

به نام خدا

گزارش آزمایشگاه مدارهای منطقی

سال تحصیلی 00-01

نویسنده

پارسا شریفی

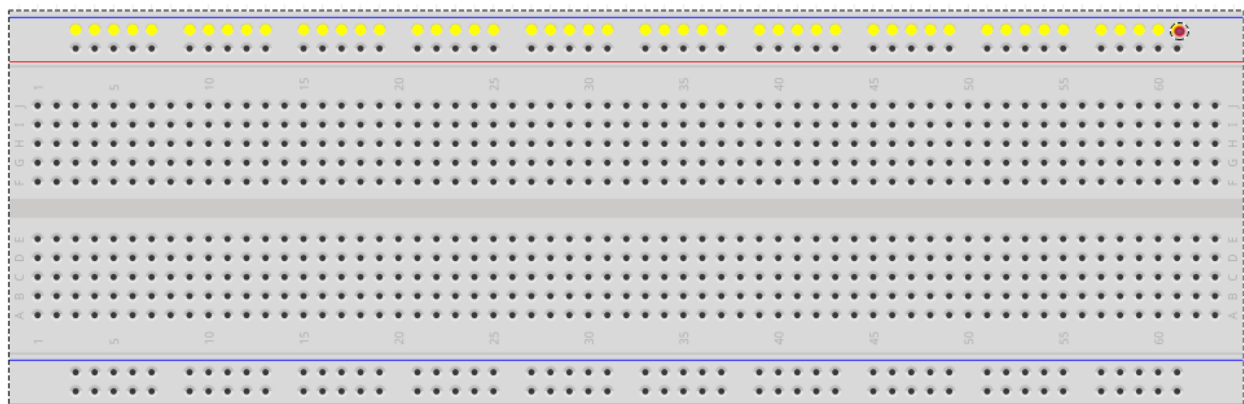
99101762

رسم مدار با Fritzing

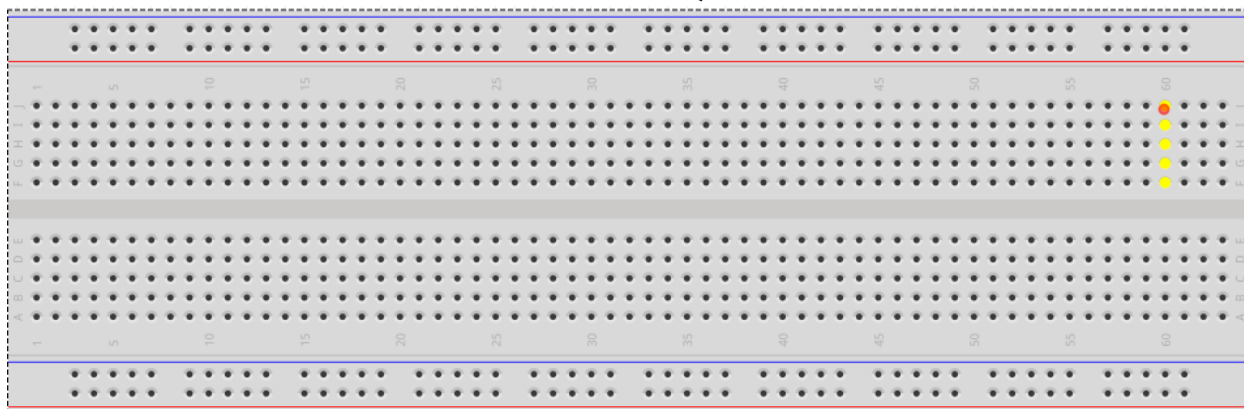
1-1-2- پیدا کردن اتصالات داخل بردبرد خالی

در این قسمت کار با نرم افزار fritzing را تمرین میکنیم هدف از این قسمت آشنایی با نحوه کار بردبرد و کار با نرم افزار است

در دو سطر اول و دو سطر آخر، اتصالات به صورت سطری می‌باشند. این قسمت‌ها برای اتصال منبع برق به بردبرد استفاده می‌شوند:



اما در بخش‌های دیگر؛ هر ستون با بخش‌های پایین خود در ارتباط است:



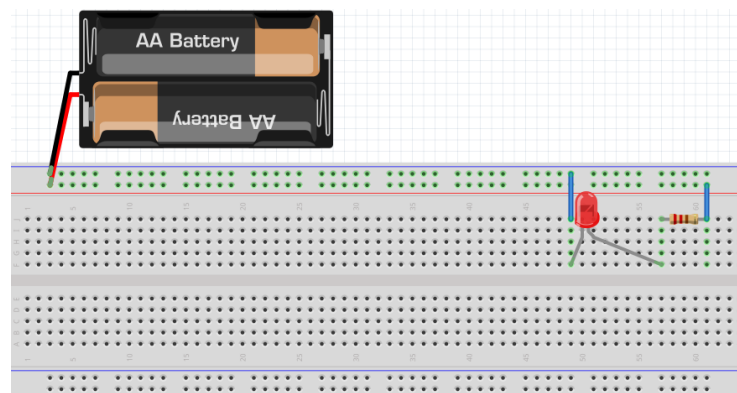
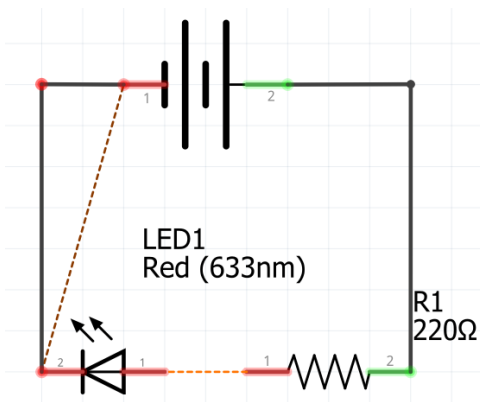
۲-۱-۲- بستن یک مدار ساده شامل باتری و لامپ

این بخش در دستور کار نبود و در کلاس اعلام شد که نیازی به انجام آن نیست ولی آزمایش انجام شد و گزارش آن به این صورت است:

