

به نام خدا

گزارش کار سوم آزمایشگاه مدارهای منطقی



دانشکده مهندسی کامپیووتر

آزمایش سوم: شمارنده

علی محمدزاده شبستری

۴۰۱۱۰۶۴۸۲

تابستان ۱۴۰۲

فهرست

<u>۳</u>	<u>مقدمه</u>
<u>۳</u>	<u>شمارنده باینری آسنکرون</u>
<u>۱۰</u>	<u>شمارنده باینری سنکرون</u>
<u>۱۴</u>	<u>شمارنده BCD</u>
<u>۱۷</u>	<u>*امتیازی: شمارنده BCD با قابلیت بارگذاری</u>
<u>۲۰</u>	<u>جمع‌بندی</u>
<u>۲۱</u>	<u>منابع</u>

مقدمه

هدف این آزمایش بررسی منطق شمارندها و ساختن انواع آن‌ها با قابلیت‌های گوناگون است. این آزمایش در سه بخش انجام می‌شود.

- در بخش اول، منطق شمارنده باینری آسنکرون را بررسی می‌کنیم، یک نمونه چهاربیتی از آن را می‌سازیم، و سپس یک شمارنده دیگر از روی آن می‌سازیم که قابلیت بارگذاری موازی هم دارد.
- در بخش دوم، یک شمارنده باینری سنکرون می‌سازیم که از صفر تا هفت را به صورت سه‌تا سه‌تا می‌شمارد.
- در بخش سوم، با استفاده از دو تراشه ۷۴۱۹۰ (یک شمارنده BCD) اعداد صفر تا ۶۳ را می‌شماریم. خروجی‌های هر تراشه را به طور مناسب به یک 7-seg LED متصل می‌کنیم تا نمایش داده شود.

شمارنده باینری آسنکرون

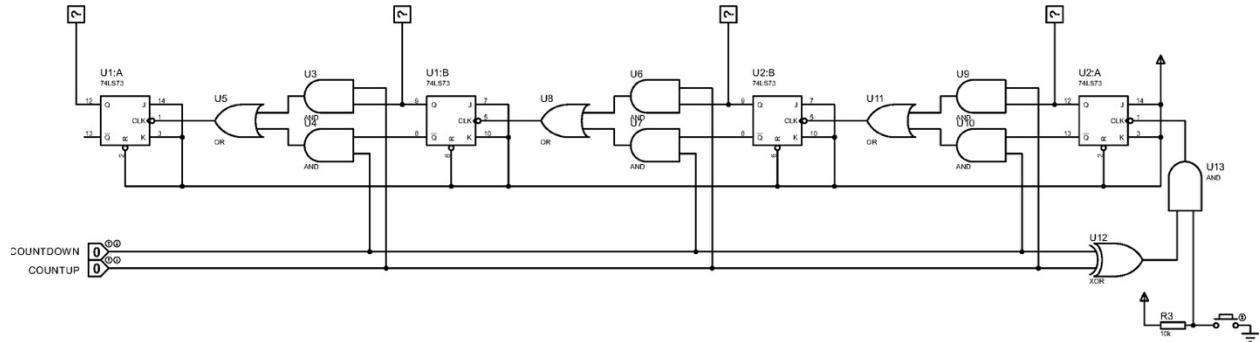
۴-۱-۱ | ساخت یک شمارنده باینری آسنکرون ::: بخش تئوری :::

در ابتدا می‌خواهیم یک شمارنده باینری آسنکرون با استفاده از JKFF (قطعه 74LS73) بسازیم. سپس در بخش بعدی قابلیت بارگذاری موازی را به آن اضافه می‌کنیم. از آنجایی که شمارنده باشد، ما ورودی کلک فلیپ‌فلاب اول را به یک clock بیرونی وصل می‌کنیم، ولی برای کلک فلیپ‌فلاب‌های بعدی، باید از خروجی فلیپ‌فلاب‌های قبلی استفاده کنیم. خروجی هر فلیپ‌فلاب مستقیماً به کلک فلیپ‌فلاب بعدی وصل نمی‌شود، بلکه باید گیت‌های لازم را درج کنیم و عملیات منطقی موردنیاز را انجام دهیم.

منطق شمارنده باینری آسنکرون در درس مدارهای منطقی بررسی شده است. ما در این آزمایش فقط آن را پیاده‌سازی می‌کنیم. زیرا در گزارش کار هم همین خواسته شده است. مدار ما می‌تواند شمارش را در دو جهت بالا یا پایین انجام دهد. تعیین جهت شمارش با استفاده از دو ورودی **countdown** و **countup** مشخص می‌شود. به این صورت که برای شمارش بالا باید **countup=۱** و **countdown=۰** باشد و برای شمارش پایین بالعکس. در دو حالت که هر دو تا یک یا هر دو تا صفر باشند، هیچ اتفاقی نمی‌افتد. زیرا آنها باهم XOR می‌شوند و حاصلش با AND وارد button می‌شود. در نتیجه اگه آن دو حالت اتفاق بیفتد حاصل XOR صفر می‌شود و پس از آن حاصل AND هم صفر می‌شود و کلک تغییری نمی‌کند. در خصوص ورودی‌های J و K و R فلیپ‌فلاب هم متنذکر می‌شویم که هر سه به power (معادل یک منطقی) متصل هستند. می‌دانیم اگر در یک JKFF هر دو J و K یک باشند، عمل تاگل اتفاق می‌افتد.

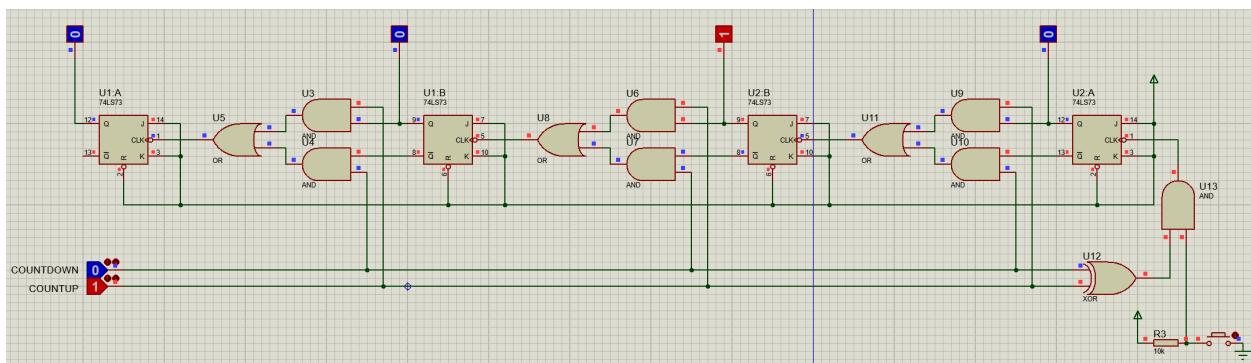
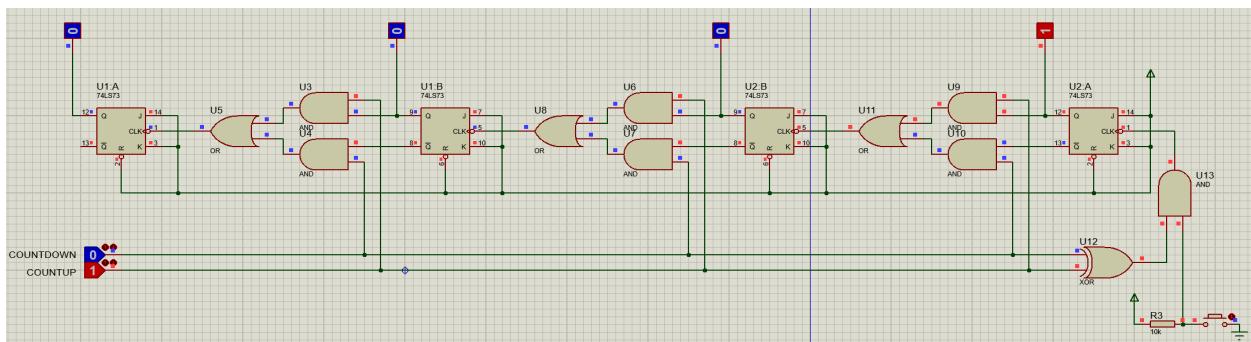
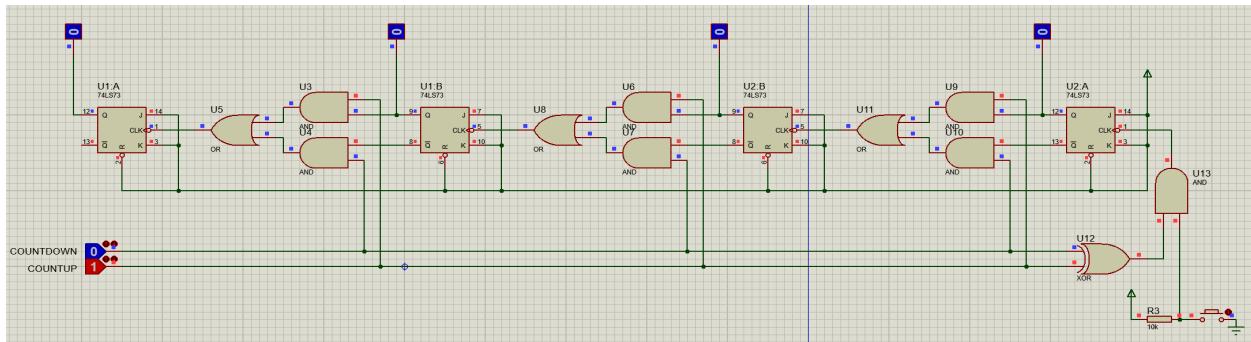
۱- بخش عملی ::

مدار زیر در برنامه پروتئوس طراحی شده و در قسمت Export Graphics خروجی تصویری گرفته شده است.



