



آزمایشگاه معماری کامپیوتر

تابستان ۱۴۰۲

آزمایش اول

جمع کننده BCD

محمدپیام تائبی – ۴۰۰۱۰۴۸۶۷

امیرحسین علمدار – ۴۰۰۱۰۵۱۴۴

ماهان بیهقی – ۴۰۰۱۰۴۸۳۴

مرحله اولیه (proteus):

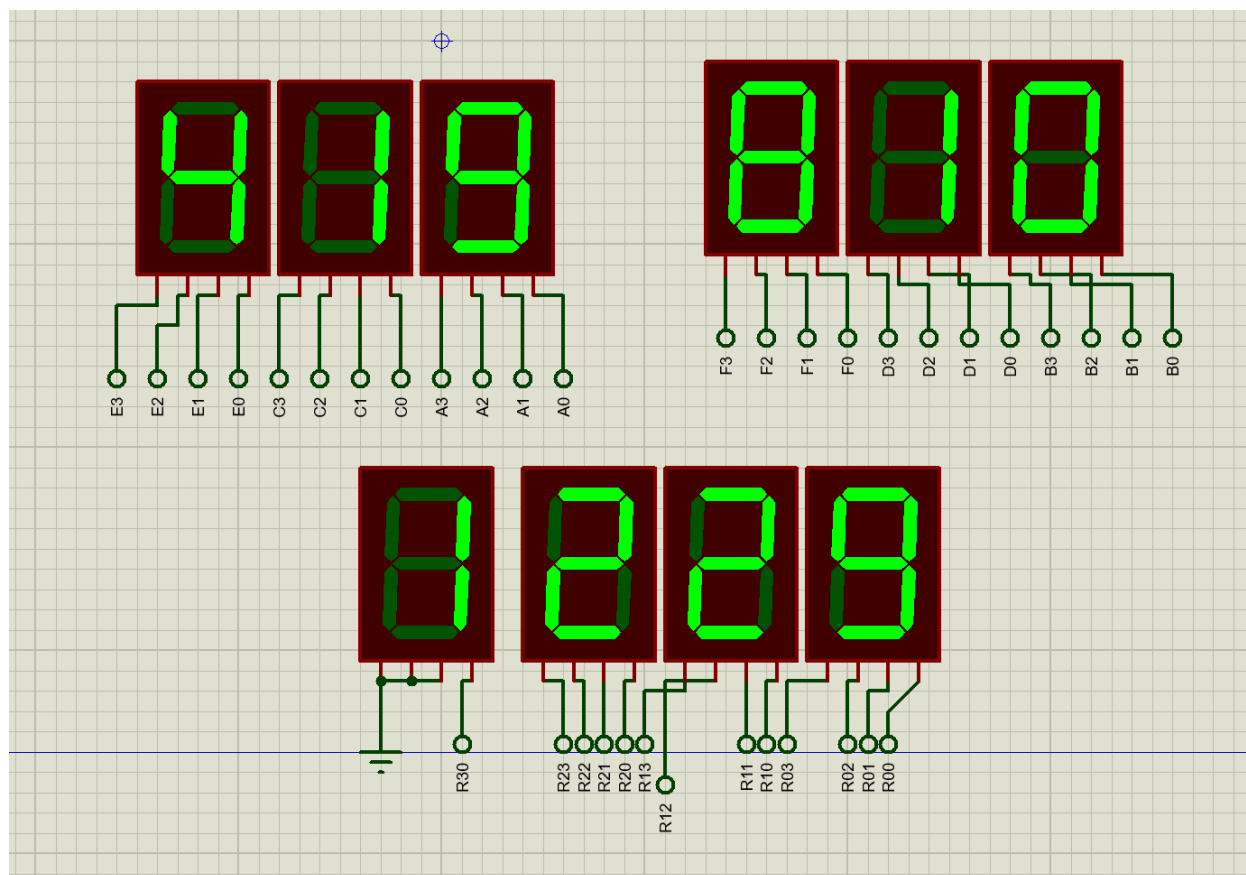
هدف و حدس از نتیجه آزمایش:

در این آزمایش قصد داریم جمع کننده دهنده بسازیم

میدانیم در نمایش دهنده اعداد، برای هر رقم یک مقدار چهار بیتی داریم که به شکل باینری میتواند از صفر تا پانزده را نمایش دهد و ما مقادیر صفر تا نه را به عنوان یک رقم برای هر رقم دهنده نگه میداریم دقت کنید منطقاً هیچ رقمی نباید بزرگ تر از نه شود و اعداد ده تا پانزده در واقع استفاده نمیشوند

