

بسم الله الرحمن الرحيم



آزمایشگاه معماری کامپیوتر
گزارش کار آزمایش پنجم: مبدل دهنده به دودویی

دانشکده مهندسی کامپیوتر
دانشگاه صنعتی شریف

تابستان ۱۴۰۳

رادین چراغی ۴۰۱۱۰۵۸۱۵
مبین پورعابدینی ۴۰۱۱۱۰۵۵۶
آرین نوری ۴۰۱۱۰۶۶۶۳

مقدمه

در این آزمایش، هدف طراحی یک مبدل دهنده به دودویی به کمک ابزار Proteus است. به طور کلی مدار شامل سه ورودی چهار بیتی است که هر کدام یک عدد BCD را نمایش می‌دهند، همچنین یک سیگنال Start نیز داریم که به معنی آغاز فرایند الگوریتم تبدیل است. در ادامه یک خروجی ده بیتی خواهیم داشت که عدد سه رقمی BCD داده شده را به صورت دودویی نمایش می‌دهد و به کمک خروجی تک بیتی End نیز آماده بودن جواب را اعلام می‌کنیم. پس به این ترتیب ورودی‌ها و خروجی‌ها به این گونه هستند:

Inputs

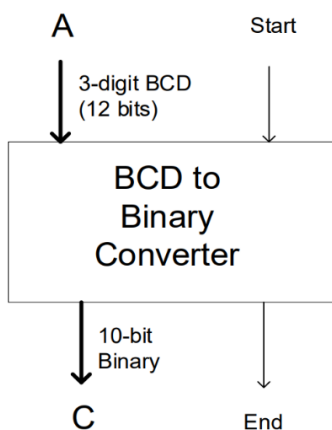
A: a twelve-bit number which contains three BCD digits

Start: start the operation

Outputs

C: a ten-bit number which is the result of the conversion

End: end of the operation



شکل ۱. نمای کلی مدار خواسته شده

شرح آزمایش

حالا به نحوه انجام آزمایش و توضیح الگوریتم تبدیل می‌پردازیم. با فعال شدن سیگنال Start، مدار شروع به کار می‌کند و ورودی دهنده را که یک عدد سه رقمی است به عدد معادل دودویی تبدیل کرده و نتیجه را روی خطوط خروجی قرار داده و سیگنال End را روشن می‌کند. در انتها نیز به کمک ورودی Start می‌توان دوباره مدار را Reset کرد و همین کار را تکرار کرد. الگوریتم تبدیل یک عدد دهنده ۲ رقمی به دودویی معادل به صورت زیر است:

- (الف) عدد دهنده ورودی را به سمت راست شیفت دهید.
- (ب) اگر با ارزش‌ترین بیت (بیت چهارم) هر رقمی از مدار یک است، از آن رقم ۳ تا کم کنید.
- (ج) مراحل (الف) و (ب) را تا زمانی که تمامی بیت‌های عدد دهنده ورودی صفر نشده‌اند، تکرار کنید.
- (در این مدار حداکثر ده بار تکرار لازم است)

در پایان بیت‌هایی که با شیفت به راست بیرون می‌آیند، عدد دودویی معادل را می‌سازند.

برای درک بهتر الگوریتم داده شده به جدول زیر توجه کنید که تبدیل عدد دهدهی ۱۱۰ به دودویی آن است:

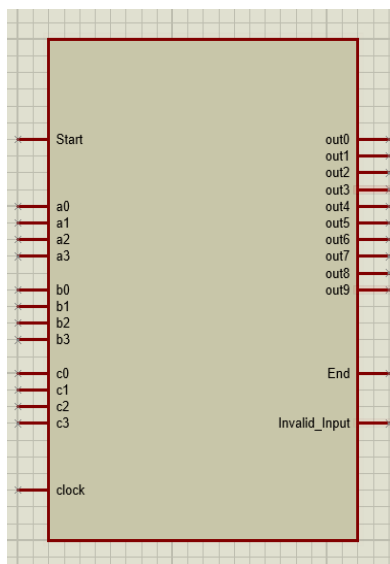
عمل	خروجی	رقم اول	رقم دوم	رقم سوم
شیفت به راست	خالی۱	...۱
از رقم‌های اول و دوم ۳ تا کم کن	.	۱۰۰۰	۱۰۰۰
شیفت به راست	.	۰۱۰۱	۰۱۰۱
از رقم اول ۳ تا کم کن	۱۰	۱۰۱۰	۰۰۱۰
شیفت به راست	۱۰	۰۱۱۱	۰۰۱۰
شیفت به راست	۱۱۰	۰۰۱۱	...۱
از رقم اول ۳ تا کم کن	۱۱۱۰	۱۰۰۱
شیفت به راست	۱۱۱۰	۰۱۱۰
شیفت به راست	۰۱۱۱۰	۰۰۱۱
شیفت به راست	۱۰۱۱۱۰	۰۰۰۱
پایان عملیات	۱۱۰۱۱۱۰

جدول ۱. نمونه‌ای برای درک بهتر فرآیند

در نهایت از مدار انتظار می‌رود الگوریتم بالا را انجام دهد و در صورتی که به خروجی مناسب رسید سیگنال End را روشن کند.