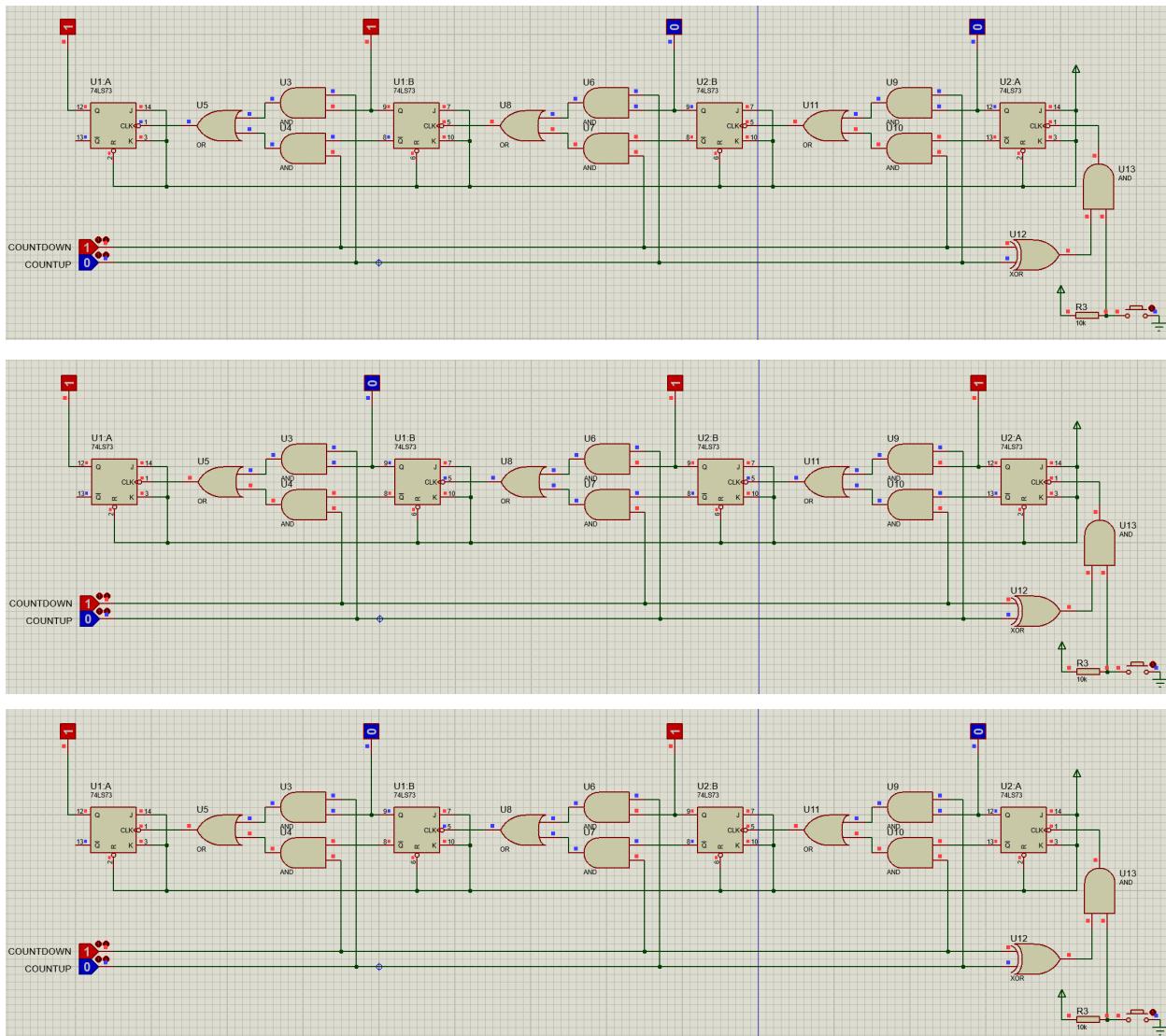
۱- مدار شمارنده باینری آستکرون (چهار بیتی)

تست مدار: یک بار هم مدارمان را امتحان می کنیم. سه تصویر مرتبه شمارش بالا (صفراً ← یک ← دو)



۲و۳- شمارش عدد صفر، یک و دو (countdown=0 , countup=1)

سه تصویر مربوط به شمارش پایین (دوازده ← یازده ← ده)



۵ و ۶ و ۷-شمارش دوازده، یازده و ده (countdown=1, countup=0)

۴-۱ | اضافه کردن قابلیت مقداردهی موازی

::: بخش تئوری :::

اکنون باید با استفاده از الگوی مدار طراحی شده در قسمت قبل، یک مدار طراحی کنیم که علاوه بر شمارش (در دو جهت) بتواند مقداردهی موازی هم داشته باشد. به این منظور ما باید فلیپفلابهای قبلی را با نوع دیگری از فلیپفلاب که دو ورودی (Preset(Set) و Clear(Reset)) دارد جایگزین کنیم. به این منظور در نرمافزار پروتئوس قطعه‌ای با نام 74LS76 را انتخاب می‌کنیم و هر چهارتا فلیپفلاب قبلی را با آن تعویض می‌کنیم. زیرا این تراشه S و R هم دارد.

برای مقدار J و K هر چهارتا فلیپفلاب مجدداً همان مقدار یک را می‌گذاریم. اما برای S و R باید گیت اضافه کنیم و عملیات منطقی انجام دهیم. کمی جلوتر در مورد جزئیات آن مطلع می‌شویم.

ما در مدارمان سه کار می‌خواهیم انجام دهیم: شمارش به بالا، شمارش به پایین، مقداردهی موازی. لذا می‌توانیم از دو ورودی `load` و `count` استفاده کنیم و از حالات ترکیب‌های این دو بیت، عمل دلخواه را انجام دهیم. جدول زیر که در دستور کار آزمایشگاه موجود بود، برای ما مشخص کرده است در هر حالت چه کاری انجام شود.

<i>load</i>	<i>count</i>	<i>function</i>
0	0	<i>count up</i>
0	1	<i>count down</i>
1	0	<i>parallel load</i>
1	1	<i>do nothing</i>

۸- جدول تعیین عملکرد مدار

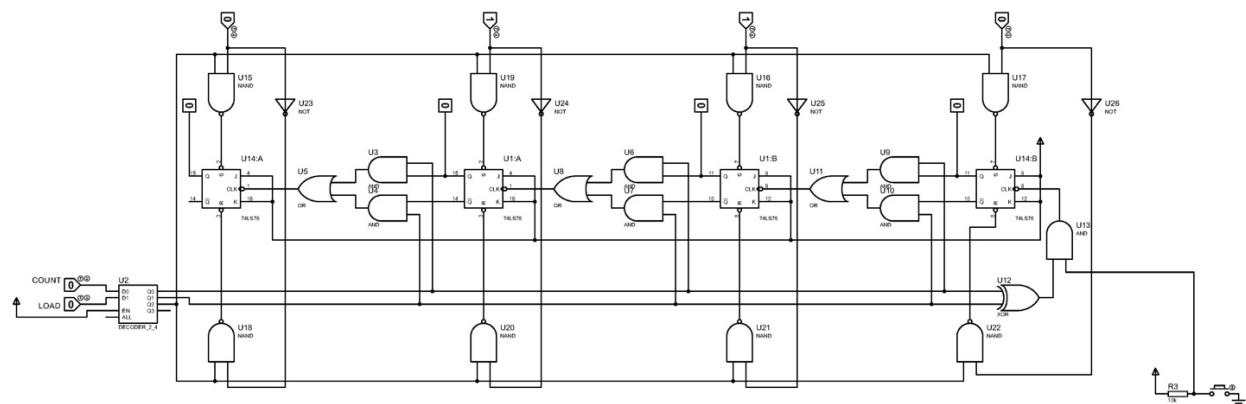
باشیم. ما از یک دی کدر استفاده می کنیم. دو خروجی اول (Q_0, Q_1) در واقع مربوط به عمل شمارش هستند. پس عین بخش قبل، همان‌ها را مستقیماً به XOR متصل می کنیم.

خروجی سوم که مربوط به عمل بارگذاری است، نیازمند تعدادی گیت است که قرار بود درمورد جزئیاتش بیشتر بحث کنیم. ما از دو گیت NAND استفاده می‌کنیم و به هردوی آن‌ها Q_2 را می‌دهیم. ورودی دوم یکی از این دو گیت برابر I_x و گیت دیگر I_x' است. برای هر بیت دو گیت NAND لازم داریم. یعنی در مجموع هشت تا NAND داریم. هر موقع در حالتی غیر از مقداردهی موازی باشیم (Q_2 صفر باشد) حاصل NAND یک می‌شود و هر کدام از Rها و Sها در فلیپ فلاب‌ها، یک می‌گیرند و این حالت یعنی فلیپ فلاب فعالیت نرم‌مال خود را انجام می‌دهد.

$$S_x = Q_2 \text{ NAND } I_x, \quad R_x = Q_2 \text{ NAND } I_x'$$

بخش عملی ::

مدار زیر در برنامه پروتئوس طراحی شده و در قسمت Export Graphics خروجی تصویری گرفته شده است.

