باسمه تعالی

جزوه درس مدار منطقی

مهندس رضا سعیدی نیا

دانشجوی دکترا معماری سیستم های کامپیوتر دانشگاه تهران

برای مشاهده ویدیوهای ضبط شده درس مدار منطقی با حل 15 سال تست های کارشناسی ارشد به سامانه زیر مراجعه کنید:

www.idars.ir

بخش دانشگاهی- فنی مهندسی- مهندسی کامپیوتر

http://www.saeedinia.ir

saeidi_reza@yahoo.com

فصل1(مبانی کامپیوتر)

تبدیل اعداد از مبنای 10 به مبنای R : قسمت صحیح را بر تقسیم‌های متوالی بر R به دست می‌آوریم تا زمانیکه خارج قسمت قابل تقسیم بر R نباشد و از آخرین خارج قسمت به چپ (باقی‌مانده‌ها) را می‌نویسیم قسمت اعشار را با ضرب‌های متوالی در R به دست می‌آوریم تا اعشار صفر شود یا به دقت مورد نظر برسد.

هررقم در مبنای R0≤ ≤R-1

تبدیل اعداد از مبنای R به مبنای 10 : قسمت صحیح را با ضرب در توان‌های R (0 تا n) و قسمت اعشار را با ضرب در توان‌های منفی R به دست می‌آوریم و حاصل جمع را محاسبه می‌کنیم:

$$N=(\underline{a_n a_{n-1}\ldots\ldots a_1 a_0}.\underline{a_{-1}a_{-2}\ldots\ldots a_{-m}})=a_n r^n+a_{n-1}r^{n-1}+\ldots+a_0 r^0+a_{-1}r^{-1}+\ldots..+a_{-m}r^{-m}=(\Sigma a_i r^i)_{10}$$

مثال:اعداد زیر را به مبنای 10 تبدیل کنید:

$(257.2)_8=2*8^2+5*8^1+7*8^0+2*8^{-1}=175.25$

$(1FA/C)_{16}=1*16^2+F*16^1+A*16^0+C*16^{-1}=506.75$

$(1011001.101)=1*2^6+1*2^4+1*2^3+1*2^0+1*2^{-1}+1*2^{-3}=89.625$

کدام یک از اعداد زیر اگر به مبنای 4 تبدیل شود خاتمه پذیر است؟

0/875(4 0/775(3 0/15(2 0/1(1

پاسخ:یعنی بعد از چند بار ضرب کردن قسمت اعشار صفر شود.جواب گزینه 4 می باشد.

تبدیل مبنای 2به 16(8 یا 4):قسمت صحیح را از ممیز به چپ و اعشار را از ممیز به راست 4 (3یا 2) رقم جدا می‌کنیم و به جای آن از جدول تبدیل مبناها که در ادامه می آید مقدار مربوطه را می‌نویسیم:

$(0010,1101.1101)_2=(2D.D)_{16}$

$(101,101,110.100)_2=(556.4)_8$

$(10,11,01.11,01)_2=(231.31)_4$

در انجام تبدیلات اگر رقم (بیت) کم باشد به سمت چپ قسمت صحیح یا سمت راست قسمت اعشار به تعداد کافی صفر اضافه می کنیم.

$(11011011)_2=(?)_8=(011,011,011)_2=(333)_8$

$(1101100.101)_2=(?)_{16}=(0110,1100.1010)_2=(6C.A)_{16}$

تبدیل مبنای 16 (8یا)4 به 2 :به جای هر رقم به ترتیب 4و3و2 رقم معادل آن‌را ازجدول تبدیل مبناها می‌نویسیم.

جدول تبدیل مبناها:

مبنای 10	مبنای 2 8421	مبنای 8 81	مبنای 16
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

مکمل پایه و مکمل پایه کاهش یافته: در هر مبنای r به مکمل r مکمل پایه و به مکمل r-1 مکمل پایه کاهش یافته می‌گوییم.

طریقه‌ی محاسبه‌ی مکمل r-1 :به تعداد رقم‌های عدد r-1 قرار می‌دهیم و عدد را از آن کم می‌کنیم.

مکمل 9 عدد $_{10}$(256) چیست؟

$$\begin{array}{r} 999 \\ -\ 256 \\ \hline 743 \end{array}$$

مکمل 1 عدد $_{2}$(110101) چیست؟

$$\begin{array}{r} 111111 \\ -110101 \\ \hline 001010 \end{array}$$

در مبنای r :مکمل r=1+(مکمل r-1)

مکمل 10 عدد $_{10}$(256) چیست؟

743+1=744

در حالت کلی مکمل 10 عدد n رقمی 10^n-N=N می باشد.

در حالت کلی مکمل r عدد n رقمی r^n-N=N می باشد.

جمع و تفریق در پایه r:

A+B

$(A-B)_r=A_r+(B\ \text{عدد}\ r\ \text{مکمل})=A_r+(r^n-B)$

اگر A≥B باشد حتماً حاصل نقلی دارد که آن را نادیده می‌گیریم .

اگر A<B خروجی نقلی ندارد و پاسخ مکمل rحاصل تفریق است. برای بدست آوردن عدد معادل مکمل r آن را حساب می‌کنیم و جلوی آن یک منفی قرار می‌دهیم.

مثال تفریق‌های زیر را با مکمل 10 بدست آورید:

(456-321)=456+679=135

(321-456)=321+544=865=-135

مثال: تفریق‌های زیر را به روش مکمل 2 حل کنید (بدون علامت)

$(10100)_2-(01111)_2=10100+10001=100101\rightarrow$ 5

$(01111)_2-(10100)_2=01111+01100=11011\rightarrow$ -5

در اعداد بدون علامت هنگام گسترش سمت چپ صفر قرار می‌دهیم در اعداد علامت‌دار هنگام گسترش سمت چپ علامت قبلی را گسترش می‌دهیم:

111→11111

0111→ 00111

در اعداد بدون علامت صحیح در n بیت از 0 تا 2^n-1 ذخیره می شود.

روش‌های ذخیره‌سازی اعداد صحیح (منفی و مثبت):

1- روش مقدار علامت (sign-magnitude): در این روش سمت چپ‌ترین بیت (MSB=Most significant bit) را به عنوان بیت علامت در نظر می‌گیریم و برای اعداد مثبت 0 و برای اعدا منفی یک قرار می‌دهیم و بقیه بیت ها را برای ذخیره کردن مقدار استفاده می‌کنیم.

مثلا در 8 بیت 12- ,12+ بصورت 00001100 و 10001100 ذخیره می شوند. در این روش در n بیت از $-(2^{n-1}-1)$ تا $+(2^{n-1}-1)$ ذخیره می شود.

دو ایراد :

1-برای عدد صفر دو نمایش داریم: مثلا در 8 بیت 00000000 نشان دهنده 0+ و 10000000 نشان دهنده 0- می باشد.

2- جمع دو عدد مختلف العلامه درست بدست نمی‌آید. یعنی A+(-A)=-2A که برای رفع ایراد آن را بصورت A-(+A) می-نویسیم یعنی به جای جمع تفریق می کنیم.

2- روش متمم یک: در این روش عدد مثبت را به شیوه ی معمولی می نویسیم و عدد منفی متمم 1 عدد مثبت می باشد. برای بدست آوردن متمم یک 0 ها را به یک و یک ها را به صفر تبدیل می کنیم. یعنی 1↔0

در این روش عدد 00001100=12+ و 11110011=12- می باشند.

در این روش نیز برای عدد صفر دو نمایش داریم. 0-=11111111 ,0+=00000000.

در این روش نیز در n بیت از $(1-2^{n-1})$-تا $(1-2^{n-1})$+ذخیره می شود.

3- روش متمم 2 (مکمل 2): عدد مثبت را به شیوه معمولی می نویسیم و عدد منفی مکمل 2 عدد مثبت است. برای بدست آوردن متمم 2 دو روش داریم روش اول: متمم 2= متمم1+1

روش دوم: عدد را از راست به چپ پیمایش می‌کنیم و تا اولین یک را می‌نویسیم از آن به بعد را not می‌کنیم. در این روش 00001100=12+ و 11110100=12- می‌باشد. این روش هیچ ایرادی ندارد. محدوده عدد در n بیت از $(1-2^{n-1})$+تا - (2^{n-1}) می باشد.

کدهای دودویی : روشی برای نمایش اعداد و حروف در متمم می باشد هر کد به منظور خاص و اهداف خاصی طراحی می شود و باید نیازهای مدنظر را برآورده کند .

کد BCD(binary coded decimal): کد ده دهی تبدیل شده به باینری هدف نمایش روش قلم و کاغذ باشد در کد دهدهی رقم ها بین صفر تا 9 می باشند بنابراین در کد BCD نیز رقم ها بین صفر تا 9 می باشد به ازای هر رقم 4 بیت از جدول مقابل استفاده می شود .

کدBCD	مبنای 10
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9

$(10)_{10}=(1010)_2$

$(10)_{10}=(00010000)_{BCD}$

$(11)_{10}=(1011)_2$

$(11)_{10}=(00010001)_{BCD}$

$(12)_{10}=(1100)_2$

$(12)_{10}=(00010010)_{BCD}$

$(20)_{10}=(10100)_2$

$(20)_{10}=(00100000)_{BCD}$

رقم های 0 تا 9 درمبنای 2 و BCD باهم برابرند ولی اعداد از 9 به بعد در BCD از لحاظ ارزش معادل مبنای دو + 6 می باشد.

کد خود مکمل کدی است که در آن مکمل 9 عدد با مکمل 1 آن عدد برابر است.

کد وزن دار : کدی است که هر رقم آن یک ارزش دارد (یک ضریب یا وزن دارد).

مبنای 10	کد وزن دار (8 4 -2 -1)
0	0000
1	0111
2	0110
3	0101
4	0100
5	1011
6	1010
7	1001
8	1000
9	1111

خود مکمل است / متمم 1= مکمل 9

کد گری(Gray) یا کد انعکاسی :

در کد گری هر عدد با عدد قبلی فقط در یک بیت تفاوت دارد(برای ارسال متوالی اعداد استفاده می شود).

$b_n = a_n$, $b_{n-1} = a_n \oplus a_{n-1}$, $b_{n-2} = a_{n-1} \oplus a_{n-2}$

مبنای 10	مبنای 2 $a_3a_2a_1a_0$	کد گری $b_3b_2b_1b_0$
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

کد اسکی: در یک بایت از 0 تا 127 برای حروف انگلیسی و از 127 تا 255 برای زبان دوم اما جدیدا" کدهای اسکی 2 بایت است یعنی تا $2^{16}-1$ داریم تا زبان های مختلف را وارد کنیم

کد توازن (parity): این کد برای تشخیص خطا استفاده می‌شود.و دارای دو نوع می‌باشد

توازن فرد (odd parity) :تعداد یک‌ها در داده توازن فرد است.

توازن زوج (even parity):تعداد یک‌ها در داده و توازن زوج است.

مبنای 2	P_o(توازن فرد)	P_e(توازن زوج)
0000	1	0
0001	0	1
0010	0	1
0011	1	0
0100	0	1
0101	1	0
0110	1	0
0111	0	1
1000	0	1
1001	1	0
1010	1	0
1011	0	1
1100	1	0
1101	0	1
1110	0	1
1111	1	0

$P_e=a_n \oplus a_{n-1} \oplus \ldots\ldots\ldots a_0$

$P_o=(a \oplus_n a_{n-1} \oplus \ldots\ldots\ldots\ldots a_0)'$

فصل2: ساده سازی توابع با جبر بول

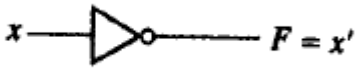
گیت‌های منطقی: برای ساخت مدارات منطقی از ترانزیستور استفاده می شود. گیت ها یکسری مدارات اصلی و آماده هستند که ساخته شده اند و در آزمایشگاها قابل استفاده هستند. انواع مهم گیت‌های منطقی عبارتند از:

گیت بافر: وظیفه‌ی آن تنظیم مقدار x است و اجازه نمی‌دهد ولتاژ مورد استفاده برای 0و1 تضعیف شود.

X	F
0	0
1	1

گیت **not**:

X	F
0	1
1	0

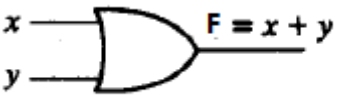


گیت **and**:

X	Y	x.y
0	0	0
0	1	0
1	0	0
1	1	1

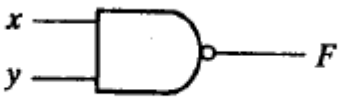
گیت **or**:

X	Y	x+y
0	0	0
0	1	1
1	0	1
1	1	1



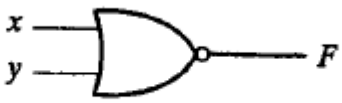
گیت **nand**(**not and**)

X	Y	(x.y)'
0	0	1
0	1	1
1	0	1
1	1	0



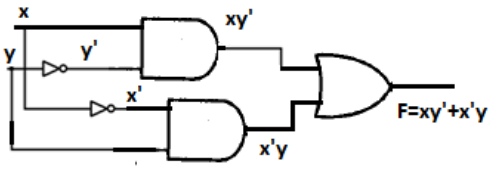
گیت **nor**(**not or**)

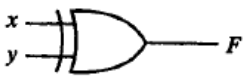
X	Y	(x+y)'
0	0	1
0	1	0
1	0	0
1	1	0



گیت **xor**

X	y	x'	y'	xy'	x'y	xy'+x'y
0	0	1	1	0	0	0
0	1	1	0	0	1	1
1	0	0	1	1	0	1
1	1	0	0	0	0	0



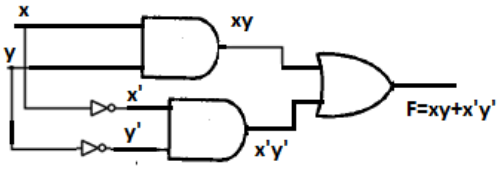


$F = xy' + x'y$
$= x \oplus y$

گیت

xnor

X	y	x'	y'	x'y'	xy	x'y'+xy
0	0	1	1	1	0	1
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	1	0	0	0	1	1



$F = xy + x'y'$
$= (x \oplus y)'$

جبر بول: یک روش ساده‌سازی عبارات (توابع منطقی) استفاده از جبر بول است برای ساده‌سازی از قوانین جبر بول استفاده می-کنیم. قوانین جبر بول عبارتند از:

1- $x+0=x$ $x.1=x$

2. $x+x'=1$ $x.x'=0$

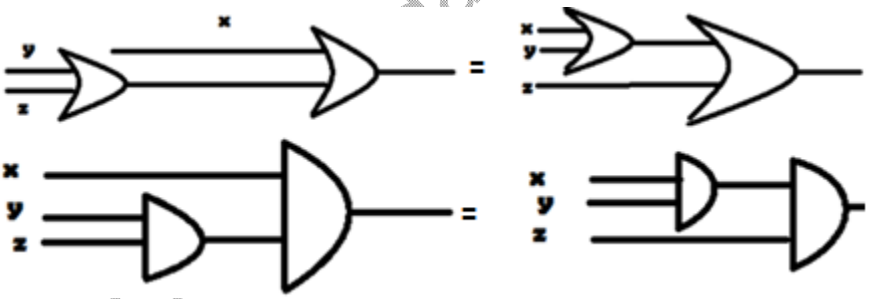
3. $x+x=x$ $x.x=x$

4. $x+1=1$ $x.0=0$

5. $(x')'=x$

6. $x+y=y+x$ $x.y=y.x$

7. $x+(y+z)=(x+y)+z$ $x(yz)=(xy)z$



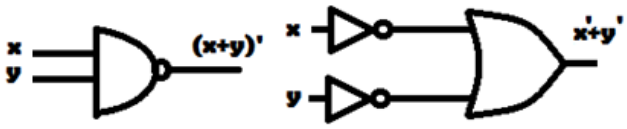
8. $x(y+z)=xy+xz$ $x+yz=(x+y)(x+z)$

9. $(x+y)'=x'y'$ $(xy)'=x'+y'$ قوانین دمورگان

10. $x+xy=x$ $x.(x+y)=x$ قوانین جذب

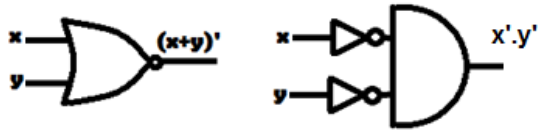
قوانین دمورگان:

$(x.y)'=x'+y'$



and_invert=invert_or

$$(x+y)'=x'.y'$$



Or_invert=invert_and

اثبات درستی قوانین : یک روش اثبات قوانین جدول درستی(جدول صحت) می باشد در این روش صحت عبارات سمت چپ را بدست آورده و صحت عبارات سمت راست را نیز بدست می آوریم که باید برابر باشد. برای بدست آوردن جدول صحت براساس تعداد ورودی های صورت مساله معادل مبنای دو را نوشته و بر اساس رابطه بین خروجی و ورودی ها مقدار خروجی به ازای هر ترکیب ورودی را بدست می آوریم. این عمل را برای هر دو عبارت مد نظر صورت مساله انجام می دهیم.

مثال: درستی $x=x+xy$ را با جدول صحت بررسی کنید:

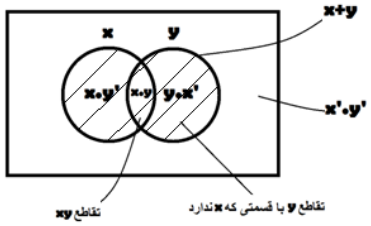
x	y	xy	x+xy
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

مثال:با جدول درستی ثابت کنید $x+yz=(x+y)(x+z)$

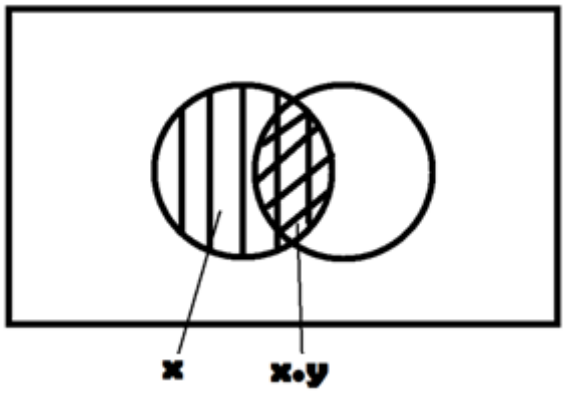
x	Y	z	yz	x+yz	x+y	x+z	(x+y)(x+z)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

دیاگرام ون: یک روش دیگر برای اثبات برابری دو عبارت است.

مثال: درستی قانون دمورگان $(x+y)'=x'y'$ را با دیاگرام ون نمایش دهید.

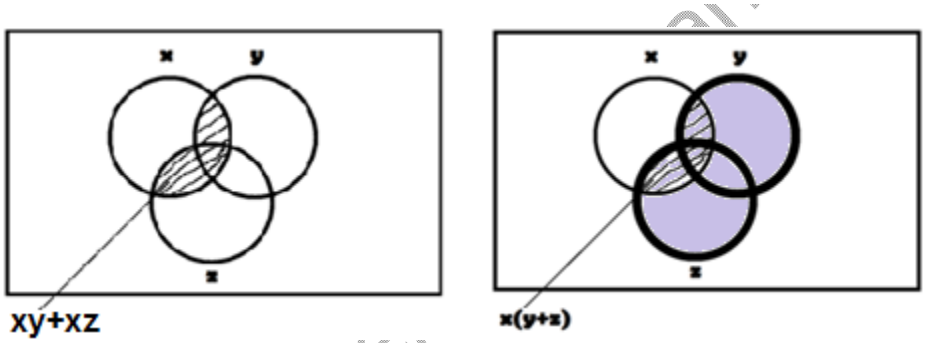


مثال: درستی قانون x+xy=x را با دیاگرام ون نشان دهید:



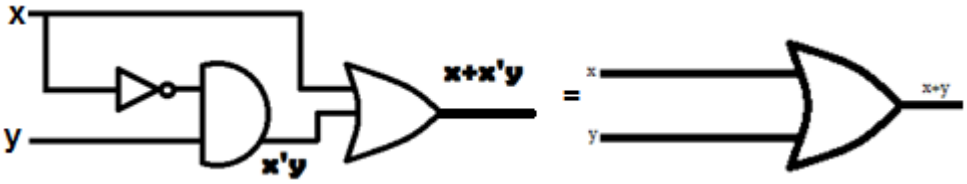
x+xy=x

مثال:مثال با دیاگرام ون قانون x(y+z)= xy+xz را ثابت کنید.

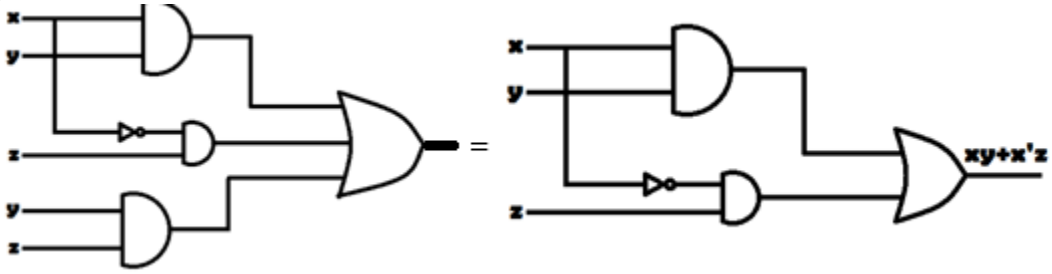


مثال: عبارات زیر را با استفاده از جدول ساده کنید.

1. x+x'y=(x+x')(x+y)=x+y



2. x(x'+y)=xx'+xy=xy
3. x'y'z+x'yz+xy'=x'z(y'+y)+xy'=x'z+xy'
4. xy+x'z+yz=xy+x'z+yz(x+x')=xy+x'z+yzx+yzx'
   xy(1+z)+x'z(1+z)=xy+x'z



- یک متغیر
- هر چه

مکمل یک

$(x+y+z)'=(x+B)'=x'.B'=x'(y+z)'=x'.y'.z'$

تعریف مینترم و ماکسترم: مینترم به عبارتی گفته می شود که فقط به ازای یک ترکیب ورودی ها 1 است و در بقیه حالات صفر است.

ماکسترم: عبارتی است که به ازای ترکیب ورودی ها صفر است و در بقیه حالات یک است.

X	y	$m_0=x'y'$	$m_1=x'y$	$m_2=xy'$	$m_3=xy$	$M_0=x+y$	$M_1=x+y'$	$M_2=x'+y$	$M_3=x'+y'$	F_1
0	0	1	0	0	0	0	1	1	1	0
0	1	0	1	0	0	1	0	1	1	1
1	0	0	0	1	0	1	1	0	1	1
1	1	0	0	0	1	1	1	1	0	0

نکته: هر تابع را می توان به صورت جمع مینترم ها (Sum Of Product=SOP) نوشت:

$F1=m1+m2=\Sigma(1,2)$

نکته:هر تابع را می توان به صورت ضرب ماکسترم ها (Product of Sum=POS) نوشت:

$F1=M0.M3$

جمع مینترم ها not ضرب ماکسترم است.

$F1=M0.M3=\pi(0,3)=POS$

مینترم ها را با حرف کوچک m_i و ماکسترم ها را با حروف بزرگ M_i نشان می دهیم.

$M0=\Sigma(1,2,3)$

$M1=\Sigma(0,2,3)$

$M2=\Sigma(0,1,3)$

$M3=\Sigma(0,1,2)$

برای نوشتن مینترم بصورت عبارت بولی معادل مبنای 2 را می نویسیم مقادیر0 هر متغیر را با پریم و مقادیر 1 آن را بدون پریم نشان می‌دهیم و بین آن ها and قرار می دهیم.

$F1(x,y,z)= \Sigma(0,1,6,7)=\pi(2,3,4,5)$

$F1=x'y'z'+x'y'z+xyz'+xyz= x'y'(z'+z)+xy(z'+z)=x'y'+xy=(x \oplus y)'$

$F2(x,y,z)=\Sigma(0,1,3,5,7)=x'y'z'+x'y'z+x'yz+xy'z+xyz=x'y'z'+x'z(y'+y)+xz(y+y')=x'y'z'+x'z+xz= x'y'z'+z.(x'+x)=x'y'z'+z=x'y'+z$

F2$=x'y'z'+z\ (x'y'+x'y+xy'+xy)=x'y'z'+z=x'y'+z$

F3(x,y,x)=Σ(0,2,4,6,7)=x’y’z’+x’yz’+xy’z’+xyz’+xyz=z’(x’y’+x’y+xy’+xy)+xyz=

z’+xyz=z’+xy

F4(x,y,z)=Σ(4,5,6,7)=xy’z’+xy’z+xyz’+xyz=x(y’z’+y’z+yz’+yz)=x

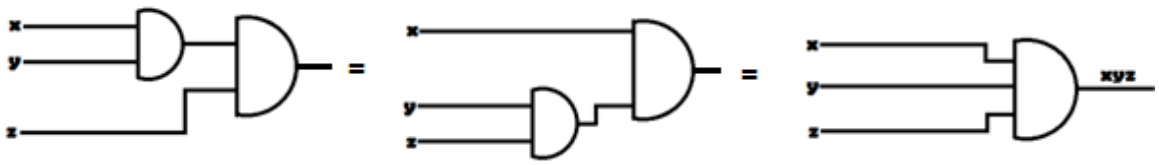
F5(x,y,z)=Σ(1,2,3,6,7)=x’y’z+x’yz’+x’yz+xyz’+xyz=x’y’z+y(x’z’+x’z+xz’+xz)=
x’y’z+y=x’z+y

F6(w,x,y,z)=Σ(0,1,2,3,12,13,14,15)=w’x’y’z’+w’x’y’z+w’x’yz’+w’x’yz+wxy’z’+wxy’z+wxyz’
+wxyz=w’x’(y’z’+y’z+yz’+yz)+wx(y’z’+y’z+yz’+yz)=w’x’+wx = wʘx

گسترش ورودی گیت ها : آیا می توان با استفاده از گیت های دو ورودی گیت های سه ورودی و بزرگتر ساخت و به عبارت دیگر آیا گیت ها شرکت پذیر هستند ؟

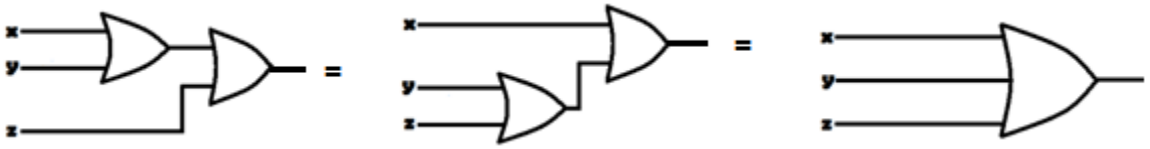
(x.y).z=x.(y.z)=x.y.z

1) گیت and:



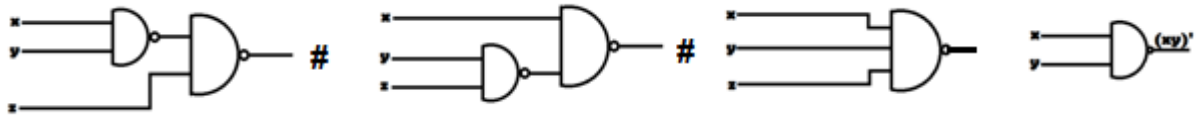
2) گیت or :

(x+y)+z=x+(y+z)=x+y+z



3) گیت Nand

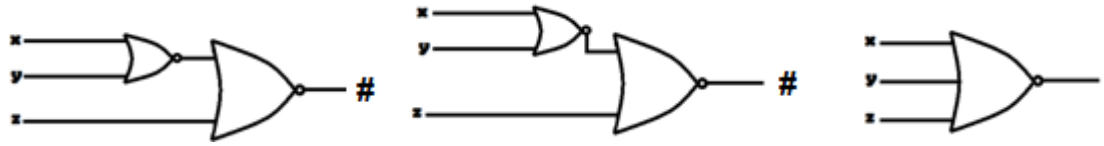
(x↑y)↑z!=x↑(y↑z)!=((x.y)’.z)’!=(x.(yz)’)’ !=(xyz)’



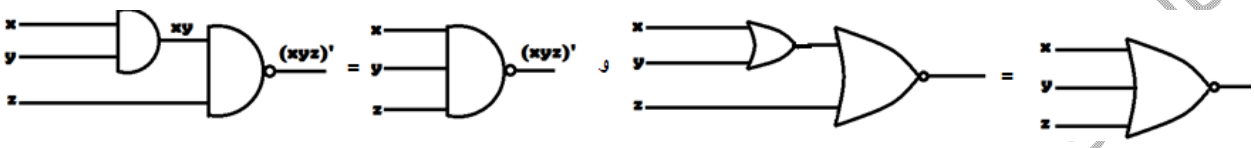
((xy)’z)’=x.y+z’ , (x(yz)')'=x'+yz, (xyz)'=x'+y'+z'

4)گیت nor

$(x\downarrow y)\downarrow z \neq x\downarrow(y\downarrow z) \neq x\downarrow y\downarrow z \neq (x+y+z)'$



نکته:



5)گیت xor: ثابت کنید $(x \oplus y) \oplus z=x \oplus(y \oplus z)=(x \oplus y \oplus z)$

x	y	z	x⊕y	(x⊕y)⊕z	y⊕z	x⊕(y⊕z)	x⊕y⊕z
0	0	0	0	0	0	0	0
0	0	1	0	1	1	1	1
0	1	0	1	1	1	1	1
0	1	1	1	0	0	0	0
1	0	0	1	1	0	1	1
1	0	1	1	0	1	0	0
1	1	0	0	0	1	0	0
1	1	1	0	1	0	1	1

$x \oplus y=x'y+xy'$

تبدیل فرم های استاندارد به یک دیگر : برای n متغیر 2^n مینترم (ماکسترم) داریم .

اگر تابعی به صورت یک فرم استاندارد بامینترم (ماکسترم) داده شود برای تبدیل به فرم متعارف دیگر حاصل را با 2^n-N ماکسترم (مینترم) می نویسیم .

مثال) :

$F(A,B,C,D)=\Sigma(0,1,5,7,9,14)=\pi(2,3,4,6,8,10,11,13,14,15)$

تبدیل یک عبارت ساده شده به فرم های استاندارد : ابتدا تعداد متغیرهای تابع را مورد توجه قرار می دهیم در هر جمله ی داده متغیرهایی که وجود ندارد را با (متغیر پریم + متغیر) به حاصل ضرب اضافه می‌کنیم وحاصل را می یابیم. ترتیب متغیرها مهم است .