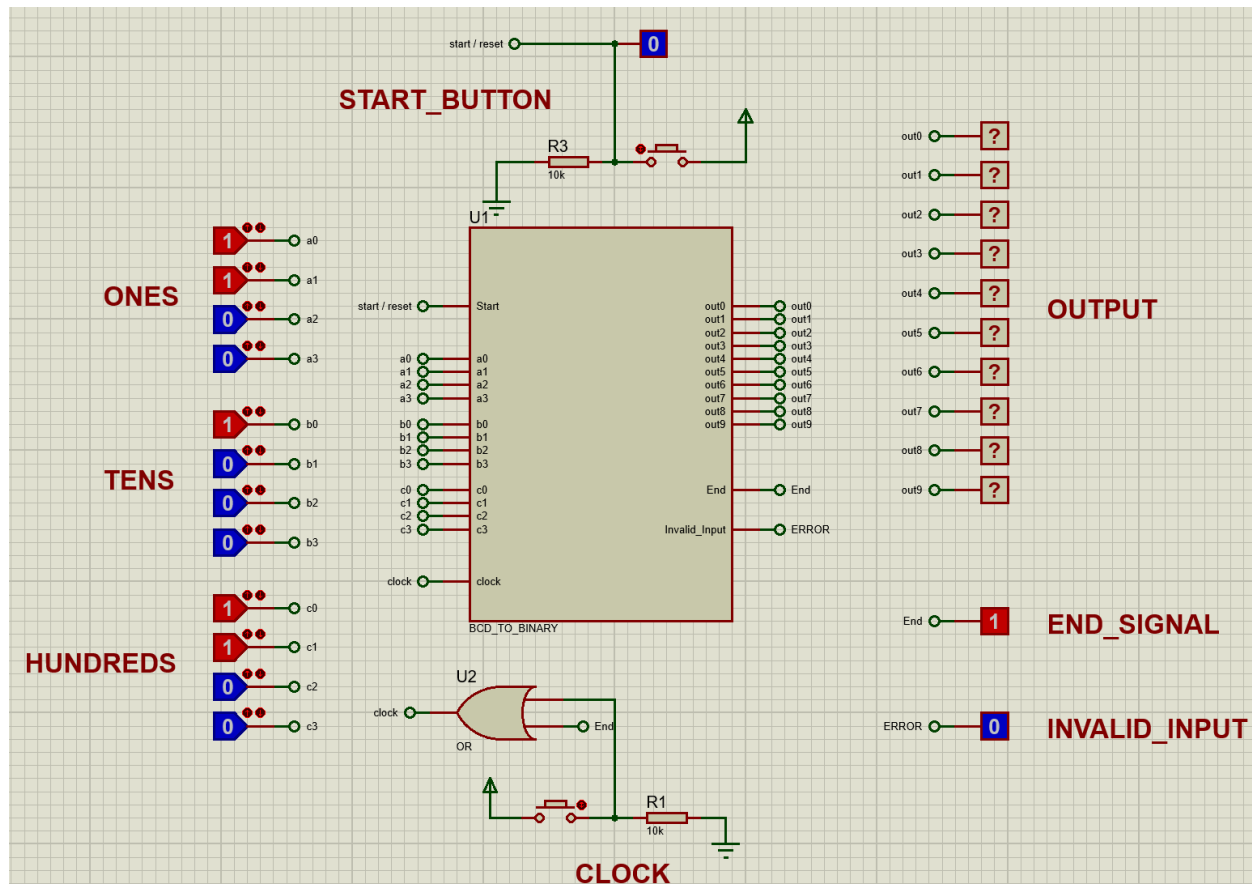
ساخت مدار در پروتئوس

حال که مراحل را متوجه شدیم به پیاده سازی بر روی برنامه‌ی پروتئوس می‌رسیم. ابتدا قطعه‌ی زیر را می‌سازیم تا در ادامه درون آن را کامل کنیم و به عنوان مبدل از آن استفاده کنیم، همان طور که می‌بینید ورودی‌های آن عبارت است از سیگنال Start، سه عدد BCD که یکان، دهگان و صدگان ما هستند، و سیگنالی برای انتقال لبه‌ی clock. خروجی‌ها نیز مطابق انتظار یک خروجی ده بیتی که تبدیل شده‌ی عدد است و سیگنال End برای اتمام عملیات. همچنین یک خروجی دیگر با نام Invalid Input در مبدل قرار می‌دهیم که عملکرد آن تنها برای تایید مقادیر ورودی‌هاست، به این شکل که کنترل می‌کند ورودی‌های ما سه رقم BCD باشند و این یعنی در صورتی که اعداد ۱۰۱۰ تا ۱۱۱۱ را در رقم‌های ورودی مشاهده کند به کاربر خطای ورودی اشتباه می‌دهد.