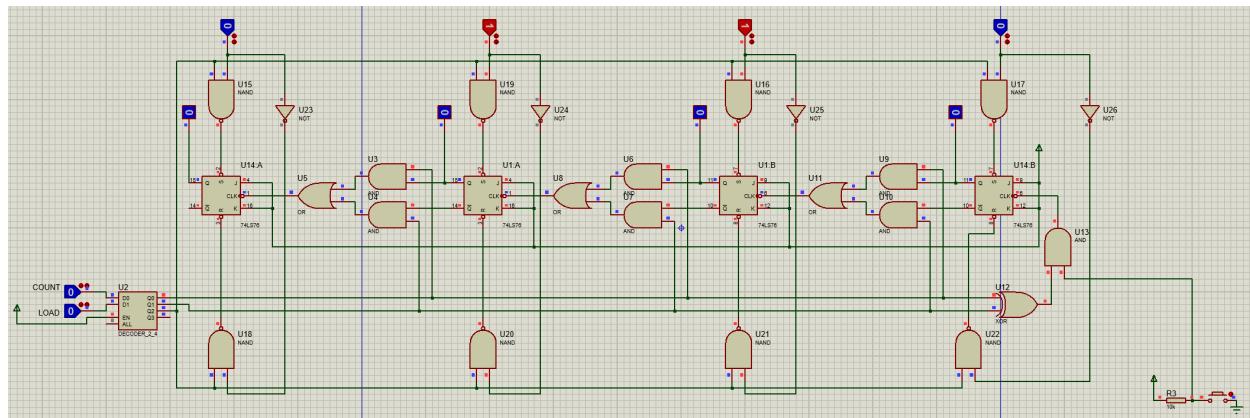


۹- مدار شمارنده به بالا، به باین، و مقدار دهن، موازی

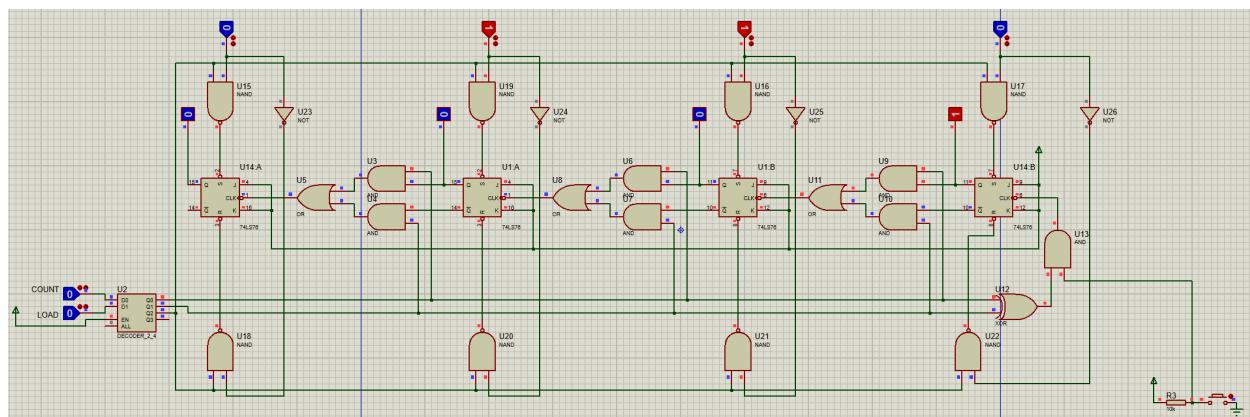
ضمناً يك حالت چهارمی در جدول وجود داشت که مربوط به do nothing بود. همانطور که در مدار پیداست، خروجی چهارم دیکدر را به هیچ جا وصل نکردیم. لذا در آن حالت هیچ اتفاقی نخواهد افتاد چون هیچ اتصالی به مدار ندارد.

تست مدار: اکنون جا دارد که مدار را تست کنیم.

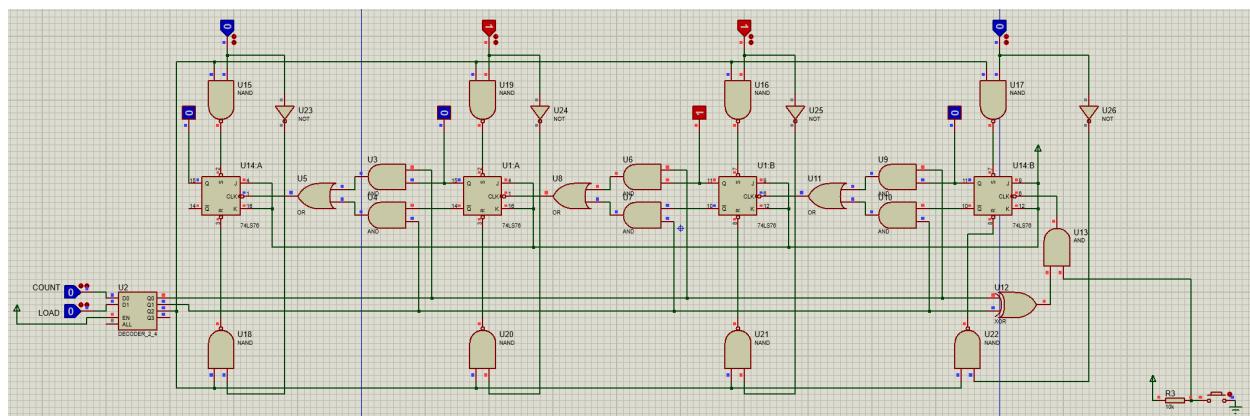
تست شمارش به بالا



-شمارش به بالا (اکنون صفر، با کلک بعدی باید به ۱ برویم)

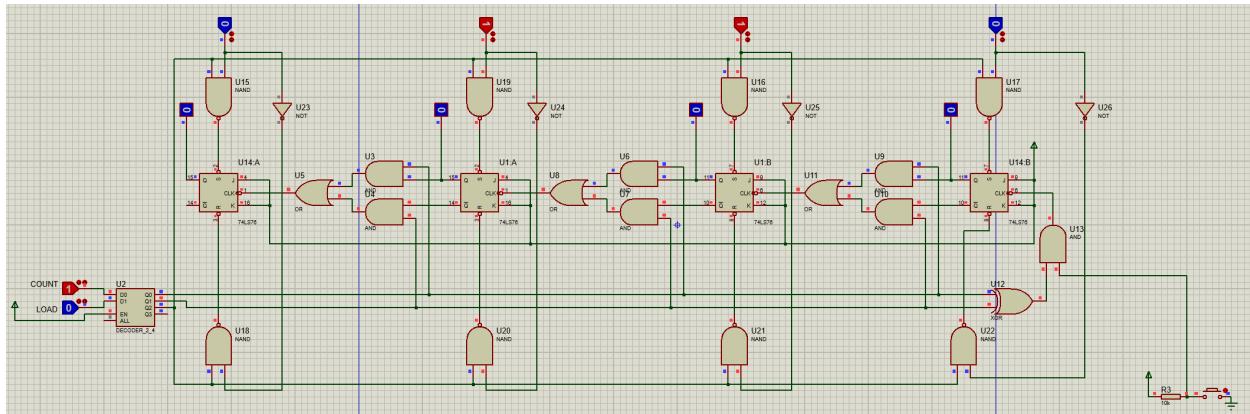


-شمارش به بالا (اکنون ۱، با کلک بعدی باید به ۲ برویم)

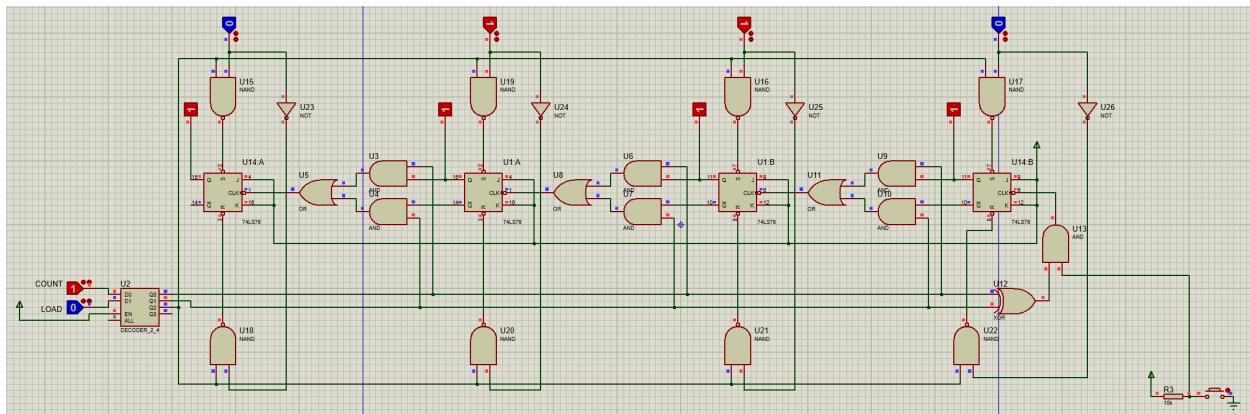


-شمارش به بالا (اکنون ۲، با کلک بعدی باید به ۳ برویم)

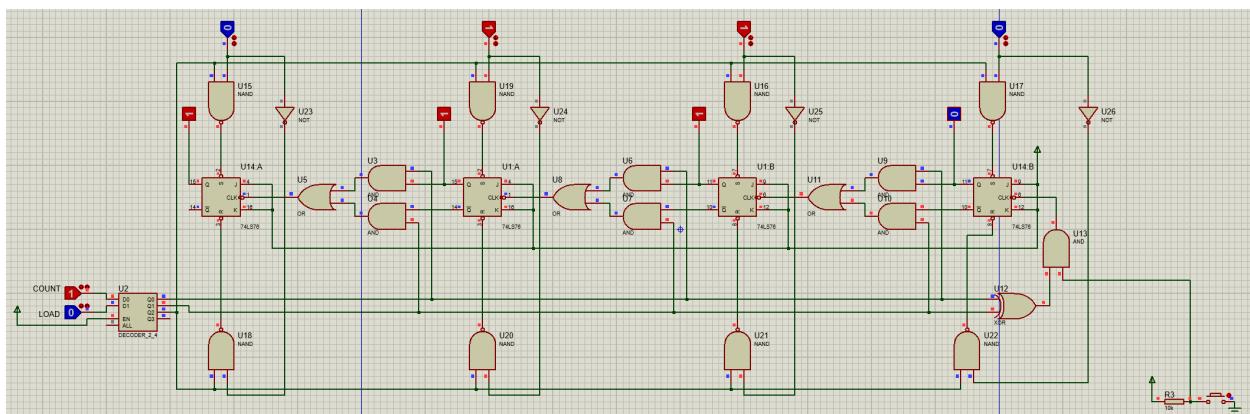
تست شمارش به پایین



-۱۳- شمارش به پایین (اکنون صفر، با کلاک بعدی باید به ۱۵ برویم)

