

به نام خدا



آزمایشگاه معماری کامپیوتر

گزارش کار دوم

ضرب کننده ممیز ثابت

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

پاییز ۱۴۰۱

استاد:

حمید سربازی آزاد

دستیار آموزشی:

عطیه یونسی

نویسندگان:

احمد رضا خناری

نگار عسکری

علی نظری

فهرست

مقدمه

۲

گزارش آزمایش

۳

بخش ۱. شمای کلی طراحی

۳

بخش ۲. مراحل طراحی

۴

نتیجه گیری

۷

مقدمه

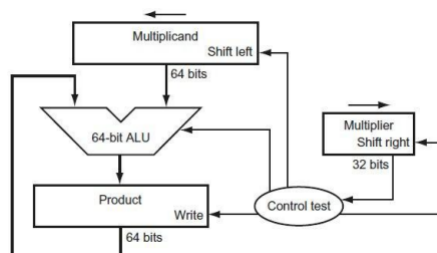
در این آزمایش، ما یک مدار ضرب کننده ی ۴ بیت در ۴ بیت با روش add and shift با استفاده از نرم افزار Proteus طراحی و شبیه سازی کردیم. این مدار به این شکل کار می کند که دو عدد ۴ بیتی و یک سیگنال start و همچنین یک سیگنال clock به مدار ورودی داده می شود. با یک شدن start عملیات ضرب شروع می شود و پس از ۴ سیکل ساعت عملیات ضرب کامل می شود و حاصل به همراه سیگنال end خروجی داده می شود.

الگوریتم مورد استفاده، الگوریتم add and shift است. یک shift register ۸ بیتی را به عنوان حاصل در نظر می گیریم که در ابتدا به صفر reset شده است. همچنین مضروب را در شیفتر رجیستر ۸ بیتی می گذاریم و مضروب فیه را در یک شیفتر رجیستر ۴ بیتی قرار می دهیم. سپس در هر مرحله رقم یکان مضروب فیه را نگاه می کنیم. اگر یکان یک باشد، مقدار حاصل ضرب تا کنون محاسبه شده را با مقدار موجود در رجیستر مضروب جمع می کنیم. سپس در هر سیکل ساعت مضروب را یکبار به چپ شیفتر می دهیم و مضروب فیه را یکبار به راست شیفتر می دهیم.

گزارش آزمایش

بخش ۱. شمای کلی طراحی

حالت کلی طراحی به شکل زیر است:

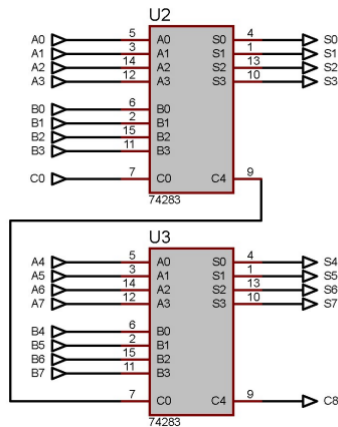


شکل ۱: Main Diagram

همانطور که مشخص است، از ۳ شیفت رجیستر استفاده شده است. یک واحد جمع کننده ۸ بیتی هم لازم است. یک واحد کنترل هم که لازم است. سیگنال های مربوط به شیفت دادن رجیستر ها و همچنین نوشتن حاصل جمع در رجیستر حاصل ضرب در واحد کنترلی تولید می شود.

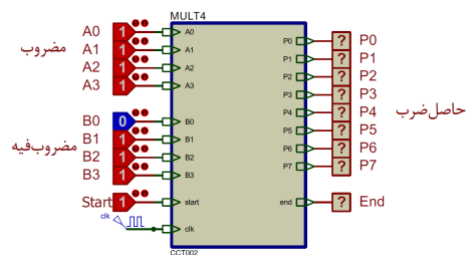
بخش ۲. مراحل طراحی

برای طراحی نخست به جمع کننده ۸ بیتی نیاز داریم. با کمک دو جمع کننده ۴ بیتی به سادگی می توانیم آن را بسازیم. پس یک sub circuit می سازیم و carry out اولی را به عنوان carry in دومی در نظر می گیریم.