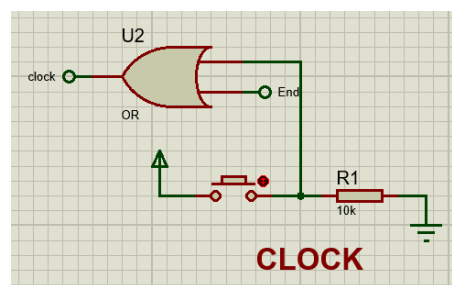
شکل ۲. تصویر قطعه‌ی اصلی مدار

حال تمامی ورودی‌ها و خروجی‌ها را به شکل مناسبی طراحی کرده و در کنار قطعه‌ی اصلی قرار می‌دهیم، برای سیگنال‌های Start و clock مطابق شکل زیر از Push Button استفاده می‌کنیم.



شکل ۳. نمای کلی مدار نهایی

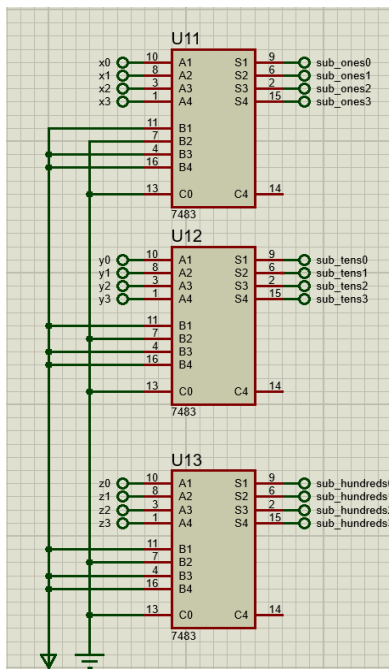
دقت دارید که در طراحی نهایی مدار ما سیگنال End را با clock با یک گیت OR ترکیب کرده‌ایم و این کار به ما کمک می‌کند تا در صورتی که مدار به پایان رسید دیگر خروجی ما با زدن clock از بین نرود.



شکل ۴. منطق طراحی ساعت

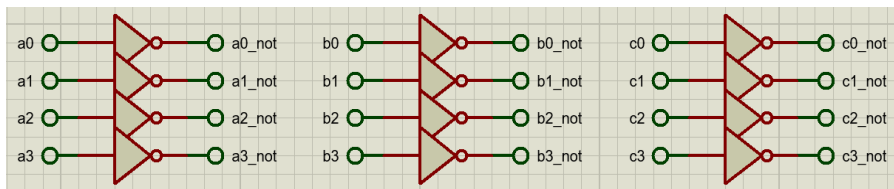
در طراحی مدار ما به دوازده D flip flop برای ذخیره‌ی مقادیر ورودی و به ده D flip flop برای ذخیره‌ی عدد نهایی خروجی نیاز داریم. همچنین می‌دانیم که باید در هر مرحله از بیت صفرم خروجی شروع کرده و تا زمان پایان مدار بیت‌ها را وارد خروجی کنیم، در نتیجه برای این کار از یک شمارنده کمک می‌گیریم که در هر مرحله به ما اطلاع دهد در صورت شیفت، بیت چندم خروجی مقدار می‌گیرد. همچنین برای این که تغییر بیت‌های ورودی در هنگام اجرای الگوریتم، خللی به آن وارد نکند، مقادیر Set و Reset فلیپ‌فلاپ‌ها را طوری درست می‌کنیم که تنها با سیگنال Start مقدار گیرند. برای تفريق با ۳ نیز به سه جمع کننده نیاز داریم که اعداد ورودی را با منفی ۳ جمع کنند. حالا با دانستن خلاصه‌ای از عملکرد مدار در ادامه به شکل دقیق جزئیات آن را بررسی می‌کنیم.

در تصویر زیر سه جمع کننده وجود دارد که هر کدام اعداد X ، Y و Z را با مقدار دودویی منفی ۳ یعنی ۱۱۰۱ جمع کرده و خروجی را در ترمینال‌هایی نگه می‌دارند.



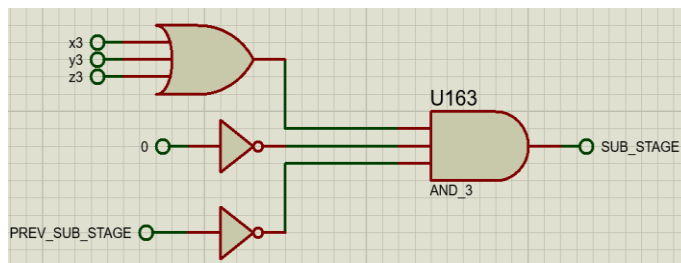
شکل ۵. جمع‌کننده‌های مدار برای تفریق با ۳

برای آسانی فهم مدار از ترمینال‌های زیر که نقیض ورودی‌ها هستند کمک می‌گیریم:



شکل ۶. ترمینال‌های نقیض کمکی

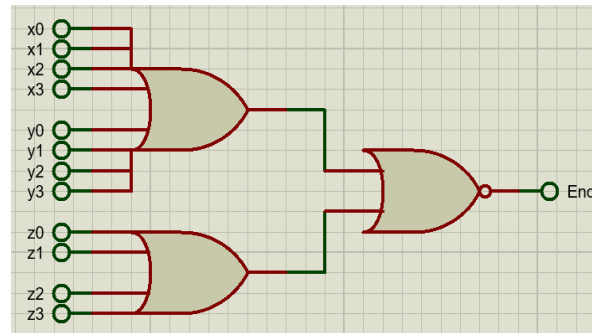
همچنین سیگنالی را با عنوان SUB_STAGE در نظر می‌گیریم که وظیفه‌ی آن این است برای ما تعیین کند در واحد زمانی حاضر باید عملیات شیفرت صورت گیرد یا تفریق.