مثال) تابع F_1 جمع چه مینترم هایی می باشد ؟ (SOP)

$F(x,y,z)=x+yz+x'y'z'=x(y+y')(z+z')+yz(x+x')+x'y'z'=x(yz+y'z+yz'+y'z')+xyz+x'yz+x'y'z'=xyz+xy'z+xyz'+xy'z'+xyz+x'yz+x'y'z'=\Sigma(7,6,5,4,3,0)=\pi(1,2)$

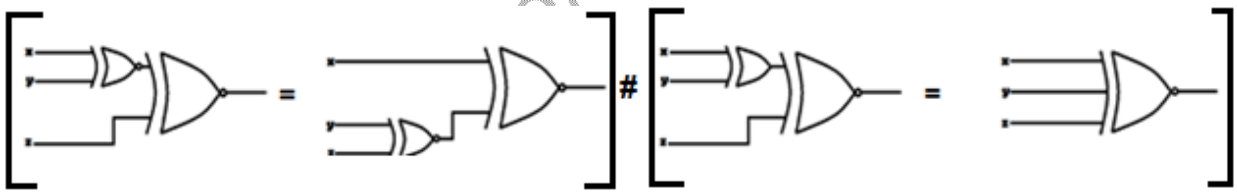
با n متغیر چند تابع می توان ساخت؟چون با n متغیر 2^n مینترم(ماکسترم)داریم و با توجه به این که هر تابع را می توان با جمع مینترم ها و یا ضرب ماکسترم ها نوشت بنابراین 2^{2^n}تابع می توان طراحی کرد.

X	y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

$F_0=0$, $F_1=m_0$, $F_2=m_1$, $F_3=m_0+m_1$, $F_4=m_2$, $F_5=m_0+m_2$,$F_6=m_1+m_2$, $F_7=m_0+m_1+m_2=M_3$
$F_8=m_3$,$F_9=m_0+m_3$, $F_{10}=m_1+m_3$, $F_{11}=M_1$, $F_{12}=m_2+m_3$,$F_{13}=x+y'$, $F_{14}=x+y$, $F_{15}=1$

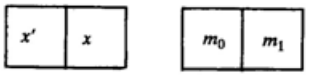
6)گیت **xnor**: [(xʘy)ʘz =xʘ(yʘz)]!=[(x⊕y)ʘz=xʘyʘz]؟

x	y	z	xʘy	x⊕y	(xʘy)ʘz	yʘz	xʘ(yʘz)	xʘyʘz	(x⊕y) ʘz
0	0	0	1	0	0	1	0	1	1
0	0	1	1	0	1	0	1	0	0
0	1	0	0	1	1	0	1	0	0
0	1	1	0	1	0	1	0	1	1
1	0	0	0	1	1	1	1	0	0
1	0	1	0	1	0	0	0	1	1
1	1	0	1	0	0	0	0	1	1
1	1	1	1	0	1	1	1	0	0



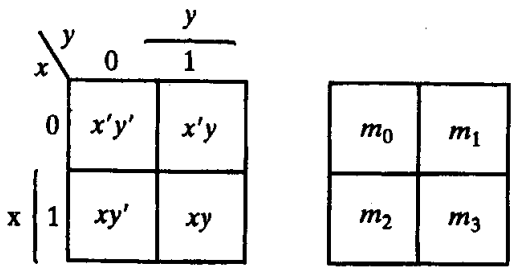
فصل سوم: ساده سازی با جدول کارنو

جدول کارنوی یک متغیره: جدولی شامل دو خانه که یک متغیر روی آن توسعه می‌یابد.



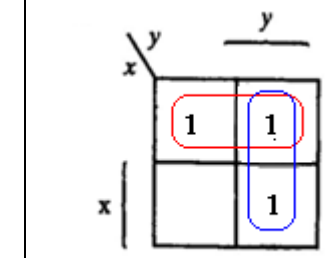
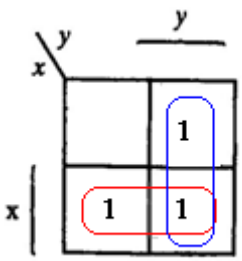
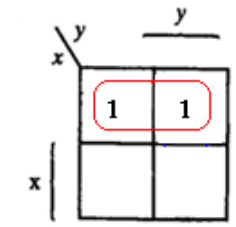
F(x) (ب) (الف)

جدول کارنوی دو متغیره: جدولی شامل چهار خانه که دو متغیر روی آن توسعه می‌یابد.

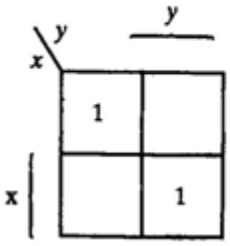
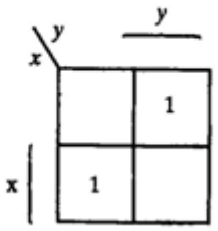


F(x,y) (ب) (الف)

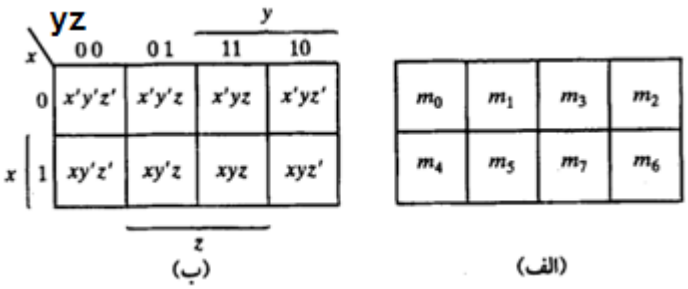
مثال) توابع زیر را با جدول کارنوی دو متغیره ساده کنید.

$F_1(x,y) = \sum(0,1,3) = x' + y$	$F_2(x,y) = \sum(1,2,3) = x + y$	$F_3(x,y) = \sum(0,1) = x'$
		

در جدول کارنوی دو متغیره همسایگی به صورت سطری و ستونی می‌باشد. مثلاً مینترم‌های 0و 1 با هم همسایه می‌باشند. همچنین مینترم‌های 0 و 2 نیز با هم همسایه‌اند ولی مینترم‌های 0 و 3 همسایه نیستند. هر دو خانه همسایه باعث حذف یک متغیر می‌شوند.

$F_4(x,y) = \sum(0,3) = x'y' + xy = x \odot y$	$F_5(x,y) = \sum(1,2) = x'y + xy' = x \oplus y$
	

جدول کارنوی سه متغیره: جدولی شامل هشت خانه که سه متغیر روی آن توسعه می یابد.



x \ yz	00	01	11	10
0	x'y'z'	x'y'z	x'yz	x'yz'
1	xy'z'	xy'z	xyz	xyz'

(ب)

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

(الف)

نکته: در جدول کارنو اگر تمام مینترم‌ها وجود داشته باشندعبارت ساده شده برابر یک است. (F=1)

نکته: در جدول کارنو همسایگی‌ها زوج هستند. یعنی همسایگی دوتایی، چهارتایی، هشت‌تایی، شانزده تایی و... داریم.

دلیل جابجایی مینترم‌ها در جدول کارنو وجود عامل مشترک بین خانه‌های همسایه است.

y	z	
0	0	y'z'
0	1	y'z
1	1	yz
1	0	yz'

این دو تا اشتراک ندارند پس در جدول کارنو جای خانه های 10 و 11 را عوض می کنیم.

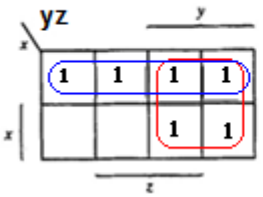
در جدول کارنوی سه متغیره هر سطر با سطر پایین و هر ستون با ستون کنارش همسایه هستند. (مثلاً m_1 وm_3 همسایه هستند).

تشریح جدول کارنوی سه متغیره: در سطرمتغیر x قرار می‌دهیم و yz را در ستون می‌گذاریم. روش مقداردهی yz بر اساس کد گری است تا بین خانه‌های مجاور عامل مشترک ایجاد شود. هر ستون با ستون کناری همسایه است و سطر اول با سطر آخر همسایه است همچنین ستون اول با ستون آخر همسایگی دارد.(همسایگی دوار)

توجه: دو خانه همسایگی یک متغیر را حذف می‌کند.چهار خانه همسایگی دو متغیر را حذف می‌کند.هشت خانه همسایگی سه متغیر را حذف می‌کند.

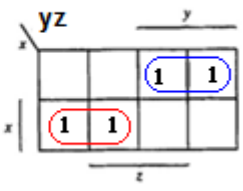
مثال) توابع زیر را با جدول کارنوی سه متغیره ساده کنید.

$F_1(x, y, z) = \sum(0,1,2,3,6,7) = x' + y$



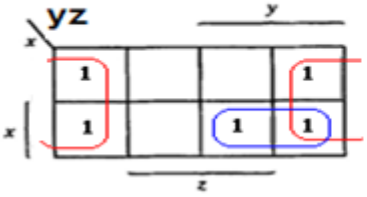
x \ yz	00	01	11	10
0	1	1	1	1
1			1	1

$F_2(x, y, z) = \sum(2, 3, 4, 5) = xy' + x'y = x \oplus y$

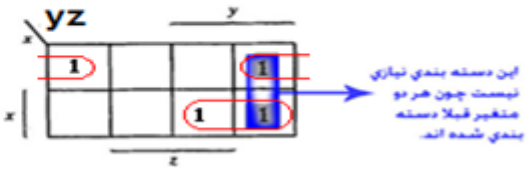


x \ yz	00	01	11	10
0			1	1
1	1	1		

$F_3(x, y, z) = \sum (0, 2, 4, 6, 7) = z' + x$

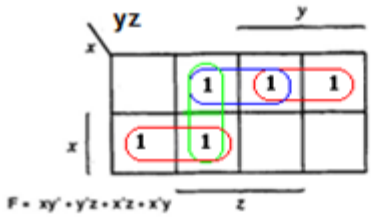


$F_4(x, y, z) = \sum (0, 2, 6, 7) = xy + x'z'$

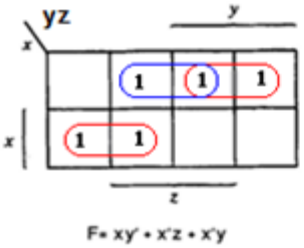


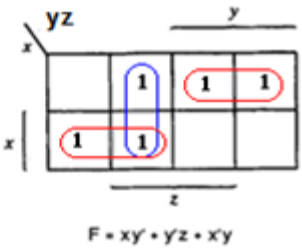
$F_5(x, y, z) = \sum (1, 2, 3, 4, 5)$

این حالت پیچیده ترین حالت است



این دو حالت ساده ترین حالت هستند:





همانطور که مشاهده می‌شود برای ساده سازی F_5 سه حالت وجود دارد که حالت های اول و دوم (سمت راست) ساده‌تر از حالت سوم سمت چپ می‌باشند.

جدول کارنوی چهار متغیره: چون با چهار متغیر $16=2^4$ مینترم داریم باید یک جدول کارنو بکشیم که 16 خانه مرتبط با هم داشته باشد. هر سطر با سطر مجاورش همسایه است. هر ستون با ستون مجاورش همسایه است. سطر اول با سطر آخر همسایه است و ستون اول با ستون آخر همسایگی دارد.(همسایگی دوار)

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6
m_{12}	m_{13}	m_{15}	m_{14}
m_8	m_9	m_{11}	m_{10}

(الف)

wx \ yz	00	01	11	10
00	$w'x'y'z'$	$w'x'y'z$	$w'x'yz$	$w'x'yz'$
01	$w'xy'z'$	$w'xy'z$	$w'xyz$	$w'xyz'$
11	$wxy'z'$	$wxy'z$	$wxyz$	$wxyz'$
10	$wx'y'z'$	$wx'y'z$	$wx'yz$	$wx'yz'$

(ب)

در جدول کارنوی چهار متغیره همسایگی ها یکی، دوتایی، چهارتایی، هشت تایی و شانزده‌تایی می‌باشند.

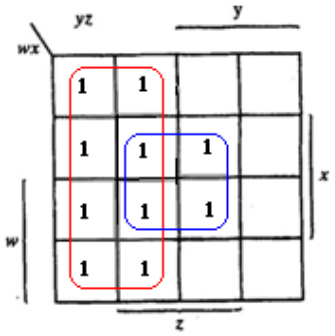
ایجاب کننده (Implicate):جمله ایست به صورت ضرب که شامل یک یا چند مینترم باشد. (تعداد مهم نیست)

ایجاب کننده اولیه (Prime Implicate=PI): ایجاب کننده ایست که توسط هیچ ایجاب کننده دیگری پوشش داده نشود یعنی دسته ایست که درون دسته بزرگتری نباشد.

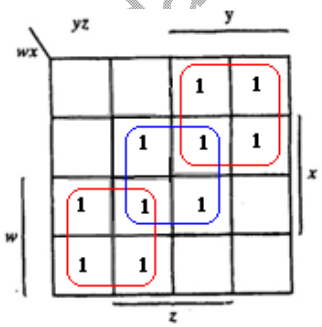
ایجاب کننده اولیه اساسی (Essential PI=EPI): یک PI که حداقل یک مینترم را که توسط هیچ PI دیگری پوشش داده نشده است پوشش دهد.

مثال) توابع زیر را با جدول کارنوی چهار متغیره ساده کنید.

$F_1(w,x,y,z) = \sum(0, 1, 4, 5, 7, 8, 9, 12, 13, 15) = y'+xz$



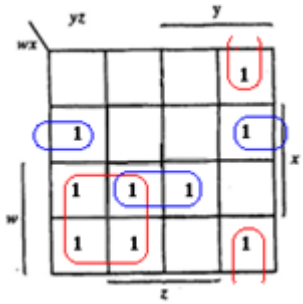
$F_2(w,x,y,z) = \sum(2, 3, 5, 6, 7, 8, 9, 12, 13, 15) = wy' + xz + w'y$



PI ها =EPI ها = 3 (دسته‌های چهارتایی)

Implicate =3+12+10 = 25 (10 تا تکی، 12 تا دوتایی، 3 تا چهارتایی)

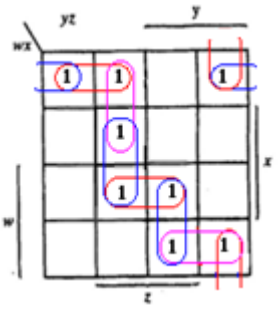
$F_3(w,x,y,z) = \sum(2, 4, 6, 8, 9, 10, 12, 13, 15)=wy'+wxz+w'xz'+x'yz'$



PI= 7 و EPI= 2و Implicate= 9+10+1 = 20 (9 تا تکی، 10 تا دوتایی، یک تا چهارتایی)

توجه: ایجاب‌کننده‌ها همه حالات PIها و EPIها را دربرمی‌گیرد یعنی برای شمارش ایجاب کننده PIها و EPIها را هم می-شماریم.

مثال) جدول زیر چند PI و EPI دارد؟

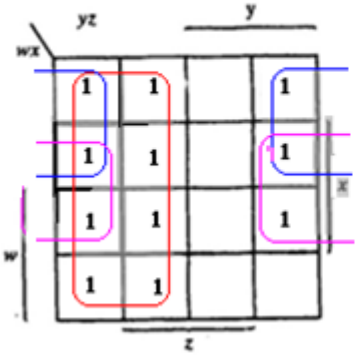


EPI=0 و PI=8 و I=8+8=16 (8 تا دوتایی ، 8 تا تکی)

مثال) توابع زیر را ساده کنید سپس تعداد PI ها و EPIها را تعیین کنید.

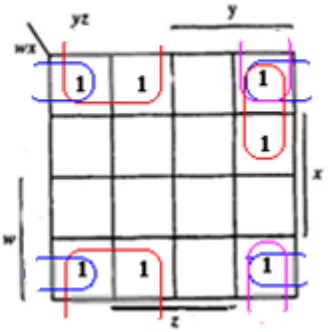
$(w,x,y,z) = \sum(0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14)$

$F = y' + w'z' + xz' = y' + z'(x + w')$ PI = EPI = ۳



$F(w,x,y,z) = w'x'y' + x'yz' + w'xyz' + wx'y'$

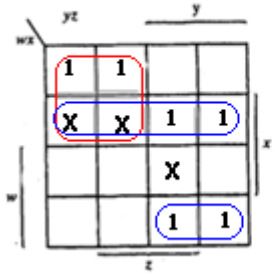
$F = x'y' + x'yz' + w'yz'$ PI = ۳ EPI = ۳



حالات بی‌اهمیت (don't care):حالاتی هستند که صفر یا یک بودن آن‌ها برای ما فرقی نمی‌کند.حالات بی‌اهمیت را با d نشان می‌دهند و در جدول کارنو معمولاً با حرف Xنشان می‌دهند. با توجه به این که مقدار این حالات مهم نیست می‌توان در ساده‌سازی از آن‌ها استفاده کرد.

مثال) تابع زیر را ساده کنید.

$F(w,x,y,z) = \sum(0, 1, 6, 7, 10, 11) + d = \sum(4, 5, 15)$



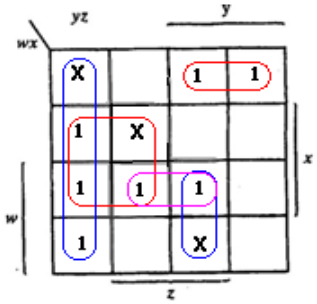
$F = w'y' + w'x + wx'y$

مثال) تابع مینیمم شده معادل با F کدام است؟ (کنکور ارشد 82)

$F(w,x,y,z) \sum (2, 3, 4, 8, 12, 13, 15) + d = \sum(0, 5, 11)$

الف) $w'x' + wxz + wyz$ ب) $xy + wxz + w'x'y'$

ج) $y'z' + w'x'y + wxz$ د) $y'z' + xy' + w'x'y'$



جواب : گزینه سه

توجه: اگر در گزینه‌ها هر سه مورد داشتیم باید همه حالات دسته بندی را در نظر می‌گرفتیم.

مثال) کدام یک از گزینه‌ها حاصل ضرب ماکسترم نمایش canonical یا استاندارد تابع زیر است؟ (دولتی 75)

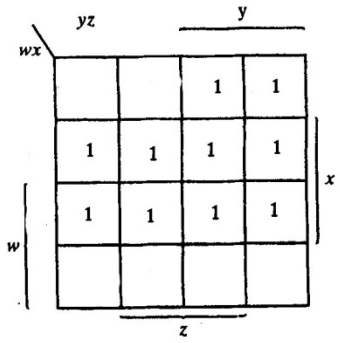
$F(w,x,y,z) = xy' + w'y + wxy$

الف) $\pi M(0, 1, 8, 9, 10, 11)$

ب) $\pi M(0, 1, 2, 3, 7, 12, 15)$

ج) $\pi M(1, 3, 4, 5, 7, 12, 13, 14, 15)$

د) $\pi M(0, 1, 2, 6, 8, 9, 11)$



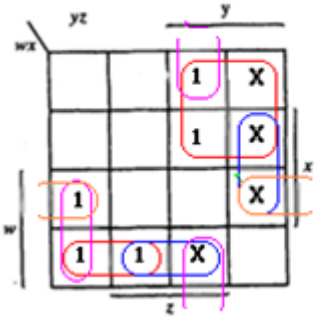
$F = \sum(2, 3, 4, 5, 6, 7, 12, 13, 14, 15) = \pi(0, 1, 8, 9, 10, 11)$

جواب: گزینه الف می باشد.

نکته: در حالتی که don't care داریم هنگام شمارش PI باید don't careها را هم در نظر بگیریم و هنگام شمارش EPI یک‌هایی را در نظر می‌گیریم که فقط در یک دسته باشند.

مثال) در تابع زیر به ترتیب چند PI و EPI وجود دارد.

$F(w,x,y,z) = \sum(3, 7, 8, 9, 12) + d = \sum(2, 6, 11, 14)$



الف) (۵و۱)　　ب) (۵و۲)

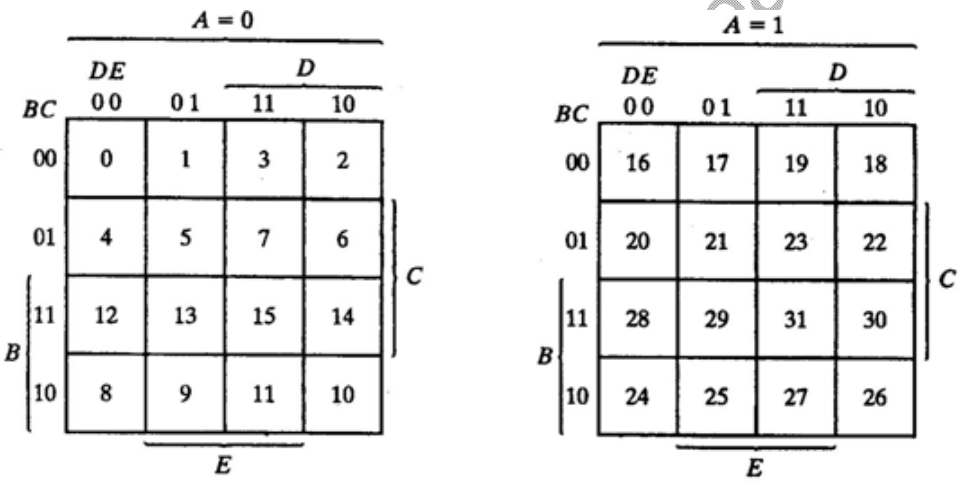
ج) (۷و۱)　　د) (۷و۲)

PI = ۷　EPI = ۱

جواب: گزینه ج می باشد.

توجه: برای شمارش EPI ها فقط یک‌ها را در نظر می‌گیریم X ها را نمی‌شماریم.

جدول کارنوی پنج متغیره : (شامل دو جدول چهار متغیره)



در هر جدول چهار متغیره همسایگی به صورت کروی می باشد. در ضمن جدول یک و دو به طور متناظر همسایه می باشند. یعنی اگر جدول دو را روی جدول یک قرار دهیم خانه‌هایی که روی هم می‌افتند همسایه هستند. مثلاً مینترم های (7,5) با مینترم‌های (23,21) همسایه‌اند.

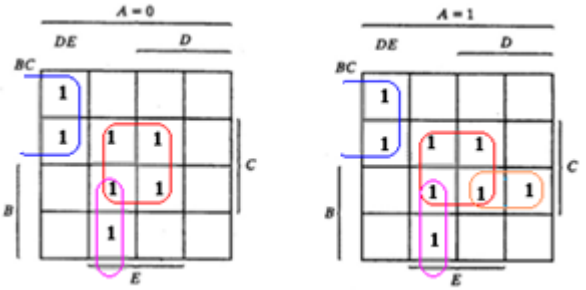
ستون اول جدول اول با ستون اول جدول دوم همسایه‌اند. سطر اول جدول با سطر اول جدول دوم همسایه‌اند و الی آخر. همسایگی ها سی و دو تایی، شانزده تایی، هشت تایی، چهارتایی، دوتایی و یکی می‌باشند.

توجه: به هیچ وجه مینترم‌های 18 و 0 باهم همسایه نیستند، مینترم 18 می‌تواند با مینترم 2 همسایه باشد.

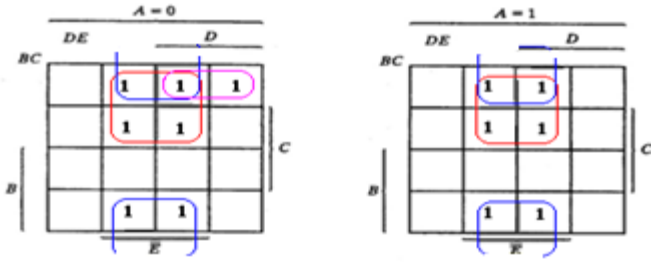
نکته: داخل هر همسایگی هشت تایی دوازده همسایگی دوتایی وجود دارد.

مثال) عبارت‌های زیر را ساده کنید.

$F(A,B,C,D,E) = \sum(0, 4, 5, 7, 9, 13, 15, 16, 20, 21, 23, 25, 29, 30, 31)$



$F(A,B,C,D,E) = \sum(1, 2, 3, 5, 7, 9, 11, 17, 19, 21, 23, 25, 27) = B'E + C'E + A'B'C'D'$



همه PI هایی که گرفته شد EPI هم هستند. (3 عدد)

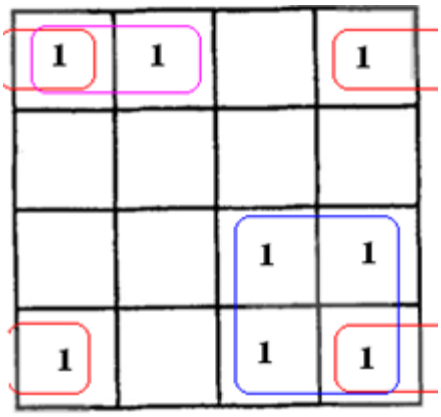
روش ساده سازی کوئین – مک کلاسکی (روش الگوریتمی):

جدول کارنو تا پنج و شش متغیر برای ساده‌سازی مناسب است ولی بیشتر از آن پیچیده خواهد شد و نوشتن برنامه‌ای که عمل ساده‌سازی را انجام دهد، مشکل می‌شود. در این روش، روشی الگوریتمی ارائه شده است که می‌توان آن را با استفاده از یک برنامه کامپیوتری پیاده‌سازی کرد. هرچند که روش جدول کارنو روی قلم و کاغذ سریع‌تر است.

روش کار: ابتدا مینترم‌هایی که در معادل باینری آنها، همه صفرند می‌نویسیم. سپس مینترم‌هایی که فقط یک بیت یک دارند را می‌نویسیم. در ادامه مینترم‌هایی را می‌نویسیم که دو بیت یک دارند، سپس مینترم‌هایی که سه بیت یک دارند و .... بین هر دسته را با یک خط جدا می‌کنیم و این نوشتن‌ها به صورت ستونی انجام می‌شود. بین هر دو دسته متوالی مینترم‌هایی را که فقط در یک بیت با هم تفاوت دارند با هم دسته‌بندی می‌کنیم و کنار آن یک علامت تیک قرار می‌دهیم و در ستون دوم می‌نویسیم و به جای بیتی که مقدار متفاوت داشته است خط تیره قرار می‌دهیم. این عمل را برای ستون دوم، سوم و .... هم تکرار می‌کنیم تا زمانی که اشتراکی وجود نداشته باشد. دسته‌های به دست آمده PI خواهد بود.

مثال) تابع زیر را به روش جدول بندی (کوئین – مک کلاسکی) ساده کنید.

$F(w,x,y,z) = \sum(0, 1, 2, 8, 10, 11, 14, 15)$



ستون اول	ستون دوم	ستون سوم
w x y z	w x y z	w x y z
0 0 0 0 0√	0,1 0 0 0 -	0,2,8,10 - 0 – 0
1 0 0 0 1√	0,2 0 0 - 0 √	0,8,2,10 - 0 – 0
2 0 0 1 0√	0,8 - 0 0 0 √	
8 1 0 0 0√		10,11,14,15 1 - 1 –
	2,10 - 0 1 0 √	10,14,11,15 1 - 1 –
10 1 0 1 0√	8,10 1 0 - 0 √	
11 1 0 1 1√	10,11 1 0 1 – √	
14 1 1 1 0√	10,14 1 – 1 0 √	
15 1 1 1 1√	11,15 1 – 1 1 √	
	14,15 1 1 1 1 √	

هر کدام که تیک نخورد EPI است.

جدول پوشش: گاهی اوقات در روش کوئین – مک کلاسکی ممکن است انتخاب‌های به دست آمده بهینه نباشد. در مثال فوق همه انتخاب‌ها EPI بودند ولی این همیشه درست نیست. برای تعیین EPIها از جدول پوشش استفاده می‌کنیم. در این جدول PIها را قرار می‌دهیم و مینترم‌های صورت مسئله را روی ستون‌ها قرار می‌دهیم.در هر PI مینترم‌هایی که پوشش داده شوند را تیک می‌زنیم اگر ستونی وجود داشته باشد که فقط یک تیک داشته باشد PI متناظر آن EPI است در غیر این صورت از PIهای باقیمانده PIهایی را انتخاب می‌کنیم که مینترم‌های باقی مانده را پوشش دهد.

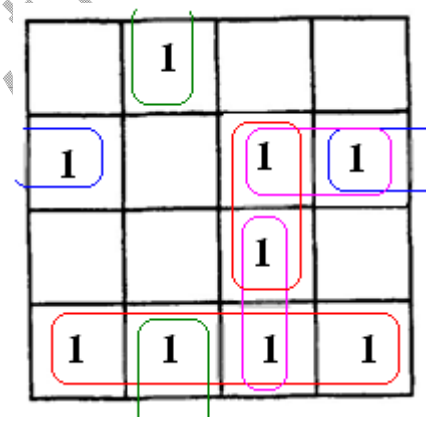
مثال) تابع زیر را به روش کوئین – مک کلاسکی ساده کنید.

$F(w,x,y,z) = \sum(1, 4, 6, 7, 8, 9, 10, 11, 15)$

ستون اول w x y z	ستون دوم w x y z	ستون سوم w x y z
0 0 0 1 ✓	_ 0 0 1 (1 , 9) PI2	1 0 _ _ (8 , 9 , 10 , 11) PI1
0 1 0 0 ✓	0 1 _ 0 (4 , 6) PI3	1 0 _ _ (8 , 9 , 10 , 11)
1 0 0 0 ✓	1 0 0 _ (8 , 9) ✓	
	1 0 _ 0 (8 , 10) ✓	
0 1 1 0 ✓	0 1 1 _ (6 , 7) PI4	
1 0 0 1 ✓	1 0 _ 1 (9 , 11) ✓	
1 0 1 0 ✓	1 0 1 _ (10 , 11) ✓	
0 1 1 1 ✓	_ 1 1 1 (7 , 15) PI5	
1 0 1 1 ✓	1 _ 1 1 (11 , 15) PI6	
1 1 1 1 ✓		

اگر در زوج (x , y) فاصله بین دو عدد توانی از دو باشد می‌توان آنها را دسته‌بندی کرد مثل (9 و 1) یا (10 و 8)اما مثلاً (6 و 8) را نمی‌توان دسته بندی کردچون 6 از 8 کمتر است پس عدد پایینی باید از عددبالایی بزرگتر باشد.

