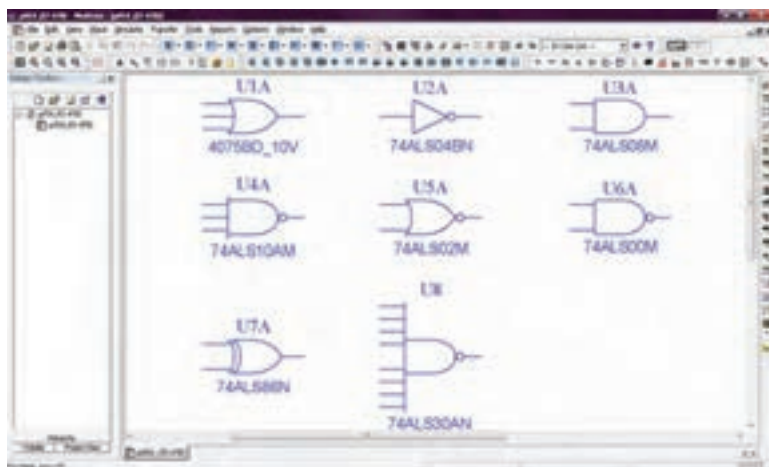
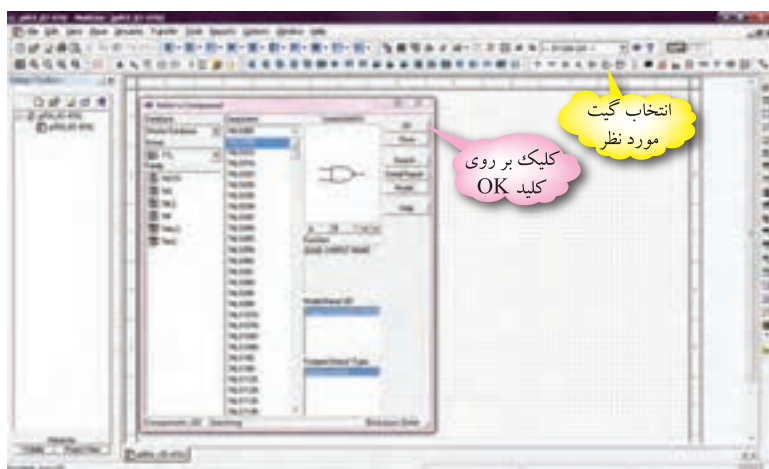


## دانش افزایی

استفاده از نرم افزار:

با استفاده از نرم افزار مولتی سیم می توانید دروازه های منطقی را بر روی میز کار نرم افزار بیاورید.  
در شکل زیر طریقه استفاده از کلیدهای مربوط به دروازه های منطقی را مشاهده می کنید.



نمونه ای از دروازه های منطقی در نرم افزار



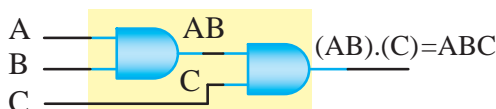
انواع دروازه‌های منطقی را در نرم‌افزار مولتی سیم شناسایی کنید.

## دانش افزایی در مورد توابع ترکیبی و پاسخ فعالیت‌های کارگاهی

### افزایش ظرفیت ورودی‌های دروازه‌های منطقی:

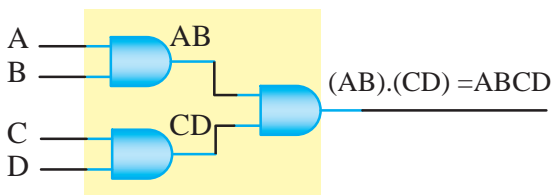
هرچند که در عمل دروازه‌های منطقی تا هشت ورودی نیز ساخته می‌شوند ولی گاهی به بیش از هشت ورودی نیاز داریم یا به دروازه‌های منطقی با ورودی حتی کمتر از هشت نیازمندیم ولی در دسترس نیستند در هر یک از این شرایط، می‌توان با استفاده از دروازه‌های منطقی موجود یک دروازه منطقی با تعداد ورودی‌های دلخواه ساخت. در این قسمت به شرح روش افزایش تعداد ورودی‌های بعضی از دروازه‌های منطقی می‌پردازیم.

افزایش تعداد ورودی‌های دروازه منطقی **AND**: فرض کنید یک سری دروازه‌های منطقی **AND** با دو ورودی در دسترس داریم و در عمل، به یک دروازه **AND** با سه ورودی نیازمندیم. مدار شکل زیر این نیاز را برآورده می‌کند.



نحوه ساخت دروازه **AND** با سه ورودی با استفاده از دو دروازه **AND** با دو ورودی

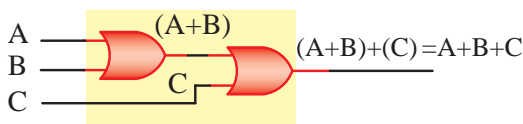
معادل دروازه منطقی **AND** با چهار ورودی را با استفاده از سه دروازه منطقی **AND** با دو ورودی نشان می‌دهد.



معادل دروازه منطقی **AND** با چهار ورودی با استفاده از سه دروازه منطقی **AND** با دو ورودی

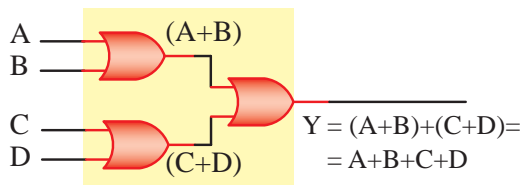
به همین شیوه می‌توان یک دروازه منطقی با هر تعداد ورودی ساخت. افزایش تعداد ورودی‌های دروازه منطقی **OR**: برای افزایش تعداد ورودی‌های دروازه منطقی **OR**، می‌توانیم از تعداد بیشتری دروازه منطقی **OR** با تعداد ورودی

کمتر استفاده کنیم. برای مثال، شکل زیر معادل یک دروازه منطقی OR با سه ورودی را با استفاده از دو دروازه منطقی OR با دو ورودی نشان می‌دهد.



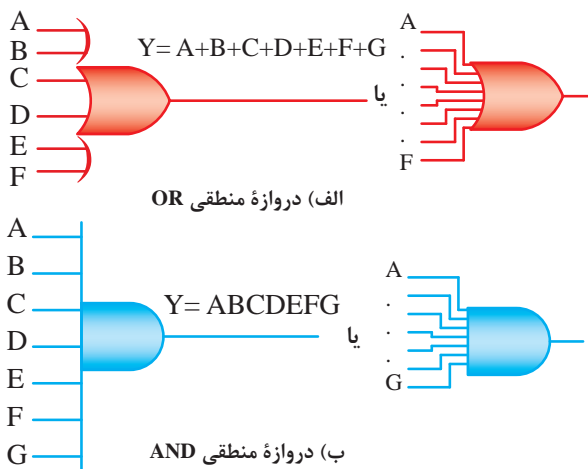
نحوه ساخت دروازه منطقی OR با سه ورودی با استفاده از دو دروازه منطقی OR با دو ورودی

شکل معادل یک دروازه منطقی OR با چهار ورودی را با استفاده از سه دروازه منطقی OR با دو ورودی نشان می‌دهد.



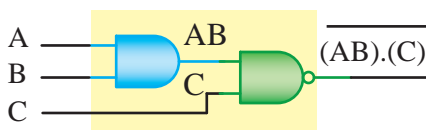
نحوه ساخت دروازه منطقی OR با چهار ورودی با استفاده از دروازه‌های منطقی OR با دو ورودی

به همین روش می‌توان دروازه منطقی OR با هر تعداد ورودی ایجاد کرد. اگر تعداد ورودی‌های یک دروازه منطقی زیاد شوند، معمولاً نماد آن دروازه منطقی را در مقایسه با دروازه منطقی معمولی بزرگ‌تر رسم نمی‌کنند بلکه آن را به اندازه استاندارد به صورت شکل الف و ب رسم می‌کنند.



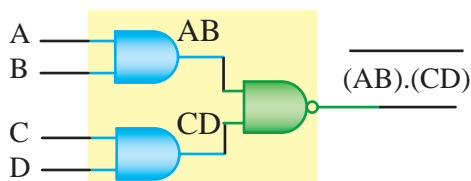
نمادهای دروازه‌های منطقی AND و OR با ورودی‌های زیاد

افزایش تعداد ورودی‌های دروازه منطقی NAND: با استفاده از دروازه‌های منطقی AND و NAND می‌توان یک دروازه منطقی NAND را با هر تعداد ورودی ساخت. شکل یک دروازه منطقی NAND با سه ورودی را با استفاده از یک دروازه منطقی AND و یک دروازه منطقی NAND نشان می‌دهد.



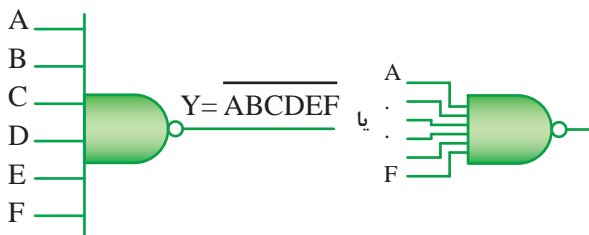
دروازه منطقی NAND با سه ورودی با استفاده از دروازه منطقی AND و NAND

شکل یک دروازه منطقی NAND با چهار ورودی را نشان می‌دهد. در این مدار از یک دروازه منطقی NAND با دو ورودی و دو دروازه منطقی AND استفاده شده است.



دروازه منطقی NAND با چهار ورودی با استفاده از دروازه‌های منطقی AND و NAND با دو ورودی

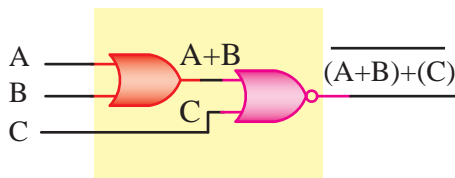
به همین روش، می‌توان دروازه منطقی NAND با هر تعداد ورودی ساخت. شکل زیر نمادهای دروازه منطقی NAND با شش ورودی را نشان می‌دهد.



نمادهای دروازه منطقی NAND با ورودی‌های زیاد

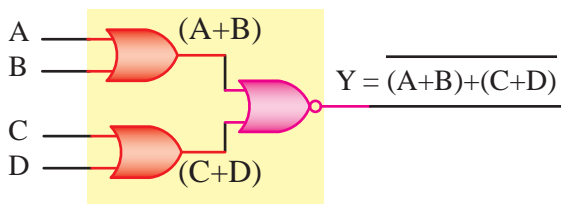
افزایش تعداد ورودی‌های دروازه منطقی NOR: با استفاده از دروازه‌های منطقی OR و NOR می‌توان یک دروازه منطقی NOR با هر تعداد ورودی ساخت. شکل صفحه بعد

یک دروازه منطقی NOR با سه ورودی را که با استفاده از یک دروازه منطقی OR و یک دروازه منطقی NOR ساخته شده است، نشان می‌دهد.