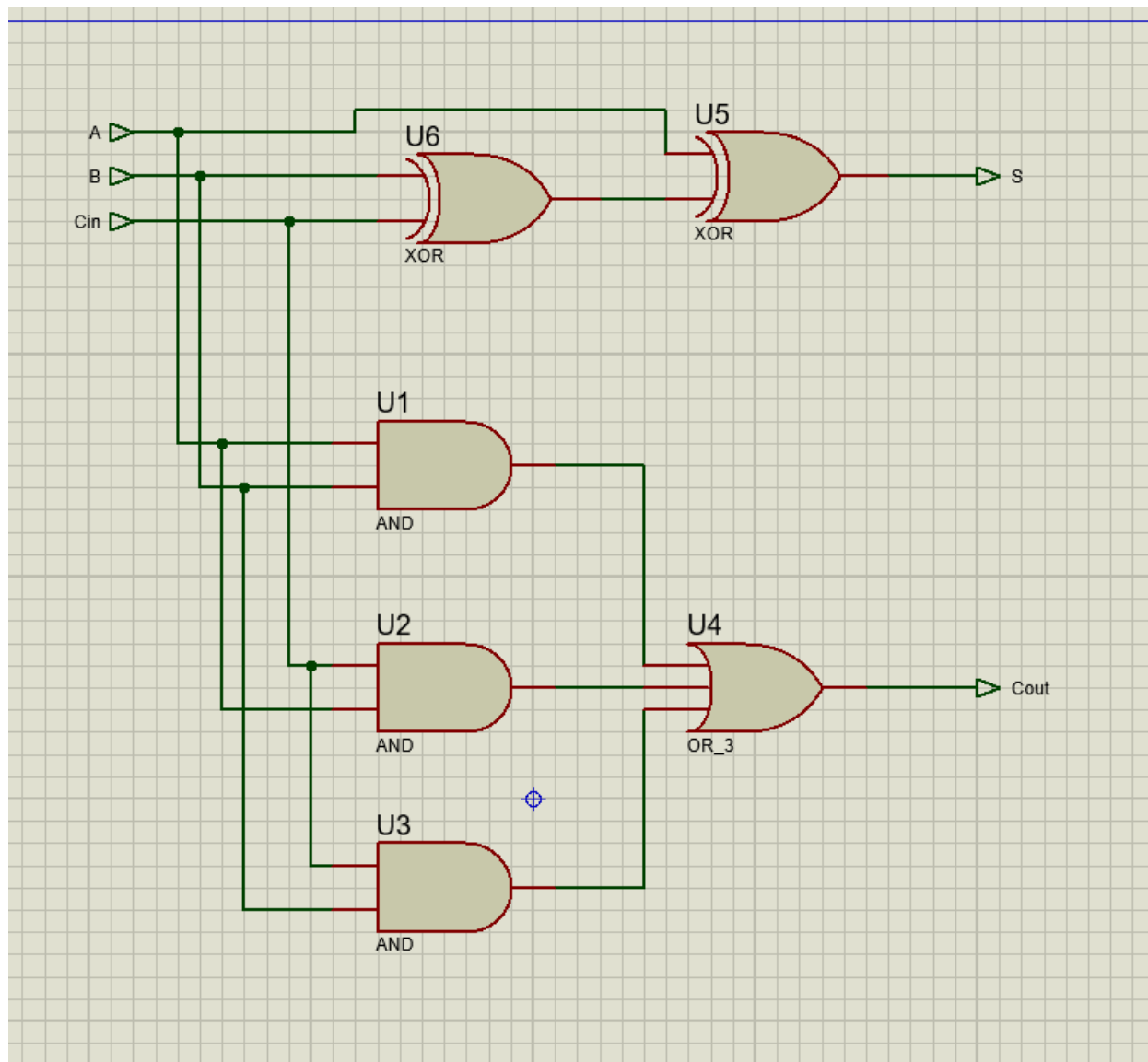
حال که اعداد BCD را شناختیم توقع داریم که یک جمع کننده ۳ رقمی به شکل دهنده بسازیم مثلاً عدد ۴۱۹ و ۸۱۰ را به عنوان ورودی بدیم (هرکدام از رقم ها به شکل یک عدد باینری ۴ بیتی است) و در خروجی یک عدد ۴ رقمی که هر کدام یک عدد باینری ۴ بیتی است بگیریم که خروجی ۱۲۲۹

که در آخر آزمایش توقع داریم به این شکل بشود :

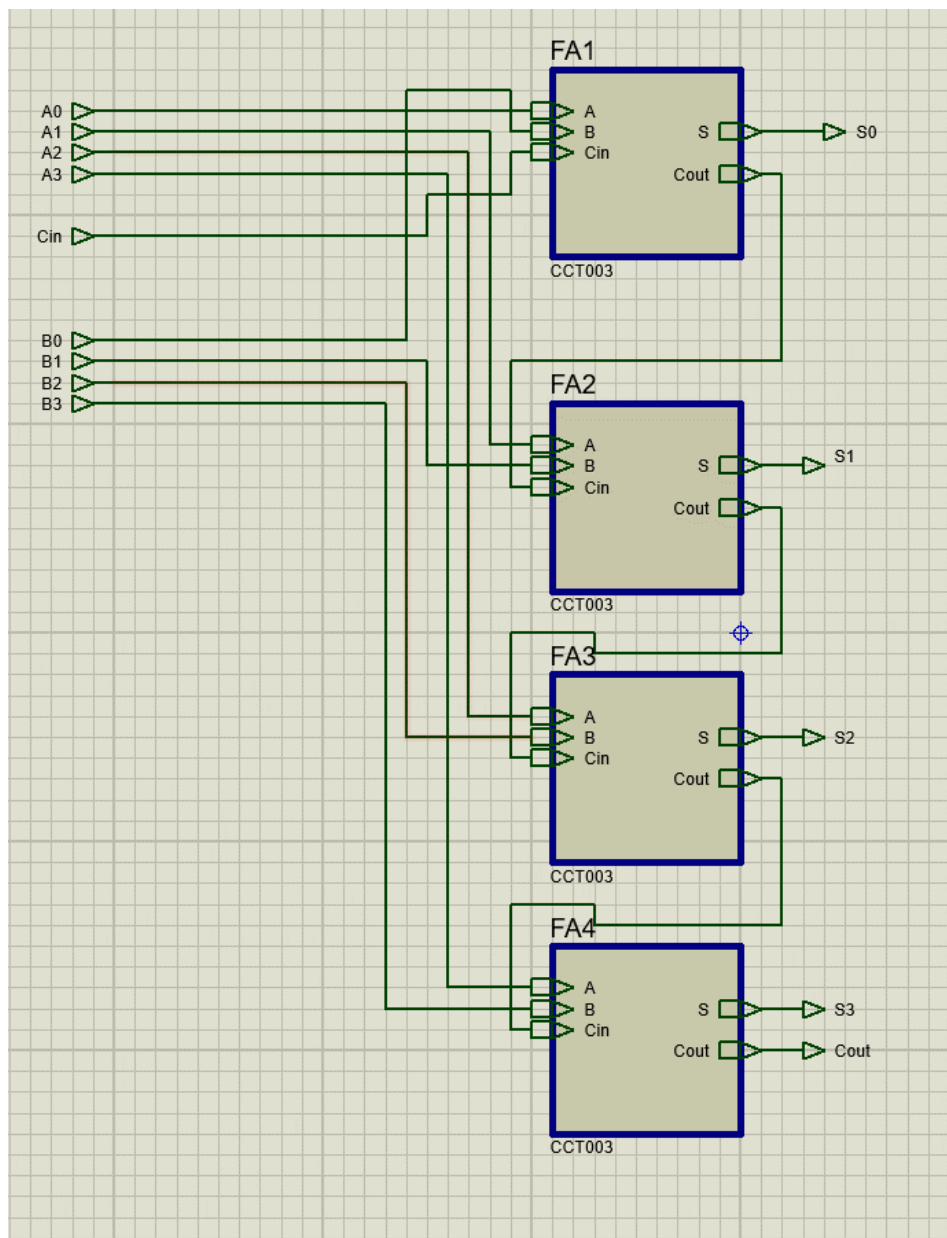


شرح آزمایش:

اولیه ترین قطعه ای که خواهیم ساخت Full Adder است ، میدانیم فول ادر ها سه ورودی دارند و این ۳ بیت را جمع میکند و به شکل دو خروجی sum و carry out میدهد یعنی اگر همه ورودی ها صفر باشد خروجی مدار صفر و اگر یکی یک باشد s یک میشود و carry صفر ، و اگر دو تا یک باشند باید s صفر شود و carry یک، و اگر هر سه ورودی یک باشد باید هر دو خروجی یک شود که مدار ترکیبی که میتوان برای full adder ساخت را مطابق این شکل پیاده میکنیم :



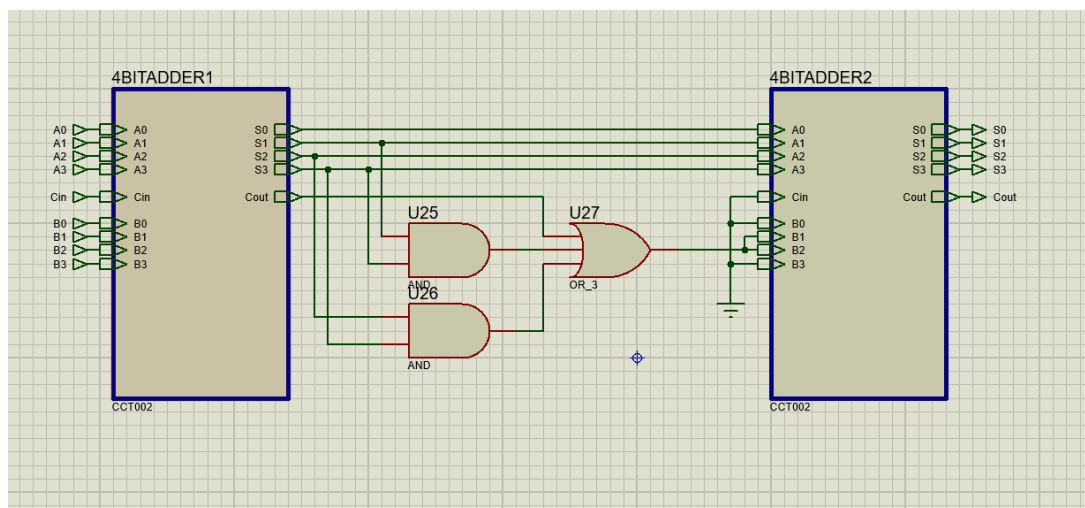
حال میتوانیم از به هم وصل کردن هر تعدادی از این full adder ها به شکل ripple carry adder جمع کننده بسازیم که در این مدار چون اعداد BCD هستند و در ۴ بیت نمایش داده میشوند منطقاً به جمع کننده های ۴ بیتی نیاز داریم پس در مرحله بعدی سعی میکنیم با استفاده از چهار فول ادر یک رپل کری ۴ بیتی بسازیم؛ کافی است کری خروجی هر فول ادر به کری ورودی فول ادر بعدی وصل باشد که در نهایت مدار شکل زیر را خواهیم داشت :



تا اینجا توانستیم جمع کننده ۴ بیتی بسازیم اما جمع کننده چهار بیتی برای اعداد BCD کافی نیست چرا؟ چون اعداد ما واقعا اعداد ۴ بیتی نیستند و اگر حاصل خارج از بازه ۰ تا ۹ قرار بگیرد یعنی در بازه ۱۰ تا ۱۵ قرار بگیرد باید ۱۰ تا از آن کم شود که فقط رقم یکان آن نگه داشته شود پس نمیتوانیم مستقیما از جمع کننده ۴ بیتی استفاده کنیم و باید تغییری اعمال کنیم که مدار طوری شود که ۲ عدد ۴ بیتی و کری را بگیرد سپس این ۲ عدد را به شکل عادی جمع کند حال اگر که خروجی ما بزرگ تر از ۹ شد باید حاصل خروجی منهای ۱۰ شود که چون عدد ۱۰ در نمایش باینری ۴ بیتی معادل ۱۰۱۰ است و ما مداری که منها کند نداریم فقط جمع کننده داریم پس از مکمل دوی عدد استفاده میکنیم (میدانیم جمع کردن با مکمل دوی هر عدد باینری معادل با تفریق آن عدد است) مکمل دوی عدد ۱۰ میشود ۰۱۱۰ پس مدار ما باید طوری باشد که اگر حاصل خروجی ما بیش از ۹ شد باید یک بار دیگر حاصل را با ۰۱۱۰ که معادل عدد ۶ است جمع کند؛ مسئله نهایی این است که به یک مدار ترکیبی نیاز داریم که بفهمیم عدد خروجی ۵ بیتی (۴ بیت عادی و یک بیت کری) بزرگ تر از ۹ هست یا نه که خیلی ساده اگر بیت کری یک باید که قطعا بزرگتر از ۱۶ هستیم که یکی از این حالات است، از طرفی اگر بیت s_3 (مطابق شکل) یک باشد یعنی عدد ما حداقل ۸ است پس فقط زمانی $s_3=1$ قابل قبول است که $s_1=s_2=0$ باشد پس در نهایت مدار ترکیبی را میشود به شکل $Cout + s_3s_1 + s_3s_2$ ساخت که اگر این حاصل یک شد باید مدار ما با ۶ جمع شود که واضحا میتوان یک جمع کننده ۴ بیتی دیگر قرار داد و همواره با 0xx0 جمع کند (کری ورودی هم صفر باشد) و x اینجا برابر است با:

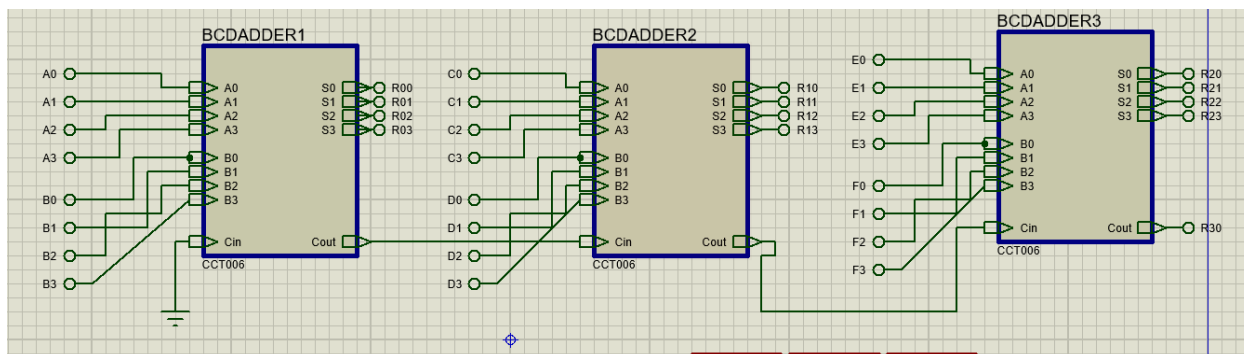
$$X = Cout + s_3s_1 + s_3s_2$$

پس مدار نهایی ما به شکل زیر میشود (به طور خلاصه مدار یک جمع کننده یک رقمی BCD است):



