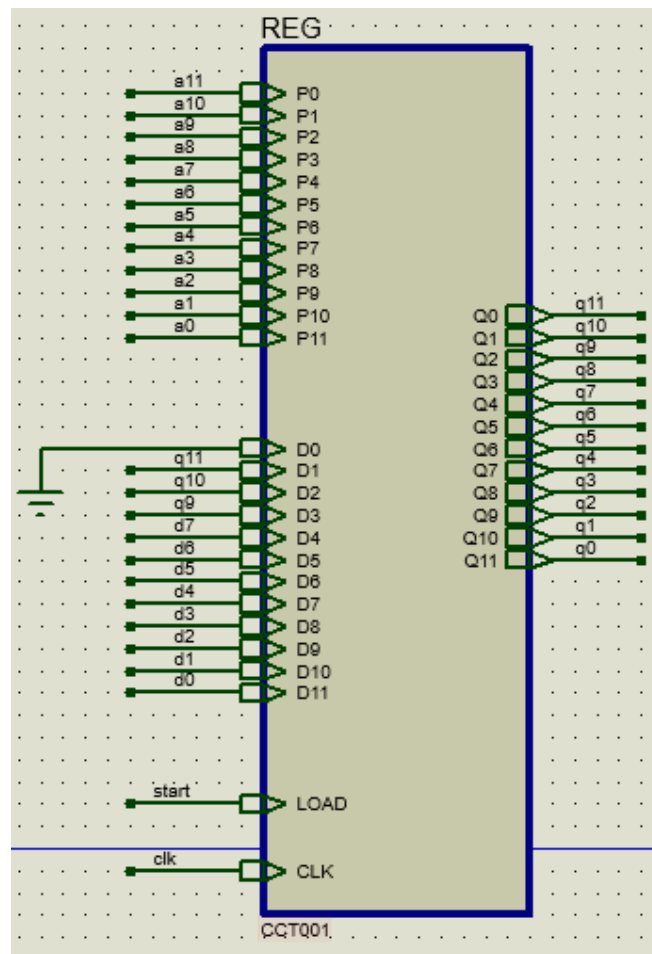


شکل ۴: رابط ورودی و خروجی ماژول مالتی پلکسر

همانطور که دیده می‌شود به ورودی‌های آن بیت‌های رقم اولیه (q ها) و بیت‌های رقم منهای سه شده (s ها) داده شده است. سلکتور آن نیز بیت باارزش رقم مورد نظر (در اینجا q_4) قرار داده شده است. (در اینستنس دیگر این ماژول، سلکتور بیت باارزش رقم دوم یعنی q_8 قرار داده شده است.) هم‌چنین دقت کنید که از بیت‌های چهارم و هشتم استفاده شده، چرا که در مدار تفریق قبل از شیفت‌دادن انجام و خروجی آن آماده شده است.