شکل ۷. سیگنال کمکی SUB_STAGE

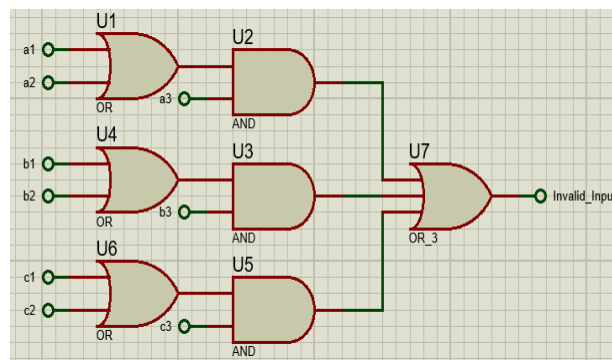
نحوه کارکرد این سیگنال این گونه است که اگر یکی از بیت‌های پرارزش اعداد ۱ باشد، و در واحد زمانی اول مدار نباشیم، و در واحد زمانی قبلی عملیات تفریق را انجام نداده باشیم، باید عملیات تفریق را انجام دهیم. لازم به ذکر است که توسط سیگنال PREV_SUB_STAGE و یک DFF نیز می‌توانیم عملیات مرتبه‌ی زمانی قبل را نیز ذخیره کنیم و از آن استفاده کنیم.

همان طور که پیش تر گفته شد، سیگنال End زمانی فعال می شود که تمامی ورودی ها صفر شوند، پس منطق آن به صورت زیر خواهد بود:



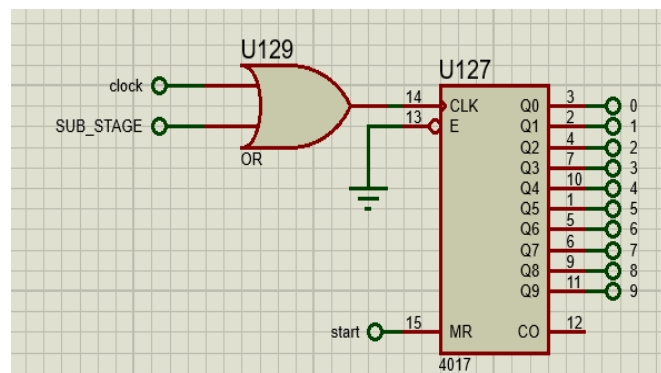
شکل ۸. نحوه پیاده سازی سیگنال End

همچنین برای بررسی صحیح بودن ورودی ها (در دامنه ی BCD) نیز به این صورت سیگنال خطا را می سازیم:



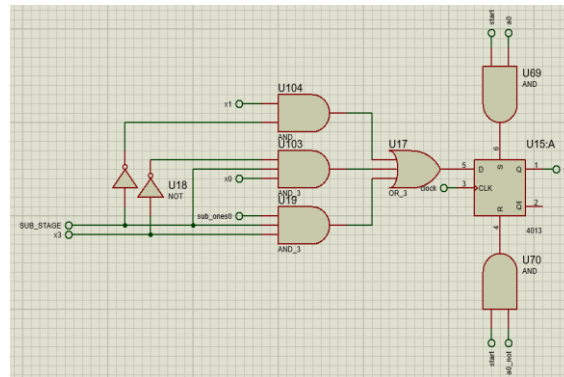
شکل ۹. سیگنال خطای ارقام ورودی

گفتیم که برای تعیین شماره ی بیتی از خروجی که با شیفِت به راست مقدار می پذیرد، از یک شمارنده استفاده می کنیم، در شکل زیر می توانید نحوه ی استفاده از آن را ببینید، مقدار clock آن را با سیگنال SUB_STAGE، مقدار clock آن را با سیگنال SUB_STAGE، مقدار clock آن را با سیگنال SUB_STAGE، مقدار clock آن را با سیگنال SUB_STAGE می کنیم که تنها در زمانی که در حالت شیفِت هستیم این شمارنده کارش را انجام دهد، همچنین مقدار Reset آن نیز به سیگنال Start متصل می شود.



شکل ۱۰. شمارنده برای تعیین بیت خروجی

حالا که تمامی جزئیات کم اهمیت تر مدار را بررسی کردیم، به توضیح دو بخش اصلی مدار می پردازیم یعنی نحوه ی ذخیره ی ورودی ها و نحوه ی نگه داشتن بیت های خروجی.

