

بسمه تعالی



گزارش کار سوم آزمایشگاه مدارهای منطقی

شمارنده

استاد:

دکتر شاهین حسابی

نویسنده:

امیررضا آذری

99101087

دانشگاه صنعتی شریف

تابستان ۱۴۰۱

فهرست

هدف	۳
بخش اول _ شمارنده دودویی آسنکرون	۳
۱.۱ _ ساخت شمارنده بالا/پایین شمار آسنکرون	۳
۱.۲ _ افزودن امکان مقدار دهی موازی به شمارنده بالا/پایین شمار آسنکرون	۷
بخش دوم _ شمارنده دودویی سنکرون	۱۳
بخش سوم _ شمارنده BCD	۱۸
نتیجه گیری:	۲۶
منابع و مراجع:	۲۷

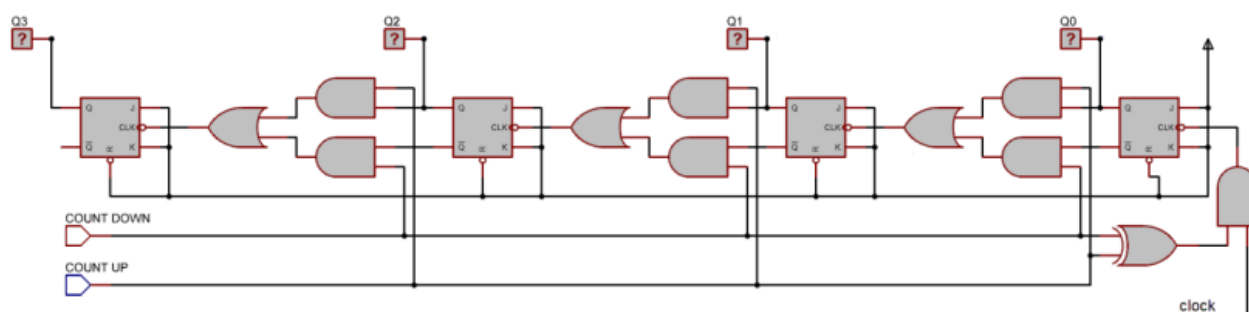
هدف

هدف از این آزمایش، آشنایی با نحوه کارکرد انواع شمارنده ها است. در بخش اول، با شمارنده دودویی آسنکرون، در بخش دوم با شمارنده دودویی سنکرون و در بخش سوم با تراشه ۷۴۱۹۰ و شمارنده BCD آشنا می‌شویم. همه بخش های این آزمایش را با نرم افزار Proteus انجام می‌دهیم.

بخش اول – شمارنده دودویی آسنکرون

۱.۱ – ساخت شمارنده بالا/پایین شمار آسنکرون

مطابق شکل زیر ، با استفاده از چهار فلیپ فلاپ JK یا T، یک شمارنده بالا/پایین شمار آسنکرون بسازید.



شکل ۱. شمارنده بالا/پایین شمار آسنکرون

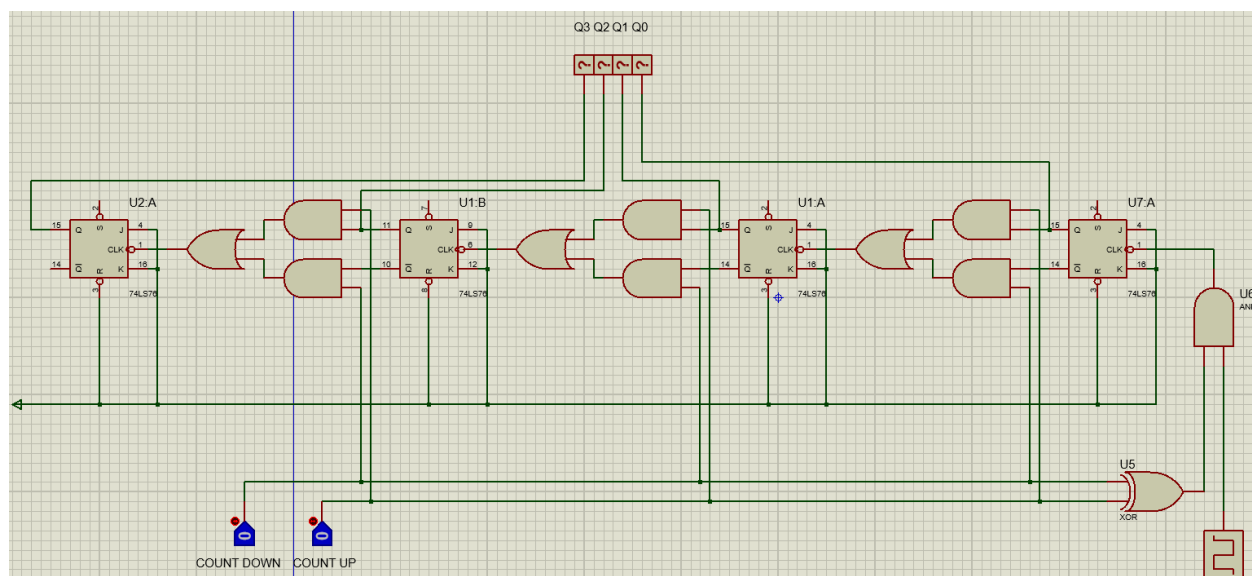
تئوری آزمایش:

در این آزمایش با استفاده از نرم افزار proteus و استفاده از گیت های AND, XOR, OR و فلیپ فلاپ JK، یک شمارنده بالا/پایین شمار آسنکرون مطابق شکل ۱ طراحی می‌کنیم.

گزارش کار بخش ۱.۱:

مطابق شکل ۲، مدار خواسته شده را رسم کرده‌ایم. در این مدار اگر COUNT UP یا COUNT DOWN یک باشد به ترتیب شمارش رو به بالا و شمارش رو به پایین انجام می‌شود. به طور مثال اگر $COUNT\ UP = 1$ ، اعداد نشان داده شده به ترتیب ۰۰۰۱، ۰۰۱۰، ۰۰۱۱، ۰۱۰۰، ...، ۱۱۱۱ خواهند بود. و اگر $COUNT\ DOWN = 1$ ، اعداد گفته شده، بالعکس نشان داده خواهند شد (۱۱۱۱، ۱۱۱۰، ۱۱۰۱، ...، ۰۰۰۰). همچنین در هر مرحله ای که باشیم، با عوض کردن COUNT UP یا COUNT DOWN، می‌توانیم به جلو یا عقب حرکت نماییم.

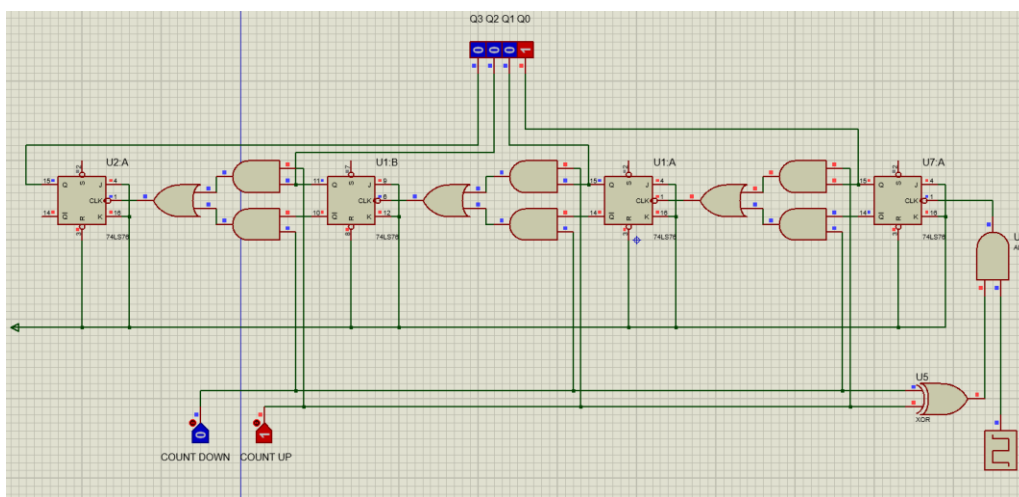
کلاک فلیپ فلاپ های بعدی نیز بر اساس تغییر خروجی فلیپ فلاپ قبلی و نوع شمارش (بالا یا پایین) با استفاده از گیت های AND و OR و XOR ساخته شده است.



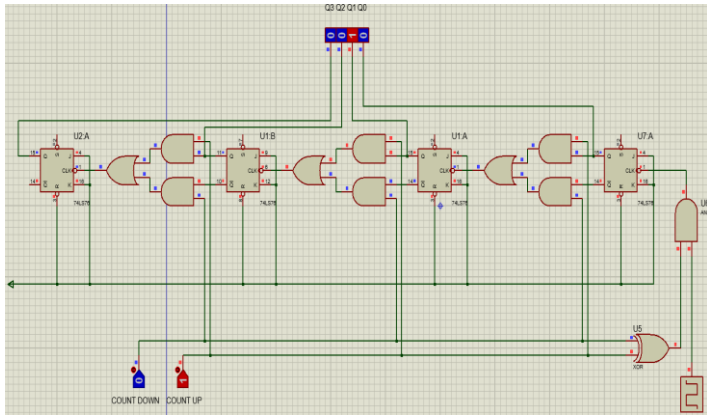
شکل ۲. مدار طراحی شده

به بررسی مثالی می پردازیم.

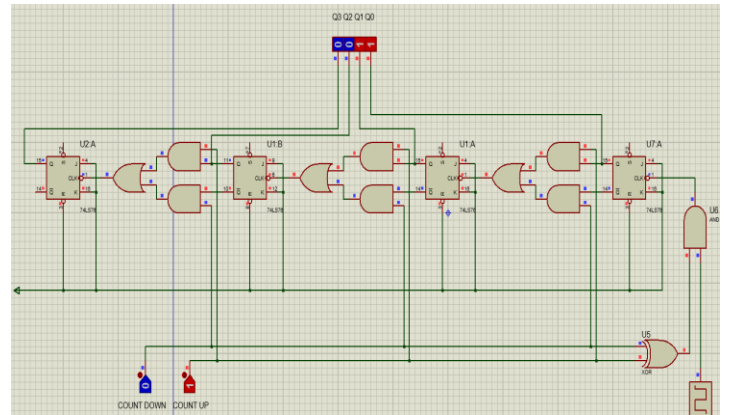
در ابتدا با یک کردن COUNT UP، مشاهده می کنیم با هر کلاکی که می خورد، عدد نشان داده شده یکی افزایش می یابد. برای مثال تا عدد ۶ (۰۱۱۰)، COUNT UP را یک نگه می داریم.



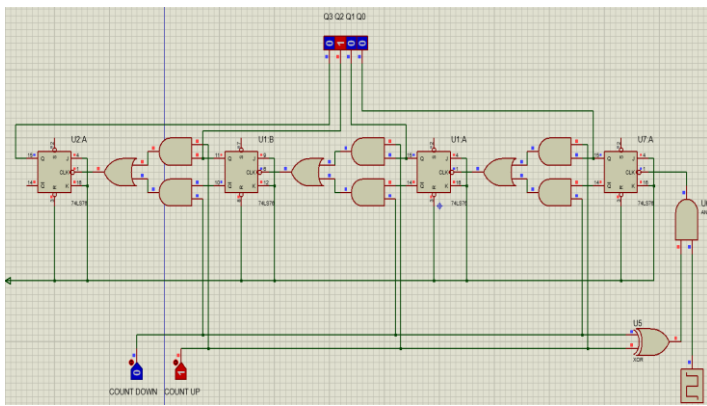
شکل ۳. نشان داده شدن عدد ۱ (۰۰۰۱) پس از کلاک



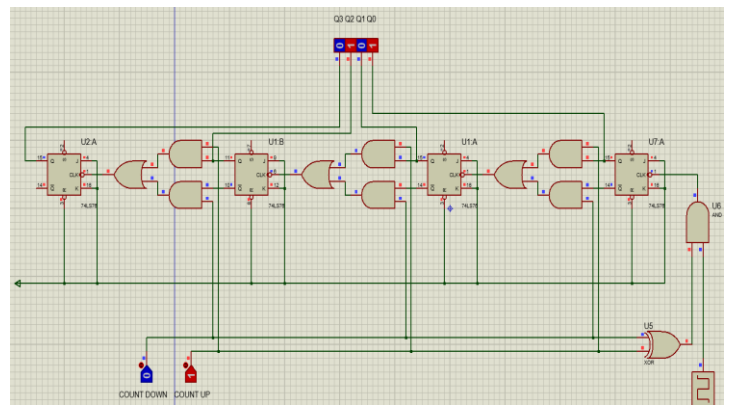
شکل ۴. نشان داده شدن عدد ۲ (۰۰۱۰) پس از خوردن کلاک