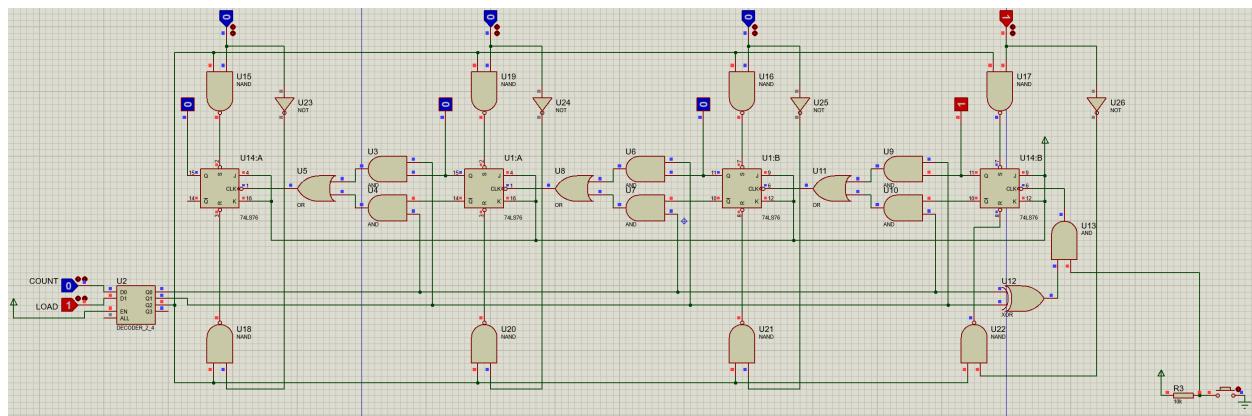


-۱۴- شمارش به پایین (اکنون ۱۵، با کلاک بعدی باید به ۱۴ برویم)

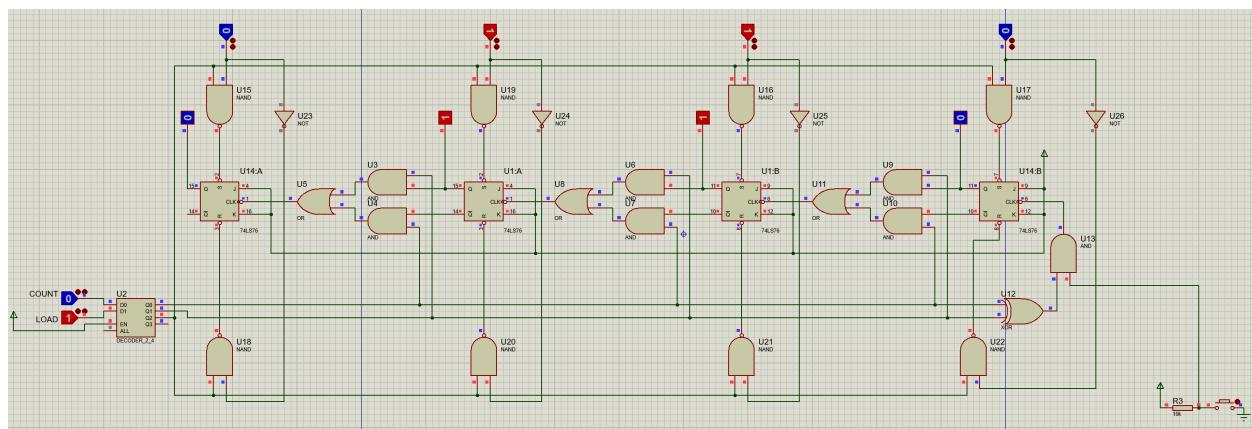


-۱۵- شمارش به پایین (اکنون ۱۴، با کلاک بعدی باید به ۱۳ برویم)

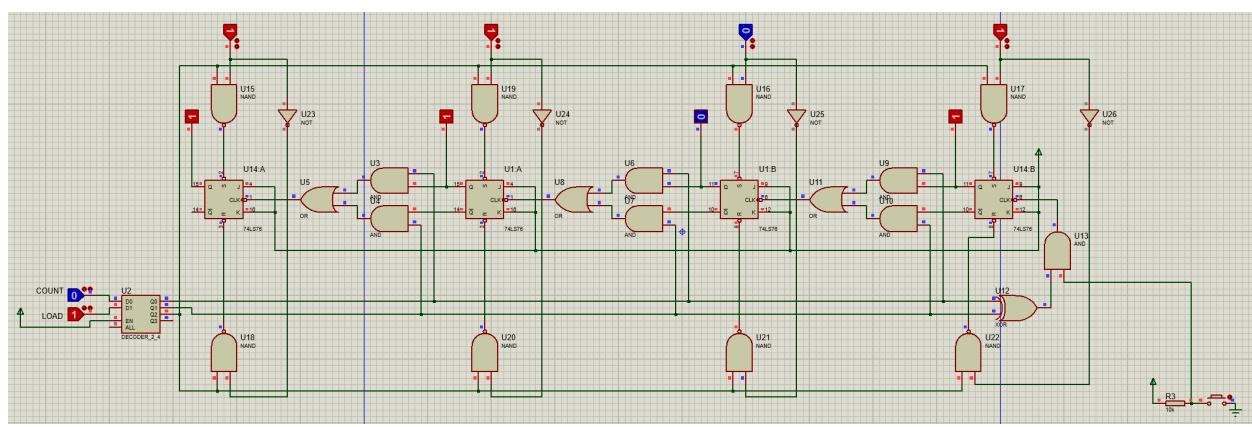
تست بارگذاری موازی



-۱۶- مقداردهی موازی عدد یک



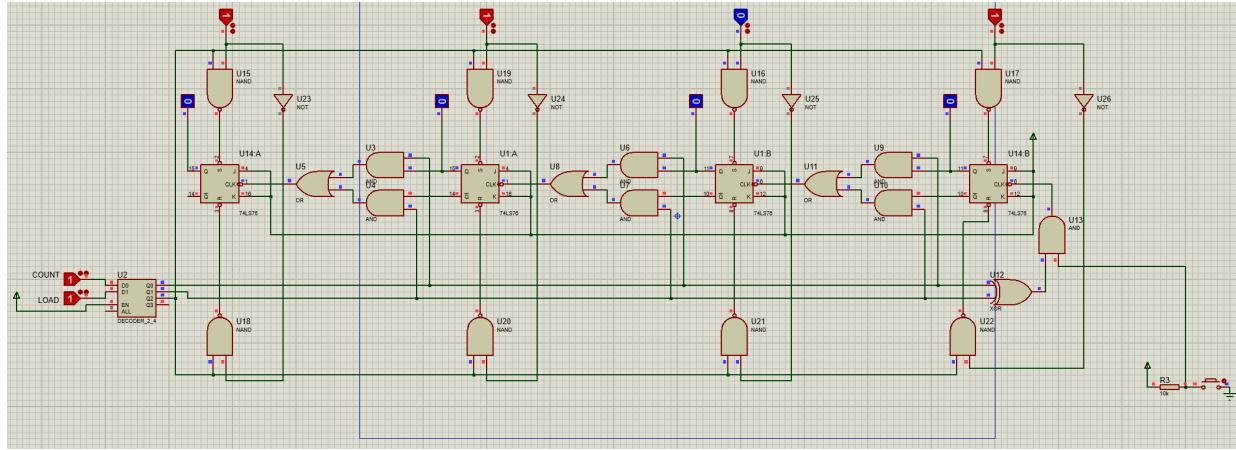
-۱۷- مقداردهی موازی عدد شش



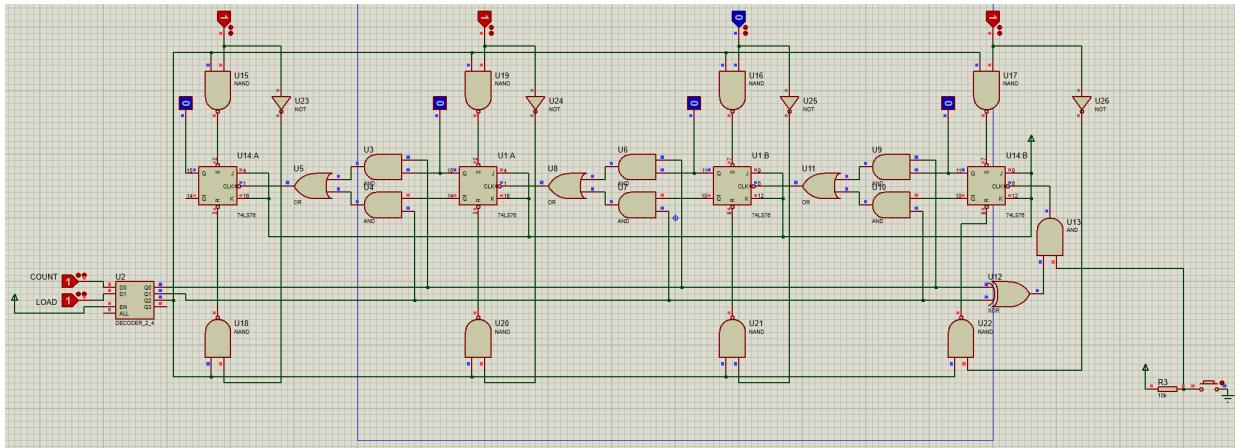
-۱۸- مقداردهی موازی عدد سیزده

تست Do Nothing

برای نشان دادن بهتر عملکرد مدار یک ویدیوی کوتاه چندثانیه‌ای ضمیمه شده است.



۱۹- در حالت Do Nothing هیچ کاری انجام نمی‌شود. (قبل فشردن کلاک)



۲۰- در حالت Do Nothing هیچ کاری انجام نمی‌شود. (بعد فشردن کلاک)

شمارنده باینری سنکرون

:: بخش تئوری ::

مدار موردنظر ما باید اعداد صفر تا هفت را سه تا سه بشارد. یعنی پس از شمارش عدد X , عدد $X+3$ را بشمارد. پس در این بخش باید با جدول درستی و جدول کارنوهای متعددی به طراحی مدار موردنظر برسیم. سنکرون بودن مدار باعث می‌شود هر سه فلیپ‌فلاپ از یک کلاک مشترک استفاده کنند و همگام باشند.