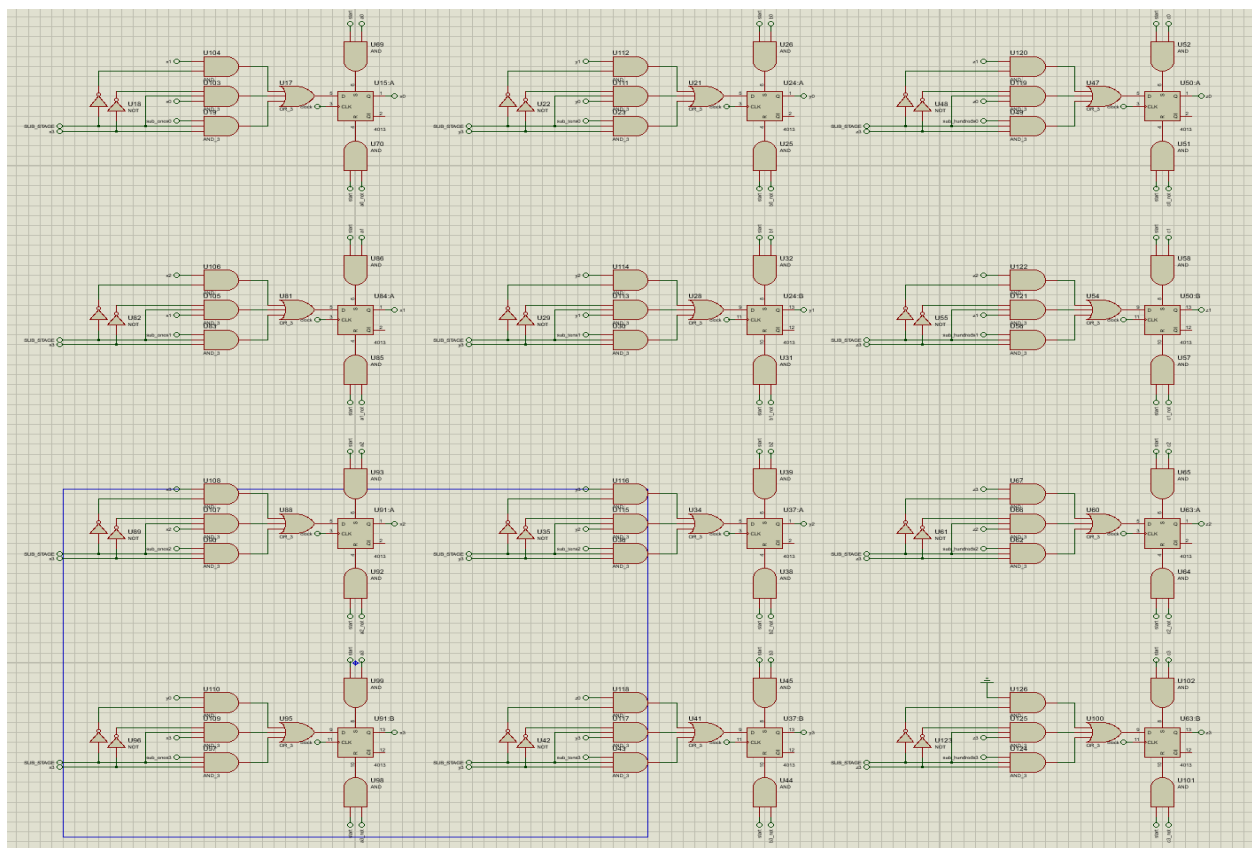


شکل ۱۱. فلیپ‌فلاپ برای ذخیره‌ی بیت ورودی و تغییر آن

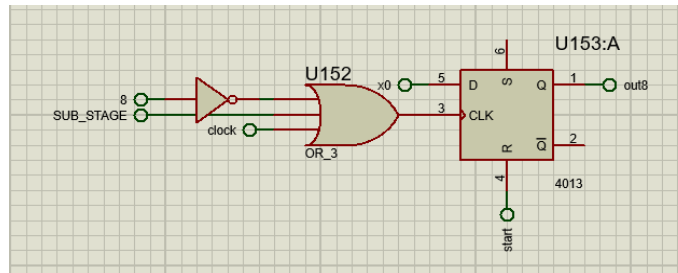
در شکل بالا همان طور که می‌بینید با فعال شدن سیگنال Start، بیت صفرم یکان وارد فلیپ‌فلاپ می‌شود (در صورتی که بیت ۱ باشد به کمک Set و در غیر این صورت به کمک Reset وارد فلیپ‌فلاپ می‌شود). هنگامی که بخواهیم مقدار جدیدی را درون این سیگنال بریزیم با ۳ حالت روبه‌رو هستیم:

- (۱) شیف‌ت به راست: در این حالت سیگنال کناری وارد می‌شود (در مثال بالا سیگنال کناری بیت صفرم یکان، بیت یکم یکان است، همچنین برای بیت آخر در این حالت چون که دیگر بیتی در سمت چپ آن نداریم مقدار صفر را وارد می‌کنیم)
- (۲) عدم تغییر: در این حالت یکی از ارقام مدار در حال انجام عملیات تفریق با ۳ است و این رقم نیازی به تفریق ندارد پس سیگنال مربوطه در این واحد زمانی احتیاجی به تغییر نداشته و مقدار خود را نگه می‌دارد.
- (۳) تفریق رقم با ۳: این حالت به این معنی است که رقم مربوطه بیت پرارزشش ۱ بوده و نیاز دارد تا با منفی ۳ جمع شود و خروجی این جمع را درون فلیپ‌فلاپ‌هایی که رقم را نگه می‌دارند بریزد.

