مبانی دیجیتال

۱. مفهوم دیجیتال و سیستم اعداد
۲. ساختمان دروازه های منطقی پایه و ترکیبی
۳. جبر بول
۴. چند مدار ترکیبی کاربردی
۵. مدارهای ترتیبی فلیپ فلاپ ها
۶. شیفت رجیسترها و شمارنده ها
۷. مدارهای منطقی پیشرفته

مفهوم دیجیتال و سیستم اعداد

دروازه های منطقی پایه
سیستم های اعداد
مکمل های اعداد
تبدیل مبناهای اعداد به یکدیگر
جمع باینری
تفریق باینری
نقش کد در سیستم دیجیتال (BCD)

ساختمان دروازه های منطقی پایه و ترکیبی

ترازهای ولتاژ
دروازه های منطقی پایه
دروازه های منطقی ترکیبی
مشخصات ویژۀ دروازه های منطقی
استفاده از data book

جبر بول

جبر بول

قوانین دمورگان

ساده سازی توابع جبر بول

ساده سازی توابع با استفاده از نقشهٔ کارنو

افزایش ظرفیت ورود ی های درواز ه های منطقی

ساخت دروازه های منطقی مختلف با استفاده از گیت NAND

ساخت دروازه های منطقی مختلف با استفاده از گیت NOR

مقدمه ای بر مدارهای ترکیبی

چند مدار ترکیبی کاربردی

مدارهای ترکیبی

مدارهای ترکیبی با کاربردهای ویژه

انواع کدها

مبدل BCDبه 7.S

مدارهای رمز گشا

مدارهای رمز گذار

مدارهای متمرکز کننده یا تسهیم کننده

مدارهای ترتیبی- فلیپ فلاپ ها

فلیپ فلاپ ها

فلیپ فلاپ RS

تقسیم بندی فلیپ فلا پ ها براساس پالس ساعت

فلیپ فلا پ J-K

فلیپ فلاپ نوع D

فلیپ فلاپ نوع T

شیفت رجیسترها و شمارنده ها

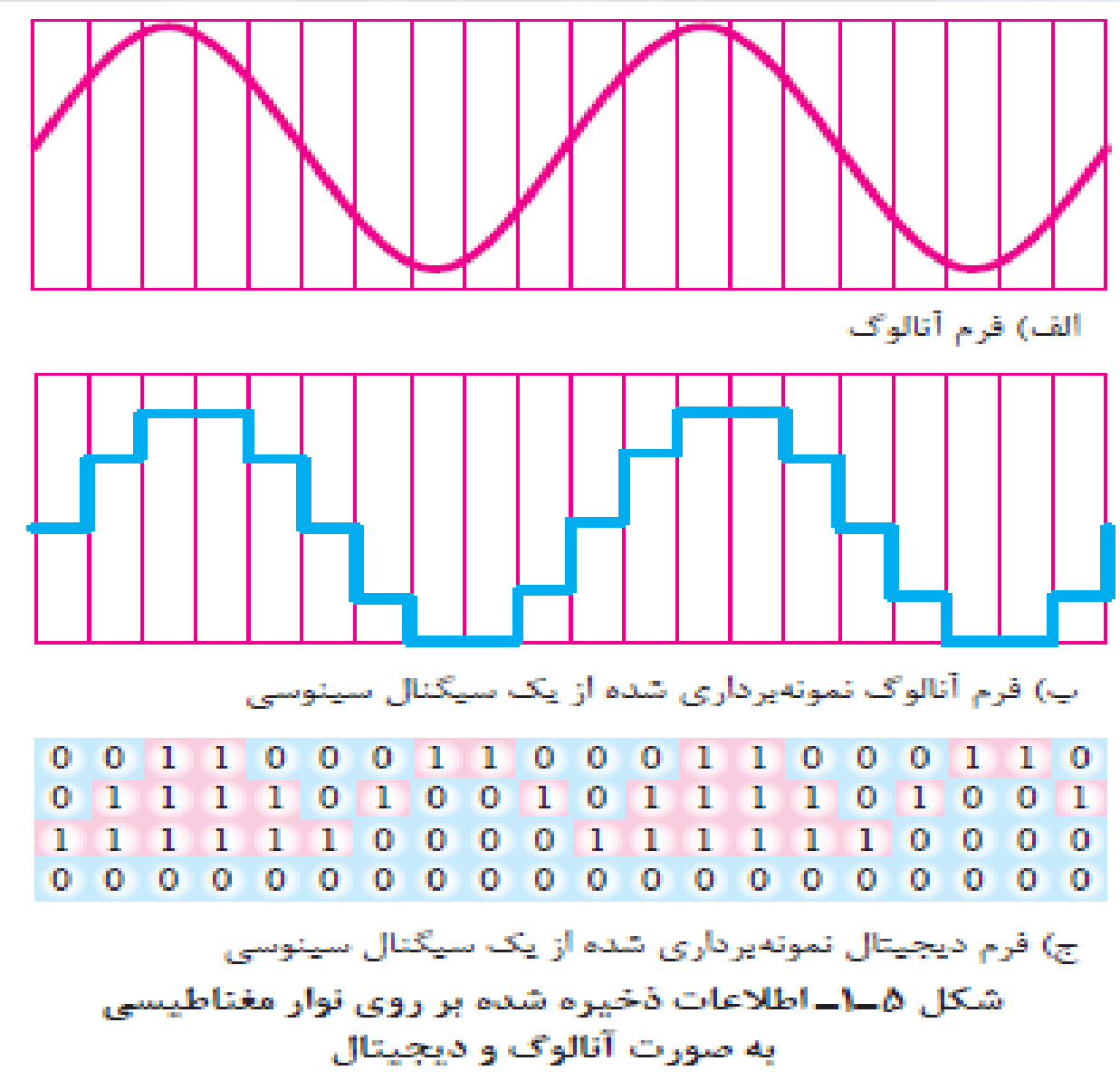
شیفت رجیسترها و شمارنده ها

شیفت رجیستر چپ رو راست رو

شمارنده ها

مفهوم دیجیتال

یک سیستم دیجیتال، سیستمی است که در آن اطلاعات به صورت رقمی ارائه و پردازش می شود.



الف) فرم آنالوگ

ب) فرم آنالوگ نمونه‌برداری شده از یک سیگنال سینوسی

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

ج) فرم دیجیتال نمونه‌برداری شده از یک سیگنال سینوسی

شکل ۵ـ۱ـ اطلاعات ذخیره شده بر روی نوار مغناطیسی به صورت آنالوگ و دیجیتال

مزایای سیستم های دیجیتال نسبت به آنالوگ

۱ قابلیت انعطاف پذیری بیشتر

۲ سرعت بالاتر

۳ دقت وضوح بیشتر

۴ بازیابی آسان اطلاعات

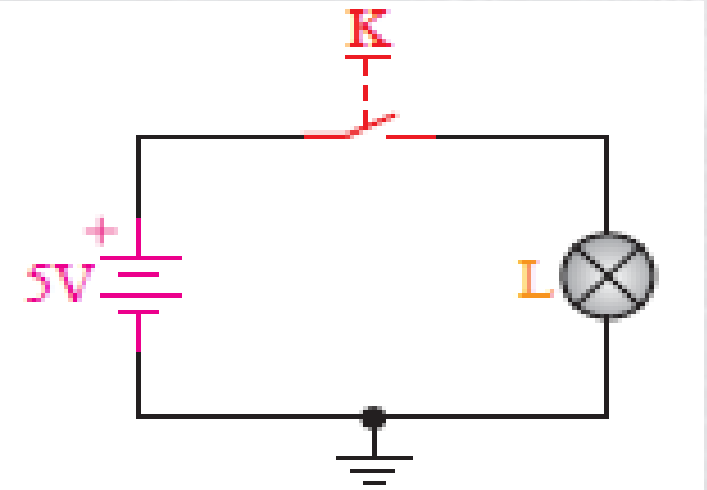
۵ ذخیرهٔ آسان اطلاعات

۶ کاهش ابعاد

مفهوم صفر و یک منطقی

اگر ولتاژ به حد معینی رسید یعنی یک است. یعنی لامپ روشن است و اگر ولتاژ در حد معینی پایین آمد و نزدیک به صفر شد مفهوم آن صفر است

هر سیستم دو وضعیتی برای نمایش دادن هر حالت از صفر و یک استفاده می شود،

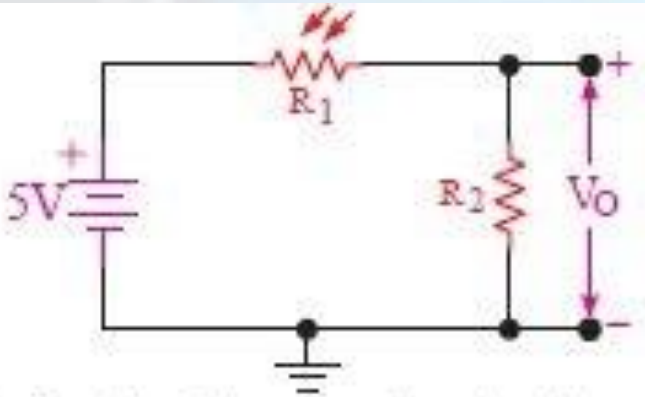


Off _Low _ خاموش → لامپ در حالت خاموش

On _High _ روشن → لامپ در حالت روشن

سطح ولتاژ صفر و یک منطقی

شکل ۷ـ۱ـ سطوح ولتاژ صفر و یک منطقی رایج



مدار مولد صفر و یک منطقی با استفاده از مقاومت

تابع نور

دروازه های منطقی پایه

دروازه های منطقی، اساس کار ماشین های حساب، کامپیوترها، مدارهای کنترل و ... است. به عبارت دیگر، یک کامپیوتر یا ماشین حساب و ... از تعدادی دروازه های منطقی تشکیل شده است.

در کامپیوتر یا ماشین حساب یک دروازهٔ منطقی در حقیقت یک مدار الکترونیکی است که یک یا چند ورودی و فقط یک خروجی دارد.

بلوک کلی یک دروازهٔ منطقی

در مدارهای غیرکامپیوتری، ساخت دروازه های منطقی با استفاده از کلیدها، شستی ها، رله ها و ... نیز امکان پذیر است اما به دلیل پایین بودن سرعت قطع و وصل این گونه قطعات، آنها با قطعات الکترونیکی قابل مقایسه نیستند. لذا در مواردی که سرعت قطع و وصل مطرح نیست و تعداد دروازه ها نیز بسیار محدود است، از این قطعات هم برای ساختن دروازه های منطقی استفاده می شود.

دروازهٔ AND (و)

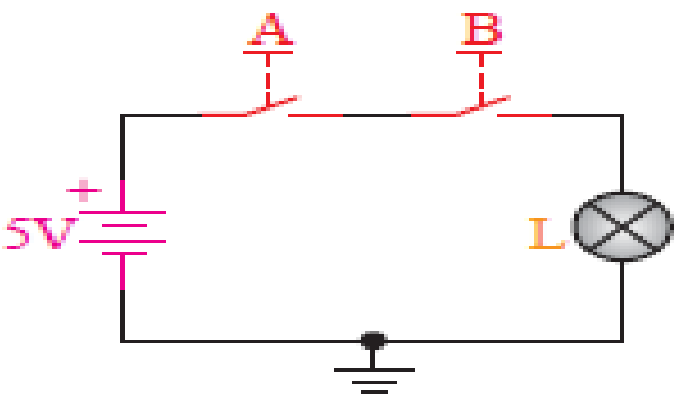
دروازه ای است که چنان چه همهٔ ، ANDدروازهٔ منطقی ورودی های آن در وضعیت یک منطقی باشند، خروجی آن نیز در وضعیت یک منطقی قرار می گیرد. در غیر این صورت حتی اگر یکی از ورودی های آن در وضعیت صفر منطقی باشد، خروجی این دروازه در وضعیت صفر منطقی خواهد بود.

مثال

فرد مراجعه کننده	داشتن دیپلم	داشتن گواهی نامهٔ مهارت در کار با کامپیوتر	وضعیت استخدام
خانم A	ندارد	ندارد	استخدام نمی‌شود
آقای B	دارد	ندارد	استخدام نمی‌شود
خانم C	ندارد	دارد	استخدام نمی‌شود
آقای D	دارد	دارد	استخدام می‌شود

دروازهٔ AND (و)

مدار شماتیک



جدول صحت

A	B	Y
۰	۰	۰
۰	۱	۰
۱	۰	۰
۱	۱	۱

وضعیت خروجی	وضعیت کلید B	وضعیت کلید A
۰ منطقی یا ۰ V	باز	باز
۰ منطقی یا ۰ V	بسته	باز
۰ منطقی یا ۰ V	باز	بسته
۱ منطقی یا ۵ V	بسته	بسته

دروازهٔ OR (یا)

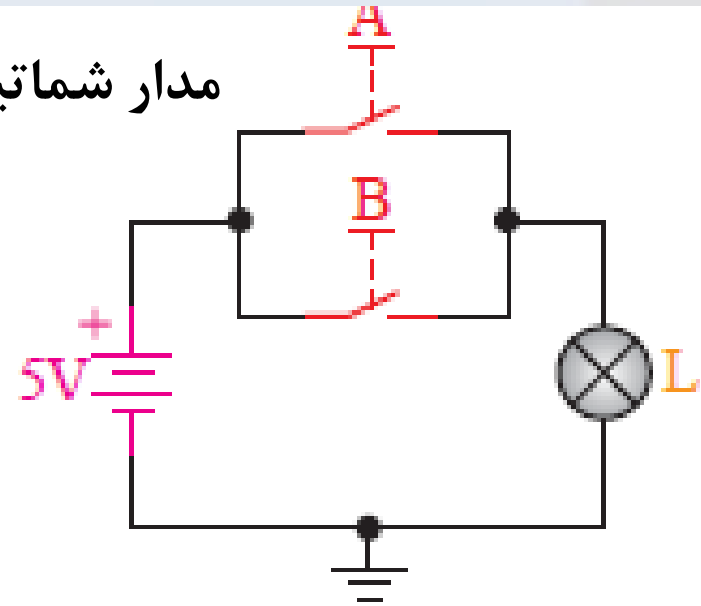
دروازه ای است که اگر دست کم ، ORدروازهٔ منطقی یکی از ورودی های آن در وضعیت یک منطقی باشد، خروجی آن نیز در وضعیت یک منطقی قرار م یگیرد و چنان چه همهٔ ورودی های آن در وضعیت صفر منطقی باشند، خروجی آن نیز در وضعیت صفر منطقی خواهد

مثال

فرد مراجعه کننده	داشتن دیپلم	داشتن گواهی‌نامهٔ مهارت در حسابداری	وضعیت استخدام
خانم A	ندارد	ندارد	استخدام نمی‌شود
آقای B	دارد	ندارد	استخدام می‌شود
خانم C	ندارد	دارد	استخدام می‌شود
آقای D	دارد	دارد	استخدام می‌شود

دروازهٔ OR (یا)

مدار شماتیک



جدول تغییرات دروازهٔ منطقی

وضعیت خروجی	وضعیت کلید B	وضعیت کلید A
۰ منطقی یا ۰ V	باز	باز
۱ منطقی یا ۵ V	بسته	باز
۱منطقی یا ۵ V	باز	بسته
۱ منطقی یا ۵ V	بسته	بسته

جدول صحت

A	B	Y
۰	۰	۰
۰	۱	۱
۱	۰	۱
۱	۱	۱

دروازهٔ NOT (نه)

دروازه ای است که اولاً یک ورودی دارد؛ ، NOTدروازهٔ

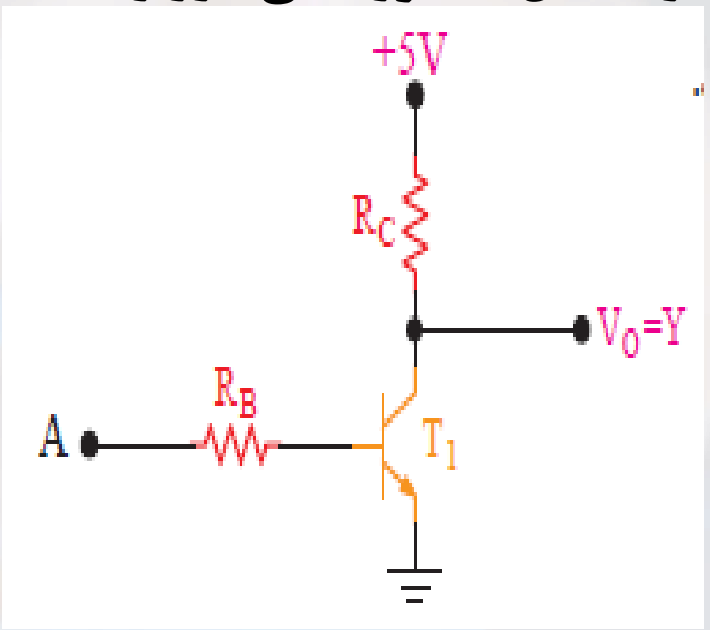
ثانیاً خروجی آن زمانی در وضعیت یک منطقی قرار

م یگیرد که ورودی آن در وضعیت صفر منطقی باشد.

مثال

فرد مراجعه کننده	داشتن سابقهٔ کیفری	وضعیت استخدام
آقای A	ندارد	استخدام می‌شود
آقای B	دارد	استخدام نمی‌شود

مدار معادل الکترونیکی دروازهٔ منطقی



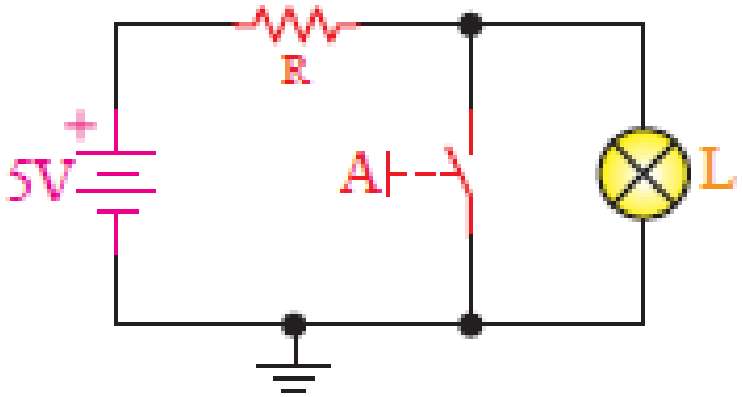
جدول تغییرات دروازهٔ

ولتاژ خروجی (Y)	ولتاژ ورودی (A)
= ۵ V	۰ V
= ۰ V	۵ V

جدول صحت دروازهٔ

A	Y
۰	۱
۱	۰

مدار شماتیک



وضعیت لامپ	وضعیت کلید
روشن	باز
خاموش	بسته

دروازهٔ NOT (نه)

دروازه ای است که اولاً یک ورودی دارد؛ ، NOTدروازهٔ

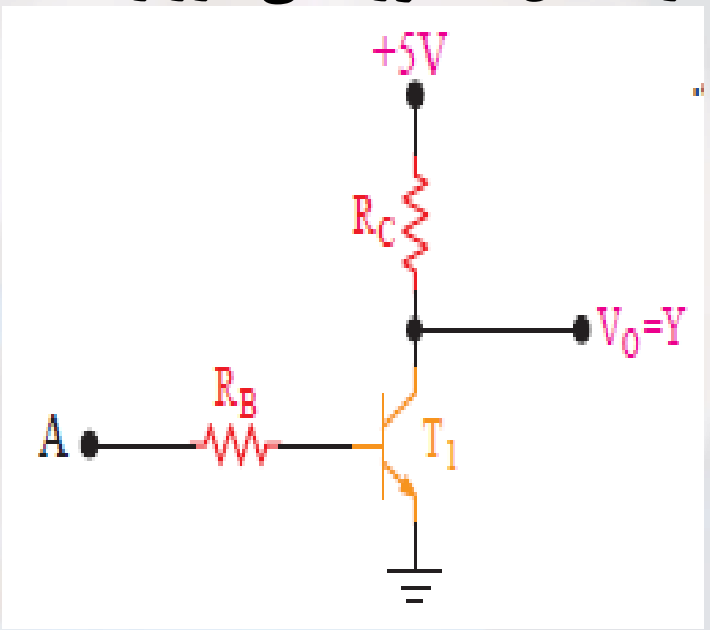
ثانیاً خروجی آن زمانی در وضعیت یک منطقی قرار

م یگیرد که ورودی آن در وضعیت صفر منطقی باشد.

مثال

فرد مراجعه کننده	داشتن سابقهٔ کیفری	وضعیت استخدام
آقای A	ندارد	استخدام می‌شود
آقای B	دارد	استخدام نمی‌شود

مدار معادل الکترونیکی دروازهٔ منطقی



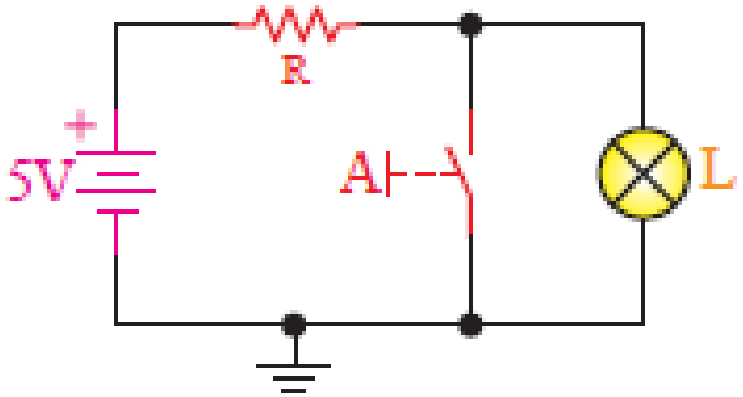
جدول تغییرات دروازهٔ

ولتاژ خروجی (Y)	ولتاژ ورودی (A)
= ۵ V	۰ V
= ۰ V	۵ V

جدول صحت دروازهٔ

A	Y
۰	۱
۱	۰

مدار شماتیک



وضعیت لامپ	وضعیت کلید
روشن	باز
خاموش	بسته

سیستم های اعداد

یکی از مبناهایی که از زمان قدیم تا کنون مورد استفاده قرار گرفته است، مبنای ۱۰ (ده دهی) است.

Decimal – سیستم د هدهی

سیستم اعداد ده دهی از ده علامت ۰و ۱و ۲و... ۹ تشکیل شد هاند. برای شمارش از صفر تا ۹ از این علامتها استفاده میکنیم و برای نشان دادن اعداد بزرگ تر از ۹، این علامتها را طبق قواعد خاصی با یک دیگر ترکیب میکنیم

موقعیت مکانی هر عدد

$$۳۲۹۶ = ۳۰۰۰ + ۲۰۰ + ۹۰ + ۶ =$$

$$۳\times۱۰^{۳} + ۲\times۱۰^{۲} + ۹\times۱۰^{۱} + ۶\times۱۰^{۰}$$

$$N = a_n\times۱۰^{n} + a_{n-1}\times۱۰^{n-1} + \dots + a_۲\times۱۰^{۲} + a_۱\times۱۰^{۱} + a_۰\times۱۰^{۰}$$

$$۴۵۵۳۱ = ۴\times۱۰^{۴} + ۵\times۱۰^{۳} + ۵\times۱۰^{۲} + ۳\times۱۰^{۱} + ۱\times۱۰^{۰}$$

ده هزارگان — هزارگان — صدگان — دهگان — یکان

سیستم دودویی – Binary

در سیستم دودویی علایم به کار رفته ۰ و ۱ هستند. برای شمارش صفر و یک از این علامت ها استفاده میکنیم و برای نمایش دادن اعداد بزرگ تر از یک، این دو علامت را طبق قواعد خاصی پشت سر هم قرار می دهیم.