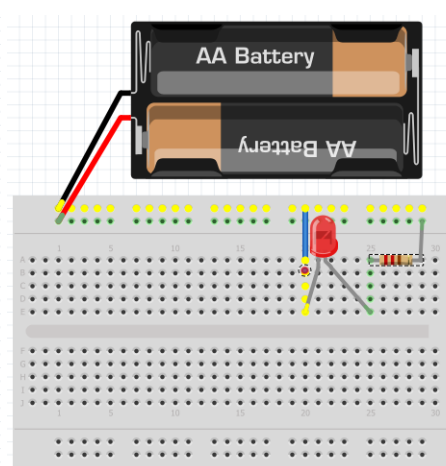
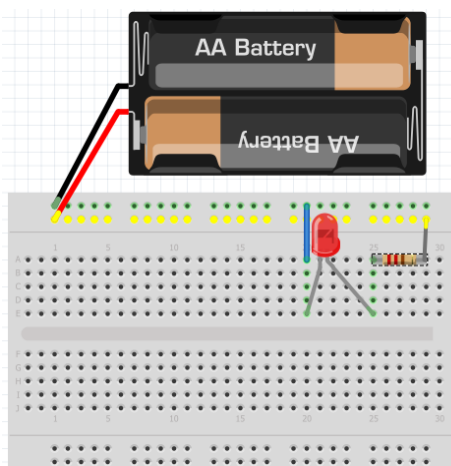
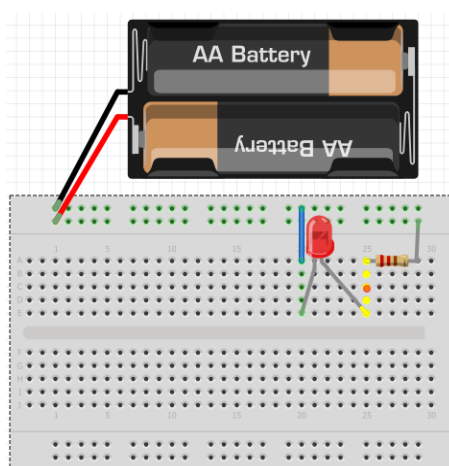
در این قسمت از ما خواسته شد که یک مدار ساده را با استفاده از این نرم افزار پیاده کنیم

نحوه بستن مدار بسیار ساده و بدیهی است

شکل کلی مدار بسته شده و قسمت شماتیک آن به صورت زیر است:



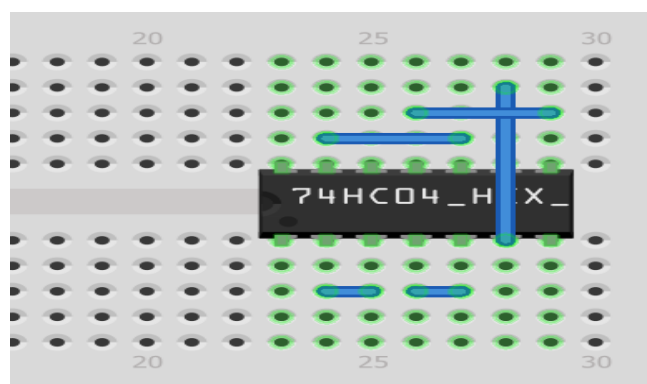
اتصالات بخش‌های مختلف آن نیز به صورت زیر است:



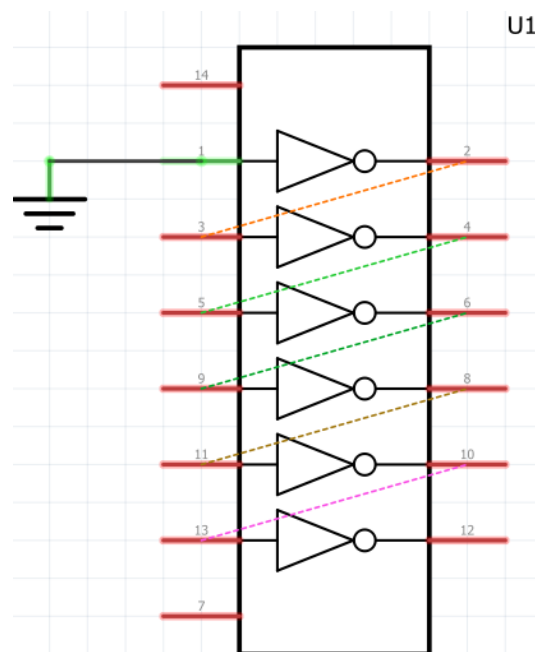
۳-۱-۲- مدار شامل 6 گیت NOT

در این قسمت هدف این است که با استفاده از یک تراشه که حاوی گیت های not است یک مدار طراحی کنیم که 6 بار ورودی را نات بکند

ابتدا تراشه مورد نظر را پیدا می‌کنیم. نام آن در نرم‌افزار 74HC04_HEX_INVERTER می‌باشد. سپس خروجی هر گیت نات را به ورودی گیت بعدی وصل می‌کنیم. مداری مانند شکل روبرو بدست می‌آید



اتصالات داخل مدار (نمای شماتیک) آن در دو تصویر زیر نشان داده شده‌اند:



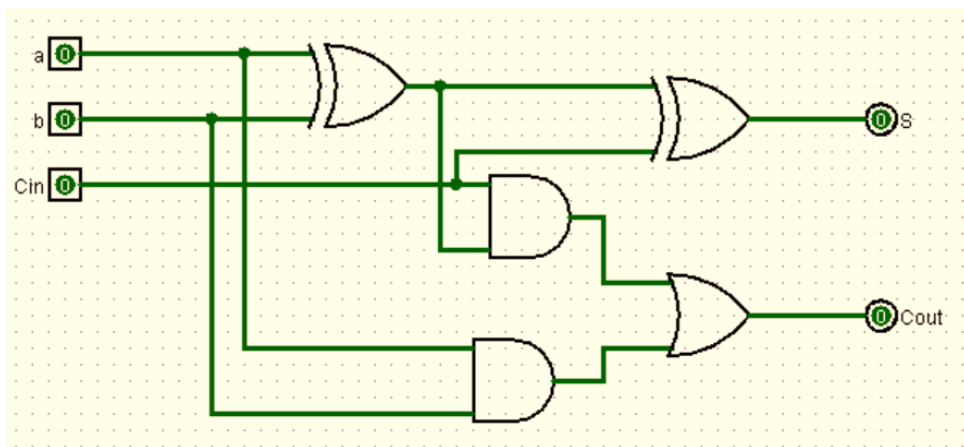
در اینجا تنها خود قطعه به نمایش درآمده و از باتری در مدار صرف نظر شده

ساخت مدار با Logisim

۱-۲-۲- مدار جمع کننده کامل

در این قسمت هدف ما این است که با نرم افزار Logisim

یک فول ادر بسازیم با توجه به روابطی که در درس مدار منطقی خواندیم و همچنین جدول داده شده، مدار ما به همراه جدول خروجی آن به صورت زیر خواهد بود:

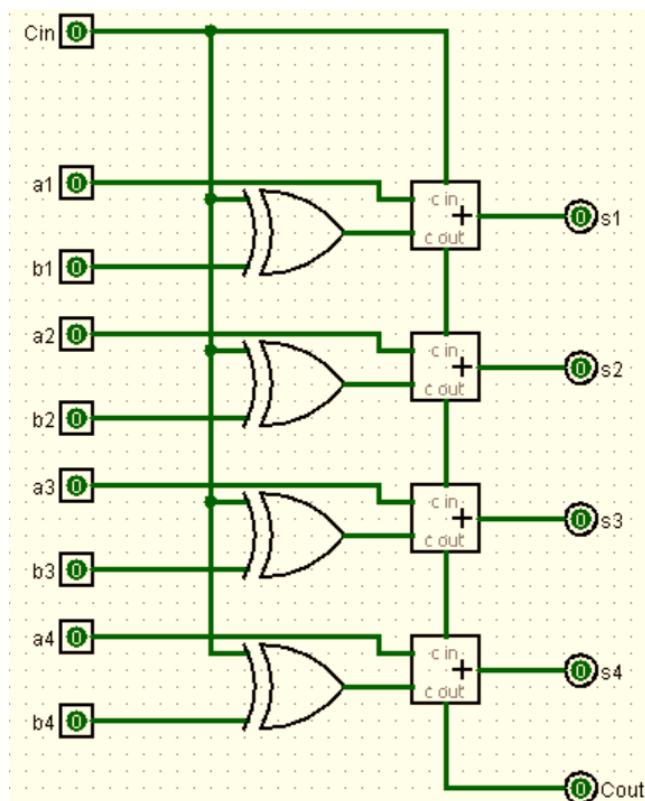


a	b	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

در واقع این مدار را با استفاده از مطالب کتاب موريس مانو انجام دادیم دقیقاً همین شکل با توضیحات در این کتاب قرار دارد

۲-۲-۲- مدار جمع کننده / تفریق کننده

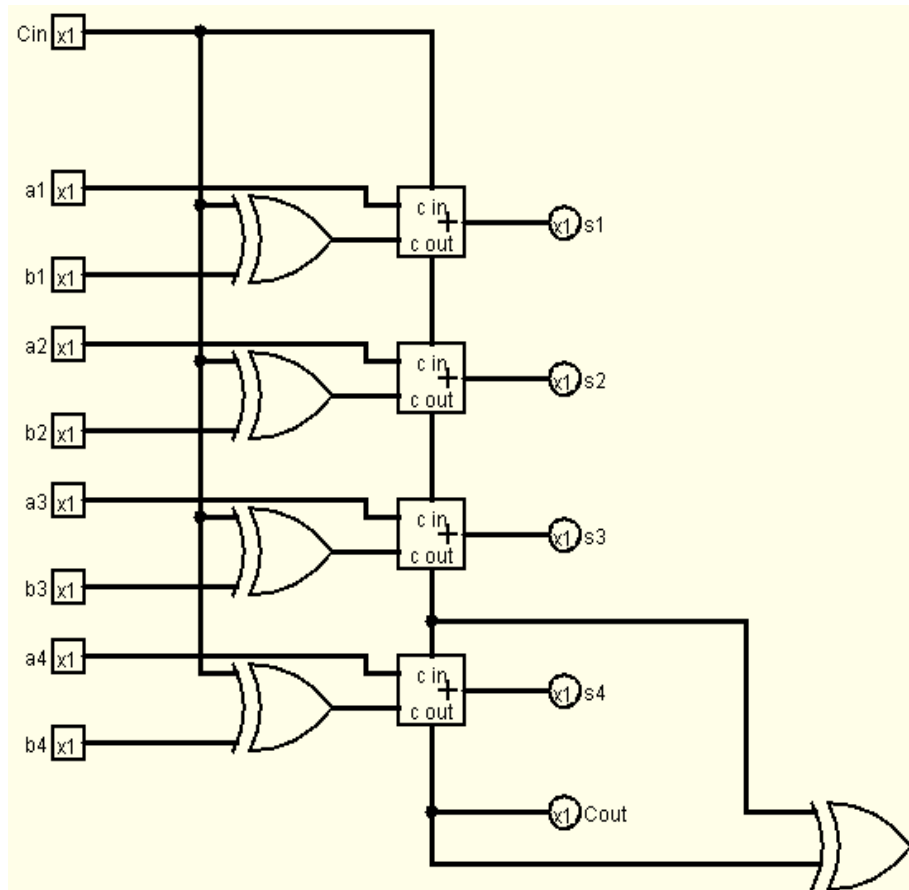
در این قسمت هدف ما طراحی مداری است که بتواند هم جمع کند و هم تفریق را انجام دهد هر کدام از این اعمال وابسته به نوع ورودی ماست که ایا جمع صورت بگیرد و یا تفریق چون اعداد ورودی ما 4 بیتی هستند، به چهار جمع کننده نیاز داریم. همچنین، با استفاده از 4 گیت XOR، در مواقعی که ورودی Cin برابر یک باشد، B را تبدیل به -B می‌کنیم به این صورت که مکمل دوم آن را با استفاده از گیت های XOR به دست آورده و آنرا با A جمع می‌کنیم. شکل مدار و خروجی آن به صورت زیر است:



Cin	a1	a2	a3	a4	b1	b2	b3	b4	s1	s2	s3	s4	Cout
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	1	1	0	0	1	1	0
0	0	0	0	0	0	1	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	1	0	1	0	1	0
0	0	0	0	0	0	1	1	0	0	1	1	0	0
0	0	0	0	0	0	1	1	1	0	1	1	1	0
0	0	0	0	0	1	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	1	1	0	0	1	0
0	0	0	0	0	1	0	1	0	1	0	1	0	0
0	0	0	0	0	1	1	0	0	1	1	0	0	0
0	0	0	0	0	1	1	0	1	1	1	0	1	0
0	0	0	0	0	1	1	1	0	1	1	1	0	0
0	0	0	0	0	1	0	0	0	0	0	0	1	0
0	0	0	0	1	0	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	1	0	0	0	1	1	0
0	0	0	0	1	0	0	1	1	0	0	1	0	1
0	0	0	0	1	0	1	0	0	0	1	0	1	0
0	0	0	0	1	0	1	1	0	0	1	0	0	1
0	0	0	0	1	1	0	0	0	1	0	0	1	0
0	0	0	0	1	1	0	0	1	1	0	0	0	1
0	0	0	0	1	1	0	1	0	1	0	1	1	0
0	0	0	0	1	1	1	0	0	1	0	1	0	1
0	0	0	0	1	1	1	1	0	1	0	1	1	0
0	0	0	0	1	1	1	1	1	1	1	0	1	0

توجه داشته باشید که این مدار نمی‌تواند سرریز (Overflow) را تشخیص دهد. برای تشخیص سرریز، نیاز به یک گیت XOR دیگر در خروجی داریم که کری از فول ادر سوم را با Cout در شکل بالا

XOR بکند تا خروجی V به دسن اید که با اسفاده از آن میتوان سرریز را شناسایی کرد.