Present State			Input	Next State			Flip-Flop Inputs					
Q ₂	Q ₁	Q ₀	X	Q ₂	Q ₁	Q ₀	J ₂	K ₂	J ₁	K ₁	J ₀	K ₀
.	.	.	1	0	1	1	0	x	1	x	1	x
.	.	1	1	1	0	0	1	x	0	x	x	1
.	1	0	1	1	0	1	1	x	x	1	1	x
.	1	1	1	1	1	0	1	x	x	0	x	1
1	.	.	1	1	1	1	x	0	1	x	1	x
1	.	1	1	0	0	0	x	1	0	x	x	1
1	1	0	1	0	0	1	x	1	x	1	1	x
1	1	1	1	0	1	0	x	1	x	0	x	1
.	.	.	0	1	0	1	1	x	0	x	1	x
.	.	1	0	1	1	0	1	x	1	x	x	1
.	1	0	0	1	1	1	1	x	x	0	x	1
.	1	1	0	0	0	0	0	x	1	1	x	1
1	.	.	0	0	0	1	x	1	0	x	1	x
1	.	1	0	0	1	0	x	1	1	x	x	1
1	1	0	0	0	1	1	x	1	x	0	1	x
1	1	1	0	1	0	0	x	0	x	1	x	1

- جدول درستی (شامل Next State و ورودی فلیپفلابها)

در جدول درستی فوق، جا دارد یک نکته در خصوص JKFF را یادآوری کنیم. ($Q^+ = JQ' + K'Q$)

- اگر $J=1$ و $K=1$ باشند، عمل تاگل اتفاق می‌افتد.
- اگر $J=1$ و $K=0$ باشند، عمل ست اتفاق می‌افتد.
- اگر $J=0$ و $K=1$ باشند، عمل ریست اتفاق می‌افتد.
- اگر $J=0$ و $K=0$ باشند، عمل حفظ اتفاق می‌افتد.

اکنون در ادامه انتظار می‌رود با رسم شش جدول کارنو برای هر ورودی (سه تا J و سه تا K) ضابطه هر کدام را مشخص کنیم.

اما با دقت کردن در جدول متوجه می‌شویم که در هیچ ردیفی J_i و K_i مقدار متفاوت ندارند. در هر ردیف همواره یکی از آن‌ها صفر یا یک است و دیگری don't care است. در نتیجه می‌توانیم J_i را هم همارا ازش با آن یکی مقدار در نظر بگیریم. و در نهایت با رسم سه جدول کارنو به راه حل خلاصه‌تری دست پیدا کنیم. (هر جدول برای هر دوی K_i و J_i استفاده می‌شود)

		Q ₂ Q ₁	00	01	11	10
		Q ₀ x	00	1	1	1
		00	0	1	1	0
		01	1	0	0	1
		11	1	1	0	1
		10	1	0	0	1

-جدول کارنوی $J_2 \oplus K_2$ ۲۴

		Q ₂ Q ₁	00	01	11	10	
		Q ₀ x	00	0	0	0	0
		00	1	1	1	1	
		01	1	0	0	1	
		11	0	0	0	0	
		10	1	1	1	1	

-جدول کارنوی $J_1 \oplus K_1$ ۲۳

		Q ₂ Q ₁	00	01	11	10	
		Q ₀ x	00	1	1	1	1
		00	1	1	1	1	
		01	1	1	1	1	
		11	1	1	1	1	
		10	1	1	1	1	

-جدول کارنوی $J_0 \oplus K_0$ ۲۲

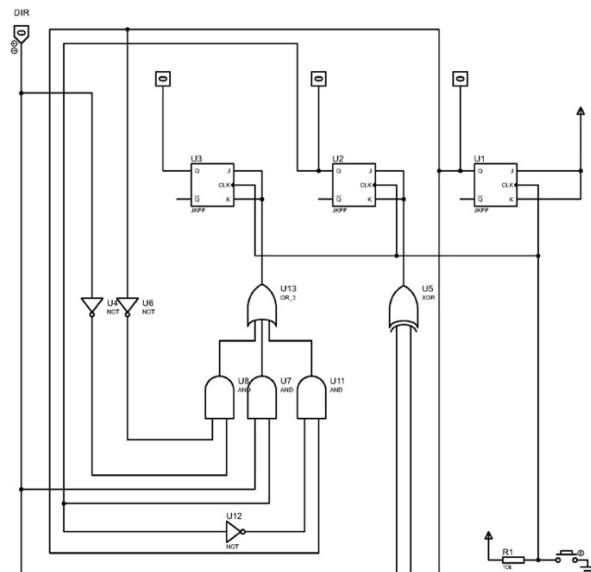
$$J_0 = K_0 = 1$$

$$J_1 = K_1 = Q_0 \oplus x$$

$$J_2 = K_2 = Q'_0 x' + Q_1 x + Q'_1 Q_0$$

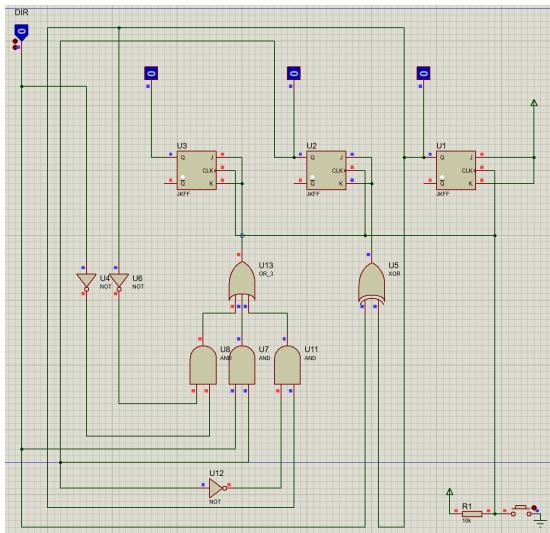
بخش عملی

مدار زیر در برنامه پروتوس طراحی شده و در قسمت Export Graphics خروجی تصویری گرفته شده است.

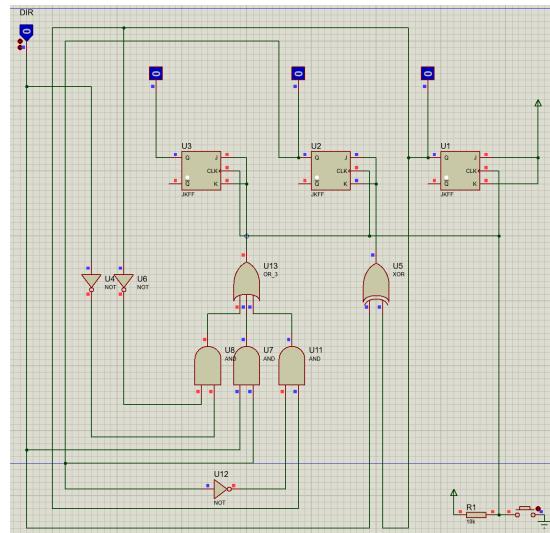


-مدار شمارنده سه تا سه تای سنکرون ۲۵

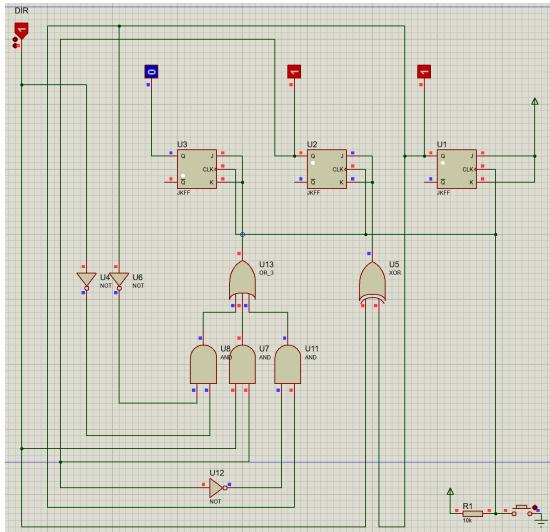
تست مدار: اکنون که مدار را ساختیم، آن را تست می‌کنیم. ابتدا در حالت صفر هستیم. با $DIR=0$ سه تا به سمت پایین می‌شماریم. در نتیجه به عدد ۵ خواهیم رسید. سپس با $DIR=1$ سه تا به سمت بالا می‌شماریم و مجدداً صفر می‌شود. سپس با همان جهت سه تا دیگر به بالا می‌شماریم و به ۳ می‌رسیم.



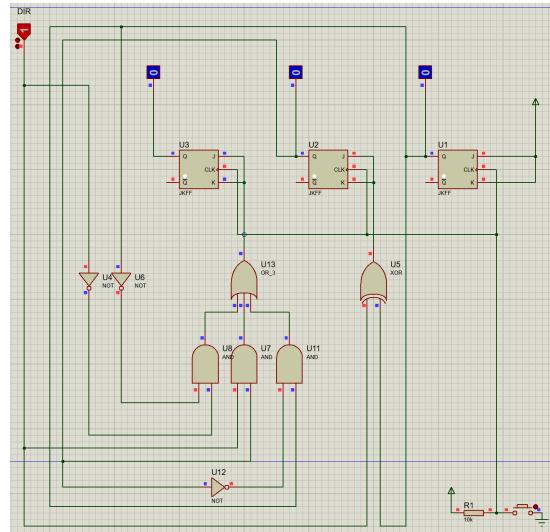
۲۷- عدد پنج (می خواهیم به بالا بشماریم و به صفر بررسیم)



۲۶- عدد صفر (می خواهیم به پایین بشماریم و به پنج بررسیم)



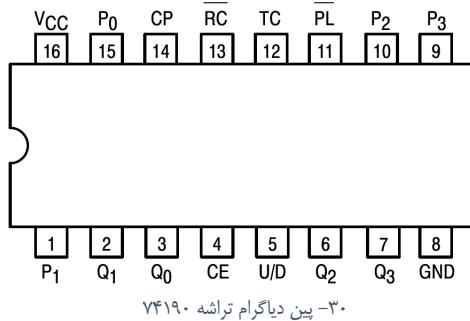
۲۹- عدد سه (اگر به بالا بشماریم به شش می رسیم)



۲۸- عدد صفر (می خواهیم به بالا بشماریم و به سه بررسیم)

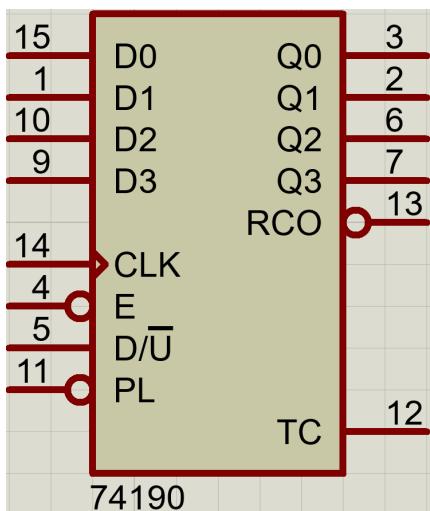
شمارنده BCD

:: بخش تئوری ::



در این قسمت جذاب آزمایش می‌خواهیم یک شمارنده بسازیم که اعداد صفر تا ۶۳ را دوجهته برای ما بشمارد و آن را با 7-segLED نمایش دهد.