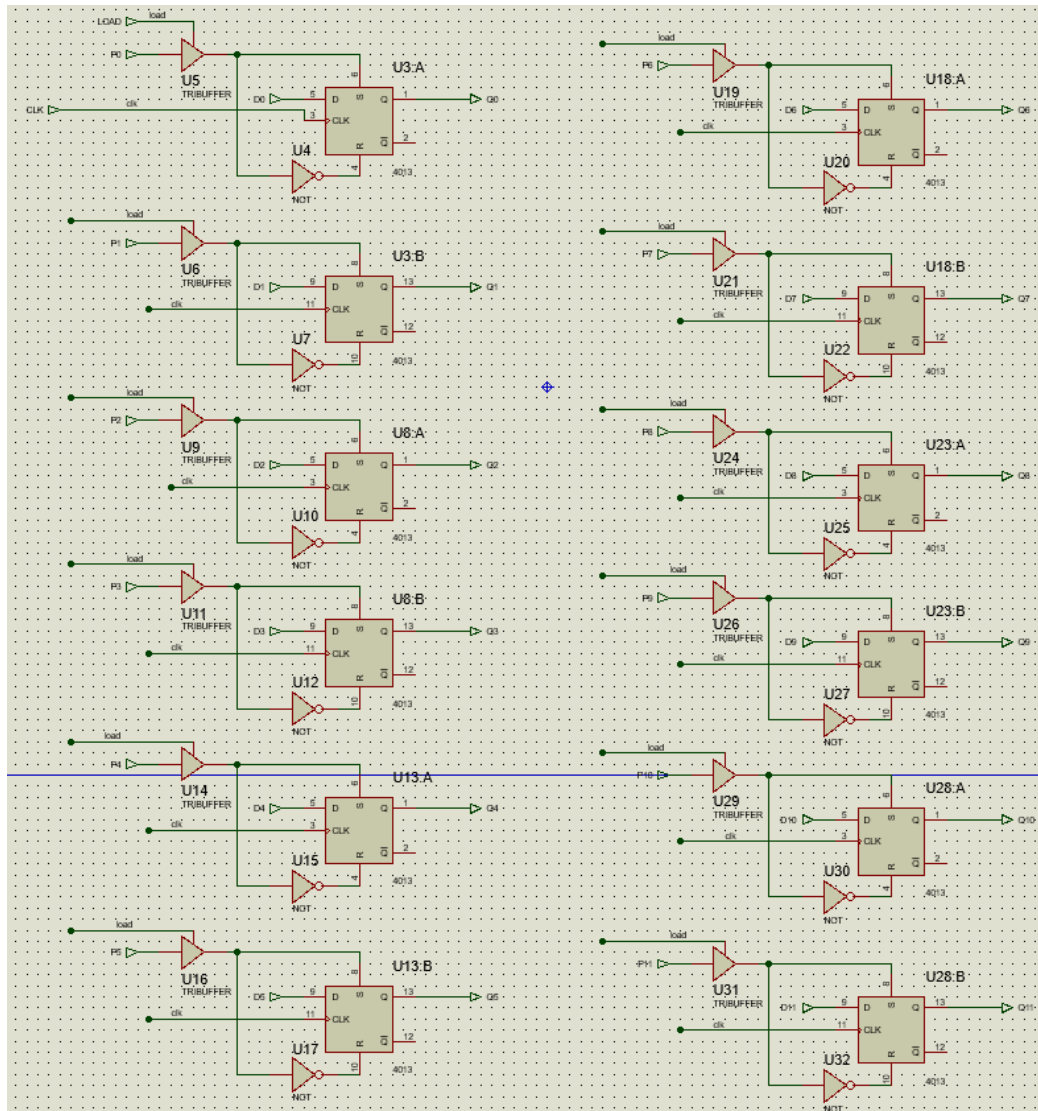
۴.۲ ماژول REG

این ماژول وظیفه‌ی نگهداری مقدار ورودی را دارد. در مراحل مختلف الگوریتم نیاز به شیفت دادن ورودی به سمت راست داریم که مقدار آن پس از انجام شیفت در این ماژول ذخیره می‌شود. هم‌چنین این ماژول قابلیت بارگذاری موازی را نیز داراست، چرا که در ابتدای اجرای مدار نیاز داریم ورودی را در آن ذخیره کنیم. رابط این ماژول در شکل ۵ آمده است.



شکل ۵: رابط ورودی و خروجی ماژول شیفت‌دهنده

همانطور که در تصویر دیده می‌شود در ابتدا ورودی مدار یعنی سیگنال‌های a_0 تا a_{11} در این ماژول بارگذاری می‌شوند. خروجی آن در هر لحظه سیگنال‌های q_0 تا q_{11} خواهد بود که مقدار ورودی را در مراحل مختلف الگوریتم نشان می‌دهد. همچنین ورودی‌های D در این ماژول مقدار جدیدی هستند که قرار است در رجیسترها ذخیره شوند. این مقدار جدید از ۱۲ بیت تشکیل شده که ۴ بیت ابتدایی آن همان بیت‌های باارزش ورودی هستند که یک واحد به سمت راست شیفت داده شده‌اند. چهار بیت دوم، سیگنال‌های d_4 تا d_7 هستند که خروجی مالتی‌پلکسر MUX_A هستند. به همین ترتیب چهار بیت سوم، سیگنال‌های d_0 تا d_3 هستند که خروجی مالتی‌پلکسر MUX_B می‌باشند. (برای مشخص شدن این که این سیگنال‌ها چه معنایی دارند به بخش (ماژول‌های MUX_A و MUX_B) مراجعه کنید.) طراحی داخلی این ماژول را در شکل ۶ می‌توان مشاهده کرد.