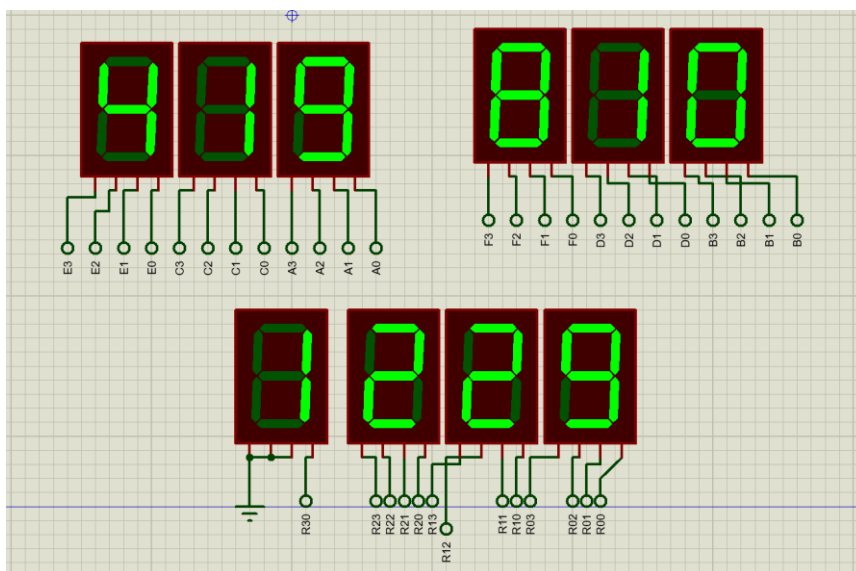
نام ماژولی که ساختیم را BCDADDER میگذاریم حال به گام آخر آزمایش رسیدیم کافی است ۳ تا از این ماژول ها قرار دهیم و کری خروجی هر جمع کننده را به کری ورودی جمع کننده ی بعدی وصل کنیم

در نهایت مدار ما به این شکل میشود:

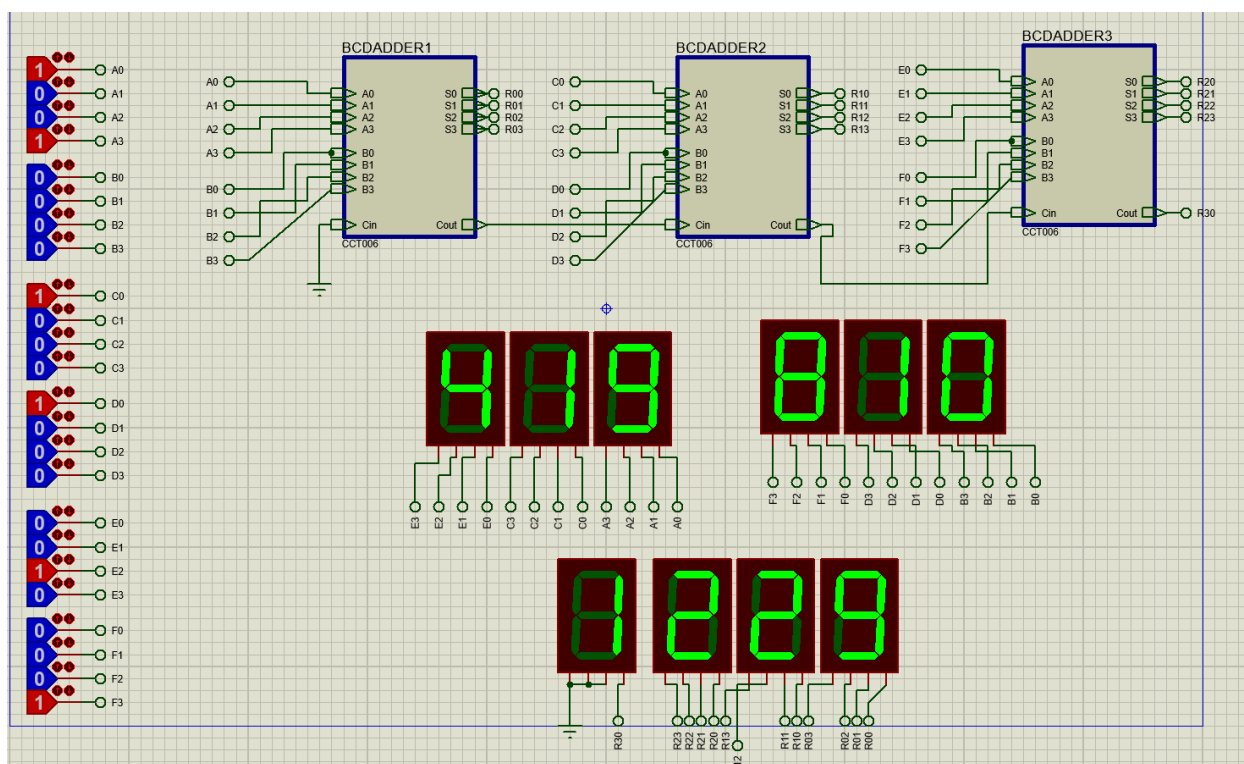


این مدار مطابق شکل عدد ۳ رقمی ای که ارقام آن با ACE نمایش داده شده را با عدد سه رقمی که به شکل BDF نمایش داده شده را جمع میزند منطقاً مدار ما ۶ رقم معادل ۲۴ بیت ورودی دارد که این مدل ورودی دادن شاید خیلی دلنشین نباشد پس از SEVEN SEGMENT استفاده کردم و ۴ بیت مربوط به هر رقم را به یک سون سگمنت دادم تا راحت تر ورودی ها قابل نمایش باشند

خروجی مدار نیز ۳ عدد ۴ بیتی به شکل BCD و یک بیت کری است که این کری هم میتوان به شکل BCD نمایش داد که برای آن هم یک سون سگمنت قرار دادم در کل سون سگمنت ها را میتوانید در شکل ببینید:



دقت کنید در این مدار من Cin را صفر دادن اما میتوانستید برای آن هم یک logic state قرار دهید و آن را نیز ورودی دهید که وقتی بخواهیم جمع کننده های بزرگ تری بسازیم کاربرد دارد در نهایت مدار نهایی ما همچین شکلی دارد :