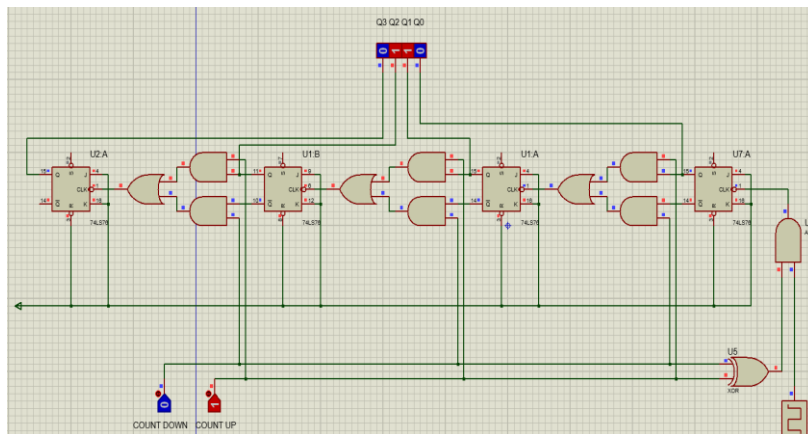
شکل ۵. نشان داده شدن عدد ۳ (۰۰۱۱) پس از خوردن کلاک



شکل ۶. نشان داده شدن عدد ۴ (۰۱۰۰) پس از خوردن کلاک

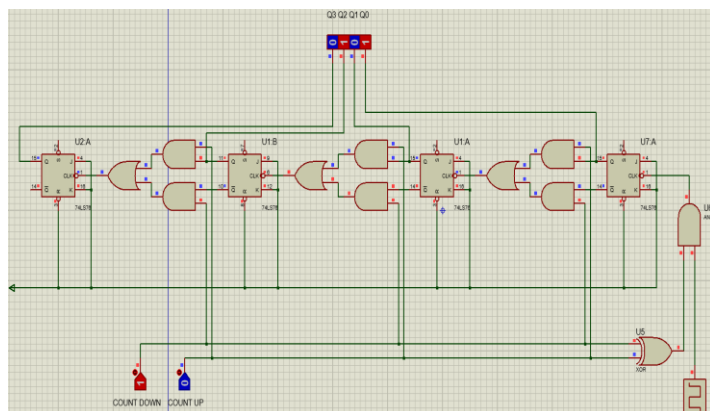


شکل ۷. نشان داده شدن عدد ۵ (۰۱۰۱) پس از خوردن کلاک

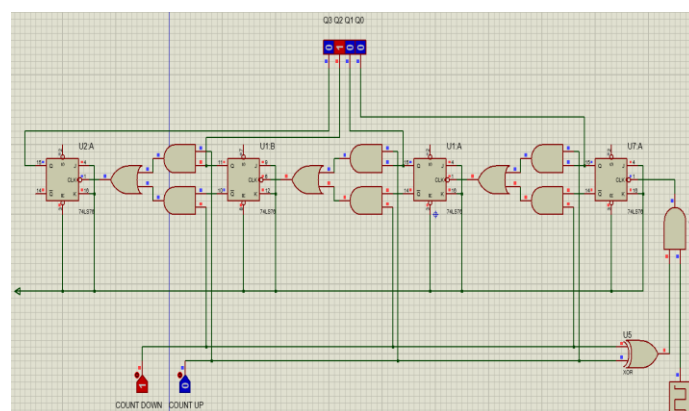


شکل ۸. نشان داده شدن عدد ۶ (۰۱۱۰) پس از خوردن کلاک

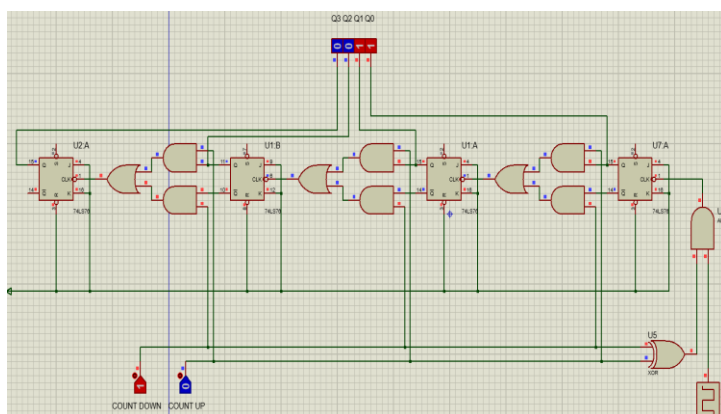
حال، COUNT UP را صفر کرده و COUNT DOWN را یک می‌نماییم. مشاهده می‌شود که بعد از خوردن هر کلاک، عدد نشان داده شده، یک عدد کمتر می‌شود.



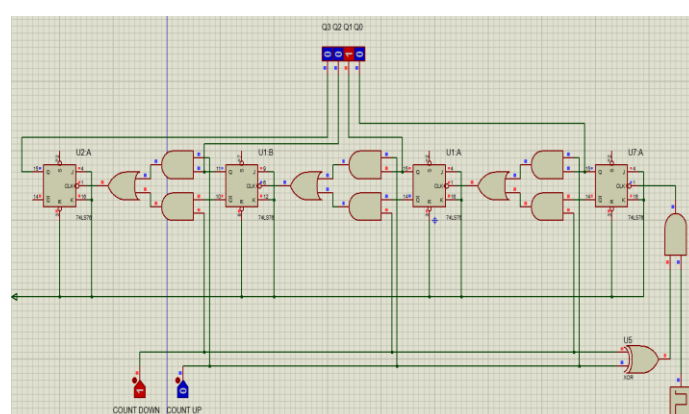
شکل ۹. نشان داده شدن عدد ۵ (۰۱۰۱) پس از خوردن کلاک



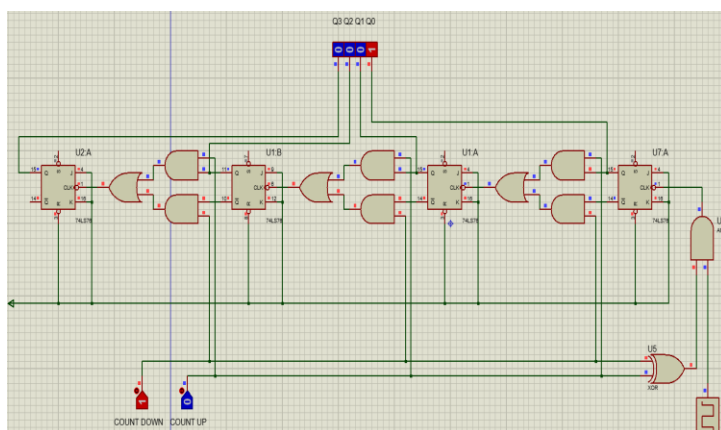
شکل ۱۰. نشان داده شدن عدد ۴ (۰۱۰۰) پس از خوردن کلاک



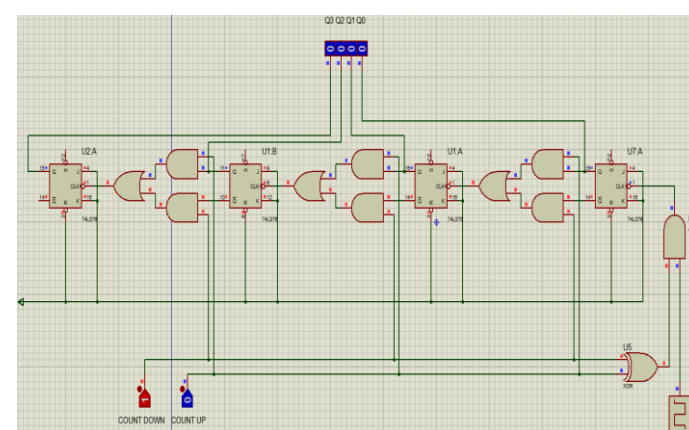
شکل ۱۱. نشان داده شدن عدد ۳ (۰۰۱۱) پس از خوردن کلاک



شکل ۱۲. نشان داده شدن عدد ۲ (۰۰۱۰) پس از خوردن کلاک



شکل ۱۳. نشان داده شدن عدد ۱ (۰۰۰۱) پس از خوردن کلاک

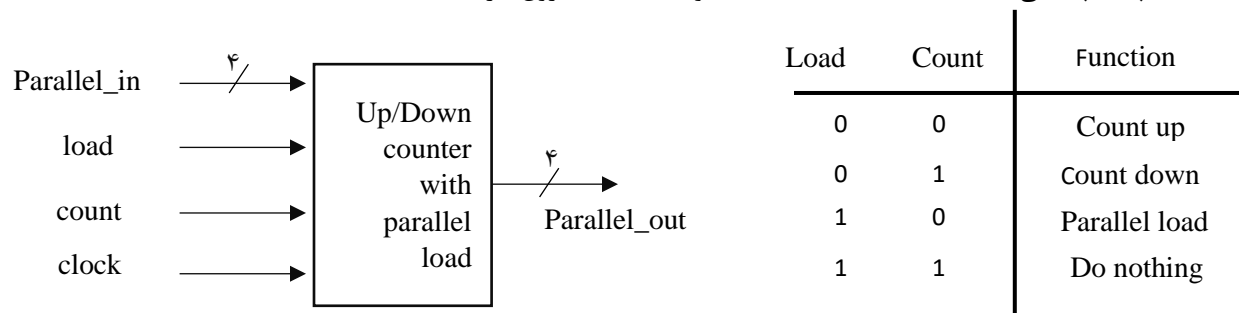


شکل ۱۴. نشان داده شدن عدد ۰ (۰۰۰۰) پس از خوردن کلاک

لازم به ذکر است برای زدن کلاک، از CLOCK در بخش COMPONENT MODE استفاده شده است.

۱.۲_ افزودن امکان مقدار دهی موازی به شمارنده بالا/پایین شمار آسنکرون

با اعمال تغییرات لازم، طبق شکل ۱۵ امکان مقداردهی موازی را به شمارنده خود اضافه کنید. برای این منظور می‌توانید از فلیپ فلاپ‌هایی استفاده کنید که قابلیت Clear و Preset آسنکرون دارند.

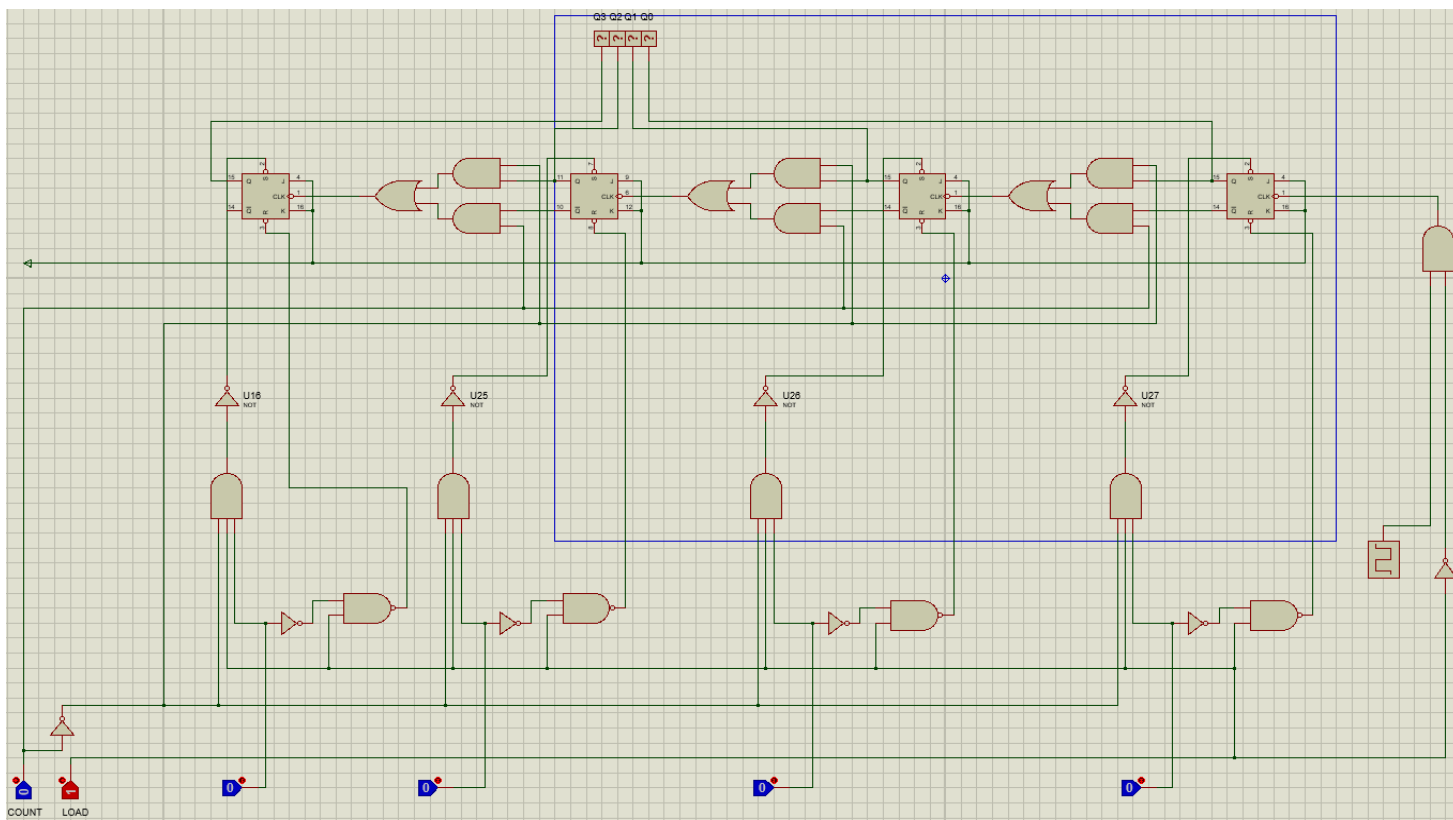


تئوری آزمایش:

با اعمال تغییرات لازم، افزودن ۲ ورودی LOAD و COUNT، افزودن ۴ بخش که هر کدام شامل ۲ گیت NOT، ۱ گیت AND و ۳ ورودی و ۱ گیت NAND و همچنین بخشی برای غیر فعال کردن CLOCK در هنگام ۱ بودن LOAD، قابلیت مقداردهی موازی را به مدار اضافه می‌کنیم (همانطور که از شکل ۱۵ مشخص است، هنگام ۱ بودن LOAD، شمارش انجام نمی‌شود).