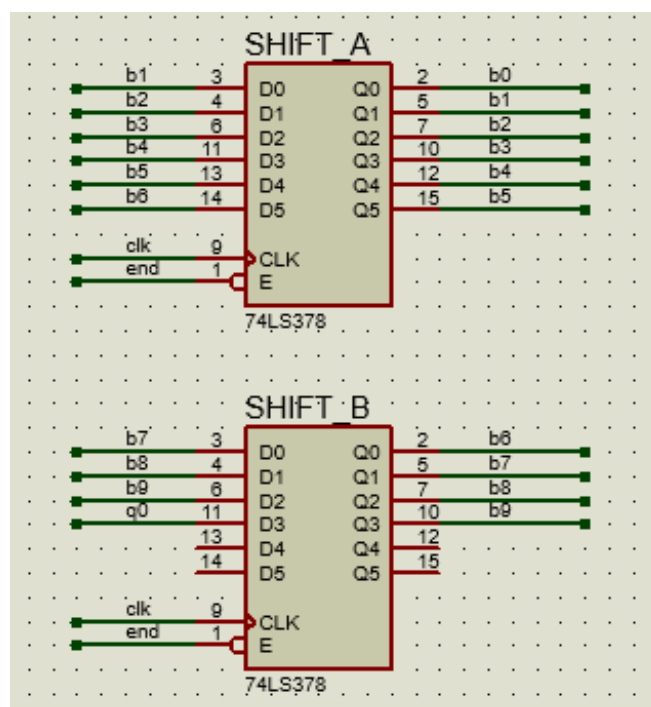


شکل ۶: طراحی داخلی ماژول REG

همانطور که توضیح داده شد ورودی های D به ورودی فلیپ فلاپ ها متصل شده و سیگنال استارت فلیپ فلاپ ها به مقدار ورودی های P با استفاده از یک ترای استیت بافر متصل شده است. این بافر زمانی فعال می شود که ورودی $Load$ ماژول فعال باشد.

۵.۲ ماژول های SHIFT_A و SHIFT_B

این ماژول ها وظیفه ی شیفت دادن به سمت راست خروجی را بر عهده دارند. ورودی و خروجی های آنها در شکل زیر آمده است.

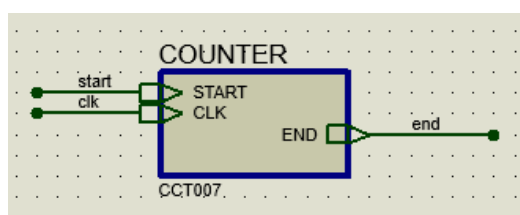


شکل ۷: ورودی و خروجی ماژول‌های SHIFT_A و SHIFT_B

برای این ماژول‌ها از ماژول‌های آماده‌ی پروتئوس استفاده شده است. همانطور که در تصویر دیده می‌شود ورودی b_i به خروجی b_{i-1} رفته و ورودی q_0 که همان بیت اول عدد ورودی است، به خروجی b_9 می‌رود. با این کار در هر کلاک سیگنال‌های b یک واحد به راست شیفت داده می‌شوند و بیت q_0 نیز از سمت چپ به آن‌ها اضافه می‌شود.

