

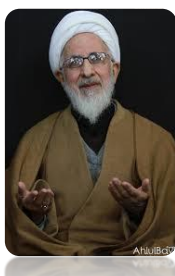
آزمایشگاه مبانی دیجیتال

تقديم به دانشجویان رشته ی مخابرات ، کامپیوتر، الکترونیک

تکنولوژی ارتباطات و فناوری اطلاعات



مؤلف : مهندس رضا علیپور



سخنی از رهبر معظم انقلاب اسلامی دانشمند محترم و فیلسوف گرامی حضرت آیت الله العظمی امام خامنه ای

﴿مَدَّ ظِلُّهُ الْعَالِي﴾ ، ﴿ادامَ اللهُ ظِلَّ الْوَارِثِ﴾

﴿تعلیم، تعلم، تأدب عبادت است﴾

سخنی از دانشمند محترم و فیلسوف گرامی حضر آیت الله العظمی نوری همدانی

﴿مَدَّ ظِلُّهُ الْعَالِي﴾ ، ﴿ادامَ اللهُ ظِلَّ الْوَارِثِ﴾

خامنه ای رهبر آست ، دشمن او آبر است

سخنی از دانشمند محترم و فیلسوف گرامی حضر آیت الله العظمی جوادی آملی

﴿مَدَّ ظِلُّهُ الْعَالِي﴾ ، ﴿ادامَ اللهُ ظِلَّ الْوَارِثِ﴾

اگر در میان افراد جامعه عاطفه و همدلی وجود نداشته باشد مانند همان سنگهای بی ملات فرو می ریزند

فهرست مطالب

حدود صفحات

مطالب

۴ الی ۱

گیت ها

۶ الی ۵

ساختن گیت ها ی مختلف به وسیله ی NAND

۹ الی ۷

قوانین جبر بول

۱۲ الی ۱۰

نمایشگر هفت قسمتی

۱۴ الی ۱۳

جمع کننده و تفریق کننده با استفاده از تراشه ی ۷۴۸۳

۱۶ الی ۱۵

تفریق کننده

۱۹ الی ۱۷

نیمه مقایسه کننده

۲۳ الی ۲۰

فلیپ فلاپ ها

۲۷ الی ۲۴

پالس ساعت و ساخت شمارنده ی آسنکرون

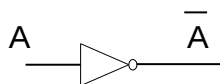
۳۱ الی ۲۸

ساخت دیکودر- انکودر- مالتی پلکسر - دی مالتی پلکسر

۳۶ الی ۳۲

شمارنده BCD

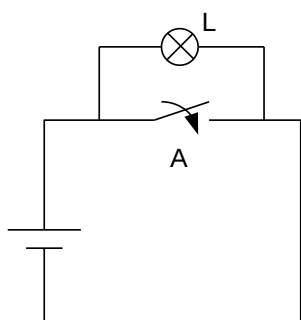
گیت ها



گیت NOT :

نکته : بر روی برد آزمایشگاه که ما کار می کنیم باید بدانیم که رنگ سبز به معنای منطقی و قرمز به معنای منطقی و همچنین نارنجی حالت خنثی می باشد.

حالت کلیدی گیت نات به صورت روبه رو می باشد و جدول صحت آن نیز مانند زیر می باشد.

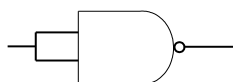


A	L
0	1
1	0

A	L
0.008	4.96
4.96	0.017

تبدیل گیت NAND و NOR به گیت NOT :

در صورتی که ورودی های گیت NAND و NOR را اتصال کوتاه کنیم تبدیل به گیت NOT می گردد.



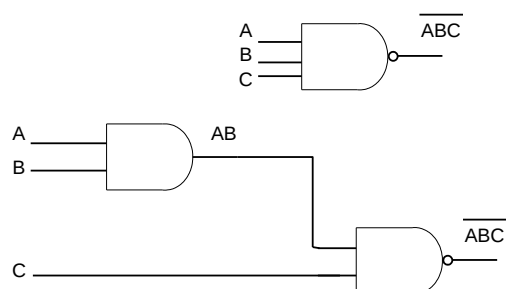
A	L	A	L
0	1	0.009	3.64
1	0	4.96	0.15

A	L
0	1
1	0

A	L
0.007	4.95
4.96	0.022

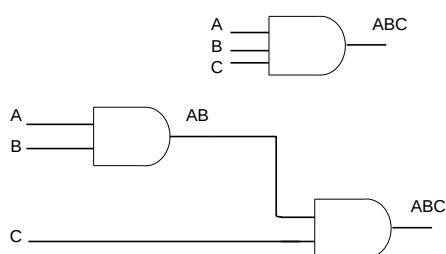
افزایش ورودی های گیت AND و OR و NOR و NAND

تبدیل گیت NAND سه پایه



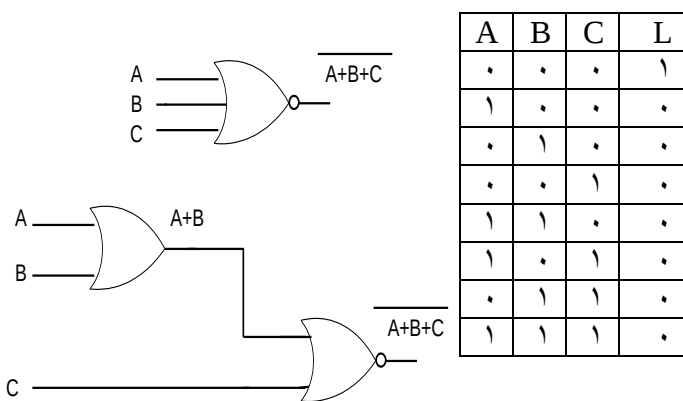
A	B	C	L
۰	۰	۰	۱
۱	۰	۰	۱
۰	۱	۰	۱
۰	۰	۱	۱
۱	۱	۰	۱
۱	۰	۱	۱
۰	۱	۱	۱
۱	۱	۱	۰

A	B	C	L
۰/۰۰۸	۰/۰۰۸	۰/۰۰۸	۳/۶۵
۴/۹۶	۰/۰۰۸	۰/۰۰۸	۳/۶۵
۰/۰۰۸	۴/۹۶	۰/۰۰۸	۳/۶۵
۰/۰۰۸	۰/۰۰۸	۴/۹۶	۳/۶۵
۴/۹۶	۴/۹۶	۰/۰۰۸	۳/۶۵
۴/۹۶	۰/۰۰۸	۴/۹۶	۳/۶۵
۰/۰۰۸	۴/۹۶	۴/۹۶	۳/۶۵
۴/۹۶	۴/۹۶	۴/۹۶	۰/۱۳

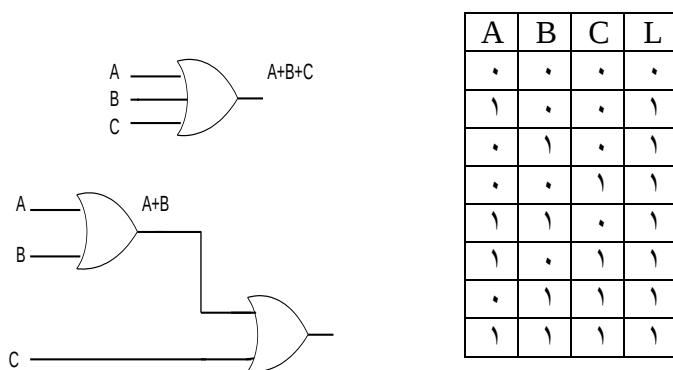


A	B	C	L
۰	۰	۰	۰
۱	۰	۰	۰
۰	۱	۰	۰
۰	۰	۱	۰
۱	۱	۰	۰
۱	۰	۱	۰
۰	۱	۱	۰
۱	۱	۱	۱

A	B	C	L
۰/۰۰۸	۰/۰۰۸	۰/۰۰۸	۰/۱۸
۴/۹۶	۰/۰۰۸	۰/۰۰۸	۰/۱۸
۰/۰۰۸	۴/۹۶	۰/۰۰۸	۰/۱۸
۰/۰۰۸	۰/۰۰۸	۴/۹۶	۰/۱۸
۴/۹۶	۴/۹۶	۰/۰۰۸	۰/۱۸
۴/۹۶	۰/۰۰۸	۴/۹۶	۰/۱۸
۰/۰۰۸	۴/۹۶	۴/۹۶	۰/۱۸
۴/۹۶	۴/۹۶	۴/۹۶	۴/۹۵



A	B	C	L
۰/۰۰۸	۰/۰۰۸	۰/۰۰۸	۴/۹۵
۴/۹۷	۰/۰۰۸	۰/۰۰۸	۰/۰۳
۰/۰۰۸	۴/۹۷	۰/۰۰۸	۰/۰۳
۰/۰۰۸	۰/۰۰۸	۴/۹۷	۰/۰۳
۴/۹۷	۴/۹۷	۰/۰۰۸	۰/۰۳
۴/۹۷	۰/۰۰۸	۴/۹۷	۰/۰۳
۰/۰۰۸	۴/۹۷	۴/۹۷	۰/۰۳
۴/۹۷	۴/۹۷	۴/۹۷	۰/۰۳

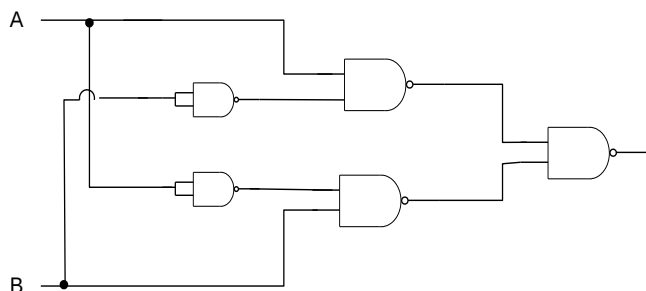


A	B	C	L
۰/۰۰۸	۰/۰۰۸	۰/۰۰۸	۰/۱۲
۴/۹۷	۰/۰۰۸	۰/۰۰۸	۳/۹۵
۰/۰۰۸	۴/۹۷	۰/۰۰۸	۳/۹۵
۰/۰۰۸	۰/۰۰۸	۴/۹۷	۳/۹۵
۴/۹۷	۴/۹۷	۰/۰۰۸	۳/۹۵
۴/۹۷	۰/۰۰۸	۴/۹۷	۳/۹۵
۰/۰۰۸	۴/۹۷	۴/۹۷	۳/۹۵
۴/۹۷	۴/۹۷	۴/۹۷	۳/۹۵



توجه

استفاده از فقط گیت NAND یک گیت XOR طراحی کرده و پیاده سازی کنید و سپس جدول صحت آن را به دست آورید.



A	B	L
۰	۰	۰
۰	۱	۱
۱	۰	۱
۱	۱	۰

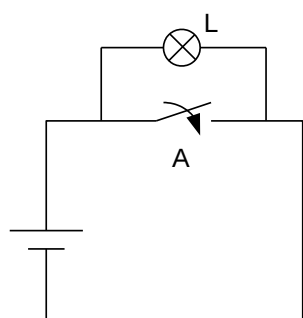
A	B	L
۰/۰۰۸	۰/۰۰۸	۰/۰۰۸
۰/۰۰۸	۴/۹۶	۴/۹۶
۴/۹۶	۰/۰۰۸	۴/۹۶
۴/۹۶	۴/۹۶	۰/۰۰۸



توجه

بر روی برد آزمایشگاه که ما کار می کنیم باید بدانیم که رنگ سبز به معنای منطقی و قرمز به معنای منطقی و همچنین نارنجی حالت خنثی می باشد.

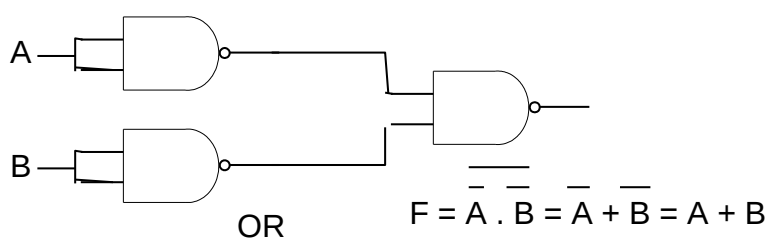
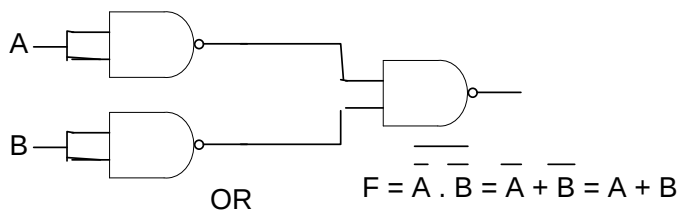
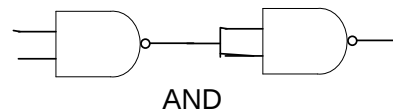
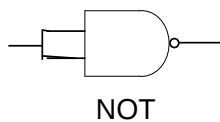
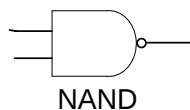
حالت کلیدی گیت نات به صورت روبه رو می باشد و جدول صحت آن نیز مانند زیر می باشد.