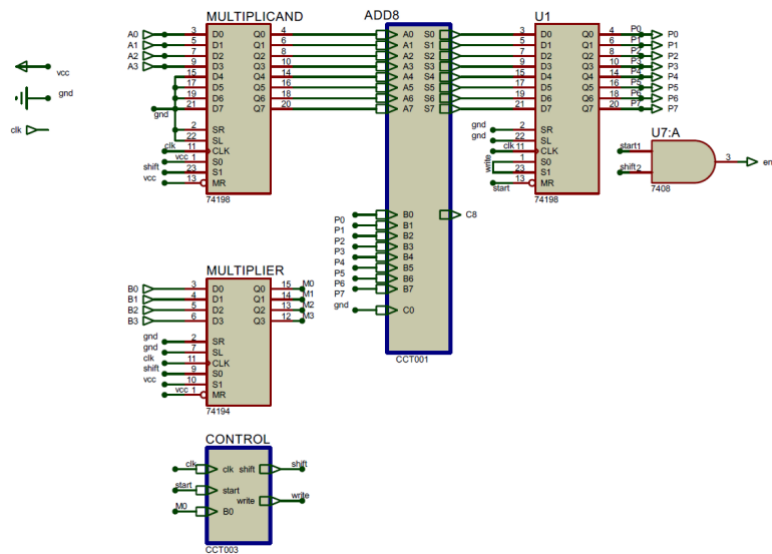
شکل ۲: 8 bit adder

حال به طراحی کلی مدار ضرب کننده می پردازیم. این ضرب کننده به شکل زیر است:



شکل ۳: multiplier

داخل این ماژول به شکل زیر است:



شکل ۴: multiplier

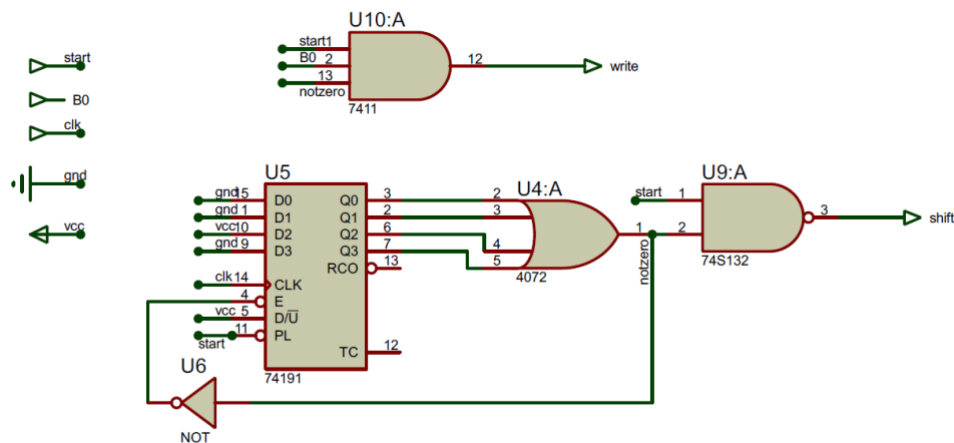
برای شیفت رجیستر های ۸ بیتی از آییسی ۷۴۱۹۸ استفاده می شود و برای شیفت رجیستر های ۴ بیتی، از ۷۴۱۹۴. داده های ورودی هم که درون مضروب و مضروب فیه بارگذاری می شود. پس به پایه های S_0 و S_1 باید مقدار یک ورودی داده شود. همچنین به پایه ی مربوط به reset که active low است، مقدار یک را ورودی می دهیم. به پایه های غیر لازم هم مقدار صفر را می دهیم.

همچنین در صورتی که بخواهیم مضروب را به راست شیفت بدهیم، باید پایه S_0 را صفر کنیم و پایه S_1 را یک کنیم. پس سیگنال شیفت را در واحد کنترل به صورتی طراحی می کنیم که ۰ به معنای انجام شیفت باشد و یک به معنای عدم انجام آن. و این ها را به پایه های مضروب و مضروب فیه وصل می کنیم.

رجیستر حاصل ضرب هم در زمانی که سیگنال write فعال است، حاصل جمع خروجی را از ALU می گیرد و درون خودش ذخیره می کند. پس دو ورودی S_0 و S_1 را برابر با سیگنال write خروجی از CU قرار می دهیم. همچنین reset که active low است را برابر با سیگنال start قرار می دهیم تا در زمان شروع ضرب این رجیستر با صفر پر شود.