حال که تمامی حالات را بررسی کردیم کل بخش ذخیره و تغییر ورودی‌ها را در عکس زیر می‌توانید مشاهده کنید:

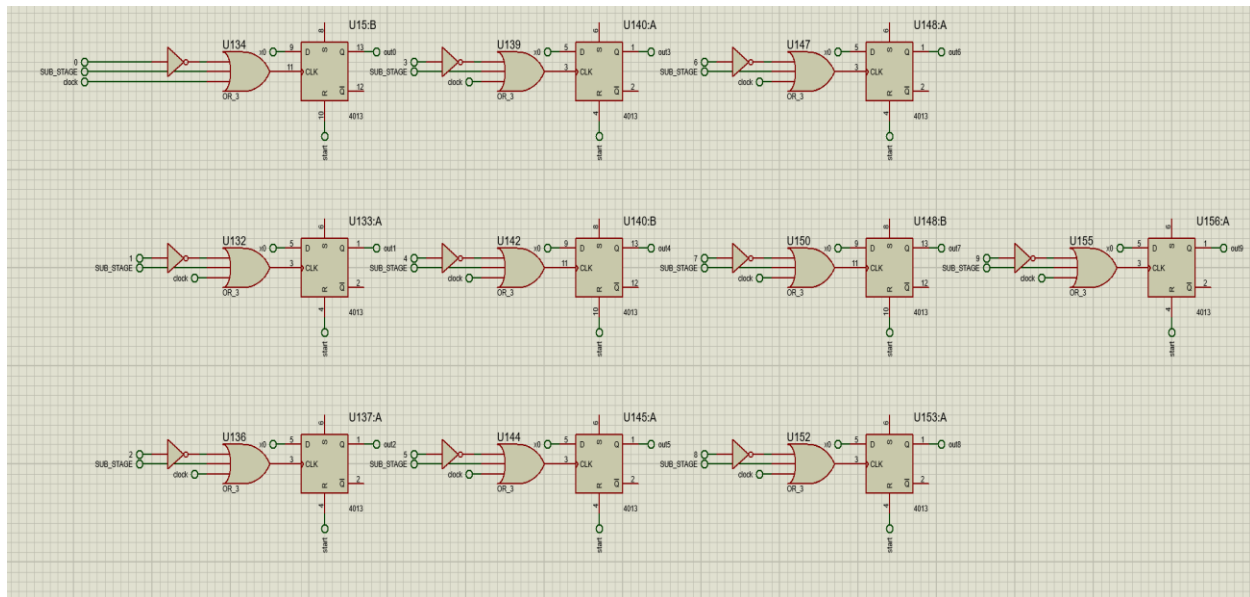


شکل ۱۲. نمای کلی تمامی فلیپ‌فلاپ‌های سه رقم ورودی



شکل ۱۳. فلیپ‌فلاپ برای مقدار دادن به بیت‌های خروجی

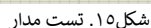
تنها بخش باقی مانده از مدار نحوه‌ی ایجاد خروجی‌هاست. ما ده بیت خروجی داریم و برای هر کدام یک فلیپ‌فلاپ در نظر می‌گیریم، ورودی این فلیپ‌فلاپ‌ها همواره بیت صفرم یکان است، چرا که هر بار با شیفت به راست دادن این بیت از مدار خارج می‌شود و طبق الگوریتم باید به خروجی برسد. نکته‌ی اصلی پیدا کردن جایگاه مناسب این بیت خروجی است که پیش‌تر گفتیم به کمک شمارنده این مشکل را حل می‌کنیم. در شکل بالا می‌بینید که clock به صورت OR سه‌تایی از یک عدد، سیگنال SUB_STAGE و clock اصلی مدار است. علت آن این است که می‌خواهیم تعیین کنیم اگر در شماردهی درست بیت خروجی نیستیم، و یا اگر در حالت تفریق هستیم، لبه‌ی clock برای ما بی‌اثر شود و فقط هنگامی تأثیر بپذیرد که در حالت شیفت باشیم و همچنین شمارنده به ما اعلام کند نوبت سیگنال خروجی مربوطه است. همچنین سیگنال Reset را نیز به Start متصل می‌کنیم تا در هنگام آغاز مجدد مدار خروجی‌ها همگی پاک شوند. در شکل زیر نیز کل بخش مربوطه را مشاهده می‌کنید:



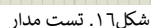
شکل ۱۴. تمامی DFF‌های خروجی با منطقی مشابه شکل ۱۳

حالا که تمامی بخش‌های مدار را توضیح دادیم، در ادامه به تست آن خواهیم پرداخت و تصاویر تست را قرار خواهیم داد.

