فصل اول

جبر بول و ساده‌سازی توابع بولی

جبر بول:

یک شبکه مکمل‌پذیر و توزیع‌پذیر را جبر بول گویند و به صورت < B , * , + , ' , o , 1 > نشان می‌دهند که در آن:

B: یک مجموعه متناهی می‌باشد (در کامپیوتر محدود به دو عضو صفر و یک است).

+ , *: عملیات دوتایی (BINARY OPERATION) روی مجموعه B

/ : عمل یگانی (UNARY OPERATION) روی مجموعه B

1 , 0 : کران‌ها هستند که صفر کوچک‌ترین عضو و یک بزرگ‌ترین عضو مجموعه B است.

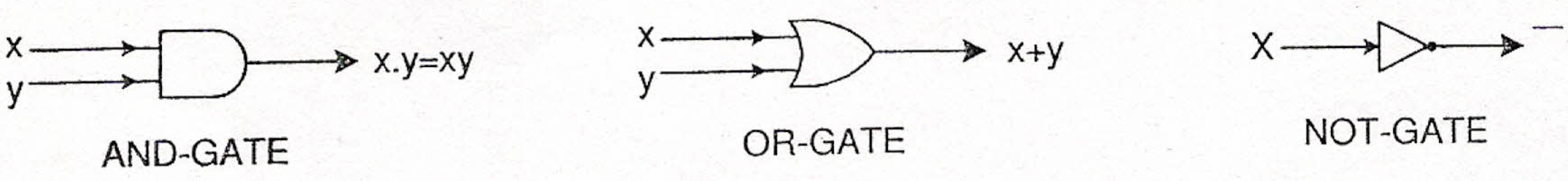
جبر بول دو مقداری:

جبر بول دو مقداری روی مجموعه‌های با دو عضو مانند B = { 0 , 1 } و عملیات دوتایی "+" ، "." و عمل یگانی " – " طبق جدول‌های زیر تعریف می‌شود:

x	$\bar{x}$
0	1
1	0

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

x	y	x.y
0	0	0
0	1	0
1	0	0
1	1	1



و در خواص زیر صدق می‌کند:

$\forall x, y, z \in B$

خاصیت بستار:

$1)\ x+y \in B$

$x.y \in B$

وجود عضو همانی:

$2)\ x+0=x$

$x.1=x$

خاصیت جابه‌جایی:

$3)\ x+y=y+x$

$x.y=y.x$

خاصیت توزیع‌پذیری:

$4)\ x+(y.z)=(x+y).(x+z)$

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

خاصیت مکمل‌پذیری:

$5)\ x+\overline{x}=1$

$x.\overline{x}=0$

خاصیت خود توانی:

$6)\ x+x=x$

$x.x=x$

وجود عضوهای صفر:

$7)\ x+1=1$

$x.0=0$

خاصیت انجمنی:

$8)\ x+(y+z)=(x+y)+z$

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

قانون دمرگان:

$9)\ (x+y)'=x'.y'$

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

خاصیت جذبی:

$10)\ x+(x.y)=x$

$x.(x+y)=x$

عبارت‌ها و توابع بولی:

تعریف:

عبارت بولی را می‌توان به صورت تعریف بازگشتی زیر بیان نمود:

۱. $0, 1, x, \overline{x}$ هر کدام به تنهایی عبارت بولی هستند.

۲. اگر E_1 ، E_2 دو عبارت بولی باشد. $E_1.E_2, E_1+E_2, \overline{E_1}, \overline{E_2}$ نیز عبارت بولی خواهد بود.

۳. تنها عبارت‌هایی بولی خواهند بود، که بتوان آن ها را طی کاربردهای متناهی از 1 و 2 به‌دست آورد.

مثال: عبارت روبرو مطابق تعریف فوق یک عبارت بولی است: $\overline{(x_1+\overline{x}_2)}$

تعریف: فرض کنیم $< B , * , + , \bar{} , 0 , 1 >$ یک جبر بول و $B = \{0 , 1\}$:

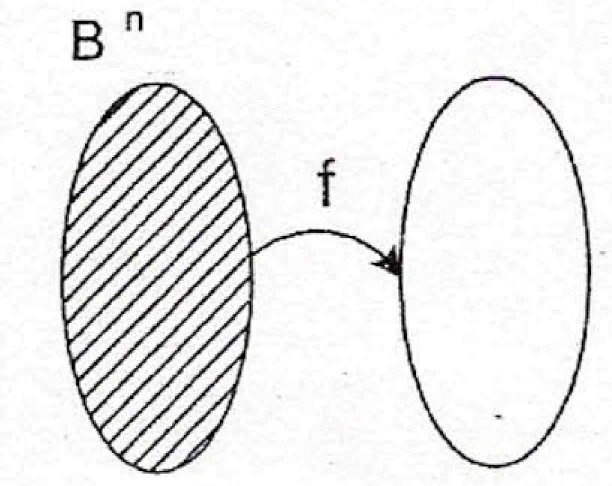
تابع $f : B^n \to B$ را که متناظر با یک عبارت بولی است " تابع بولی" گویند.

که در آن عضوهای B^n ، n تایی مرتب $< x_1 , x_2 ,...x_n >$ و $x_i \in \{0,1\}$ می‌باشد.

۱. در صورتی که f یک نگاشت از B^n به سوی B باشد، یعنی هر n تایی مانند: $< x_1 , x_2 ,..., x_n > \in B^n$

با یک و فقط یک عضو از مجموعه $B = \{0 , 1\}$ به‌وسیله f در رابطه باشد آن‌گاه f را تابع بولی "کاملاً مشخص شده" گویند.

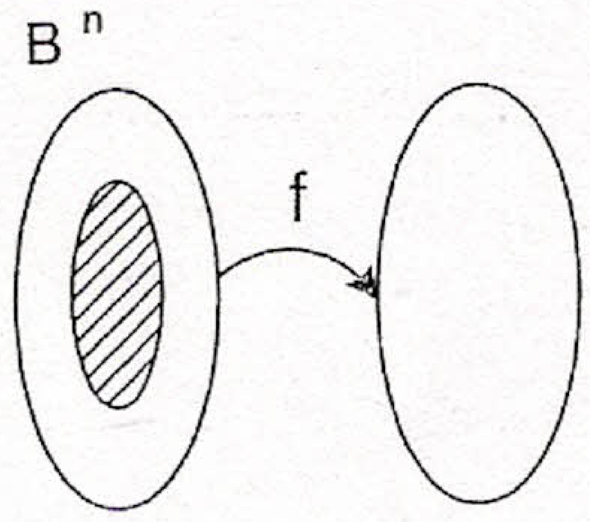
(بنابراین حداکثر $(2)^{2^n}$ تابع کاملاً مشخص شده قابل تعریف است)



که در آن $D_f = B$

۲. هر گاه هر n تایی مرتب از مجموعه B^n حداکثر با یک عضو از مجموعه B در رابطه f باشد، "آن‌گاه f را تابع بولی گویند که به طور کامل مشخص نشده است".

$D_f \subseteq B^n$



بنابراین مقدار تابع برای برخی از عضوهای B^n نامشخص خواهد بود. یعنی برای هر n تایی مرتب از B_n مقدار تابع می‌تواند 1 , 0 و یا نامشخص باشد. پس حداکثر $(3)^{2^n}$ تابع قابل تعریف است.

۳. اگر مکمل تابع f را با $\overline{f}$ نشان دهیم، آن‌گاه $\overline{f}$ را می‌توان با استفاده از عبارت بولی تابع f و تعویض "." و "+" و تعویض "0" و "1" و مکمل نمودن همه متغیرها به‌دست آورد.

مثال:

$$f = \left(x_1 + (x_2 , x_3)\right).\left(x_4 .(\overline{x}_3 + \overline{x}_5)\right)$$

$$\overline{f} = \left(\overline{x}_1 .(\overline{x}_2 + \overline{x}_3)\right) + \left(\overline{x}_4 + (x_3 . x_5)\right)$$

۴. اگر همزاد (DUAL) تابع f را با f_d نشان دهیم، آن‌گاه عبارت بولی تابع f_d را می‌توان با استفاده از عبارت بولی تابع f و تعویض "+"، "." و تعویض "0" , "1" به‌دست آورد.

رابطه بین همزاد و مکمل تابع:

$$f_d = \overline{f}\left(\overline{x}_1 , \overline{x}_2 ,...., \overline{x}_n\right)$$

در صورتی‌که $f = f_d$ باشد، آن‌گاه تابع را خود همزاد (SELF DUAL) گویند.

۵ـ قضیه شانون: هر تابع ترکیبی n متغیری را می‌توان به صورت مجموع دو تابع (n-1) متغیری نوشت یعنی:

$$f\left(x_1 , x_2 , \dots , x_n\right) = x_i . g_1\left(x_1 , x_2 , \dots , x_{i-1} , x_{i+1} , \dots , x_n\right) + \overline{x}_i . g_0\left(x_1 , x_2 , \dots , x_{i-1} , x_{i+1} , \dots , x_n\right)$$

در این صورت گوییم تابع f را حول متغیر x_i بسط داده‌ایم.
که در آن:

$$g_1(x_1, x_2, ..., x_{i-1}, x_{i+1}, ..., x_n) = f(x_1, x_2, ..., x_{i-1}, 1, x_{i+1}, ..., x_n)$$
$$g_0(x_1, x_2, ..., x_{i-1}, x_{i+1}, ..., x_n) = f(x_1, x_2, ..., x_{i-1}, 0, x_{i+1}, ..., x_n)$$

مثال:

$$f(x_1, x_2, x_3) = x_1\overline{x}_2 + x_3$$
$$f(x_1, x_2, x_3) = x_1 . g_1(x_2, x_3) + \overline{x}_1 g_0(x_2, x_3)$$
$$g_1(x_2, x_3) = f(1, x_2, x_3) = 1.\overline{x}_2 + x_3 = \overline{x}_2 + x_3$$
$$g_0(x_2, x_3) = f(0, x_2, x_3) = 0.\overline{x}_2 + x_3 = x_3$$
$$f(x_1, x_2, x_3) = (x_1.(\overline{x}_2 + x_3)) + (\overline{x}_1 . x_3)$$

اگر تابع f را حول k متغیر آن بسط دهیم، می‌توان f را به صورت مجموع 2^k جمله حاصل‌ضرب نوشت که در آن هر جمله حاصل‌ضرب متشکل از k متغیر به صورت x_i یا $\overline{x}_i$ و یک تابع (n-k) متغیری خواهد بود.

اگر k = n باشد، آن‌گاه توابع (n-n) متغیری یا مساوی صفر و یا مساوی یک خواهند بود و در نتیجه تابع به صورت مجموع جملات حاصل‌ضرب n متغیری بیان خواهد شد که آن را مجموع مین‌ترم‌ها (Minterms) گویند. بنابراین با توجه به قضیه شانون هر تابع بولی را می‌توان به صورت مجموع حاصل‌ضرب‌ها نوشت و در نتیجه مدار آن را به صورت دو سطحی AND-OR پیاده‌سازی نمود. که سریع‌ترین مدار ترکیبی خواهد بود.

کاربرد جبر بول دو مقداری در طراحی مدارهای ترکیبی:

جبر بول دو مقداری شامل متغیرهای بولی است، مانند $x_i \in \{0,1\}$ و عملیات منطقی OR , AND و NOT است.هر تابع بولی متناظر با یک عبارت بولی است که رشته‌ای است متناهی و متشکل از متغیرهای بولی و عملیات •,+,– که برای هر یک از ترکیبات ارزش ممکن برای متغیرها مقدار تابع مساوی 1 , 0 و یا نامشخص خواهد بود. مثلا با فرض تابع $f = x + \overline{y}z$، مقدار تابع در یکی از دو وضعیت زیر مساوی با یک خواهد بود:

a) اگر x=1 باشد، آن‌گاه: f = 1 است. یعنی در این صورت متغیرها z , y تاثیری در مقدار تابع دارد و یا don't – care هستند و در نتیجه مقدار تابع به ازای هر یک از چهار ترکیب زیر مساوی یک خواهد بود.

x	y	z
1	0	0
1	0	1
1	1	0
1	1	1

$\Rightarrow f = 1 \quad \text{if} \quad x = 1$

b) اگر $\overline{y}z = 1$ باشد یا (y =0 , z=1) باشد، آن‌گاه f = 1 است. یعنی متغیر x برای مقدار تابع don't – care خواهد بود و در نتیجه مقدار تابع به ازای هر یک از دو ترکیب زیر مساوی با یک خواهد بود.