گزارش کار بخش ۱.۲:

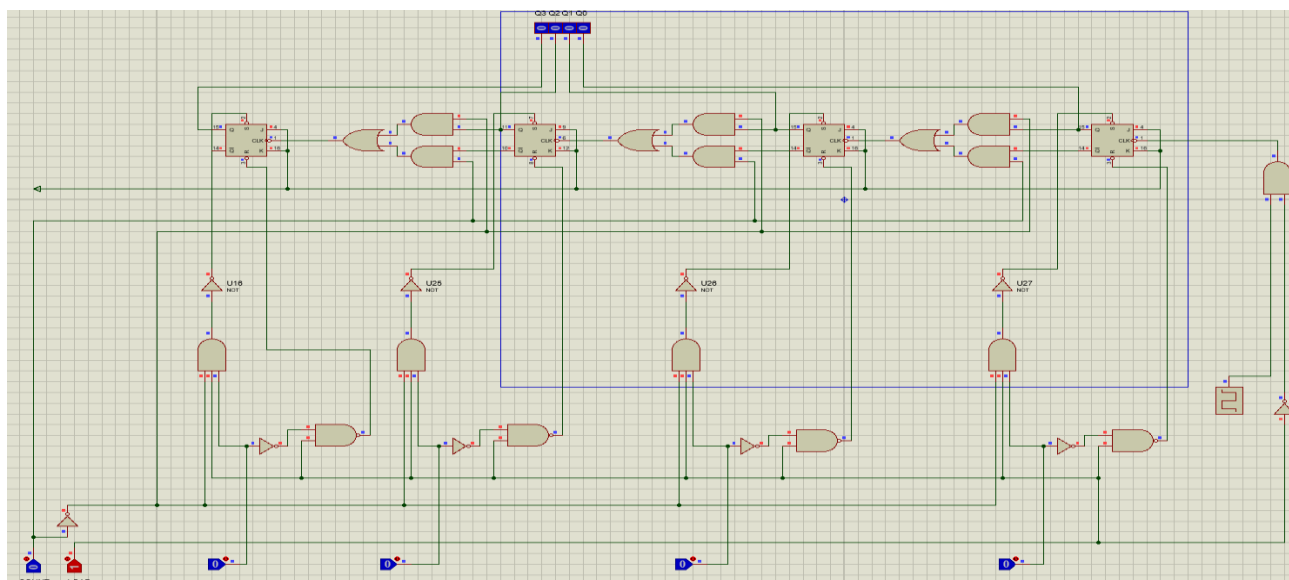
همانطور که طبق شکل ۱۶ مشخص است از قسمت قبل COUNT UP _ COUNT DOWN حذف شده اند و بجای آنها یک ورودی COUNT با کارایی مشابه قرار داده شده است. در حالتی که شمارش نداریم، برای توقف شمارش به این صورت عمل شده است که اگر LOAD یک باشد، از آنجا که کلاک را با 'LOAD' AND کرده ایم، فرآیند شمارش متوقف می‌شود. برای LOAD شدن نیز اگر ورودی کاربر صفر باشد، در اثر NAND شدن با LOAD، باعث می‌شود فلیپ فلاپ مربوطه ریست شود و اگر ورودی ۱ باشد، در اثر AND شدن با LOAD و 'COUNT' (هدف اطمینان از ۱ بودن LOAD و صفر بودن COUNT است)، اگر شرایط درست باشد، فلیپ فلاپ مربوطه را ست می‌کند.



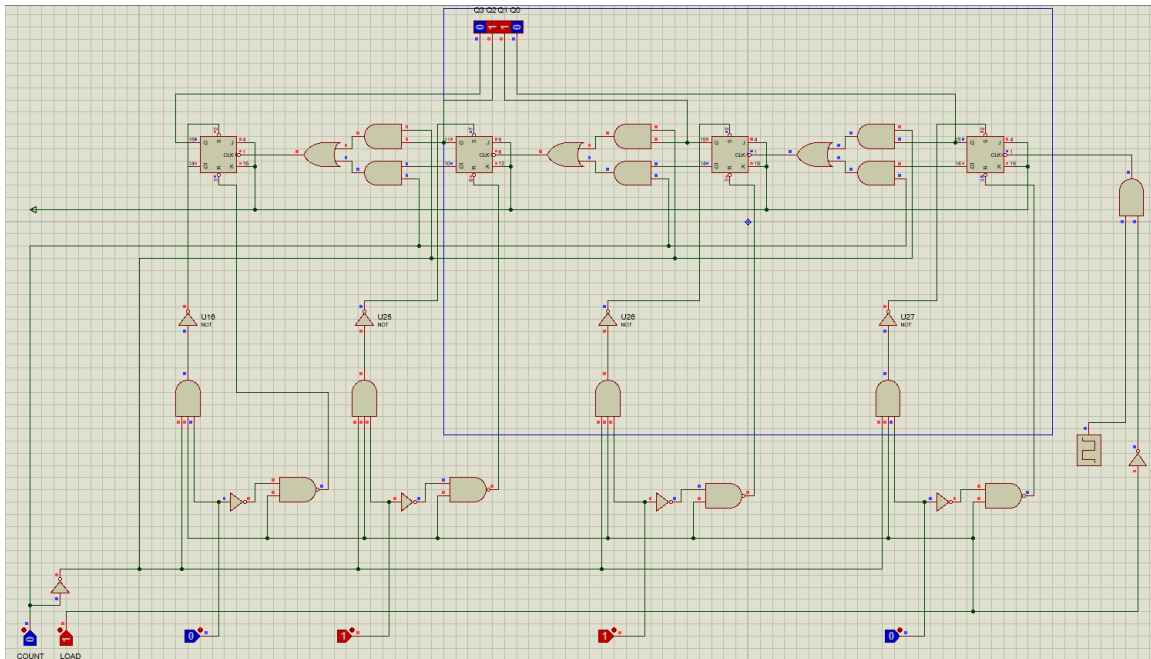
شکل ۱۶. مدار طراحی شده

به بررسی حالات می پردازیم:

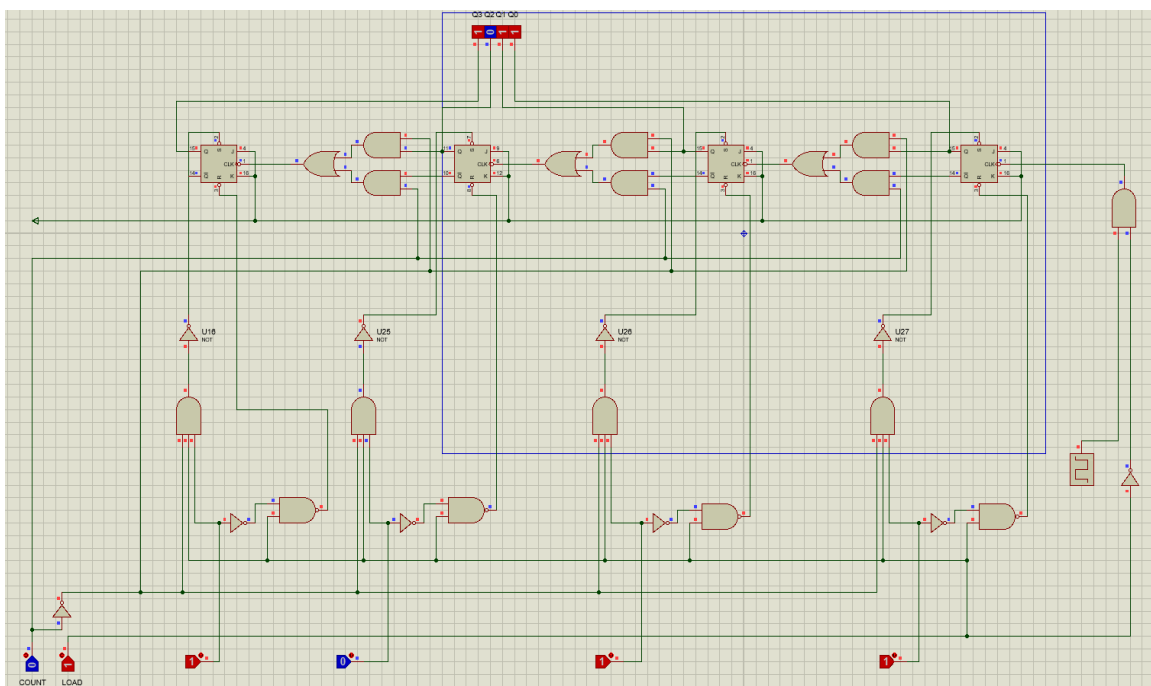
اگر LOAD، یک باشد و COUNT را صفر قرار بدهیم، باید مقدار دهی موازی انجام بشود.



شکل ۱۷. در ابتدا تمام ورودی ها ۰، و طبعاً خروجی ها نیز ۰ می باشند.



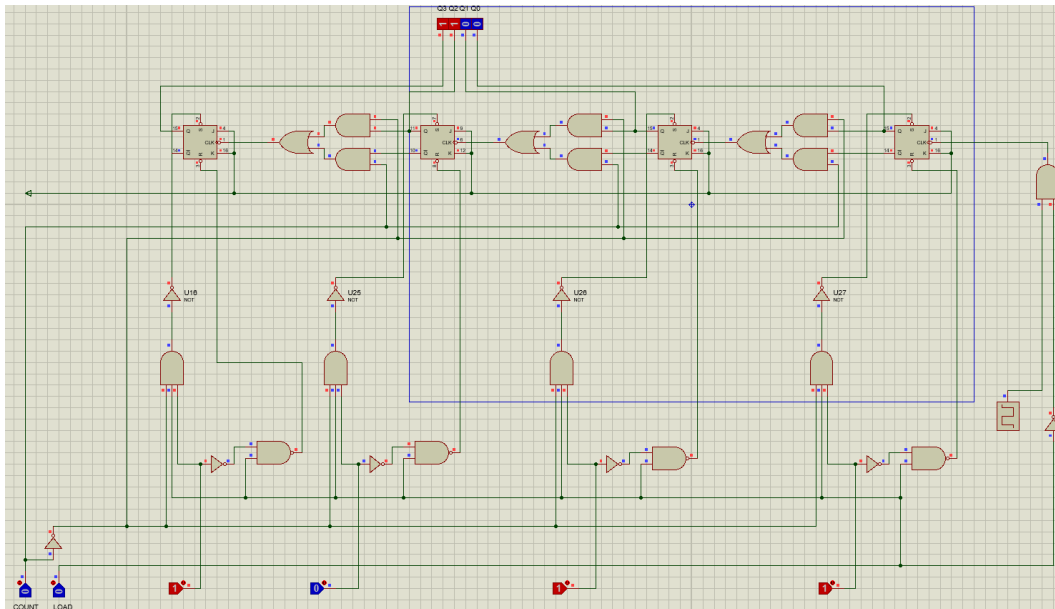
شکل ۱۸. با تبدیل ورودی به ۰۱۱۰، مشاهده می‌شود که مقداردهی انجام شده است.



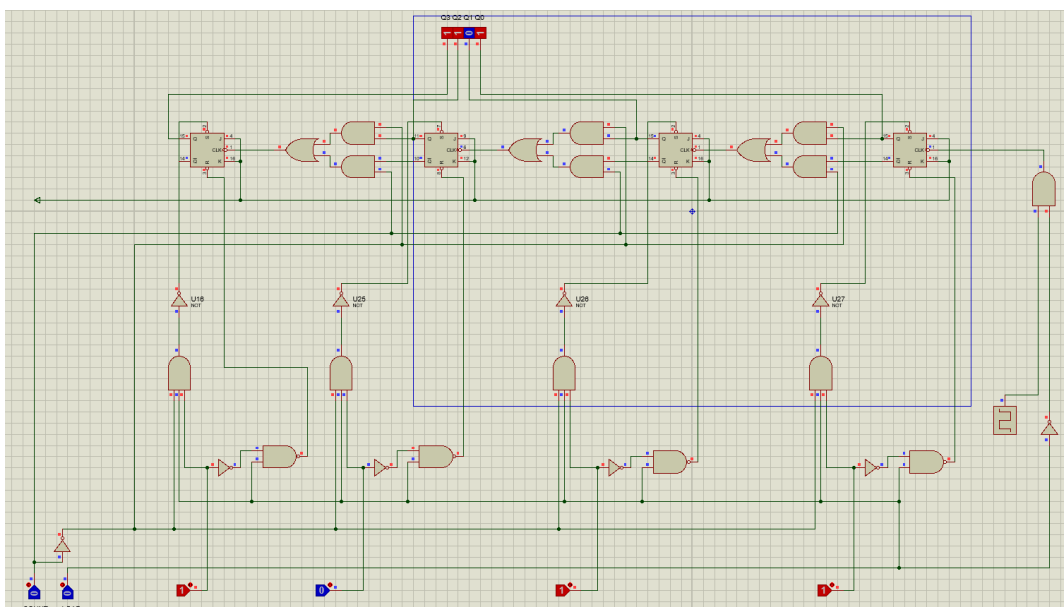
شکل ۱۹. با تبدیل ورودی به ۱۰۱۱، مشاهده می‌شود که مقداردهی انجام شده است.

حال با همان ورودی شکل ۱۹، $LOAD = 0$ و $COUNT = 0$ را امتحان می‌کنیم:

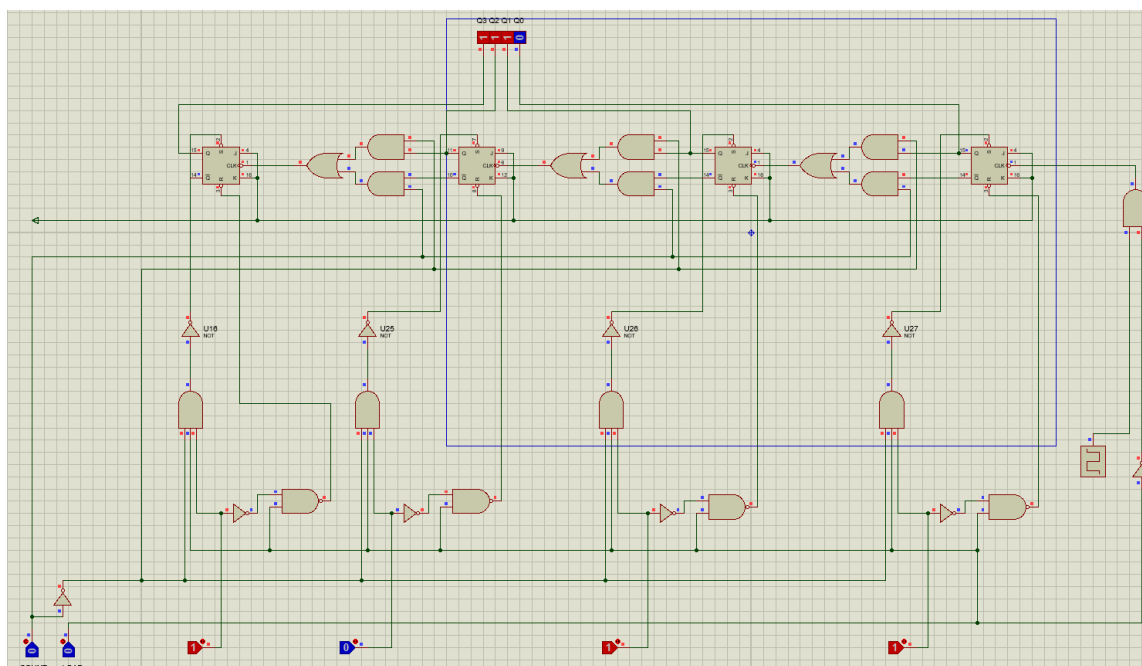
عدد ورودی ۱۰۱۱ می‌باشد و انتظار می‌رود که بعد از هر کلاک، به ترتیب اعداد نشان داده شده ۱۱۰۰، ۱۱۰۱، ۱۱۱۰ و ... باشد (COUNT UP).



شکل ۲۰. پس از صفر شدن LOAD و COUNT، همانطور که انتظار می‌رفت، COUNT UP انجام شده است.



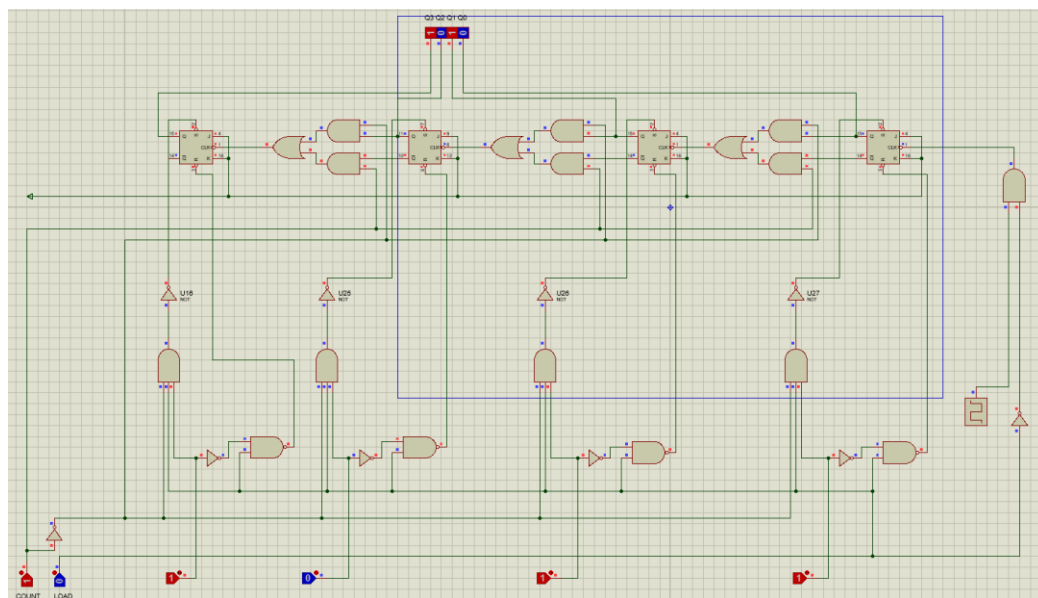
شکل ۲۱. در مرحله بعد نیز COUNT UP انجام شده و عدد نشان داده شده به ۱۱۰۱ تبدیل شده است.



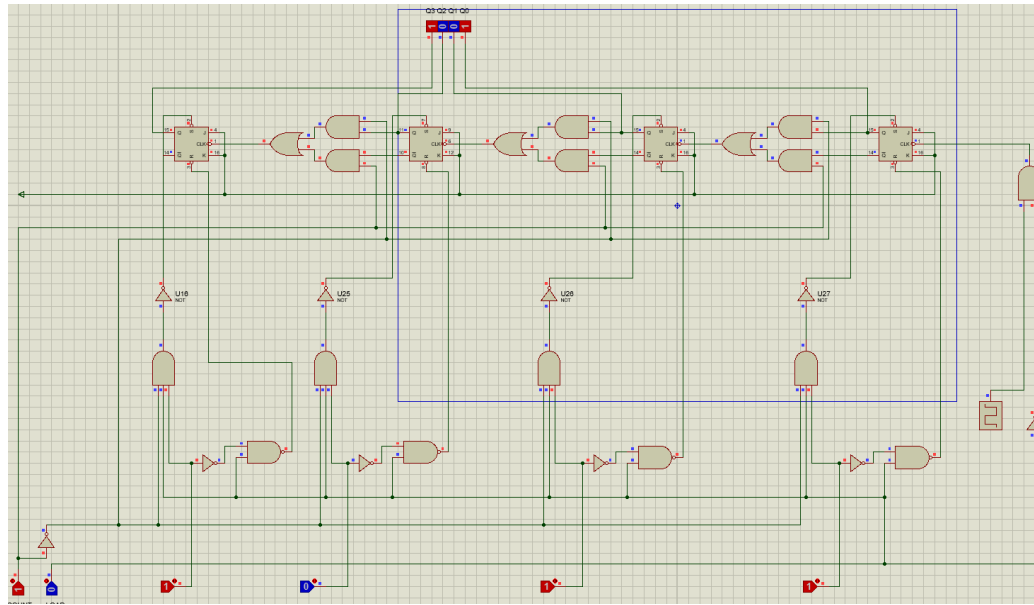
شکل ۲۲. باز هم COUNT UP انجام شده و عدد نشان داده شده ۱۱۱۰ می باشد.

حال با همان ورودی شکل ۱۹، $LOAD = 0$ و $COUNT = 1$ را امتحان می کنیم:

عدد ورودی ۱۰۱۱ می باشد و انتظار می رود اعداد نشان داده شده به ترتیب ۱۰۱۰، ۱۰۰۱، ۱۰۰۰ و ... باشد (COUNT DOWN).



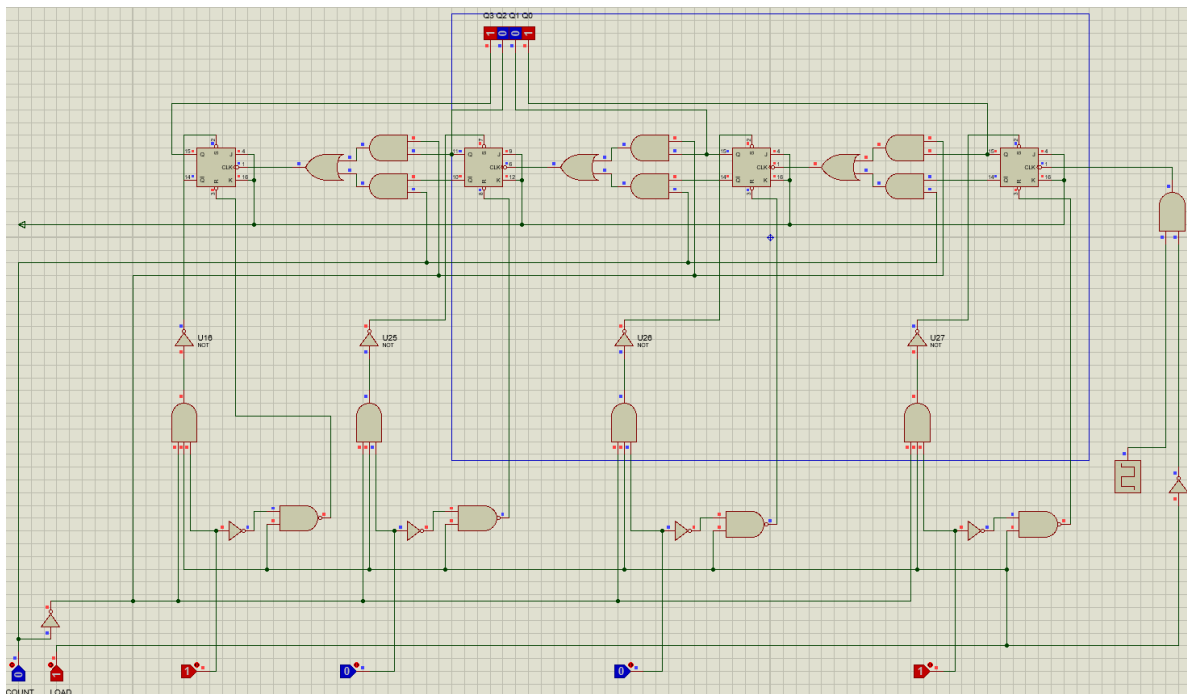
شکل ۲۳. همانطور که انتظار می رود بعد از کلاک، عدد نشان داده شده، ۱۰۱۰ می باشد.



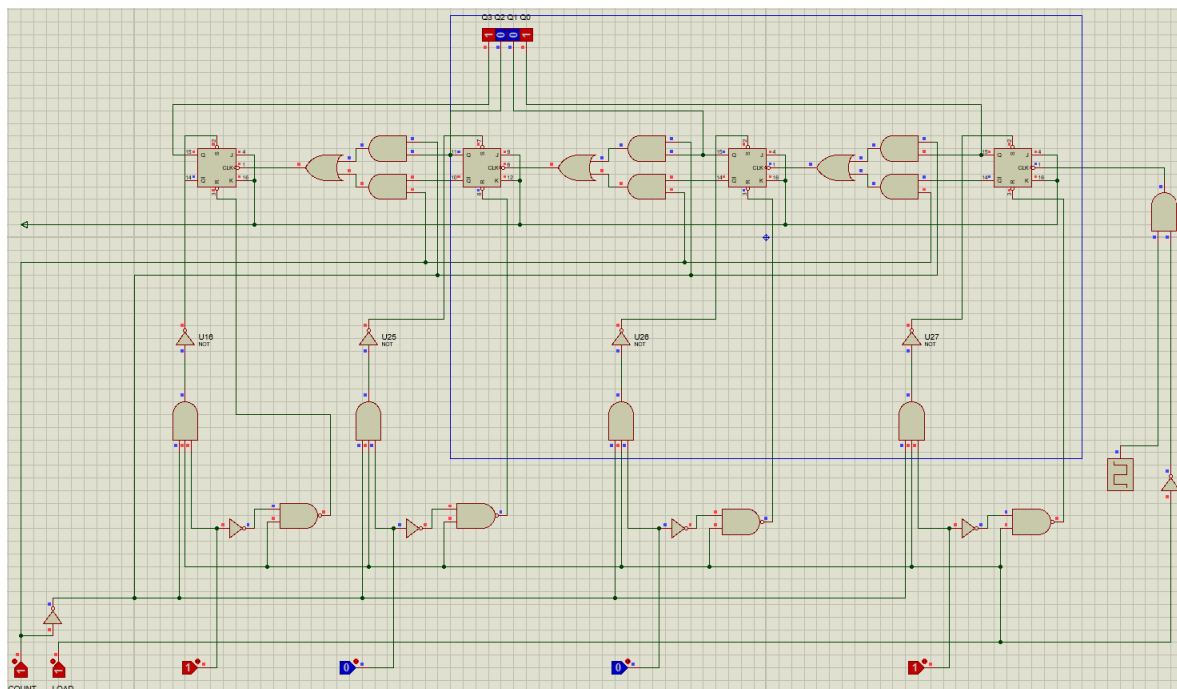
شکل ۲۴. در مرحله بعد، بعد از کلاک، عدد نشان داده شده، ۱۰۰۱ می‌باشد.

حال $LOAD = 1$ و $COUNT = 1$ را امتحان می‌کنیم:

طبق دستورالعمل هیچ اتفاقی نباید بیفتد، پس با ورودی ۱۰۰۱ امتحان می‌کنیم.



شکل ۲۵. ابتدا مقداردهی را انجام می‌دهیم.



شکل ۲۶. همانطور که مشاهده می‌شود، با یک کردن COUNT، هیچ اتفاقی نمی‌فتد (DO NOTHING).

بخش دوم _ شمارنده دودویی سنکرون

با استفاده از سه فلیپ فلاپ JK یک شمارنده سنکرون بسازید که اعداد صفر تا هفت را سه تا سه تا بشمارد. این شمارنده، یک ورودی X دارد که جهت شمارش را مشخص می‌کند، به این ترتیب که اگر $X = 0$ شمارش رو به پایین و اگر $X = 1$ شمارش رو به بالا خواهد بود.

تئوری آزمایش:

در این آزمایش، با استفاده از فلیپ فلاپ های JK، یک شمارنده سنکرون می‌سازیم که ۳ تا ۳ بشمارد.

بدین شکل که اگر از ۰ شروع کند و COUNT UP بکند، داریم:

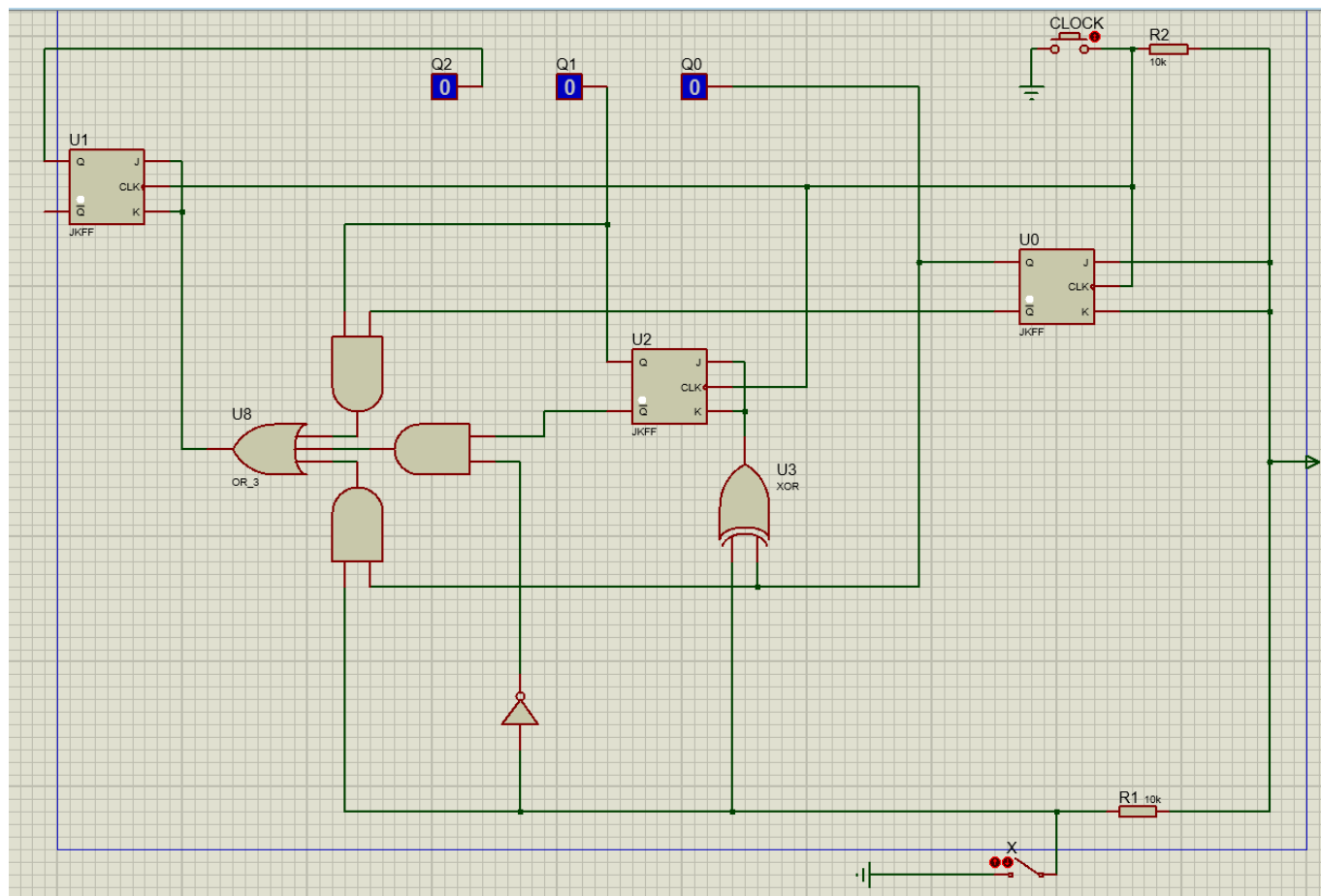
$$0 - 1 - 2 - 3 - 4 - 5$$

گزارش کار بخش دوم:

جدول درستی و مقادیر J, K محاسبه می‌شود. بر اساس آن مدار مطابق شکل ۲۷ ساخته می‌شود (X اگر ۱ باشد بالا شمار و اگر صفر باشد پایین شمار است).

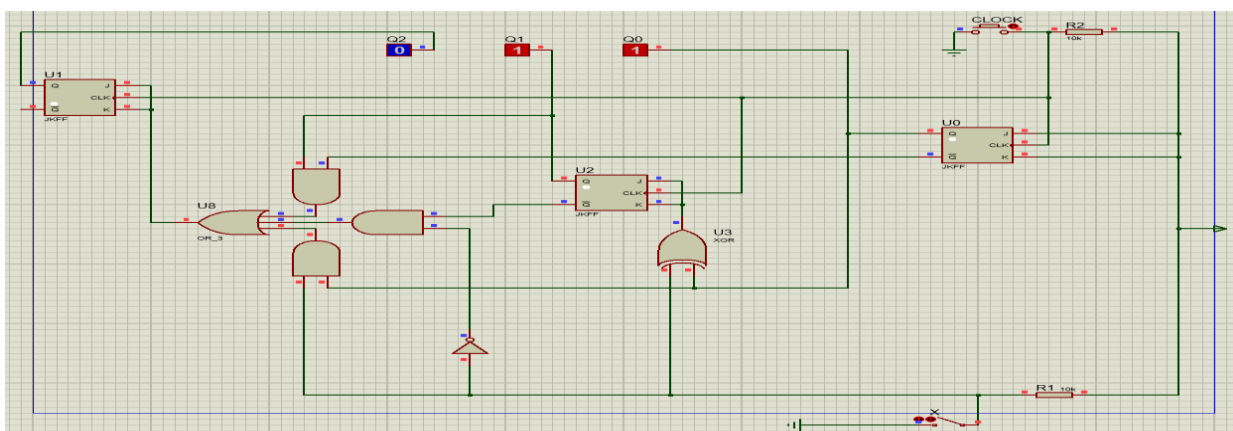
Q2	Q1	Q0		X	Q2'	Q1'	Q0'	J2	K2	J1	K1	J0	K0
0	0	0		0	1	0	1	1	X	0	X	1	X
0	0	0		1	0	1	1	0	X	1	X	1	X
0	0	1		0	1	1	0	1	X	1	X	X	1
0	0	1		1	1	0	0	1	X	0	X	X	1
0	1	0		0	1	1	1	1	X	X	0	1	X
0	1	0		1	1	0	1	1	X	X	1	1	X
0	1	1		0	0	0	0	0	X	X	1	X	1
0	1	1		1	1	1	0	1	X	X	0	X	1
1	0	0		0	0	0	1	X	1	0	X	1	X
1	0	0		1	1	1	1	X	0	1	X	1	X
1	0	1		0	0	1	0	X	1	1	X	X	1
1	0	1		1	0	0	0	X	1	0	X	X	1
1	1	0		0	0	1	1	X	1	X	0	1	X
1	1	0		1	0	0	1	X	1	X	1	1	X
1	1	1		0	1	0	0	X	0	X	1	X	1
1	1	1		1	0	1	0	X	1	X	0	X	1

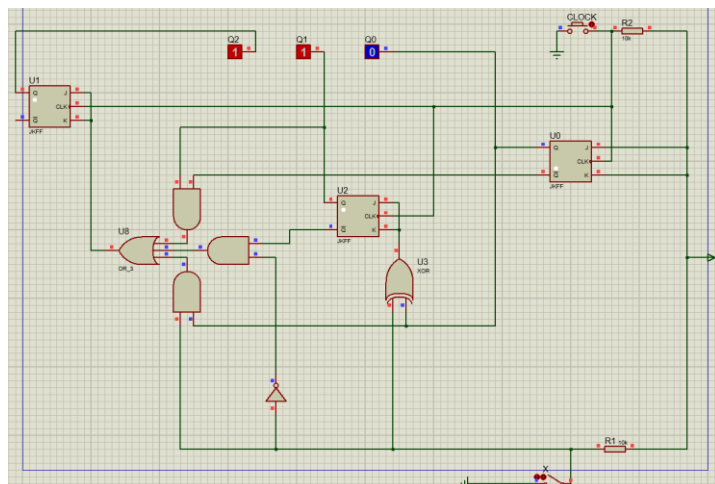
جدول کارنو



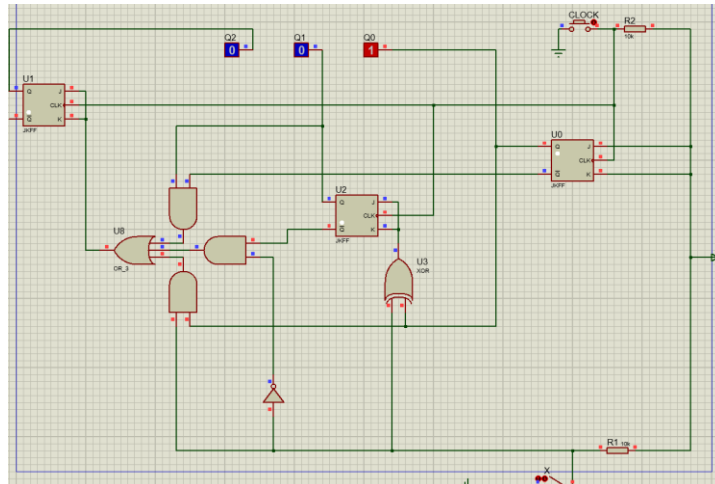
حال، یک مثال COUNT UP و یک مثال COUNT DOWN می‌زنیم:

در مثال اول، با باز گذاشتن سویچ X و بعد از هر کلاک، حالات را بررسی می‌کنیم (ورودی اولیه \bullet می‌باشد).



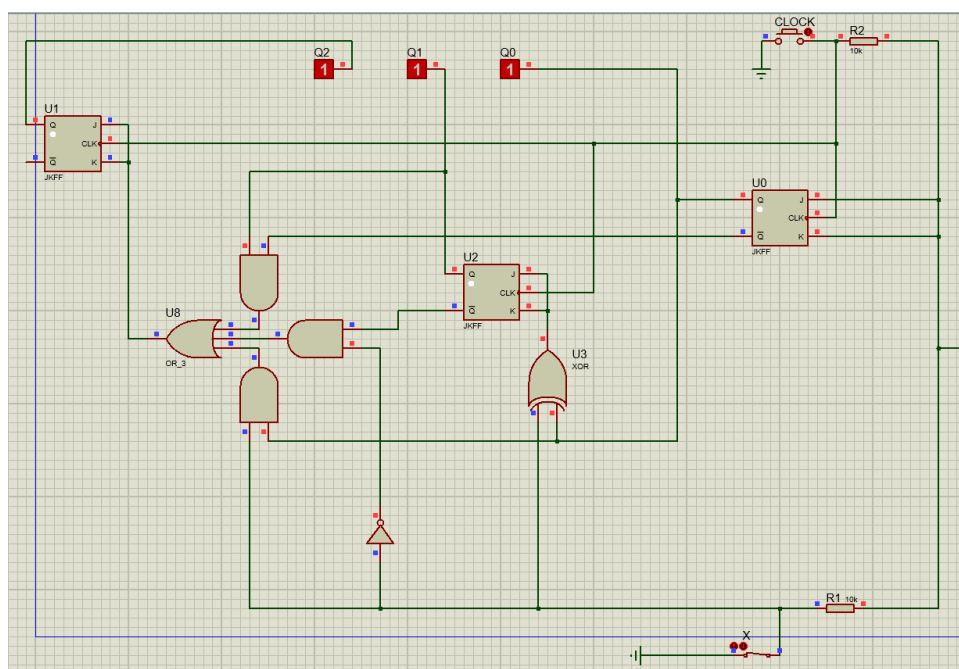


شکل ۲۹. بعد از کلاک، عدد نشان داده شده برابر ۶ (۱۱۰) می باشد.

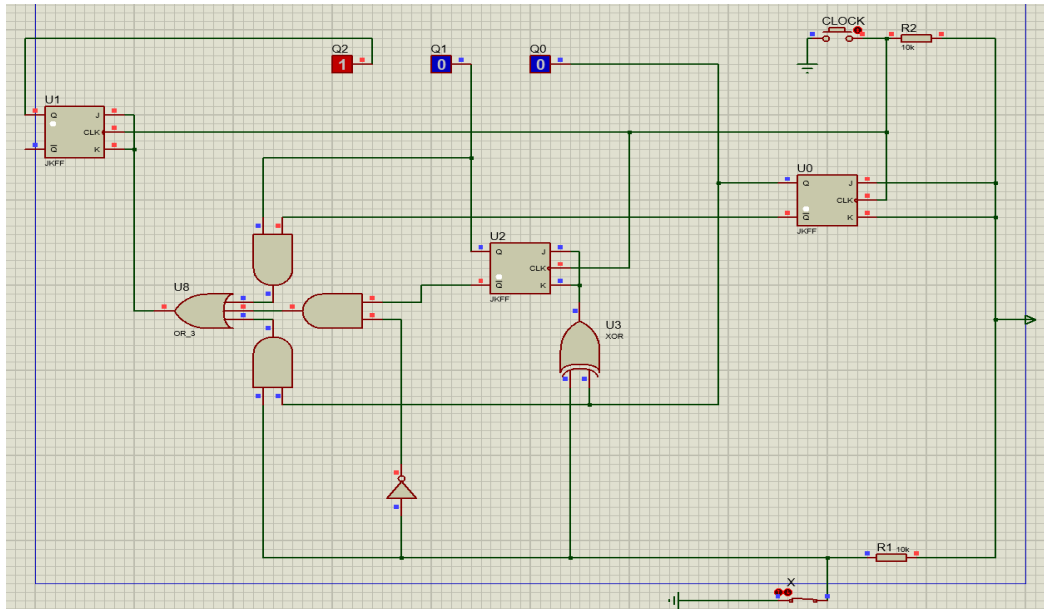


شکل ۳۰. بعد از کلاک، عدد نشان داده شده برابر ۱ (۰۰۱) می باشد.

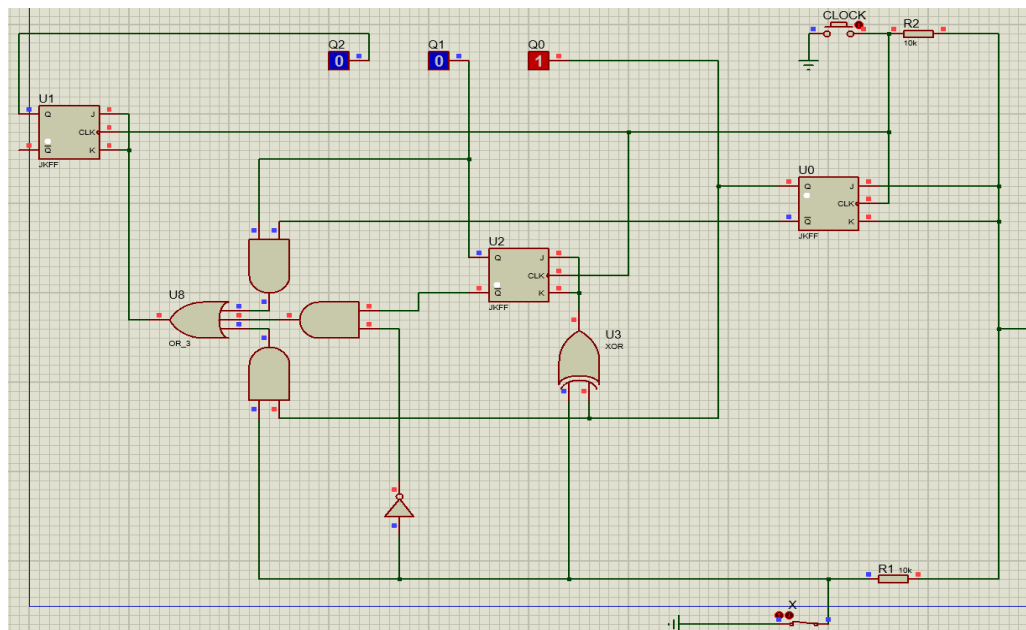
حال در مثال بعد با ورودی ۷ (۱۱۱)، سوئیچ X را می بندیم؛ خواهیم داشت:



شکل ۳۱. ورودی ۷ در ابتدا و بسته بودن سوئیچ X



شکل ۳۲. بعد از کلاک، عدد نشان داده شده ۴ خواهد بود (COUNT DOWN).



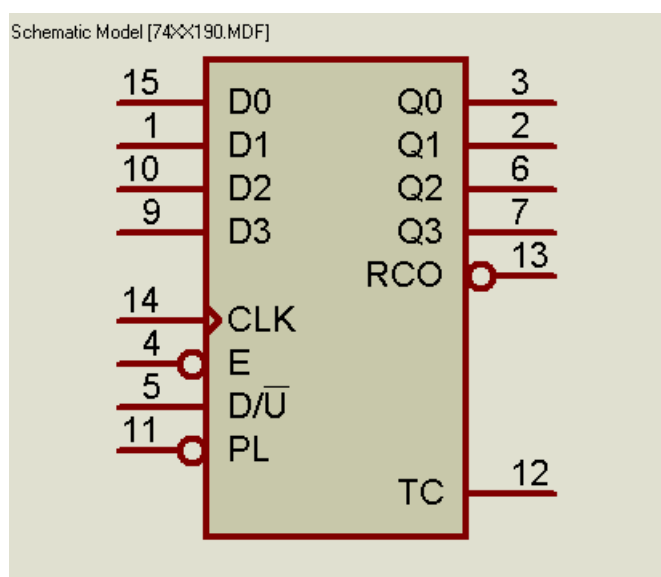
شکل ۳۳. بعد از کلاک، عدد نشان داده شده ۱ خواهد بود (COUNT DOWN).

بخش سوم _ شمارنده BCD

با استفاده از دو تراشه ۷۴۱۹۰ که یک شمارنده BCD با قابلیت مقداره‌ی اولیه و شمارش رو به بالا و پایین است، یک شمارنده برای شمارش اعداد صفر تا ۶۳ بسازید. خروجی شمارنده ها را به نمایشگرهای ۴ قطعه ای (۷-seg LED) وصل کنید.

تئوری آزمایش:

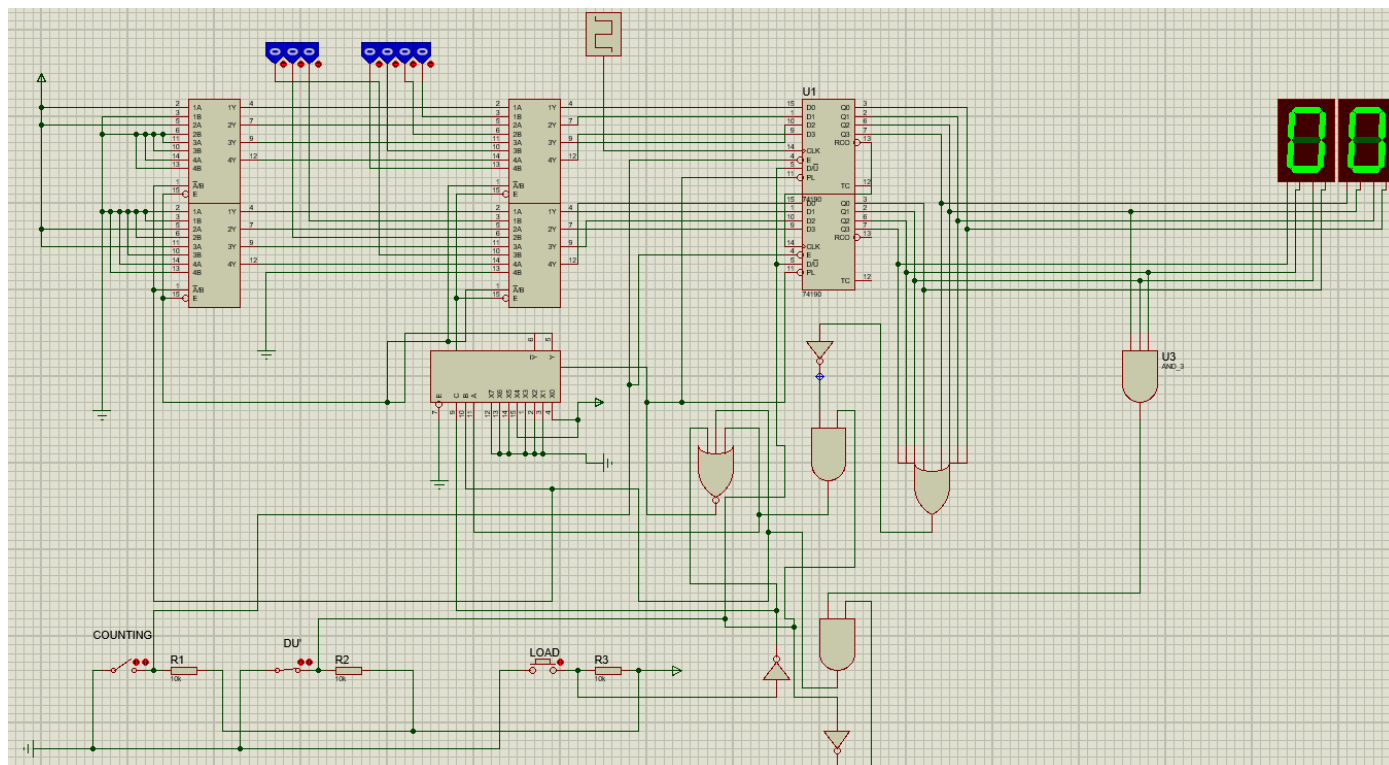
طبق خواسته دستور کار آزمایش، با استفاده از تراشه ۷۴۱۹۰، تلاش می‌کنیم شمارنده ای برای شمارش ۰ تا ۹۹ بسازیم و بعد از آن شروطی را با استفاده گیت های AND, OR, NOR, NOT و مولتی پلکسر تعیین می‌کنیم تا به شمارنده ۰ تا ۶۳ تبدیل بشود.



شکل ۳۴. تراشه ۷۴۱۹۰

گزارش کار بخش سوم:

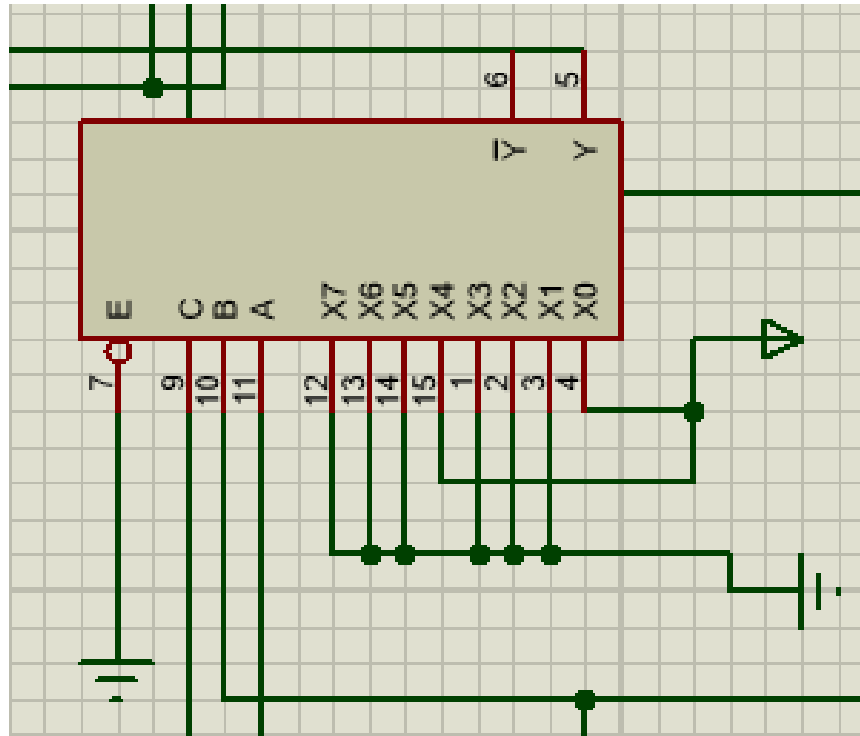
مدار طراحی شده را در شکل ۳۵ مشاهده می‌کنید. کارکرد LOAD و COUNTING مشخص است؛ LOAD ورودی را بارگذاری می‌کند و COUNTING در صورت وصل بودن، شمارش را آغاز می‌کند. کلید DU' نیز همانطور که از نامش پیداست در صورتی که متصل باشد، بالا می‌شمارد و اگر قطع باشد پایین می‌شمارد (در صورت وصل بودن COUNTING). کار شمارش را خود تراشه انجام می‌دهد. حال باید ۲ حالت را بررسی کنیم، وقتی که شمارش رو به بالاست و به عدد ۶۳ رسیدیم، برای مرحله بعد ۰ را در شمارنده لود می‌کنیم؛ وقتی شمارش رو به پایین است و به ۰ رسیدیم، برای مرحله بعد ۶۳ را در شمارنده لود می‌کنیم.



شکل ۳۵. مدار طراحی شده

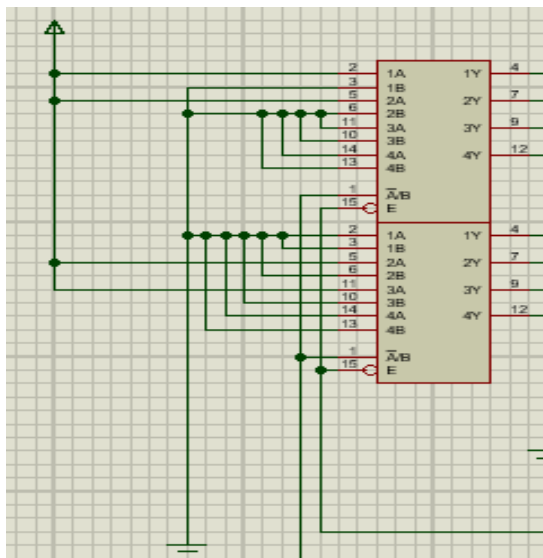
گیت های AND, OR در سمت راست تصویر برای تشخیص نوع شمارش و وقوع حالت مذکور قرار داده شده‌اند. سپس اگر یکی از حالت فوق رخ دهد یا کاربر بخواهد لود کند، با استفاده از گیت NOR، ورودی LOAD(PL) مشخص می‌شود. در سمت چپ تصویر از چند مولتی پلکسر استفاده شده است. این مولتی پلکسرها با تشخیص اینکه در چه حالتی قرار داریم (یکی از دو حالت که بالاتر ذکر شد یا حالت لود توسط کاربر) برای ما مقداری که باید لود شود را مشخص می‌کنند.

مطابق شکل ۳۶، ورودی A به حالتی وصل شده که شمارش به پایین باشد و به صفر برسیم. ورودی B مربوط به حالت شمارش بالا و رسیدن به ۶۳ است و ورودی C مشخص کننده حالتی است که کاربر قصد دارد LOAD کند. اگر هر سه ورودی فوق صفر باشند، خروجی Y را یک در نظر می‌گیریم. (تفاوتی نمی‌کند چه چیزی در نظر گرفته شود زیرا در این حالت قطعاً PL در تراشه‌ها غیرفعال است). اگر تنها C برابر ۱ باشد، خروجی ۱ است ($1=X^4$) در مولتی پلکسرهای بالایی شکل مدار، مقادیر ورودی عدد B که همان ورودی‌های کاربر برای LOAD هستند فعال شوند. اگر هر یک از A, B, نیز یک باشد، Y برابر ۰ می‌شود تا مقادیر ورودی عدد A که از مولتی پلکسرهای قبل تر مشخص می‌شوند لود شوند.



شکل ۳۶. تراشه ۷۴۱۵۱

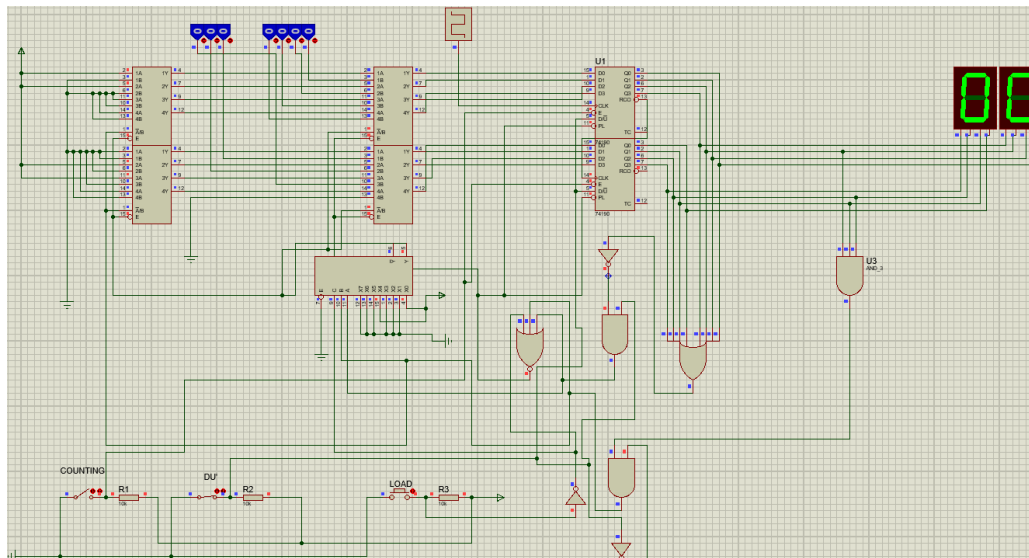
مطابق شکل ۳۷، در این دو مولتی پلکسر اعداد ۶۳ و ۰ مشخص شده‌اند تا بر اساس ورودی A/B' یکی به عنوان خروجی مشخص و لود شود. توجه شود ورودی A/B' به خروجی گیت اندی وصل شده است که تشخیص می‌داد در حالت بالا شمار به ۶۳ رسیدیم یا نه. اگر آن باشد اعداد متناظر ورودی B به خروجی وصل می‌شوند و در غیر اینصورت اعداد متناظر با ورودی های A.



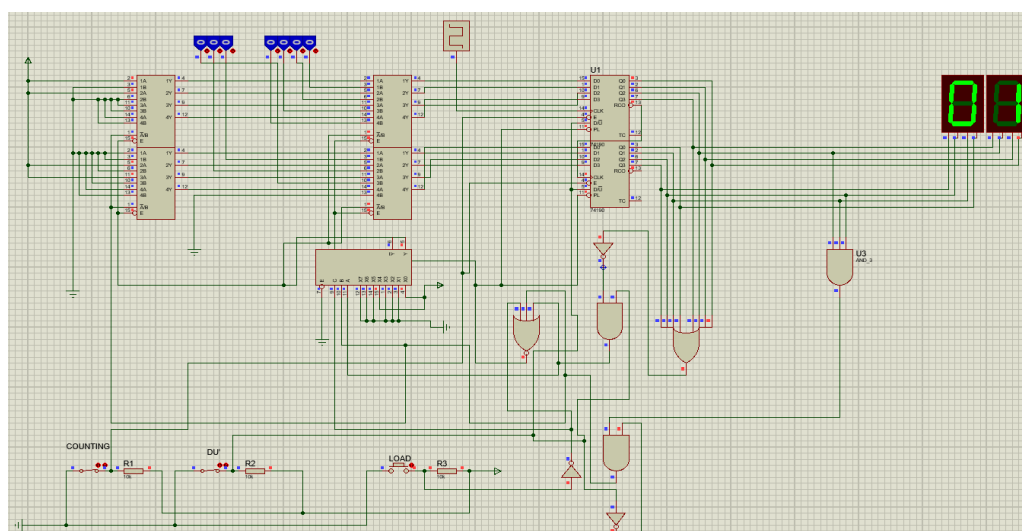
شکل ۳۷. تراشه ۷۴HC۱۵۷

حال به بررسی عملکرد مدار می‌پردازیم:

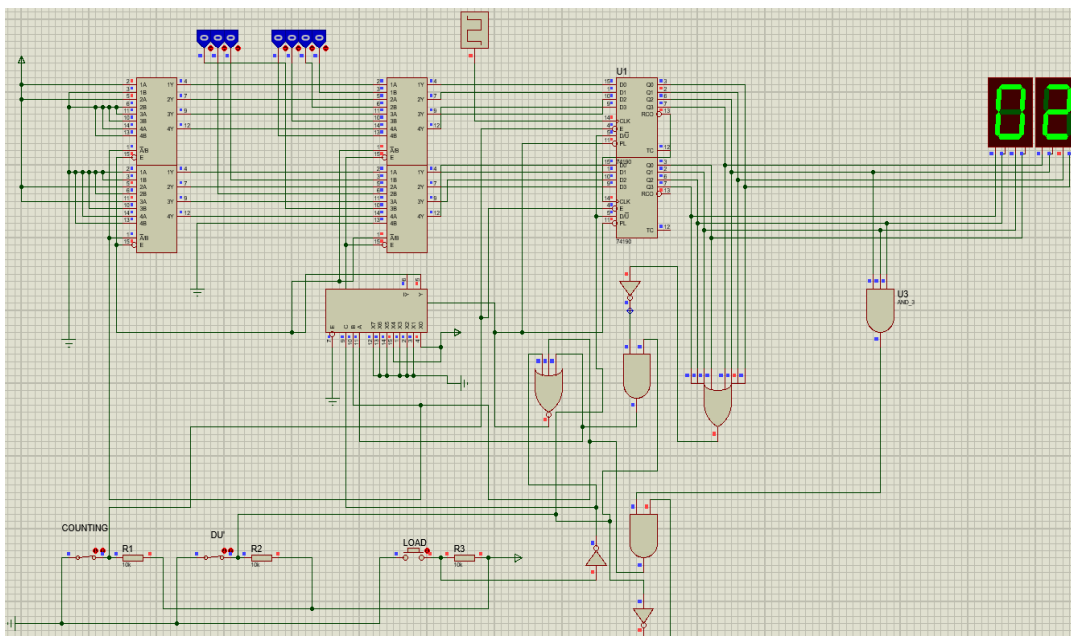
در ابتدا، سوییچ 'DU' را وصل کرده و COUNTING را باز می‌گذاریم.



شکل ۳۸. مدار در ابتدا

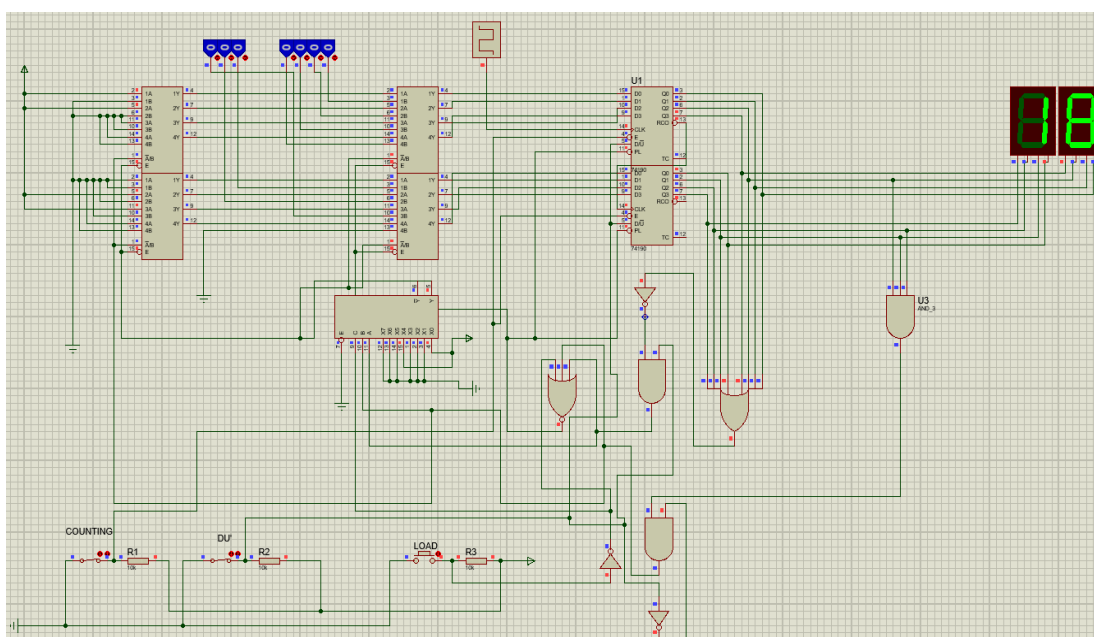


شکل ۳۹. شروع به شمارش به بالا بعد از وصل کردن COUNTING



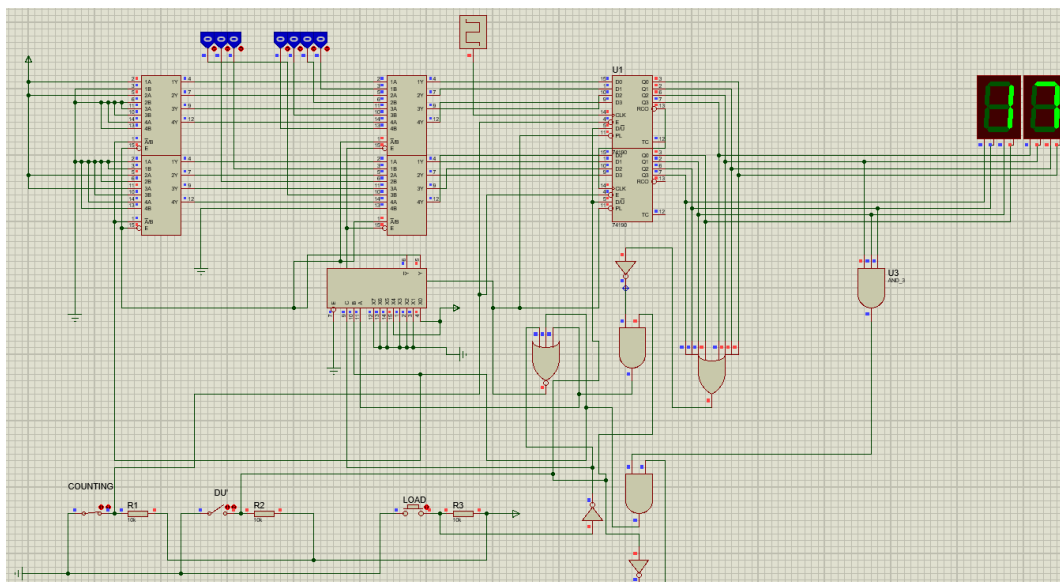
شکل ۴۰. شمارش به بالا

همینطور ادامه می‌دهیم و برای مثال تا ۱۸ پیش می‌رویم.



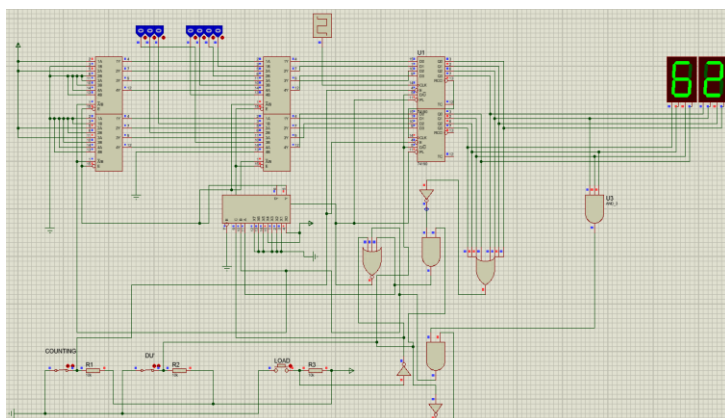
شکل ۴۱. نشان داده شدن عدد ۱۸

حال طبق شکل ۴۲، سویچ DU' را باز می‌کنیم تا شمارش به پایین آغاز شود.

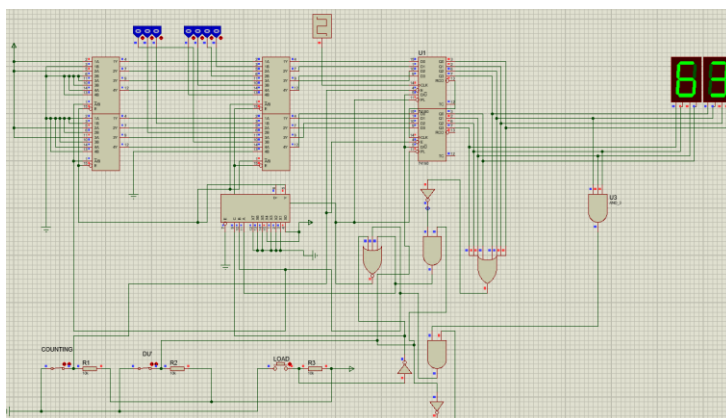


شکل ۴۲. نشان داده شدن عدد ۱۷ بعد از قطع کردن سویچ DU'

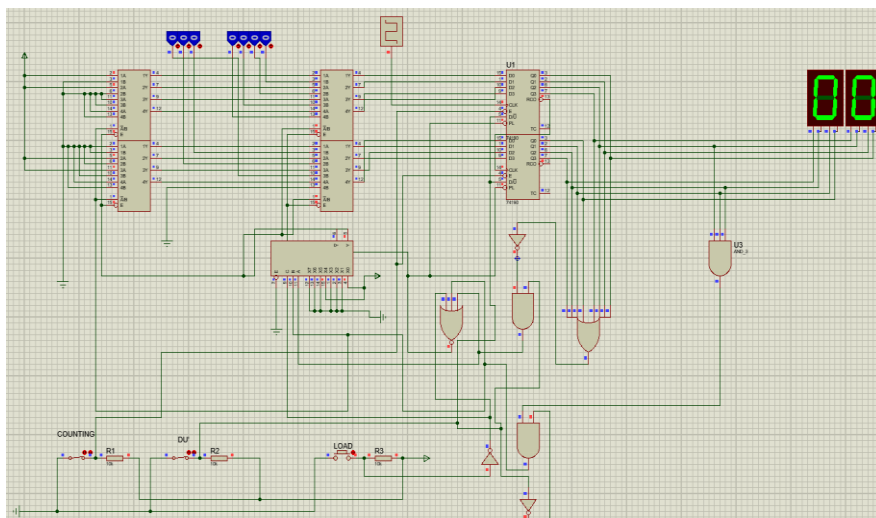
در مرحله بعد، مطابق شکل ۴۳ و ۴۴ و ۴۵، سویچ DU' را وصل می‌کنیم تا به ۶۲ و ۶۳ برسیم و بعد از آن به ۰ برسیم (نشان بدهیم تا ۶۳ شمارش می‌کند).



شکل ۴۳. رسیدن به ۶۲ با شمارش رو به بالا

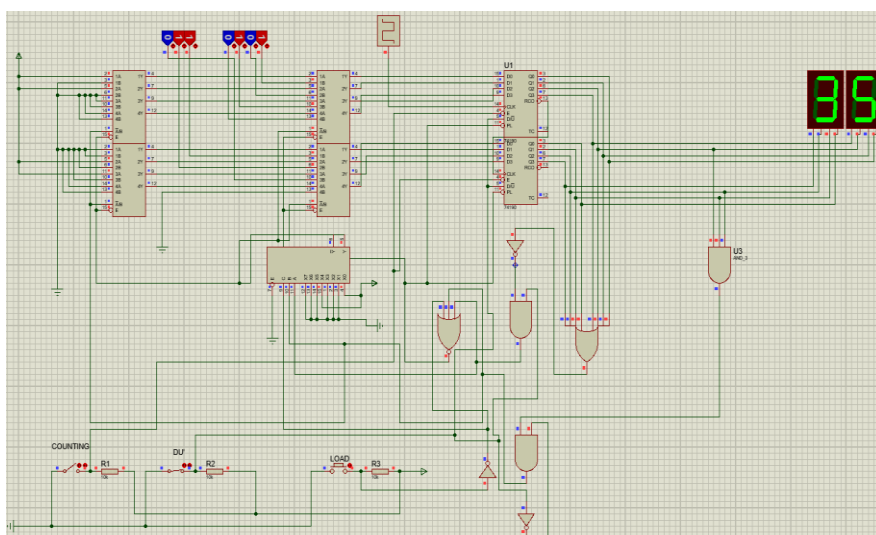


شکل ۴۴. رسیدن به ۶۳ با شمارش رو به بالا

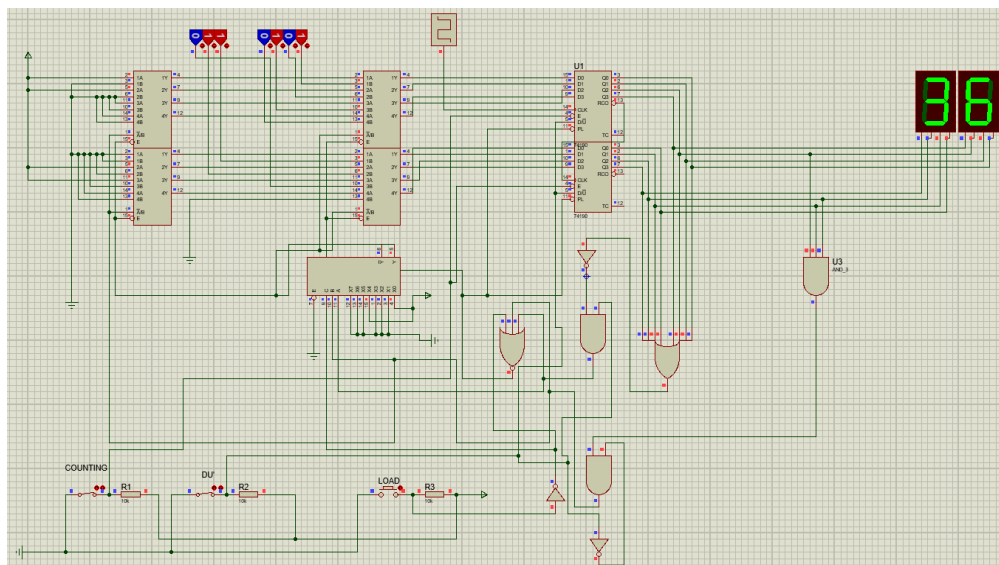


شکل ۴۵. رسیدن به ۰ بعد از ۶۳

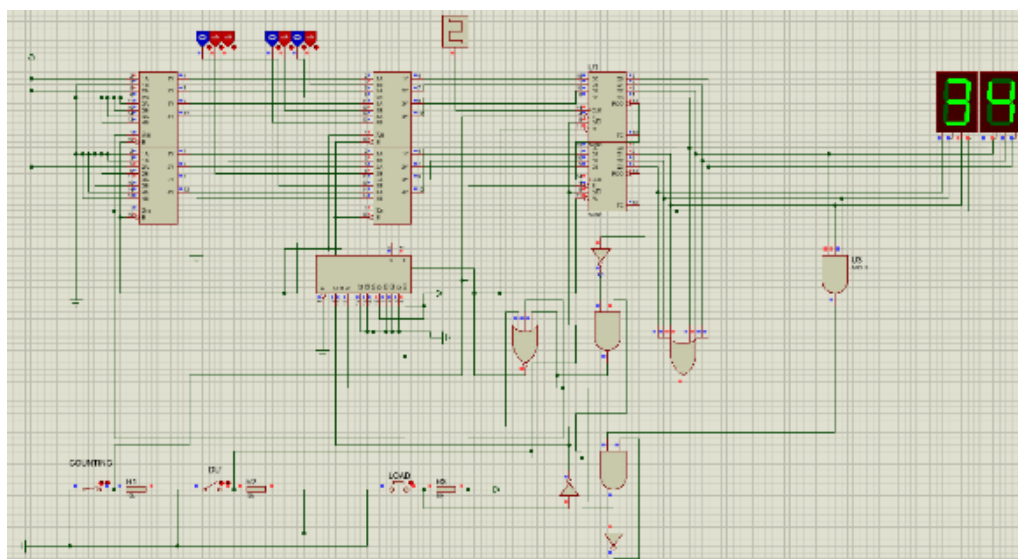
حال مقداردهی با استفاده از LOAD را امتحان می‌کنیم، عدد ۳۵ را ورودی می‌دهیم، و بعد از زدن دکمه LOAD، مقداردهی کرده و سپس شمارش رو به بالا و پایین را امتحان می‌کنیم. نکته قابل ذکر برای ورودی دادن این است که برای ورودی دادن از ۳ LOGICTOGGLE برای دهگان و ۴ LOGICTOGGLE برای یکان استفاده می‌کنیم و پرارزش ترین رقم دهگان را ۰ می‌گذاریم زیرا شمارشگر ما تا ۶۳ می‌باشد و ۱ شدن این رقم، بی معنی می‌باشد. خروجی را نیز طبق خواسته سوال به یک ۷SEG-BCD-GRN می‌دهیم.



شکل ۴۶. مقداردهی عدد ۳۵ بعد از زدن کلید LOAD (۰۰۱۱ ۰۱۰۱)



شکل ۴۷. نشان داده شدن عدد ۳۶ بعد از شمارش رو به بالا (COUNTING و DU' هر دو وصل می‌باشند).



شکل ۴۸. نشان داده شدن عدد ۳۴ بعد از شمارش رو به پایین (COUNTING وصل و DU' قطع می‌باشند).

در شکل ۴۸، شمارش به پایین را هم امتحان کرده‌ایم.

نتیجه گیری:

در آزمایش سوم درس آزمایشگاه مدارهای منطقی، به آشنایی با شمارنده‌ها و انجام چند آزمایش ساده و پیچیده پرداختیم.

در بخش اول، با شمارنده دودویی آسنکرون و آشنایی با مقداردهی موازی پرداختیم؛ با استفاده از نرم‌افزار PROTEUS مدار را طراحی کردیم و سپس قابلیت مقداردهی موازی را به آن اضافه کردیم و چند مثال برای درستی مدار را شرح دادیم.

در بخش دوم، با استفاده از نرم‌افزار PROTEUS، شمارنده دودویی سنکرون را طراحی کردیم به شکلی که ۳ تا ۳ تا بشمارد و اعداد را نمایش بدهد. همچنین قابلیت شمارش به بالا و پایین را هم در این مدار در نظر گرفتیم.

اما در بخش سوم کار سخت‌تری در پیش داشتیم و شمارنده BCD را که تا ۶۳، هم بالاشمار و هم پایین‌شمار، بشمارد، طراحی کردیم. این مدار باید قابلیت مقداردهی را نیز می‌داشت که چند مثال از هر کدام را شرح دادیم و همچنین با کارکرد تراشه ۷۴۱۹۰ نیز آشنا شدیم.

منابع و مراجع:

Mano, M. Morris. Computer system architecture. Prentice-Hall of India, 2003.