

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

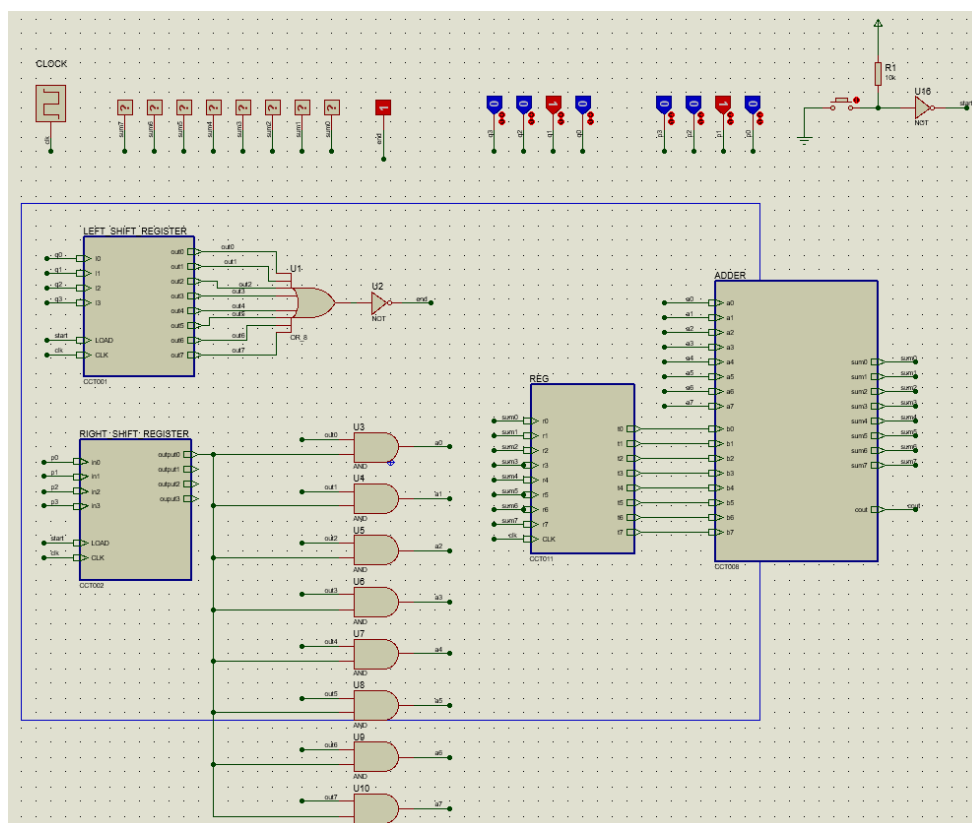
|   |                                          |
|---|------------------------------------------|
| ۲ | ۱ هدف آزمایش                             |
| ۳ | ۲ مراحل طراحی و پیاده‌سازی مدار          |
| ۳ | ۱.۲ مازول شیفتهنده به سمت راست . . . . . |
| ۴ | ۲.۲ مازول شیفتهنده به سمت چپ . . . . .   |
| ۵ | ۳.۲ مازول جمع‌کننده . . . . .            |
| ۶ | ۴.۲ مازول REG . . . . .                  |
| ۸ | 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.