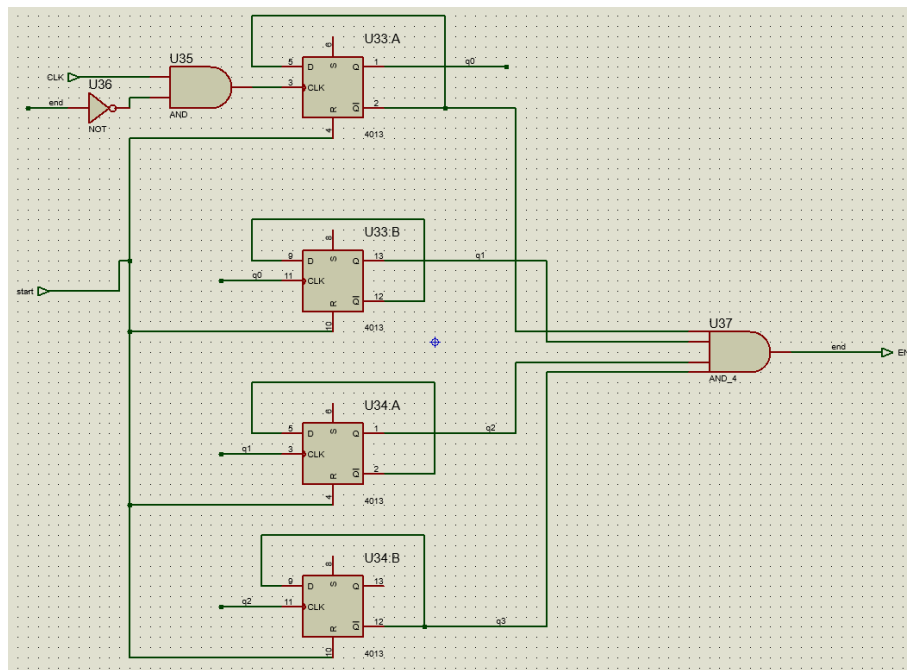
۶.۲ ماژول Counter

وظیفه این ماژول شمارش تا عدد ۱۰ است. ورودی و خروجی‌های آن را در شکل ۸ می‌توان دید.



شکل ۸: رابط ورودی و خروجی ماژول شمارنده

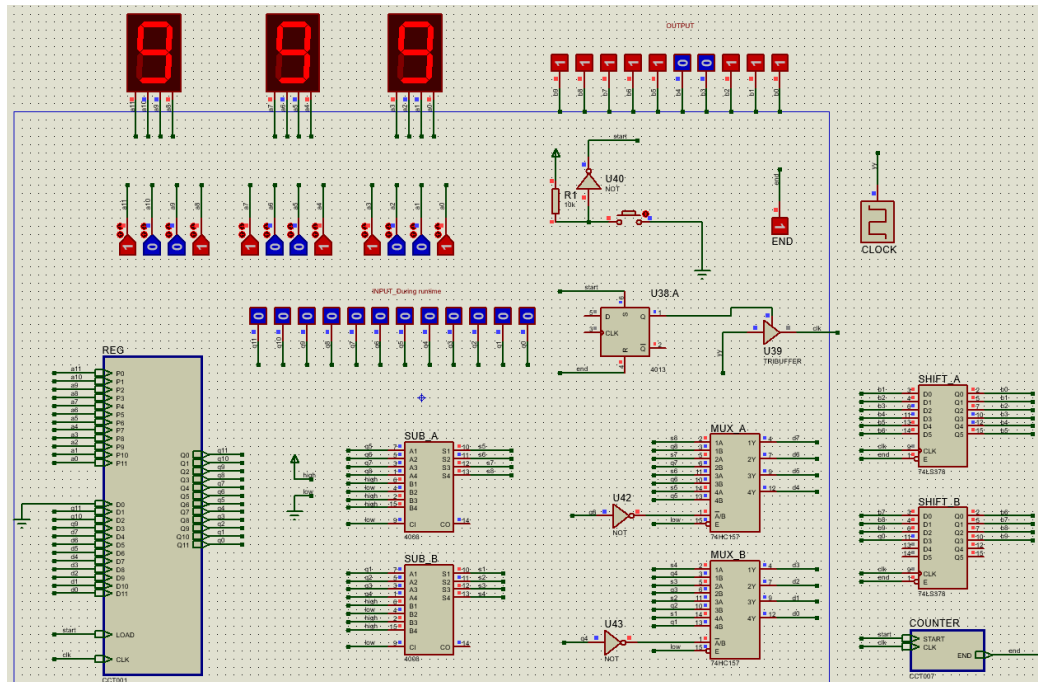
همانطور که در تصویر دیده می‌شود، با آمدن سیگنال $start$ این ماژول شمارش خود را آغاز می‌کند و تا ۱۰ می‌شمارد و پس از اتمام شمارش سیگنال end را فعال می‌کند. طراحی داخلی این ماژول را در شکل ۹ می‌توان مشاهده کرد.