خروجی XOR اضافه شده همان V میشود که با آن میتوان سرریز را شناسایی کرد

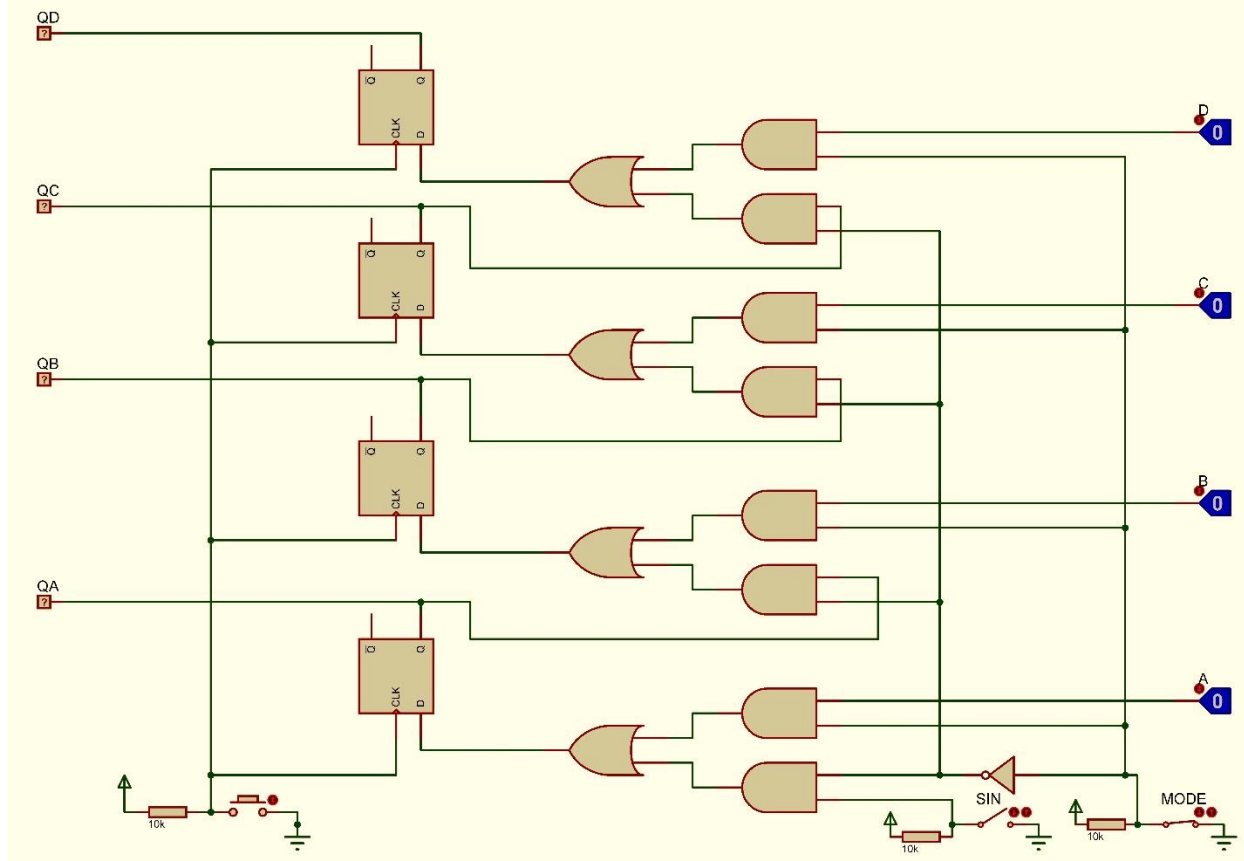
۳-۲- ساخت مدار با Proteus

در کلاس اعلام شد که این بخش از گزارش از دستور کار حذف شده این قسمت مربوط به بخش carry generator است.

آزمایش دوم: شیفتر رجیسترها

بخش اول: طراحی و ساخت یک شیفتر رجیستر

3-1-1 در این قسمت از ما میخواهد که یک شیفتر رجیستر بسازیم مدار طراحی شده به شکل زیر می باشد:



در این مدار یک شیفت رجیستر با چهار D-FlipFlop پیاده‌سازی شده است. چهار ورودی A تا D که روی شکل نام‌گذاری شده‌اند به ترتیب به خروجی‌های Q_A تا Q_D مرتبط می‌شوند. کلید Mode تعیین‌کننده حالت مدار می‌باشد؛ به این صورت که اگر بسته باشد، یک شیفت به بالا (در واقع ما شیفت به بالا نداریم و این اصطلاح فقط در صورت سوال آمده و بسته به این که کدام بیت را بیت پر ارزش بگیریم می‌تواند شیفت به چپ و یا شیفت به راست باشد در این حالت اگر A را بیت با ارزش تر بگیریم منظور از شیفت به بالا شیفت به راست می‌شود) انجام شده و ورودی Sin وارد خروجی می‌شود. در غیر این صورت چهار ورودی A تا D به صورت موازی وارد خروجی می‌شوند.

3-1-2- در این قسمت از ما خواسته شده که مقدار داده شده را درون شیفت رجیستر ثبت کنیم