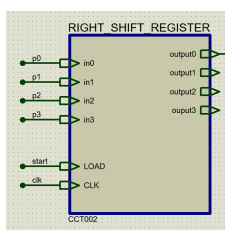
شکل ۱: نمای کلی مدار ضرب‌کننده

## ۲ مراحل طراحی و پیاده سازی مدار

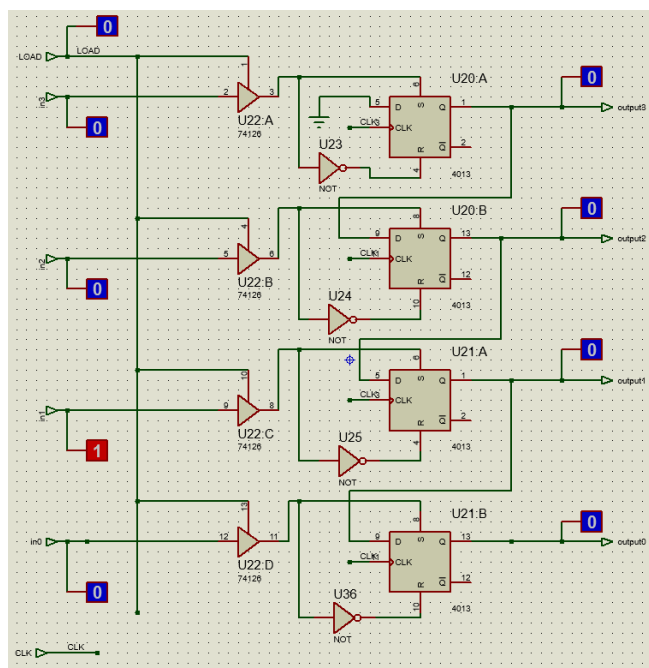
همانطور که در بخش قبل گفته شد، مدار مورد نظر از سه ماژول اصلی شیفت دهنده به سمت راست، شیفت دهنده به سمت چپ و جمع کننده هشت بیتی تشکیل شده است. همچنین ماژول REG ماژول کمکی بوده و برای جلوگیری از شلوغ شدن نمای اصلی مدار اضافه شده است. در ادامه به بررسی طراحی هر یک از این ماژول ها می پردازیم:

### ۱.۲ ماژول شیفت دهنده به سمت راست

همانطور که گفته شد این مدار برای شیفت دادن ضرب کننده در هر مرحله مورد استفاده قرار می گیرد. ورودی ها و خروجی های آن را در شکل ۲ می توان دید. طراحی داخلی این ماژول نیز به کمک فلیپ فلاپ های نوع D صورت گرفته که خروجی هر یک به صورت سریال به فلیپ فلاپ بعدی داده شده تا عملیات شیفت با آمدن کلاک صورت بگیرد. طراحی داخلی این ماژول را در شکل ۳ می توان مشاهده کرد.

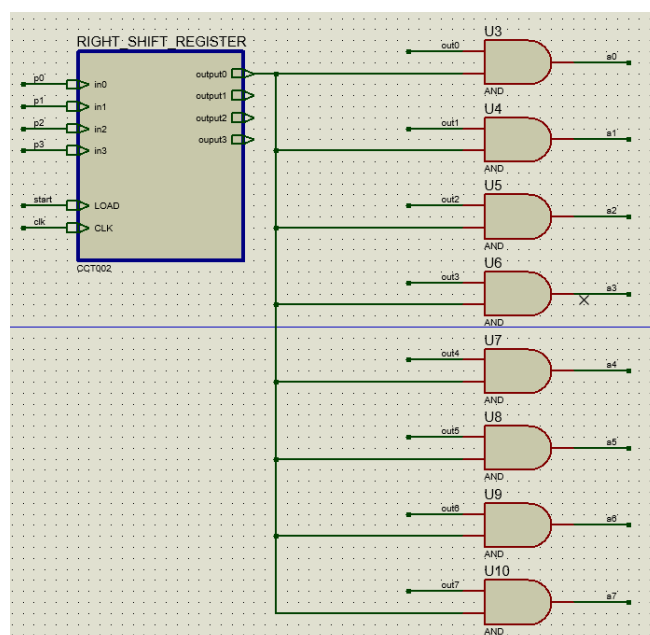


شکل ۲: ورودی ها و خروجی های ماژول شیفت دهنده به سمت راست



شکل ۳: طراحی داخلی ماژول شیفت دهنده به سمت راست

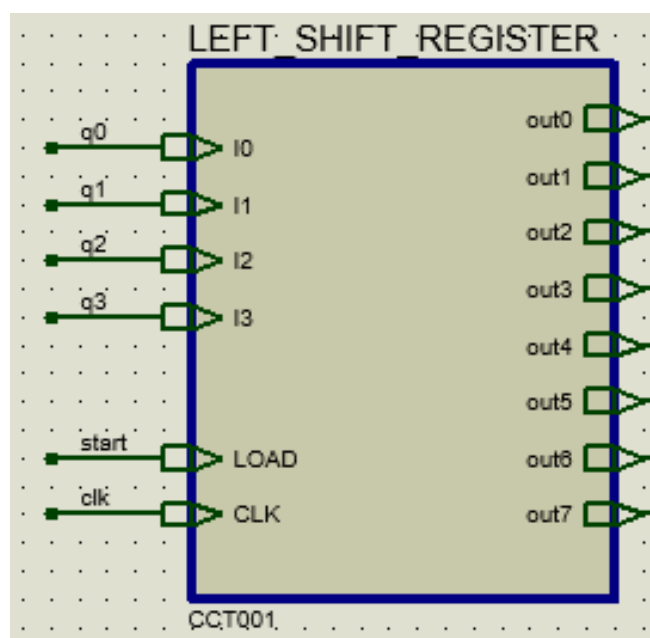
◀ همانطور که در الگوریتم Add & Shift توضیح داده شد، از بیت اول مقدار شیفت داده شده ضرب کننده برای این منظور که ضرب شوند را به حاصل اضافه کنیم یا خیر استفاده می شود، بنابراین بیت اول خروجی این ماژول را با بیت های ضرب شوند And می کنیم تا خواسته ی ما برآورده شود.



شکل ۴: تصمیم‌گیری برای اضافه کردن ضرب‌شونده

## ۲.۲ ماژول شیفت‌دهنده به سمت چپ

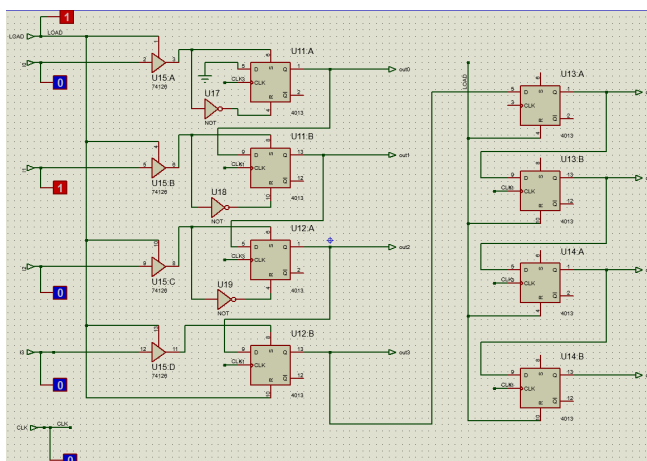
از این ماژول برای شیفت دادن ضرب‌شونده به سمت چپ استفاده می‌شود. رابط این ماژول را در شکل ۵ می‌توان دید.



شکل ۵: ماژول شیفت‌دهنده به سمت چپ

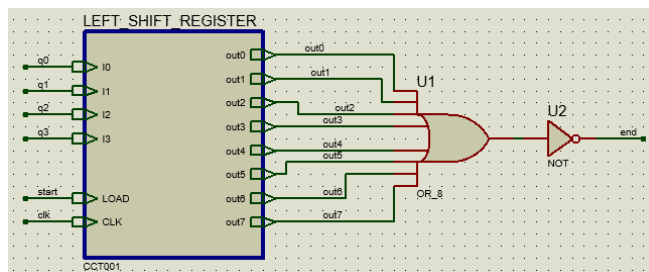
در طراحی داخلی این مدار نیز از فلیپ‌فلاپ‌های نوع D استفاده شده که مشابه ماژول قبلی خروجی هر فلیپ‌فلاپ به ورودی فلیپ‌فلاپ قبلی متصل شده تا عملیت شیفت به چپ صورت بگیرد. طراحی

داخلی این ماژول را نیز می‌توان در شکل ۶ مشاهده کرد.



شکل ۶: طراحی داخلی ماژول شیفت‌دهنده به سمت چپ

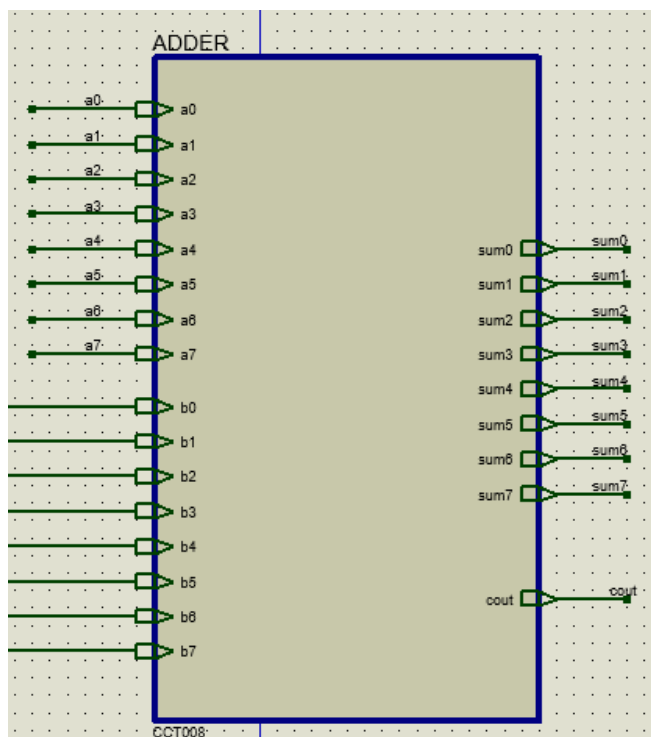
از آنجایی که شیفت به سمت چپ با اضافه شدن صفر به سمت راست عدد به وجود می‌آید، می‌توان با صفر شدن همه‌ی بیت‌های مقدار حاصل از شیفت، به این موضوع پی برد که آیا الگوریتم به پایان رسیده است یا خیر. بنابراین از همین موضوع برای مشخص کردن سیگنال خروجی end استفاده شده است. مدار این قسمت در شکل ۷ آورده شده است.



شکل ۷: مشخص کردن مقدار سیگنال خروجی end

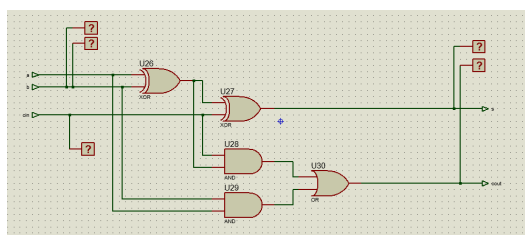
### ۳.۲ ماژول جمع کننده

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

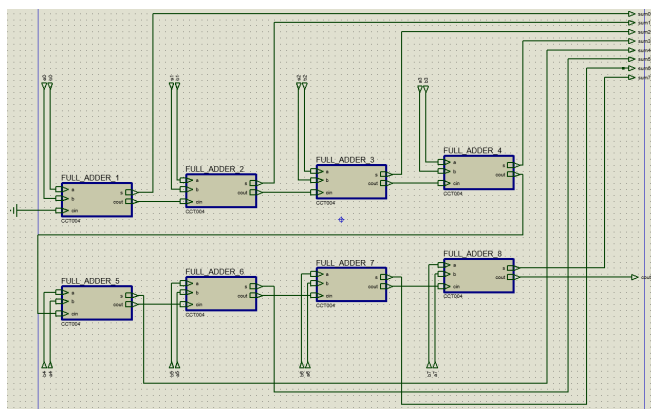


شکل ۸: ماژول جمع‌کننده

طراحی داخلی این جمع‌کننده 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

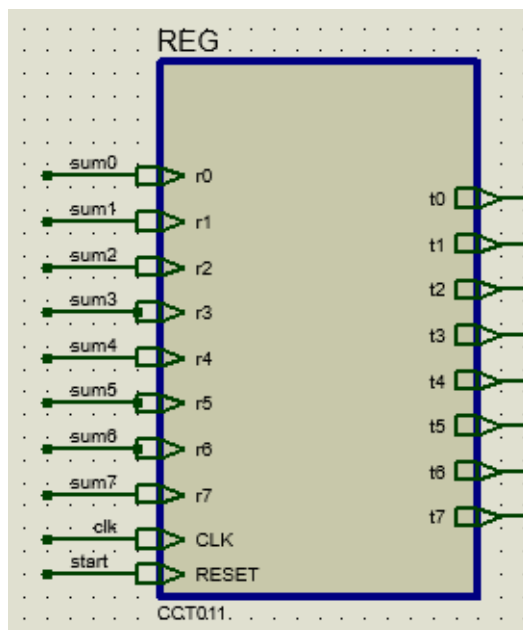


(آ) طراحی داخلی ماژول جمع‌کننده

شکل ۹: طراحی داخلی Adder و FullAdder

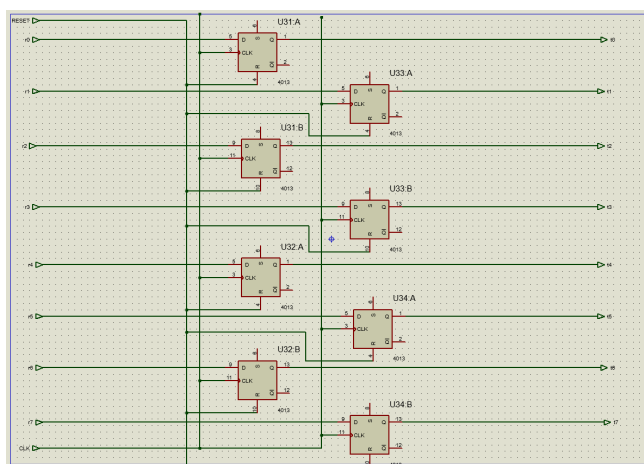
## ۴.۲ ماژول REG

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 فلیپ فلاپ نوع D استفاده شده است که طراحی آن را در شکل ۱۱ می توان دید.

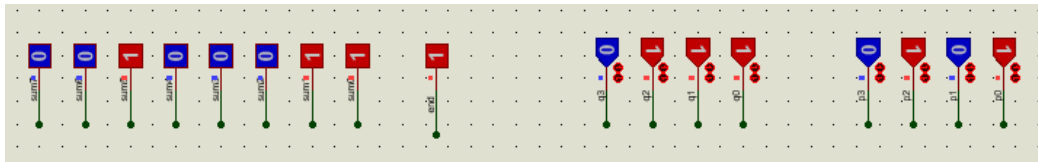
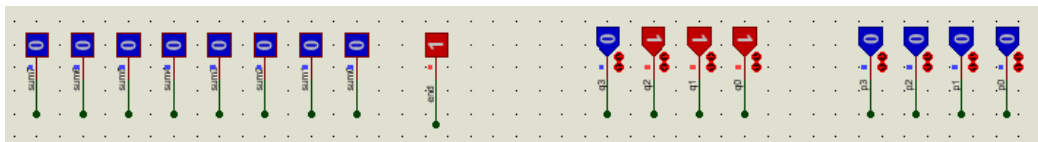


شکل ۱۱: طراحی داخلی ماژول REG



## 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:  $15 \times 15 = 225$ شکل ۱۴:  $7 \times 0 = 0$ شکل ۱۵:  $15 \times 1 = 15$