موقعیت مکانی هر عدد

$$N=a_n\times 2^n+a_{n-1}\times 2^{n-1}+\dots+a_2\times 2^2+a_1\times 2^1+a_0\times 2^0$$

$$(1\,0\,0\,1\,1)_2 = 1\times 2^4+0\times 2^3+0\times 2^2+1\times 2^1+1\times 2^0$$

یکی — ۲تایی — ۴تایی — ۸تایی — ۱۶تایی

برای تبدیل کردن اعداد اعشاری به باینری، می توانیم از تقسیمات متوالی عدد اعشاری به عدد دو استفاده کنیم. برای مثال عدد اعشاری ۸۷ را به عدد باینری تبدیل میکنیم.

تقسیمات را تا جایی ادامه م یدهیم تا آخرین خارج قسمت یک شود و سپس در سمت چپ آخرین خارج قسمت را م ینویسیم و به ترتیب باقیماند ه های به دست آمده را د جلوی آن قرار می دهیم

اعداد اعشاری و معادل باینری آن

باینری	اعشاری ده دهی
۰	۰
۱	۱
۱۰	۲
۱۱	۳
۱۰۰	۴
۱۰۱	۵
۱۱۰	۶
۱۱۱	۷
۱۰۰۰	۸
۱۰۰۱	۹
۱۰۱۰	۱۰
۱۰۱۱	۱۱
۱۱۰۰	۱۲
۱۱۰۱	۱۳
۱۱۱۰	۱۴
۱۱۱۱	۱۵

واحد بزرگ تر از بایت، کیلوبایت معادل ۲ به توان ۱۰ بایت یا ۱۰۲۴ بایت و مگابایت معادل ۲ بتوان ۲۰ بایت یا ۱۰۲۴ کیلوبایت است.

در سیستم دودویی هر کیلو بایت معادل 2^{10} بایت اســـت و با واحد کیلو در بقیهٔ کمیت‌هایی که تاکنون شناخته‌ایم متفاوت است :

به همین ترتیب داریم:

۱کیلوبایت=۱KB=2^{10} B

۱مگابایت=۱MB=2^{10}KB=2^{20} B

۱گیگابایت=۱GB=2^{10}MB=2^{20}KB=2^{30} B

۱ترا بایت=۱TB=2^{10}GB=2^{20}MB=2^{30}KB=2^{40} B

سیستم هشت تایی- Octal

در سیستم مبنا، عدد ۸ و تعداد علامت ها هشت رقم به صورت (۰و ۱و ۲و...و ۷) است. برای نمایش دادن اعداد از صفر تا هفت، از این علامت ها استفاده می شود. برای اعداد بزرگ تر از هفت، این علامت ها را طبق قواعد خاصی پشت سر هم قرار م یدهیم.

موقعیت مکانی هر عدد

$$N = a_n \times 8^n + a_{n-1} \times 8^{n-1} + \dots + a_2 \times 8^2 + a_1 \times 8^1 + a_0 \times 8^0$$

$(۵۲۳۶)_8 =$

$۵\times۸^۳+۲\times۸^۲+۳\times۸^۱+۶\times۸^۰ =$

$۵\times۵۱۲+۲\times۶۴+۳\times۸+۶\times۱ =$

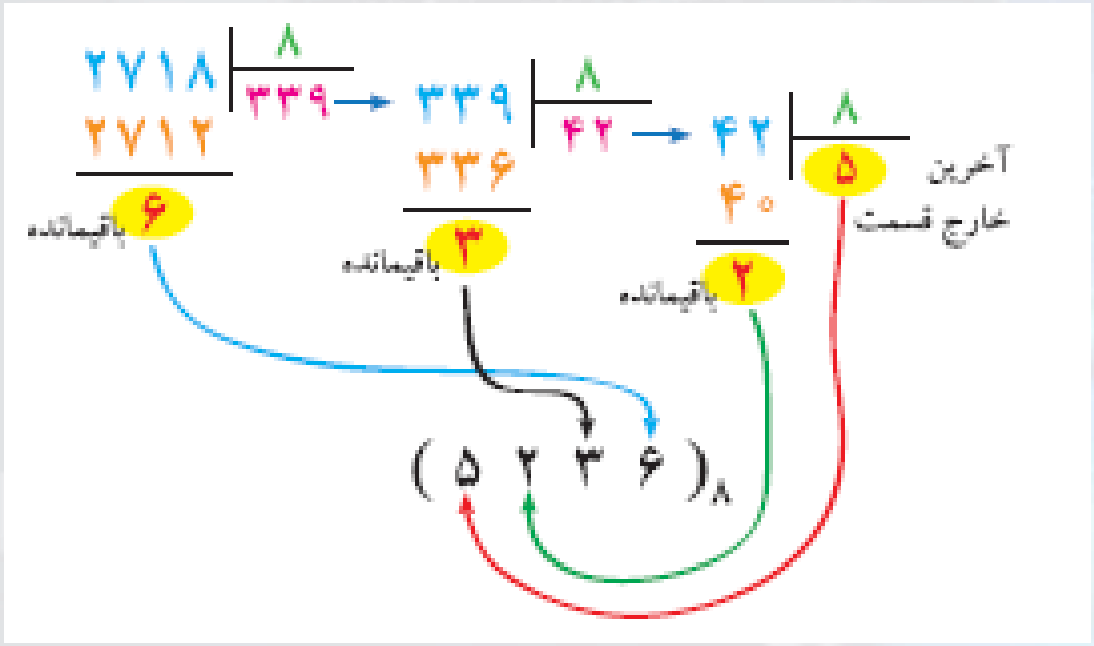
$۲۵۶۰+۱۲۸+۲۴+۶ = (۲۷۱۸)_{10}$

$(۳۰۴۵)_8 = ۳\times۸^۳+۰\times۸^۲+۴\times۸^۱+۵\times۸^۰ =$

$۳\times۵۱۲+۰\times۶۴+۴\times۸+۵\times۱ =$

$۱۵۳۶+۰+۳۲+۵ = (۱۵۷۳)_{10}$

برای تبدیل کردن اعداد اعشاری به اکتال، از تقسیم های متوالی عدد اعشاری به عدد ۸ استفاده می کنیم و همان قواعد خاصی که ا اشاره کردیم



تقسیمات را تا جایی ادامه م یدهیم که خارج قسمت با عدد ۷ مساوی یا کوچکتر شود. مشابه سیستم باینری از سمت چپ شروع به نوشتن عدد می کنیم. به این ترتیب که آخرین خارج قسمت را سمت چپ نوشته و به ترتیب باقیمانده را در جلوی آن م ینویسیم تا به اولین باقی ماندۀ تقسیم برسیم.

سیستم شانزده تایی – Hexa decimal

در این سیستم ۱۶ علامت شامل ۰

به کار می رود. در این سیستم برای F و E، D، C، A،B

نمایش عددهای بیشتر از ۹ و کمتر از ۱۶ باید از یک

علامت استفاده کرد و نمی توان مثلاً عدد ۱۰ را به همین

صورت نشان داد چون یک عدد دو رقمی است که هم

صفر و هم یک دارد و با صفر و یک اصلی اشتباه می شود.

به همین دلیل از حروف استفاده می شود که:

$$A=10,\ B=11,\ C=12,\ D=13,\ E=14,\ F=15$$

موقعیت مکانی هر عدد

$$N=a_n\times16^n+a_{n-1}\times16^{n-1}+\dots+a_1\times16^1+a_0\times16^0$$

$(A14E)_{16}$

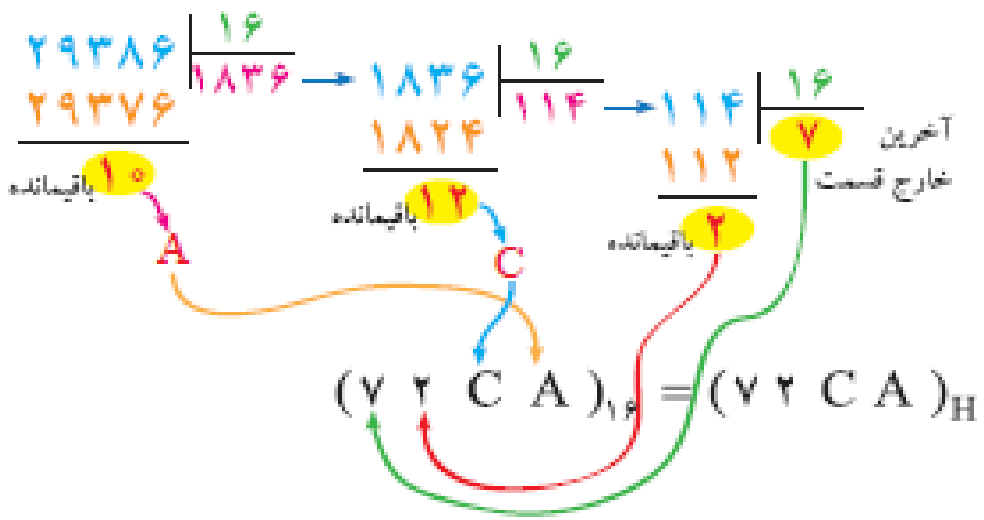
$$N=A\times16^3+1\times16^2+4\times16^1+E\times16^0=$$

$$N=(10)\times4096+1\times256+4\times16+(14)\times1=$$

$$40960+256+64+14=(41294)_{10}$$

برای تبدیل کردن اعداد اعشاری به اعداد هگزادسی مال از تقسیم های متوالی عدد اعشاری به عدد ۱۶ استفاده میکنیم. هنگام تقسیم کردن توجه داشته باشید که تا Aاگر باقیمانده بین ۱۰ تا ۱۵ باشد، باید از حروف F استفاده کنید.

$(29386)_{10}$

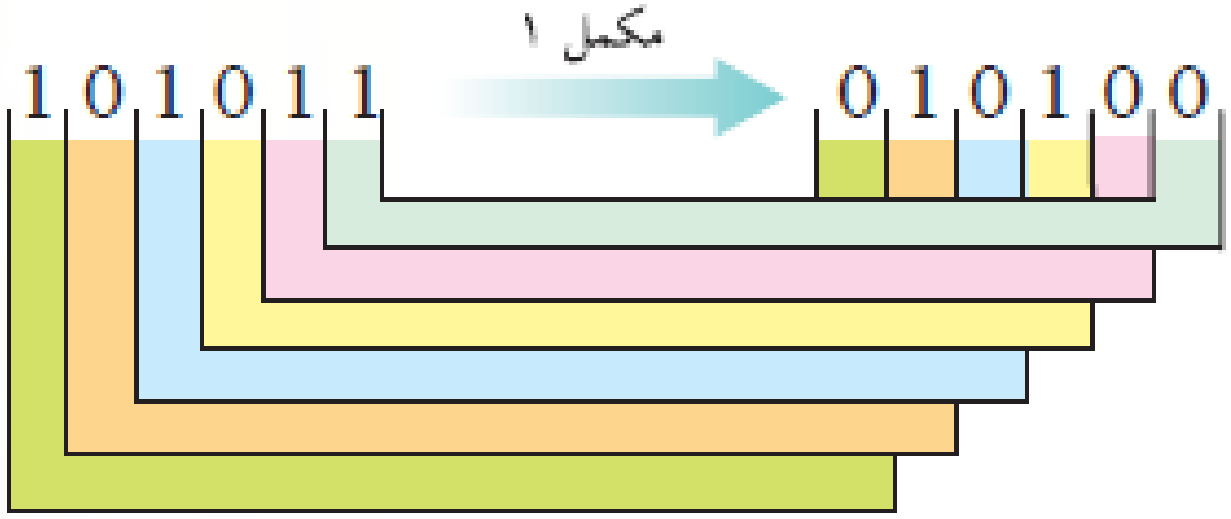


مکمل های اعداد

مکملها یا متممها در کامپیوترهای دیجیتال برای ساده کردن عمل تفریق و یا عملیات منطقی به کار میروند. در هر مبنایی دو نوع مکمل برای هر سیستم وجود دارد: یکی مکمل مبنا یا پایه و دیگری مکمل مبنا منهای یک یا پایهٔکاهش یافته است. در سیستم دودویی چون مبنا ۲ است مکمل ۲ را داریم و مکمل کاهش یافتهٔ پایه که آن را مکمل ۱ می نامیم.

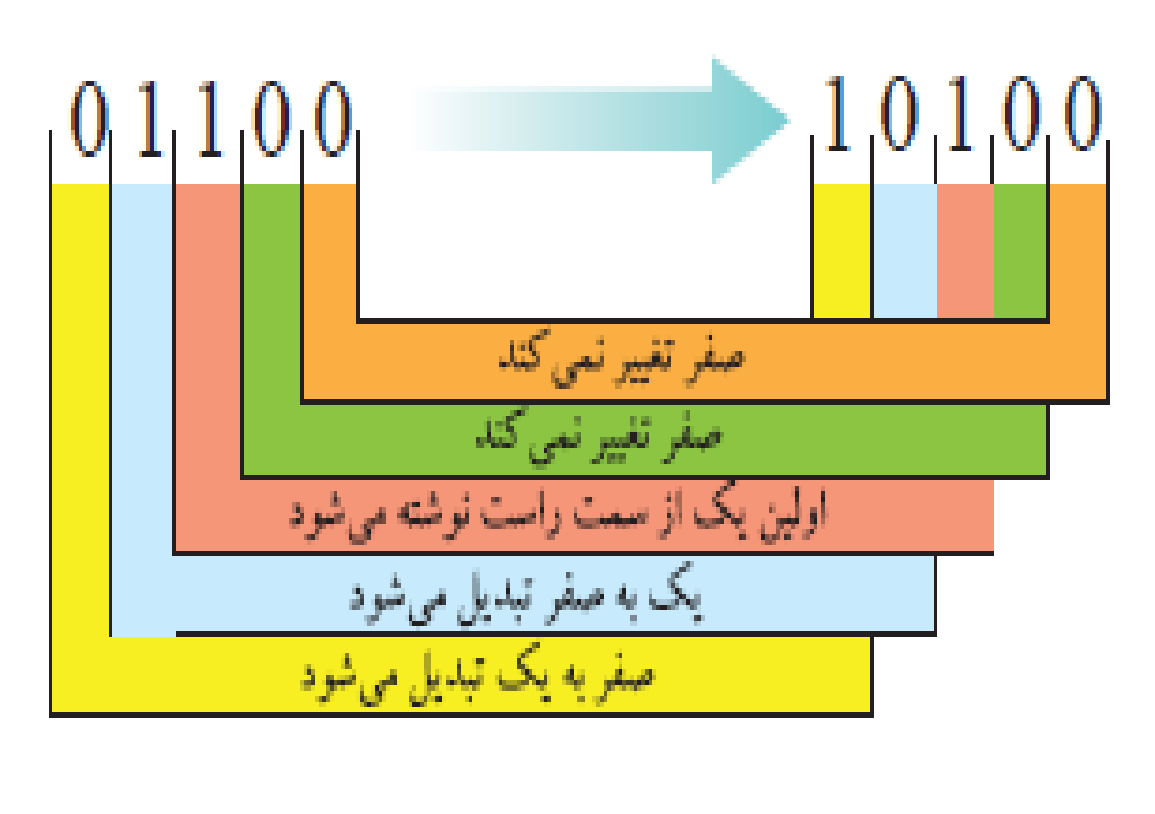
مکمل ۱

برای بدست آوردن مکمل ۱ در هر عدد دودویی کافی است صفرها را یک و یک های آن را به صفر تبدیل کنیم. مثلاً برای عدد باینری ۰۱۱۱۰۱ مکمل یا متمم یک آن به صورت ۱۰۰۰۱۰ خواهدشد.



مکمل ۲

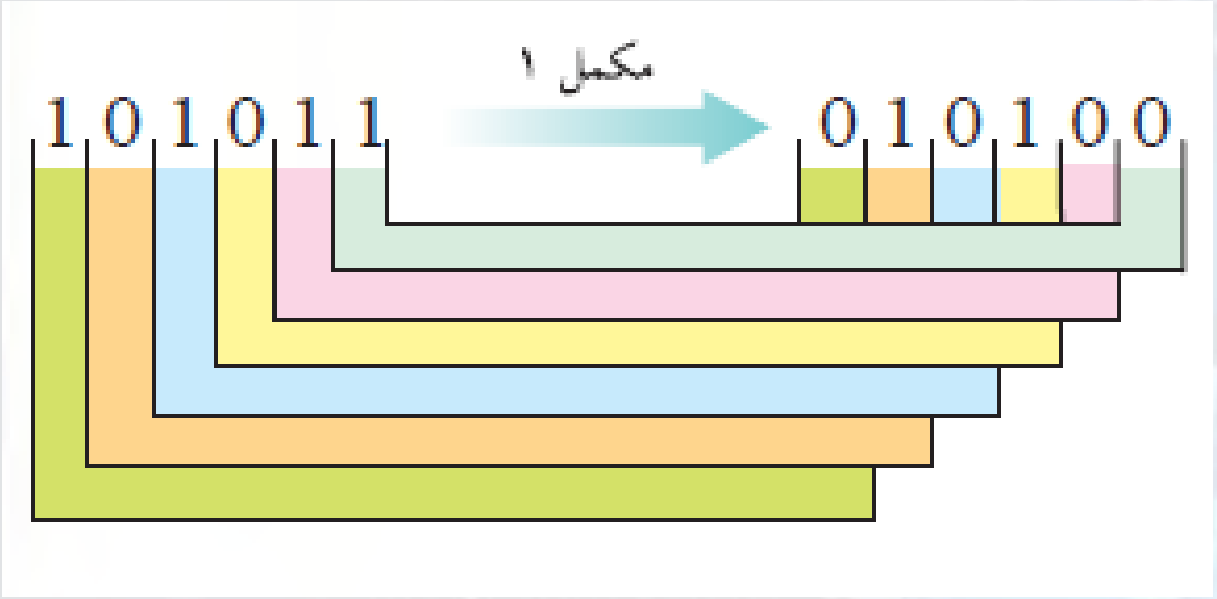
در سیستم دودویی مکمل ۲ براساس مبنا یا پایهٔآن تعریف شده است. برای به دست آوردن مکمل ۲ در هر عدد باینری به صورت زیر عمل می کنیم. ابتدا اولین ارقام صفر را از سمت راست می نویسیم به اولین یک که رسیدیم آن را نوشته، سپس بقیهٔ بیت ها را متمم می کنیم یعنی یک ها را به صفر و صفرها را به یک تبدیل می کنیم. مثلاً برای عدد باینری ۰۱۱۰۰ از سمت راست، ابتدا دو صفر آن را نوشته و اولین یک از سمت راست را نیز می نویسیم سپس دومین یک از سمت راست را صفر م یکنیم و بعد صفر را یک کرده و آن را می نویسیم .

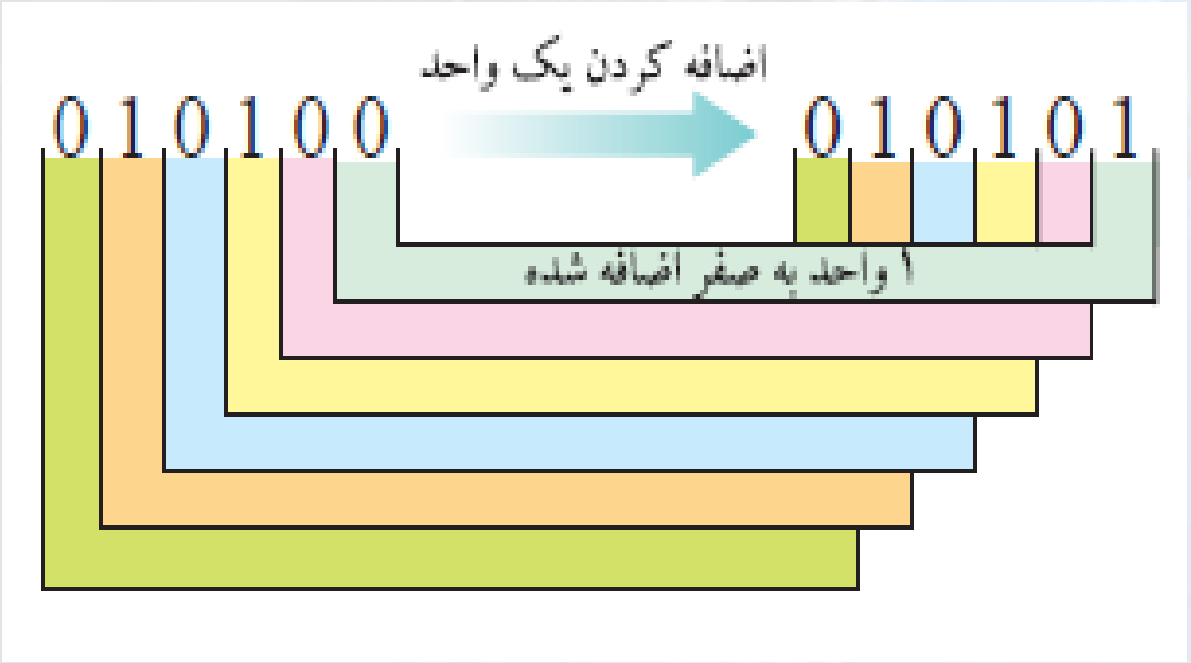


مکمل ۲

روش دیگری نیز برای مکمل ۲ وجود دارد به این ترتیب که ابتدا مکمل ۱ عدد باینری را می نویسیم، سپس یک واحد به عدد به دست آمده اضافه میکنیم.

به طور مثال مکمل ۲ عدد باینری ۱۰۱۰۱۱ را از روش دوم به دست می آوریم





تبدیل مبناهای اعداد به یکدیگر

وقتی که ما بی شتر از یک سیستم عددی داریم،

تبدیل اعداد از یک سیستم به سیستم دیگر بسیار مهم است. برای ما آسا نتر است که با اعداد دسی مال سروکار داشته باشیم ولی در سیست مهای دیجیتال اعداد دودویی بیشتر به کار میرود.

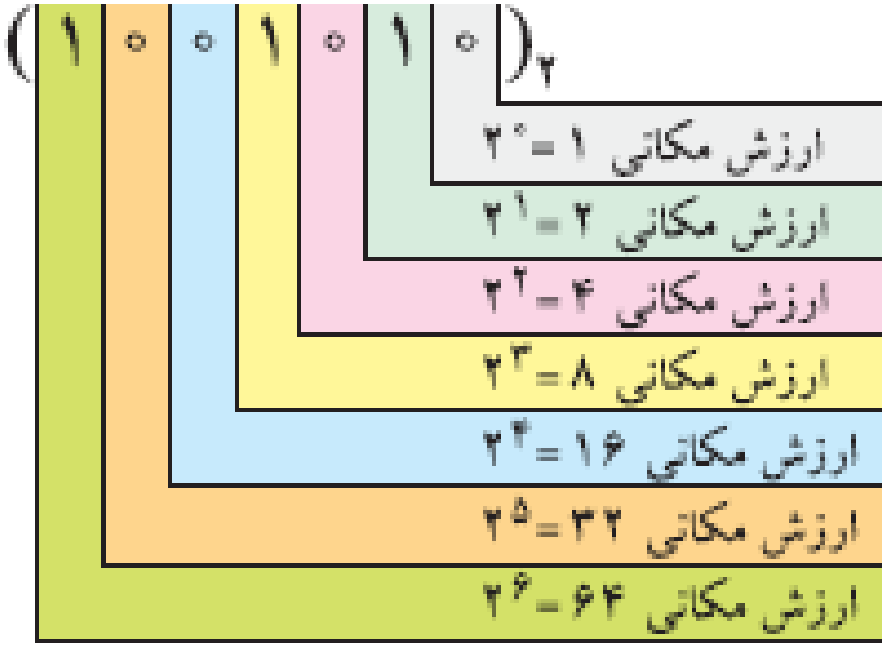
از طرفی ما هم به اعداد دسی مال احتیاج داریم و هم به اعداد دودویی، زیرا ماشین اعداد دودویی را می شناسد درصورتی که روی نمایشگر باید اعداد ده دهی ظاهر شود.

درنتیجه همواره در سیستم های دیجیتالی تبدیل اعداد دسی مال به اعداد دودویی در مورد اطلاعات ورودی و برعکس تبدیل اعداد دودویی به اعداد دسی مال در مورد اطلاعات خروجی مورد نیاز است.