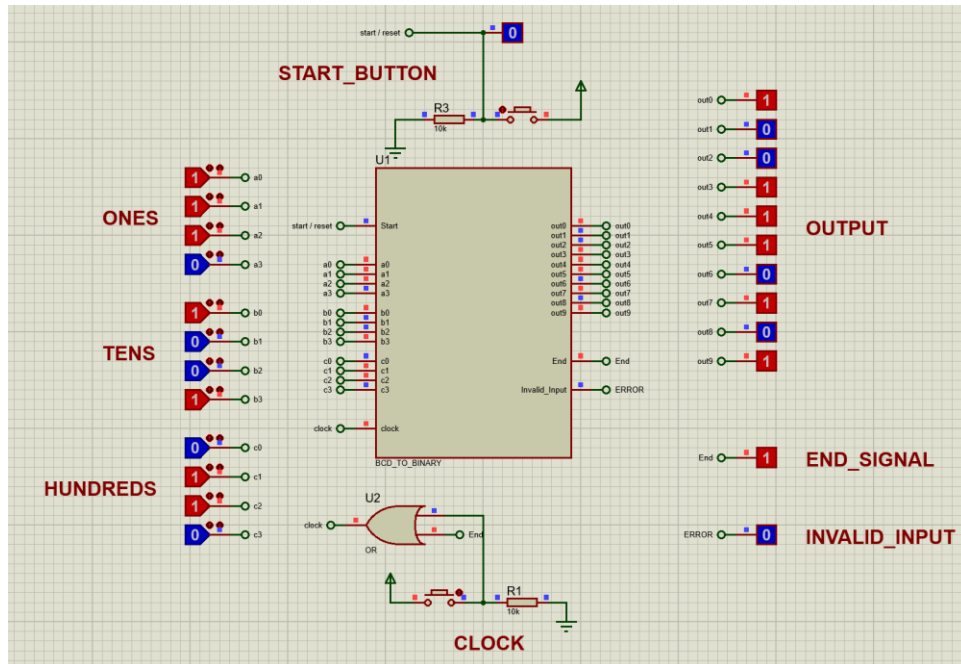
ورودی: ...
خروجی مورد انتظار: Invalid input