ما می‌خواهیم از یک تراشه به نام 74190 استفاده کنیم که درواقع یک شمارنده BCD است. ابتدا به بررسی این تراشه می‌پردازیم.



۷۴۱۹۰-تراشه در نرم‌افزار پروتوس

(صفر تا نه) توسط شمارنده انجام شده است و از آن به عنوان کلاک شمارنده بعدی استفاده می‌کنیم.

- ورودی‌های D_0 و D_1 و D_2 و D_3 برای مقداردهی هستند.
- خروجی‌های Q_0 و Q_1 و Q_2 و Q_3 از فلیپ‌فلاپ خارج می‌شوند و ما آن‌ها را به پایه‌های الای‌دی وصل می‌کنیم.
- ورودی CLK کلاک است. (بدیهی)
- ورودی E فعال‌کننده شمارش است. (enable)
- ورودی D/U مشخص کننده شمارش به بالا (Up) یا پایین (Down) است.
- خروجی RCO زمانی استفاده می‌شود که یک چرخه کامل

قطعات موردنیاز:

1. دو تراشه 74190
2. دو 7-SEG-BCS (GRN سبز)
3. گیت‌های منطقی موردنیاز، کلاک و LogicState

نحوه ساخت مدار:

به این صورت است که شمارنده ما شمارش در جهت بالا را از صفر شروع می‌کند و وقتی خواست از ۶۴ به ۶۳ برود، به جای ۶۴ صفر می‌شود. همچنین در شمارش پایین هرگاه از صفر خواست به ۹۹ برود، به جای ۹۹ به ۶۳ می‌رود. لذا همانطور که پیداست باید در خروجی‌های دو شمارنده، تعدادی گیت به کار ببریم که ۹۹ و ۶۴ را شناسایی کنند. به این منظور از یک گیت AND برای شناسایی عدد ۶۴ استفاده می‌کنیم. هنگامی که شمارنده کوچکتر ۴ و شمارنده بزرگ‌تر ۶ را نشان داد، خروجی AND یک می‌شود. به طور مشابه یک گیت AND دیگر برای شناسایی ۹۹ با همان روش می‌گذاریم. خروجی این دوتا AND را باهم وارد یک OR می‌کنیم. به عبارت دیگر به محض اتفاق افتادن یکی از آن دو حالت (رفتن به ۶۴ یا ۹۹) دستگاه به حالت load می‌رود و مقادیر مخصوصی را از input می‌گیرد. به همین منظور خروجی OR را به PL و نات آن را به E وصل می‌کنیم. مشخص کردن ورودی‌ها هم به این صورت است:

- اگر در حال شمارش به بالا باشیم (DIR=0) و از ۶۳ بخواهیم بالاتر برویم، ورودی‌های هر دو شمارنده باید 0000 باشند. پس به چهار ورودی هر دو شمارنده صفر را می‌دهیم.
- اگر در حال شمارش به پایین باشیم (DIR=1) و از صفر بخواهیم پایین‌تر برویم، ورودی‌های شمارنده یکان باید 0011 و ورودی‌های شمارنده دهگان باید 0110 باشد تا عدد ۶۳ نمایش داده شود.

اما یک روش خلاقانه برای تخصیص بیت‌های ورودی این است که بینیم شمارنده اول و دوم در هر دو حالت چگونه ورودی می‌گیرند و آن بیت‌هایی که در هر دو حالت صفر هستند را از gnd بگیریم. و بیت‌های دیگر که متغیر هستند را از DIR. *

شمارنده یکان دو حالت 0000 و 0011 را دارد. پس دو صفر سمت چپ را به gnd که همیشه صفر است وصل می‌کنیم. دو بیت سمت راستی دقیقاً معادل همان DIR در حالت مربوطه هستند و به همان DIR وصل می‌شوند.

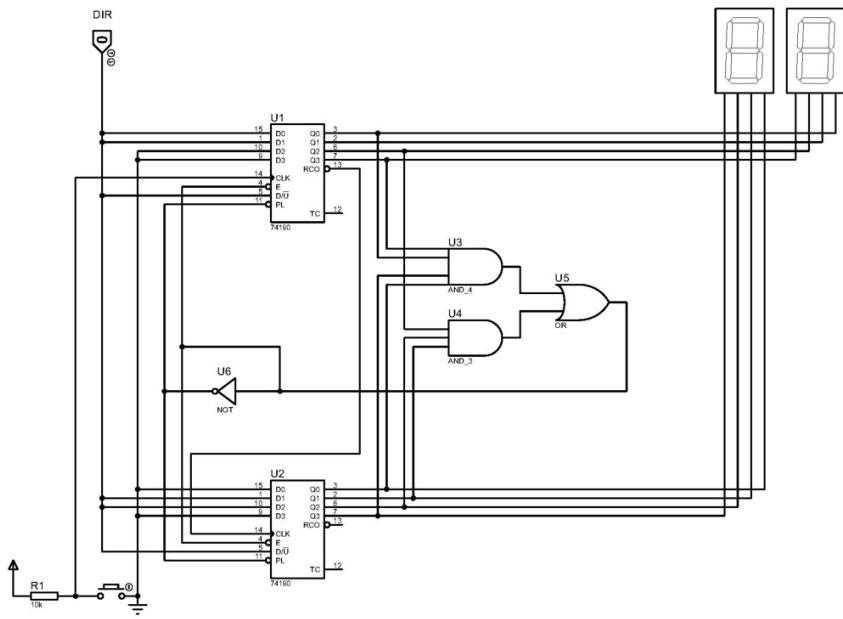
* شمارنده دهگان دو حالت 0000 و 0110 را دارد. دو بیت ابتدا و انتهای طبق توضیحات بالا به gnd وصل می‌شوند و دو بیت وسطی به DIR.

...بخش عملی ...

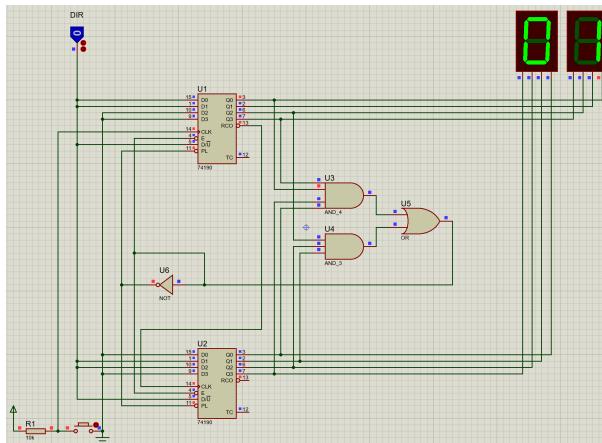
مدار را طبق توضیحات گفته شده طراحی می‌کنیم. خروجی‌های فلیپ‌فلاپ را به ترتیب به LED‌ها وصل می‌کنیم. از خروجی TC استفاده‌ای نکرده‌ایم.

مدار زیر در برنامه پروتئوس طراحی شده و در قسمت Export Graphics خروجی تصویری گرفته شده است.

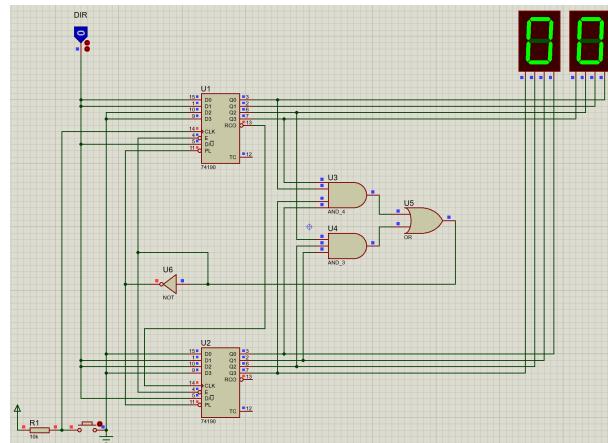
تست مدار: در ادامه آن را تست هم می‌کنیم تا از صحت عملکرد آن مطمئن شویم.