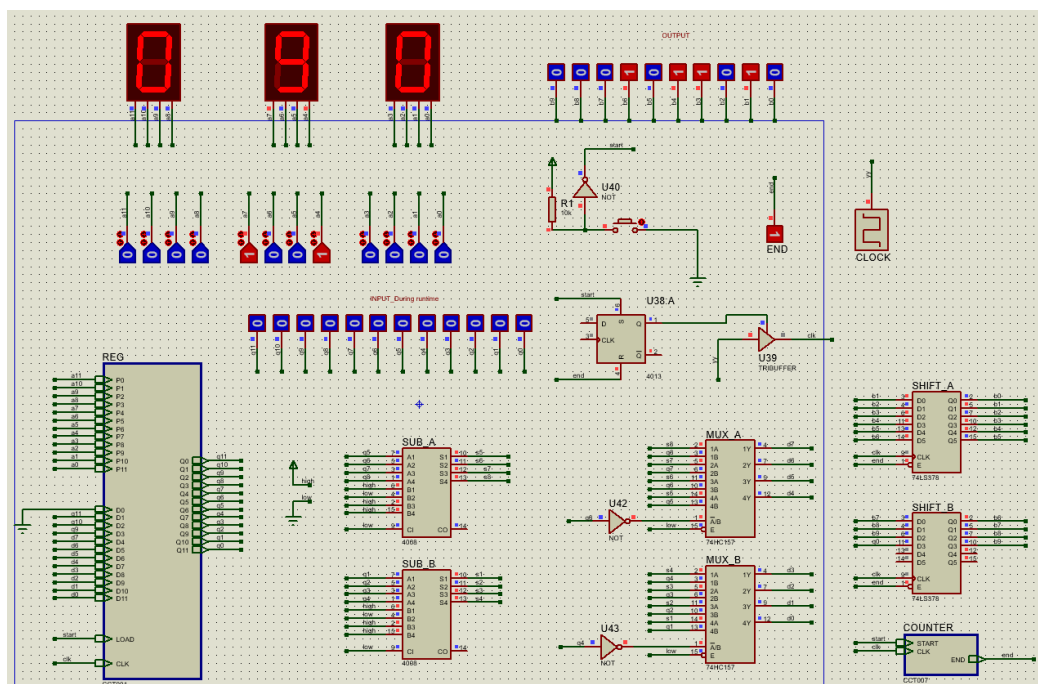
شکل ۹: طراحی داخلی مازول شمارنده

۳ تست مدار

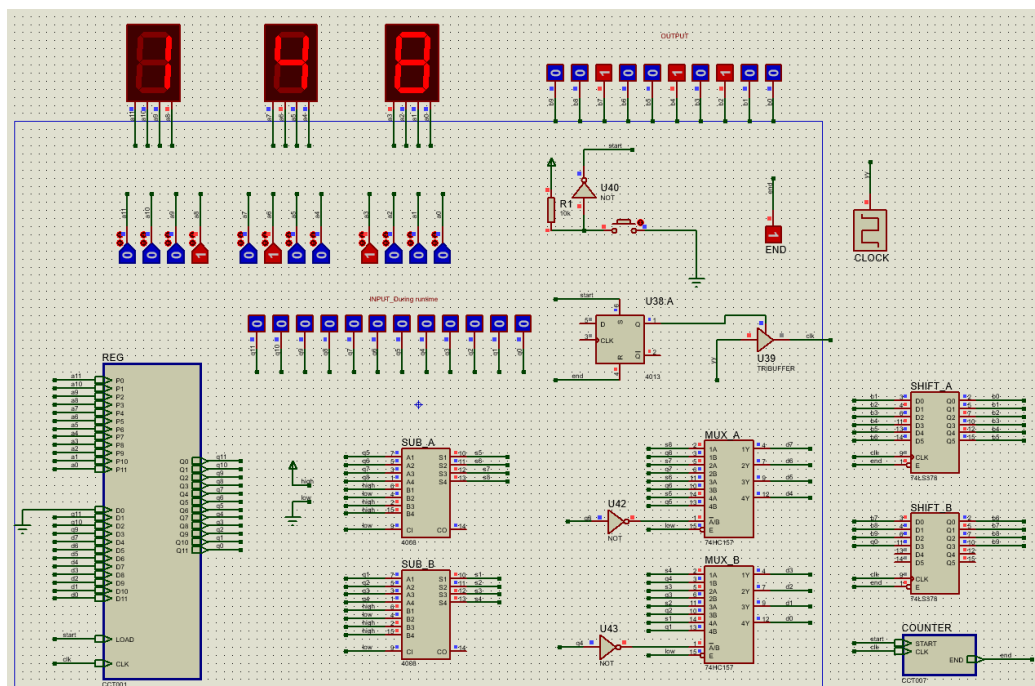
به منظور تست و شبیه‌سازی مدار، عملکرد مدار روی ورودی‌های مختلف بررسی شده است. در شکل‌هایی که در ادامه آمده است می‌توان خروجی و کارکرد مدار را به ازای ورودی‌های مختلف مشاهده کرد.



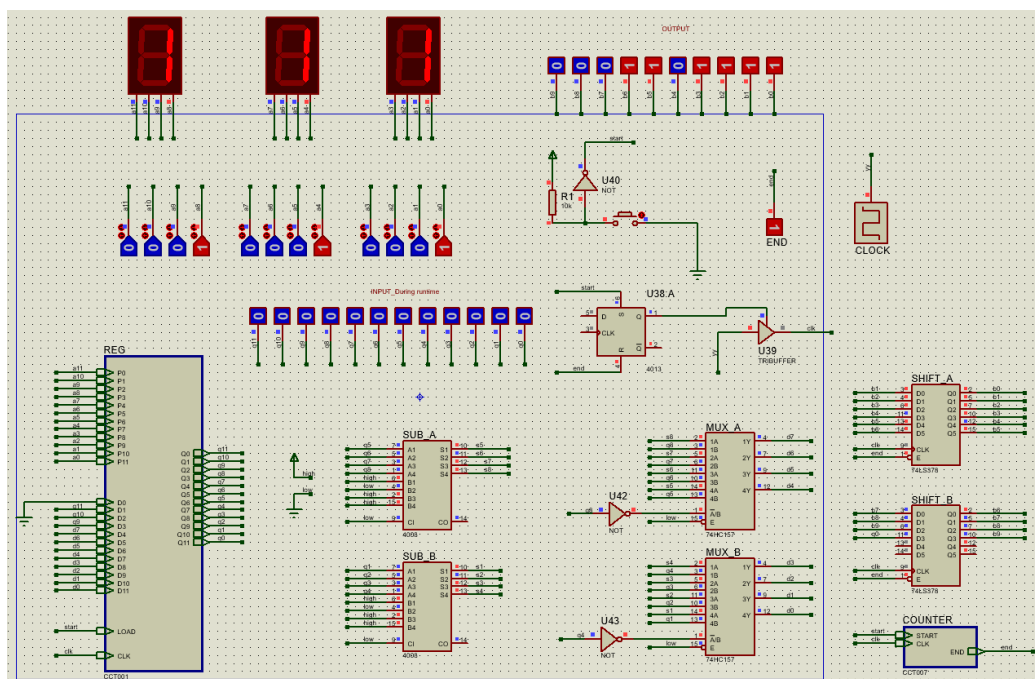
شکل ۱۰: تست مدار به‌ازای ورودی ۹۹۹



شکل ۱۱: تست مدار به‌ازای ورودی ۹۰



شکل ۱۲: تست مدار به‌ازای ورودی ۱۴۸



شکل ۱۳: تست مدار به‌ازای ورودی ۱۱۱