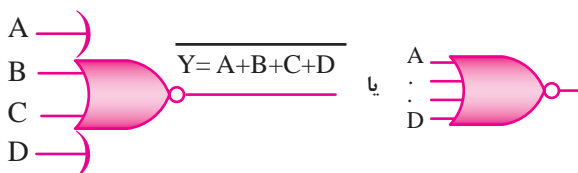
دروازه منطقی NOR با سه ورودی که با استفاده از دروازه‌های منطقی OR و NOR ساخته شده است.

شکل زیر یک دروازه منطقی NOR با چهار ورودی را که با استفاده از دروازه‌های منطقی OR و NOR با دو ورودی ساخته شده‌اند را نشان می‌دهد.



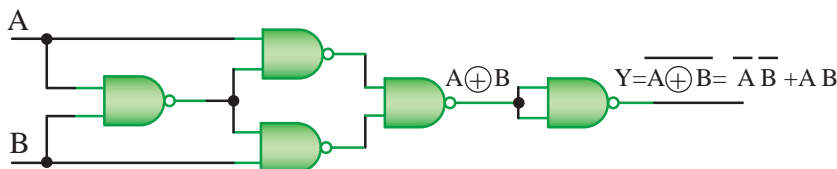
ایجاد دروازه منطقی NOR با چهار ورودی با استفاده از دروازه‌های منطقی OR و NOR با دو ورودی

به همین روش، می‌توان دروازه منطقی NOR را با هر تعداد ورودی ساخت. شکل زیر نمادهای دروازه منطقی NOR با چهار ورودی (یا بیشتر) را نشان می‌دهد.



نمادهای دروازه منطقی NOR با چهار ورودی

دروازه منطقی NOR انحصاری (XNOR): برای ساخت این دروازه منطقی با استفاده از دروازه‌های منطقی NAND، کافی است خروجی دروازه منطقی XOR را که با استفاده از گیت NAND ساخته شده است NOT کنیم. در شکل صفحه بعد دروازه منطقی XNOR را با استفاده از دروازه منطقی NAND مشاهده می‌کنید.

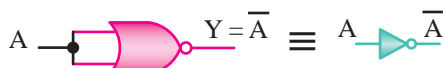


ساخت دروازه منطقی NOR انحصاری با استفاده از NAND

### ساخت دروازه‌های منطقی مختلف با استفاده از گیت NOR:

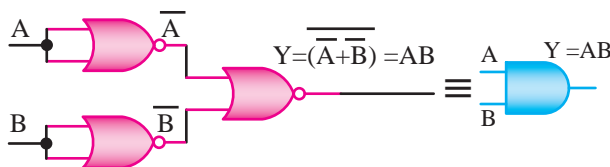
در مدارهای منطقی دروازه منطقی NOR، نیز دروازه منطقی پایه محسوب می‌شود. بنابراین، با این دروازه منطقی می‌توان سایر دروازه‌های منطقی را ساخت. در این قسمت نحوه ساخت سایر دروازه‌های منطقی به کمک دروازه منطقی NOR را مورد بررسی قرار می‌دهیم.

**ساخت دروازه منطقی NOT (نه):** اگر تمامی ورودی‌های دروازه منطقی NOR را به یکدیگر وصل کنیم، یک دروازه منطقی NOT به دست می‌آید. شکل این گیت را نشان می‌دهد.



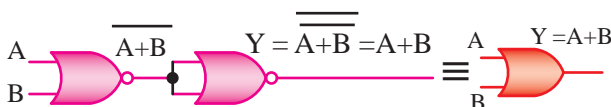
ساخت دروازه منطقی NOT با استفاده از NOR

**ایجاد دروازه منطقی AND:** به کمک سه عدد دروازه منطقی NOR می‌توان یک دروازه منطقی AND مطابق شکل ساخت:



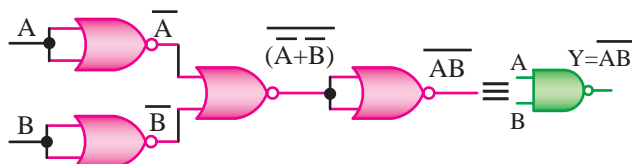
ساخت دروازه منطقی AND با استفاده از NOR

**تولید دروازه منطقی OR:** با کمک دو عدد دروازه منطقی NOR می‌توان یک دروازه منطقی OR مطابق شکل زیر ساخت زیرا اگر هر تابع دو بار NOT شود خود تابع به دست می‌آید.



ساخت دروازه منطقی OR با استفاده از NOR

**دروازه منطقی NAND:** در شکل زیر چگونگی ساخت دروازه منطقی NAND را با استفاده از دروازه منطقی NOR مشاهده می کنید.

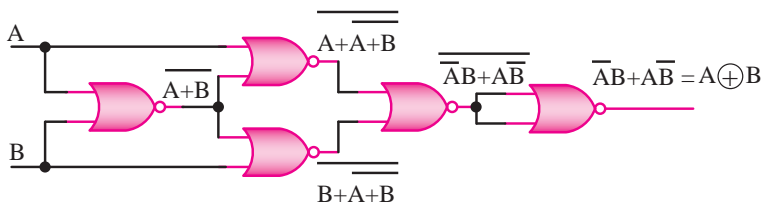


ساخت دروازه منطقی NAND با استفاده از NOR

همان طور که مشاهده می شود ابتدا با استفاده از دروازه NOR، دو دروازه NOT تشکیل می دهیم. سپس خروجی ها را مجدداً NOR می کنیم خروجی را دوباره NOT می کنیم و خروجی نهایی گیت NAND خواهد شد. دروازه منطقی XOR: می دانیم رابطه منطقی XOR به صورت زیر است.

$$Y = A \oplus B = \overline{A}B + A\overline{B}$$

و مدار ساخته شده در شکل را با گیت های NOR مطابق شکل رسم می کنیم.

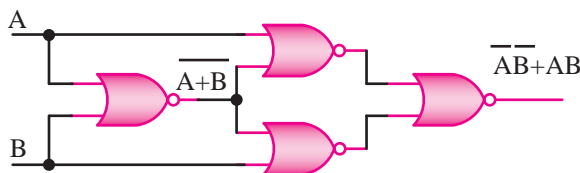


ساخت دروازه منطقی XOR با استفاده از NOR

**دروازه منطقی XNOR:** می دانیم رابطه XNOR به صورت زیر است:

$$Y = A \oplus B = \overline{A}B + AB$$

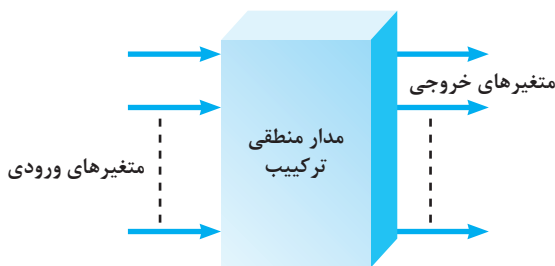
از شکل مدار استفاده می کنیم و با حذف آخرین مدار NOT شکل زیر طراحی می شود.



ساخت دروازه منطقی XNOR با استفاده از NOR

### مدارهای ترکیبی

مدارهای ترکیبی، مدارهایی هستند که خروجی‌های آنها به‌طور هم‌زمان به ورودی‌های آنها بستگی دارد. به عبارت دیگر اگر ورودی‌هایی را به شبکه اعمال کنیم و به اندازه کافی صبر کنیم تا روی خروجی‌ها اثر بگذارند، در این شرایط مقادیر خروجی‌ها تحت تأثیر آخرین مجموعه مقادیر ورودی‌ها قرار می‌گیرند و مقدار آنها را مشخص می‌کند. «رفتار مدار» به آنچه که از مدارهای ترکیبی بیان شد گفته می‌شود. می‌توانیم مدارهای ترکیبی را براساس ساختار مدار نیز تعریف کنیم. به عبارت ساده، مدارهای ترکیبی مدارهایی بدون فید بک (پس‌خورد) و بدون عنصر حافظه هستند. یک مدار ترکیبی شامل متغیرهای ورودی، دروازه‌های منطقی و متغیرهای خروجی است. شکل زیر بلوک دیاگرام یک مدار ترکیبی را نشان می‌دهد.



شکل بلوک دیاگرام یک مدار ترکیبی

روش طراحی مدارهای ترکیبی: طراحی مدارهای ترکیبی با تعریف یک مسئله شروع مجموعه‌های «یا» منطقی مدار «می‌شود و با دیاگرام که با استفاده از آنها می‌توان به سادگی» از توابع بول دیاگرام منطقی را به‌دست آورد پایان می‌یابد. مراحل زیر روند طراحی مدارهای ترکیبی را نشان می‌دهد.

- ۱ تعریف دقیق مسئله
- ۲ تعیین تعداد ورودی‌ها و خروجی‌های لازم
- ۳ تشکیل جدول درستی مدار که ارتباط بین ورودی‌ها و خروجی‌ها را برقرار کند
- ۴ نوشتن تابع منطقی
- ۵ ساده‌سازی تابع منطقی بولی به‌دست آمده برای هر یک از خروجی‌های مدار
- ۶ رسم مدار منطقی با حداقل گیت یا با گیت‌های خواسته شده



نکته



در طراحی مدار سعی می‌کنیم تعداد دروازه‌های منطقی و تعداد ورودی‌های آن حداقل باشد.

کار در کلاس



جدول درستی یک مدار ترکیبی از ستون‌های ورودی و ستون‌های خروجی تشکیل می‌شود. براساس صورت مسئله، با استفاده از  $2^n$  حالت ورودی حالت‌های خروجی را به دست می‌آوریم. مشخصات مسئله ممکن است به گونه‌ای باشد که بعضی از ترکیب‌های ورودی هرگز به وجود نیایند. که این حالت‌ها را «حالت‌های بی‌اهمیت» (don't care) می‌گویند.

مداری با سه ورودی A، B و C طراحی کنید، که اگر ورودی B یک باشد خروجی یک شود.

پاسخ:

الف) رسم جدول صحت: ابتدا جدول صحت را با سه متغیر ورودی و یک خروجی رسم می‌کنیم. با توجه به صورت مسئله در جدول صحت، در شرایطی که ورودی B یک است، خروجی را یک می‌نویسیم.

جدول صحت مربوط به مثال

شماره سطر	A	B	C	F
۰	۰	۰	۰	۰
۱	۰	۰	۱	۰
۲	۰	۱	۰	۱
۳	۰	۱	۱	۱
۴	۱	۰	۰	۰
۵	۱	۰	۱	۰
۶	۱	۱	۰	۱
۷	۱	۱	۱	۱

ب) نوشتن تابع منطقی مدار: تابع منطقی را با توجه به حاصل نهایی که خروجی

یک است می‌نویسیم برای نوشتن تابع منطقی از حاصل جمع حاصل ضرب‌ها (مینترم) استفاده می‌کنیم.

$$F = \overline{A} \overline{B} \overline{C} + \overline{A} B \overline{C} + A \overline{B} \overline{C} + A B \overline{C} + A B C$$

سطر ۷ سطر ۶ سطر ۳ سطر ۲

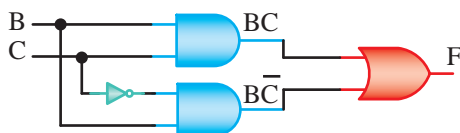
ج) ساده کردن تابع منطقی: از سطر ۲ و ۶ عبارت  $\overline{B} \overline{C}$  و از سطر ۳ و ۷ عبارت  $BC$  را فاکتورگیری می‌کنیم.

$$F = \overline{B} \overline{C} (\overline{A} + A) + BC (\overline{A} + A)$$

می‌دانیم هر متغیر که با مکمل خودش جمع شود حاصل آن یک است. همچنین اگر «یک» منطقی در عبارتی ضرب شود، حاصل همان عبارت خواهد بود. بنابراین عبارت خروجی به صورت زیر در می‌آید.

$$F = \overline{B} \overline{C} + BC$$

د) طراحی مدار: با توجه به عبارت منطقی به دست آمده برای تابع  $F$ ، مدار شکل زیر را طراحی می‌کنیم.

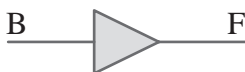


شکل مدار مربوط به مثال

کمی به عبارت خروجی  $F$  توجه کنید، در می‌یابید که می‌توانید تابع  $F$  را ساده‌تر کنید. از متغیر  $B$  فاکتورگیری می‌کنیم، حاصل به صورت زیر در می‌آید.

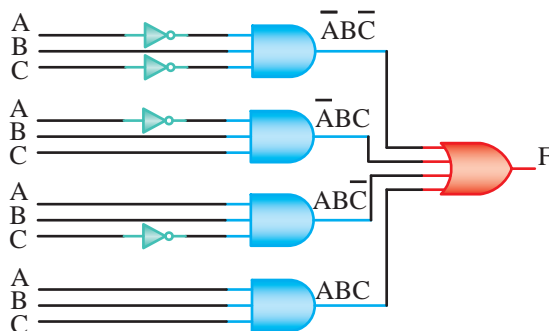
$$F = B(\overline{C} + C) = B$$

در نتیجه وقتی خروجی یک است که ورودی  $B$  یک باشد. به عبارت دیگر، سایر متغیرهای ورودی تأثیری در خروجی ندارند و می‌توانند حذف شوند. مدار شکل زیر مدار ساده شده عبارت خروجی  $F$  است.



شکل مدار ساده شده مثال

اگر از قوانین مربوط به ساده کردن عبارت‌های منطقی استفاده نمی‌کردیم باید مدار ساده‌ی مثال را به صورت مدار پیچیده‌ی شکل زیر طراحی کنیم.



شکل مدار مثال قبل از ساده شدن

مثال



مداری طراحی کنید که یک قفل را با ۳ کلید دو وضعیت  $A$ ،  $B$  و  $C$  کنترل کند. رمز قفل در حالتی باز می‌شود که فقط یک کلید بسته باشد (بسته بودن به مفهوم «یک» منطقی است).

پاسخ:

الف) ترسیم جدول صحت: ابتدا با توجه به تعداد متغیرها (۳ متغیر) و تعداد حالت‌ها ( $2^3 = 8$ ) جدول صحت را مطابق جدول زیر رسم می‌کنیم. در این جدول ستون ۱ شماره سطر و ستون ۲ حالت‌های ورودی‌ها را نشان می‌دهد.

جدول صحت مربوط به مثال

شماره سطر	A	B	C	F
۰	۰	۰	۰	۰
۱	۰	۰	۱	۱
۲	۰	۱	۰	۱
۳	۰	۱	۱	۰
۴	۱	۰	۰	۱
۵	۱	۰	۱	۰
۶	۱	۱	۰	۰
۷	۱	۱	۱	۰

ستون ۴ همان خروجی مدار یا F است که براساس تعریف کارمدار، آن را تکمیل می‌کنیم. بنابر صورت مسئله، اگر فقط یکی از ورودی‌ها، یک و بقیه صفر باشند، ستون خروجی یعنی F یک خواهد شد.

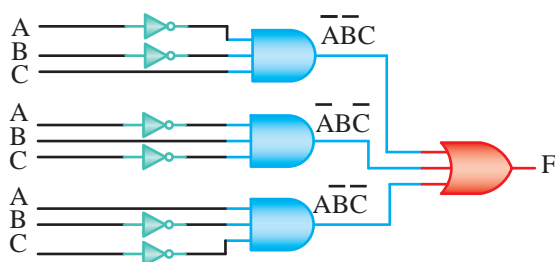
(ب) نوشتن تابع منطقی مدار: حال باید تابع منطقی را برای حالت‌هایی که مقدار خروجی یک است، بنویسیم. برای نوشتن تابع منطقی می‌توانیم از حاصل جمع حاصل ضرب‌ها (مین ترم) استفاده کنیم.

$$F = \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C}$$

سطر ۴    سطر ۲    سطر ۱

(ج) ساده کردن تابع منطقی: تابع منطقی به‌دست آمده ساده نمی‌شود.

(د) طراحی مدار: برای طراحی مدار ابتدا از تابع منطقی به‌دست آمده (F) کمک می‌گیریم، سپس با استفاده از دروازه‌های منطقی AND، OR و NOT مدار را طراحی می‌کنیم.



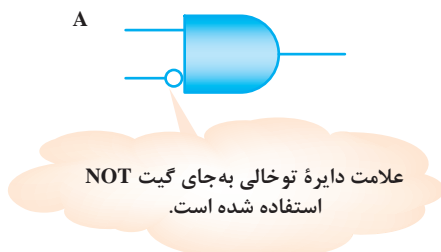
شکل مدار مثال

همان‌طور که در شکل مشاهده می‌کنید بعضی از ورودی‌ها برای اتصال به گیت بعدی، ابتدا وارد گیت منطقی NOT می‌شوند، تا به‌صورت متمم درآیند. برای هر یک از گیت‌های NOT، یک نماد مداری مشابه شکل زیر رسم شده است.



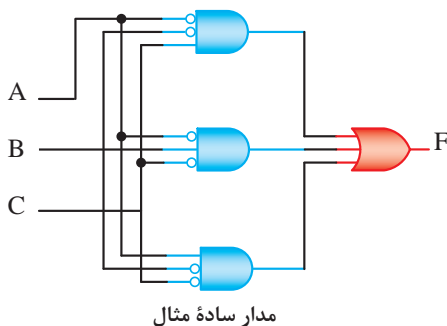
شکل دروازه منطقی NOT

برای ساده‌تر شدن مدارهای ترسیمی، در رسم مدارها معمولاً گیت NOT را با یک دایره تو خالی در ورودی گیت مورد نظر (مثلاً گیت AND) نشان می‌دهند، در شکل این اتصال را مشاهده می‌کنید.



شکل نماد گیت NOT به صورت دایره تو خالی

با استفاده از نماد جدید، مدار شکل بالا را می توان به صورت ساده تر در آورد.



مدار ساده مثال

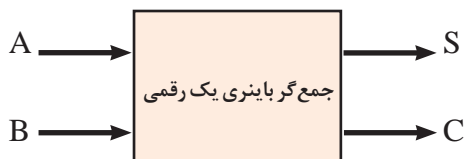
پاسخ و دانش افزایی فعالیت کارگاهی جمع کننده ناقص H.A (Half Adder):

جمع دو رقم دودویی را می توان بامداری به نام جمع کننده ناقص یا به اختصار H.A انجام دهیم. مدار H.A مداری است که دو ورودی (A و B) و دو خروجی (S و C) دارد. رقم اول S حاصل جمع SUM و C رقم نقلی یا Carry را مشخص می کند. جدول صحت جمع دوبیت A و B را در جدول زیر مشاهده می کنید.

جدول صحت نیم جمع کننده

A	B	C	S	
۰	۰	۰	۰	$0+0=0$
۰	۱	۰	۱	$0+1=1$
۱	۰	۰	۱	$1+0=1$
۱	۱	۱	۰	$1+1=10$

بلوک دیاگرام مداری که بتواند دو رقم باینری را با هم جمع کند، در شکل زیر نشان داده‌ایم.



شکل بلوک دیاگرام جمع‌کننده ناقص

با توجه به جدول به این نتیجه می‌رسیم که اگر فقط دو بیت با هم جمع شوند، رقم نقلی از قبل وجود ندارد و جمع‌کننده ناقص (نیم جمع‌کننده) می‌تواند این عمل جمع را انجام دهد.

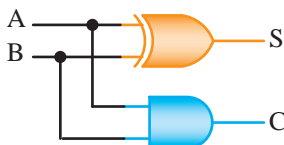
مطابق جدول صحت مدار، تابع S شامل دو جمله  $\overline{A}B$  و  $A\overline{B}$  است یعنی:

$$S = \overline{A}B + A\overline{B} = A \oplus B$$

از طرفی تابع C فقط شامل یک جمله AB است یعنی:

$$C = AB$$

با استفاده از مقادیر S و C به کمک یک دروازه XOR و یک دروازه AND می‌توانیم مداری مطابق شکل زیر را طراحی کنیم.



شکل مدار نیم جمع‌گر باینری یک رقمی

در این جمع‌گر، ورودی رقم نقلی پیش‌بینی نشده است. به همین دلیل، به آن جمع‌گر ناقص یا نیم جمع‌گر می‌گویند.

**پاسخ و دانش افزایی فعالیت کارگاهی جمع‌کننده کامل F.A (Full Adder):**

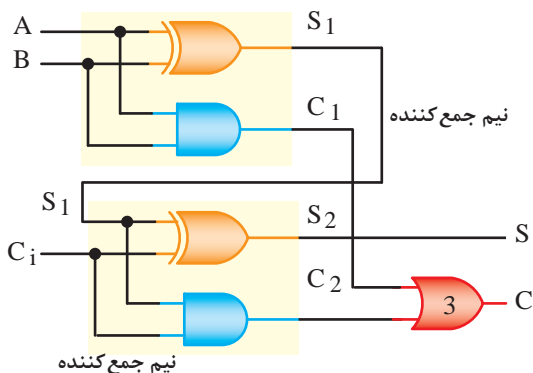
برای انجام عملیات جمع اعداد دودویی نیاز به مداری داریم که بتواند ۳ رقم یک بیتی باینری را با هم جمع کند. چنین مداری را جمع‌کننده کامل می‌گویند و با F.A نشان می‌دهند.

جمع‌کننده کامل یا تمام جمع‌گر مداری است که ۳ خط ورودی (A، B و  $C_i$ ) و دو خط خروجی (S و C) را دارد. به این ترتیب مدار جمع‌کننده کامل می‌تواند دو رقم دودویی و یک Carry که از مرحله قبل حاصل شده است را با هم جمع کند.

با بیان مثالی در جمع اعداد ده دهی درک موضوع را ساده تر می کنیم. در جمع دو عدد ۹۵۶ و ۳۸۷ وقتی می خواهیم دو رقم دهگان ۵ و ۸ را جمع کنیم ابتدا باید حاصل جمع دو رقم یکان را به دست آوریم از حاصل جمع دو رقم یکان ۶ و ۷ عدد ۱۳ حاصل می شود که ۳ را در ستون یکان می نویسیم و رقم یک را به ستون دهگان انتقال می دهیم که اصطلاحاً ده بر یک می گوییم.

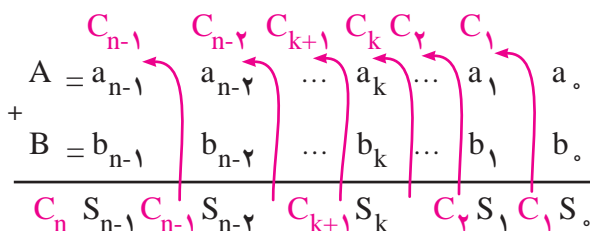
$$\begin{array}{r}
 \text{رقم نقلی} \quad 1 \\
 \phantom{1} \quad 1 \phantom{0} \phantom{0} \\
 \phantom{1} \quad 3 \phantom{0} \phantom{0} \\
 + \quad 9 \phantom{0} \phantom{0} \\
 \hline
 1 \phantom{0} \phantom{0} 3 \phantom{0} 4 \phantom{0} 3
 \end{array}$$

عدد یک، رقم نقلی است که در جمع ۸ و ۵ تأثیر می گذارد در سیستم باینری نیز به همین ترتیب رقم نقلی وجود دارد و روی عمل جمع اثر می گذارد. به کمک دو نیم جمع کننده می توان یک تمام جمع گر یا یک جمع گر یک بیتی کامل مطابق شکل ساخت.



شکل مدار تمام جمع گر یک بیتی

به طور مثال برای جمع دو عدد باینری چهار بیتی ۱۰۱۱ و ۱۱۰۱ خواهیم داشت:



یعنی، تمام جمع گری که در موقعیت مکانی  $K$  قرار می گیرد، باید سه بیت،  $C_k$ ،  $a_k$  و  $b_k$  را با یک دیگر جمع کند.

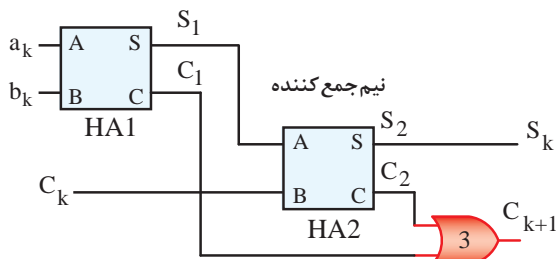
$$C_k + a_k + b_k$$

رابطه بالا را می توانیم به صورت زیر بنویسیم.

$$C_k + a_k + b_k = C_k + (a_k + b_k)$$

یعنی، ابتدا ارقام  $a_k$  و  $b_k$  را با هم جمع کنیم و سپس حاصل جمع آنها را با  $C_k$  جمع می کنیم. در طراحی مدار نیز باید به همین ترتیب عمل کنیم. در شکل زیر مدارهای نیم جمع کننده شکل به صورت بلوک دیاگرام نشان داده شده است در این مدار ابتدا نیم جمع کننده  $AH_1$  ارقام  $a_k$  و  $b_k$  را با هم جمع می کند و سپس نیم جمع کننده  $AH_2$  حاصل جمع این دو رقم را با  $C_k$  به دست می آورد.

توجه داشته باشید که هرگز ورودی های دروازه OR (گیت شماره ۳) هم زمان یک نمی شود، زیرا این امر مستلزم آن است که هر دو نیم جمع کننده رقم نقلی یک ایجاد کنند، و این تنها در صورتی تحقق می یابد که هر دو ورودی نیم جمع کننده یک باشند.



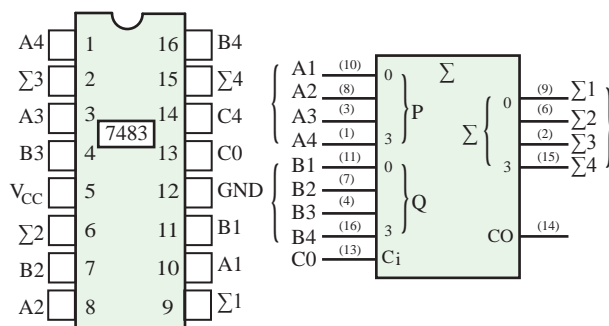
شکل بلوک دیاگرام تمام جمع گر



حال اگر هر دو ورودی  $a_k$  و  $b_k$  برابر یک باشند، خروجی  $S_1=0$  و  $C_1=1$  می‌شود. چون خروجی  $S_1=0$  است، خروجی  $C_2=0$  می‌شود، (آیا می‌دانید چرا؟) اگر ورودی‌های  $a_k$  و  $b_k$  هم‌زمان یک نشوند، خروجی  $C_1=0$  می‌شود؛ یعنی، هرگز خروجی‌های  $C_1$  و  $C_2$  هم‌زمان یک نمی‌شوند.

پاسخ و دانش افزایی فعالیت کارگاهی جمع‌کننده چهار بیتی:

جمع‌گرهای باینری چهار بیتی با رقم نقلی موجی به صورت آی سی عرضه می‌شود. نمونه‌هایی از این آی سی‌ها به صورت تجارتي ۱۶ پایه با شماره‌های ۷۴LS۸۳، ۷۴HC ۷۴۸۳، ۷۴A۸۳A، در بازار وجود دارند. در شکل زیر مشخصات تراشه ۷۴۸۳A را مشاهده می‌کنید.

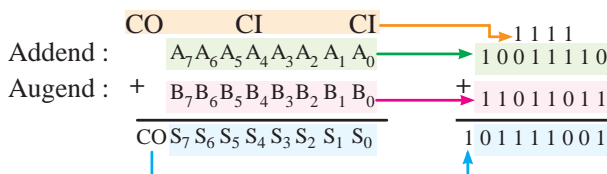
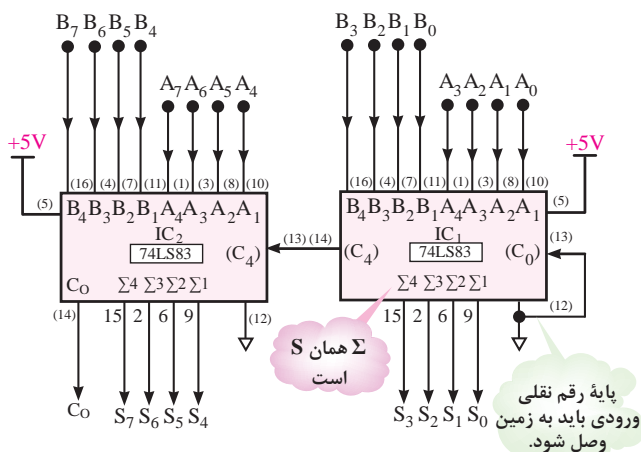


شکل مشخصات جمع‌گر چهار بیتی ۷۴۸۳A

برای جمع کردن اعداد بزرگ‌تر از چهار رقم می‌توان از چند تراشه تجارتي استفاده کرد.

در شکل بالا چگونگی انجام دادن عمل جمع دو عدد هشت بیتی را با استفاده از دو جمع‌گر چهار بیتی ۷۴LS ۸۳ مشاهده می‌کنید. در تراشه سمت راست چون رقم نقلی از مرحله اولی وجود ندارد پایه ۱۳ این آی سی به زمین اتصال داده می‌شود. دو عدد هشت بیتی  $A_7A_6A_5A_4A_3A_2A_1A_0$  و  $B_7B_6B_5B_4B_3B_2B_1B_0$  را با هم جمع می‌کنیم. ابتدا در آی سی  $IC_1$  (سمت راستی) ارقام  $A_3A_2A_1A_0$  با  $B_3B_2B_1B_0$  جمع می‌شوند و اگر رقم نقلی ایجاد شود به پایه ۱۴ آی سی سمت چپ انتقال می‌یابد. سپس ارقام  $A_7A_6A_5A_4$  با  $B_7B_6B_5B_4$  جمع شده و در صورتی که رقم نقلی ایجاد شود در پایه ۱۴ آی سی دوم ظاهر می‌شود. حاصل جمع چهار بیت کم‌ارزش‌تر در پایه‌های ۹، ۶، ۲ و ۱۵ آی سی سمت راست و حاصل جمع چهار بیت با ارزش بالاتر را در پایه‌های ۹، ۶، ۲ و ۱۵ آی سی سمت چپ مشاهده می‌کنید.

کار پیشنهادی: با استفاده از نرم افزار مولتی سیم یا پروتئوس دو عدد آی سی ۷۴۸۳ مدار شکل را ببندید و با تغییر ورودی ها به حالت صفر و یک نتیجه را در خروجی های مدار مشاهده کنید.



شکل چگونگی انجام دادن عمل جمع دو عدد هشت بیتی به کمک دو جمع گر ۸۳ SL ۷۴

کار پیشنهادی: با استفاده از نرم افزار مولتی سیم مدار تمام جمع گر ۴ بیتی و تمام جمع گر هشت بیتی را ببندید و نتیجه جمع را در آزمایشگاه مجازی تجربه کنید. مدار شبیه سازی شده را به کلاس ارائه نمایید.