همچنین استفاده از سیستم اعداد در مبنای اکتال و هگزا دسی مال که به صورت توا نهایی از ۲ نوشته میشوند، در ساده کردن این تبدیلات بسیار مؤثر هستند.

تبدیل مبنای ۲ به ۱۰

برای تبدیل اعداد دودویی به دسی مال، ابتدا ارزش مکانی بیت های عدد باینری را مشخص می کنیم، سپس باتوجه به مقدار بیت در آن ارزش مکانی آنها را با هم جمع می کنیم. به عنوان مثال ارزش مکانی عدد زیر را تعیین میکنیم.



۱×۶۴+۰×۳۲+۰×۱۶+۱×۸+۰×۴+۱×۲+۰×۱=

۶۴+۰+۰+۸+۰+۲+۰=۷۴

(۱۰۰۱۰۱۰)۲=۷۴

تبدیل مبنای ۲ به ۸

برای این که اعداد را از مبنای باینری به مبنای اکتال تبدیل کنیم، ابتدا باید عدد باینری را به مبنای دسی مال برده و سپس با تقسیم های متوالی بر ۸ به مبنای اکتال تبدیل کنیم.

$(۱۱۱۰۰۱۱)_۲=$

$۱\times۲^۶+۱\times۲^۵+۱\times۲^۴+۰\times۲^۳+۰\times۲^۲+۱\times۲^۱+۱\times۲^۰=$

$۱\times۶۴+۱\times۳۲+۱\times۱۶+۰\times۸+۰\times۴+۱\times۲+۱\times۱=$

$۶۴+۳۲+۱۶+۲+۱=۱۱۵$

۱۱۵ | ۸ → ۱۴ | ۱۱۲ | باقیمانده ۳

۱۴ | ۸ → ۱ | ۸ | باقیمانده ۶

آخرین خارج قسمت

$۱۱۵=(۱۶۳)_۸=(۱۱۱۰۰۱۱)_۲$

مرحله اول تبدیل به مبنای دسی مال

$۱۰۰۱۱۰=$

$۱\times۳۲+۰\times۱۶+۰\times۸+۱\times۴+۱\times۲+۰\times۱=$

$۳۲+۴+۲=۳۸$

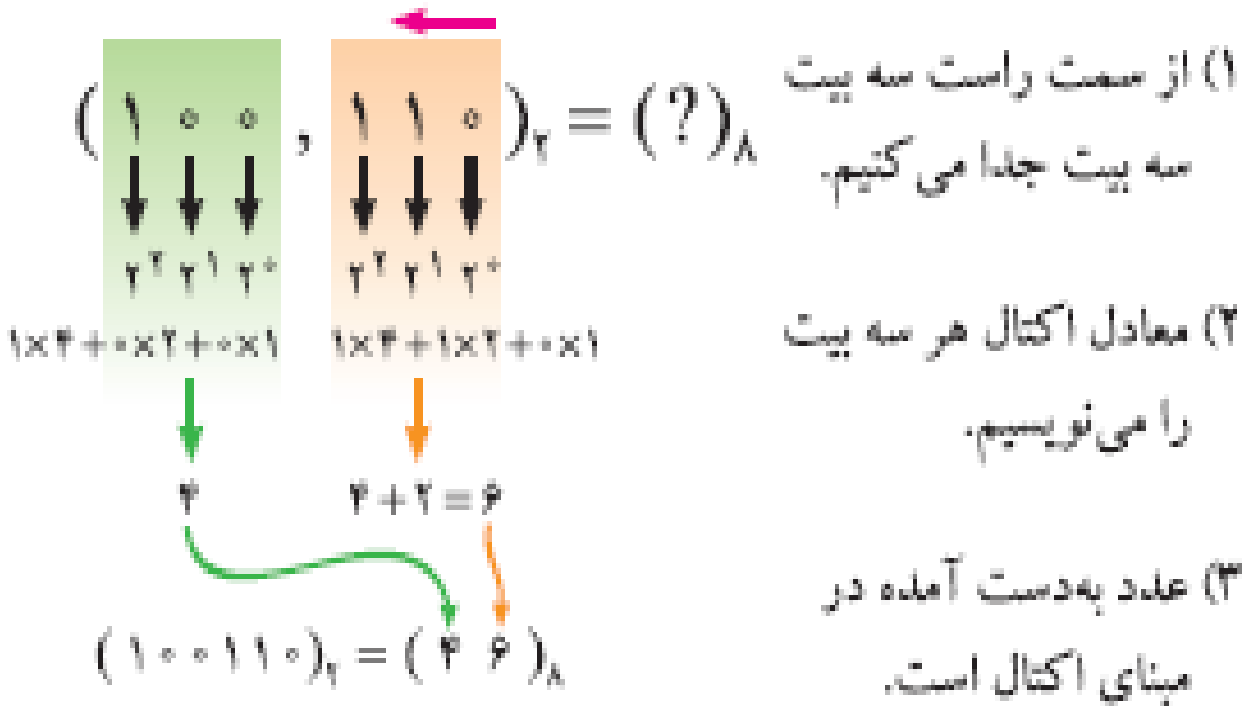
مرحله دوم تبدیل عدد اعشاری به مبنای اکتال:

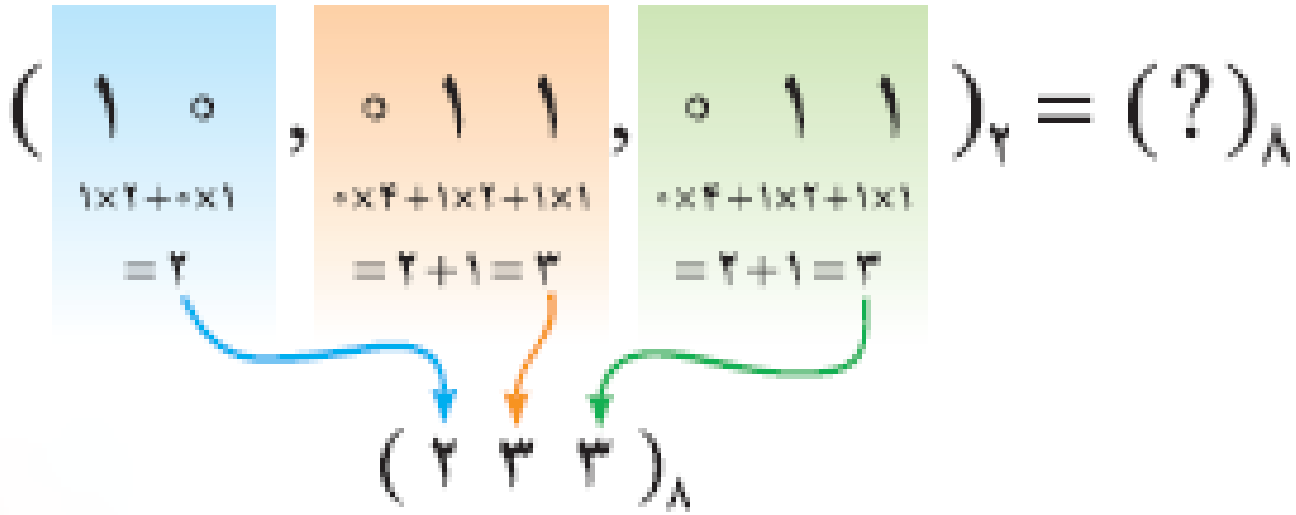
۳۸ | ۸ → ۴ | ۳۲ | باقیمانده ۶

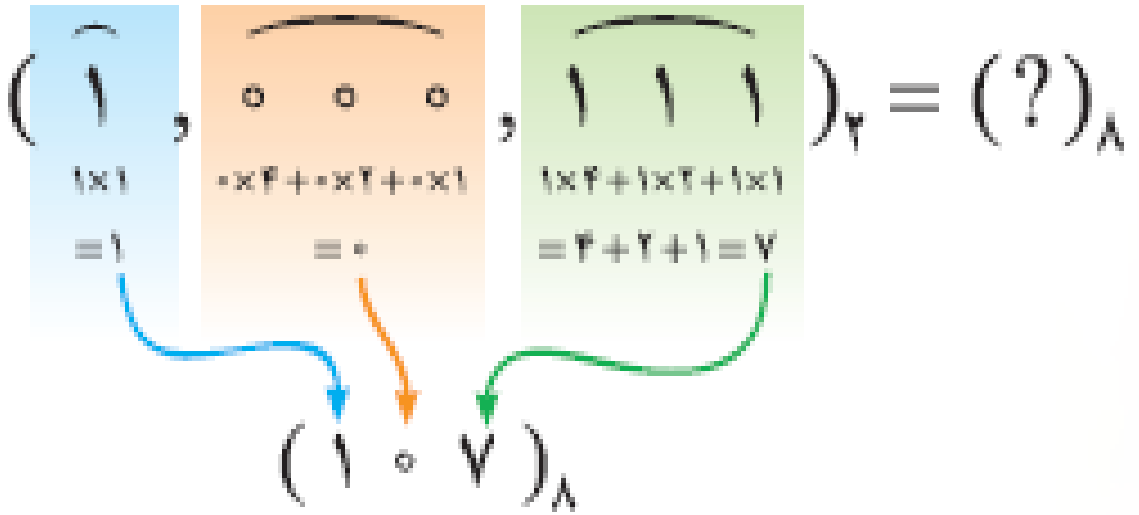
خارج قسمت

تبدیل مبنای ۲ به ۸

روش ساده تری نیز برای این تبدیل وجود دارد که سرعت کار را بالاتر میبرد. می توان عدد باینری را از سمت راست سه بیت سه بیت جدا کنیم و معادل هر قسمت آن را به صورت اکتال بنویسیم، به طور مثال:







تبدیل مبنای ۸ به ۲

برای تبدیل اعداد در مبنای اکتال به مبنای دودویی،
ابتدا باید عدد در سیستم اکتال را به
سیستم دسی مال برده، سپس با تقسیم های متوالی
بر ۲ به مبنای دودویی تبدیل کنیم.:

مرحلهٔ اول تبدیل به مبنای دسی مال:

$$(۷۵۲)_۸ =$$

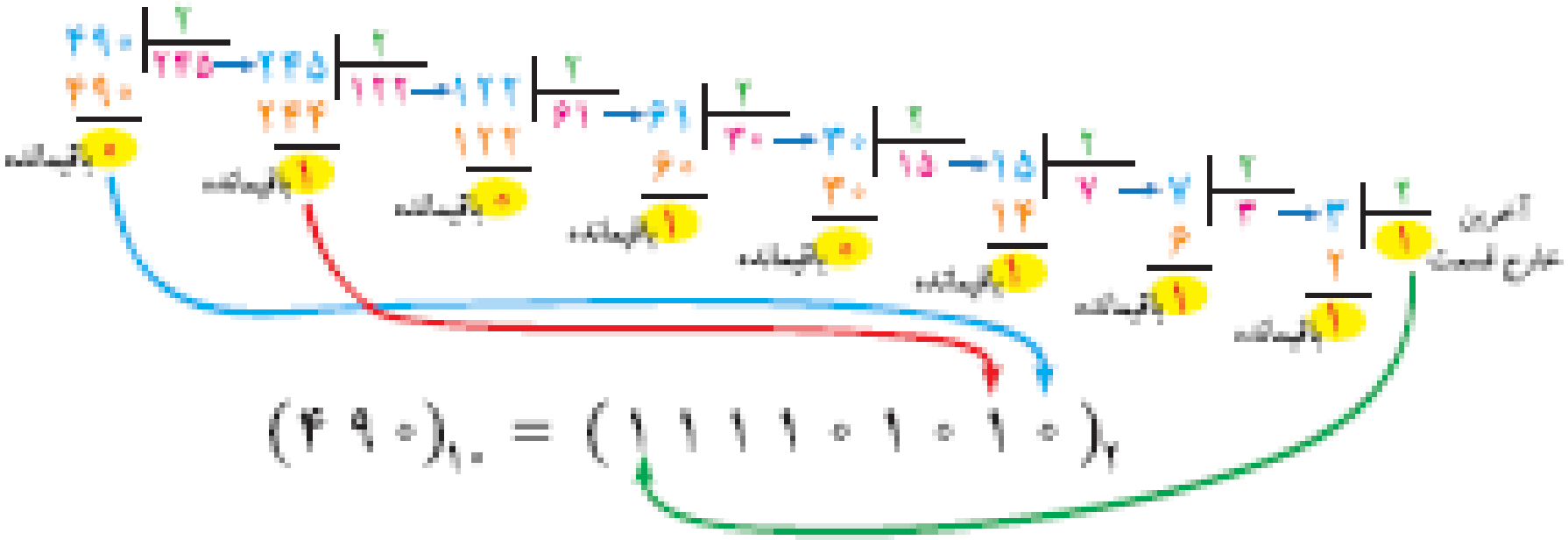
$$۷\times۸^۲+۵\times۸^۱+۲\times۸^۰=$$

$$۷\times۶۴+۵\times۸+۲\times۱=$$

$$۴۴۸+۴۰+۲=۴۹۰$$

$$(۷۵۲)_۸=(۴۹۰)_{۱۰}$$

مرحلهٔ دوم تبدیل عدد اعشاری به دست آمده به مبنای باینری:



$$(۴۹۰)_{۱۰} = (۱۱۱۱۰۱۰۱۰)_۲$$

تبدیل مبنای ۸ به ۲

مرحلهٔ اول تبدیل به مبنای دسی مال:

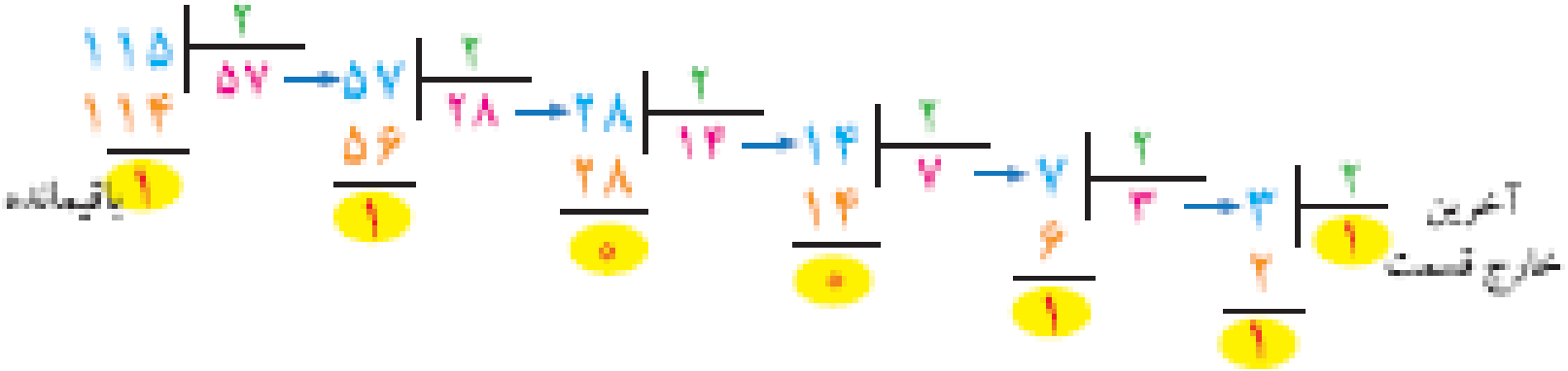
$$(۱۶۳)_۸=$$

$$۱\times۸^۲+۶\times۸^۱+۳\times۸^۰=$$

$$۱\times۶۴+۶\times۸+۳\times۱=$$

$$۶۴+۴۸+۳=(۱۱۵)_{۱۰}$$

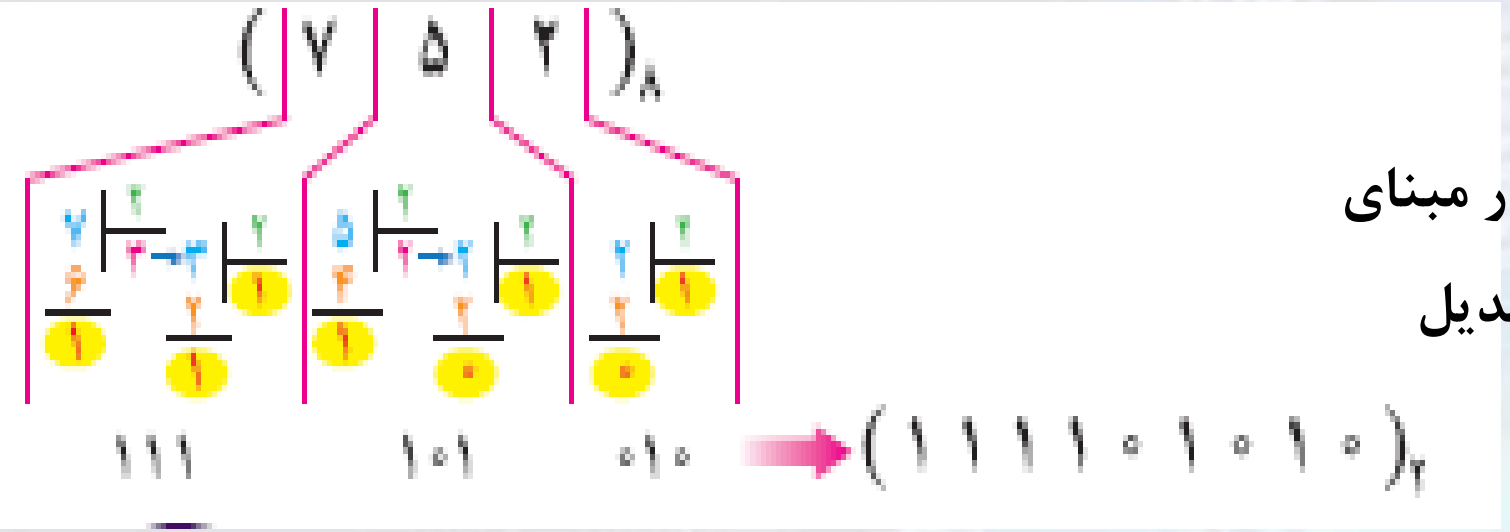
مرحلهٔ دوم تبدیل عدد اعشاری به دست آمده به مبنای باینری:

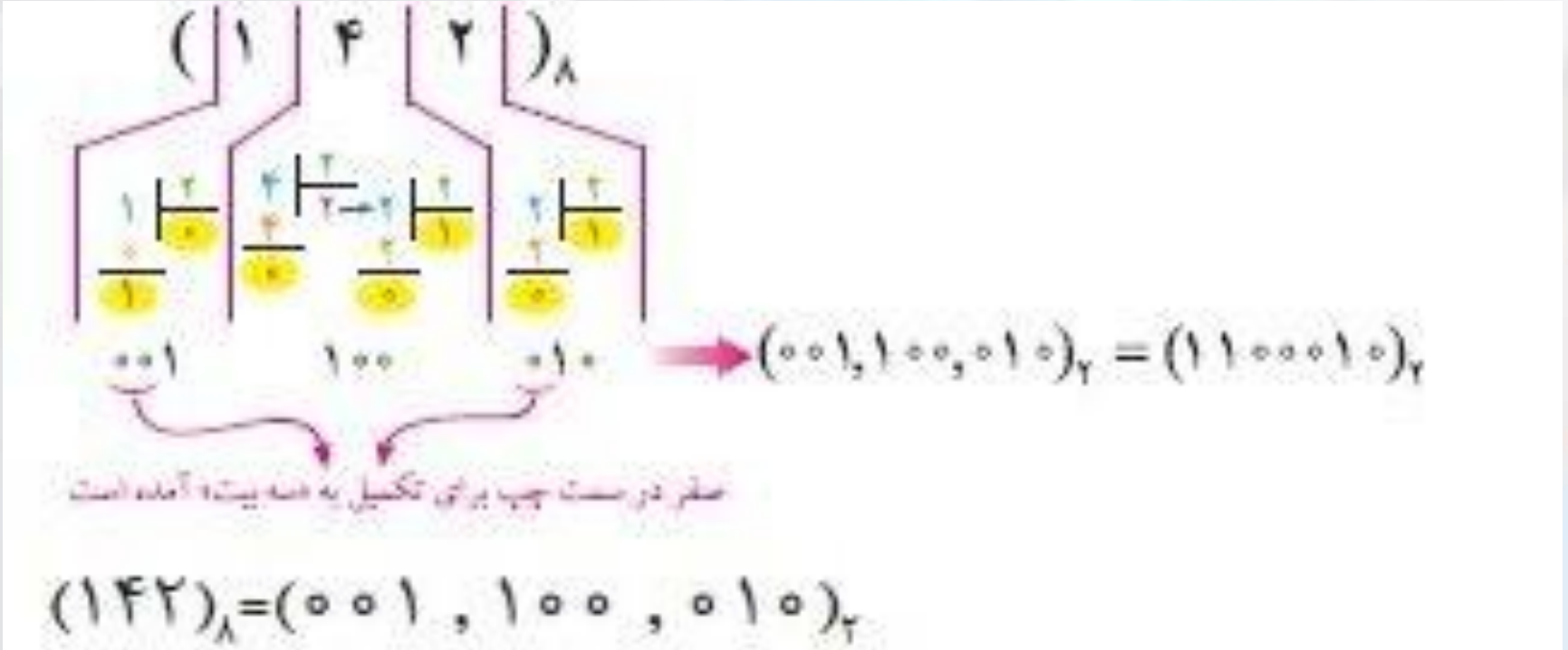


$$(۱۱۱۰۰۱۱)_۲=(۱۱۵)_{۱۰}=(۱۶۳)_۸$$

تبدیل مبنای ۸ به ۲

از روش ساده تر و سریع تری نیز برای این تبدیل می توان استفاده کرد، به این ترتیب که هر رقم در مبنای اکتال را به یک عدد سه بیتی در مبنای باینری تبدیل می کنیم.





