

به نام خدا



آزمایشگاه معماری

آزمایش سوم: طراحی ضرب کننده دو عدد ۴ بیتی

اعضای گروه:

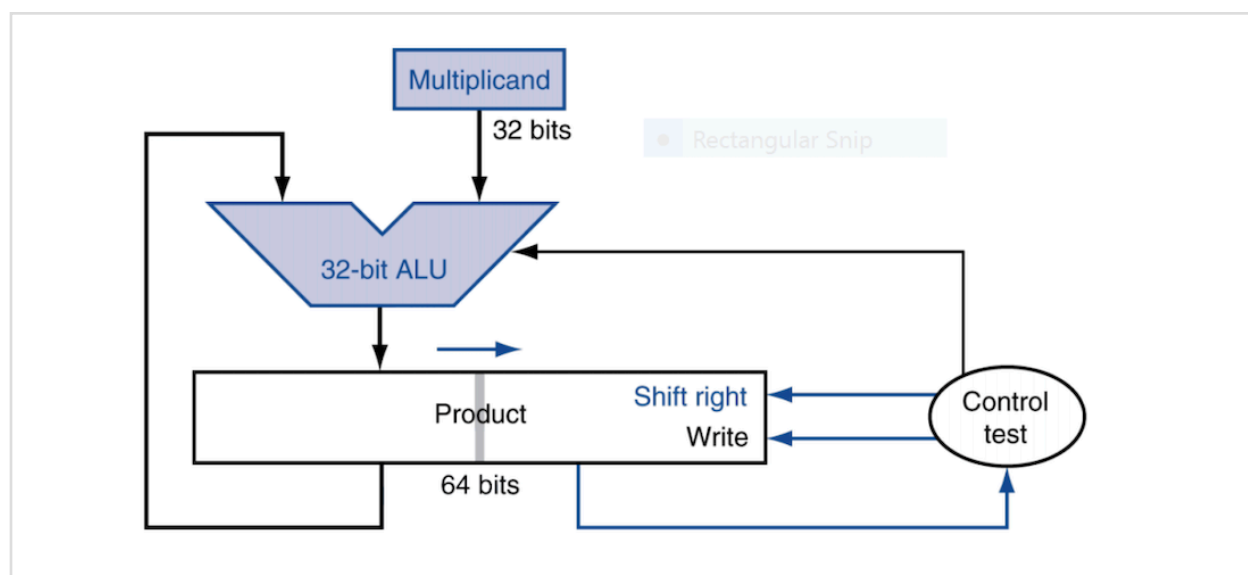
امیراردلان دهقانپور 401105901

رادین شاهدایی 401106096

باربد شهرآبادی 401106125

**مقدمه:**

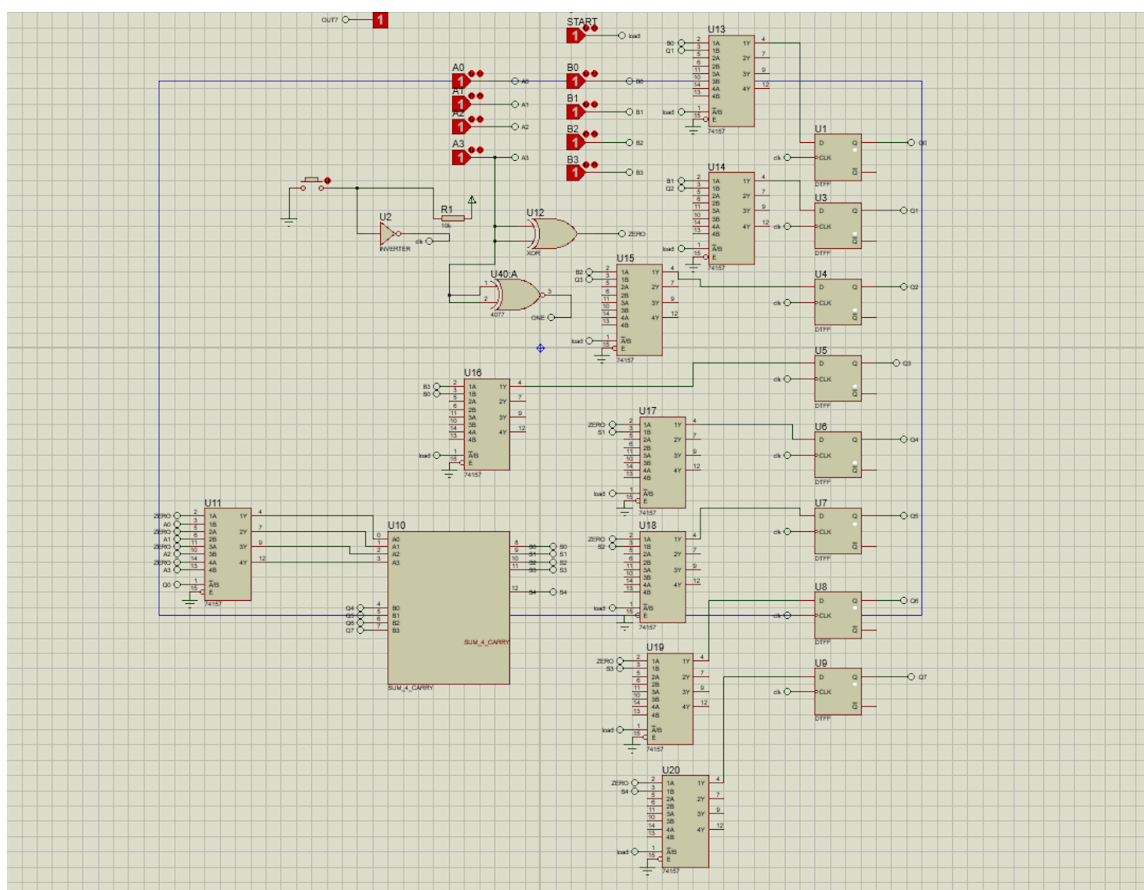
در این آزمایش قصد داریم که يك ضرب کننده برای دو عدد ۴ بیتی را به روش add & shift بسازیم. الگوریتمی که برای ضرب کردن دو عدد در این سوال از آن استفاده کرده ایم تا حدودی مشابه با همان روش کاغذ و قلمی است که در حالت عادی و روی کاغذ نیز از آن استفاده می کنیم، این الگوریتم البته بهینه تر از حالت عادی این الگوریتم است زیرا در الگوریتم عادی از ALU ۸ بیتی برای جمع کردن باید استفاده می کردیم اما در اینجا از ALU ۴ بیتی برای جمع استفاده می کنیم، به صورت کلی الگوریتم مشابه با همان الگوریتم برای اعداد ۳۲ بیتی است تنها با این تفاوت که در اینجا اعداد ۴ بیتی هستند، این الگوریتم برای اعداد ۳۲ بیتی به صورت زیر اس



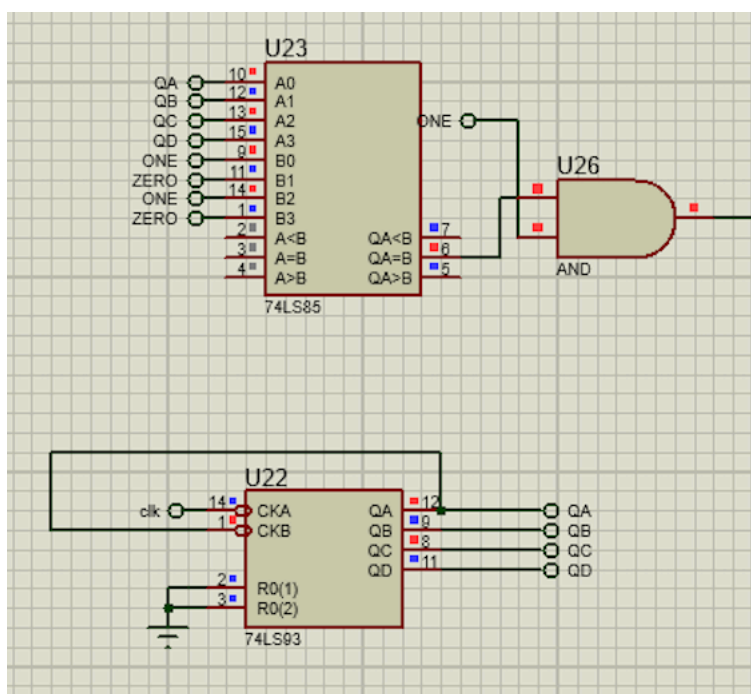
شکل ۱

همانطور که مشخص است در این الگوریتم یکی از دو عدد را در رجیستر Multiplicand قرار می دهیم و عدد دیگر را نیز در ابتدا در ۴ بیت سمت راست رجیستر Product قرار می دهیم و سپس بعد از آن در هر مرحله این رجیستر را يك بیت به سمت چپ شیفت می دهیم و ۴ بیت سمت چپ آن را با همان رجیستر Multiplicand در هر مرحله جمع می کنیم و حاصل را در این ۴ بیت قرار می دهیم، واحد Control نیز کاربردی که دارد این است که اگر بیت سمت راستی که خارج کرده ایم يك باشد جمع را انجام دهد و

در غیر این صورت جمع را انجام ندهد، حال با توجه به این الگوریتم متوجه می شویم که به چهار کلاک برای محاسبه حاصل ضرب دو عدد ۴ بیتی به کمک این الگوریتم نیاز داریم. برای پیاده سازی این مدار در پروتئوس، نیاز داریم که یک رجیستر ۸ بیتی برای Product داشته باشیم که این کار را به کمک کنار هم قرار دادن ۸ d-flip flop انجام می دهیم، همچنین یک واحد جمع کننده ۴ بیتی نیز می سازیم که همان کار واحد ALU را انجام می دهد و همچنین برای بخش Control نیز به یک مالتی پلکسر نیاز داریم تا به کمک بیت سمت راستی که خارج می شود بتوانیم انتخاب کنیم که چه چیزی به عنوان ۴ بیت سمت راست رجیستر Product شود، همچنین در صورت سوال گفته شده است که در هنگامی که کار تمام می شود و حاصل جمع محاسبه می شود باید سیگنال end برابر با یک شود، برای این کار نیز یک شمارنده گذاشته ایم که با هر بار کلاک خوردن مدار یک واحد افزایش پیدا می کند و هر هنگامی که برابر با ۵ شد، خروجی را نمایش می دهیم زیرا به ۴ کلاک برای محاسبه حاصل ضرب نیاز است و همچنین در ابتدای کار نیز به یک کلاک نیاز داریم تا عدد دوم را در ۴ بیت سمت راست رجیستر Product قرار بدهیم. پس می توانیم مدار ضرب کننده را به صورت زیر نمایش دهیم:

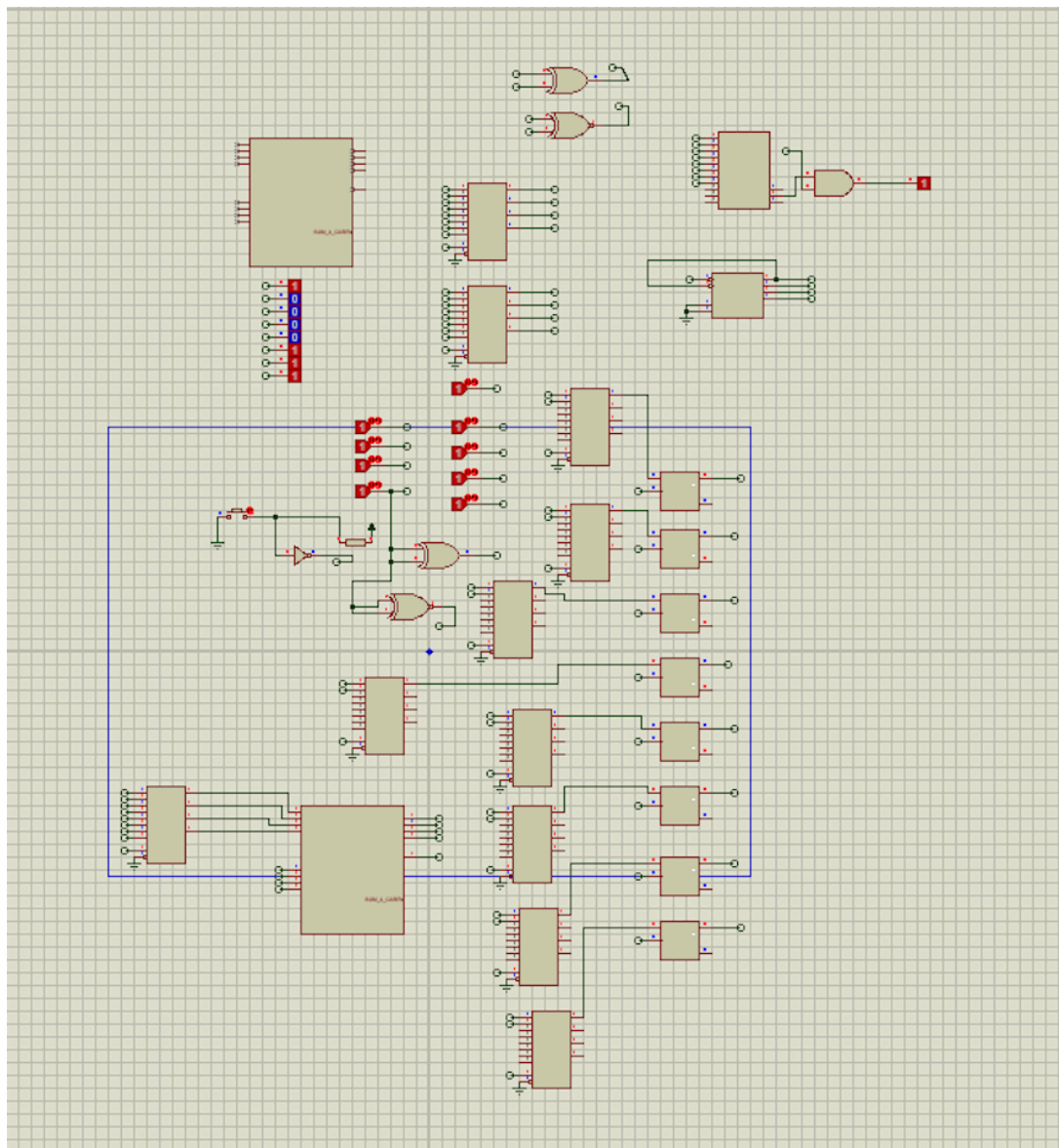


همانطور که در مدار مشخص است در ابتدا به کمک سیگنال load که به start متصل است مشخص کنیم که این سیگنال فعال باشد یا خیر، در حالت ابتدایی باید غیرفعال و صفر باشد زیرا در ابتدای امر می دانیم که باید در ۴ بیت سمت راست عدد B قرار بگیرد و در ۴ بیت سمت راست صفر قرار بگیرد، سپس بعد از آنکه سیگنال load برابر با یک شد، در مراحل بعدی در هر مرحله یک بیت شیفต์ به راست داریم که در مدار نیز مشخص است که Q1, Q2, Q3 یک بیت به راست رفته اند و در ۵ بیت دیگر حاصل جمع قرار داده شده است که شامل بیت های S0 تا S4 می باشد، روند این بخش نیز به این صورت است که با توجه به بیت Q0 که همان بیت سمت راست خارج شده از مدار است در مالتی پلکسر تصمیم گیری می کنیم که از ۴ بیت صفر و یا ۴ بیت عدد A استفاده کنیم سپس این خروجی با ۴ بیت سمت چپ رجیستر Product جمع می شود که در اینجا برای مثال اگر بیت Q0 برابر با صفر بود، در این جمع صفر با این ۴ بیت جمع می شد و صرفاً باعث می شد که یک بیت شیفต์ به راست داشته باشیم و صفر وارد می شد، پس نحوه کارکرد مدار ما به صورت کلی مشابه با روندی است که توضیح داده ایم و الگوریتم ضربی که در کتاب مانو آورده شده است، بخش شمارنده که به کمک آن می توانیم سیگنال end را آماده کنیم نیز به این صورت می باشد:



شکل ۳

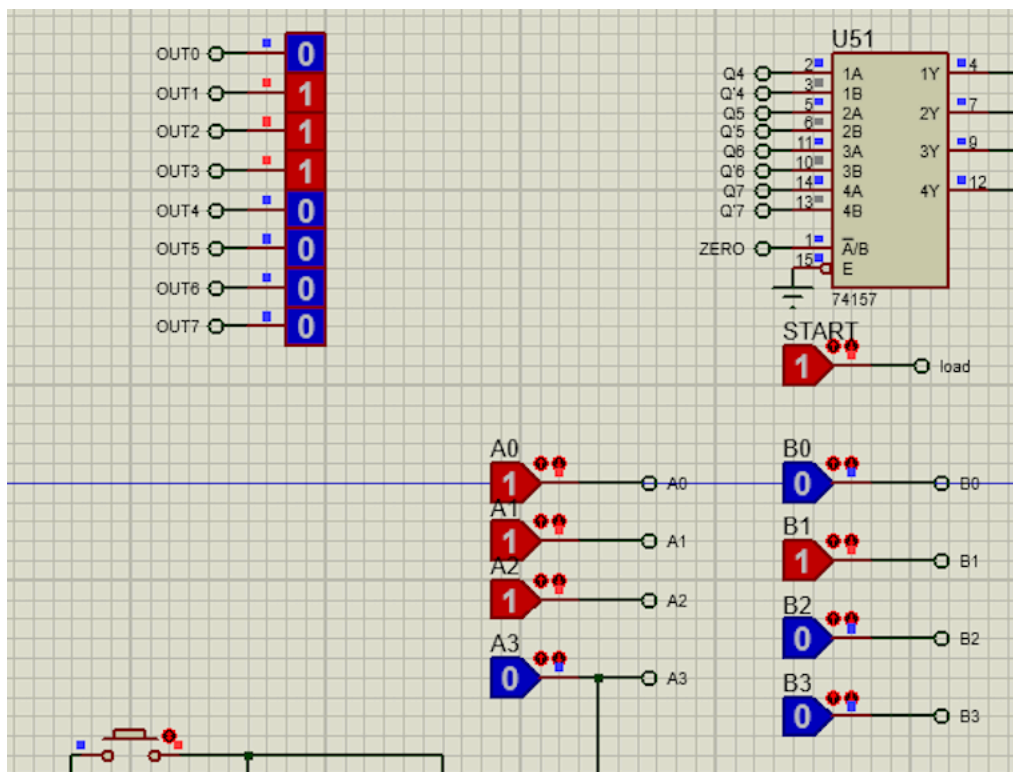
در این بخش نیز مشاهده می‌کنیم که از IC 74LS93 برای شمارش استفاده کرده ایم که به آن سیگنال کلاک متصل است همچنین از يك مقایسه‌کننده استفاده کرده ایم که زمانی که این عدد برابر با ۵ شد سیگنال آن برابر با يك شود و بتوانیم نشان دهیم که عملیات پایان یافته است، پس در نهایت مدار ما آماده شده است و شمای کلی آن به صورت شکل زیر می‌باشد در ادامه چند مثال نیز از کارکرد آن آورده می‌شود.



شکل ۴

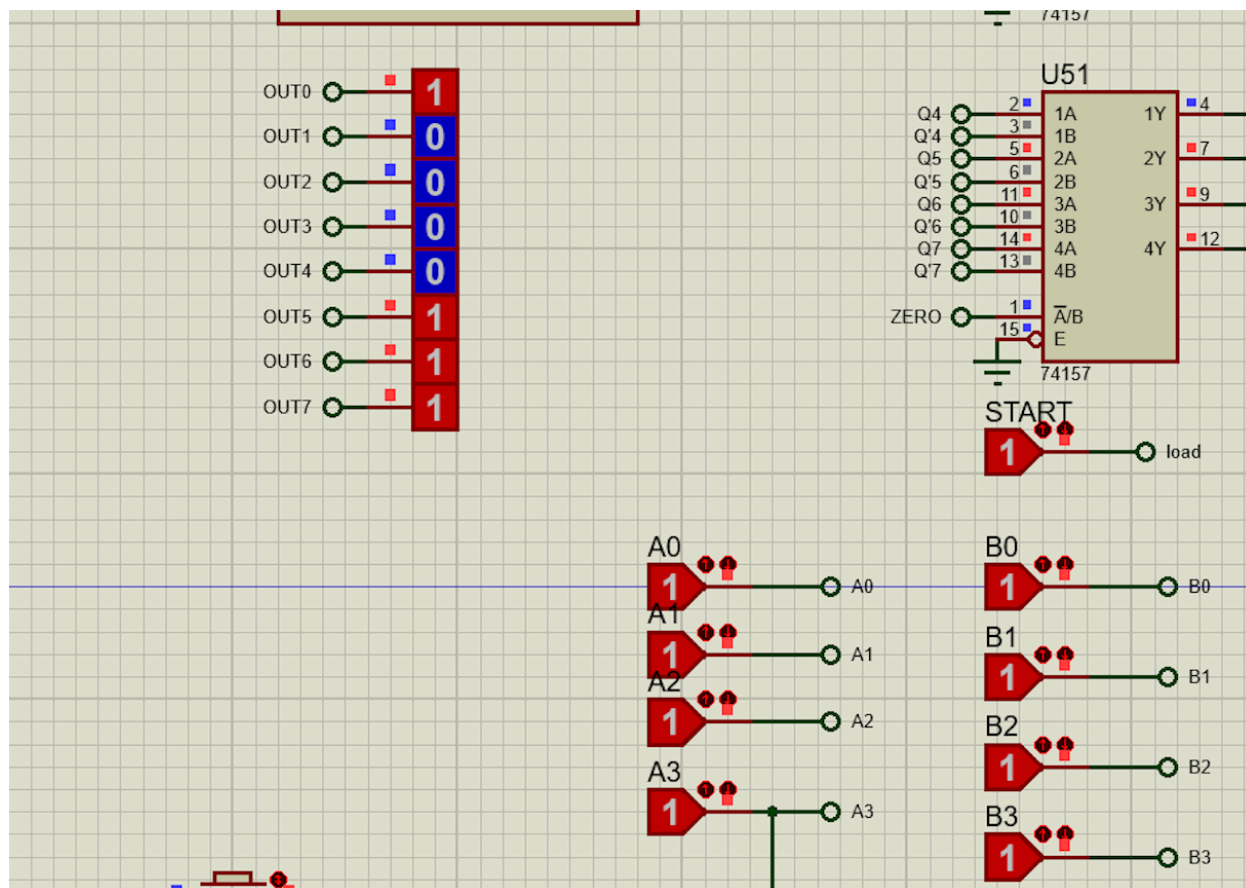
حال چند مثال از ورودی و خروجی ها را مشاهده خواهیم کرد:

- ورودی A برابر با ۷ و ورودی B برابر با ۲ بوده است و حاصل برابر با ۱۴ شده است.



شکل ۵

- ورودی A برابر با ۱۵ و B نیز برابر با ۱۵ بوده است و حاصل برابر با ۲۲۵ است.



شکل ۶

### منابع مورد استفاده:

- کتاب معماری کامپیوتر موریس مانو