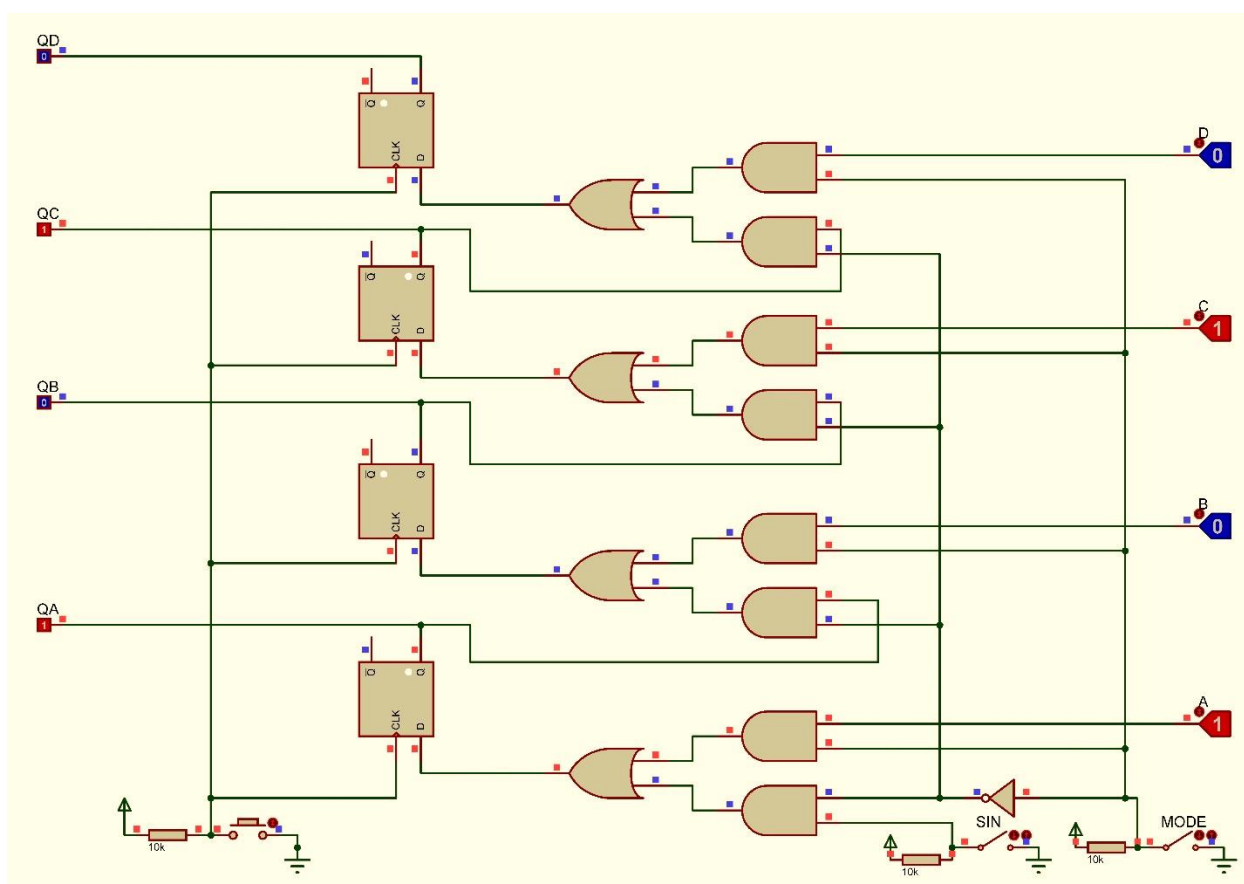
برای ذخیره مقدار 1010، دو روش داریم:

روش اول: می‌توانیم کلید Mode را ببندیم، و با تغییر ورودی S_{in} به مقادیر 0، 1، 0، 1 و اعمال کردن کلاک بعد از هر تغییر خروجی 1010 را داشته باشیم. در واقع ورودی را به طور مداوم شیفت می‌دهیم تا به حالت مورد نظرمان برسد

روش دوم: می‌توانیم کلید Mode را باز کنیم و ورودی‌ها را به صورت مقابل مقداردهی کنیم و از بارگذاری موازی استفاده کرده و مقدار خروجی‌ها را تعیین می‌کنیم:

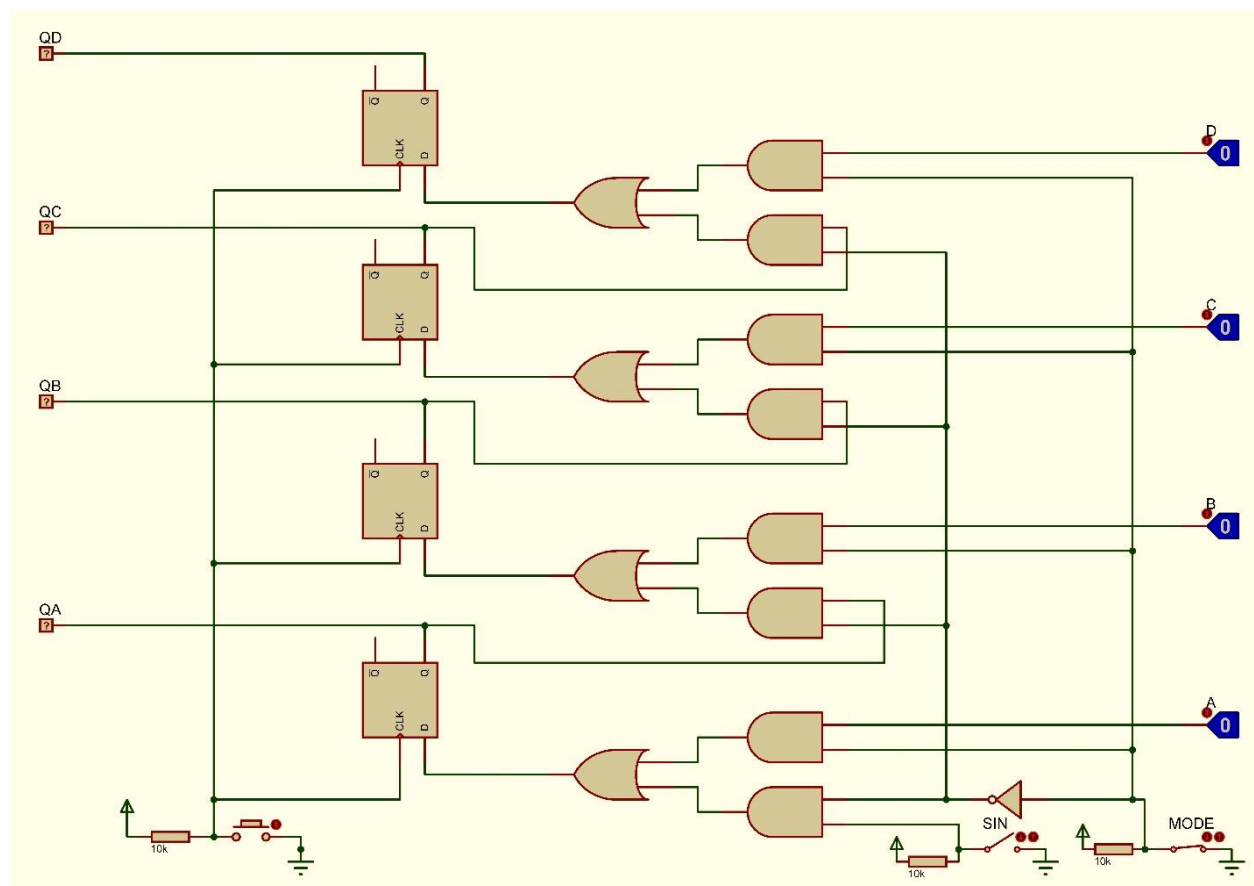
A=1 B=0 C=1 D=0

اکنون با اعمال یک کلاک پالس، دقیقاً همین خروجی را خواهیم داشت:

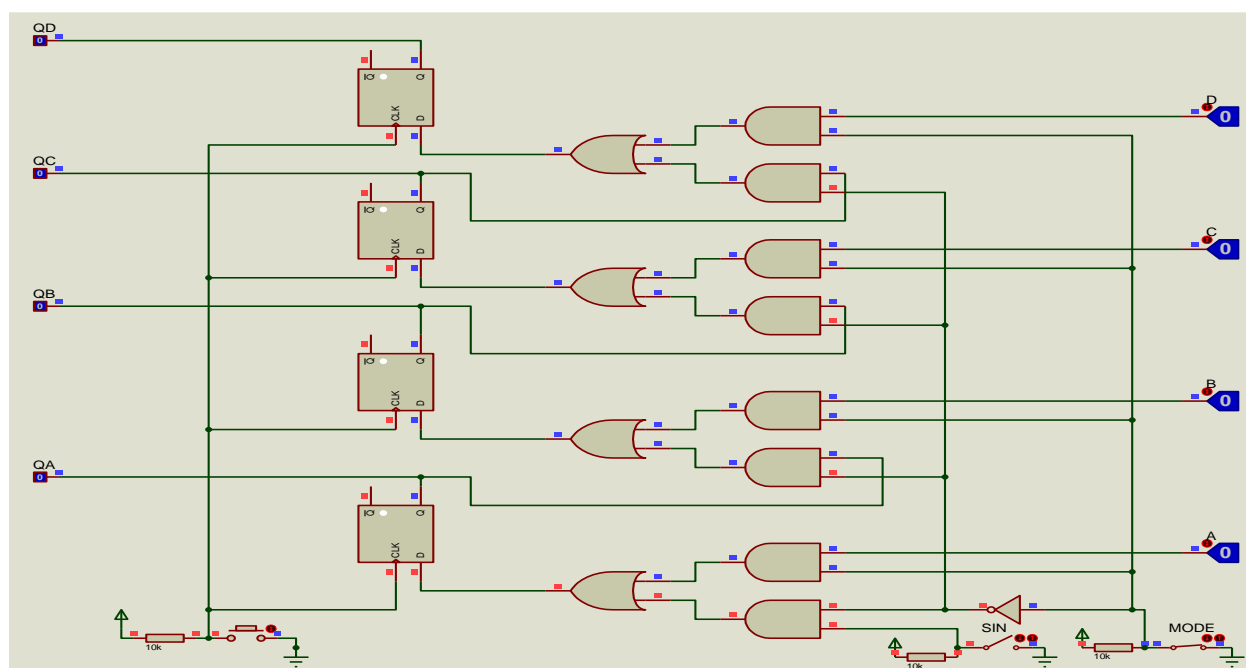


۳-۲-۳- در این قسمت از ما خواسته شده که یک شیفت رجیستر بسازیم که به سکت راست شیفت بدهد با فرض این که بیت A بیت پرارزش تر باشد همانطور که در قسمت بالاتر توضیح داده شد خود شکل سوال میتواند جواب این سوال باشد

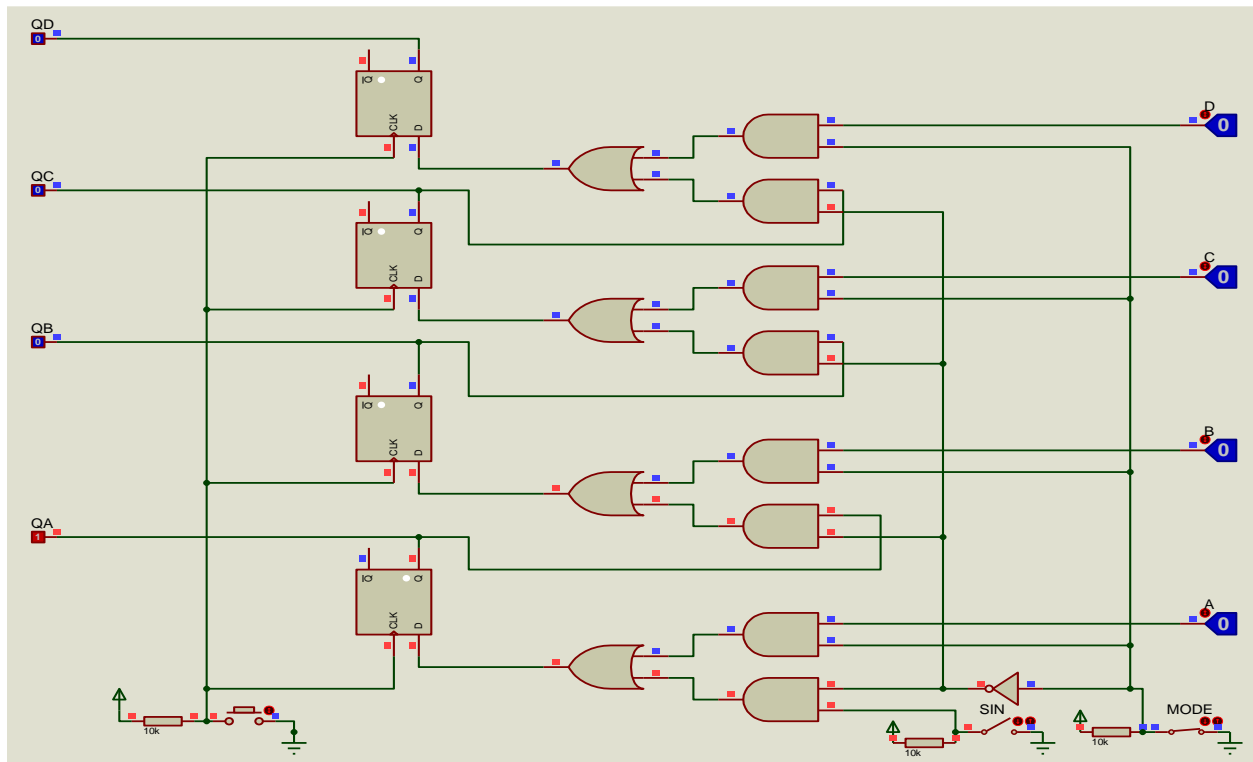
در واقع منظور از سوال از شیفت به بالا در صورتی که A بیت پرارزش باشد شیفت به راست است و ورودی از سمت چپ وارد میشود درست مانند شکل زیر:



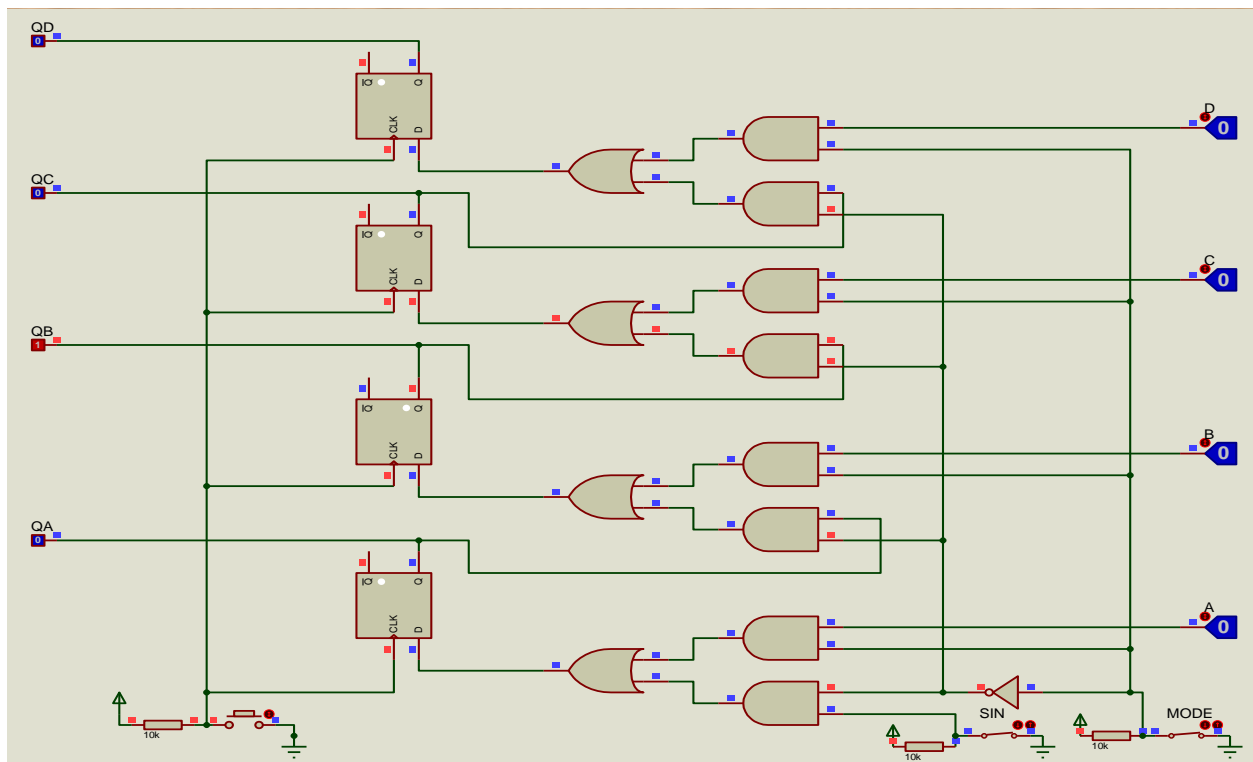
حال مدار را آزمایش میکنیم و شکل ها را بعد از هر پالس میبینیم
مرحله 1 شروع کار مدار که همه خروجی ها در حالت 0 هستند



حال در این حالت با زدن فقط یک کلاک ورودی 1 وارد شده و در مقدار A قرار میگیرد:



حال ورودی را با بستن کلید 0 میکنیم که متوجه حرکت بیت 1 شده باشیم و دوباره کلاک میزنیم:

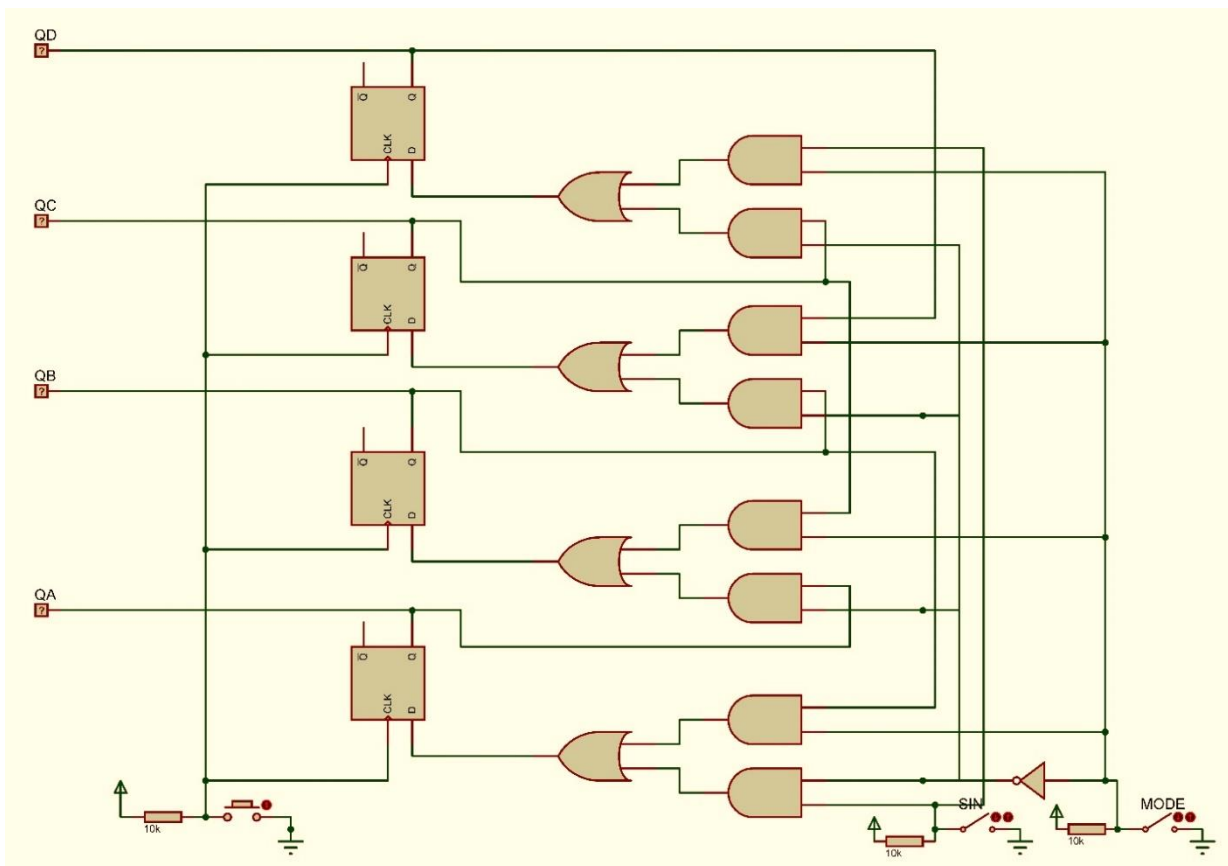


[illegible]

ابتدا با توجه به ویژگی خواسته شده، معادله ورودی هر فلیپ‌فلاپ را بدست می‌آوریم در این قسمت از کتاب مدار منطقی موریس مانو استفاده شده و روابط را با استفاده از آن کتاب نوشتیم:

$$D_d = Mode' \cdot Q_c + Mode \cdot S_{in}$$

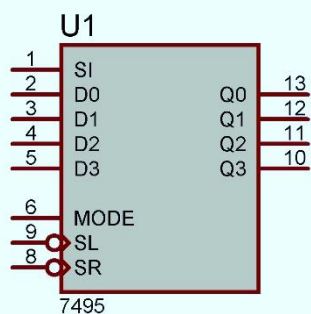
اکنون با توجه به معادلات فوق، مدار را رسم می‌کنیم (برای رسم این مدار از کتاب مانو استفاده شده با استفاده از بخش شیفتر رجیستر یونیورسال ولی تغییری روی آن اعمال شده تا قابلیت بارگذاری موازی نداشته باشند):



بخش دوم: استفاده از شیفتر رجیستر آماده

3-2-1- ابتدا تراشه 7495 را مورد بررسی قرار می‌دهیم. شکل آن به صورت زیر می‌باشد:

در این قسمت ما با استفاده از این تراشه شیفتر رجیستر به سمت راست می‌سازیم درواقع قسمت قبلی را اینبار کمی ساده‌تر انجام می‌دهیم



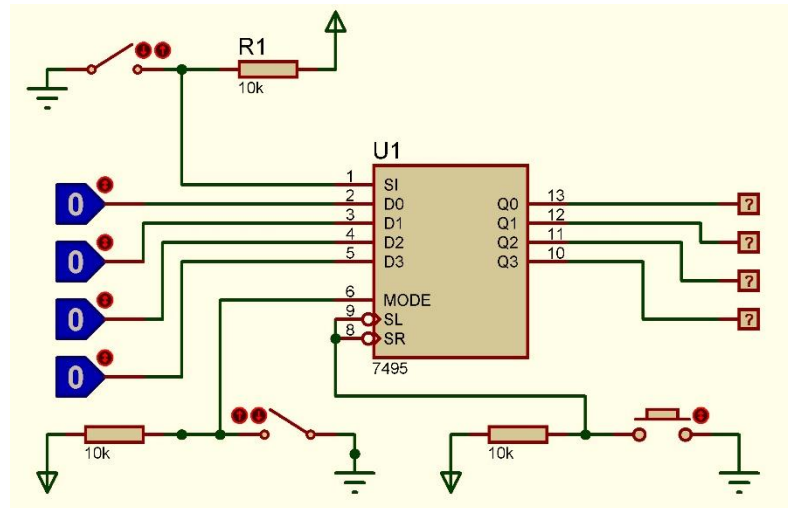
ورودی‌های D0 تا D3 به عنوان ورودی موازی و ورودی SI به

عنوان ورودی شیفتر استفاده می‌شوند. همچنین ورودی Mode تعیین

کننده حالت مدار است: در صورت یک بودن مدار خاصیت پذیرش ورودی موازی و در صورت صفر بودن مدار خاصیت شیفتر رجیستر را دارد.

دو ورودی SR و SL نیز ورودی‌های کلاک جداگانه برای شیفتر و ورودی موازی هستند؛ که در اینجا چون ما نیازی به جدا بودن این دو کلاک نداریم، آن‌ها را به یک کلاک پالس متصل خواهیم کرد.

با توجه به توضیحات فوق، مدار نهایی خواسته شده به صورت زیر می‌باشد:

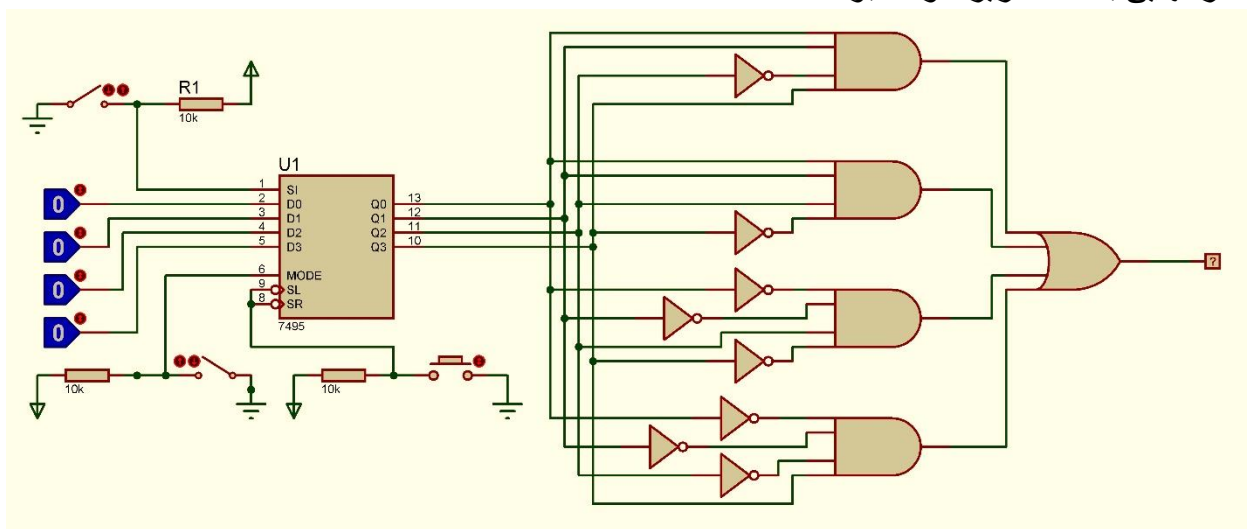


3-2-2- در این قسمت از ما خواسته شده که شیفت رجیستر ما بتواند بعضی از حالات در خروجی را شناسایی کند که حالات مورد نظر در متن گذارش و شیوه انجام ان آمده

برای اینکه مدار ما هر چهار رشته را در ورودی تشخیص دهد، باید از 4 گیت AND با چهار ورودی و یک گیت OR با چهار ورودی استفاده کنیم. خروجی رجیستر را طوری به ورودی AND ها متصل می‌کنیم که در صورت رخ دادن هرکدام از رشته‌ها، خروجی آن گیت 1 شود در واقع باید با استفاده از تک تک خروجی ها مدار را طوری تکمیل کنیم که حالت های مورد نظر را پیدا کند. نهایتا خروجی همه گیت‌های AND را به ورودی OR متصل می‌کنیم تا در صورت وقوع هرکدام از حالت ها خروجی نهایی نیز یک شود. ورودی گیت‌های AND به صورت زیر هستند:

- 1) $Q_0 Q_1 Q_2' Q_3 \equiv 1101$
- 2) $Q_0 Q_1 Q_2 Q_3' \equiv 1110$
- 3) $Q_0' Q_1' Q_2 Q_3' \equiv 0010$
- 4) $Q_0' Q_1' Q_2' Q_3 \equiv 0001$

مدار نهایی به شکل زیر خواهد بود:



پایان