سیگنال end نیز برابر با حاصل and دو سیگنال start و shift است. یعنی زمانی که عملیات ضرب در حال انجام است و دیگر نیازی نیست شیفت انجام دهیم.

پس بخش CU مانند شکل زیر می شود.



شکل ۵: Control Unit

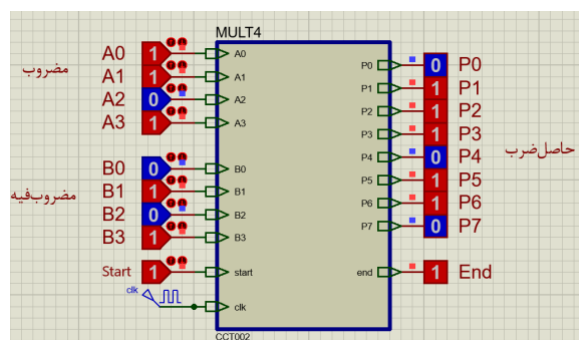
این ماژول رقم یکان رجیستر multiplier را می گیرد و در صورتی که یک باشد، باید حاصل جمع در رجیستر product ریخته شود. همچنین باید سیگنال start نیز یک باشد و شمارنده صفر نباشد. یعنی عملیات ضرب در حال انجام باشد. پس سیگنال write برابر با and سه سیگنال notzero و start و B₀ است. همچنین در این ماژول یک شمارنده ۷۴۱۹۱ قرار می دهیم که به حالت شمارش نزولی تنظیم شده است. در صورتی که start برابر با صفر باشد، این شمارنده روی عدد ۴ تنظیم می شود. سپس با هر سیکل ساعت، یک واحد از آن کم می شود تا زمانی که به صفر برسد. پس از آن باید سیگنال enable آن که active low است، به یک تبدیل شود. به این منظور، خروجی های شمارنده را با هم OR کرده و not آن را به ورودی enable شمارنده می دهیم. همچنین سیگنال شیف، باید در زمان انجام عمل ضرب، برابر با یک باشد. یعنی در هر سیکل ساعت، باید مضروب و مضروب فیه را شیف دهیم. بنابراین سیگنال شیف برابر با NAND سیگنال start و حاصل OR خروجی شمارنده است. چون می خواهیم shift ما active low باشد.

نتیجه گیری

به عنوان بخش پایانی، مدار را روی چند حالت تست می کنیم.

$$11 \times 10 = 110$$

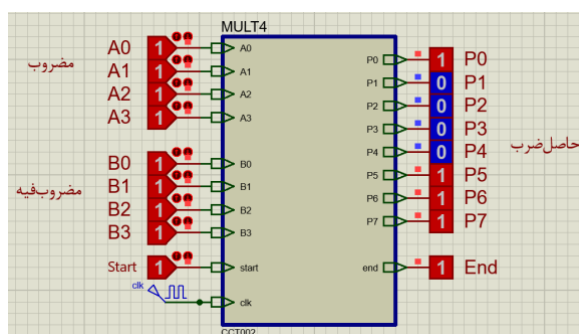
$$1011 \times 1010 = 01101110$$



شکل ۶: multiplier

$$15 \times 15 = 225$$

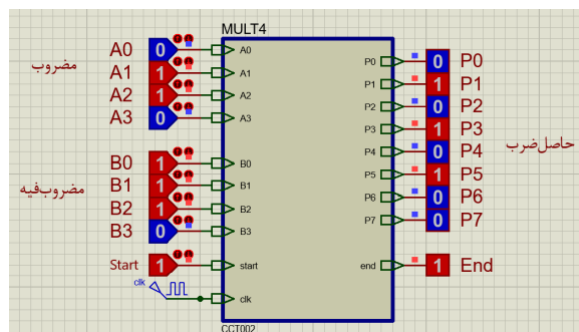
$$1111 \times 1111 = 11100001$$



شکل ۷: multiplier

$$6 \times 7 = 42$$

$$0110 \times 0111 = 00101010$$



شکل ۸: multiplier