

آزمایش اول: طراحی جمع کننده BCD

ایمان محمدی ۹۹۱۰۲۲۰۷

علی پاشا منتصری ۹۹۱۰۹۱۸۸

مهدی قائم پناه ۹۹۱۰۹۱۹۹

چکیده

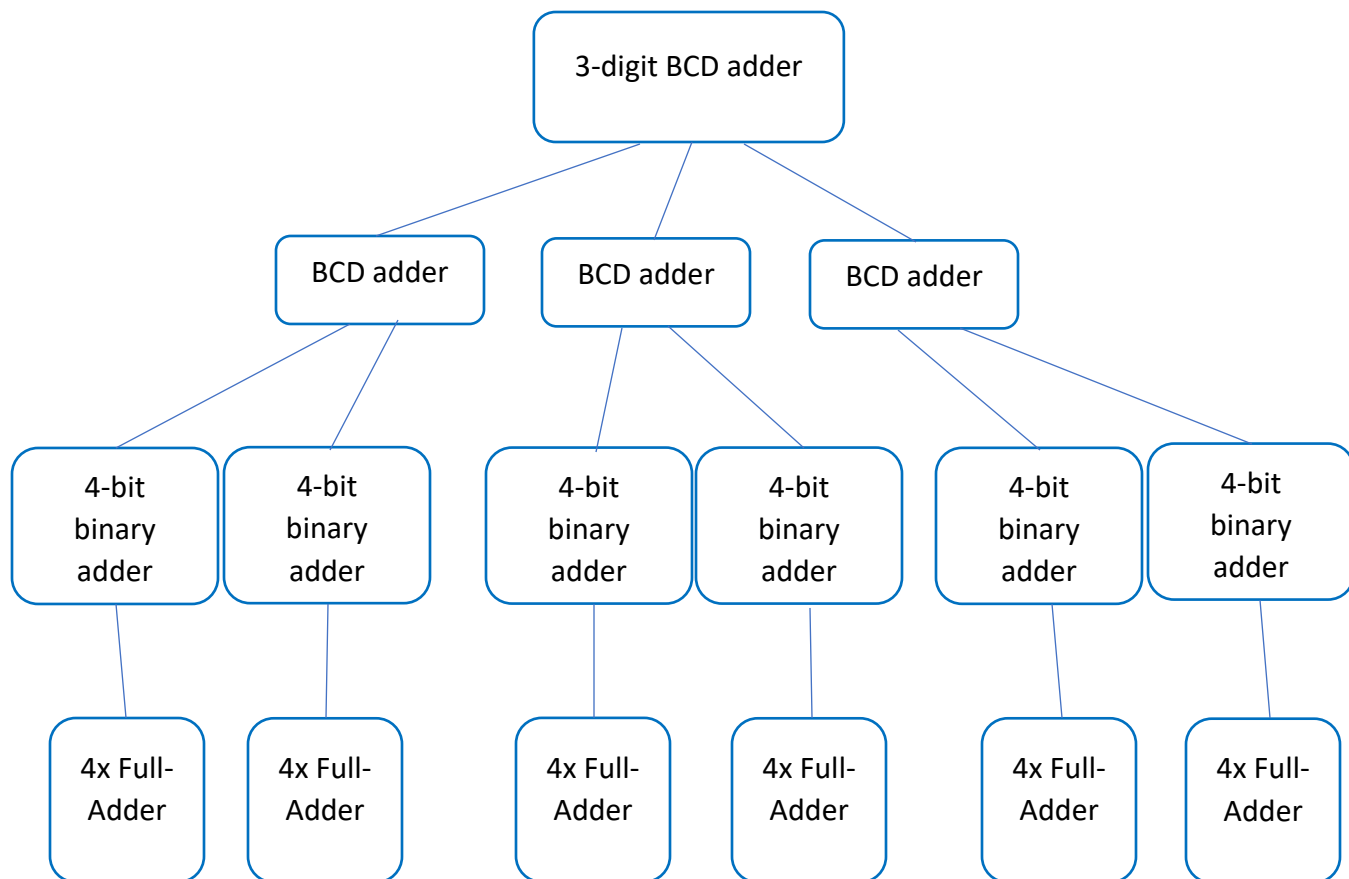
در این آزمایش، بصورت سلسله‌مراتبی و از پایین به بالا، یک جمع کننده دهدهی سه رقمی پیاده می‌کنیم. در ابتدا Full-Adder با استفاده از تراشه‌های گیت‌های پایه پیاده شده و سپس جمع کننده باینری ۴ بیتی و در ادامه جمع کننده دو عدد BCD پیاده می‌شود. در نهایت جمع کننده سه رقمی پیاده‌سازی می‌شود.

فهرست

- ۱ آزمایش اول: طراحی جمع‌کننده‌ی BCD
- ۱ چکیده
- ۳ سلسله‌مراتب طراحی صورت گرفته
- ۴ جمع‌کننده‌ی یک بیتی دودویی
- ۵ جمع‌کننده‌ی چهاربیتی دودویی
- ۸ جمع‌کننده‌ی یک رقمی BCD
- ۱۲ جمع‌کننده‌ی BCD سه رقمی
- ۱۴ نمونه ورودی اول: ۲۰۰ و ۱۲۲ بدون cin
- ۱۵ نمونه ورودی دوم: ۱۹۹ و ۲۹۹ بدون cin
- ۱۶ نمونه ورودی سوم: ۱۹۹ و ۲۹۹ با cin
- ۱۷ نمونه ورودی چهارم: ۴۹۹ و ۵۰۰ با cin
- ۱۸ نمونه ورودی پنجم: ۷۹۹ و ۸۹۹ بدون cin
- ۱۹ نمونه ورودی ششم: ۷۹۹ و ۸۹۹ با cin
- ۲۰ فایل‌های آزمایش

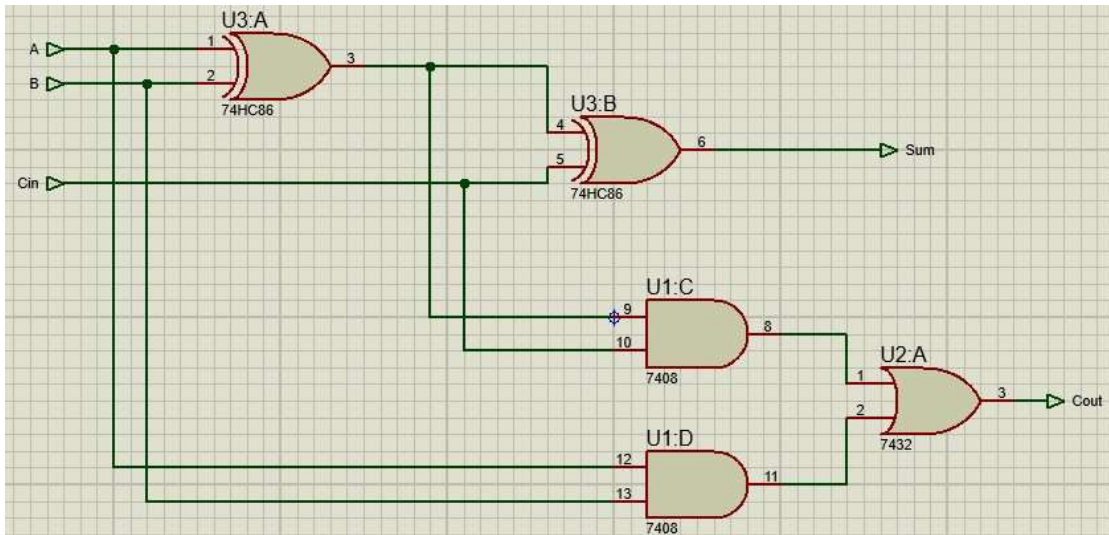
سلسله‌مراتب طراحی صورت گرفته

به طور مختصر، طرح ایجاد شده دارای چنین معماری‌ای است:

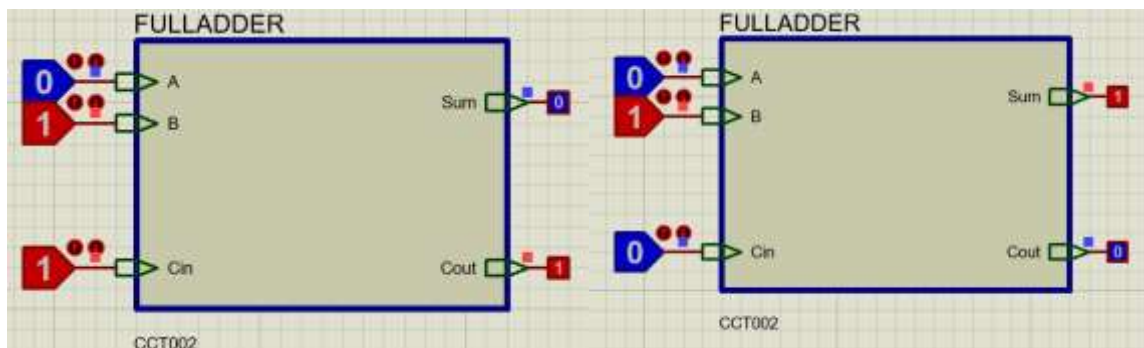


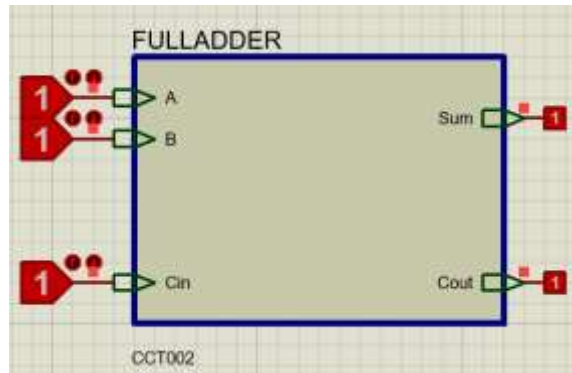
جمع کننده یک بیتی دودویی

در ابتدا با استفاده از تراشه های سری ۷۴ برای گیت های پایه، Full-Adder طراحی می کنیم. برای گیت های AND و OR و XOR به ترتیب از تراشه های 7408 و 7432 و 7486 استفاده شده است که همگی دو ورودی دارند.



مدار ایجاد شده داخل subcircuit قرار داده شده است. در ادامه، با چندین ورودی مختلف، این قسمت از طراحی را آزمایش می کنیم:



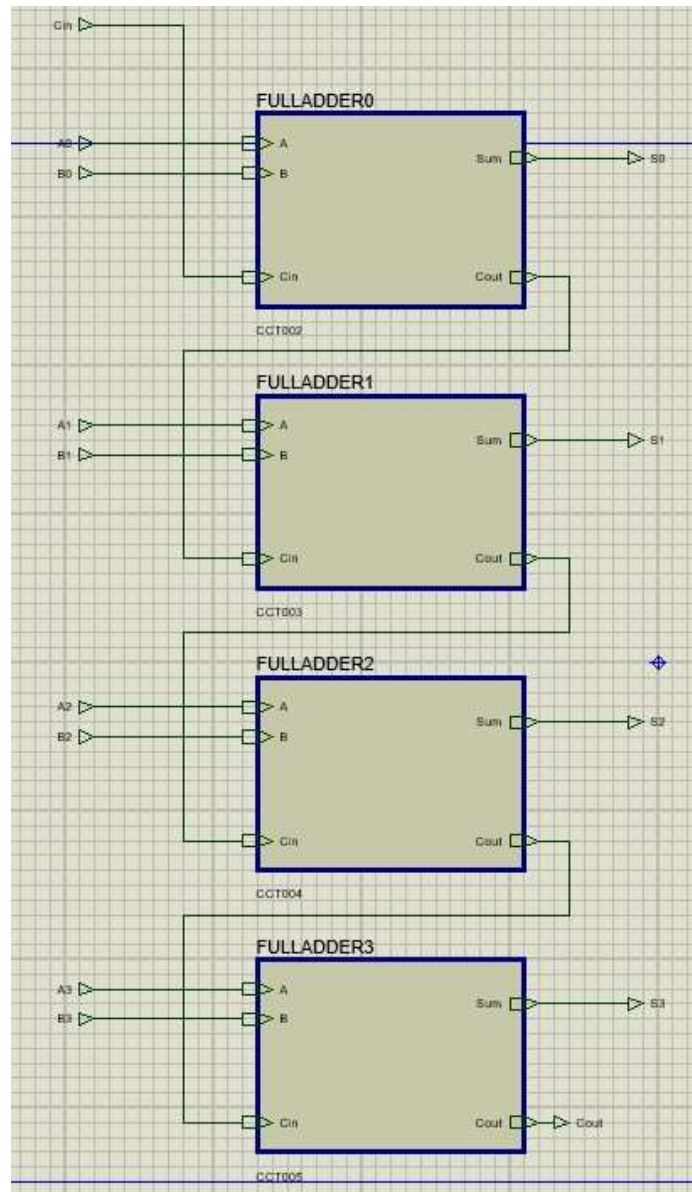


مشاهده می‌گردد که Full-Adder به درستی کار می‌کند. در ادامه، با استفاده از چند واحد Full-Adder، یک مدار جمع‌کننده‌ی باینری چهاربیتی پیاده می‌شود.

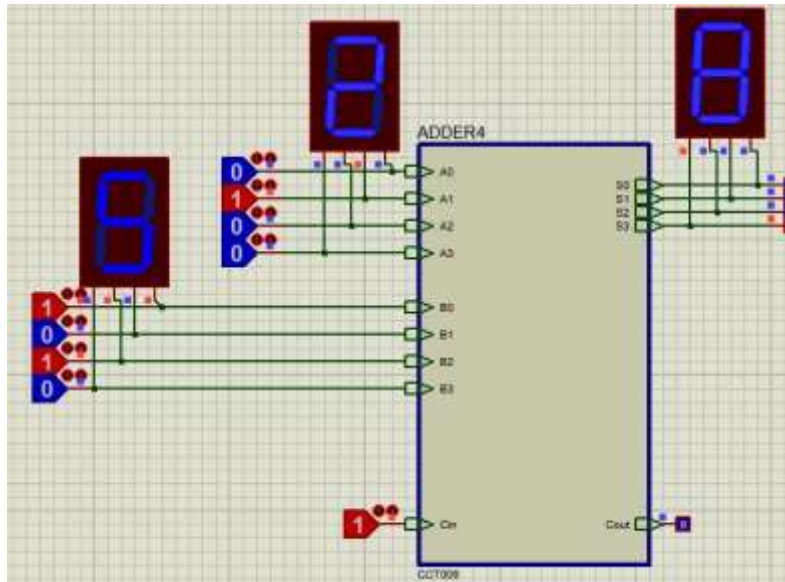
جمع‌کننده‌ی چهاربیتی دودویی

از آنجایی که BCDها دارای چهار بیت هستند (کافیست اعداد ۰ الی ۹ را نمایش دهند) برای جمع زدن آنها از چندین جمع‌کننده‌ی باینری چهاربیتی استفاده می‌شود. همانطور که جلوتر مشاهده خواهد شد، نیازمند ایجاد جمع‌کننده‌ی چهاربیتی هستیم.

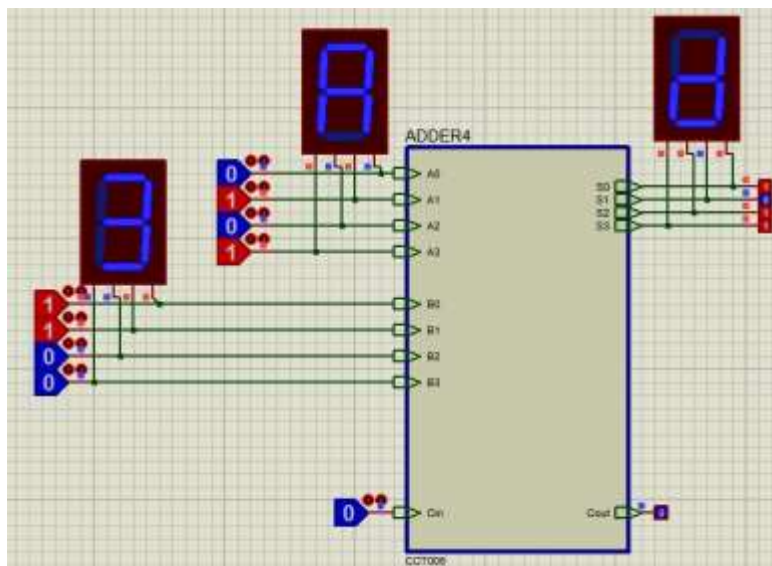
برای پیاده‌سازی جمع‌کننده‌ی چهاربیتی، از چهار عدد Full-Adder بصورت ripple-carry استفاده شده است. همچنین یک ورودی carry-in و خروجی carry-out برای این جمع‌کننده‌ی چهاربیتی در نظر گرفته شده است که در پیاده کردن جمع‌کننده‌ی BCD از آنها استفاده خواهد شد.



مجدداً، مدار طراحی شده در قالب یک subcircuit قرار گرفته تا در ادامه‌ی سلسله‌مراتب طراحی از آن استفاده کنیم. سپس با دادن چند ورودی، این واحد را نیز آزمایش می‌کنیم. چند مثال از ورودی‌های بررسی شده در ادامه آورده شده است.

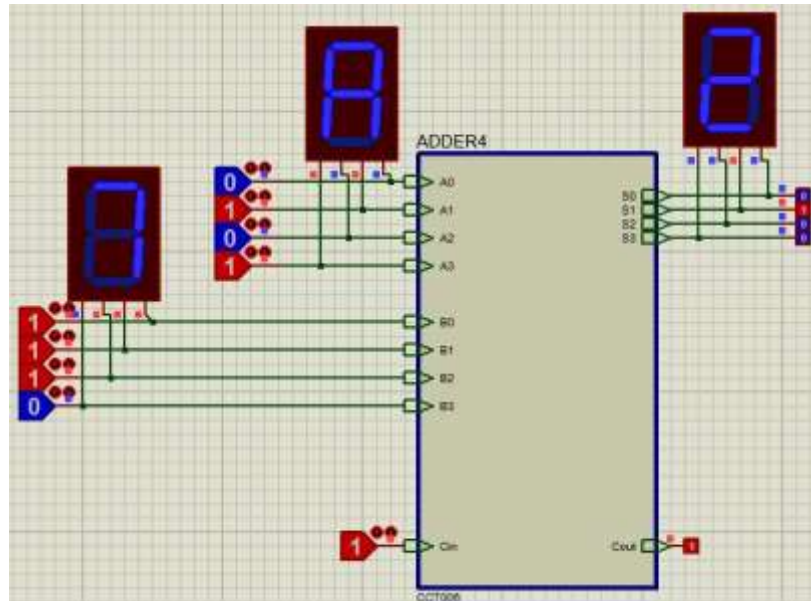


در این نمونه، دو عدد ۲ و ۵ وارد شده و ورودی رقم نقلی داریم. حاصل ۸ است.



در این نمونه، دو عدد ۳ و ۱۰ وارد شده و ورودی رقم نقلی نداریم. حاصل ۱۳ است.

دقت داریم که در ادامه کاری با اعداد بزرگتر از ۹ نداریم ولی در این قسمت آنها را بررسی می‌کنیم.

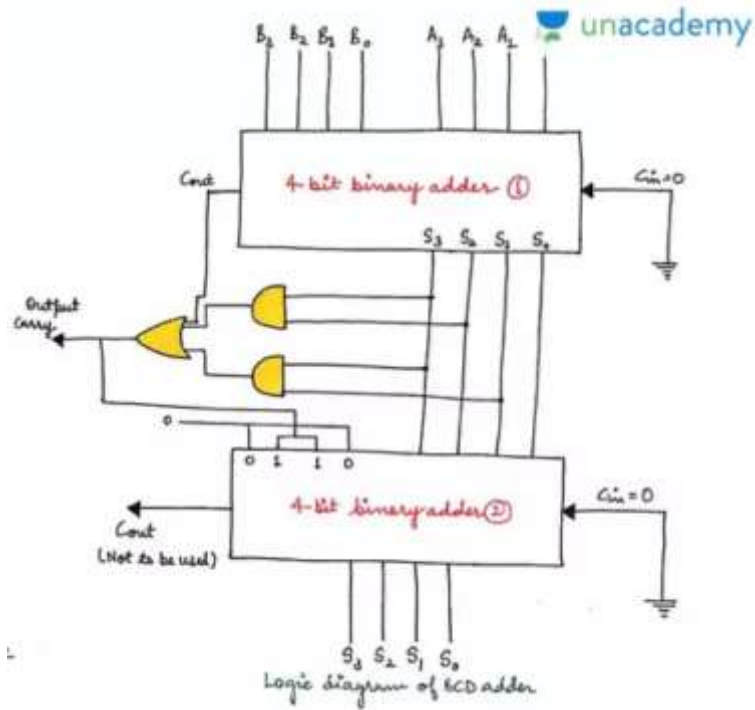


در این نمونه، دو عدد ۷ و ۱۰ به همراه رقم نقلی وارد شده‌اند. حاصل برابر ۱۸ خواهد بود، که بصورت یک خروجی نقلی (برابر ۱۶) و عدد ۲ ظاهر می‌شود.

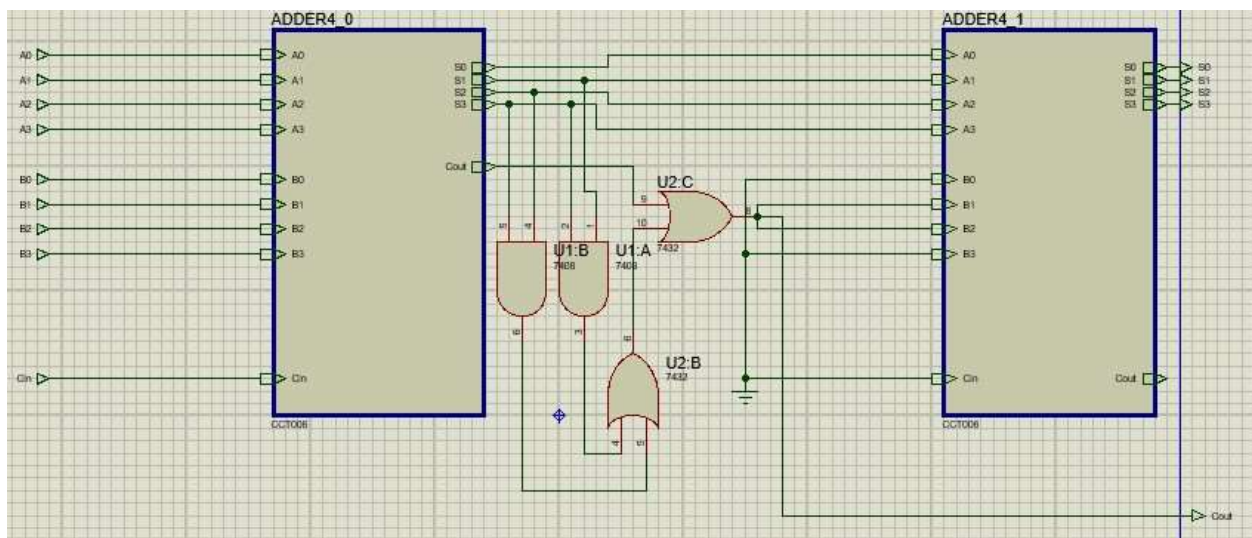
جمع‌کننده‌ی یک رقمی BCD

برای پیاده‌سازی این مدار، مطابق شماتیک‌های موجود برای جمع‌کننده‌ی یک رقمی BCD عمل می‌کنیم. یک نمونه از این شماتیک‌ها در ادامه آورده شده است. برای پیاده‌سازی جمع‌کننده‌ی یک رقمی BCD نیازمند دو جمع‌کننده‌ی چهار بیتی هستیم: یکی برای جمع کردن دو عدد داده شده، و دیگری برای حالاتی که نیازمند جمع زدن این حاصل با عدد ۶ هستیم.

دقت داریم که گاهی اوقات حاصل جمع دو BCD از ۹ زیادتر می‌شود، در این حالات باید حاصل با عدد ۶ جمع شود و رقم نقلی تولید شود، حال آنکه جمع‌کننده‌ی چهار بیتی ما زمانی که حاصل از ۱۵ زیادتر شود رقم نقلی تولید می‌کند، پس باید یک جمع‌کننده‌ی جدا برای جمع با ۶ (در صورت لزوم) داشته باشیم.

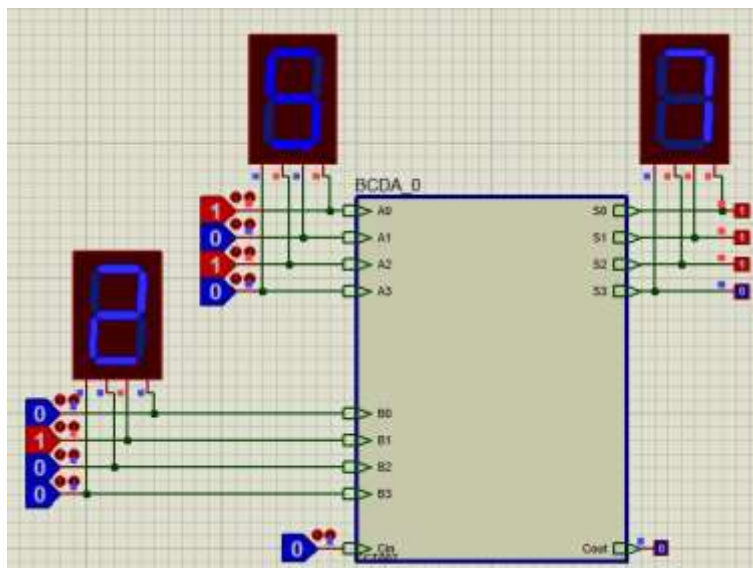


در تصویر فوق نمونه شماتیک مورد استفاده مشاهده می شود، البته طرح پیاده شده دارای رقم نقلی ورودی است که به ورودی c_{in} از 1 4-bit binary adder (جمع کننده ی بالایی) متصل می شود.

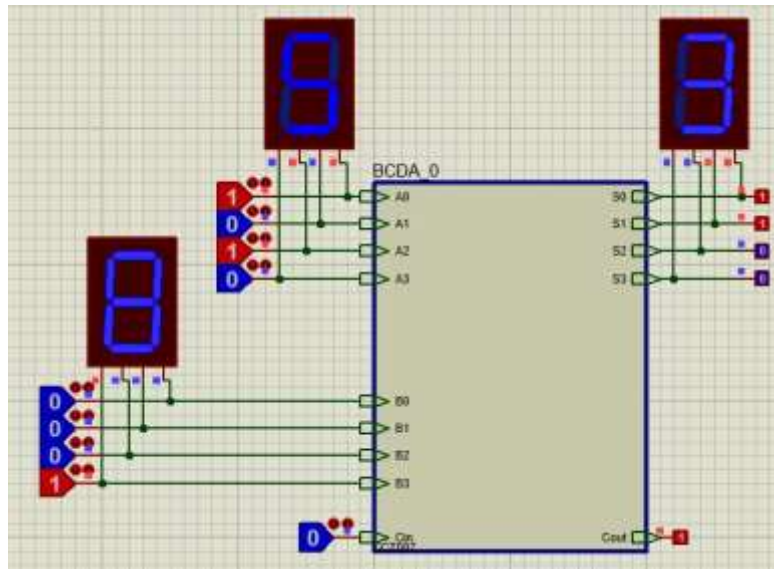


دقت داریم که بجای گیت OR سه ورودی، از دو گیت OR دو ورودی استفاده شده است که تنوع تراشه و در برخی حالات تعداد تراشه مصرفی کمتر شود.

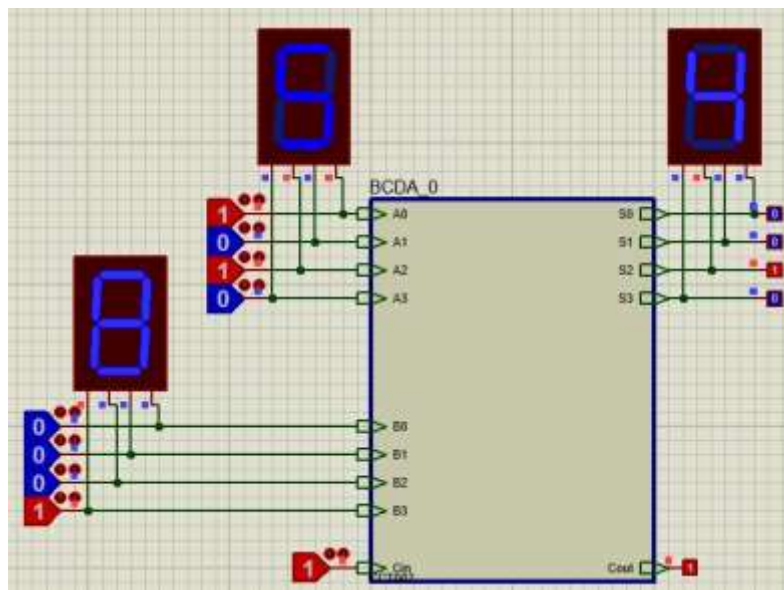
حال مجدداً این طرح را بصورت یک subcircuit در آورده و با دادن چند ورودی، عملکرد آن را بررسی می‌کنیم. چند مثال از ورودی‌های داده شده در ادامه آورده شده است. دقت داریم که بر خلاف قسمت قبل، دیگر ارقام بزرگتر از ۹ مشاهده نخواهیم کرد و اگر جمع ورودی‌های داده شده از ۹ زیادتر شود رقم نقلی تولید خواهد شد.



در این حالت، دو ورودی ما ۲ و ۵ هستند و carry-in نداریم. خروجی ما ۷ خواهد بود و رقم نقلی تولید نمی‌شود.



در این حالت، دو ورودی ما ۸ و ۵ هستند و carry-in نداریم. جمع این دو رقم برابر ۱۳ است، پس خروجی تولید شده ۳ است و خروجی نقلی ایجاد می‌شود.



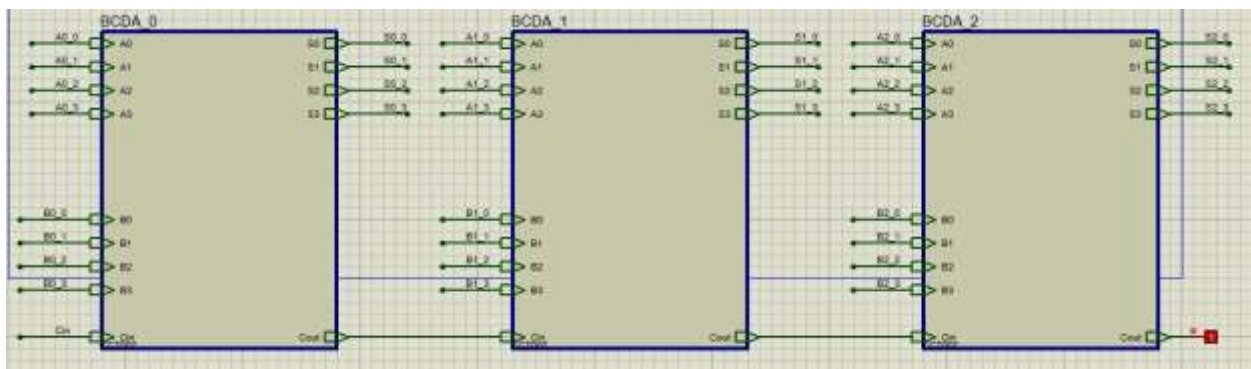
در این حالت همان ورودی‌های قبلی را داریم ولی carry-in هم داریم. مشاهده می‌کنیم که رقم تولید شده یکی زیاده‌تر خواهد بود.

جمع کننده ی BCD سه رقمی

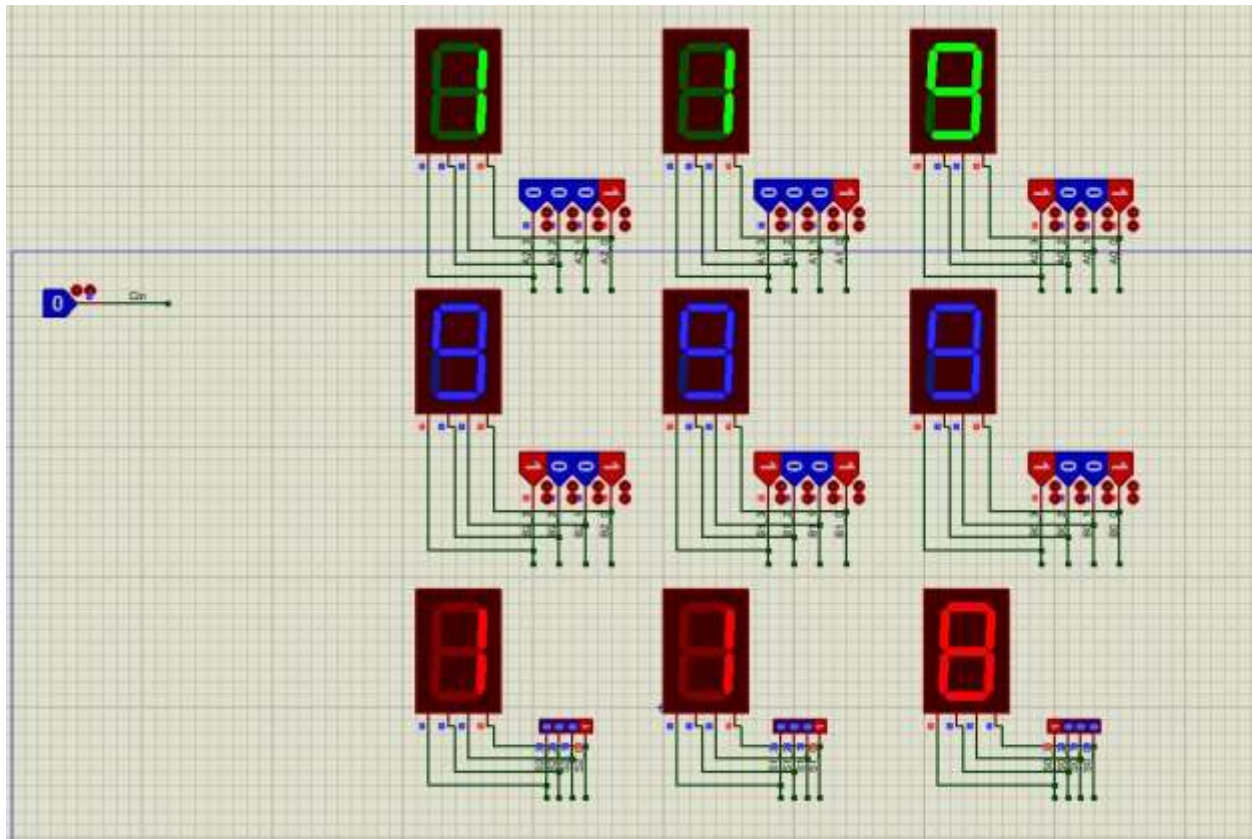
در این قسمت، مشابه قسمت دوم عمل می کنیم و جمع کننده های BCD را بصورت ripple-carry وصل کرده و برای ورودی ها مشابه قسمت های دوم و سوم از 7-segment استفاده می کنیم. دو عدد ورودی بصورت سه رقم سبز و سه رقم آبی هستند و حاصل جمع آنها در قالب یک عدد قرمز و یک LOGICPROBE که بیانگر خروجی رقم نقلی مجموعه است نشان داده شده است.

همچنین دقت داریم که برای پرهیز از شلوغی، از ترمینال برای برقراری اتصالات استفاده شده است.

در تصویر اول، سه واحد جمع کننده مشاهده می شود:



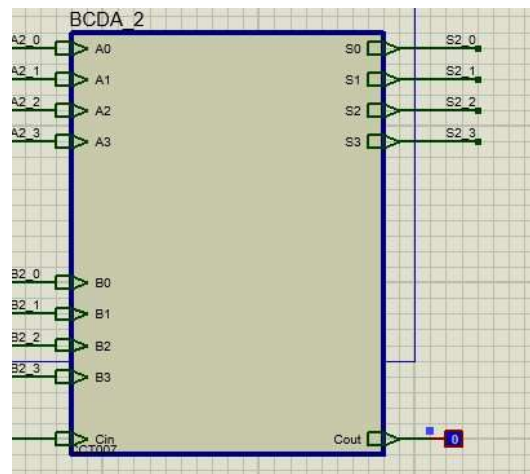
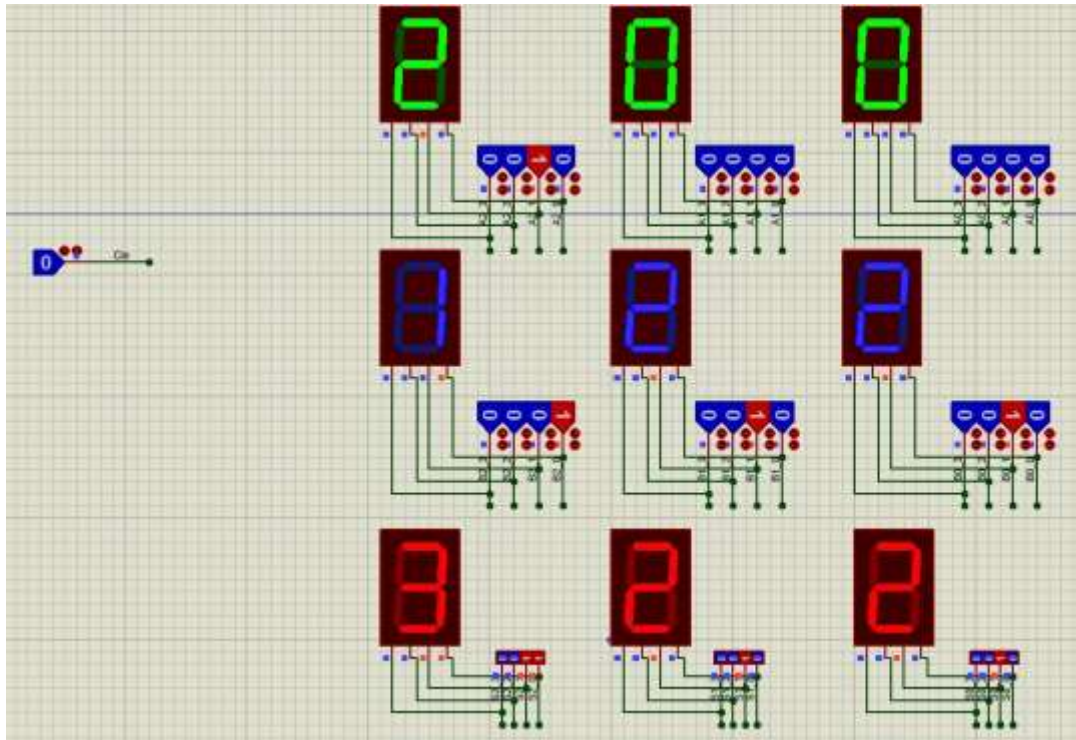
همچنین خروجی carry-out (خروجی رقم نقلی مجموعه) در گوشه ی این تصویر دیده می شود. اتصالات مدار با استفاده از ترمینال انجام شده است. در ادامه، ورودی ها و سه رقم تولید شده ی خروجی را مشاهده می کنیم که آنها نیز با ترمینال به بقیه ی مدار وصل شده اند. برای مشاهده ی آسانتر BCD ها، از bcd-7-segment استفاده شده است.



در ادامه نیز با دادن چند ورودی، خروجی مدار را بررسی می‌کنیم که صحت عملکرد این مدار بررسی شود.
 (تصاویر بالا نیز از شمای شبیه‌سازی بودند و مشاهده شد که ۱۱۹ و ۹۹۹ جمع زده شده و ۱۱۸ و یک رقم نقلی
 خروجی ایجاد شده است)

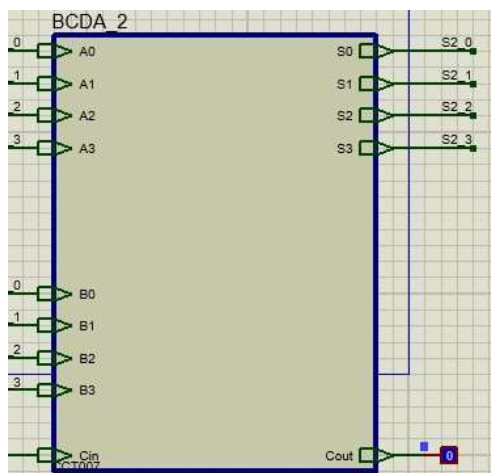
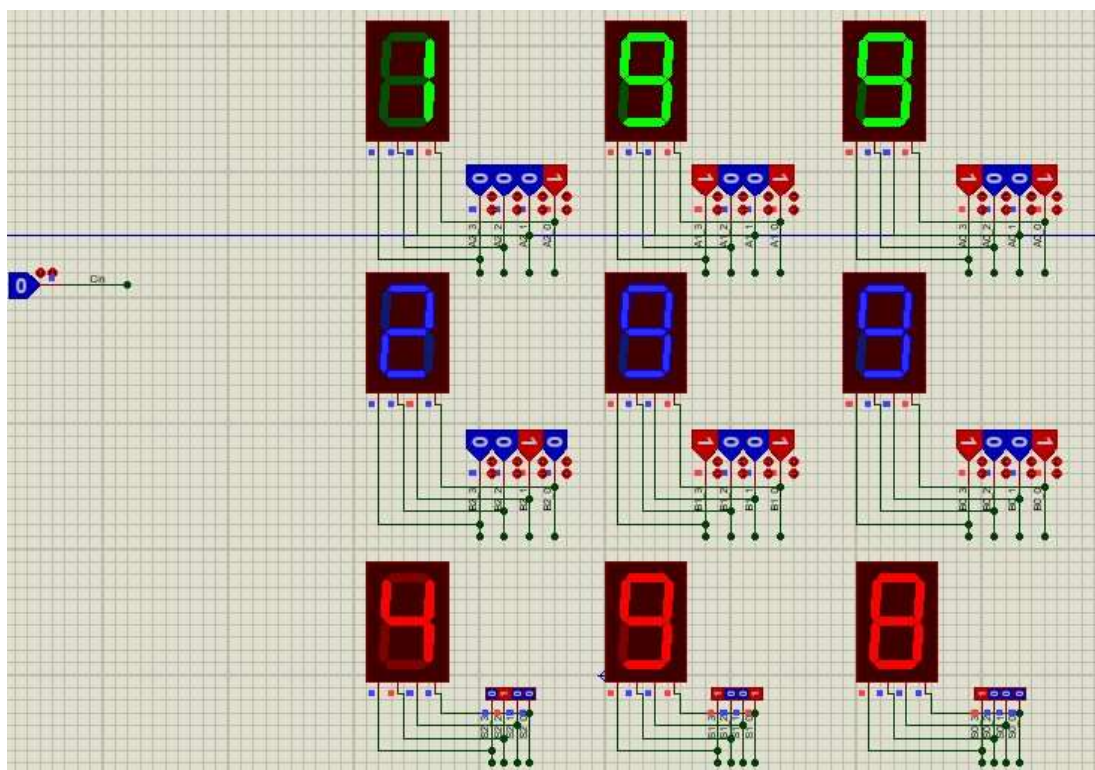
در هر صفحه به بررسی یک نمونه ورودی و تحلیل خروجی‌های ایجاد شده می‌پردازیم.

نمونه ورودی اول: ۲۰۰ و ۱۲۲ بدون cin



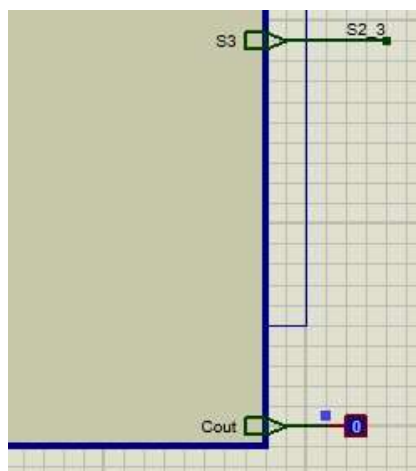
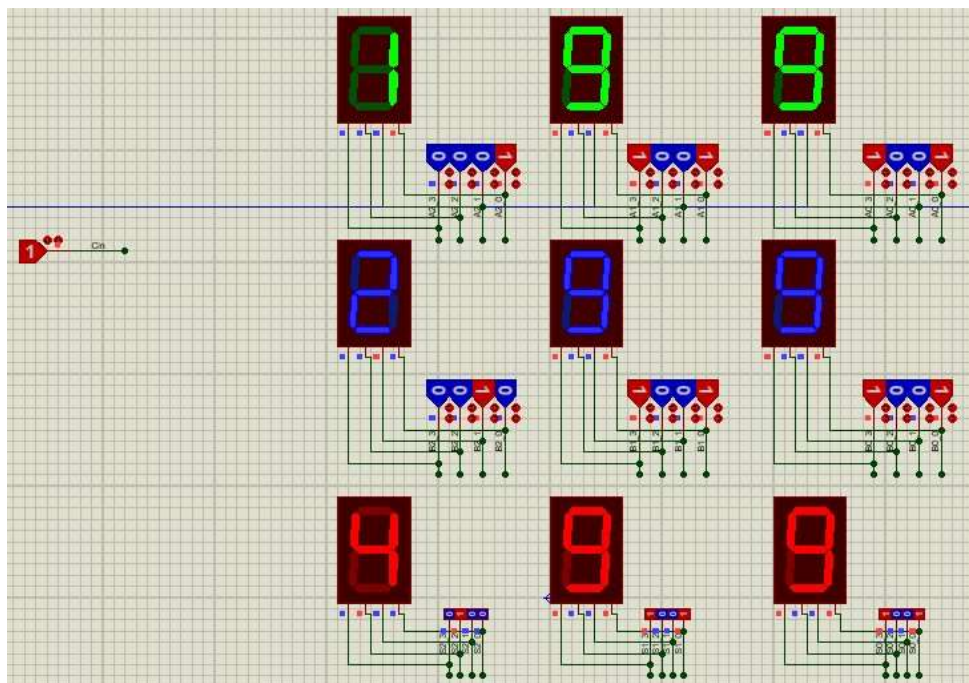
مشاهده می شود که خروجی ایجاد شده ۳۲۲ است و خروجی نقلی نداریم.

نمونه ورودی دوم: ۱۹۹ و ۲۹۹ بدون cin



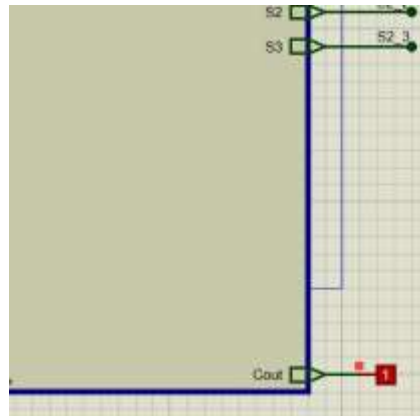
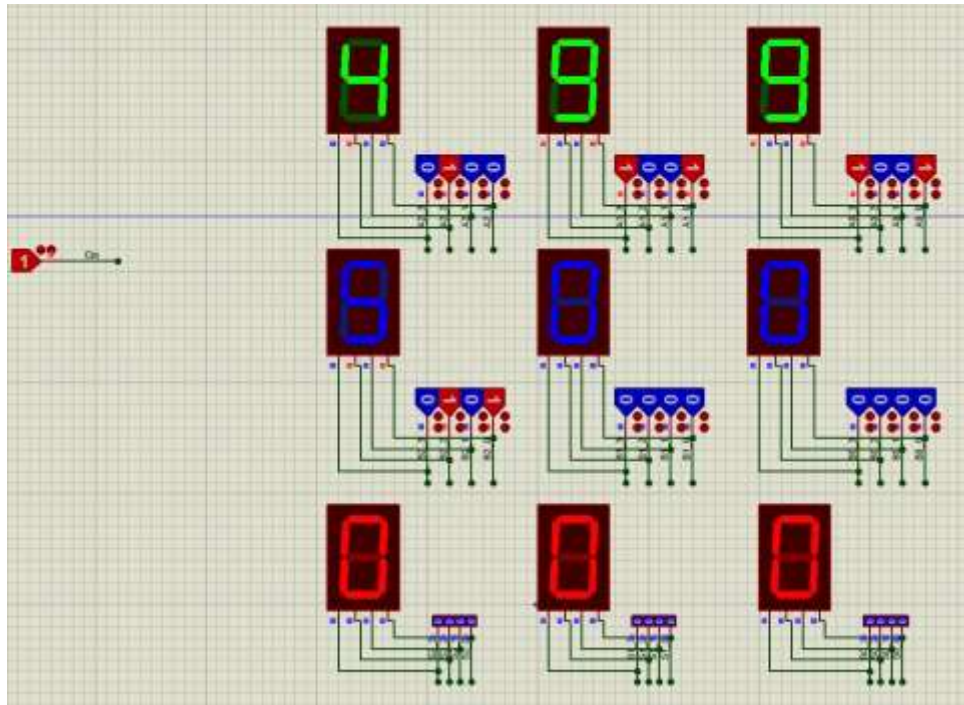
مشاهده می‌شود که خروجی ایجاد شده ۴۹۸ است و خروجی نقلی نداریم.

نمونه ورودی سوم: ۱۹۹ و ۲۹۹ با cin



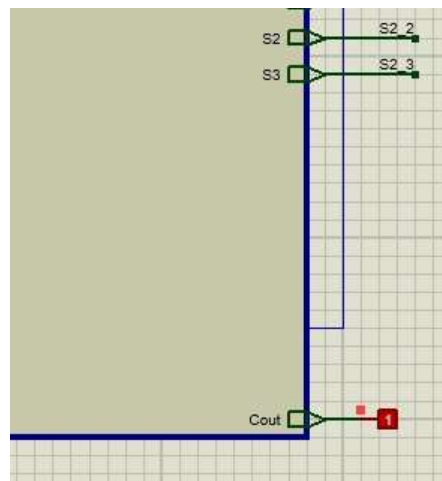
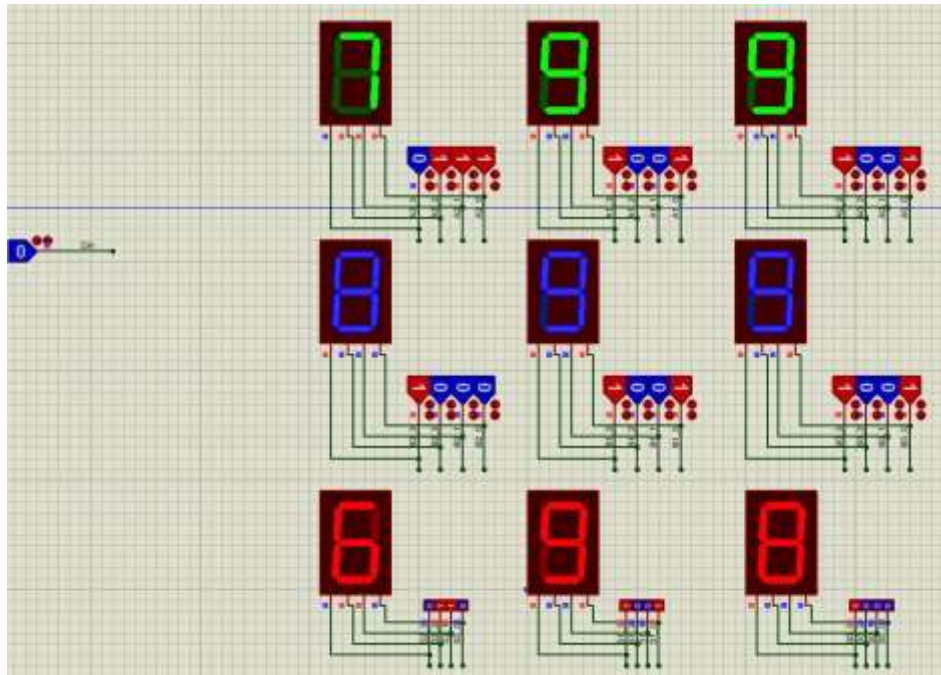
همانطور که انتظار داریم حاصل فقط یکی از ورودی‌های سابق زیادتر است (۴۹۹) است و خروجی نقلی و دو رقم پرارزش‌تر تغییر نکرده‌اند.

نمونه ورودی چهارم: ۴۹۹ و ۵۰۰ با cin



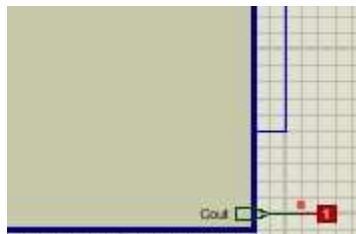
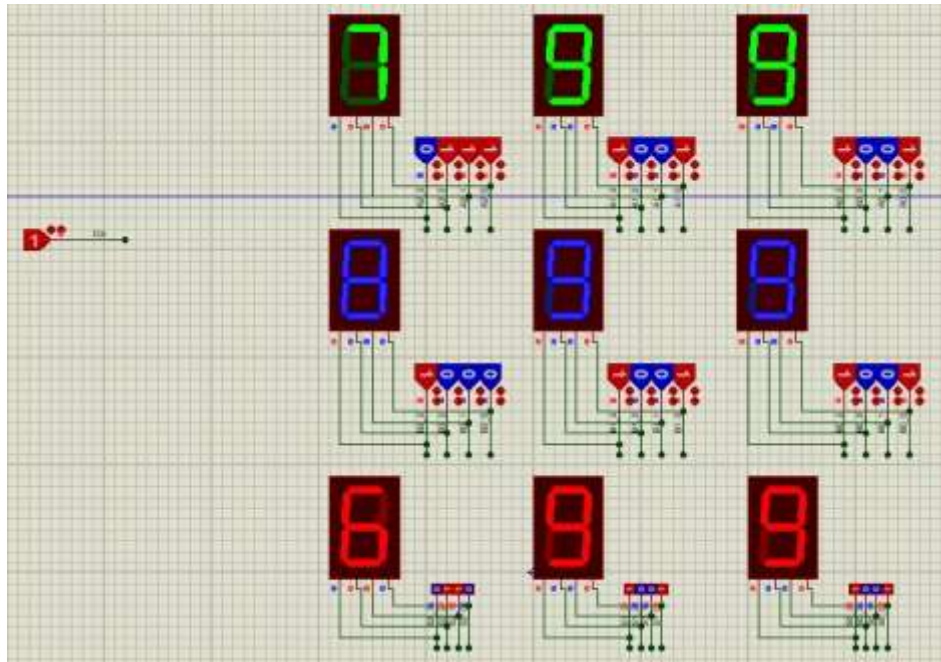
مجموع ورودی‌های داده شده برابر ۱۰۰۰ است، بنابراین مطابق انتظارمان ارقام نشان داده شده صفر هستند و خروجی رقم نقلی فعال شده است.

نمونه ورودی پنجم: ۷۹۹ و ۸۹۹ بدون cin



مجموع ورودی‌های داده شده ۱۶۹۸ است، بنابراین مطابق انتظارمان خروجی رقم نقلی فعال شده و عدد ۶۹۸ نمایش داده شده است.

نمونه ورودی ششم: ۷۹۹ و ۸۹۹ با cin



مجموع ورودی‌ها از نمونه قبلی یکی زیادتر است، بنابراین فقط رقم یکان حاصل تغییر کرده و سایر ارقام ثابت می‌مانند.

فایل های آزمایش

فایل های آزمایش بصورت پوشه بندی شده ارسال شده اند، یک پروژه پروتئوس مربوط به مدار نهایی است و یک پروژه پروتئوس شامل نمونه از ماژول های ساخته شده (فول ادر و ادر ۴ بیتی و ...) است که به منظور تست از آن استفاده شده است.