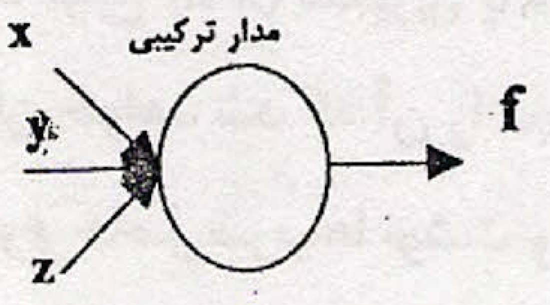
x	y	z
0	0	1
1	0	1

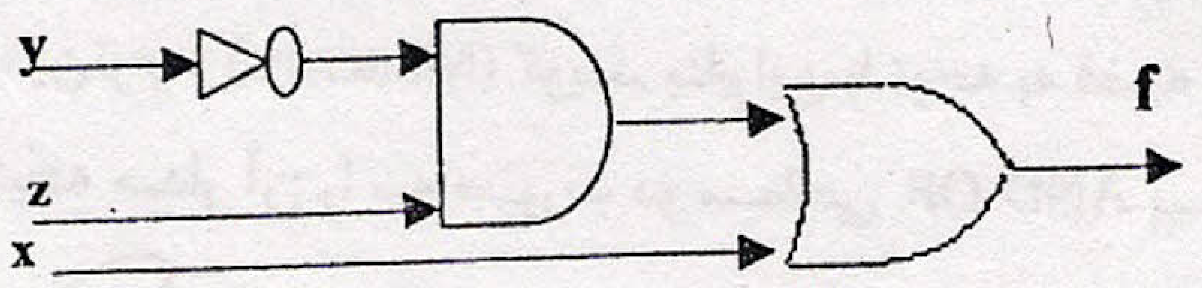
$\Rightarrow f = 1 \quad \text{if} \quad \overline{y}z = 1$

رابطه بین مقدار تابع و متغیرهای آن را می‌توان به‌وسیله جدول ارزش (TRUTH TABLE) نشان داد. اگر تابع n متغیری باشد، آن‌گاه جدول ارزش آن 2^n ترکیبات ارزش متمایز صفر و یک خواهد داشت. مثلاً برای تابع فوق جدول ارزش به صورت زیر خواهد بود:

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

هر تابع بولی را می‌توان با استفاده از عبارت بولی متناظر با آن و gateهای NOT ، OR و AND به نمودار منطقی یا Logical Diagram تبدیل نمود که در آن متغیرها به عنوان ورودی و خروجی مدار، مقدار تابع f را به ازای هر یک از ترکیبات ورودی مشخص می‌کند.





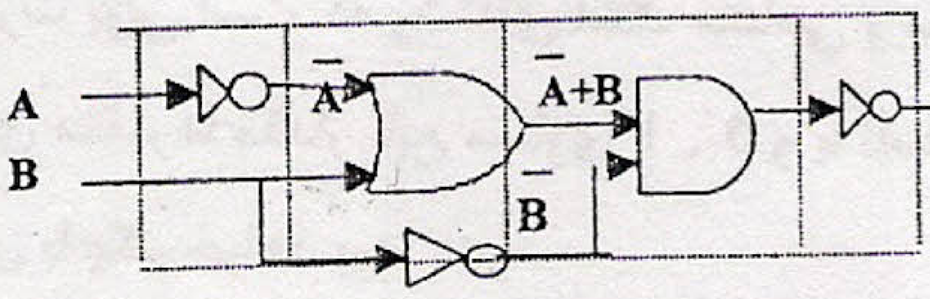
بنابراین از جبر بول دو مقداری می‌توان در تجزیه و تحلیل و طراحی مدارهای ترکیبی استفاده نمود و به کمک آن:

۱. جدول ارزش را می‌توان به صورت عبارت جبری نمایش داد.

۲. رابطه بین ورودی و خروجی مدار را به صورت جبری نشان داد. (که صورت‌های مختلفی دارد)

۳. مدار ساده‌تری برای تابع ایجاد نمود.

مثال: با فرض $f(A,B,C)=\overline{\left((A+B).\overline{B}\right)}$



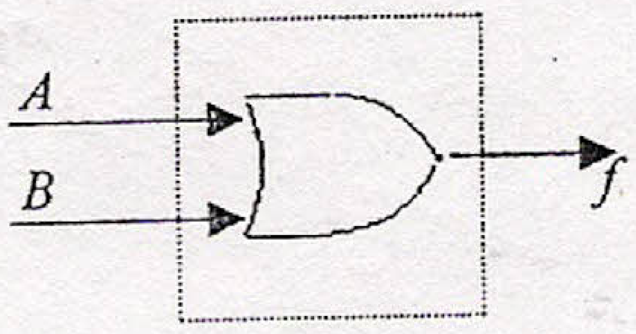
مدار عبارت به این صورت است.

با استفاده از قوانین جبر بول، می‌توان عبارت بولی تابع f را ساده نمود و مداری بهینه طرح نمود.

$$f(A,B,C)=\overline{\left(\left(\overline{A}+B\right).\overline{B}\right)}$$

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

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



طراحی مدارهای ترکیبی:

در یک تابع ترکیبی n متغیری مانند $f(x_1,x_2,...,x_n)$ که در آن $x_i\in\{0,1\}$ تنها 2^n ترکیبات ارزش ممکن برای متغیرها وجود دارد که این مجموعه 2^n ترکیبات ارزش، را یک فضا یا مکعب n بعدی و هر یک از 2^n ترکیبات ارزش ممکن برای متغیرها مانند $X_i=<x_1,x_2,....,x_n>$ یک نقطه از فضا گویند که مقدار تابع به ازا نقطه X_i ممکن است 1 یا 0 یا نامشخص باشد.

اگر $f(X_i)=1$ باشد آن‌گاه X_i را نقطه -1 تابع گویند.

اگر $f(X_i)=0$ باشد آن‌گاه X_i را نقطه -0 تابع گویند.

اگر $f(X_i) = -$ باشد آن‌گاه X_i را نقطه بی‌تفاوت یا Dont - Care گویند. Don't care (d.c.) به مفهومی ترکیبی است که هرگز در ورودی مدار رخ نخواهد داد و طراح مقدار تابع را به ازای نقاط بی‌تفاوت به دلخواه می‌تواند صفر و یا یک در نظر بگیرید.

هر تابع ترکیبی را می‌توان به وسیله جدول ارزش نمایش داد.

x_1	x_2	x_3	f_1	f_2	f_3
0	0	0	1	1	1
0	0	1	0	0	0
0	1	0	1	-	0
0	1	1	0	-	0
1	0	0	1	1	-
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	0	0	0

f_1 تابع بولی کاملاً مشخص شده است ولی f_3, f_2 توابع بولی هستند که به طور کامل مشخص نشده‌اند.

$f_1(0,1,0) = 1$ بنابراین (0 , 1 , 0) نقطه یک تابع f_1 است.

$f_2(0,1,0) = -$ بنابراین (0 , 1 , 0) نقطه بی‌تفاوت تابع f_2 است.

$f_3(0,1,0) = 0$ بنابراین (0 , 1 , 0) نقطه صفر تابع f_3 است.

مراحل طراحی مدارهای ترکیبی

۱. اولین قدم در طراحی مدارهای ترکیبی، استنتاج جدول ارزش از توصیف لفظی صورت مساله است.

هر چند برای طراحی مدارهای ترکیبی همیشه نمی‌توان از جدول ارزش استفاده نمود و هم‌چنین به علت مبهم بودن توصیف لفظی رویه اصولی و منظم برای تولید جدول ارزش وجود ندارد، ولی در بیشتر موارد می‌توان جدول ارزش را طی سه مرحله زیر به‌دست آورد.

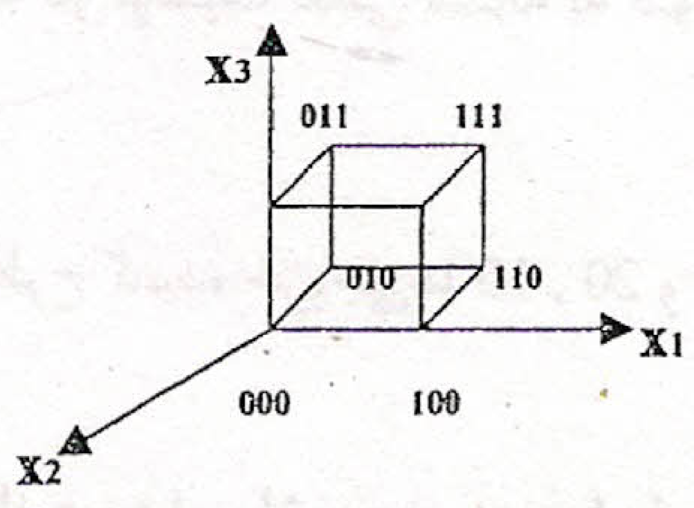
۱) ورودی‌ها را با متغیرهای دودویی مانند x_i نشان می‌دهیم.

۲) خروجی‌ها را با متغیرهای دودویی مانند z_i نشان می‌دهیم.

۳) خروجی متناظر با هر یک از ورودی‌ها را به‌دست می‌آوریم.

مثال ۱: مداری با سه ورودی و یک خروجی طرح کنید. به طوری که خروجی آن تنها هنگامی مساوی با یک باشد که ورودی مضربی از چهار باشد.

x_1	x_2	x_3	Z
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0



فضای سه بعدی

مثال ۲: مدار ترکیبی با 3 ورودی طرح کنید که بتواند جذر صحیح ورودی را تولید کند.

$$f(x)=\lfloor \sqrt{x} \rfloor$$

چون مقادیر ورودی 7,...,1,0 می‌باشد، بنابراین خروجی مدار می‌تواند یکی از سه مقادیر 2,1,0 باشد، بنابراین برای کدگذاری سه کمیت به 2-bit یعنی z_2, z_1 نیاز داریم.

x_1	x_2	x_3	z_1	z_2
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

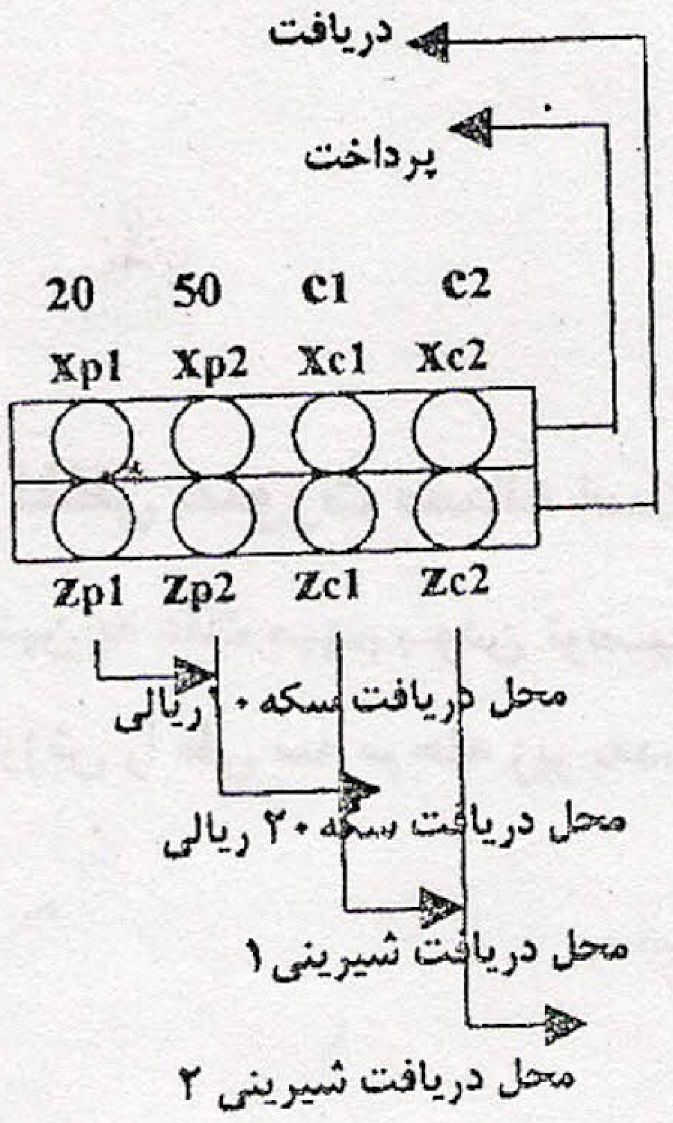
مثال ۳: ماشین خودکار برای توزیع شیرینی و باز پس دادن پول باقی‌مانده به شرح زیر طرح کنید:

ماشین دارای دو نوع شیرینی c_1 و c_2 به ترتیب به قیمت 20, 50 ریال می‌باشد.

شیرینی c_1 را می‌توان با قرار دادن سکه 20 ریالی در محل مربوطه و فشار دادن دکمه c_1 و یا با قرار دادن یک سکه 50 ریالی در محل مربوطه و فشار دادن c_1 و باز پس گرفتن یک سکه 10 ریالی و یک سکه 20 ریالی خریداری نمود.

شیرینی c_2 را می‌توان با قرار دادن یک سکه 50 ریالی در محل سکه شماره 2 و فشار دادن کلید c_2 خریداری نمود.

x_{p1}	x_{p2}	x_{c1}	x_{c2}	z_{p1}	z_{p2}	z_{c1}	z_{c2}
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	1	1	1	0
0	1	1	1	0	0	0	1
1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0
1	0	1	0	0	0	1	0
1	0	1	1	0	0	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	1	0	1
1	1	1	0	1	1	1	0
1	1	1	1	0	0	1	1



با توجه به جدول متوجه می‌شویم که خروجی مربوط به برخی از ورودی‌ها در توصیف لفظی مساله به صورت صریح بیان نشده‌اند ولی در جدول ارزش به صورت قابل قبول مشخص گردیده‌اند.

تمرین: ماشین خودکار برای خرد کردن یک سکه 10 تومانی و 5 تومانی طرح کنید، خروجی‌ها 10 , 20 و 50 ریالی هستند.

۲. دومین قدم در طراحی مدارهای ترکیبی به‌دست آوردن تابع بولی از روی جدول ارزش می‌باشد.

چند اصطلاح: در یک فضای n بعدی جمله حاصل‌ضربی را که در آن همه متغیرها تنها یک بار یا به صورت واقعی و یا به صورت مکمل ظاهر شده باشند Minterm یا جمله حاصل‌ضرب کامل و یا جمله مینیمم گویند.

مثال: با دو متغیر x_1, x_2 چند مین‌ترم یا جمله حاصل‌ضرب کامل می‌توان نوشت؟

4 جمله می‌توان نوشت: $\overline{x}_1\,\overline{x}_2, \overline{x}_1 x_2, x_1\overline{x}_2, x_1 x_2$

	x_1	x_2	$\overline{x}_1\overline{x}_2$	$\overline{x}_1 x_2$	$x_1\overline{x}_2$	$x_1 x_2$
m0	0	0	1	0	0	0
m1	0	1	0	1	0	0
m2	1	0	0	0	1	0
m3	1	1	0	0	0	1

با توجه به جدول متوجه می‌شویم که مین‌ترم‌ها متمایز هستند و هر مین‌ترم تنها به ازا یکی از ترکیبات ارزش ممکن برای متغیرها دارای مقدار 1 و به ازای بقیه دارای مقدار صفر می‌باشد.

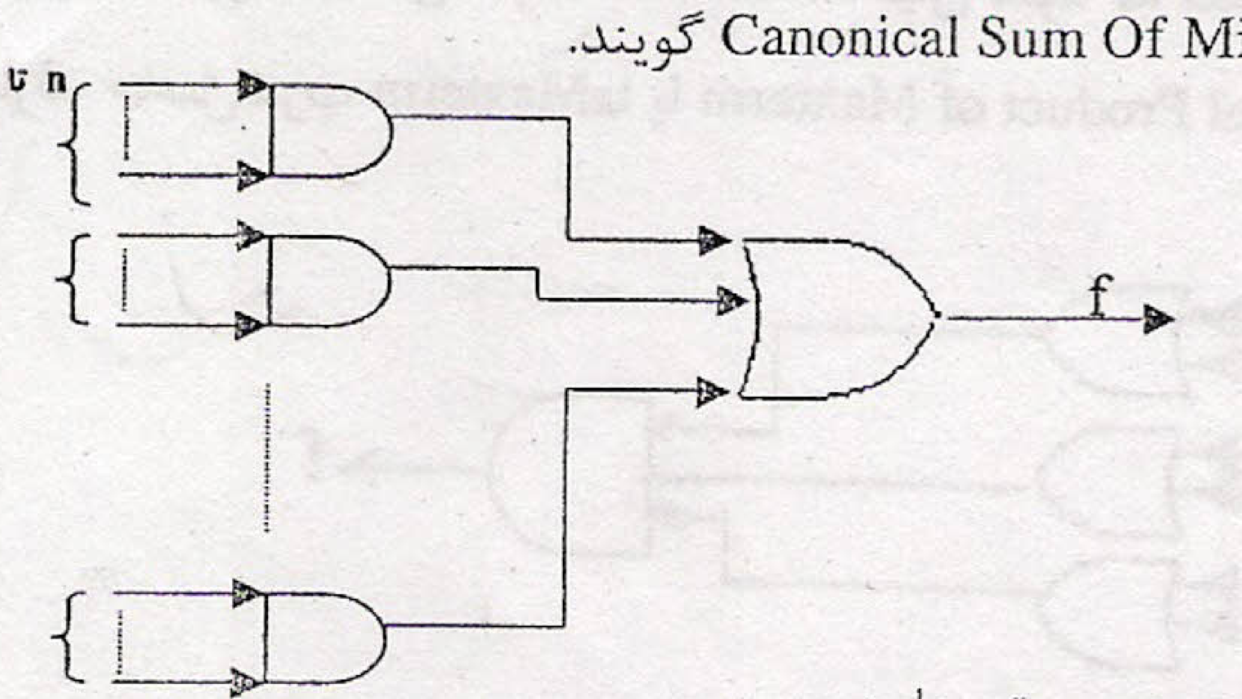
به طور کلی اگر n متغیر متمایز وجود داشته باشد، آن‌گاه 2^n مین‌ترم متمایز وجود خواهد داشت که آن‌ها را به ترتیب با $m_0, m_1, \ldots, m_{(2^n-1)}$ نشان می‌دهند. $x_1, x_2, x_3 \Rightarrow n = 3 \Rightarrow 2^n = 2^3 = 8 \Rightarrow m_0, m_1, \ldots, m_7$

و بر عکس هر m_i را می‌توان به مین‌ترم معادل آن تبدیل نمود و برای این منظور اندیس دهدهی، را به معادل دودویی آن تبدیل می‌کنیم در صورت نیاز تعدادی صفر به سمت چپ آن اضافه می‌کنیم تا تعداد صفر و یک‌ها مساوی تعداد متغیرها شود و سپس در هر جا یک ظاهر شده باشد، خود متغیر و اگر صفر ظاهر شده باشد مکمل متغیر را جای‌گزین می‌کنیم.

مثال: با فرض 3 متغیر x_1, x_2, x_3 مین‌ترم معادل m_2 را به‌دست آورید.

$$(2)_{10} = (?)_2 = (10)_2 = (010)_2 \Rightarrow m_2 = \overline{x}_1 x_2 \overline{x}_3$$

هر تابع بولی مانند $f(x_1, x_2, \ldots, x_n)$ را می‌توان به صورت مجموع مین‌ترم‌ها بیان نمود که مدار معادل آن به فرم دو سطحی AND – OR خواهد بود که در آن AND ها برای ایجاد مین‌ترم‌ها و OR برای ایجاد مجموع مین‌ترم‌ها به کار می‌رود. چنین پیاده‌سازی را فرم متعارف مجموع جملات مینیمم و یا Canonical Sum Of Minterms گویند.



مثال: با توجه به جدول ارزش، توابع بولی را به فرم متعارف مجموع مین‌ترم‌ها بنویسید.

x_1	x_2	x_3	f_1	f_2
0	0	0	1	1
0	0	1	0	0
0	1	0	1	-
0	1	1	0	-
1	0	0	1	1
1	0	1	1	1
1	1	0	0	0
1	1	1	0	0

$$f_1 = \overline{x}_1\overline{x}_2\overline{x}_3 + \overline{x}_1 x_2 \overline{x}_3 + x_1\overline{x}_2\overline{x}_3 + x_1\overline{x}_2 x_3 = \sum_1 (0,2,4,5) = \prod_0 (1,3,6,7)$$

توابع بولی را که به‌طور کامل مشخص نشده‌اند می‌توان با جدا کردن نقاط 1 تابع و نقاط نامشخص تابع به صورت زیر نمایش داد:

$$f_2 = \sum_1 (m_i) + \sum_d (m_j)$$

که در آن m_iها نشان‌دهنده نقاط -1 تابع و m_j ها نشان‌دهنده نقاط بی‌تفاوت (.d.c) تابع هستند.

پس تابع f_2 را به صورت زیر می‌توان نوشت.

$$f_2 = \sum_1 (0,4,5) + \sum_d (2,3) = \sum_1 (0,4,5) + d(2,3)$$

در یک فضای n بعدی جمله حاصل جمع را که در آن همه متغیرها تنها یک بار یا به صورت واقعی و یا به صورت مکمل ظاهر شده باشد، Maxterm یا جمله حاصل جمع کامل و یا جمله ماکزیمم گویند.

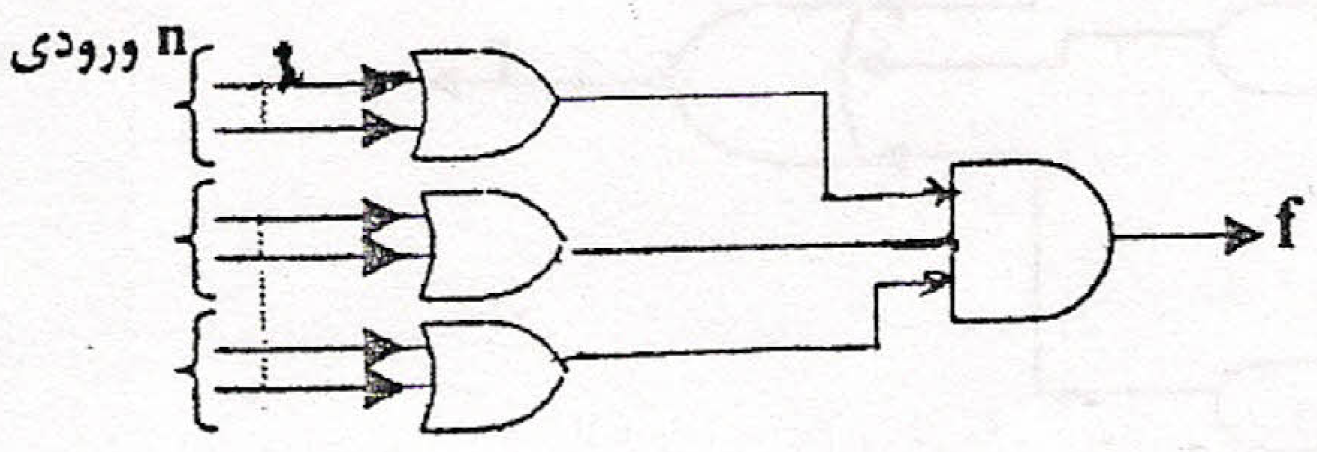
مثال: با دو متغیر x_1, x_2 چند جمله حاصل جمع کامل می‌توان نوشت:

	x_1	x_2	$\overline{x}_1 + \overline{x}_2$	$x_1 + x_2$	$x_1 + x_2$	$x_1 + x_2$
M_1	0	0	1	1	1	0
M_2	0	1	1	1	0	1
M_3	1	0	1	0	1	1
M_4	1	1	0	1	1	1

با توجه به جدول متوجه می‌شویم که Maxtermها متمایزند، هم‌چنین هر ماکسترم تنها به ازا یکی از ترکیبات دارای ارزش صفر می‌باشد و به ازا بقیه دارای ارزش یک می‌باشد.

۳. اگر n متغیر متمایز وجود داشته باشد، آن‌گاه 2^n – Maxterm متمایز وجود خواهد داشت که آن‌ها را به صورت $M_0, M_1, ..., M_{(2^n-1)}$ نمایش می‌دهیم.

هر تابع بولی را می‌توان به صورت حاصل ضرب Maxtermها بیان نمود که مدار معادل آن به صورت دو سطحی OR – AND خواهد بود و آن را فرم متعارف حاصل‌ضرب Maxtermها یا Canonical Product of Maxterm گویند.



مثال: با استفاده از جدول ارزش مثال قبل فرم متعارف حاصل‌ضرب Maxtermهای تابع را به‌دست آورد.

$$f_1 = (x_1 + x_2 + \overline{x}_3)(x_1 + \overline{x}_2 + \overline{x}_3)(\overline{x}_1 + \overline{x}_2 + x_3)(\overline{x}_1 + \overline{x}_2 + \overline{x}_3) = \Pi(1,3,6,7)$$

با توجه به مثال فوق متوجه می‌شویم اندیس‌هایی که در یک فرم متعارف ظاهر نشده، در فرم متعارف دیگر ظاهر می‌شود.

اگر تابع $\overline{f}$ را به صورت مجموع "Minterm"ها نشان دهیم یعنی:

$$\overline{f} = \sum_{l \in I} (m_l)$$

آن‌گاه با استفاده از قانون دمرگان داریم:

$$f = \overline{\overline{f}} = \overline{\sum (m_i)} = \Pi(\overline{m}_i) = \Pi(M_i)$$

$$\overline{m_i} = M_i$$

یعنی Maxtermها و Minterm ها مکمل یکدیگرند.

۴. ساده کردن توابع بولی:

توابع بولی را می‌توان به یکی از سه روش جبری، جدول کارنو و رویهٔ کویین مک کلاسکی ساده نمود.

روش جبری:

با استفاده از قوانین جبر بول می‌توان توابع ترکیبی را ساده نمود.

مثال: با استفاده از جدول ارزش زیر تابع را ساده نمایید:

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

فرم مجموع مین‌ترم‌ها $f=\bar{x}_1\bar{x}_2x_3+x_1\bar{x}_2x_3+x_1x_2x_3$

فرم مجموع حاصل‌ضرب‌ها $f=\bar{x}_2x_3\left(\bar{x}_1+x_1\right)+x_1x_2x_3=\bar{x}_2x_3+x_1x_2x_3$

فرم مجموع حاصل‌ضرب‌ها $f=x_1x_3\left(\bar{x}_2+x_2\right)+\bar{x}_1\bar{x}_2x_3=x_1x_3+\bar{x}_1\bar{x}_2x_3$

فرم بهینه مجموع حاصل‌ضرب‌ها $f=\bar{x}_1\bar{x}_2x_3+\left(x_1\bar{x}_2\bar{x}_3+x_1\bar{x}_2\bar{x}_3\right)+x_1x_2x_3=\bar{x}_2x_3+x_1x_3$

در این قسمت هدف ایجاد مدارهای دو سطحی است که از جنبه‌هایی بهینه باشند. ضوابط متعددی برای بهینگی وجود دارد که ما دو ضابطه زیر را در نظر می‌گیریم:

۱) مدار دو سطحی که در آن تعداد کل ورودی به GATE ها حداقل باشد.

۲) مدار دو سطحی که در آن تعداد کل GATE ها حداقل باشد و بین تمام مدارهایی که از نظر تعداد GATEها می‌نیمم باشند آن یکی که دارای کم‌ترین ورودی باشد.

برای رسیدن به این هدف ابتدا مدارهای دو سطحی موسوم به مجموع حاصل‌ضرب‌ها (Sum of Products) را در نظر می‌گیریم.

می‌دانیم هر مین‌ترم تنها یک نقطه از فضا را می‌پوشاند، بنابراین جمله حاصل‌ضربی که فاقد برخی از متغیرها باشد نمایشگر نقاط متعددی از فضا خواهد بود. مثلا فرض کنید جمله حاصل‌ضرب P (n-1 متغیری) در فضای n بعدی فاقد متغیر x_n باشد، آن‌گاه داریم:

$$P=P\cdot 1=P\left(x_n+\bar{x}_n\right)=P\cdot x_n+P\bar{x}_n$$

مثال: با فرض 3 متغیر x_1, x_2, x_3 جمله حاصل‌ضرب $\bar{x}_1x_3$ که فاقد متغیر x_2 می‌باشد، دو نقطه از فضا را می‌پوشاند (زیرا متغیر x_2 برای جمله‌ها حاصل‌ضرب don't – care می‌باشد، بنابراین متغیر x_2 خواه مساوی با صفر خواه مساوی با یک باشد، اثری در مقدار جمله حاصل‌ضرب ندارد).

$$P=\bar{x}_1x_3=\bar{x}_1x_3.1=\bar{x}_1x_3.\left(x_2+\bar{x}_2\right)=\bar{x}_1x_2x_3+\bar{x}_1\bar{x}_2x_3 \Rightarrow \left\{\begin{array}{ccc} x_1 & x_2 & x_3 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{array}\right.$$

به طور کلی در یک فضای n بعدی اگر جمله حاصل‌ضربی فاقد k متغیر باشد، آن‌گاه نشان‌دهنده 2^k نقطه از فضا خواهد بود.

مثال: با فرض فضای 3 بعدی (x_1, x_2, x_3) جمله $\overline{x}_2$ چهار نقطه از فضا را می‌پوشاند، زیرا متغیرهای x_3, x_1 برای جمله حاصل‌ضرب don't – care می‌باشد.

یعنی جمله $\overline{x}_2$ به ازای هر یک از چهار ترکیب $\begin{cases} x_1 & x_2 & x_3 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{cases} \Rightarrow P = \overline{x}_2$ دارای ارزش یک می‌باشد.

تعریف ۱: در یک فضای n بعدی، اگر جمله حاصل‌ضربی مانند P حداقل یک نقطه - 1 تابع را بپوشاند و هیچ نقطه – 0 تابع را نپوشاند، ایجاب کننده (Implicant) نامیده می‌شود. بنابراین مقدار تابع به ازا هر یک از نقاط پوشانده شده به‌وسیله ایجاب کننده یا مساوی با یک و یا نامشخص (Don't – care) خواهد بود.

تعریف ۲: ایجاب کننده P را ایجاب کننده نخستین (Prime Implicant) گویند، هرگاه برای هر ایجاب کننده دیگر مانند P' نقطه‌ای وجود داشته باشد که به‌وسیله P پوشانده بشود ولی به وسیله P' پوشانده نشود.

مثال: با استفاده از جدول ارزش داریم:

m_i	x_1	x_2	x_3	f
0	0	0	0	1
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	–
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

$\Rightarrow P_1 = \overline{x}_1\overline{x}_2 \Rightarrow \begin{cases} 0 & 0 & 0 \\ 0 & 0 & 1 \end{cases} \Rightarrow \begin{cases} f(0\,0\,0)=1 \\ f(0\,0\,1)=1 \end{cases} \Rightarrow$ P_1 یک ایجاب کننده هست.

$\Rightarrow P_2 = \overline{x}_1\overline{x}_2 \Rightarrow \begin{cases} 0 & 1 & 0 \\ 0 & 1 & 1 \end{cases} \Rightarrow \begin{cases} f(0\,1\,0)=0 \\ f(0\,1\,1)=1 \end{cases} \Rightarrow$ P_2 یک ایجاب کننده نیست زیرا یک نقطه -0 تابع را می‌پوشاند.

$\Rightarrow P_3 = \overline{x}_1\overline{x}_2 \Rightarrow \begin{cases} 1 & 0 & 0 \\ 1 & 0 & 1 \end{cases} \Rightarrow \begin{cases} f(1\,0\,0)=- \\ f(1\,0\,1)=1 \end{cases} \Rightarrow$ P_3 یک ایجاب کننده هست.

$P_4 = P_1 + P_3 = \overline{x}_2 \Rightarrow \begin{cases} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{cases} \Rightarrow \begin{cases} f(0\,0\,0)=1 \\ f(0\,0\,1)=1 \\ f(1\,0\,0)=- \\ f(1\,0\,1)=1 \end{cases}$ P_4 یک ایجاب کننده نخستین هست.

توجه داشته باشید هر ایجاب کننده نخستین یک جمله حاصل‌ضرب می‌باشد، بنابراین اگر بتوانیم تابع را به صورت مجموع حاصل‌ضرب‌ها بنویسیم، تعداد GATE ها کاهش می‌یابد و اگر جملات حاصل‌ضرب‌ها از نوع PI باشد، آن‌گاه تعداد ورودی به GATE ها نیز کاهش می‌یابد.

قضیه: در یک تابع ترکیبی f به فرم مجموع حاصل‌ضرب‌ها، اگر مدار از نظر تعداد کل ورودی‌ها و یا تعداد کل GATE ها بهینه باشد، آن‌گاه خروجی هر "AND – GATE" جمله حاصل‌ضربی است که برای تابع f یک Prime Implicant محسوب می‌شود.

اثبات: برهان خلف: فرض کنیم چنین نباشد یعنی مدار می‌نیمم برای تابع f وجود دارد که در آن خروجی یک "AND GATE" مانند G_1 جمله حاصل‌ضرب P_1 باشد که PI تابع نیست. در این صورت ایجاب کننده‌ای مانند P_2 باید وجود داشته باشد که همه نقاط پوشیده شده به‌وسیله P_1 را بپوشاند.

اگر "AND GATE" متناظر با P_2 یعنی G_2 را جای‌گزین G_1 کنیم، در رفتار مدار تغییری ایجاد نخواهد شد زیرا به ازا هر یک از نقاطی که به‌وسیله P_2 پوشانده می‌شود، مقدار تابع مساوی یک و یا نامشخص خواهد بود. حال اگر P_1 شامل K_1 متغیر باشد، جمله P_2 شامل $k_2 > k_1$ متغیر خواهد بود. زیرا P_2 نسبت به P_1 نقاط بیشتری را می‌پوشاند و در نتیجه با استفاده از G_2 تعداد کل ورودی‌ها تقلیل می‌یابد که با فرض قضیه متناقض است و در نتیجه قضیه ثابت می‌شود.

بنابراین مداری که از نظر تعداد ورودی‌ها و تعداد GATEها بهینه باشد باید به صورت مجموع ایجاب‌کننده‌های نخستین (PIهای) تابع نوشته شود به طوری که هر نقطه یک تابع حداقل به‌وسیله یکی از PI ها پوشانده شود.
(هر چند که در حالت کلی در مدار بهینه همه PI ها ظاهر نخواهد شد)

مثال: با توجه به جدول ارزش داریم:

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$$\left.\begin{array}{l}4-\text{AND}\\1-\text{OR}\\16-\text{Input}\end{array}\right\} \Rightarrow f=\bar{x}_1\bar{x}_2\bar{x}_3+\bar{x}_1x_2x_3+x_1x_2\bar{x}_3+x_1x_2x_3$$

مجموعه PI های تابع $=\{x_1x_2,\bar{x}_1x_3,x_2x_3\}$

$$\left.\begin{array}{l}3-\text{AND}\\1-\text{OR}\\9-\text{Input}\end{array}\right\} \Rightarrow f=x_1x_2+\bar{x}_1x_3+x_2x_3$$

بهینه $$\left.\begin{array}{l}2-\text{AND}\\1-\text{OR}\\5-\text{Input}\end{array}\right\} \Rightarrow f=x_1x_2+\bar{x}_1x_3$$

برای ایجاد مدارهای بهینه می‌توان رویه کلی ایجاد نمود.

رویه:

۱) مجموعه کامل از PIهای تابع f را به‌دست آورید.

۲) زیر مجموعه‌های با هزینه می‌نیمم از PIها انتخاب کنید به طوری که هر نقطه 1- تابع حداقل به‌وسیله یکی از PI ها پوشانده بشود.

۳) تابع f را به صورت مجموع PI های به‌دست آمده از مرحله دوم بنویسید و مدار آن را طرح کنید.

واضح است که هزینه یک PI مانند P_k با هدف نهایی بهینه‌سازی مدار ارتباط دارد.

یک PI با L متغیر ورودی به یک AND – GATE با L ورودی و یک خروجی به OR – GATE واقع در سطح دوم نیاز دارد. بنابراین اگر هدف بهینه‌سازی تعداد کل ورودی‌ها باشد، داریم:

$$C_2(P_L)=\begin{cases}L+1 & \text{اگر } L>1\\ L & \text{اگر } L=1\end{cases}$$

اگر هدف بهینه‌سازی تعداد کل GATE ها باشد، آن‌گاه داریم:

$$C_1(P_L)=\begin{cases}1 & \text{اگر } L>1\\ 0 & \text{اگر } L=1\end{cases}$$

تولید *PI* های تابع:

۱. روش جدولی: می‌دانیم هر جمله حاصل‌ضرب کامل (مین‌ترم) تنها یک نقطه از فضا را می‌پوشاند و از ترکیب دو مین‌ترم که فقط در یک لفظ $(x_j, \overline{x}_j)$ متفاوت باشند، می‌توان جمله حاصل‌ضربی ایجاد نمود که فاقد لفظ متفاوت (x_j) و شامل لفظهای ثابت می‌باشد که دو نقطه از فضا را خواهد پوشاند.

مثلاً مجموع دو جمله حاصل‌ضرب $P\overline{x}_1 + Px_1$ یک جمله حاصل‌ضرب n-1 متغیری P را به وجود می‌آورد که هر دو نقطه از فضا را می‌پوشانند. به طور مشابه اگر دو جمله n-k متغیری به جز یک لفظ در بقیه متغیرها یکسان باشند، با یکدیگر ترکیب و یک جمله حاصل‌ضرب n-k-1 متغیری تولید می‌کنند که 2^{k+1} نقطه از فضا را می‌پوشاند. مثلا در یک فضای 4 بعدی فرض کنید:

$$x_2x_3\overline{x}_4 + x_2\overline{x}_3\overline{x}_4 = x_2\overline{x}_4$$

ولی دو جمله حاصل‌ضرب $x_1x_2x_3$ و $x_2x_3x_4$ قابل ترکیب نیستند، زیرا در یک متغیر متفاوت هستند. PI های تابع بولی را می‌توان به طریق فوق با ترکیب مجموعه‌هایی از نقاط 1- و نامشخص به‌دست آورد. این رویه تا جایی ادامه پیدا می‌کند که دیگر امکان هیچ ترکیب جدیدی باقی نماند. این رویه را می‌توان به طریق سیستماتیک زیر بیان نمود.

۲. رویه کویین مک کلاسکی:

Binary Represention (a):

۱ـ در جدولی که ستون‌های آن با متغیرهای تابع یعنی $x_1, x_2, \ldots, x_n$ نام‌گذاری شده‌اند، نقاط 1- و نامشخص تابع را رده‌بندی می‌کنیم، به‌طوری‌که رده s_i شامل نقاط 1- و نامشخص تابع باشد که در ترکیب آن i تا یک و بقیه صفر می‌باشند.

۲ـ برای هر $i = 1, 2, \ldots, n-1$ هر عضو از رده s_i را با هر عضو از ردهٔ s_{i+1} مقایسه می‌کنیم. در صورتی که فقط در یک لفظ متفاوت باشند از ترکیب آن‌ها جمله حاصل‌ضربی می‌نویسیم که فاقد لفظ متفاوت و شامل لفظهای ثابت باشد. آن‌را در ردهٔ s'_i قرار می‌دهیم و نقاط ترکیب شده را با علامت ✓ مشخص می‌کنیم تا معلوم شود که Prime Implicant نیستند.

۳ـ عضوی از رده s'_i با عضوی از رده s'_{i+1} به دو شرط قابل ترکیب خواهد بود.

اولاًـ موضع don't-Care آن‌ها یکسان باشد (یعنی هر دو فاقد آن متغیر باشند).

ثانیاًـ فقط در یک لفظ متفاوت باشند.

از ترکیب آن‌ها جمله حاصل‌ضربی می‌نویسیم که فاقد لفظ متفاوت و شامل لفظهای ثابت باشد. آن را در رده s''_i قرار داده و نقاط قابل ترکیب را با علامت ✓ مشخص می‌کنیم. این عمل را تا جایی ادامه می‌دهیم که دیگر نتوان جمله حاصل‌ضرب جدیدی به‌دست آورد. از مرحله اول تا پایان جملات حاصل‌ضربی که علامت ✓ ندارد، مجموعه Prime Implicant تابع را تشکیل می‌دهند.

مثال: کلیه PI های تابع $f(x_1, x_2, x_3, x_4)$ که به‌وسیله جدول ارزش زیر تعریف شده‌اند را به کمک رویهٔ کویین مک کلاسکی به‌دست آورید:

	x_1	x_2	x_3	x_4	f
m_0	0	0	0	0	1
m_1	0	0	0	1	1
m_2	0	0	1	0	1
m_3	0	0	1	1	0
m_4	0	1	0	0	0
m_5	0	1	0	1	1
m_6	0	1	1	0	0
m_7	0	1	1	1	–
m_8	1	0	0	0	1
m_9	1	0	0	1	0
m_{10}	1	0	1	0	1
m_{11}	1	0	1	1	0
m_{12}	1	1	0	0	0
m_{13}	1	1	0	1	0
m_{14}	1	1	1	0	1
m_{15}	1	1	1	1	1

(1

رده	x_1	x_2	x_3	x_4		
s_0	0	0	0	0	m_0	✓
s_1	0	0	0	1	m_1	✓
	0	0	1	0	m_2	✓
	1	0	0	0	m_8	✓
s_2	0	1	0	1	m_5	✓
	1	0	1	0	m_{10}	✓
s_3	0	1	1	1	m_7	✓
	1	1	1	0	m_{14}	✓
s_4	1	1	1	1	m_{15}	✓

(2

	x_1	x_2	x_3	x_4		
S'_0	0	0	0	–	(m_0, m_1) $\Delta_1 = 1 = (1-0)$	
*	0	0	–	0	(m_0, m_2) $\Delta_1 = 2 = (2-0)$	✓
	–	0	0	0	(m_0, m_8) $\Delta_1 = 8$	✓
S'_1	0	–	0	1	(m_1, m_5) $\Delta_1 = 4$	
*	–	0	1	0	(m_2, m_{10}) $\Delta_1 = 4$	✓
	1	0	–	0	(m_8, m_{10}) $\Delta_1 = 2$	✓
S'_2 *	0	1	–	1	(m_5, m_7) $\Delta_1 = 2$	
*	1	–	1	0	(m_{10}, m_{14}) $\Delta_1 = 4$	
S'_3 *	–	1	1	1	(m_7, m_{15}) $\Delta_1 = 8$	
*	1	1	1	–	(m_{14}, m_{15}) $\Delta_1 = 1$	

(3

	x_1	x_2	x_3	x_4	
s''_0 *	–	0	–	0	m_0, m_2, m_8, m_{10}, $\Delta_1 = 2$, $\Delta_2 = 8$

مجموعه *PI* های تابع *f*: $\{\bar{x}_1\bar{x}_2\bar{x}_3 , \bar{x}_1\bar{x}_3 x_4 , \bar{x}_1 x_2 x_4 , x_1 x_3 \bar{x}_4 , x_2 x_3 x_4 , x_1 x_2 x_3 , \bar{x}_2\bar{x}_4\}$

توجه نمایید، آن‌هایی که علامت * دارند و حذف نشده‌اند مجموعه کامل از هفت PI را تشکیل می‌دهند.

:Decimal Representation (b)

رویه کویین مک کلاسکی را با استفاده از اندیس دهدهی Minterm ها نیز می‌توان انجام داد. دو مین‌ترم تنها هنگامی قابل ترکیب خواهند بود که تفاوت اندیس آن‌ها توان صحیحی از 2 باشد (مانند 2^k). در جمله حاصل‌ضرب به‌دست آمده از ترکیب کلیه متغیرهای دو مین‌ترم ظاهر می‌شوند، بجز متغیری که ارزش مکانی آن 2^k می‌باشد که باید حذف گردد. (متغیر زاید)

مثال: با فرض $f(x_1,x_2,x_3,x_4)=\sum(0,1,8,9)$ دو مین‌ترم m_1 , m_9 قابل ترکیب هستند، زیرا: $\Delta=9-1=8=2^3$ که در جمله حاصل‌ضرب به‌دست آمده متغیری که وزن آن مساوی 8 می‌باشد زاید است باید حذف گردد.

تذکر: در ترکیب دو جمله حاصل‌ضرب شرط این‌که تفاوت اندیس‌ها توان صحیحی از 2 باشد، لازم هست ولی کافی نیست زیرا دو استثنا وجود دارد:

۱) دو مین‌ترم که تفاوت اندیس آن‌ها توان صحیحی از 2 باشد ولی تعداد یک‌های موجود در اندیس کوچک‌تر بیش از تعداد یک‌های موجود در اندیس بزرگ‌تر باشد قابل ترکیب نخواهند بود. مثلا $m_7(0111), m_9(1001)$ قابل ترکیب نیستند. زیرا هر چند $\Delta_1=9-7=2^1$ چون در بیش از یک لفظ تفاوت دارند قابل ترکیب نیستند.

۲) دو مین‌ترم که تفاوت اندیس آن‌ها توان صحیح از 2 باشد ولی تعداد یک‌های موجود در دو اندیس با هم برابر باشند، آن‌گاه آن دو مین‌ترم قابل ترکیب نخواهند بود، مانند m_9 , m_5 زیرا هر چند $\Delta_1=9-5=4=2^2$ ولی تعداد یک‌های موجود در دو اندیس با هم برابرند.

ایجاب‌کننده‌هایی را که نمایشگر مجموعه‌های از 2^r نقطه هستند، می‌توان به‌وسیله مجموعه‌های از مین‌ترم‌ها در مبنای 10 نمایش داد. دو ایجاب کننده در صورتی با هم قابل ترکیب خواهند بود، اولاً Δ_1 آن‌ها یکسان، ثانیاً تفاوت اندیس‌های دو مجموعه مقداری ثابت و توان صحیحی از 2 باشد.

در این‌صورت تفاوت اندیس‌های دو مجموعه را با Δ_2 نشان می‌دهند که Δ_2 , Δ_1 نشان‌دهندهٔ مواضع متغیرهای حذف شده خواهد بود.

$$\left.\begin{array}{ll}(m_0 , m_2) & \Delta_1=2\\ (m_8 , m_{10}) & \Delta_1=2\end{array}\right\}\Rightarrow(m_0 , m_2 , m_8 , m_{10}) \qquad \Delta_1=2 \quad \Delta_2=8$$ (تفاوت ثابت بین اندیس‌های دو مجموعه)

$\Delta_1=2 \leftarrow$ تفاوت اولیه

$\Delta_2=8 \leftarrow$ تفاوت ثابت بین اندیس‌های دو مجموعه $\left.\begin{array}{l}m_8-m_0=8\\ m_2-m_{10}=8\end{array}\right\}$

مثال: کلیه PIهای تابع $f(x_1,x_2,x_3,x_4)=\sum_1(0,1,2,5,8,9,10,14,15)+\sum_d(7)$ را با استفاده از نمایش دهدهی برای اندیس مین‌ترم‌ها و با استفاده از روش کویین مک کلاسکی به‌دست آورید:

			x_1	x_2	x_3	x_4	
s_0	✓	m_0	0	0	0	0	✓
s_1	✓	m_1	0	0	0	1	
	✓	m_2	0	0	1	0	
	✓	m_8	1	0	0	0	
	✓	m_5	0	1	0	1	
s_2	✓	m_6	1	0	0	1	
	✓	m_{10}	1	0	1	0	
s_3	✓	m_7	0	1	1	1	
	✓	m_{13}	1	0	1	1	
s_4	✓	m_{15}	1	1	1	1	

			x_1	x_2	x_3	x_4
	S'_0	$\Delta=1(m_0, m_1)$	0	0	0	–
✓		$\Delta=2(m_0, m_2)$	0	0	–	0
✓		$\Delta=8(m_0, m_8)$	–	0	0	0
✓	S'_1	$\Delta=4(m_1, m_5)$	0	–	0	1
✓		$\Delta=8(m_1, m_9)$	–	0	0	1
✓		$\Delta=8(m_2, m_{10})$	–	0	1	0
✓		$\Delta=2(m_8, m_{10})$	1	0	–	1
✓	S'_2	$\Delta=2(m_7, m_7)$	0	1	–	1
✓		$\Delta=8(m_5, m_{13})$	–	1	0	1
✓		$\Delta=4(m_9, m_{13})$	1	–	0	1
✓	S'_3	$\Delta=8(m_7, m_{15})$	–	1	1	1
✓		$\Delta=2(m_{13}, m_{15})$	1	1	–	1

			x_1	x_2	x_3	x_4
S''_0	$\Delta=2$, $\Delta=8$	(0,2,8,10)	-	0	-	0
	$\Delta=8$, $\Delta=1$	(0,1,8,9)	-	0	0	-
S''_1	$\Delta=4$, $\Delta=5$	(1,5,9,13)	-	-	0	1
S''_2	$\Delta=2$, $\Delta=5$	(5,7,13,15)	-	1	-	1

مجموعه PI های تابع $=\left\{\bar{x}_2\bar{x}_4, \bar{x}_2\bar{x}_3, \bar{x}_3\bar{x}_4, x_2x_4\right\}$

تمرین ۱: PI های تابع $f=\sum_{1}(0,1,4,5,6,7,15)+\sum_{d}(8,12)$ را به روش QMC به‌دست آورید.

تمرین ۲: برای توزیع یک لیوان شربت با یخ به قیمت 15 ریال مداری طرح کنید.
(ورودی‌ها 5 و 10 و 20 ریالی).

تولید *PI* های تابع به کمک جدول کارنو: *(Karnough Map)*

از جدول کارنو می‌توان برای ساده کردن توابع بولی استفاده نمود که حداکثر شامل 6 متغیر باشند.

۱ـ جدول کارنو در واقع نمایش دو بعدی از یک فضای n بعدی است که در آن به ازای هر مین‌ترم یک خانه وجود دارد.

۲ـ جدول کارنو برای تابع n متغیری شامل 2^n خانه خواهد بود که در آن خانه‌های مجاور (دو خانه که در یک ضلع مشترکند) فقط در یک لفظ متفاوتند.

۳ـ در جدول کارنو مین‌ترم متناظر با هر خانه را می‌توان با بررسی بر چسب‌های ستون و سطر آن خانه به‌دست آورد.

۴ـ در جدول کارنو دقیقا نصف خانه ها مربوط به $x_i=1$ و نصف دیگر مربوط به $x_i=0$ می‌باشد.

۵ـ در جدول کارنو هر خانه دقیقاً با n خانه دیگر مجاور است.

۶ـ در جدول کارنو سطرهای اول و آخر و ستون‌های چپ و راست با یکدیگر مجاورند.

۷ـ هر تابع بولی را می‌توان با درج نقاط 1 و نامشخص تابع در جدول کارنو نمایش داد (با درک این نکته که مقدار تابع به ازای سایر نقاط صفر می‌باشد).

$x_2 \backslash x_1$	0	1
0	00 m_0 $\bar{x}_1\bar{x}_2$	10 m_1 $x_1\bar{x}_2$
1	01 m_1 $\bar{x}_1 x_2$	11 m_3 $x_1 x_2$

جدول کارنو دومتغیری

$x_3 \backslash x_1x_2$	00	01	11	10
0	m_0	m_2	m_6	m_4
1	m_1	m_3	m_7	m_5

جدول کارنو تابع سه متغیری

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

جدول کارنو تابع چهار متغیری

تمرین: جدول کارنو برای $n = 5$, $n = 6$ را به‌دست آورید. مین‌ترم متناظر با هر خانه را به صورت اندیس‌دهی مشخص نمایید و خانه‌های مجاور با هر خانه را معین نمایید.

مثال: تابع زیر را به وسیله جدول کارنو نمایش دهید:

$$f = \sum_{1}(0,1,6) + \sum_{d}(3)$$

	x_1	x_2	x_3	f
m_0	0	0	0	1
m_1	0	0	1	1
m_2	0	1	0	0
m_3	0	1	1	–
m_4	1	0	0	0
m_5	1	0	1	0
m_6	1	1	0	1
m_7	1	1	1	0

	00	01	11	10
0	0 1	2	6 1	4
1	1 1	3 –	7	5

نکته مهم: چون ایجاب‌کننده‌های یک تابع از ترکیب نقاط 1 و نامشخص تابع به‌وجود می‌آیند که تنها در یک لفظ تفاوت دارند، بنابراین چنین جملاتی در جدول کارنو به صورت گروهی از نقاط 1 و نامشخص مجاور به هم ظاهر می‌شوند، لذا PI های یک تابع را می‌توان با پیدا کردن تمام گروه های 2^k خانه‌ای مجاور که یکی به طور کامل در داخل گروه دیگر نباشند به‌دست آورد.

در واقع برای پیدا کردن Prime Implicatant ها از جدول کارنو باید هر چهار نکته زیر را رعایت نمود:

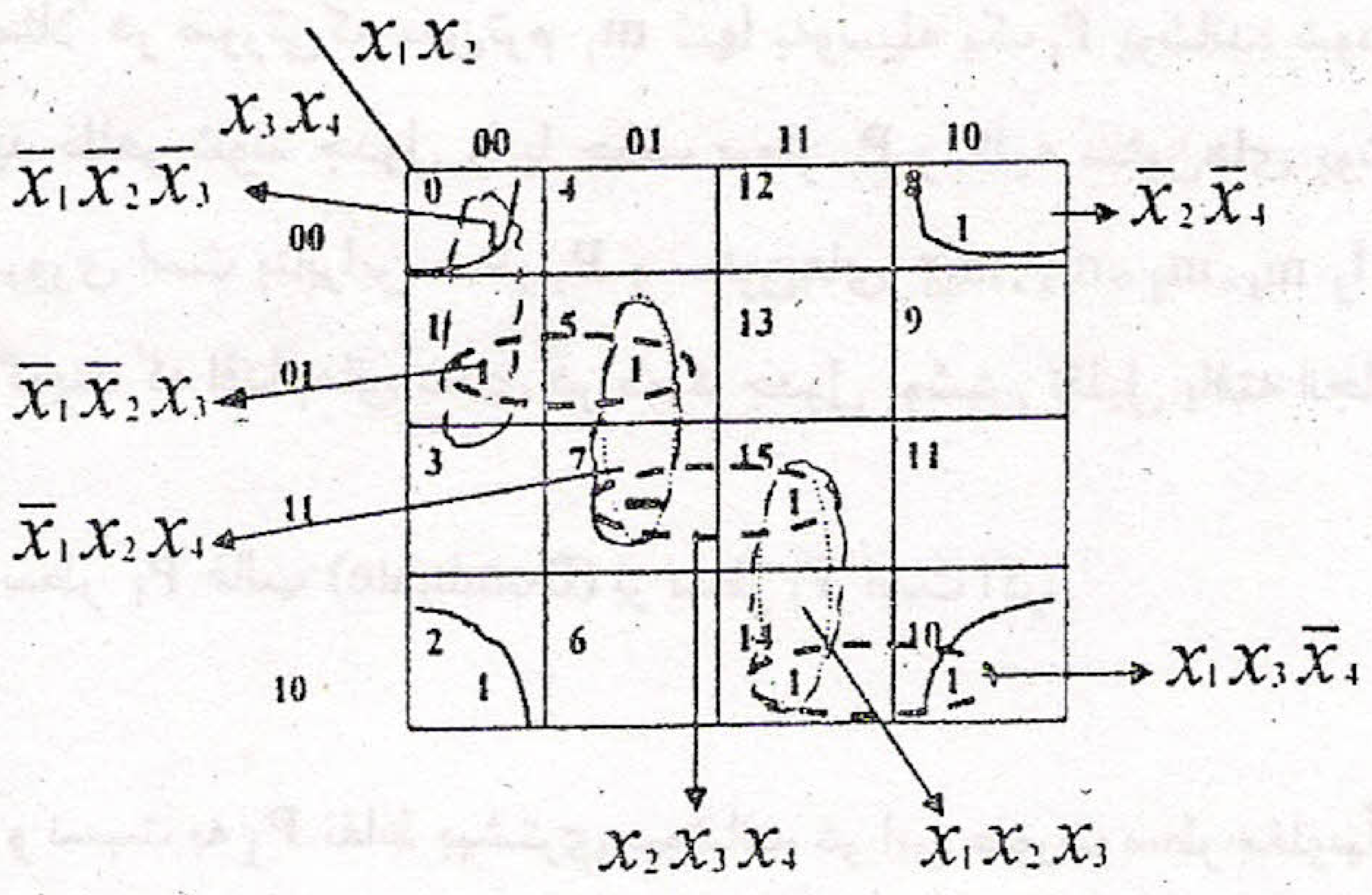
۱) خانه‌های متناظر با نقاط 1 و don't-care مجاور به هم را انتخاب می‌کنیم.

۲) تعداد خانه‌ها باید توان صحیحی از 2 باشد.

۳) تعداد خانه‌های انتخاب شده باید حداکثر تعداد ممکن باشد.

۴) گروهی به‌طور کامل در داخل گروه دیگر نباشد.

مثال: با استفاده از جدول کارنو کلیه PI های تابع $f(x_1,x_2,x_3,x_4)=\sum_1(0,1,2,5,8,9,10,14,15)+\sum_d(7)$ را به‌دست آورید.



انتخاب یک پوشش می‌نیمم از مجموعه *PI* های تابع:

مساله پوشش عبارت است از انتخاب حداقل تعداد از PIهای تابع با کمترین هزینه به‌طوری‌که هر نقطه ‎-1 تابع حداقل به‌وسیله یکی از PI های تابع پوشانده شود. انتخاب چنین پوششی را می‌توان به کمک جدول پوشش انجام داد. در جدول پوشش به ازا هر PI یک سطر و به ازای هر یک از نقاط ‎-1 تابع یک ستون وجود دارد. دو ستون اضافی c_1 و c_2 نیز در ارتباط با هزینه‌های مربوط به تعداد ورودی‌ها و تعداد gate ها در جدول پوشش ظاهر می‌شوند. در تقاطع سطر iام و ستون jام، یک درج می‌کنیم. اگر ایجاب کننده نخستین p_i مین‌ترم m_j را بپوشاند.

توجه داشته باشیم که نقاط بی‌تفاوت در جدول پوشش درج نمی‌شوند، زیرا نیازی به پوشش آن‌ها نیست.

مثال: جدول پوشش تابع $f=\sum_i(0,1,2,5,8,9,10,14,15)+\sum_d(7)$ را به‌دست آورید.

مجموعه‌های *PI* های تابع *f*:

مجموعه PI های تابع f $=\{\overline{x}_1\overline{x}_2\overline{x}_3,\ \overline{x}_1\overline{x}_3x_4,\ \overline{x}_1x_2x_4,\ x_1x_3\overline{x}_4,\ x_1x_2x_3,\ x_2x_3x_4,\ \overline{x}_2\overline{x}_4\}$

جدول پوشش تابع *f*:

		m_0	m_1	m_2	m_5	m_8	m_{10}	m_{14}	m_{15}	c_1	c_2
P_1	$\overline{x}_1\overline{x}_2\overline{x}_3$	1	1								
P_2	$\overline{x}_1\overline{x}_3x_4$		1		1						
P_3	$\overline{x}_1x_2x_4$				1						
P_4	$x_1x_3\overline{x}_4$						1	1			
P_5	$x_1x_2x_3$							1	1		
P_6	$x_2x_3x_4$								1		
P_7	$\overline{x}_2\overline{x}_4$	1		1		1	1				

طی قواعد زیر می‌توان جدول پوشش را ساده کرد و در نتیجه مدار مینیممی را تولید نمود:

۱) **حذف PI های ضروری:** جدول پوشش را ستون به ستون بررسی می‌کنیم، در صورتی‌که ستونی فقط یک درآیه یک داشته باشد، PI نظیر آن را PI ضروری گویند. مثلاً در صورتی‌که مین‌ترم m_j تنها به‌وسیله یک P_i پوشانده شود، آن را PI ضروری گویند، به مفهوم این‌که در هر پوشش می‌نیمم باید ظاهر شود. جدول را با حذف سطر P_i و کلیه ستون‌های پوشانده شده به‌وسیله آن ساده می‌کنیم. مثلاً در مورد مثال فوق P_7 ضروری است بنابراین سطر P_7 و ستون‌های m_1, m_2, m_8, m_{10} را از جدول حذف کنیم. آن‌چه که باقی ماند جدول پوشش تقلیل یافته گویند که اقدام‌های بعدی در مورد جدول پوشش تقلیل یافته انجام خواهد گرفت.

۲) **حذف سطرهای مغلوب:** گوییم سطر P_i غالب (Dominate) بر سطر P_j است اگر:

الف: $c_1(p_i) \le c_1(p_j)$

ب: P_i کلیه نقاط متناظر با P_j را بپوشاند و نسبت به P_j نقاط بیشتری بپوشاند، در این صورت سطر مغلوب یعنی p_j را حذف می‌کنیم.

(یعنی در هر موضعی که سطر مغلوب یک داشته باشد، در همان مواضع نیز سطر غالب یک داشته باشد ولی تعداد یک‌های سطر غالب بیشتر از تعداد یک‌های مغلوب می‌باشد).

مثلا در مثال فوق P_2 غالب بر P_1, P_3 و هم‌چنین P_5 غالب بر P_4, P_6 می‌باشد، بنابراین P_1, P_3, P_4, P_6 را از جدول حذف می‌کنیم. بعد از اعمال قاعده 2 می‌توان برای پیدا کردن P_i ضروری دوباره از قاعده 1 استفاده کرد. مثلا در مورد مثال فوق، پس از اعمال مرحله 2، P_5, P_2 نیز PI ضروری و در نتیجه عضوهایی از پوشش می‌نیمم خواهد بود. در صورتی‌که چند سطر یکسان وجود داشته باشد به جز یکی بقیه را از جدول حذف می‌کنیم.

۳) **حذف ستون‌های غالب:** گوییم ستون m_i غالب بر m_j است که هر سطری که m_j یک دارد، m_i را نیز یک داشته باشد و m_i نسبت به m_j نقاط یک بیشتری داشته باشد، در این صورت سطری‌که m_j را می‌پوشاند m_i را نیز خواهد پوشاند و بنابراین ستون غالب یعنی m_i را از جدول حذف می‌کنیم.

مثال فوق در مرحله 2 به نتیجه می‌رسد و نیاز به حذف ستون غالب ندارد و پوشش می‌نیمم (بهینه) عبارت است از $\{P_2, P_5, P_7\}$.

جدول دوره‌ای: *Cycle table*

جدول پوششی را که نتوان با استفاده از قواعد 2،1 و 3 ساده نمود، جدول دوره‌ای گویند.

مثلاً این جدول دوره‌ای است:

	m_a	m_b	m_c	m_d	m_e	c
P_1	1	1		1		k
P_2		1	1			k
P_3	1		1			k
P_4				1	1	k
P_5			1		1	k
P_6		1			1	k

دو روش کلی برای به‌دست آوردن پوشش می‌نیمم برای جداول دوره‌ای وجود دارد:

۱. **رویه انشعاب:** ستونی را به دلخواه مانند m_i (معمولاً با تعداد یک‌های کمتر) انتخاب می‌کنیم و مجموعه PI هایی که آن را می‌پوشانند R می‌نامیم. عضوی از مجموعه R را به دلخواه انتخاب می‌کنیم و جدول را با حذف آن سطر و کلیه ستون‌های پوشانده

شده به‌وسیله آن ساده می‌کنیم. اگر جدول از حالت دوره‌ای خارج شود با استفاده از سه قاعده قبل، پوشش بهینه و شامل این عضو را به‌دست می‌آوریم. همین عمل را برای تک تک عضوهای باقی‌مانده از R تکرار می‌کنیم و پوشش‌های به‌دست آمده را از نظر هزینه با هم مقایسه می‌کنیم و آن یکی را که دارای کمترین هزینه باشد به عنوان پوشش بهینه با هزینه می‌نیمم برای جدول دوره‌ای انتخاب می‌کنیم. رویه انشعاب را می‌توان برای هر یک از ستون‌ها اعمال نمود و هم‌چنین از عمل انشعاب در چندین سطح می‌توان استفاده نمود. مثلاً در مورد مثال فوق، ستون m_a را انتخاب می‌کنیم که برای پوشاندن آن انتخاب یکی از دو ایجاب کننده نخستین P_3, P_1 ضروری است. یعنی $R=\{P_1, P_3\}$

الف : P_1 را انتخاب می‌کنیم و پوشش بهینه شامل P_1 را به‌دست می‌آوریم که با حذف سطر P_1 و ستون‌های m_a, m_b, m_d داریم:

	m_c	m_e	c
P_2	1		k
P_3	1		k
P_4		1	k
P_5	1	1	k
P_6		1	k

جدول پوشش تقلیل یافته

در جدول پوشش تقلیل یافته سطر P_5 غالب بر P_2, P_3, P_4, P_6 خواهد بود، در نتیجه مجموعه $\{P_1, P_5\}$ پوشش می‌نیمم با هزینه 2k خواهد بود.

ب . حال P_3 را انتخاب و پوشش بهینه شامل P_3 را به‌دست می‌آوریم:

	m_b	m_d	m_e	c
P_1	1	1		k
P_2	1			k
P_3		1		k
P_4			1	k
P_5			1	k
P_6	1		1	k

جدول پوشش تقلیل یافته

سطرهای P_1 و P_6 غالب بر P_2 است و سطر P_4 غالب بر P_5 پس پوشش بهینه شامل P_3 و دو سطر از مجموعه سطرهای P_1, P_6, P_4 خواهد بود، بنابراین پوشش‌های به‌دست آمده عبارتند از:

$\{P_1, P_5\}$ با هزینه 2K

$\{P_1, P_3, P_4\}$ با هزینه 3K

$\{P_3, P_4, P_6\}$ با هزینه 3K

و در نتیجه پوشش $\{P_1, P_5\}$ که دارای حداقل هزینه می‌باشد به عنوان پوشش بهینه انتخاب می‌شود.

۲. روش پتریک : برای هر P_i متغیر بولی p_i را به صورت زیر تعریف می‌کنیم:

$$\text{متغیر بولی } p_i = \begin{cases} 1 : \text{اگر } P_i \text{ جزء پوشش می‌نیمم انتخاب شده باشد.} \\ 0 : \text{است اگر } P_i \text{ جزء پوشش می‌نیمم انتخاب نشده باشد.} \end{cases}$$

در این صورت ستون m_j به‌وسیله تعدادی P_i پوشیده می‌شود که شرط پوشش برای ستون m_j را با عبارت بولی $\sum_{i \in I} P_i = 1$ و در نتیجه در حالت کلی شرط پوشش برای همه ستون‌های یک جدول دوره‌ای m ستونی به صورت:

$$\prod_{i=1}^{m}\left(\sum_{i \in IJ} P_I\right) = 1$$

که اگر آن را به فرم معادل مجموع حاصل‌ضرب‌ها تبدیل کنیم، هر جمله حاصل‌ضرب به‌دست آمده نشان دهنده یک پوشش برای جدول خواهد بود که هر کدام از آن‌ها تمام نقاط یک تابع را خواهند پوشاند. در نتیجه جمله حاصل‌ضربی را که شامل تعداد کمتر از متغیر بولی باشد به عنوان پوشش بهینه انتخاب می‌کنیم. در مورد مثال فوق داریم:

$$(p_1 + \overset{m_a}{p_3})(p_1 + \overset{m_b}{p_2} + p_6).(p_1 + \overset{m_d}{p_4})(p_4 + \overset{m_e}{p_5} + p_6).(p_2 + \overset{m_c}{p_3} + p_5) = 1$$

و با استفاده از توزیع‌پذیری داریم:

$$p_1 p_5 + p_1 p_2 p_4 + p_1 p_2 p_4 + p_1 p_2 p_0 + p_2 p_3 p_4 + p_3 p_4 p_6 + p_1 p_3 p_4 + p_1 p_3 p_6 = 1$$

در نتیجه جواب همان پوشش $\{P_1, P_5\}$ خواهد بود که با استفاده از رویه انشعاب نیز قبلاً به‌دست آوریم.

تمرین: برای تابع $f(x_1, x_2, x_3, x_4) = \sum_i (0,3,4,6,9,13,15) + \sum_i (1,5,11,12,14)$

۱) با استفاده از رویه QMC مجموعه PI ها و پوشش با هزینه می‌نیمم را به‌دست آورید.

۲) با استفاده از جدول کارنو PI ها و پوشش با هزینه می‌نیمم را به‌دست آورید.

نکته مهم: با استفاده از جدول کارنو می‌توان دو مرحله تولید و انتخاب حداقل تعداد از PI ها را در هم ادغام نمود و بدون استفاده از جدول پوشش، طی یک مرحله پوشش بهینه برای تابع به‌دست آورد. برای این منظور ابتدا به سراغ خانه‌ای می‌رویم که برای پوشاندن آن فقط یک PI وجود دارد و این عمل را تا جایی ادامه می‌دهیم که PI ضروری وجود نداشته باشد. سپس به ترتیب قواعد سطر غالب و مغلوب را اعمال می‌کنیم.

مثال: برای تابع زیر، عبارت مجموع حاصل‌ضرب‌ها با هزینه می‌نیمم را به کمک جدول کارنو به‌دست آورید.

$$f(x_1, x_2, x_3, x_4) = \sum_i (4,5,7,12) + \sum_d (8,10,15,14)$$

x_3x_4 \ x_1x_2	00	01	11	10
00	0	4 1	12 1	8 –
01	1	5 1	13	9
11	3	7 1	15 –	11
10	2	6	14 –	10 –

$$f = \bar{x}_2 x_3 \bar{x}_4 + \bar{x}_1 x_2 x_4 + x_2 \bar{x}_3 \bar{x}_4 \rightarrow \begin{cases} 3 \text{ and} \\ 1 \text{ or} \\ 12 \text{ input} \end{cases}$$

برای توابع کاملاً مشخص شده این حدس وجود دارد که مجموع حاصل‌ضرب‌ها با تعداد ورودی می‌نیمم از نظر تعداد gate ها نیز می‌نیمم باشد و بالعکس. ولی این مطلب برای توابعی که کاملاً مشخص نشده‌اند صادق نیست.

مثال: تابع $f(x_1, x_2, ..., x_{10})$ را در نظر بگیرید که به صورت زیر تعریف شده است:

$$f(0000000000) = f(1100000000) = 1$$

$$f(1000000000) = f(0100000000) = \text{don't care}$$

$$f(1,0,x_3,x_4,.....,x_{10}) = 0 \text{ اگر } (x_3+x_4+x_5+...+x_{10}) = 1$$

$$f(0,1,x_3,x_4,......,x_{10}) = 0 \text{ اگر } (x_3+x_4+x_5+....+x_{10}) = 1$$

و به ازا بقیه نقاط مقدار تابع نامشخص می‌باشد. یعنی:

$$\left.\begin{matrix} f(00,x_3,....,x_{10}) = - \text{ if } x_3+x_4+....+x_{10} = 1 \\ f(11,x_3,....,x_{10}) = - \text{ if } x_3+x_4+....+x_{10} = 1 \end{matrix}\right\}$$

مثلاً از ترکیب چهار نقطه اول یعنی دو نقطه -1 و دو نقطه .d.c دو متغیر x_2, x_1 حذف و هشت متغیر ظاهر می‌شود. یعنی ایجاب‌کننده نخستین $P_3 = \overline{x}_3\,\overline{x}_4 \ldots \overline{x}_{10}$ به‌دست می‌آید.

هم‌چنین از ترکیب نقطه -1 یعنی $f(0000\ldots0) = 1$ و 255 نقطه .d.c تابع یعنی $f(00, x_3, ..., x_{10}) = -$ که در مجموع 256 نقطه است، می‌توان PI ای را به‌دست آورد که فاقد 8 متغیر و شامل 2 متغیر $x_2 \cdot x_1$ یعنی $\overline{x}_1.\overline{x}_2$ باشد. به همین ترتیب می‌توان x_1x_2 را نیز از ترکیب نقطه -1 تابع یعنی $f(11,00\ldots0) = 1$ و 255 نقطه .d.c تابع یعنی $f(11, x_3, \ldots, x_{10}) = -$ به‌دست آورد.

$$P_3 = \overline{x}_3\overline{x}_4\overline{x}_5,....,\overline{x}_{10}\ ,\ P_2 = x_1x_2\ ,\ P_1 = \overline{x}_1\overline{x}_2$$

	m_0	m_2	c_1	c_2
P_1	1			
P_2		1		
P_3	1	1		

اگر مدار با حداقل تعداد ورودی‌ها را در نظر می‌گیریم:

$$f = \overline{x}_1\overline{x}_2 + x_1x_2 \left\{\begin{matrix} 2-\text{AND} \\ 1-\text{OR} \\ 6-\text{input} \end{matrix}\right.$$

و اگر هدف مدار با حداقل تعداد gate ها باشد:

$$f = \overline{x}_3\overline{x}_4....\overline{x}_{10} \left\{\begin{matrix} 1-\text{AND} \\ 8-\text{input} \end{matrix}\right.$$

بنابراین برای تابع بولی که به‌طور کامل مشخص نشده‌اند ممکن است مداری طرح کنیم که در نظر تعداد Gateها بهینه باشد ولی از نظر تعداد inputها بهینه نباشد و بالعکس.

طراحی مدار بهینه با استفاده از حاصل‌ضرب مجموع‌ها: *Product Of Sums*

مدار دو سطحی بهینه متناظر با حاصل‌ضرب مجموع‌های یک تابع را می‌توان با استفاده از مجموع حاصل‌ضرب‌های تابع $\overline{f}$ و اعمال قوانین دمورگان به صورت زیر به‌دست آورد:

اگر

$$\overline{f} = \sum_{i \in I} P_i$$

باشد آن‌گاه:

$$f = \overline{\overline{f}} = \overline{\sum_{i \in I} P_i}$$

که در آن P_i یک تابع PI برای تابع $\overline{f}$ می‌باشد.

رویه:

۱. تابع بولی $\overline{f}$ را به صورت زیر تعریف می‌کنیم:

برای نقطه m_i اگر $f(m_i)=1$ آن‌گاه $\overline{f}(m_i)=0$

هم‌چنین اگر $f(m_i)=0$ آن‌گاه $\overline{f}(m_i)=1$

و اگر $f(m_i)=-$ آن‌گاه $\overline{f}(m_i)=-$

۲. برای تابع $\overline{f}$ مجموع حاصل‌ضرب‌ها یا $\overline{f}=\sum_{i\in I}P_1$ با هزینه می‌نیمم را به‌دست می‌آوریم.

۳. با اعمال قانون دمورگان به عبارت بولی $\overline{f}$ به فرم می‌نیمم مجموع حاصل‌ضرب‌ها، می‌توان عبارت بولی را به فرم می‌نیمم حاصل‌ضرب مجموع‌ها به‌دست آورد.

تذکر: به عوض رویه فوق، برای به‌دست آوردن فرم می‌نیمم حاصل‌ضرب مجموع‌ها برای تابع f، می‌توان نقاط -0 و don't –care تابع f را با هم ترکیب نمود.

تذکر: در حالت کلی برای تابع f فرم حاصل‌ضرب مجموع‌ها و مجموع حاصل‌ضرب‌ها دارای هزینه یکسان نمی‌باشند. بنابراین باید هر دو فرم را برای تابع f به‌دست آورد. آن‌گاه فرمی با کمترین هزینه را به منظور پیاده‌سازی مدار انتخاب شود.

مثال: با استفاده از جدول کارنو فرم می‌نیمم حاصل‌ضرب مجموع‌ها و مجموع حاصل‌ضرب‌ها را برای تابع زیر به‌دست آورید.

$$f(x_1,x_2,x_3,x_4)=\sum_1(3,4,6,9,11,12,13)+\sum_d(7)$$

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00	0	4 1	12 1	8
01	1	5	13 1	9 1
11	3 1	7 –	15	11 1
10	2	6 1	14	10

$\Rightarrow$

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00	0 1	4	12	8 1
01	1 1	5 1	13	6
11	3	7 –	15 1	10
10	2 1	6	14 1	11 1

$$\left.\begin{matrix}4-\text{AND}\\1-\text{OR}\\16-\text{input}\end{matrix}\right\} f=\overline{x}_1x_3x_4+\overline{x}_1x_2\overline{x}_4+\overline{x}_1x_2\overline{x}_3+x_1\overline{x}_2x_3$$

$$\overline{f}=\overline{x}_2\overline{x}_4+\overline{x}_1\overline{x}_3x_4+x_1x_2x_3$$

$$\left.\begin{matrix}3 & \text{OR}\\1 & \text{AND}\\11 & \text{input}\end{matrix}\right\} f=\overline{\overline{f}}=(x_2+x_4)(x_1+x_3-\overline{x}_4)(\overline{x}_1+\overline{x}_2+\overline{x}_3)$$

البته می‌توان با ترکیب نقاط -0 و d.c تابع طی یک مرحله فرم می‌نیمم حاصل‌ضرب مجموع‌ها را به‌دست آورد.

x_3x_4 \ x_1x_2	00	01	11	10
00	0: 0	4	12	8: 0
01	1: 0	5: 0	13	9
11	3	7: –	15: 0	11
10	2: 0	6	14: 0	10: 0

$$f=(x_2+x_4).(x_1+x_3+\overline{x}_4).(\overline{x}_1+\overline{x}_2+\overline{x}_3)$$

مثال: برای تابع f که در جدول کارنو تعریف شده است، مدار دو سطحی می‌نیمم به فرم حاصل‌ضرب مجموع‌ها به‌دست آورید:

x3x4 \ x1x2	00	01	11	10
00	0: 1	4: 1	12: 1	8: 0
01	1: 0	5: 1	13: 0	9: 0
11	3: 0	7: 0	15: 1	11: 0
10	2: 0	6: 0	14: 0	10: 0

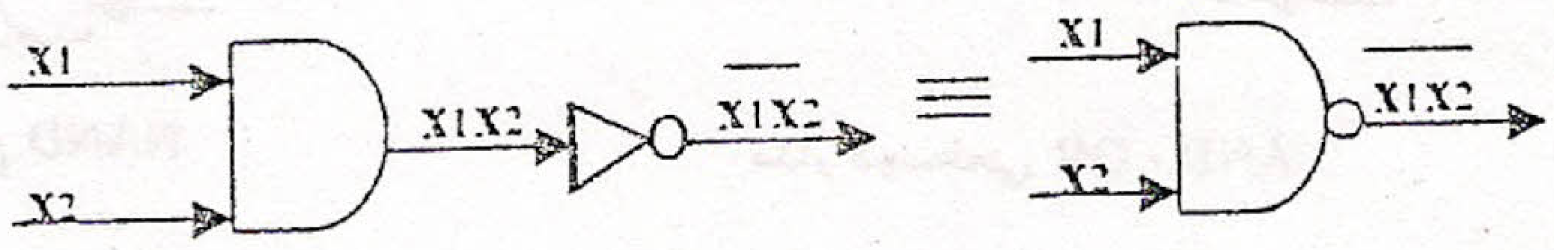
$$f=(\overline{x}_3+x_4)(\overline{x}_1+x_2)(x_1+x_3)(x_2+\overline{x}_4)(\overline{x}_1+x_3+\overline{x}_4)$$

چند *Gate* دیگر:

۱. NAND – Gate:

ترکیبی است از NOT و AND

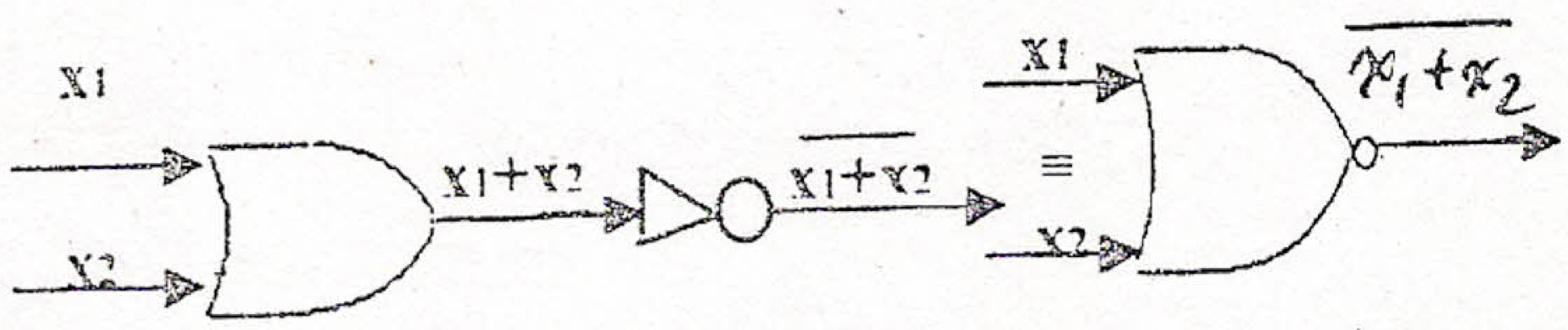
x_1	x_2	$\overline{x_1x_2}$
0	0	1
0	1	1
1	0	1
1	1	0