$F = PI_1 + PI_2 + PI_3 + PI_4 + PI_5 + PI_6 = wx' + x'y'z + w'xz' + w'zy + xyz + wyz$



همان طور که در جدول کارنو می بینیم w'zy و wyz قبلاً دسته بندی شده‌اند ولی در جدول بندی کوئین مک کلاسکی باز نوشته شده اند. حال جدول پوشش را رسم می‌کنیم:

	1	4	6	7	8	9	10	11	15
$EPI_1 = PI_1 = (8, 9, 10, 11)$					✓	✓	✓	✓	
$EPI_2 = PI_2 = (1, 9)$	✓					✓			
$EPI_3 = PI_3 = (4, 6)$		✓	✓						
$PI_4 = (6, 7)$			✓	✓					
$PI_5 = (7, 15)$				✓					✓
$PI_6 = (11, 15)$								✓	✓

در جدول فوق نگاه می کنیم کدام ستون‌ها فقط یک تیک دارند. PI هایی که دور تیک آنها خط کشیده شده است EPI هستند. از این سه EPI مینترم های 1, 4, 6, 8, 9, 10,11 را پوشش می‌دهند تنها میترم‌های 7و15 پوشش داده نشده اند که جواب نهایی را می‌توان به دو صورت زیر نمایش داد:

1) $EPI_1 + EPI_2 + EPI_3 + PI_5 = wx' + x'y'z + w'xz' + xyz$

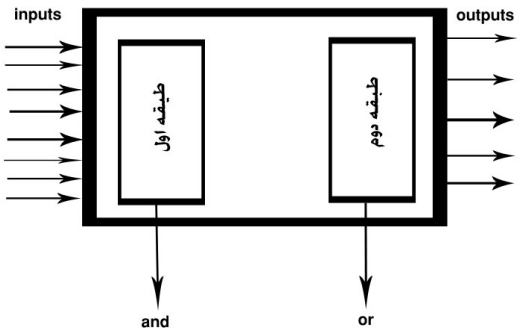
2) $EPI_1 + EPI_2 + EPI_3 + PI_4 + PI_6$

حالت اول از حالت دوم بهتر است. در حالت اول برای انتخاب مینترم‌های 7 و 15 می‌توان از PI_5 استفاده کرد و در حالت دوم برای انتخاب مینترم‌های 7 و 15 می‌توان از $PI_4 + PI_6$استفاده کرد.

پیاده‌سازی توابع با گیت‌های and-or , or-and , nor- nand

1- پیاده سازی با گیت‌های and-or

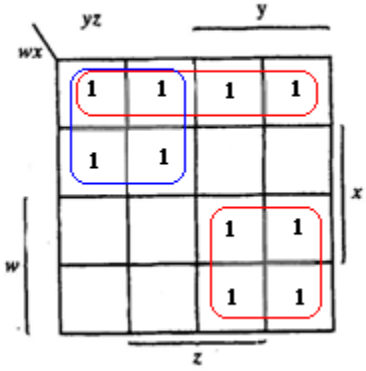
مدار



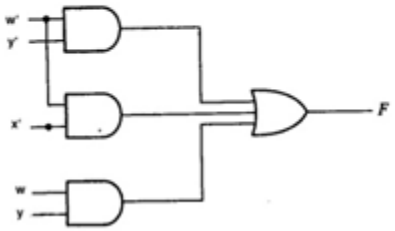
برای پیاده سازی به روش معمول تابع را با جدول کارنوساده‌سازی می‌کنیم و یک‌ها‌را دسته‌بندی می‌کنیم.(SOP) مدار ساده شده تابع را رسم می‌کنیم.

مثال) تابع زیر را با گیت‌های and-or پیاده‌سازی کنید.

$F(w,x,y,z) = \sum(0, 1, 2, 3, 4, 5, 10, 11, 14, 15)$



$F = w'y' + wy + w'x'$



(مدار double-rail)

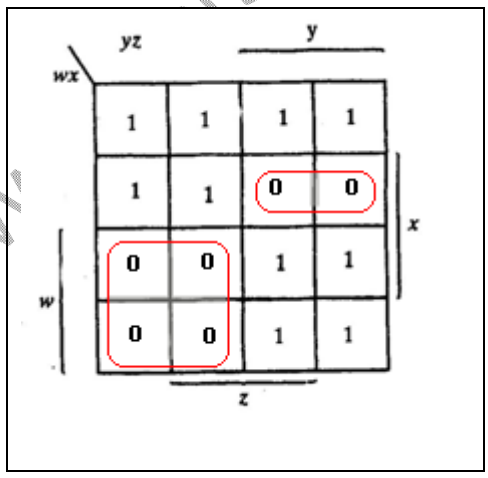
نکته: مدار single-rail: فقط متغیرهای ساده را داریم.(notها را نداریم.)
مدار double-rail: هم متغیرها را داریم و هم not آنها را.

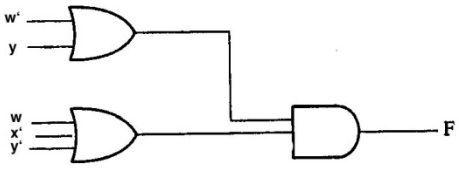
2-پیاده سازی با گیت‌های or-and

به جای یک‌ها در جدول کارنو صفرها را دسته‌بندی می‌کنیم. تابع به دست آمده F' خواهد بود.سپس با قوانین دمورگان F را به دست می‌آوریم.(POS)

مثال) تابع مثال قبل را با گیت‌های or-and پیاده‌سازی کنید.

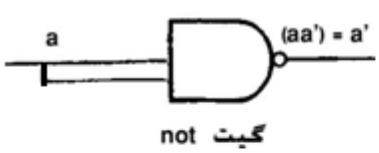
$F' = wy' + w'xy \rightarrow F = (wy' + w'xy)' = (w' + y)\,(w + x' + y')$



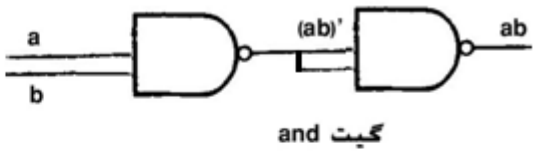


گیت‌های کامل(Universal): به گیت (هایی) کامل می‌گوییم که بتوان با استفاده از آنها هر گیت دیگری را پیاده سازی نمود.

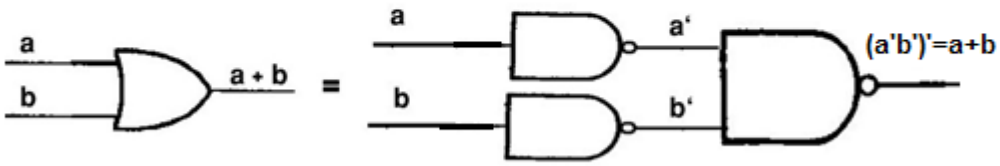
✓ بررسی گیت Nand:زمانی خروجی nand یک است که حداقل یکی از ورودی‌های آن یک باشد.



گیت not



گیت and



a+b=(a'b')'

گیت OR

به همین ترتیب می‌توان سایر گیت‌ها را هم با گیت nand ساخت در نتیجه nand کامل است.

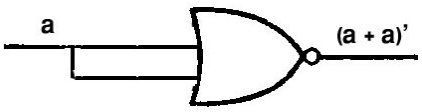
آیا گیت and کامل است؟ خیر، چون نمی‌توان با آن not ساخت.

آیا گیت or کامل است؟ خیر، چون نمی‌توان با آن not ساخت.

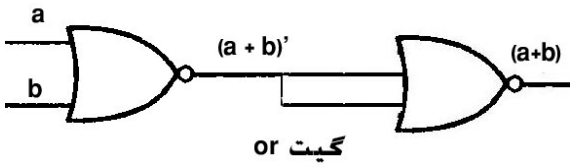
(and و not) باهم کامل هستند ولی به تنهایی کامل نیستند.

گیت nor یک گیت کامل است زیرا با آن می‌توان هر گیتی ساخت.

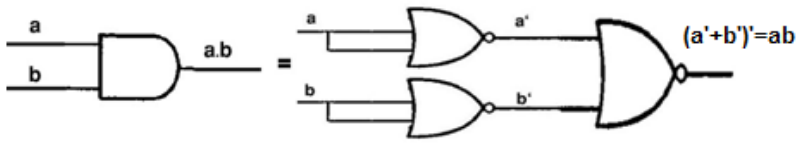
✓ بررسی گیت nor:



گیت not



گیت or



گیت and

گیت nor کامل است زیرا می‌توان با آن هرگیتی ساخت. زمانی خروجی nor یک است که هر دو ورودی آن صفر باشد یا زمانی صفر است که یکی از ورودی‌های آن یک باشد. از لحاظ کامل بودن چند تعریف وجود دارد که عبارتند از:

کامل قوی: فقط با وجود متغیرهای $x_1,x_2,\ldots,x_n$ همه توابع را می‌توان با n متغیر ساخت مثل (and-not),(or-not),nand, nor

کامل ضعیف: فقط با وجود متغیرهای $0,1x_1,x_2,\ldots,x_n$, همه توابع را می‌توان با n متغیر ساخت مثل (xor-or),(xor-and)

کامل متممی قوی: ورودی عبارت است از $x_1,x_2,\ldots,x_n$ و $x'_1,x'_2,\ldots,x'_n$ مثل or و and

کامل متممی ضعیف: ورودی عبارت است از $x_1,x_2,\ldots,x_n$ و $x'_1,x'_2,\ldots,x'_n$ و 0 و 1 مثل $\{x_1 + x_2 . x_3\}$

تمرین 1) آیا گیت xor کامل است؟

تمرین 2) آیا گیت xor-and کامل است؟

تمرین 3) آیا گیت xor-or کامل است؟

مثال) گزاره های زیر را در نظر بگیرید و گزینه صحیح را انتخاب نمایید. (مکاترونیک 87)

1- مجموعه nor از نظر عملیاتی کامل است.
2- مجموعه {xor-and} از نظر عملیاتی کامل است.

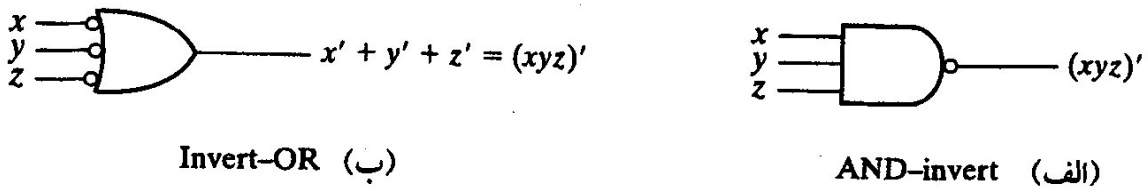
الف) هر دو گزینه صحیح هستند. ب) هر دو گزینه نا صحیح هستند.

ج) گزاره اول ناصحیح و گزاره دوم صحیح است. د) گزاره دوم ناصحیح و گزاره اول صحیح است.

جواب : گزینه الف صحیح می باشد.

3- پیاده سازی تابع با گیت‌های nand

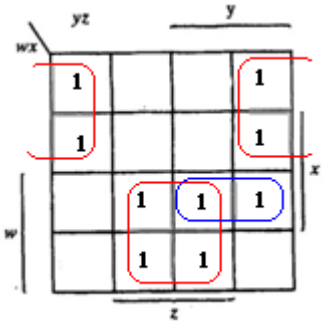
برای پیاده سازی مدار با گیت nand ابتدا آن را به صورت SOP پیاده‌سازی می‌کنیم. مدار به دست آمده به صورت and-or خواهد بود در خروجی هر گیت and یک not قرار می‌دهیم و آن را با not دیگری به ورودی گیت or خنثی می‌کنیم. طبقه اول تبدیل به nand می‌شود و با توجه به این جمله "or-invert = and-invert" طبقه دوم نیز nand می‌شود.

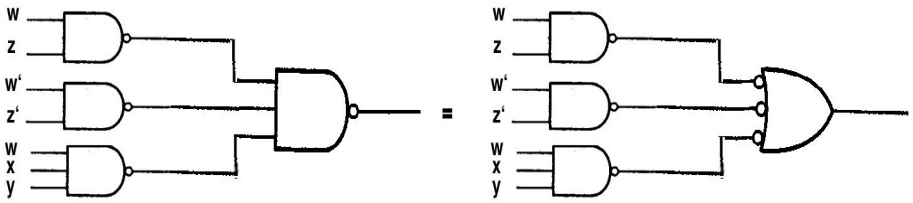


نکته: اگر گیت or دارای یک ورودی باشدکه از خروجی گیت and نیامده باشد، یعنی از ورودی‌ها مستقیماً آمده باشد، کافیست ورودی را not کنیم و یک not در ورودی گیت or قرار دهیم.

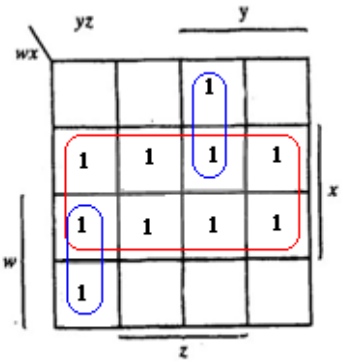
مثال) تابع زیر را با گیت nand طراحی کنید.

$F(w,x,y,z) = \sum(0, 2, 4, 6, 9, 11, 13, 14, 15) = wz + w'z' + wxy$



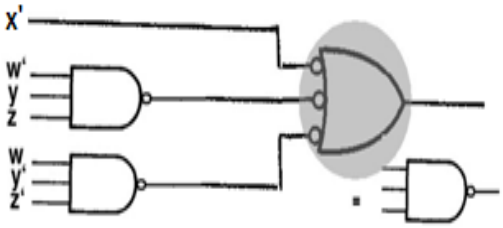


مثال) تابع زیر را با گیت nand طراحی کنید.



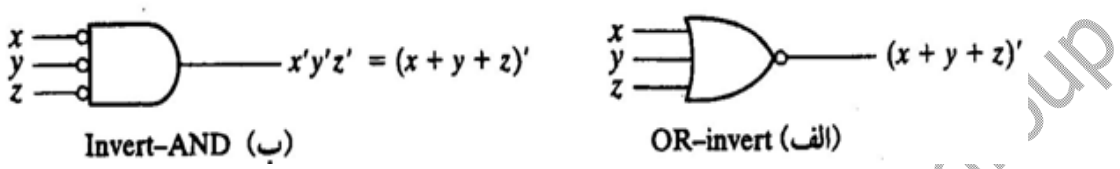
$F(w,x,y,z) = \sum(3, 4, 5, 6, 7, 8, 12, 13, 14, 15)$

$F = x + w'yz + wy'z'$ همه دسته‌بندی‌ها EPI هستند.



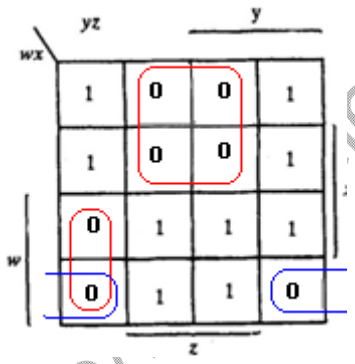
4- پیاده‌سازی با گیت‌های nor:

برای پیاده سازی ابتدا مدار را به صورت POS ساده می‌کنیم. مدارات به دست آمده به صورت or-and خواهد بود. در خروجی های گیتهای or ، Not می‌گذاریم و آن را با یک not دیگر در ورودی and خنثی می‌کنیم. طبقه اول تبدیل به nor شده است و طبقه دوم با توجه به اینکه "or-invert = and-invert" به nor تبدیل شده است.

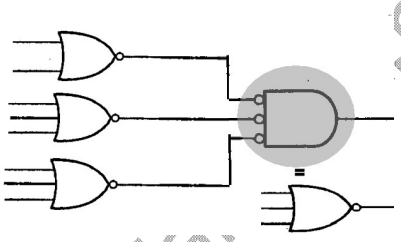


مثال) تابع زیر را با nor پیاده سازی کنید.

$F(w,x,y,z) = \sum(0, 2, 4, 6, 9, 11, 13, 14, 15)$, $F' = w'z + wx'z' + wy'z'$



همه دسته‌بندی‌ها EPI هستند.

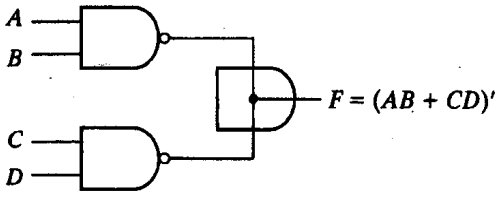


تکنولوژی‌های ساخت گیت‌ها:

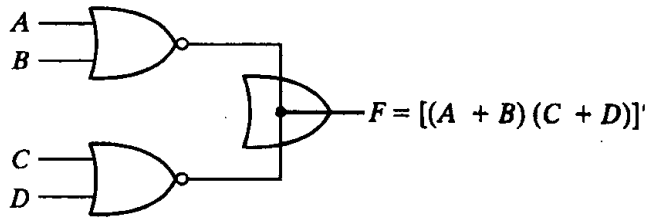
برای طراحی گیت‌ها از تکنولوژی‌های مختلف می‌توان استفاده کرد که عبارتند از:

RTL, DTL, TTL, ECL, NMOS, CMOS,PMOS

- ✓ اگر در ساختن گیت nand از تکنولوژی TTL (کلکتور باز) استفاده شود، اتصال دو خروجی آن‌ها به معنی and خواهد بود.(wired and)

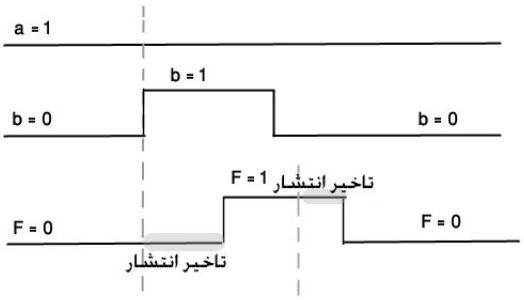


✓ اگر در ساختن گیت nor از تکنولوژی ECL استفاده شده باشد اتصال دو خروجی آن‌ها به معنی or است. (wired or)



تاخیر انتشار (propagation delay): مدت زمانی است که تغییر روی ورودی‌ها به خروجی‌ها منتقل می‌شود. معمولاً تأخیر بر اساس ns است.

مثلاً اگر فرض کنیم در ابتدا ab=10 باشد و به ab=11 تغییر پیدا کند:

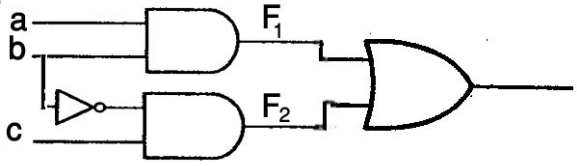


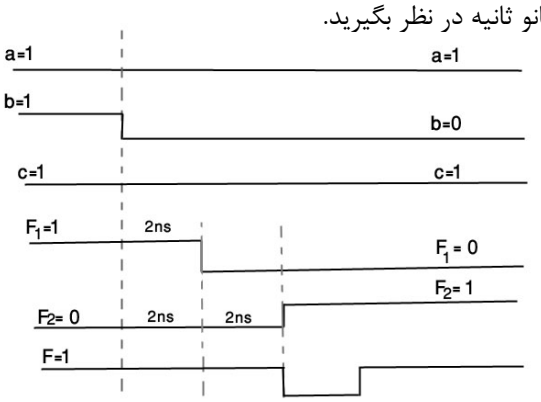
مخاطره (Hazard): اگر از مدار انتظار یک خروجی داشته باشیم که در یک زمان کوتاه آن خروجی بر آورده نشود می‌گوییم مخاطره اتفاق افتاده است.علت ایجاد مخاطره وجود تأخیر انتشار بین گیت‌ها می‌باشد.

انواع مخاطره:

1- مخاطره ایستای سطح یک: انتظار داریم خروجی یک باشد، در یک زمان کوتاه خروجی به صفر تغییر می‌کند.

مثال)در مدار زیر در ابتدا abc=111می‌باشد و در زمان t_0، abc=101 میـشود. آیا مدار مخاطره دارد؟ تأخیر انتشار هر گیت را 2 نانو ثانیه در نظر بگیرید.

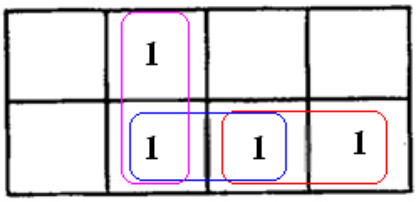




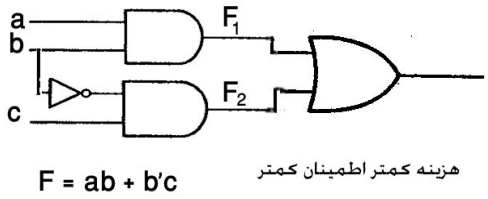
با توجه به نمودار زمانی مشاهده می‌شود که مدار مخاطره ایستای سطح یک دارد.

چه موقع در مدار مخاطره داریم و چگونه می‌توانیم آن را رفع کنیم؟

برای تشخیص مخاطره جدول کارنوی عملکرد مدار را می‌کشیم اگر یک PI وجود داشته باشد که در دو همسایگی تکرار شده باشد و در ساده‌سازی در دسته بندی قرار نگرفته باشد امکان مخاطره وجود دارد.برای رفع مخاطره باید همه‌PIها را نیز در دسته-بندی بگیریم.

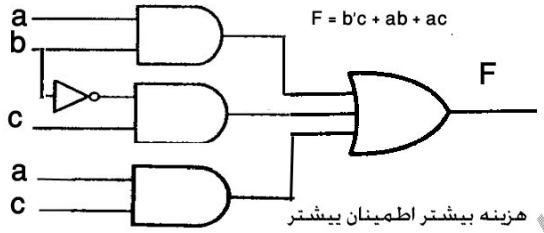


مخاطره بین پنج و هفت اتفاق می افتد.



F = ab + b'c

هزینه کمتر اطمینان کمتر

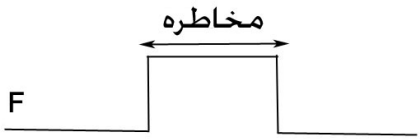


هزینه بیشتر اطمینان بیشتر

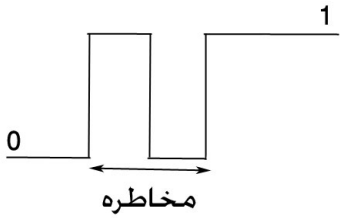
مدار دوم بدون مخاطره است پس اگر بخواهیم مخاطره انفاق نیافتدباید همه PI ها را در نظر بگیریم.

در مدار اول تابع مدار F=ab+b'c می باشد که با توجه به جدول کارنو می‌بینیم یک PI وجود دارد که در دسته بندی در نظر گرفته نشده است (مینترم 5و7). بنابراین مخاطره با تغییر از 5 به 7 و بالعکس اتفاق می‌افتد برای حذف مخاطره تابع را بصورت F=b'c+ab+ac در نظر می گیریم. تابع دوم کاملاً ساده نشده است اما مخاطره هم ندارد.

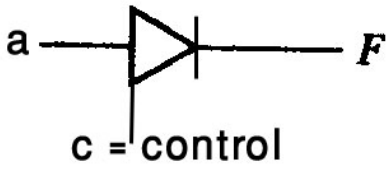
2- مخاطره ایستای سطح صفر: باید خروجی مدار صفر باشد ولی در یک زمان کوتاه خروجی به یک تغییر پیدا می‌کند.



3-مخاطره پویا: ممکن است چند پالس صفر و یک اشتباه جواب دهد.



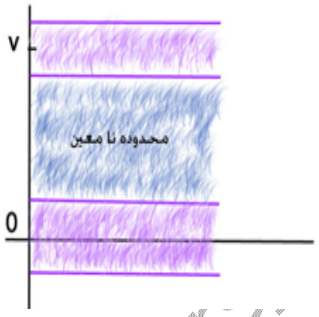
گیت بافر سه حالته:



اگر c = 1 باشد، خروجی =a

اگر c = 0 باشد، خروجی = هیچی (hize, high impedance).

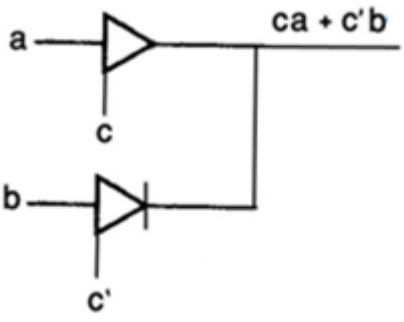
خروجی یک گیت می تواند بصورت نمودار زیر باشد اگر ولتاژ V معادل 1 باشد یا بک دامنه تغییرات d و ولتاژ 0 با دامنه تغییرات d2 معادل بیت 0 باشد آنگاه فاصله بین این دو ولتاژ نامعین است یعنی نه صفر است و نه یک.که در محدوده نامعین قرار می گیرد.

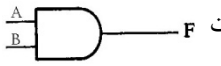


آیا خروجی یک بافر سه حالته برابر با یک گیت and می باشد؟ پاسخ خیر. زیرا اگر c = 1 باشد، آنگاه در گیت and خروجی c.a=0 ولی اگر c=0 باشد خروجی c.a≠0 خواهد بود.

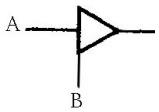
نکته: خروجی گیت‌های بافر سه حالته را با سیم به هم متصل می‌کنند.

معمولاً کنترل دو گیت بافر سه حالته not هم می‌باشند. (در گیت‌های بافر همواره باید رعایت شود.)

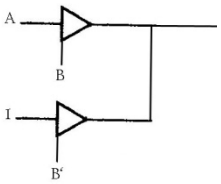


مثال) کدام گزینه معادل گیت  می باشد؟

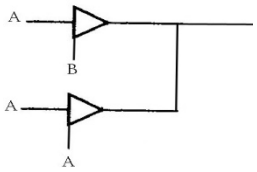
الف)



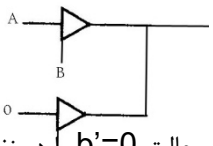
ب)



ج)

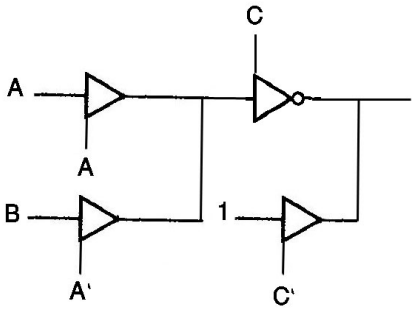


د)



جواب: گزینه الف اصلاً بافر نیست. گزینه ج نکته فوق را رعایت نکرده است. بین گزینه ب و د باید حالت b'=0 را در نظر گرفت به طوری که خروجی جواب داشته باشد. پس گزینه د صحیح می باشد.

مثال) مدار زیر چه تابعی را پیاده می‌کند؟



الف) (A' + B') C'

ب) A'B' + C'

ج) ((A'B' + C')'

د) (A' – B') C'

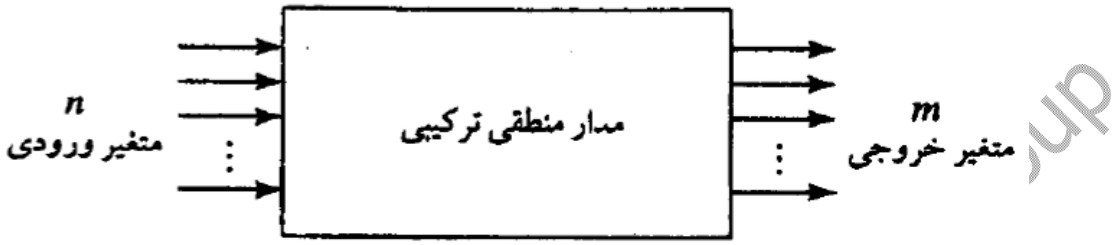
[(A + A'B) C]' + C' = [(A)' (A+B') + C'] + C' = A'A + A'B' + C' + C' =A'B' + C'

جواب : گزینه دو

فصل4

طراحی مدارات ترکیبی (Combinational Circuit Design)

مداراتی می‌باشندکه خروجی‌ها فقط به ورودی‌ها وابسته است.



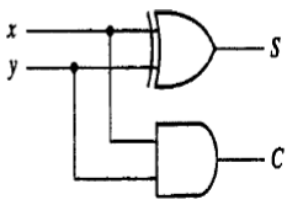
برای طراحی یک مدار باید مراحل زیر را انجام داد:

1- دریافت صورت مسئله
2- مشخص کردن تعداد ورودی‌ها و خروجی‌ها
3- مشخص کردن رابطه بین ورودی‌ها و خروجی‌ها
4- ساده‌سازی‌مدار
5- پیاده‌سازی مدار(کشیدن شکل مدار)

مدارنیم جمع کننده (Half Adder=HA): می خواهیم دو بیت a,b را با هم جمع کنیم.

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

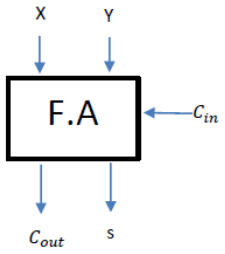
$S=X'Y+XY'=X \oplus Y$ $C=XY$



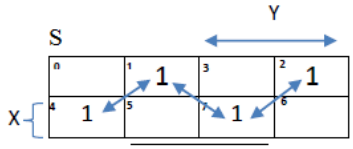
مدار تمام جمع کننده (Full Adder=FA): مدار نیم جمع کننده دو تک بیت را جمع می کند ولی تمام جمع کننده تو تک بیت و یک بیت نقلی که از طبقه قبل آمدا است را با هم جمع می کند.

$$\begin{array}{r} C_2\ C_1C_0 \\ +\ X_3X_2X_1\ X_0 \\ \underline{Y_3Y_2\ Y_1\ Y_0} \\ C_{OUT}\ S_3S_2\ S_1\ S_0 \end{array}$$

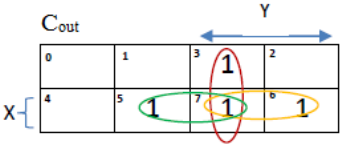
x	y	C_{in}	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



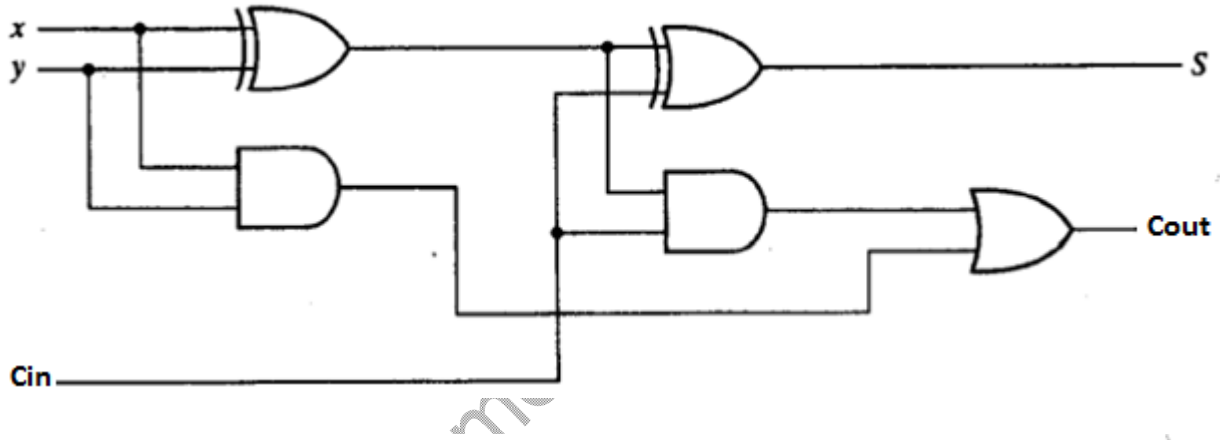
X	Y	$X+Y$	$X \oplus Y$
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	0



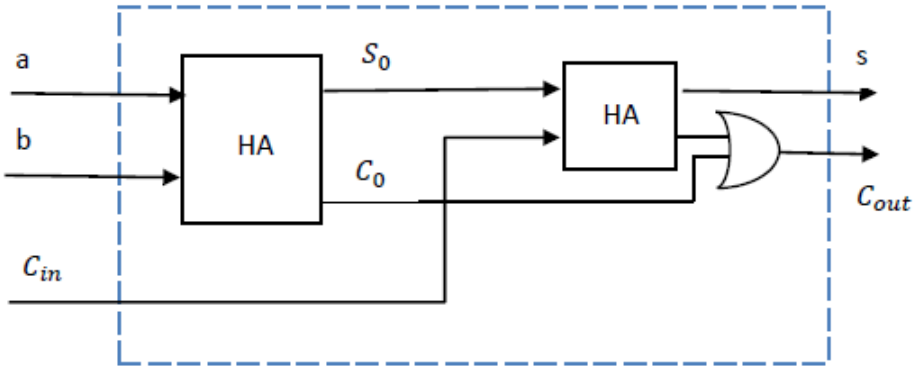
$S = X \oplus Y \oplus C_{in}$



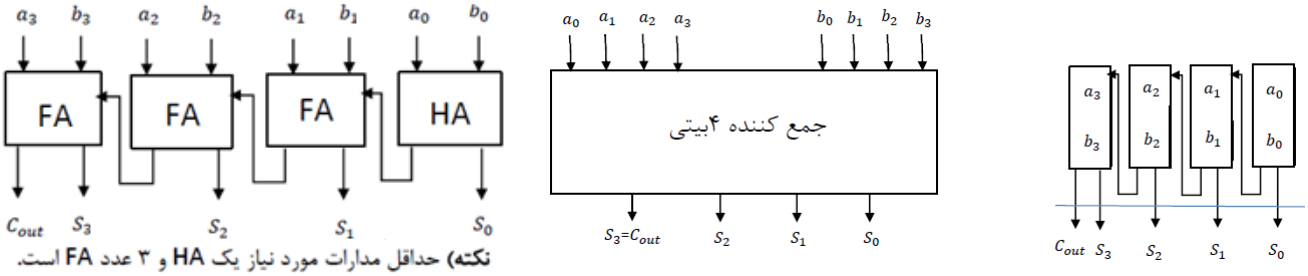
$$C_{out} = X.C_{in} + Y.C_{in} + XY = C_{in}(X+Y) + XY = C_{in}(X \oplus Y) + XY$$



طراحی FA توسط دو HA و یک گیت OR دو ورودی



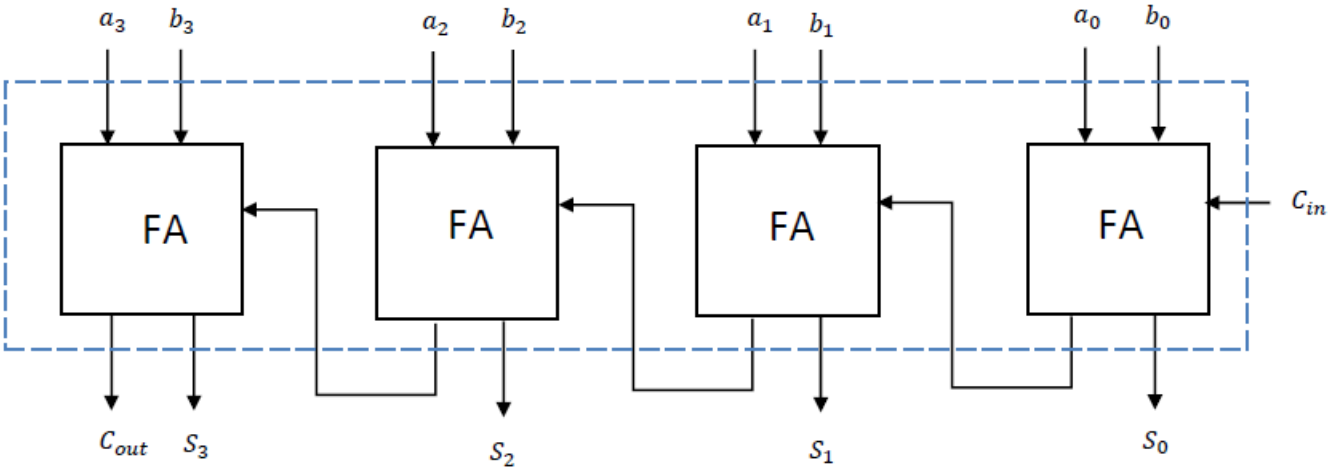
مسئله) به تعداد کافی FA و HA داریم. با استفاده از آنها یک جمع کننده ۴بیتی طراحی کنید.



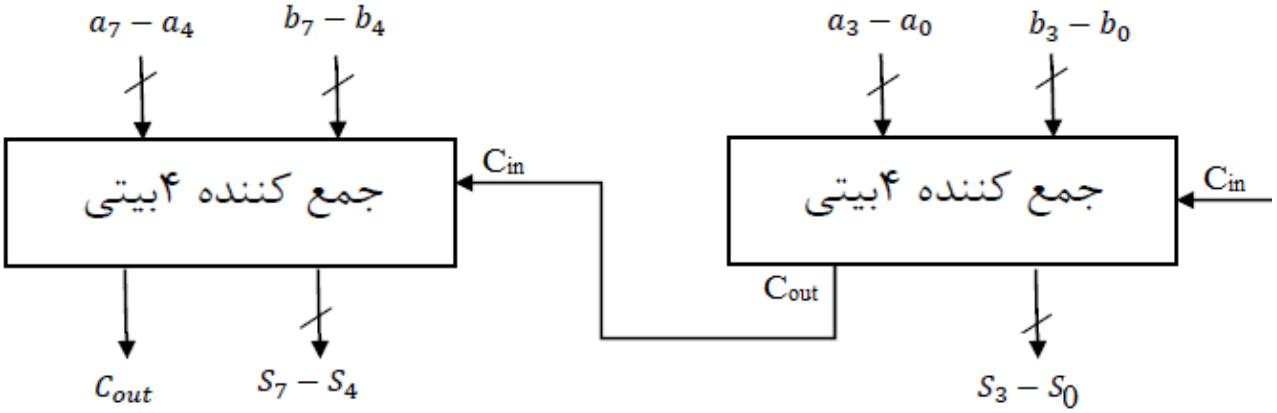
نکته) حداقل مدارات مورد نیاز یک HA و ۳ عدد FA است.

نکته) حداقل مدارات مورد نیاز برای یک جمع کننده nبیتی توسط FA و HA، یک HA و n-1 عدد FA است.

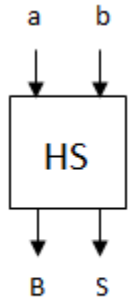
این مدار قابل گسترش نیست برای قابل گسترش شدن HA را نیز با FA تعویض می‌کنیم.



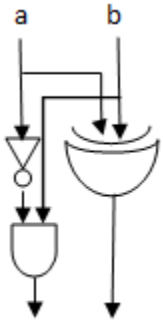
مسئله) توسط جمع کننده ۴بیتی جمع کننده ۸بیتی بسازید.



نیم تفریق گر(Half Subtractor):



a	b	B	S
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

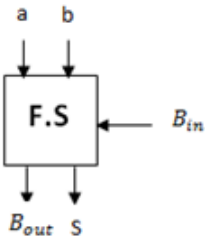


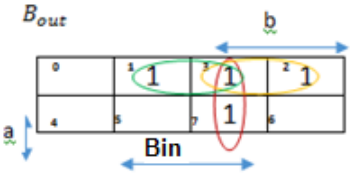
$S = a \oplus b$ $B = a'b$

تمام تفریق گر(Full Subtractor) :

a	b	B_{in}	B_{out}	S
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

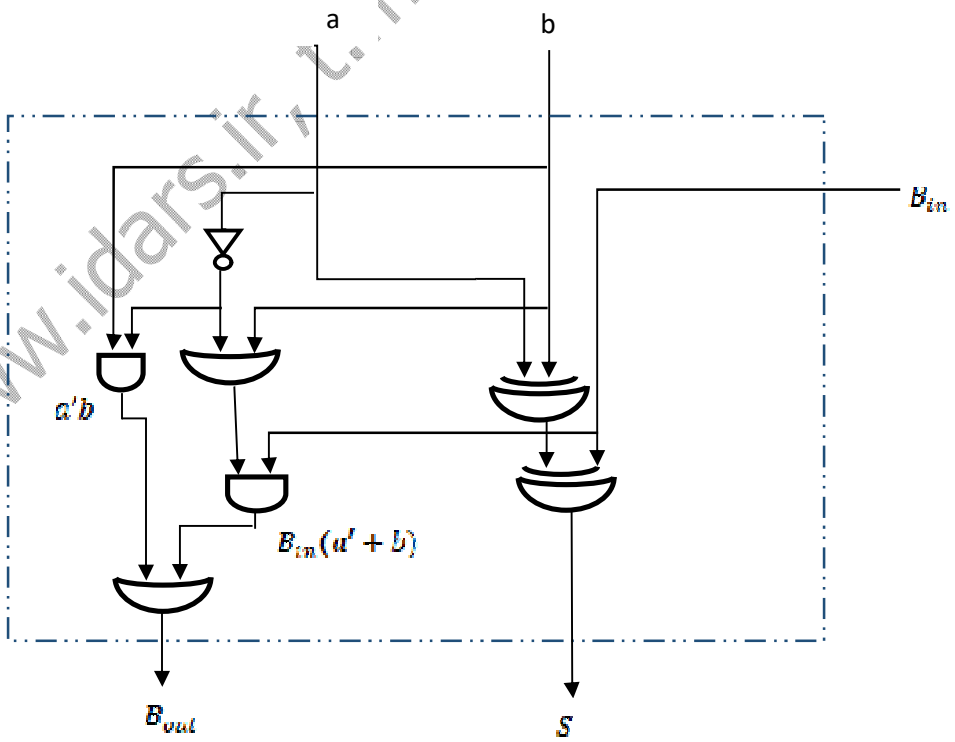
$S = a \oplus b \oplus B_{in}$





$B_{out} = a'B_{in} + bB_{in} + a'b = B_{in}(a' + b) + a'b$

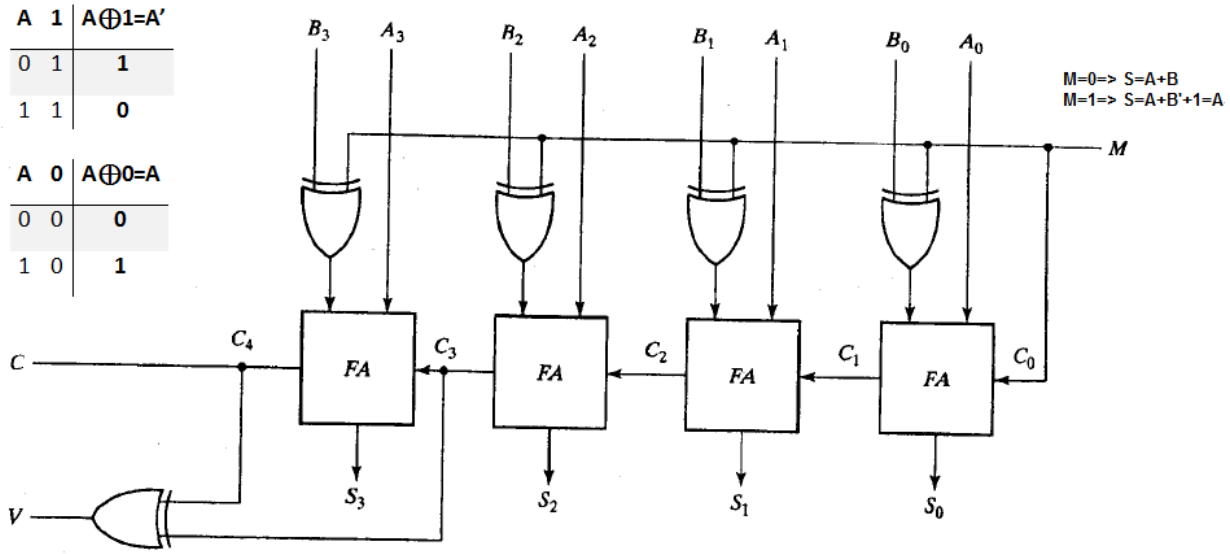
$= B_{in}(a' \oplus b) + a'b$



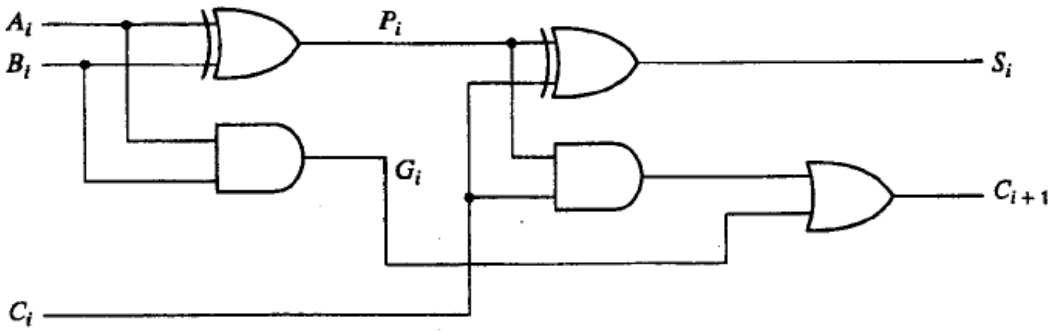
مدار جمع کننده و تفریق کننده:

A	1	A⊕1=A′
0	1	1
1	1	0

A	0	A⊕0=A
0	0	0
1	0	1



اگر هر FA تاخیر t_{FA} داشته باشد خروجی S_3 و C_{out} بعد از $4t_{FA}$ بدست می‌آید. چون FA ها باهم سری هستند. هر FA دو طبقه گیت دارد برای محاسبه S. برای محاسبه c سه طبقه دارد.



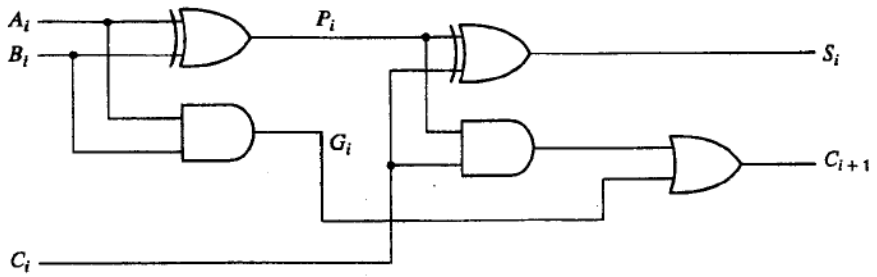
اگر تاخیر هر FA،2ns باشد تاخیر یک جمع‌کننده 4بیتی چقدر است؟

t_{FA}=2*2ns=4ns

4t=4*4ns=16ns=تاخیر کل

تاخیر یک جمع‌کننده nبیتی $n*t_{FA}$ یا 2nt می‌باشد که t تاخیر گیت می‌باشد. بنابراین برای n بزرگ تاخیر مدار زیاد است.

مدار جمع‌کننده با پیش بینی نقلی:



می‌خواهیم با حذف سری نقلی‌ها سرعت مدار جمع‌کننده را زیاد کنیم.

=Generate G_i (تولیدکننده نقلی) =propagate , Gi= $a.b$ p_i (انتشار نقلی) Pi= $a \oplus_i b_i$

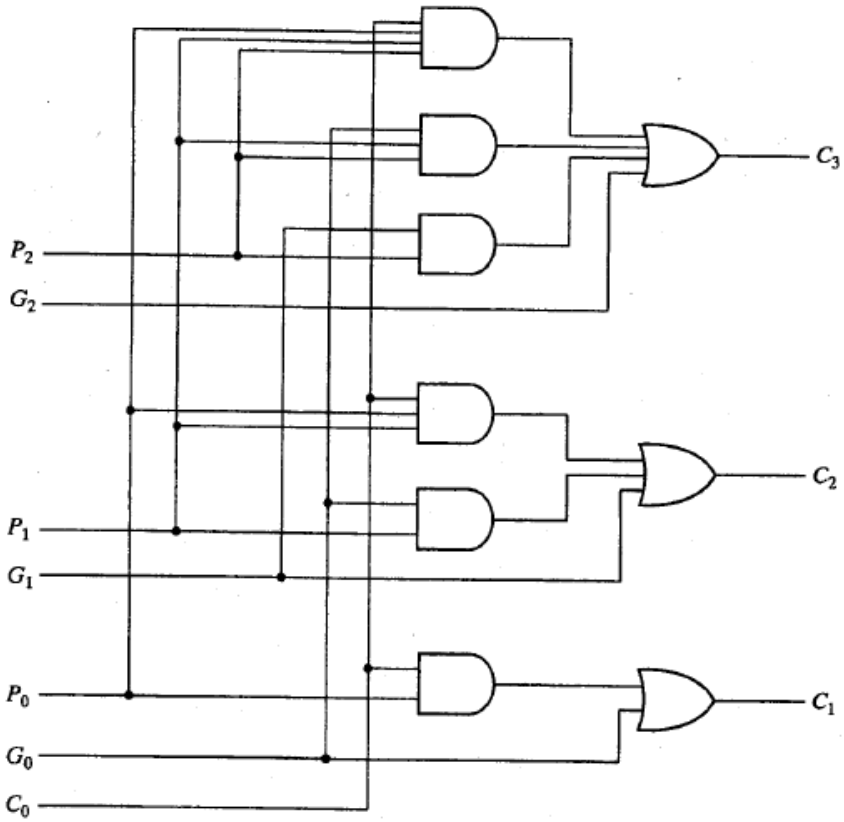
$S_i = p_i \oplus C_i,\ C_{i+1} = p_i.C_i + G_i$

C_0 =نقلی ورودی

$C_1 = G_0 + P_0C_0$

$C_2 = G_1 + P_1C_1 = G_1 + P_1(G_0 + P_0C_0) = G_1 + P_1G_0 + P_1P_0C_0$

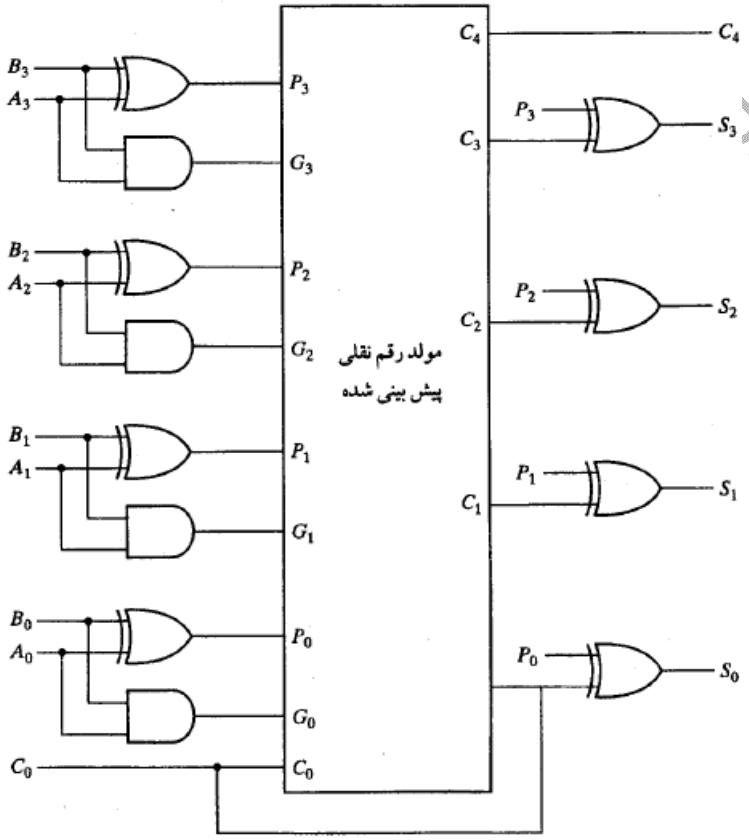
$C_3 = G_2 + P_2C_2 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0$



نکته) در مدار پیش بینی نقلی محاسبه هر نقلی در دو طبقه گیت اتفاق می‌افتد برای محاسبه C_i نیازمند i عدد گیت می‌باشیم و تعداد ورودی گیت OR و AND حداکثر i است.

نکته) اگر بدون در نظر گرفتن تعداد ورودی های گیت‌ها تاخیر گیت AND و OR را t در نظر بگیریم محاسبه همه carryها 2t طول می‌کشد.

مثال) از مدار پیش بینی نقلی استفاده کنید و یک جمع کننده 4بیتی بسازید.



با توجه به شکل بدون توجه به ورودی‌ها خروجی مدار جمع کننده با 4طبقه گیت محاسبه می‌شود در صورتیکه تاخیر AND-OR و XOR باهم برابر باشد.

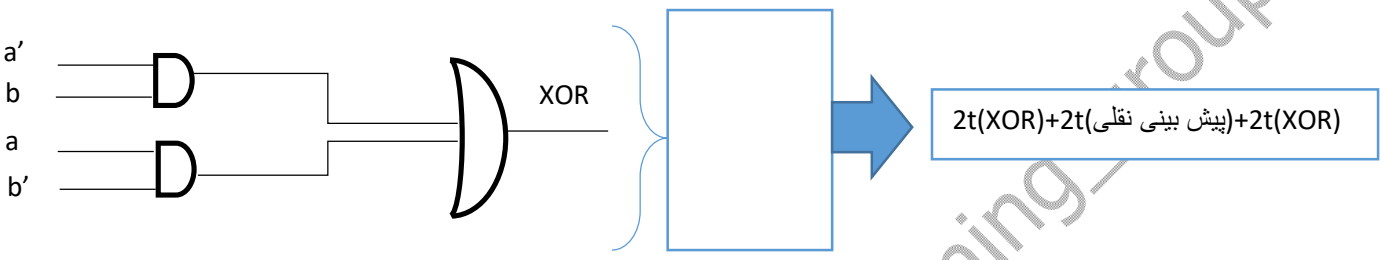
تست) فرض کنید تاخیر گیت‌های ANDوORوXOR برابر t است تاخیر یک جمع کننده 4بیتی با پیش‌بینی نقلی کدام گزینه است؟

4t(1) 6t(2) 8t(3) 16t(4

تست) فرض کنید تاخیر گیت AND و OR بدون توجه به تعداد ورودی‌ها t باشد و XOR نیز از AND و OR بدست آید و تاخیر گیت NOR صفر است. تاخیر یک جمع‌کننده 4بیتی با پیش‌بینی نقلی کدام است؟

4t((1 6t(2) 8t(3 16t(4

$$a \oplus b = a'b + ab'$$



جمع‌کننده BCD) : می خواهیم دو عدد A,B را که BCD می باشند با یکدیگر جمع کنیم و نتیجه جمع نیز BCD باشد برای این عمل از مدار جمع کننده چهار بیتی باینری استفاده می کنیم و نتیجه آن را به BCD تبدیل می کنیم.

اعداد A و B صفر تا 9 می‌باشند. حداکثر جمع دوعدد BCD، 19 می‌باشد. $A+B+C_{in}=9+9+1=19$

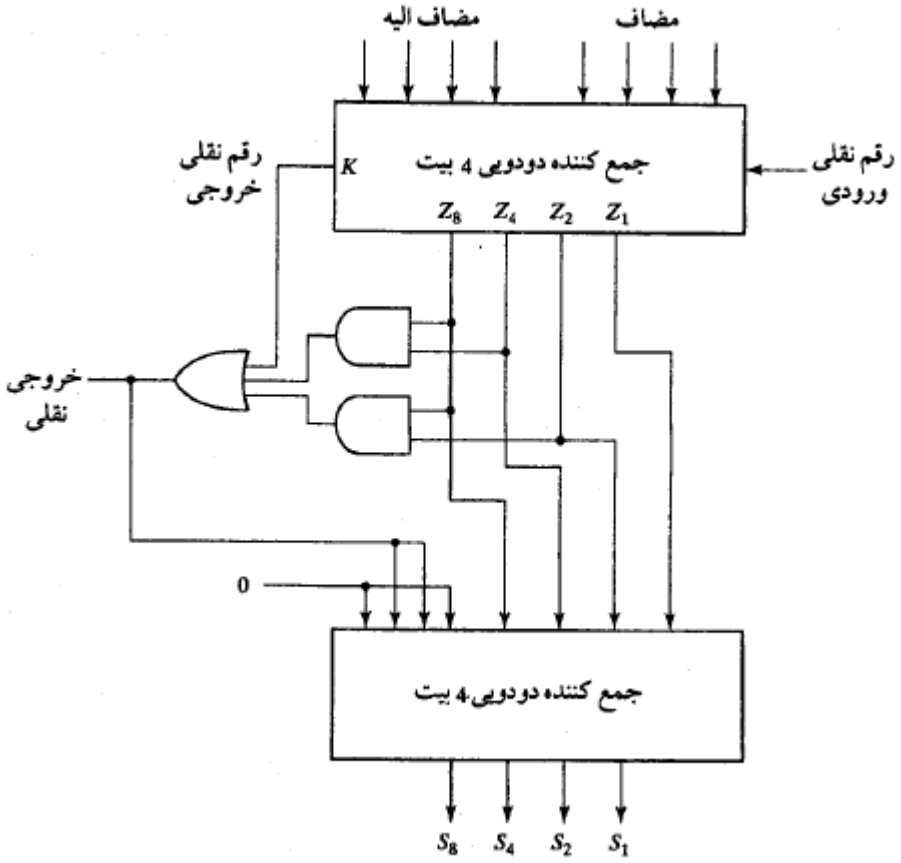
خروجی تولید شده توسط جمع کننده باینری را $Z_8Z_4Z_2Z_1$ نام گذاری می کنیم و خروجی BCD را $CS_8S_4S_2S_1$ می نامیم رابطه آنها بصورت زیر است.

جمع دودویی					جمع BCD					دهدهی
K	Z_8	Z_4	Z_2	Z_1	C	S_8	S_4	S_2	S_1	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19

$$C_{out}=K+Z_8Z_4+Z_8Z_2$$

اگر حاصل جمع باینری 0 تا 9 باشد، حاصل BCD نیز همان است. اعداد 10 به بعد در باینری با BCD، 6واحد تفاوت دارند.

اگر عدد >=10 ➡ عدد BCD = عدد باینری +6(0110)



تبدیل کدها :

برای طراحی مدارهای تبدیل کد باید عملیات زیر را انجام داد.

1. کد ورودی و خروجی را در یک جدول صحت می نویسیم.
2. با استفاده از جدول کارنو به ازای هر خروجی عمل ساده سازی را بر اساس ورودی ها انجام می دهیم.
3. شکل مدار را می کشیم .

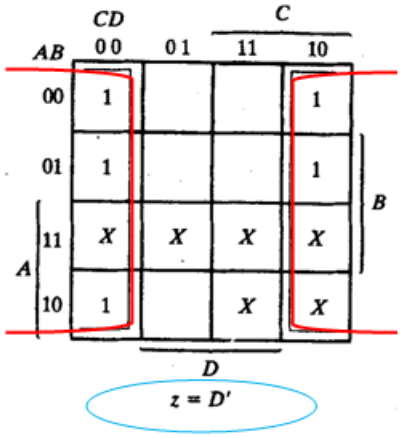
مثال : مداری طراحی کنید که کد BCD را به افزونی3 تبدیل کند .

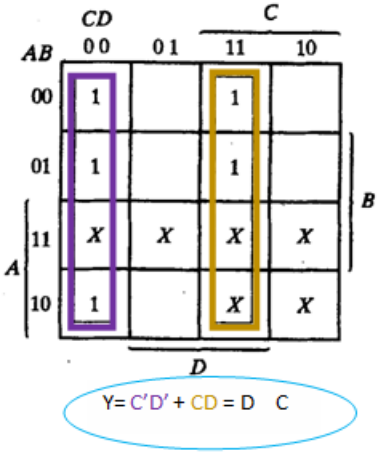
1 - (جدول صحت)

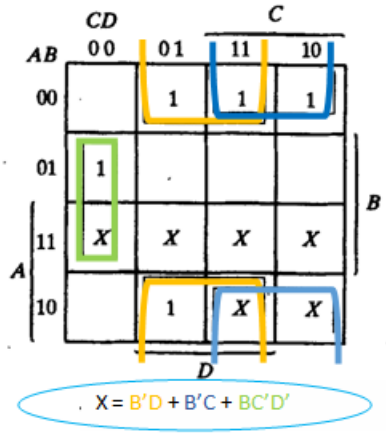
ورودی BCD				خروجی کد افزونی - 3			
A	B	C	D	w	x	y	z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

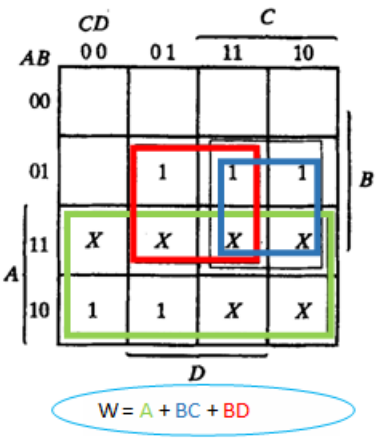
نکته : جدول ورودی تا 9 است .10 تا 15 بی اهمیت می باشد.

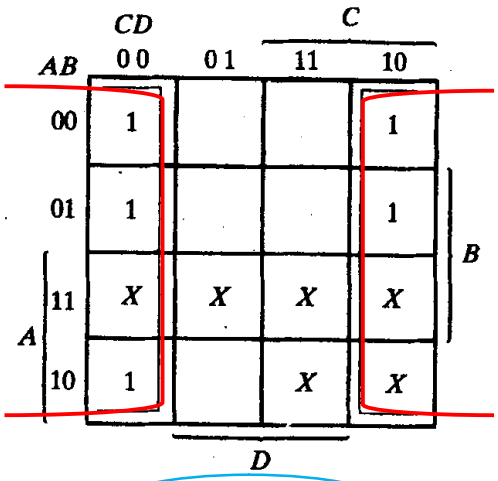
2- (جدول کارنو)



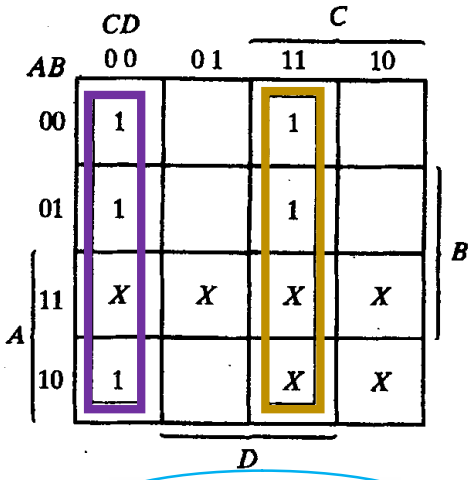




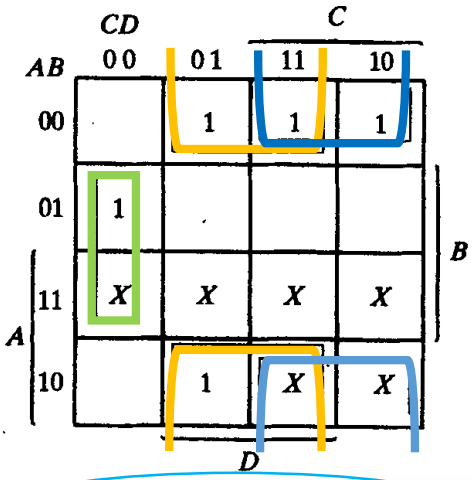




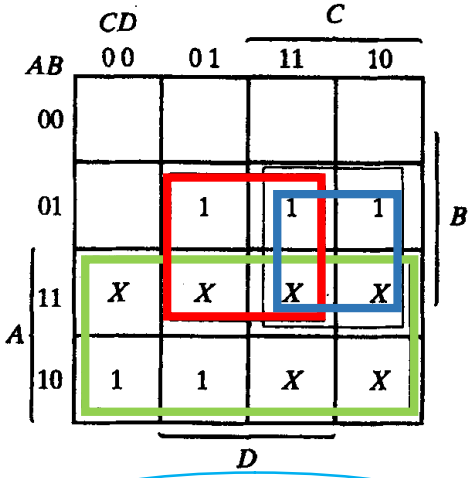
$z = D'$



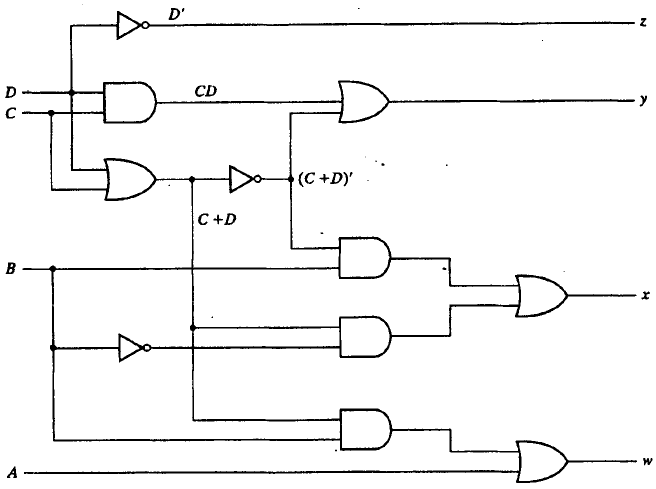
Y= C'D' + CD = D C



X = B'D + B'C + BC'D'

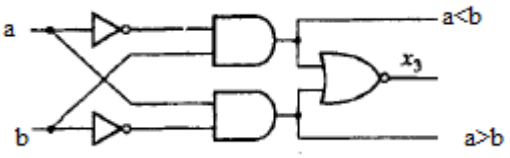


W = A + BC + BD



مقایسه گر مقدار: می خواهیم دو بیت a و b را با هم مقایسه کنیم. جدول درستی مقایسه a,b بصورت زیر است.

a	b	a=b=a'b'+ab	a<b=a'b	a>b=ab'	
0	0	1	0	0	
0	1	0	1	0	
1	0	0	0	1	
1	1	1	0	0	



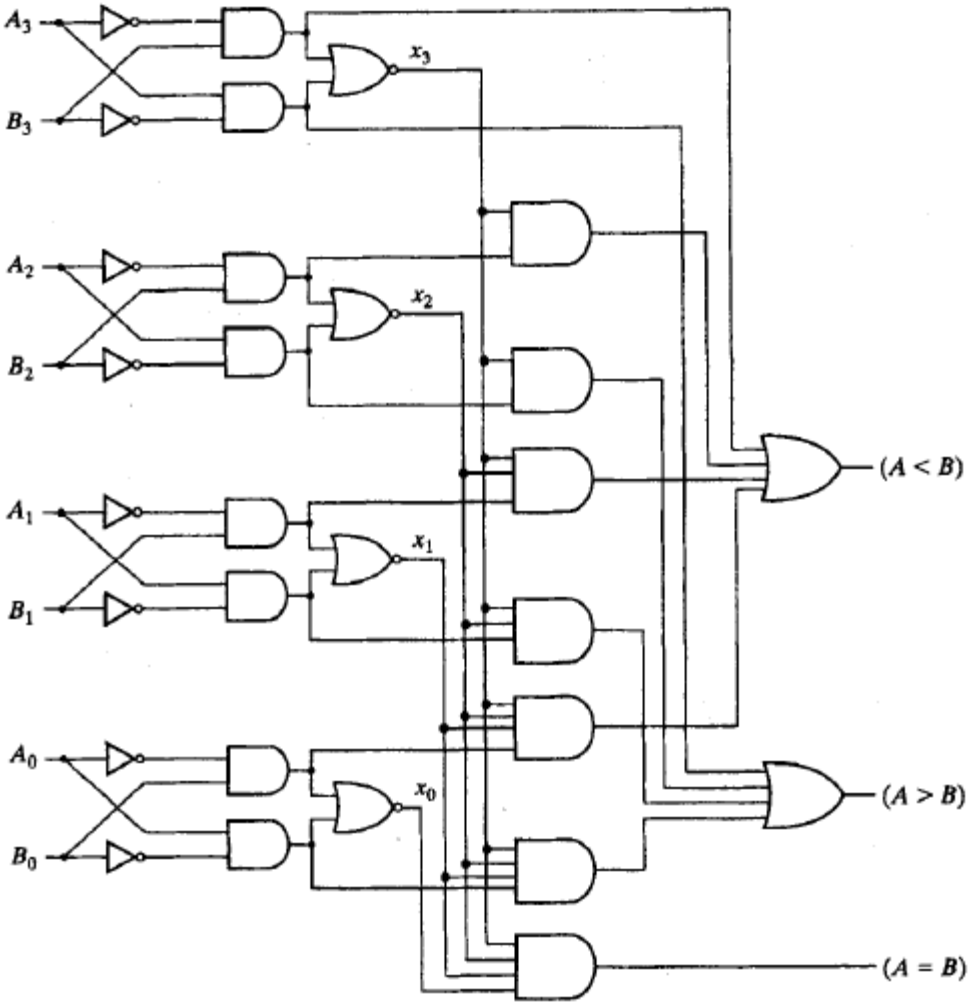
حال می خواهیم دو عدد 4 بیتی A=A3A2A1A0 را با B=B4B2B1B0 مقایسه کنیم اگر بخواهیم جدول صحت این دو داده را رسم کنیم شامل 256 حالت خواهد شد که زمانبر است برای بدست آوردن عبارت مقایسه استفاده می کنیم.

هنگامی A=B می باشد که (A3=B3).(A2=B2).(A1=B1).(A0=B0).

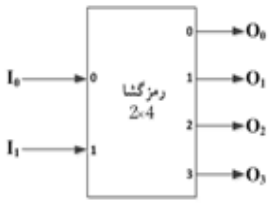
هنگامی A>B که A3>B3+ (A3=B3).(A2>B2)+(A3=B3)(A2=B2).(A1>B1)+(A3=B3).(A2=B2).(A1=B1).(A0>B0)

هنگامی A<B که A3<B3+ (A3=B3).(A2<B2)+(A3=B3)(A2=B2).(A1<B1)+(A3=B3).(A2=B2).(A1=B1).(A0<B0)

بنابراین مدار مقایسه گر مقدار 4 بیتی بصورت زیر است:

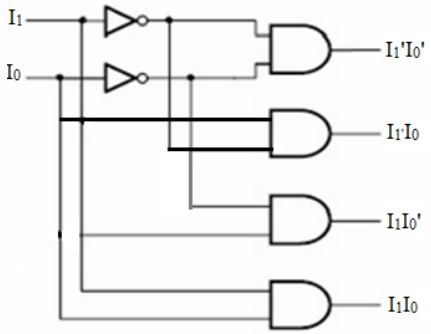


دیکدر (Decoder): مداری است با n ورودی و 2^n خروجی. به ازای مقدار روی ورودی ها، خروجی متناظر با آن را یک می کند و سایر خروجی ها صفر است . یا به عبارتی مینترم ها را تولید می کند . مثلا برای دیکدر 2 به 4 داریم:



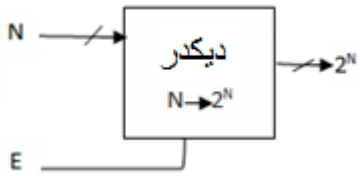
I_1	I_0	O_3	O_2	O_1	O_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

$$O_0=I_1'I_0' \quad O_1=I_1'I_0 \quad O_2=I_1I_0' \quad O_3=I_1I_0$$

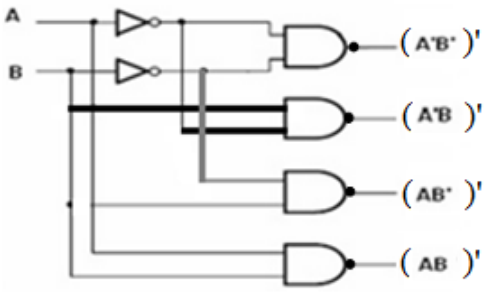


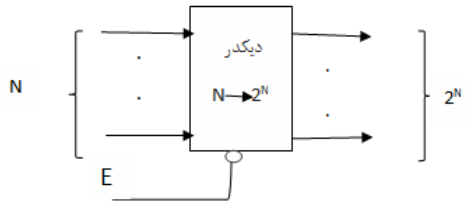
❖ نکته: یک دیکدر $N\rightarrow 2^N$نیازمند 2^N گیت and ، N ورودی است .

دیکدر بدون توانا ساز همیشه در حال فعالیت است اگر بخواهیم دیکدر را کنترل کنیم یک ورودی E(Enable) به دیکدر اضافه می‌کنیم. هرگاه E=0 کل خروجی‌ها صفر خواهند بود. هرگاه E=1دیکدربطور عادی کار خواهد کرد.



❖ نکته : اگر گیت‌های AND به NAND تبدیل شوند به جای تولید مینترم، ماکسترم تولید خواهد شد.



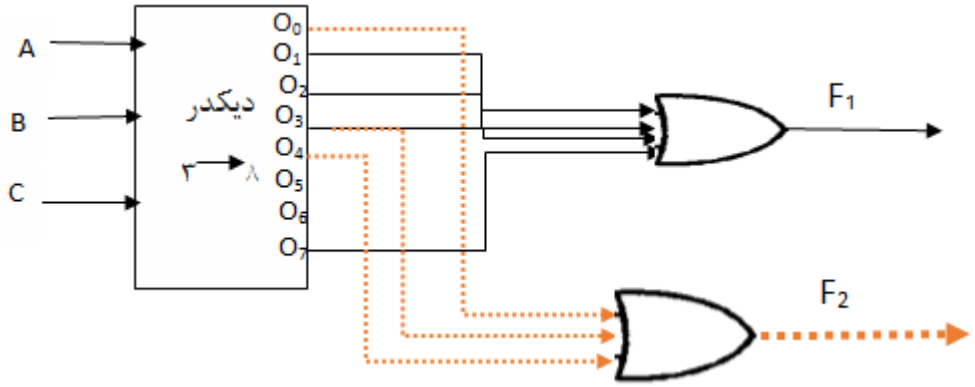


وقتی E=0 باشد کار می کند

وقتی E=1 باشد کار نمی کند

هر تابعی را می توان با استفاده از دیکدر طراحی کرد به این منظور کافی است خروجی منترم های تابع را با یک گیت (OR) به هم وصل کنیم.

✓ مثال : $F(A,B,C)=\Sigma(1,2,3,7)$ و $F_2(A,B,C)=\Sigma(0,3,4)$ را با دیکدر طراحی کنید.



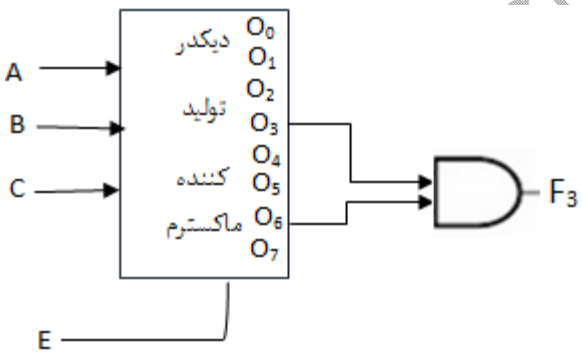
در طراحی با دیکدر اگر تعداد مینترم ها بیشتر از نصف خروجی بود می توانیم عبارت را به جای (SOP) با (POS) نوشته و با گیت NOR طراحی می کنیم.

مثال :$F_3\ (A,B,C)=\Sigma(0,1,2,4,5,7)$ را با دیکدر طراحی کنید.

برای طراحی باید یک گیت OR، 6 ورودی طراحی کنیم.

اما می دانیم.$\Sigma(0,1,2,4,5,7)=\pi(3,6)$

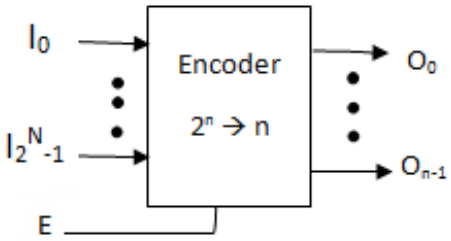
که با یک AND دو ورودی قابل انجام است.



انکدر (Encoder)(رمزگذار):

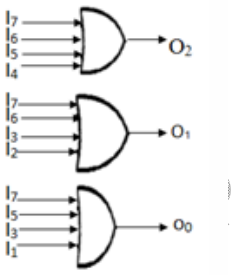
❖ بدون اولویت:

همیشه یکی از ورودی ها یک است (فرض) بر اساس مقدار ورودی که یک است. خروجی متناظر آن ظاهر خواهد شد



✓ مثال: انکدر 8→3

ورودی‌ها								خروجی ها		
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	O_2	O_1	O_0
1	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0



$O_2 = I_7 + I_6 + I_5 + I_4$ $O_1 = I_7 + I_6 + I_3 + I_2$ $O_0 = I_7 + I_5 + I_3 + I_1$

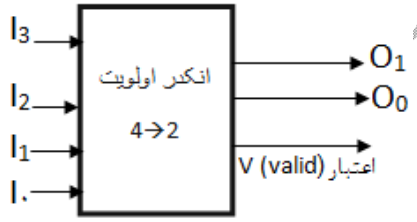
در مدار فوق فرض کردیم که فقط یک ورودی یک است اگر بیشتر از یک ورودی یک شود جوابی که روی خروجی ها ایجاد خواهد شد ممکن است غلط شود.

به عنوان مثال : اگر I_3 وI_4 هر دو یک شوند خروجی 111 خواهد شد و بیانگر هیچکدام از ورودی ها نیست.

❖ انکدر با اولویت:

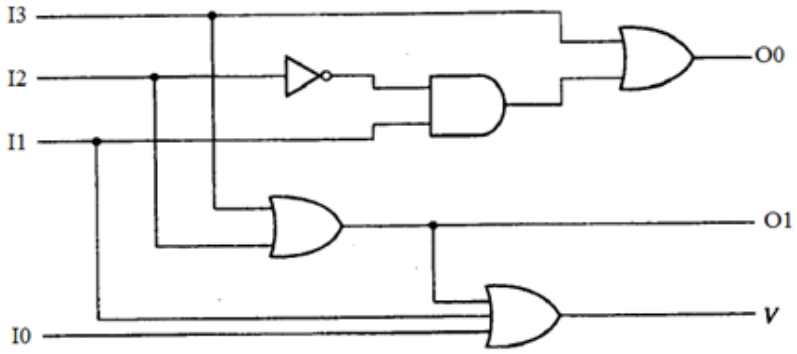
می‌خواهیم مدار را طوری اصلاح کنیم که همیشه درست جواب دهد و اولویت را به ورودی بزرگتر بدهیم.

✓ مثال : یک انکدر اولویت 4 به 2 طراحی کنید.



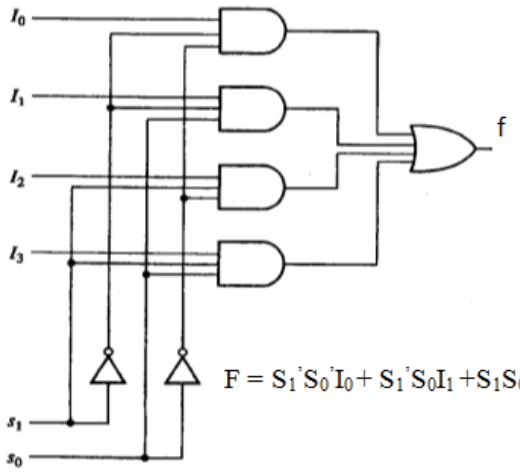
I_3	I_2	I_1	I_0	O_1	O_0	V
0	0	0	0	x	x	0
1	x	x	x	1	1	1
0	1	x	x	1	0	1
0	0	1	x	0	1	1
0	0	0	1	0	0	1

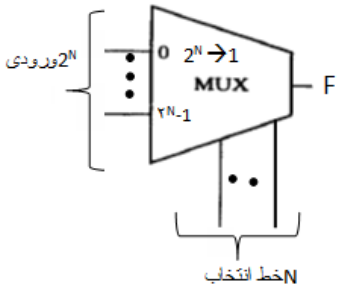
$O_1 = I_3 + I_3'I_2 = I_3 + I_2$ $O_0 = I_3 + I_3'I_2'I_1 = I_3 + I_2'I_1$ $V = I_3 + I_2 + I_1 + I_0$



❖ مالتی پلکسر: (Multiplxer):(MUX):انتخاب گر، تسهیم ساز :

مداری است که دارای N ورودی انتخاب و 2^N ورودی و یک خروجی می باشد. بر اساس مقدار ورودی های انتخاب، ورودی مربوط به خروجی منتقل می شود.



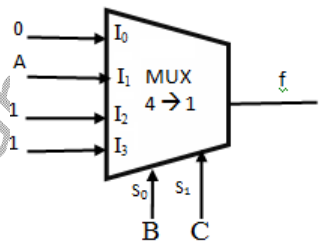


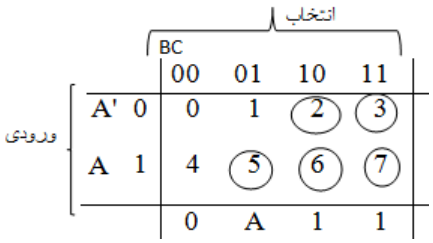
S_1	S_0	f
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

- پیاده سازی توابع با مالتی پلکسر :

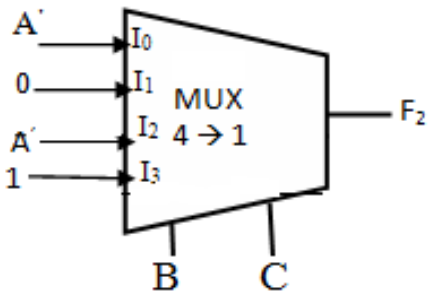
هر تابعی را می‌توان با مالتی پلکسر پیاده‌سازی کرد اگر تابع n ورودی داشته باشد (mux) انتخاب شده دارای (n-1 یا کمتر) ورودی انتخاب خواهد داشت در این صورت n-1 ورودی یا کمتر، ورودی را روی خط انتخاب قرار می‌دهیم و باقیمانده ورودی‌ها را روی ورودی تقسیم می‌کنیم.

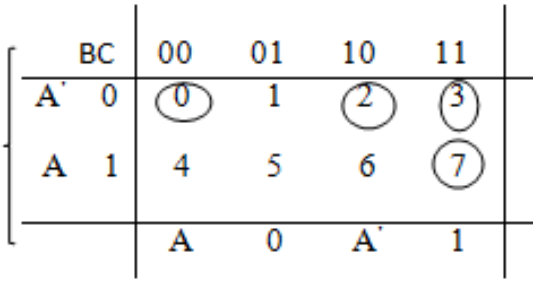
✓ مثال : تابع $f(a,b,c)=\Sigma(2,3,5,6,7)$را با استفاده از mux طراحی کنید .



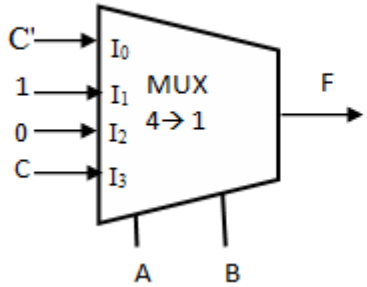


مثال : $f_2(a,b,c) = \Sigma(0,2,3,7)$ را با مالتی پلکسر پیاده سازی کنید(bوc را روی select قرار دهید).



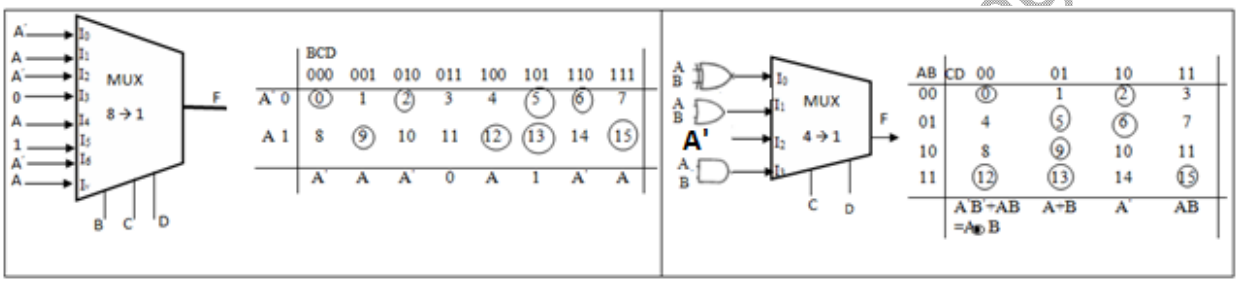


مثال : مثال فوق را با قراردادن AB رو انتخاب حل کنید.

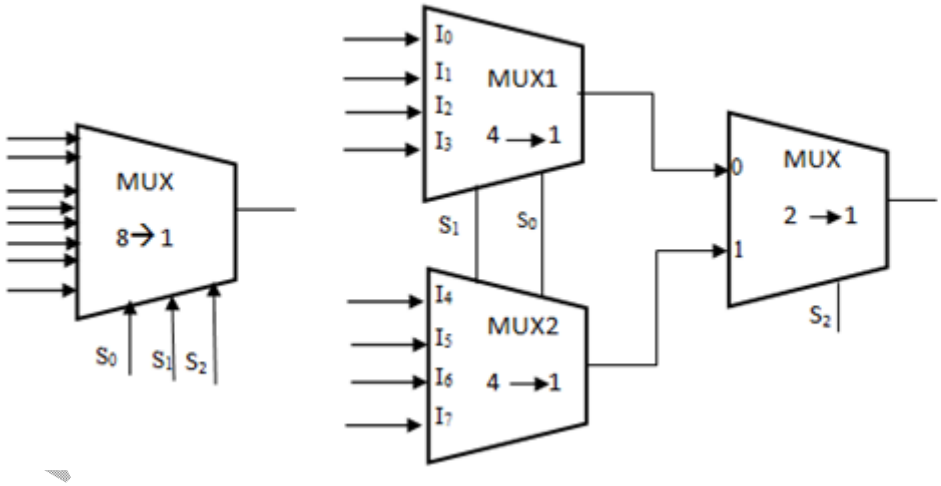


	AB	00	01	10	11
C'	0	0	2	4	6
C	1	1	3	5	7
		C'	1	0	C

✓ مثال : تابع $F_3(A,B,C,D) = \Sigma(0,2,5,6,9,12,13,15)$ را یک بار با مالتی پلکسر 8به 1و یک بار با مالتی پلکسر 4به1 طراحی کنید .



✓ مثال : به تعداد کافی مالتی پلکسر 4 به 1 و 2 به 1 داریم با استفاده از آن ها مالتی پلکسر 8 به 1 بسازید.

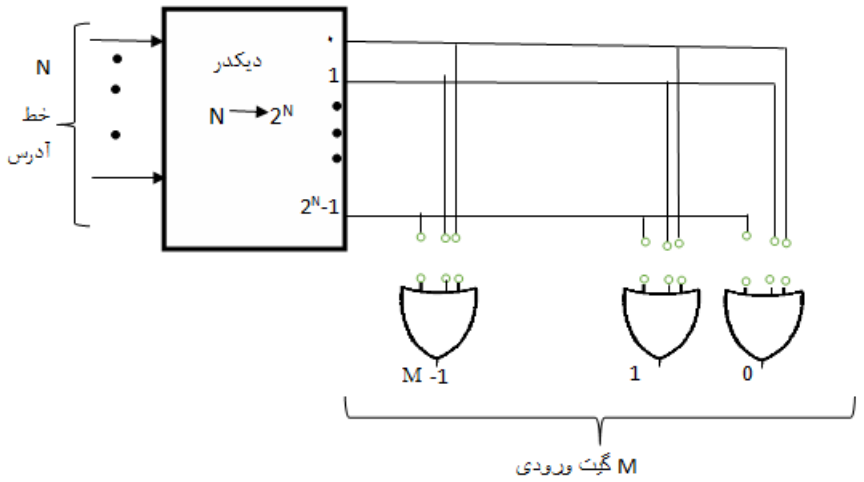


S_2	S_1	S_0	
0	0	0	
0	0	1	
0	1	0	MUX1
0	1	1	
1	0	0	
1	0	1	MUX2
1	1	0	
1	1	1	

طراحی حافظه ROM(Read Only Memory):

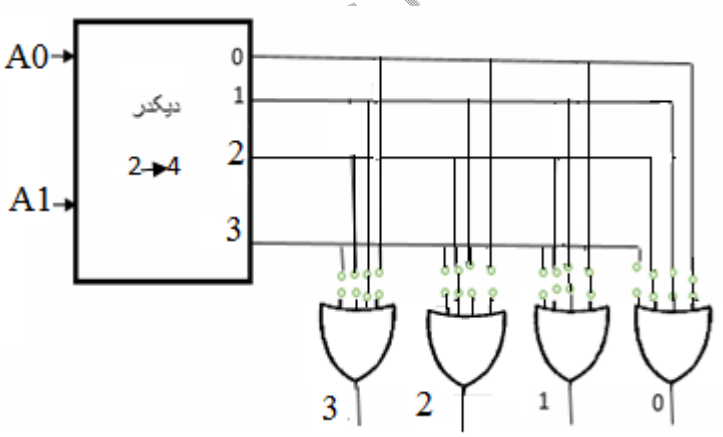
هر حافظه nخط آدرس و 2^n خانه حافظه، هر خانه حافظه نیز m بیت است.(4،8،16،32،64،128)برای حافظه ROM از یک دیکدر N به 2^N استفاده می شود و از M گیتOR که دارای 2^N ورودی می‌باشد. برای ساخت خانه های حافظه استفاده می‌شود. 2^N ورودی گیت OR به صورت فیوز می‌باشد. یعنی اتصالی برقرار نیست و وقتی محتوای ROM مشخص شد فیوزها متصل می شوند.

شکل کلی حافظه ROM



گیت های or برنامه پذیر می باشند.

✓ مثال : یک حافظه ROM با 4 خانه 4 بیتی با محتوی زیر طراحی کنید.

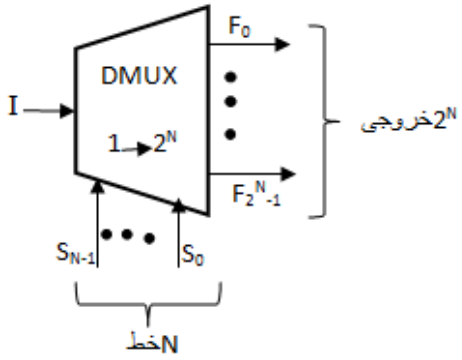


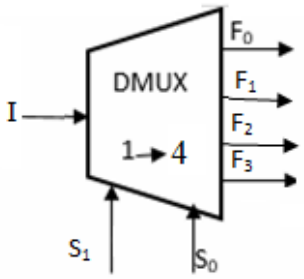
A_1	A_0	M_3	M_2	M_1	M_0
0	0	0	0	1	1
0	1	1	0	1	1
1	0	0	1	0	1
1	1	0	0	0	0

در حافظه ROM گیت های AND ثابت و گیت های OR برنامه پذیر می باشد. اگر گیت های AND نیز قابل برنامه نویسی باشند به آن PLA می گویند و اگر گیت های OR ثابت باشند و گیت های AND برنامه پذیر باشند به آن PLA می گویند.

دی مالتی پلکسر (DMUX):

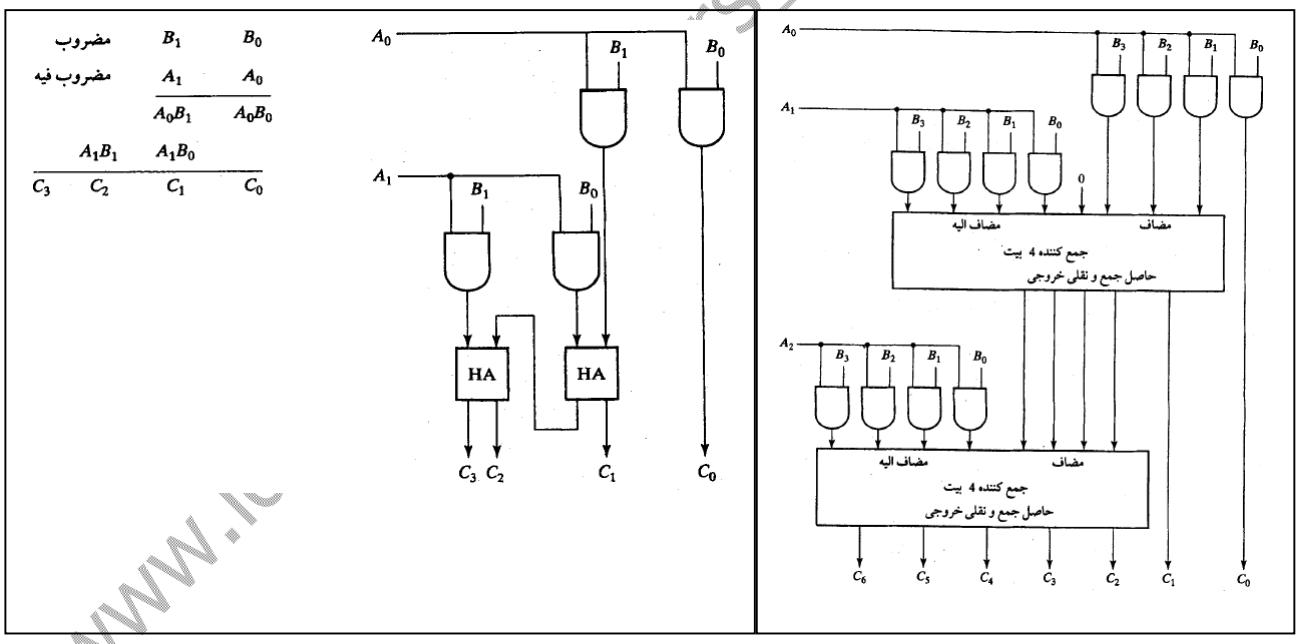
مداری است با 1 ورودی و 2^N خروجی که شامل N خط انتخاب است. بر اساس مقدار روی ورودی خطوط انتخاب ورودی به یکی از خروجی‌هامتصل می‌شود و به صورت شکل زیر نشان می‌دهیم.





S_1	S_0	F_0	F_1	F_2	F_3
0	0	I	0	0	0
0	1	0	I	0	0
1	0	0	0	I	0
1	1	0	0	0	I

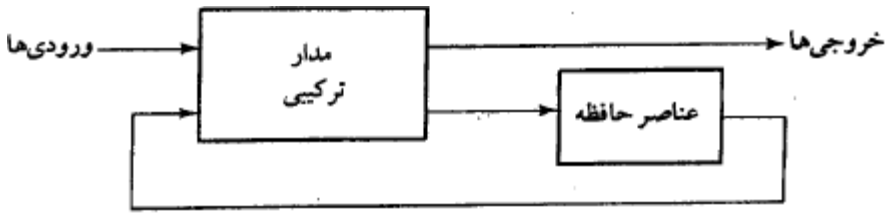
مدار ضرب کننده:



در کل برای ضرب یک مضروب n بیتی در یک مضروب فیه m بیتی نیازمند n-1 جمع کننده m بیتی و m*n گیت and دو ورودی هستیم.

فصل 5 : مدارات ترتیبی (Sequential Circuits)

درفصل4مدارات ترکیبی رابررسی نمودیم که در آنها خروجی فقط به ورودی وابسته بود. در مدارات ترتیبی، خروجی علاوه بر ورودی به خروجی(های) قبلی نیز وابسته است. به عبارتی حافظه دارد (وجود حافظه به این معناست که خروجی قبلی را نگه می دارد.) شکل کلی مدارات ترتیبی به صورت زیر می‌باشد:

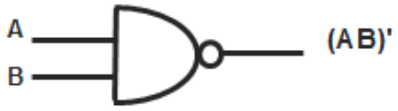


یادآوری گیتnor : هرگاه هر دو ورودی صفر باشند، خروجی یک است. اگر یکی از ورودی ها یک باشدخروجی صفر می شود.

A	B	$\overline{(A+B)}$
0	0	1
0	1	0
1	0	0
1	1	0

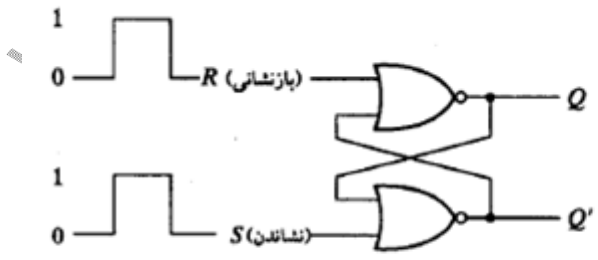
یادآوری گیتnand: هرگاه هر دو ورودی یک باشند، خروجی صفراست. اگر یکی از ورودی‌ها یک باشد خروجی صفر می‌شود.

A	B	(AB)'
۰	۰	۱
۰	۱	۱
۱	۰	۱
۱	۱	۰



اولین مدار ترتیبی که معرفی می‌کنیم فلیپ‌فلاپ پایه‌ی SR است.

فلیپ فلاپSR / SR flip-flop (Set reset) :



S	R	Q(t)	Q(t+۱)	Q'(t+۱)	
۰	۰	۰	۰	۱	
۰	۰	۱	۱	۰	
۰	۱	۰	۰	۱	
۰	۱	۱	۰	۱	
۱	۰	۰	۱	۰	
۱	۰	۱	۱	۰	
۱	۱	۰	۰	۰	نامعین
۱	۱	۱	۰	۰	نامعین

جدول صحت S-R

در فلیپ فلاپ SR، Q هم به R وS و هم به مقدارقبلی خودش وابست هاست.

*نکته 1:وقتی Sو R هر دو صفر هستند، Q(t) هر مقداری داشته باشد،Q(t+1) نیز همان مقدار را خواهد داشت.یعنی مقدار Q(t) حفظ می شود. Q(t+1) و Q'(t+1)، not یکدیگر می باشند.

*نکته 2: هرگاه R=1 و S=0، مهم نیست Q(t) چه باشد، Q(T+1)، Reset می شود یعنی مقدار Q(t+1) برابر با 0 می شود.

*نکته 3:هرگاه S=1 و R=0، مهم نیست Q(t) چه باشد،Q(t+1)، set می شود یعنی مقدار Q(t+1) برابر با 1 می شود.

*نکته 4: هرگاه هم S=1 و هم R=1، خروجی نامعین می شود، زیرا همان طور که گفته شد Q(t+1) و Q'(t+1) باید not یکدیگر باشند. ولی طبق جدول فلیپ فلاپ SR، مقدار مساوی دارند.

(برای این که حالت نامعین وجود نداشته باشد، در طراحی مدار با استفاده از فلیپ فلاپ SR، باید طوری عمل کنیم که هیچ گاه S R = 1 1 نشود.)

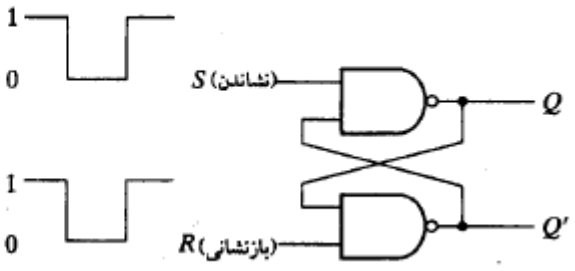
S	R	Q(t+۱)	
۰	۰	Q(t)	بدون تغییر
۰	۱	۰	Reset
۱	۰	۱	Set
۱	۱	نامعین	

جدول صحت فلیپ فلاپ SR

طراحی فلیپ فلاپ SR با گیت Nand :

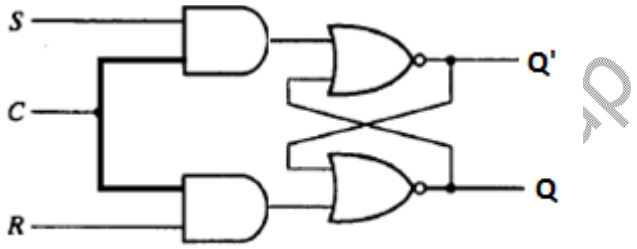
کافی است در مدار فلیپ فلاپ SR که با گیت Nor طراحی شده، جای Q و Q' را عوض کنیم و به جای گیت های nor از گیت های nand استفاده نماییم. (این مدار معادل همان مدار فلیپ فلاپ SR طراحی شده با گیت nor می باشد.)

S	R	Q(t+۱)
۰	۰	نامعین
۰	۱	۱
۱	۰	۰
۱	۱	Q(t)

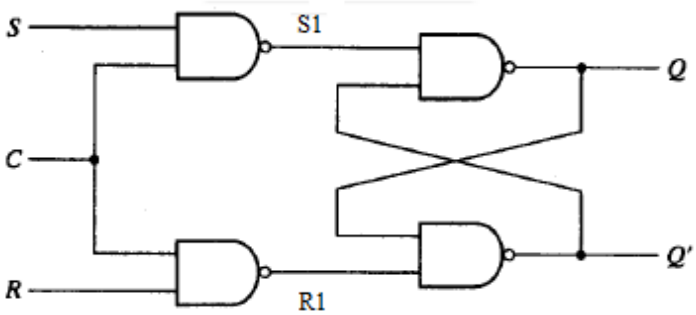


فلیپ فلاپ SR با ورودی کنترل clock

در فلیپ فلاپ SR، که با گیت nor یا nand طراحی شده است، با تغییر S و R خروجی Q تغییر خواهد کرد. می خواهیم کنترل فلیپ فلاپ را دقیق تر کنیم، به این معنی که هر گاه کنترل فعال شد، مدار کار کند.



با تغییر زیر مدار به گیت NAND تبدیل می شود:



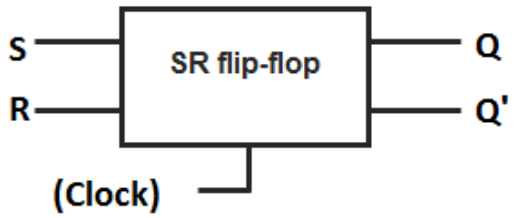
در مدار فوق، اگر C=0 آن گاه بدون توجه به مقادیر S و R، خروجی گیت Nand برابر یک خواهد بود و در نتیجه S1 و R1یک می شوند و خروجی بدون تغییر خواهد ماند.

اگر C=1، خروجی‌های گیت‌های nand تنها به مقادیر S و R وابسته است و مدار کار خودش را انجام می‌دهد.

C	S	R	Q(t+۱)
۰	۰	۰	Q(t)
۰	۰	۱	
۰	۱	۰	
۰	۱	۱	
۱	۰	۰	Q(t)
۱	۰	۱	۰
۱	۱	۰	۱
۱	۱	۱	نامعین

→

C	S	R	Q(t+۱)
۰	x	x	Q(t)
۱	۰	۰	Q(t)
۱	۰	۱	۰
۱	۱	۰	۱
۱	۱	۱	نامعین

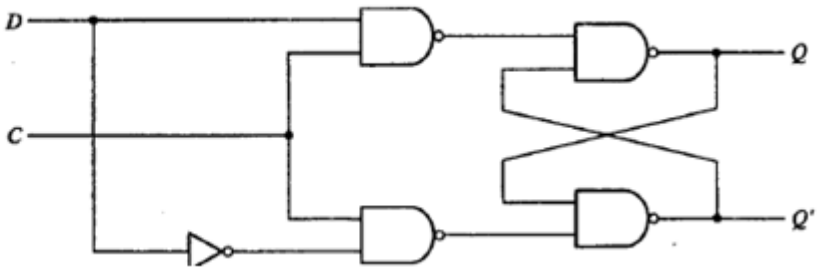


فلیپ فلاپ D / D flip-flop (flip-flop Data storing) :

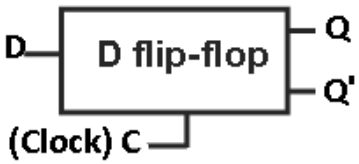
اگر D را به ورودی S فلیپ فلاپ و not آن را به ورودی R فلیپ فلاپ وصل کنیم، فلیپ فلاپ D ساخته می شود. در این فلیپ فلاپ مقدار D به Q(t+1) منتقل می شود. D=Q(t+1)

D	Q(t+۱)
۰	۰
۱	۱

جدول صحت فلیپ فلاپ D



فلیپ فلاپ D با ورودی کنترل clock: از این به بعد فلیپ فلاپ D را بطور خلاصه بصورت زیر نشان می‌دهیم.



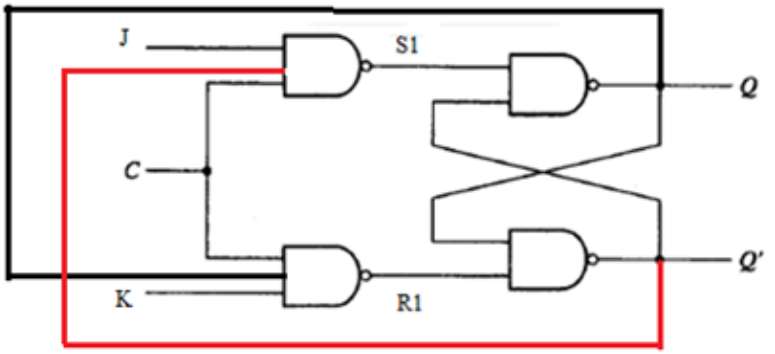
هنگامی که C=1، خروجی و ورودی با هم برابر می شوند.

فلیپ فلاپ JK / JK flip-flop :

به منظور رفع ایراد فلیپ فلاپ SR از فلیپ فلاپ JK استفاده می نماییم.

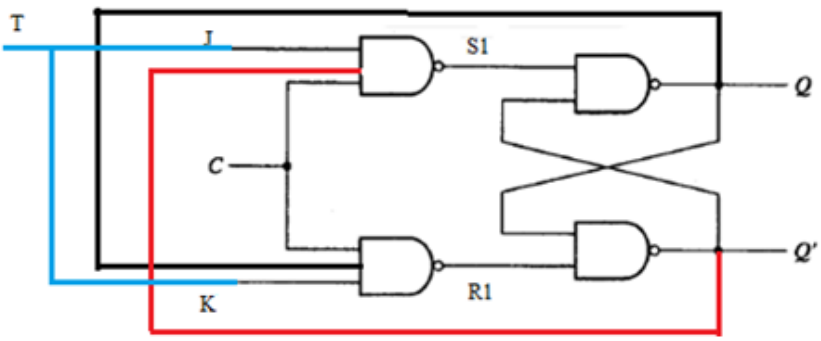
J	K	Q(t+۱)	
۰	۰	Q(t)	بدون تغییر
۰	۱	۰	Reset
۱	۰	۱	Set
۱	۱	Q'(t)	مکمل

جدول صحت فلیپ فلاپ JK



فلیپ فلاپ T / T flip-flop (Trigger):

فلیپ فلاپ T یا حافظه را نگه می دارد یا آن را not می کند. شبیه فلیپ فلاپ JK عمل می کند با این تفاوت که یک ورودی T به J و T' به K دارد.



T	Q(t+۱)
۰	Q(t)
۱	Q'(t)

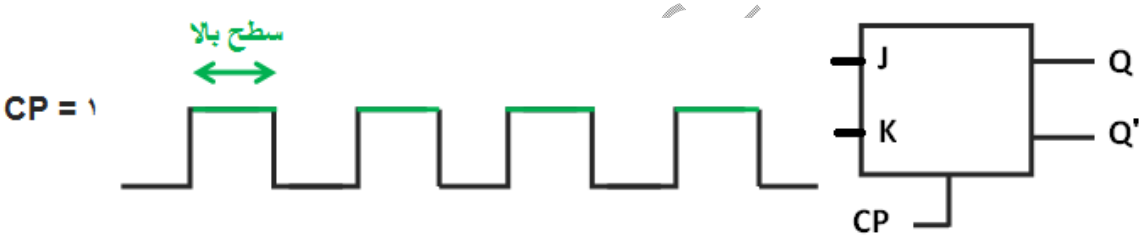
جدول صحت فلیپ فلاپ T

دسته بندی فلیپ فلاپ ها بر اساس کار کردن در نواحی مختلف پالس Clock :

پالس ساعت (level trigger flip-flop) پشت سر هم مثبت و منفی می شود. فلیپ فلاپ ها را بر اساس نوع clock، به 4 دسته تقسیم می کنیم.

مدارات حساس به سطح مثبت :

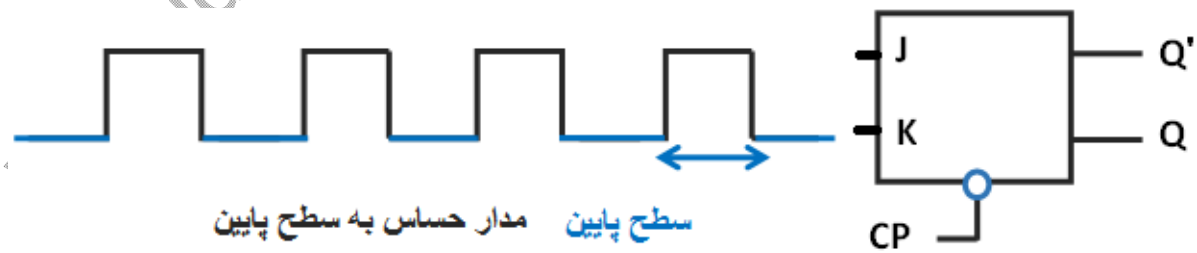
مدارات حساس به سطح مثبت زمانی کار می کنند که CP=1



مدار حساس به سطح بالا

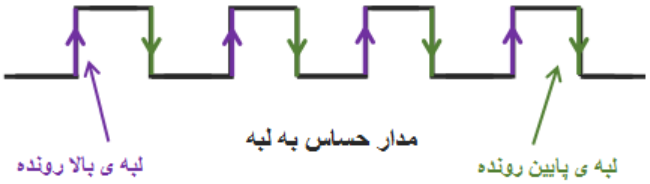
مدارات حساس به سطح منفی :

مدارات حساس به سطح مثبت زمانی کار می کنند که CP=0. اگر جلوی CP در مدار حساس به سطح مثبت، یک not بگذاریم، مدار حساس به سطح پایین خواهد شد.

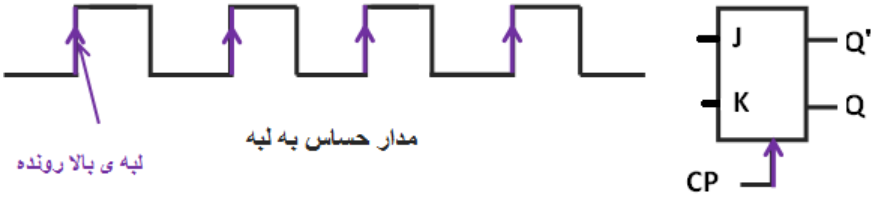


فلیپ فلاپ های حساس به لبه/ Edge trigger flip-flop:

فلیپ فلاپ هایی که فقط در یک زمان کوچک اجازه ی کار به مدار می دهند را فلیپ فلاپ های حساس به لبه یا(Edge trigger flip-flop) می گویند.

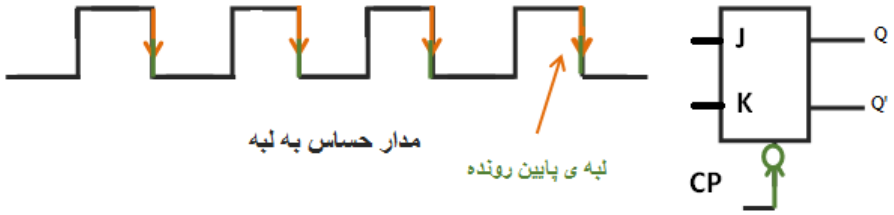


مدارات حساس به لبه ی بالارونده : زمان فعالیت این مدارات در لبه ی بالارونده می باشد.



مدارات حساس به لبه ی پایین رونده :

زمان فعالیت این مدارات در لبه ی پایین رونده می باشد. اگر جلوی CP در مدار حساس به سطح بالارونده، یک not بگذاریم، مدار حساس به سطح پایین رونده حاصل خواهد شد.



مدارات ترتیبی:

در مدارات ترتیبی دو مسأله مطرح می شود: مسأله ی اول تجزیه و تحلیل و مسأله ی دوم طراحی می باشد.در تجزیه و تحلیل مداری به ما داده می شود و عملکرد آن را باید تشخیص دهیم. ولی در طراحی باید با استفاده از فلیپ فلاپی خاص یک عمل مشخص را طراحی نماییم.

1-تجزیه و تحلیل:

در تجزیه و تحلیل ابتدا ورودی‌ها و خروجی‌ها را با توجه به صورت مسأله مشخص می‌کنیم و بر اساس جدول صحت حالات فعلی و بعدی را مشخص می‌کنیم و دیاگرام حال مدار را نیز می‌یابیم.

مثال: مداری با دو فلیپ فلاپ D و یک ورودی x و یک خروجی y داریم. اسامی فلیپ فلاپ ها را A و B در نظر بگیرید. عبارات مربوط به ورودی های D به صورت زیر است:

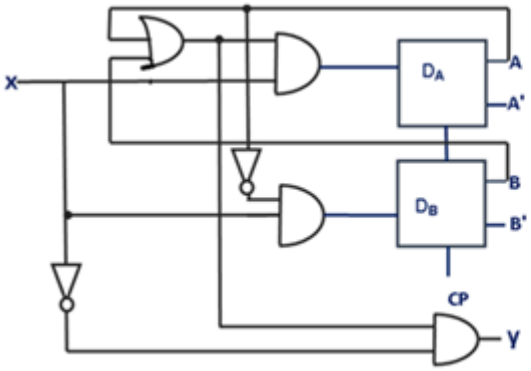
$D_A = A(t).x(t)+B(t).x(t)$

$D_B = A'(t).x(t)$

$y =[A(t)+B(t)].x'(t)$

1) شکل مدار را بکشید.

2) جدول صحت و دیاگرام حالت مدار را به دست آورید.



حالات فعلی			حالات بعدی		
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

$A(t+1)= D_A = A(t).x(t)+B(t).x(t)$

$B(t+1) = D_B = A'(t).x(t)$

$y = [A(t)+B(t)].x'(t)$

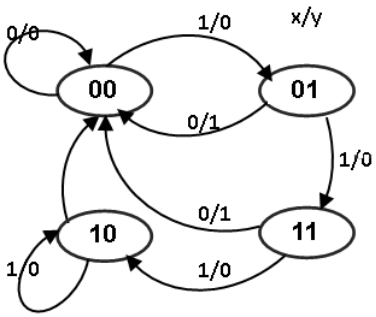
با توجه به جدول صحت فلیپ فلاپ می دانیم که $D=Q(t+1)$ و در نتیجه :

$D_A =A(t+1)=(A+B).x$ و $D_B =B(t+1)= A'.x$

A جاری را $A(t)$ می گویند و A یک لحظه بعد تر را $A(t+1)$ می گویند.

جدول صحت مدار:(تمامی ترکیبات ممکن از ورودیx و حالات فعلی مدار(A و B) را در سمت چپ جدول می نویسیم. A و B و y بعدی را با استفاده از رابطه ایکه در صورت سوال داده شده با توجه به حالات فعلی به دست می آوریم.)

دیاگرام حالت مدار: برای رسم دیاگرام حالت، حالات ممکن که در جدول صحت آمده را درون دوایری می نویسیم و با توجه به جدول صحت مدار می بینیم به ازای مقادیر ورودی و خروجی (در صورت وجود) از کدام حالت به حالت دیگر می رود. مقدار ورودی و خروجی را بالای فلش های وصل کننده ی حالتی به حالت دیگر می نویسیم.با توجه به دیاگرام حالت مدار،اگر x=0 یعنی ورودی x صفر باشد، آن گاه مدار در هر حالتی که باشد به 00 می‌رود.

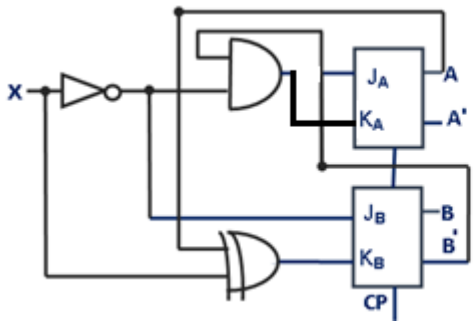


مثال: مداری با دو فلیپ فلاپ JK و یک ورودی x داریم. عبارات مربوط به ورودی فلیپ فلاپ ها به صورت زیر است:

$$\begin{cases} J_A = B \\ K_A = B.x' \end{cases} \qquad \begin{cases} J_B = x' \\ K_B = A.x' + A'x = A \oplus x \end{cases}$$

1) شکل مدار را بکشید.

2) جدول صحت و دیاگرام حالت مدار را به دست آورید.



حالات فعلی			حالت بعدی					
A	B	x	A	B	JA	KA	JB	KB
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0

J	K	Q(t+1)	
0	0	Q(t)	بدون تغییر
0	1	0	Reset
1	0	1	Set
1	1	Q'(t)	

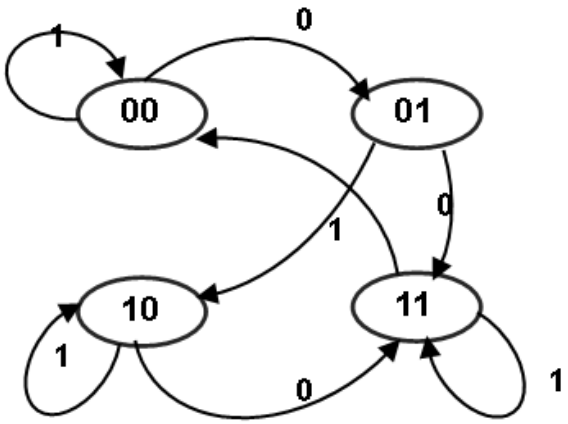
جدول صحت فلیپ فلاپ JK

*برای به دست آوردن جداول صحت فلیپ فلاپ ها به جز فلیپ فلاپ D، باید ورودی های فلیپ فلاپ ها را نیز به دست آوریم و بر اساس آن ها حالات بعدی را مشخص کنیم.

با استفاده از مقدار A و B و x و روابط داده شده در صورت سوال JAوKA وJB و KB را می یابیم. سپس با استفاده از جدول صحت فلیپ فلاپ JK، A و B بعدی را می یابیم.

به طور مثالهنگامی که A و B و X ورودی مقادیر 0و 1و 0 را دارند، با توجه به روابط صورت سوال، داریم:A= 1 KA=1 ,و KB = 0و1=JB

وقتی JA=1 و KA=1، A بعدی not حالت قبلی هستند. پس A بعدی صفر می‌شود. وقتی JB= 1 و KB=0، Bبعدی 1 می-شود. دیاگرام حالت مدار به صورت زیر می شود:



مدل‌های میلی و مور :

در دیاگرام حالت نوع میلی، خروجی به ورودی و حالت قبلی وابسته است. (مانند مثال1). در دیاگرام حالت نوع مور، خروجی فقط به حالت قبلی وابسته است.

طراحی با کمک فلیپ فلاپ‌ها:

برای طراحی ابتدا با استفاده از صورت مسأله، ورودی‌ها و خروجی‌ها و تعداد فلیپ‌فلاپ‌های مورد نیاز را مشخص می-کنیم.سپس با استفاده از حالات فعلی و بعدی و جدول تحریک فلیپ فلاپ‌ها، ورودی‌های آن‌ها را مشخص می‌کنیم و با استفاده از جدول کارنو ساده می‌کنیم و شکل مدار را می‌کشیم.

در جدول تحریک فلیپ فلاپ ها Q جاری و بعدی را داریم و ورودی ها را می یابیم.

Q(t)	Q(t+۱)	J	K
۰	۰	۰	X
۰	۱	۱	X
۱	۰	X	۱
۱	۱	X	۰

جدول تحریک فلیپ فلاپ JK

Q(t)	Q(t+۱)	S	R
۰	۰	۰	X
۰	۱	۱	۰
۱	۰	۰	۱
۱	۱	X	۰

جدول تحریک فلیپ فلاپ SR

Q(t)	Q(t+۱)	T
۰	۰	۰
۰	۱	۱
۱	۰	۱
۱	۱	۰

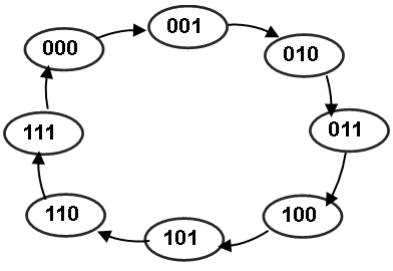
جدول تحریک فلیپ فلاپ T

Q(t)	Q(t+۱)	D
۰	۰	۰
۰	۱	۱
۱	۰	۰
۱	۱	۱

جدول تحریک فلیپ فلاپ D

مثال: با استفاده از فلیپ فلاپ T، مداری طراحی کنید که از 0 تا 7 را بشمارد.

برای اعداد 0 تا 7 حداقل به سه بیت احتیاج داریم. در نتیجه از 3 فلیپ فلاپ استفده می کنیم. *تعداد بیت های مورد نیاز طبق صورت مسأله = تعداد فلیپ فلاپ های مورد نیاز ورودی ندارد و خروجی تنها به حالت قبلی وابسته است. بنابراین مدل مور می باشد.



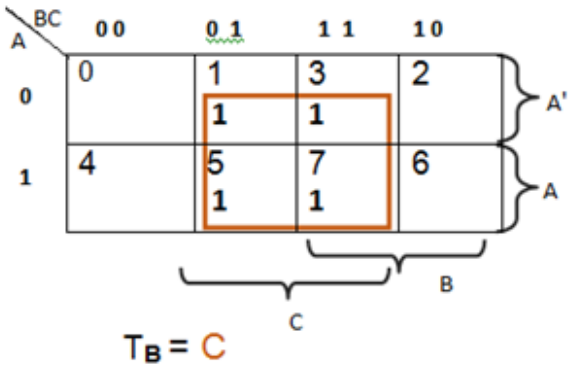
حالت قبلی و بعدی را داریم، T_A و T_B و T_C را می‌یابیم.

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

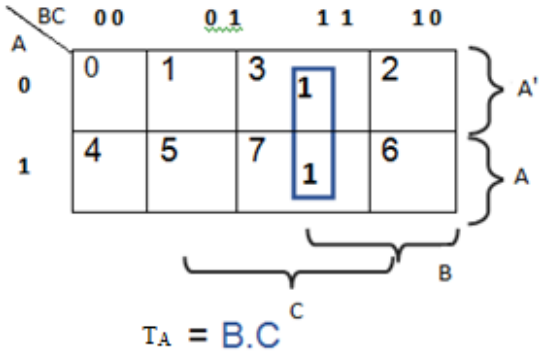
جدول تحریک فلیپ فلاپ T

حالات فعلی			حالات بعدی					
A	B	C	A	B	C	TA	TB	TC
0	0	0	0	0	1	·	0	1
0	0	1	0	1	0	·	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

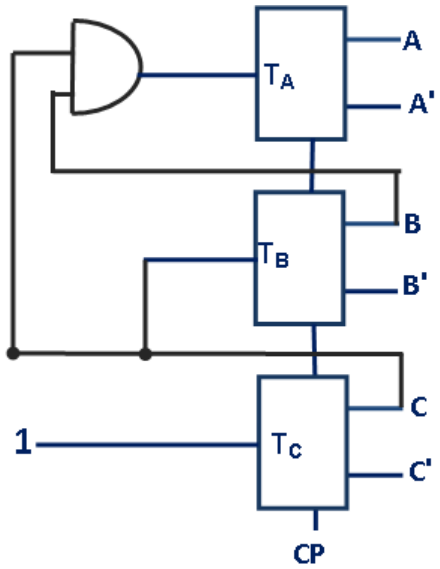
به طور مثال برای یافتن مقدار T_A در سطر نشان داده شده، مقدار A فعلی برابر با 1و A بعدی برابر با 0 است.طبق جدول تحریک فلیپ فلاپ ها، در این حالت T_Aبرابر با 1 می شود.مشخص است که $T_C=1$.با استفاده از جدول کارنو مقادیرT_A و T_B را بر حسبمقادیر فعلی A و B می یابیم.



