

به نام خدا



دانشگاه صنعتی شریف  
دانشکده مهندسی کامپیوتر

## آزمایشگاه معماری کامپیوتر

آزمایش دوم:  
ضرب کننده ممیز ثابت

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

پاییز ۱۴۰۰

## فهرست مطالب

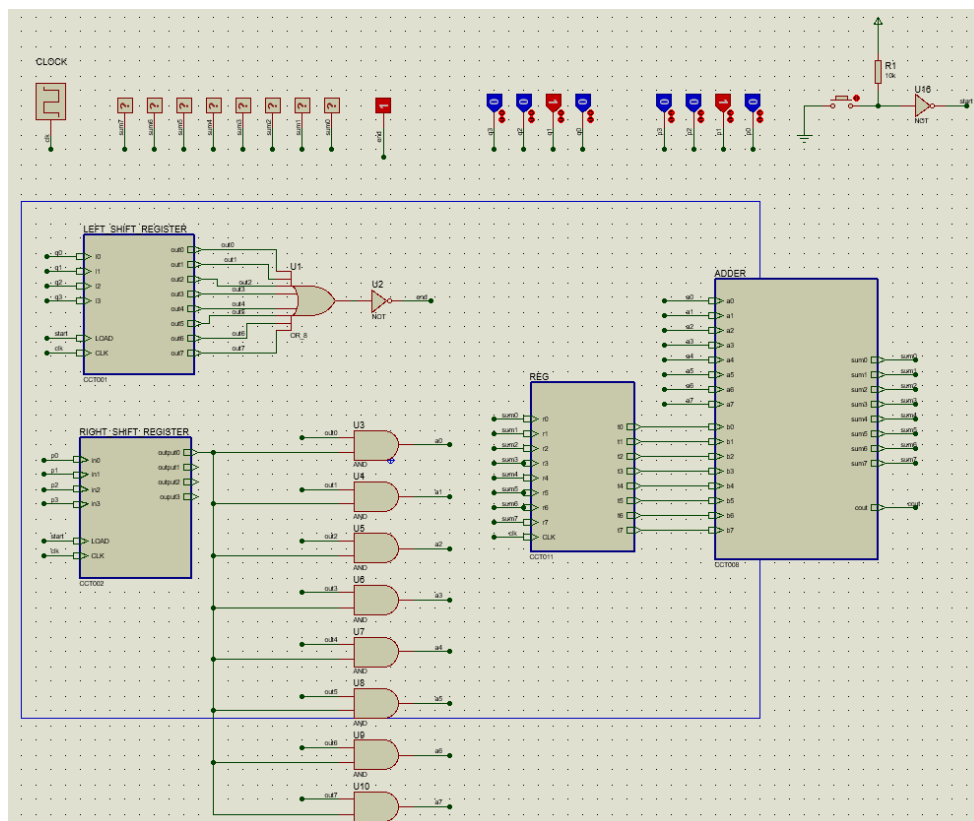
۲	۱ هدف آزمایش
۳	۲ مراحل طراحی و پیاده‌سازی مدار
۳	۱.۲ مازول شیفتهنده به سمت راست . . . . .
۵	۲.۲ مازول شیفتهنده به سمت چپ . . . . .
۶	۳.۲ مازول جمع‌کننده . . . . .
۷	۴.۲ مازول REG . . . . .
۹	۳ تست مدار

## ۱ هدف آزمایش

در این آزمایش هدف پیاده‌سازی مدار ضرب‌کننده دو عدد باینری ممیز ثابت بود. از آنجایی که ضرب دو عدد ممیز ثابت تفاوتی با ضرب دو عدد عادی ندارد و صرفاً کافیست مکان ممیز را در انتها لحاظ کنیم، طراحی یک مدار ضرب‌کننده چهاربیتی خواسته‌ی ما را برآورده می‌کند.