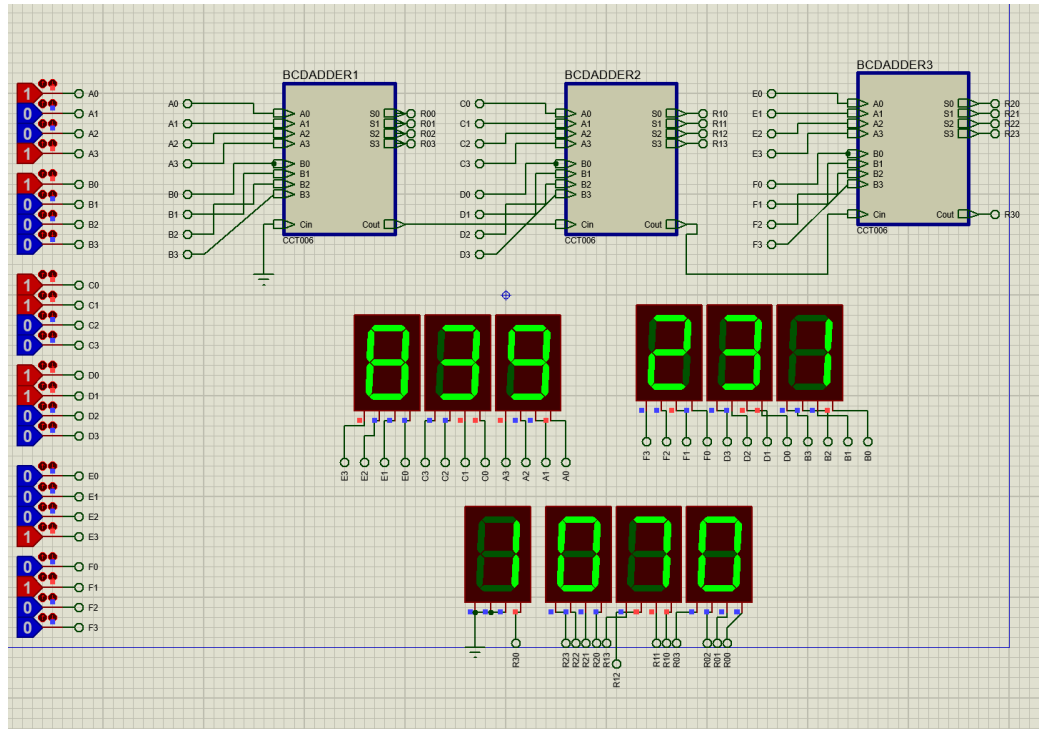


در ادامه قصد دارم چند نمونه ورودی و خروجی روی مدار نمایش دهم تا کارکرد مدار به درستی بررسی شود :

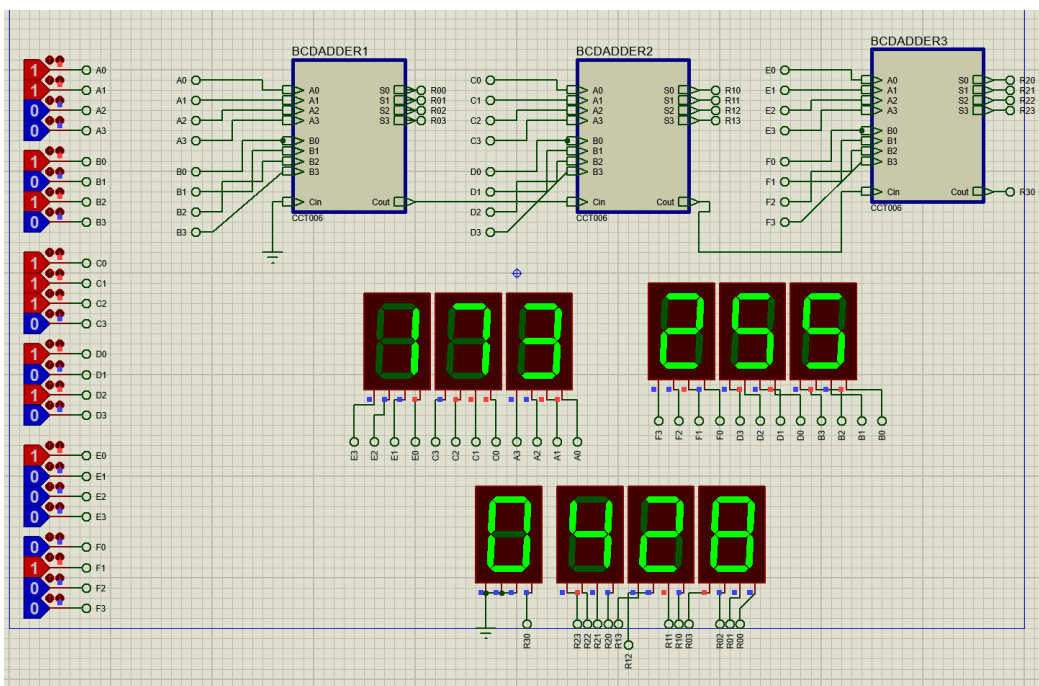
واضحا ورودی و خروجی ها روی سون سگمنت ها قابل مشاهده هستند و از این جهت کار با این مدار به شدت ساده است

صحت سنجی خروجی :

نمونه ی اول:



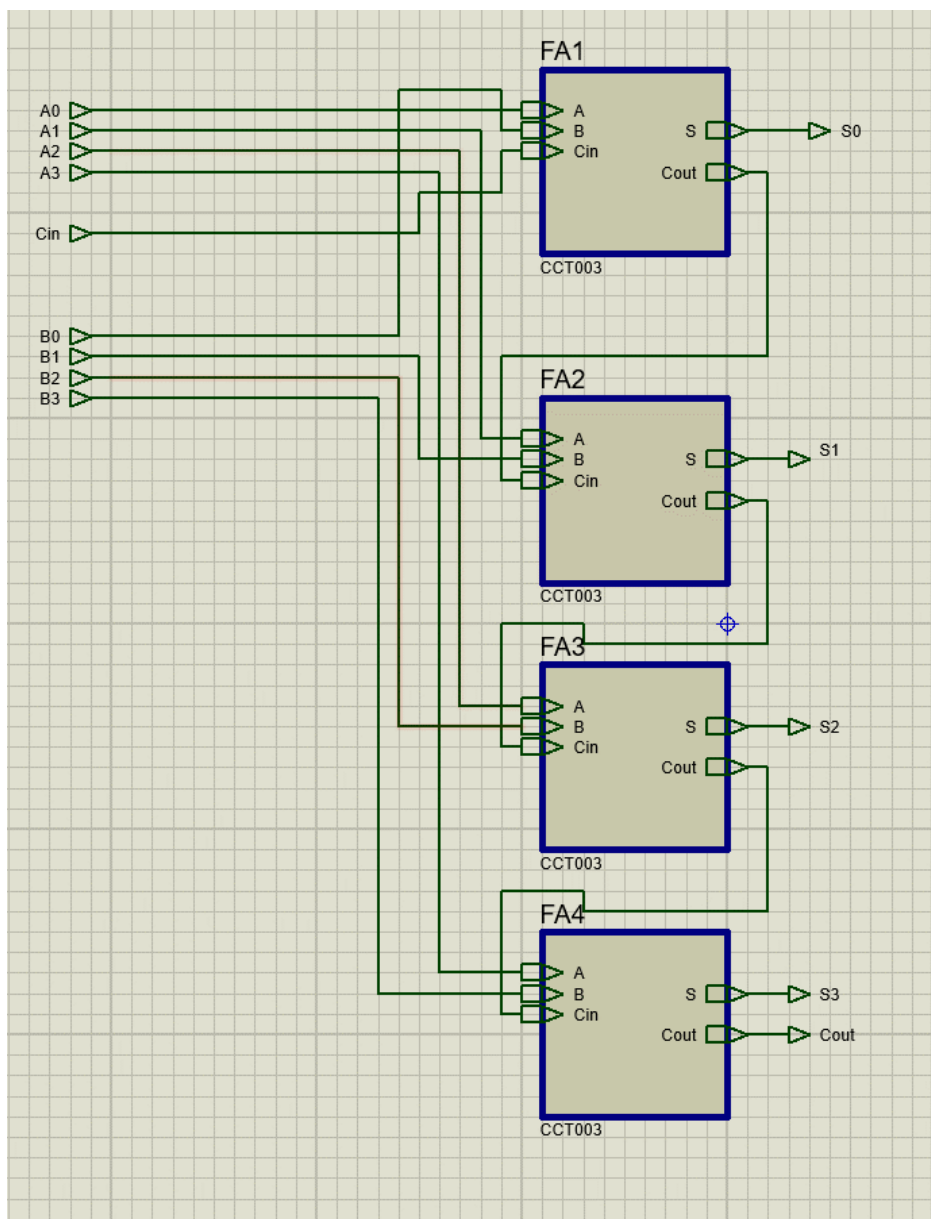
نمونه ی دوم:



مرحله دوم(پیاده سازی روی برد بورده):

شرح آزمایش:

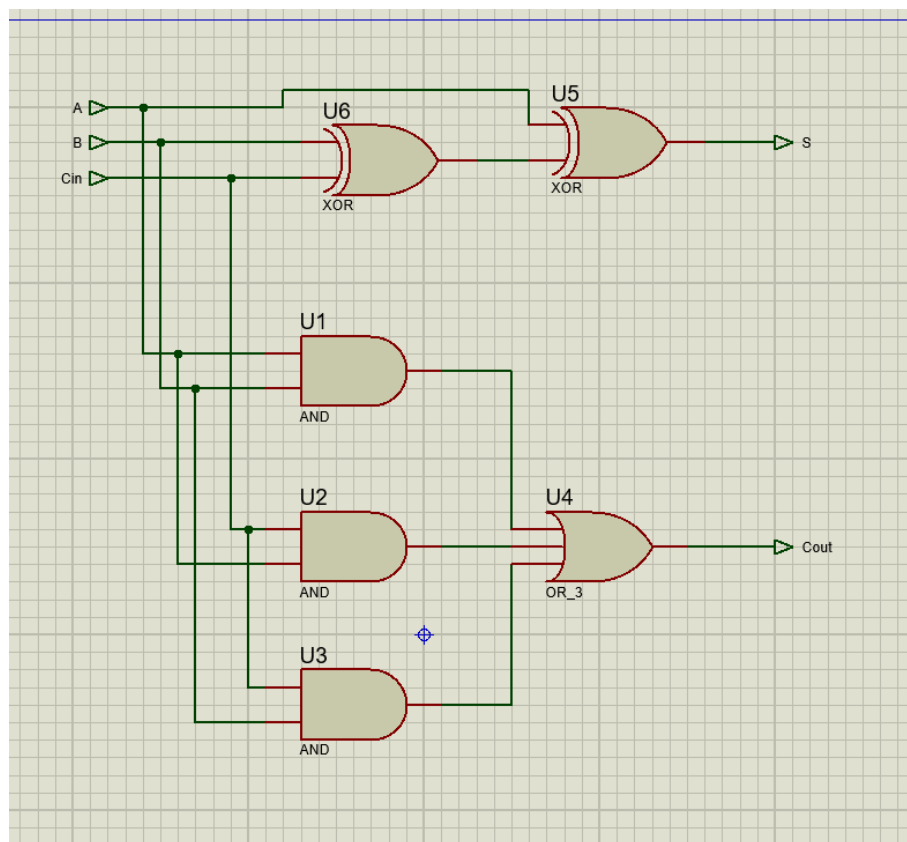
در این بخش قصد نداریم که جمع کننده BCD پیاده سازی کنیم بلکه میخواهیم جمع کننده ۴ بیتی پیاده سازی کنیم یعنی هدف نهایی آزمایش این بخش از پروتئوس است:



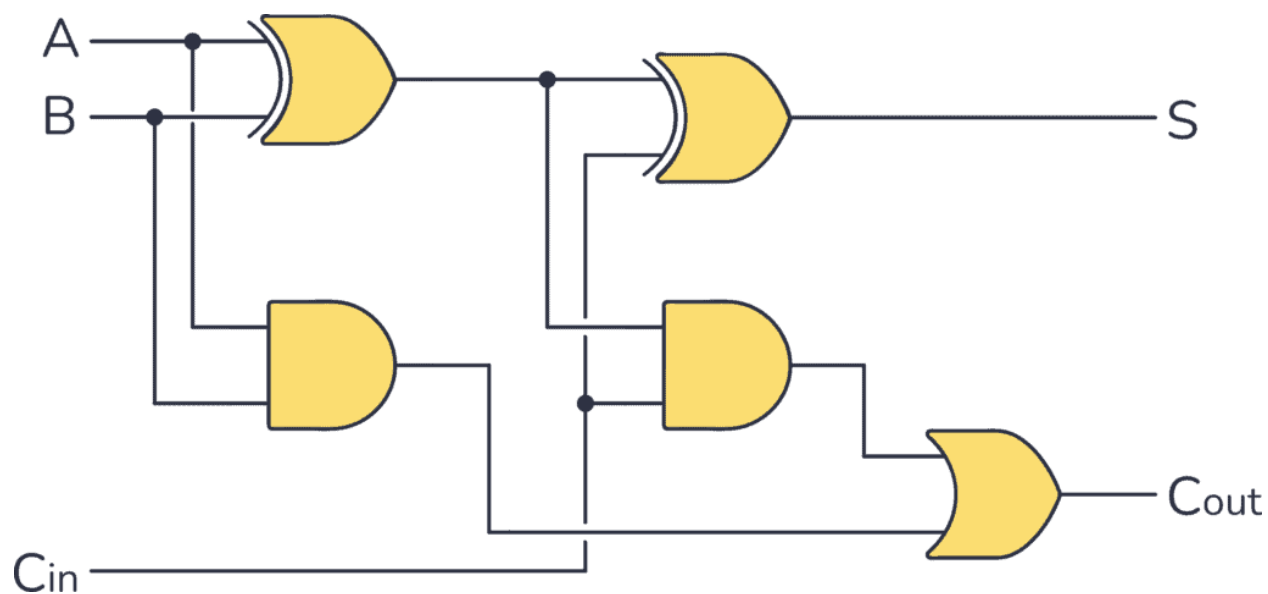
و باقی بخش های آزمایش را پیاده سازی نمیکنیم

برای این پیاده سازی ابتدا باید ۴ فول ادر بسازیم :

در پروتئوس از این ساختار فول ادر استفاده کردیم:

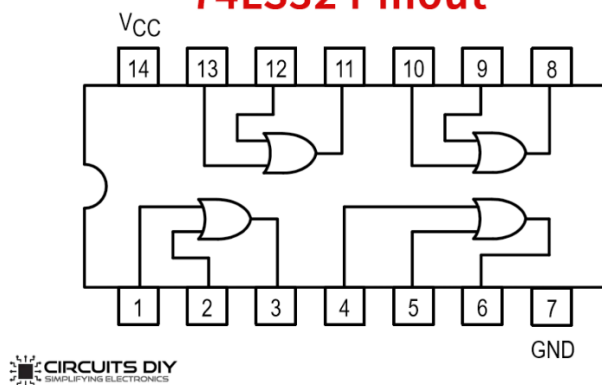


اما چون در آزمایشگاه و پیاده سازی واقعی هر چه تعداد گیت های کمتری استفاده کنیم بهتر است و سیم کشی ها ساده تر است از ساختار زیر برای ساختن فول ادر ها استفاده میکنیم :

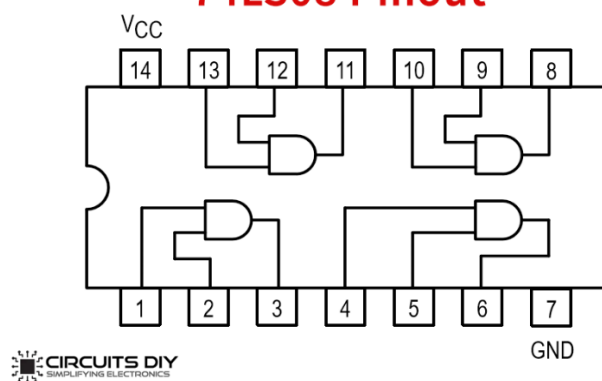


پس منطقاً به گیت های xor or and نیاز داریم که برای XOR کردن از 74LS86 استفاده کردیم , برای OR کردن هم از گیت 74LS32 و برای AND هم از 74LS08 استفاده کردیم دیتا شیت این گیت ها را میتوانید در ادامه مشاهده کنید :