۲. NOR ـ Gate:

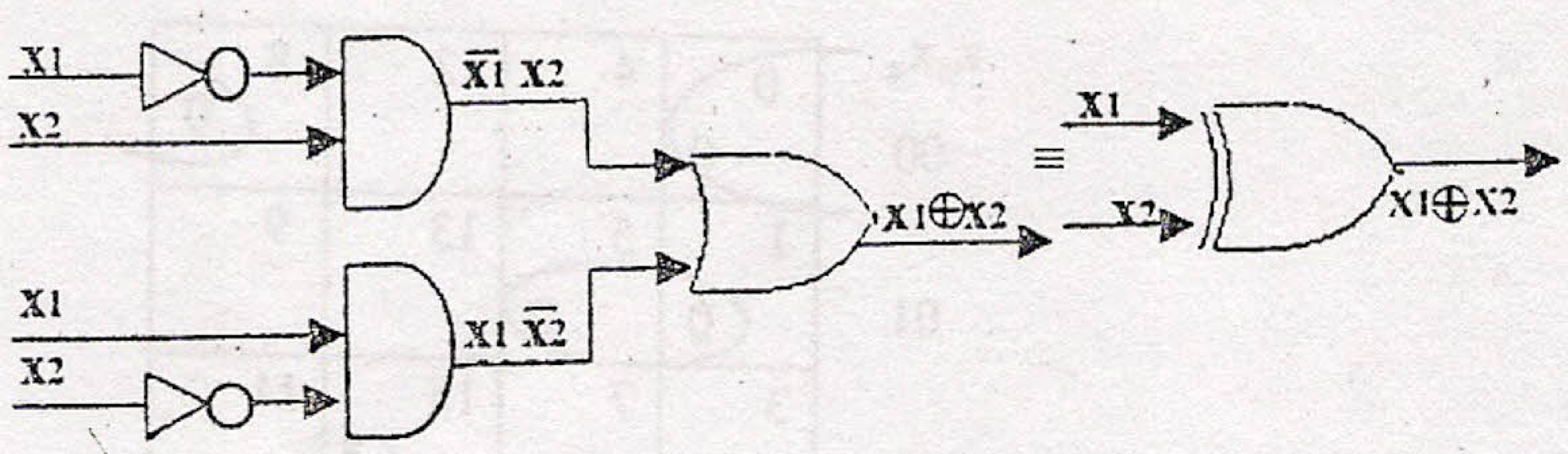
ترکیبی است از NOT و OR:

x_1	x_2	$\overline{x_1x_2}$
0	0	1
0	1	0
1	0	0
1	1	0



۳. Gate (X-OR):

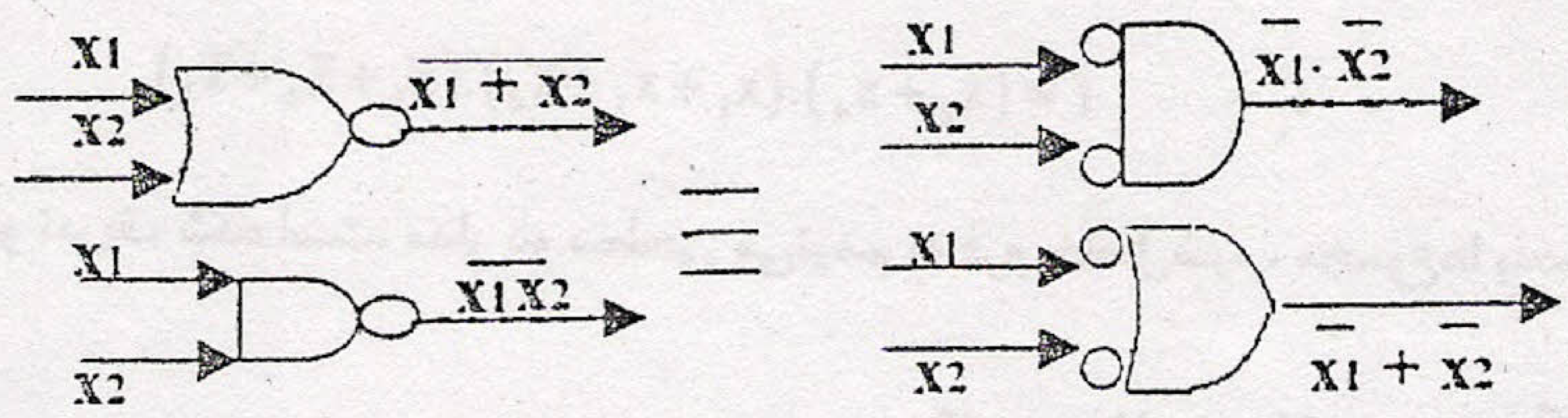
x_1	x_2	$x_1 \oplus x_2$
0	0	0
0	1	1
1	0	1
1	1	0



با استفاده از قوانین دمورگان داریم:

$$\overline{x_1 + x_2} = \bar{x}_1 . \bar{x}_2$$

$$\overline{x_1 . x_2} = \bar{x}_1 + \bar{x}_2$$

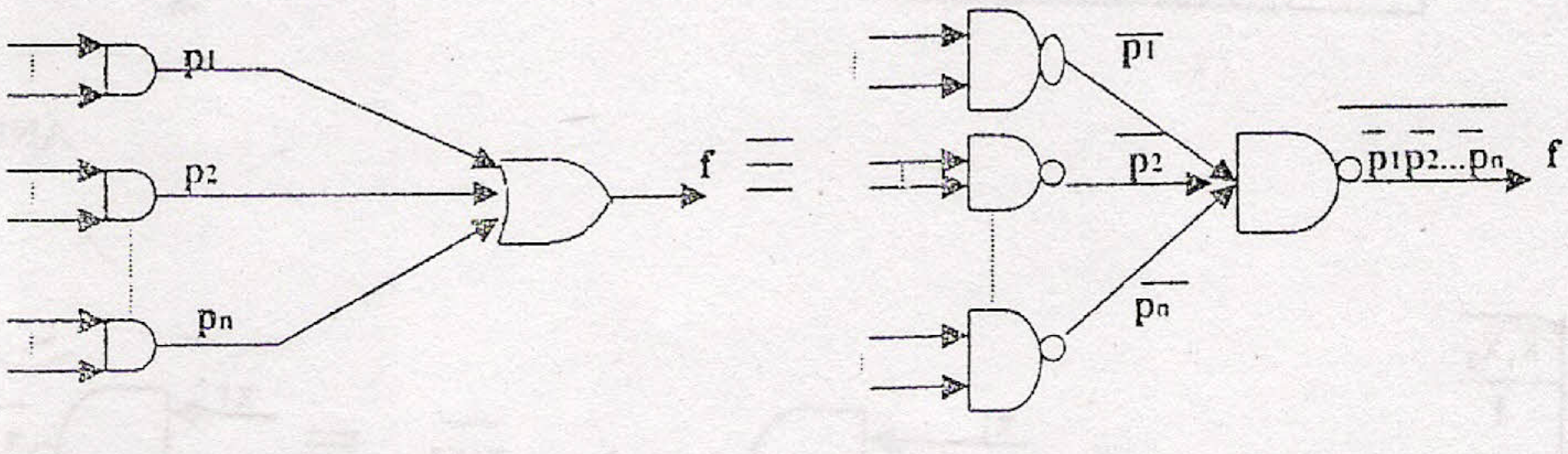


مدارهای دو سطحی تمام *NAND*

هر تابع بولی را می‌توان فقط با استفاده از NAND – Gate به صورت دو سطحی طرح نمود. زیرا هر تابع بولی را می‌توان بـا اسـتفاده از نقاط 1- و d.c تابع به فرم می‌نیمم مجموع حاصل‌ضرب‌ها تبدیل و مدار آن را به صورت دو سطحی AND – OR طرح نمود.

$$f = P_1 + P_3 + ... + P_n = \overline{\bar{P}_1 . \bar{P}_2 .... \bar{P}_n}$$

فرم می‌نیمم مجموع حاصل‌ضرب‌ها که در آن هر P_i یک PI می‌باشد.



مدار دوسطحی AND - OR   مدار دو سطحی تمام NAND

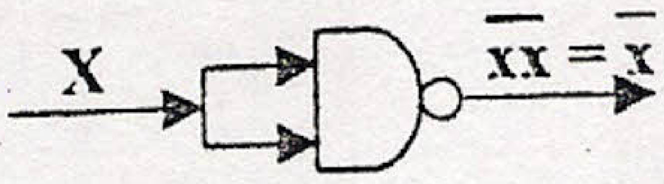
بنابراین مدار تمام NAND معادل مدار دو سطحی AND – OR می‌باشد.

یعنی برای طراحی مدار تمام NAND همان روش طراحی مدار دو سطحی AND – OR را در نظر می‌گیریم و فقط ورودی‌هـایی کـه از تعداد فردی از سطوح Gateها عبور می‌کند، در مدار تمام NAND به مکمل آن تبدیل می‌کنیم.

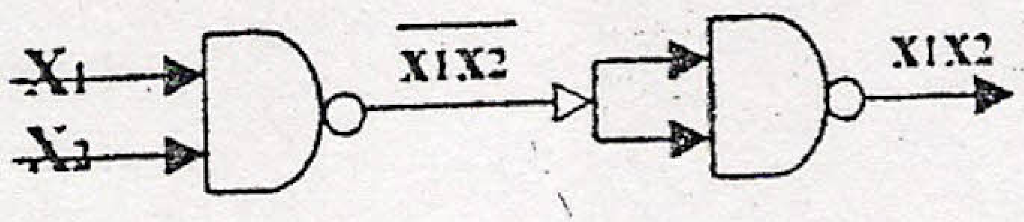
NAND – Gate به تنهایی از نظر عملیاتی کامل است یعنی تمام مدارهای ترکیبی را می‌توان فقط بـا اسـتفاده از NAND پیـاده‌سـازی کرد.

۱) NAND معادل NOT –Gate:

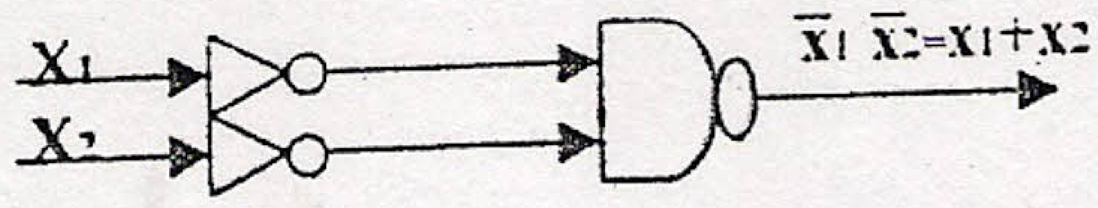
$\overline{xx} = \bar{x} + \bar{x} = \bar{x}$



۲) NAND معادل AND – gate:



۳) NAND معادل OR – gate:



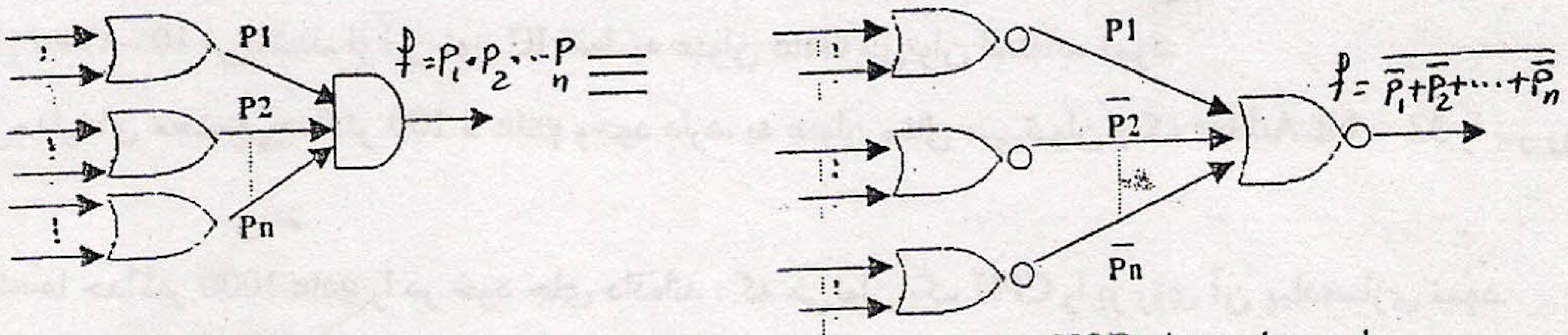
تمرین:

نشان دهید NOR - Gate به تنهایی از نظر عملیاتی کامل است.

مدارهای دو سطحی می‌نیمم با استفاده از *NOR*:

تابع بولی f را می‌توان فقط با استفاده NOR به صورت دو سطحی زیر طرح نمود، زیرا:

$$f = P_1 + P_2 + \ldots P_n \equiv \overline{\bar{P}_1 \bar{P}_2 \ldots \bar{P}_n}$$

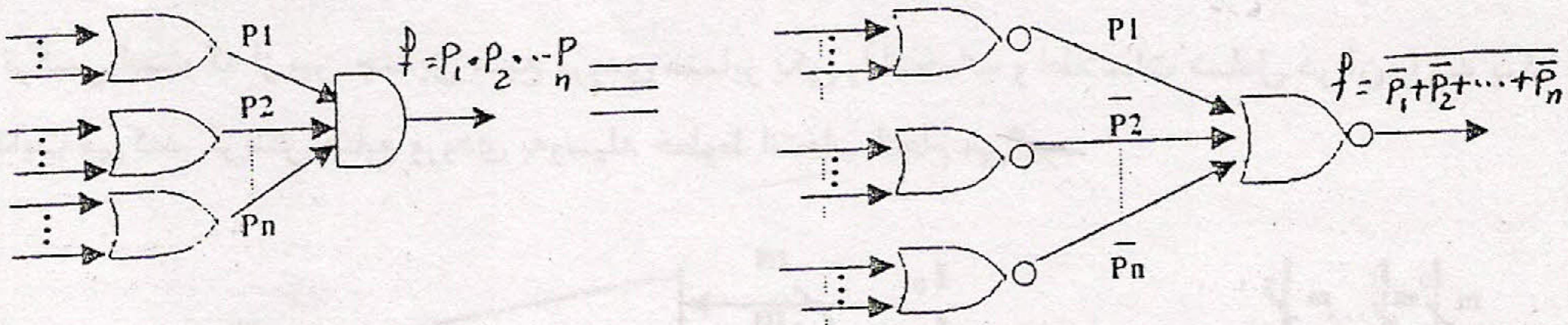


مدار دوسطحی OR - AND

مدار دوسطحی تمام NOR

مدار تمام NOR و دو سطحی تابع f را مستقیماً می‌توان از فرم می‌نیمم حاصل‌ضرب مجموع‌های تابع f به‌دست آورد، زیرا:

$$f = P_1 \cdot P_2 \ldots P_n = \overline{\bar{P}_1 + \bar{P}_2 + \ldots + \bar{P}_n}$$



مدار دو سطحی OR – AND

$f = P_1.P_2 \ldots\ldots p_n$

مدار تمام NOR

$f = \overline{\bar{P}_1 + \bar{P}_2 + \ldots + \bar{P}_n}$

بنابراین مدار تمام NOR معادل مدار دو سطحی OR – AND می‌باشد. طراحی مدار تمام NAND و تمام NOR، باعث پیدایش IC ها شده است.