برای ضرب دو عدد باینری نیز از الگوریتم Add & Shift استفاده شده است که یک الگوریتم ترتیبی است. در این الگوریتم در هر مرحله با بررسی کردن بیت اول ضرب‌کننده، تصمیم گرفته می‌شود که ضرب‌شونده را به حاصل اضافه کنیم یا خیر. سپس ضرب‌کننده را یک واحد به سمت راست شیفت داده تا بیت بعدی آن مورد بررسی قرار گیرد، هم‌چنین ضرب‌شونده نیز یک واحد به سمت چپ شیفت داده می‌شود چرا که مرتبه‌ی بیتی که قرار است در آن ضرب شود، یک واحد افزایش پیدا کرده است.

بنابراین برای طراحی مدار نیاز به ماژول‌های شیفت‌دهنده به سمت چپ، شیفت‌دهنده به سمت راست و یک جمع‌کننده هشت بیتی نیاز بوده که جزئیات طراحی این ماژول‌ها در بخش بعدی آمده است. در شکل ۱ نیز نمای کلی مدار طراحی شده آمده است که سیگنال‌های ورودی آن، شامل ضرب‌کننده و ضرب‌شونده به ترتیب با p و q نشان داده شده‌اند. سیگنال ورودی start برای شروع کار مدار آمده است و خروجی در هشت بیت sum نمایش داده شده که با یک شدن سیگنال end، خروجی مدار آماده شده است.



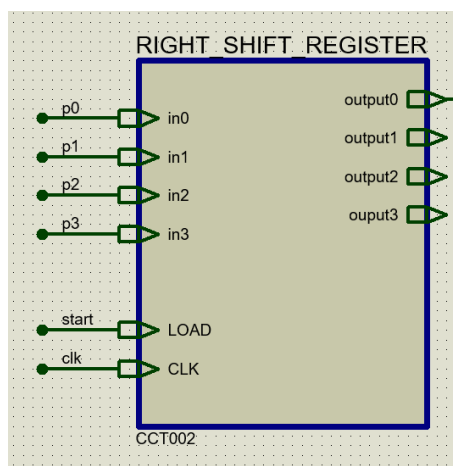
شکل ۱: نمای کلی مدار ضرب‌کننده

## ۲ مراحل طراحی و پیاده‌سازی مدار

همانطور که در بخش قبل گفته شد، مدار مورد نظر از سه ماژول اصلی شیفت‌دهنده به سمت راست، شیفت‌دهنده به سمت چپ و جمع‌کننده هشت‌بیتی تشکیل شده است. همچنین ماژول REG ماژول کمکی بوده و برای جلوگیری از شلوغ شدن نمای اصلی مدار اضافه شده است. در ادامه به بررسی طراحی هر یک از این ماژول‌ها می‌پردازیم:

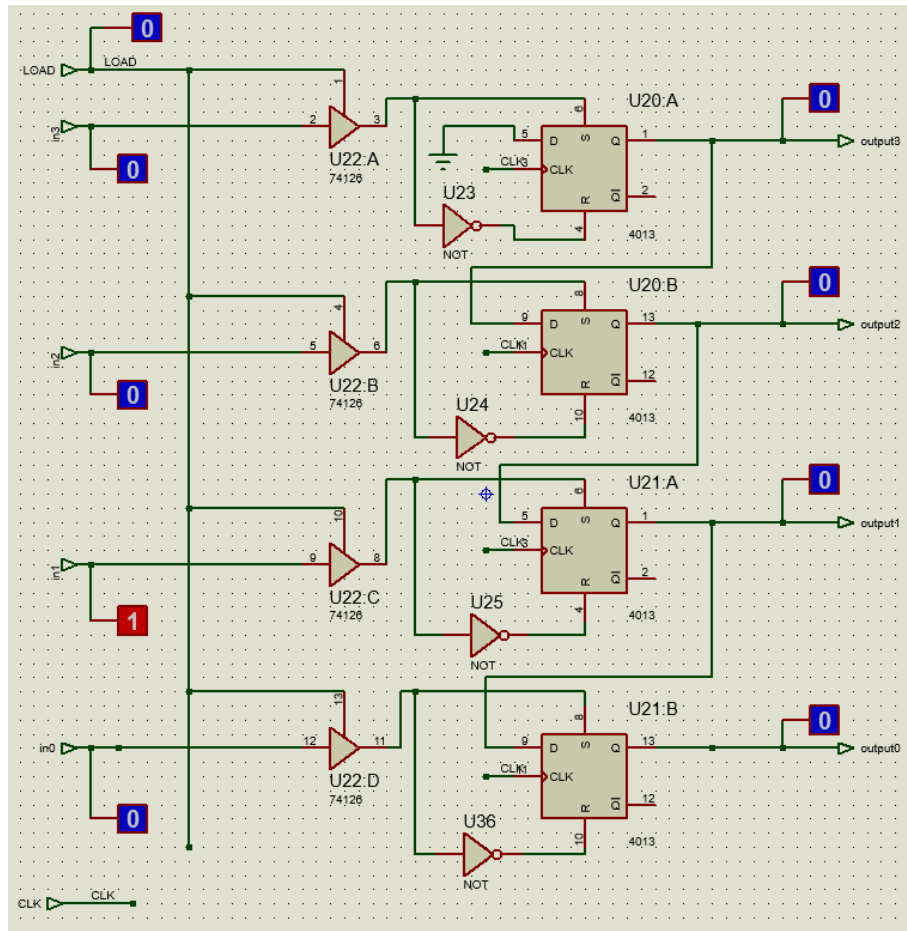
### ۱.۲ ماژول شیفت‌دهنده به سمت راست

همانطور که گفته شد این مدار برای شیفت دادن ضرب‌کننده در هر مرحله مورد استفاده قرار می‌گیرد. ورودی‌ها و خروجی‌های آن را در شکل ۲ می‌توان دید.



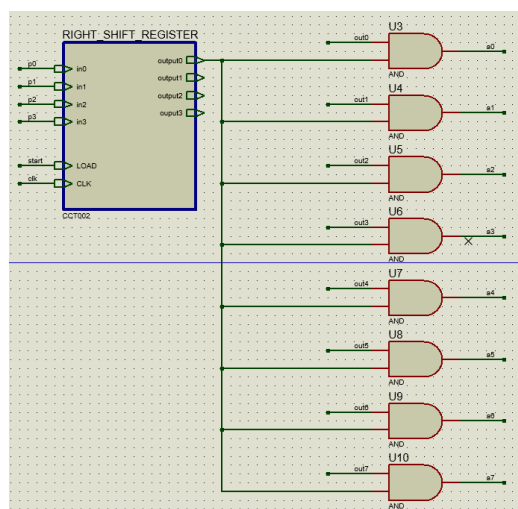
شکل ۲: ورودی‌ها و خروجی‌های ماژول شیفت‌دهنده به سمت راست

طراحی داخلی این ماژول نیز به کمک فلیپ‌فلاپ‌های نوع D صورت گرفته که خروجی هر یک به فلیپ‌فلاپ قبلی داده شده تا عملیات شیفت به راست با آمدن کلاک صورت بگیرد. همچنین سیگنال start به ورودی LOAD ماژول داده شده تا در صورت یک شدن این سیگنال، رجیسترها ریست شوند و مقدار صفر در آن‌ها قرار بگیرد. داده‌ی ورودی نیز با استفاده از تری‌استیت بافر به ورودی رجیسترها متصل شده تا در صورت یک بودن سیگنال LOAD، مقدار ورودی در خروجی ظاهر شود یا به عبارتی داده‌ی ورودی در رجیسترها بارگذاری شود. طراحی داخلی این ماژول را در شکل ۳ می‌توان مشاهده کرد.



شکل ۳: طراحی داخلی ماژول شیفت دهنده به سمت راست

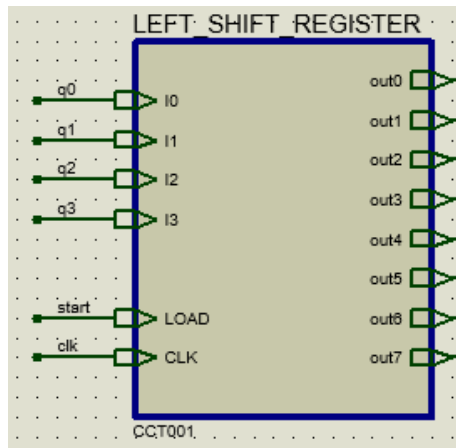
◁ همانطور که در الگوریتم Add & Shift توضیح داده شد، از بیت اول مقدار شیفت داده شده ضرب کننده برای این منظور که ضرب شونده را به حاصل اضافه کنیم یا خیر استفاده می شود، بنابراین بیت اول خروجی این ماژول را با بیت های ضرب شونده And می کنیم تا خواسته ی ما برآورده شود.



