مبانی دیجیتال (مدارهای منطقی)

saber norouzi
info836saber@gmail.com
info836saber@yahoo.com

برنامه ترم :

- فصل ۱ : ۳ جلسه
- میان ترم ۱ : ۴ نمره

- فصل ۲ : ۱ جلسه
- فصل ۳ : ۲ جلسه
- میان ترم ۲ : ۴ نمره

- فصل ۴ : ۳ جلسه
- میان ترم ۳ : ۴ نمره

- فصل ۵ : ۲ جلسه
- فصل ۶ : ۱ جلسه
- میان ترم ۴ : ۴ نمره

- فصل ۷ : ۱ جلسه
- فصل ۸ : ۱ جلسه
- فصل ۹ : ۲ جلسه
- فایнал : ۶ نمره

فصل اول : مفهوم دیجیتال و سیستم اعداد

مفهوم دیجیتال:

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

سامانه‌های پایه‌ریزی شده بر مبنای سیگنال‌های پیوسته را سامانه های آنالوگ می نامند.

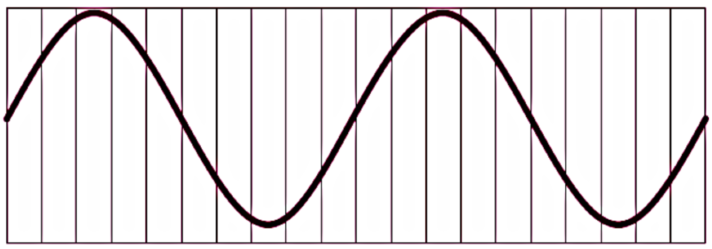
بعضی از ساعت‌هایی که زمان را به وسیله ی عقربه‌های ساعت، دقیقه و ثانیه‌شمار نشان می‌دهند و حرکتی پیوسته دارند، (نه حرکتی که عقربه های ثانیه شمار یک ثانیه، یک ثانیه پرش دارد) مثالی از یک وسیله آنالوگ است .

دستگاه سنجش فشارخون عقربه‌ای، میزان فشار خون را به صورت آنالوگ نشان می دهد که در این حالت نیاز به همراهی شخص دیگری است. در صورتی که بیمار به تنهایی می‌تواند دستگاه سنجش فشارخون دیجیتالی را مورد استفاده قرار دهد. همچنین او احتیاج به مهارت خاص برای اندازگیری فشارخون توسط دستگاه دیجیتالی ندارد.

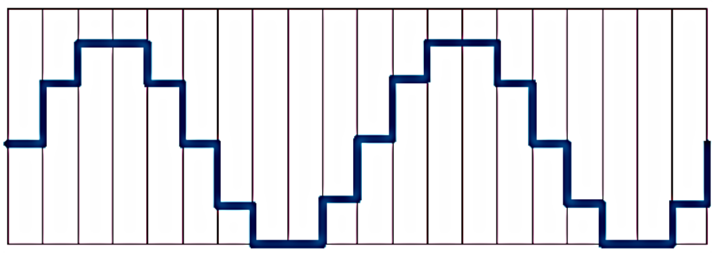
اطلاعات روی نوارهای کاست صوتی به صورت آنالوگ ذخیره می شوند در حالی که دیسک‌های (CD) لیزری فشرده، اطلاعات را به صورت دیجیتال ذخیره می کنند.

در شکل‌های زیر ، اطلاعات ذخیره شده بر روی نوار مغناطیسی را به صورت آنالوگ و دیجیتال مشاهده می‌کنید .

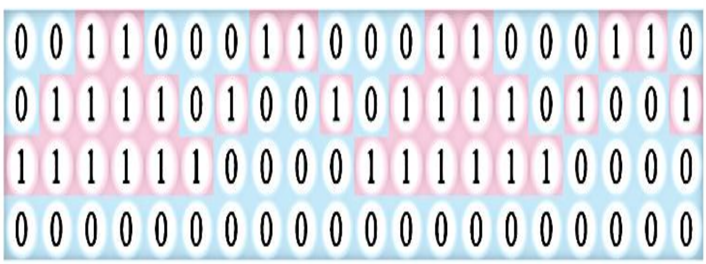
الف) فرم آنالوگ :



ب) فرم آنالوگ نمونه برداری شده از یک سیگنال سینوسی :



ج) فرم دیجیتال نمونه برداری شده از یک سیگنال سینوسی :



مزایای سیستم های دیجیتال نسبت به آنالوگ :

۱- در سیستم‌های دیجیتالی، قابلیت انعطاف پذیری بیشتر است.

۲- در سیستم‌های دیجیتالی، سرعت بالاتر است.

۳- در سیستم های دیجیتالی، دقت و وضوح بالاتر است.

۴- در سیستم های دیجیتالی، بازیابی اطلاعات آسان‌تر است.

۵- در سیستم های دیجیتالی، ذخیره اطلاعات آسان‌تر است.

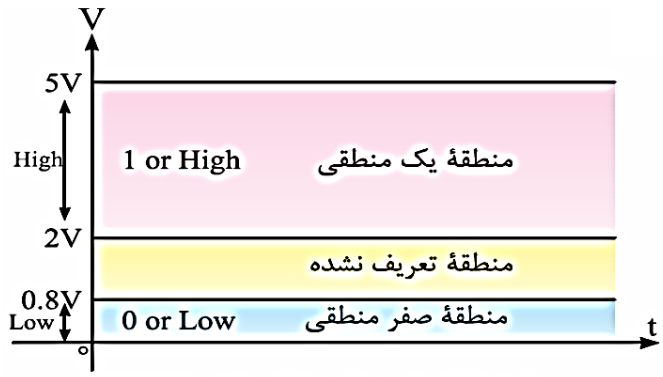
۶- در سیستم های دیجیتالی، ابعاد سیستم‌ها کاهش می‌یابند.

۷- در سیستم های دیجیتالی، می‌توانیم از پردازشگرهای قویتر، مانند کامپیوتر، استفاده کنیم.

مفهوم صفر و یک منطقی:

در سیستم‌های دیجیتال ، ولتاژ حدود صفر ولت (عملاً از ۰ تا ۰/۸ ولت) به منزله صفر منطقی ؛ و ولتاژ حدود ۵ ولت (عملاً از ۲ تا ۵ ولت) به منزله یک منطقی در نظر گرفته می‌شود .

هر چند سطوح صفر و یک منطقی ممکن است در سیستم‌های گوناگون با یکدیگر تفاوت داشته باشد اما ولتاژهای حوالی صفر ولت و ۵ ولت از بقیه رایج تر است .



سطوح ولتاژ صفر و یک منطقی رایج

سیستم‌های اعداد :

سیستم ده‌دهی یا اعشاری (Decimal) :

سیستم ده‌دهی یا اعشاری از ده رقم ۰ و ۱ و ۲ و ..... و ۹ تشکیل شده‌اند. موقعیت مکانی هر رقم معنی خاصی دارد.

مثلاً :

$$3296 \; = \; 3000 + 200 + 90 + 6 = 3 \times 10^3 + 2 \times 10^2 + 9 \times 10^1 + 6 \times 10^0$$

به طور کلی، در سیستم اعشاری (ده‌دهی) هر عدد صحیح را می توان به صورت زیر نوشت.

$$N = a_n \times 10^n + a_{n-1} \times 10^{n-1} + \; \dots \dots \; + a_2 \times 10^2 + a_1 \times 10^1 + a_0 \times 10^0$$

ارقام a_n , a_{n-1} , ... , a_2 , a_1 و a_0 می توانند بین صفر تا ۹ باشند. توان‌های ۱۰ ، ارزش مکانی هر یک از رقم‌ها را مشخص می‌کند. مثلاً :

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

یکان　　دهگان　　صدگان　　هزارگان　　ده هزارگان

سیستم دودویی یا باینری (Binary) :

در سیستم دودویی هر عدد را می توان به صورت زیر نوشت:

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

دراین جا ارقام a_n , a_{n-1} , ... , a_2 , a_1 و a_0 می توانند 0 , 1 باشند.

در سیستم باینری به هر رقم صفر یا یک، یک بیت (Binary Digit = Bit) می‌گویند، مثلاً عدد ۱۱۰۱ یک عدد چهار بیتی است.

به هر چهار بیت یک نیبل (Nibel) ، و به هر هشت بیت یک بایت (Byte) گفته می‌شود. واحد بزرگتر از بایت ، کیلوبایت معادل $۲^{۱۰}$ بایت یا ۱۰۲۴ بایت و مگابایت معادل $۲^{۲۰}$ بایت یا ۱۰۲۴ کیلوبایت است.

نیبل = ۴ بیت = xxxx

بایت = ۸ بیت = xxxxxxxx

1 k byte = 2^{10} byte = $2^{10} * 2^3$ bit = 2^{13} bit

1 MEGA byte = 2^{10} K byte = 2^{20} byte = 2^{23} bit

1 GIGA byte = 2^{10} MEGA byte = 2^{20} k byte = 2^{30} byte = 2^{33} bit

1 TERA byte = 2^{10} GIGA byte = 2^{20} MEGA byte = 2^{30} k byte = 2^{40} byte = 2^{43} bit

تبدیل از مبنای ۲ به ۱۰:

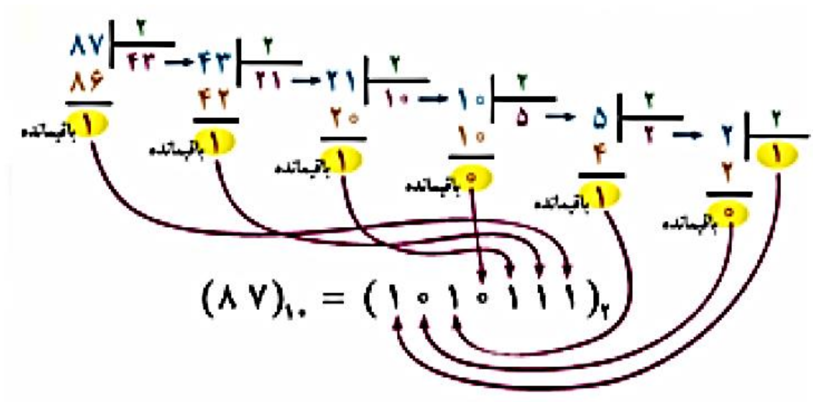
برای این کار کافی است هر رقم را در ارزش جایگاهش ضرب کرده، و جواب‌ها را باهم جمع کنیم. مثلاً :

$$10011 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (19)_{10}$$

$$(1010.011)^2 = 2^3 + 2^1 + 2^{-2} + 2^{-3} = (10.375)_{10}$$

تبدیل از مبنای ۱۰به ۲:

برای تبدیل کردن اعداد اعشاری به باینری، می‌توانیم از تقسیمات متوالی اعداد اعشاری به عدد ۲ استفاده کنیم. برای مثال عدد اعشاری ۸۷ را به عد باینری تبدیل می کنیم.



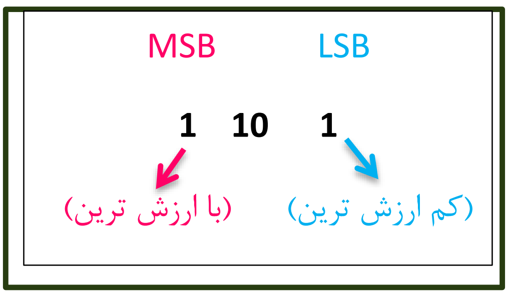
تقسیمات را تا جایی ادامه می‌دهیم تا آخرین خارج قسمت یک شود .

سپس در سمت چپ ، آخرین خارج قسمت را می‌نویسیم و به ترتیب باقی‌مانده‌های بدست آمده را در جلوی آن قرار می‌دهیم.

در یک عدد باینری (مثلاً ۱۱۰۱۱۱) بیت اول از سمت راست کم ارزش‌ترین بیت است که به آن LSB (Least siginificant Bit) می‌گویند.

به آخرین بیت در سمت چپ که با ارزش‌ترین بیت است MSB (Most siginificant Bit) گفته می شود.

توجه داشته باشید که ارزش ارقام دقیقاً مشابه سیستم اعشاری است.



اعداد اعشاری و معادل باینری

باینری	ده دهی اعشاری
0	0
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

سیستم هشت‌تایی یا اُکتال (Octal) :

در سیستم اُکتال (هشت‌تایی) مبنا عدد ۸ ، و ارقام به صورت (۰ و ۱ و ۲ و ۰۰۰ و ۷) است .

تبدیل از مبنای ۸ به ۱۰:

برای این کار کافی است هر رقم را در ارزش جایگاهش ضرب کرده، و جواب‌ها را با هم جمع کنیم.

$$N = a_n \times ۸^n + a_{n-۱} \times ۸^{n-۱} + \dots + a_۲ \times ۸^۲ + a_۱ \times ۸^۱ + a_۰ \times ۸^۰$$

مثلاً:

$$(۵۲۳۶)_۸ =$$

$$۵ \times ۸^۳ + ۲ \times ۸^۲ + ۳ \times ۸^۱ + ۶ \times ۸^۰ =$$

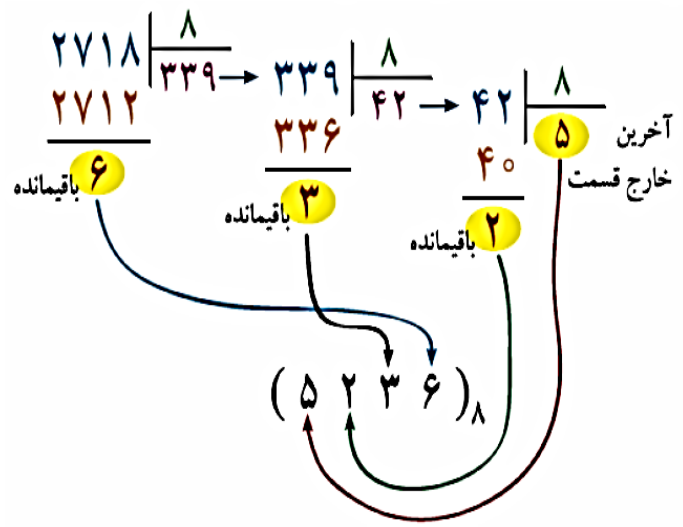
$$۵ \times ۵۱۲ + ۲ \times ۶۴ + ۳ \times ۸ + ۶ \times ۱ =$$

$$۲۵۶۰ + ۱۲۸ + ۲۴ + ۶ = (۲۷۱۸)_{۱۰}$$

تبدیل از مبنای ۱۰ به ۸:

برای تبدیل کردن اعداد اعشاری به مبنای ۸ ، می‌توانیم از تقسیمات متوالی اعداد اعشاری به عدد ۸ استفاده کینم. تقسیمات را تا جایی ادامه دهیم که خارج قسمت با عدد ۷ مساوی یا کوچکتر شود. مشابه سیستم باینری از سمت چپ شروع به نوشتن عدد می کنیم.

مثال: عدد اعشاری ۱۰(۲۷۱۸) را به عدد اُکتال تبدیل کنید (به مبنای ۸ ببرید).



سیستم شانزده‌تایی یا هگزادسی‌مال (**Hexa decimal**) :

در این سیستم (۱۶تایی) ، ارقام شامل ۰و۱و۲و......۹، E,D,C,B,A و F هستند . در این سیستم برای 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 می‌شود. به همین دلیل از حروف استفاده می‌شود که :

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

تبدیل از مبنای ۱۶ به ۱۰:

برای این کار کافی است هر رقم را در ارزش جایگاهش ضرب کرده، و جواب‌ها را با هم جمع کنیم.

$$N = a_n \times 16^n + a_{n-1} \times 16^{n-1} + \ldots\ldots + a_2 \times 16^2 + a_1 \times 16^1 + a_0 \times 16^0$$

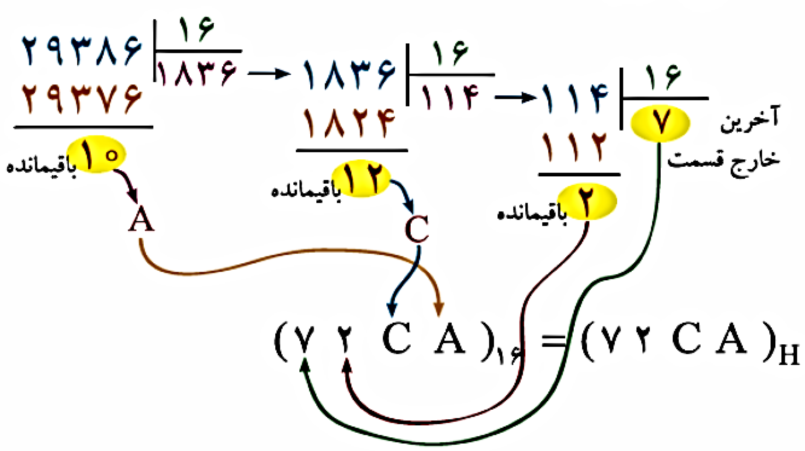
مثلاً عدد $(A14E)_{16}$ در مبنای ۱۶ نوشته شده است. معادل اعشاری آن برابر است با :

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

$N= (10)\times4096 + 1\times256 + 4\times16 + (14)\times1 = 40960 + 256 + 64 + 14 = (41294)_{10}$

تبدیل از مبنای ۱۰ به ۱۶:

برای تبدیل کردن اعداد اعشاری به هگزادسیمال ، می‌توانیم از تقسیمات متوالی عدد اعشاری به عدد ۱۶ استفاده کنیم . تقسیمات را تا جایی ادامه می دهیم که خارج قسمت با عدد ۱۶ مساوی یا کوچکتر شود. مشابه سیستم باینری از سمت چپ شروع به نوشتن عدد می کنیم .



تبدیل از مبنای m به n:

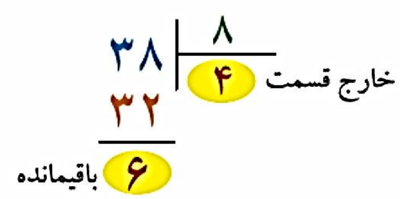
برای این کار ابتدا عدد را از مبنای m به مبنای ۱۰برده ، سپس مبنای ۱۰ را به n میبریم.

به طور مثال برای تبدیل عدد باینری ۱۰۰۱۱۰ به مبنای اُکتال به روش زیر عمل می کنیم.

مرحله اول تبدیل به مبنای دسی‌مال :

$$(100110)_B = 1 \times 320 + 0 \times 16 + 0 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1 = 32\ +\ 4\ +\ 2 = (38)_D$$

مرحله ی دوم تبدیل عدد اعشاری به مبنای اُکتال :



تبدیلات سریع :

تبدیل سریع از مبنای ۲ به ۱۰ و برعکس :

ابتدا ارزش هر جایگاه را نوشته ، و سپس سعی می‌کنیم عدد مورد نظر را ایجاد کنیم.

	۸	۴	۲	۱
(۱۳)۱۰ =	۱	۱	۰	۱

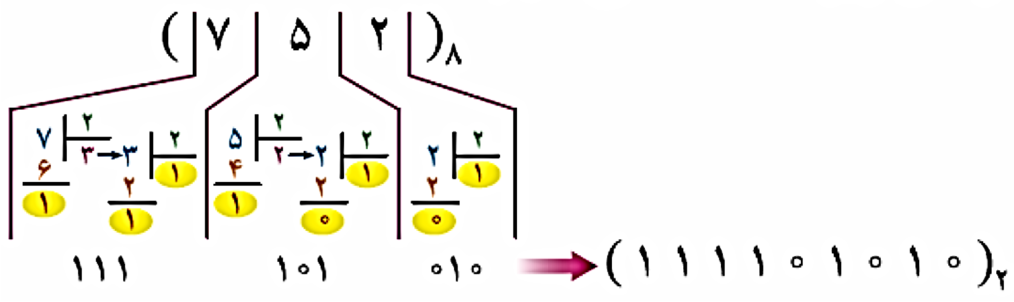
	۶۴	۳۲	۱۶	۸	۴	۲	۱
(۱۰۰)۱۰ =	۱	۱	۰	۰	۱	۰	۰

۲۵۶	۱۲۸	۶۴	۳۲	۱۶	۸	۴	۲	۱	
۱	۰	۰	۱	۱	۱	۰	۰	۱	= (۳۱۳)۱۰

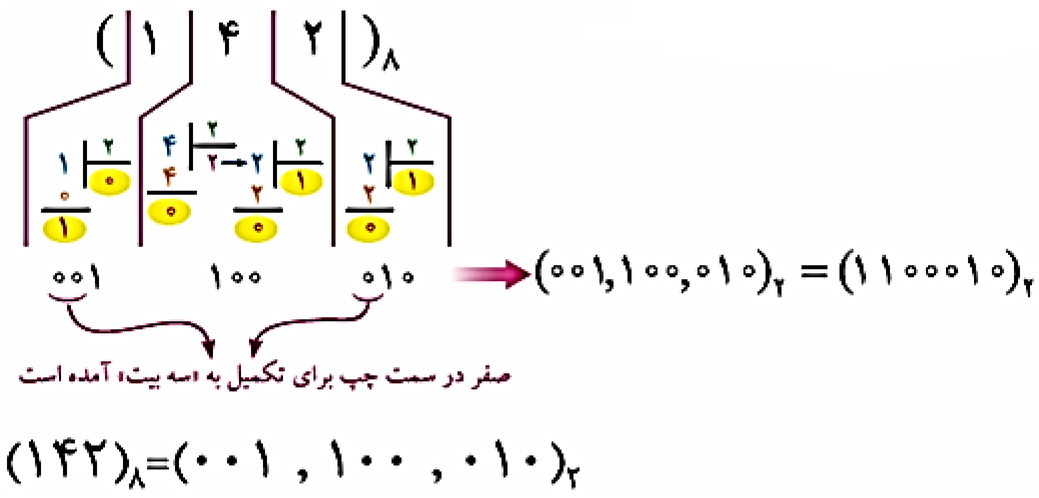
تبدیل سریع از مبنای ۸ به ۲:

از آنجاییکه $8 = 2^3$ است ، هر رقم در مبنای اُکتال معادل سه بیت باینری است. بنابراین هر رقم در مبنای اُکتال را به یک عدد سه بیتی تبدیل می کنیم. برای این کار می توان از روش تقسیم‌های متوالی استفاده کرد . با تمرین زیاد به راحتی می‌توانید معادل باینری هر عدد را بدون استفاده از محاسبات به دست آورید.

به طور مثال برای تبدیل عدد $(۷۵۲)_۸$ به مبنای ۲ از مراحل زیر استفاده می‌کنیم.



مثال: عدد $(۱۴۲)_۸$ را از روش سریعتر به مبنای دودویی تبدیل کنید.



تبدیل سریع از مبنای ۲ به ۸ :

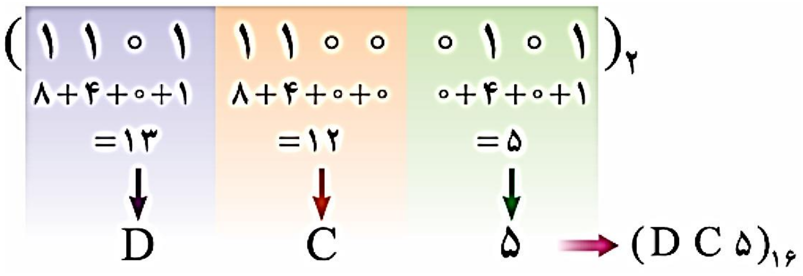
بنابر آنچه گفته شد می‌توان عدد باینری را از سمت راست سه‌بیت، سه‌بیت جدا کنیم و معادل هر قسمت آن را به صورت اُکتال بنویسیم. به طور مثال :

$$(\,1001110010011\,)_B = (\,1\ \ 001\ \ 110\ \ 010\ \ 011\,)_B = (\,1\,1\,6\,2\,3\,)_O$$

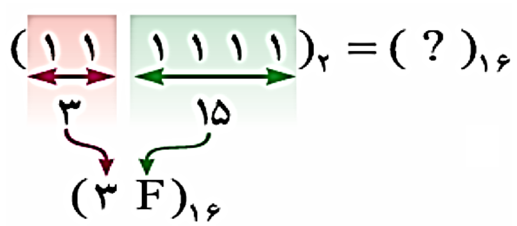
$$(\,10011000110111\,)_B = (\,10\ \ 011\ \ 000\ \ 110\ \ 111\,)_B = (\,2\,3\,0\,6\,7\,)_O$$

تبدیل سریع از مبنای ۲ به ۱۶:

از آنجاییکه ۲۴ = ۱۶ است می توان عدد باینری را از سمت راست چهار بیت، چهار بیت جدا کینم و معادل هر قسمت آن را به صورت شانزده تایی بنویسیم. به طور مثال:



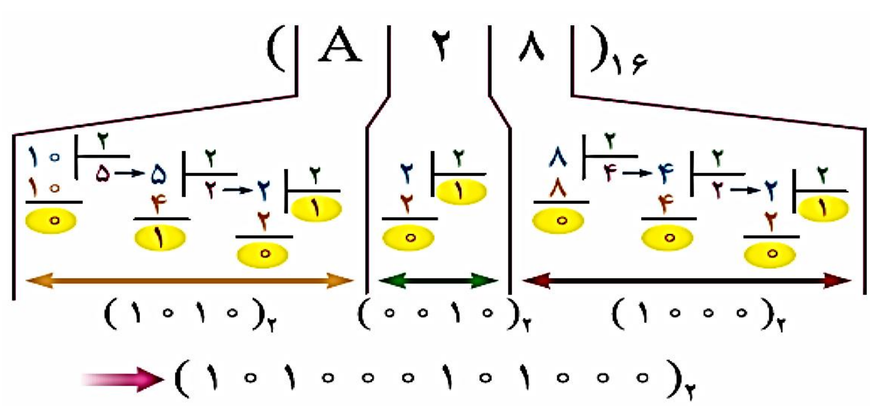
مثال: عدد ۲(۱۱۱۱۱۱) را به مبنای هگزا دسی مال ببرید، (از روش ساده و سریع)



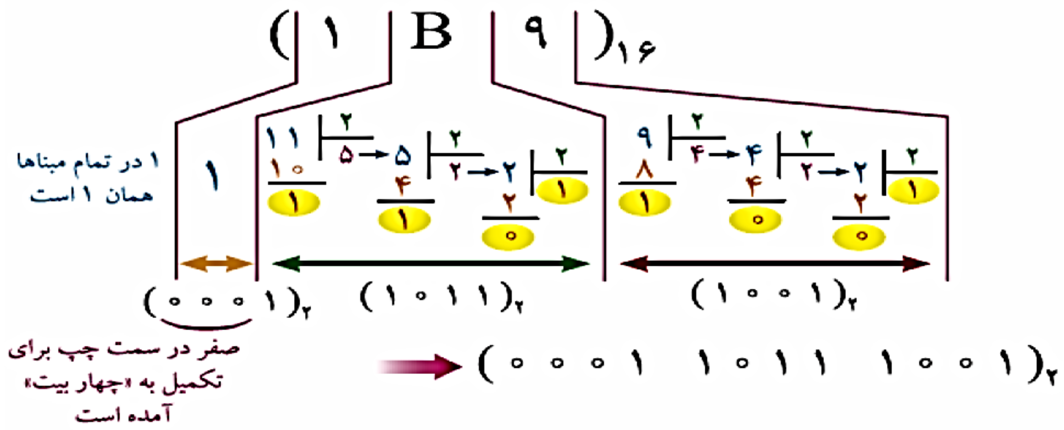
تبدیل سریع از مبنای ۱۶ به ۲:

از آنجاییکه ۲۴= ۱۶ است ، هر رقم در مبنای هگزادسی‌مال را به یک عدد چهاربیتی در مبنای باینری تبدیل می‌کنیم. برای این کار می توان از روش تقسیم‌های متوالی استفاده کرد. با تمرین فراوان، به راحتی می توانید معادل باینری هر عدد را بدون استفاده از محاسبات به دست آورید.

به طور مثال برای تبدیل عدد ۱۶(A۲۸) به مبنای ۲ از مراحل زیر استفاده می کنیم.



مثال: عدد $(۱B۹)_{۱۶}$ را از روش سریعتر به مبنای دودویی تبدیل کیند.



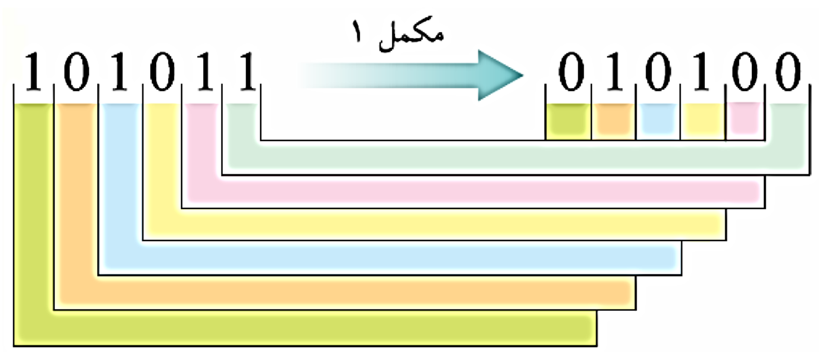
مکمل۱:

برای به دست آوردن مکمل ۱ در هر عدد دودویی (باینری) کافی است صفرها را یک و یک‌های آن را به صفر تبدیل کینم. مثلاً برای عدد باینری ۰۱۱۱۰۱ مکمل یا متمم یک آن به صورت ۱۰۰۰۱۰ خواهد شد.

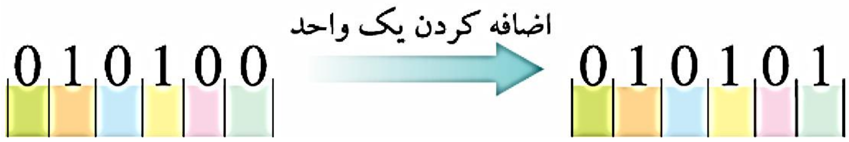
مکمل۲:

برای این کار ۲روش وجود دارد. روش اول : برای به دست آوردن مکمل ۲ در هر عدد باینری ، اولین رقم یک از سمت راست و صفرهای ماقبلش را ثابت نوشته ، و بقیه بیت ها را متمم می‌کنیم 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 می‌نویسیم سپس دومین یک از سمت راست را صفر میکنیم و بعد صفر را یک کرده و آن را می‌نویسیم.

روش دوم : ابتدا مکمل ۱ عدد باینری را می‌نویسیم ، 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.



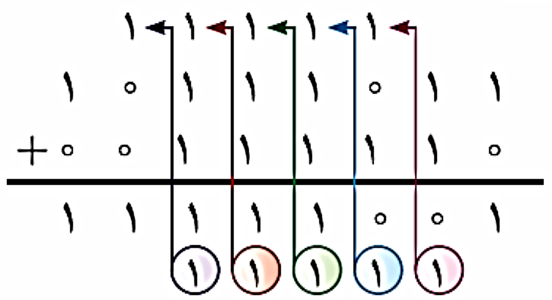
در این روش همانطور که ملاحظه شد یک‌ها را به صفر و صفرها را به یک تبدیل می‌کنیم که ابتدا مکمل ۱ عدد به دست می‌آید. سپس به مکمل ۱ عدد به دست آمده یک واحد اضافه می شود که این روش را پس از فراگیری جمع در سیستم باینری بهتر درک خواهید کرد.



جمع در سیستم باینری :

جمع در این سیستم ، شبیه به جمع در سیستم اعشاری است. در سیستم اعشاری ، هر گاه جمع دو رقم از ده بیشتر می‌شود، یک واحد به رقم بعد آن اضافه می کنیم که به آن **ده بر یک** می گوییم. در سیستم باینری، هر گاه جمع دو رقم دو شود (حالت ۱+۱)، ایجاد **دو بر یک** می کند و باید عدد یک را به رقم بعدی اضافه کرد.

مثال:

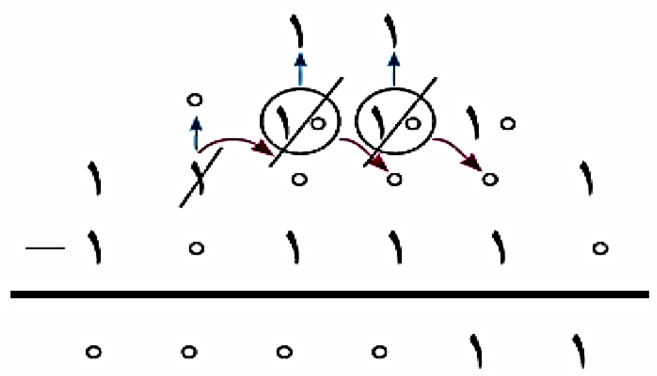


$(10111011)_2+(111110)_2=(11111001)_2$

تفریق باینری:

تفریق یک عدد باینری از عدد باینری دیگر با روشی مانند تفریق اعداد دسی‌مال انجام می‌پذیرد، یعنی اگر رقم بزرگتر از رقم کوچکتر کم شود یک واحد از مکان بعدی قرض گرفته می‌شود و در مکانی که یک واحد قرض گرفته شده یک را به صفر تبدیل می‌کنیم. واحد قرض گرفته شده را Borrow می گویند.

مثال: دو عدد باینری را از هم کم کنید.



نکته: اگر یک عدد بزرگتر از کوچکتر کم شد، یک منفی می‌گزاریم، و جای دو عدد را عوض می کنیم.

مثال: حاصل عبارت $(101110)_2 - (110001)_2$ را بیابید .

ابتدا جای اعداد را عوض می‌کنیم و حاصل را می‌یابیم :

$$(110001)_2 - (101110)_2 = (000011)_2$$

پس می‌توان گفت :

$$(101110)_2 - (110001)_2 = -(000011)_2$$

نکته : برای درک بهتر می‌توان عملیات تفریق را در مبنای ۱۰ نیز انجام داد :

$$46 - 49 = -3$$

ضرب باینری:

نحوه ضرب دو عدد باینری را با یک مثال نشان میدهیم.

```
    ۱۰۱۱
 *
     ۱۰۱
  ------
    ۱۰۱۱
+  ۰۰۰۰
  ۱۰۱۱
  ------
  ۱۱۰۱۱۱
```

نقش کد در سیستم دیجیتال:

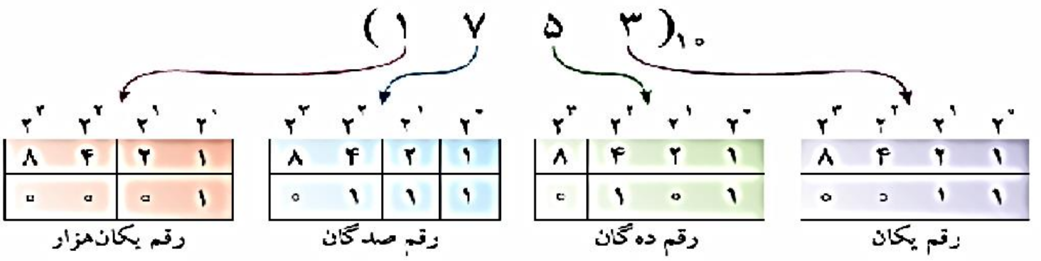
معنای واقعی"کد کردن" همان "رمز کردن" یا به صورت رمز در آوردن اطلاعات است. اولین سؤالی که مطرح می‌شود آن است که چرا باید اطلاعات را به صورت رمز در آوریم؟

لازم به ذکر است که تبدیل اطلاعات به کد ، نه فقط برای ایجاد ارتباط لازم است ، بلکه یکی از روش‌های با اهمیت در تشخیص خطا برای اطلاعات پردازش شونده در سیستم می‌باشد.

کد BCD :

بعضی از ماشین‌های محاسبه‌گر الکترونیکی عملیات ریاضی را در کد (Binary Coded Decimal) BCD انجام می‌دهند.

در کد BCD هر رقم ده‌دهی را با چهار بیت باینری معادل آن نشان می‌دهند.



نکته : کدBCD با کد باینری در ارقام بالای ۹ تفاوت دارد . به مثال زیر دقت کنید .

مثال : کدBCD و کد باینری عدد $(23)_D$ را بیابید .

$$(23)_D = (10111)_B = (00100011)_{BCD}$$

در جدول زیربرخی کدهای دیگر نشان داده شده است.

رقم دهدهی	(BCD) ۸۴۲۱	افزونی ۳-	۸۴-۲-۱	۲۴۲۱	دوپنجی ۵۰۴۳۲۱۰
۰	۰۰۰۰	۰۰۱۱	۰۰۰۰	۰۰۰۰	۰۱۰۰۰۰۱
۱	۰۰۰۱	۰۱۰۰	۰۱۱۱	۰۰۰۱	۰۱۰۰۰۱۰
۲	۰۰۱۰	۰۱۰۱	۰۱۱۰	۰۰۱۰	۰۱۰۰۱۰۰
۳	۰۰۱۱	۰۱۱۰	۰۱۰۱	۰۰۱۱	۰۱۰۱۰۰۰
۴	۰۱۰۰	۰۱۱۱	۰۱۰۰	۰۱۰۰	۰۱۱۰۰۰۰
۵	۰۱۰۱	۱۰۰۰	۱۰۱۱	۱۰۱۱	۱۰۰۰۰۰۱
۶	۰۱۱۰	۱۰۰۱	۱۰۱۰	۱۱۰۰	۱۰۰۰۰۱۰
۷	۰۱۱۱	۱۰۱۰	۱۰۰۱	۱۱۰۱	۱۰۰۰۱۰۰
۸	۱۰۰۰	۱۰۱۱	۱۰۰۰	۱۱۱۰	۱۰۰۱۰۰۰
۹	۱۰۰۱	۱۱۰۰	۱۱۱۱	۱۱۱۱	۱۰۱۰۰۰۰

هر کدام از روش‌های کدینگ فوق، خصوصیت ویژه‌ی دارند. مثلاً کد دو- پنجی ، در هر حالت تنها دو تا یک دارد ، و برای خطا سنجی مناسب است. در کد افزونی ۳ هر رقم با رقم قبلی‌اش تنها یک تفاوت دارد.

کدهای آشکار سازی خطا:

اطلاعات دودویی ممکن است از یک مکان به مکان دیگر به‌کمک وسایل ارتباطی مثل سیم‌ها یا موج‌های رادیویی انتقال یابند. هر پارازیت خارجی که وارد وسایل فیزیکی شود ارزش بیت‌ها را از ۰ به ۱ و یا برعکس تغییر می‌دهد. معمول‌ترین روش خطایابی، استفاده از توازن است. یک بیت توازن، بیتی است اضافی که جزئی از پیام است و سبب می شود که تعداد کل ۱ ها در پیام زوج یا فرد گردد اگر بیت توازن فرد انتخاب شده باشد P طوری انتخاب می گردد که مجموع ۱ ها در پنج بیت فرد باشد و در توازن زوج P طوری انتخاب شده تا مجموع همه ۱ ها زوج باشد.

مثلاً برای ارسال پیام 100011 در توازن زوج از کد 1100011 و در توازن فرد از کد 0100011 استفاده می‌شود .

کدگری (انعکاسی) :

مزیت کد گری نسبت به اعداد دودویی محض این است که وقتی از یک عدد به عدد بعدی می رویم فقط یک بیت تغییر می کند.

کد گری ۴ بیتی

معادل دهدهی	کد گری
۰	۰۰۰۰
۱	۰۰۰۱
۲	۰۰۱۱
۳	۰۰۱۰
۴	۰۱۱۰
۵	۰۱۱۱
۶	۰۱۰۱
۷	۰۱۰۰
۸	۱۱۰۰
۹	۱۱۰۱
۱۰	۱۱۱۱
۱۱	۱۱۱۰
۱۲	۱۰۱۰
۱۳	۱۰۱۱
۱۴	۱۰۰۱
۱۵	۱۰۰۰

کدهای اسکی (ASCII) :

در بسیاری از کاربردهای کامپیوترهای دیجیتال ، نه تنها نیاز به دستکاری روی داده‌های عددی ، بلکه روی حروف نیز می‌باشد. کد دودویی استاندارد برای کاراکترهای الفبا ، عددی ASCII ست. این کد از هفت بیت برای کد نمودن ۱۲۸ کاراکتر استفاده می‌کند.

کد همینگ:

جهت تشخیص و تصحیح خطا به کار می رود.

اعداد دودویی علامت دار:

معمول است که سمت چپ‌ترین بیت عدد را به علامت اختصاص می‌دهند. قرار این است که اعداد مثبت را با گذاشتن ۰ و اعداد منفی را با گذاشتن ۱ در محل مزبور نشان دهند. مثلاً :

$$(+13)_{10} = (0\ 0001101)_2 \quad , \quad (-13)_{10} = (1\ 0001101)_2$$

تمرین :

۱- کمیت های آنالوگ و دیجیتال به چه معناست؟

جواب :

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

سامانه‌های پایه‌ریزی شده بر مبنای سیگنال‌های پیوسته را سامانه های آنالوگ می نامند.

****

۲- مزایای استفاده از سیستم دیجیتال نسبت به سیستم آنالوگ چیست؟

جواب :

۱- در سیستم‌های دیجیتالی، قابلیت انعطاف پذیری بیشتر است.
۲- در سیستم‌های دیجیتالی، سرعت بالاتر است.
۳- در سیستم های دیجیتالی، دقت و وضوح بالاتر است.
۴- در سیستم های دیجیتالی، بازیابی اطلاعات آسان‌تر است.
۵- در سیستم های دیجیتالی، ذخیره اطلاعات آسان‌تر است.
۶- در سیستم های دیجیتالی، ابعاد سیستم‌ها کاهش می‌یابند.
۷- در سیستم های دیجیتالی، می‌توانیم از پردازشگرهای قویتر، مانند کامپیوتر، استفاده کنیم.

****

۳- مزایای کدکردن اطلاعات چیست؟

جواب :

۱- برای ایجاد ارتباط لازم است .
۲- یکی از روش‌های با اهمیت در تشخیص و تصحیح خطا برای اطلاعات پردازش شونده در سیستم می‌باشد.
۳- موجب به رمز درآوردن اطلاعات و محرمانه کردن آن می‌شود .

****

۴- حاصل عبارت زیر را به دست آوردید.

الف) $(۱۰۱۱۰۰۱۱۱)_{۲} = (؟)_{۱۰}$ ‌ ‌ ب) $(۱۰۱۱۰۱۱۰۰)_{۲} = (؟)_{۸} = (؟)_{۱۶}$ ‌ ‌ ج) $(۲۵۶)_{۸} = (؟)_{۲}$

د) $(۱۱۰۱۱)_{۲} + (۱۱۰۰۰)_{۲} = (؟)_{۸}$ ‌ ‌ ح) $(۱۱۰۰۰۰)_{۲} - (۱۰۱۱۰)_{۲} = (؟)_{۱۶}$ ‌ ‌ و) $(۷۷)_{۸} - (۷۷)_{۱۶} = (؟)_{۲}$

جواب :

$(101100111)_2 = (359)_{10}$

$(101101100)_2 = (554)_8 = (16C)_{16}$

$(256)_8 = (10101110)_2$

$(11011)_2 + (11000)_2 = (110011)_2 = (63)_8$

$(110000)_2 - (10110)_2 = (11010)_2 = (1A)_{16}$

$(77)_8 - (77)_{16} = (111111)_2 - (01110111)_2 = (-111000)_2$

****

۵- معادل باینری و کد BCD موارد زیر را بیابید.

الف) ۳۴۵ ب) ۲۷ ج) ۹۵۸۱ د) ۸

جواب :

$(345)_D = (101011001)_B = (0011\ 0100\ 0101)_{BCD}$

$(27)_D = (11011)_B = (0010\ 0111)_{BCD}$

$(9581)_D = (10010101101101)_B = (1001\ 0101\ 1000\ 0001)_{BCD}$

$(8)_D = (1000)_B = (1000)_{BCD}$

****

۶- اگر عدد ۱۰۰۱۱۰ یک عدد علامت‌دار باشد، معادل ده‌دهی آن کدام است؟

جواب :

$(100110)_B = (-3)_D$

****

۷- اعداد زیر را که در سیستم ده‌دهی هستند به سیستم‌های باینری، اُکتال و هگزادسی‌مال تبدیل کنید.

الف)۱۴۲ ب)۷۵۶ پ)۱۰۳۰ ت)۹۵۹

جواب :

$(142)_D = (10001110)_B = (216)_O = (8E)_H$

$(756)_D = (1011110100)_B = (1364)_O = (2F4)_H$

$(1030)_D = (10000000110)_B = (2006)_O = (406)_H$

$(959)_D = (1110111111)_B = (1677)_O = (3BF)_H$

****

۸- اعداد باینری زیر را با استفاده از روش‌هایی که فرا گرفتید به مبنای ۸ و ۱۶ تبدیل کنید و نتیجه‌ی استفاده از هر روش را با هم مقایسه کنید.

الف) ۲(۱۰۰۱۰۰۰۱۱) ب) ۲(۱۰۱۰۱۰۱) ج) ۲(۰۱۱۱۱۰۰۰۱۱)

جواب :

$(100100011)_B = (443)_O = (123)_H$

$(1010101)_B = (125)_O = (55)_H$

$(0111100011)_B = (743)_O = (1E3)_H$

****

۹- مکمّل‌های ۱ و۲ اعداد زیر را به دست آورید.

الف) ۲(۱۰۰۱۱۰۱) ب) ۲(۱۰۰۱۰۰۱) ج) ۲(۱۰۱۰۱۰۱۰۰۱۰)

جواب :

1001101 → مکمل 1 → 0110010

1001101 → مکمل 2 → 0110011

1001001 → مکمل 1 → 0110110

1001001 → مکمل 2 → 0110111

10101010010 → مکمل 1 → 01010101101

10101010010 → مکمل 2 → 01010101110

****

۱۰- جمع، تفریق و ضرب باینری اعداد زیر را به دست آورید.

الف	a) ۱۰۰۰۱۰۱	b) ۱۰۰۱۱۱
ب	a) ۱۰۰۱۱۱	b) ۱۰۰۰۱۱۱
ج	a) ۱۱۰۰۰۰	b) ۱۱۰۱۱

جواب :

الف	$a + b = 1101100$	$a - b = 11110$	$a \times b = 101010000011$
ب	$a + b = 1101110$	$a - b = -100000$	$a \times b = 101011010001$
ج	$a + b = 1001011$	$a - b = 10101$	$a \times b = 10100010000$

****

۱۱- اعداد زیر را به صورت نمایش کد BCD بنویسید.

الف) ۷۵۱ ب) ۴۲۰ ج) ۹۸۳ ت) ۶۱۲

جواب :

$751 \rightarrow$ کد BCD $\rightarrow 111\ 0101\ 0001$ $420 \rightarrow$ کد BCD $\rightarrow 0100\ 0010\ 0000$

$983 \rightarrow$ کد BCD $\rightarrow 1001\ 1000\ 0011$ $612 \rightarrow$ کد BCD $\rightarrow 0110\ 0001\ 0010$

****

۱۲- کد کردن اطلاعات یعنی چه؟

جواب :

معنای واقعی"کد کردن" همان "رمز کردن" یا به صورت رمز در آوردن اطلاعات است.

****

۱۳- پنج نوع کدینگ مختلف را نام برده و به اختصار توضیح دهید.

جواب :

۱) کد BCD : در کد BCD هر رقم ده‌دهی را با چهار بیت باینری معادل آن نشان می‌دهند.

۲) کدگری (انعکاسی) : مزیت کد گری نسبت به اعداد دودویی محض این است که وقتی از یک عدد به عدد بعدی می رویم فقط یک بیت تغییر می کند.

۳) کدهای اسکی (ASCII) : کد دودویی استاندارد برای کاراکترهای الفبا ، عددی ASCII ست. این کد از هفت بیت برای کد نمودن ۱۲۸ کاراکتر استفاده می‌کند.

۴) کد همینگ : جهت تشخیص و تصحیح خطا به کار می رود.

۵) کد دو- پنجی : این کد در هر حالت تنها دو تا یک دارد ، و برای خطا سنجی مناسب است.

۶) کد افزونی ۳ : در این کد هر رقم با رقم قبلی‌اش تنها یک تفاوت دارد.

****

۱۴-عدد 673.124 در مبنای 8 معادل چه اعدادی در مبناهای دودویی و دهدهی است.

جواب :

$$(673.124)_O = (443.164)_D = (110\ 111\ 011\ .\ 001\ 010\ 100)_B$$

****

۱۵- در هر یک از موارد زیر چند سلول (بیت) وجود دارد؟

الف) یک CD با ظرفیت700 MEGA BYTE

ب) یک DVD با ظرفیت4.7 GIGA BYTE

ج) یک FLASH MEMORY با ظرفیت8 GIGA BYTE

د) یک HARD با ظرفیت2 TERA BYTE

جواب :

$$700\ MEGA\ BYTE = 700 \times 2^{10} KBYTE = 700 \times 2^{20} BYTE = 700 \times 2^{23} BIT$$

$$4.7\ GIGA\ BYTE = 4.7 \times 2^{30}\ BYTE = 4.7 \times 2.33\ BIT$$

$$8\ GIGA\ BYTE = 2^3 \times 2^{30}\ BYTE = 2^3 \times 2^{33}\ BIT = 2^{36}\ BIT$$

$$2\ TERA\ BYTE = 2 \times 2^{40}\ BYTE = 2 \times 2^{43}\ BIT = 2^{44}\ BIT$$

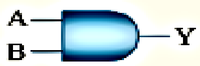
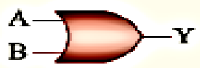
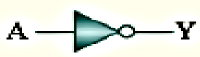
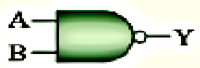
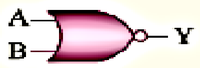
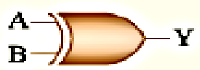
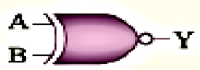
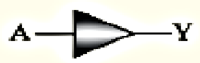
****

فصل دوم : گیت‌های منطقی پایه و ترکیبی

گیت‌ها یا دروازه‌های منطقی پایه (Basic Logic Gates) :

دروازه‌های منطقی، اساس کار ماشین‌های حساب، کامپیوترها، مدارهای کنترل و.... است. به عبارت دیگر، یک کامپیوتر یا ماشین حساب و ..... از تعدادی دروازه‌های منطقی تشکیل شده است. به طور خلاصه یک دروازه‌ی منطقی، یک مدار الکتریکی یا مدار الکترونیکی یا .... است که با توجه به حالت‌هایی که به ورودی آن داده می شود (صفر یا یک منطقی) ، خروجی آن نیز در وضعیت صفر یا یک منطقی قرار می‌گیرد. بدین ترتیب، انواع دروازه‌های منطقی وجود دارد که به شرح آن‌ها می‌پردازیم.

گیت‌های پایه و ترکیبی همراه با نماد، تابع منطقی و جدول درستی

نام دروازه	نماد (سمبل گرافیکی)	تابع منطقی	جدول درستی
AND		$Y=AB$	A B Y 0 0 0 0 1 0 1 0 0 1 1 1
OR		$Y=A+B$	A B Y 0 0 0 0 1 1 1 0 1 1 1 1
NOT		$Y=\overline{A}$	A Y 0 1 1 0
NAND		$Y=\overline{AB}=(AB)'$	A B Y 0 0 1 0 1 1 1 0 1 1 1 0
NOR		$Y=\overline{A+B}=(A+B)'$	A B Y 0 0 1 0 1 0 1 0 0 1 1 0
XOR		$Y=A\oplus B=\overline{A}B+A\overline{B}$	A B Y 0 0 0 0 1 1 1 0 1 1 1 0
XNOR		$Y=\overline{A\oplus B}=\overline{A}\,\overline{B}+AB$	A B Y 0 0 1 0 1 0 1 0 0 1 1 1
Buffer		$Y=A$	A Y 0 0 1 1

ترازهای ولتاژ (Voltage Levels) :

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

الف) ولتاژ بالا (High)

ب) ولتاژ پایین (Low)

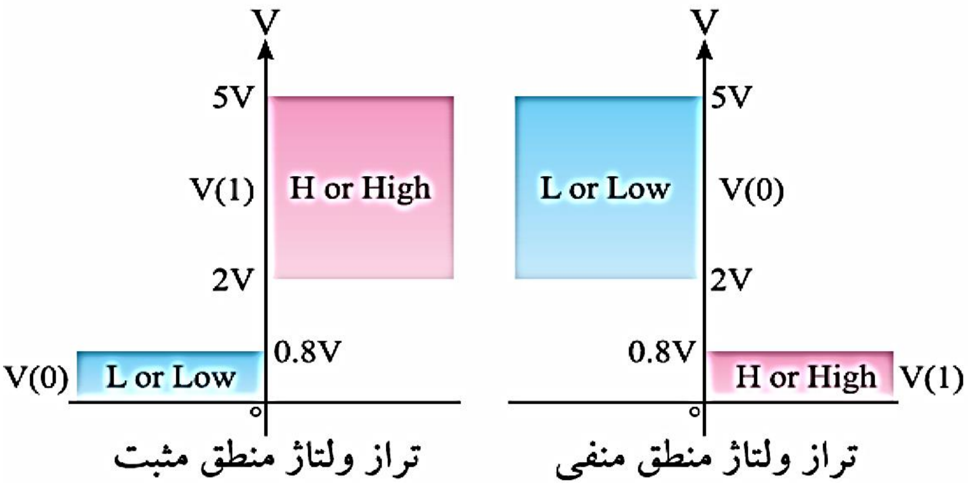
معمولاً به جای استفاده از حروف H و L ، اکثراً از نمادهای ۱ و۰ برای توصیف حالت ورودی‌ها و خروجی‌های مدارهای منطقی استفاده می‌کنند. ورودی‌ها و خروجی‌ها با دو حالت زیر تعریف می‌شوند.

الف)منطق مثبت (Positive logic)

ب)منطق منفی (Negative logic)

درمنطق مثبت عدد "۱" نشان دهنده ولتاژ بالا (H) و صفر نشان دهنده ولتاژ پایین (L) است.

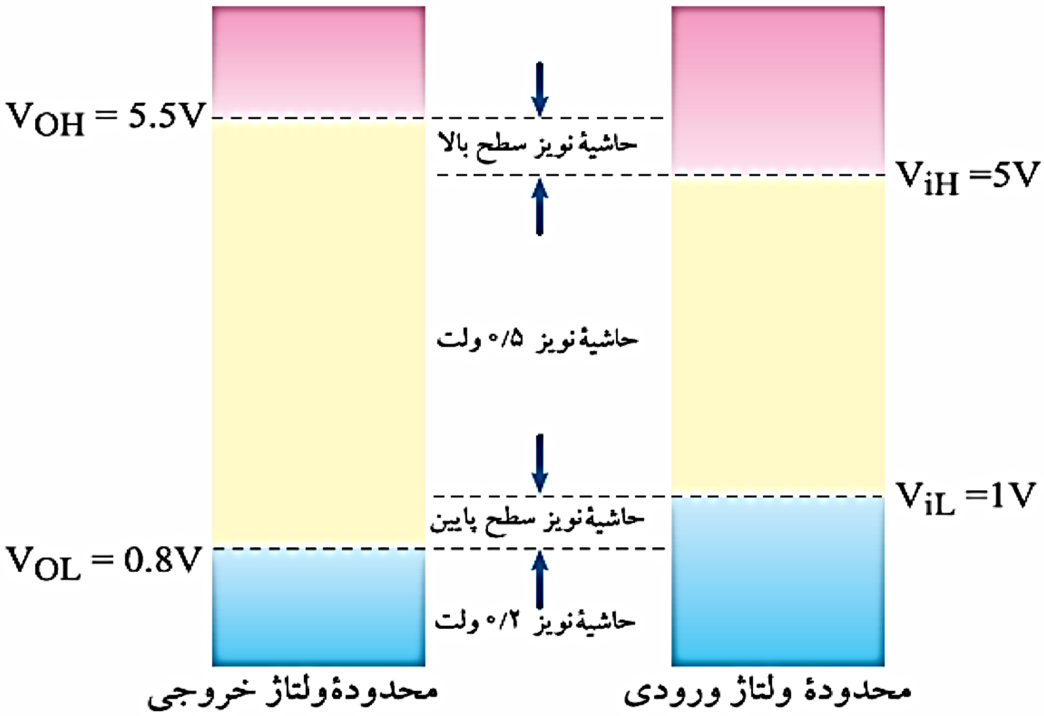
در منطق منفی عدد "۱" نشان دهنده ولتاژ پایین (L) و صفر نشان دهنده ولتاژ بالا (H) است.



Fan-in : حداکثر تعداد ورودی که یک گیت منطقی می تواند قبول کند را fan-in آن گیت می‌گویند. مثلاً اگر یک گیت محدود به ۶ ورودی باشد،گوییم fan-in این گیت برابر۶ است.

Fan-out : حداکثر تعداد گیت‌هایی که می‌توانند از طریق خروجی یک گیت تغذیه شود را fan-out آن گیت گوییم. مثلاً اگر fan-out یک گیت برابر با ۵ باشد ، یعنی خروجی این گیت را می‌توان حداکثر به ۵ گیت دیگر متصل نمود .

حاشیه نویز (Marginal noise) : حاشیه نویز در یک گیت منطقی ، تأثیر دامنه نویز در ورودی مدار منطقی است.

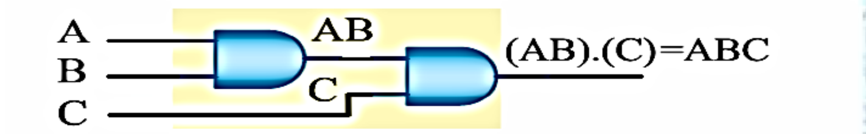


تأخیر در انتشار (Propagation delay) : تأخیر درانتشار عبارت است از زمانی که خروجی یک دروازه منطقی لازم دارد تا تغییر ورودی از یک حالت به حالت دیگر ظاهر نماید ، به عبارت دیگر هرچه تأخیر کمتر باشد سرعت انتقال اطلاعات بیشتر می شود.

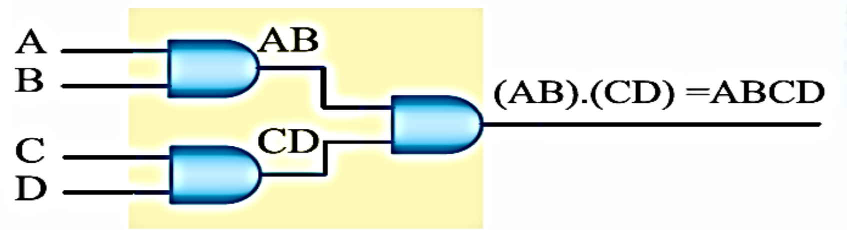
توان تلف شده (Power dissipation) : مقدار توانی که در هر گیت به صورت حرارت تلف می شود را توان تلف شده آن گیت می‌گویند ومقدار توان تلف شده بر حسب میلی وات اندازه‌گیری می‌شود.

افزایش ظرفیت ورودی های دروازه های منطقی:

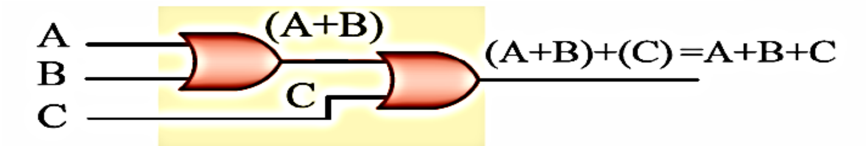
اغلب گیت‌های منطقی ساخته شده و موجود در بازار دو ورودی دارند . اما می‌توان با استفاده از دروازه‌های منطقی موجود یک دروازه منطقی با تعداد ورودی‌های دلخواه ساخت. در این قسمت به شرح روش افزایش تعداد ورودی‌ها در بعضی گیت‌های منطقی می پردازیم.



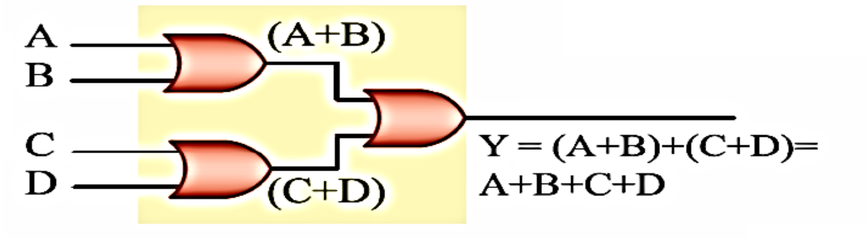
ساخت دروازه AND با سه ورودی با استفاده از AND با دو ورودی



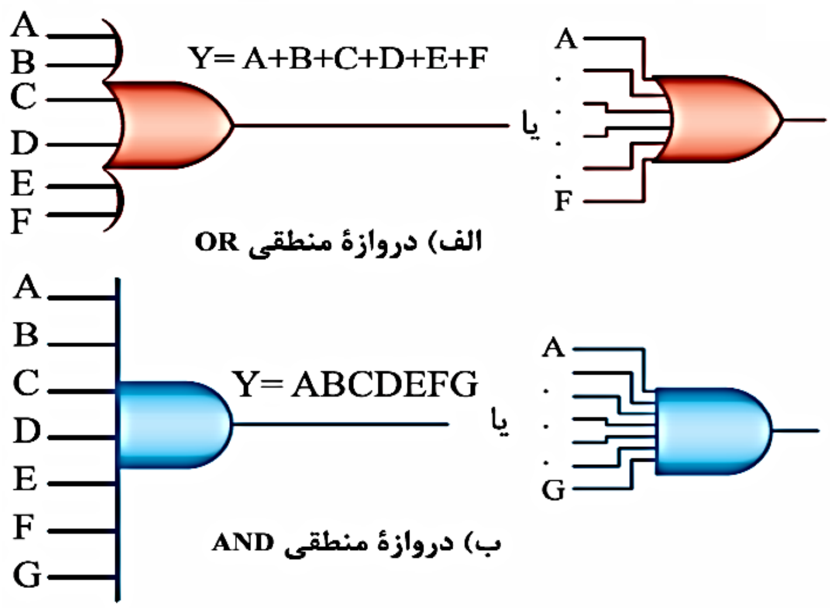
دروازهٔ منطقی AND با چهار ورودی با استفاده از AND با دو ورودی



ساخت دروازهٔ منطقی OR با سه ورودی با استفاده از OR با دو ورودی

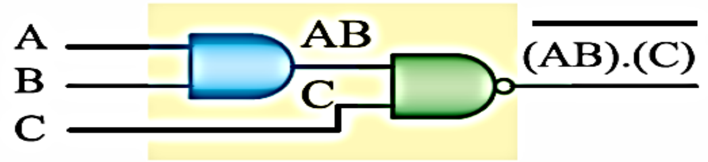


دروازهٔ منطقی OR با چهار ورودی با استفاده از OR با دو ورودی

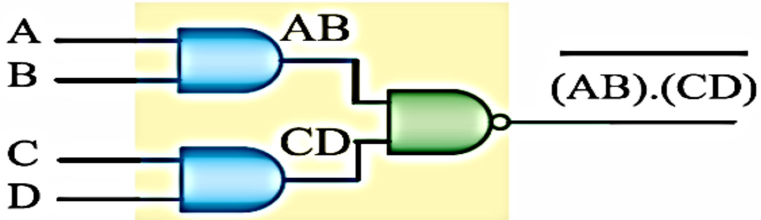


نمادهای دروازه‌های منطقی AND و OR با ورودی‌های زیاد

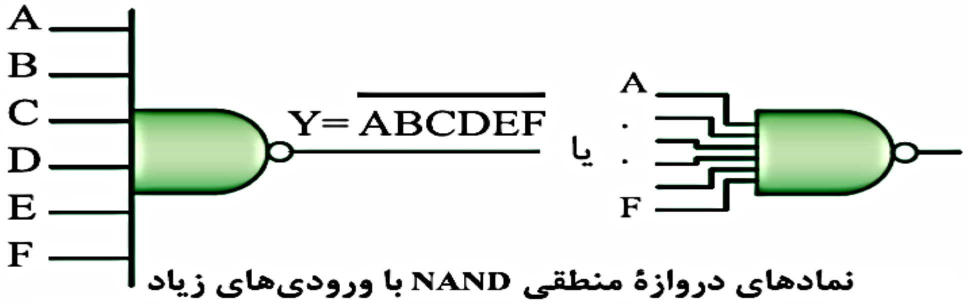
نکته : برای ساخت گیت‌های NAND , NOR فقط گیت آخری را NOT می‌کنیم .



دروازهٔ منطقی NAND با سه ورودی با منطقی AND و NAND

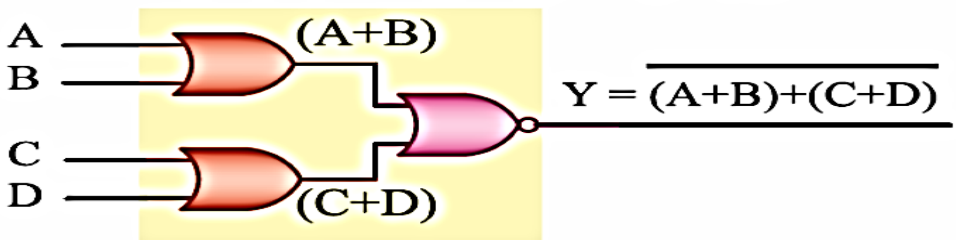


دروازهٔ منطقی NAND با چهار ورودی با استفاده از دروازه‌های منطقی AND و NAND با دو ورودی

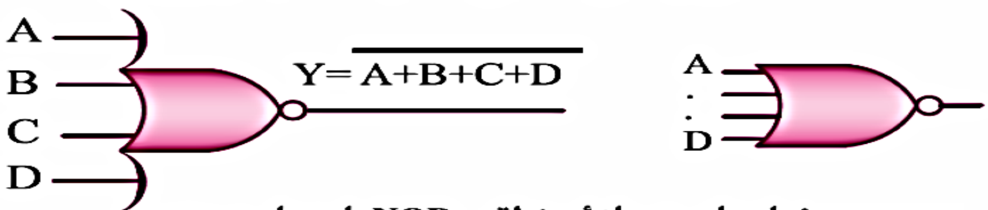


نمادهای دروازهٔ منطقی NAND با ورودی‌های زیاد

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

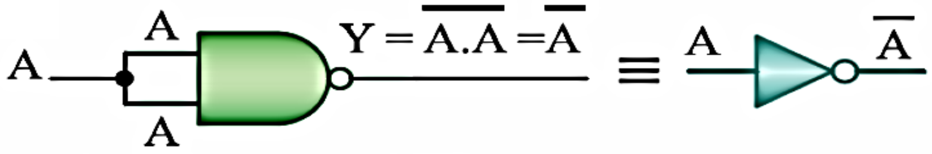


ایجاد دروازهٔ منطقی NOR با چهار ورودی با استفاده از دروازه‌های منطقی OR و NOR با دو ورودی

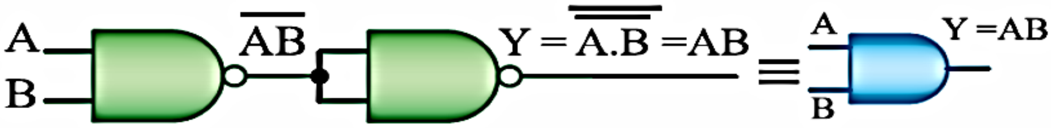


نمادهای دروازهٔ منطقی NOR با چهار ورودی

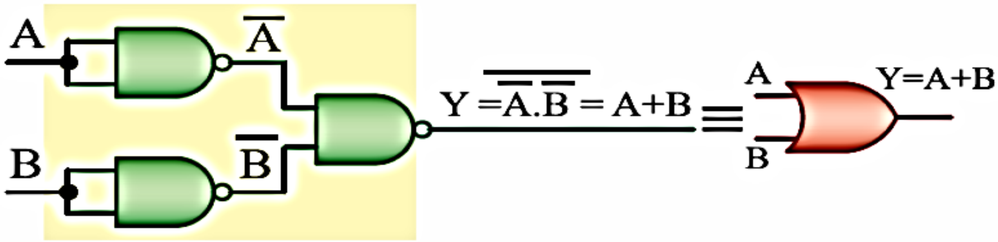
نمادهای دروازهٔ منطقی NOR با چهار ورودی



ساخت دروازه منطقی NOT با استفاده از NAND



ساخت دروازهٔ منطقی AND با استفاده از NAND



ساخت دروازهٔ منطقی OR با استفاده ازNAND

تمرین :

۱- مفاهیم زیر را تعریف کنید.

الف) منطق مثبت ب) منطق منفی

جواب :

الف) منطق مثبت (Positive logic) : درمنطق مثبت عدد "۱" نشان دهنده ولتاژ بالا (H) و صفر نشان دهنده ولتاژ پایین (L) است.

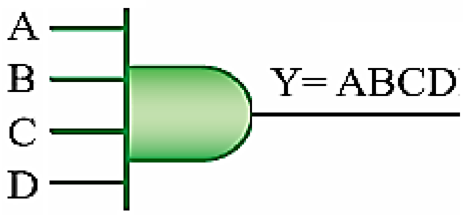
ب) منطق منفی (Negative logic) : در منطق منفی عدد "۱" نشان دهنده ولتاژ پایین (L) و صفر نشان دهنده ولتاژ بالا (H) است.

****

۲- یک دروازه AND با چهار ورودی رارسم کنید و جدول صحت آن را بنویسید.

جواب :

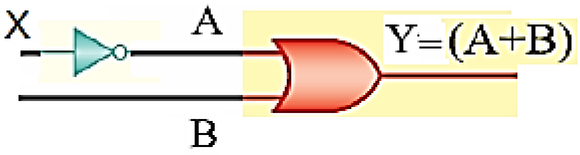
A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1



****

۳- یک گیت NOT را در مسیر یکی از ورودی‌های گیت OR قرار می دهیم . مدار آن را رسم کرده و جدول صحت آن‌را بیابید .

جواب :



X	A=X’	B	Y=A+B
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1

۴- جدول صحت دروازه XOR را با سه ورودی بنویسید.

جواب : میدانیم که $Y = A \oplus B \oplus C$ و اگر $X = A \oplus B$ باشد ، میتوان گفت : $Y = X \oplus C$

و می‌دانیم اگر ورودی‌های یک گیت XOR همانند باشند خروجی آن صفر است . پس داریم :

A	B	$X = A \oplus B$	C	$Y = X \oplus C$
0	0	0	0	0
0	0	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1

→

A	B	C	$Y = A \oplus B \oplus C$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

۵- مشخصات ویژه دروازه‌های منطقی را به طورخلاصه توضیح دهید.

جواب :

Fan-in : حداکثر تعداد ورودی که یک گیت منطقی می تواند قبول کند را fan-in آن گیت می‌گویند. مثلاً اگر یک گیت محدود به ۶ ورودی باشد،گوییم fan-in این گیت برابر۶ است.

Fan-out : حداکثر تعداد گیت‌هایی که می‌توانند از طریق خروجی یک گیت تغذیه شود را fan-out آن گیت گوییم. مثلاً اگر fan-out یک گیت برابر با ۵ باشد ، یعنی خروجی این گیت را می‌توان حداکثر به ۵ گیت دیگر متصل نمود .

حاشیه نویز (Marginal noise) : حاشیه نویز در یک گیت منطقی ، تأثیر دامنه نویز در ورودی مدار منطقی است.

تأخیر در انتشار (Propagation delay) : تأخیر درانتشار عبارت است از زمانی که خروجی یک دروازه منطقی لازم دارد تا تغییر ورودی از یک حالت به حالت دیگر ظاهر نماید ، به عبارت دیگر هرچه تأخیر کمتر باشد سرعت انتقال اطلاعات بیشتر می شود.

توان تلف شده (Power dissipation) : مقدار توانی که در هر گیت به صورت حرارت تلف می شود را توان تلف شده آن گیت می‌گویند ومقدار توان تلف شده بر حسب میلی وات اندازه‌گیری می‌شود.

****

۶- fan-in یک دروازه منطقی برابر با چهار است این دروازه منطقی چه ویژگی خاصی دارد؟

جواب :

یعنی این گیت چهار پایه ورودی دارد .

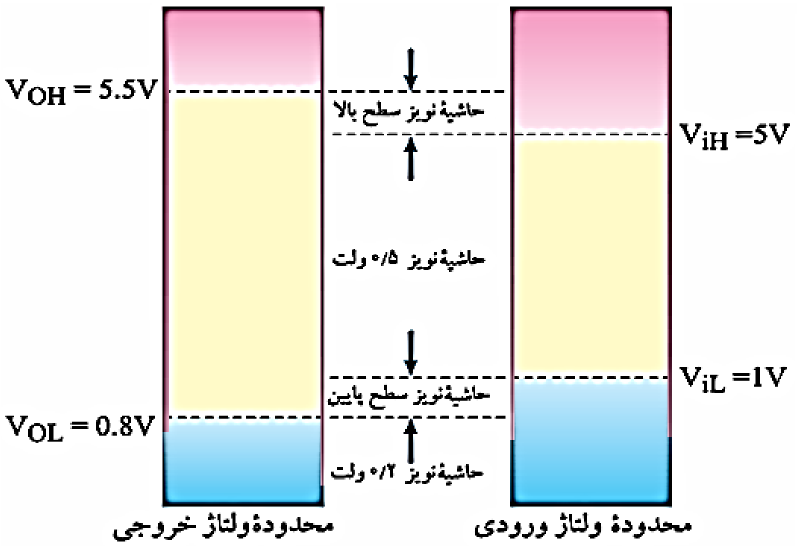
****

۷- اگر مشخصات یک گیت به شکل زیر باشد ، حاشیه نویز سطح پایین و بالا را بیابید .

$V_{OH} = 5.5V \quad V_{iH} = 5V$

$V_{OL} = 0.8V \quad V_{iL} = 1V$

جواب : با توجه به مقادیر داده شده و شکل زیر داریم :



حاشیه نویز سطح بالا = VOH - ViH = 0.5 V

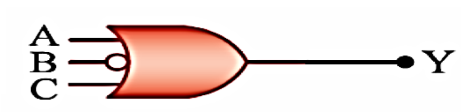
حاشیه نویز سطح پایین = ViL - VOL = 0.2 V

****

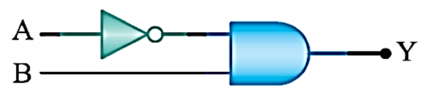
۸- جدول صحت وخروجی تابع هریک از شکل های زیر رابنویسید.

الف)

ب)



ج)



جواب :

الف : $Y = ((A.B)')' = A.B$

A	B	F=AB
0	0	0
0	1	0
1	0	0
1	1	1

ب : $Y = AB'C$

A	B	C	B’	Y=AB’C
0	0	0	1	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	0
1	0	0	1	0
1	0	1	1	1
1	1	0	0	0
1	1	1	0	0

ج : $Y = A'B$

A	B	A’	Y=A’B
0	0	1	0
0	1	1	1
1	0	0	0
1	1	0	0

۹- جدول صحت را برای گیت‌های زیر که هر یک دارای سه ورودی هستند بنویسید.

NAND – AND – OR- NOR

جواب :

NAND

A	B	C	Y=(ABC)’
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

AND

A	B	C	Y=ABC
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

OR

A	B	C	Y=A+B+C
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

NOR

A	B	C	Y=(A+B+C)’
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

****

۱۰- هر یک از جملات زیر به چه گیتی اشاره دارد ؟

الف) خروجی آن همیشه صفر است مگر آنکه تمام ورودی‌ها یک باشند .

ب) برای صفر شدن خروجی آن کافی است حداقل یک ورودی صفر داشته باشد .

ج) خروجی آن همیشه یک است مگر آنکه تمام ورودی‌ها صفر باشند .

د) برای یک شدن خروجی آن کافی است حداقل یک ورودی یک داشته باشد .

ه) خروجی آن همواره مخالف ورودی است .

و) خروجی آن همیشه یک است مگر آنکه تمام ورودی‌ها یک باشند .

ز) برای یک شدن خروجی آن کافی است حداقل یک ورودی صفر داشته باشد .

ح) خروجی آن همیشه صفر است مگر آنکه تمام ورودی‌ها صفر باشند .

ط) برای صفر شدن خروجی آن کافی است حداقل یک ورودی یک داشته باشد .

ی) اگر تمام ورودی‌ها یکسان باشند ، قطعاً خروجی آن صفر است .

ک) اگر تمام ورودی‌ها یکسان باشند ، قطعاً خروجی آن یک است .

ل) خروجی آن همواره برابر با ورودی است .

جواب :

الف) AND　　ب) AND　　ج) OR　　د) OR　　ه) NOT　　و) NAND

ز) NAND　　ح) NOR　　ط) NOR　　ی) XOR　　ک) XNOR　　ل) BUFFER

فصل سوم : جبر بول ، قوانین دمُرگان و طراحی مدارهای ساده منطقی

جبر بول(Boolean algebra) :

این جبر روش‌های مفید و ساده‌ای را برای تجزیه و تحلیل و ترکیب مدارهای دیجیتالی، از جمله مدارهای ترکیبی (Combinational) و مدارهای ترتیبی (Sequential) ارائه می دهد.

برای درک بهتر جبر بول و استفاده مؤثر از آن، ابتدا روش‌های کلی مربوط به این جبر را بیان می کنیم.

جدول قوانین جبر بول

$A+0=A$	تأثیر عضو خنثی «صفر» در جمع منطقی
$A+1=1$	تأثیر «یک» منطقی در جمع منطقی
$A+A=A$	جمع منطقی یک تابع با خودش
$A+\overline{A}=1$	جمع منطقی یک تابع با معکوس خودش
$A\cdot 1=A$	تأثیر عضو خنثی «یک» در ضرب منطقی
$A\cdot 0=0$	تأثیر «صفر» منطقی در عمل ضرب منطقی
$A\cdot A=A$	ضرب یک تابع در خودش
$A\cdot\overline{A}=0$	ضرب یک تابع در معکوس خودش
$A\cdot(B+C)=AB+AC$	ضرب یک تابع در پرانتز
$A+(BC)=(A+B)\cdot(A+C)$	جمع یک تابع با پرانتز

$0+0=0$	$0*0=0$
$0+1=1$	$0*1=0$
$1+0=1$	$1*0=0$
$1+1=1$	$1*1=1$

قوانین دِمُرگان (Domorgan) :

قانون اول:

$$\boxed{\overline{(A+B)} = \overline{A}.\overline{B}}$$

قانون دوم:

$$\boxed{\overline{A.B} = \overline{A} + \overline{B}}$$

مثال: عبارت زیر را با استفاده از قوانین دِمُرگان و جبر بول به ساده‌ترین شکل ممکن بنویسید.

$$Y = \overline{ABC} + \overline{A}BC + A\overline{BC} + A\overline{B}C + ABC$$

$$Y = \underbrace{\overline{A}\,\overline{B}\,\overline{C}}_{\text{۱}} + \underbrace{\overline{A}BC}_{\text{۲}} + \underbrace{A\overline{B}\,\overline{C}}_{\text{۳}} + \underbrace{A\overline{B}C}_{\text{۴}} + \underbrace{ABC}_{\text{۵}}$$

$$Y = \underbrace{\overline{B}\,\overline{C}(\overline{A}+A)}_{\text{۱و۳}} + \underbrace{AC(\overline{B}+B)}_{\text{۴و۵}} + \underbrace{BC(\overline{A}+A)}_{\text{۲و۵}}$$

$$Y = \overline{B}\,\overline{C} + AC + BC = \underset{\text{۱}}{\overline{B}\,\overline{C}} + \underset{\text{۲}}{AC} + \underset{\text{۳}}{BC} = AC + (\overline{B \oplus C})$$

روش طراحی مدارهای کلیدی (سوئیچی) :

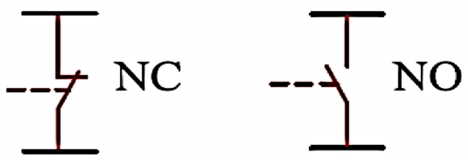
۱-بجای ضرب (AND) میتوان از کلیدهای سری استفاده کرد.

۲-بجای جمع (OR) میتوان از کلیدهای موازی استفاده کرد.

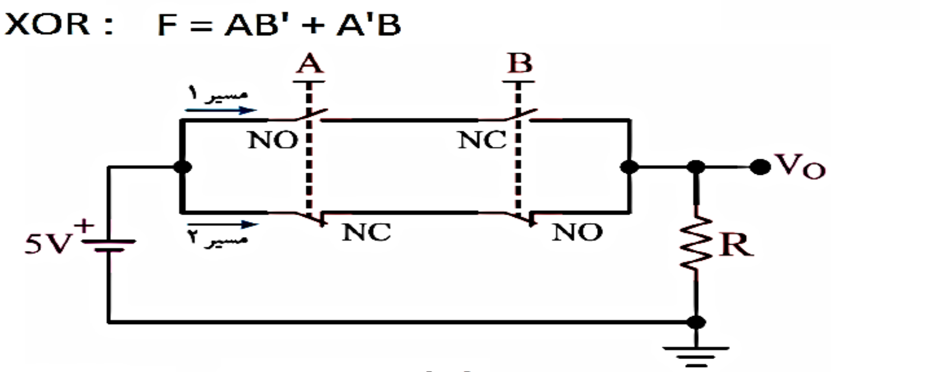
۳-بجای مکمل (NOT) میتوان از کلیدهای موازی یا NC استفاده کرد.

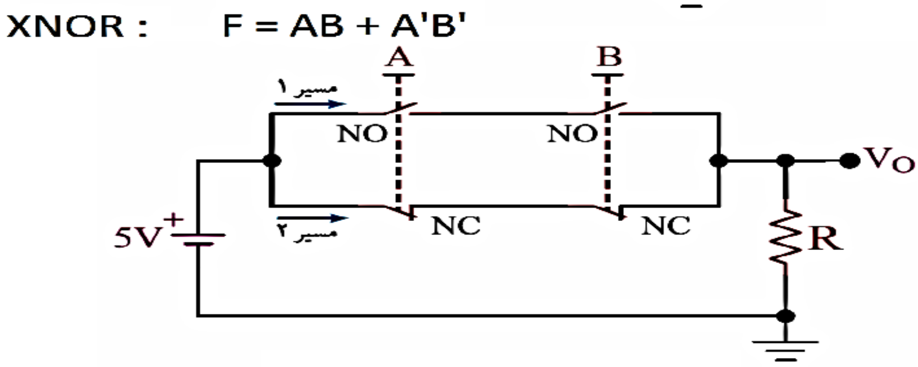
کلیدهای NO و NC :

کلیدهای فشاری nc (normally closed) در حالت عادی بسته و no (normally open) در حالت عادی باز، از جمله کلیدهایی هستند که در مدار معادل کلیدی گیت های منطقی استفاده دارند.



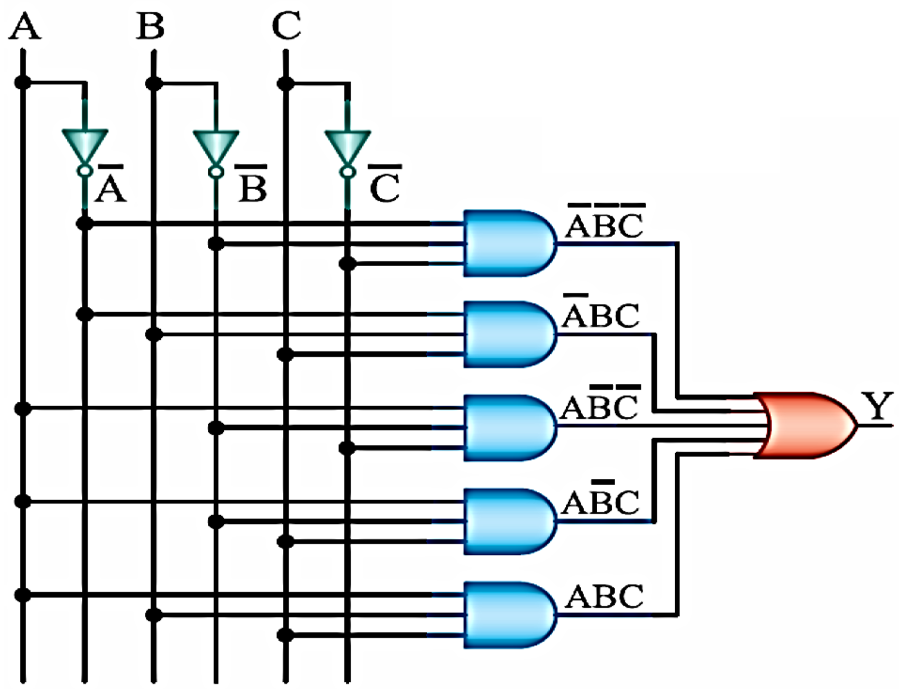
مثال : نمودار کلیدی گیت‌های XOR و XNOR را رسم کنید.

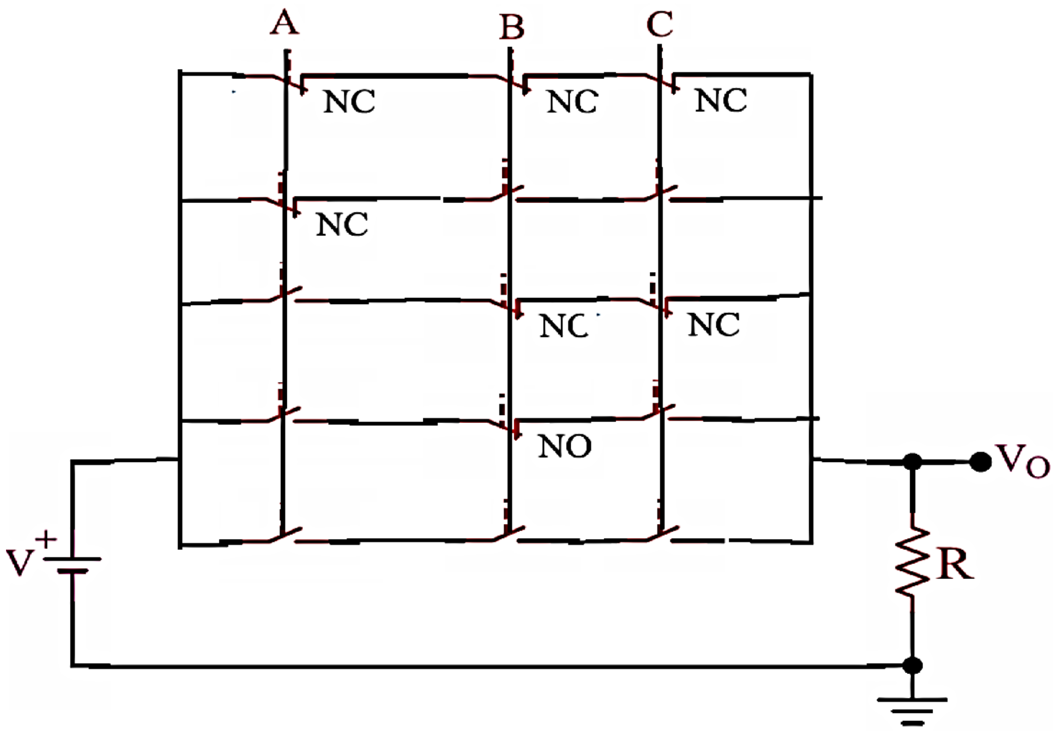




مثال : مدار منطقی و مدار کلیدی تابع زیر را رسم کنید.

$$Y = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}BC + A\overline{B}\,\overline{C} + A\overline{B}C + ABC$$





مثال: تابع سؤال قبل را با استفاده از قوانین دِمُرگان و جبر بول به ساده‌ترین شکل ممکن بنویسید و مدار آن‌را بازطراحی کنید.

همان‌طور که نشان دادیم :

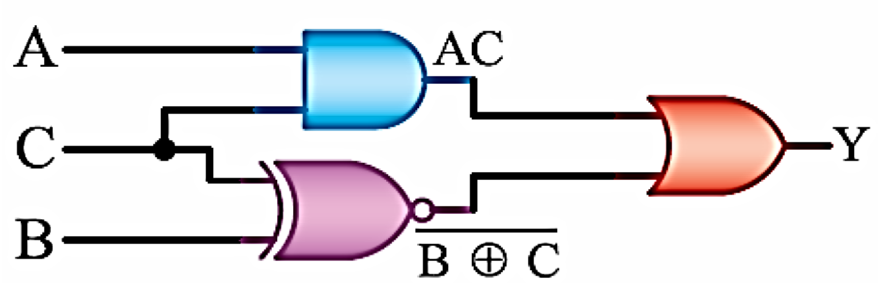
$$Y=\bar{A}\bar{B}\bar{C}+\bar{A}BC+A\bar{B}\bar{C}+A\bar{B}C+ABC$$

$$Y=\underbrace{\bar{A}\bar{B}\bar{C}}_{۱}+\underbrace{\bar{A}BC}_{۲}+\underbrace{A\bar{B}\bar{C}}_{۳}+\underbrace{A\bar{B}C}_{۴}+\underbrace{ABC}_{۵}$$

$$Y=\underbrace{\bar{B}\bar{C}(\bar{A}+A)}_{۱و۳}+\underbrace{AC(\bar{B}+B)}_{۴و۵}+\underbrace{BC(\bar{A}+A)}_{۲و۵}$$

$$Y=\bar{B}\bar{C}+AC+BC=\underset{۱}{\bar{B}\bar{C}}+\underset{۲}{AC}+\underset{۳}{BC}=AC+(\overline{B\oplus C})$$

نکته : همان‌طور که نشان دادیم به جای مدار مثال قبل، میتوان از مدار زیر نیز استفاده نمود.



مراحل طراحی یک مدار :

مرحله ی (۱) : تنظیم جدول صحّت

مرحله ی (۲) : تکمیل جدول صحّت

مرحله ی (۳) : به دست آوردن رابطه‌ی خروجی با جمع عبارت‌های یک

مرحله ی (۴) : ساده‌سازی تابع خروجی

مرحله ی (۵) : طراحی مدار با استفاده از گیت‌های منطقی یا مدار سوئیچی

مثال : مداری طراحی کنید که جدول صحّت زیر را ایجاد کند.

A	B	Y
۰	۰	۱
۰	۱	۱
۱	۰	۰
۱	۱	۰

مرحلهٔ (۱) نداریم

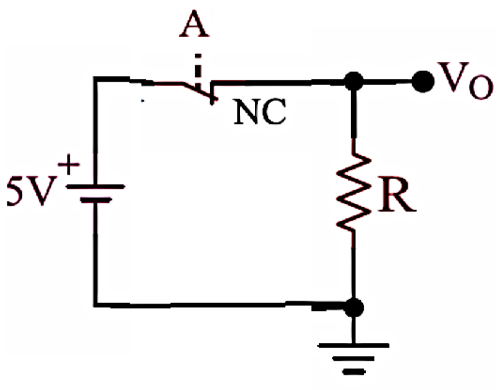
مرحلهٔ (۲)

A	B	Y	
0	0	1	$\bar{A}\cdot\bar{B}=1$
0	1	1	$\bar{A}\cdot B=1$
1	0	0	
1	1	0	

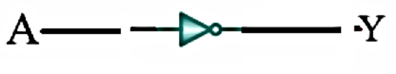
مرحلهٔ (۳) $Y = A'B' + A'B$

مرحلهٔ (۴) $Y = A'B' + A'B = A'(B' + B) = A' \cdot 1 = A'$

مرحلهٔ (۵)



مدار سوئیچی



مدار منطقی

مثال : سیستم دزدگیری با دو سنسور طوری طراحی کنید که اگر حداقل یک سنسور فعال شد، آژیر فعال شود.

رحلهٔ (۱)

A	B	Y
۰	۰	۰
۰	۱	۱
۱	۰	۱
۱	۱	۱

مرحلهٔ (۲)

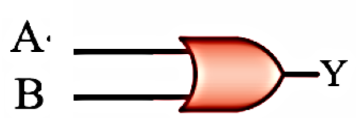
A	B	Y	
۰	۰	۰	
۰	۱	۱	→ A'B
۱	۰	۱	→ AB'
۱	۱	۱	→ AB

مرحلهٔ (۳) $Y = \overline{A}B + \overline{A}B + AB$

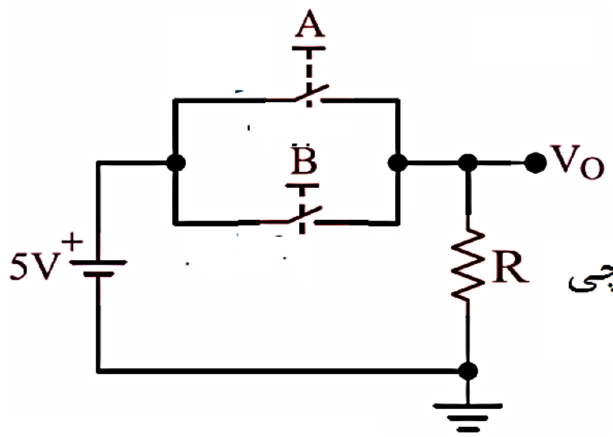
مرحلهٔ (۴)

$$Y = A\overline{B} + \overline{A}B + AB = A\overline{B} + B(\overline{A} + A) = B + A\overline{B}$$
$$= (B + A)(B + \overline{B}) = (B + A)(1) = A + B$$

مرحلهٔ (۵)



مدار منطقی



مدار سوئیچی

مثال : مداری طراحی کنید که جدول صحت زیر را ایجاد کند.

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

A	B	C	Y	
0	0	0	1	$\rightarrow \overline{A}\cdot\overline{B}\cdot\overline{C}$
0	0	1	0	
0	1	0	0	
0	1	1	1	$\rightarrow \overline{A}\cdot B\cdot C$
1	0	0	1	$\rightarrow A\cdot\overline{B}\cdot\overline{C}$
1	0	1	1	$\rightarrow A\cdot\overline{B}\cdot C$
1	1	0	0	
1	1	1	1	$\rightarrow A\cdot B\cdot C$

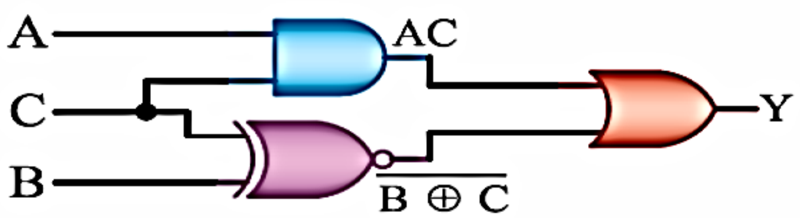
$$Y=\overline{A}\,\overline{B}\,\overline{C}+\overline{A}BC+A\overline{B}\,\overline{C}+A\overline{B}C+ABC$$

$$Y=\underbrace{\overline{A}\,\overline{B}\,\overline{C}}_{۱}+\underbrace{\overline{A}BC}_{۲}+\underbrace{A\overline{B}\,\overline{C}}_{۳}+\underbrace{A\overline{B}C}_{۴}+\underbrace{ABC}_{۵}$$

$$Y=\underbrace{\overline{B}\,\overline{C}(\overline{A}+A)}_{۱\text{و}۳}+\underbrace{AC(\overline{B}+B)}_{۴\text{و}۵}+\underbrace{BC(\overline{A}+A)}_{۲\text{و}۵}$$

$$Y=\overline{B}\,\overline{C}+AC+BC=\underbrace{\overline{B}\,\overline{C}}_{۱}+\underbrace{AC}_{۲}+\underbrace{BC}_{۳}$$

$$Y=AC+(B\odot C)$$



مثال : تابع زیر را ساده کنید.

$$Y=\bar{A}B\bar{C}+AB\bar{C}+\bar{A}\bar{B}\bar{C}+A\bar{B}C+\bar{A}\bar{B}C+A\bar{B}\bar{C}$$

$$Y=\underbrace{\bar{A}B\bar{C}}_{\text{۱}}+\underbrace{AB\bar{C}}_{\text{۲}}+\underbrace{\bar{A}\bar{B}\bar{C}}_{\text{۳}}+\underbrace{A\bar{B}C}_{\text{۴}}+\underbrace{\bar{A}\bar{B}C}_{\text{۵}}+\underbrace{A\bar{B}\bar{C}}_{\text{۶}}$$

$$Y=\underbrace{B\bar{C}(\bar{A}+A)}_{\text{از جمله‌های ۱و۲}}+\underbrace{\bar{A}\bar{B}(\bar{C}+C)}_{\text{از جمله‌های ۳ و ۵}}+\underbrace{A\bar{B}(C+\bar{C})}_{\text{از جمله‌های ۴ و ۶}}$$

$$Y=B\bar{C}+\bar{A}\bar{B}+A\bar{B}=\underbrace{B\bar{C}}_{\text{۱}}+\underbrace{\bar{A}\bar{B}}_{\text{۲}}+\underbrace{A\bar{B}}_{\text{۳}}$$

$$Y=B\bar{C}+\underbrace{\bar{B}(\bar{A}+A)}_{\text{از جمله ۲ و ۳}}=B\bar{C}+\bar{B}$$

$$Y=(B+\bar{B})(\bar{B}+\bar{C})=\bar{B}+\bar{C}$$

فرم استاندارد توابع بول : یک تابع بول را در صورتی فرم استاندارد بول می‌گویند که در هر جمله‌ی آن همه‌ی متغیرها از جمله خود متغیر یا NOT آن ظاهر شده باشد.

برای مثال، تابع $Y=\bar{A}BC+AB\bar{C}+\overline{ABC}+ABC+\bar{A}B\bar{C}$ یک تابع استاندارد بول است؛ امّا تابع $Y=AB\bar{C}+\overline{ABC}+ABC+\bar{B}C$ یک تابع استاندارد بول نیست.

نکته : از آنجاییکه (A+A’) = (B+B’) = (C+C’) = 1 و میدانیم ضرب هر عبارت در عدد ۱ اشکالی ندارد ، برای استاندارد کردن یک تابع غیر استاندارد هر جمله ناقص آن را در (A+A’) یا (B+B’) یا (C+C’) ضرب می‌کنیم.

مثال: تابع F = A’B’ + A’C + BC را به فرم استاندارد در آوردید.

$$Y=\bar{A}\bar{B}(C+\bar{C})+\bar{A}C(B+\bar{B})+BC(A+\bar{A})$$

$$Y=\underbrace{\bar{A}\bar{B}C}_{\text{۱}}+\underbrace{\bar{A}\bar{B}\bar{C}}_{\text{۲}}+\underbrace{\bar{A}BC}_{\text{۳}}+\underbrace{\bar{A}\bar{B}C}_{\text{۴}}+\underbrace{ABC}_{\text{۵}}+\underbrace{\bar{A}BC}_{\text{۶}}$$

$$Y=\bar{A}\bar{B}C+\bar{A}\bar{B}\bar{C}+\bar{A}BC+ABC$$

مثال : تابع زیر را به صورت استاندارد بول درآورید.

$$Y=A+BC$$

$$Y=A.1+BC.1=A(B+\overline{B})+BC(A+\overline{A})$$

$$Y=AB+A\overline{B}+ABC+\overline{A}BC$$

$$Y=AB.1+A\overline{B}.1+ABC+\overline{A}BC$$

$$Y=AB(C+\overline{C})+A\overline{B}(C+\overline{C})+ABC+\overline{A}BC$$

$$Y=\underset{۱}{ABC}+\underset{۲}{AB\overline{C}}+\underset{۳}{A\overline{B}C}+\underset{۴}{A\overline{B}\overline{C}}+\underset{۵}{ABC}+\underset{۶}{\overline{A}BC}$$

$$Y=ABC+AB\overline{C}+A\overline{B}C+A\overline{B}\overline{C}+\overline{A}BC$$

مجموع حاصل ضرب‌ها (Sums of Products) یا مین‌ترم‌ها (minterm) :

اگر در هر عبارت مجموع حاصل ضرب‌ها، به همان تعداد متغیری که در تابع وجود دارد، متغیرها یا مکمل‌های (NOT) آن‌ها وجود داشته باشد، آن عبارت را مین‌ترم می گویند.

مثال: اگر جدول صحت Y به شکل زیر باشد، آنرا به شکل مجموع حاصلضرب ها یا مین‌ترم بنویسید.

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

شمارهٔ سطر جدول	ورودی‌ها A	B	C	خروجی Y	
0	0	0	0	0	
1	0	0	1	1	$\overline{A}\cdot\overline{B}\cdot C$
2	0	1	0	1	$\overline{A}\cdot B\cdot\overline{C}$
3	0	1	1	0	
4	1	0	0	1	$A\cdot\overline{B}\cdot\overline{C}$
5	1	0	1	0	
6	1	1	0	0	
7	1	1	1	1	$A\cdot B\cdot C$

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

$$Y=\Sigma_m\,(m_{\text{۱}}\ ,m_{\text{۲}}\ ,m_{\text{۴}},m_{\text{۷}})$$

$$Y=\Sigma_m\,(\text{۱,۲,۴,۷})$$

مثال : تابع خروجی $Y=\Sigma_m$ (۶،۵،۳،۱) را بنویسید.

حل : تابع خروجی در سطرهای ۶،۵،۳،۱ یک است.

$$Y=\underbrace{\overset{0\,0\,1}{\overline{A}\,\overline{B}C}}_{\text{سطر ۱}}+\underbrace{\overset{0\,1\,1}{\overline{A}BC}}_{\text{سطر ۳}}+\underbrace{\overset{1\,0\,1}{A\overline{B}C}}_{\text{سطر ۵}}+\underbrace{\overset{1\,1\,0}{AB\overline{C}}}_{\text{سطر ۶}}$$

عبارت حاصل ضرب مجموع‌ها (Products of sums) یا ماکس‌ترم‌ها (Maxterm) :

اگر در هر عبارت حاصل ضرب مجموع‌ها به همان تعداد متغیری که در تابع وجود دارد، متغیرها یا مکمل‌های (NOT) آن‌ها وجود داشته باشد، آن‌ها را ماکس‌ترم می‌گوییم.

مثلاً عبارت $Y=(A'+B+C')(A+B'+C')(A+B'+C)$ یک عبارت ماکس‌ترم است .

معمولاً یک عبارت ماکس‌ترم را می‌توان به صورت زیر نشان داد.

$Y=\Pi_M\,(M_2,M_3,M_5)$ یا $Y=\Pi_M\,(2,3,5)$ می‌خوانیم پی اِم ....

به عنوان مثال جدول صحت مربوط به رابطه‌ی ماکس ترم Y را در جدول مشاهده می کنید. $Y=\Pi_M\,(2,3,5)$

شمارهٔ سطر جدول	ورودی‌ها A	B	C	خروجی Y	
0	0	0	0	1	
1	0	0	1	1	
2	0	1	0	0	$A+\overline{B}+C$
3	0	1	1	0	$A+\overline{B}+\overline{C}$
4	1	0	0	1	
5	1	0	1	0	$\overline{A}+B+\overline{C}$
6	1	1	0	1	
7	1	1	1	1	

$$Y=\underbrace{(A+\overline{B}+C)}_{\text{۲}}\underbrace{(A+\overline{B}+\overline{C})}_{\text{۳}}\underbrace{(\overline{A}+B+\overline{C})}_{\text{۵}}$$

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

مثال : رابطه‌ی خروجی $F = \Pi_M\,(M0, M1, M3)$ را بنویسید.

حل :

$$\begin{array}{ccc} \underline{\;\;0\;\;} & \underline{\;\;1\;\;} & \underline{\;\;3\;\;} \\ 0\;0\;0 & 0\;0\;1 & 0\;1\;1 \end{array}$$

$$F = (A+B+C)\,(A+B+C')\,(A+B'+C')$$

مجموعه‌ی حاصل ضرب‌ها و حاصل ضرب مجموع‌های هر سطر برای یک جدول با سه ورودی در جدول نشان داده شده است.

شمارهٔ سطر جدول	ورودی ها			عبارت حاصل‌ضرب	عبارت حاصل‌جمع
	A	B	C		
0	0	0	0	$\overline{A}\cdot\overline{B}\cdot\overline{C}$	$A+B+C$
1	0	0	1	$\overline{A}\cdot\overline{B}\cdot C$	$A+B+\overline{C}$
2	0	1	0	$\overline{A}\cdot B\cdot\overline{C}$	$A+\overline{B}+C$
3	0	1	1	$\overline{A}\cdot B\cdot C$	$A+\overline{B}+\overline{C}$
4	1	0	0	$A\cdot\overline{B}\cdot\overline{C}$	$\overline{A}+B+C$
5	1	0	1	$A\cdot\overline{B}\cdot C$	$\overline{A}+B+\overline{C}$
6	1	1	0	$A\cdot B\cdot\overline{C}$	$\overline{A}+\overline{B}+C$
7	1	1	1	$A\cdot B\cdot C$	$\overline{A}+\overline{B}+\overline{C}$

مثال : برای تابعی با جدول صحت زیر، عبارت مین‌ترم‌ها و ماکسترم‌ها را بنویسید.

	ورودی ها			خروجی
	A	B	C	Y
۰	۰	۰	۰	۰
۱	۰	۰	۱	۱
۲	۰	۱	۰	۰
۳	۰	۱	۱	۰
۴	۱	۰	۰	۱
۵	۱	۰	۱	۱
۶	۱	۱	۰	۰
۷	۱	۱	۱	۱

$$Y = \Sigma_m\,(۱,۴,۵,۷) = \Pi_M\,(۰,۲,۳,۶)$$

تمرین :

۱- عبارات زیر را ساده کنید.

$F1 = \overline{(A+B+C+D)}$ $\qquad$ $F2 = \overline{(ABCD)}$ $\qquad$ $F3 = \bar{A}B+\bar{A}\bar{B}+AB$

$F4 = \bar{A}\bar{B}+\bar{A}B+A\bar{B}+AB$ $\qquad$ $F5 = \bar{A}\bar{B}\bar{C}+\bar{A}\bar{B}C+\bar{A}B\bar{C}+\bar{A}BC$

جواب :

$$F_1 = (A+B+C+D)' = A'.B'.C'.D'$$

$$F_2 = (A.B.C.D)' = A'+B'+C'+D'$$

$$F_3 = A'B + A'B' + AB = A'(B+B') + AB = A' + AB = (A'+A).(A'+B) = A'+B$$

$$F_4 = A'B' + A'B + AB' + AB = A'(B'+B) + A(B'+B) = A' + A = 1$$

$$F_5 = A'B'C' + A'B'C + A'BC' + A'BC = A'B'(C'+C) + A'B(C'+C)$$

$$= A'B' + A'B = A'(B'+B) = A'$$

****

۲- ابتدا عبارات زیر را به فرم استاندارد تبدیل کنید ؛ سپس شماره مین‌ترم‌ها را بنویسید .

$F_1(A,B,C) = AB + B'C$ $\qquad$ $F_2(A,B,C) = AB' + AC + BC'$ $\qquad$ $F_3(A,B,C,D) = ABCD + BC$

جواب :

$$\begin{aligned} F_1(A,B,C) &= AB + B'C = AB(C+C') + B'C(A+A') \\ &= ABC + ABC' + AB'C + A'B'C = \sum m\,(1,5,6,7) \end{aligned}$$

$$\begin{aligned} F_2(A,B,C) &= AB' + AC + BC' = AB'(C+C') + AC(B+B') + BC'(A+A') \\ &= AB'C + AB'C' + ABC + AB'C' + ABC' + A'BC' \\ &= AB'C + AB'C' + ABC + ABC' + A'BC' \\ &= \sum m(2,4,5,6,7) \end{aligned}$$

$$\begin{aligned} F_3(A,B,C,D) &= ABCD + BC = ABCD + BC(A+A') = ABCD + ABC + A'BC \\ &= ABCD + ABC(D+D') + A'BC(D+D') \\ &= ABCD + ABCD + ABCD' + A'BCD + A'BCD' \\ &= \sum m(6,7,14,15) \end{aligned}$$

****

۳- الف : جدول صحت رابطه‌ی Y = A’B’C’ + A’B’C+AB’C’ را رسم کنید .   ب : خروجی را به صورت ساده شده بنویسید.   ج : مدار سوئیچی آن‌را رسم کنید .   د : مدار منطقی آن‌را رسم کنید .

جواب :

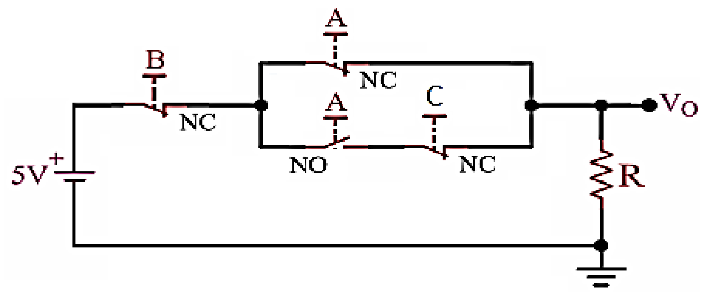
الف :

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

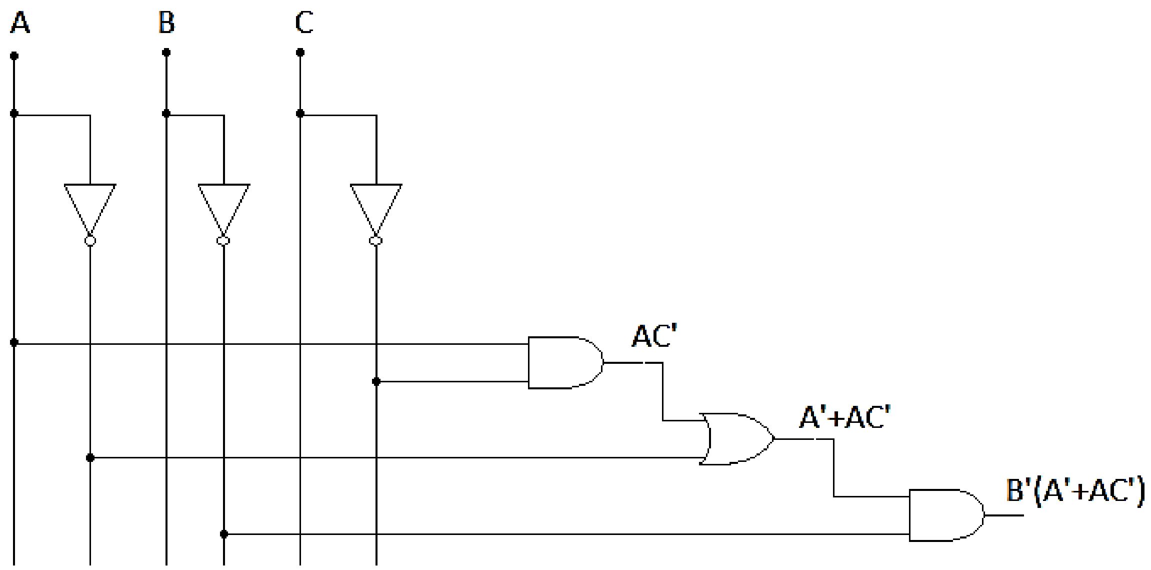
ب :

$$Y = A'B'C' + A'B'C + AB'C' + AB'C$$
$$= A'B'(C + C') + AB'C'$$
$$= A'B' + AB'C' = B'(A' + AC')$$

ج :



د :



****

۴- الف : جدول صحت $F(A,B,C,D) = \Sigma_m (10,11,14,15)$ را رسم کنید .

ب : رابطه ی خروجی را بنویسید. ج : مدار آن را رسم کنید .

د : تابع خروجی را ساده کنید . ه : مدار ساده شده را رسم کنید .

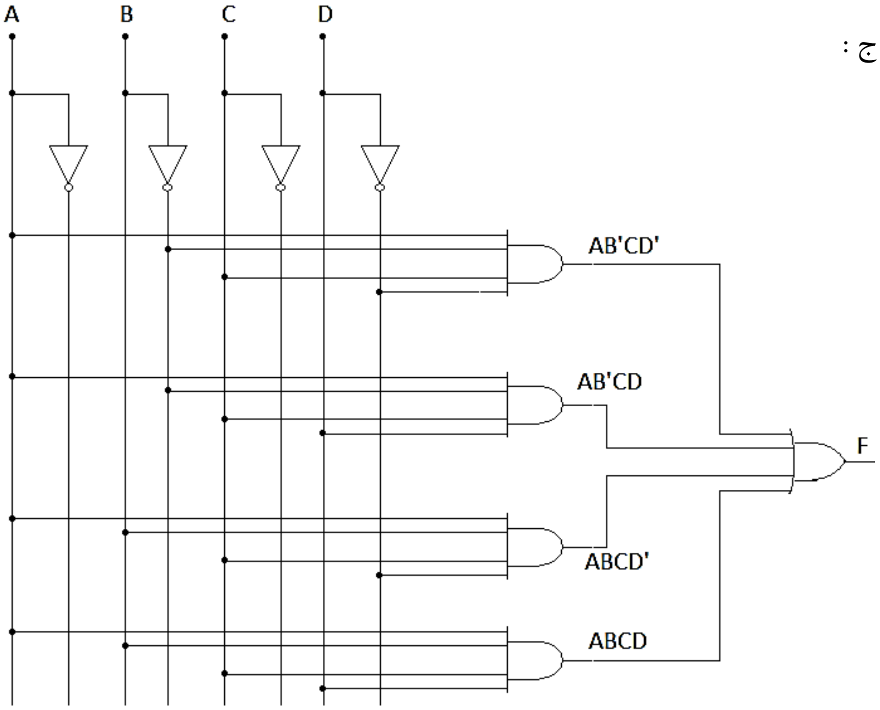
جواب :

الف :

	A	B	C	D	F
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	0	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	1

ب :
$$F = AB'CD' + AB'CD + ABCD' + ABCD$$

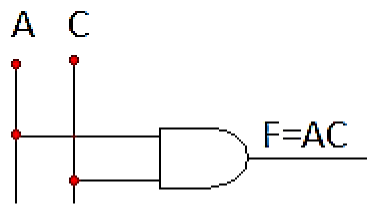
ج :



د :

$$F = AB'CD' + AB'CD + ABCD' + ABCD = AB'C(D + D') + ABC(D + D')$$
$$= AB'C + ABC = AC(B + B') = AC$$

ه :



****

۵- الف : رابطه‌ی خروجی $Y(A, B, C) = \prod (M1, M3, M6)$ را بنویسید.

ب : جدول صحت آن را رسم کنید .　　ج : شماره مین‌ترم‌های آن‌را بنویسید .

د : رابطه مین‌ترم آن را بنویسید .　　ه : مدار آن را رسم کنید .

جواب :

الف :

1	3	6
0 0 1	0 1 1	1 1 0

Y = (A+B+C').(A+B'+C').(A'+B'+C)

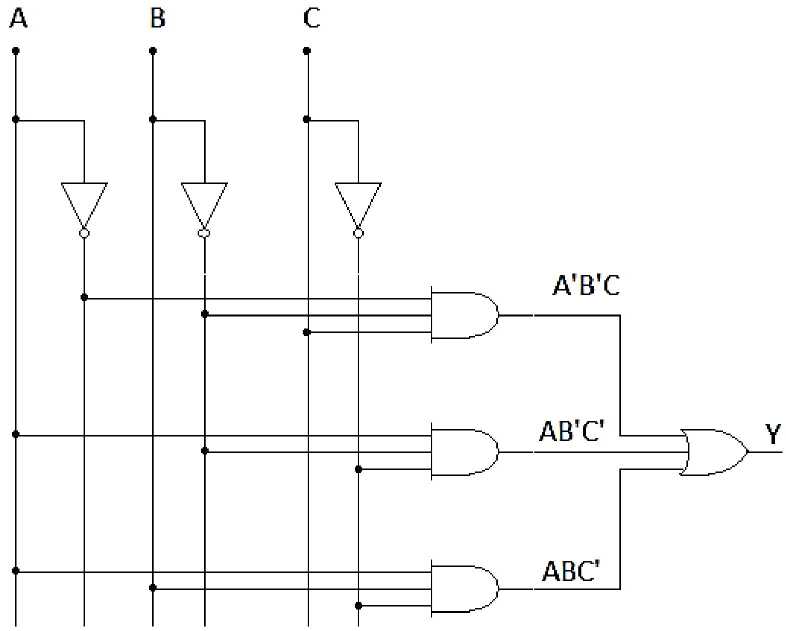
ب :

	A	B	C	Y
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

ج : $Y = \sum \,(m1\,, m4\,, m6\,)$

د : $Y = A'B'C + AB'C' + ABC'$

ه :



****

۶- جدول صحت خروجی Y را در جدول مشاهده می کنید. مین‌ترم و ماکس‌ترم خروجی را بنویسید.

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

جواب :

$$minterm \;\; \rightarrow \;\; Y = A'B'C + AB'C' + AB'C + ABC$$

$$maxterm \;\; \rightarrow \;\; Y = (A + B + C').(A' + B + C).(A' + B + C').(A' + B' + C')$$

****

۷- توابع زیر را به کمک روابط جبر بول ساده کنید.

الف : $F_1(A,B,C) = A'B'C' + AB'C' + AB'C$

ب : $F_2(A,B,C,D) = A'B'CD' + A'B'CD + AB'CD' + AB'CD$

ج : $F_3(A,B,C) = A'BC + A'BC' + AB'C + AB'C'$

جواب :

$$F_1(A,B,C) = A'B'C' + AB'C' + AB'C = A'B'C' + AB'C' + AB'C' + AB'C$$
$$= B'C'(A'+A) + AB'(C'+C) = B'C' + AB'$$

$$F_2(A,B,C,D) = A'B'CD' + A'B'CD + AB'CD' + AB'CD$$
$$= A'B'C(D'+D) + AB'C(D'+D) = A'B'C + AB'C = B'C(A'+A)$$
$$= B'C$$

$$F_3(A,B,C) = A'BC + A'BC' + AB'C + AB'C' = A'B(C+C') + AB'(C+C')$$
$$= A'B + AB' = A \oplus B$$

****

۸- تابع زیر را به فرم استاندارد بول (عبارت‌های مین‌ترم) درآوردید.

$$F(A,B,C,D) = ABC' + AD' + ABC$$

جواب :

$$F(A,B,C,D) = ABC' + AD' + ABC = ABC'(D+D') + AD'(B+B') + ABC(D+D') = ABC'D + ABC'D' + ABD' + AB'D' + ABCD + ABCD' = ABC'D + ABC'D' + ABD'(C+C') + AB'D'(C+C') + ABCD + ABCD' = ABC'D + ABC'D' + ABCD' + ABC'D' + AB'CD' + AB'C'D' + ABCD + ABCD' = ABC'D + ABC'D' + ABCD' + AB'CD' + AB'C'D' + ABCD = \sum m(13,12,14,10,8,15)$$

****

۹- تابع زیر را به فرم استاندارد بول (عبارت ماکس‌ترم) درآوردید.

$$\mathrm{F(A,B,C)} = (\mathrm{A}+\mathrm{B}')(\mathrm{B}+\mathrm{C}') = (A+B'+CC')(B+C'+AA')$$

جواب :

$$\mathrm{F(A,B,C)} = (\mathrm{A}+\mathrm{B}')(\mathrm{B}+\mathrm{C}') = (A+B'+CC')(B+C'+AA')$$
$$= (A+B'+C)(A+B'+C')(A+B+C')(A'+B+C') = \prod M(2,3,1)$$

****

۱۰- منزلی با سه سنسور را در نظر بگیرید. فرض کنید می‌خواهیم مداری طراحی کنیم که اگر حداقل دو سنسور فعال شد دزدگیر به صدا درآید. مطلوب است:

الف) رسم و تکمیل جدول صحت ب) نمایش خروجی به صورت مین‌ترم‌ها

ج) نمایش خروجی به صورت ماکس‌ترم‌ها د) طراحی مدار منطقی بدون ساده سازی آن

ه) طراحی مدار سوئیچی بدون ساده سازی آن

جواب :

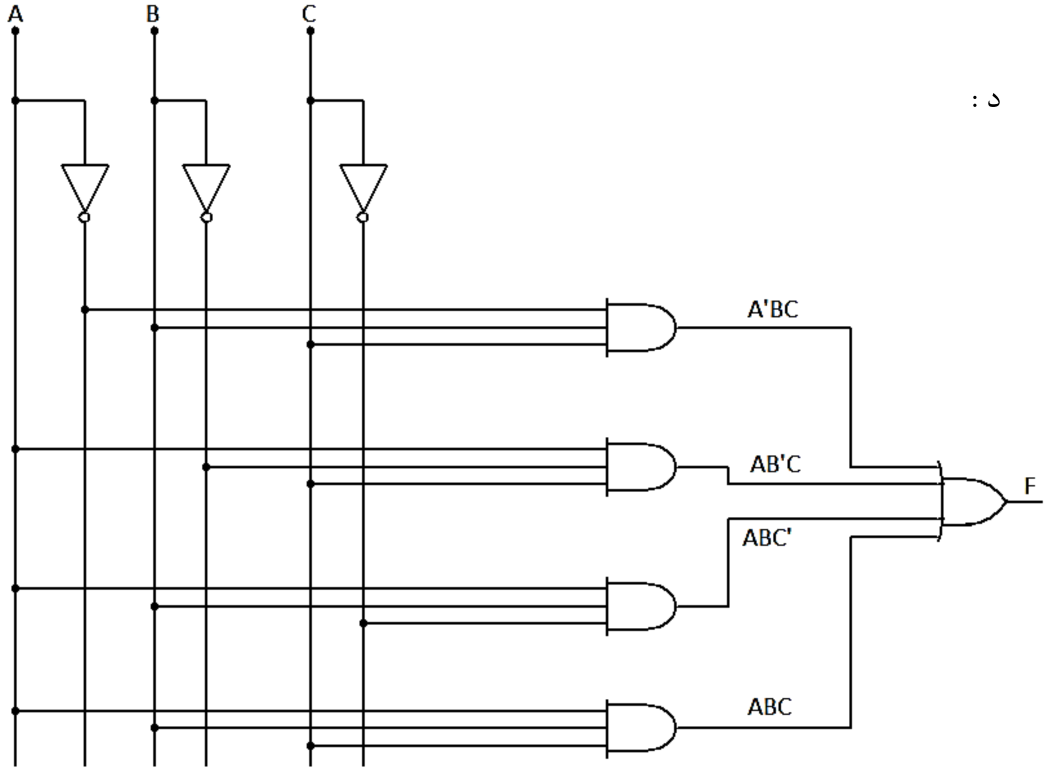
الف :

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

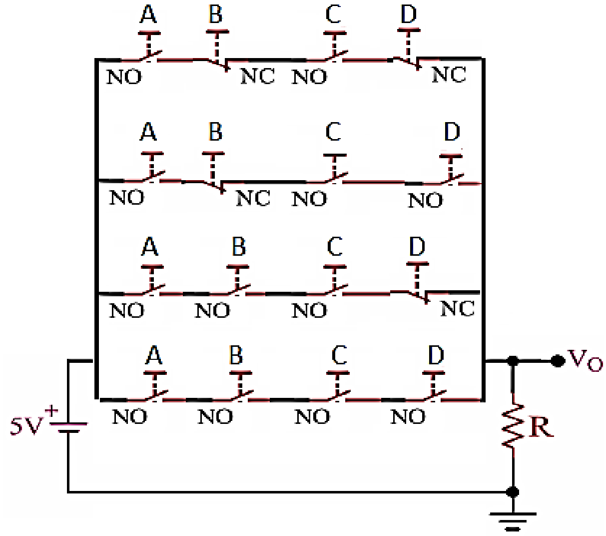
ب : $F = A'BC + AB'C + ABC' + ABC$

ج : $F = (A + B' + C').(A' + B + C').(A' + B' + C).(A' + B' + C')$

د :



ه :

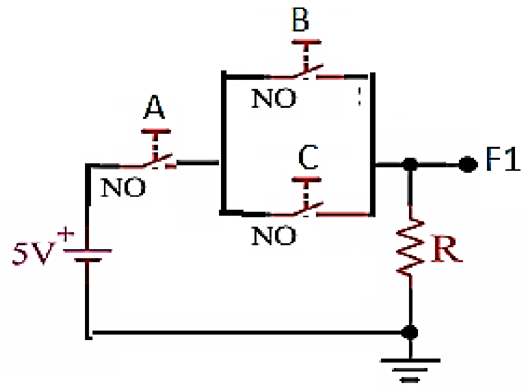


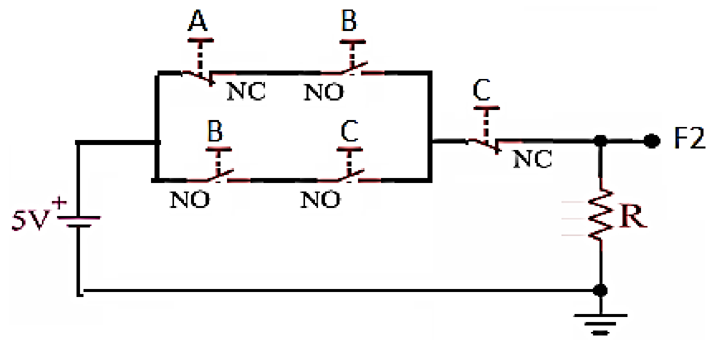
۱۱- مدار سوئیچی هر یک از مواد زیر را رسم کنید.

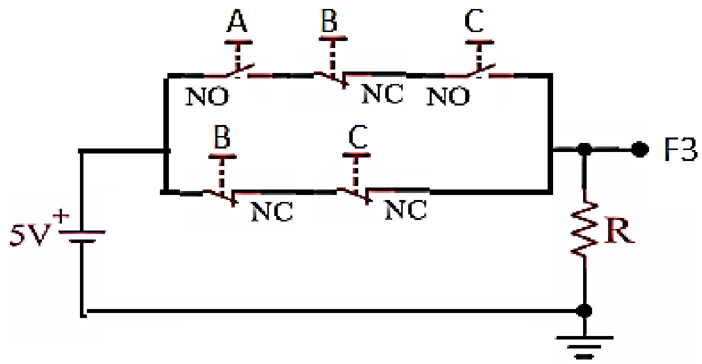
$F1 = A(B + C)$ $\qquad$ $F2 = [A'B + BC]\,[C']$

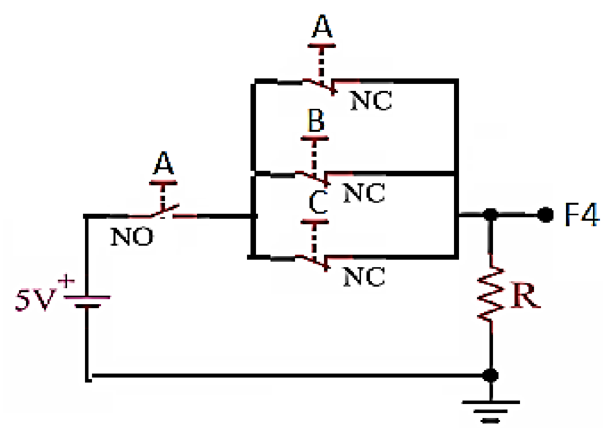
$F3 = [AB'C] + [B'C']$ $\qquad$ $F4 = A[A' + B' + C']$

جواب :









فصل چهارم : نقشه کارنو و روش طراحی مدارهای ترکیبی

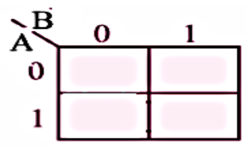
ساده‌سازی توابع با استفاده از نقشه کارنو :

همانطور که قبلاً نیز گفته شد، اساس ساده‌سازی توابع بول بر مبنای فاکتورگیری و حذف متغیرهاست. اما این کار می‌تواند بسیار پیچیده و وقت گیر باشد. لذا برای ساده‌سازی توابع ، اغلب از نقشه کارنو استفاده می شود. نقشه کارنو روشی است که به صورت کلاسیک، ساده‌ترین شکل یک تابع دیجیتالی را نشان می‌دهد. برای ساده‌سازی یک تابع با نقشه کارنو مراحل زیر را به ترتیب طی می‌کنیم.

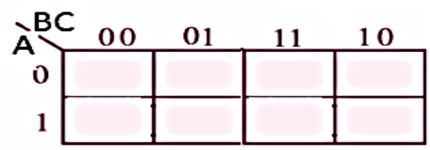
۱- استاندارد کردن ضابطه تابع

۲- تکمیل جدول صحت

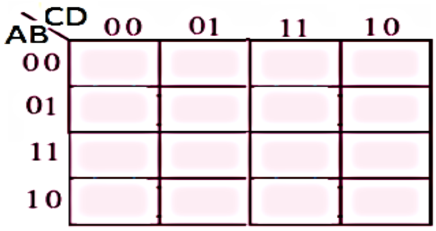
۳- تشکیل نقشه کارنو با توجه به تعداد متغیرهای ورودی



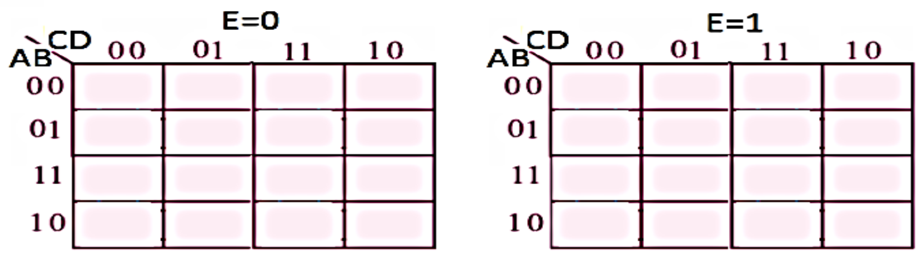
نقشه کارنو ۲ متغیره



نقشه کارنو ۳ متغیره



نقشه کارنو ۴ متغیره



نقشه کارنو ۵ متغیره

۴- تکمیل نقشه کارنو با استفاده از جدول صحت

۵- دسته بندی یک‌ها در نقشه کارنو

الف) دسته‌ها باید ۲ تایی یا ۴ تایی یا ۸ تایی یا ۱۶ تایی باشند.

ب) دسته‌های بزرگتر اولویت دارند.

ج) تمام یک‌ها باید پوشش داده شوند.

د) هر سلول یک ، می تواند در چند دسته عضو باشد.

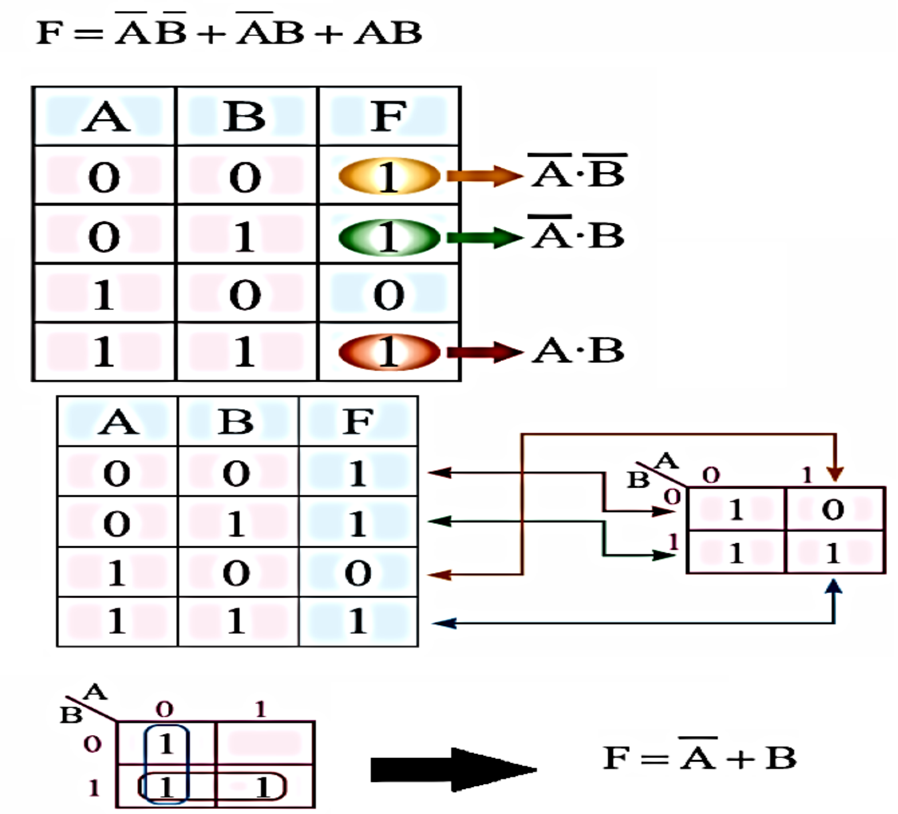
ه) تمام اعضای هر دسته باید مجاور باشند.

و) خانه‌های متناظر موجود در لبه‌ها، و گوشه‌ها مجاور محسوب می‌شوند.

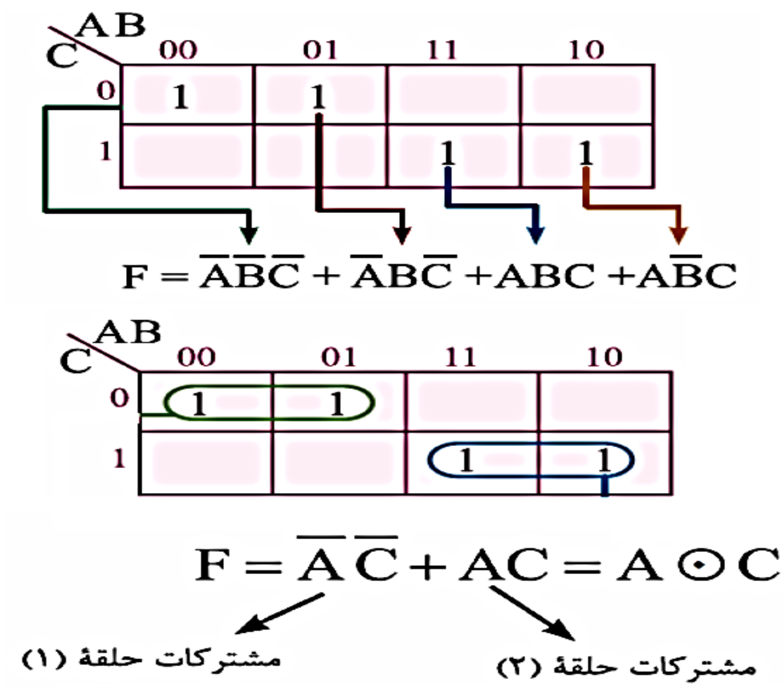
۶- عامل‌های مشترک هر دسته را می‌نویسیم.

۷- همه را با هم جمع می کنیم.

مثال : تابع $F = A'B' + A'B + AB$ را ساده کنید.

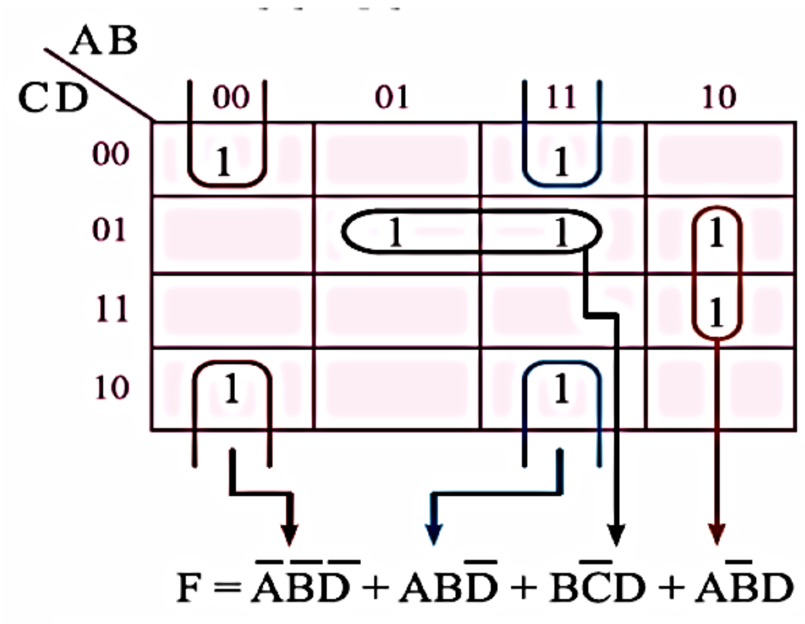


مثال : تابع $\mathrm{F = A'B'C' + A'BC' + ABC + AB'C}$ را به کمک نقشه کارنو ساده کنید.



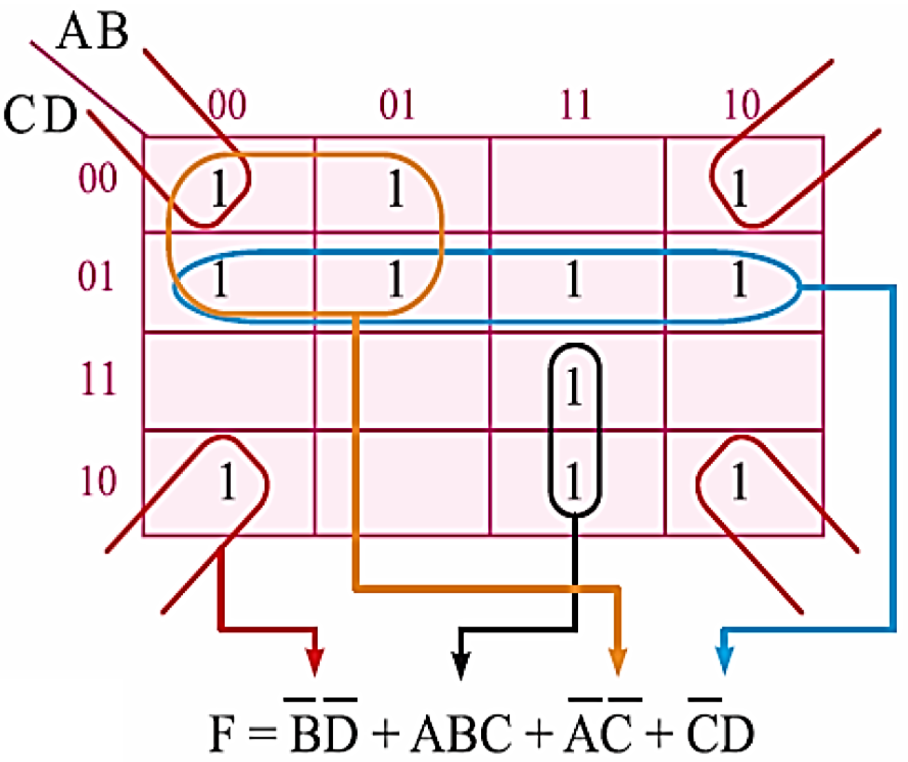
مثال : تابع زیر را ساده کنید.

$$F = A'B'C'D' + A'B'CD + A'BC'D + A'BCD + A'BCD' + ABCD' + AB'C'D' + AB'CD$$



مثال : تابع زیر را ساده کنید.

$$F(A,B,C,D) = A'B'C'D' + A'B'C'D + A'B'CD' + A'BC'D' + A'BC'D + A'BCD + A'BCD' + ABCD + AB'C'D' + AB'CD + AB'CD'$$



مثال : تابع بول زیر را ساده کنید.

F(A,B,C,D,E)= (0,2,4,6,9,13,21,23,25,29,31)

E=0

AB \ CD	00	01	11	10
00	1	1	1	1
01				
11				
10				

E=1

AB \ CD	00	01	11	10
00				
01	1			1
11	1		1	1
10			1	1

$$F = A'B'E' + BD'E + ACE$$

نکته: اگر در یک تابع ، تعداد صفرها کمتر باشد، میتوان اول صفرها را ترکیب کرده و F' را به دست آورد، سپس با معکوس کردن آن ، تابع F را یافت.

مثال: یک دزدگیر ۴ سنسوره را طوری طراحی کنید که اگر حداقل یکی از آن‌ها فعال شد آژیر به صدا درآید.

$$F = \Sigma_m(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16)$$

AB \ CD	00	01	11	10
00	0	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$$F' = A'B'C'D' \Rightarrow F = \overline{(A'B'C'D')} = A + B + C + D$$

حالات بی‌اهمّیّت :

یک‌ها و صفرهای نقشه ، بیانگر ترکیبی از متغیرهاست که به ترتیب تابع را ۱ و یا ۰ می‌کنند. معمولاً ترکیبات حاصل از جدول درستی ، حالاتی هستند که تحت آن‌ها تابع برابر یک می باشد و در سایر مکان‌های نقشه ، مقدار تابع ۰ فرض می‌گردد. این فرض همیشه صحیح نیست. زیرا در بعضی از کاربردها ترکیبات معینی از متغیرهای ورودی هرگز وجود ندارد. تابعی که خروجی‌های نامشخص یا بلا استفاده در ازای بعضی از ترکیبات ورودی دارد به توابع ناقص معروف‌اند. مین‌ترم‌های نامعین را در توابع، حالات بی‌اهمیت می‌نامیم. این حالات بی اهمیت در نقشه برای ساده سازی بیشتر عبارت بول بکار می‌روند.

حالت بی اهمیت را با X نمایش می‌دهیم. بنابراین یک X بیانگر این واقعیت است که ما به ازای مین‌ترم خاصی به ۰ یا ۱ شدن F اهمیتی نمی‌دهیم.

به هنگام انتخاب مربع‌های همجوار در نقشه، برای ساده نمودن تابع، با این ایده که ساده‌ترین عبارت حاصل گردد، X ها را برابر ۰ و یا ۱ فرض می نماییم.

مثال: مداری طراحی کنید که یک عدد یک رقمی (۰ تا۹) از ورودی بگیرید، و اعداد فرد را مشخص کند.

ابتدا جدول صحت را می‌کشیم . سپس نقشه کارنو را رسم می‌کنیم و با دسته‌بندی یک‌ها تابع خروجی را ساده می‌کنیم .
برای ساده‌تر شدن خروجی ، X های درون دسته را یک درنظر می‌گیریم .

	A	B	C	D	F
0	0	0	0	0	X
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	X
11	1	0	1	1	X
12	1	1	0	0	X
13	1	1	0	1	X
14	1	1	1	0	X
15	1	1	1	1	X

AB \ CD	00	01	11	10
00	X	1	1	0
01	0	1	1	0
11	X	X	X	X
10	0	1	X	X

$$F = D$$

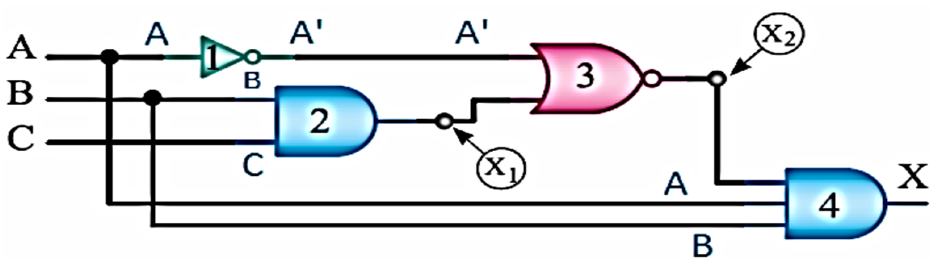
مدار ترکیبی :

مدار ترکیبی به مداری اطلاق می شود که وضعیت خروجی‌های آن در هر لحظه، منحصراً به وضعیت ورودی‌های آن، در همان لحظه بستگی دارد. در چنین مداری، هیچ خروجی‌ای به هیچ ورودی‌ای از مدار برگشت داده نمی شود. (فیدبک نداریم)

آنالیز مدارهای ترکیبی :

برای بررسی رفتار یک مدار ترکیبی، ابتدا ورودی‌ها و خروجی‌های تمام گیت‌ها را نام‌گذاری می‌کنیم. سپس تک‌تک آن‌ها را از ابتدا تا انتها به دست می آوریم.

مثال: در مدار شکل زیر نخست تابع خروجی X را بر حسب متغیرهای A ,B و C به صورت مجموع حاصل ضرب‌ها به دست آورید و سپس جدول صحت مدار را رسم کنید.



$$X1 = BC$$

$$X2 = (A'+X1)' = (A'+BC)'$$
$$= A(BC)' = A(B'+C') = AB'+AC'$$

$$X = BAX2 = AB(AB'+AC')$$
$$= ABB'+ABC' = ABC'$$

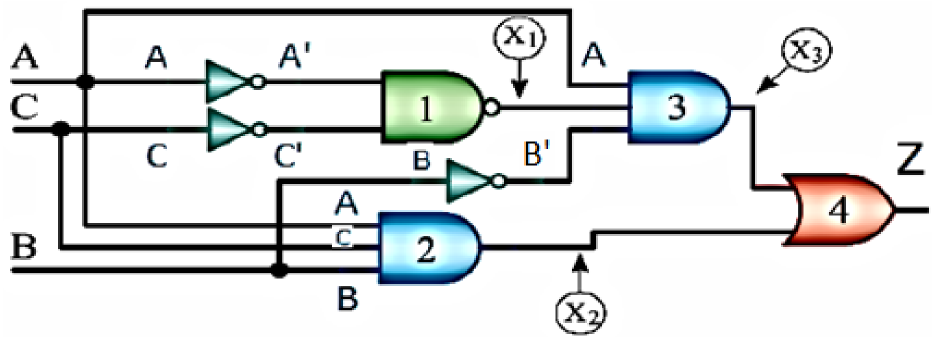
A	B	C	X
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**0**
1	0	0	**0**
1	0	1	**0**
1	1	0	**1**
1	1	1	**0**

مثال: در مدار زیر :

الف) ابتدا تابع خروجی را بیابید. ب) آن‌را به استاندارد مین‌ترم‌ها تبدیل کنید.

ج) جدول صحت آن را رسم کنید. د) آن‌را را به ساده‌ترین شکل ممکن درآورید.

ه) مدار ساده شده آن‌را رسم کنید.



الف)

$$X_1=\overline{\overline{A}.\overline{C}}=A+C$$

$$X_2=A.x_1.\overline{B}=A(A+C)\overline{B}=A\overline{B}+A\overline{B}C$$

$$=A\overline{B}(1+C)=A\overline{B}$$

$$X_3 = ABC$$

$$Z = X_2 + X_3$$

$$Z = ABC + A\overline{B}$$

ب)

$$Z = ABC + A\overline{B} = ABC + A\overline{B}(\overline{C}+C)$$

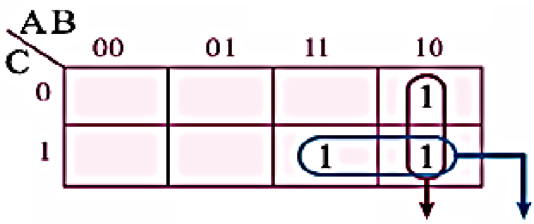
$$= \underset{\text{سطر ۷}}{ABC} + \underset{\text{سطر ۴}}{A\overline{B}\overline{C}} + \underset{\text{سطر ۵}}{A\overline{B}C}$$

ج)

$$Z = \Sigma_m (m_7, m_4, m_5)$$

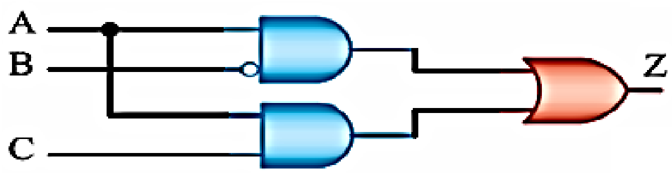
A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

د)

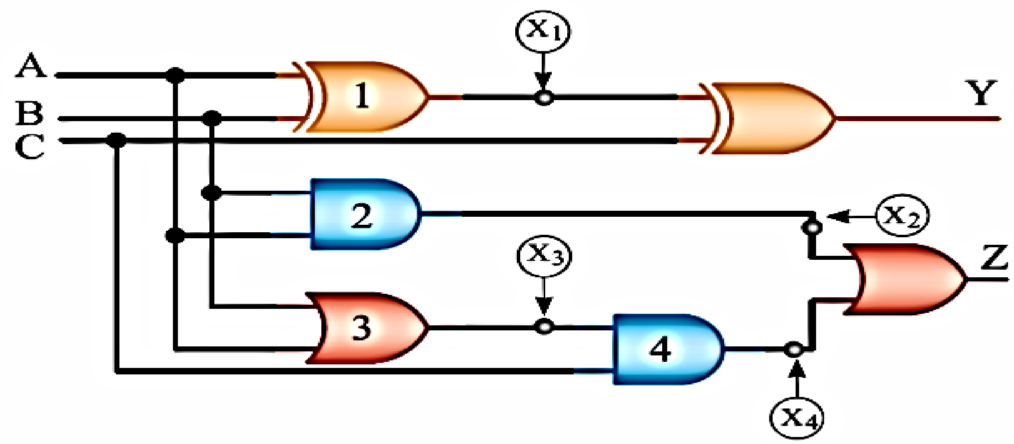


$$Z = A\overline{B} + AC$$

ه)



مثال: در شکل زیر هر یک از توابع Y و Z را بر حسب متغیرهای A ,B و C در فرم مجموع حاصل ضرب‌ها به دست آورید. سپس جدول صحت مدار را رسم کنید.



$X_۱ = A \oplus B$

$Y = x_۱ \oplus C$

$Y = A \oplus B \oplus C$

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

$Y = \Sigma m\ (۱,۲,۴,۷)$

$x_۲=AB$

$x_۳ =A+B$

$x_۴=Cx_۳$

$Z=x_۲+x_۴\ =AB+Cx_۳$

$Z=AB+C(A+B)=AB+AC+BC$

$Z = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$

$Z = \Sigma m\ (۳,۵,۶,۷)$

شماره‌های سطر	A	B	C	Y	Z
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

مروری بر روش طراحی مدارهای ترکیبی :

چنان‌چه قبلاً نیز ذکر شد، برای طراحی یک مدار ترکیبی، باید به ترتیب زیر عمل کنیم.

الف) جدول صحت مدار را به دست آوریم.

ب) به کمک جدول صحت مدار هر یک از توابع خروجی را در فرم مجموع حاصل ضرب‌ها (یا حاصل ضرب مجموع‌ها) بیان کنیم.

ج) هر یک از توابع فوق را به روش جبری یا با استفاده از جدول کارنو ساده می‌کنیم.

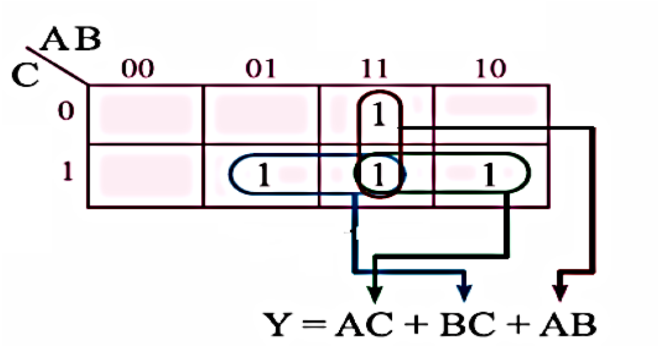
د) مدار منطقی یا سوئیچی مورد نظر را پیاده‌سازی می‌کنیم.

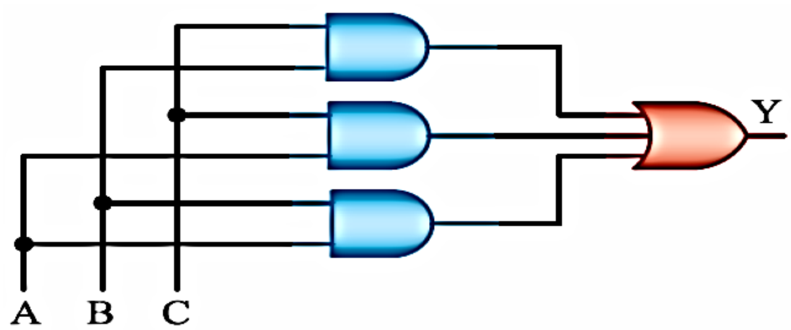
ه) هر یک از توابع ساده شده را به صورت یک ترکیب NAND یا NOR پیاده‌سازی می‌کنیم.

مثال : مداری با سه ورودی A,B و C و یک خروجی Y طرح کنید که در حالت‌هایی که اکثریت نسبی ورودی‌ها یک باشد، خروجی آن یک شود. مدار را در ساده ترین فرم مجموع حاصل ضرب‌ها اجرا کنید.

شماره سطرها	A	B	C	Y
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

$$Y = A'BC + AB'C + ABC' + ABC = \Sigma m(3,5,6,7)$$





مثال : مداری طراحی کنید که یک عدد یک رقمی را از ورودی بگیرد و اعداد بخش پذیر بر ۲ را در خروجی F1، اعداد بخش پذیر بر ۳ را در خروجی F2 و اعداد بخش پذیر بر ۴ را در خروجی F3، نشان دهد.

	A	B	C	D	F1	F2	F3
۰	0	0	0	0	0	0	0
۱	0	0	0	1	1	1	1
۲	0	0	1	0	1	0	0
۳	0	0	1	1	0	1	0
۴	0	1	0	0	1	0	1
۵	0	1	0	1	0	0	0
۶	0	1	1	0	1	1	0
۷	0	1	1	1	0	0	0
۸	1	0	0	0	1	0	1
۹	1	0	0	1	0	1	0
۱۰	1	0	1	0	X	X	X
۱۱	1	0	1	1	X	X	X
۱۲	1	1	0	0	X	X	X
۱۳	1	1	0	1	X	X	X
۱۴	1	1	1	0	X	X	X
۱۵	1	1	1	1	X	X	X

F1 :

AB \ CD	00	01	11	10
00	0	1	0	1
01	1	0	0	1
11	X	X	X	X
10	1	0	X	X

F1 = CD' + BC'D' + AC'D' + A'B'C'D

F2 :

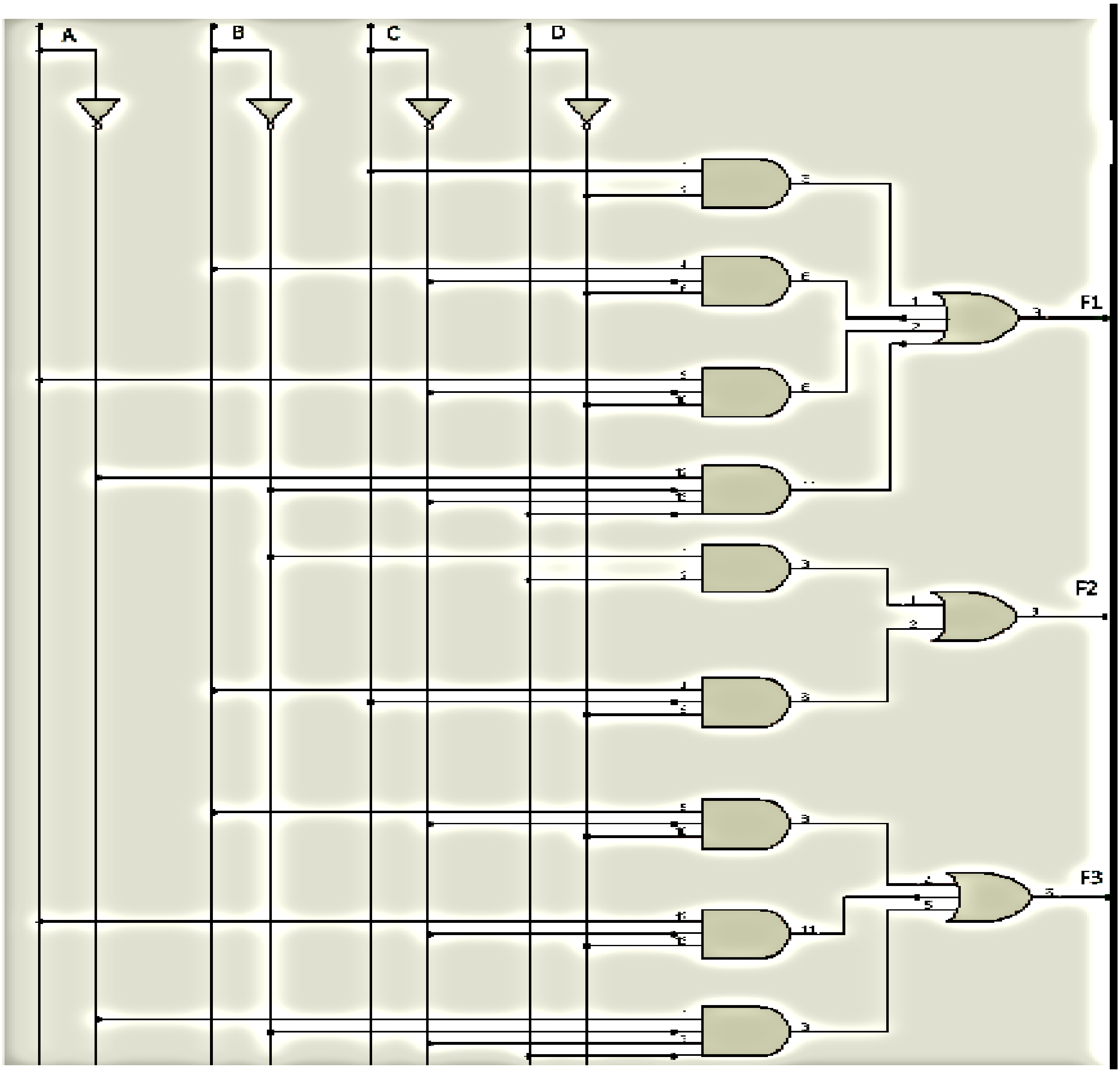
AB \ CD	00	01	11	10
00	0	1	1	0
01	0	0	0	1
11	X	X	X	X
10	0	1	X	X

F2 = B'D + BCD'

F3 :

AB \ CD	00	01	11	10
00	0	1	0	0
01	1	0	0	0
11	X	X	X	X
10	1	0	X	X

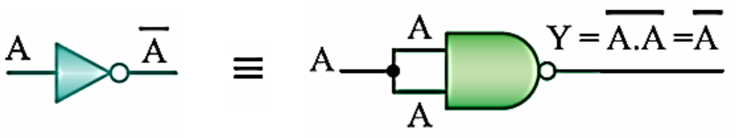
F3 = BC'D' + AC'D' + A'B'C'D



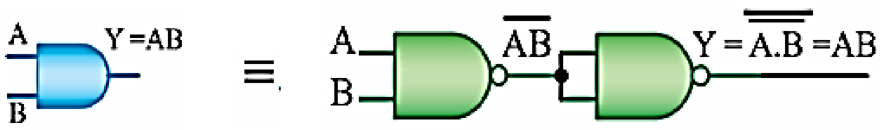
نکته : همانگونه که LED , LCD از معروف‌ترین استانداردهای ساخت تلویزیون هستند، TTL و CMOS نیز دو استاندارد معروف در ساخت گیت‌های منطقی‌اند. در استاندارد TTL بهترین گیت، گیت NAND و در استاندارد CMOS بهترین گیت، گیت NOR است. علّت این امر را میتوان در سرعت بالا، تکنولوژی ساخت آسان، قیمت ارزان، زمان تأخیر کم و.... جستجو کرد. از این رو اغلب سعی می‌شود در طراحی مدارهای منطقی، فقط از این دو گیت استفاده گردد.

تبدیل گیت‌ها به گیت NAND :

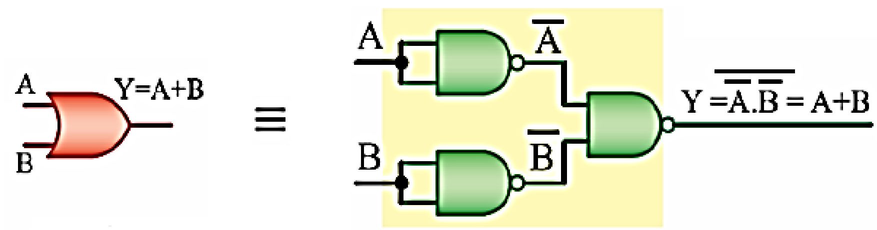
نکته ۱ : برای تبدیل گیت NOT به گیت NAND کافی است دو پایه ورودی NAND را به هم وصل کنیم. داریم:



نکته ۲ : برای تبدیل گیت AND به گیت NAND کافی است خروجی NAND را NOT کنیم. داریم:



نکته ۳: برای تبدیل گیت OR به گیت NAND کافی است در پایه‌های ورودی NAND یک NOT اضافه کنیم. داریم:



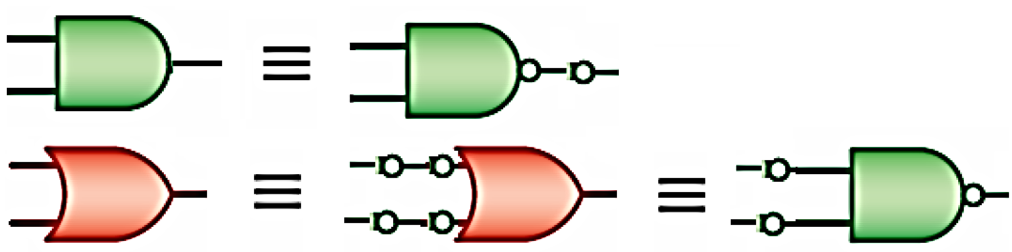
روش کار :

۱- بعد از تمام گیت‌های AND دو عدد NOT می‌گذاریم.

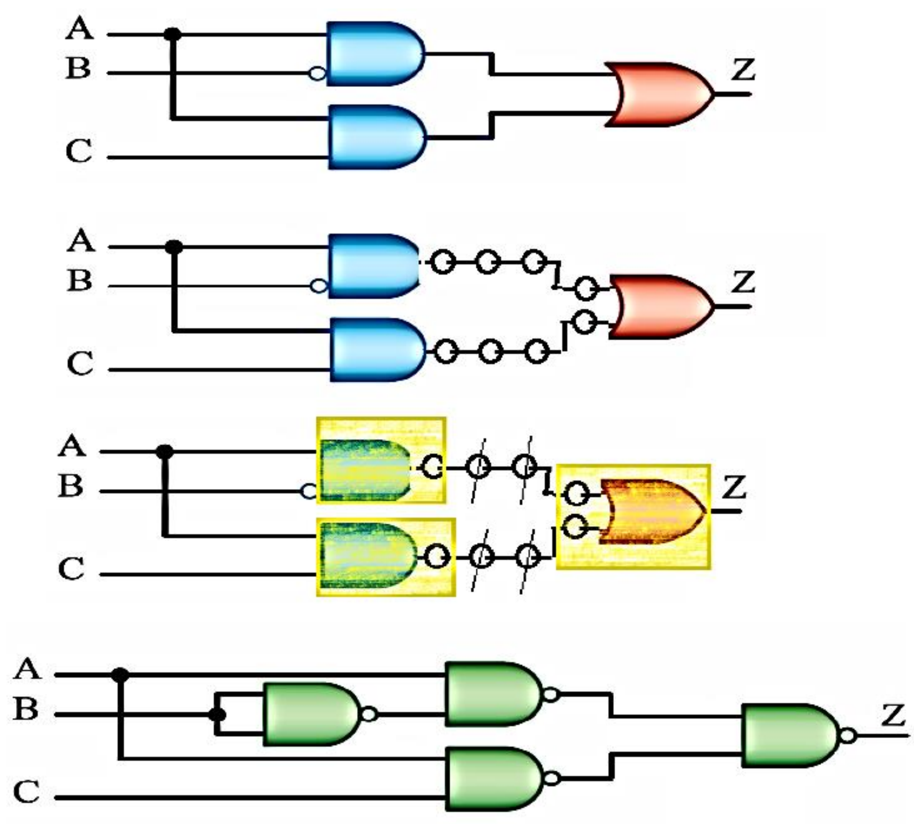
۲- قبل از هر ورودی گیت OR دو عدد NOT می‌گذاریم.

۳- NOT های اضافه را خنثی می‌کنیم.

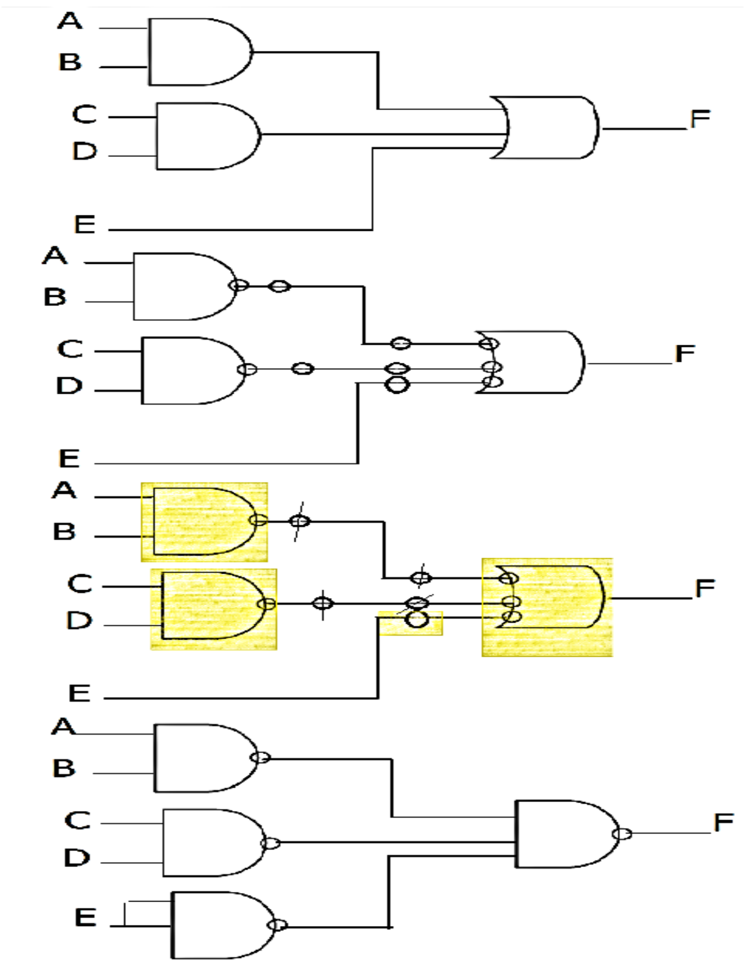
۴- تمام گیت‌ها را به گیت NAND تبدیل می‌کنیم.



مثال : در مدار زیر تمام گیت‌ها را به گیت NAND تبدیل کنید.



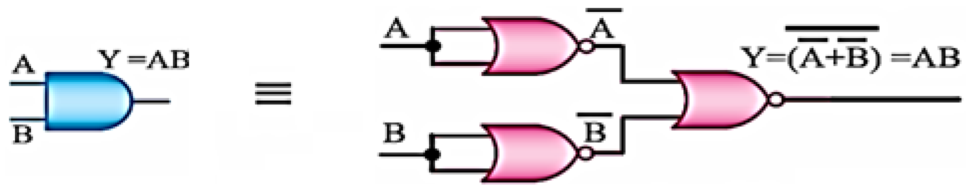
مثال : در مدار زیر تمام گیت‌ها را به گیت NAND تبدیل کنید.



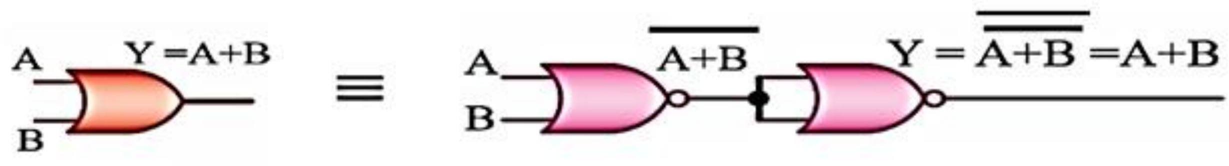
تبدیل گیت‌ها به گیت NOR :

نکته ۱ : برای تبدیل گیت NOT به گیت NOR کافی است پایه‌های ورودی NOR را به هم وصل کنیم. داریم :

نکته ۲ : برای تبدیل گیت AND به گیت NOR کافی است به ورودی‌های NOR یک NOT اضافه کنیم. داریم :

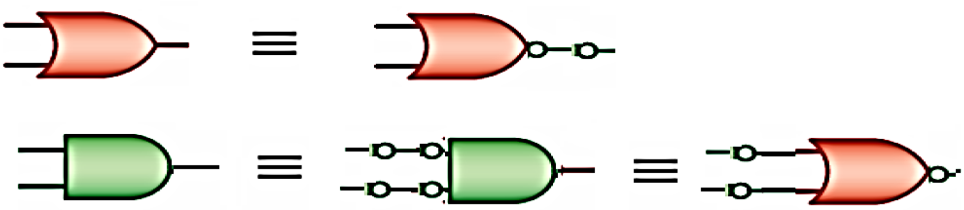


نکته ۳ : برای تبدیل گیت OR به گیت NOR کافی است خروجی NOR را NOT کنیم. داریم :

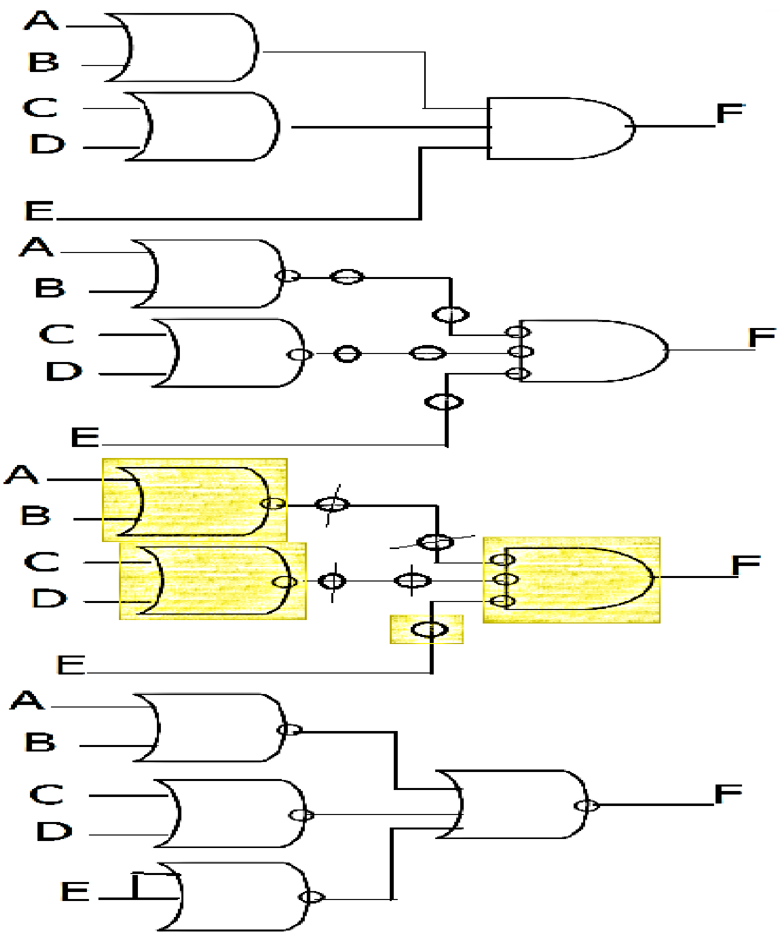


روش کار :

۱- بعد از تمام گیت‌های OR دو عدد NOT می‌گذاریم.
۲- قبل از هر ورودی گیت AND دو عدد NOT می‌گذاریم.
۳- NOT های اضافی را خنثی می‌کنیم.
۴- تمام گیت‌ها را به گیت NOR تبدیل می‌کنیم.



مثال : در مدار زیر تمام گیت‌ها را به گیت NOR تبدیل کنید.



تمرین:

۱- تابع $F(A,B,C) = A'B'C' + AB'C' + A'BC' + ABC'$ را به کمک جدول کارنو ساده کنید.

جواب :

A \ BC	00	01	11	10
0	1	0	0	1
1	1	0	0	1

$F = C'$

****

۲- تابع مین ترم $F(A,B,C) = \Sigma m(m1, m2, m5, m6, m7)$ را به کمک جدول کارنو ساده کنید.

جواب :

A \ BC	00	01	11	10
0	0	1	0	1
1	0	1	1	1

$F = B'C + BC' + AB$

****

۳- در نقشه کارنو زیر ، عبارت خروجی را به ساده‌ترین شکل بنویسید.

C \ AB	00	01	11	10
0	1	0	0	1
1	1	0	1	1

C \ AB	00	01	11	10
0	1	0	0	1
1	1	0	1	1

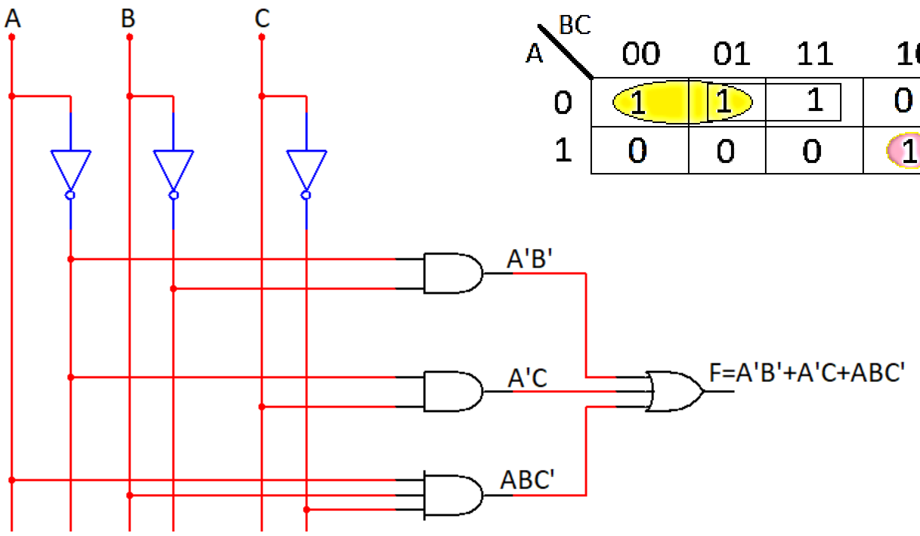
$F = B' + AC$

****

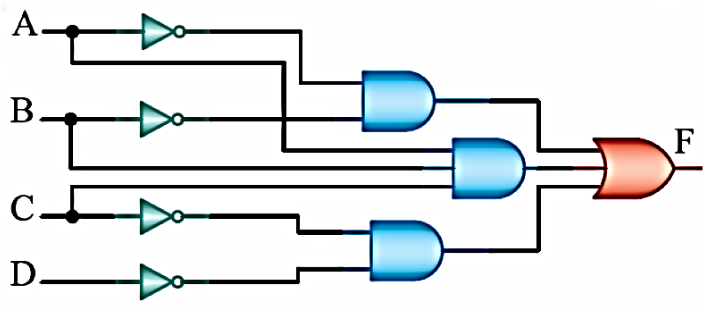
۴- تابع $F(A,B,C) = \Sigma m(0,1,3,6)$ را با استفاده از جدول کارنو ساده نمایید و مدار آن را رسم کنید.

A \ BC	00	01	11	10
0	1	1	1	0
1	0	0	0	1

$F = A'B' + A'C + ABC'$



۵- در مدار شکل زیر :



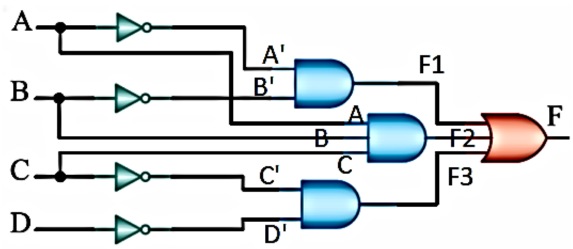
الف : تابع خروجی را بیابید .

ب : آن را استاندارد کنید .

ج : توسط نقشه کارنو آن را به ساده‌ترین شکل ممکن درآورید .

د : مدار ساده شده را بازطراحی کنید .

جواب :



F1 = A'B'
F2 = ABC
F3 = C'D'
F = F1 + F2 + F3
= A'B' + ABC + C'D'

AB \ CD	00	01	11	10
00	1	1	1	1
01	1	0	0	0
11	1	0	1	1
10	1	0	0	0

F = C'D' + A'B' + ABC

F = A'B'(C+C') + ABC(D+D') + C'D'(A+A')
= A'B'C + A'B'C' + ABCD + ABCD' + AC'D' + A'C'D'
= A'B'C(D+D') + A'B'C'(D+D') + ABCD + ABCD' + AC'D'(B+B') + A'C'D'(B+B')
= A'B'CD + A'B'CD' + A'B'C'D + A'B'C'D' + ABCD + ABCD' + ABC'D' + AB'C'D'
+ A'BC'D' + A'B'C'D' = $\sum$m (3 , 2 , 1 , 0 , 15 , 14 , 12 , 8 , 4)

چنان‌چه دیده می‌شود همان تابع قبل به دست آمد . یعنی خودش در ساده‌ترین شکل ممکن بوده است . پس مدار از این ساده‌تر نمی‌شود .

****

۶- تابع زیر را با استفاده از جدول کارنو به ساده‌ترین شکل بنویسید.

F (A,B,C,D) = Σm(1,3,6,10,11,13)

جواب :

AB \ CD	00	01	11	10
00	0	1	1	0
01	0	0	0	1
11	1	1	0	0
10	0	0	0	1

F = A'B'D + ABC' A'BCD' + AB'CD'

****

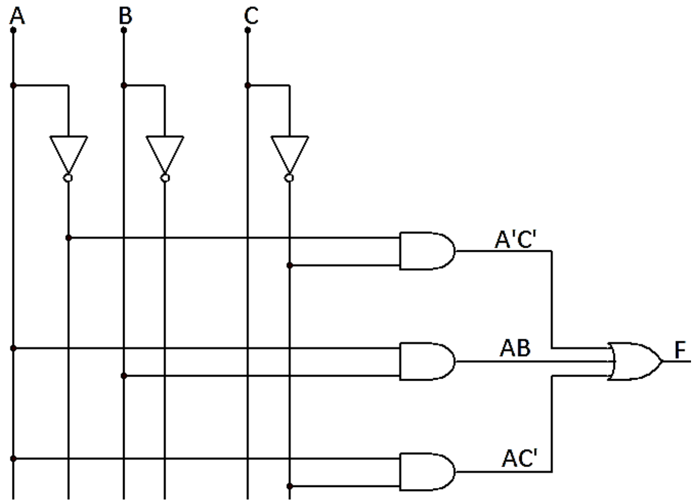
۷- مداری طراحی کنید که جدول صحت زیر را ایجاد کند.

A	B	C	F	LED
0	0	0	0	
0	0	1	1	
0	1	0	0	
0	1	1	1	
1	0	0	1	
1	0	1	0	
1	1	0	1	
1	1	1	1	

جواب :

A \ BC	00	01	11	10
0	0	1	1	0
1	1	0	1	1

F = A'C' + AB + AC'



****

۸- ساده‌ترین عبارت استخراجی از نقشه‌های کارنوی زیر را بنویسید.

الف

C \ AB	00	01	11	10
0	1	1	0	1
1	0	1	1	1

ب

C \ AB	00	01	11	10
0	1	0	0	1
1	1	0	1	1

پ

CD \ AB	00	01	11	10
00	1	1	0	1
01	1	1	1	1
11	1	0	1	1
10	1	0	0	1

ت

CD \ AB	00	01	11	10
00	1	0	0	1
01	1	1	0	0
11	0	1	1	0
10	0	1	1	0

الف

C \ AB	00	01	11	10
0	1	1	0	1
1	0	1	1	1

F = A'C' + BC + AB'

ب

C \ AB	00	01	11	10
0	1	0	0	1
1	1	0	1	1

F = B' + AC

پ

CD \ AB	00	01	11	10
00	1	1	0	1
01	1	1	1	1
11	1	0	1	1
10	1	0	0	1

F = B' + A'C' + ABD

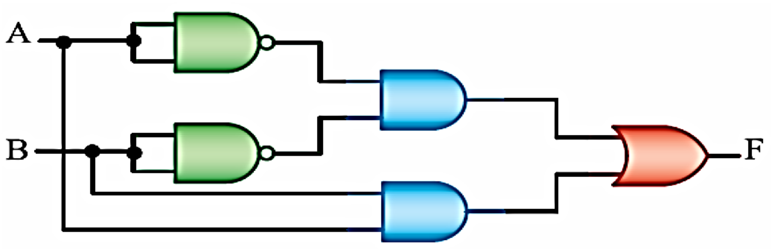
ت

CD \ AB	00	01	11	10
00	1	0	0	1
01	1	1	0	0
11	0	1	1	0
10	0	1	1	0

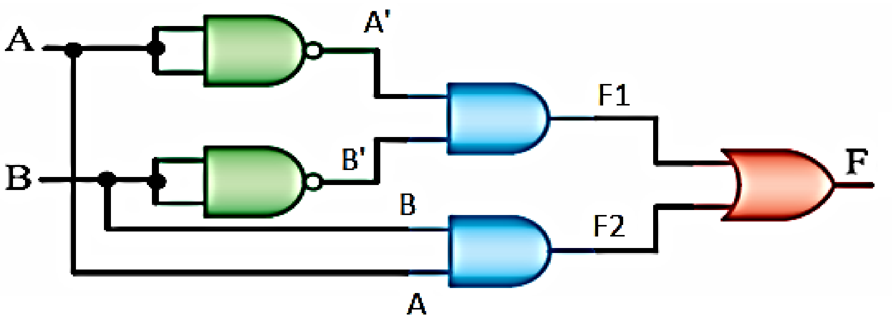
F = BC + A'C'D + B'C'D'

****

۹- در مدار شکل زیر پس از به دست آوردن رابطه‌ی خروجی و ساده کردن آن، بگویید چه مدار ساده‌ای را می‌توان جایگزین کرد؟



جواب :



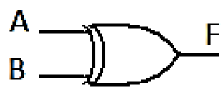
F1 = A'B'

F2 = AB

F = F1 + F2

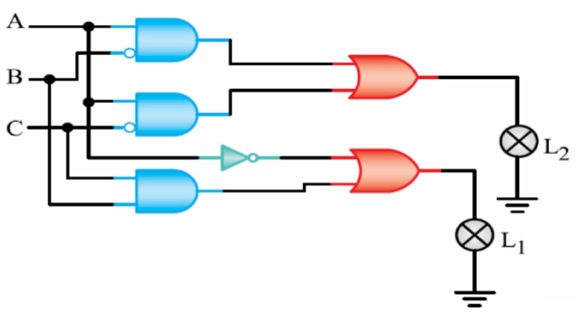
= A'B' + AB

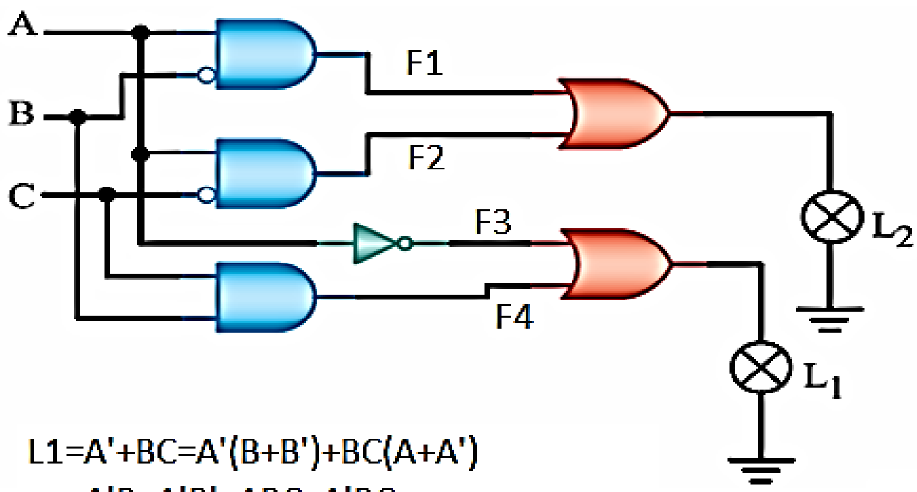
= A ⊕ B



****

۱۰- در مدار شکل زیر به ازای کدام حالت ورودی‌ها لامپ L_1 خاموش و لامپ L_2 روشن می شود؟





جواب :

$F1 = AB'$

$F2 = AC'$

$F3 = A'$

$F4 = BC$

$L1 = F3 + F4 = A' + BC$

$L2 = F1 + F2 = AB' + AC'$

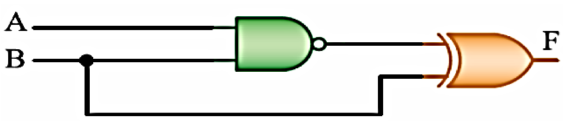
$$
\begin{aligned}
L1 &= A'+BC = A'(B+B')+BC(A+A') \\
&= A'B+A'B'+ABC+A'BC \\
&= A'B(C+C')+A'B'(C+C')+ABC+A'BC \\
&= A'BC+A'BC'+A'B'C+A'B'C'+ABC+A'BC
\end{aligned}
$$

$$
\begin{aligned}
L2 &= AB'+AC' = AB'(C+C')+AC'(B+B') \\
&= AB'C+AB'C'+ABC'+AB'C'
\end{aligned}
$$

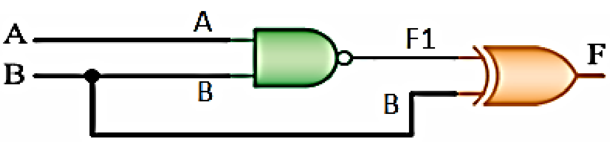
A	B	C	L1	L2
0	0	0	1	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

****

۱۱- در شکل زیر تابع خروجی را بنویسید و جدول صحت آن را به دست آورید.



جواب :



$$
\begin{aligned}
F1 &= AB + A'B' + B' = AB + A'B' + B(A+A') \\
&= AB + A'B' + AB + A'B \\
&= AB + A'B' + A'B
\end{aligned}
$$

$$
\begin{aligned}
F1 &= (AB)' = A' + B' \\
F &= F1 \oplus B = F1'.B + F1.B' \\
&= (A'+B')'.B + (A'+B')B' \\
&= (A.B).B + A'B' + B' \\
&= AB + A'B' + B'
\end{aligned}
$$

A	B	F
0	0	1
0	1	1
1	0	0
1	1	1

****

۱۲- مداری طراحی کنید که ورودی‌های از صفر تا ۲۰ را بگیرد ، و اعداد بخش پذیر بر ۳ را مشخص کند.

جواب :

	E	D	C	B	A	F
0	0	0	0	0	0	0
1	0	0	0	0	1	0
2	0	0	0	1	0	0
3	0	0	0	1	1	1
4	0	0	1	0	0	0
5	0	0	1	0	1	0
6	0	0	1	1	0	1
7	0	0	1	1	1	0
8	0	1	0	0	0	0
9	0	1	0	0	1	1
10	0	1	0	1	0	0
11	0	1	0	1	1	0
12	0	1	1	0	0	1
13	0	1	1	0	1	0
14	0	1	1	1	0	0
15	0	1	1	1	1	1
16	1	0	0	0	0	0
17	1	0	0	0	1	0
18	1	0	0	1	0	1
19	1	0	0	1	1	0
20	1	0	1	0	0	0
21	1	0	1	0	1	X
22	1	0	1	1	0	X
23	1	0	1	1	1	X
24	1	1	0	0	0	X
25	1	1	0	0	1	X
26	1	1	0	1	0	X
27	1	1	0	1	1	X
28	1	1	1	0	0	X
29	1	1	1	0	1	X
30	1	1	1	1	0	X
31	1	1	1	1	1	X

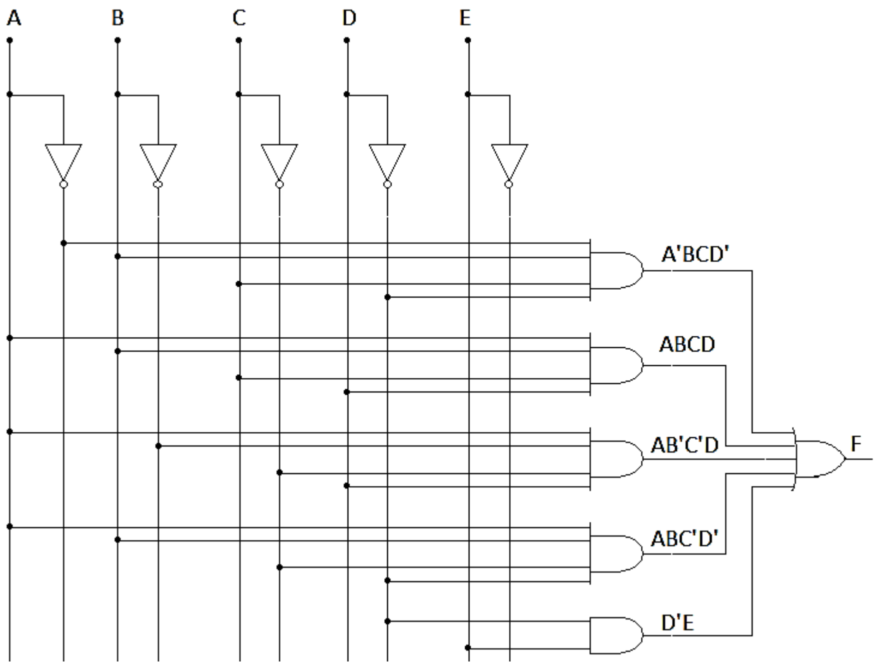
E = 0

AB \ CD	00	01	11	10
00	0	0	1	0
01	0	0	0	1
11	1	0	1	0
10	0	1	0	0

E = 1

AB \ CD	00	01	11	10
00	0	0	0	1
01	0	X	X	X
11	X	X	X	X
10	X	X	X	X

F = D'E + A'BCD' + ABCD + AB'C'D + ABC'D'



****

۱۳- توابع زیر را به کمک نقشه کارنو ساده کنید.

$$F1(A,B,C) = \Sigma m(0,1,5,6,7)$$

$$F2(A,B,C,D) = \Sigma m(0,2,5,7,9,11,12,14,15)$$

$$F(A,B,C,D) = \Sigma m(2,5,6,7,9,10,11,15)$$

جواب :

A \ BC	00	01	11	10
0	1	1	0	0
1	0	1	1	1

F1 = A'B' + B'C + AB

AB \ CD	00	01	11	10
00	1	0	0	1
01	0	1	1	0
11	1	0	1	1
10	0	1	1	0

F2 = A'B'D' + A'BD + ABC + AB'D

AB \ CD	00	01	11	10
00	0	0	0	1
01	0	1	1	1
11	0	0	1	0
10	0	1	1	1

F = A'BC' + A'BD + BCD + AB'D + AB'C

****

۱۴- توابع زیر را با کمک نقشه کارنو ساده کنید.

$F1(A,B,C) = \prod M\,(0,2,4,6)$ $\qquad$ $F2(A,B,C,D) = \prod M\,(0,1,4,7,9,11,13,14)$

جواب :

$$F1 = \sum m(1,3,5,7)$$

$$F2 = \sum m(2,3,5,6,8,10,12,15)$$

A \ BC	00	01	11	10
0	0	1	1	0
1	0	1	0	1

F1 = B'C + A'C + ABC'

AB \ CD	00	01	11	10
00	0	0	1	1
01	0	1	0	1
11	1	0	1	0
10	1	0	0	1

F2 = AC'D' + A'B'C + B'CD' + A'CD'

****

۱۵- توابع زیر را ابتدا ساده کنید و سپس به کمک دروازه های منطقی رسم کنید.

$F1(A,B,C) = \Sigma m(0,1,2,3,5,7)$ $\qquad$ $F2(A,B,C,D) = \Sigma m(2,4,6,8,10,12,13,14)$

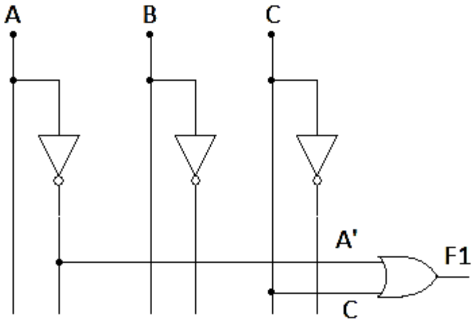
جواب :

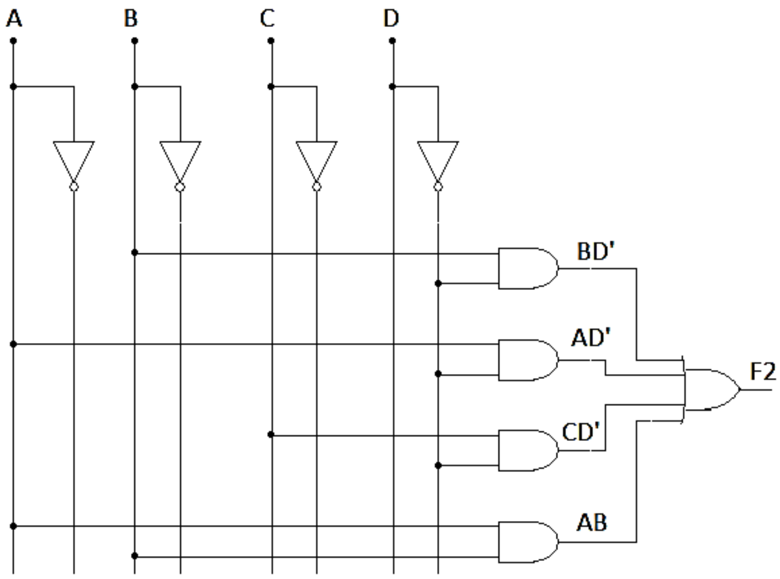
A \ BC	00	01	11	10
0	1	1	1	1
1	0	1	1	0

F1 = A' + C

AB \ CD	00	01	11	10
00	0	0	0	1
01	1	0	0	1
11	1	1	0	1
10	1	0	0	1

F2 = BD' + AD' + CD' + AB





****

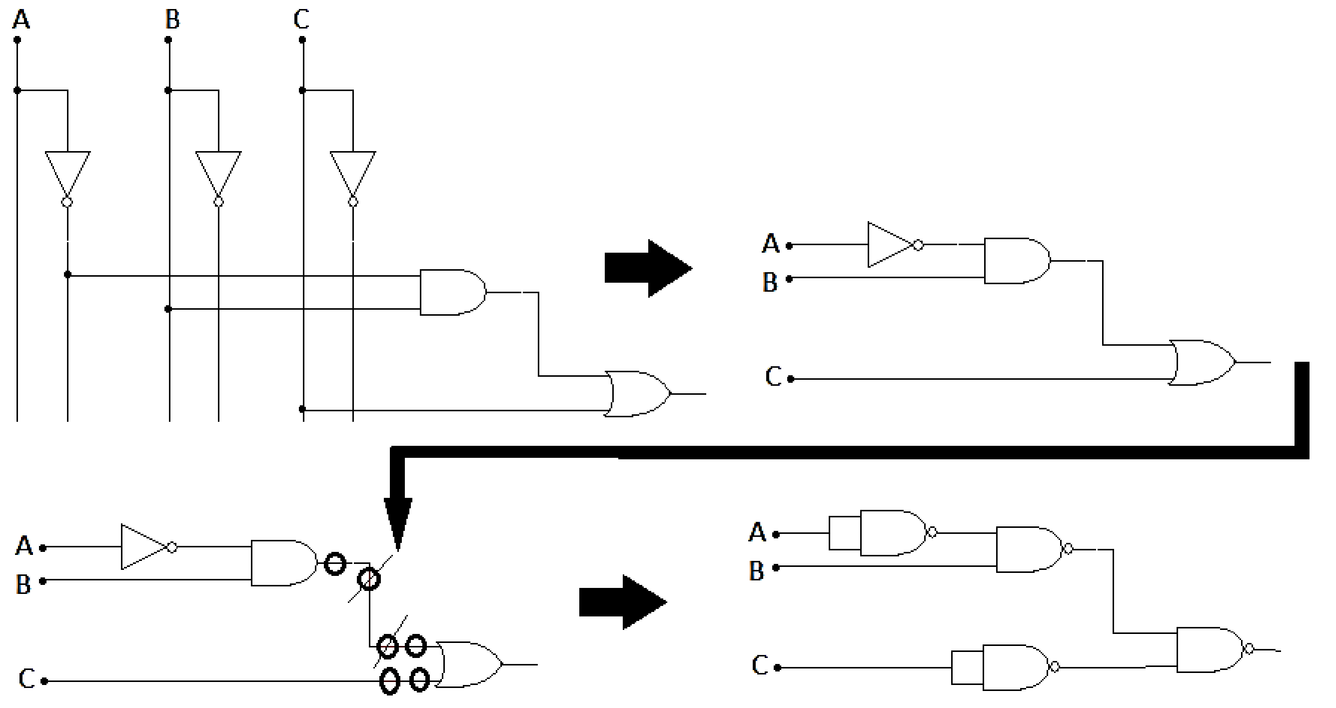
۱۶- تابع زیر را بعد از ساده نمودن به کمک دروازه منطقی NAND رسم کنید.

$F(A,B,C) = \Sigma m(1,2,3,5,7)$

جواب :

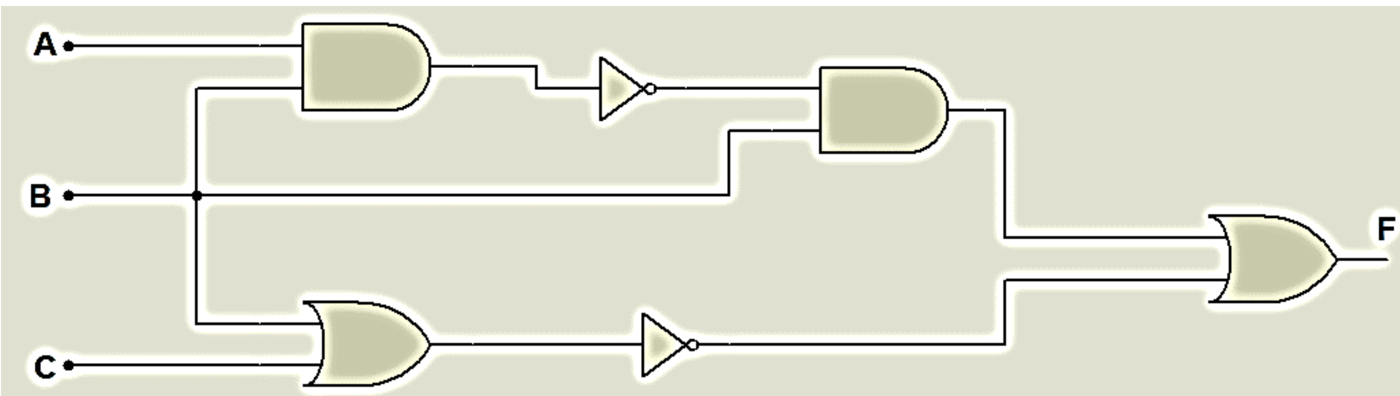
A \ BC	00	01	11	10
0	0	1	1	1
1	0	1	1	0

F = C + A'B



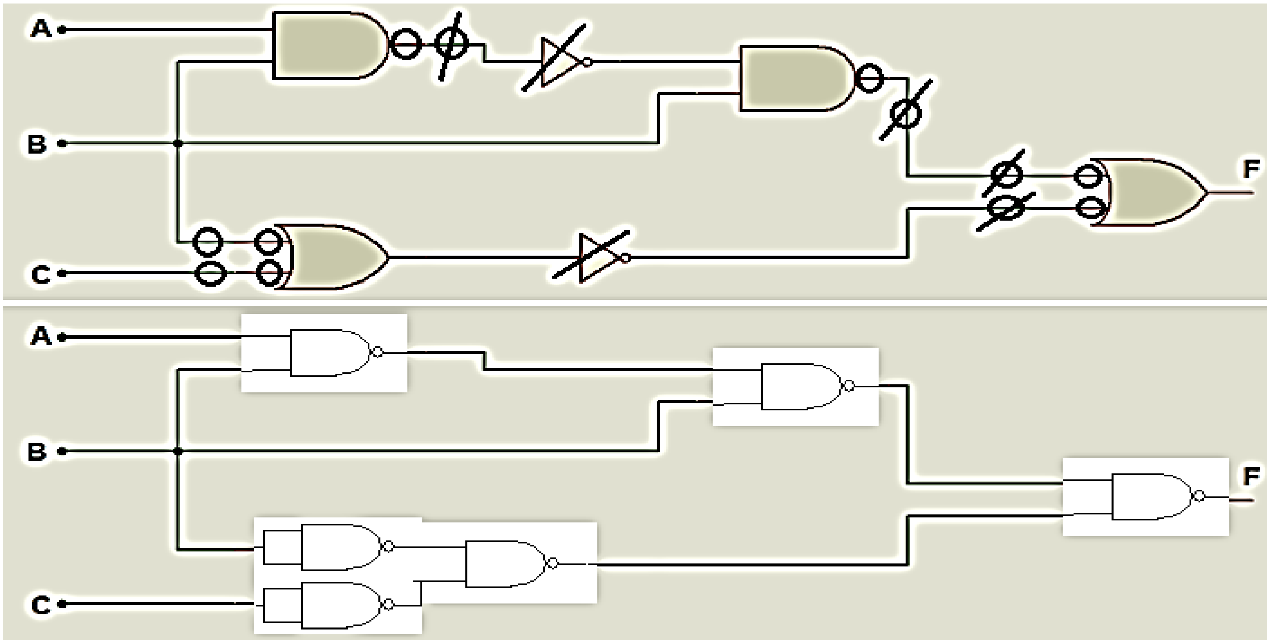
۱۷- در مدار شکل زیر :

الف) آن‌را فقط با استفاده از گیت NAND بازسازی کنید.   ب) آن‌را فقط با استفاده از گیت NOR بازسازی کنید.

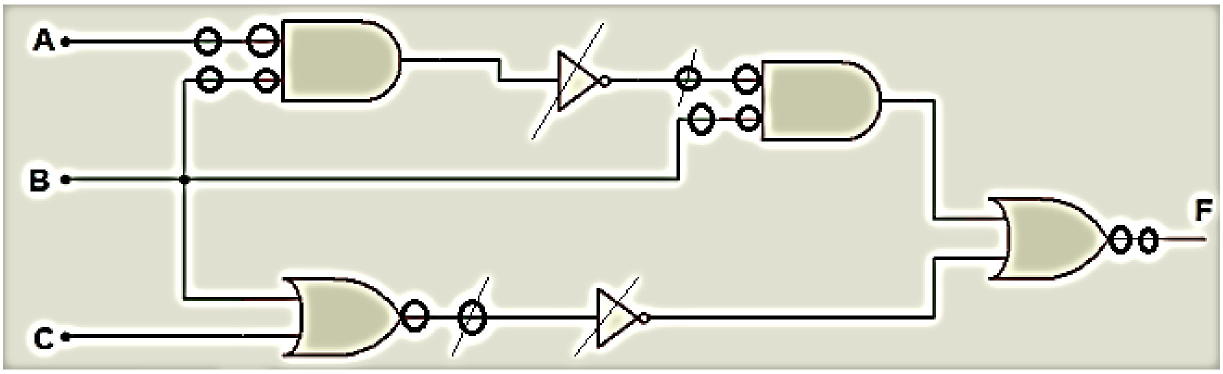


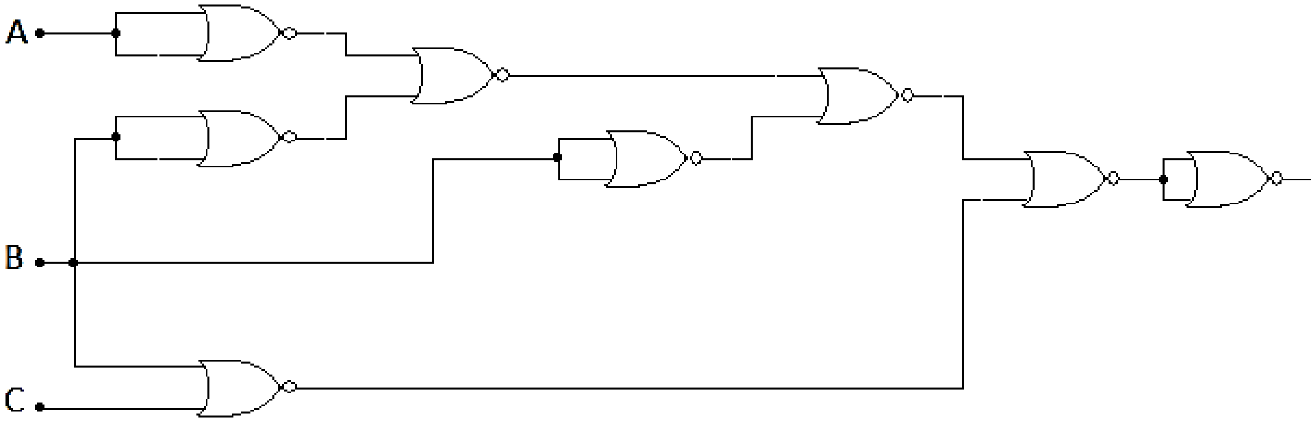
جواب :

الف :



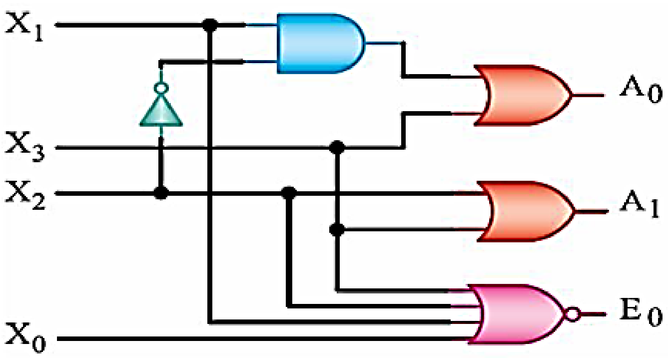
ب



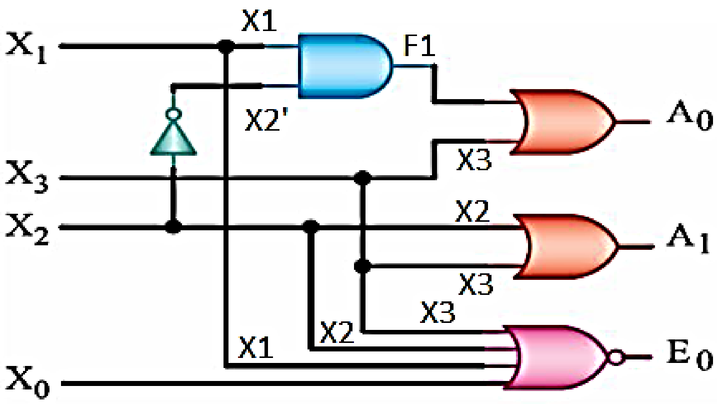


****

۱۸- در مدار شکل زیر ، معادله جبری توابع E0 , A1 , A0 را بیابید.



جواب :



F1 = X1.X2'
A0 = F1 + X3 = X1.X2' + X3
A1 = X2 + X3
E0 = (X0 + X1 + X2 + X3)'
= X0'.X1'.X2'.X3'

****

۱۹-یک مقایسه کننده دو بیتی طراحی کنید که در آن دو عدد A و B را از ورودی بگیرد، و حاصل مقایسه آنها در سه خروجی A=B، A<B و A>B نشان دهد.

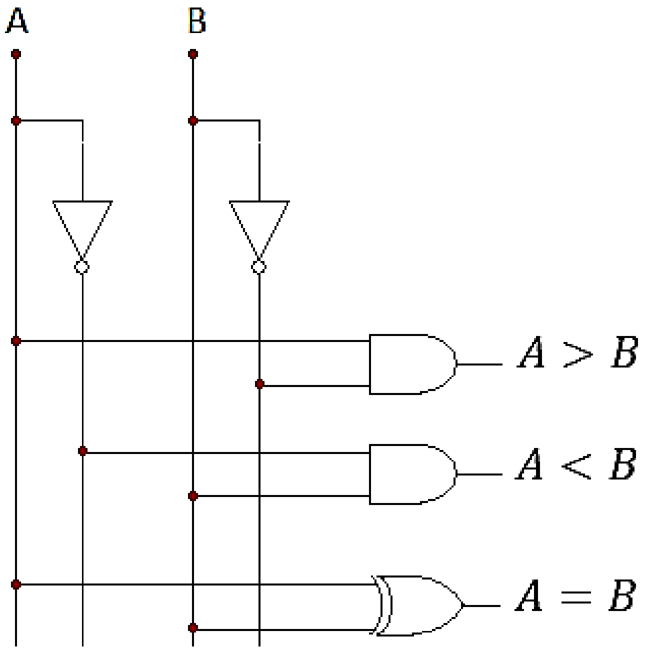
جواب :

A	B	A>B	A<B	A=B
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

$$A > B : F = AB'$$

$$A < B : F = A'B$$

$$A = B : F = A'B' + AB = A \oplus B$$

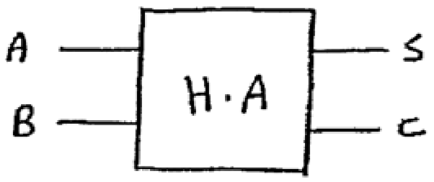


فصل پنجم : مدارهای ترکیبی با کاربردهای ویژه

در این بخش، بعضی از مدارهای ترکیبی با کاربردهای ویژه ، که به علت مصرف عام به صورت تراشه‌ها (IC) تجارتی عرضه می‌شوند را معرفی می‌کنیم.

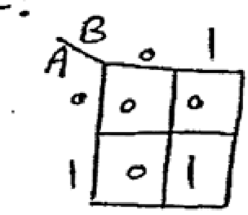
۱- نیم جمع کننده یا جمع کننده ناقص یا H.A (Half Adder) :

جمع دو عدد یک بیتی را می‌توان با مداری به نام نیم جمع کننده یا به اختصار H.A انجام دهیم. مدار H.A مداری است که دو ورودی (B ,A) و دو خروجی (S و C) دارد. S رقم اول حاصل جمع (SUM) و C رقم نقلی (Carry) را مشخص می‌کند.



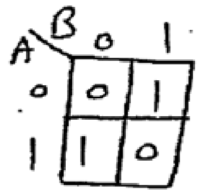
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

C:

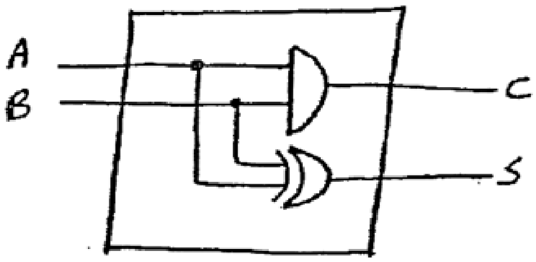


$C = AB$

S:



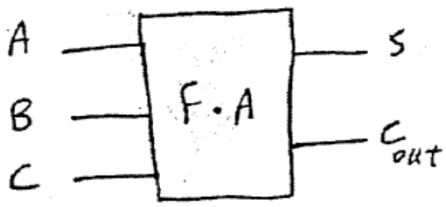
$S = A'B + AB' = A \oplus B$



۲- تمام جمع کننده یا جمع کننده کامل یا 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 تمام جمع کننده می‌گویند و با F.A نشان می دهند.

تمام جمع کننده مداری است که ۳ خط ورودی (C_1 ,B,A) و دو خط خروجی (S و C) دارد.



A	B	C	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

C_{out}:

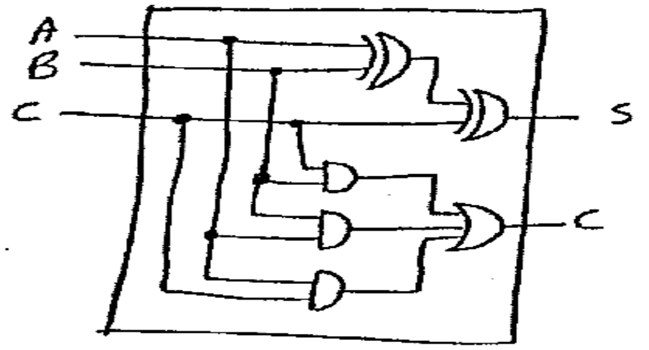
A \ BC	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$C_{out} = AB + BC + AC$$

S:

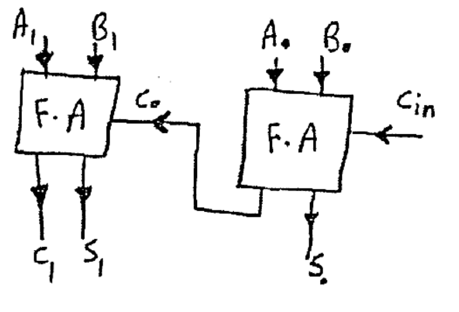
A \ BC	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$S = A'B'C + A'BC' + AB'C' + ABC = C \oplus (A \oplus B)$$

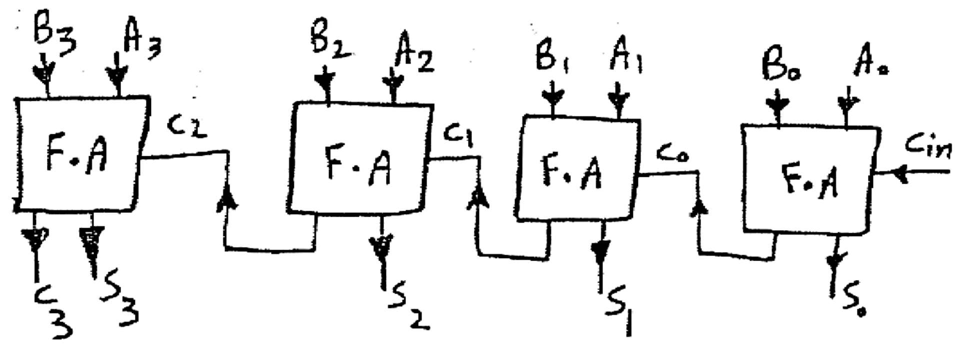


جمع کننده موازی چهار بیتی :

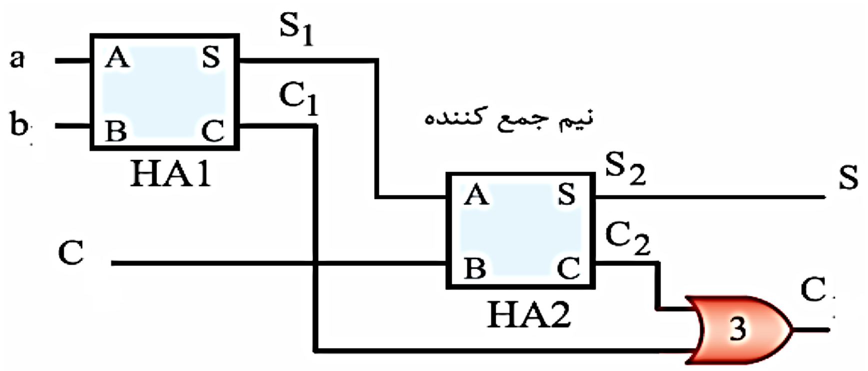
فرض کنید بخواهیم دو عدد A=47 و B=36 را با هم جمع کنیم. برای این کار ابتدا یکان‌ها را با هم جمع می‌کنیم. (۱۳=۶+۷) سپس دهگان‌ها و رقم نقلی با هم جمع می‌کنیم (۸= ۱+۴+۳) و جواب را به دست می‌آوریم. (S= 083). حال فرض کنید بخواهیم دو عدد دو بیتی $A=A_1A_0$ و $B=B_1B_0$ را درمبنای دو با هم جمع کنیم. برای این کار باید مانند مثال بالا ابتدا A0 را با B0 و سپس A_1 وB_1 و C_0 را با هم جمع کنیم. جواب نهایی شامل ($S=C_1S_1S_0$) می باشد. اگر بخواهیم عملیات فوق را با F.A پیاده سازی نماییم داریم:



و از آنجا که هر F.A سه ورودی دارد میبایست Cin= 0 باشد تا به جمع آسیب نزند. به همین ترتیب برای جمع دو عدد چهار بیتی دودویی $A= A_3A_2A_1A_0$ و $B=B_3B_2B_1B_0$ خواهیم داشت :



مثال : با استفاده از بلوک‌های H.A و گیت‌های اضافه یک F.A بسازید.



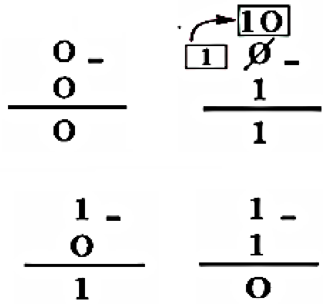
۳- نیم تفرق کننده یا تفریق کننده ناقص یا H.S (Half Subtractor) :

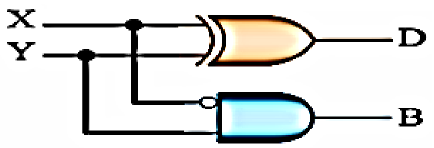
مدار تفریق کننده ناقص یا نیم تفریق کننده شامل دو ورودی (y,x) و دو خروجی (B,D) است. خروجی D (decrement) اصل تفریق و خروجی B رقم قرضی (Borrow) می‌باشد. در حین تفریق اگر x ≥ y باشد یعنی (۰-۰ ، ۰-۱ ، ۱-۱) عمل تفریق انجام می‌شود و به رقم قرضی نیاز نداریم. ولی اگر x≤y باشد یعنی (۱ - ۰) بایستی یک واحد از مرتبه بالاتر قرض بگیریم. یک واحد قرض گرفته شده از مرتبه بالاتر، ۲ واحد به بیت مورد نظر اضافه می‌کند . (سیستم دودویی)

ورودی‌ها X	ورودی‌ها Y	خروجی‌ها B	خروجی‌ها D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

$$D = \overline{x}y + x\overline{y} = x \oplus y$$

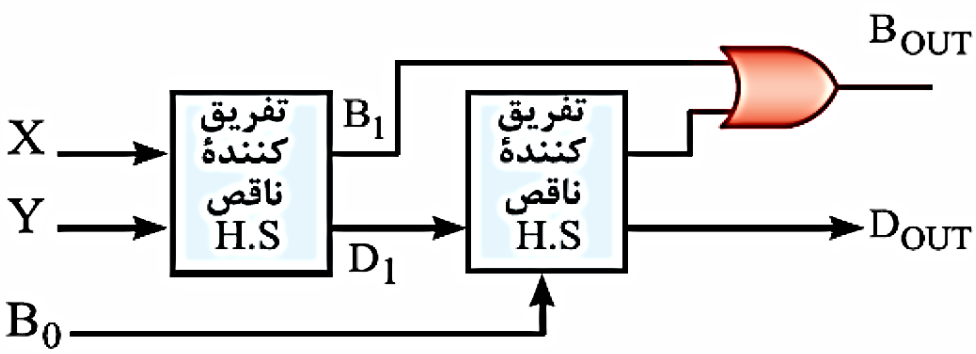
$$B = \overline{x}y$$





۴- تمام تفریق کننده یا تفریق کننده کامل یا F.S (Full Subtractor) :

تفریق کننده کامل یا تمام تفریق کننده مداری است که ۳ خط ورودی (C1 , B , A) و دو خط خروجی (B , D) دارد. می توان مطابق روشی که برای مدارهای تمام جمع کننده ذکر شد ، مدار تمام تفریق کننده یا تفریق کننده کامل را نیز با دو نیم تفریق کننده و یک گیت OR ساخت.



تفریق کننده موازی ۴ بیتی :

تفریق دو عدد دودویی به روش مکمل دو :

فرض کنید بخواهیم تفریق دو عدد دودویی دو بیتی $A=A_1A_0$ و $B=B_1B_0$ را به روش مکمل دو محاسبه نماییم. قبلاً گفتیم برای یافتن مکمل ۲ یک عدد دو دویی تمام بیت های آن را معکوس کرده و حاصل را با عدد ۱ جمع میکنیم. مثلاً مکمل دو عدد ۱۰۱۰۱ در مبنای دو برابر ۰۱۰۱۱ است. روش دیگر یافتن مکمل دو یک عدد دودویی ، آن است که کم‌ارزشترین رقم ۱ و تمام صفرهای سمت راست آن بدون تغییر ، و بقیه ارقام سمت چپ آن مکمل می‌شوند. مثلاً مکمل دو عدد ۱۰۱۰ در مبنای دو ، برابر ۰۱۱۰ است. بنابر آن‌چه گفته شد ، مکمل دو عدد B_1B_0 برابر $(B_1'B_0'+1)$ است. حال آن را با عدد A جمع می‌کنیم.

حال اگر جواب حاصل بیت نقلی نداشت، کافی است بیت نقلی را حذف کنیم و باقیمانده را به عنوان جواب بپذیریم. اما اگر جواب حاصله بیت نقلی نداشت قطعا B از A بزرگتر بوده و پاسخ یک عدد منفی است. برای یافتن پاسخ صحیح در این حالت کافی است بیت نقلی را حذف کرده و مکمل دو باقیمانده را محاسبه نمود. در انتها نیز به آن یک علامت منفی بیافزاییم.

مثال : تفریق مستقیم دو عدد باینری :

$$\begin{array}{r} 11011 \\ -\,01110 \\ \hline 01101 \end{array} \qquad \begin{array}{r} 27 \\ -\,14 \\ \hline +\,13 \end{array}$$

حاصل تفریق

تفریق به روش مکمل۲ :

$$۰۱۱۱۰ \xrightarrow{\text{متمم ۲}} ۱۰۰۱۰$$

$$\begin{array}{r} 11011 \\ -\,01110 \\ \hline 11011 \\ +\,10010 \\ \hline 1\;01101 \end{array} \qquad \begin{array}{r} 27 \\ -\,14 \\ \hline +\,13 \end{array}$$

حاصل تفریق

رقم نقلی نهایی +

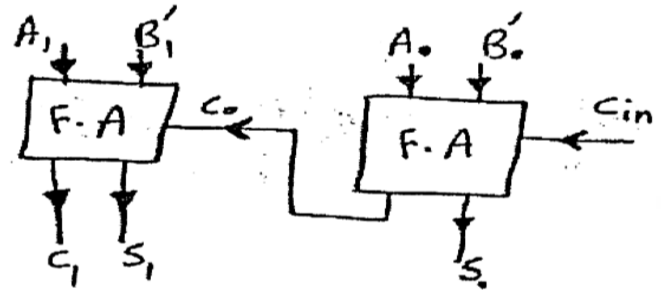
تفریق به روش مکمل ۲ :

$$\begin{array}{r} 01110 \\ -\;\;11011 \\ \hline 110011 \end{array} \qquad \begin{array}{r} 14 \\ -\,27 \\ \hline -\,13 \end{array}$$

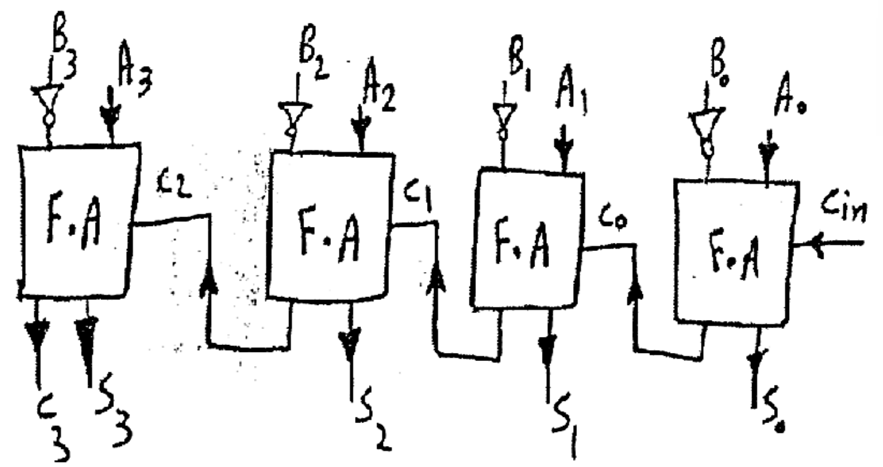
$$110011 \xrightarrow{\text{متمم ۲}} -\,01101$$

رقم نقلی نهایی -

اگر بخواهیم عملیات فوق را با F.A پیاده‌سازی نماییم داریم :



همانطور که قبلاً نیز ذکر شد ، برای جمع کردن عدد A با مکمل دو عدد B می‌بایست تمام بیت‌های B مکمل گردند و حاصل با عدد ۱ نیز جمع شود. قرار دادن Cin=1 این امر را سبب می‌شود. به همین ترتیب برای تفریق دو عدد چهاربیتی دودویی $A=A_3A_2A_1A_0$ و $B=B_3B_2B_1B_0$ خواهیم داشت :

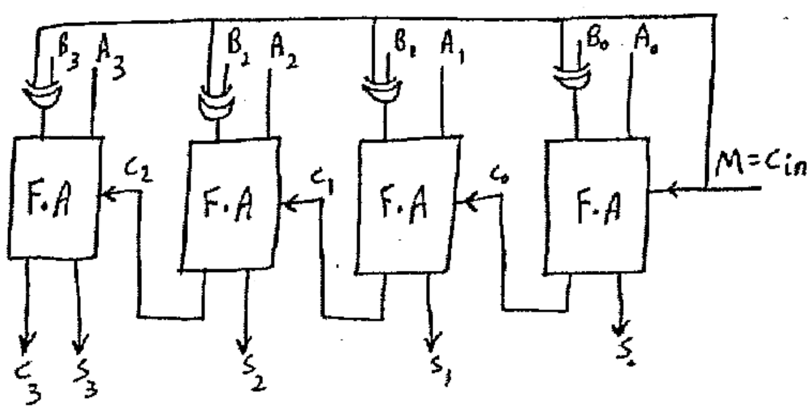


جمع کننده / تفریق کننده موازی چهاربیتی :

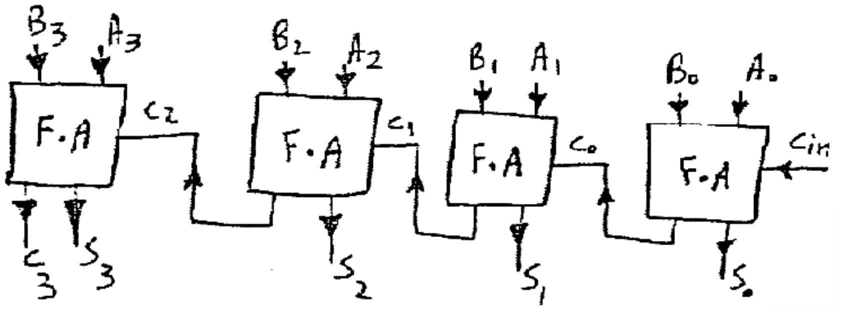
نکته۱ : می‌دانیم : $0 \oplus 1 = 1 \; , \; 0 \oplus 0 = 0$ است . بنابراین میتوان گفت : $0 \oplus A = A$. به عبارتی XOR هر عدد با صفر ، خود آن عدد می‌شود.

نکته۲: می‌دانیم : $1 \oplus 1 = 0 \; , \; 1 \oplus 0 = 1$ است . بنابراین میتوان گفت : $1 \oplus A = A'$. به عبارتی XOR هر عدد با یک ، مکمل آن عدد می‌شود.

حال به شکل زیر دقت کنید.

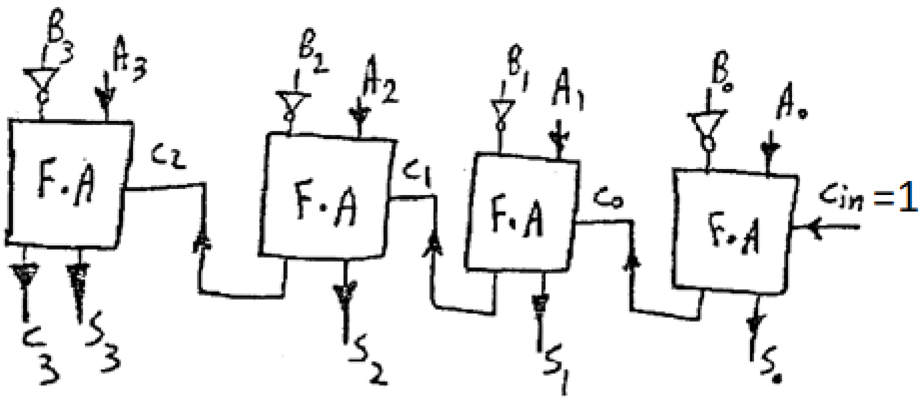


فرض کنید M=0 باشد. در این صورت با توجه به نکته 1 می‌توان مدار بالا به صورت زیر ساده نمود.



چنان چه می بینید این مدار همان مدار جمع کننده موازی چهار بیتی است.

این بار فرض کنید M=1 باشد. در این صورت با توجه به نکته 2 می‌توان مدار فوق را به صورت زیر ساده نمود.



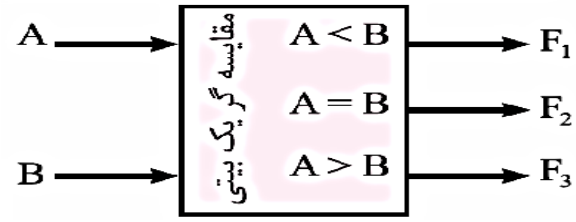
چنان‌چه می‌بینید این مدار به یک تفریق کننده موازی چهار بیتی تبدیل شده است.

بنابراین می‌توان گفت ، با قرار دادن M=0 ، مدار به یک جمع کننده و با قرار دادن M=1 مدار به یک تفریق کننده موازی چهار بیتی تبدیل می شود.

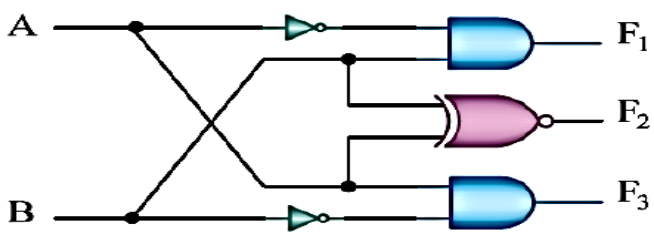
۵-مقایسه کننده یک بیتی :

در مقایسه بین دو بیت ، ممکن است بزرگتر ، کوچکتر یا مساوی بودن بیت ها مورد نظر باشد.

A	B	F_1 $A<B$	F_2 $A=B$	F_3 $A>B$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0



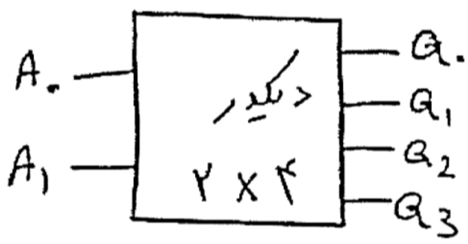
$$F_1 = \bar{A}B$$
$$F_2 = \bar{A}\bar{B} + AB$$
$$F_3 = A\bar{B}$$



۶- مدارهای رمز گشا یا دیکدر (Decoder) :

هررمز گشا با n ورودی داری حداکثر 2^n خروجی است و در هر لحظه تنها یکی از 2^n خروجی فعال است. به عبارت دیگر، هر یک از خروجی‌های آن متناظر با یک ترکیب خاص ورودی (یک جمله حاصل ضرب نرمال یا مین ترم) است.

در شکل زیر یک مدار دیکدر ۲ به ۴ (بافرض Active High) و جدول صحت آن را می بینید.

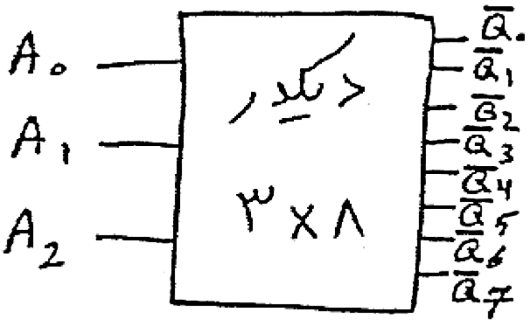


A_1	A_0	Q_3	Q_2	Q_1	Q0
0	0	**0**	**0**	**0**	**1**
0	1	**0**	**0**	**1**	**0**
1	0	**0**	**1**	**0**	**0**
1	1	**1**	**0**	**0**	**0**

همان‌طورکه دیده می‌شود درخروجی دیکدر ، همواره یک خط فعال است .

(در حالت Active High خط فعال با ۱ نمایش داده می شود.)

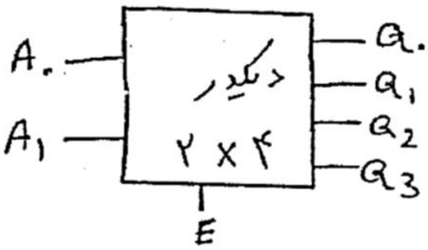
واین خط فعال توسط خط آدرس A_1A_0 تعیین می‌شود. شکل زیر یک دیکدر ۳ به ۸ (با فرض Active Low) و جدول صحت آن را نشان می‌دهد.



A_2	A_1	A_0	Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0
0	0	0	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	0	1
0	1	0	1	1	1	1	1	0	1	1
0	1	1	1	1	1	1	0	1	1	1
1	0	0	1	1	1	0	1	1	1	1
1	0	1	1	1	0	1	1	1	1	1
1	1	0	1	0	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1

در بسیاری از آی‌سی‌ها یک پایه کنترلی وجود دارد که به کمک آن می‌توان کل مدار را غیر فعال (خاموش) کرد. این پایه را فعال‌ساز گویند و معمولاً با E یا CS نشان می‌دهند.

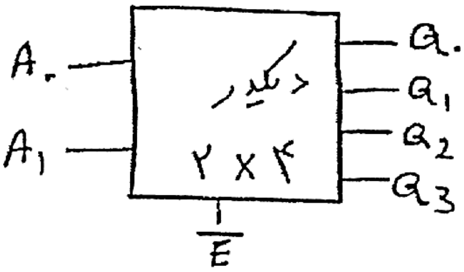
در شکل زیر یک مدار دیکدر ۲ به ۴ (بافرض Active High) با فعال‌ساز مثبت ، همراه با جدول صحت آن را می بینید.



E	A_1	A_0	Q_3	Q_2	Q_1	Q_0
0	X	X	0	0	0	1
1	0	0	0	0	0	0
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

همانطورکه دیده می شود ، اگر E=0 باشد ، فرقی نمی‌کند که A_0 و A_1 صفر باشند یا یک . در هر صورت تمام خروجی‌ها صفر هستند. اما اگر E=1 باشد آی‌سی‌فعال بوده و به کار خود ادامه می‌دهد.

در شکل زیر یک مدار دیکدر ۲ به ۴ (با فرض Active High) با فعال سازی منفی، همراه با جدول صحت آن را می‌بینید.



E	A_1	A_0	Q_3	Q_2	Q_1	Q_0
1	X	X	0	0	0	1
0	0	0	0	0	0	0
0	0	1	0	0	1	0
0	1	0	0	1	0	0
0	1	1	1	0	0	0

طراحی مدار به کمک دیکدر :

برای این کار مراحل زیر را به ترتیب پی می‌گیریم :

۱- جدول صحت مدار خواسته شده را می‌نویسیم.

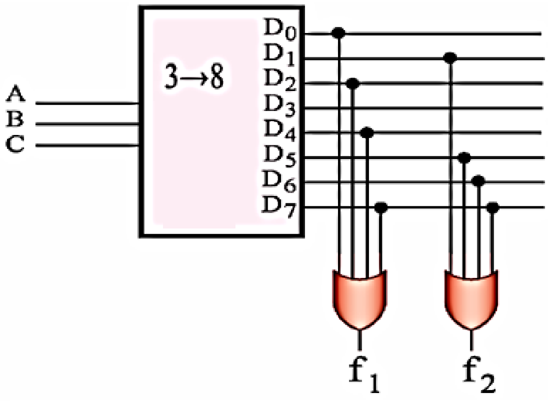
۲- از روی تعداد متغیرهای ورودی، دیکدر مربوطه را رسم می‌کنیم.

۳- در دیکدرهای active high خروجی‌های 1 را OR و در دیکدرهای active low خروجی های ۱ را NAND می‌کنیم.

مثال: با توجه به جدول زیر توابع منطقی F_1 و F_2 را طراحی کنید.

	C	B	A	f_1	f_2
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

چون : $F2 = \Sigma m(1,5,6,7)$ و $F1 = \Sigma m(0,2,4,7)$ است ، پس :



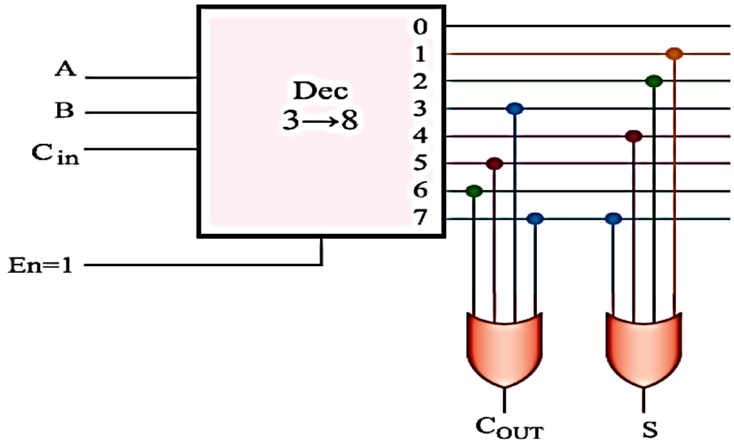
مثال : یک مدار تمام جمع کننده با استفاده از رمزگشا طراحی کنید.

حل : ابتدا جدول صحت S و C مربوط به تمام جمع کننده را به دست می‌آوریم.

	A	B	C_{in}	C_{OUT}	S
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1

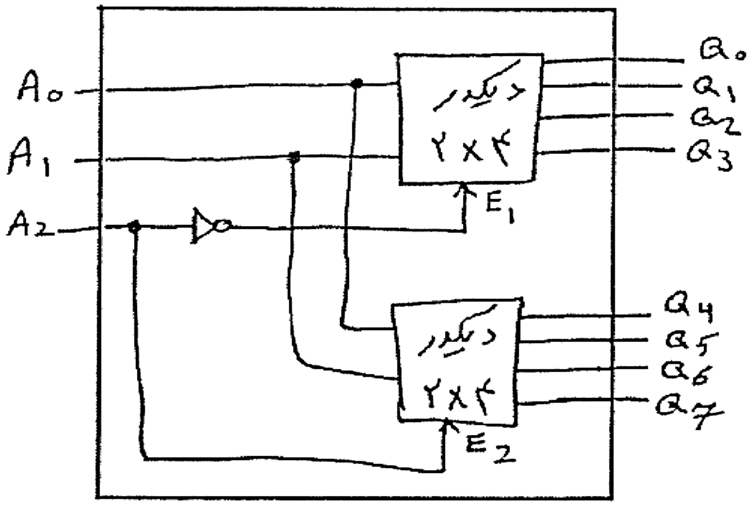
$$S = \Sigma m\ (1,2,4,7)$$

$$Cout = \Sigma m(3,5,6,7)$$



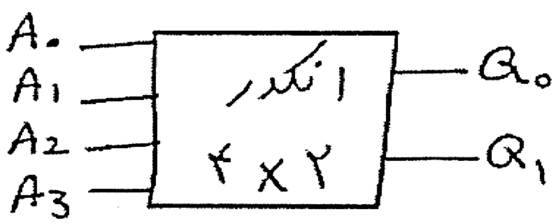
ساخت دیکدرهای بزرگتر:

به کمک پایه E می‌توان آی‌سی‌های دیکدر را به گونه‌ای به هم وصل کرد که یک دیکدر بزرگتر به دست آید. در شکل زیر توسط دو دیکدر ۲ به ۴ (با فرض Active High و فعال سازی مثبت) یک دیکدر ۳ به ۸ درست شده است.



۷-مدار رمز کننده یا انکدر (Encoder) :

در شکل زیر یک مدار انکدر ۴ به ۲ (با فرض Active High) و جدول صحت آن را می‌بینید.

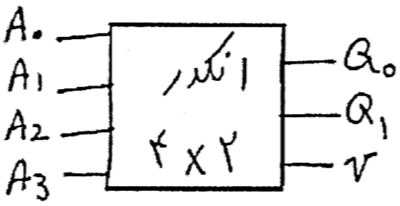


A_3	A_2	A_1	A_0	Q_1	Q_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

همان‌طور که دیده می‌شود، در خروجی انکدر، معادل باینری خط فعال شده ورودی، نمایش داده می‌شود. ایراد اصلی مدار فوق آن است که طبق تعریف می‌بایست در هر حالت، فقط یکی از ورودی‌های A0تا A3 فعال باشد. بنابراین می‌توان گفت اگر بیش از یک ورودی فعال باشد، مدار دچار اشکال می‌گردد. برای رفع مشکل فوق از انکدر با اولویت (Priority Encoder) استفاده می‌کنند. در انکدر اولویت‌دار، موقعی که بیش از یک ورودی فعال می‌شود ، عدد با اولویت بالاتر به خروجی ارسال می گردد.

مشکل دیگر زمانی است که هیچ‌کدام از خطوط ورودی فعال نباشند. در این صورت همه خطوط برابر 0 خواهند بود ؛ که مشابه وقتی است که ورودی 0001 باشد. برای حل این مشکل نیز اغلب از یک خروجی اضافی (V) استفاده می‌کنند.

اگر V=1 باشد، یعنی حداقل یکی از ورودی‌ها فعال است. اما اگر V=0 باشد، یعنی هیچ‌کدام از ورودی‌ها فعال نیستند. در شکل زیر یک مدار انکدر اولویت دار ۴ به ۲ (با فرض Active High) با خروجی V و جدول صحت آن را می‌بینید.

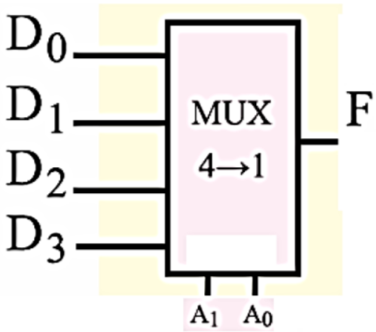


A_3	A_2	A_1	A_0	Q_1	Q_0	V
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

۸-مدارهای متمرکز کننده یا تسهیم کننده یا مالتی‌پلکسر (Multiplexer) :

مالتی پلکسر یکی از پرکاربردترین مدارهای ترکیبی است. و مانند یک انتخاب کننده (Selector) عمل می‌کند. این مدار با توجه به آدرسی که در خط آدرس آن انتخاب می شود، به یکی از ورودی‌ها اجازه عبور می دهد. به عبارت دیگر، عملکرد آن درست شبیه یک کلید چند حالته است؛ با این تفاوت که حالت کلید به صورت دیجیتالی انتخاب می شود.

شکل زیر یک مولتی پلکسر ۴ به ۱ و جدول صحت آن را صحت نشان می‌دهد.



A_1	A_0	F
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

همان‌طور که می‌بینید ، ورودی که خط آدرس نشان می‌دهد ، به خروجی منتقل می‌شود.

طراحی مدار با استفاده از مالتی پلکسر :

برای این کار :

۱- جدول صحت مدار خواسته شده را می‌نویسیم.
۲- آن‌را به تعداد خطوط ورودی مالتی پلکسر موجود تقسیم می‌کنیم.
۳- خطوط ورودی مالتی پلکسر را می‌یابیم.
۴- مدار را ترسیم می‌کنیم.

مثال: اگر تابع F به شکل زیر باشد، مطلوب است :

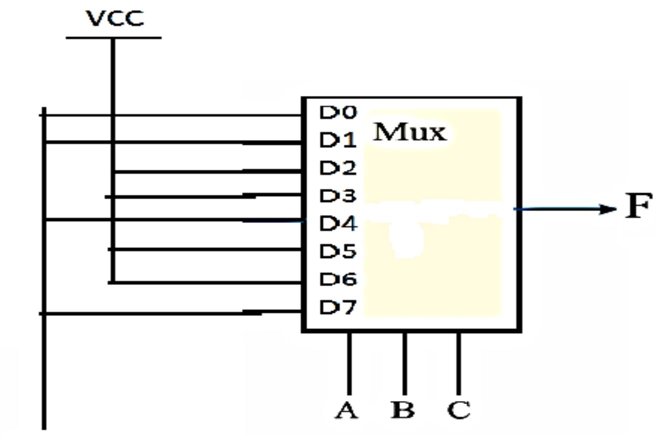
$F(A,B,C) = \Sigma m(2,3,5,6)$

الف) طراحی آن با استفاده از یک MUX8*1

ب) طراح آن با استفاده از یک MUX4*1

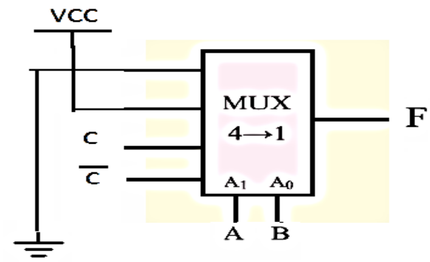
الف :

A	B	C	F	
0	0	0	0	D0 = 0
0	0	1	0	D1 = 0
0	1	0	1	D2 = 1
0	1	1	1	D3 = 1
1	0	0	0	D4 = 0
1	0	1	1	D5 = 1
1	1	0	1	D6 = 1
1	1	1	0	D7 = 0



ب :

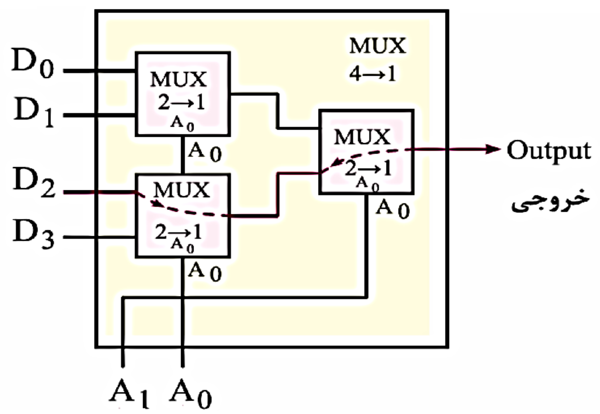
A	B	C	F	
0	0	0	0	D0 = 0
0	0	1	0	
0	1	0	1	D1 = 1
0	1	1	1	
1	0	0	0	D2 = C
1	0	1	1	
1	1	0	1	D3 = $\overline{C}$
1	1	1	0	



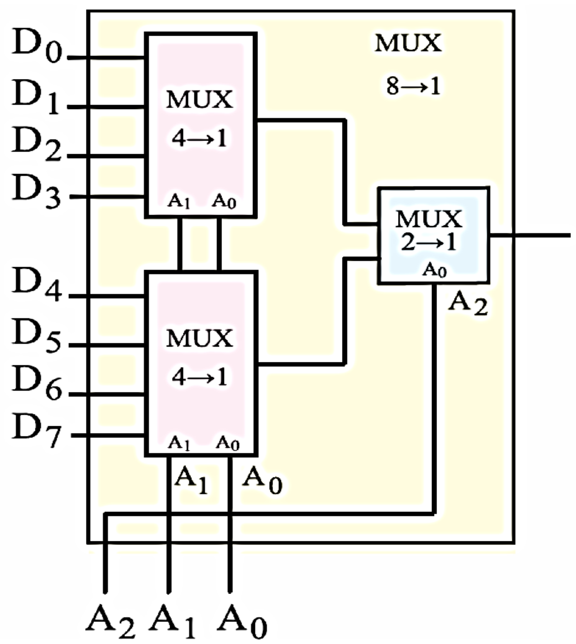
ساخت مالتی پلکسرهای بزرگتر :

با روشی مشابه ساخت دیکدرهای بزرگتر، می‌توان مالتی پلکسرهای بزرگتر را نیز ساخت.

مثال: یک مالتی پلکسر ۴ به ۱ را با سه عدد مالتی پلکسر ۲ به ۱ طراحی کنید.

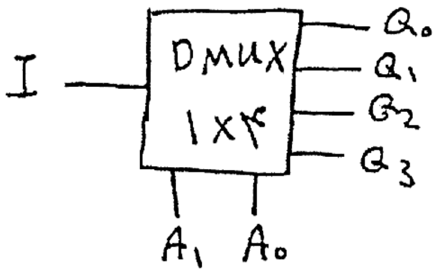


مثال: یک مالتی پلکسر ۸ به ۱ را با دو عدد مالتی پلکسر ۴ به ۱ و یک مالتی پلکسر ۲ به ۱ طراحی کنید.



۹- مدار توزیع کننده یا دی‌مالتی‌پلکسر (Demultiplexer) :

شکل زیر یک دی مولتی پلکسر ۱ به ۴ را با جدول صحت آن نشان می‌دهد.



A_1	A_0	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0	I
0	1	0	0	۱	0
1	0	0	۱	0	0
1	1	۱	0	0	0

همان‌طور که می‌بینید، خط I به خروجی‌ی منتقل می‌شود که خط آدرس آن‌را نشان می‌دهد.

نکته : اگر I=1 باشد، مدار دی‌مالتی‌پلکسر به یک دیکدر Active High تبدیل می‌شود.

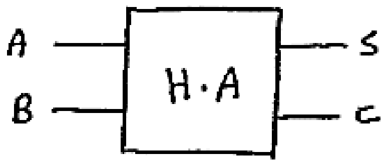
تمرین:

۱- نشان دهید چگونه می‌توان سه تابع زیر را به کمک مدار نیم جمع کننده اجرا کرد.

$F1 = X \oplus Y \oplus Z$ $\quad$ $F2 = ABC$ $\quad$ $F3 = XY' + XZ' + X'YZ$

جواب :

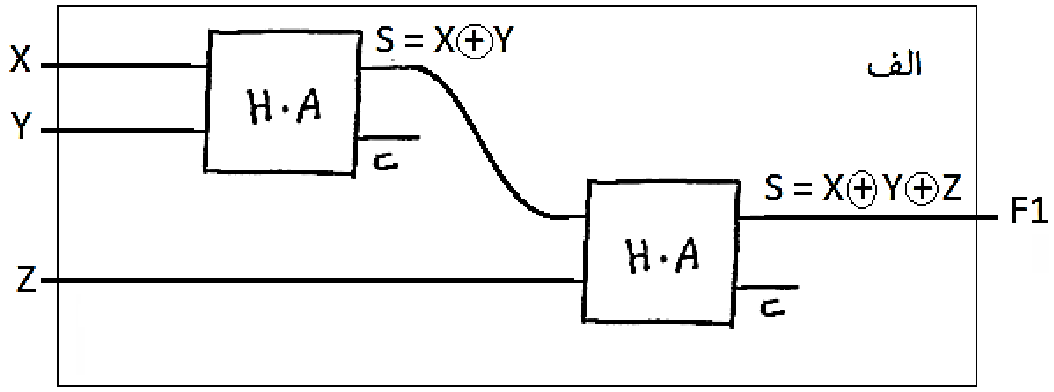
می‌دانیم :



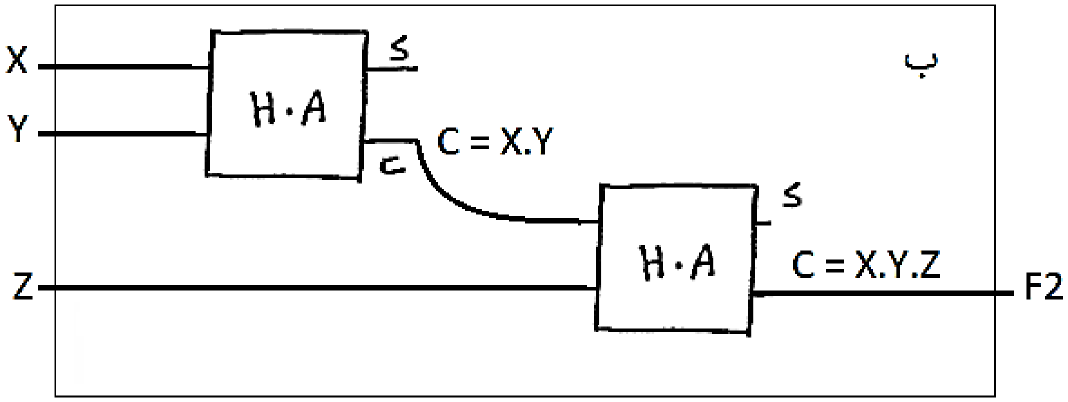
$s = A'B + AB' = A \oplus B$

$c = AB$

الف :

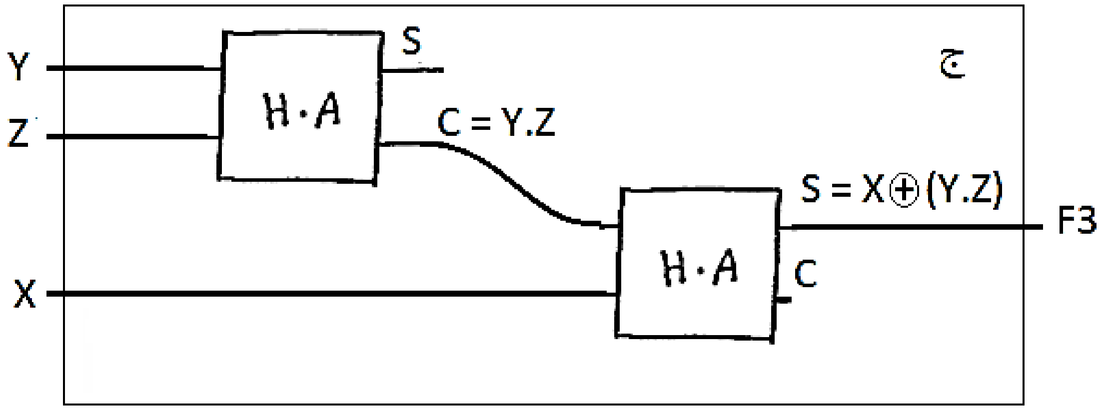


ب :



ج :

$$F3 = XY' + XZ' + X'YZ = X(Y' + Z') + X'YZ = X(YZ)' + X'YZ = X \oplus (YZ)$$



۲- تابع منطقی رأی اکثریت یک کمیته سه نفره را با یک دیکدر اجرا کنید.

الف : با یک دیکدر active high

ب : با یک دیکدر active low

جواب :

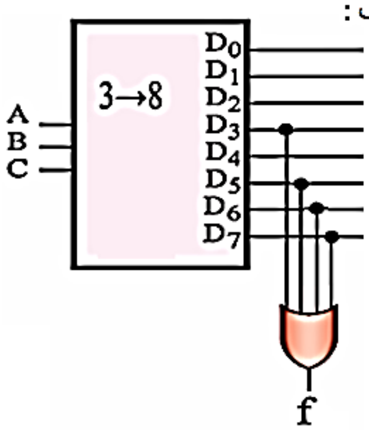
ابتدا جدول صحت را می‌کشیم .

چون ۳ ورودی دارد ، پس از یک دیکدر ۸ به ۳ استفاده می‌کنیم .

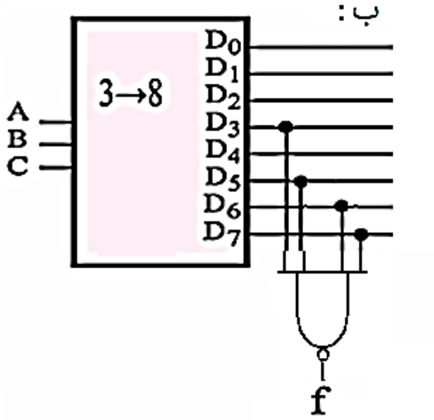
	A	B	C	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

حال در دیکدرهای active high خروجی‌های 1 را OR و در دیکدرهای active low خروجی های ۱ را NAND می‌کنیم. داریم :

الف :



ب :



****

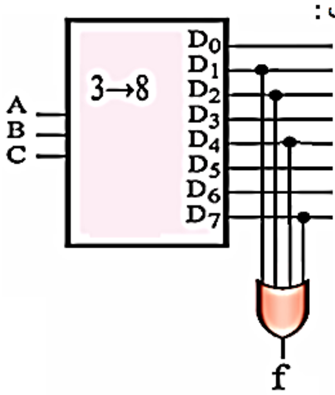
۳- تابع $F(A,B,C) = \Sigma\, \mathrm{m}\ (1,2,4,7)$ را با یک دیکدر یا رمزگشا اجرا کنید.

الف)active high

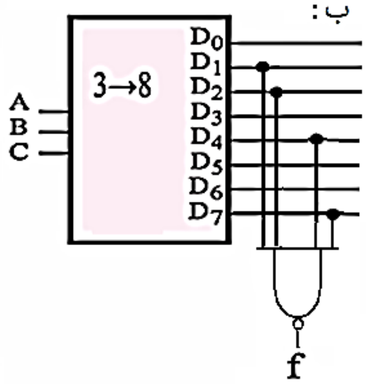
ب)active low

	A	B	C	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

جواب : الف :



ب :

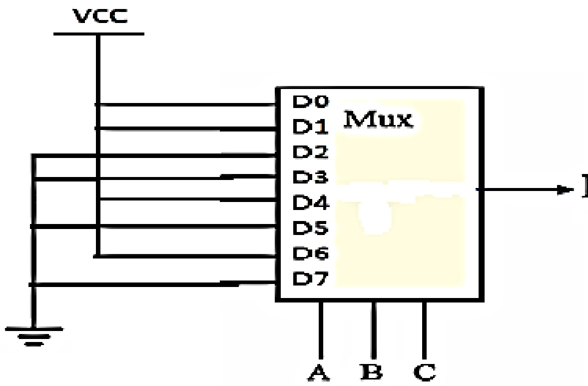


۴- تابع $F(A,B,C) = \Sigma m(0,1,4,6)$ را با یک مالتی پلکسر اجرا کنید.

الف : با یک $MUX\ 8 \to 1$ ب : با یک $MUX\ 4 \to 1$

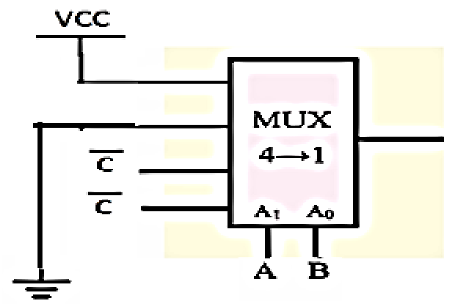
الف :

	A	B	C	F	
0	0	0	0	1	D0 = 1
1	0	0	1	1	D1 = 1
2	0	1	0	0	D2 = 0
3	0	1	1	0	D3 = 0
4	1	0	0	1	D4 = 1
5	1	0	1	0	D5 = 0
6	1	1	0	1	D6 = 1
7	1	1	1	0	D7 = 0



ب :

A	B	C	F	
0	0	0	1	D0 = 1
0	0	1	1	
0	1	0	0	D1 = 0
0	1	1	0	
1	0	0	1	D2 = $\overline{C}$
1	0	1	0	
1	1	0	1	D3 = $\overline{C}$
1	1	1	0	



****

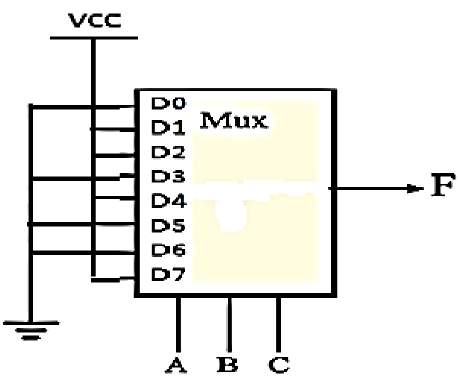
۵- تابع $F = \Sigma m\ (1,2,4,7)$ را با استفاده از رمز گشا طراحی کنید.

الف : با یک $MUX\ 8 \to 1$ ب : با یک $MUX\ 4 \to 1$

جواب :

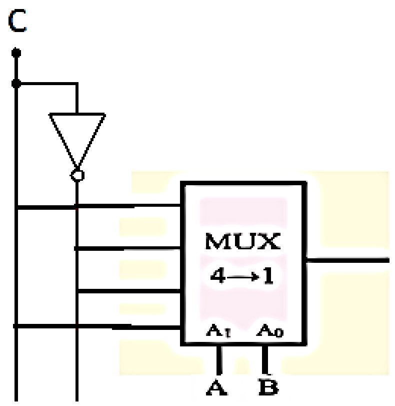
الف :

	A	B	C	F	
0	0	0	0	0	D0 = 0
1	0	0	1	1	D1 = 1
2	0	1	0	1	D2 = 1
3	0	1	1	0	D3 = 0
4	1	0	0	1	D4 = 1
5	1	0	1	0	D5 = 0
6	1	1	0	0	D6 = 0
7	1	1	1	1	D7 = 1



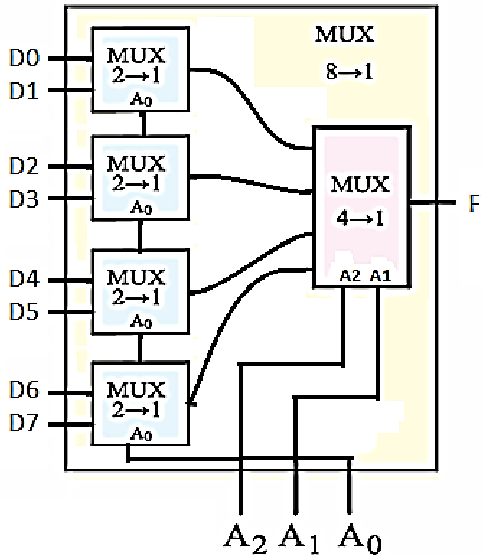
ب :

A	B	C	F	
0	0	0	0	D0 = C
0	0	1	1	
0	1	0	1	D1 = $\overline{C}$
0	1	1	0	
1	0	0	1	D2 = $\overline{C}$
1	0	1	0	
1	1	0	0	D3 = C
1	1	1	1	



۶- با استفاده از ۴عدد mux×2*1 و ۱ عدد mux×4*1 یک عدد mux×8*1 طراحی کنید.

جواب :



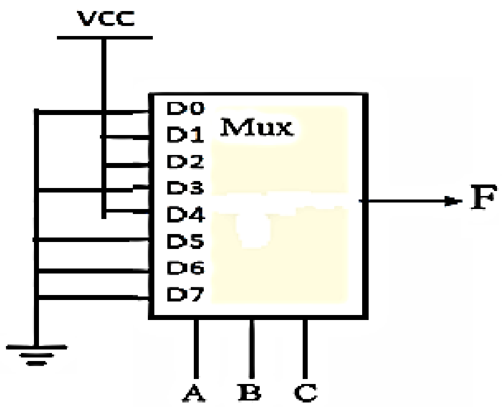
****

۷- با استفاده از مالتی پلکسر، سیستمی با سه سنسور ورودی را طوری طراحی کنید که اگر <u>فقط</u> یک سنسور فعال شود دزدگیر به صدا درآید.   الف : با یک $MUX\ 8 \rightarrow 1$ ب : با یک $MUX\ 4 \rightarrow 1$

جواب :

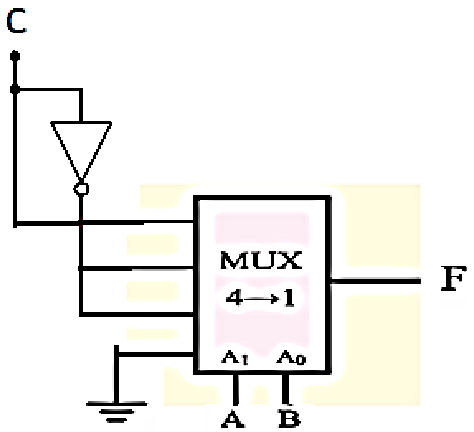
الف :

	A	B	C	F	
0	0	0	0	0	D0 = 0
1	0	0	1	1	D1 = 1
2	0	1	0	1	D2 = 1
3	0	1	1	0	D3 = 0
4	1	0	0	1	D4 = 1
5	1	0	1	0	D5 = 0
6	1	1	0	0	D6 = 0
7	1	1	1	0	D7 = 0



ب :

A	B	C	F	
0	0	0	0	D0 = C
0	0	1	1	
0	1	0	1	D1 = $\overline{C}$
0	1	1	0	
1	0	0	1	D2 = $\overline{C}$
1	0	1	0	
1	1	0	0	D3 = 0
1	1	1	0	



****

۸- تابع $F(A,B,C) = AB'C + A'BC + ABC' + ABC$ را طراحی کنید.

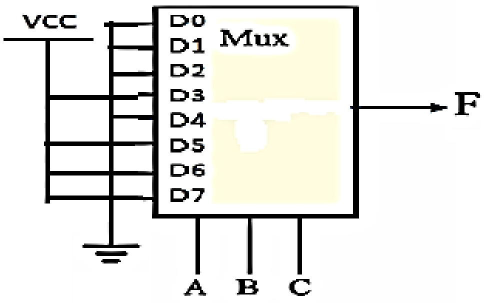
الف : با یک $MUX\ 8 \rightarrow 1$ ب : با یک $MUX\ 4 \rightarrow 1$

ج : با یک دیکدر active high د : با یک دیکدر active low

جواب : الف :

F = AB'C + A'BC + ABC' + ABC

	A	B	C	F	
0	0	0	0	0	D0 = 0
1	0	0	1	0	D1 = 0
2	0	1	0	0	D2 = 0
3	0	1	1	1	D3 = 1
4	1	0	0	0	D4 = 0
5	1	0	1	1	D5 = 1
6	1	1	0	1	D6 = 1
7	1	1	1	1	D7 = 1



ب :

F = AB'C + A'BC + ABC' + ABC

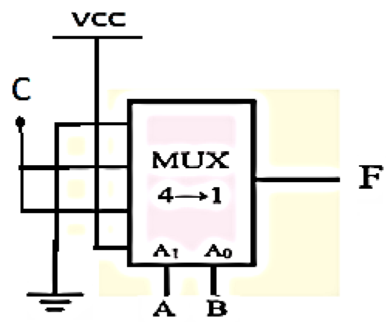
	A	B	C	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

D0 = 0

D1 = C

D2 = C

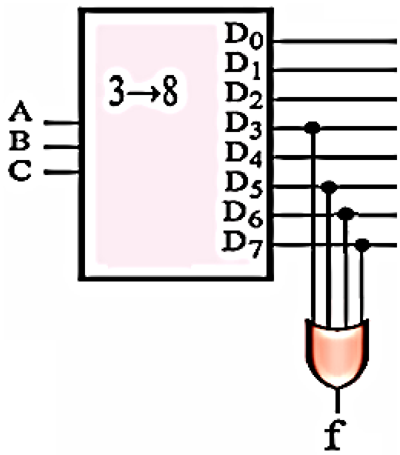
D3 = 1



ج :

F = AB'C + A'BC + ABC' + ABC

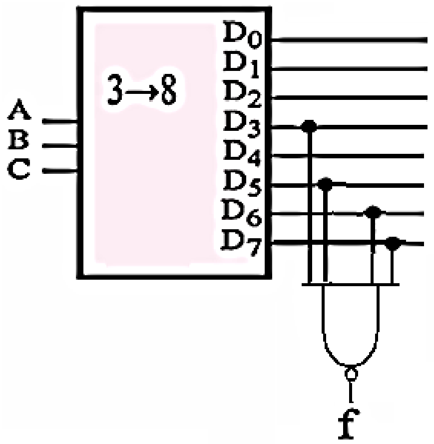
	A	B	C	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1



د :

F = AB'C + A'BC + ABC' + ABC

	A	B	C	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

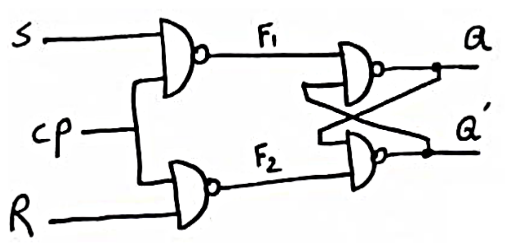


فصل ششم : فیلیپ‌فلاپ‌ها

فلیپ فلاپ‌ها عناصری حافظه‌دار هستند که خروجی آن‌ها در هر لحظه ، می‌تواند به ورودی در همان لحظه و خروجی حالت قبل بستگی داشته باشد.

فیلیپ فیلاپ SR :

به مدار زیر دقت کنید.



در این مدار ابتدا فرض کنید CP = 0 باشد . بنابراین حتماً F1=F2=1 است.

(می‌دانیم خروجی گیت NAND یک است، مگر در حالتی‌که دو ورودی یک باشند . و چون یکی از دو ورودی صفر است پس خروجی حتماً یک است.) بنابراین داریم:

$$\mathrm{Q} = (\mathrm{F1.Q'})' = (1.\mathrm{Q'})' = (\mathrm{Q'})' = \mathrm{Q}$$

$$Q' = (F1.Q)' = (1.Q)' = (Q') = Q'$$

به عبارتی ، مادامیکه CP = 0 باشد خروجی‌ها بدون تغییر باقی می‌مانند. حال فرض کنید CP = 1 باشد. می‌خواهیم با بررسی حالت‌های مختلف ، جدول زیر را تکمیل کنیم.

حالت اول: $\mathrm{S} = 1\,,\mathrm{R} = 0 \quad \rightarrow \quad \mathrm{F1} = 0\,,\mathrm{F2} = 1 \quad \rightarrow \quad \mathrm{Q} = 1\,,\mathrm{Q} = 0$

حالت دوم: $S = 0\,,R = 0 \quad \rightarrow \quad F1 = 1\,,F2 = 1 \quad \rightarrow \quad Q = 1\,,Q = 0$

حالت سوم: $S = 1\,,R = 0 \quad \rightarrow \quad F1 = 1\,,F2 = 0 \quad \rightarrow \quad Q = 0\,,Q = 1$

حالت چهارم: $S = 0\,,R = 0 \quad \rightarrow \quad F1 = 1\,,F2 = 1 \quad \rightarrow \quad Q = 0\,,Q = 1$

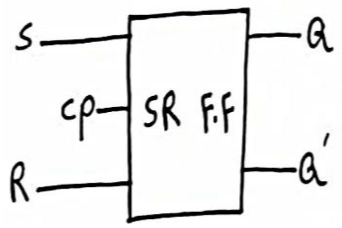
حالت پنجم: $S = 1\,,R = 1 \quad \rightarrow \quad F1 = 0\,,F2 = 0 \quad \rightarrow \quad Q = 1\,,Q = 1$

S	R	Q	Q’
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	غ ق ق	غ ق ق

با بررسی جدول و نتایج به دست آمده می‌توان گفت :

- اگر $R \neq S$ باشد ، خروجی تابع S است یعنی $Q = S$
- اگر $S = R= 0$ باشد، خروجی همان خروجی حالت قبل است و بدون تغیر باقی می ماند یعنی $Q(t+1) = Q(t)$
- اگر $S = R= 1$ باشد خروجی غیر قابل قبول است.

نتایج بالا را می‌توان به شکل زیر نشان داد.

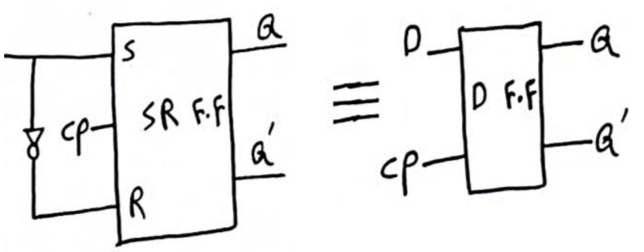


جدول صحت SR F.F :

S	R	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	غ ق ق

فلیپ فلاپ D :

همان‌طور که گفتیم ، در حالتی‌که $S = R= 1$ باشد ، خروجی SR F.F غیر قابل قبول است. برای رفع این عیب روش‌های متعددی وجود دارد. یک راه این است که ، کاری کنیم تا هیچ وقت $S = R$ نشود. برای این منظور کافی است پایه‌ی S را با یک گیت NOT به پایه‌ی R متصل نماییم. با انجام این کار فیلیپ فلاپ جدیدی به نام D F.F به وجود می آید.



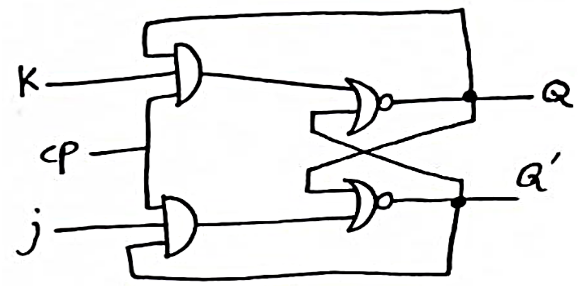
جدول صحت D F.F :

D	Q(t+1)
0	0
1	1

واضح است که حالت D=0 در فیلیپ فلاپ D ، معادل حالت S=0 و R=1 در فیلیپ فلاپ SR است . پس خروجی تابع S و برابر صفر است. همچینین حالت D=1 در فیلیپ فلاپ D معادل حالت S=1 وR=0 در فیلیپ فلاپ SR است پس خروجی تابع S و برابر یک است.

فلیپ فلاپ JK:

به مدار زیر دقت کنید.



با بررسی حالت‌های مختلف ، جدول زیر را تکمیل می کنیم.

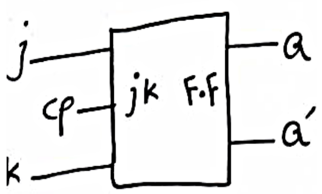
J	K	Q	Q’
1	0	1	0
0	0	1	0
1	1	0	1
0	1	0	1
0	0	0	1
1	1	1	0

با بررسی جدول و نتایج به دست آمده می‌توان گفت :

- اگر $K \neq J$ باشد ، خروجی تابع J است یعنی Q = J
- اگر J = K= 0 باشد، خروجی همان خروجی حالت قبل است و بدون تغیر باقی می ماند یعنی Q(t+1) = Q(t)
- اگر J = K =1 باشد، خروجی مکمل خروجی حالت قبل است یعنی Q(t+1)=Q’(t)

نتایج بالا را می‌توان به شکل زیر نشان داد.

جدول صحت JK F.F :

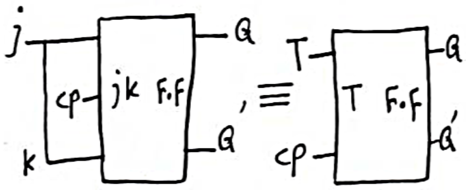


J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	Q/(t)

فلیپ فلاپ T:

با اتصال ورودی‌های J وk به یک‌دیگر ، فیلیپ فلاپ جدیدی به نام T F.F به وجود می آید.

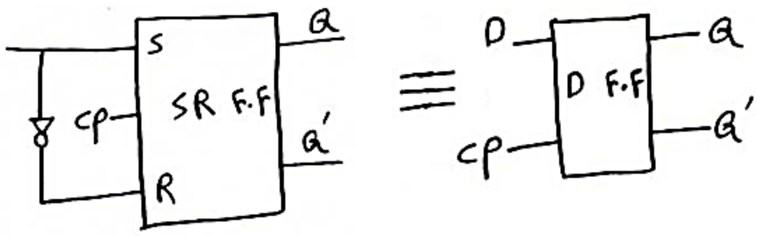
جدول صحت T F.F :



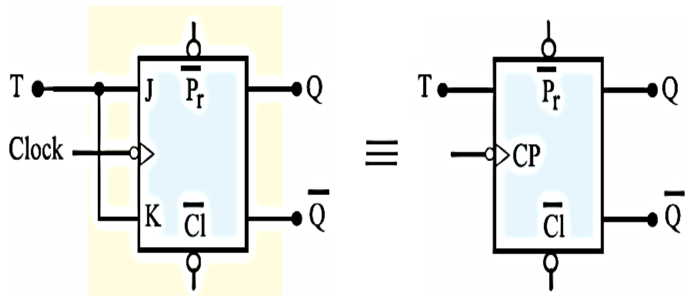
T	Q(t+1)
0	Q(t)
1	Q/(t)

واضح است که حالت T=0 در فیلیپ فلاپ T ، معادل حالت J = K = 0 در فیلیپ فلاپ JK است . پس خروجی تابع J و برابر صفر است. همچنین حالت T=1 در فیلیپ فلاپ T ، معادل J = K = 1 در فیلیپ فلاپ JK است . پس خروجی تابع J و مکمل حالت قبل است.

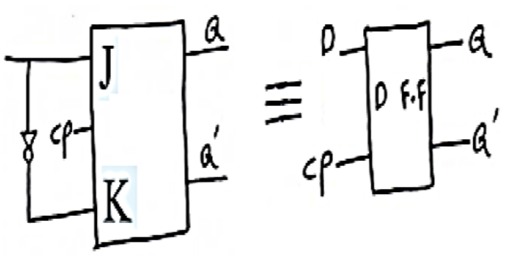
ساخت D F.F با SR F.F :



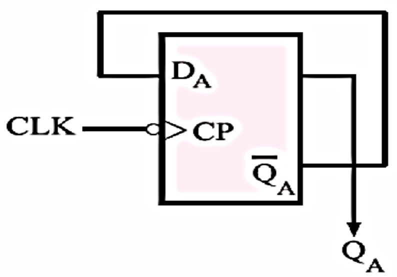
ساخت T F.F با JK F.F :



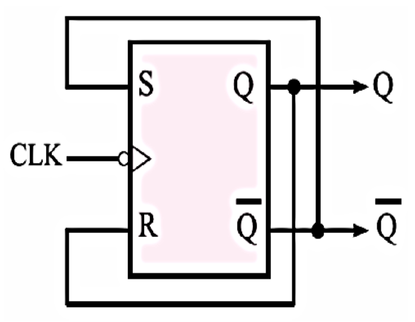
ساخت D F.F با JK F.F:



ساخت T F.F با D F.F:



ساخت T F.F با SR F.F:

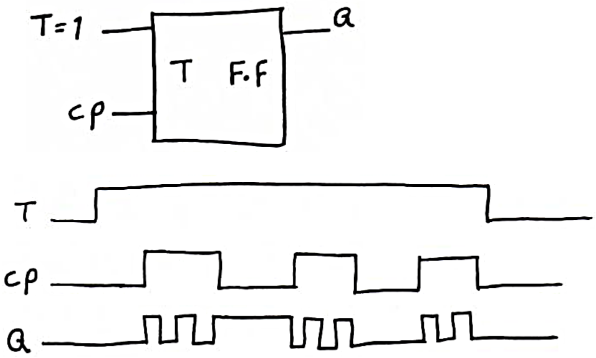


آی‌سی‌های حساس به سطح و حساس به لبه :

آی‌سی‌حساس به سطح مثبت :

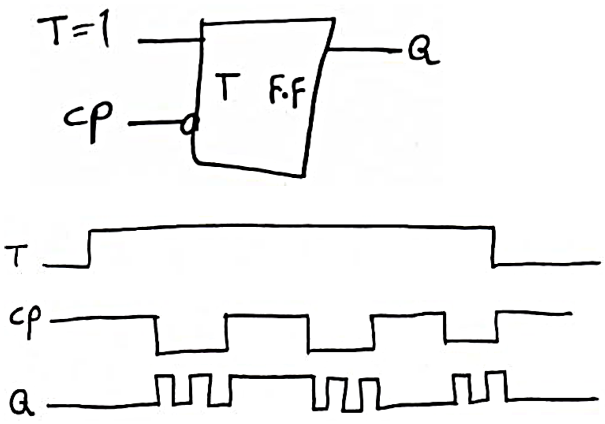
در این نوع آی‌سی مادامی‌که CP = 1 باشد ، خروجی می‌تواند با توجه به ورودی تغییر کند.

مدار زیر را در نظر بگیرید. از آنجا که T=1 است و خروجی در هر لحظه مرتباً مکمل حالت قبل می‌شود. به عبارتی مادامی‌که CP =1 باشد خروجی مرتباً تغیر می‌کند. یعنی:



آی‌سی‌حساس به سطح منفی:

در این آی‌سی مادامی‌که CP =1 باشد ، خروجی می‌تواند با توجه به ورودی تغییر کند.



آی سی حساس به لبه مثبت:

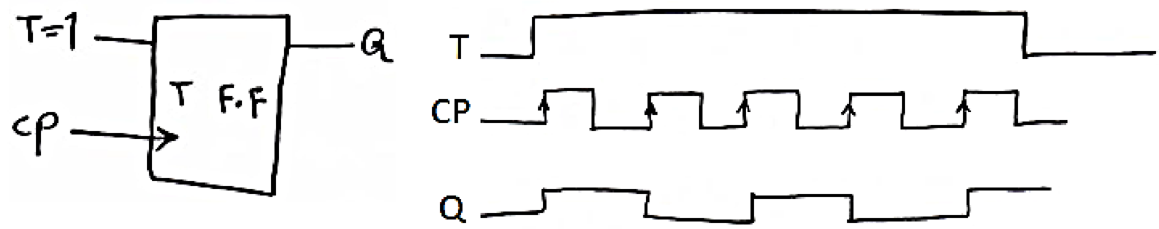
قبلاً گفتیم در آی‌سی حساس به سطح ، مادامی‌که CP =1 باشد خروجی مرتباً تغییر می‌کند. به مثال اول توجه کنید. همان‌طور که می‌بینید مادامی‌که CP =1 باشد ، خروجی مرتباً تغییر می‌کند . اما در حالتی‌که CP =0 باشد خروجی بدون تغییر باقی می‌ماند. حال سؤال اینجاست که در حالت بدون تغییر ، خروجی صفر است یا یک ؟ یا اینکه اصلاً مادامی‌که CP =1 باشد خروجی چند بار تغییر می کند؟ پاسخ به این سؤالات به چند عامل بستگی دارد.

۱- مدت زمانی که CP =1 است.

۲- مدت زمان تأخیر فلیپ فلاپ ، برای تغییر حالت خروجی.

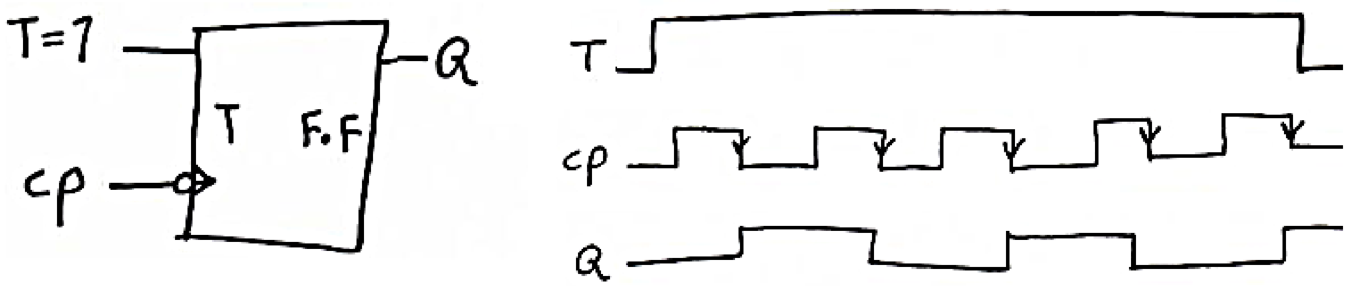
برای کنترل تعداد دفعات تغییر حالت در خروجی ، از آی‌سی‌های حساس به لبه استفاده می‌کنند.

در آی‌سی حساس به لبه مثبت ، با ایجاد هر لبه مثبت ، خروجی می‌تواند با توجه به ورودی یک بار تغییر کند.



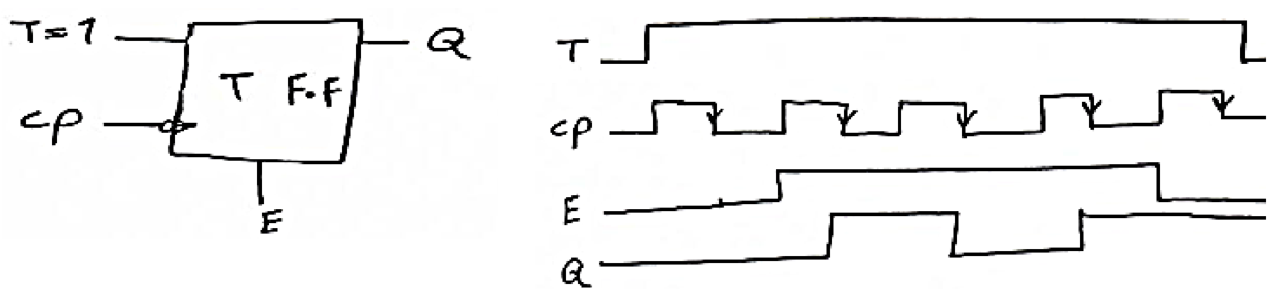
آی‌سی با لبه منفی :

در این نوع آی‌سی با ایجاد هر لبه منفی ، خروجی میتواند با توجه به ورودی یکبار تغییر کند.



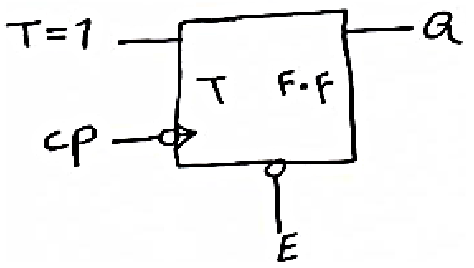
آی‌سی با فعال ساز مثبت :

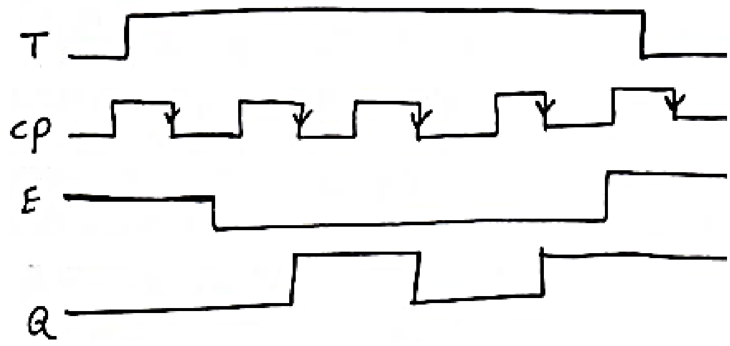
در این نوع آی‌سی مادامی‌که E=0 باشد ، خروجی بدون تغییر باقی می‌ماند . اما اگر E=1 باشد آی‌سی روشن است و می‌تواند با توجه به ورودی‌هایش تغییر کند. در شکل زیر یک آی‌سی حساس به لبه منفی با فعال ساز مثبت را به همراه خروجی‌هایش مشاهده می‌کنید.



آی‌سی با فعال ساز منفی:

در این نوع آی‌سی ، مادامی‌که E=1 باشد ، خروجی بدون تغییر باقی می‌ماند . اما اگر E=0 باشد، آی‌سی روشن است و می‌تواند با توجه به ورودی‌هایش تغیر کند. در شکل زیر یک آی‌سی حساس به لبه منفی با فعال ساز مثبت را به همان خروجی‌هایش مشاهده می‌کنید.





آی‌سی همراه با Reset :

آی‌سی‌های با پایه‌ی Reset مثبت :

در این نوع آی‌سی مادامی‌که R=0 باشد، آی‌سی روشن است . (یعنی Q’=0 و Q=1 می‌شود.)

و اگر R=1 باشد آی‌سی خاموش است . (یعنی Q=0 و Q’=1 می‌شود.)

آی سی با پایه Reset منفی :

در این نوع آی‌سی مادامی‌که R=1 باشد، آی‌سی روشن است . (یعنی Q’=0 و Q=1 می‌شود.)

و اگر R=0 باشد آی‌سی خاموش است . (یعنی Q=0 و Q’=1 می‌شود.)

نکته ۱ : گاهی پایه Reset را پاک کننده یا Clear نیز می‌نامند.

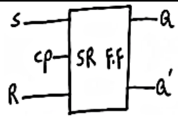
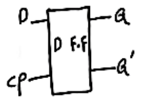
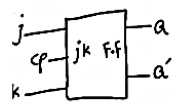
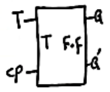
نکته ۲ : تفاوت پایه R با پایه E آن است که با فعال شدن R خروجی صفر می‌شود. اما با غیر فعال شدن E خروجی بدون تغییر باقی می‌ماند و حالت قبل خود را حفظ می‌کند.

تمرین :

۱- الف) انواع فیلیپ‌فلاپ‌ها را نام ببرید . ب) نمودار بلوکی هر کدام را رسم کنید .

ج) جدول صحّت هر یک را بکشید .

جواب :

	SR F.F	D F.F	JK F.F	T F.F
الف	SR F.F	D F.F	JK F.F	T F.F
ب				

ج)

SR F.F:

S	R	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	غ ق ق

D F.F:

D	Q(t+1)
0	0
1	1

JK F.F:

J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	Q/(t)

T F.F:

T	Q(t+1)
0	Q(t)
1	Q/(t)

****

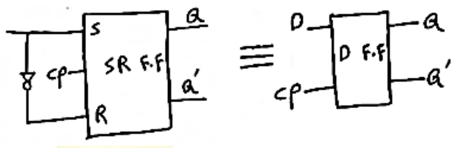
۲- الف) با یک SR.F.F یک D.F.F بسازید . ب) با یک JK.F.F یک T.F.F بسازید .

ج) با یک JK.F.F یک D.F.F بسازید . د) با یک D.F.F یک T.F.F بسازید .

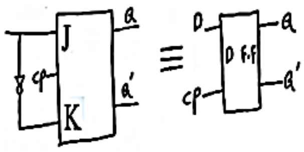
ه) با یک SR.F.F یک T.F.F بسازید .

جواب :

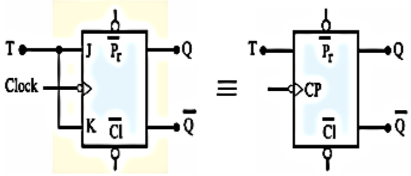
الف) ساخت D F.F با SR F.F :



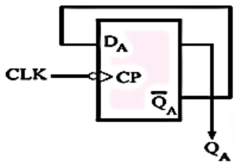
ج) ساخت D F.F با JK F.F:



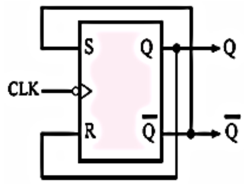
ب) ساخت T F.F با JK F.F :



د) ساخت T F.F با D F.F:

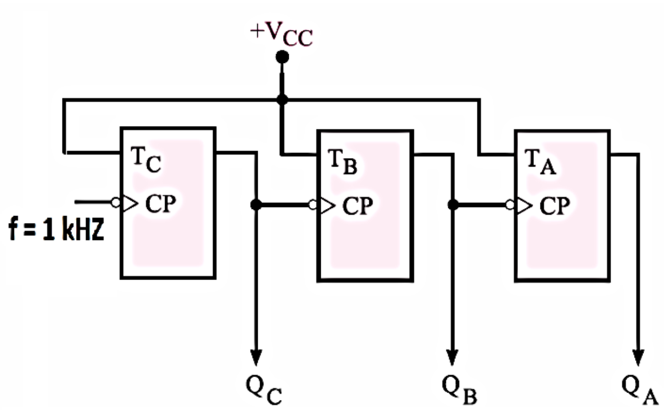


ه) ساخت T F.F با SR F.F:



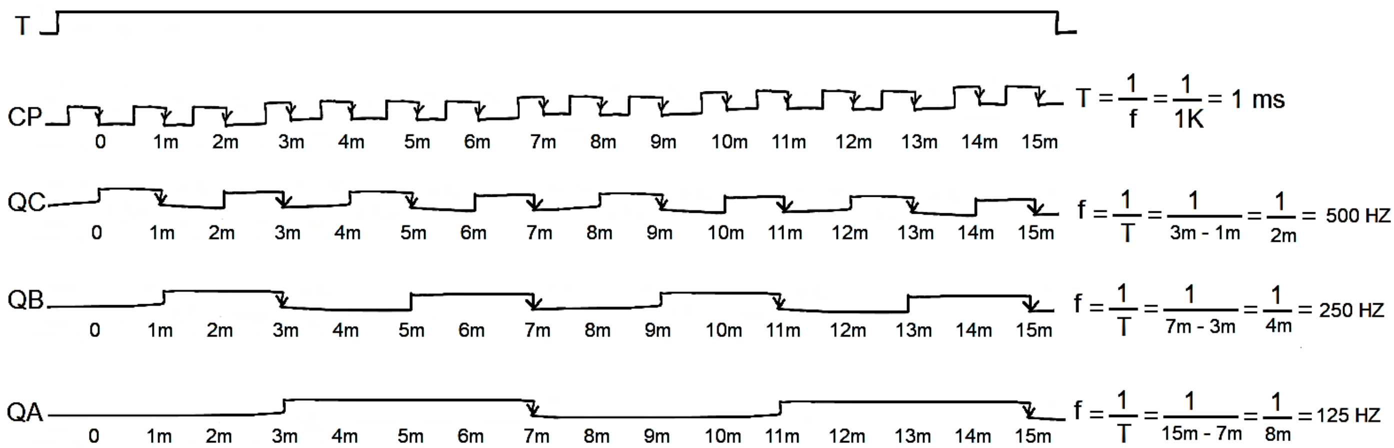
****

۳- اگر سه طبقه فلیپ فلاپ را مطابق شکل به هم وصل کنیم یک شمارشگر باینری سه بیتی تشکیل می‌شود . فرکانس و شکل موج خروجی هر یک از فلیپ فلاپ ها را به دست اورید.



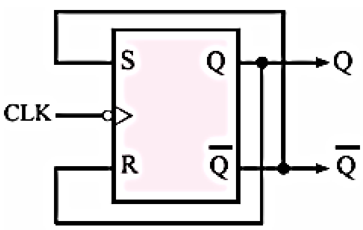
جواب :

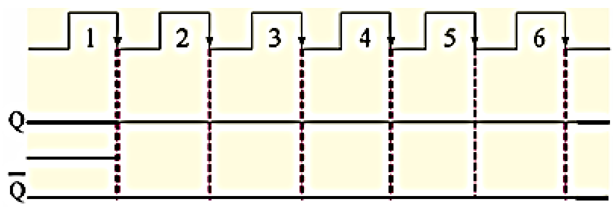
همه فیلیپ‌فلاپ‌ها حساس به لبه منفی هستند . پس داریم :



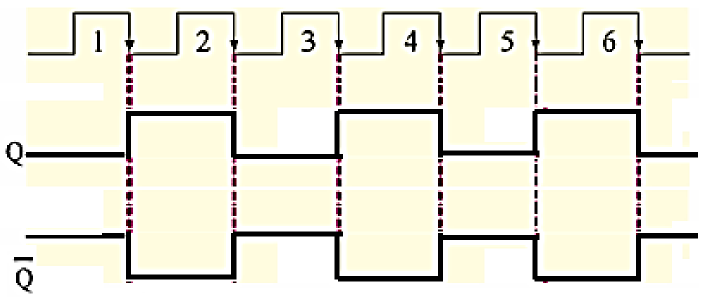
****

۴- با فرض آن که فلیپ فلاپ شکل زیر در ابتدا Reset باشد تغییرات بعدی وضعیت خروجی Q را در نمودار رسم کنید. نتیجه حاصل رفتار کدام یک از فلیپ فلاپ ها را نشان می دهد؟



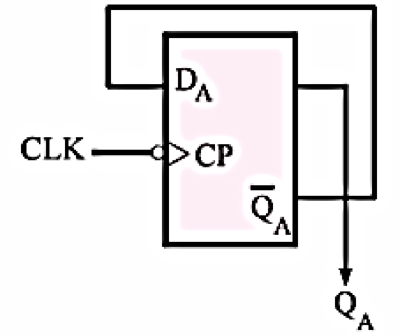


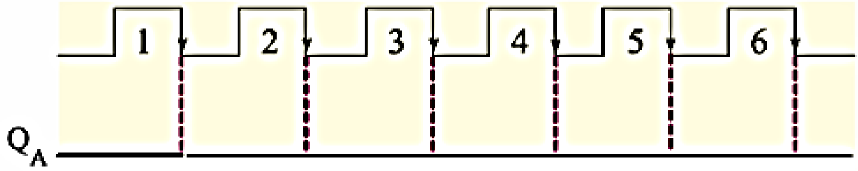
جواب :



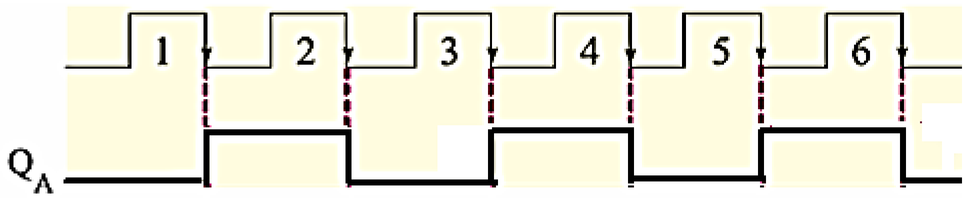
پس معادل یک T.F.F عمل می‌کند .

۵- فرض کنید فلیپ فلاپ شکل زیر در آغاز در حالت Reset قرار دارد. پالس دیاگرام زمانی QA را در نمودار رسم کنید. رفتار این مدار معادل کدام فلیپ فلاپ است؟





جواب :



پس معادل یک T.F.F عمل می‌کند .

****

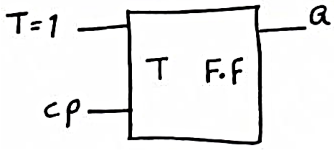
۶- انواع CP ها را نام برده و هر یک را تشریح کنید .

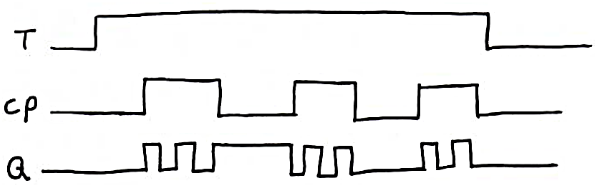
جواب :

<u>آی‌سی‌حساس به سطح مثبت :</u>

در این نوع آی‌سی مادامی‌که CP = 1 باشد ، خروجی می‌تواند با توجه به ورودی تغییر کند.

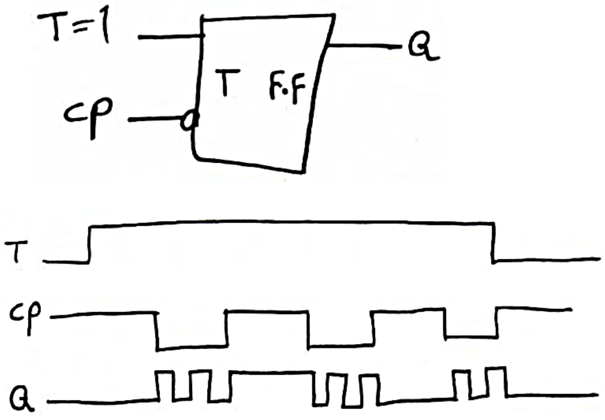
مدار زیر را در نظر بگیرید. از آنجا که T=1 است و خروجی در هر لحظه مرتباً مکمل حالت قبل می‌شود. به عبارتی مادامی‌که CP =1 باشد خروجی مرتباً تغیر می‌کند. یعنی:





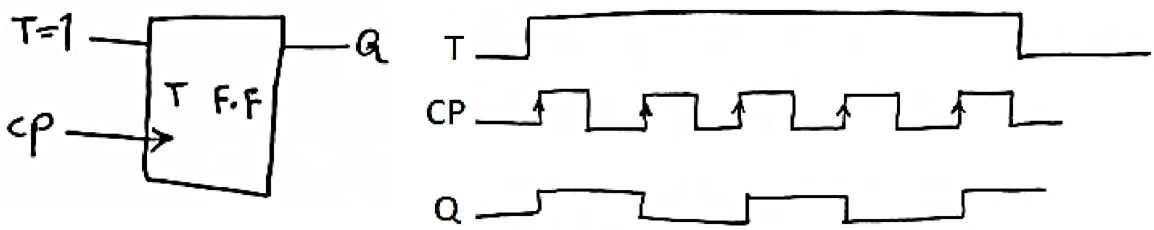
آی‌سی‌حساس به سطح منفی:

در این آی‌سی مادامی‌که CP =1 باشد ، خروجی می‌تواند با توجه به ورودی تغییر کند.



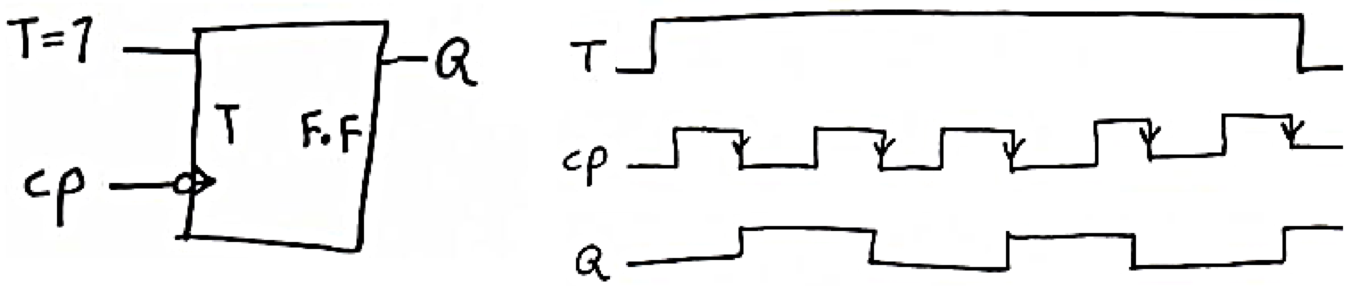
آی سی حساس به لبه مثبت:

در آی‌سی حساس به لبه مثبت ، با ایجاد هر لبه مثبت ، خروجی می‌تواند با توجه به ورودی یک بار تغییر کند.



آی‌سی با لبه منفی :

در این نوع آی‌سی با ایجاد هر لبه منفی ، خروجی میتواند با توجه به ورودی یکبار تغییر کند.



۷- الف) عیب آی‌سی‌های حساس به سطح چیست ؟
ب) چگونه آن‌را برطرف می‌کنند .

جواب :

الف) قبلاً گفتیم در آی‌سی حساس به سطح ، مادامی‌که CP =1 باشد خروجی مرتباً تغییر می‌کند. بعبارتی مادامی‌که CP =1 باشد ، خروجی مرتباً تغییر می‌کند . اما در حالتی‌که CP =0 باشد خروجی بدون تغییر باقی می‌ماند. حال سؤال اینجاست که در حالت بدون تغییر ، خروجی صفر است یا یک ؟ یا اینکه اصلاً مادامی‌که CP =1 باشد خروجی چند بار تغییر می کند؟ پاسخ به این سؤالات به چند عامل بستگی دارد.

۱- مدت زمانی که CP =1 است.
۲- مدت زمان تأخیر فلیپ فلاپ ، برای تغییر حالت خروجی.

ب) برای کنترل تعداد دفعات تغییر حالت در خروجی ، از آی‌سی‌های حساس به لبه استفاده می‌کنند.

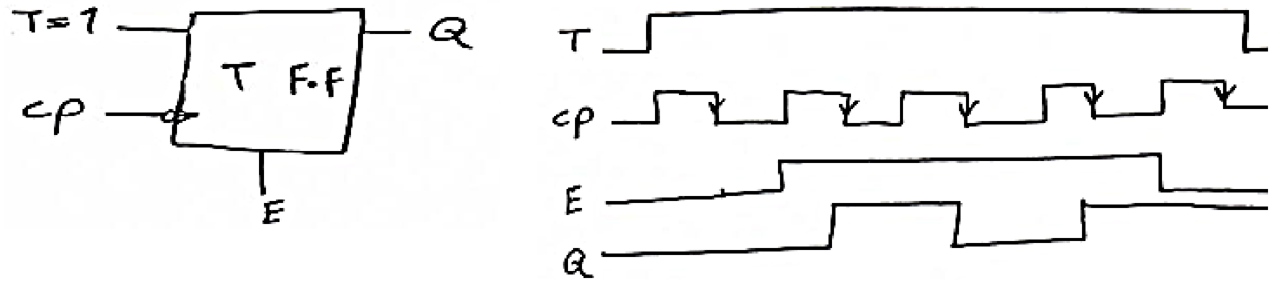
****

۸- انواع فعال‌سازها را نام برده و هر یک را تشریح کنید .

جواب :

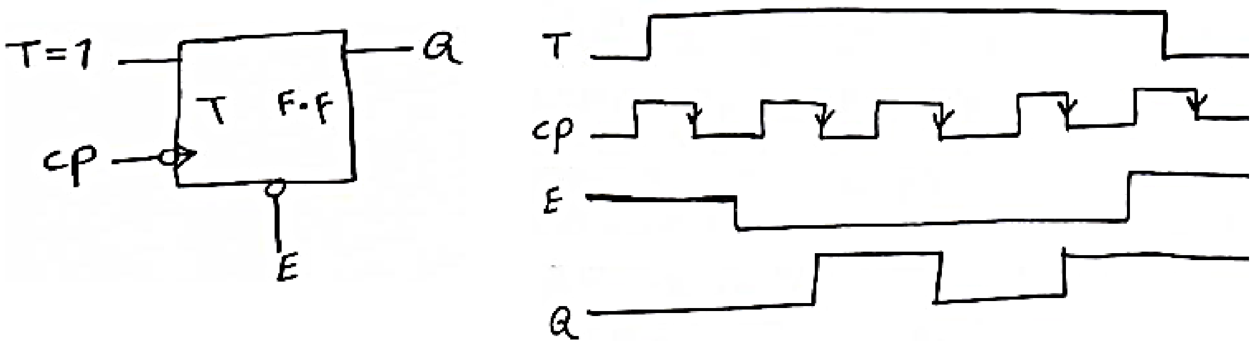
<u>آی‌سی با فعال ساز مثبت :</u>

در این نوع آی‌سی مادامی‌که E=0 باشد ، خروجی بدون تغییر باقی می‌ماند . اما اگر E=1 باشد آی‌سی روشن است و می‌تواند با توجه به ورودی‌هایش تغییر کند. در شکل زیر یک آی‌سی حساس به لبه منفی با فعال ساز مثبت را به همراه خروجی‌هایش مشاهده می‌کنید.



<u>آی‌سی با فعال ساز منفی:</u>

در این نوع آی‌سی ، مادامی‌که E=1 باشد ، خروجی بدون تغییر باقی می‌ماند . اما اگر E=0 باشد، آی‌سی روشن است و می‌تواند با توجه به ورودی‌هایش تغیر کند. در شکل زیر یک آی‌سی حساس به لبه منفی با فعال ساز مثبت را به همان خروجی‌هایش مشاهده می‌کنید.



****

۹- انواع ریست ها را نام برده و هر یک را تشریح کنید .

جواب :

<u>آی‌سی‌های با پایه‌ی Reset مثبت :</u>

در این نوع آی‌سی مادامی‌که R=0 باشد، آی‌سی روشن است . (یعنی Q’=0 و Q=1 می‌شود.)

و اگر R=1 باشد آی‌سی خاموش است . (یعنی Q=0 و Q’=1 می‌شود.)

<u>آی سی با پایه Reset منفی :</u>

در این نوع آی‌سی مادامی‌که R=1 باشد، آی‌سی روشن است . (یعنی Q’=0 و Q=1 می‌شود.)

و اگر R=0 باشد آی‌سی خاموش است . (یعنی Q=0 و Q’=1 می‌شود.)

<u>نکته :</u> گاهی پایه Reset را پاک کننده یا Clear نیز می‌نامند.

****

۱۰- پایه ریست و پایه فعال‌ساز چه تفاوتی دارند ؟

جواب :

تفاوت پایه R با پایه E آن است که با فعال شدن R خروجی صفر می‌شود. اما با غیر فعال شدن E خروجی بدون تغییر باقی می‌ماند و حالت قبل خود را حفظ می‌کند.

****

فصل هفتم : شیفت رجیسترها (Shift Registers)

یک ثبات یا رجیستر ، مجموعه‌ای از فلیپ فلاپ‌ها (سلول‌های حافظه) است ، که می‌تواند اطلاعات دودویی (باینری) را در خود نگه دارد. رجیستری که قادر است اطلاعات باینری ذخیره شده در خود را به سمت راست یا چپ انتقال دهد ، شیفت رجیستر می‌نامند.

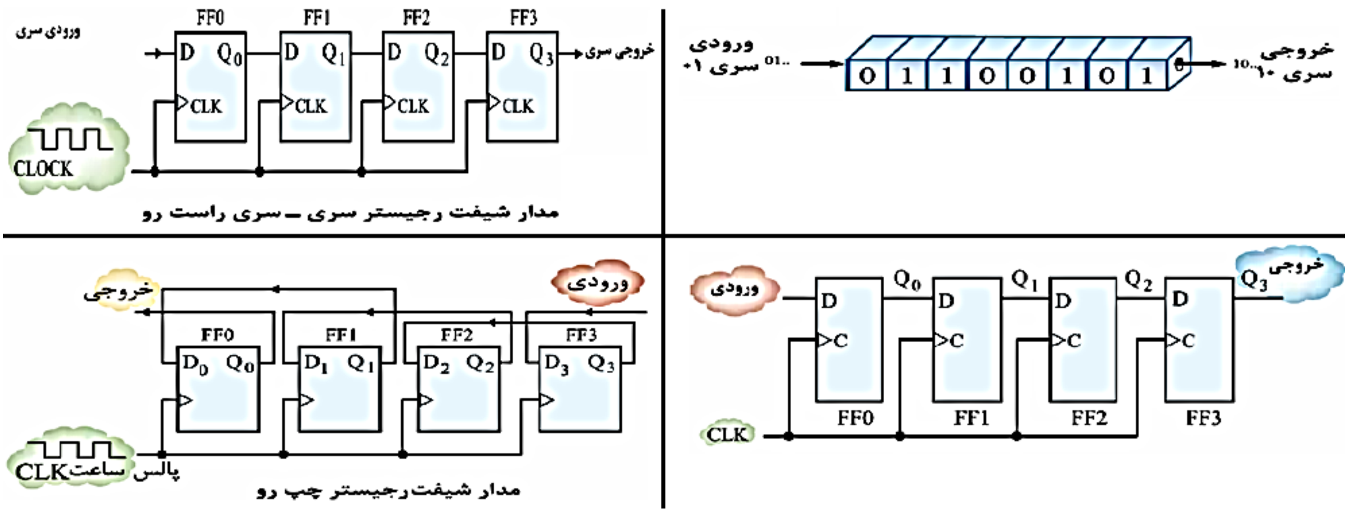
اتصال فلیپ فلاپ‌ها به گونه‌ای است که یک رشته ارقام باینری به آن‌ها وارد ، یا از آن‌ها خارج می‌شود. این نوع مدارها را معمولاً شیفت رجیستر یا ثبات انتقالی می‌نامند. یک شیفت رجیستر n بیتی از n فلیپ فلاپ تشکیل می‌شود و می‌تواند n بیت اطلاعات را در خود ذخیره کند. یک نمونه از کاربرد شیفت رجیستر در ماشین حساب است.

یک شیفت رجیستر ۸ بیتی از ۸ فلیپ فلاپ تشکیل شده است . انتقال اطلاعات از ورودی به خروجی ممکن است از طریق لبه‌های بالارونده ، یا پایین رونده مثبت یا منفی پالس ساعت ورودی ، صورت بگیرد.

انواع شیفت رجیستر:

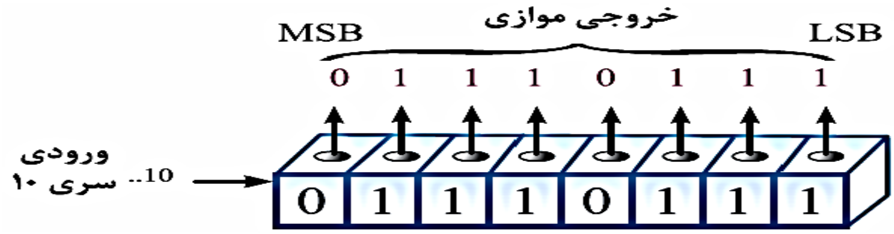
بر حسب اینکه اطلاعات چگونه ثبت و به چه صورت خوانده شود ، شیفت رجیسترها را به چهار گروه زیر دسته‌بندی می‌کنند :

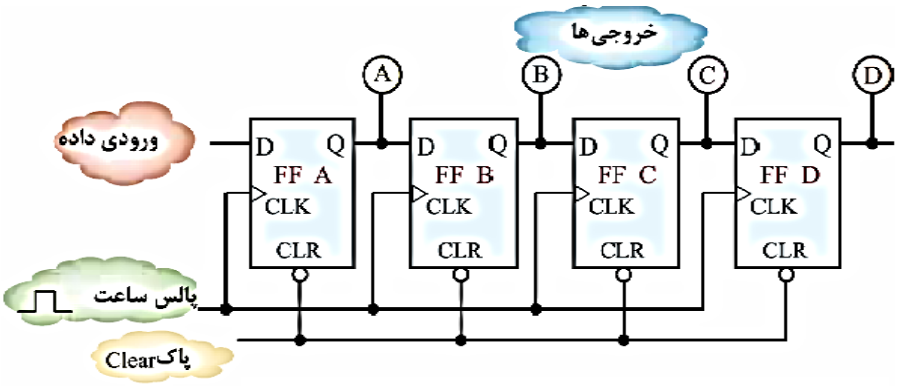
۱- ورودی سری ، خروجی سری یا متوالی ، متوالی یا سری ، سری یا SISO (Serial input – Serial output) :



۲- ورودی سری ، خروجی موازی یا متوالی ، موازی یا سری ، موازی

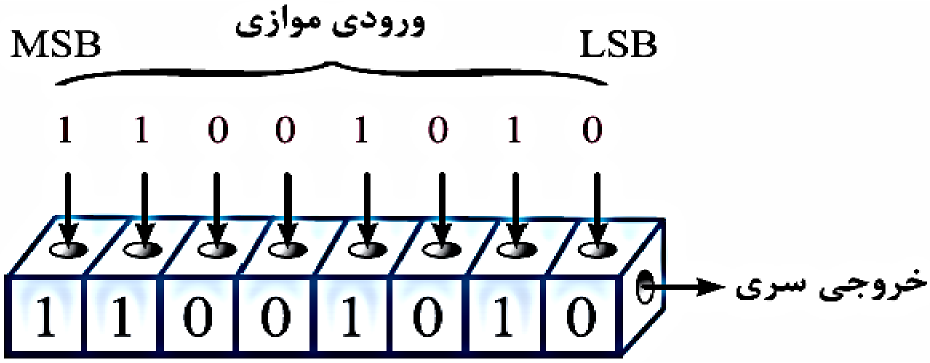
یا SIPO (Serial input – Paralell output) :

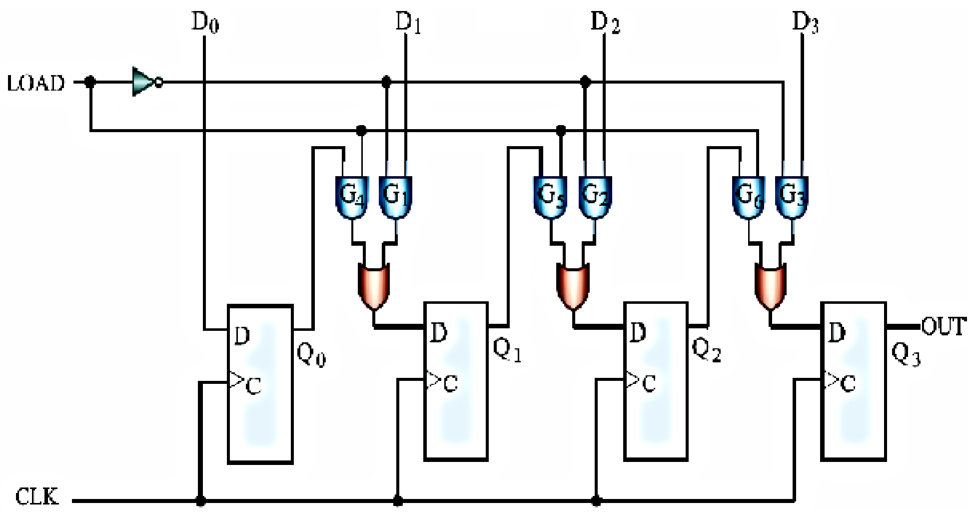




۳- ورودی موازی ، خروجی سری یا موازی ، متوالی یا موازی ، سری

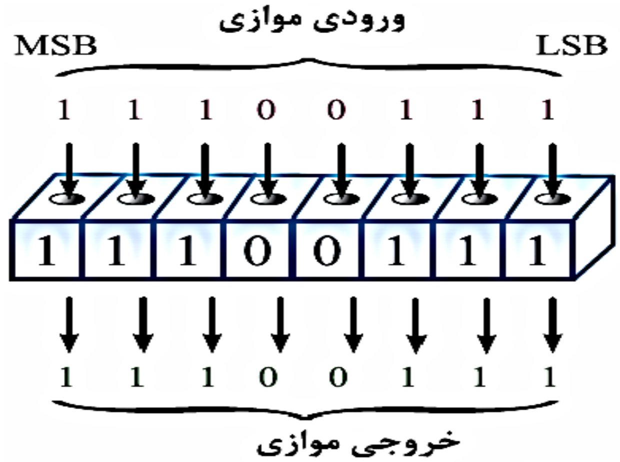
یا PISO (Serial output – Paralell input) :

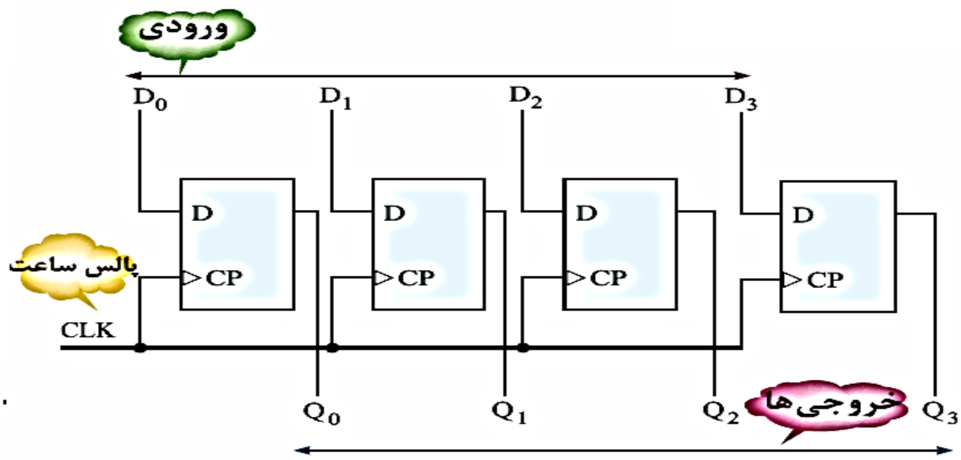


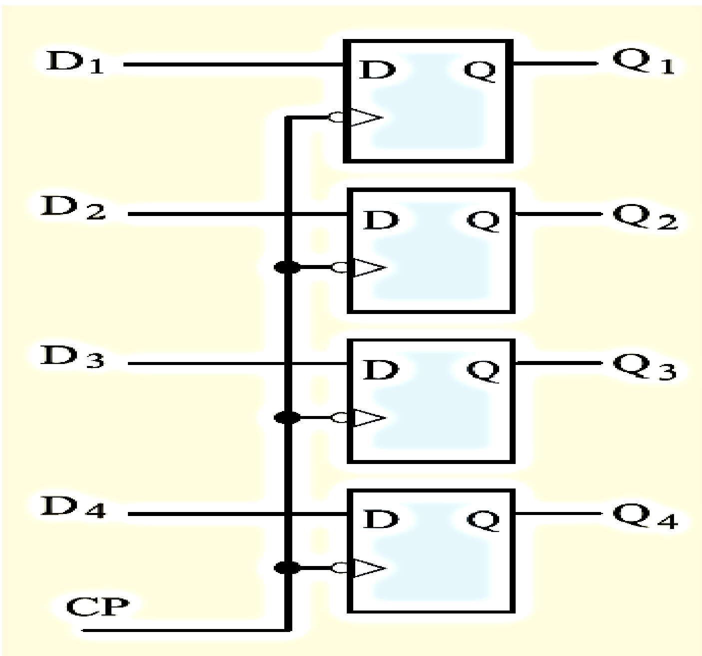


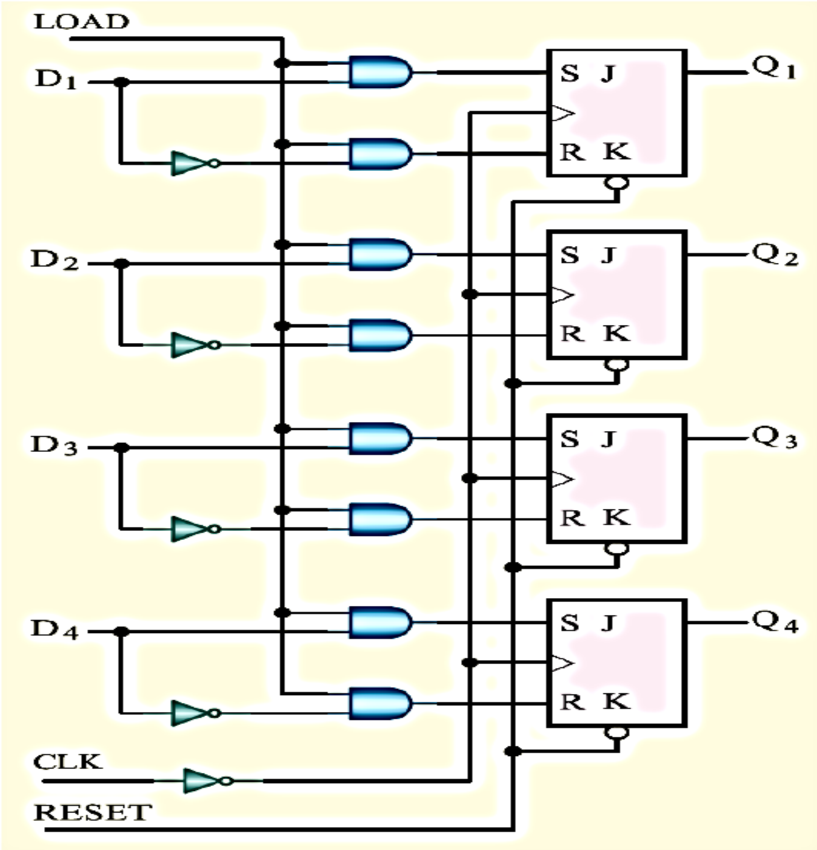
۴- ورودی موازی ، خروجی موازی یا موازی ، موازی

یا PIPO (Paralell input –Paralell output) :

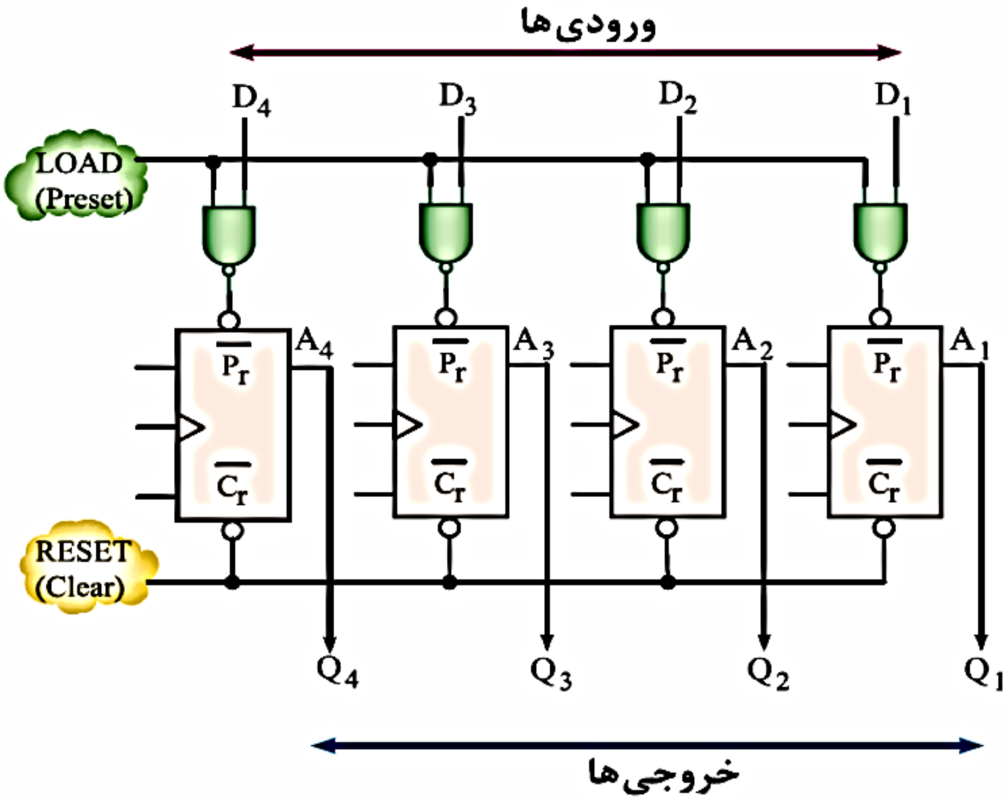








انتقال اطلاعات در شیفت رجیستر ، می‌تواند از طریق ورودی‌های Preset مطابق شکل زیر نیز انجام شود. توجه داشته باشید که قبل از انتقال اطلاعات باید محتویات قبلی فلیپ فلاپ با استفاده از ورودی Preset پاک شود.



عملیات سری ، به دلیل زمانی که برای انتقال این اطلاعات به داخل و خارج از شیفت رجیسترها صرف می‌کند، کندتر است ؛ ولی از نظر سخت افزاری به مدارهای کمتری نیاز دارد. با مقایسه شیفت رجیسترهای سری و موازی می‌توان نتیجه گرفت که حالت‌های موازی از سرعت بیشتری نسبت به حالت‌های سری برخوردارند.

تمرین :

۱- فرق بین انتقال سری و موازی چیست؟

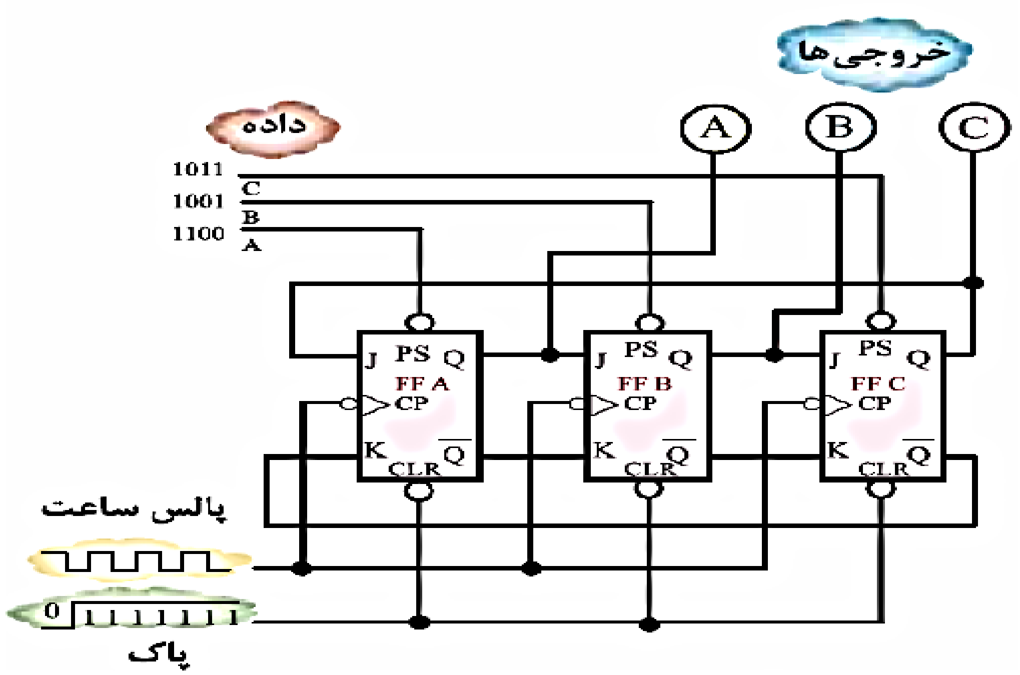
جواب :

عملیات سری ، به دلیل زمانی که برای انتقال این اطلاعات به داخل و خارج از شیفت رجیسترها صرف می‌کند، کندتر است ؛ ولی از نظر سخت افزاری به مدارهای کمتری نیاز دارد. با مقایسه شیفت رجیسترهای سری و موازی می‌توان نتیجه گرفت که حالت‌های موازی از سرعت بیشتری نسبت به حالت‌های سری برخوردارند.

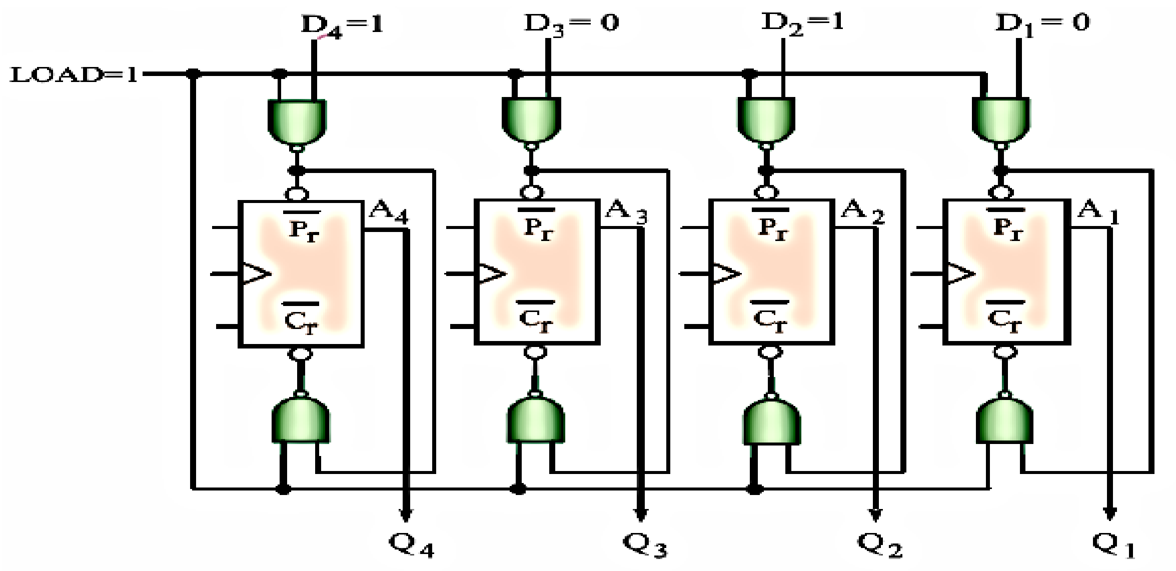
****

۲- با تحلیل هر یک از مدارهای زیر، نوع شیفت رجیستر را مشخص کنید.

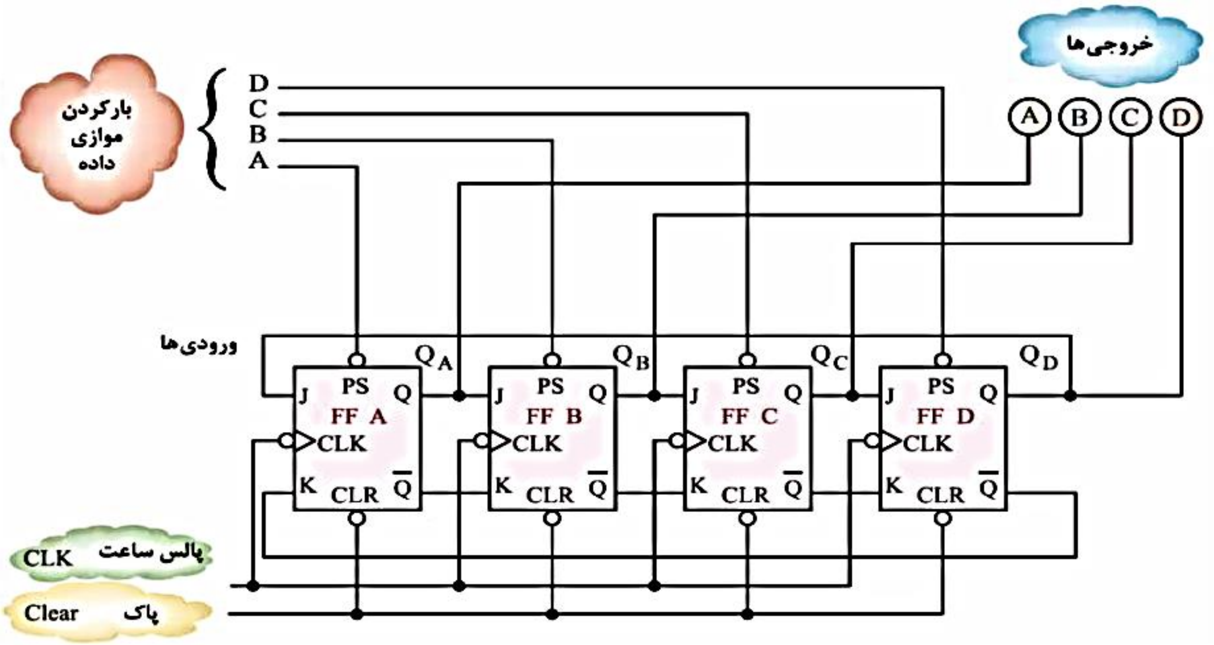
الف :



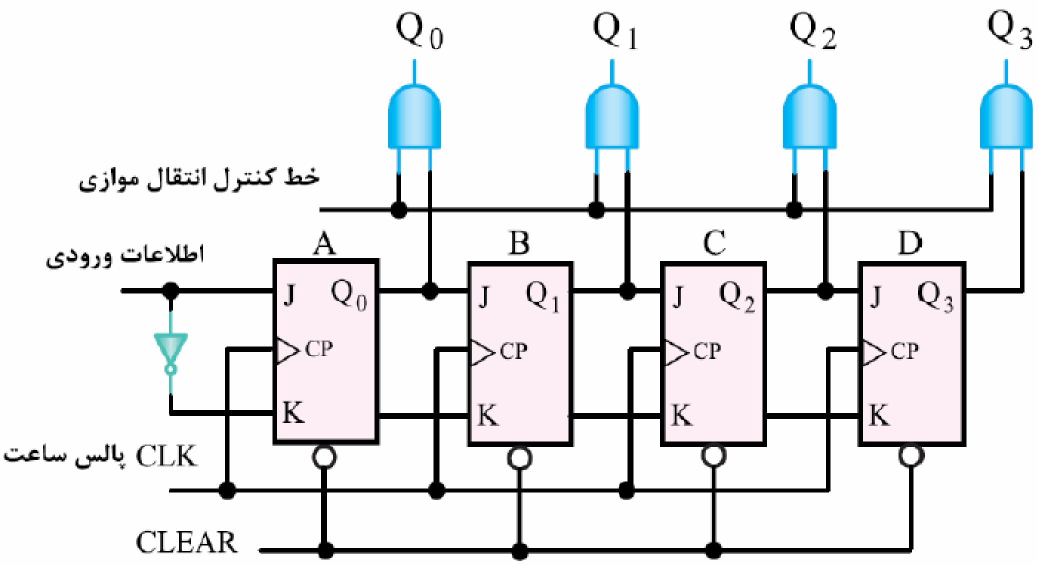
ب :



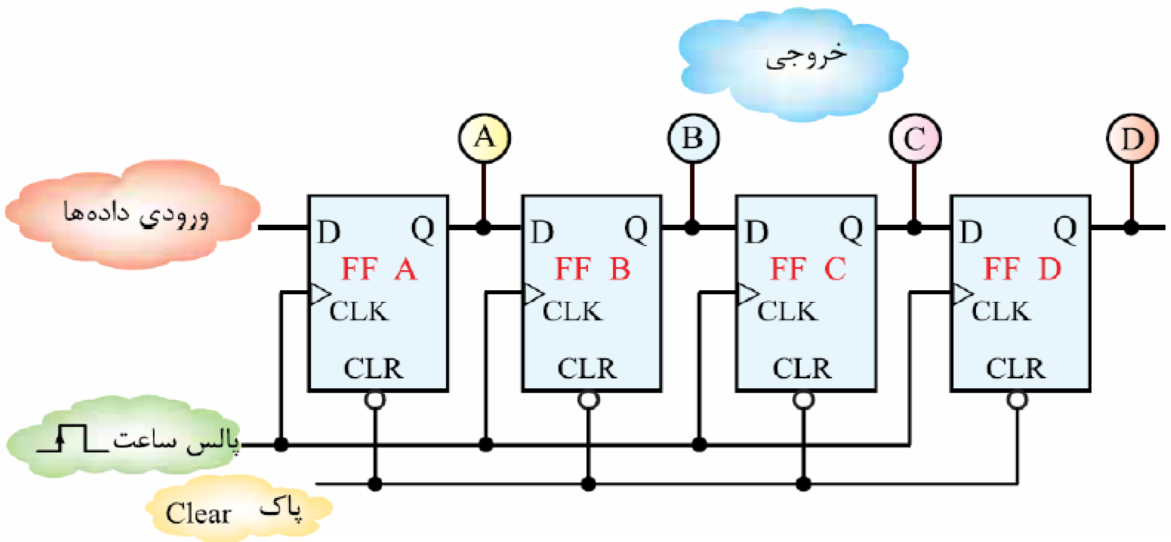
ج :



د :



ه :



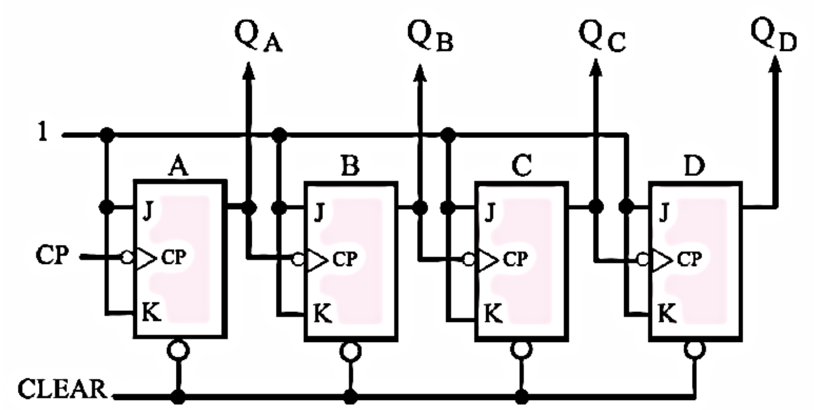
جواب :

الف : موازی ، موازی (PIPO)   ب : موازی ، موازی (PIPO)   ج : موازی ، موازی (PIPO)

د : سری ، موازی (SIPO)   ه : سری ، موازی (SIPO)

فصل هشتم : شمارنده‌های آسنکرون (Asynchronou’s counters)

شمارنده‌ها از تعدادی فلیپ فلاپ تشکیل می‌شوند که به صورت سری به هم متصل شده‌اند . و عملاً پالس‌های ورودی به مدار را شمارش می‌کنند. شمارنده‌ها به عنوان تقسیم کننده فرکانس نیز به کار می‌روند. شمارش ممکن است بر مبنای ۱۰ یا هر مبنای دیگری انجام شود. عنصر اصلی هر شمارنده فلیپ فلاپ است. یک شمارنده با n طبقه فلیپ فلاپ ، حداکثر می تواند 2^n حالت تعریف شده داشته باشد. شکل زیر یک شمارنده ۴ بیتی را نشان می دهد.



تعداد وضعیت‌هایی که یک شمارنده قبل از رسیدن به حالت اولیه طی می‌کند را ، مدول (modules) یا پیمانه شمارنده می‌نامند. مثلاً یک شمارنده باینری ۳ بیتی از مدول ۸ یعنی ۸ وضعیتی ، و یک شمارنده باینری ۴ بیتی از مدول ۱۶ یعنی ۱۶ وضعیتی است.

انواع شمارنده‌ها :

شمارنده‌ها بر اساس نحوه کار آن‌ها به دو دسته شمارنده سنکرون (synchronous) یا هم زمان ، و شمارنده آسنکرون (Asynchronous) یا غیر هم زمان و تقسیم‌بندی می‌شوند.

شمارنده آسنکرون (غیر هم‌زمان) :

در شمارنده آسنکرون ، پالس ساعت فلیپ فلاپ‌ها به طور همزمان به آنها اعمال نمی‌شود . و هر طبقه پالس ساعت خود را از خروجی طبقه ماقبل خود دریافت می‌کند.

شمارنده آسنکرون به شمارنده ضربانی (Ripple Counter) نیز معروف است.

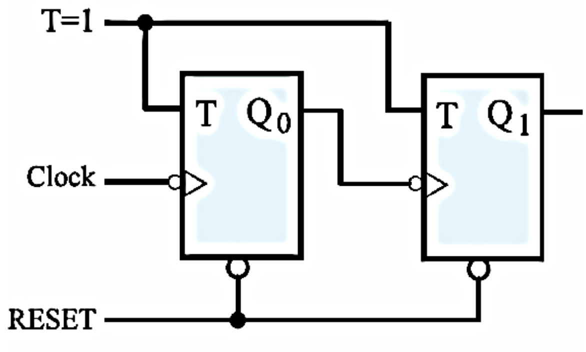
فلیپ فلاپ‌های به کار رفته در این نوع شمارنده از نوع T (یا معادل‌های آن) است و باید همواره T=1 باشد.

در شمارنده آسنکرون ، تغییر وضعیت هر فلیپ فلاپ به تغییر وضعیت فلیپ فلاپ طبقه ماقبل بستگی دارد. به همین دلیل سرعت عمل شمارنده های آسنکرون نسبت به شمارنده سنکرون مشابه کم تر است.

در شمارنده آسنکرون از فلیپ فلاپ نوع T (یا معادل‌های آن) استفاده میشود. مثلاً اگر از فلیپ فلاپ نوع JK استفاده کنیم ، باید ورودی های J و K را در کلیه طبقات همواره به هم وصل ، و در حالت یک نگه داریم.

شمارنده‌های آسنکرون می‌توانند اعداد را به طور منظم شمارش کنند. این نوع شمارنده‌ها نمی‌توانند هر ترتیب شمارش دل‌خواهی را اجرا کنند.

در شکل زیر یک شمارنده آسنکرون با دو فلیپ فلاپ نوع T نشان داده شده است. این شمارنده دارای چهار وضعیت مختلف (مدول۴) یا دوبیتی است.



شمارنده سنکرون (هم‌زمان) :

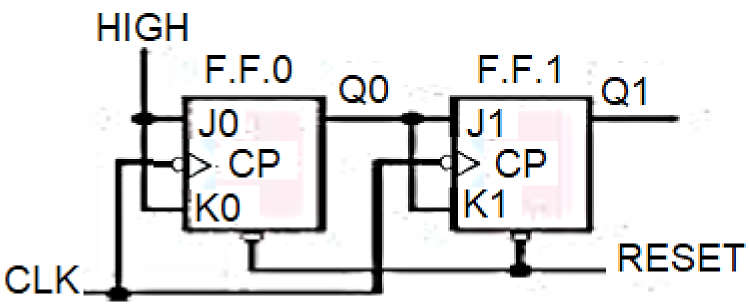
همانطور که ذکر شد ، در شمارنده آسنکرون ، تغییر وضعیت هر فلیپ فلاپ به تغییر وضعیت فلیپ فلاپ طبقه ماقبل آن بستگی دارد. به همین جهت سرعت عمل در این نوع شمارنده کم است.

هم‌چنین اگر فرکانس پالس ساعت زیاد شود، در شمارش خطا به وجود می‌آید.

در شمارنده سنکرون پالس ساعت کلیه فلیپ فلاپ ها از یک منبع تأمین می‌شود. بدین ترتیب اشکال مربوط به شمارنده آسنکرون را برطرف می‌کند.

اما در شمارنده سنکرون از گیت های بیشتری نسبت به شمارنده آسنکرون استفاده می‌شود.

در شکل زیر یک شمارنده‌ی سنکرون دوبیتی نشان داده شده است. که در آن از دو فلیپ فلاپ نوع T استفاده شده است.



شمارنده صعودی : در این نوع شمارنده ، خروجی‌ها به ترتیب از کمتر به بیشتر شمرده می‌شوند.

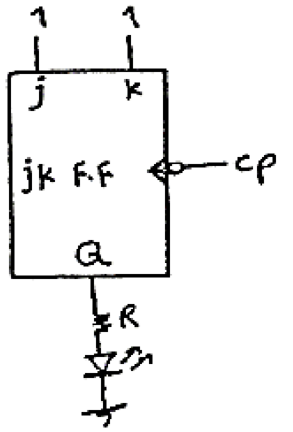
شمارنده نزولی : در این نوع شمارنده ، خروجی‌ها به ترتیب از بیشتر به کمتر شمرده می‌شوند.

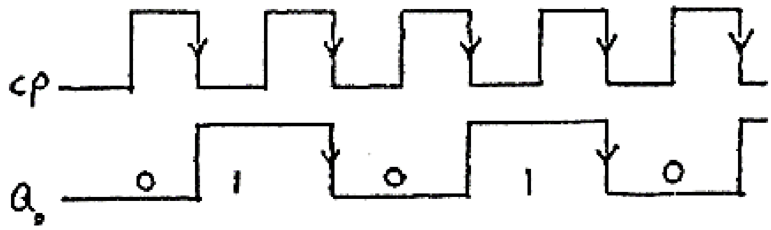
طراحی شمارنده‌های آسنکرون :

شمارنده مُد ۲ :

به مدار زیر دقت کنید.

از آنجاییکه J= K =1 است ، به ازای ایجاد هر لبه منفی ورودی JK F.F ، خروجی یک‌بار مکمّل می‌شود. یعنی :

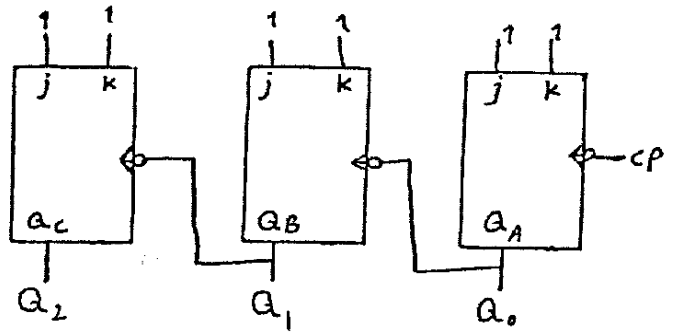




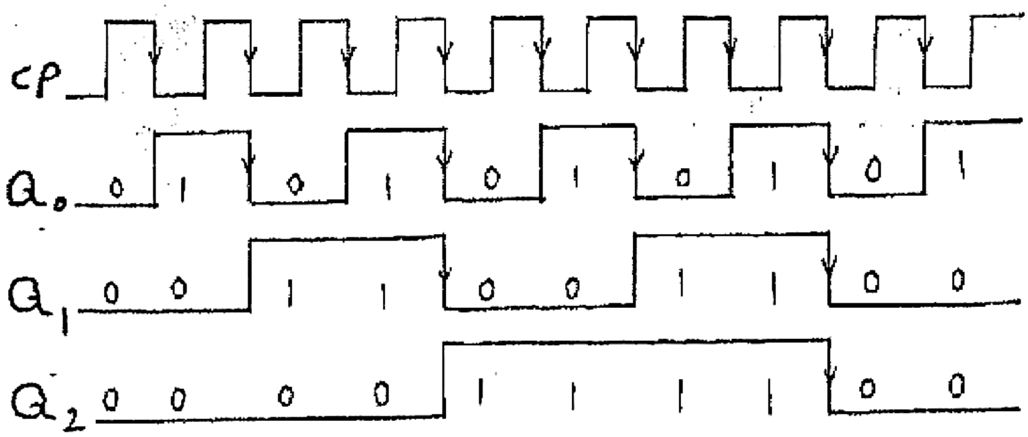
بعبارتی خروجی Q ، اعداد صفر و یک را مرتباً می‌شمارد. به مدار فوق شمارنده مُد۲ گویند.

شمارنده مُد ۸ :

به مدار زیر دقت کنید.



واضح است که به ازای ایجاد هر لبه منفی در CP ، خروجی فلیپ فلاپ A یکبار مکمّل می‌شود . و از آنجاییکه QA خود نقش CP فلیپ فلاپ B را بازی می‌کند ، ایجاد لبه منفی در QA ، تغییر در QB را سبب می‌شود. به همین ترتیب با ایجاد هر لبه منفی در QB ، خروجی QC یکبار مکمّل می‌شود.



حال بیایید خروجی‌ها را به ترتیب بنویسیم. داریم :

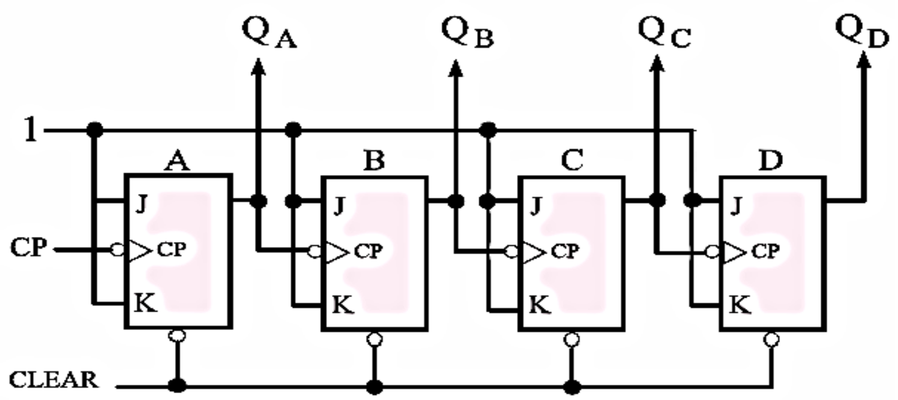
Q_C	Q_B	Q_A	خروجی
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7
0	0	0	0
0	0	1	1

همان‌طور که می‌بینید ، با ایجاد هر لبه منفی در CP ، در خروجی اعداد صفر تا هفت به ترتیب شمرده شده و مجدداً شمارش از اول صورت می‌گیرد.

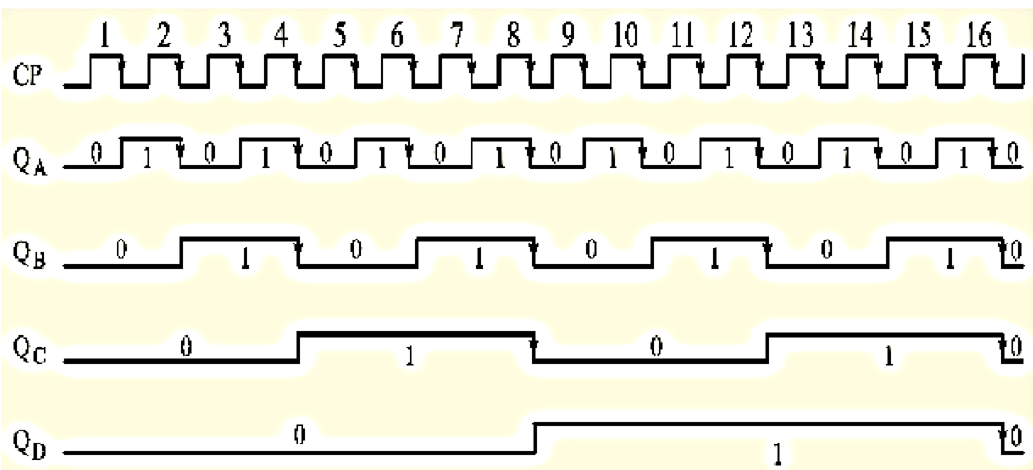
به مدار فوق شمارنده آسنکرون صعودی مُد ۸ می‌گویند.

شمارنده مُد ۱۶ :

به مدار زیر دقت کنید.



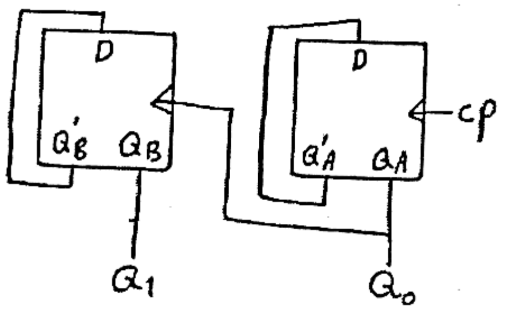
داریم :



پالس ساعت	Q_D	Q_C	Q_B	Q_A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

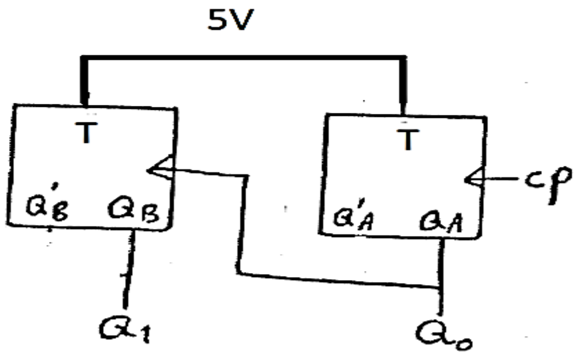
به مدار فوق شمارنده آسنکرون صعودی مُد ۱۶ می‌گویند.

مثال : مدار زیر را تحلیل کنید ، و بگویید چه نوع شمارنده‌ای است؟

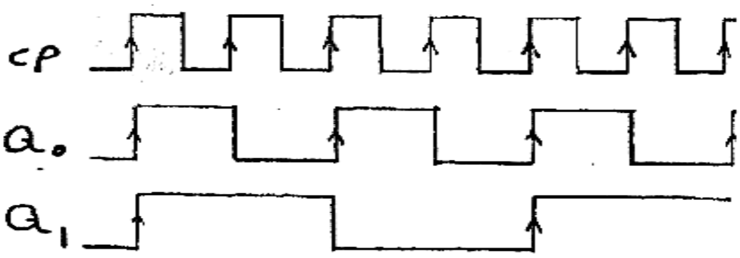


حل :

مدار فوق را می‌توان به شکل زیر نیز رسم کرد.



پس داریم :



Q1	Q0	عدد
1	1	3
1	0	2
0	1	1
0	0	0

بنابراین یک شمارنده آسنکرون نزولی مُد ۴ است.

نکته : برای طراحی شمارنده نزولی ، کافی‌است از لبه مثبت کلاک پالس استفاده کنیم.

روش طراحی مُدهای مختلف شمارنده آسنکرون :

برای این کار مراحل زیر را به ترتیب دنبال می‌کنیم :

۱- تعداد فلیپ‌فلاپ‌ها را تعیین کرده و آن‌ها را پشت سر هم می‌کشیم.

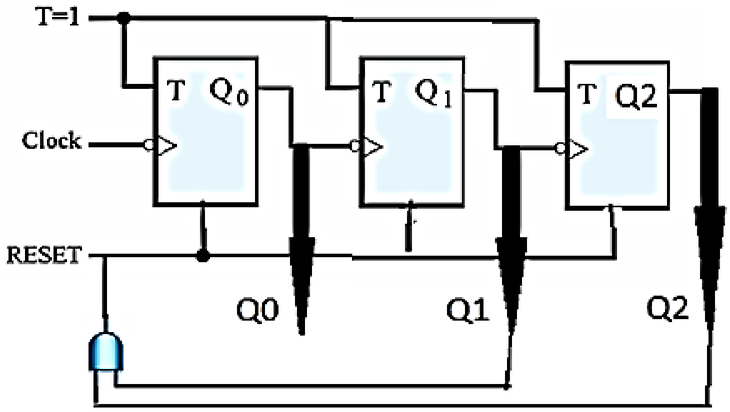
۲- برای شمارنده‌های صعودی از کلاک پالس حساس به لبه منفی ، و برای شمارنده‌های نزولی از کلاک پالس حساس به لبه مثبت استفاده می‌کنیم.

۳- با استفاده از گیت‌های منطقی ، کاری می‌کنیم که در عدد مورد نظر تمام فلیپ فلاپ‌ها ریسِت شوند.

مثال : یک شمارنده صعودی آسنکرون مُد ۶ طراحی کنید.

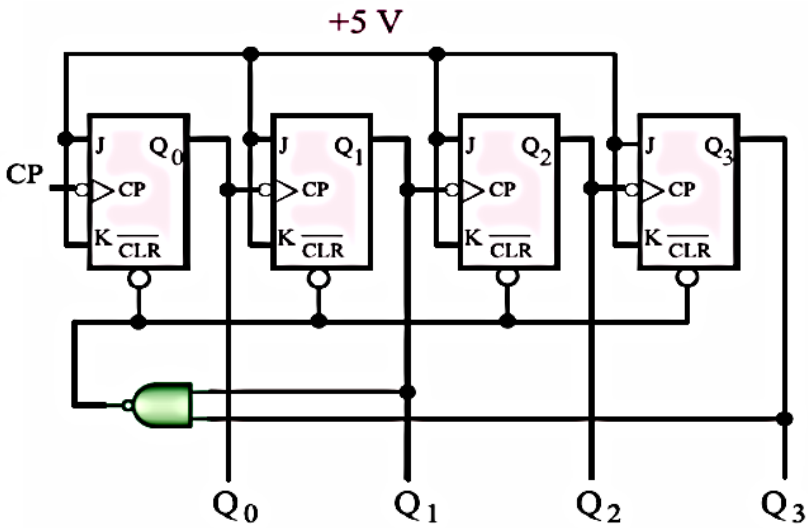
می‌خواهیم اعداد صفر (000) تا پنج (101) را بشماریم. پس سه عدد فلیپ فلاپ احتیاج داریم . و می‌خواهیم مدار در عدد شش (110) ریسِت شود.

از آنجایی‌که در عدد شش ، Q1 , Q2 هم‌زمان 1 می‌شوند ، آن‌ها را با هم AND کرده و به پایه RESET متصل می‌کنیم. با این کار به محض اینکه خروجی به عدد ۶ رسید ، تمام فیلیپ‌فلاپ‌ها ریسِت می‌شوند و شمارش از اول شروع می‌گردد. داریم :



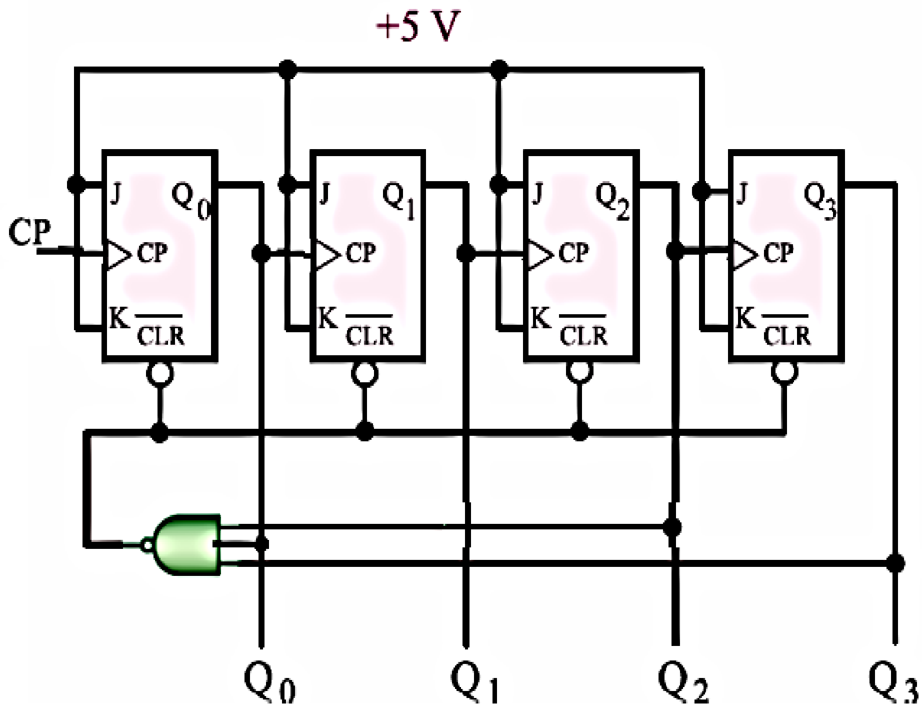
مثال : با استفاده از JK.F.F های حساس به RESET منفی ، یک شمارنده صعودی آسنکرون مُد ۱۰ طراحی کنید.

حل : عدد ۱۰ برابر است با : 1010 . پس باید Q3 , Q1 را AND کنیم . و چون RESET ها منفی هستند ، به‌جای AND از NAND استفاده می‌کنیم . داریم :



مثال : با استفاده از JK.F.F های حساس به RESET منفی ، یک شمارنده نزولی آسنکرون مُد ۱۳ طراحی کنید.

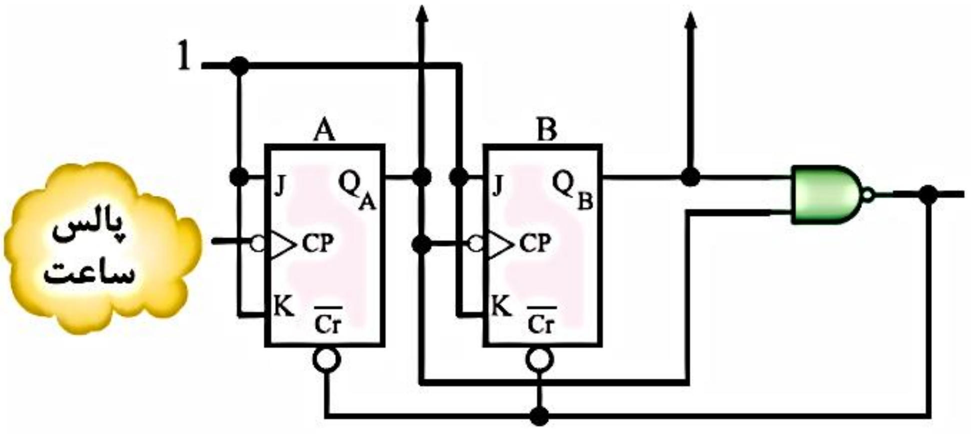
حل : چون شمارنده نزولی است ، از CP با لبه منفی استفاده می‌کنیم . عدد ۱۳ برابر است با : 1101
پس باید Q0 , Q2 , Q3 را AND کنیم . و چون RESET ها منفی هستند، به‌جای AND از NAND استفاده کرده‌ایم.
داریم :



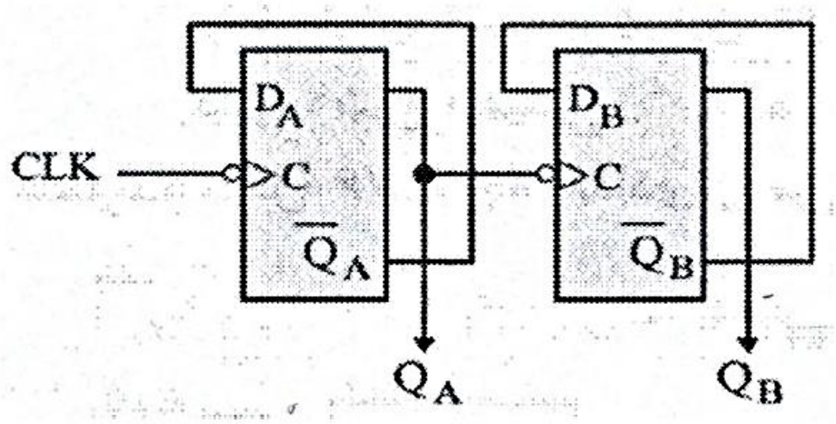
تمرین :

۱- هر یک از مدارهای زیر را به طور کامل تشریح کرده ، و نوع شمارنده را تعیین کنید.

الف :



ب :



جواب :

الف : CP هر طبقه از طبقه قبلش گرفته شده ، پس یک شمارنده آسنکرون است . از دو JK.F.F با کلاک پالس‌های حساس به لبه منفی ، و ریسِت منفی تشکیل شده است . چون از لبه منفی برای کلاک پالس استفاده شده ، شمارنده صعودی است. وقتی که QA QB = 11 شوند ، مدار ریسِت می‌شود ، پس مُد ۳ است .

بنابراین یک شمارنده صعودی ، دوبیتی ، موازی ، آسنکرون ، مُد ۳ است که اعداد ۰ تا ۲ را به ترتیب می‌شمارد .

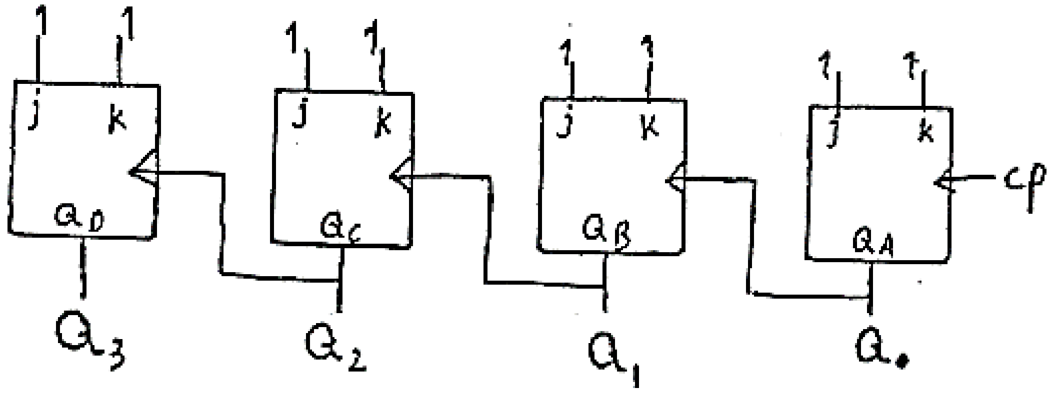
ب : CP هر طبقه از طبقه قبلش گرفته شده ، پس یک شمارنده آسنکرون است . از دو D.F.F با کلاک پالس‌های حساس به لبه منفی تشکیل شده است . چون از لبه منفی برای کلاک پالس استفاده شده ، شمارنده صعودی است. مدار ریسِت نمی‌شود ، پس مُد ۴ است .

بنابراین یک شمارنده صعودی ، دوبیتی ، موازی ، آسنکرون ، مُد ۴ است که اعداد ۰ تا ۳ را به ترتیب می‌شمارد .

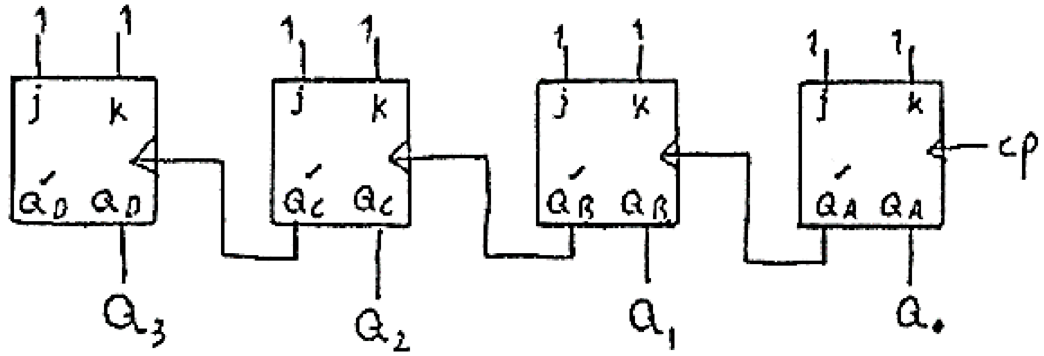
۲- هر یک از مدارهای زیر را تحلیل کرده و عملکرد آن‌را به طور کامل تشریح نمایید.
در صورت امکان بگویید چه نوع شمارنده‌ای است.

نکته : لبه مثبت Q' مانند لبه منفی Q عمل می‌کند و بالعکس.

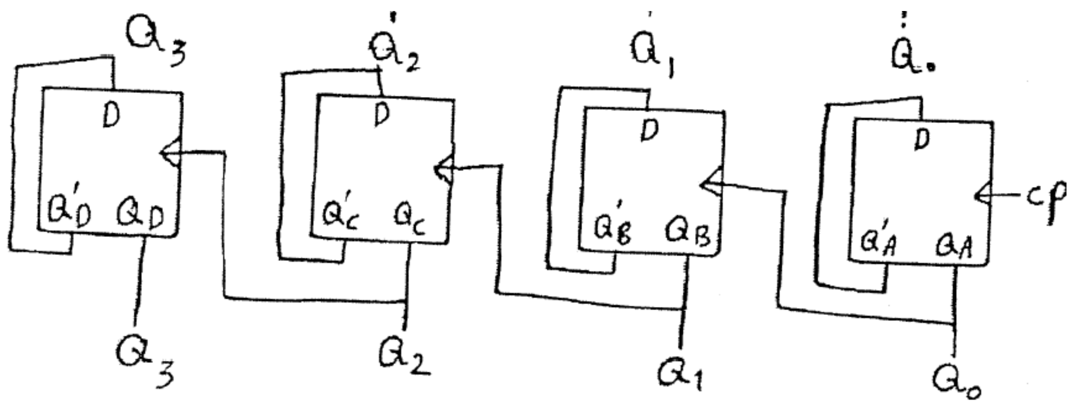
الف :



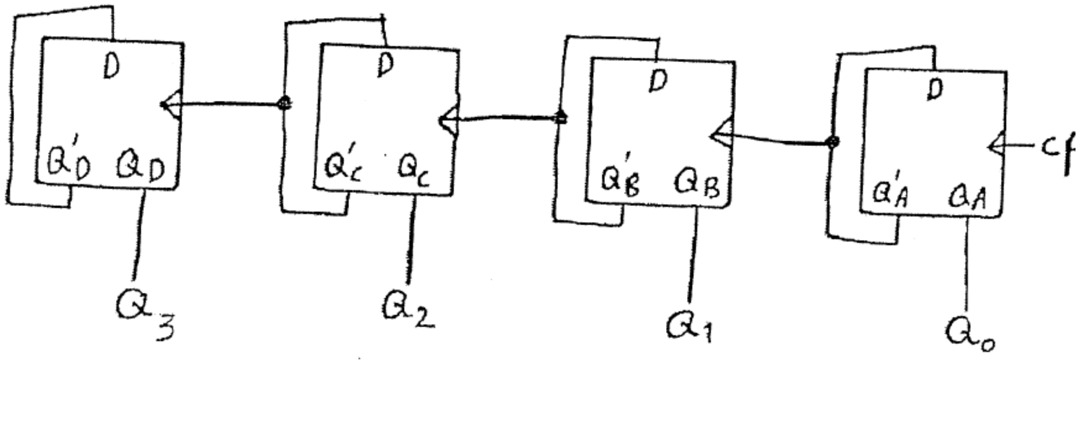
ب :



ج :



د :



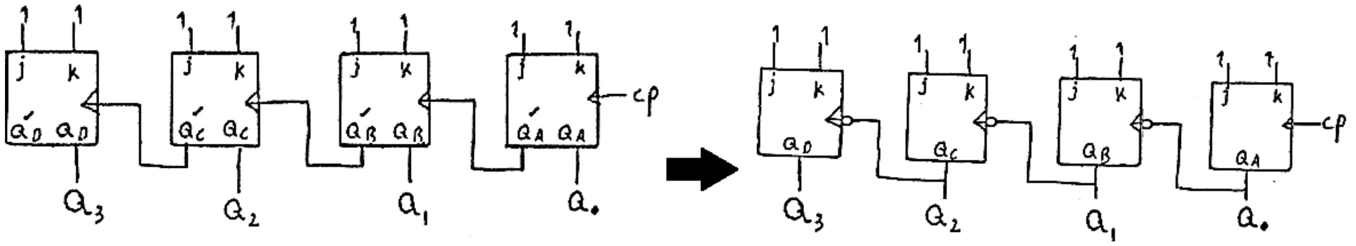
جواب :

<u>الف :</u> CP هر طبقه از طبقه قبلش گرفته شده ، پس یک شمارنده آسنکرون است . از چهار JK.F.F با کلاک پالس‌های حساس به لبه مثبت تشکیل شده است . چون از لبه مثبت برای کلاک پالس استفاده شده ، شمارنده نزولی است.

مدار ریسِت نمی‌شود ، پس مُد ۱۶ است .

بنابراین یک شمارنده نزولی ، چهاربیتی ، موازی ، آسنکرون ، مُد ۱۶ است که اعداد ۱۵ تا ۰ را به ترتیب می‌شمارد .

<u>ب :</u> می‌دانیم لبه مثبت Q' مانند لبه منفی Q عمل می‌کند و بالعکس. پس مدار فوق را می‌توان به شکل زیر کشید .



چنانچه مشاهدا می‌شود مدار مانند قسمت الف است با این تفاوت که از لبه منفی CP استفاده شده ، پس شمارنده صعودی است. بنابراین یک شمارنده صعودی ، چهاربیتی ، موازی ، آسنکرون ، مُد ۱۶ است که اعداد ۰ تا ۱۵ را به ترتیب می‌شمارد.

<u>ج :</u> مانند قسمت الف است با این تفاوت که از D.F.F استفاده شده است . بنابراین :

CP هر طبقه از طبقه قبلش گرفته شده ، پس یک شمارنده آسنکرون است . از چهار D.F.F با کلاک پالس‌های حساس به لبه مثبت تشکیل شده است . چون از لبه مثبت برای کلاک پالس استفاده شده ، شمارنده نزولی است.

مدار ریسِت نمی‌شود ، پس مُد ۱۶ است .

بنابراین یک شمارنده نزولی ، چهاربیتی ، موازی ، آسنکرون ، مُد ۱۶ است که اعداد ۱۵ تا ۰ را به ترتیب می‌شمارد .

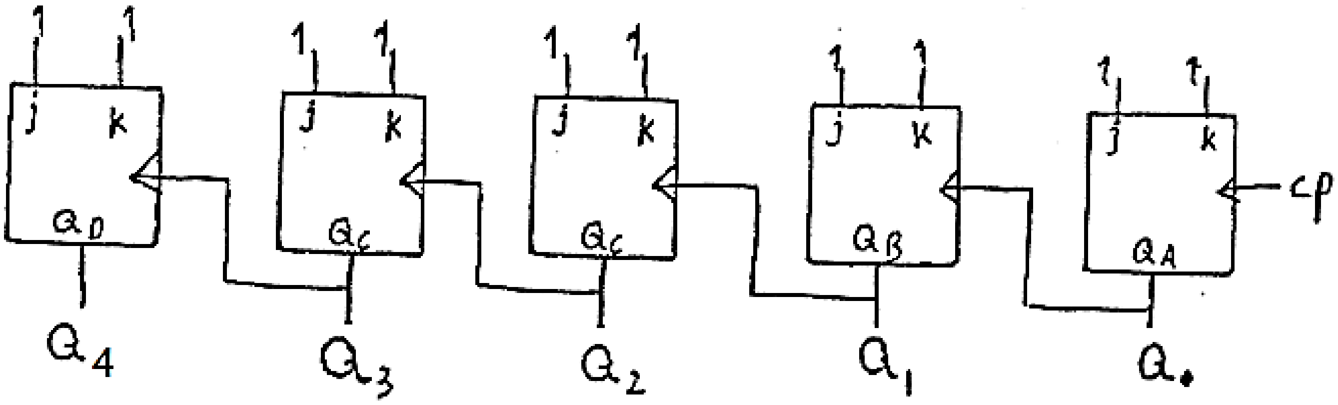
<u>د :</u> مانند قسمت ب است با این تفاوت که از D.F.F استفاده شده است . بنابراین :

یک شمارنده صعودی ، چهاربیتی ، موازی ، آسنکرون ، مُد ۱۶ است که اعداد ۰ تا ۱۵ را به ترتیب می‌شمارد.

****

۳- یک شمارنده آسنکرون نزولی مُد ۳۲ طراحی کنید.

جواب : برای شمارنده‌های نزولی از CP های با لبه مثبت استفاده می‌کنیم . و از آنجاییکه $32 = 2^5$ است ، داریم :

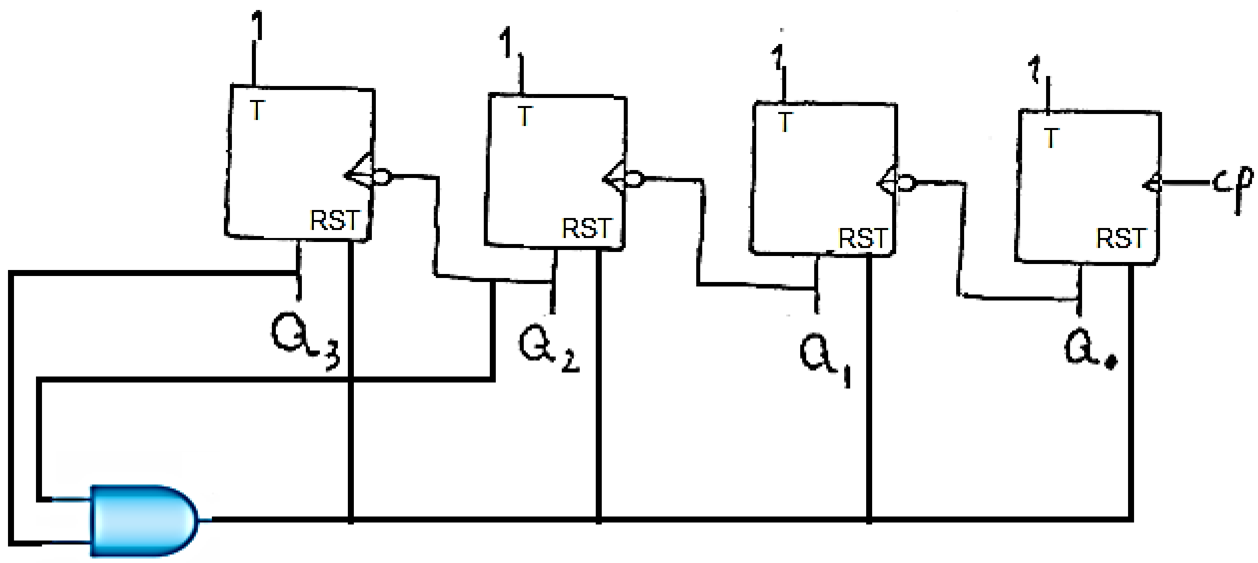


****

۴- با استفاده از T.F.F های حساس به CLR مثبت ، یک شمارنده آسنکرون صعودی مد ۱۲ طراحی کنید.

جواب :

برای شمارنده‌های صعودی از CP های با لبه منفی استفاده می‌کنیم . و از آنجاییکه $12 = 1100$ است ، از چهار فیلیپ فلاپ استفاده کرده و Q_2 , Q_3 را AND می‌کنیم و به RESET می‌دهیم . داریم :

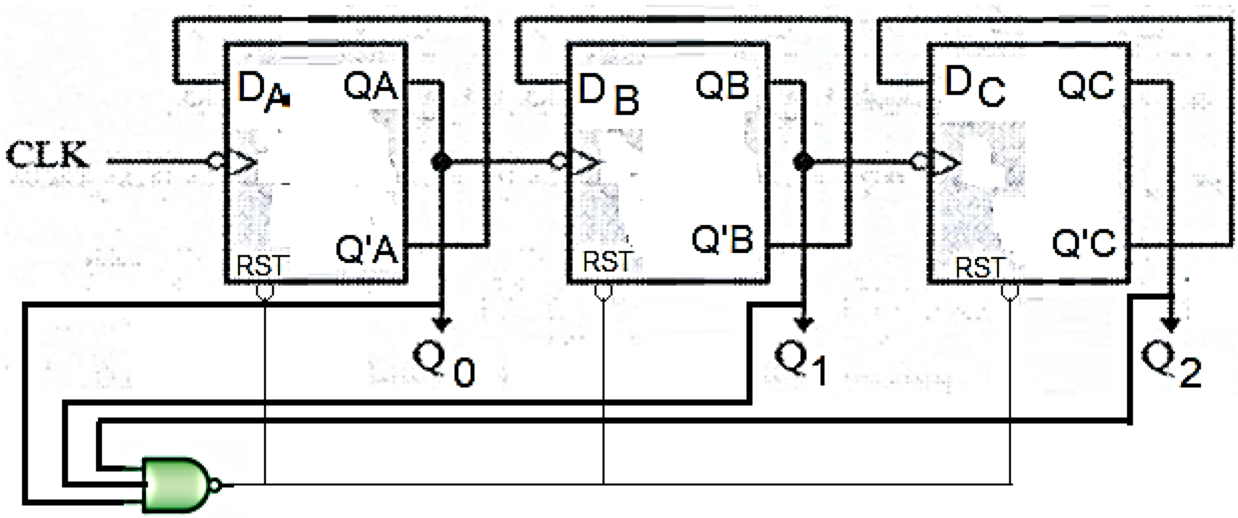


****

۵- با استفاده از D.F.F های حساس به CLR منفی ، یک شمارنده آسنکرون صعودی مد ۷ طراحی کنید.

جواب :

ابتدا باید با سه عدد D.FF ، سه عدد T.FF بسازیم . برای شمارنده‌های صعودی از CP های با لبه منفی استفاده می‌کنیم.
از آنجاییکه حساس به CLR منفی است و $7 = 111$ است ، از سه عدد فیلیپ فلاپ استفاده کرده و Q_2 , Q_1 , Q_0 را ANDN می‌کنیم و به RESET می‌دهیم . داریم :



****

فصل نهم : شمارنده‌های سنکرون (synchronou’s counters)

در این فصل ابتدا به روش تحلیل ، و سپس روش طراحی شمارنده‌های سنکرون خواهیم پرداخت .

روش تحلیل شمارنده‌های سنکرون :

برای این کار مراحل زیر را به ترتیب دنبال می‌کنیم :

۱- ورودی‌ها و خروجی‌های مدار را نام‌گذاری می‌کنیم.
۲- رابطه خروجی‌ها را می‌یابیم.
۳- از روی جدول تحریک، جدول حالت را رسم می‌کنیم.

نکته : جدول زیر را جدول تحریک می‌گویند.

فعلی	بعدی	S	R	D	J	K	T
۰	۰	۰	x	۰	۰	x	۰
۰	۱	۱	۰	۱	۱	x	۱
۱	۰	۰	۱	۰	x	۱	۱
۱	۱	x	۰	۱	x	۰	۰

جدول حالت :

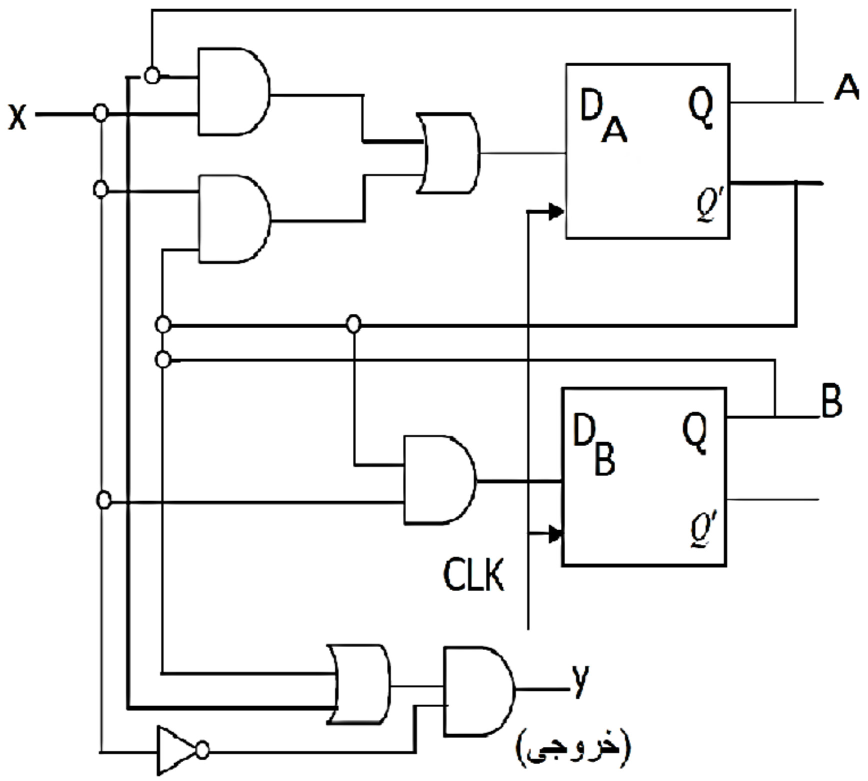
ترتیب زمانی ورودی‌ها، خروجی‌ها و وضعیت فیلیپ فلاپ‌ها را در جدولی به نام جدول حالت می‌توان بیان نمود. این جدول شامل حالت فعلی، ورودی، حالت بعدی و خروجی است.

۴- دیاگرام حالت مدار را رسم می‌کنیم.

دیاگرام حالت (State Diagram) :

نموداری است که حالت فعلی، حالت بعدی و ورودی‌های مدار را یک‌جا نمایش می‌دهد.

مثال : مدار زیر را تحلیل کرده و طرز کار آن‌را بیان کنید.



برای این کار مراحل زیر را به ترتیب دنبال می‌کنیم :

۱- ورودی‌ها و خروجی‌های مدار را نام‌گذاری می‌کنیم.

۲- رابطه خروجی‌ها را می‌یابیم.

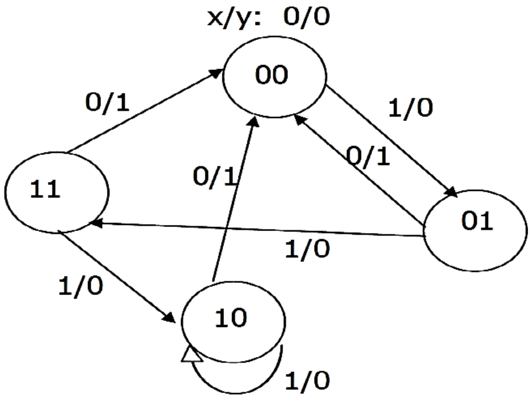
۳- جدول حالت مدار را رسم می‌کنیم.

۴- دیاگرام حالت مدار را رسم می‌کنیم.

$$A(t+1) = A(t)x(t) + B(t)x(t)$$
$$B(t+1) = A'(x)x(t)$$
$$y = (A(t) + B(t)\ )x'(t)$$

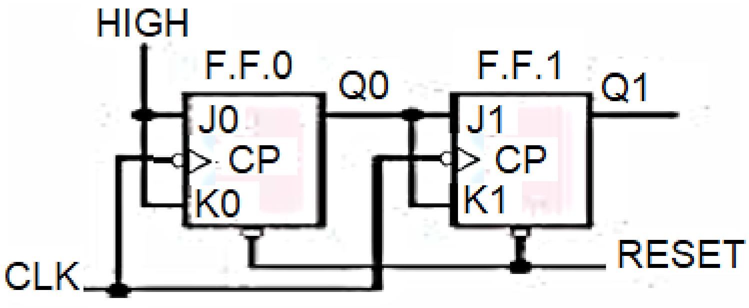
حالت فعلی		ورودی	حالت بعدی		خروجی
A	B	x	A	B	y
۰	۰	۰	۰	۰	۰
۰	۰	۱	۰	۱	۰
۰	۱	۰	۰	۰	۱
۰	۱	۱	۱	۱	۰
۱	۰	۰	۰	۰	۱
۱	۰	۱	۱	۰	۰
۱	۱	۰	۰	۰	۱
۱	۱	۱	۱	۰	۰
A	B	x	A	B	y

جدول حالت



دیاگرام حالت

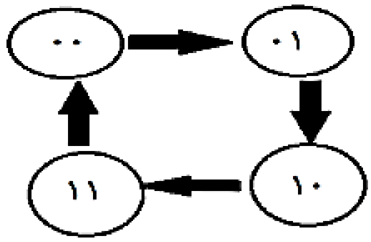
مثال: مدار زیر را تحلیل کرده، و طرز کار آنرا بیان کنید.



J0=1 K0=1 J1=Q0 K1=Q0

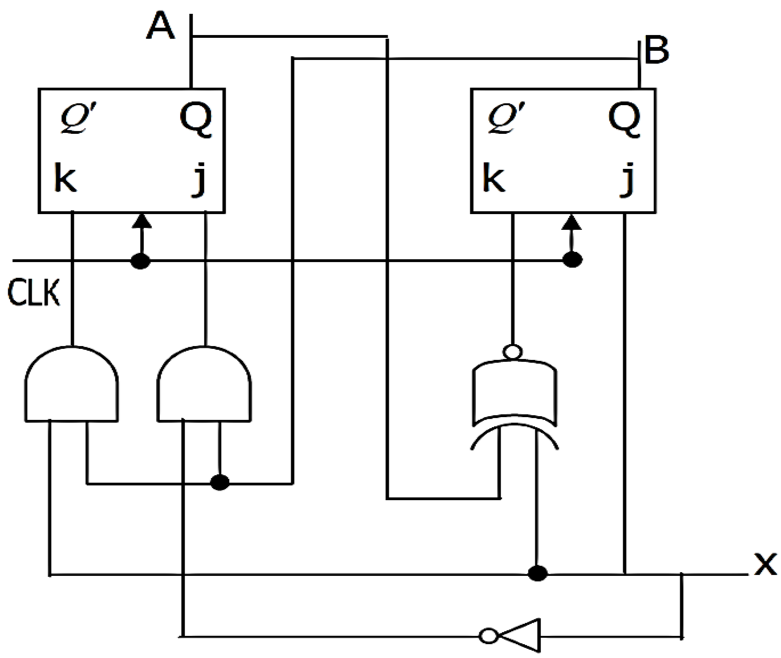
حالت فعلی Q1	Q0	J1	K1	J0	K0	خروجی Q1	Q0
0	0	0	0	1	1	0	1
0	1	1	1	1	1	1	0
1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0

جدول حالت



دیاگرام حالت

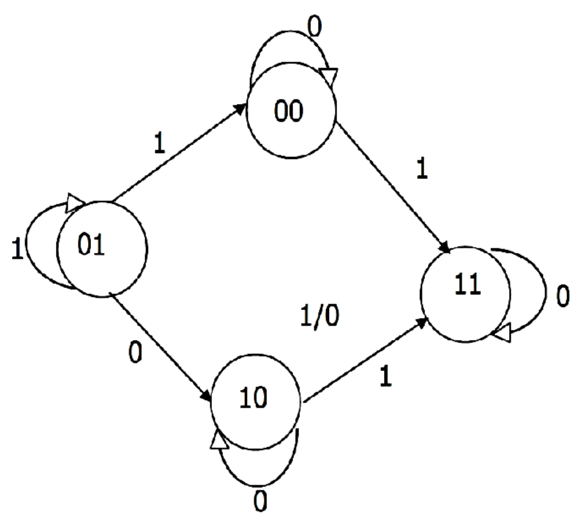
مثال : مدار زیر را تحلیل کرده ، و طرز کار آن‌را بیان کنید.



$j_A = Bx' \quad k_A = Bx \quad j_B = x \quad k_B = (A \oplus x)'$

فعلی		ورودی					بعدی	
A	B	x	j_A	k_A	j_B	k_B	A	B
۰	۰	۰	۰	۰	۰	۱	۰	۰
۰	۰	۱	۰	۰	۱	۰	۰	۱
۰	۱	۰	۱	۰	۰	۱	۱	۰
۰	۱	۱	۰	۱	۱	۰	۰	۱
۱	۰	۰	۰	۰	۰	۰	۱	۰
۱	۰	۱	۰	۰	۱	۱	۱	۱
۱	۱	۰	۱	۰	۰	۰	۱	۱
۱	۱	۱	۰	۱	۱	۱	۰	۰

جدول حالت



دیاگرام حالت

روش طراحی شمارنده‌های سنکرون :

برای این کار مراحل زیر را به ترتیب دنبال می‌کنیم :

۱- دیاگرام حالت را رسم می‌کنیم.

۲- از روی دیاگرام حالت و جدول تحریک، جدول حالت را رسم می‌کنیم.

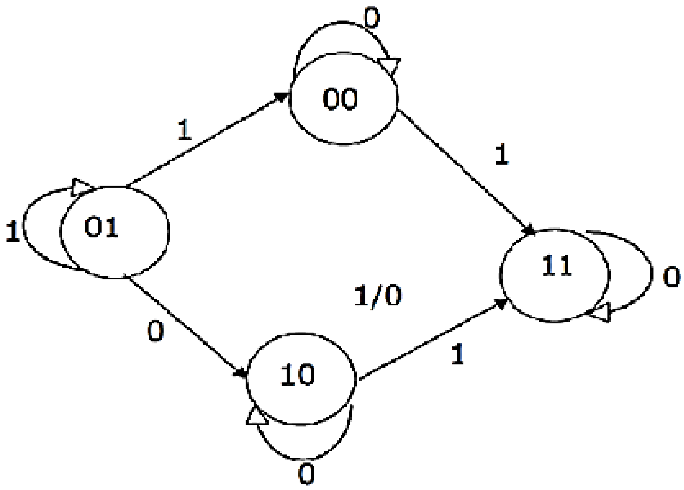
نکته : جدول زیر را جدول تحریک می‌گویند.

فعلی	بعدی	S	R	D	J	K	T
۰	۰	۰	x	۰	۰	x	۰
۰	۱	۱	۰	۱	۱	x	۱
۱	۰	۰	۱	۰	x	۱	۱
۱	۱	x	۰	۱	x	۰	۰

۳- با استفاده از نقشه کارنو ، معادله تابع ورودی فلیپ فلاپ‌ها را به دست آوریم.

۴- مدار مورد نظر را طراحی می‌کنیم.

مثال : با استفاده از JK F.F یک شمارنده سنکرون طراحی کنید که اعداد زیر را به ترتیب بشمارید.



حل :

برای این کار مراحل زیر را به ترتیب دنبال می‌کنیم :

۱- دیاگرام حالت را رسم می‌کنیم.

۲- از روی دیاگرام حالت و جدول تحریک، جدول حالت را رسم می‌کنیم.

۳- با استفاده از نقشه کارنو ، معادله تابع ورودی فلیپ فلاپ‌ها را به دست آوریم.

۴- مدار مورد نظر را طراحی می‌کنیم.

فعلی		ورودی	بعدی					
A	B	x	A	B	J_A	K_A	J_B	K_B
۰	۰	۰	۰	۰	۰	x	۰	x
۰	۰	۱	۰	۱	۰	x	۱	x
۰	۱	۰	۱	۰	۱	x	x	۱
۰	۱	۱	۰	۱	۰	x	x	۰
۱	۰	۰	۱	۰	x	۰	۰	x
۱	۰	۱	۱	۱	x	۰	1	x
۱	۱	۰	۱	۱	x	۰	x	۰
۱	۱	۱	۰	۰	x	۱	x	۱

A \ BX

۰	۰	۰	۱
x	x	x	x

$j_A = Bx'$

A \ Bx

x	x	x	x
۰	۰	۱	۰

$k_A = Bx$

A \ BX

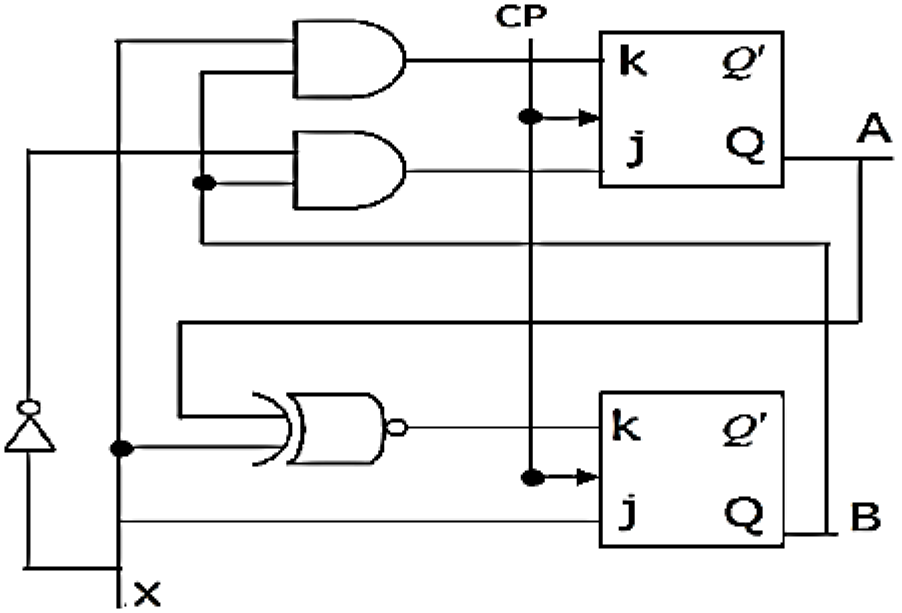
۰	۱	x	x
۰	۱	x	x

$j_B = x$

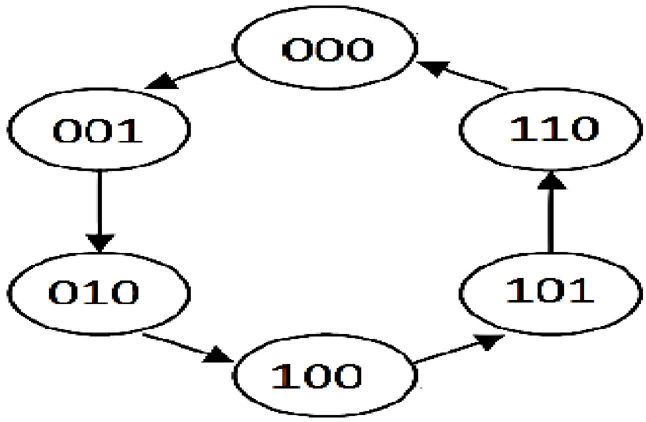
A \ Bx

x	x	۰	۱
x	x	۱	۰

$k_B = (A \odot x)$



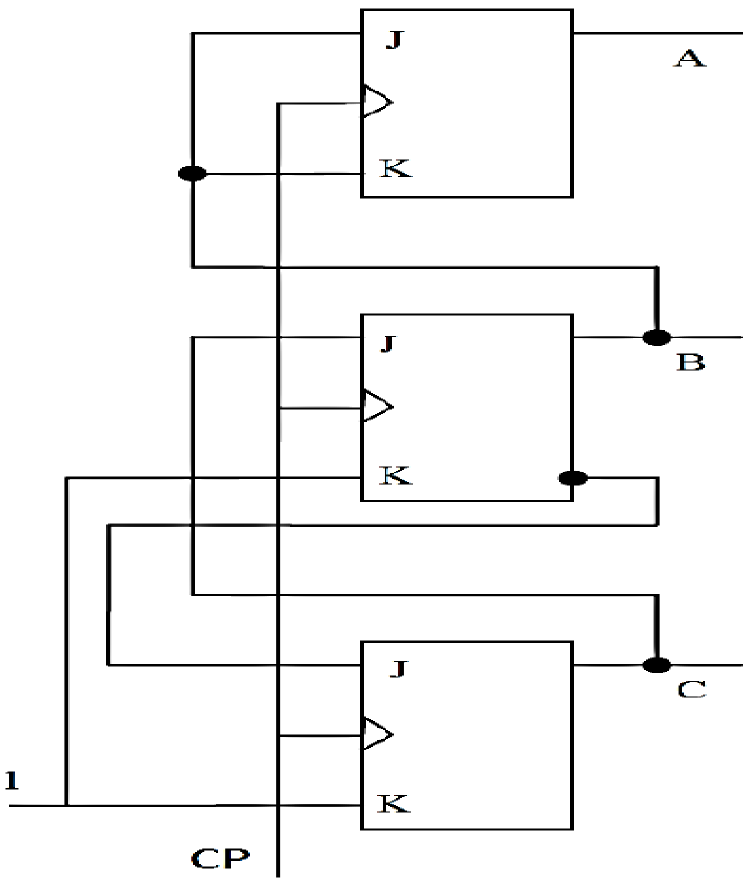
مثال : با استفاده از JK F.F یک شمارنده سنکرون طراحی کنید که اعداد زیر را ترتیب بشمارد.



حالت فعلی			حالت بعدی			ورودی‌های فلیپ‌فلاپ‌ها					
A	B	C	A	B	C	JA	KA	JB	KB	JC	KC
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	1	0	0	1	X	X	1	0	X
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	0	0	0	X	1	X	1	0	X

$JA = B \quad KA = B \quad JB = C \quad KB = 1$

$JC = \overline{B} \quad KC = 1$



مثال : با استفاده از T F.F یک شمارنده سنکرون سه بیتی را طوری طراحی کنید که یک عدد سه بیتی از ورودی بگیرد و حاصل جمع ارقام را در خروجی خود نشان دهد.

فعلی			بعدی					
A	B	C	A	B	C	TA	TB	TC
0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0
0	1	0	0	0	1	0	1	1
0	1	1	0	1	0	0	0	1
1	0	0	0	0	1	1	0	1
1	0	1	0	1	0	1	1	1
1	1	0	0	1	0	1	0	0
1	1	1	0	1	1	1	0	0

A \ BC	00	01	11	10
0	٠	٠	٠	٠
1	١	١	١	١

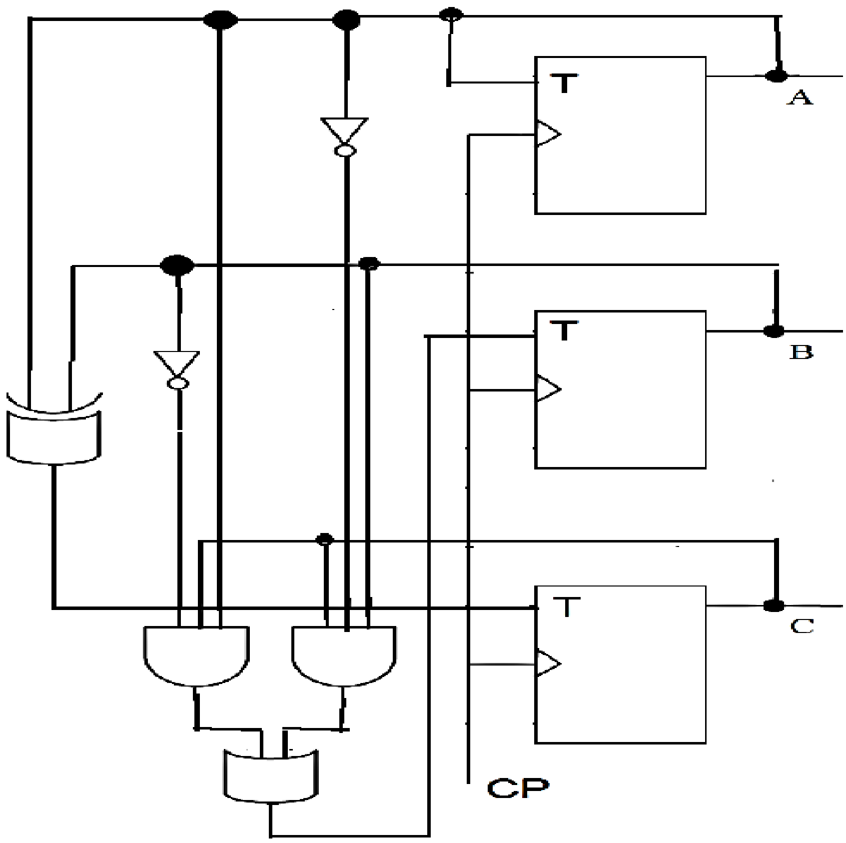
TA = A

A \ BC	00	01	11	10
0	٠	٠	٠	١
1	٠	١	٠	٠

TB = A' B C + A B' C

A \ BC	00	01	11	10
0	٠	٠	١	١
1	١	١	٠	٠

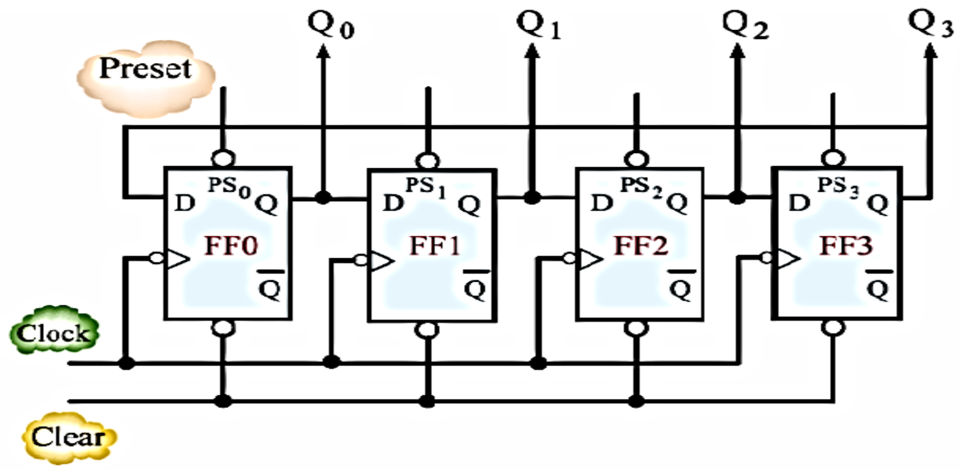
$TC = AB' + A'B = A \oplus B$



شمارنده های خاص :

شمارنده حلقوی یا دایره‌ای (Ring Counter) :

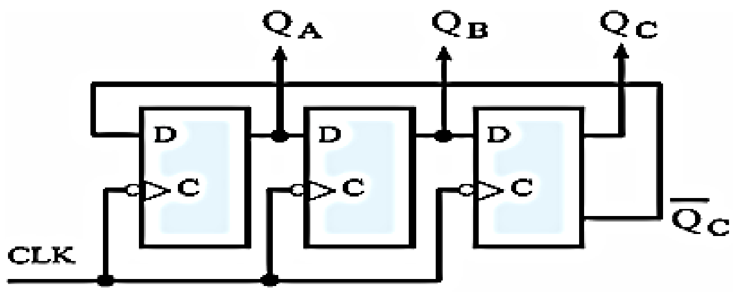
شمارنده حلقوی از ترکیب فلیپ فلاپ‌های نوع D به گونه‌ای شکل می گیرد ، که خروجی Q آخرین فلیپ فلاپ ، به ورودی D اولین فلیپ فلاپ فیدبک شده باشد . شکل زیر شمارنده حلقوی ۴ بیتی را نشان می‌دهد.



تعداد پالس‌های ساعت ورودی	خروجی‌ها Q_3	Q_2	Q_1	Q_0	شمارش ده‌دهی خروجی $N=(Q_3Q_2Q_1Q_0)$
0	0	0	0	1	1
1	0	0	1	0	2
2	0	1	0	0	4
3	1	0	0	0	8
4	0	0	0	1	1

شمارنده جانسون (Johnson Counter) :

این شمارنده یک شیفت رجیستر با ورودی سری و خروجی سری است ؛ که در آن ′Q آخرین فلیپ فلاپ به ورودی (D) اولین فلیپ فلاپ متصل شده است. در شکل زیر شمارنده جانسون ۳ بیتی را مشاهده می‌کنید.

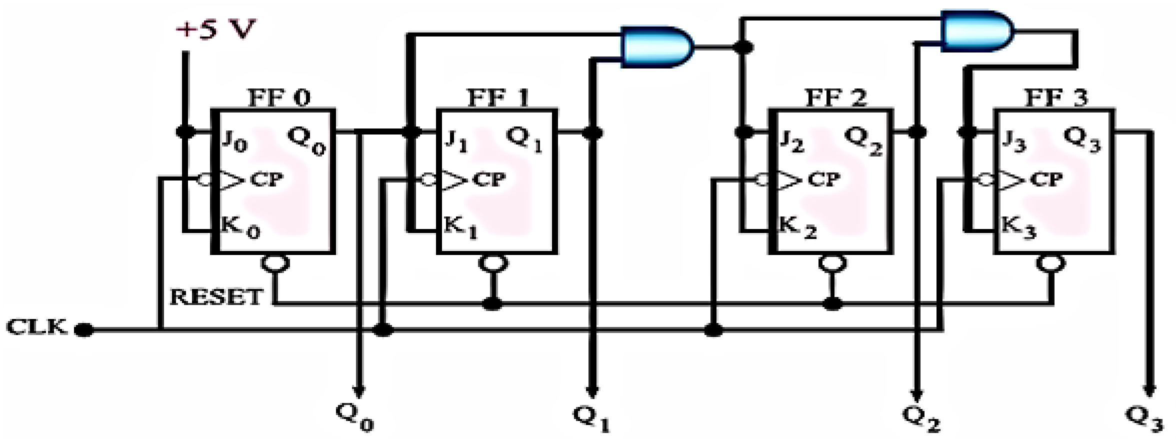


پالس ساعت	Q_A	Q_B	Q_C
0	0	0	0
1	1	0	0
2	1	1	0
3	1	1	1
4	0	1	1
5	0	0	1

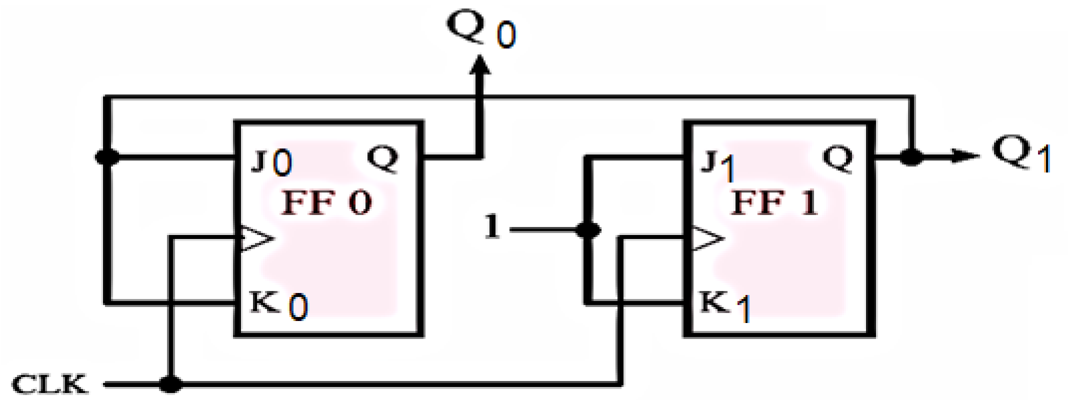
تمرین :

۱- هر یک از مدارهای زیر را به طور کامل تشریح کرده ، و نوع شمارنده را تعیین کنید.

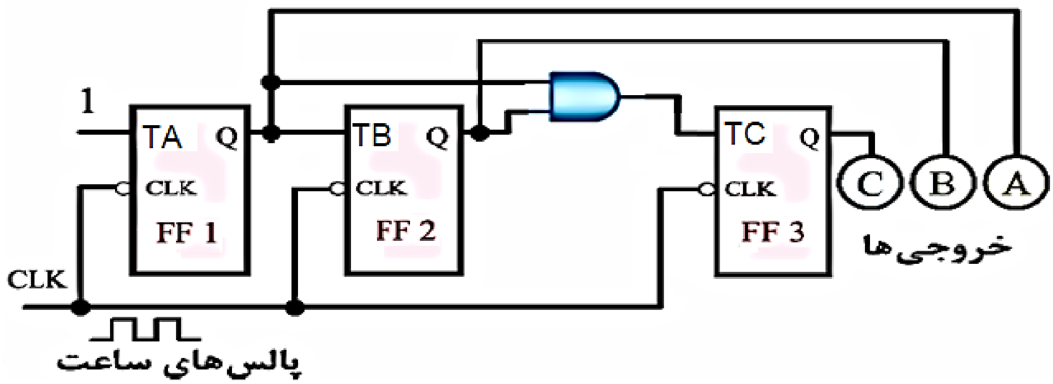
الف :



ب :



ج :



جواب :

<u>الف :</u> CP تمام طبقات یکی است ، پس یک شمارنده سنکرون است . پس برای تحلیل آن مراحل زیر را به ترتیب دنبال می‌کنیم : ۱- ورودی‌ها و خروجی‌های مدار را نام‌گذاری می‌کنیم. ۲ - رابطه خروجی‌ها را می‌یابیم.

۳- از روی جدول تحریک، جدول حالت را رسم می‌کنیم.

نکته : جدول تحریک JK.F.F عبارت است از :

فعلی	J	K	بعدی
۰	۰	x	۰
۰	۱	x	۱
۱	x	۱	۰
۱	x	۰	۱

۴- دیاگرام حالت مدار را رسم می‌کنیم.

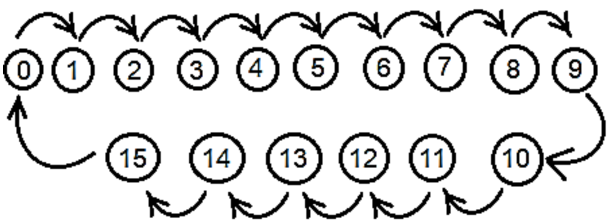
ورودی فیلیپ فلاپ‌ها عبارتند از :

$$J_0 = K_0 = 1 \quad , \quad J_1 = K_1 = Q_0 \quad , \quad J_2 = K_2 = Q_0Q_1 \quad , \quad J_3 = K_3 = Q_0Q_1Q_2$$

جدول حالت :

	حالت فعلی				ورودی فیلیپ فلاپ‌ها								حالت بعدی				
	Q_3	Q_2	Q_1	Q_0	J_3	K_3	J_2	K_2	J_1	K_1	J_0	K_0	Q_3	Q_2	Q_1	Q_0	
0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1
1	0	0	0	1	0	0	0	0	1	1	1	1	0	0	1	0	2
2	0	0	1	0	0	0	0	0	0	0	1	1	0	0	1	1	3
3	0	0	1	1	0	0	1	1	1	1	1	1	0	1	0	0	4
4	0	1	0	0	0	0	0	0	0	0	1	1	0	1	0	1	5
5	0	1	0	1	0	0	0	0	1	1	1	1	0	1	1	0	6
6	0	1	1	0	0	0	0	0	0	0	1	1	0	1	1	1	7
7	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	8
8	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	9
9	1	0	0	1	0	0	0	0	1	1	1	1	1	0	1	0	10
10	1	0	1	0	0	0	0	0	0	0	1	1	1	0	1	1	11
11	1	0	1	1	0	0	1	1	1	1	1	1	1	1	0	0	12
12	1	1	0	0	0	0	0	0	0	0	1	1	1	1	0	1	13
13	1	1	0	1	0	0	0	0	1	1	1	1	1	1	1	0	14
14	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	15
15	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0

دیاگرام حالت :



پس مدار فوق یک شمارنده چهار بیتی ، سنکرون ، صعودی ، مُد ۱۶ است ، و اعداد 0 تا 15 را به ترتیب می‌شمارد .

ب : CP تمام طبقات یکی است ، پس یک شمارنده سنکرون است . پس برای تحلیل آن مراحل زیر را به ترتیب دنبال می‌کنیم :   ۱- ورودی‌ها و خروجی‌های مدار را نام‌گذاری می‌کنیم.   ۲ - رابطه خروجی‌ها را می‌یابیم.

۳- از روی جدول تحریک، جدول حالت را رسم می‌کنیم.

نکته : جدول تحریک JK.F.F عبارت است از :

فعلی	J	K	بعدی
۰	۰	x	۰
۰	۱	x	۱
۱	x	۱	۰
۱	x	۰	۱

۴- دیاگرام حالت مدار را رسم می‌کنیم.

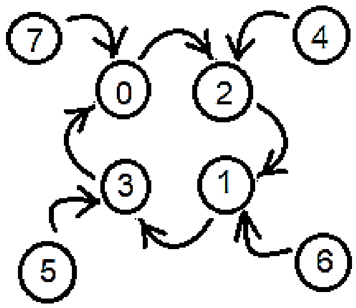
ورودی فیلیپ فلاپ‌ها عبارتند از :

$$J_0 = K_0 = Q_1 \qquad , \qquad J_1 = K_1 = 1$$

جدول حالت :

	حالت فعلی		ورودی فیلیپ فلاپ‌ها				حالت بعدی		
	Q_1	Q_0	J_1	K_1	J_0	K_0	Q_1	Q_0	
0	0	0	1	1	0	0	1	0	2
1	0	1	1	1	0	0	1	1	3
2	1	0	1	1	1	1	0	1	1
3	1	1	1	1	1	1	0	0	0
4	0	0	1	1	0	0	1	0	2
5	0	1	1	1	0	0	1	1	3
6	1	0	1	1	1	1	0	1	1
7	1	1	1	1	1	1	0	0	0

دیاگرام حالت :



پس مدار فوق یک شمارنده دو بیتی ، سنکرون ، نامنظم است ، و اعداد بالا را می‌شمارد .

ج : CP تمام طبقات یکی است ، پس یک شمارنده سنکرون است . پس برای تحلیل آن مراحل زیر را به ترتیب دنبال می‌کنیم :   ۱- ورودی‌ها و خروجی‌های مدار را نام‌گذاری می‌کنیم.   ۲ - رابطه خروجی‌ها را می‌یابیم.

۳- از روی جدول تحریک، جدول حالت را رسم می‌کنیم.

نکته : جدول تحریک T.F.F عبارت است از :

فعلی	T	بعدی
۰	۰	۰
۰	۱	۱
۱	۱	۰
۱	۰	۱

۴- دیاگرام حالت مدار را رسم می‌کنیم.

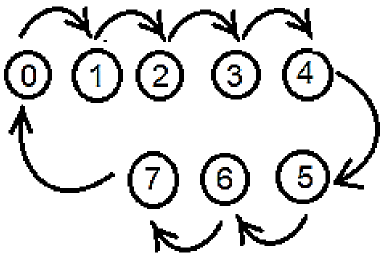
ورودی فیلیپ فلاپ‌ها عبارتند از :

$$T_A = 1 \quad , \quad T_B = Q_A \quad , \quad T_C = Q_A Q_B$$

جدول حالت :

	حالت فعلی			ورودی f.f ها			حالت بعدی			
	Q_C	Q_B	Q_A	T_C	T_B	T_A	Q_C	Q_B	Q_A	
0	0	0	0	0	0	1	0	0	1	1
1	0	0	1	0	1	1	0	1	0	2
2	0	1	0	0	0	1	0	1	1	3
3	0	1	1	1	1	1	1	0	0	4
4	1	0	0	0	0	1	1	0	1	5
5	1	0	1	0	1	1	1	1	0	6
6	1	1	0	0	0	1	1	1	1	7
7	1	1	1	1	1	1	0	0	0	0

دیاگرام حالت :

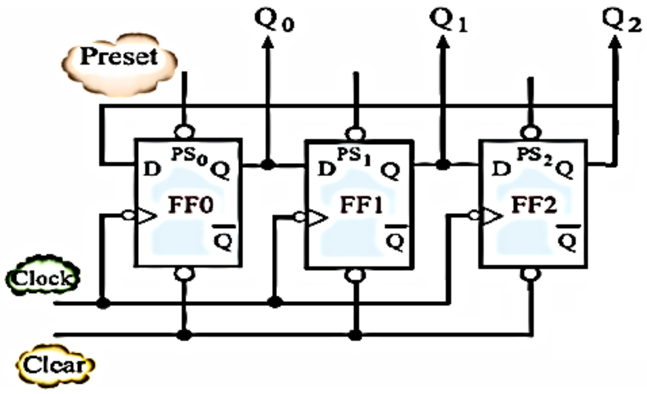


پس مدار فوق یک شمارنده سه بیتی ، سنکرون ، صعودی ، مُد ۸ است ، و اعداد 0 تا ۷ را به ترتیب می‌شمارد .

****

۲- یک مدار شمارنده حلقوی سه بیتی را رسم کرده، و جدول صحت آن‌را بکشید.

جواب :

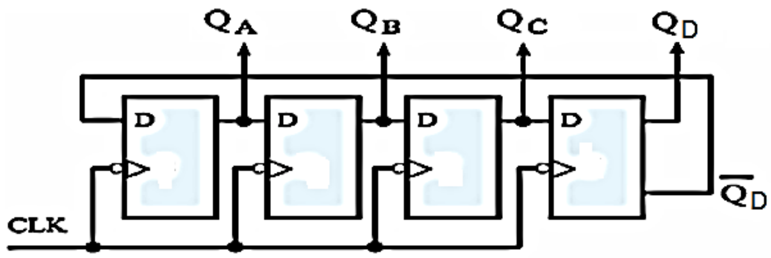


تعداد پالس‌های ساعت ورودی	خروجی‌ها Q_2	Q_1	Q_0	شمارش ده‌دهی خروجی $N=(Q_3Q_2Q_1Q_0)$
0	0	0	1	1
1	0	1	0	2
2	1	0	0	4
3	0	0	1	1

****

۳- یک مدار شمارنده جانسون سه بیتی را رسم کرده، و جدول صحت آنرا بکشید.

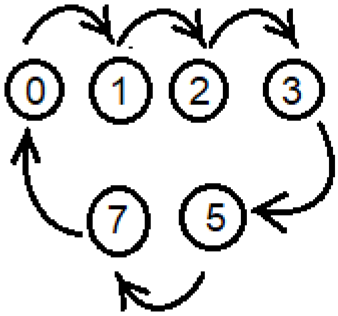
جواب :



پالس ساعت	Q_A	Q_B	Q_C	Q_D
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0

****

۴- شمارنده‌ای با استفاده از فلیپ فلاپ T طراحی نمایید که اعداد زیر را به ترتیب بشمارد.



جواب :

برای این کار مراحل زیر را به ترتیب دنبال می‌کنیم :

۱- دیاگرام حالت را رسم می‌کنیم.

۲- از روی دیاگرام حالت و جدول تحریک، جدول حالت را رسم می‌کنیم.

نکته : جدول تحریک T.F.F عبارت است از :

فعلی	T	بعدی
۰	۰	۰
۰	۱	۱
۱	۱	۰
۱	۰	۱

۳- با استفاده از نقشه کارنو ، معادله تابع ورودی فلیپ فلاپ‌ها را به دست آوریم.

۴- مدار مورد نظر را طراحی می‌کنیم.

جدول حالت :

	حالت فعلی			حالت بعدی				ورودی f.f ها		
	Q_C	Q_B	Q_A	Q_C	Q_B	Q_A		T_C	T_B	T_A
0	0	0	0	0	0	1	1	0	0	1
1	0	0	1	0	1	0	2	0	1	1
2	0	1	0	0	1	1	3	0	0	1
3	0	1	1	1	0	1	5	1	1	0
4	1	0	0	X	X	X	X	X	X	X
5	1	0	1	1	1	1	7	0	1	0
6	1	1	0	X	X	X	X	X	X	X
7	1	1	1	0	0	0	0	1	1	1

حال ورودی‌ها را طراحی می‌کنیم .

TA :

QC \ QB QA	00	01	11	10
0	1	1	0	1
1	X	0	1	X

$$TA = Q'A + QC.QB + Q'C.Q'B$$

TB :

QC \ QB QA	00	01	11	10
0	0	1	1	0
1	X	1	1	X

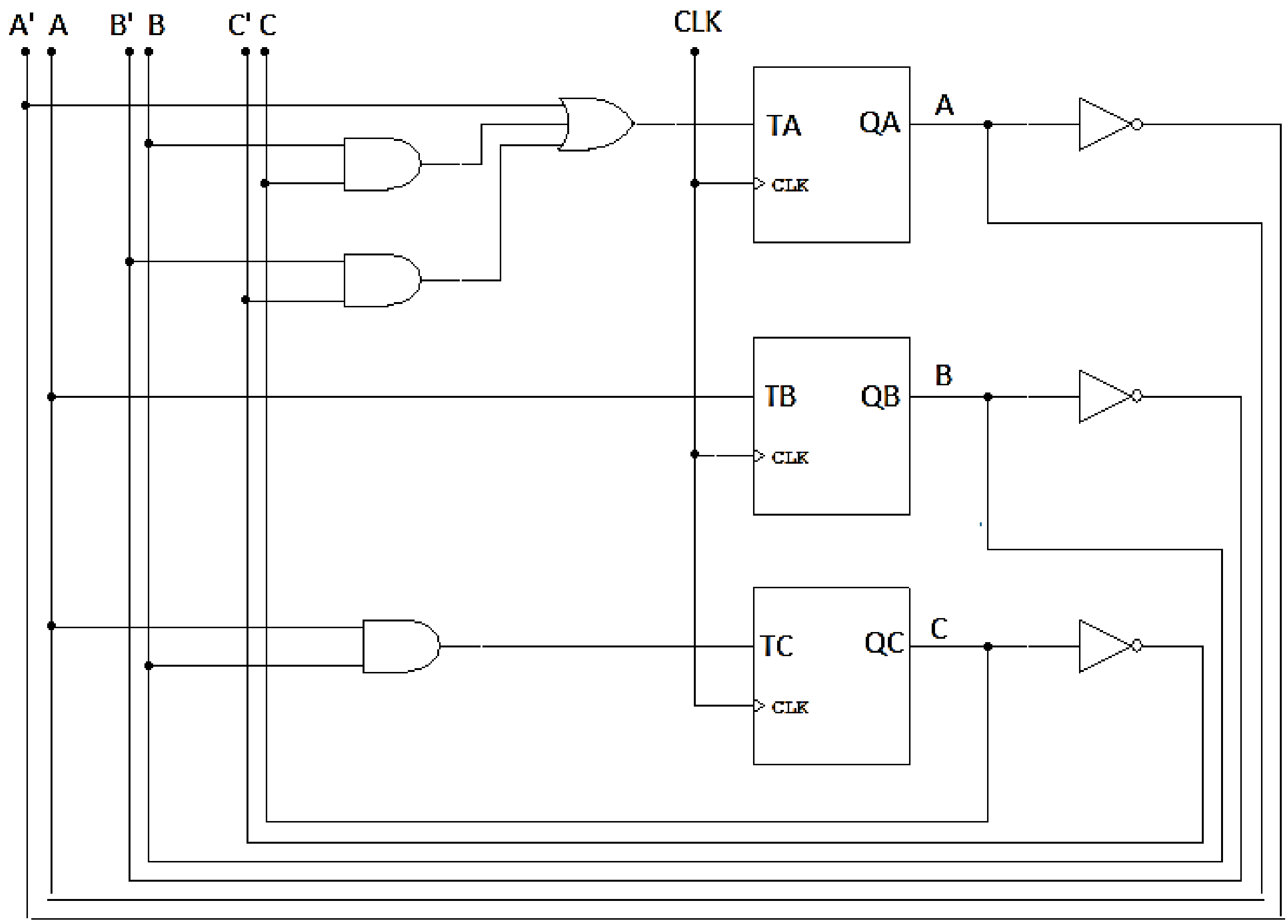
$$TB = QA$$

TC :

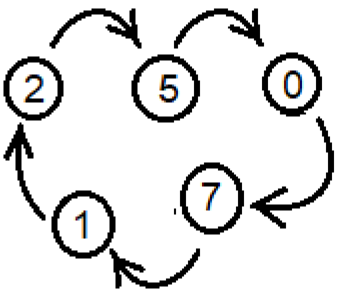
QC \ QB QA	00	01	11	10
0	0	0	1	0
1	X	0	1	X

$$TC = QB.QA$$

و در پایان مدار را طراحی می‌کنیم .



۵- با استفاده از T F.F یک شمارنده سنکرون را طوری طراحی کنید که اعداد زیر را به ترتیب بشمارد.



جواب :

برای این کار مراحل زیر را به ترتیب دنبال می‌کنیم :

۱- دیاگرام حالت را رسم می‌کنیم.

۲- از روی دیاگرام حالت و جدول تحریک، جدول حالت را رسم می‌کنیم.

نکته : جدول تحریک T.F.F عبارت است از :

فعلی	T	بعدی
۰	۰	۰
۰	۱	۱
۱	۱	۰
۱	۰	۱

۳- با استفاده از نقشه کارنو ، معادله تابع ورودی فلیپ فلاپ‌ها را به دست آوریم.

۴- مدار مورد نظر را طراحی می‌کنیم.

جدول حالت :

	حالت فعلی			حالت بعدی				ورودی f.f ها		
	Q_C	Q_B	Q_A	Q_C	Q_B	Q_A		T_C	T_B	T_A
0	0	0	0	1	1	1	7	1	1	1
1	0	0	1	0	1	0	2	0	1	1
2	0	1	0	1	0	1	5	1	1	1
3	0	1	1	X	X	X	X	X	X	X
4	1	0	0	X	X	X	X	X	X	X
5	1	0	1	0	0	0	0	1	0	1
6	1	1	0	X	X	X	X	X	X	X
7	1	1	1	0	0	1	1	1	1	0

حال ورودی‌ها را طراحی می‌کنیم .

TA :

QC \ QB QA	00	01	11	10
0	1	1	X	1
1	X	1	0	X

$TA = Q'A + Q'B$

TB :

QC \ QB QA	00	01	11	10
0	1	1	X	1
1	X	0	1	X

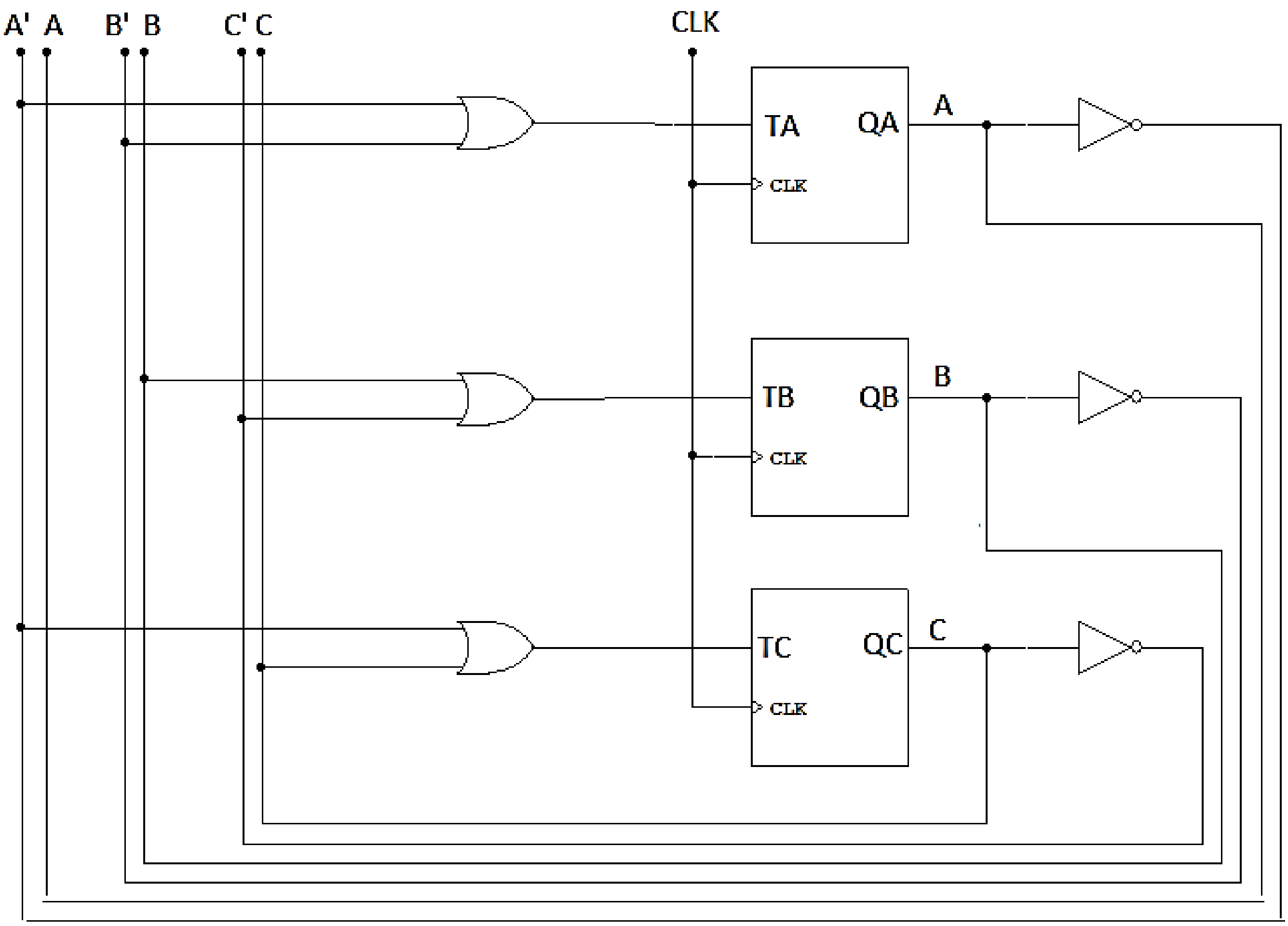
$TB = Q'C + QB$

TC :

QC \ QB QA	00	01	11	10
0	1	0	X	1
1	X	1	1	X

$TC = Q'A + QC$

و در پایان مدار را طراحی می‌کنیم .



۶- با استفاده از D F.F یک شمارنده سنکرون طراحی کنید که اعدد ۰ تا ۹ را به ترتیب بشمارد.

جواب :

برای این کار مراحل زیر را به ترتیب دنبال می‌کنیم :

۱- دیاگرام حالت را رسم می‌کنیم.

۲- از روی دیاگرام حالت و جدول تحریک، جدول حالت را رسم می‌کنیم.

نکته : جدول تحریک T.F.F عبارت است از :

فعلی	D	بعدی
۰	۰	۰
۰	۱	۱
۱	۰	۰
۱	۱	۱

۳- با استفاده از نقشه کارنو ، معادله تابع ورودی فلیپ فلاپ‌ها را به دست آوریم.

۴- مدار مورد نظر را طراحی می‌کنیم.

جدول حالت :

	حالت فعلی				حالت بعدی					ورودی فیلیپ فلاپ‌ها			
	Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0		D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0	1	1	0	0	0	1
1	0	0	0	1	0	0	1	0	2	0	0	1	0
2	0	0	1	0	0	0	1	1	3	0	0	1	1
3	0	0	1	1	0	1	0	0	4	0	1	0	0
4	0	1	0	0	0	1	0	1	5	0	1	0	1
5	0	1	0	1	0	1	1	0	6	0	1	1	0
6	0	1	1	0	0	1	1	1	7	0	1	1	1
7	0	1	1	1	1	0	0	0	8	1	0	0	0
8	1	0	0	0	1	0	0	1	9	1	0	0	1
9	1	0	0	1	0	0	0	0	0	0	0	0	0
10	X	X	X	X	X	X	X	X	X	X	X	X	X
11	X	X	X	X	X	X	X	X	X	X	X	X	X
12	X	X	X	X	X	X	X	X	X	X	X	X	X
13	X	X	X	X	X	X	X	X	X	X	X	X	X
14	X	X	X	X	X	X	X	X	X	X	X	X	X
15	X	X	X	X	X	X	X	X	X	X	X	X	X

حال ورودی‌ها را طراحی می‌کنیم .

D0 :

Q3Q2 \ Q1Q0	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	X	X	X	X
10	1	0	X	X

D0 = Q'1.Q'0 + Q'3.Q'0

D1 :

Q3Q2 \ Q1Q0	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	X	X	X	X
10	0	0	X	X

D1 = Q1.Q'0 + Q'3.Q'1.Q0

D2 :

Q3Q2 \ Q1Q0	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	X	X	X	X
10	0	0	X	X

D2 = Q2.Q'0 + Q2.Q'1 + Q'2.Q1.Q0

D3 :

Q3Q2 \ Q1Q0	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	X	X	X	X
10	1	0	X	X

D3 = Q3.Q'0 + Q2.Q1.Q0

و در پایان مدار را طراحی می‌کنیم .