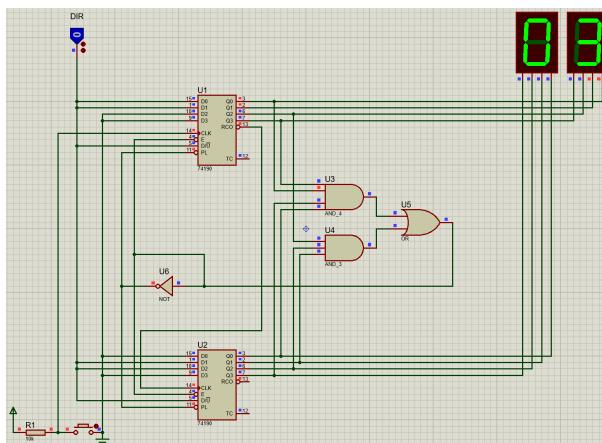
٣٢- شمارنده BCD صفر تا



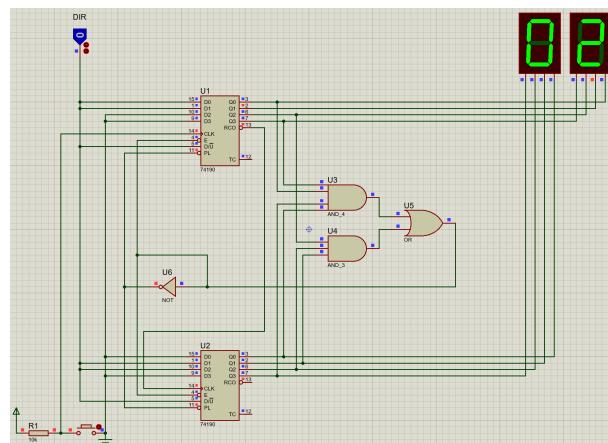
٣٤- نمایش عدد یک (در شمارش بالا به ۲ خواهیم رفت)



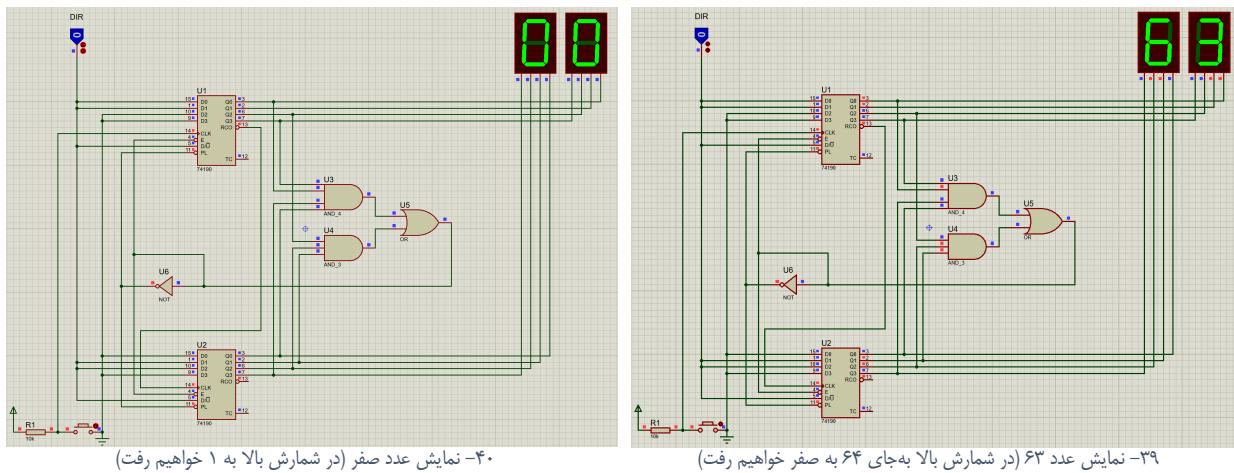
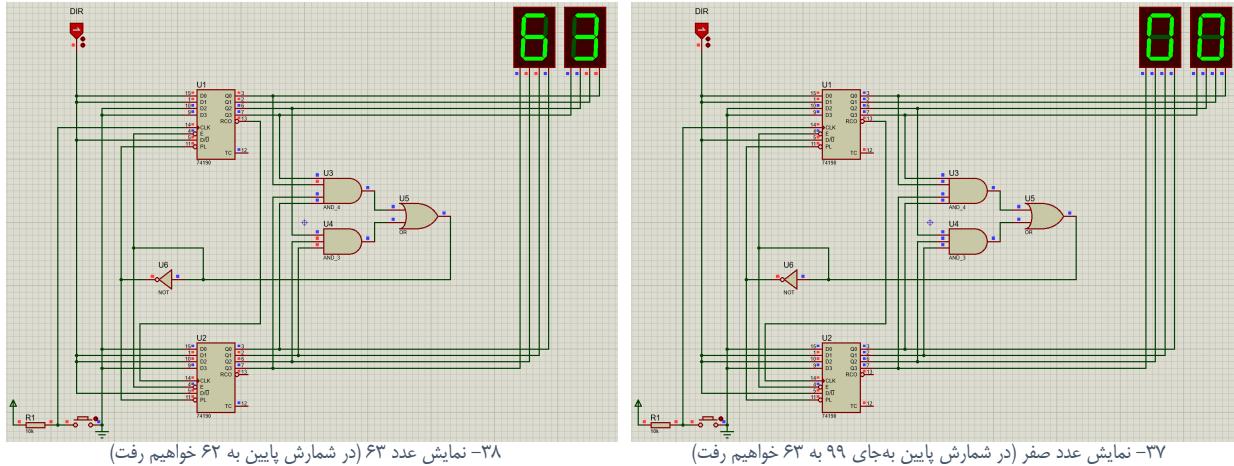
٣٣- نمایش عدد صفر (در شمارش بالا به ۱ خواهیم رفت)



٣٥- نمایش عدد دو (در شمارش بالا به ۴ خواهیم رفت)



٣٦- نمایش عدد سه (در شمارش بالا به ٣ خواهیم رفت)



*امتياز: شمارنده BCD با قابليت بارگذاري

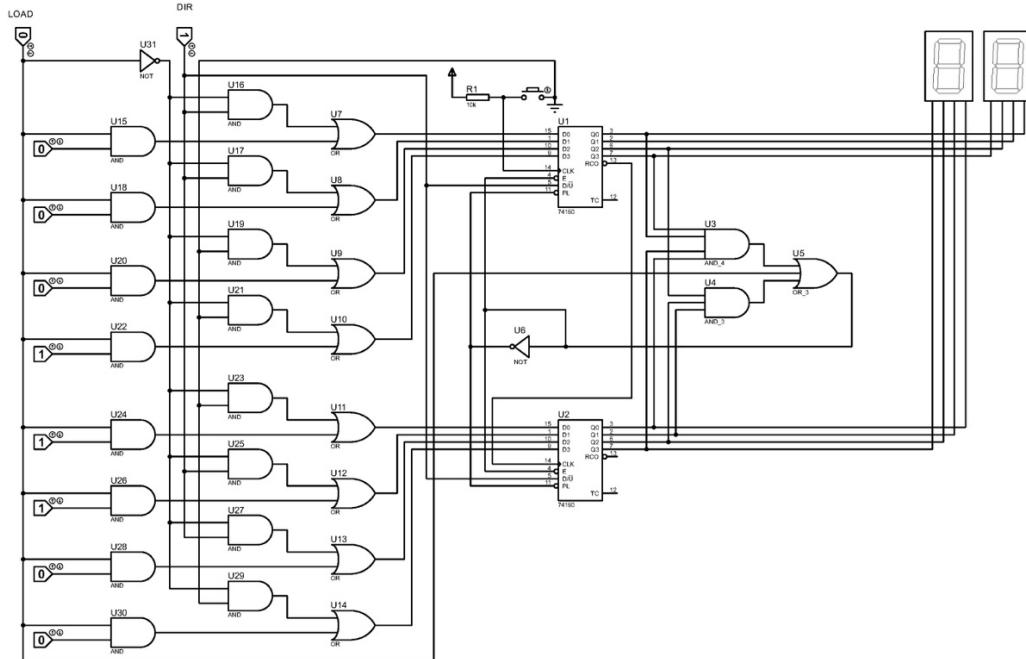
:: بخش تئوري ::

مي خواهيم همان مدار قبلی را به گونه‌اي بسازيم که اين قابليت را داشته باشد که يك عدد دو رقمي که ما برایش مشخص می‌کنيم را هم نشان دهد. به اين منظور ما از ۸ لاجيك استيت برای بيان عدد موردنظرمان بهره می‌بريم. چهارتا را برای شمارنده يکان و چهارتا را برای شمارنده دهگان استفاده می‌کنيم. به منظور تعين عملكرد مدار يك ورودي ديگر به اسم load را به مدار اضافه می‌کنيم. هرگاه مقدار آن ۱ باشد، يعني مي خواهيم عمل بارگذاري را انجام دهيم و شمارش نداريم. هرگاه صفر باشد يعني بارگذاري نداريم و شمارش انجام مي شود. به همين خاطر خود load را با هر بيت ورودي AND می‌کنيم. همچنين نات load را با بيتی که قبلًا وارد شمارنده می‌شد AND می‌کنيم. خروجي اين دو گيت AND را باهم OR می‌کنيم. به علاوه، برای اينكه ورودي PL مدار هم تحت تأثير قرار بگيرد، گيت OR قدими که به E و P وصل

کرده بودیم، را با یک OR سه ورودی تعویض می‌کنیم و مقدار load را هم به آن می‌دهیم. به این ترتیب مدار ما عملکرد موردنظر را خواهد داشت.

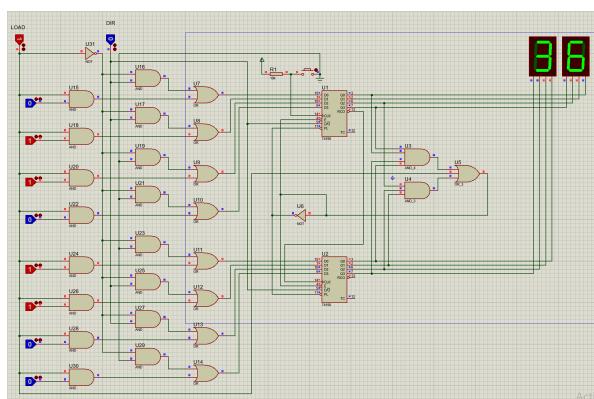
...بخش عملی ...

مدار را طبق توضیحات گفته شده می‌سازیم و از آن خروجی تصویری با استفاده از گزینه Export Graphic می‌گیریم.
یک نکته: هنگام ورودی دادن باید حواسمن باشد که ورودی هر شمارنده باید صفر تا نه باشد. اگر مثلاً ۱۱۰۱ معادل ۱۳ را بدھیم، نمایشگر LED ممکن است تصویر غیرمنتظره‌ای نمایش دهد.