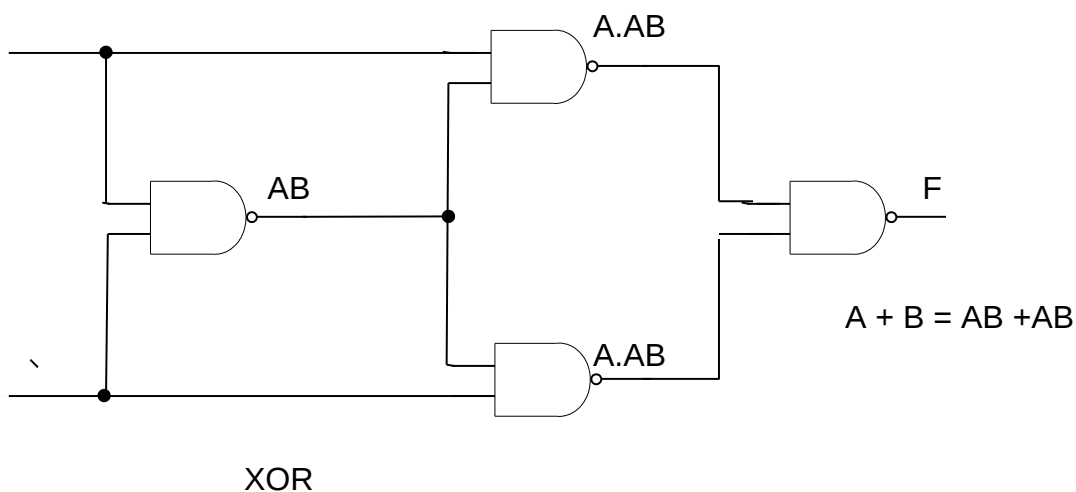
A	L
0	1
1	0

A	L
0.008	4.96
4.96	0.017

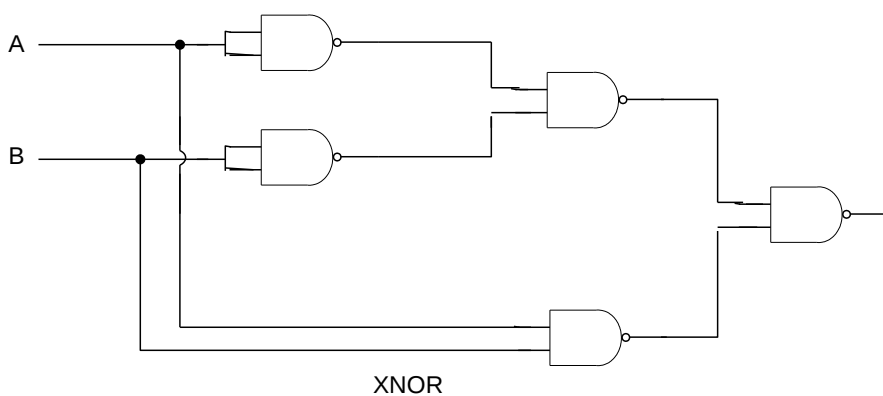
ساختن گیت های مختلف به وسیله ی NAND



A	B	F
۰	۰	۰
۰	۱	۱
۱	۰	۱
۱	۱	۱



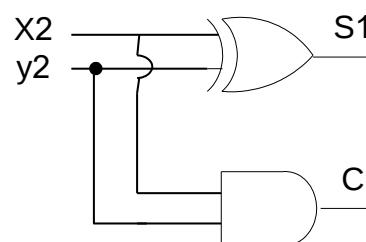
A	B	C
۰	۰	۰
۰	۱	۱
۱	۰	۱
۱	۱	۰



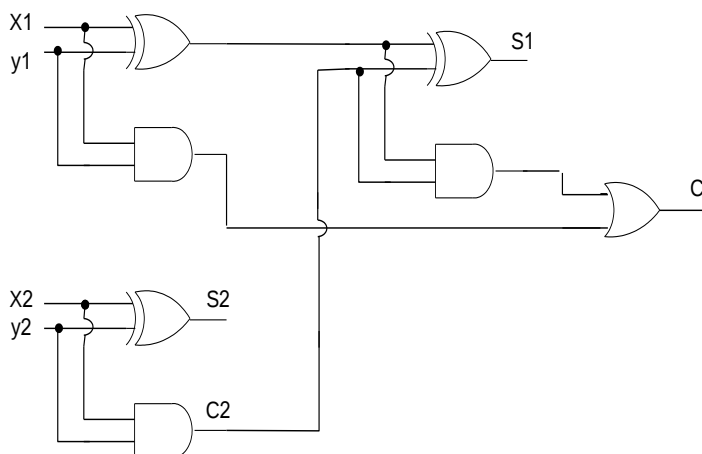
A	B	F
۰	۰	۱
۱	۰	۰
۰	۱	۰
۱	۱	۱

نیمه جمع کننده:

A	B	C	S1
۰	۰	۰	۰
۰	۱	۰	۱
۱	۰	۰	۱
۱	۱	۱	۰



تمام جمع کننده:



X1	X2	Y1	Y2	C	S1	S2
۰	۰	۰	۰	۰	۰	۰
۰	۰	۰	۱	۰	۰	۱
۰	۰	۱	۰	۰	۱	۰
۰	۱	۰	۰	۰	۱	۰
۰	۱	۰	۱	۰	۱	۱
۰	۱	۱	۰	۰	۱	۱
۰	۱	۱	۱	۰	۱	۰
۱	۰	۰	۰	۰	۱	۰
۱	۰	۰	۱	۰	۱	۱
۱	۰	۱	۰	۰	۱	۱
۱	۰	۱	۱	۰	۱	۰
۱	۱	۰	۰	۱	۰	۰
۱	۱	۰	۱	۱	۰	۱
۱	۱	۱	۰	۱	۱	۰
۱	۱	۱	۱	۱	۱	۰

قوانین جبر بول

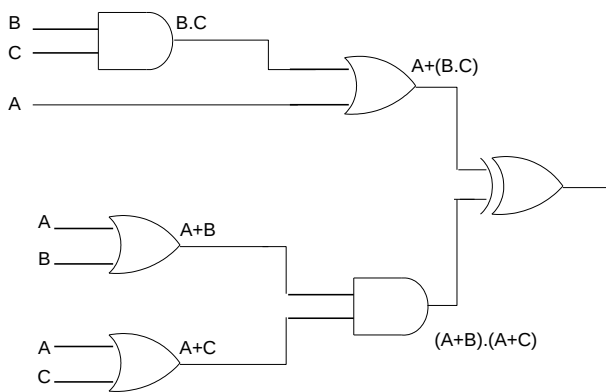
A	B	A+B	A.B
۰	۰	۰	۰
۱	۰	۱	۰
۰	۱	۱	۰
۱	۱	۱	۱

$$A \cdot (B+C) = (A \cdot B) + (A \cdot C) \quad -۲$$

$$A+(B \cdot C) = (A+B) \cdot (A+C) \quad -۱ \text{ قانون تویع پذیری:}$$

$$(A \cdot B) = A+B \quad -۴$$

$$(A+B) = A \cdot B \quad -۳ \text{ قانون دمورگان:}$$

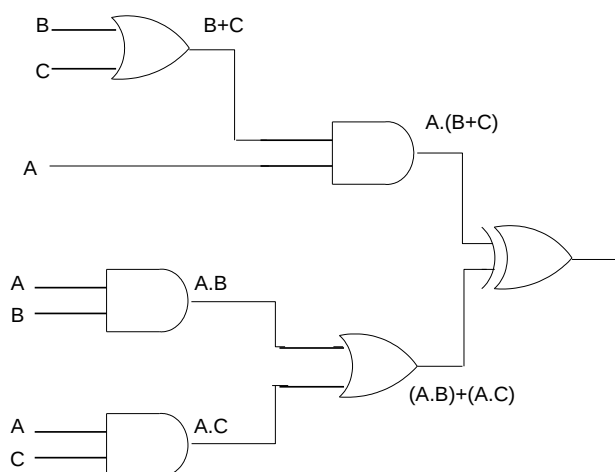


$$A+(B \cdot C) = (A+B) \cdot (A+C)$$

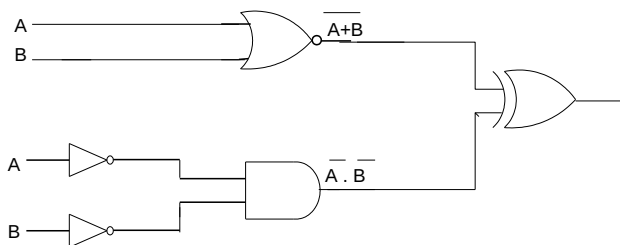
$$A+(B \cdot C) = (A+B) \cdot (A+C)$$

$$A \cdot (B+C) = (A \cdot B) + (A \cdot C)$$

A	B	C	BC	A+(B.C)	A+B	A+C	(A+B).(A+C)
۰	۰	۰	۰	۰	۰	۰	۰
۱	۰	۰	۰	۱	۱	۱	۱
۰	۱	۰	۰	۰	۱	۰	۰
۰	۰	۱	۱	۰	۰	۱	۰
۱	۱	۰	۰	۱	۱	۱	۱
۱	۰	۱	۱	۱	۱	۱	۱
۰	۱	۱	۱	۱	۱	۱	۱
۱	۱	۱	۱	۱	۱	۱	۱

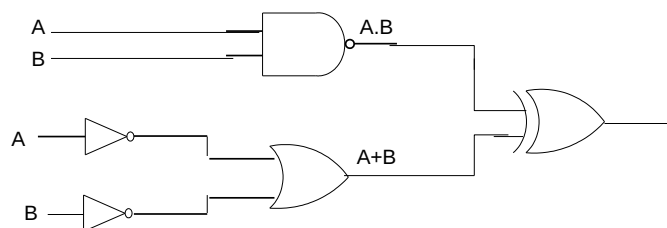


A	B	C	B+C	A.(B+C)	A.B	A.C	(A.B)+(A.C)
۰	۰	۰	۰	۰	۰	۰	۰
۱	۰	۰	۰	۰	۰	۰	۰
۰	۱	۰	۱	۰	۰	۰	۰
۰	۰	۱	۱	۰	۰	۰	۰
۱	۱	۰	۱	۱	۱	۰	۱
۱	۰	۱	۱	۱	۰	۱	۱
۰	۱	۱	۱	۱	۰	۰	۰
۱	۱	۱	۱	۱	۱	۱	۱



$$(A+B) = A.B$$

A	B	A.B	A+B
۰	۰	۰	۰
۱	۰	۰	۱
۰	۱	۰	۱
۱	۱	۱	۱

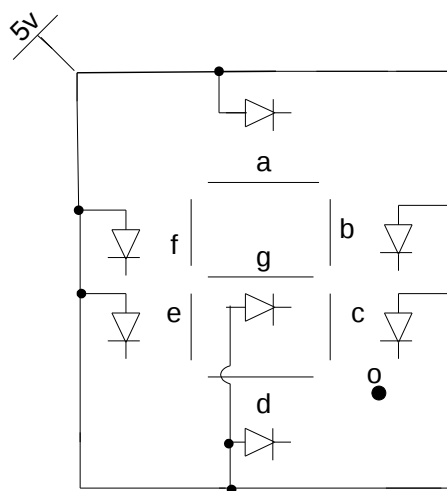


$$(A.B) = A+B$$

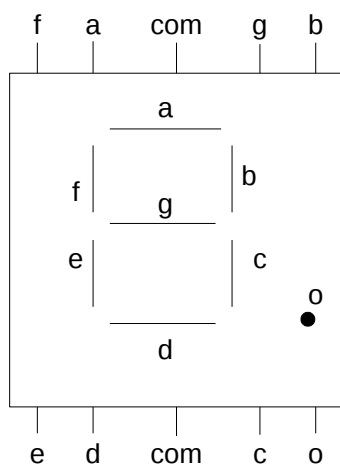
نمایشگر هفت قسمتی

نمایشگر هفت قسمتی:

این نمایشگر دارای هفت LED است. همان طور که در شکل روبرو دیده می شود می توان به صورت آند مشترک و یا کاتد مشترک در آورد. در شکل روبرو به صورت آند مشترک است که همه آند ها را به هم وصل کرده و مثلاً برای روشن کردن شماره یک باید سر های کاتد b و c را به زمین وصل کرد تا LED های مربوطه روشن شوند. برای اینکه یک سون سگمنت بتواند اعداد را نشان دهد ما باید ابتدا کدهای باینری را به BCD تبدیل کرد تا نشان دهد. این کدهای BCD را به دیکودر می دهیم و آن این کدها را گرفته و به سون سگمنت می دهد.



دیکودر ها هم آند مشترک و کاتد مشترک دارند.



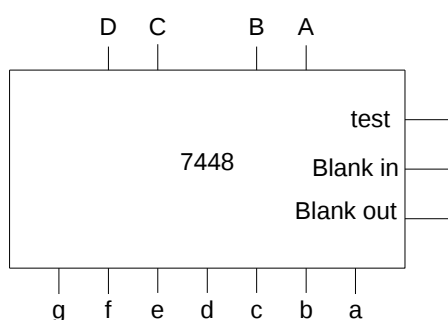
7446 ←
7447 ← : آند مشترک

7448 ←
7449 ← : کاتد مشترک

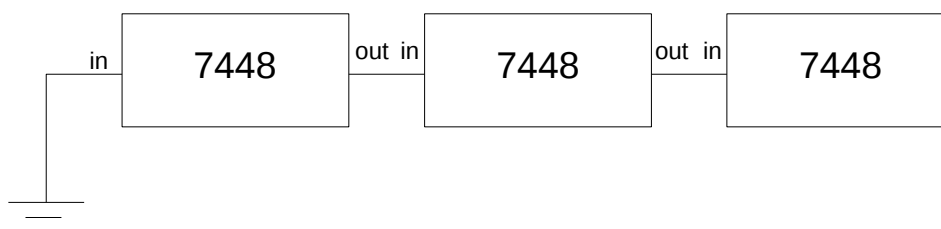
سون سگمنت در صورت واقعی مانند شکل روبرو می باشد.

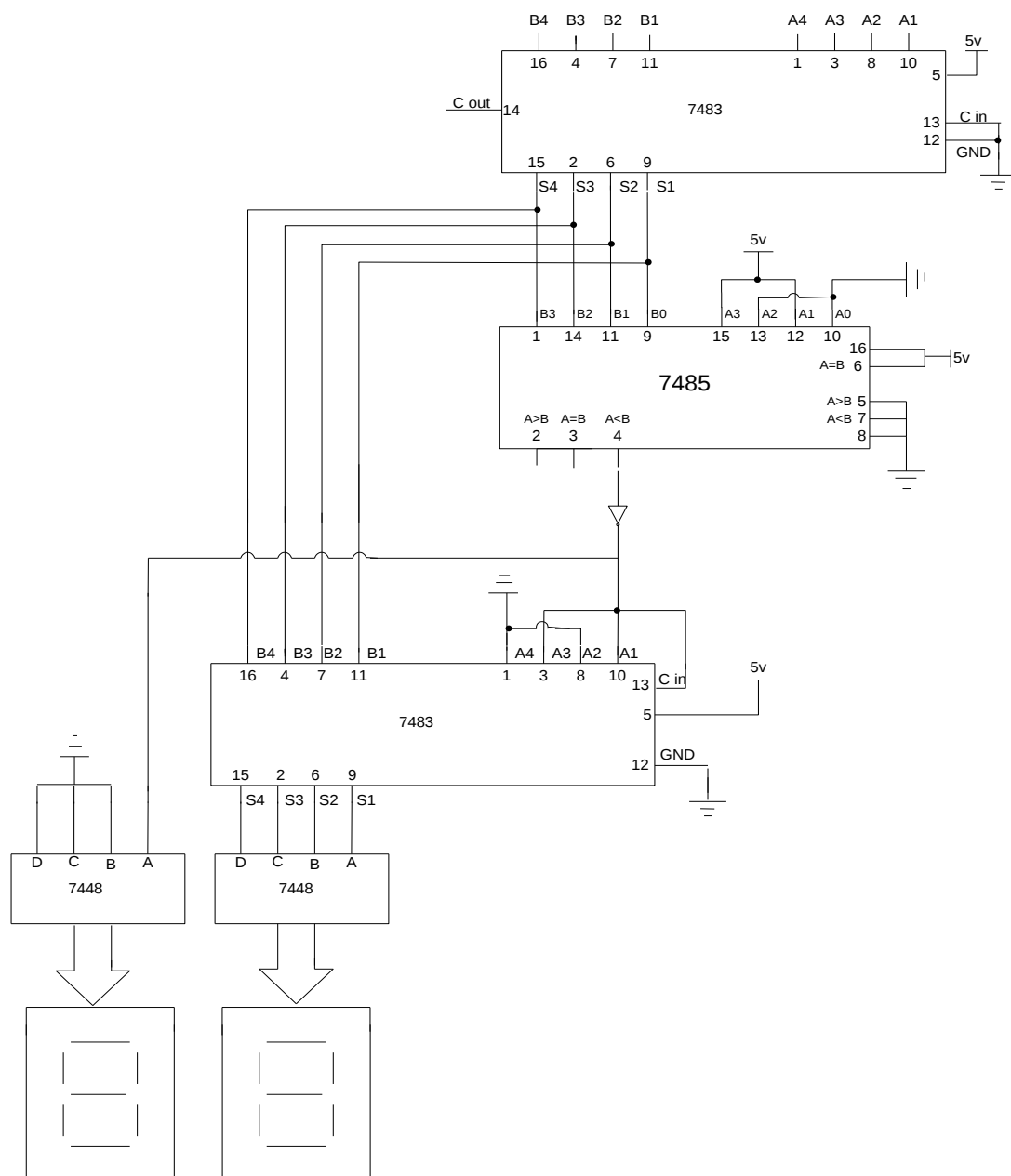
پایه test: این پایه برای تست کردن تمام لامپ های LED از اینکه سالم هستند صورت می گیرد به این صورت که این پایه را low (منطقی) کرده .

پایه Blank in: این پایه به این خاطر وجود دارد که اگر عدد وارد شده به دیکودر سمت چپش صفر باشد آن را خاموش کند یعنی سون سگمنت صفر را نشان ندهد.



پایه Blank out: در صورتی که عدد سمت چپ صفر شود Blank in به Blank out دیکودر بعدی دستور داده که اگر عدد سمت چپ تو صفر شد آن را خاموش کند. ولی اگر عدد سمت چپ دیکودر اولی صفر نشد Blank out آن خاموش بود و دیگر به دیکودر بعدی دستور نمی دهد که اگر تو هم صفر شدی خاموش باش . مثلاً عدد ۱۰۶





توضیح: ما می خواهیم دو عدد باینری را با هم جمع کنیم و جواب آن را در سون سگمنت نشان دهیم.

توجه: سون سگمنت تا عدد نیاز به تبدیل عدد باینری به کد BCD نمی باشد. از ۹ به بعد مثلاً ۱۰ باید به کد BCD تبدیل کرد.

ابتدا دو عدد را با هم جمع کرده خروجی آن را به یک مقایسه کننده می دهیم اگر جواب این جمع کردن تا ۹ بود که یکسره به

دیکودر داده و آن را در سون سگمنت نشان می دهیم. ولی اگر جواب برابر ۱۰ و یا از ۱۰ بیشتر شد باید به کد BCD تبدیل کرد به خاطر همین کار خروجی را در یک مقایسه کننده برده تا آن عدد را با ۱۰ مقایسه کند. و بعد به یک تفریق کننده

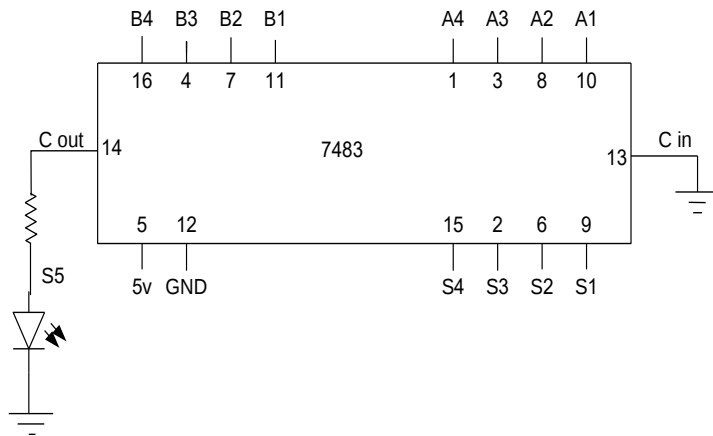
برده تا از ۱۰ کم کند و بعد یک ۱ را به سون سگمنت کنارش بدهد تا به طور مثال عدد ما ۱۳ شده این عدد را از ۱۰ کم کرده

و باقیمانده ۳ میشود که در سون سگمنت راستی نشان داده و عدد ۱ را به سون سگمنت سمت چپی داده تا عدد ۱۳ را نشان

دهد.

جمع کننده و تفریق کننده با استفاده از تراشه ۷۴۸۳

جمع کننده: اگر بخواهیم از این تراشه به عنوان جمع کننده دو عدد ۴ بیتی $A+B=S$ استفاده کرد باید طرز اتصال را به گونه شکل زیر انجام داد.

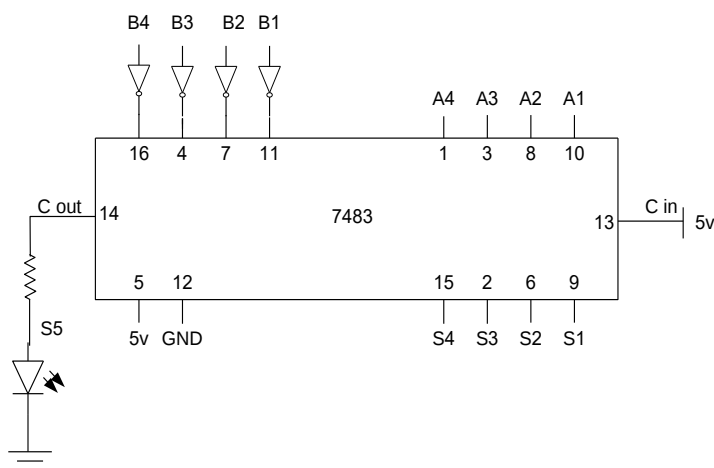


برای تفریق کردن دو عدد ۴ بیتی باید در سر راه پایه های B گیت نات گذاشت و پایه 13 را هم به 5V چون برای منفی کردن باید برعکس یک عدد را نوشت و با ۱ جمع کرد.

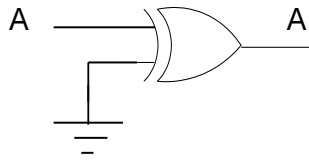
$$\begin{array}{r} 101 \\ + \\ 1 \\ \hline 011 \end{array}$$

طرز اتصال تراشه برای تفریق کردن دو عدد به صورت شکل زیر است.

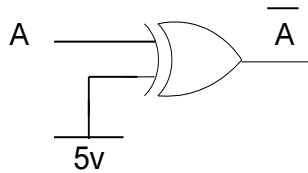
در این شکل برای تفریق کردن باید عدد ۴ بیتی A را با منفی عدد B جمع کرد. برای این کار عدد های B1 تا B4 بوسیله ی گیت نات در سر راهشان آنها را برعکس کرده و بعد چون به پایه 13، 5V را اعمال کرده به معنای عدد 1 است که با عدد 4 بیتی B جمع شده و یک عدد منفی به دست می آید.



چون از این تراشه می خواهیم هم به عنوان جمع کننده و هم به عنوان تفریق کننده استفاده کنیم.



می توان از این خاصیت XOR که در شکل روبرو آمده استفاده کرد.



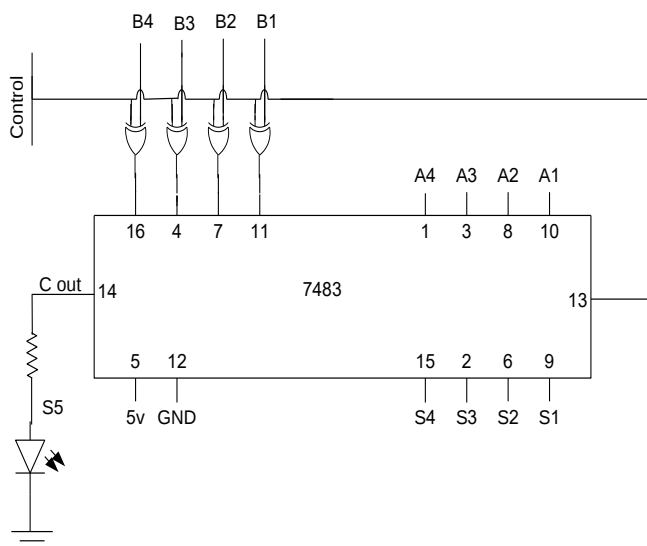
به این ترتیب این تراشه را به صورت شکل زیر اتصال داده. در سر اعداد ۴ بیتی B گیت XOR گذاشته و یک سر ورودی

از XOR ها را اعداد ۴ بیتی B داده و پایه ورودی دیگر را به هم وصل کرده و پایه شماره ۱۳ را هم به آن وصل کرده و پایه

CONTROL می نامیم و اگر حال ما به پایه CONTROL، 5v را اعمال کنیم از این تراشه می توانیم به عنوان تفریق

استفاده کنیم و اگر ما پایه CONTROL را به زمین وصل کنیم می توانیم از این تراشه به عنوان جمع کننده استفاده کنیم.

به طور مثال چند عدد را با هم جمع و تفریق می کنیم.



A	۰۱۱۰	۱۱۰۱	۱۰۰۱
B	۰۱۰۱	۱۰۱۰	۱۰۰۱
Control=0	۰۱۰۱۱	۱۰۱۱۱	۱۰۰۱۰
Control=1	۰۰۰۱	۰۰۱۱	۰۰۰۰

تفریق کننده

در یک سیستم n بیتی، منفی هر عدد مانند a برابر است با: $2^{n+1} - a$

$$2 \longrightarrow 10$$

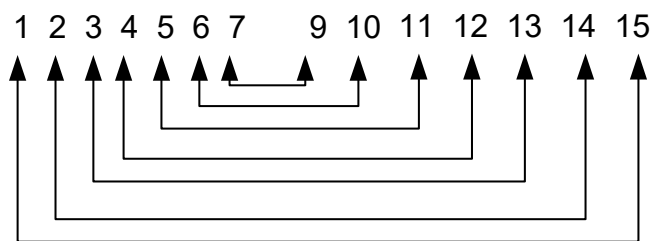


$$2^{2+1} - 2 = 6 \rightarrow 110$$

دوبیتی بدون منفی یا مثبت

سه بیتی با منفی

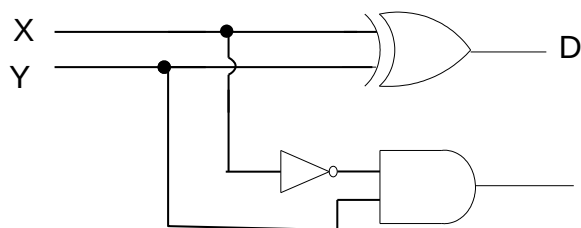
بنابراین سیستم کامل دارای $n+1$ بیت می شود اگر بیت چپ ترین (MSB) برابر ۱ باشد عدد منفی است.



برای منفی کردن اعداد، عدد را بر عکس کرده و با ۱ جمع می کنیم.

$$\begin{array}{r} 0101 \\ 1011 \\ \hline 0000 \end{array} \longrightarrow \begin{array}{r} 1010 \\ + \\ 1 \\ \hline 1011 \end{array}$$

نیمه تفریق کننده:

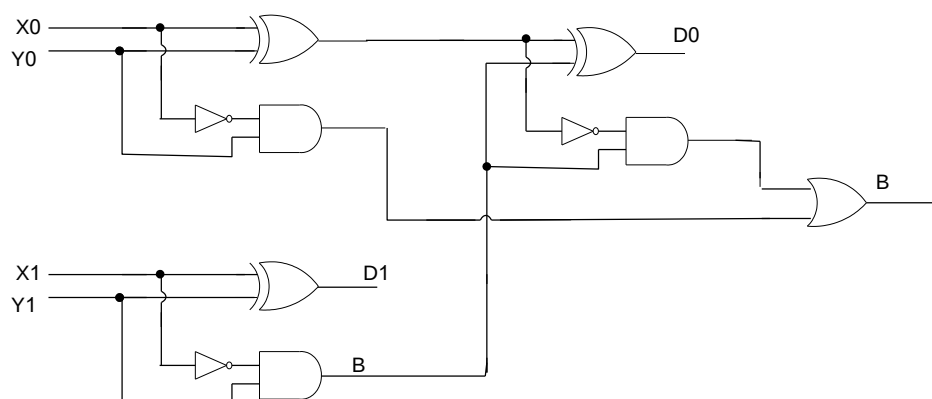


$$D = X + Y$$

$$B = X \cdot Y$$

X	Y		B
۰	۰	۰	۰
۰	۱	۱	۰
۱	۰	۱	۰
۱	۱	۰	۱

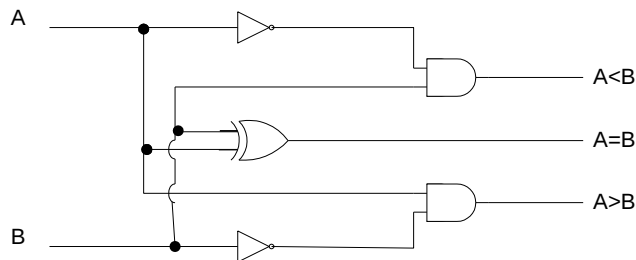
تمام تفریق کننده :



X1	X0	Y1	Y0	B	D1	D0
0	0	0	0	0	0	0
0	0	0	1	1	1	1
0	0	1	0	1	1	0
0	0	1	1	0	0	1
0	1	0	0	0	1	0
0	1	0	1	1	1	0
0	1	1	0	0	0	1
0	1	1	1	0	1	1
1	0	0	0	0	0	1
1	0	0	1	1	1	0
1	0	1	0	0	1	1
1	0	1	1	1	0	0
1	1	0	0	0	0	0
1	1	0	1	1	1	1
1	1	1	0	0	1	0
1	1	1	1	0	0	1

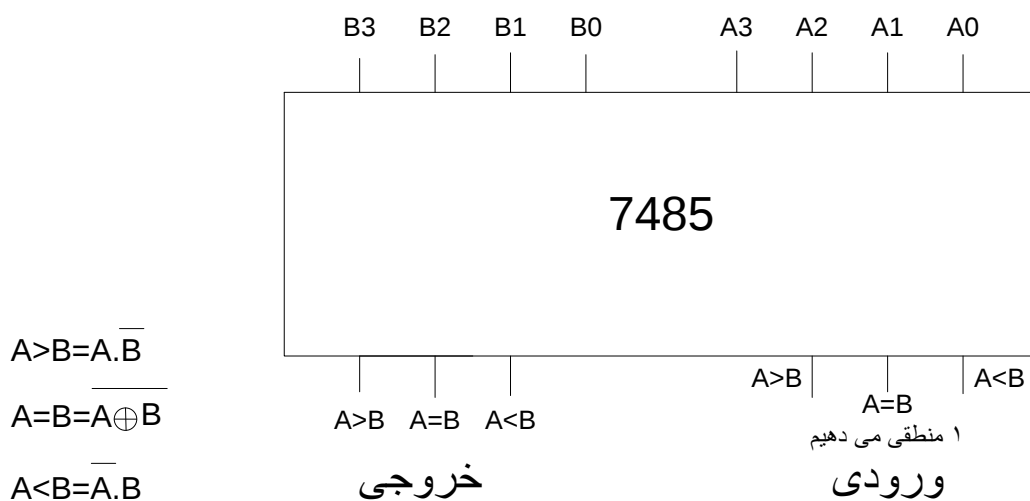
نیمه مقایسه کننده

برای مقایسه کردن دو عدد تک بیتی از این گیت استفاده می شود. که بزرگتر و یا کوچکتری یک عدد را نسبت به دیگری نشان می دهد، و همچنین تساوی دو عدد را نیز نشان می دهد، جدول صحت آن نیز مانند شکل زیر می باشد، این مقایسه گر به این صورت کار می کند که از طرف چپ شروع کرده و ستون به ستون مقایسه میکند اگر برابر بودند که تساوی دو عدد را نشان میدهد. ولی اگر به اولین ۱ که رسید کار می کند.



A	B	A>B	A=B	A<B
۰	۰	۰	۱	۰
۰	۱	۰	۰	۱
۱	۰	۱	۰	۰
۱	۱	۰	۱	۰

مقایسه کننده به صورت یک تراشه نیز وجود دارد به نام ۷۴۸۵ که شکل آن و نیز نوع سیم بندی آن به صورت زیر می باشد.



این تراشه برای مقایسه دو عدد ۴ بیتی مورد استفاده قرار می گیرد. اگر بخواهیم دو عدد ۴ بیتی را با هم مقایسه کنیم به سرهای A و B اعداد ما هستند.

در اینجا ما باید به سر $A=B$ از ورودی ۱ منطقی بدهیم و دو سر دیگر را کاری نداریم و سرهای خروجی را به LED می دهیم.

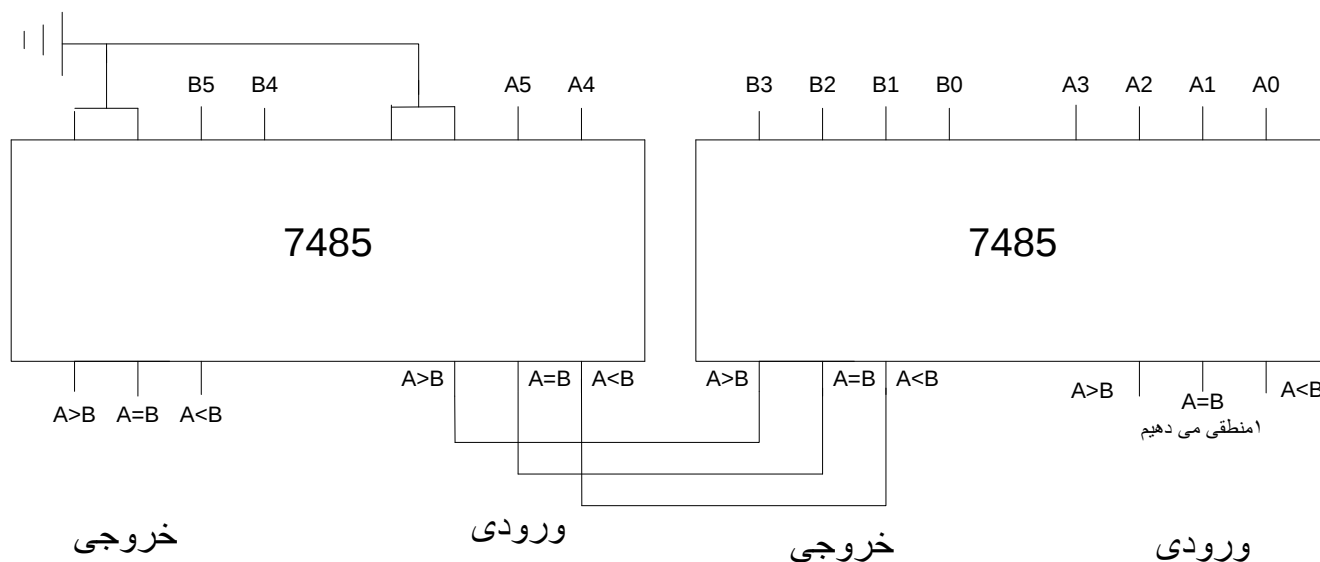
با این کار مقایسه را به طور صحیح انجام می دهد. حال اگر بخواهیم به طور مثال دو عدد ۶ بیتی را با هم مقایسه کنیم باید از

یک تراشه دیگر نیز استفاده کرد، که ۴ عدد در این تراشه و دو عدد دیگر نیز به تراشه دیگر وصل شود. سرهای خروجی تراشه

اول را به سرهای ورودی تراشه دوم وصل کرده و سرهای خروجی تراشه دوم را نیز به LED داده، ورودی تراشه اول را نیز فقط به سر $A=B$ آن ۱ منطقی داده و دو سر دیگر را کاری نداریم.

نکته: چون عدد ما ۶ بیتی است در نتیجه از سرهای A و B تراشه دوم ۲ تا باقی می ماند که آن دو سر را به زمین وصل کرده

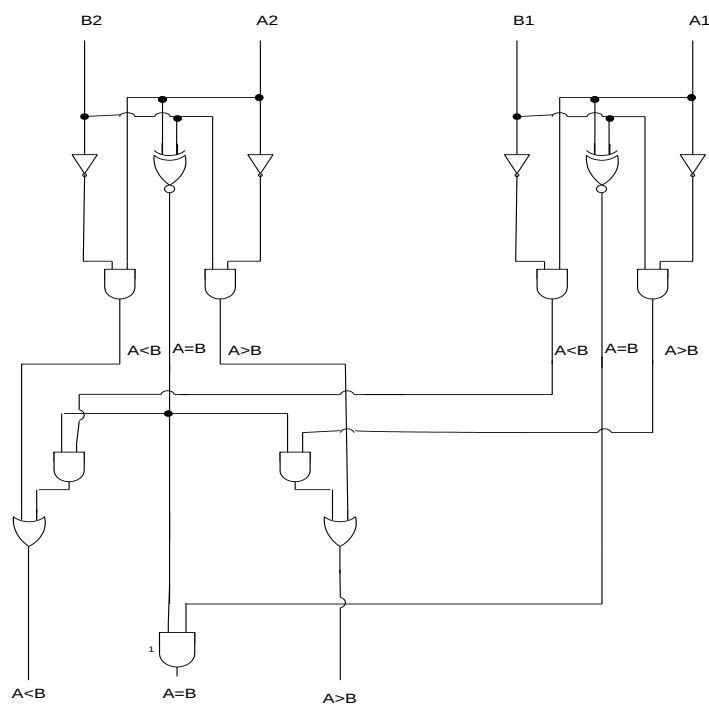
در این شکل نشان داده شده است.



تمام مقایسه کننده : این تمام مقایسه کننده در شکل زیر فقط دو عدد ۲ بیتی را با هم مقایسه می کند که از یک نیم مقایسه کننده

و از یک تمام مقایسه کننده درست شده است، برای مقایسه کردن اعداد بیشتر از چند بیتی یک تمام مقایسه کننده دیگر می

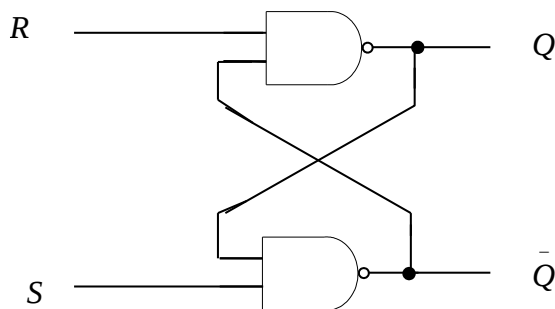
گذاریم. ترتیب کلید ها هم به صورت زیر می باشد. $A_2 A_1 B_2 B_1$.



A2	A1	B2	B1	A<B	A=B	A>B
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	1	0	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	0	1
1	0	1	1	0	0	1
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

فلیپ فلاپ ها

الف - فلیپ فلاپ نوع RS:



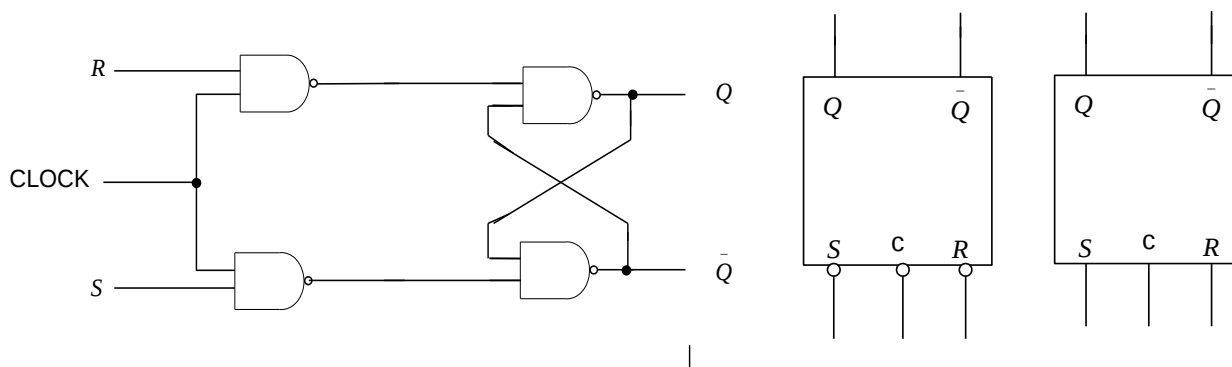
R	S	Q	$\bar{Q}$
۰	۰	۰	۰
۰	۱	۰	۱
۱	۰	۱	۰
۱	۱	Q_0	$\bar{Q}_0$

زمانی که هر دو ورودی های R و S را صفر بدهیم خروجی ها هر دو صفر می باشند که این حالت را غیر مجاز می گویند.

زمانی که هر دو ورودی را ۱ بدهیم ، خروجی ها حالت قبل خود را حفظ می کنند که با اندیس ۰ ، مثل Q_0

(این نوع فلیپ فلاپ را فعال LOW می گویند یعنی مقدار ۰ برایش با ارزش است)

ب - فلیپ فلاپ کامل :



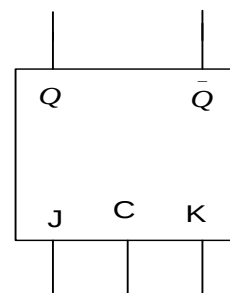
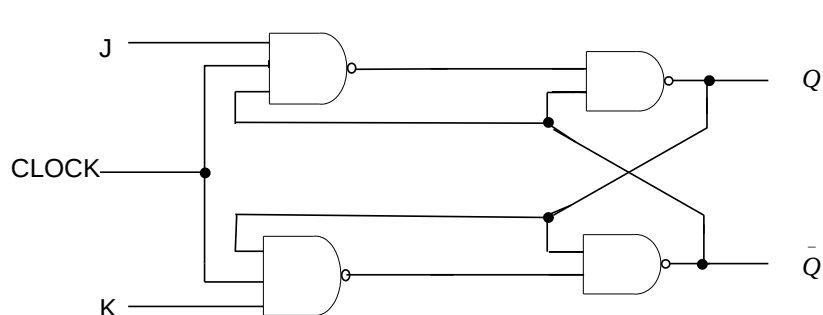
اگر Clock را صفر بدهیم باعث شده که خروجی ها به حالت قبل خود وابسته باشد و دیگر ورودی ها برای ما تاثیری ندارند ولی اگر Clock را ۱ بدهیم خروجی به ورودی ها R و S بستگی دارند.

clk	S	R	Q	$\bar{Q}$
۰	x	x	Q_0	$\bar{Q}_0$
۱	۱	۰	۱	۰
۱	۰	۱	۰	۱
۱	۱	۱	۱	۱

⇒

این حالت غیر مجاز می باشد.

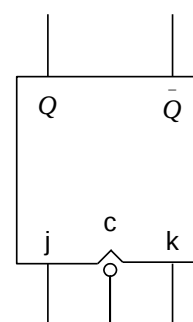
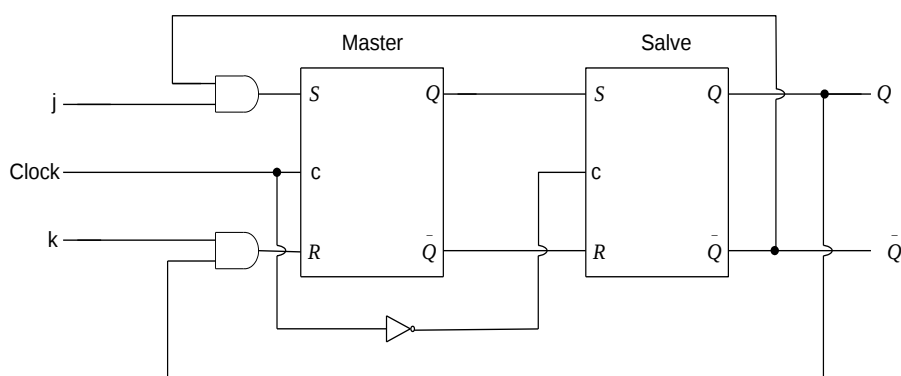
این فلیپ فلاپ هنوز دارای مشکل غیر مجاز می باشد که هر دو خروجی ها همسان هستند در نتیجه برای حل این مشکل یک فلیپ فلاپ نوع JK را درست کردیم.



این فلیپ فلاپ هنوز به سطح بستگی دارد ، ما می خواهیم یک فلیپ فلاپی بسازیم که به لبه بستگی داشته باشد.

Clk	j	k	Q	$\bar{Q}$
۰	x	x	Q_0	$\bar{Q}_0$
۱	۱	۰	۱	۰
۱	۰	۱	۰	۱
۱	۰	۰	Q_0	$\bar{Q}_0$
۱	۱	۱	$\bar{Q}_0$	Q_0

در نتیجه فلیپ فلاپ ساختیم که به لبه پایین رونده حساس است ، این فلیپ فلاپ معروف به (ارباب) master و (برده) slave می باشد.



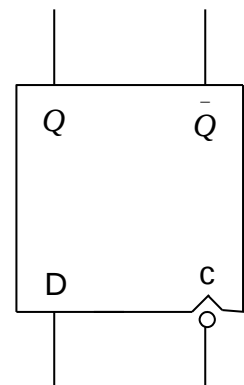
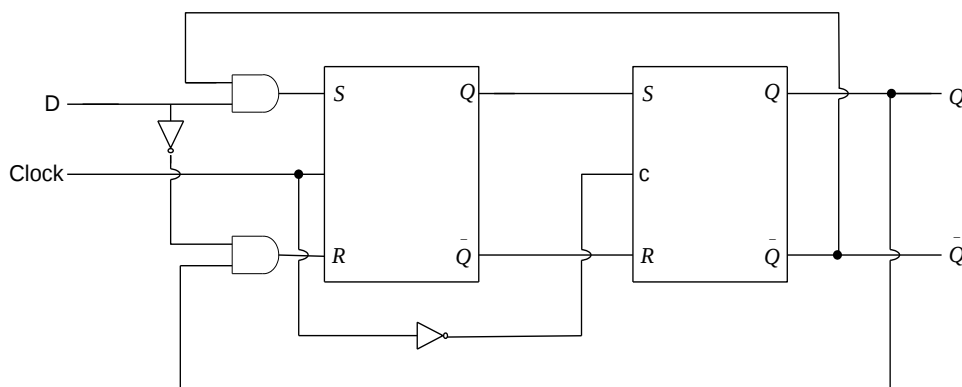


clk	j	k	Q	$\bar{Q}$
۱	×	×	Q_0	$\bar{Q}_0$
↓	۱	۰	۱	۰
↓	۰	۱	۰	۱
↓	۰	۰	Q_0	$\bar{Q}_0$
↓	۱	۱	$\bar{Q}_0$	Q_0

این فلیپ فلاپ به لبه پایین رونده حساس است.

اگر بخواهیم این فلیپ فلاپ به لبه بالا رونده حساس باشد ، باید در سر راه پایه Clock یک گیت Not گذاشت.

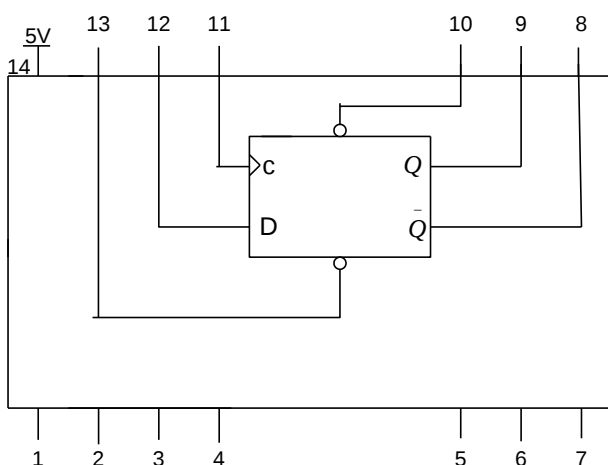
ج - فلیپ فلاپ نوع D :





تعریف تراشه ی سی ۷۴۷۴:

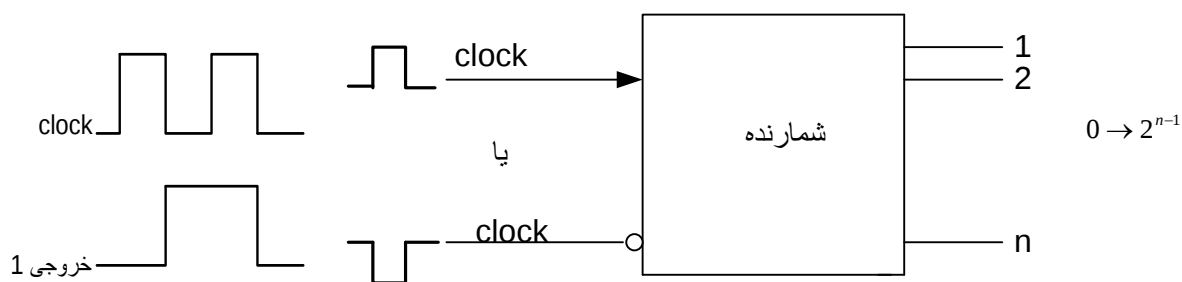
در این تراشه دو فلیپ فلاپ نوع D وجود دارد که یکی را نکشیدیم. و همچنین در این فلیپ فلاپ دو پایه Set و Reset را گذاشتند که فعال Low هستند یعنی در حالت عادی باید ۱ باشند. و با صفر دادن به آنها اثر می کنند. که مثلاً اگر پایه Set را بدهیم $Q=1$ و $\bar{Q}=0$ خواهد شد ولی اگر به Reset را، بدهیم $Q=0$ و $\bar{Q}=1$ خواهد شد، در این آی سی پایه های Set و Reset بر پایه های D و Clock اولویت دارند، یعنی اگر در وسط مدار یکی از پایه های set و reset را صفر بدهیم دیگر تغییرات در پایه های D و Clock هیچ تاثیری بر مدار نمی گذارند.



clk	D	Set	Reset	Q	$\bar{Q}$
۱	x	۱	۱	Q_0	$\bar{Q}_0$
↓	۱	۱	۱	$Q = D$	$\bar{Q} = \bar{D}$
x	x	۰	۱	۱	۰
x	x	۱	۰	۰	۱

تراشه ی ۷۴۷۴

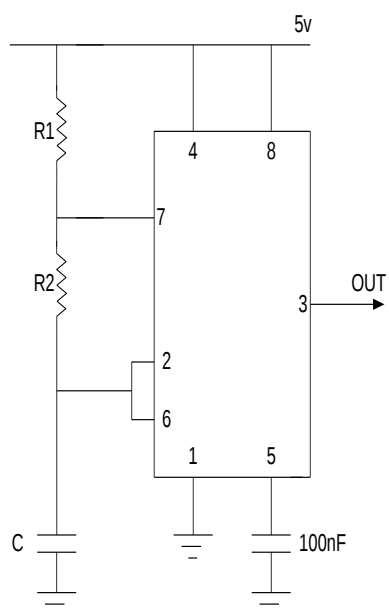
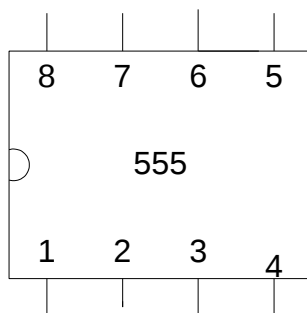
پالس ساعت و ساخت شمارنده ی آسنکرون



کار این شمارنده : ۱- شمارش ۲- تقسیم فرکانس

اگر از پایه ۱ و clock بگیریم فرکانس را بر ۲ تقسیم می کند. و اگر از پایه ۲ و clock بگیریم فرکانس را بر ۴ تقسیم می کند.

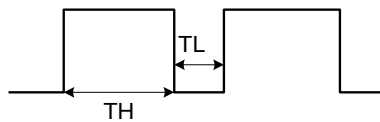
مدار تولید کننده پالس:



اتصال آی سی ۵۵۵ به صورت شکل روبرو می باشد.

شارژ و دشارژ خازن C زمان روشن و یا خاموش بودن لامپ را تعیین میکند.

نحوه محاسبه زمان TH و TL



$$T_H = 0.7 \times (R_1 + R_2) C$$

$$T_L = 0.7 \times R_2 \times C$$

$$F = \frac{1}{T_H + T_L}$$

تولید شمارنده :

0 0 0 0
0 0 0 1
0 0 1 0
0 0 1 1
0 1 0 0
0 1 0 1
0 1 1 0
0 1 1 1

قانون شمارش : هر بیت به بیت سمت راست خود نگاه کرده و وقتی آن بیت

از ۰ به ۱ تغییر حالت داد ، خود آن بیت نیز تغییر حالت می کند.

این شمارنده آسنکرون نام دارد یعنی غیر همزمان که به طور مثال ابتدا بیت اول از ۰ به ۱

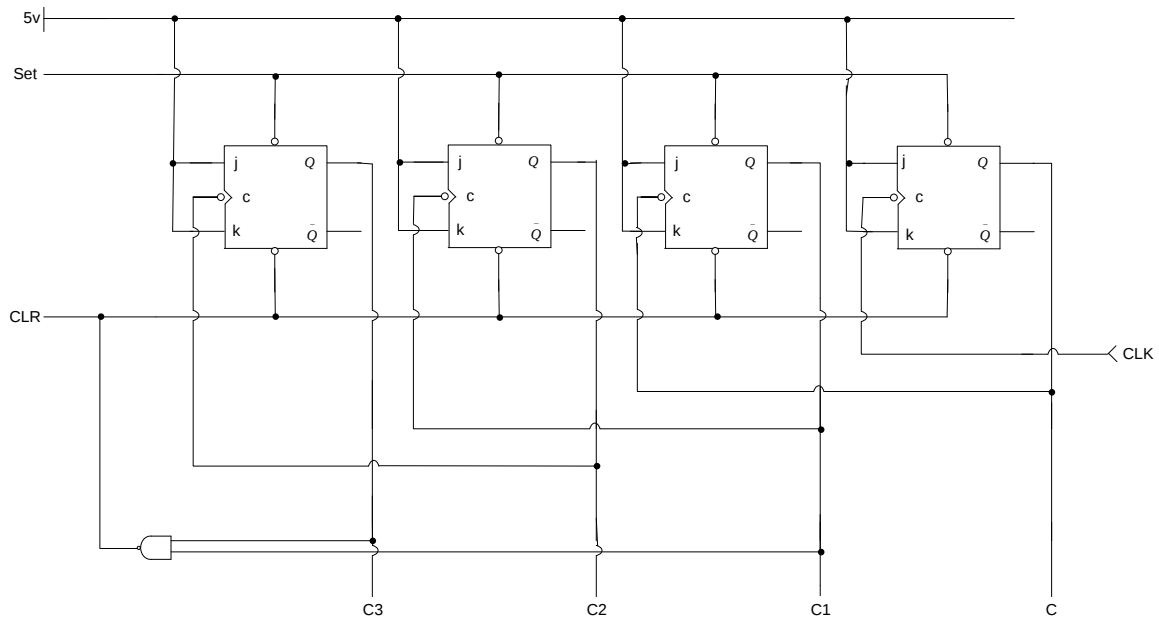
رفته و بعد بیت دوم تغییر حالت می دهد و اگر تغییرات بیت دوم از ۰ به ۱ باشد بیت سوم

نیز تغییر حالت می دهد. و بعد هم بیت چهارم. که این کار زمانی را صرف می کند که خیلی

1 1 1 0
1 1 1 1

کوچک است و بسیار مهم است. که برای این کار می آیند و شمارنده سنکرون می سازند یعنی

همزمان با هم تغییر حالت می دهند.

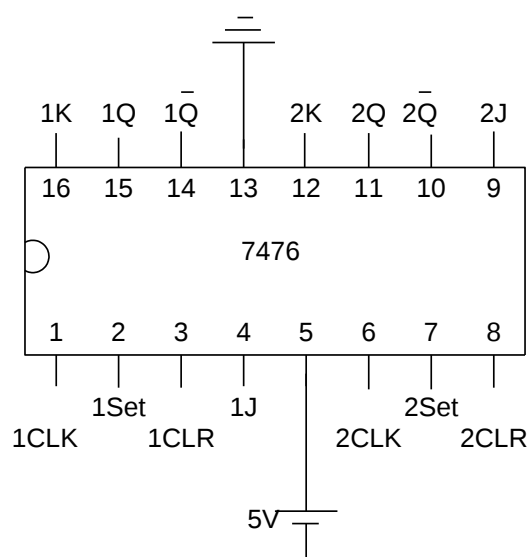


CLR همیشه در هر موقعیت و هر زمانی که مقدار آن را از ۱ به ۰ (چون ۰ نات دارد در نتیجه مقدار ۰ برایش ارزش دارد یعنی در حالت کار عادی باید ۱ باشد.) تغییر داد کل خروجی ها را صفر می کند و باعث می شود که دوباره از صفر تا ۱۵ بشمارد.

Set همیشه اگر مقدار آن را از ۱ به ۰ تغییر داد باعث شده که تمام خروجی ها را ۱ کرده (یعنی ۱۵ دسیمال) و یعنی وقتی این کلید را از ۱ به ۰ تغییر بدیم شمارنده از ۱۵ شروع به شمارش کرده و ۰ شده و بعد هم ۱ تا دوباره به ۱۵ برسد.

اگر بخواهیم که شمارنده تا مثلاً ۹ بشمارد باید کاری کنیم که موقعی که خواست عدد ۱۰ را نشان دهد همان لحظه مقدار را صفر کنیم. عدد ۱۰ به صورت باینری در روبرو نشان داده شده است ۱۰۱۰ حال برای اینکه شمارنده تا ۹ بیشتر نشمارد باید یک گیت nanad را در مدار قرار داد و ورودی های آن را به بیت دوم و چهارم وصل کرد و خروجی آن را به CLR وصل کرد. که هرگاه بیت دوم و چهارم ۱ شده شمارنده ۰ شود.

CLR چون با لبه پایین رونده کار می کند یک گیت NANAD می گذاریم.

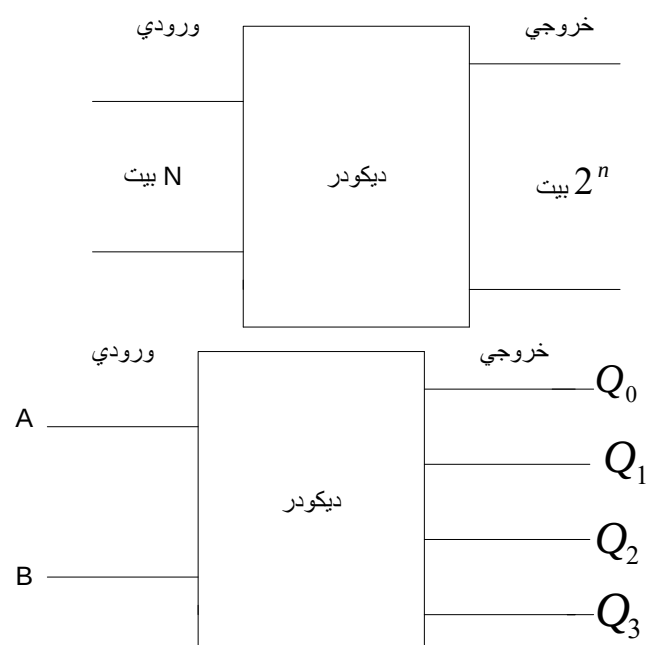


این آی سی دارای دو JK می باشد.

ساخت دیکودر- انکودر- مالتی پلکسر - دی مالتی پلکسر

دیکودر: تولید دیکودر به وسیله ی گیت AND و NOT

A	B	Q0	Q1	Q2	Q3
۰	۰	۱	۰	۰	۰
۰	۱	۰	۱	۰	۰
۱	۰	۰	۰	۱	۰
۱	۱	۰	۰	۰	۱



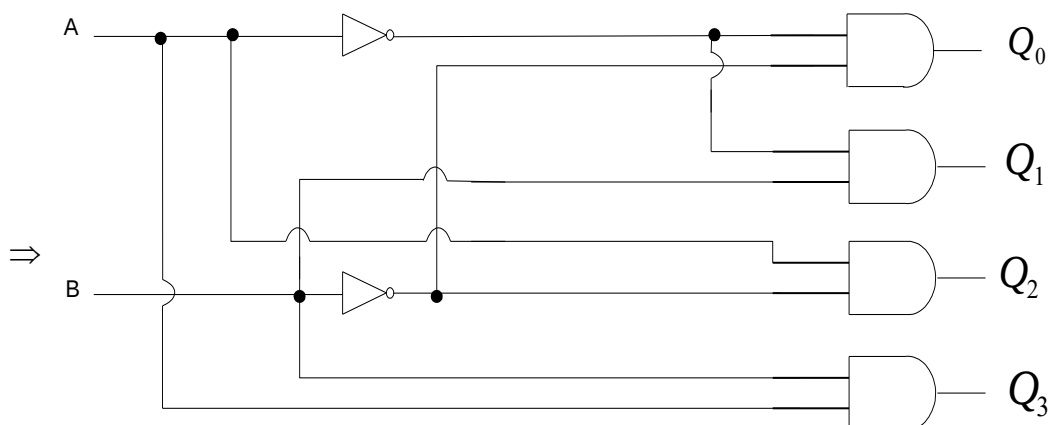
طبق جدول صحت روبرو یک دیکودر را می توان به صورت زیر ساخت ابتدا باید برای هر دام از خروجی ها طیف جدول صحت بک رابطه نوشت و به وسیله این روابط دیکودر را ساخت.

$$Q_0 = \bar{A} \cdot \bar{B}$$

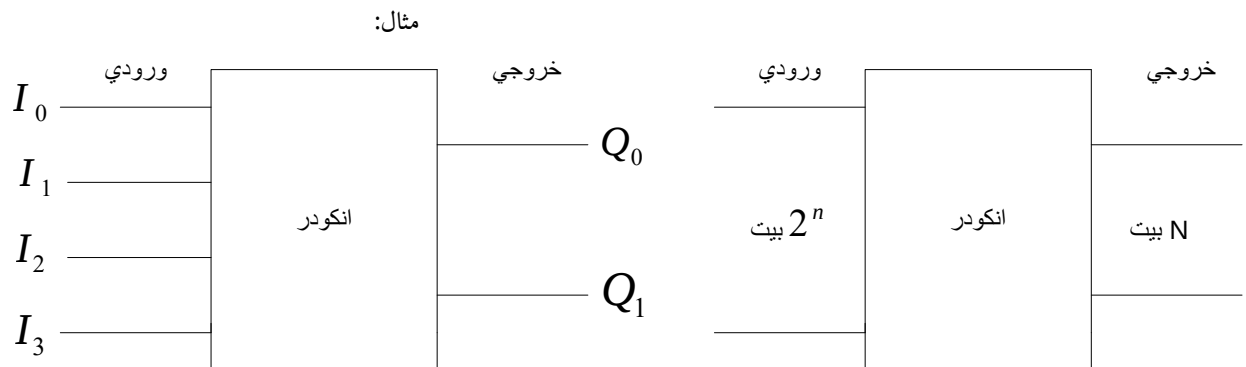
$$Q_1 = \bar{A} \cdot B$$

$$Q_2 = A \cdot \bar{B}$$

$$Q_3 = A \cdot B$$



انکودر : ساخت اکودر با گیت OR

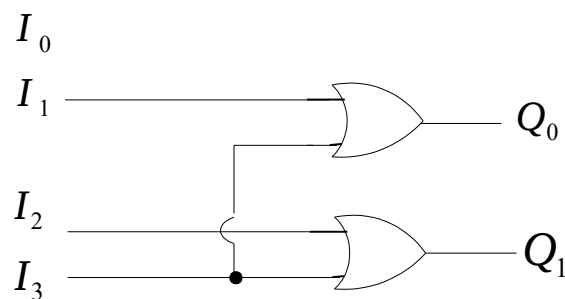


در این جدول صحت ما معجز نیستیم که در یک لحظه ورودی ها صفر باشند. در نتیجه با گیت OR ساخته می شود چون در هر

لحظه یکی از ورودی ها فعال هست.

$$\Rightarrow \begin{aligned} Q_1 &= I_2 + I_3 \\ Q_2 &= I_1 + I_3 \end{aligned}$$

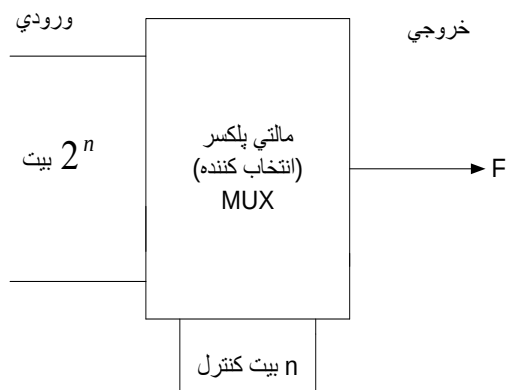
بر این مبنا کار می کند که مثلاً ما اگر I_2 را فعال کنیم در خروجی به ما باینری ۲ می دهد و Q_1 روشن می شود و اگر I_3 را فعال کنیم Q_0 و Q_1 روشن می شوند و عدد ۳ باینری را نمایش می دهند.



I_0	I_1	I_2	I_3	Q_0	Q_1
۱	۰	۰	۰	۰	۰
۰	۱	۰	۰	۰	۱
۰	۰	۱	۰	۱	۰
۰	۰	۰	۱	۱	۱

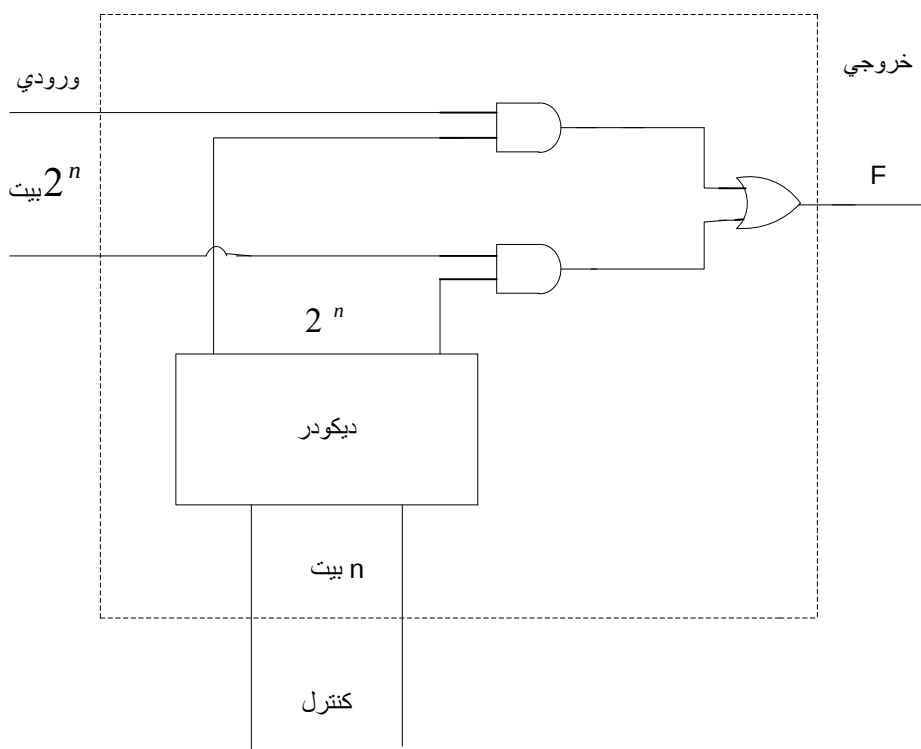
مالتی پلکسر:

کنترل		ورودی				خروجی
A	B	I3	I2	I1	I0	F
۰	۰					I0
۰	۱					I1
۱	۰					I2
۱	۱					I3

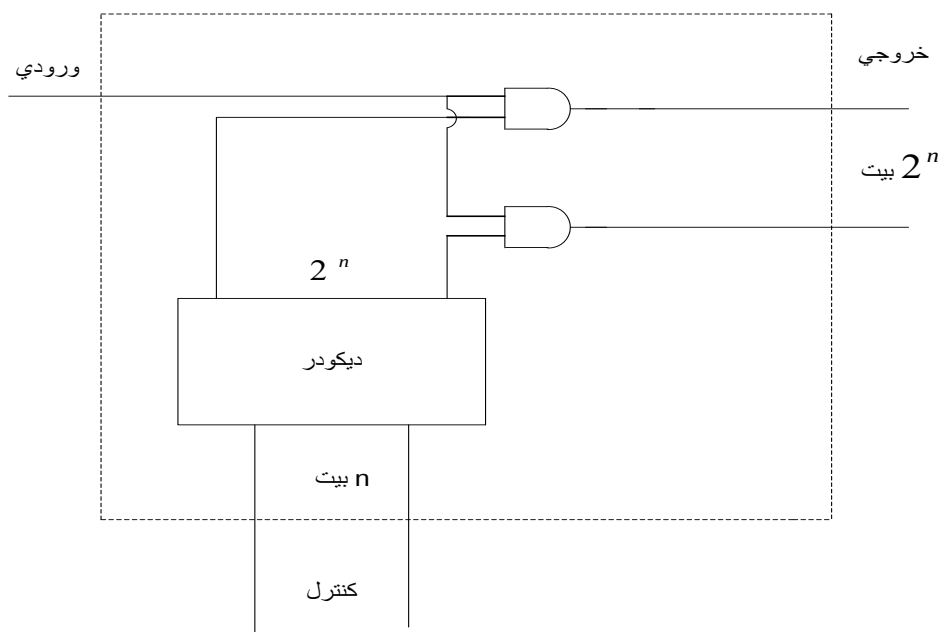
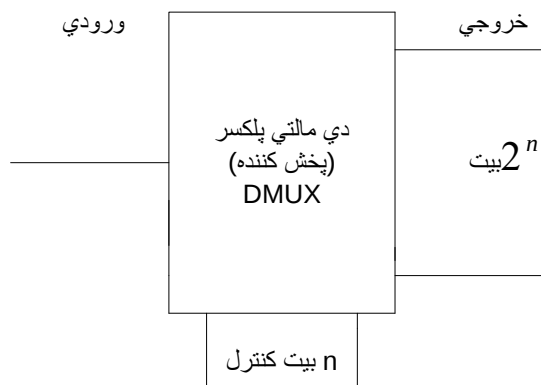


طرز کار این گیت به این صورت می باشد که با تغییرات در پایه کنترل میتوان خروجی را به یکی از ورودی ها وصل کرد، خروجی تابع آن ورودی باشد.

به طور مثال همانطور که در جدول صحت بالا دیده می شود در صورتی که پایه A و b هر دو صفر باشند خروجی f به I0 وصل میشود.

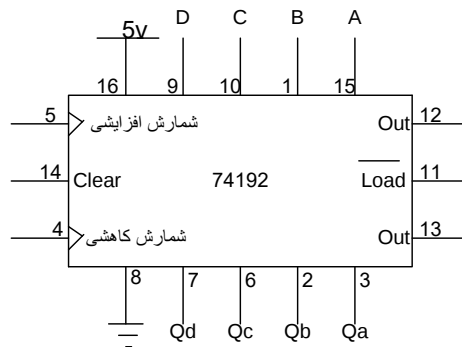


دی مالتی پلکسر: طرز کار به این صورت است که با تغییرات در کنترل می توان ورودی را به یکی از پایه های خروجی وصل کرد و آن خروجی تابع ورودی باشد.



کنترل		ورودی	خروجی			
A	B	I	Q3	Q2	Q1	Q0
۰	۰		۰	۰	۰	I
۰	۱		۰	۰	I	۰
۱	۰		۰	I	۰	۰
۱	۱		I	۰	۰	۰

شمارنده BCD



همه ی شمارنده های BCD می توانند تا عدد ۹ بشمارند. این شمارنده می تواند از ۱ تا ۹ و یا بر عکس از ۹ تا ۱ بشمارد.

آی سی ۷۴۱۹۲

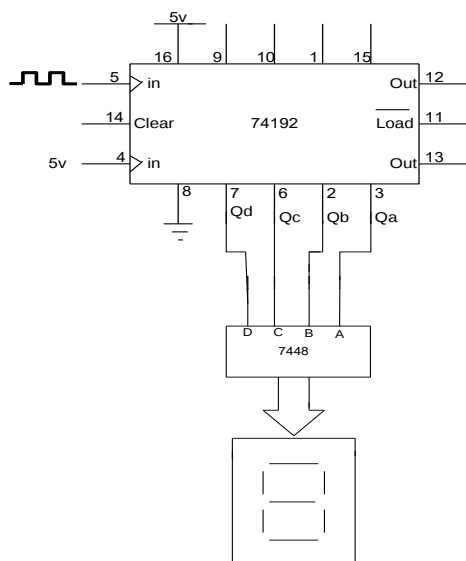
موقعی که ما بخواهیم این شمارنده از ۱ تا ۹ بشمارد به پایه شمارش افزایشی پالس داده و پایه شمارش کاهشی را در وضعیت high قرار داده یعنی ۵ ولت.

برای نشان دادن اعداد بیشتر مثلاً تا ۹۹ باید این شمارنده را با یک شمارنده دیگر به آبشاری بست. به طور مثال اگر بخواهیم شمارنده از ۱ تا ۹۹ را بشمارد، شمارنده بیت اول را به صورت افزایشی در می آوریم یعنی به پایه ۵ پالس و به پایه ۴ هم ۵ ولت داده و پایه های out متناظر آنها را به پایه های ۵ و ۴ شمارنده بعدی وصل می کنیم.

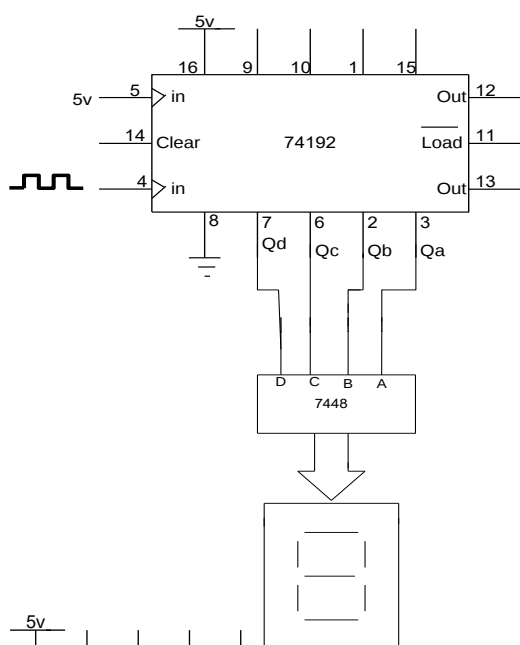
Clear این پایه در حالت عادی باید low باشد یعنی صفر که اگر ما به این پایه ۱ منطقی بدهیم شمارش را صفر می کند و شمارنده از صفر شروع به شمارش می کند.

Load این پایه در حالت عادی باید ۱ منطقی باشد و وقتی آن را صفر کنیم هر چه در ورودی است در خروجی ریخته اگر در ورودی عدد ۱۲ داریم وقتی پایه load را صفر بدهیم در وسط کار مدار در خروجی عدد ۱۲ نشان داده می شود حال اگر در همین موقعیت به پایه های in آن پالس بدهیم از ۱۲ بقیه شمارش را انجام می دهد.

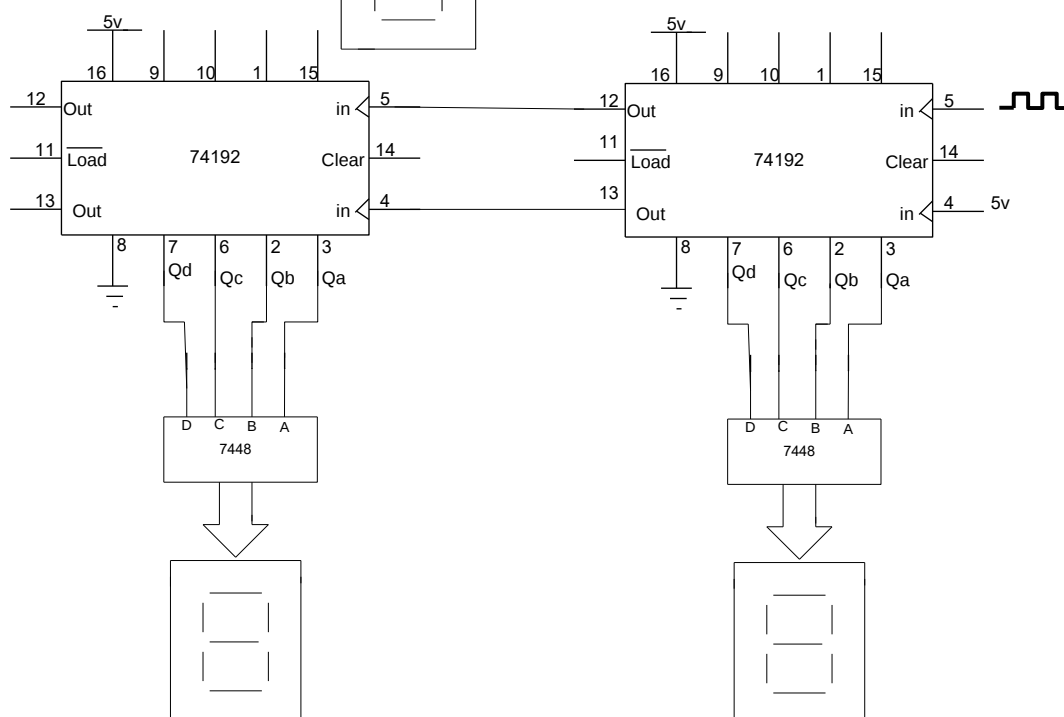
مثال: شمارنده را طوری طراحی کنید که از ۰ تا ۹، ۹۹ تا ۰، ۰ تا ۹۹ را نشان دهد.



۰ تا ۹ ←



۹ تا ۰ ←

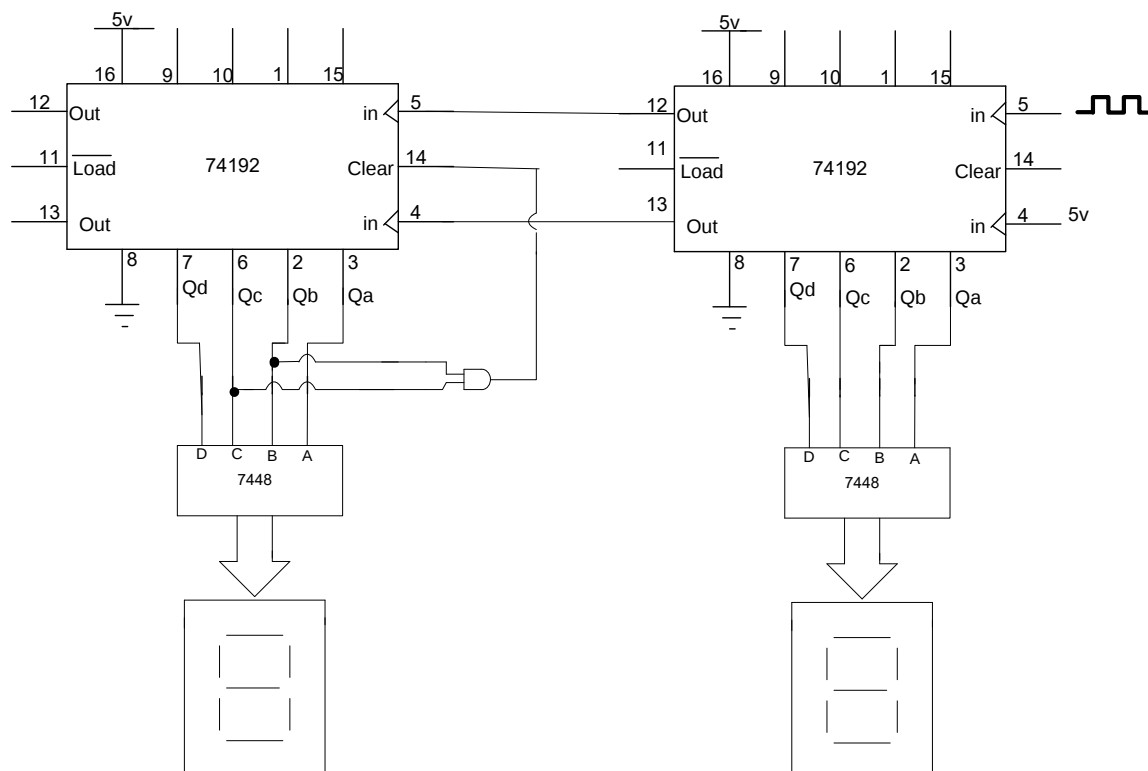


۰ تا ۹۹ ←

برای ۹۹ تا ۰

کافی است به پایه ۵ شمارنده بیت اول ۵ ولت را داده و همچنین به پایه ۴ شمارنده بیت اول پالس داده.

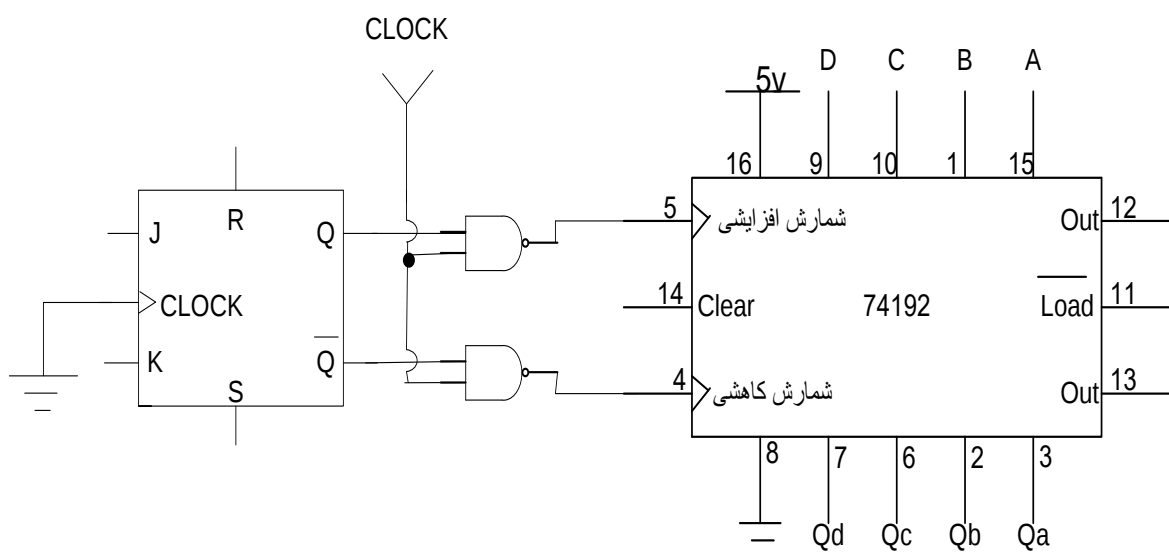
۰ تا ۵۹



توضیح: این شمارنده از ۰ تا ۵۹ می شمارد بعد موقعی که ۶۰ شد یعنی عدد ۰ در شمارنده بیت اولی و عدد ۶ در شمارنده بیت دومی در نتیجه موقعی که عدد ۶ در شمارنده بیت دوم درست شد از پایه های ۲ و ۶ به این خاطر که عدد ۶ به باینری ۰۱۱۰ انشعاب گرفته و بعد یک گیت and برده و وقتی هر دو تا ۱ شد خروجی آن ۱ و به پایه clear شمارنده برده که شمارنده را صفر کند در نتیجه به این خاطر که شمارنده اولی ۰ می باشد و شمارنده دومی را که خود ما صفر کردیم عدد صفر نمایش داده می شود و شمارنده از ۰ دوباره تا ۵۹ می شمارد.

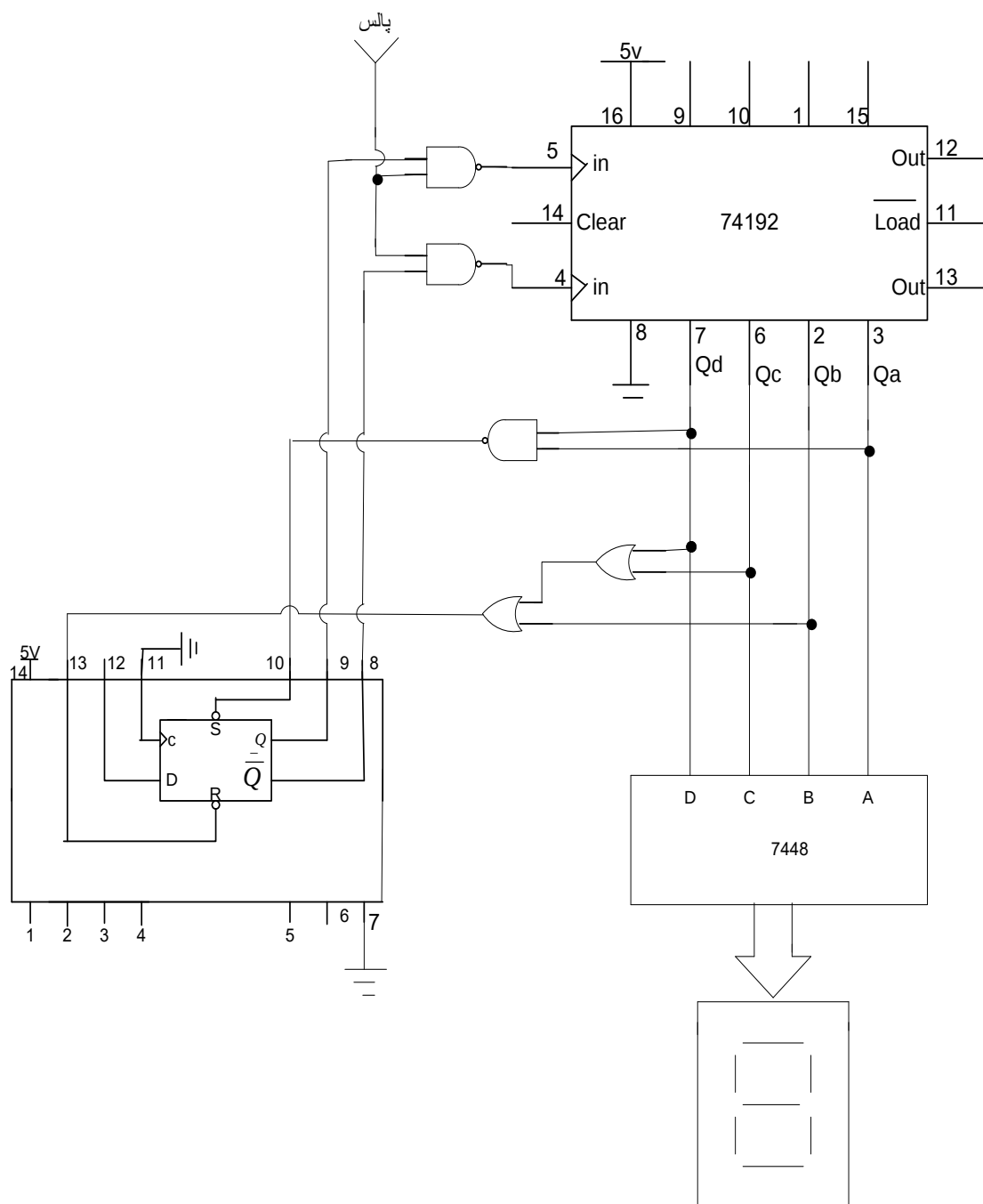
مداری طراحی کنید که از ۰ تا ۹ بشمارد و بعد وقتی به ۹ رسید به صورت معکوس تا ۰ بشمارد.

برای این باید از فلیپ فلاپ (حافظه) استفاده کرد. ما می دانیم که در فلیپ فلاپ نوع JK اگر به پایه clock آن صفر یا ۵ ولت بدهیم از حالت JK به RS تبدیل می شود.



توضیح: به طور مثال اگر در JK به Clock، ۰ یا ۵ ولت بدهیم و به طور مثال اگر به پایه $R \leftarrow 1$ و پایه $S \leftarrow 0$ بدهیم در نتیجه $Q \leftarrow 1$ و $\bar{Q} \leftarrow 0$ خواهد بود و پایه CLOCK هم پالس می دهیم حال اگر فرض شود پالس در قسمت مثبت خود قرار دارد و Q هم ۱ است در نتیجه خروجی NAND ۰ خواهد بود و اگر در لحظه بعدی که پالس در لحظه صفر خود است و Q هم که ۱ می باشد در نتیجه در خروجی گیت NAND ۱ می باشد به همین خاطر ما در خروجی گاهی ۰ و گاهی ۱ یعنی پالس داریم و پایه ۴ شمارنده هم همیشه ۱ است در نتیجه این شمارنده افزایشی است حال اگر پایه $R \leftarrow 0$ و پایه $S \leftarrow 1$ باشد در نتیجه $Q \leftarrow 0$ و $\bar{Q} \leftarrow 1$ خواهد بود در نتیجه شمارنده کاهشی می شود.

۹ → ۰ و ۰ → ۹



توضیح: هرگاه شمارنده عدد ۹ را نشان داد، خروجی گیت NAND شده و در نتیجه پایه های $Q \leftarrow 1$ و $\bar{Q} \leftarrow 0$ خواهد شد و در نتیجه شمارنده کاهشی می شود.

منابع اخذ شده :



اساتید محترم دانشگاهای برق و کامپیوتر داخل وخارج از کشور

سایت های اینترنتی برق و کامپیوتر داخل وخارج از کشور

با تشکر از اساتید محترم دانشگاه صنعتی شریف تهران

 Site address: www.aduelect.ir

 E-mail address: Rezaalipour124@gmail.com

 Author: Rezaalipour

موفق و سربلند باشید

مؤلف : مهندس رضا علیپور

آزمایشگاه مبانی دیجیتال

تقديم به دانشجویان رشته ی مخابرات ، کامپیوتر، الکترونیک

تکنولوژی ارتباطات و فناوری اطلاعات



 Site address: www.aduelect.ir

 E-mail address: Rezaalipour124@gmail.com

 Author: Engineer Rezaalipour

مؤلف : مهندس رضا علیپور