ورودی: ۱۰۱۰۱۰۰۱ معادل ۸۱۵ دهدهی
خروجی مورد انتظار: ۱۱۰۰۱۰۱۱۱۱

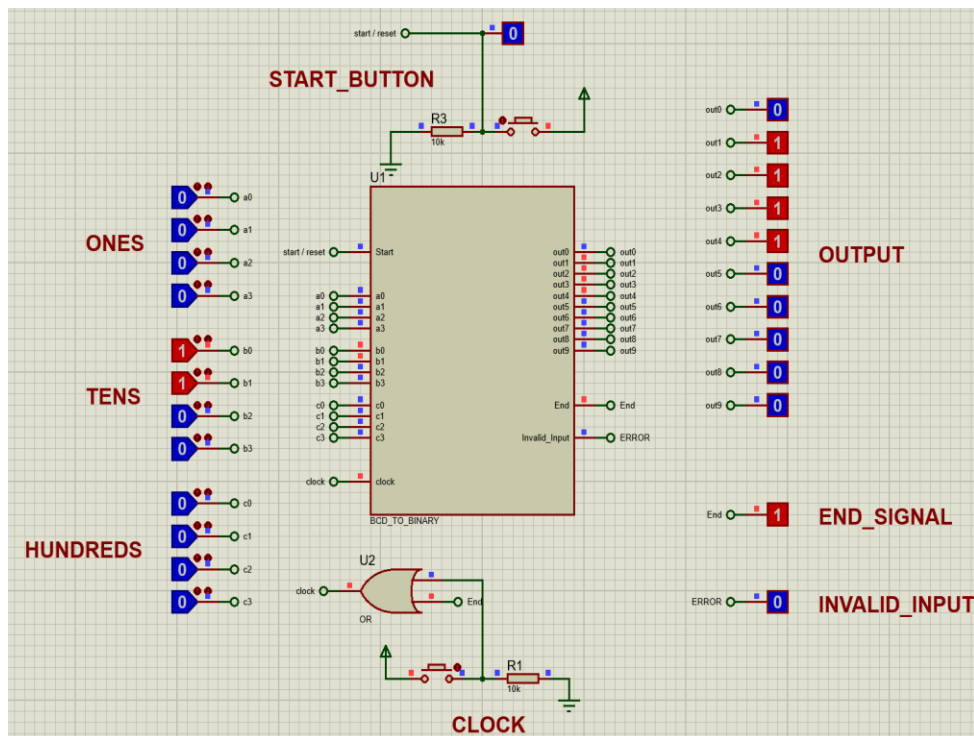


ورودی: ۱۱۱۱۰۱۰۰۱۰۱۱۰۰
معادل ۶۹۷ دهدهی
خروجی مورد انتظار: ۱۰۱۰۱۱۱۰۰۱



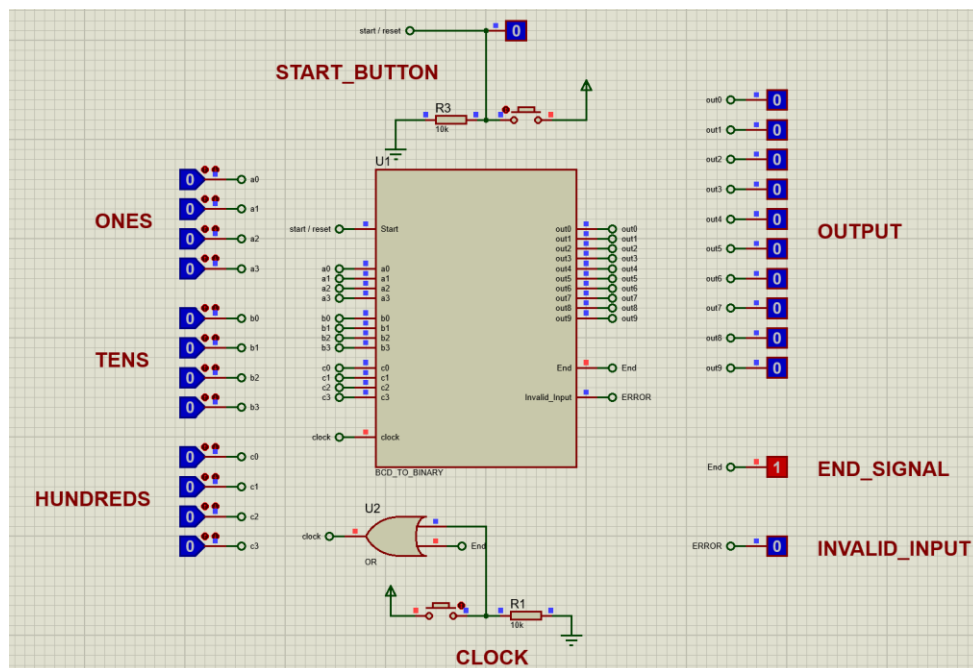
شکل ۱۷. تست مدار

ورودی: ۱۱۰۰۰۰۰۰۰۰۰۰۰۰
معدل ۳۰ دهدهی
خروجی مورد انتظار: ۱۱۱۱۰



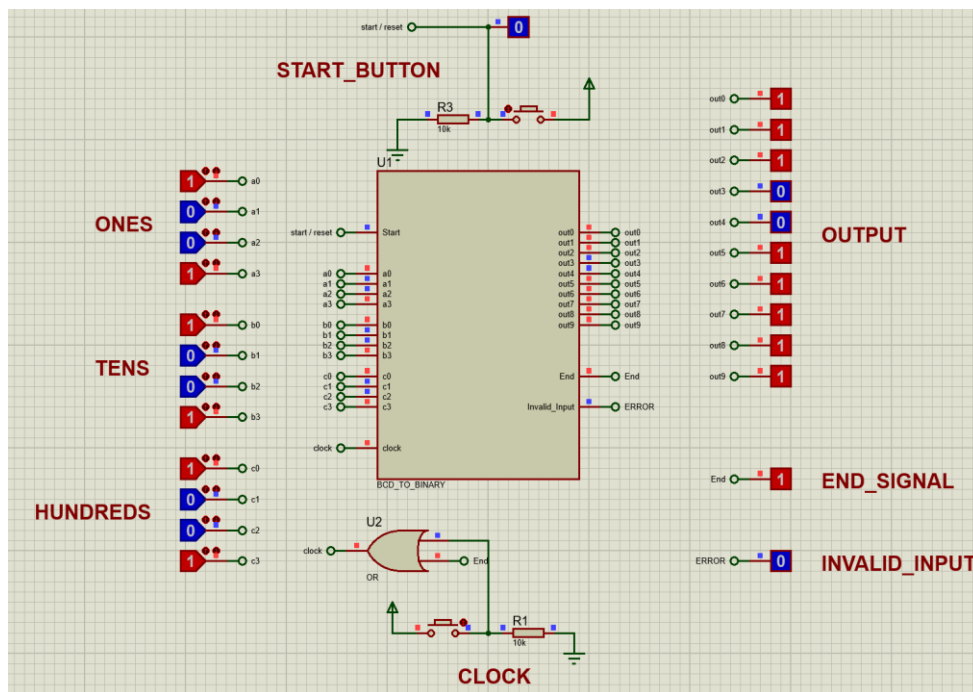
شکل ۱۸. تست مدار

ورودی: معادل صفر دهدهی
خروجی مورد انتظار: صفر



شکل ۱۹. تست مدار

ورودی: ۱۰۰۱ ۱۰۰۱ ۱۰۰۱ معادل ۹۹۹ دهدهی
خروجی مورد انتظار: ۱۱۱۱۱۰۰۱۱



شکل ۲۰. تست مدار

پایان گزارش کار آزمایش پنجم