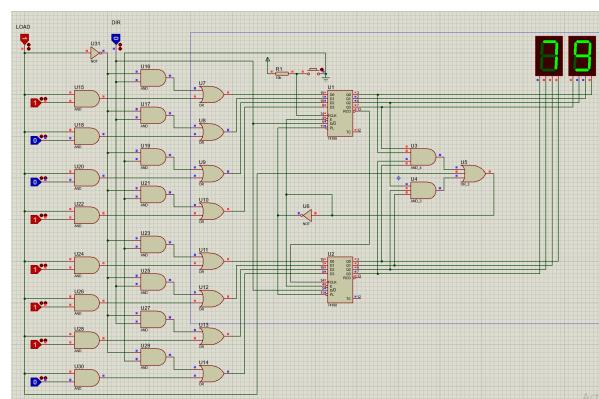


۹۹- شمارنده BCD صفر تا ۶۳ + قابلیت بارگذاری اعداد صفر تا ۴۱

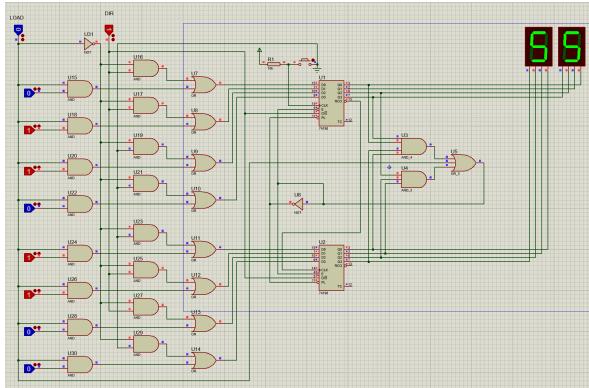
تست مدار: در ادامه آن را تست هم می‌کنیم تا از صحت عملکردش مطمئن شویم.



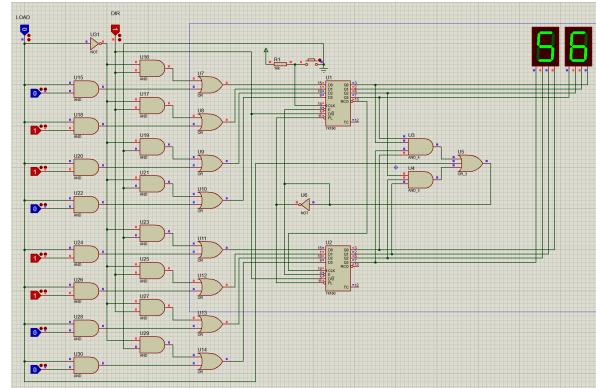
(load=1) ۴۳- بارگذاری عدد ۳۶



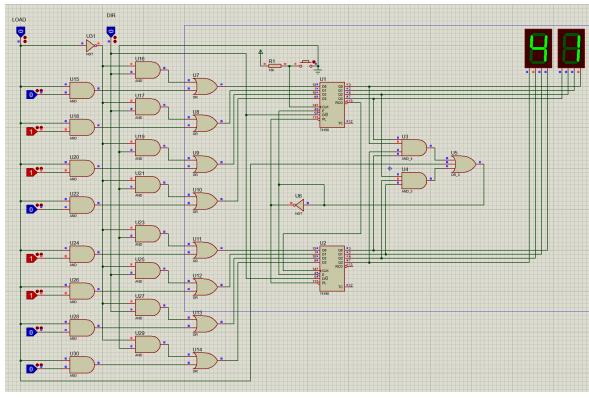
(load=1) ۴۴- بارگذاری عدد ۷۹



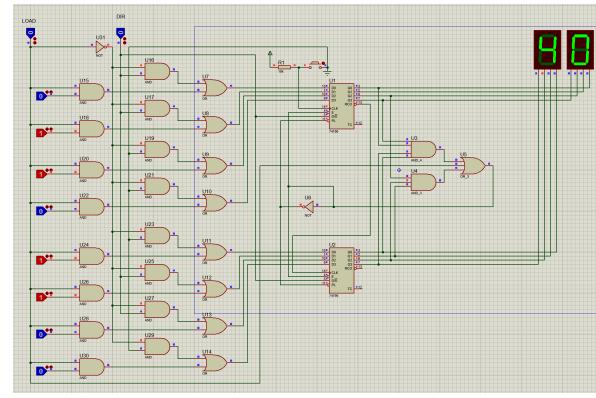
-نمایش عدد ۵۵ (در شمارش پایین به ۵۴ خواهیم رفت) (load=0)



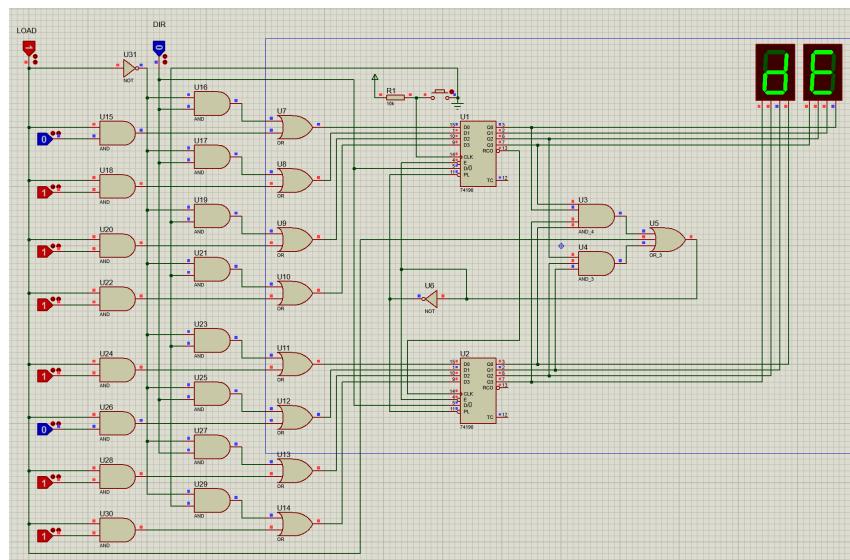
-نمایش عدد ۵۶ (در شمارش پایین به ۵۵ خواهیم رفت) (load=0)



-نمایش عدد ۴۱ (در شمارش بالا به ۴۱ خواهیم رفت) (load=0)



-نمایش عدد ۴۰ (در شمارش بالا به ۴۱ خواهیم رفت) (load=0)



-بارگذاری غیرمجاز منجر به نمایش حروف غیرمنتظره شد. یکان و دهگان باید از صفر تا ۹ باشند. (load=1)

جمع‌بندی

در آزمایش سوم درس آزمایشگاه مدارهای منطقی به آشنایی با نحوه پیاده‌سازی یک شمارنده آسنکرون برای شمارش، یک شمارنده آسنکرون دیگر مشابه آن با قابلیت بارگذاری موازی، یک شمارنده سنکرون با قابلیت شمارش سه‌تا و یک شمارنده BCD با قابلیت نمایش اعداد روی LED پرداختیم.

ما در بخش اول شمارنده آسنکرون در دو جهت را با فلیپ‌فلاب‌های 74LS73 از نوع JK ساختیم که طرح آن در دستور کار آزمایشگاه موجود بود. در بخش بعدی از روی همان شمارنده، یک شمارنده دیگر ساختیم که هم قابلیت شمارش در دو جهت را دارد و هم قابلیت بارگذاری موازی. به این منظور ما از ورودی‌های کنترلی برای تعیین عملکرد مدار استفاده کردیم. همچنین نوع فلیپ‌فلاب‌ها را به 74LS76 از نوع JK تغییر دادیم تا ورودی R و S داشته باشد. همچنین تعدادی گیت برای تعیین ورودی‌های R و S به منطق مدار اضافه شد.

ما در بخش دوم، یک شمارنده سنکرون برای شمارش سه‌تا از صفر تا هفت ساختیم. برای پیاده‌سازی منطق این مدار ما یک جدول درستی تشکیل دادیم که در ردیف‌های آن حالت فعلی، حالت بعدی و ورودی‌های JKFF را مشخص کردیم.

همه این ردیف‌ها یک بار به ازای $X=0$ و یک بار به ازای $X=1$ نوشته شدند. (X همان جهت شمارش است) پس از آنکه ورودی‌های فلیپ‌فلاب در جدول مشخص شدند، برای هر کدام جدول کارنو رسم کردیم و از روی جدول کارنو ضابطه آن‌ها را به دست آوردیم. ضابطه K_i و J_i به خاطر حضور don't care ‌ها یکسان شد. سپس با توجه به این ضابطه‌های نهایی، مدار را ساختیم.

ما در بخش سوم، یک شمارنده BCD برای شمردن از صفر تا ۶۳ در دو جهت ساختیم. نکته مهم در ساختن شمارنده این است که در شمارش بالا، پس از ۶۳ (به جای ۶۴) به صفر برود. به طور مشابه در شمارش پایین، از صفر (به جای ۹۹) به ۶۳ برود. ما در مدارمان از تراشه ۷۴۱۹۰ استفاده کردیم. خروجی‌های آن را به 7-SEG-LED سبز وصل کردیم. همچنین خروجی‌های آن را به تعدادی گیت AND بردیم تا وقوع ۶۴ و ۹۹ را شناسایی کند. هر زمان که یکی از این دو حالت اتفاق بیفتد، مدار ما به جای شمارش عدد بعدی، حالت load را به خود می‌گیرد و رشته‌هایی که خودمان به عنوان ورودی مشخص کردہ‌ایم را دریافت می‌کند و در LED نمایش می‌دهد. رشته‌های ورودی ما فقط دو حالت صفر و ۶۳ را دارند که بیت‌های آن‌ها از DIR (بیت جهت) و gnd می‌آیند. به این صورت مدار ما توانست از صفر تا ۶۳ را بشمارد و حالت ابتدا و انتهای را به درستی پشت سر گذاشت.

سپس به عنوان بخش امتیازی یک مدار دیگر طراحی کردیم که علاوه بر قابلیت شمارش مطابق مدار قبلی، بتواند بارگذاری عدد داده شده را هم انجام دهد. لذا با این مدار می‌توانیم به اعداد ۶۴ الی ۹۹ که در مدار قبلی هرگز دیده نمی‌شوند، دسترسی داشته باشیم و آن‌ها را از طریق بارگذاری روی LED نمایش دهیم.

منابع

Digital Design, M. Morris Mano

منبعی که از تصاویر موجود در آن استفاده شده است:

http://www.icdictionary.com/single_device.php?DeviceName=74190,%2074190.IEC,%20SN74LS190