تبدیل مبنای ۲ به ۱۶

به همان روشی که در تبدیل مبنای ۲ به ۸ آموختید،

ابتدا باید عدد در مبنای باینری را به سیستم ده دهی تبدیل کرد،

سپس با تقسیمهای متوالی بر ۱۶ به مبنای هگزا دسی مال

تبدیل کنیم

$(۱۱۱۰۰۱۱)_۲=۱\times۶۴+۱\times۳۲+۱\times۱۶+۰\times۸+۰\times۴+$

$۱\times۲+۱\times۱=۱۱۵$

۱۱۵ | ۱۶

خارج قسمت ۷

۱۱۲

باقیمانده ۳

$(۱۱۵)_{۱۰}=(۷۳)_{۱۶}=(۱۱۱۰۰۱۱)_۲$

مرحله اول تبدیل مبنای دسی مال

$۱۰۰۱۱۰=۱\times۳۲+۰\times۱۶+۰\times۸+۱\times۴+۱\times۲+۰\times۱=$

$۳۲+۴+۲=۳۸$

مرحله دوم تبدیل به مبنای هگزا دسی مال

۳۸ | ۱۶

خارج قسمت ۲

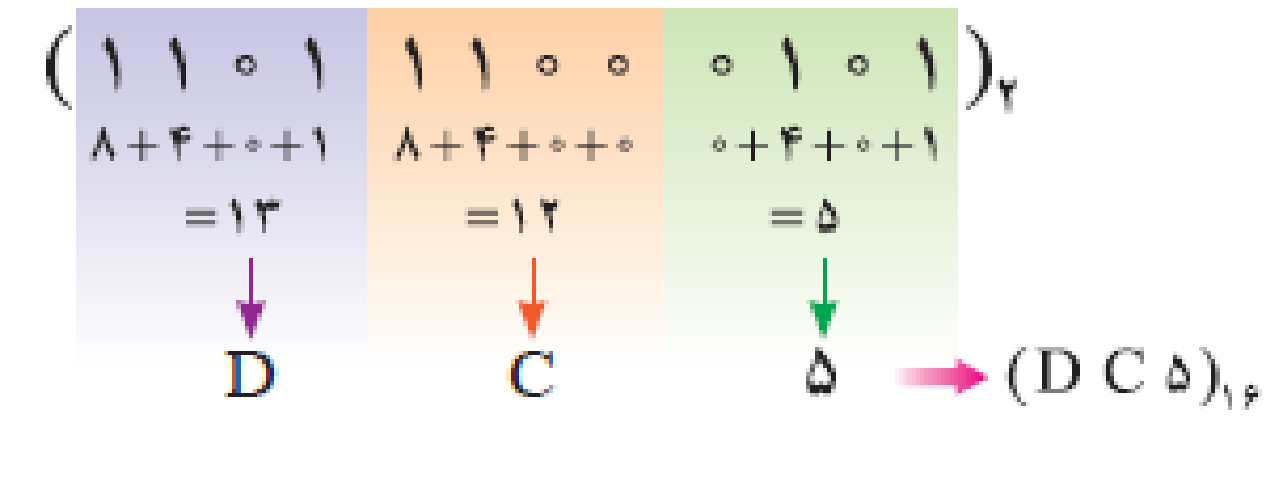
۳۲

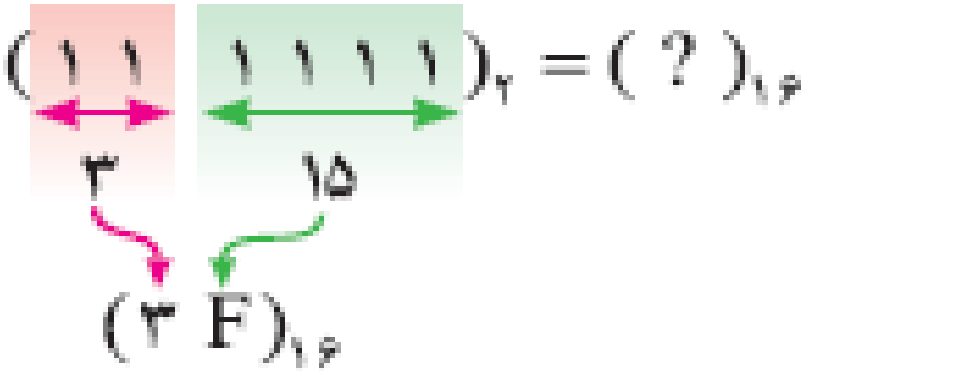
باقیمانده ۶

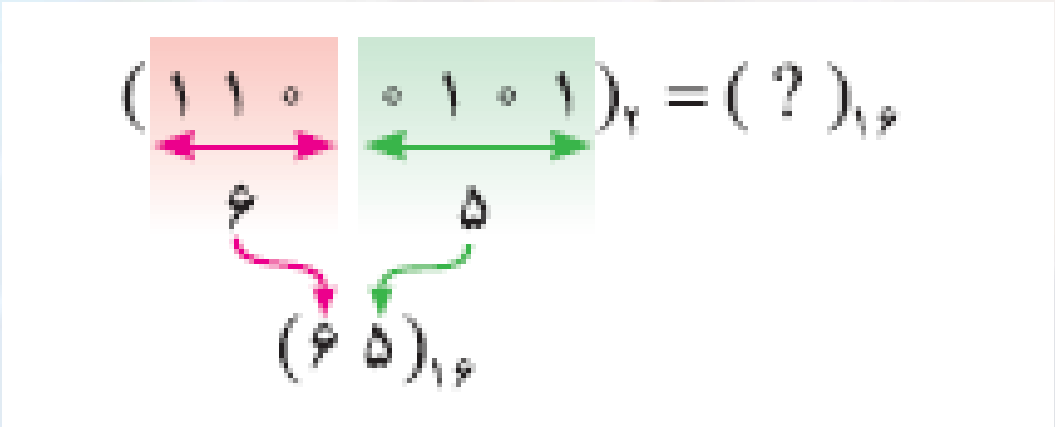
تبدیل مبنای ۲ به ۱۶

روش ساده تری نیز برای این تبدیل وجود دارد که سرعت کار را بالاتر م یبرد. می توان عدد باینری را از سمت راست چهار بیت، چهار بیت جدا کنیم و معادل هر قسمت آن را به صورت شانزده تایی بنویسیم. به طور مثال:

۱ از سمت راست چهار بیت چهار بیت جدا م یکنیم.
۲ معادل هگزا دسی مال هر چهار بیت را می نویسیم.
۳ عدد به دست آمده در مبنای هگزا دسی مال است.







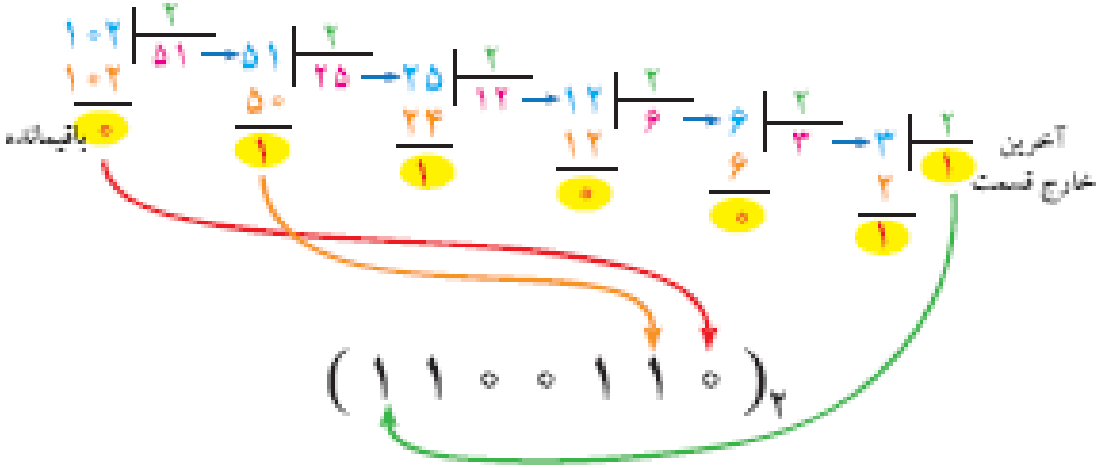
تبدیل مبنای ۱۶ به ۲

برای تبدیل اعداد در مبنای شانزده تایی به مبنای دودویی، ابتدا باید عدد در سیستم شانزده تایی را به سیستم ده دهی برده، سپس با تقسیم های متوالی بر ۲ به مبنای دودویی به روش زیر عمل میکنیم.

$(۶۶)_{۱۶}=۶\times۱۶^{۱}+۶\times۱۶^{۰}=$

$۹۶+۶\times۱=۹۶+۶=۱۰۲$

$(۶۶)_{۱۶}=(۱۰۲)_{۱۰}$

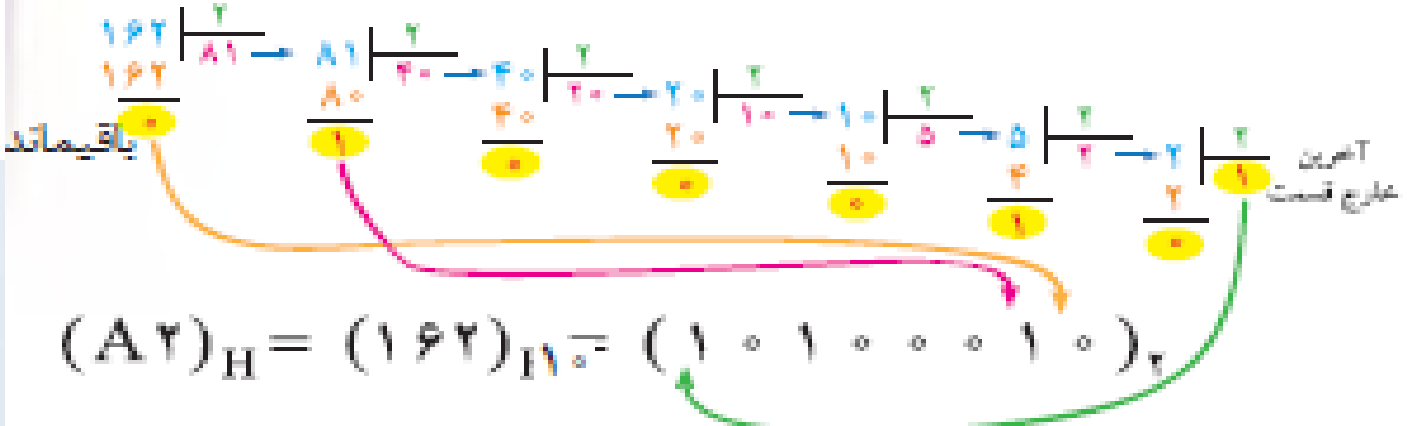


$(۶۶)_{۱۶}=(۱۰۲)_{۱۰}=(۱۱۰۰۱۱۰)_{۲}$

$(A۲)_{H}=A\times۱۶^{۱}+۲\times۱۶^{۰}=$

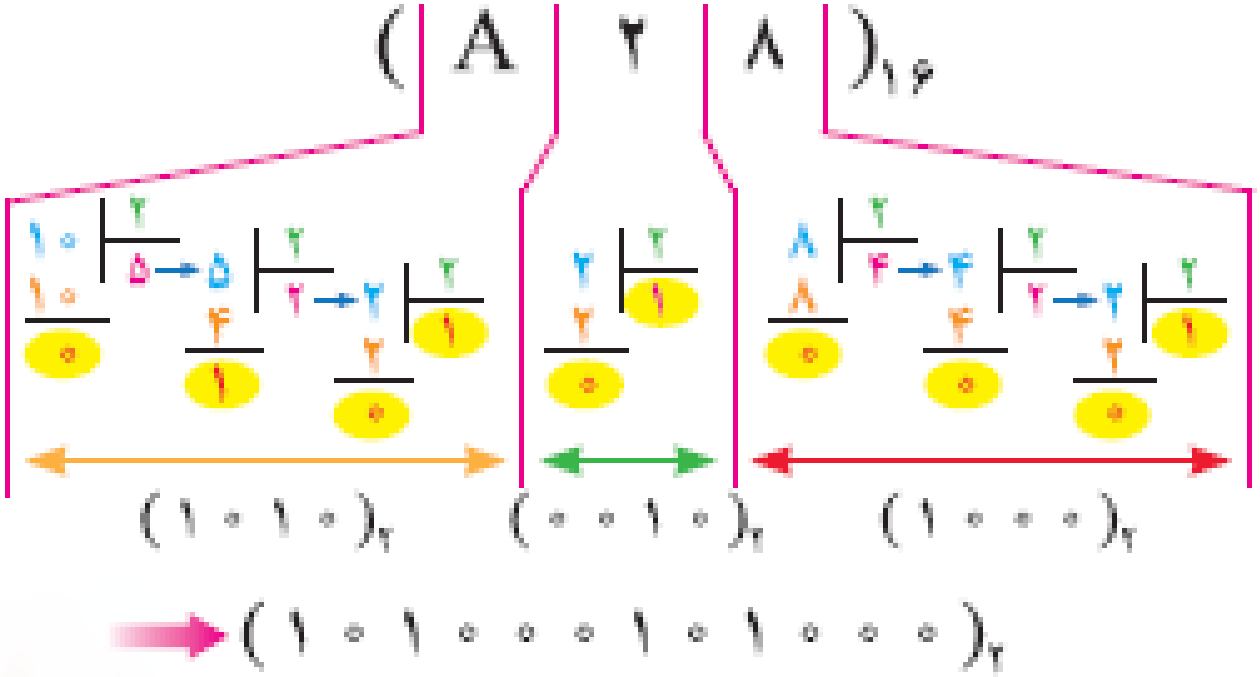
$(۱۰)\times۱۶+۲\times۱=۱۶۰+۲=۱۶۲$

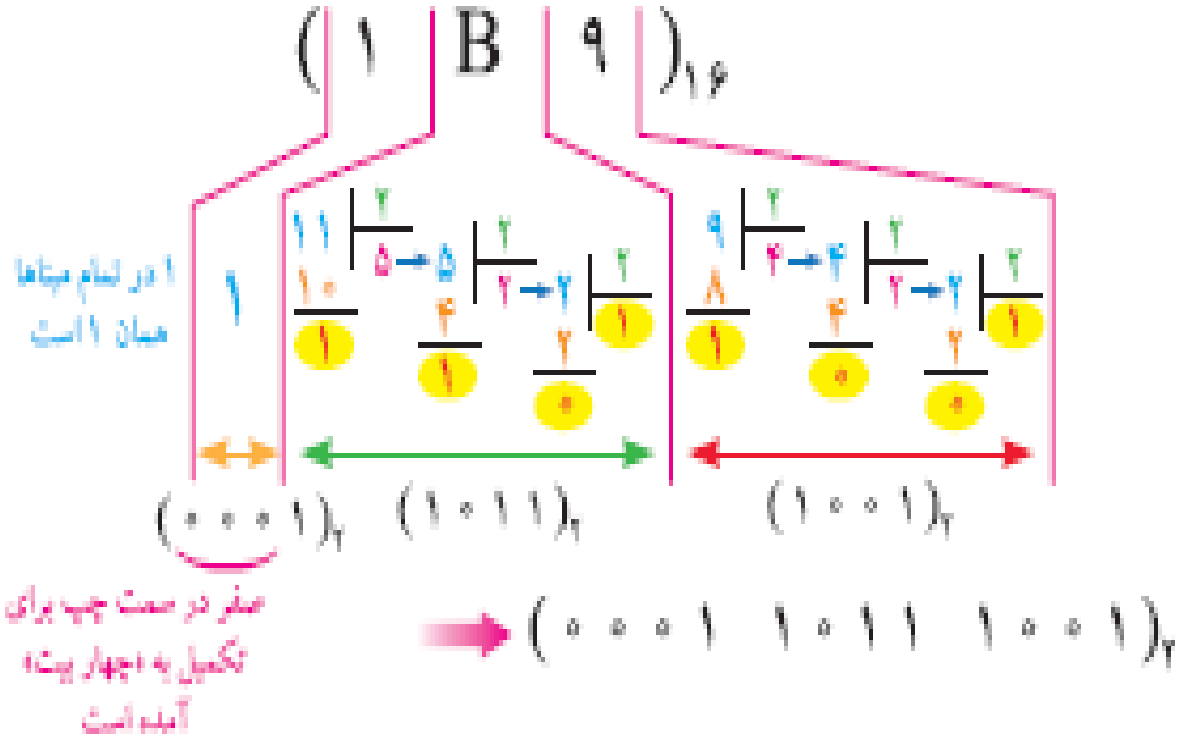
$(A۲)_{H}=(۱۶۲)_{۱۰}$



تبدیل مبنای ۱۶ به ۲

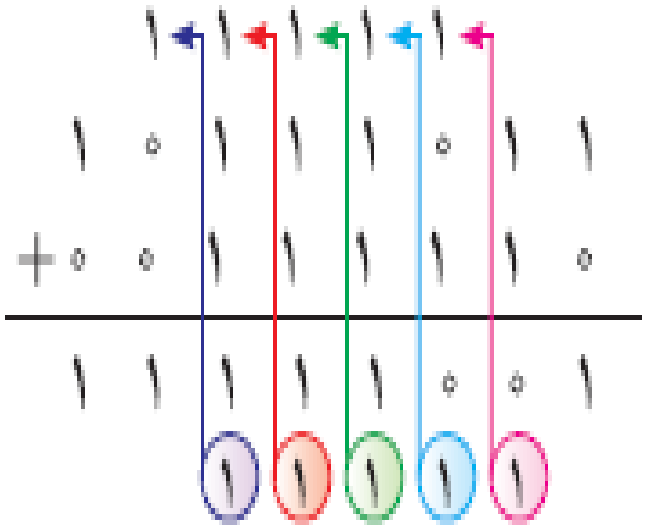
روش ساده تر و سریع تر نیز برای این تبدیل می توان استفاده کرد، به این ترتیب که هر رقم درمبنای هگزا دسی مال را به یک عدد چهار بیتی در مبنای باینری تبدیل می کنیم.



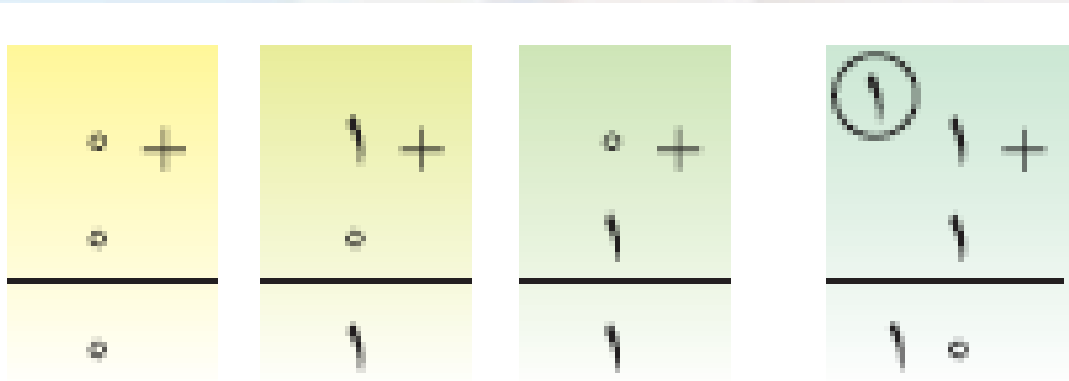


جمع باینری

جمع در سیستم باینری: جمع در این سیستم، شبیه به جمع در سیستم اعشاری است. در سیستم اعشاری، هرگاه جمع دو رقم از ده بیش تر می شود، یک واحد به رقم بعد آن اضافه م یکنیم که به آن ده بر یک می گوییم. در سیستم باینری، هرگاه جمع دو رقم دو شود (۱+۱) ایجاد دو بریک می کند و باید عدد یک یک را به رقم بعدی اضافه کرد

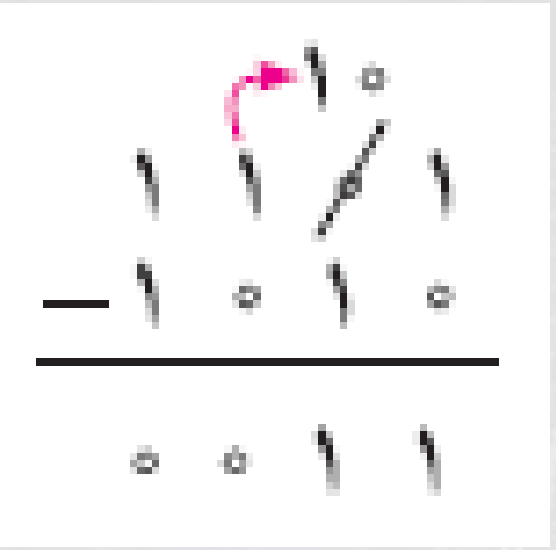


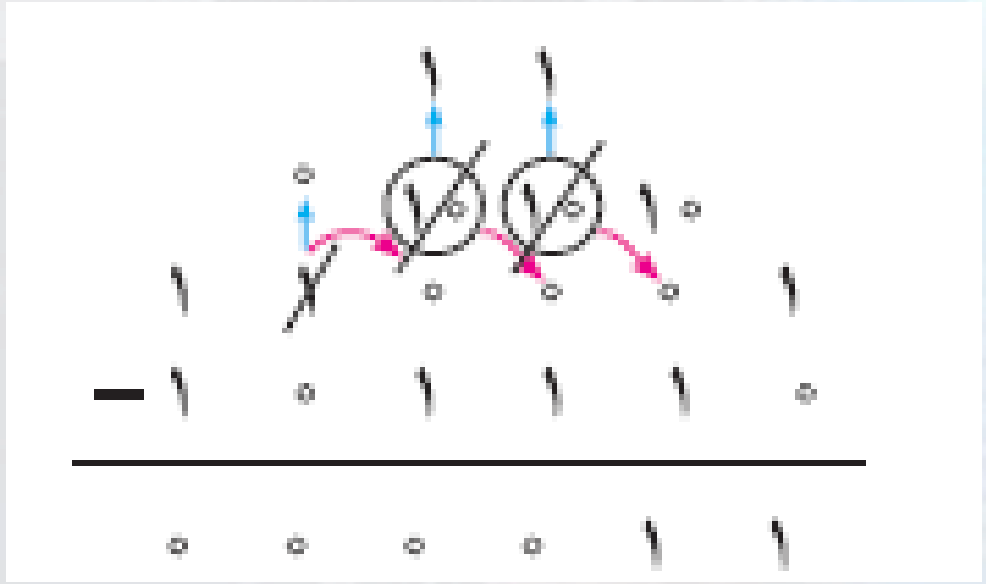
$$(1\circ111\circ11)_2+(11111\circ)_2=(111111\circ\circ1)_2$$

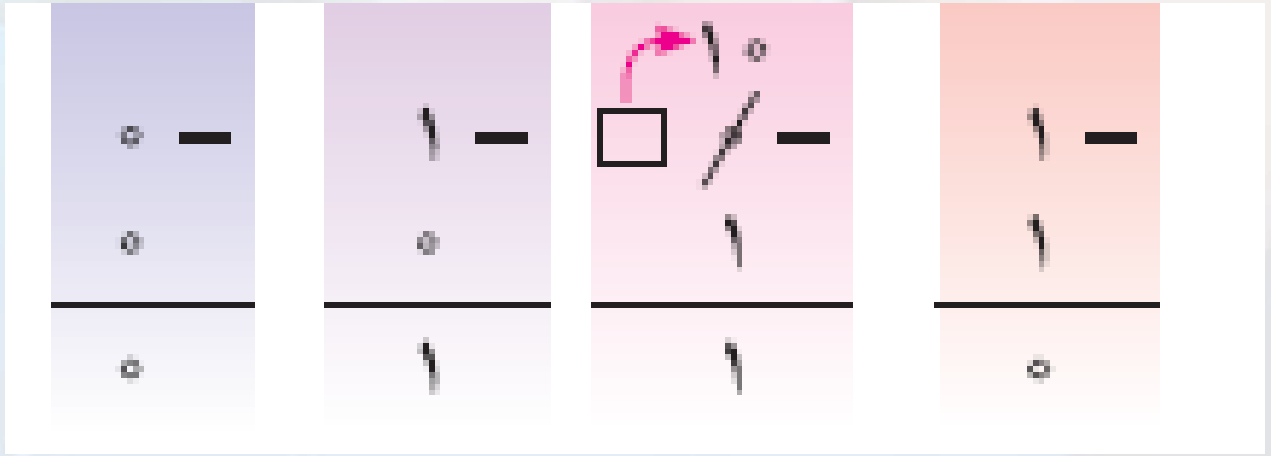


تفریق باینری

تفریق یک عدد باینری از عدد باینری دیگر با روشی مانند تفریق اعداد دسی مال انجام م یپذیرد، یعنی اگر رقم بزرگتر از رقم کوچک تر کم شود یک واحد از مکان بعدی قرض گرفته م یشود و در مکانی که یک واحد قرض گرفته شده یک را به صفر تبدیل میکنیم. واحد قرض گرفته شده را Borrow میگویند.







نقش کد در سیستم دیجیتال

معنای واقعی (کد کردن)همان رمز کردن یا به صورت رمز درآوردن اطلاعات است.

لازم به ذکر است که تبدیل اطلاعات به کد، نه فقط برای ایجاد ارتباط لازم است، بلکه یکی از روش های با اهمیت در تشخیص خطا و در صورت لزوم برطرف کردن خطا برای اطلاعات پردازش شونده در سیستم می باشد. از آنجا که رمز کردن اطلاعات برای ایجاد ارتباط با کامپیوتر صورت می گیرد و از طرفی کامپیوتر تنها صفرها و یک ها را می تواند بفهمد، برای کد کردن اطلاعات کافی است آنها را به صورت رشته ای از صفرها و یک ها درآوریم.