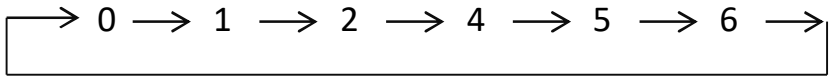
$T_B = C$



$T_A = B.C$



مثال: با استفاده از فلیپ فلاپ JK مداری طراحی کنید که دنباله ی زیر را بشمارد.



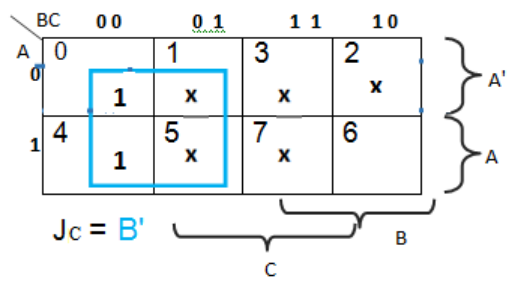
اعداد از 0 تا 6 هستند در نتیجه به سه فلیپ فلاپ نیاز داریم.شمارنده ترتیب شمار نیست، زیرا شماره ی 3 و 7 را ندارد.

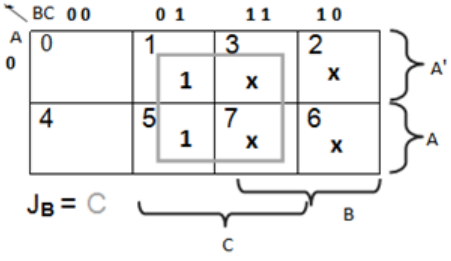
Q(t)	Q(t+1)	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

جدول تحریک فلیپ فلاپ JK

حالات فعلی			حالات بعدی								
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

با استفاده از جدول تحریک فلیپ فلاپ JK و داشتن مقادیر فعلی و بعدی A و B و C مقادیر ورودی های فلیپ فلاپها را می-یابیم و با استفاده از جدول کارنو، ورودی‌های فلیپ فلاپ‌ها را بر اساس مقادیر A و B و C فعلی می‌یابیم. جاهایی از جدول که x هستند، حکم don't care را دارند و برای دسته بندی استفاده می شوند.

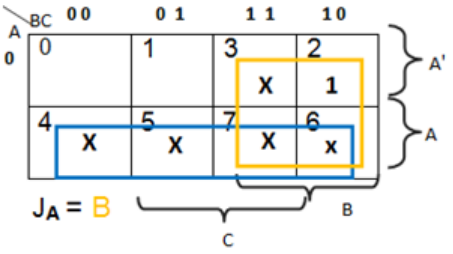


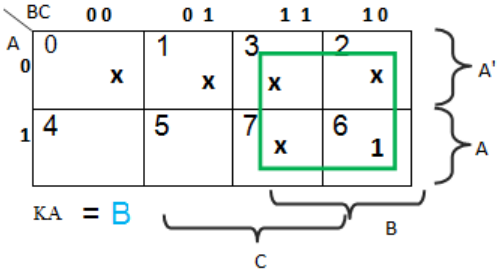


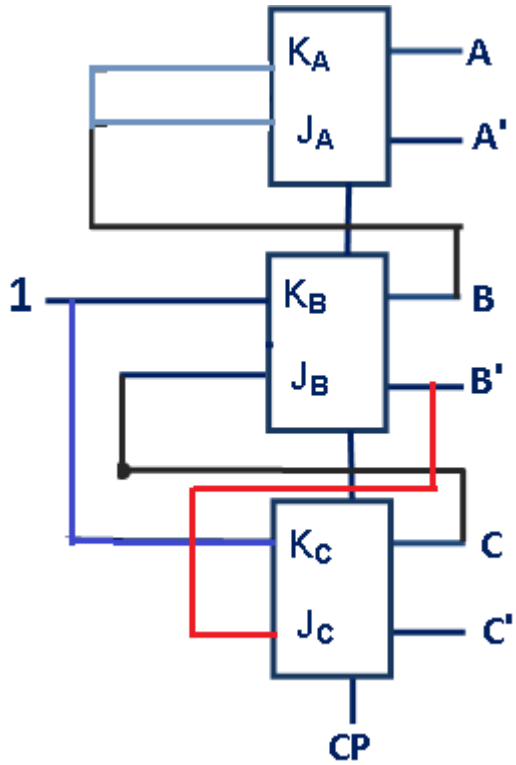
$K_A = B \quad J_A = B$

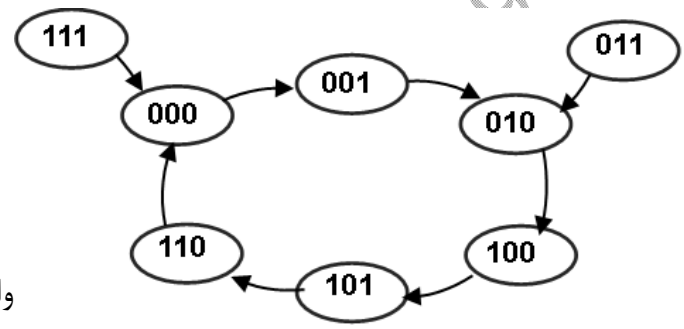
$K_B = 1 \quad J_B = C$

$K_C = 1 \quad J_C = B'$







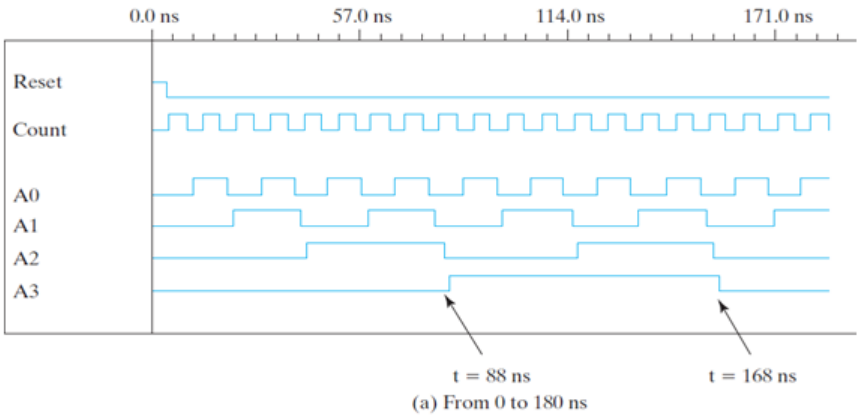


وارد

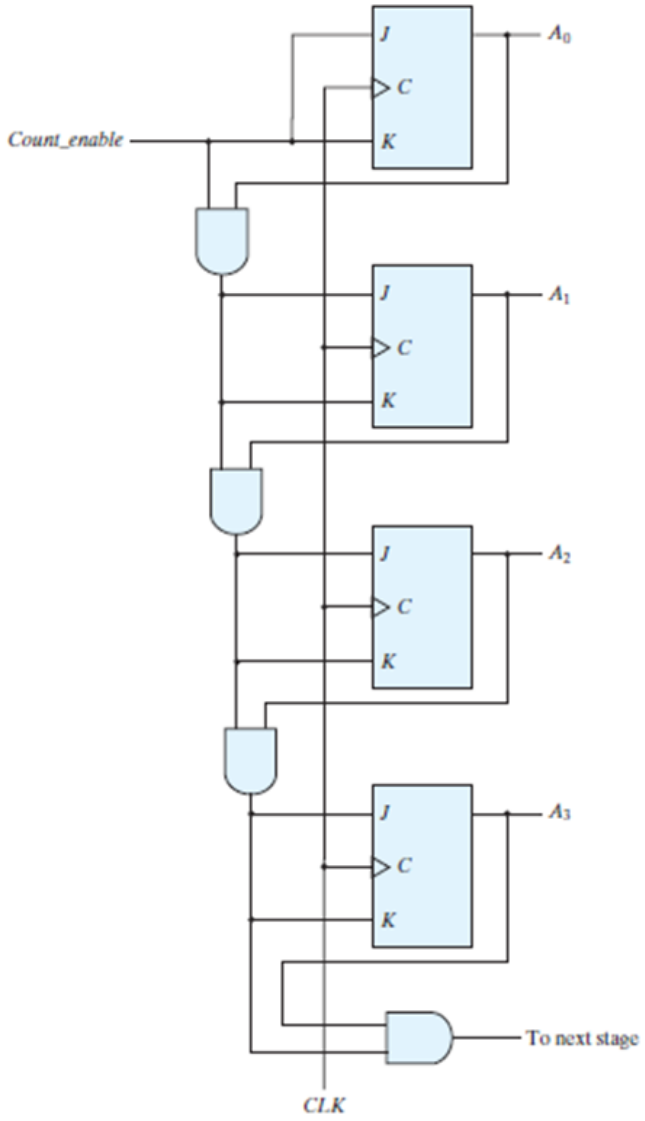
مثلا در این جا اگر 3 یا 7 شد، آیا مدار پس از آن دوباره به کار خود ادامه می‌دهد و شمارش عادی خود را انجام می‌دهد یا خیر؟!
اگر این اتفاق بیفتد مدار خودمصحح می باشد.

مدار مقابل خود مصحح است، زیرا در صورتی که یکی از حالات 111 یا 011 رخ دهد، دوباره به مدار باز می گردد.

فصل ششم: نمودار زمانی شمارنده دودویی



(a) From 0 to 180 ns

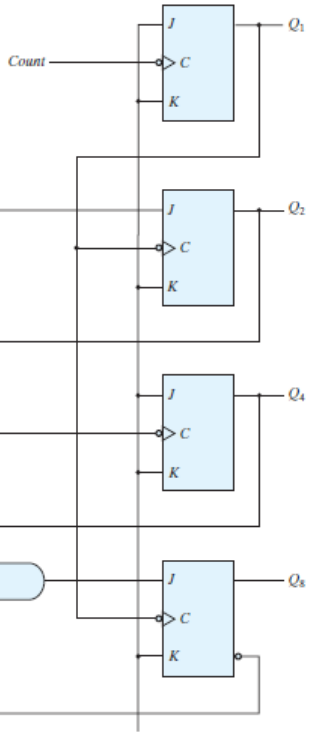


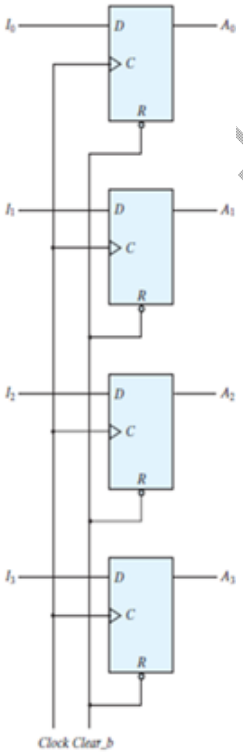
A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0

شمارنده موج گونه (RRIPPLE COUNTER):

در این شمارنده ها CLOCK فلیپ فلاپ ها با هم متفاوت است . با توجه به شکل هنگامیکه پالس CP، از 1 به صفر تغییر کند C، NOT خواهد شد .

و هنگامیکه C از یک به صفر تغییر کند B ، NOT خواهد شد و هنگامیکه B از یک به صفر تغییر کند A، NOT خواهد شد و الی آخ





ثبات ها:حافظه های موقت CPU هستند.

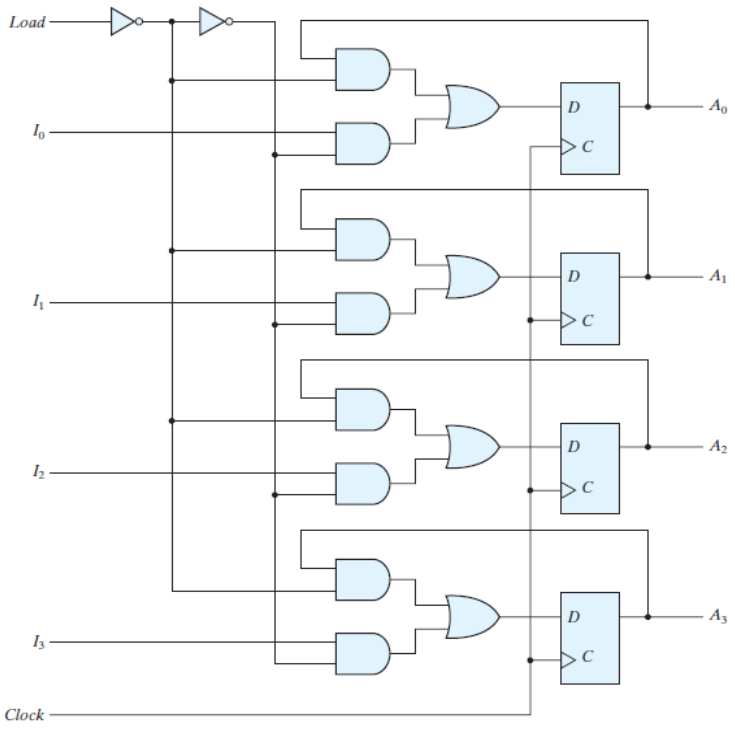
ساده ترین ثبات ها به صورت شکل ایجاد میشود .

اگر clear =1 خروجی clear بدون توجه به cp صفر می شود .

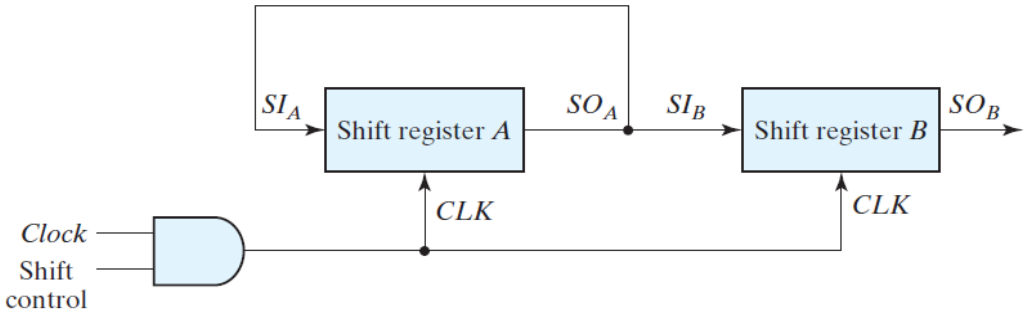
ثبات ها با امکان بارشدن موازی :

load=0 A=A

load=1 A= I

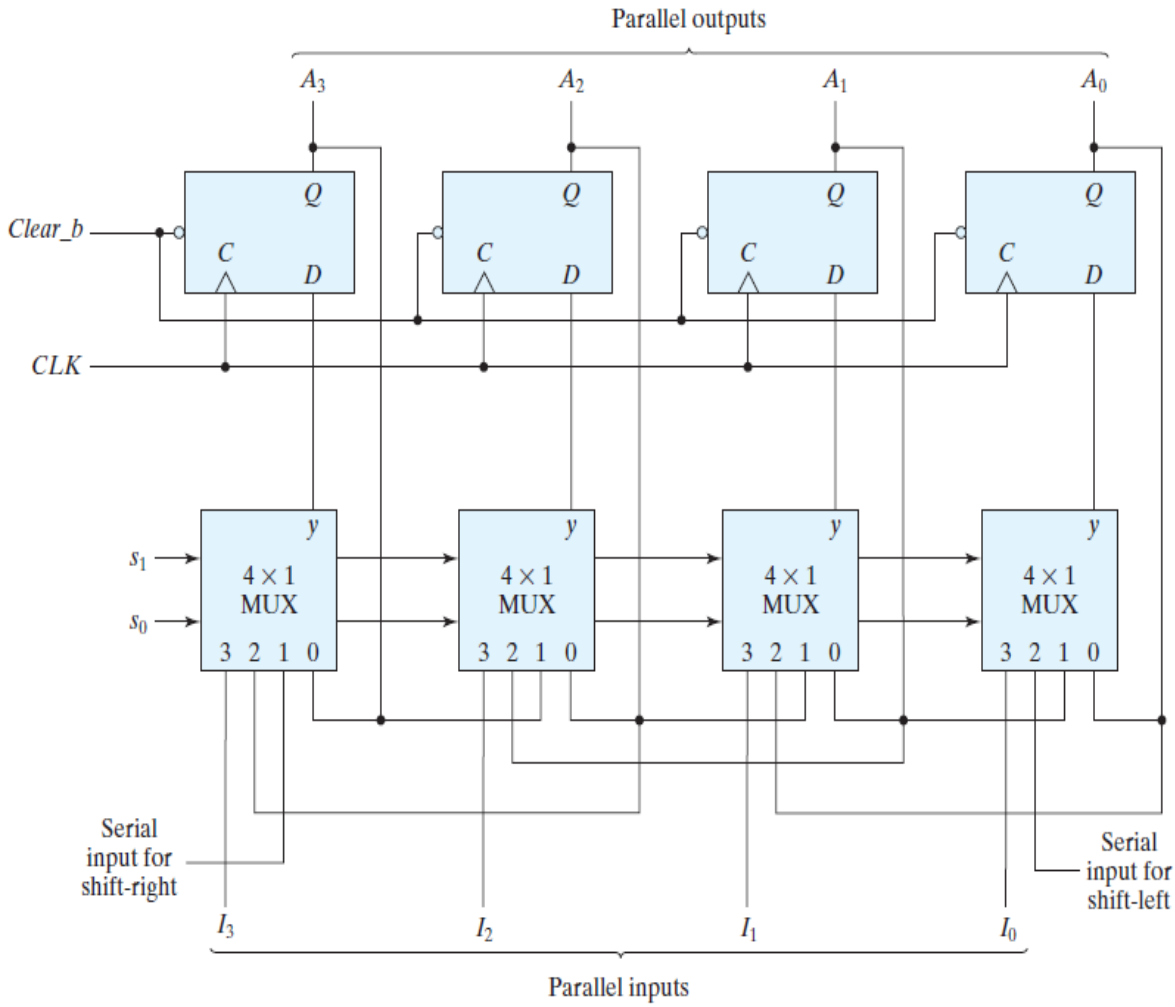


شیفت رجیستر سریال: اگر ورودی کنترل شیفت 0= هیچ اتفاقی نمی افتد .



با فرض اینکه A,B چهار بیتی باشند اگر ورودی کنترل شیفت 1 باشدبعد از چهار پالس مقدار A به B منتقل می شود و A بدون تغییر می ماند .

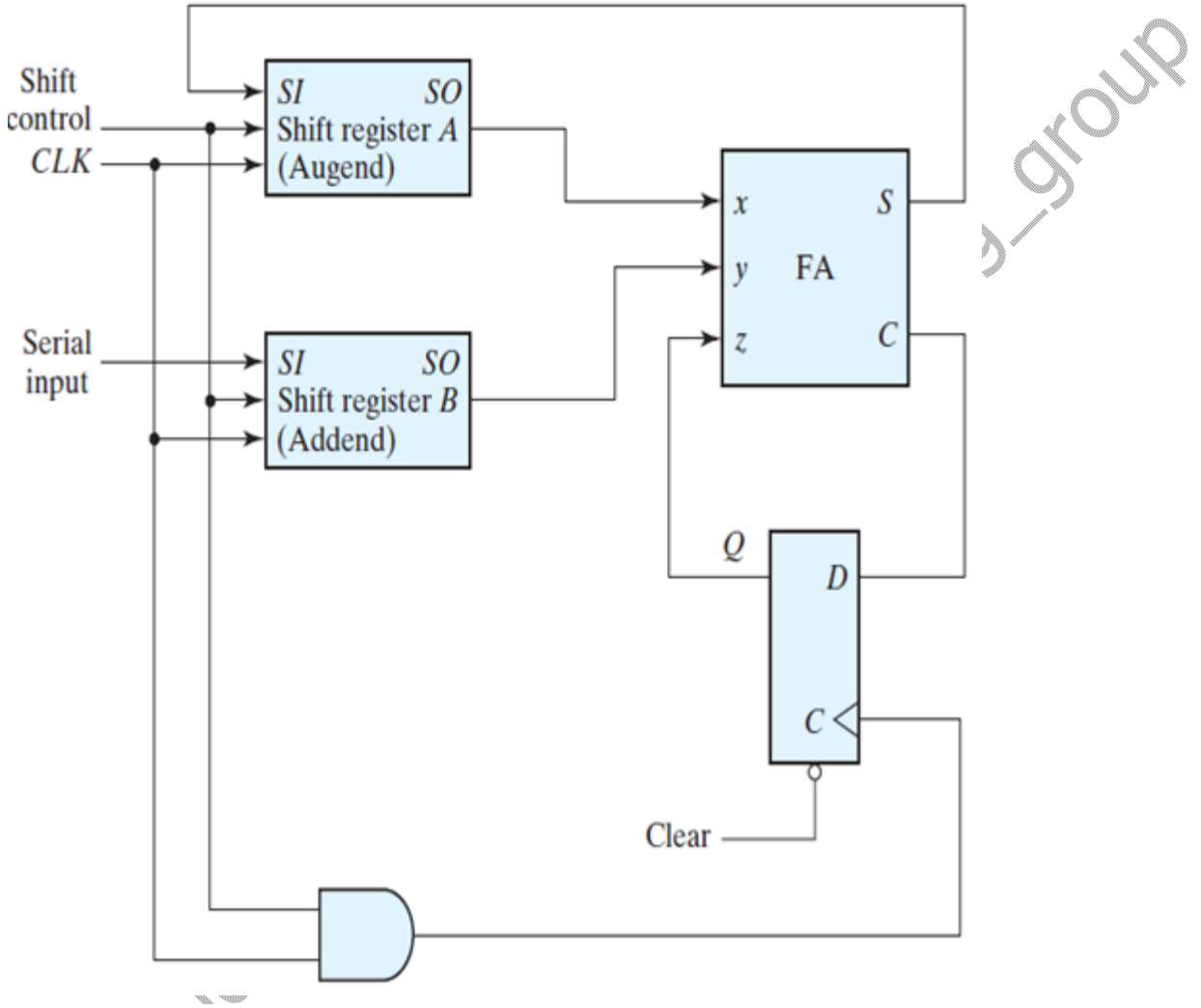
مدار شیفت رجیستر دو جهت با امکان load موازی :



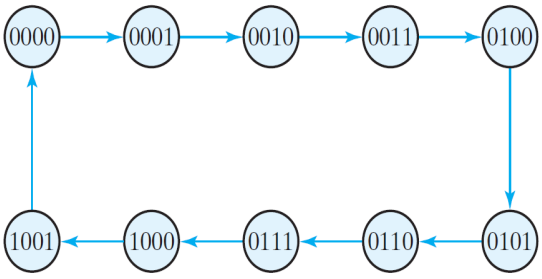
Mode Control

s_1	s_0	Register Operation
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

مدار جمع کننده سریال:در جمع دو عدد 4 بیتی از 4 FULL ADDER استفاده می‌شد. اما این جا همانطور که از اسمش پیداست به صورت سری فقط از یک FULL ADDER استفاده می‌شود.

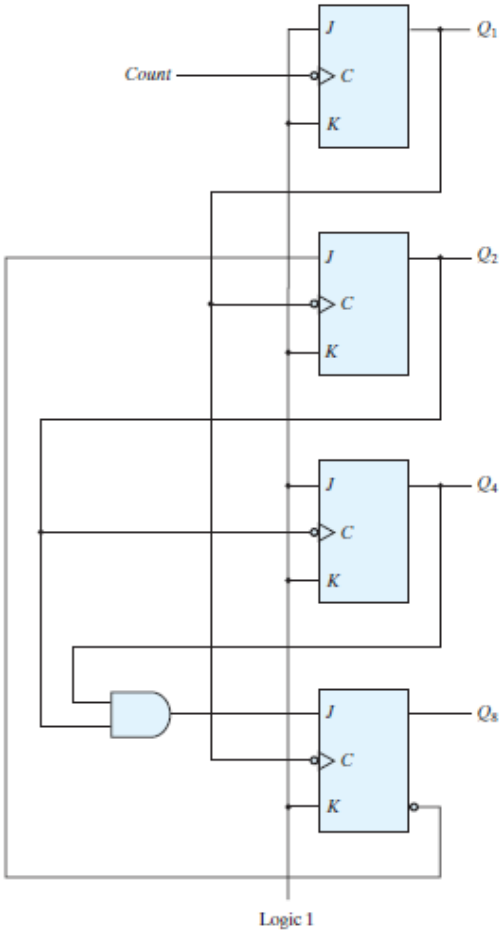


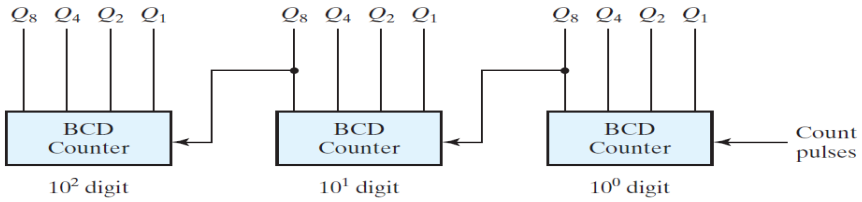
شمارنده ی موج گونه ی BCD :



- Q_1 در هر پالس NOT می شود .
- Q_2 زمانی NOT می شود که Q_2 از 1 به 0 بیاید و $Q_n = 0$
- Q_4 زمانی NOT می شود کهQ_2 از یک به صفر بیاید .

- Q_8 زمانی که Q_1 از یک به صفر بیاید و $Q_2Q_4=11$ باشد NOT شود و زمانی که Q_1 از یک به صفر بیاید و $Q_4=0$ یا Q_2 باشند، صفر می شود .





برای شمارش از 0 تا 999 BCDاستفاده از شمارنده

نمونه تمرینات حل شده در کلاس-فصل اول:

12) بدون تبدیل دهدهی جمع و ضرب های زیر را انجام دهید:

الف)اعداد دودویی 1011و101

```
  1011
 *101
------
  1011
 0000
1011
------
110111
```

```
1011 | 101
101  | 10
-----
0001
```

6)حل معادله درجه دو $x^2-11x+22=0$ برابر $x=3,x=5$است.مبنای اعداد چیست؟

$(22)_r=2*r^1+2*r^0=2*r+2 \rightarrow 3^2-(1*r+1)3+(2*r+2)=0 \rightarrow 9-3*r-3+2r+2=0 \rightarrow -r+8=0 \rightarrow r=8$

$(152)_8=(211)_x$مبنای x چیست؟

$2*x^2+1*x^1+1=1*8^2+5*8^1+2*8^0 \rightarrow 2*x^2+1*x^1+1=64+40+2 \rightarrow 2*x^2+1*x^1+1=106 \rightarrow x=7$

19)اعداد دهدهی زیر به فرم مقدار-علامت نشان داده شده است:9286_+و801_+.آن ها را به متمم 10 علامت دار تبدیل کنید و اعمال زیر را انجام دهید:(توجه کنید که حاصل جمع 10627_+ و شش رقم نیاز دارد.)

جمع معمولی =$(+9286)+(+801)$

متمم 10عدد$(801)+(+9286)$ =$(-801)+(+9286)$

$(+9286)+ 99199= 1\ 08485$

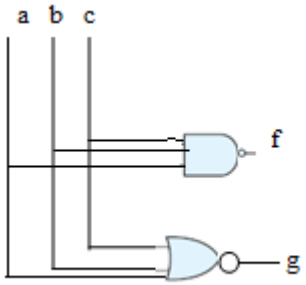
عدد باید متمم 10 شود. حذف می شود.

```
99999
-  801
-------
99198
+     1
-------
99199
```

متمم 10 عدد

35)به کمک نمودار زمان بندی مشابه شکل 5-1 سیگنال های خروجی g وf را در شکل 35-1 را به صورت تابعی از سه ورودی a وb وc نشان دهید.تمام ترکیبات هشتگانه ی aوbوc را به کار ببرید.

a	b	c	f=(abc)'	g=(a+b+c)'
0	0	0	1	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	0	0



با توجه به شکل مشخص هست برای تست صفر بودن یک مقدار از گیت nor استفاده می شود.

فصل دوم:

1)$f_1(x,y)=x+x'y$

قانون=$x+yz=(x+y)(x+z)$

(جذب) $(x+x')+(x+y)=1.(x+y)=(x+y)$

2)$f_2(x,y)=x(x'+y)=xx'+xy=0+xy=xy$

3)$f_3(x,y,z)=x'y'z+x'yz+xy'=x'z(y'+y)+xy'=xz'+xy'$

4)$f_4(x,y,z)=x'y'z'+x'y'z+xyz'+xyz$

$=x'y'(z'+z)+xy(z'+z)=x'y'+xy=x\odot y$

5)$f_5(x,y,z)=x'y'z'+x'y'z+xyz'+xyz$

روش اول:$x'y'(z'+z)+xy(z'+z)+xyz=x'y'+x'y+xyz=x'(y'.y)+xyz$

$=x'+xyz=(x'+x)(x'+yz)=x'+yz$

روش دوم:$x'(y'z'+y'z+yz'+yz)+xyz=x'+xyz=(x'+x)(x'+yz)=x'+yz$

*هرگاه 4 حالت دو عملوند باهم or شوند حاصلش یک می شود.

y	Z	
0	0	y'z'=1
0	1	y'z=1
1	0	yz'=1
1	1	yz=1

مثال:تابع زیر را ساده کنید.