شکل ۴: تصمیم گیری برای اضافه کردن ضرب شونده

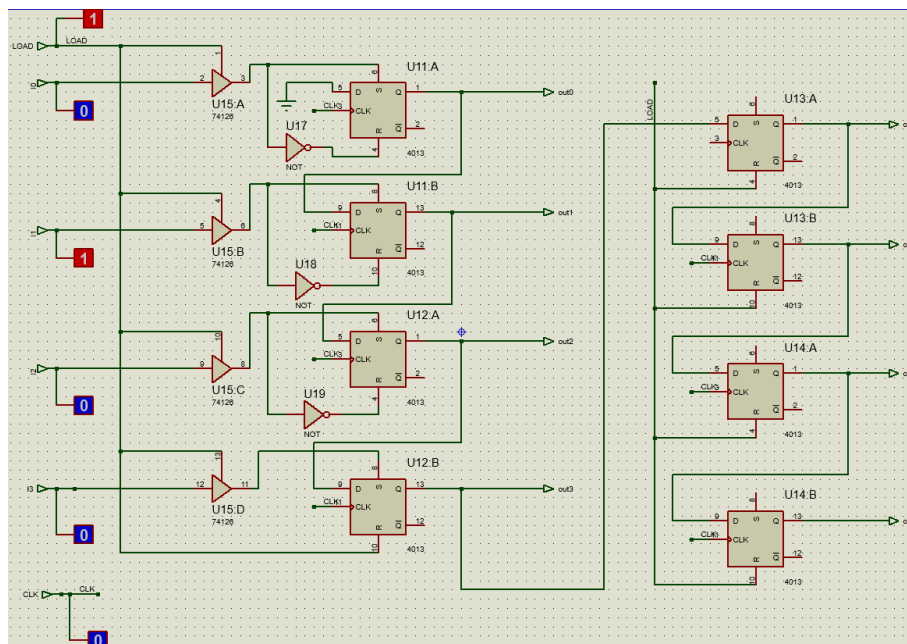
## ۲.۲ ماژول شیفت دهنده به سمت چپ

از این ماژول برای شیفت دادن ضرب شونده به سمت چپ استفاده می شود. رابط این ماژول را در شکل ۵ می توان دید.



شکل ۵: ماژول شیفت دهنده به سمت چپ

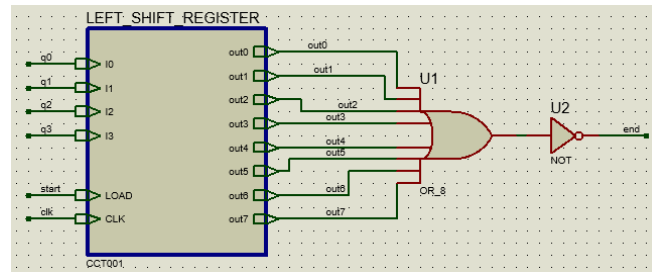
در طراحی داخلی این مدار نیز از فلیپ فلاپ های نوع D استفاده شده که مشابه ماژول قبلی خروجی هر فلیپ فلاپ به ورودی فلیپ فلاپ بعدی متصل شده تا عملیت شیفت به چپ صورت بگیرد. به طور مشابه سیگنال start به ورودی LOAD مدار متصل خواهد شد. با یک شدن این سیگنال مقدار ورودی در خروجی بارگذاری می شود. طراحی داخلی این ماژول را نیز می توان در شکل ۶ مشاهده کرد.



شکل ۶: طراحی داخلی ماژول شیفت دهنده به سمت چپ

◁ از آنجایی که شیفت به سمت چپ با اضافه شدن صفر به سمت راست عدد به وجود می آید، می توان

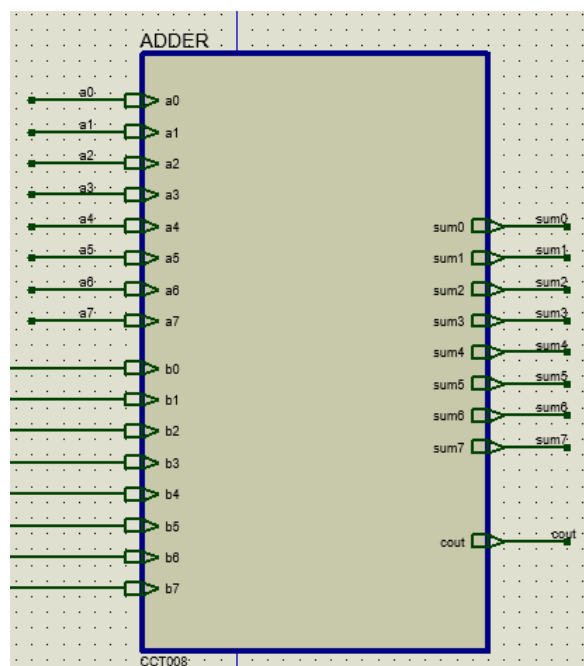
با صفر شدن همه‌ی بیت‌های مقدار حاصل از شیف، به این موضوع پی برد که آیا الگوریتم به پایان رسیده است یا خیر. بنابراین از همین موضوع برای مشخص کردن سیگنال خروجی end استفاده شده است. مدار این قسمت در شکل ۷ آورده شده است.



شکل ۷: مشخص کردن مقدار سیگنال خروجی end

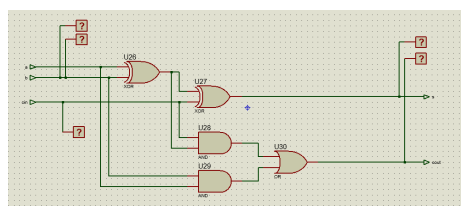
### ۳.۲ ماژول جمع کننده

این ماژول وظیفه جمع کردن مقدار حاصل تا این لحظه، با ضرب شونده را داراست. در واقع مقدار ضرب شونده در سیگال‌های  $a_i$  به این ماژول می‌رسد که این سیگنال‌ها همان and شده‌ی بیت‌های ضرب شونده با بیت اول ضرب کننده هستند. رابط این ماژول در شکل آمده است.

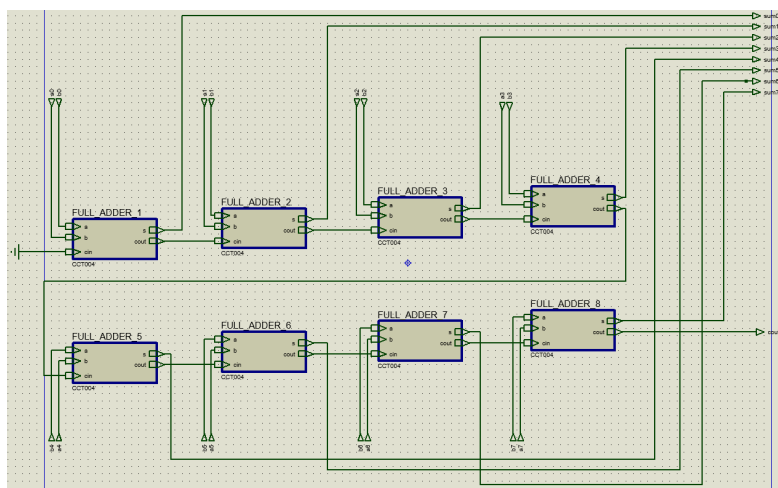


شکل ۸: ماژول جمع کننده

طراحی داخلی این جمع کننده نیز با استفاده از هشت Full Adder صورت گرفته که طراحی داخلی این ماژول و ماژول Full Adder را در شکل ۹ می‌توان دید.



(ب) طراحی داخلی ماژول Full Adder

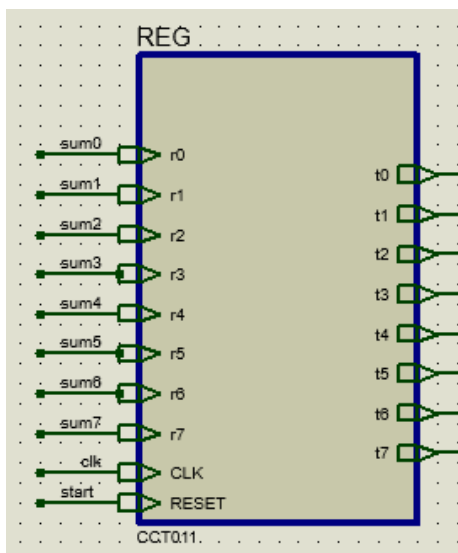


(آ) طراحی داخلی ماژول جمع‌کننده

شکل ۹: طراحی داخلی Adder و FullAdder

## ۴.۲ ماژول REG

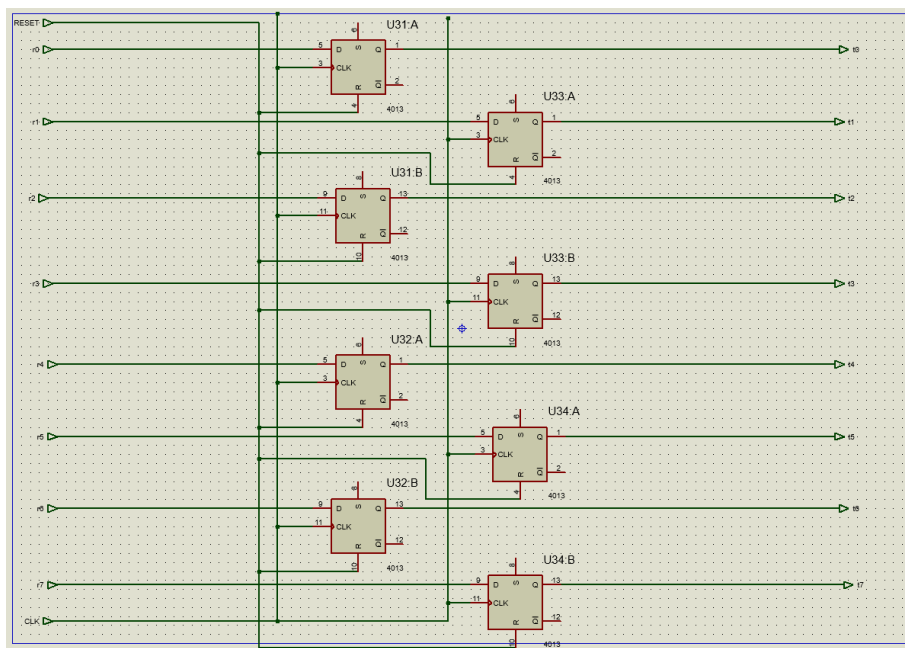
همانطور که گفته شد این ماژول به عنوان رجیستر برای نگه‌داری مقدار حاصل ضرب در هر مرحله استفاده می‌شود. خروجی‌های این ماژول به ماژول جمع‌کننده داده شده تا مقدار حاصل ضرب آپدیت شود. ورودی و خروجی‌های این ماژول در شکل ۱۰ آورده شده است.



شکل ۱۰: ماژول REG

در مدار داخلی آن نیز از هشت فلیپ‌فلاپ نوع D استفاده شده است که طراحی آن را در شکل ۱۱ می‌توان دید.

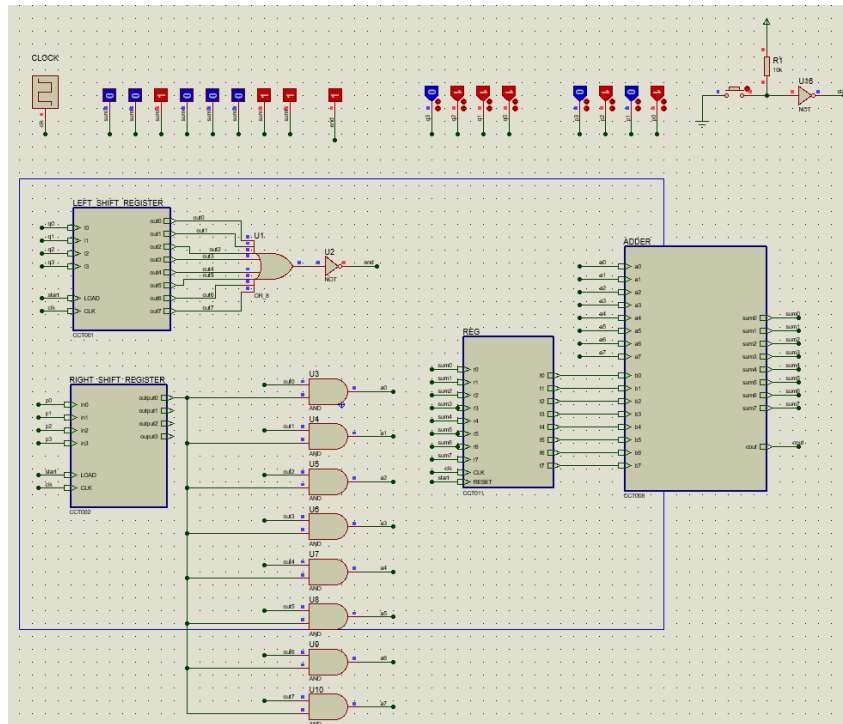




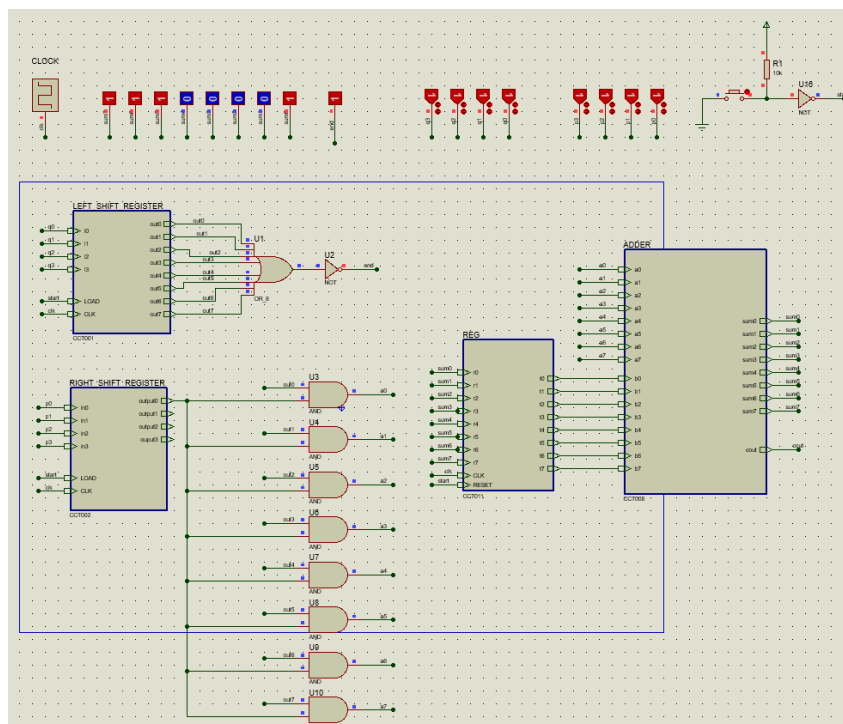
شکل ۱۱: طراحی داخلی ماژول REG

## ۳ تست مدار

برای تست کارکرد مدار نیز چند نمونه ورودی داده شده و حاصل ضرب آن‌ها توسط مدار محاسبه شده است. نمونه‌ای از تست‌ها را در شکل‌های زیر می‌توان دید.



شکل ۱۲:  $7 \times 5 = 35$



شکل ۱۳:  $15 \times 15 = 225$