در کد BCD هر رقم ده دهی را با چهار بیت باینری معادل آن نشان می دهند

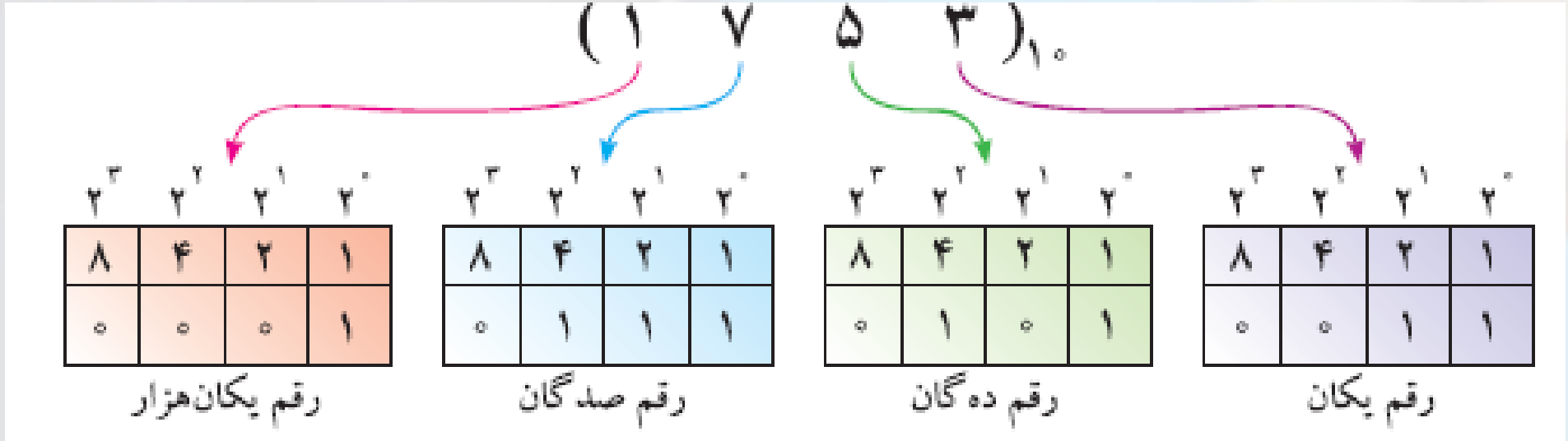
$(۳)_{۱۰} \longrightarrow (۱۱)_{۲} \longrightarrow (۰۰۱۱)_{BCD}$

$(۹)_{۱۰} \longrightarrow (۱۰۰۱)_{۲} \longrightarrow (۱۰۰۱)_{BCD}$

$(۵)_{۱۰} \longrightarrow (۱۰۱)_{۲} \longrightarrow (۰۱۰۱)_{BCD}$

اعداد یک رقمی ده دهی و معادل باینری و BCD انها

عدد ده‌دهی	عدد باینری	عدد BCD
۰	۰	۰۰۰۰
۱	۱	۰۰۰۱
۲	۱۰	۰۰۱۰
۳	۱۱	۰۰۱۱
۴	۱۰۰	۰۱۰۰
۵	۱۰۱	۰۱۰۱
۶	۱۱۰	۰۱۱۰
۷	۱۱۱	۰۱۱۱
۸	۱۰۰۰	۱۰۰۰
۹	۱۰۰۱	۱۰۰۱

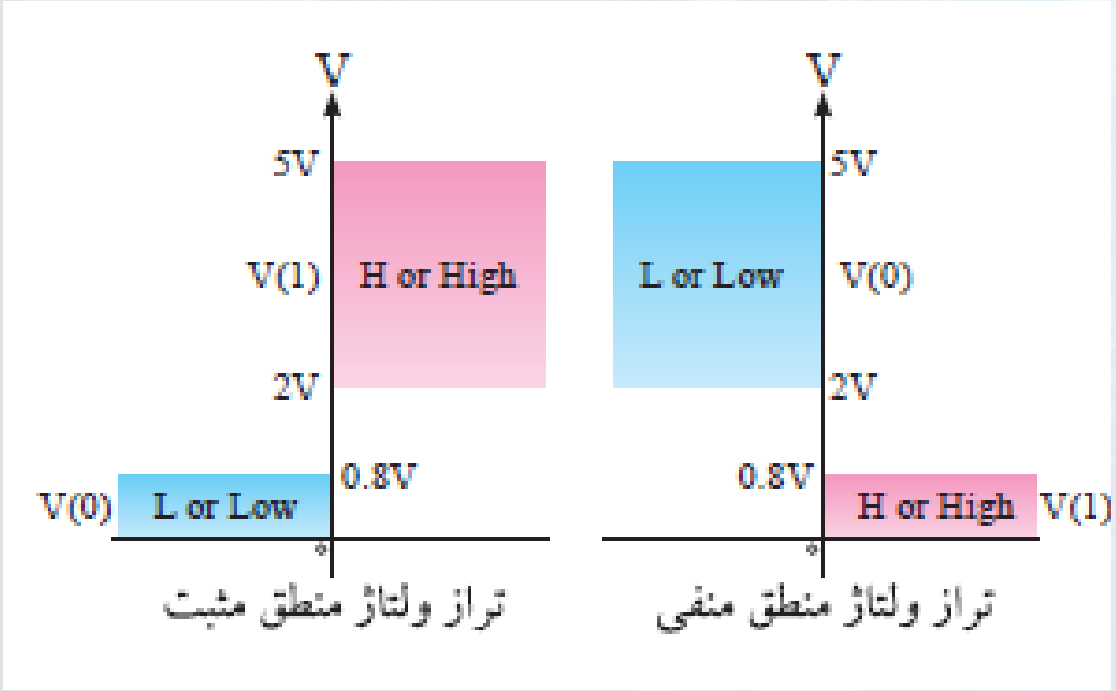


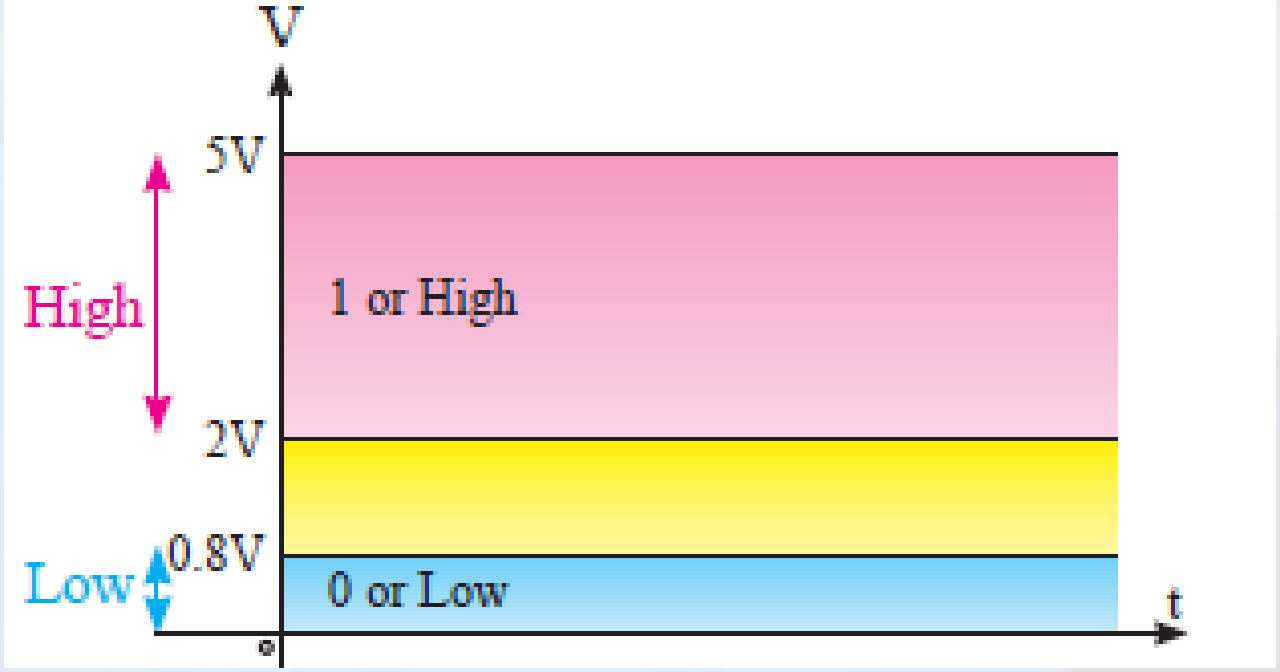
ترازهای ولتاژ – Voltage levels

مدارهای منطقی مدارهایی هستند که م یتوانند دو نوع ولتاژ زیر را از یکدیگر تشخیص دهند:

- ولتاژ بالا
- ولتاژ پایین

معمولاً مقدار واقعی ولتاژ چندان مهم نیست و در یک محدودهٔ مشخصی از ولتاژ ممکن است این دو حالت اتفاق بیفتد.



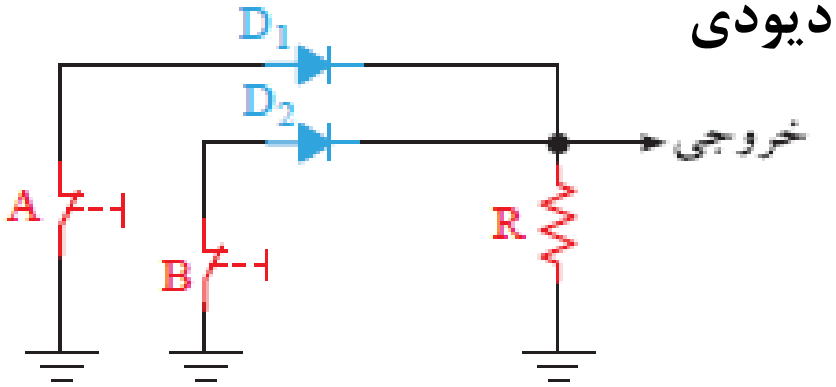


دروازه های منطقی پایه:

دروازه های منطقی پایه یا گیت ها مدارهایی هستند که تعداد یک یا بیشتر از یک ورودی و یک خروجی دارند.

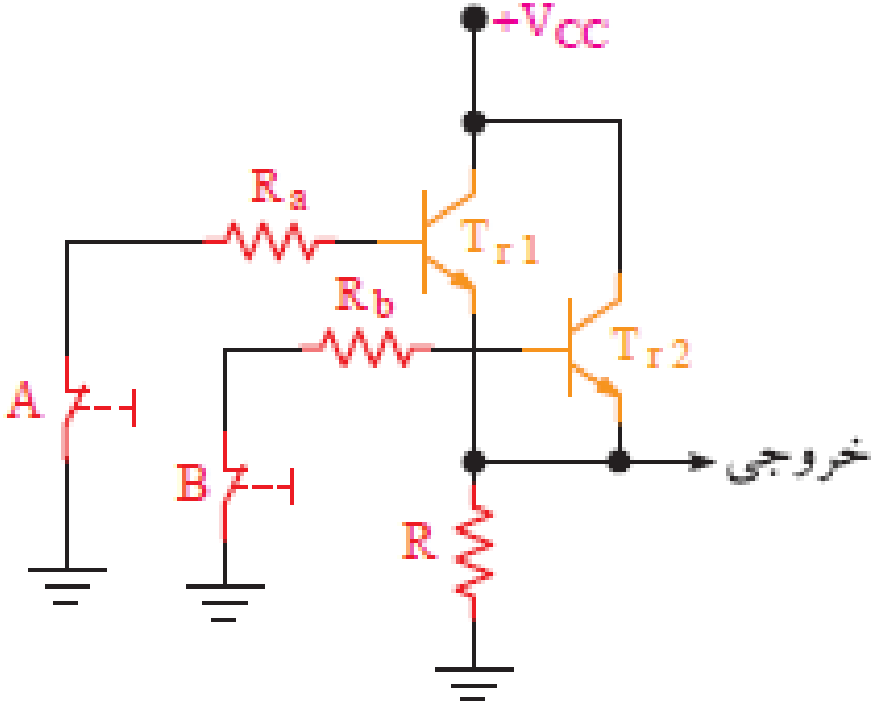
گیت OR (یا)

ساختمان دیودی



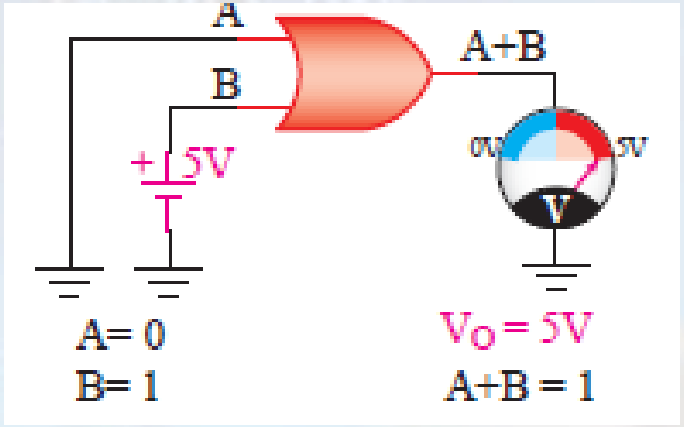
A B	D_1	D_2	V_o
L L	قطع	قطع	L
L H	قطع	هدایت	H
H L	هدایت	قطع	H
H H	هدایت	هدایت	H

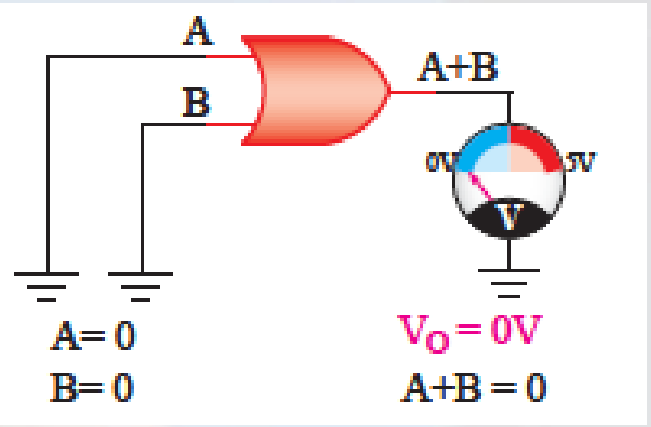
ساختمان ترانزیستوری

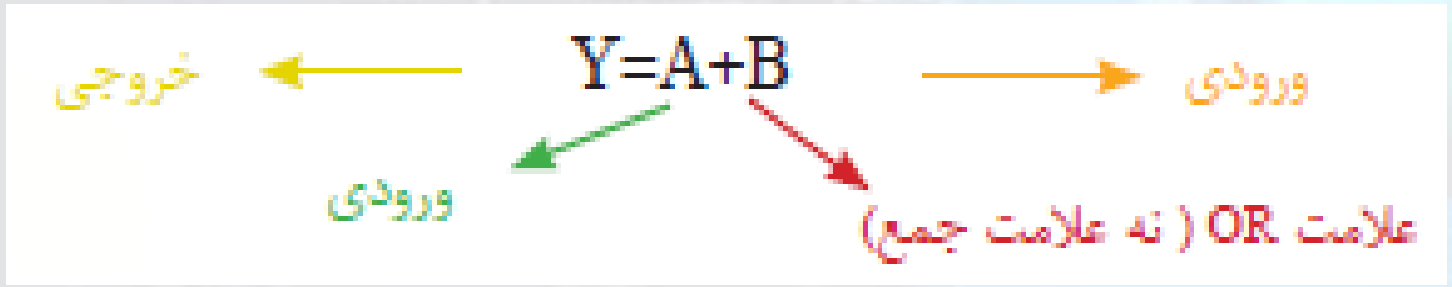


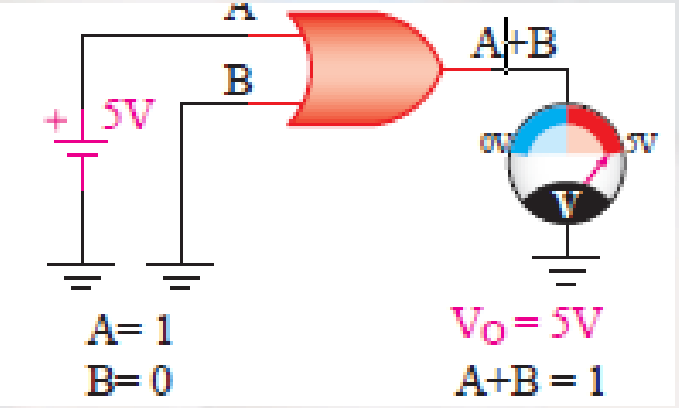
A B	Tr_1	Tr_2	V_o
L L	قطع	قطع	L
L H	قطع	وصل	H
H L	وصل	قطع	H
H H	وصل	وصل	H

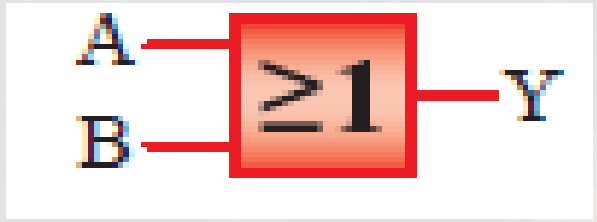
ورودی‌ها A B	خروجی Y
۰ ۰	۰
۰ ۱	۱
۱ ۰	۱
۱ ۱	۱