1)$f_1(x,y)=\sum(0,2)=x'y+xy'=y(x'+x)=y'$

2)$f(x,y)=\sum(1,2,3)=x'y+xy'+xy=xy'+x(y'+y)=x'y+x=$

$(x'+x)+(x+y) =x+y$

3)$f(x,y,z)=\sum(2,4,6,7)=x'yz+xy'z'+xyz'+xyz=yz'(x'+x)+xy'z'+$

$xyz=yz'+xy'z'+xyz=z'(y+xy')+xyz=z'(y+x)(y+y')+xyz=z'y+z'x+xyz=$

$z'y+x(z'+yz)=z'y+x(z'+y)(z'+z)=z'y+xz'+xy$

4)$f_4(x,y,z)=\sum(0,1,2,3,7)=x'y'z'+x'y'z+x'yz+xyz+xyz=yz'\ (x'+x)+xy'z'+xyz=$
$yz'+xy'z'+xyz=z'\ (y+xy')+xyz=z'\ (y+x)(y+y')+xyz=z'y+z'x+xyz=x'y+x(z'+yz)$
$=z'y+x(z'+y)(z'+z)=z'y+xz'+xy$

5)$f_5(w,x,y,z)=\sum(4,5,6,7,8,9,10,11)=w'xy'z'+ w'xy'z+ w'xyz'+ w'xyz+ wx'y'z'+$
$wx'y'z+ wx'yz'+ wx'yz=w'x(y'z'+y'z+yz'+yz)+wx'$
$(y'z'+y'z+yz'+yz)=w'x+wx'=w(xor)x$

6)$f_6(w,x,y,z)=\sum(5,7,9,11,15)=w'xy'z+w'xyz+wx'y'z+wx'yz+wxyz=$

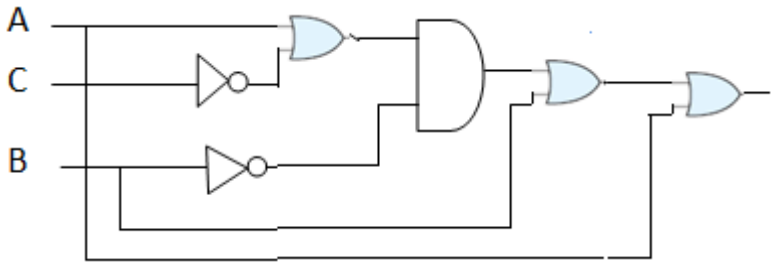
$w'xz(y'+y)+wx'z(y'+y)+wxyz=w'xz+wx'z+wxyz=w'xz+wz(x'+xy)=w'xz+wz(x'+x)$
$(x'+y)=w'xz+wzx'+wzy=z(w'x+wx')+wzy=z[(wxor)+wy]$

3)عبارت بولی زیر را با حداقل لیترال ساده نمایید:

$(BC'+A'D)(AB'+CD')=BC'AB'+BC'CD'+A'DA'B+A'DCD'=0$

13)نمودار های منطقی را برای عبارات بولی زیر رسم کنید.

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



19)تابع زیر را به صورت جمع مینترم ها ضرب ماکسترم ها بیان کنید.

F(A,B,C,D)=B'D+A'D+BD=

(A+A')B' (C+C')D+A' (B+B')(C+C')D+(A+A')B+(C+C')D=(AB'+A'B')(C+C')D=

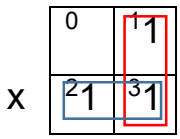
AB'CD+AB'C'D+A'B'CD+A'B'C'D+A'BCD+A'BC'D+A'B'CD+A'B'C'D+ABCD+A BC'D+A'BCD+A'BC'D=SOP(1,3,5,7,9,11,13,15)= POS(0,2,4,6,8,10,12,14)

B'D+A'D+BD=D(B'+B)+A'D=D+A'D=D(1+A')=D

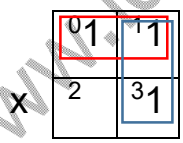
فصل سوم:

جدول کارنوی دو متغیره:

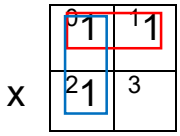
$F_1(x,y)=\sum (1,2,3)=x+y$



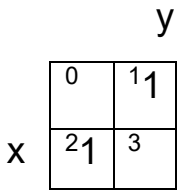
$F_2(x,y)=\sum (0,1,3)=x'+y$



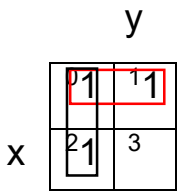
$F_3(x,y)=\sum (0,1,2)= x'+y'$



$F_5(x,y)=\sum(1,2)= x'y+xy=x \text{ xor } y$

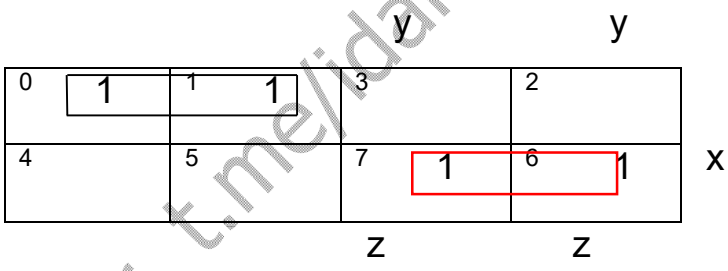


$F_6(x,y)=\sum(0,3)= xy+x'y'=x\odot y$

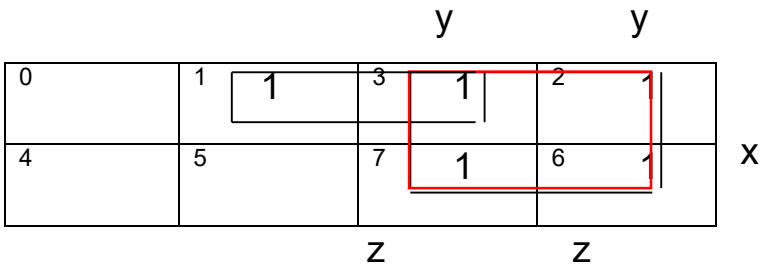


جدول کارنوی سه متغیره:

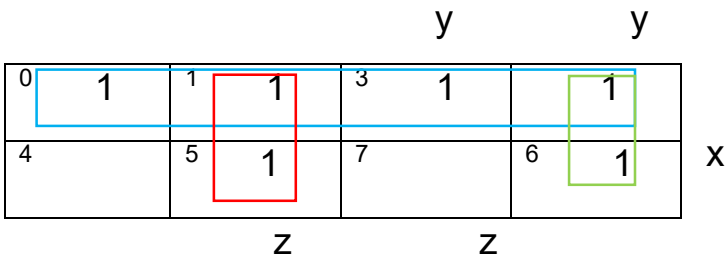
$F_1(x,y,z)=\sum(0,1,6,7)= x'y'+ xy$



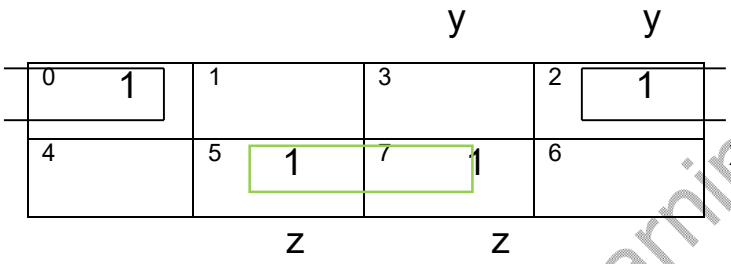
$F_2(x,y,z)=\sum(1,2,3,6,7)= y+x'z$



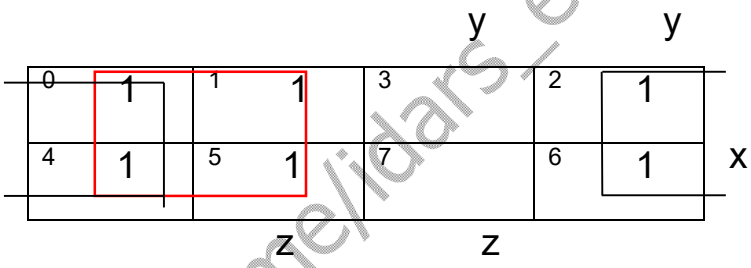
$F_3(x,y,z)=\sum(0,1,2,3,5,6)= x'+y'z+yz'$



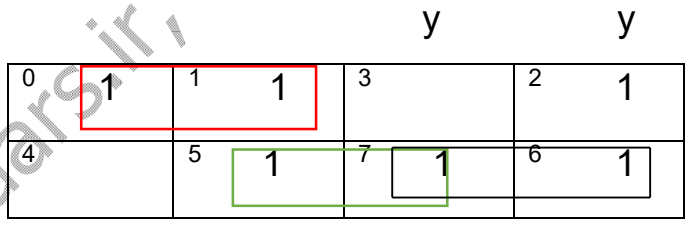
$F_4(x,y,z)=\sum(0,2,5,7)= x'z'+xz$



$F_5(x,y,z)=\sum(0,1,2,4,5,6)= y'+z'$

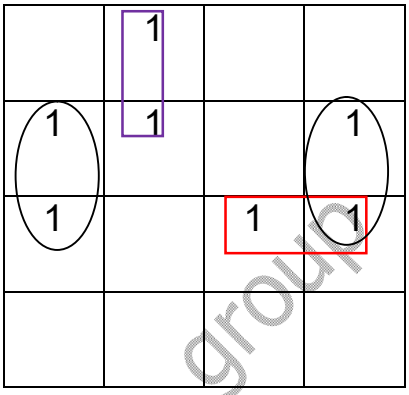


$F_6(x,y,z)=\sum(0,1,2,5,6,7)= xz+x'y'+yx$



5- با استفاده از نقشه ی 4 متغیره توابع زیر را ساده کنید.

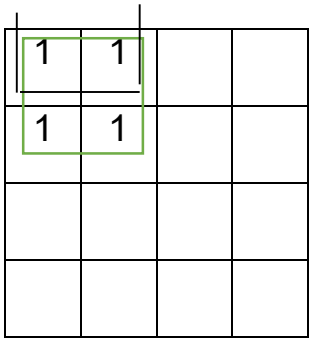
F(w,x,y,z)=∑(1,4,5,6,8,10,11)

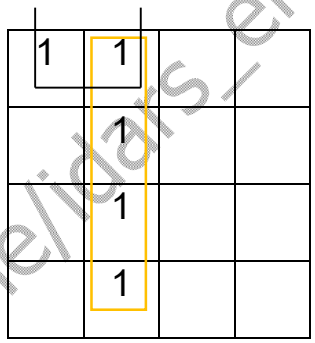


F=xz'+wxy+w'y'z

توابع زیر را ساده کنید:

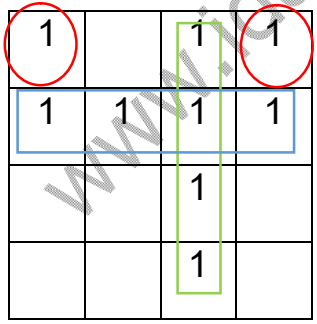
F(A,B,C,D,E)=∑(0,1,4,5,16,17,21,25,29)

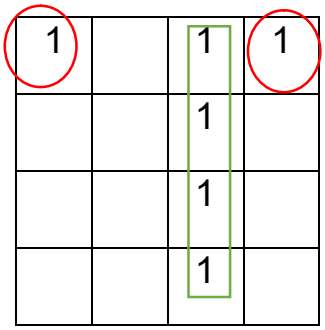




=AD'E+B'C'D'+A'B'D'

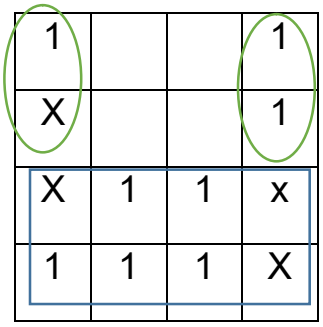
F(A,B,C,D,E)=∑(0,2,3,4,5,6,7,11,15,16,18,19,23,27,31)

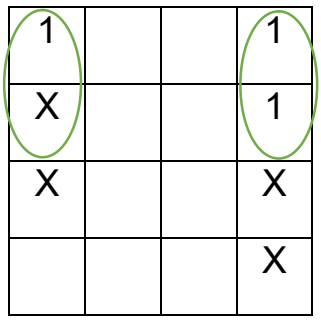




=A'B'C+B'C'E'+DE

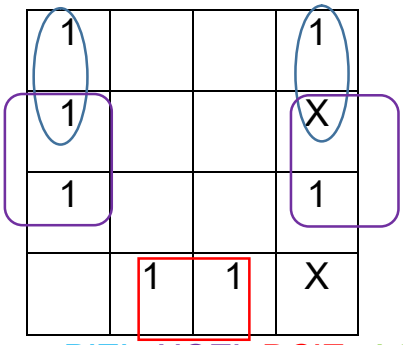
$F(A,B,C,D,E)=\sum(0,2,6,8,9,11,13,15,16,18,22),d=\sum(4,10,12,14,20,26,28,30)$

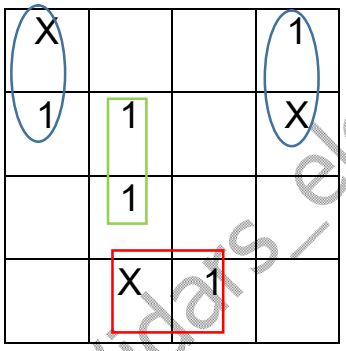




$=A'B+B'E'$

$F(A,B,C,D,E)=\sum(0,2,4,9,11,12,14,18,20,21,27,29),d=\sum(6,10,16,22,25)$



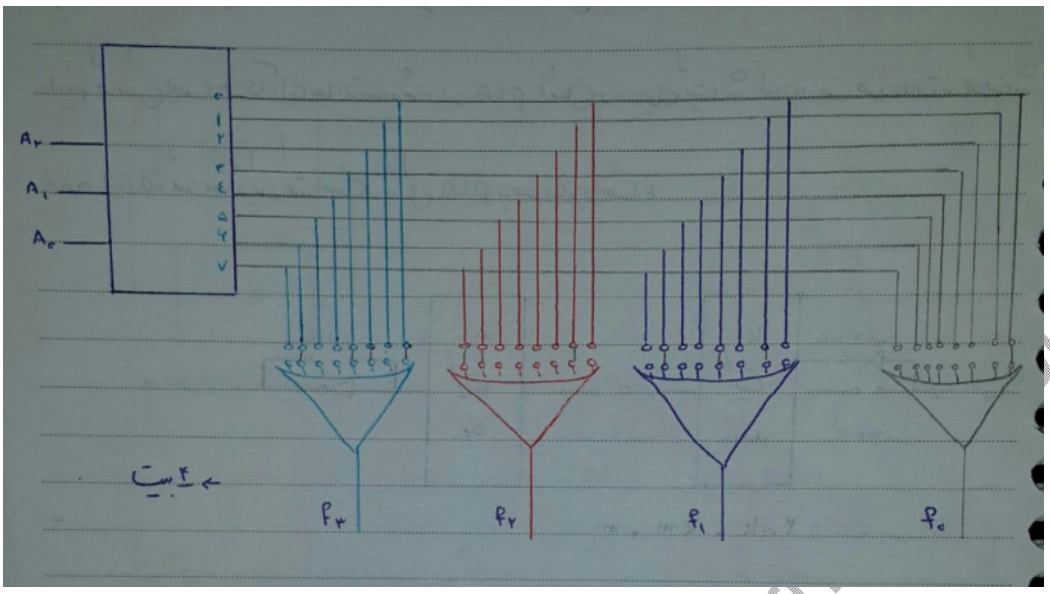


$=B'E'+A'CE'+BC'E+ACD'E$

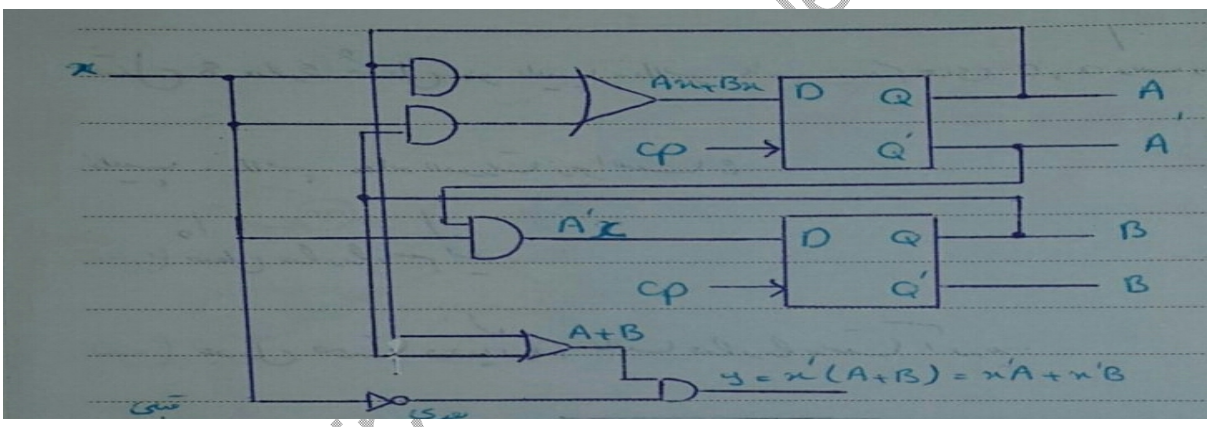
یک حافظه ROMبا 8خانه ی حافظه طراحی کنید به طوری که هرخانه 4 بیت باشد و محتویات زیر در آن قرار داشته باشد:

A_2	A_1	A_0	F_0	F_1	F_2	F_3
0	0	0	1	0	0	1
0	0	1	0	1	1	0
0	1	0	1	0	0	0
0	1	1	1	0	1	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	1	1	1	1
1	1	0	0	1	0	0

در آدرس0000 هم F0, F3 یک می شود, بنابراین خروجی صفر دیکدر به F_0 و F_3 وصل می شود.



مدار زیر چه عملی را انجام می دهد؟

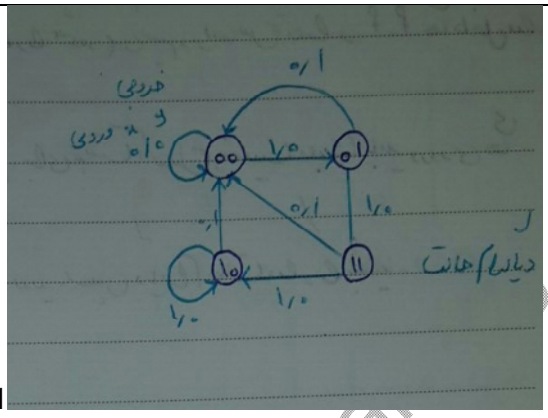


قبلی **بعدی**

A	B	X	A	B	Y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

A(t+1)=Ax+Bx B(t+1)=A'x y=x'(A+B)

A	B	AB(x=0)	AB(x=1)	y(x=0)	y(x=1)
0	0	00	01	0	0
0	1	00	11	1	0
1	0	00	10	1	0
1	1	00	10	1	0



0-->1-->2-->3 x=1

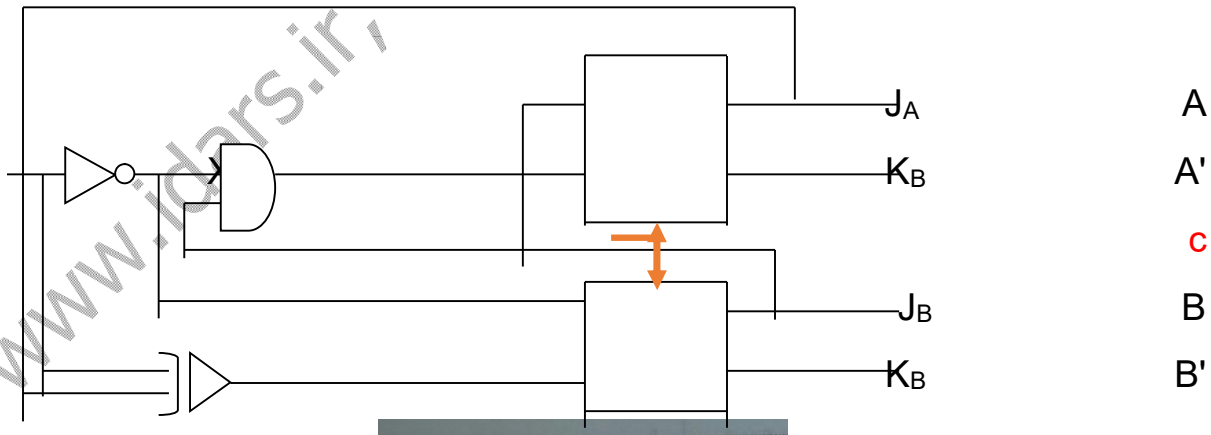
مثال:مداری شامل دو فلیپ فلاپ j-k به نام های a و b و یک ورودی x داریم,عبارات ورودی های فلیپ فلاپ به صورت زیر است:

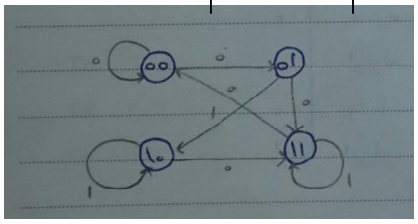
$J_A=B$, $J_B=X'$

$K_A=BX'$, $K_B=A \oplus X$

الف)شکل مدار را رسم کنید.

ب)جدول صحت و دیاگرام حالت مدار را به دست آورید:





فعلی بعدی

A	B	X	A	B	J_A	K_A	JB	K_B
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0

نمونه سوال میانترم

1– مکمل 8 عدد $(32.67)_8$ کدام است؟

۱) $(45.10)_8$ ۲) $(45.11)_8$ ۳) $(44.12)_8$ ۴) $(44.13)_8$

۲– عدد 1001 در کدگذاری $Excess-3$ مفروض است. معادل باینری آن کدام است؟

۱) ۰۱۱۰ ۲) ۱۱۰۰ ۳) ۱۰۰۰ ۴) ۰۱۰۱

۳– حاصل تفریق عدد $(25)_b$ از عدد $(51)_b$ برابر با $(20)_{10}$ است b با پایه کدام است؟

۱) ۷ ۲) ۱۰ ۳) ۸ ۴) ۹

4 – جدول کارنو داده شده چه تابع منطقی را مشخص می کند؟

	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}\bar{D}$	0	0	0	0
$\bar{C}D$	0	1	0	1
CD	0	0	0	0
$C\bar{D}$	0	1	0	1

۱) $(A\oplus B)\overline{(C\oplus D)}$

۲) $(A\oplus B)(C\oplus D)$

۳) $\overline{(A\oplus B)}(C\oplus D)$

۴) $\overline{(A\oplus B)(C\oplus D)}$

۵ – برای عبارت جبری زیر: (ارشد ۱۳۸٤)

$$F(W,X,Y,Z)=X'Z'+WYZ+W'Y'Z'+X'Y$$

عبارت جبری مینیمم حاصلضرب جمع را پیدا کنید. **(POS)**

۱) $F(W,X,Y,Z)=(Y'+Z)(W+X+Z')(W'+X+Y)$

۲) $F(W,X,Y,Z)=(Y+Z')(W'+X'+Z)(W+X'+Y')$

۳) $F(W,X,Y,Z)=(X+Z)(W'+Y'+Z')(W+X+Z)(X+Y')$

۴) $F(W,X,Y,Z)=(Z'+Y)(W'+Y'+Z')(W+Z+Y)(X'+Z')$

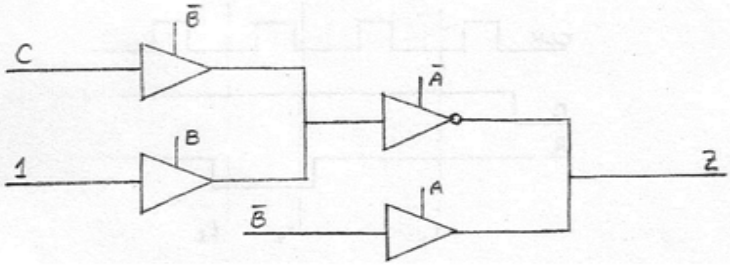
۶– کدام گزینه تابع خروجی مدار شکل مقابل را نمایش می‌دهد؟ (ارشد ۱۳۸۵)

۱) $z=1$

۲) $z=A+\bar{B}\bar{C}$

۳) $z=\bar{A}B+BC$

۴) $z=A\bar{B}+B\bar{C}$



۷ – ساده‌ترین صورت تابع مقابل کدام است؟ (ارشد ۱۳۸۱)

$$F(A,B,C,D,E)=\sum(0,2,5,8,10,16,23,26,29),d(7,18,21,24)$$

۱) $F=\bar{C}\bar{E}+\bar{B}CE+AC\bar{D}E$

۲) $F=A\bar{C}\bar{E}+B\bar{C}\bar{E}+AC\bar{D}E$

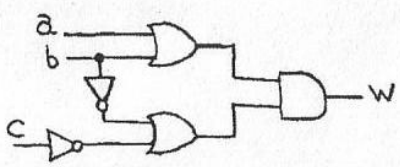
۳) $F=AB+\bar{B}C+AB\bar{E}+B\bar{D}$

۴) $F=\bar{C}\bar{E}\bar{D}+B\bar{C}E+BC\bar{D}E$

۸ – در تابع $f(a,b,c,d)=\sum m(3,7,8,9,12)+d(2,6,11,14)$ به ترتیب چند Prime Implicant و Essential Prime Implicant وجود دارد؟ **(ارشد ۱۳۸۸)**

۱) ۵ و ۱ ۲) ۵ و ۲ ۳) ۷ و ۱ ۴) ۷ و ۲

9- در مدار زیر اضافه کردن کدام گیت باعث از بین رفتن Potential Hazard خواهد شد؟



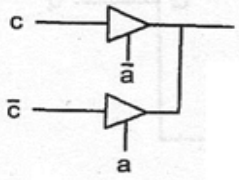
۱) AND : $\bar{a} \cdot c$

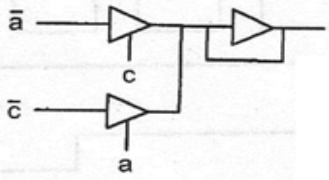
۲) NAND : $\overline{\bar{a} \cdot c}$

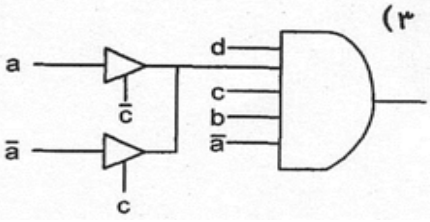
۳) NOR : $\overline{\bar{a} + b}$

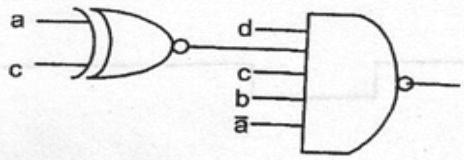
۴) NAND : $\overline{\bar{b} \cdot a}$

10- کدامیک از مدارهای زیر، تابع $(a \oplus c) + \bar{a}bcd$ را پیاده‌سازی می‌نماید؟ (ارشد ۱۳۸۰)

۱)


۲)


۳)


۴)


11- برای تابع بولی ۵ متغیرهٔ زیر ساده‌ترین صورت حاصل جمع حاصلضربها کدام است؟ (ارشد ۱۳۷۹)

$$F(A,B,C,D,E) = \sum(0,3,8,14,15,16,18,24,26,27,29) \text{ و } d(6,7,9,19,22)$$

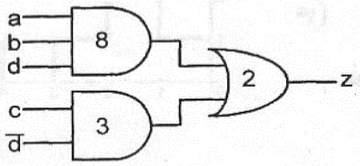
۱) $\bar{C}\bar{D}\bar{E}+A\bar{B}\bar{C}+\bar{A}\bar{B}\bar{C}E+AB\bar{C}\bar{D}\bar{E}$

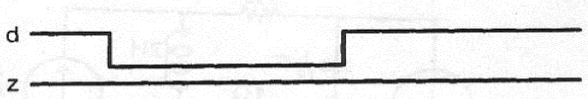
۲) $\bar{A}CD+\bar{C}\bar{D}E+\bar{B}\bar{C}D+\bar{A}\bar{B}DE+AB\bar{C}\bar{D}E$

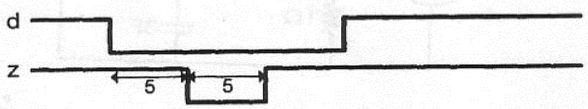
۳) $\bar{A}CD+\bar{C}\bar{D}\bar{E}+A\bar{C}D+\bar{A}\bar{B}\bar{C}E+AB\bar{C}\bar{D}E$

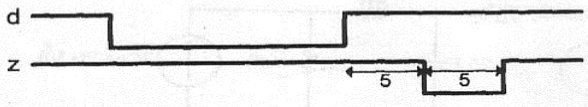
۴) $\bar{A}CD+A\bar{C}+\bar{A}\bar{B}\bar{C}\bar{E}+AB\bar{C}\bar{D}E+\bar{C}\bar{D}E$

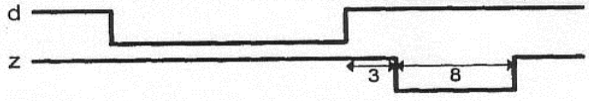
12- در مدار زیر، تأخیر گیت‌ها در داخل آنها نوشته شده است. در زمانی که $a = b = c = 1$ است، d از یک به صفر و دوباره به یک برمی‌گردد در خروجی z کدام موج دیده خواهد شد؟ (زمانها به نانو ثانیه است.) (ارشد ۱۳۸۰)



۱)


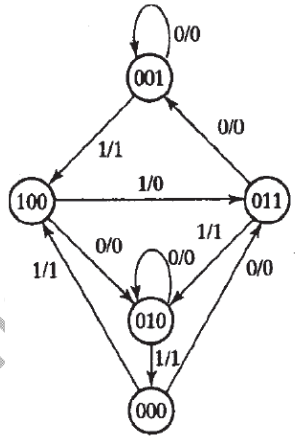
۲)


۳)


۴)


نمونه سوال پایان ترم:

1. یک مدار مقایسه گر طراحی کنید که دو عدد 3 بیتی را با هم مقایسه کند.
2. با استفاده از جمع کننده های 4 بیتی باینری جمع کننده BCDطراحی کنید.
3. تابع $F(A,B,C,D)=\sum(0,4,5,6,7,10,13,15)$ را یکبار با مالتی پلکسر 8→1 و سپس با مالتی پلکسر 4→1 (ورودی های انتخاب باید B,C باشند) طراحی کنید.
4. مدار یک انکدر اولویت 8→3 را طراحی کنید در طرحتان اعداد کوچکتر را دارای اولویت بیشتر در نظر بگیرید.
5. به تعداد کافی دیکدر 4→2 و 8→3 داریم با استفاده از آنها یک دیکدر 32→5 طراحی کنید.
6. با استفاده از فلیپ فلاپ SR مداری طراحی کنید که دیاگرام حالت زیر را پیاده سازی کند. آیا مدارتان خود مصحح است؟



7. یک مدار ترتیبی دارای دو فلیپ-فلاپ JK با نام های A,B و یک ورودی X و یک خروجی Y است. تابع ورودی فلیپ فلاپ و خروجی آن بصورت زیر است.

JA=BX+A', KA=BX', JB=A'X, KB=A+B, Y=AX+BX'

الف- شکل مدار را رسم کنید.

ب- جدول صحت مدار را بدست آورید.

ج- دیاگرام حالت مدار را بکشید.

اهداف بلند همت‌های بلند می طلبد.

موفق و پیروز باشید.