مقایسه کننده یک بیتی: در مقایسه دو بیت، اگر فقط قرار باشد که تساوی یا عدم تساوی نشان داده شود، ساده ترین مدار استفاده از گیت XNOR است. در این مقایسه کننده اگر دو بیت مساوی باشند خروجی «۱» و در غیر این صورت خروجی «۰» می شود.

جدول صحت گیت XNOR را نشان می‌دهد.

جدول صحت گیت X - NOR

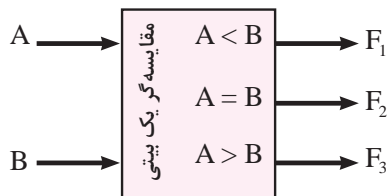
A	B	F
۰	۰	۱
۰	۱	۰
۱	۰	۰
۱	۱	۱

در مقایسه بین دو بیت، ممکن است بزرگ‌تر، کوچک‌تر و مساوی بودن بیت‌ها مورد نظر باشد. شکل زیر جدول مقایسه بین دو بیت A و B را نشان می‌دهد. مطابق جدول زیر سه خروجی داریم.

جدول مقایسه دو بیت A و B

A	B	F <sub>۱</sub> A<B	F <sub>۲</sub> A=B	F <sub>۳</sub> A>B
۰	۰	۰	۱	۰
۰	۱	۱	۰	۰
۱	۰	۰	۰	۱
۱	۱	۰	۱	۰

در شکل بلوک دیاگرام مقایسه کننده یک بیتی را مشاهده می‌کنید:



شکل بلوک دیاگرام مقایسه گر یک بیتی



با توجه به جدول صفحه قبل تابع منطقی  $F_1$ ,  $F_2$  و  $F_3$  را بنویسید.

$F_1 =$  .....

$F_2 =$  .....

$F_3 =$  .....

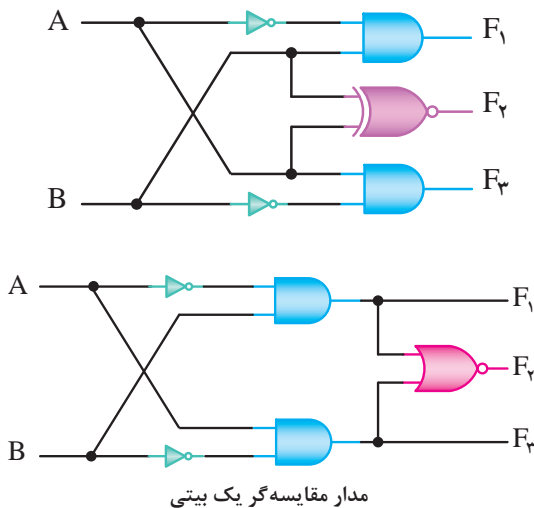
پاسخ: L به معنی Lower (کمتر)، E به معنی Equal (معادل) و G به معنی Greater (بزرگ تر) است.

$$F_L = AB$$

$$F_E = AB + \bar{A}\bar{B}$$

$$F_H = \bar{A}B$$

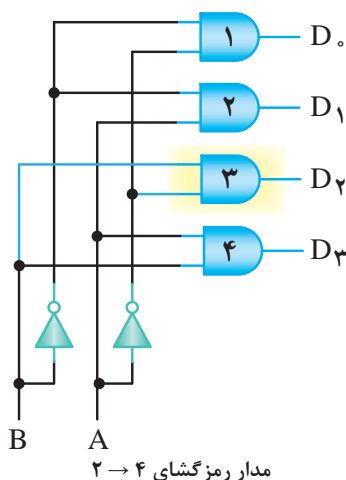
با استفاده از توابع به دست آمده از خروجی ها می توان مدار شکل زیر را طراحی کرد.



## دانش افزایی مدارهای رمزگشا (decoder)

برای دریافت اطلاعات از دستگاه های محاسباتی دیجیتالی مداری مورد نیاز است که اطلاعات را از حالت دودویی به اعشاری تبدیل کند. خروجی این مدارها وصل می شود. این تبدیل کننده ها را به نمایشگرها معمولاً رمزگشا و عملی را که انجام می دهند رمزگشایی می نامیم. خروجی است و  $2^n$  ورودی دارای حداکثر  $n$  هر رمزگشا با خروجی فعال است. به عبارت  $2^n$  در هر لحظه تنها یکی از دیگر، هر یک از خروجی های آن متناظر با یک ترکیب خاص ورودی (یک جمله حاصل ضرب نرمال یا مینترم) است.

در شکل زیر یک رمزگشای  $4 \rightarrow 2$  مشاهده می‌شود.



جدول صحت مدر رمزگشا

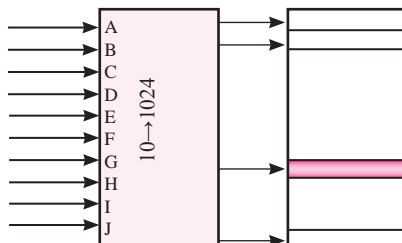
B	A	$D_0$	$D_1$	$D_2$	$D_3$
۰	۰	۱	۰	۰	۰
۰	۱	۰	۱	۰	۰
۱	۰	۰	۰	۱	۰
۱	۱	۰	۰	۰	۱

ورودی‌های A و B را ورودی‌های آدرس و خروجی‌های  $D_0, D_1, D_2$  و  $D_3$  را خروجی‌های داده می‌نامیم. همان‌طور که در شکل دیده می‌شود، خروجی دروازه شماره ۱ فقط به‌ازای ترکیب ورودی  $AB = 00$  فعال می‌شود. (۱ می‌شود) یعنی

$$D_0 = AB$$

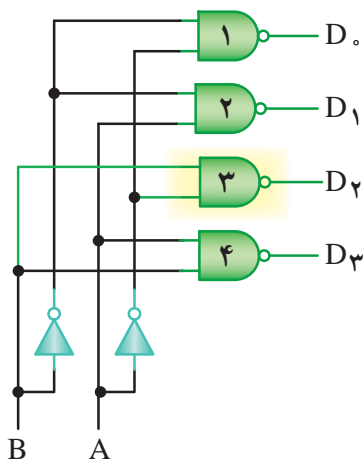
به همین ترتیب، می‌توانیم بنویسیم:  $D_1 = AB, D_2 = AB, D_3 = AB$ . (توجه کنید که در اینجا A کم ارزش‌تر و  $B_1$  متغیر با ارزش‌تر از آن است یعنی  $A=2^0=1, B=2^1=2$ ) از رمزگشاها برای آدرس دهی اجزای مختلف گیرنده یا فرستنده اطلاعات یک سیستم نیز استفاده می‌شود؛ مثلاً اگر بخواهیم کلمه‌ای را در یک سطر معینی از حافظه بخوانیم، نخست باید محل سطر مورد نظر را مشخص کنیم.

در شکل زیر سطر ۵ از یک حافظه با ظرفیت ۱۰۲۴ کلمه آدرس داده شده است (در بخش مدارهای ترتیبی با ساختمان حافظه‌های نیمه‌هادی آشنا خواهید شد).



آدرس‌دهی مکان یک کلمه معین از حافظه

ممکن است رمزگشا با دروازه‌های NAND ساخته شده باشد. در این صورت، حالت فعال خروجی‌ها «۰» خواهد بود. در شکل یک رمزگشای ۴ → ۲ نشان داده شده است.

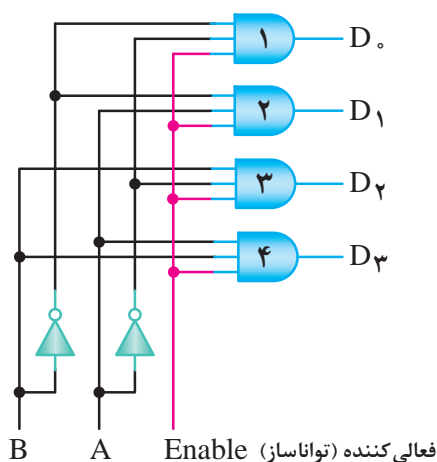


رمزگشای ۴ → ۲

جدول صحت رمزگشای ۴ → ۲

B	A	$D_0$	$D_1$	$D_2$	$D_3$
۰	۰	۰	۱	۱	۱
۰	۱	۱	۰	۱	۱
۱	۰	۱	۱	۰	۱
۱	۱	۱	۱	۱	۰

جدول صحت دکو در  $4 \rightarrow 2$  با حالت فعال LOW را نشان می‌دهد. همان‌طور که در شکل نشان داده شده، دروازه شماره ۳ در حالت فعال است، ورودی‌های این دروازه از A, B گرفته شده است. در بعضی از رمزگشاهای علاوه بر ورودی‌های آدرس، یک ورودی فعال‌کننده (تواناساز) (Enable) نیز پیش‌بینی شده است. اگر این ورودی در حالت غیرفعال نگه داشته شود، رمزگشایی انجام نخواهد شد. در شکل زیر یک رمزگشا  $4 \rightarrow 2$  با خط تواناساز به همراه جدول صحت آن را در جدول زیر مشاهده می‌کنید.



رمزگشای  $4 \rightarrow 2$  با خط تواناساز

جدول صحت دکو در  $4 \rightarrow 2$  با خط تواناساز

E	B	A	$D_0$	$D_1$	$D_2$	$D_3$
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

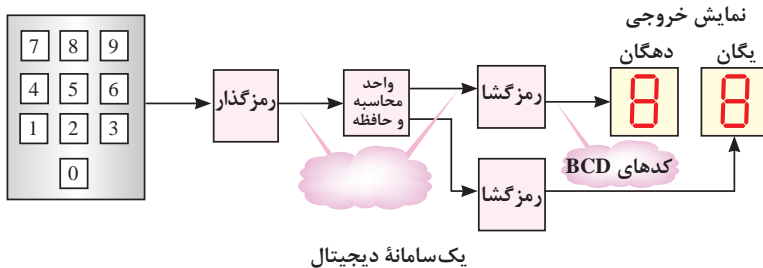


در آی سی‌ها از کلمات Enable و Disable استفاده می‌کنند. کلمه Enable به معنی فعال کننده و Disable به معنی غیر فعال کننده به کار می‌رود.

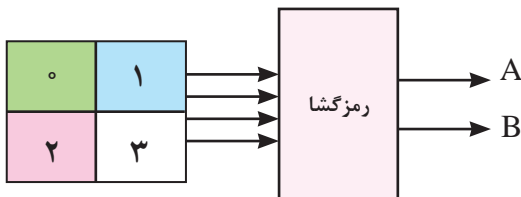
با توجه به جدول صحت صفحه قبل هرگاه یک ورودی را با X نشان دهند به معنای این است که اگر ارزش منطقی صفر یا یک باشد برای خروجی مدار بی تفاوت است. و در سطر اول جدول صحت چون ورودی E در صفر منطقی قرار گرفته است عمل رمزگشایی انجام برابر E نمی‌شود. در سطرهای دوم تا پنجم چون ورودی با «یک» است عمل رمزگشایی انجام خواهد شد.

### دانش افزایی مدارهای رمزگذار (Encoder)

اعدادی که به کامپیوترها یا سامانه دیجیتالی داده می‌شوند، در سیستم ده دهی هستند. چون کامپیوترها با اعداد باینری کار می‌کنند، اعداد ده دهی باید به اعداد باینری تبدیل شوند. مداری که اطلاعات را از حالت ده دهی به باینری تبدیل می‌کند، رمزگذار نام دارد. شکل زیر یک سامانه دیجیتالی را نشان می‌دهد.



یک رمزگذار با  $m$  ورودی ده دهی باید دارای  $n$  خروجی باشد؛ به طوری که همواره نامساوی  $m \leq 2^n$  برقرار باشد. برای مثال یک صفحه کلید با شماره‌های صفر تا سه را می‌توان به دو خط باینری تبدیل کرد. در شکل زیر بلوک دیاگرام یک رمزگذار به همراه جدول صحت در جدول زیر نشان داده شده است.



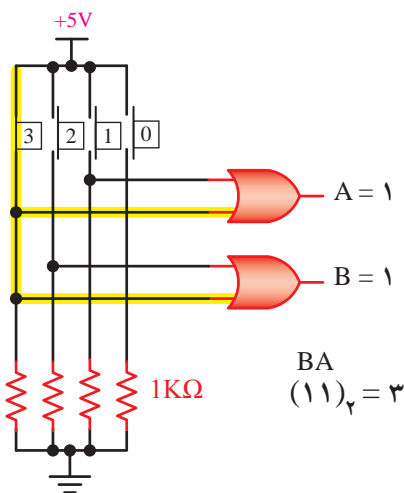
بلوک دیاگرام رمزگذار ۴ → ۲



جدول صحت رمزگذار ۲→۴

$I_r$	$I_2$	$I_1$	$I_0$	B	A
۰	۰	۰	۱	۰	۰
۰	۰	۱	۰	۰	۱
۰	۱	۰	۰	۱	۰
۱	۰	۰	۰	۱	۱

همان‌طور که مشاهده می‌کنید، در هر لحظه باید فقط یکی از ورودی‌های رمزگذار در حالت فعال قرار گیرد تا درست عمل کند. ورودی  $I_0$  و  $I_1$  ... معرف کلیدهای ورودی است و ۱ در جدول نشان‌دهنده فعال بودن آن کلید است. مدار رمزگذار ۲→۴ را می‌توان مطابق شکل زیر طراحی کرد. چنانچه کلید ۳ فشرده شود یکی از خطوط «۱» منطقی می‌شود. که سبب ایجاد عدد OR دو گیت باینری سه «۱۱» در خروجی رمزگذار می‌شود.



مدار رمزگذار ۲→۴

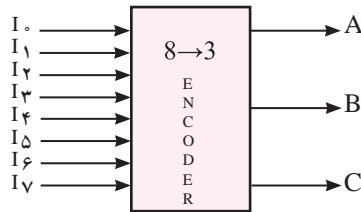
#### تحقیق



با مراجعه به منابع معتبر، آی سی با عملکرد رمزگذار ۸ به ۳ را بیابید. شماره فنی آی سی، مشخصات پایه‌ها و جدول صحت آن را به صورت گزارش به کارگاه ارائه دهید.

پاسخ:

در شکل زیر دیاگرام یک رمزگذار ۸→۳ (بخوانید ۸ به ۳) و در جدول زیر جدول صحت شده است. همان طور که مشاهده می کنید، در هر لحظه باید فقط یکی از ورودی های رمزگذار در حالت فعال باشد.



بلوک دیاگرام رمزگذار ۸→۳

جدول صحت یک رمزگشا ۸→۳

ورودی ها								خروجی ها		
I <sub>۷</sub>	I <sub>۶</sub>	I <sub>۵</sub>	I <sub>۴</sub>	I <sub>۳</sub>	I <sub>۲</sub>	I <sub>۱</sub>	I <sub>۰</sub>	D	B	A
۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۰
۰	۰	۰	۰	۰	۰	۱	۰	۰	۰	۱
۰	۰	۰	۰	۰	۱	۰	۰	۰	۱	۰
۰	۰	۰	۰	۱	۰	۰	۰	۰	۱	۱
۰	۰	۰	۱	۰	۰	۰	۰	۱	۰	۰
۰	۰	۱	۰	۰	۰	۰	۰	۱	۰	۱
۰	۱	۰	۰	۰	۰	۰	۰	۱	۱	۰
۱	۰	۰	۰	۰	۰	۰	۰	۱	۱	۱

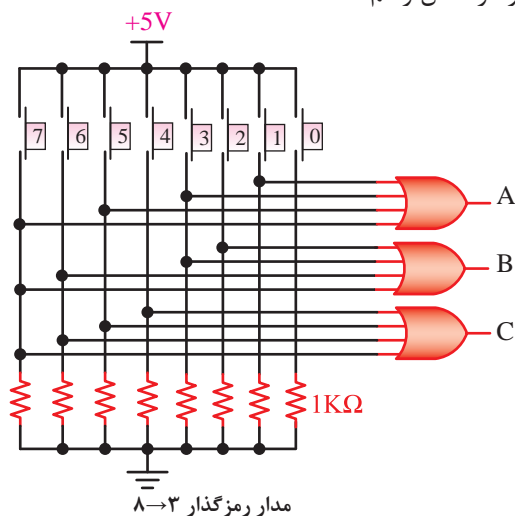
مطابق جدول فوق، معادلات بول هر یک از خروجی های A، B و C را می نویسیم:

$$A = I_1 + I_3 + I_5 + I_7$$

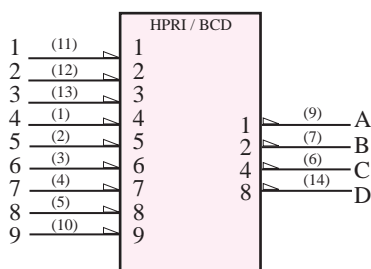
$$B = I_2 + I_3 + I_6 + I_7$$

$$C = I_4 + I_5 + I_6 + I_7$$

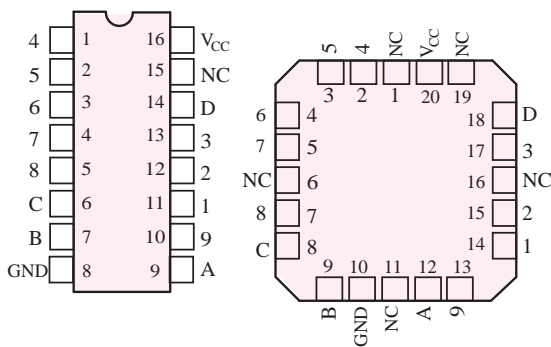
مدار این رمز گذار در شکل رسم شده است.



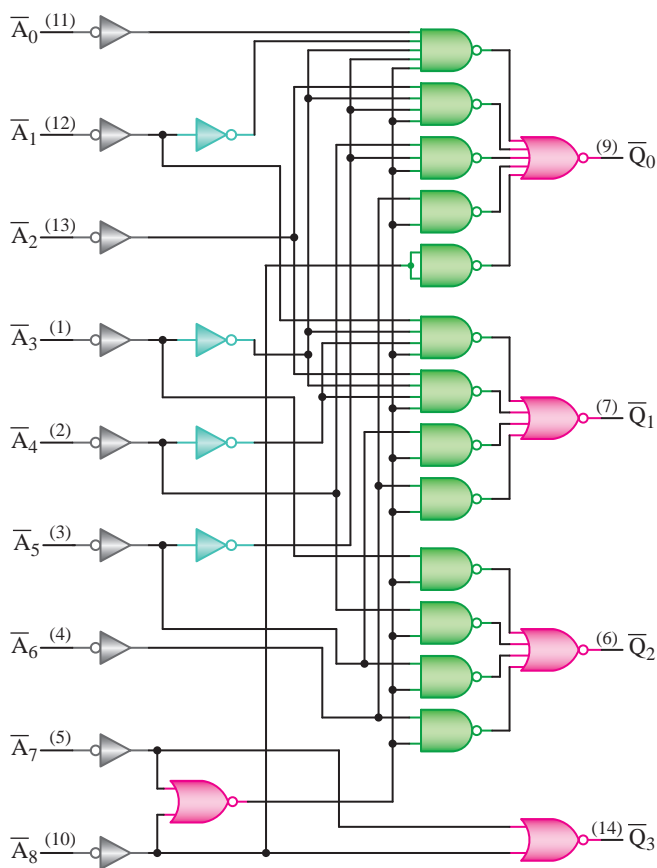
تراشه (آی سی) ۷۴۱۴۷ SN یک رمز گذار ده دهی به باینری با رعایت تقدم است. مشخصات کامل آن را در شکل زیر مشاهده می کنید. در شکل الف علامت استاندارد آی سی، در شکل ب نمای بالا و شماره پایه های آی سی و در شکل پ مدارهای داخلی آن را ملاحظه کنید.



الف) علامت استاندارد IEEE/ANSI



ب) نمای بالا و شماره پایه ها



پ مدارهای داخلی

رمزگذار تقدیمی SNV4147

جدول زیر جدول صحت آی سی ۷۴۱۴۷ SN را نشان می‌دهد.

جدول صحت رمزگذار تقدّمی SN۷۴۱۴۷

INPUTS										OUTPUTS			
۱	۲	۳	۴	۵	۶	۷	۸	۹		A <sub>۲</sub>	A <sub>۳</sub>	A <sub>۱</sub>	A <sub>۰</sub>
H	H	H	H	H	H	H	H	H		H	H	H	H
X	X	X	X	X	X	X	X	L		L	H	H	L
X	X	X	X	X	X	X	L	H		L	H	H	H
X	X	X	X	X	X	L	H	H		H	L	L	L
X	X	X	X	X	L	H	H	H		H	L	L	H
X	X	X	X	L	H	H	H	H		H	L	H	L
X	X	X	L	H	H	H	H	H		H	L	H	H
X	X	L	H	H	H	H	H	H		H	H	L	L
X	L	H	H	H	H	H	H	H		H	H	L	H
L	H	H	H	H	H	H	H	H		H	H	H	L

نکته



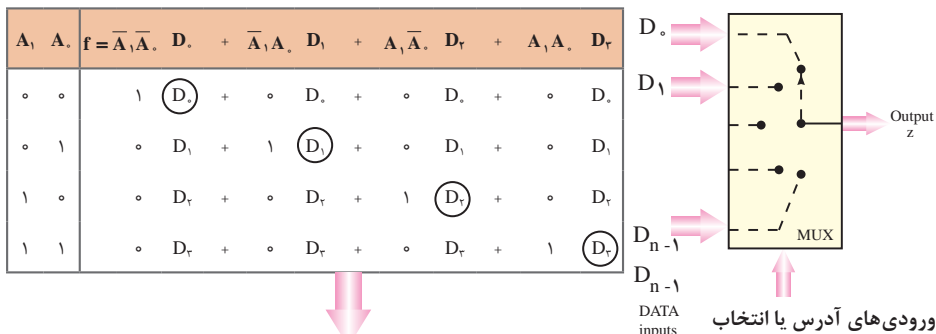
هدف از نمایش مدارهای داخلی آی سی‌ها آشنایی با مدارهاست و به هیچ عنوان در آزمون‌ها نباید سؤالی در این رابطه طراحی شود.

تحقیق



با مراجعه به رسانه‌های مختلف، نمونه‌های دیگری از آی سی مالتی پلکسر بیابید و عملکرد آنها را با استفاده از برگه اطلاعات تجزیه و تحلیل کنید. نتیجه را در قالب یک گزارش به کارگاه ارائه دهید.

مدارهای متمرکزکننده یا تسهیم‌کننده (Multiplexer): مالتی پلکسر یکی از پرکاربردترین مدارهای ترکیبی است که مانند یک انتخاب‌کننده Slector عمل می‌کند. این مدار با توجه به آدرسی که برای آن انتخاب می‌شود، به یکی از ورودی‌ها اجازه عبور می‌دهد. به عبارت دیگر، عملکرد آن درست شبیه یک کلید چندحالتی است؛ با این تفاوت که حالت کلید به صورت دیجیتالی انتخاب می‌شود. دیگرام عملیاتی یک مالتی پلکسر در شکل صفحه بعد نشان داده شده است.

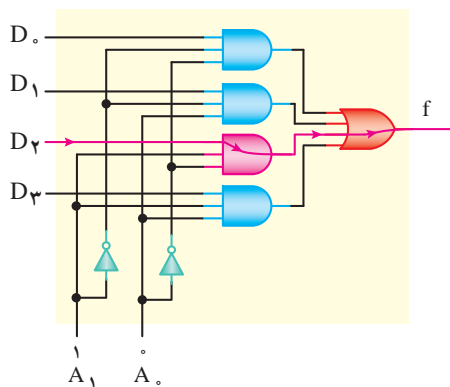


جدول ۴-۲۱- مدار تسهیم کننده در رمزگذار ۴→۱

$A_1$	$A_0$	$f$
0	0	$D_0$
0	1	$D_1$
1	0	$D_2$
1	1	$D_3$

ورودی های انتخاب حالت و ورودی های آدرس (Address Inputs) را ورودی های اصلی مالتی پلکسر ورودی های داده (Data inputs) خروجی مالتی پلکسر را خروجی تابع یا به اختصار خروجی می نامند. یک مالتی پلکسر با  $n$  خط آدرس می تواند یکی از  $2^n$  ورودی اصلی را انتخاب کند.

در شکل زیر مدار یک مالتی پلکسر ۴→۱ (بخوانید ۴ به ۱) نشان داده شده است. در این شکل ورودی های اصلی مالتی پلکسر با  $D_0$  و  $D_1$  و  $D_2$  و  $D_3$  ورودی های آدرس آن با  $A_1$  و  $A_0$  و خروجی مدار با  $F$  مشخص شده است.



مدار متمرکز کننده ۴→۱

با توجه به شکل قبل معادله بولی تابع  $F$  را می‌نویسیم.

$$F = \overline{A_1} \overline{A_0} D_0 + \overline{A_1} A_0 D_1 + A_1 \overline{A_0} D_0 + A_1 A_0 D_1$$

تابع  $F$  به فرم مجموع حاصل ضرب‌ها است. یعنی در هر حالت فقط یکی از آنها ممکن است (۱) باشد. مثلاً اگر حالت  $A_0=0$  و  $A_1=1$  را انتخاب کنیم، فقط جمله  $\overline{A_1} \overline{A_0}$  برابر (۱) است. لذا خواهیم داشت

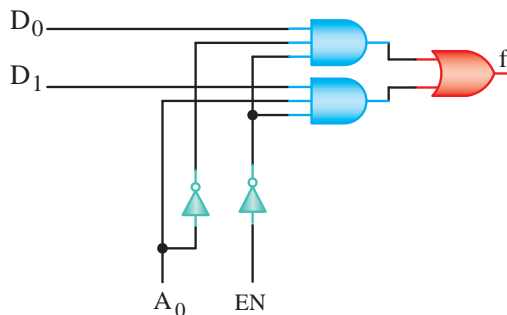
$$F = 0 \times D_0 + 0 \times D_1 + 1 \times D_0 + 1 \times D_1 = D_0 + D_1$$

یعنی  $F$  عیناً از  $D_0 + D_1$  تبعیت می‌کند. به عبارت دیگر آدرس  $\overline{A_1} \overline{A_0} = 10$  دروازه شماره ۳ را برای عبور  $D_0$  باز می‌کند. بقیه حالت‌های تابع را نیز به همین ترتیب می‌توانیم مشخص کنیم. در جدول کلیه حالت‌های تابع مشخص شده است. در اغلب مالتی پلکسرها علاوه بر ورودی‌های آدرس، یک خط کنترل اضافی نیز پیش‌بینی شده است. این ورودی اضافی با نام‌های فعال ساز، تواناساز، مشخص می‌شود.

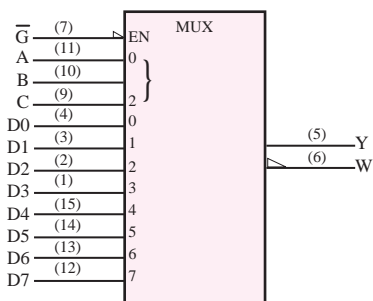
چنانچه این ورودی در یک حالت از پیش تعریف شده  $<0>$  یا  $<1>$  باشد. همه دروازه‌های AND را مسدود می‌سازد و از انتقال اطلاعات ورودی به خروجی مدار پیش‌گیری می‌کند. به عبارت دیگر، این ورودی مقدم بر ورودی‌های آدرس است. در شکل زیر مالتی پلکسر  $2 \rightarrow 1$  با ورودی تواناساز به همراه جدول صحت آن نشان داده شده است.

جدول صحت مالتی پلکسر  $2 \rightarrow 1$

$A_0$	EN	$f$
X	۱	۰
۰	۰	$D_0$
۱	۰	$D_1$

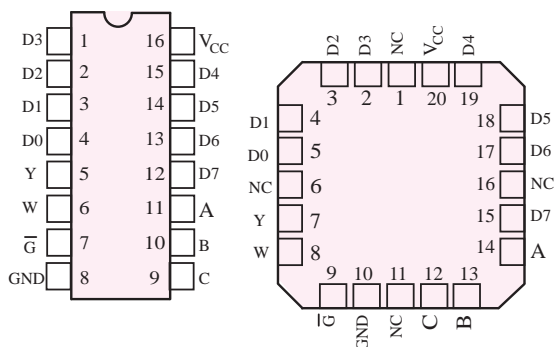


مدار یک مالتی پلکسر  $2 \rightarrow 1$

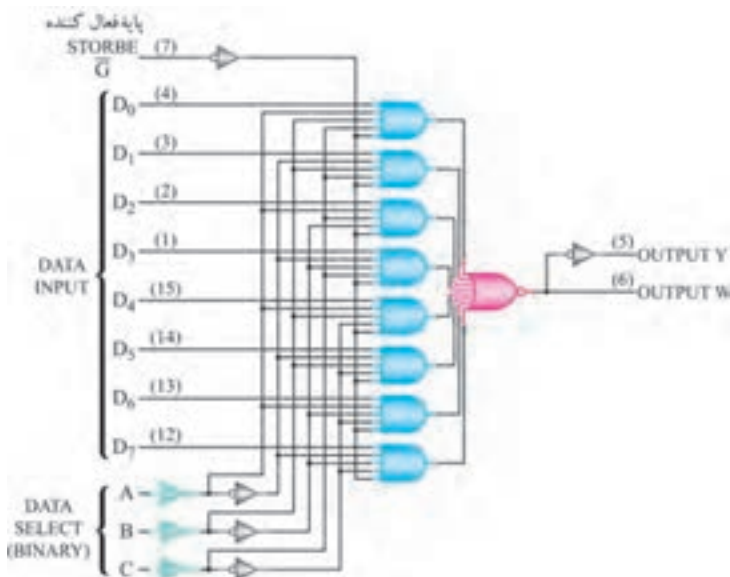


در شکل روبه‌رو مدار یک مالتی پلکسر ۸→۱ را که با شماره تجارتي ۷۴۱۵۱ به بازار عرضه می‌شود مشاهده می‌کنید. در شکل الف نماد استاندارد آی سی، در شکل ب نمای بالا و شماره پایه‌های آی سی و در شکل پ مدارهای داخلی آی سی نشان داده شده است.

الف) نماد استاندارد IEEE/ANSI



ب) نمای بالا و شماره پایه‌ها



پ) مدارهای داخلی

مدار مالتی پلکسر ۸→۱ تجارتي (۷۴۱۵۱) SN

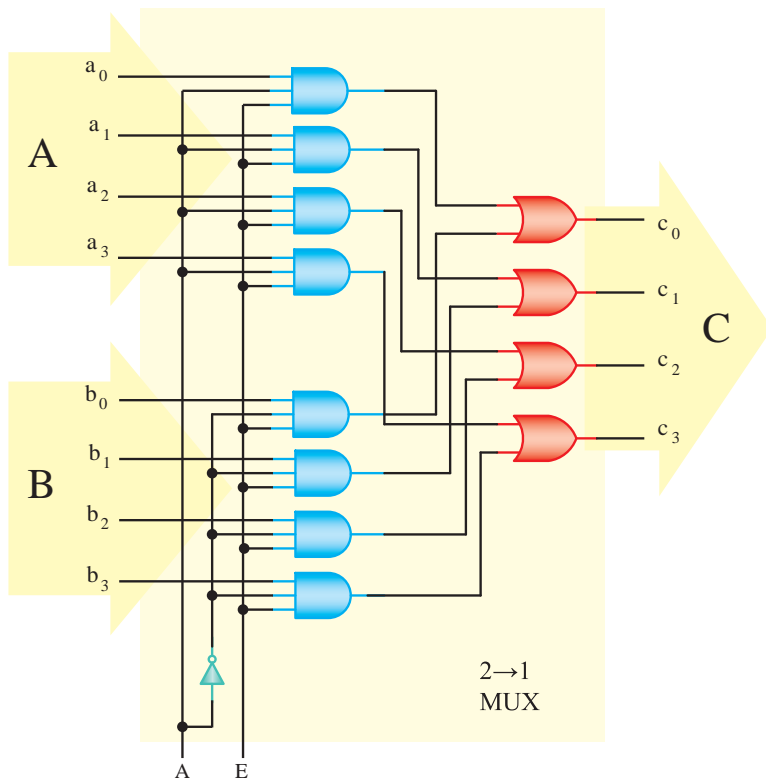


جدول صحت آی سی SN۷۴۱۴۷ را در جدول زیر مشاهده می کنید.

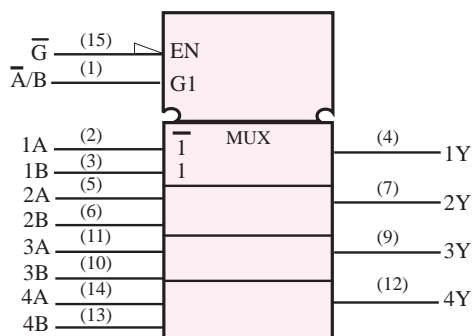
جدول صحت مالتی پلکسر SN۷۴۱۵۱

inputs				outputs	
SELECT			STROBE $\bar{G}$	Y	W
C	B	A			
X	X	X	H	L	H
L	L	L	L	D0	$\bar{D0}$
L	L	H	L	D1	$\bar{D1}$
L	H	L	L	D2	$\bar{D2}$
L	H	H	L	D3	$\bar{D3}$
H	L	L	L	D4	$\bar{D4}$
H	L	H	L	D5	$\bar{D5}$
H	H	L	L	D6	$\bar{D6}$
H	H	H	L	D7	$\bar{D7}$

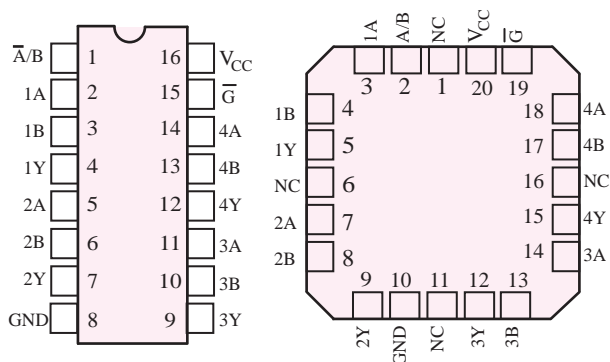
همان طور که در جدول بالا مشاهده می کنید. خروجی W معکوس خروجی Y است. در کامپیوترها مالتی پلکسرها به عنوان گذرگاه داده‌ها Data Bus ایفای نقش می کنند. در این موارد، معمولاً به مالتی پلکسرها چند بیتی نیاز داریم؛ مثلاً، اگر بخواهیم اطلاعات را توسط یک خط انتقال از دو سیستم B و A به سیستم C برسانیم و این اطلاعات در قالب کلمات چهار بیتی باشند، به یک مالتی پلکسر ۱→۲، ۴ بیتی نیاز داریم. امکان عرضه مالتی پلکسرها چند بیتی به صورت تراشه‌های تجارتي وجود ندارد دلیل این امر، افزایش تعداد پایه‌های ضروري آنهاست. لذا باید با استفاده از مالتی پلکسرها یک بیتی، مالتی پلکسر مورد نیاز خود را طراحی کنیم. در مورد مثال فوق، با استفاده از یک تراشه ۷۴۱۵۷ که شامل چهار مالتی پلکسر ۱→۲ یک بیتی است، یک مالتی پلکسر ۱→۲ چهار بیتی مطابق شکل الف، ب، پ، ت، و ث طراحی می کنیم: یعنی کلاً چهار مالتی پلکسر یک بیتی را به صورت موازی به کار می بریم.



الف) مدار اصلی یک Mux،  $2 \rightarrow 1$  چهار بیتی که با استفاده از چهار Mux،  $2 \rightarrow 1$  یک بیتی ساخته شده است.



ب) نماد استاندارد IEEE/ANSI

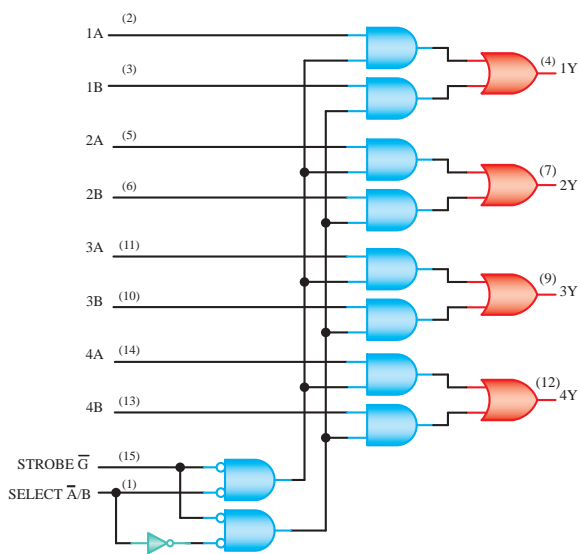


(پ) نمای بالا و شماره پایه‌ها

INPUTS				OUTPUT Y
STROBE $\overline{G}$	SELECT $\overline{A/B}$	A	B	"157, "LS157,"S157"
H	X	X	X	L
L	L	L	X	L
L	L	H	X	H
L	H	X	L	L
L	H	X	H	H

H = high level , L = Low level , X = irrelevant

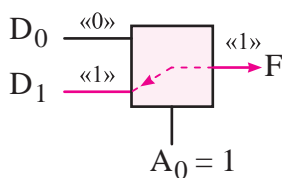
(ت) جدول صحت



### ث) مدارهای داخلی

(الف) چگونگی ساخت Mux جند پیتی؛ ب تا ث) مشخصات تراشه تجارتي SN۷۴۱۵۷

همان طور که گفته شد، افزایش تعداد پایه‌های ورودی مالتی پلکسر در طراحی مدارهای منطقی ضروری است. بنابراین در این بخش نحوه توسعه ورودی‌های یک مالتی پلکسر را با طراحی یک مالتی پلکسر  $4 \rightarrow 1$  به وسیله سه عدد مالتی پلکسر  $2 \rightarrow 1$  تشریح می‌کنیم. در شکل زیر بلوک دیاگرام یک مالتی پلکسر  $2 \rightarrow 1$  را همراه با جدول صحت آن در جدول بعد مشاهده می‌کنید. در این مدار به ازای آدرس  $A_0 = 1$  ورودی  $D_1$  به خط خروجی وصل شده است.



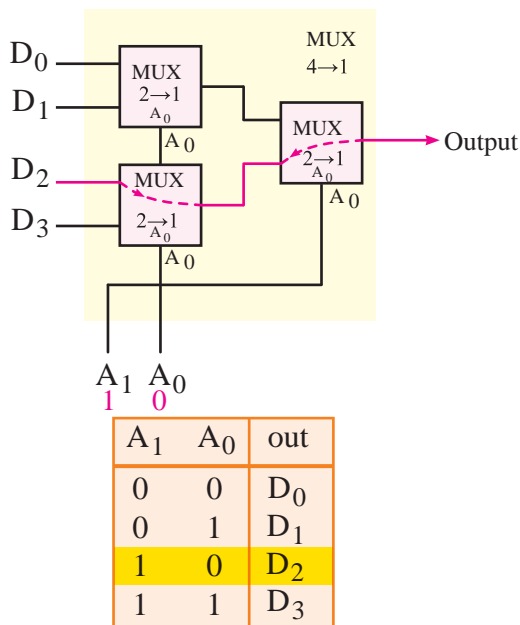
بلوک دیاگرام پلکسر  $2 \rightarrow 1$

جدول صحت مالتی پلکسر  $2 \rightarrow 1$

$A_0$	$f$
0	$D_0$
1	$D_1$

اگر سه عدد مالتی پلکسر  $2 \rightarrow 1$  را مطابق شکل زیر به یکدیگر اتصال دهیم یک مالتی پلکسر  $4 \rightarrow 1$  شکل می‌گیرد.

شکل صفحه بعد خط آدرس  $A_0$  مالتی پلکسرهای ورودی با یکدیگر موازی شده‌اند و خط آدرس کم ارزش تر  $A_1$  مالتی پلکسر  $4 \rightarrow 1$  را تشکیل می‌دهد و خط آدرس  $A_2$  مالتی پلکسر سوم (خروجی) با ارزش تر  $A_1$  مالتی پلکسر  $4 \rightarrow 1$  قرار می‌گیرد.



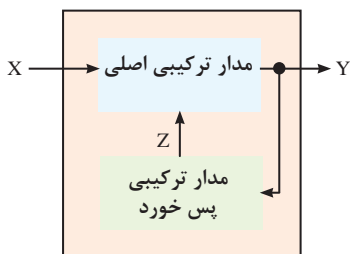
یک مالتی پلکسر ۴→۱ همراه جدول صحت آن

### مدارهای ترتیبی

به طور کلی مدارهای ترکیبی توانایی نگهداری حالت‌های ورودی را ندارند. ماشین‌های حسابگر و سیستم‌های کنترل، برای نگهداری اطلاعات و استفاده مکرر از آنها نیاز به حافظه دارند. مدارهای دارای حافظه را مدارهای ترتیبی می‌نامند. این مدارها قابلیت نگهداری (در حافظه نگه داشتن) ترتیب پیامدها را دارند. یک مدار ترتیبی شامل دو بخش اساسی به شرح زیر است:

الف) بخشی که بر اساس وضعیت قبلی خروجی‌های مدار، توابعی را ایجاد می‌کند. این بخش، مدار پس‌خورد نامیده می‌شود.

ب) بخشی که وضعیت مدار را بر اساس حالت ورودی‌های مدار و ورودی‌های دریافتی از مدار فیدبک مشخص می‌کند و وضعیت جدید سیستم را به وجود می‌آورد. این بخش را مدار اصلی می‌نامند. شکل روبه‌رو بلوک دیاگرام مدار ترتیبی را نشان می‌دهد.



بلوک دیاگرام یک مدار ترتیبی