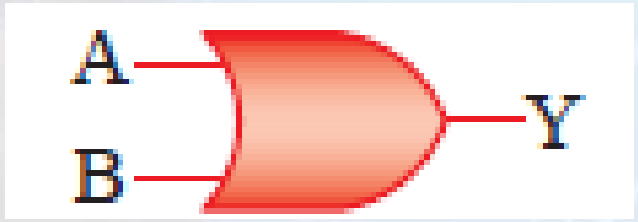


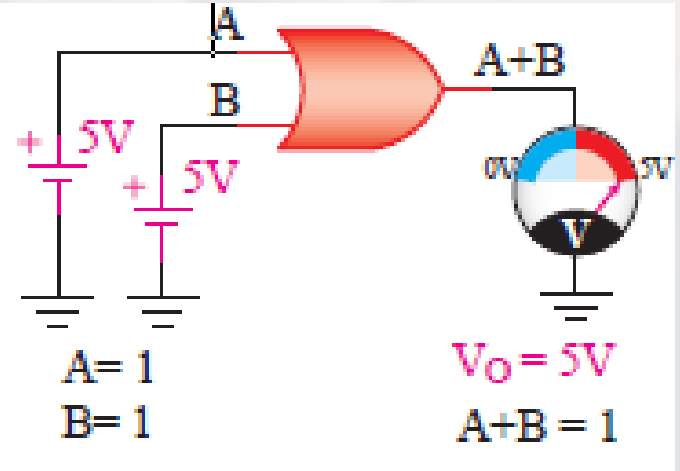






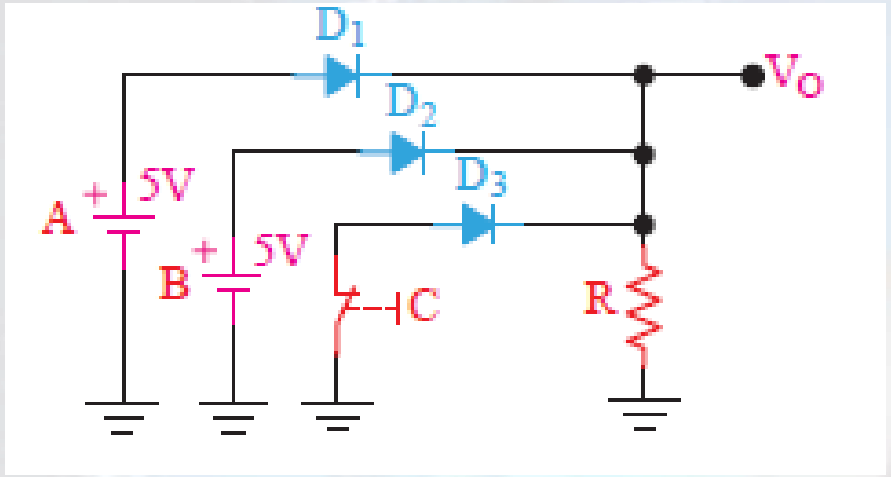


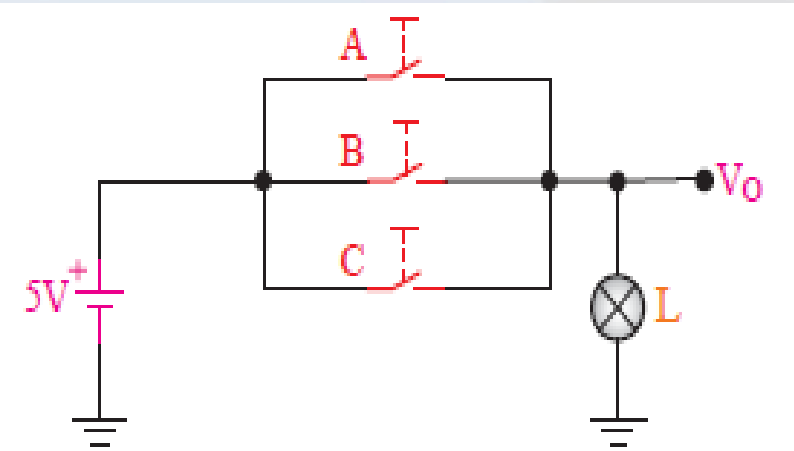


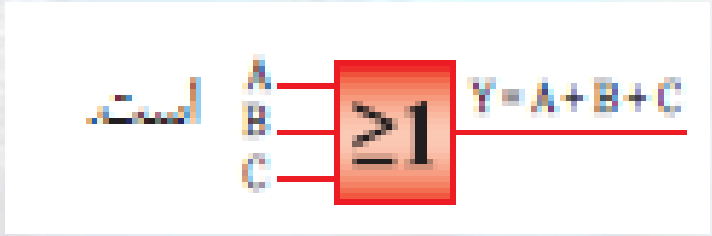


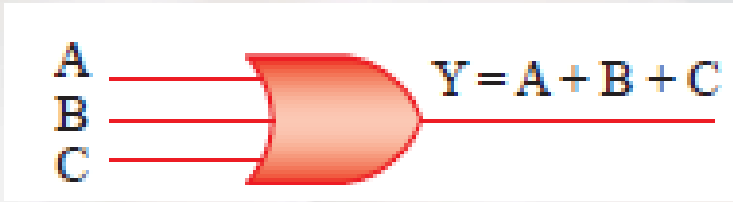
گیت OR با ورودی بیشتر

A	B	C	Y
۰	۰	۰	۰
۰	۰	۱	۱
۰	۱	۰	۱
۰	۱	۱	۱
۱	۰	۰	۱
۱	۰	۱	۱
۱	۱	۰	۱
۱	۱	۱	۱



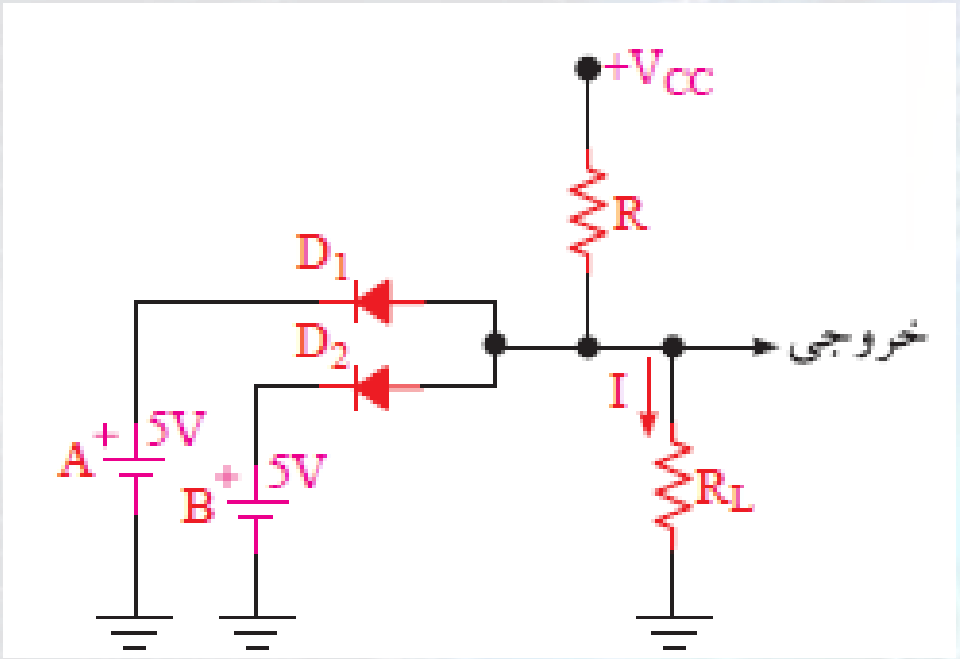


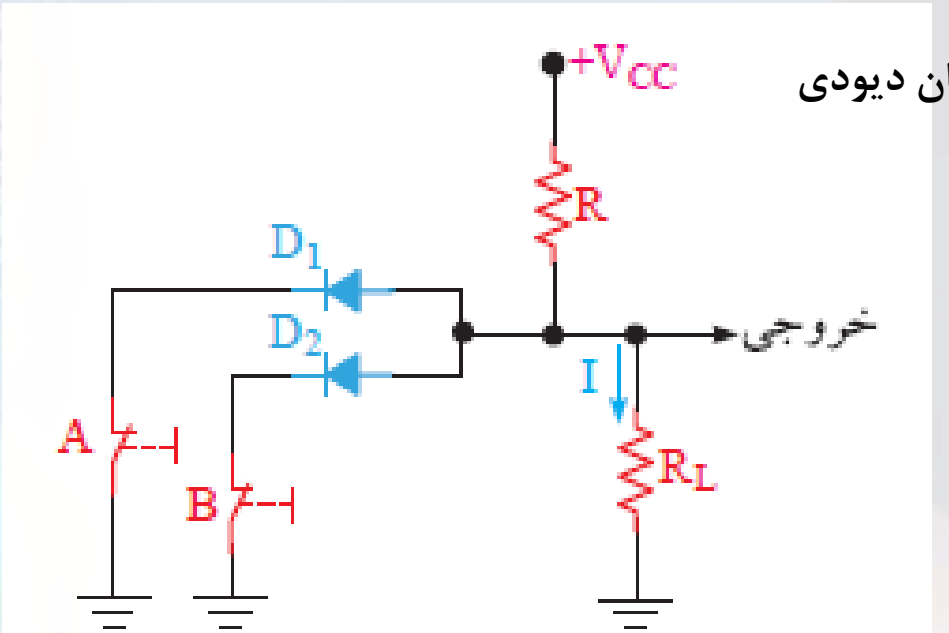




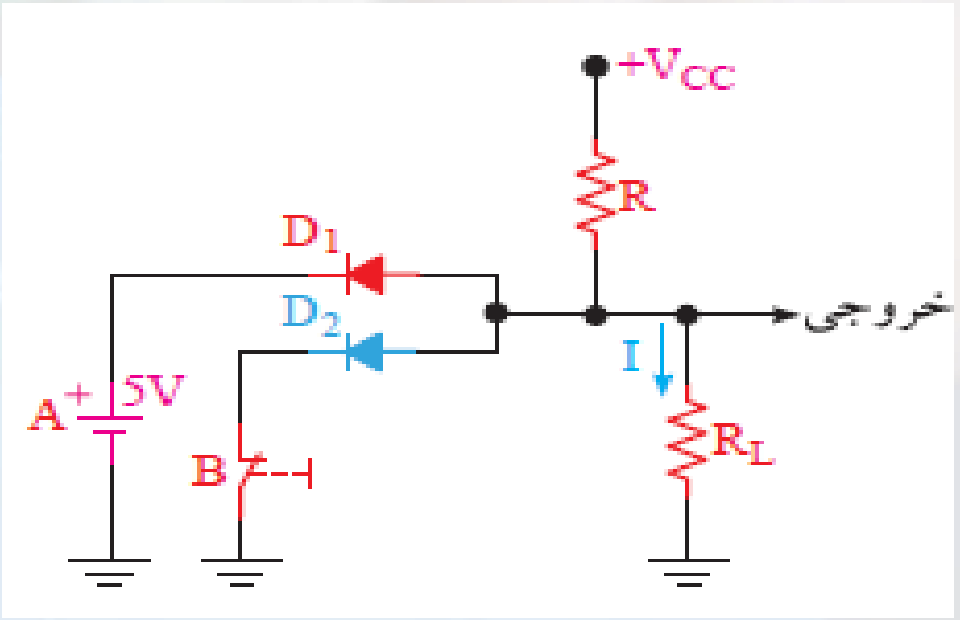
گیت AND (و)

ساختمان دیودی



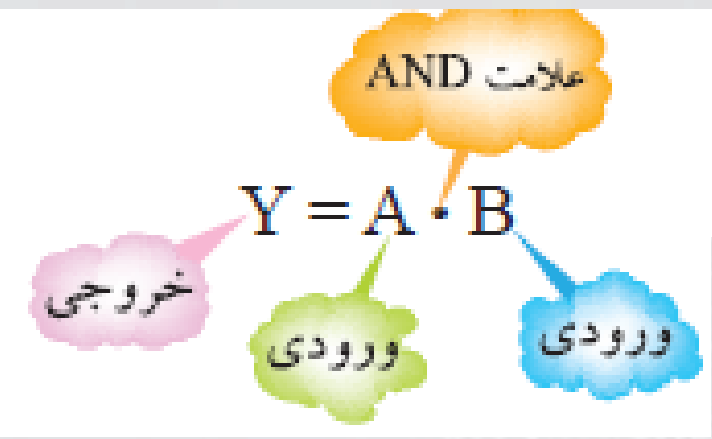


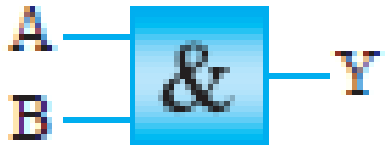
A B	D_1	D_2	V_o
L L	هدایت	هدایت	L
L H	هدایت	قطع	L
H L	قطع	هدایت	L
H H	قطع	قطع	H



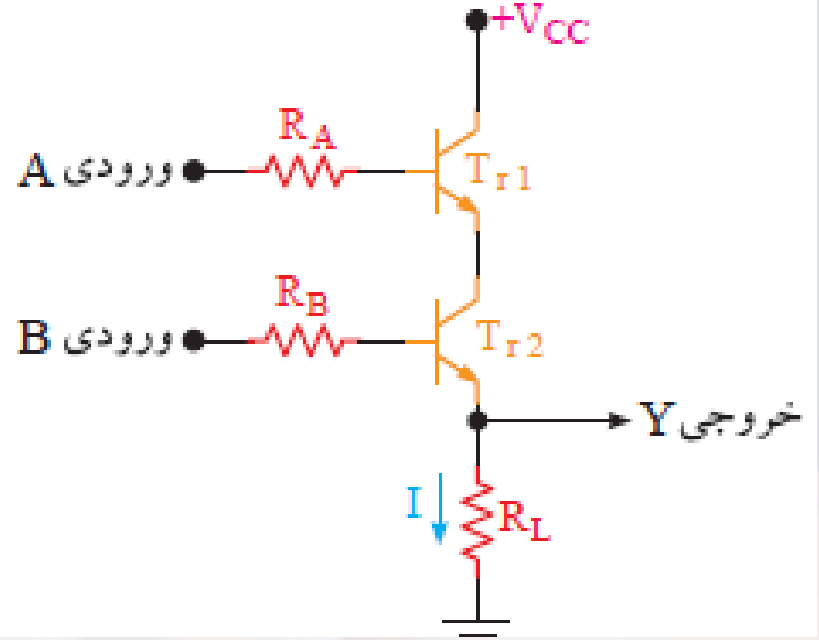
گیت AND (و)

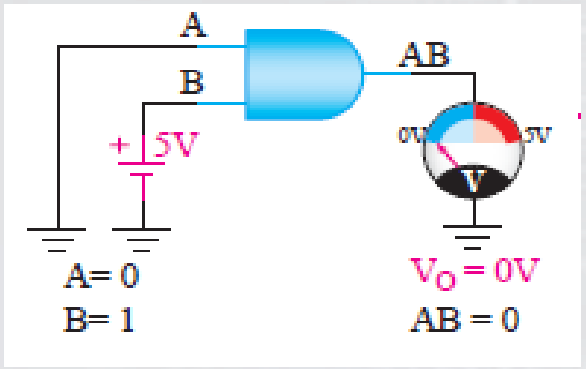
ساختمان ترانزیستوری

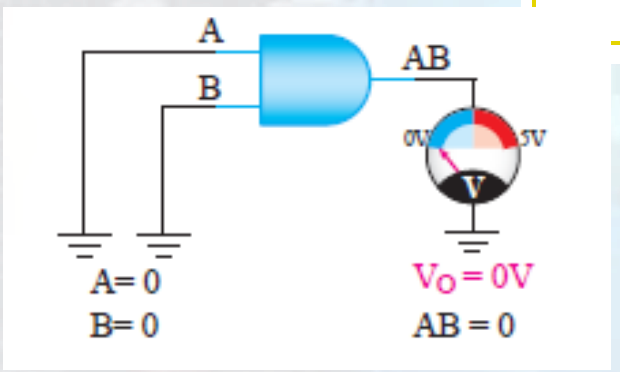


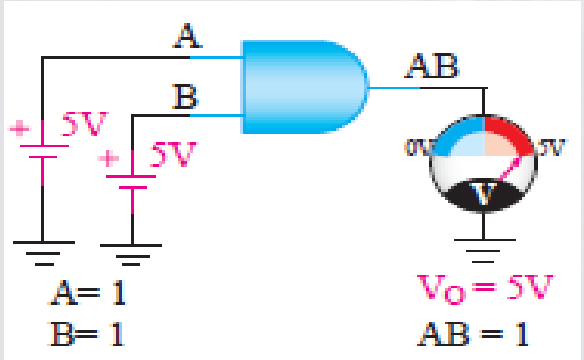


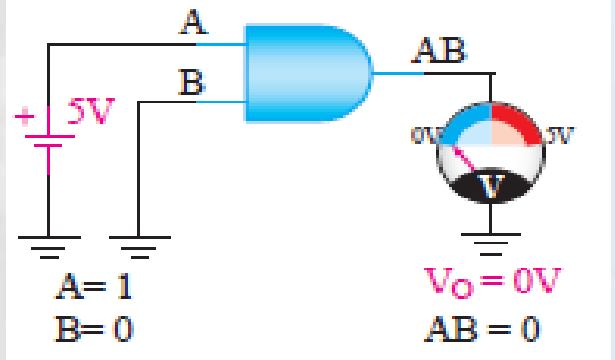
A B	Y
۰ ۰	۰
۰ ۱	۰
۱ ۰	۰
۱ ۱	۱







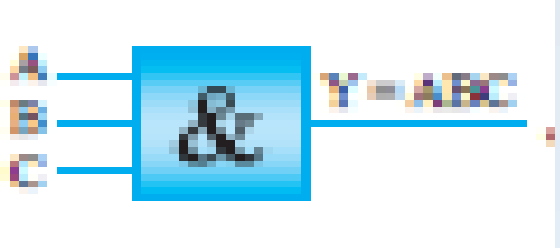


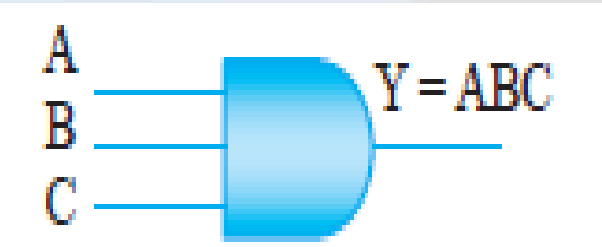


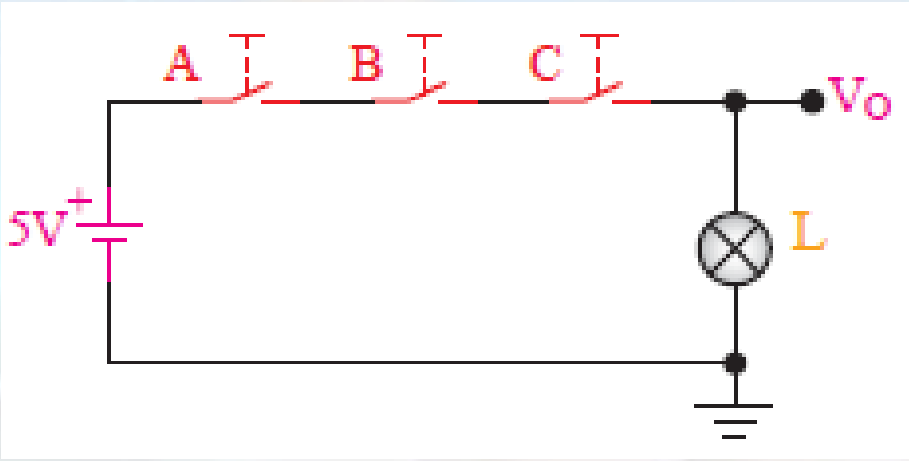
خروجی Y	Tr_2	Tr_1	ورودی B	ورودی A
L	قطع	قطع	L	L
L	وصل	قطع	H	L
L	قطع	وصل	L	H
H	وصل	وصل	H	H

گیت AND (و) با ورودی بیشتر

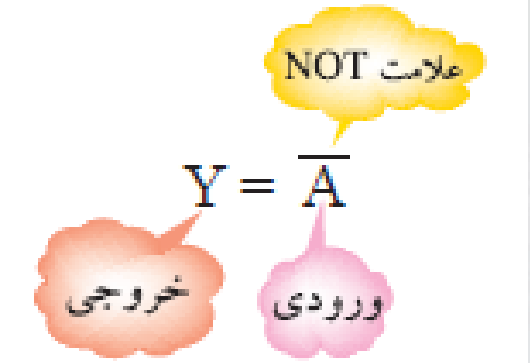
A	B	C	Y
۰	۰	۰	۰
۰	۰	۱	۰
۰	۱	۰	۰
۰	۱	۱	۰
۱	۰	۰	۰
۱	۰	۱	۰
۱	۱	۰	۰
۱	۱	۱	۱

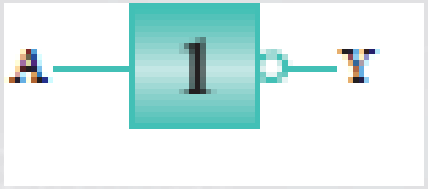


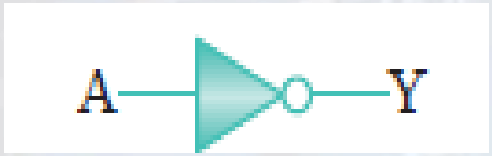




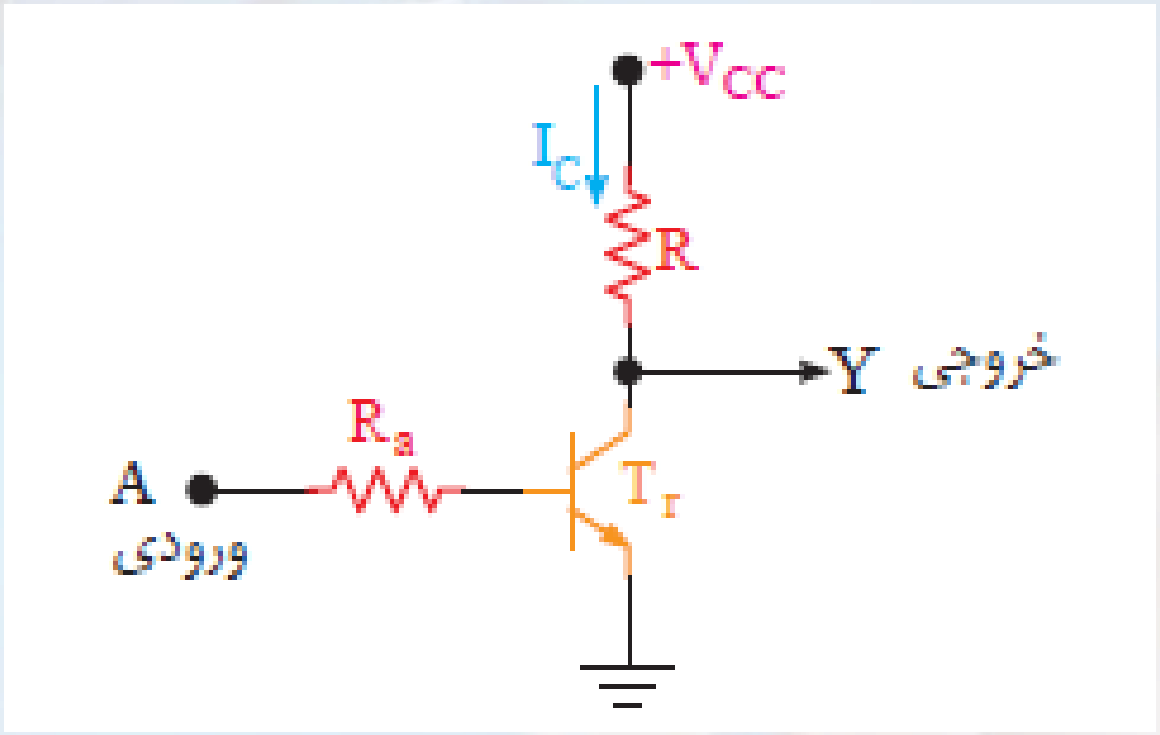
گیت NOT (نه)







ساختمان ترانزیستوری

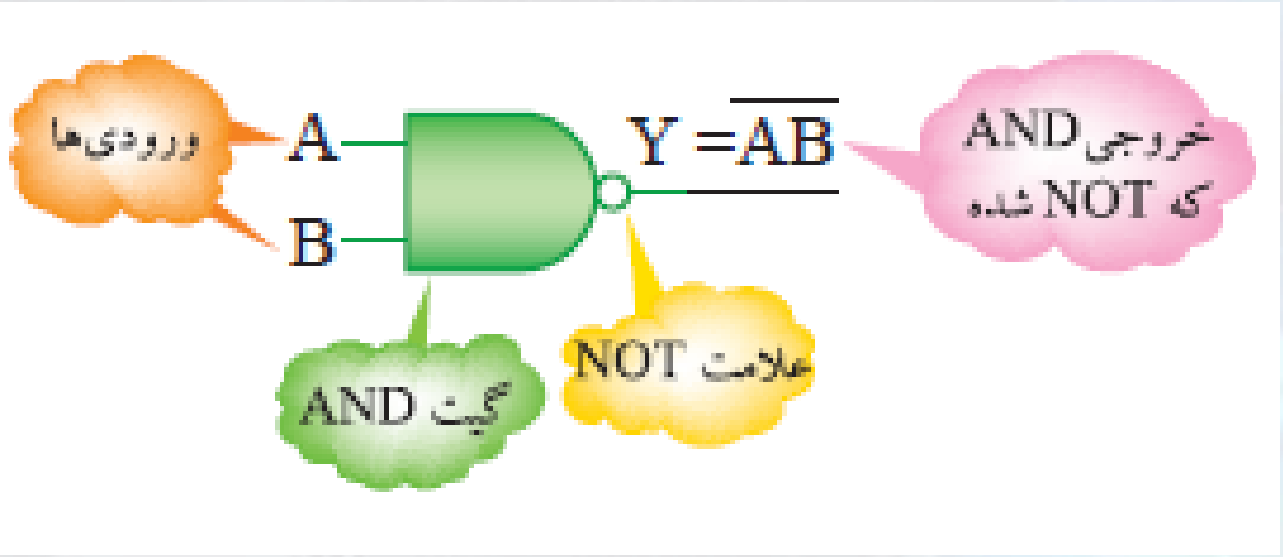


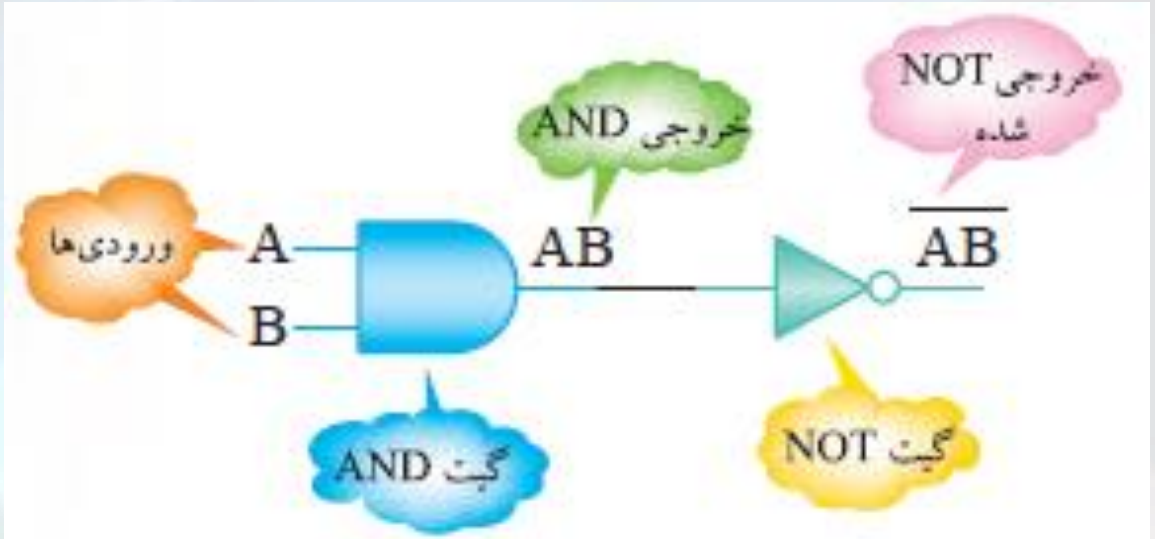
A	Y
۰	۱
۱	۰

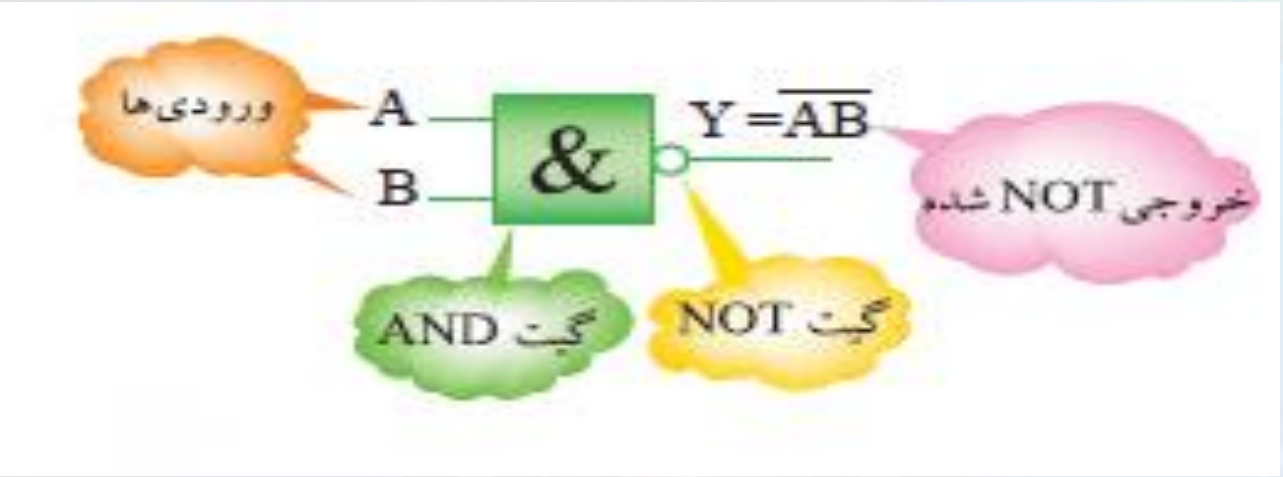
ورودی A	Tr	خروجی Y
L	قطع	H
H	وصل	L

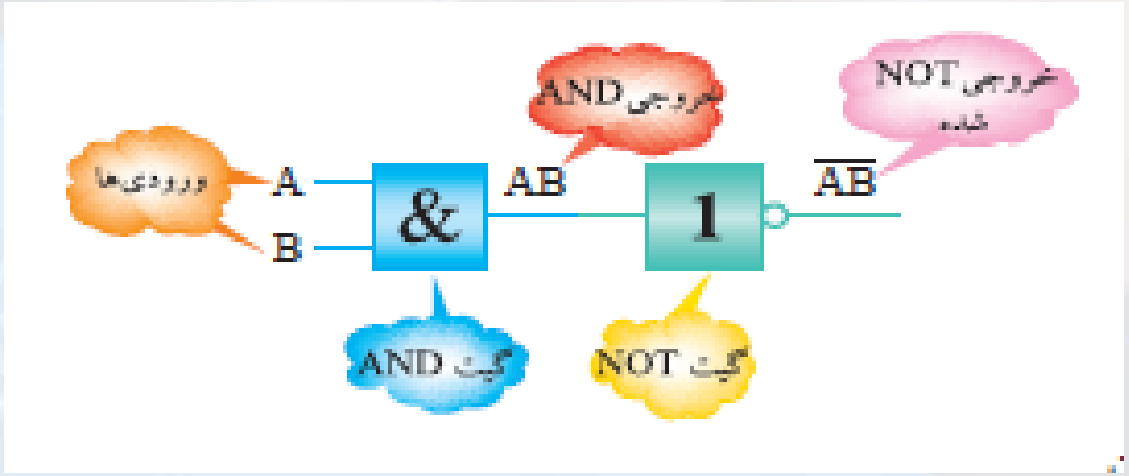
دروازه های منطقی ترکیبی:

دروازه منطقی NANAD



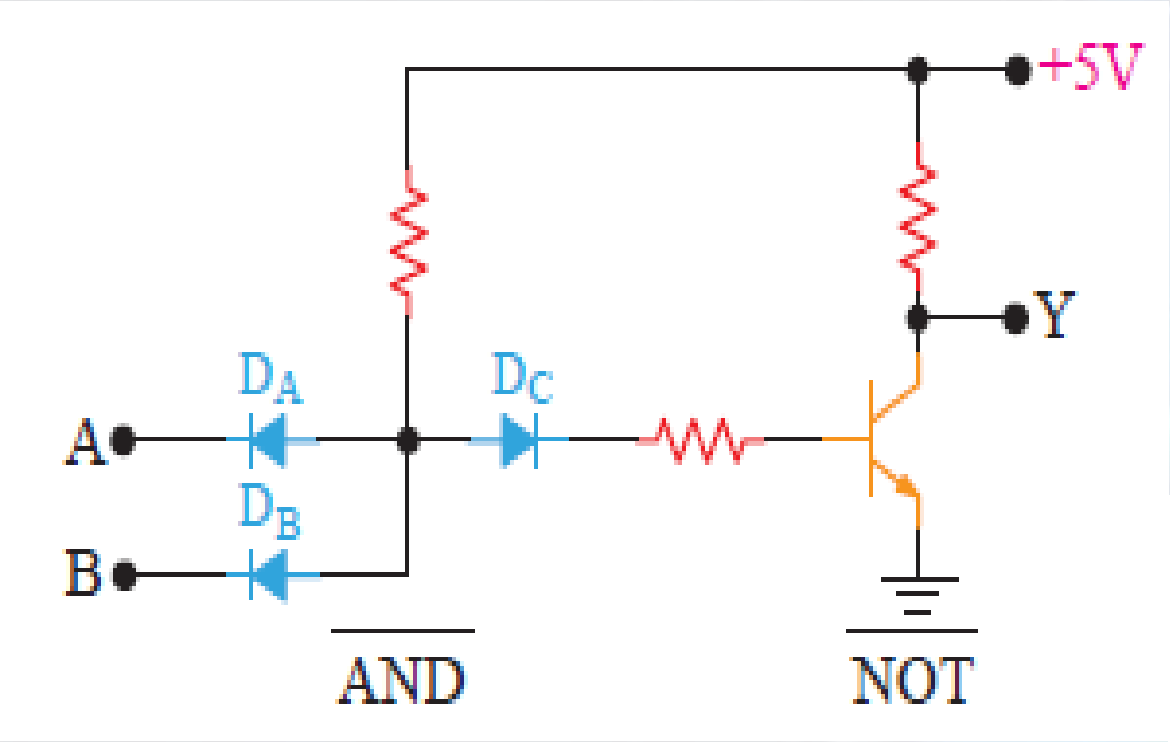


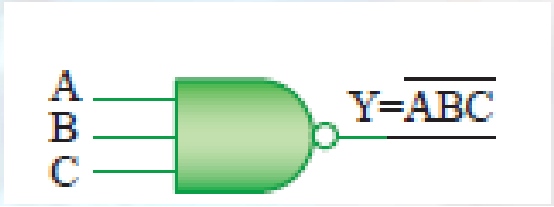




دروازه های منطقی ترکیبی:

دروازه منطقی NANAD



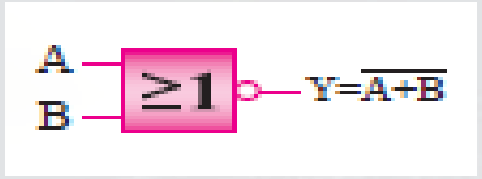


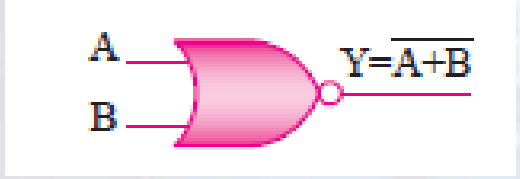
A	B	C	$\overline{Y-ABC}$
۰	۰	۰	۱
۰	۰	۱	۱
۰	۱	۰	۱
۰	۱	۱	۱
۱	۰	۰	۱
۱	۰	۱	۱
۱	۱	۰	۱
۱	۱	۱	۰

A	B	AB	$\overline{AB}$
۰	۰	۰	۱
۰	۱	۰	۱
۱	۰	۰	۱
۱	۱	۱	۰

دروازه های منطقی ترکیبی:

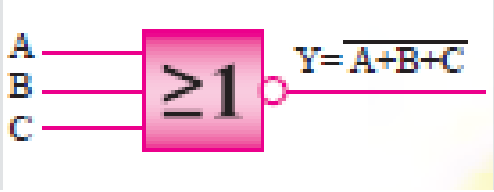
دروازه منطقی NOR

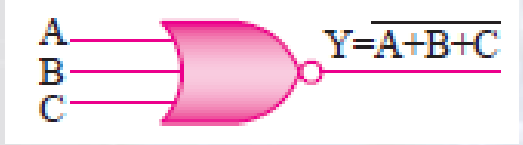




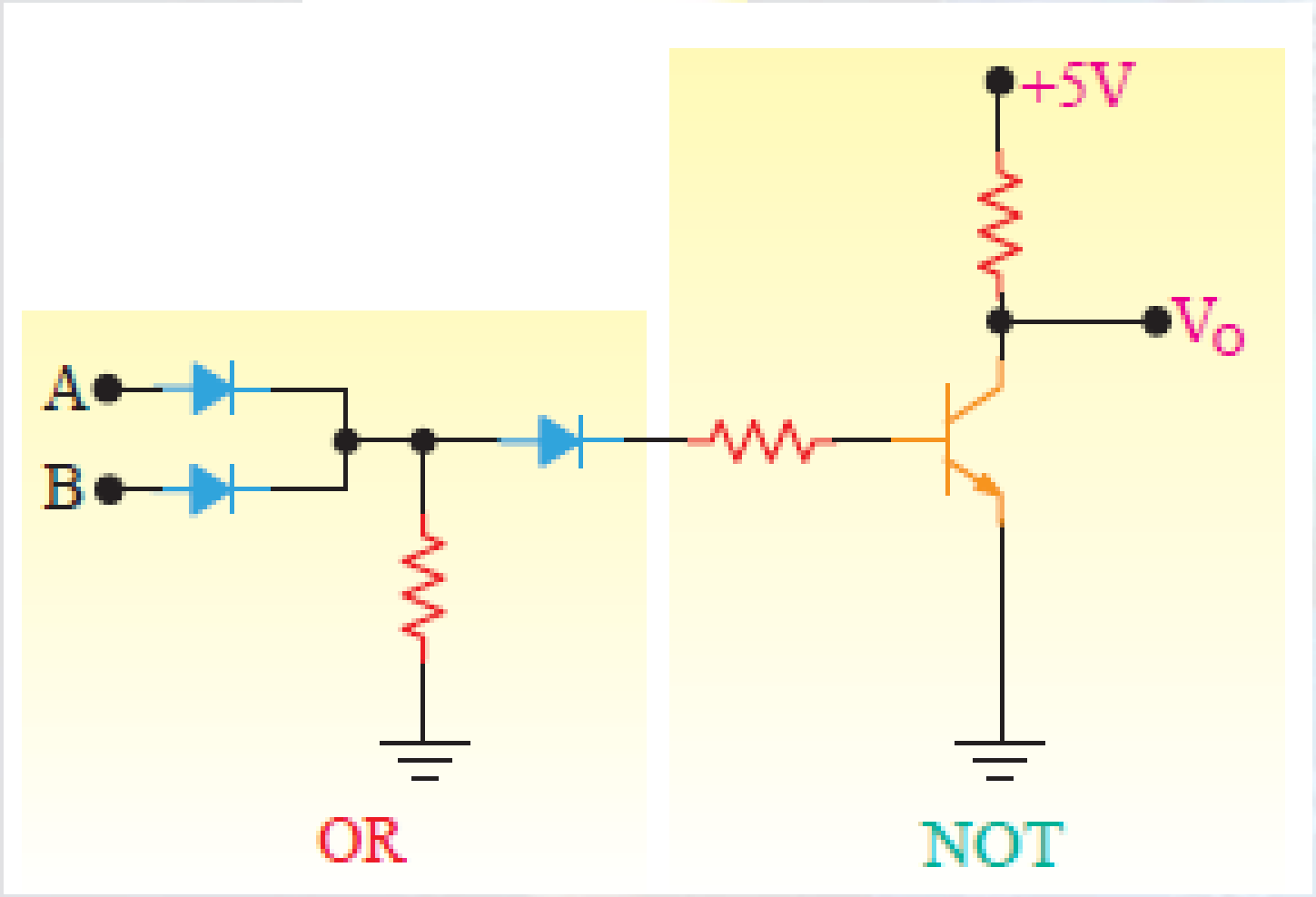
A	B	A + B	$\overline{A+B}$
۰	۰	۰	۱
۰	۱	۱	۰
۱	۰	۱	۰
۱	۱	۱	۰

دروازه های منطقی ترکیبی:





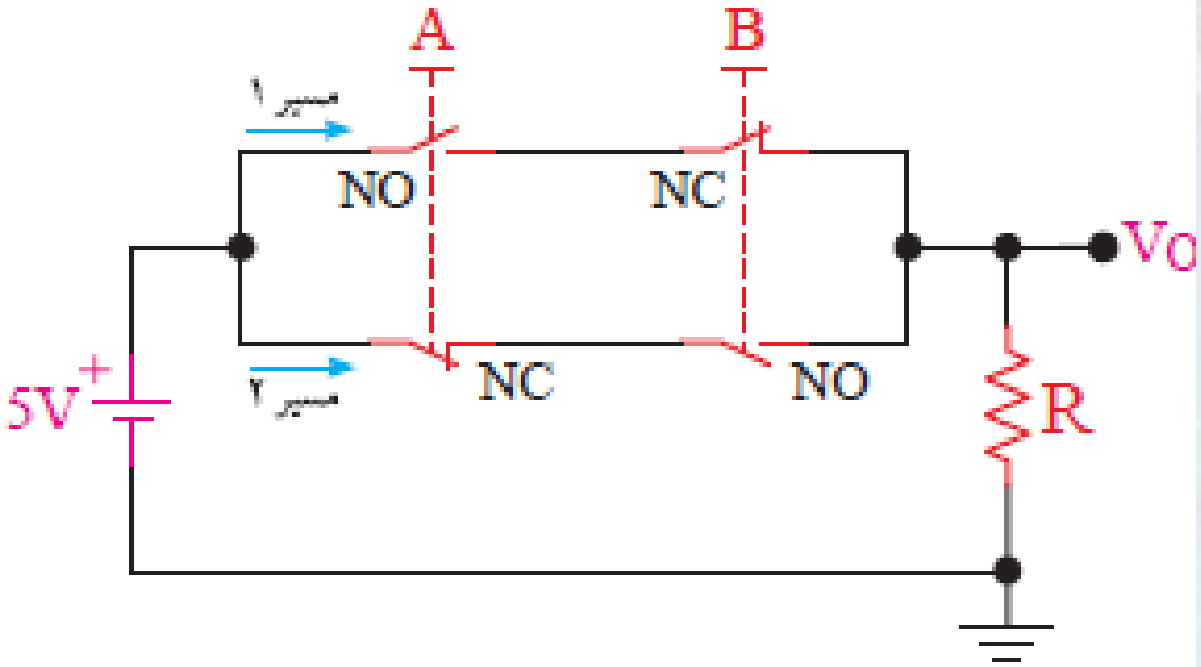
دروازه منطقی NOR



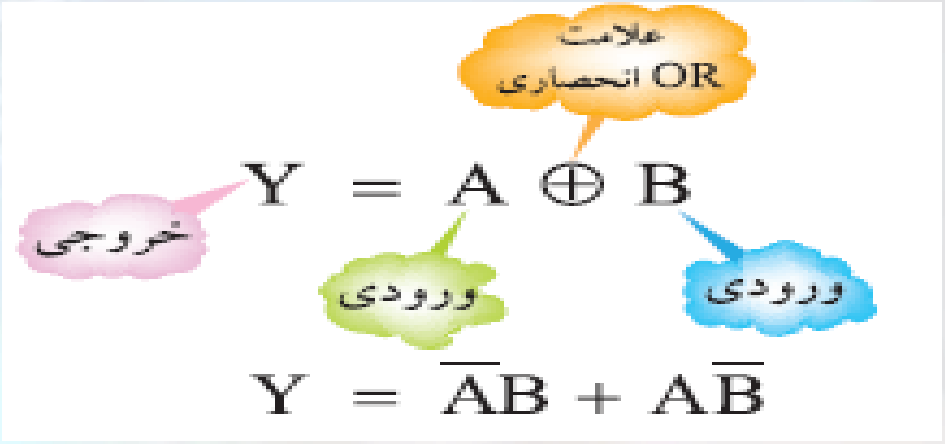
A	B	C	Y
۰	۰	۰	۱
۰	۰	۱	۰
۰	۱	۰	۰
۰	۱	۱	۰
۱	۰	۰	۰
۱	۰	۱	۰
۱	۱	۰	۰
۱	۱	۱	۰

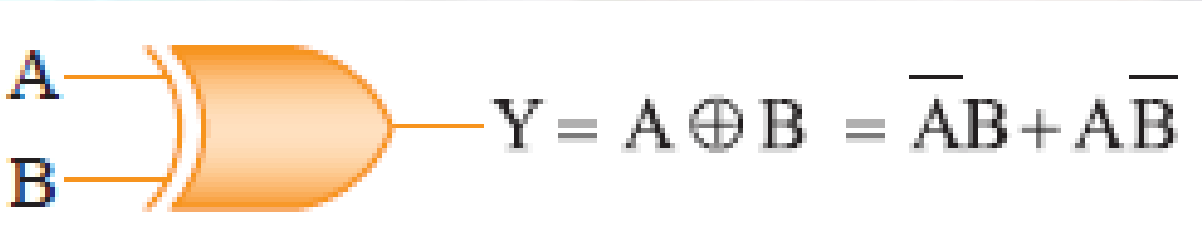
دروازه های منطقی ترکیبی:

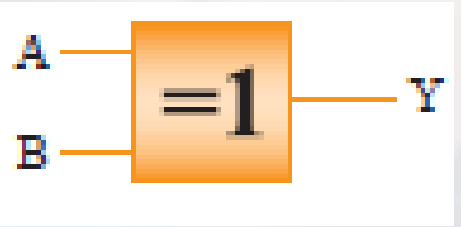
دروازه منطقی XOR



A	B	Y
۰	۰	۰
۰	۱	۱
۱	۰	۱
۱	۱	۰

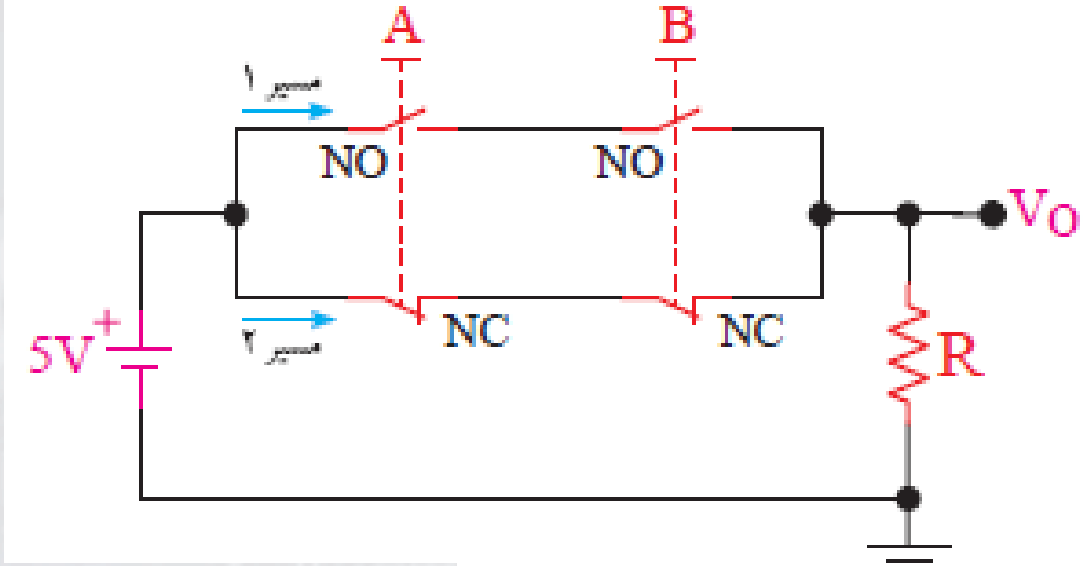






دروازه های منطقی ترکیبی:

دروازه منطقی XNOR



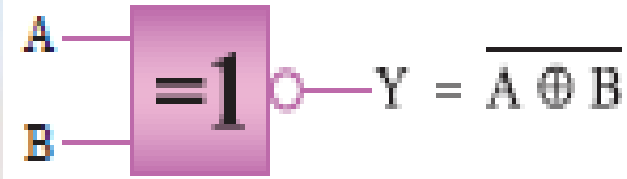
A	B	Y
۰	۰	۱
۰	۱	۰
۱	۰	۰
۱	۱	۱

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

خروجی — ورودی — ورودی

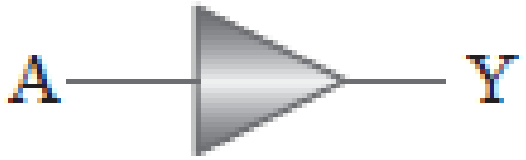
$$Y = \overline{A}\,\overline{B} + A\,B$$

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



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

دروازه منطقی Buffer



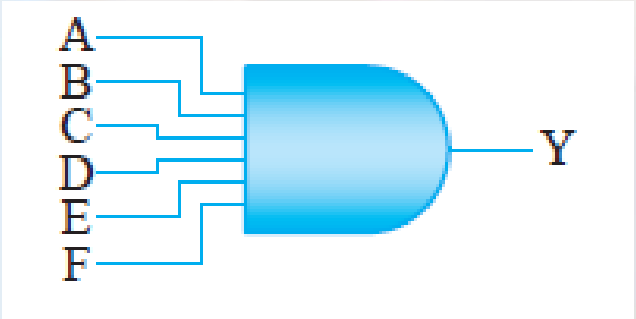
A	Y
۰	۰
۱	۱

مشخصات ویژۀ درواز ه های منطقی

مشخصات خانوادۀ آ ی سی های دیجیتال معمولاً از طریق تحلیل مدار گیت های پایه ای در هر خانواده با هم مقایسه میشوند. مهمترین پارامترهای مورد ارزیابی و مقایسه در این خانواد ه ها مقادیر fan - in، fan – out حاشیۀ نویز، تأخیر در انتشار و توان تلف شده است.

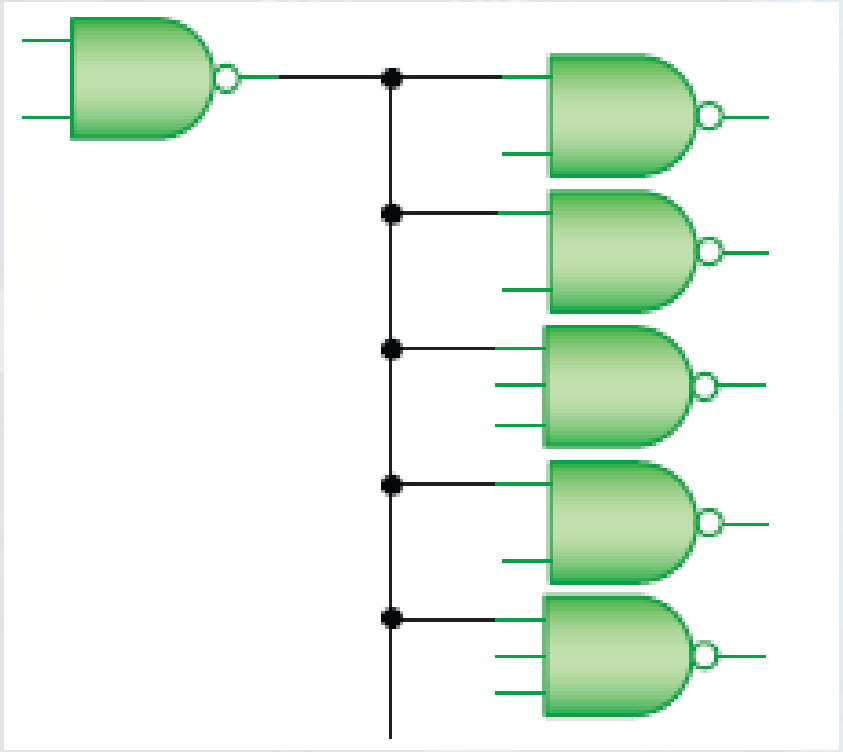
:fan - in

حداکثر تعداد ورودی که یک گیت منطقی می تواند قبول کند را fan – in آن گیت می گویند

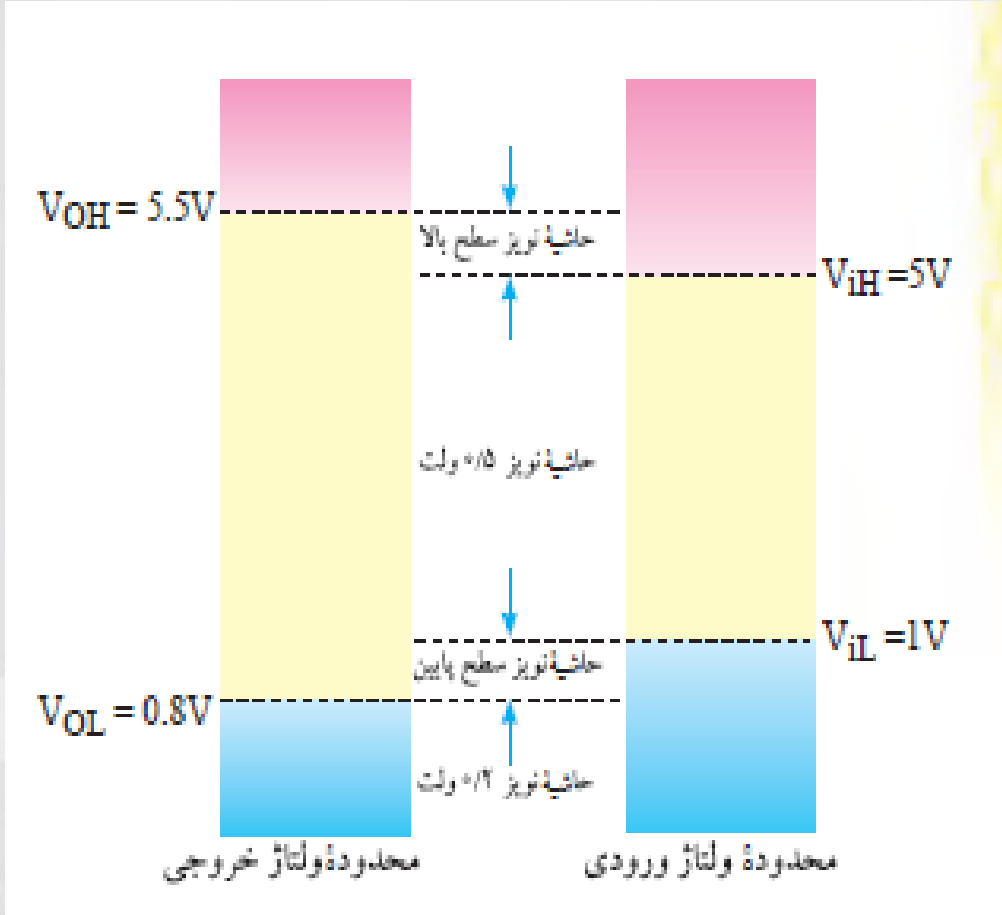


fan-out

حداکثر تعداد گی تهایی که میتواند از طریق خروجی یک گیت تغذیه شود را fan – out می گویند.