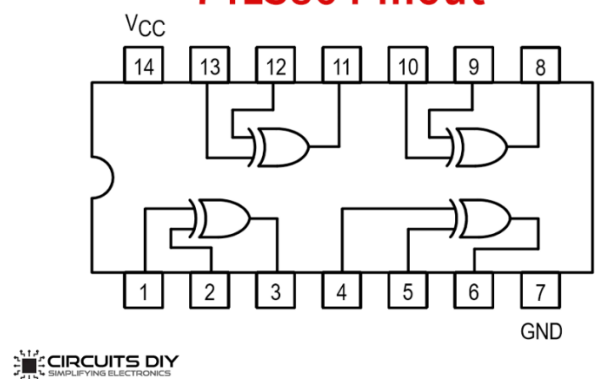
74LS32 Pinout



74LS08 Pinout

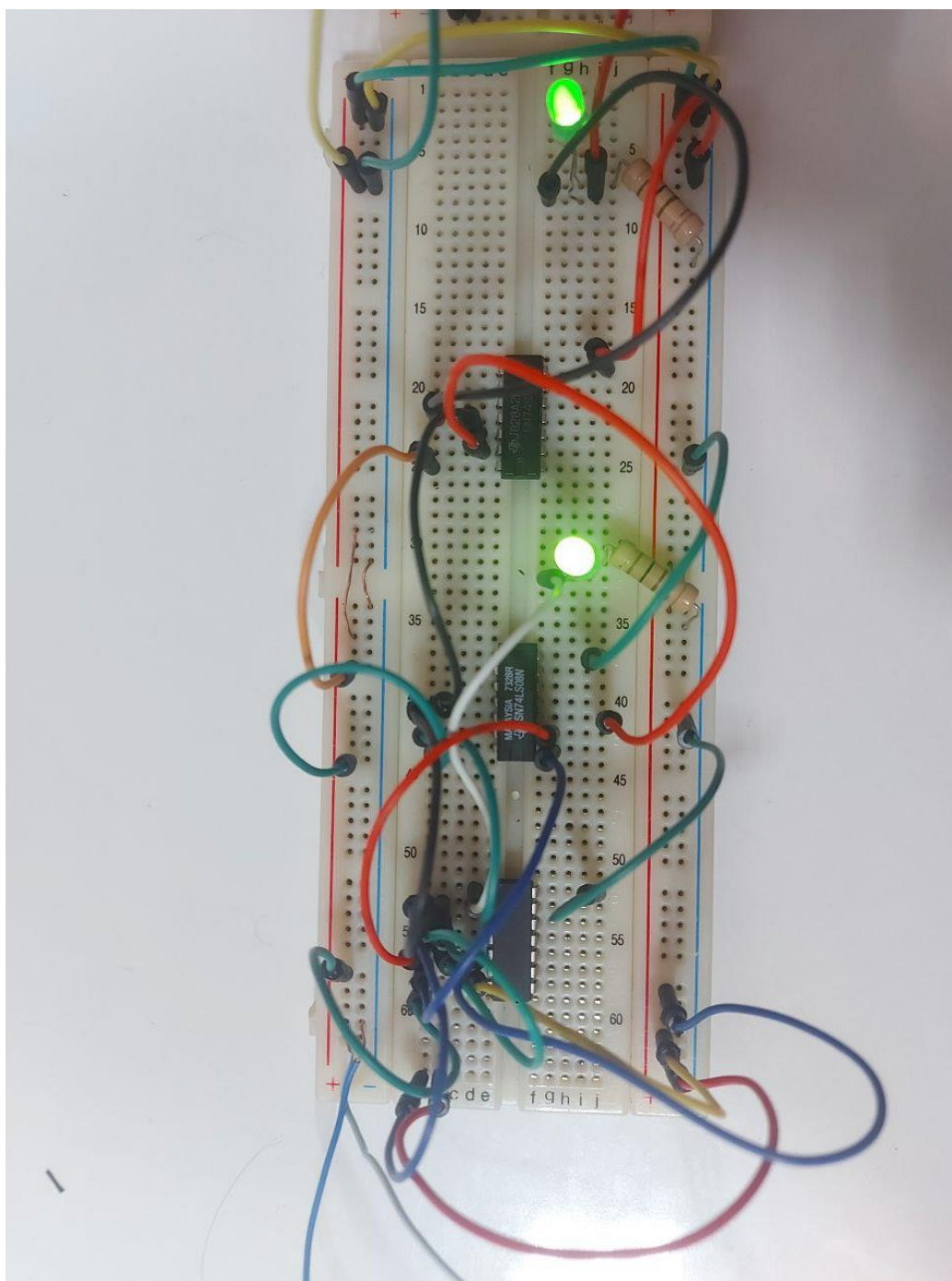


74LS86 Pinout

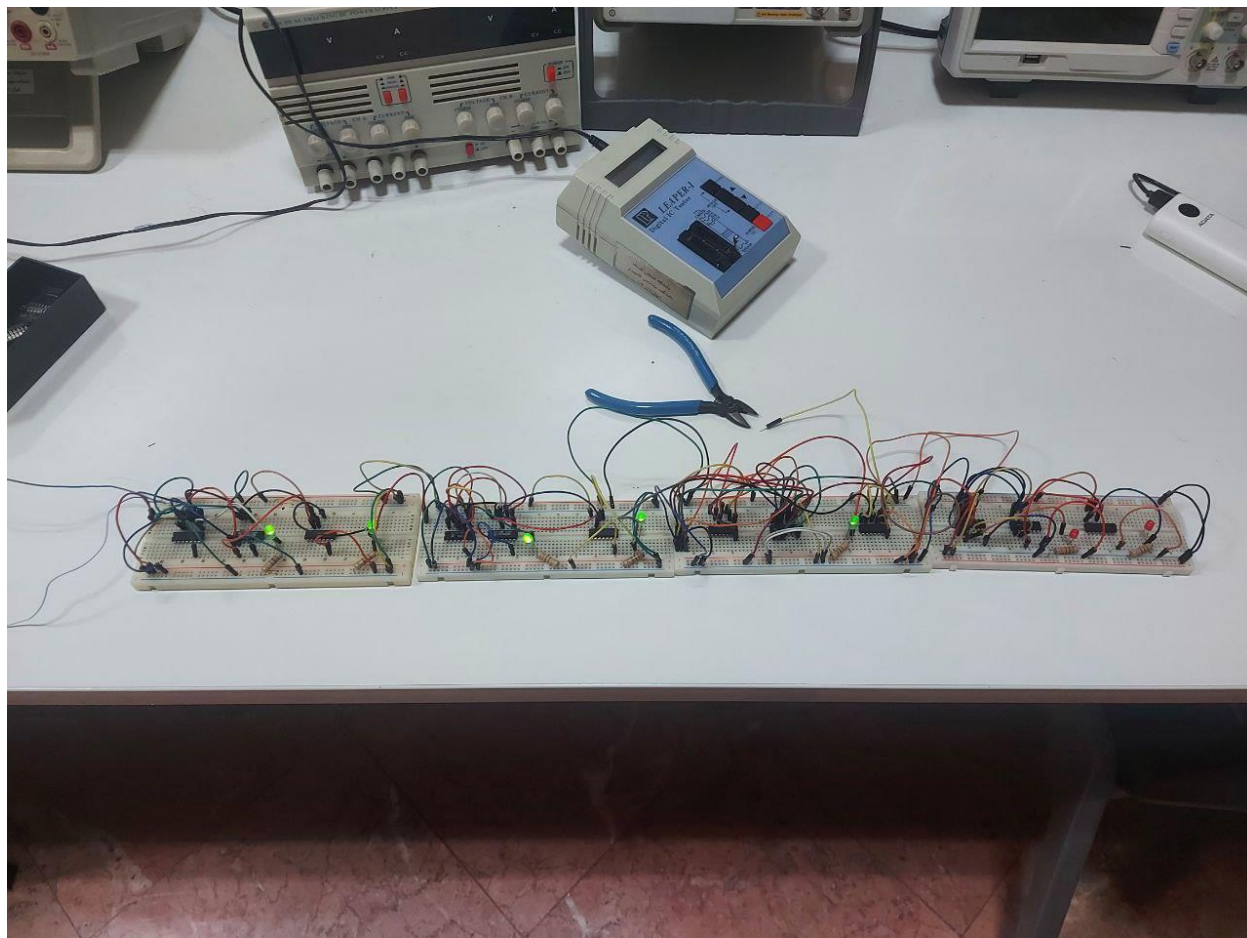


هر فول ادر یکی ازین ۳ را استفاده میکند ابتدا vcc و gnd آن ها وصل میکنیم سپس تمام اتصالات را مطابق مداری که بالاتر ارائه دادم وصل میکنیم و برای کری خروجی و سام خروجی یک LED قرار میدهیم به همراه یک مقاومت ۱۰۰ اهمی و الان فول ادر ما آماده است

شکل فول ادر نهایی :



۳ فول ادر دیگر مشابه همین میسازیم و کنار هم قرار میدهیم
و کری اوت هر کدام را به کری این بعدی میدهیم مطابق شکل زیر در نهایت مدار خروجی
ما به این شکل میشود :



در نهایت خروجی های مختلفی را روی مدار تست کردیم و خروجی ها کاملاً مطابق انتظار
به درستی کار میکنند و مدار صحیح است.