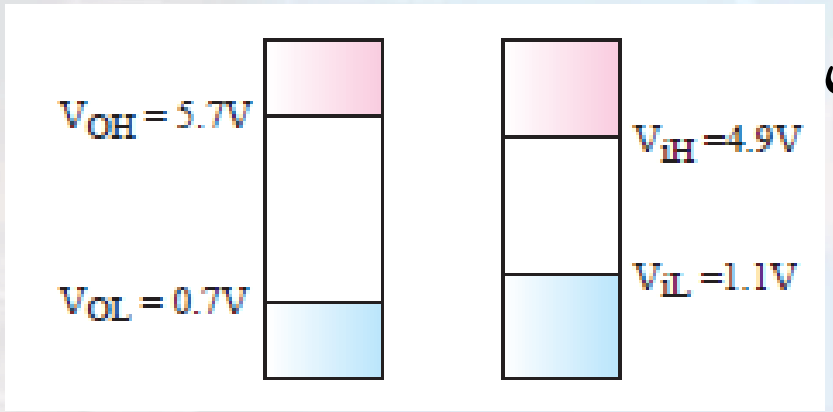


حاشیهٔ نویز



مدر یک گیت منطقی، تأثیر دامنهٔ نویز در ورودی مدار منطقی است. به عبارت دیگر میزان امنیتی است که با ظاهر شدن نویز (هر نوع ولتاژ ناخواسته)در ورودی یک مدار منطقی، بتوانیم اطلاعات را بدون خطا انتقال دهیم و دریافت کنیم. به عبارت دیگر اگر دامنهٔ ولتاژ ناخواسته بیشتر از حاشیهٔ نویز تعریف شده باشد، موجب تغییر وضعیت مدار شده و خروجی نادرست را نتیجه می دهد.

مثال: حاشیهٔ نویز سطح بالاو سطح پایین را در شکل روبرو را بدست آورید؟



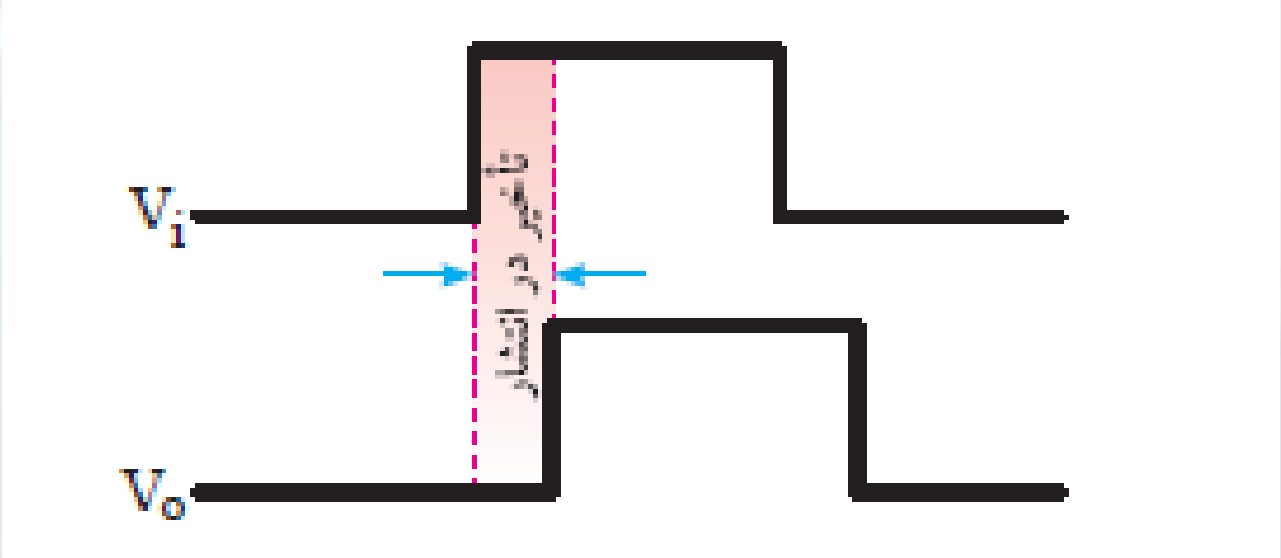
حاشیه نویز سطح بالا ۵/۷-۴/۹ = ۰/۸

حاشیه نویز سطح پایین ۱/۱-۰/۷ = ۰/۴

تأخیر در انتشار

تاخیر در انتشار عبارت است از زمانی که خروجی یک دروازهٔ منطقی لازم دارد تا تغییر ورودی را از یک حالت به حالت دیگر ظاهر نماید، به عبارت دیگر هر چه تأخیر کمتر باشد سرعت انتقال اطلاعات بیشتر می شود

هر چه تعداد گیت ها کمتر باشد تأخیر در انتشار کمتر است.



توان تلف شده

مقدار توانی که در هر گیت به صورت حرارت تلف می شود را توان تلف شدهٔ آن گیت میگویند و مقدار توان تلف شده برحسب میلی وات انداز ه گیری میشود.

استفاده از data book

معمولاً در ساخت درواز ه های منطقی که عملاً به صورت آ ی سی در اختیار ما قرار می گیرند، از ترانزیستورهای معمولی یا ترانزیستورهای MOSFET استفاده می شود.

اگر د ر یک آی سی از فناوری ترانزیستورهای معمولی استفاده شود، نام آن با حروف SN74 آغاز می شود

شماره هایی که بعد از عدد ۷۴ می آید نوع دروازۀ منطقی یا مدارهای منطقی دیگر را مشخص می کند.

با مراجعه به کتاب های اطلاعات آی سی TTL Data Book میتوان به نوع آی سی پی برد. س

اگر در ساخت آ ی سی مدارهای منطقی از فناوری ترانزیستورMOSFET استفاده شده باشد نام آی سی با حروف CD40 شروع می شود. شماره هایی که بعد از عدد ۴۰ قرار می گیرند، نوع دروازۀ منطقی یا مشخصه های دیگر آن را مشخص می کند.

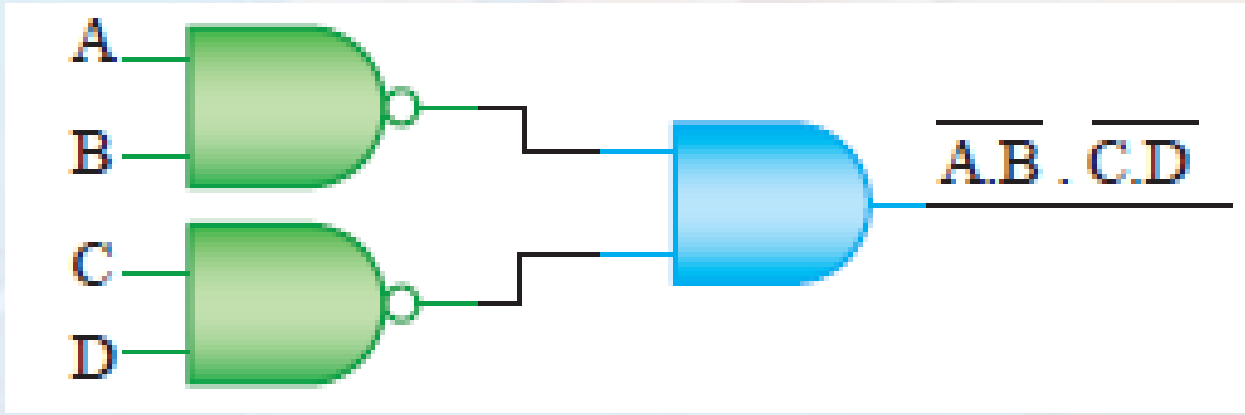
ولتاژ تغذیۀ آی سی های سری TTL از ۴/۷۵ ولت تا ۵,۲۵ ولت و ولتاژ آی سی های سری CMOS از ۳ تا ۱۵ ولت است

ولتاژ سطح منطقی یک در این نوع آی سی ها، حدود ولتاژ تغذیۀ آن است.

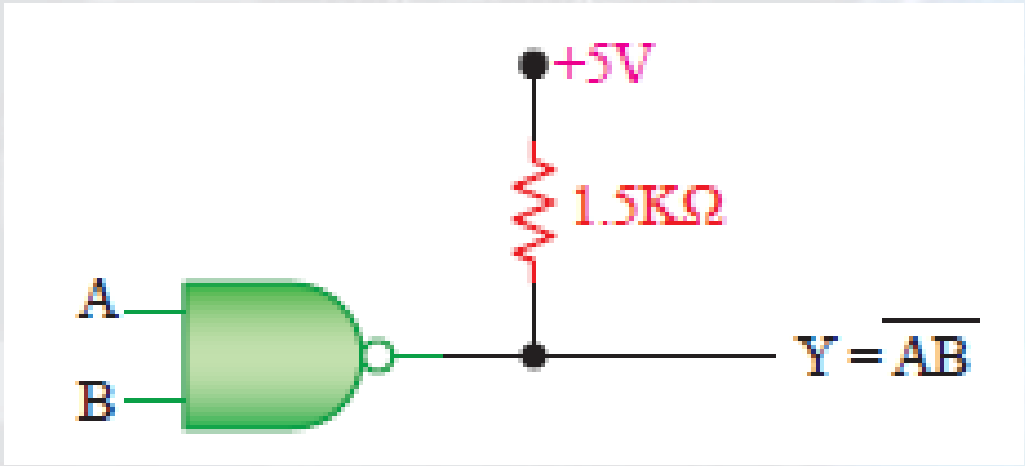
استفاده از data book

خروجی آی سی های سری TTL دو نوع معمولی Totem pole و کلکتور باز Open Collector ساخته می شود.

در نوع معمولی مدار را به همان صورتی که طرح کرده ایم می توانیم بسازیم برای مثال برای ساختن مدار شکل زیر در عمل سه دروازهٔ منطقی استفاده شده است



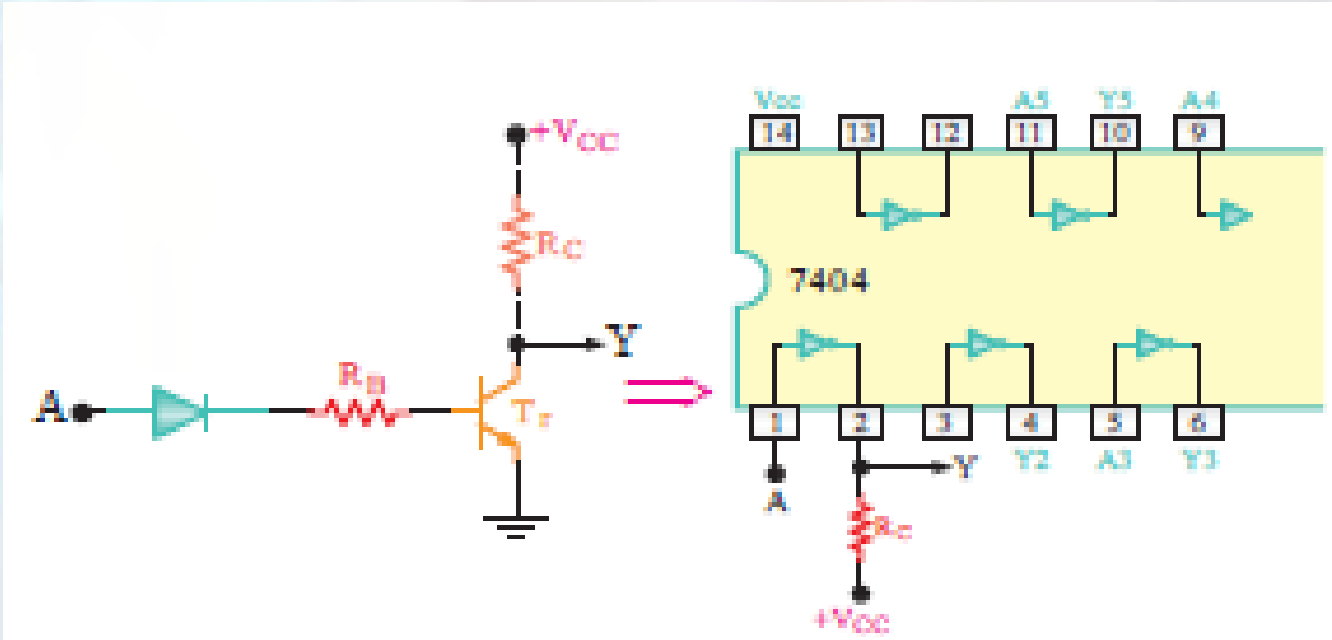
در آی سی های نوع کلکتور باز خروجی دروازه های منطقی را با یک مقاومت حدود ۱ کیلو اهم به +vcc متصل می کنیم



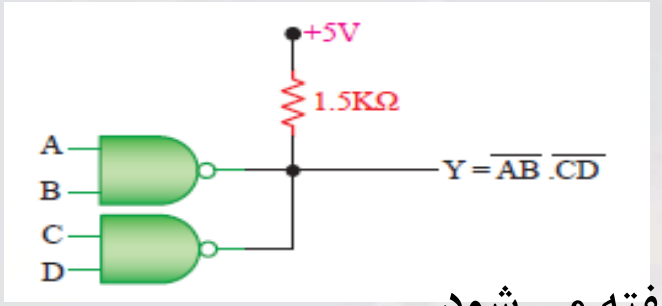
ساختمان داخلی گیت NOT به صورت زیر است:

RC که از خارج به آی سی متصل می شود را مقاومت pull up می گویند

Pull up به معنی بالا کشیدن است ودر اینجا به معنی کامل کردن مدار داخلی به کار می رود..



هنگام تعویض یک دروازهٔ منطقی معیوب با یک دروازهٔ منطقی سالم باید به نوع دروازه)معمولی یا کلکتور باز(توجه کنید. زیرا شکل ظاهری و نماد هر دو نوع آی سی مشابه است. معمولاً در کتاب های مرجع برای هر شمارهٔ آی سی، نوع معمولی یا کلکتور بازبودن را مشخص می کنند. در اغلب این کتاب ها فرض را ب ر این م یگیرند که همهٔ آی سی ها معمولی اند و فقط آی سی های کلکتور باز را مشخص می کنند. یکی از مزایای آی سی های کلکتور باز این است که اگر خروجی آنها را به یکدیگر وصل کنیم، مانند دروازهٔ منطقی AND عمل میکنند.



به این نوع AND، AND سیمی گفته می شود.
توجه داشته باشید که دروازه های معمولی را به هیچ عنوان نباید به یکدیگر متصل کنید.

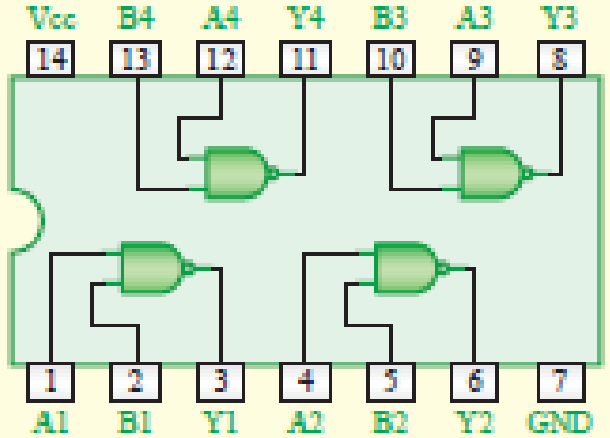
Data sheet

دروازه های منطقی پایه را معمولاً به صورت آی سی می سازند و غالباً در هر آی سی ۲ تا ۶ دروازهٔ منطقی قرار می گیرند. اطلاعات مربوط به IC ها را معمولا در data sheet قرار می دهند.

Symbol	Parameter	Min	Typ	Max	Unit
VCC	Supply Voltage	4.75	5.0	5.25	V
TA	Operating Ambient Temperature Range	0	25	70	°C
IOH	Output Current - High			- 0.4	mA
IOL	Output Current - Low			8.0	mA

علامت	مشخصه	کمترین	نوع	بیشترین	واحد
V_{CC}	ولتاژ تغذیه	۴/۷۵	۵/۰	۵/۲۵	V
T_A	حد محدودهٔ دمای عملکرد	۰	۲۵	۷۰	°C
I_{OH}	جریان بالای خروجی			— ۰/۴	mA
I_{OL}	جریان پایین خروجی			۰/۸	mA

SN74LS00
Quad 2-Input NAND Gates



Data sheet

Device	Package
SN74LS00N	Pin Dip 14
SN74LS00D	14Pin

قطعه	بسته‌بندی
SNV۴LS۰۰N	۱۴ پایه دو ردیفه
SNV۴LS۰۰D	۱۴ پایه

SOIC مدارمجتمع با پایه‌های کوچک

D Suffix با پسوند D

Case 751A شماره‌بدنه ۷۵۱A

SOIC: Small Outline Integrated Circuit

Plastic پلاستیکی

N Suffix با پسوند N

Case 646 شماره‌بدنه ۶۴۶

آشنایی با سری خانوادهٔ TTL :

۱- تراشه های TTL استاندارد – Std TTL

در دسترس ترین، ارزان ترین و در عین حال متنوع ترین نوع آی سی هاست. این آی سی ها دارای تأخیر انتشار حدود ۱۰ نانو ثانیه توان مصرفی هر دروازه حدود ۱۰ میلی وات است.

۲- تراشه های TTL شاتکی پیشرفته – AS TTL

برای سرعت های بسیار بالا ساخته شده است که این افزایش سرعت میزان جریان مصرفی آن را زیاد کرده است تأخیر انتشار در این نوع آی سی ها حدود ۱,۵ نانو ثانیه و توان مصرفی هر دروازه حدود ۲۲ میلی وات است

۳- تراشه های TTL شاتکی ک ممصرف پیشرفت– ALS TTL

شبیه سریLS است ولی در فرایندهای ساخت آن از فناوری پیشرفته تری استفاده شده است. ضریب تقویت بالا، تأخیر انتشار حدود ۴ نانو ثانیه توان مصرفی هر دروازه حدود ۱میلی وات است.

۴- تراشه های TTL شاتکی سریع– F TTL

از نظر سرعت و توان مصرفی مانند سری AS است

۵-تراشه های TTL توان بالا– H TTL

به علت جریان مصرفی بسیار بالا از رده خارج شده است.

۶-تراشه های TTL شاتکی توان پایین – L TTL

به علت سرعت پایین از رده خارج شده است.

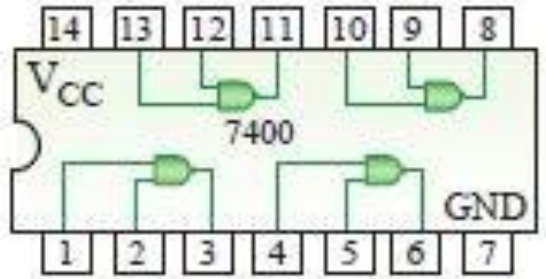
۷-تراشه های TTL شاتکی کم مصرف – LS TTL

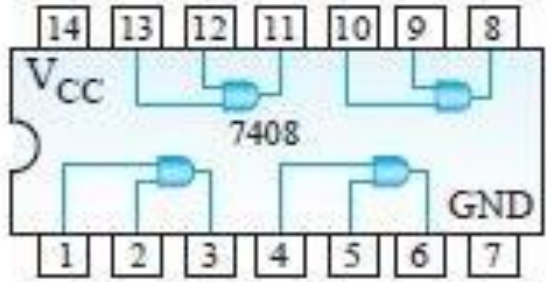
نسبت به سری استاندارد سرعت بیشتری دارد و توان مصرفی آن ۱/۵ برابر نوع استاندارد و شبیه سری ALS است

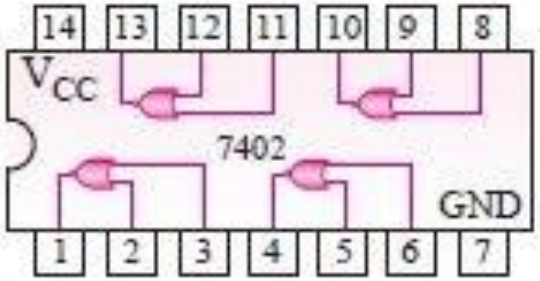
۸-تراشه های TTL شاتکی کم مصرف – S TTL

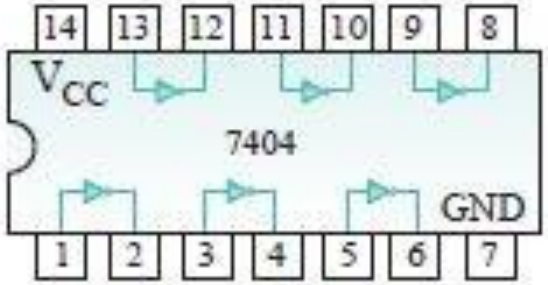
نوع اصلاح شده با سرعت بالا و مصرف پایین است.

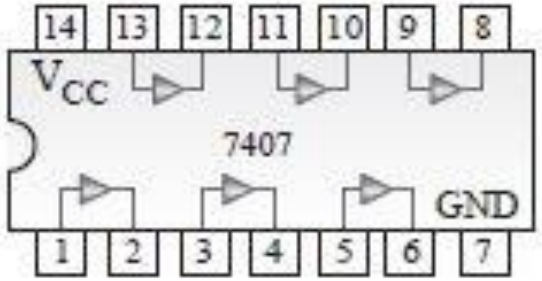
آشنایی با سری خانوادهٔ TTL :

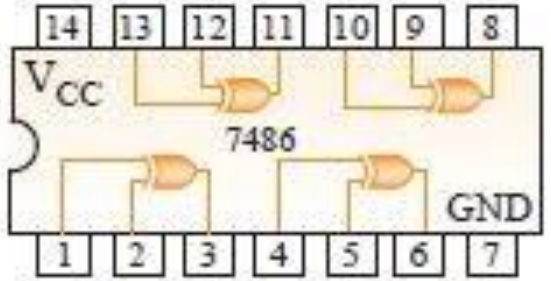




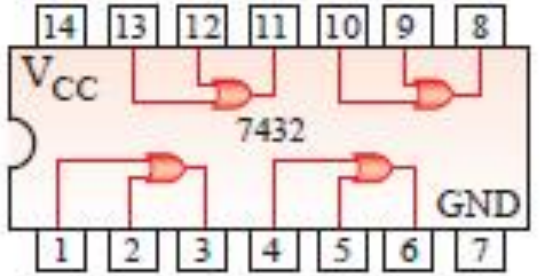








شکل ظاهری آی سی



جبر بول: