فصل اول

جبر بول و ساده‌سازی توابع بولی

جبر بول:

یک شبکه مکمل‌پذیر و توزیع‌پذیر را جبر بول گویند و به صورت $\langle B , * , + , ' , 0 , 1 \rangle$ 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.

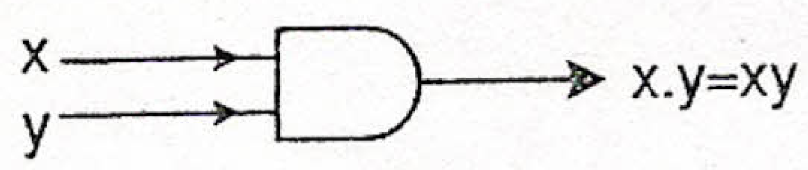
جبر بول دو مقداری:

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

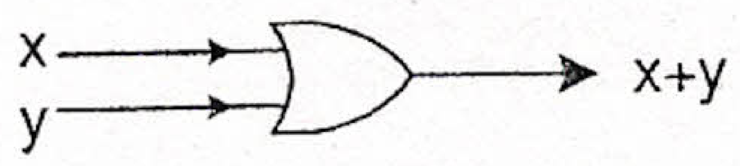
x	$\bar{x}$
0	1
1	0

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

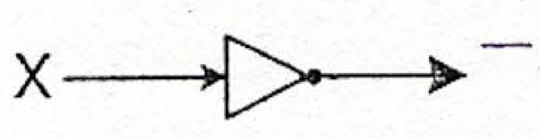
x	y	x.y
0	0	0
0	1	0
1	0	0
1	1	1



AND-GATE



OR-GATE



NOT-GATE

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:

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

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

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 $<B , * , + , ^{-} , 0 , 1>$ یک جبر بول و $B = \{0 , 1\}$:

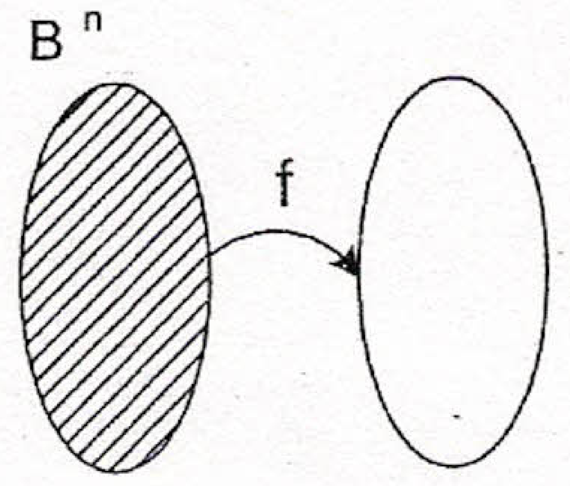
تابع $f : B^n \rightarrow B$ را که متناظر با یک عبارت بولی است " تابع بولی" گویند.

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.

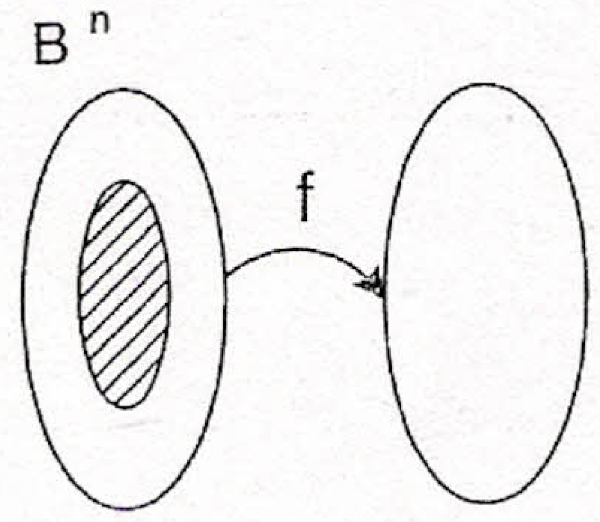
(بنابراین حداکثر $(2)^{2^n}$ تابع کاملاً مشخص شده قابل تعریف است)



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:

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

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

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:

$$g_1(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = f(x_1, x_2, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)$$

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

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.

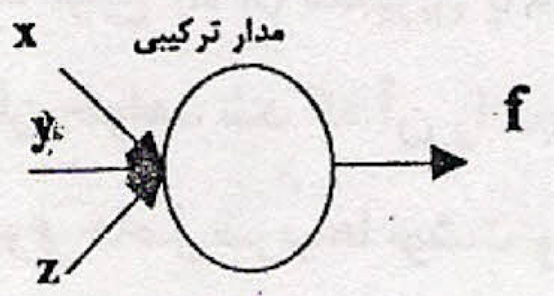
x	y	z
0	0	1
1	0	1

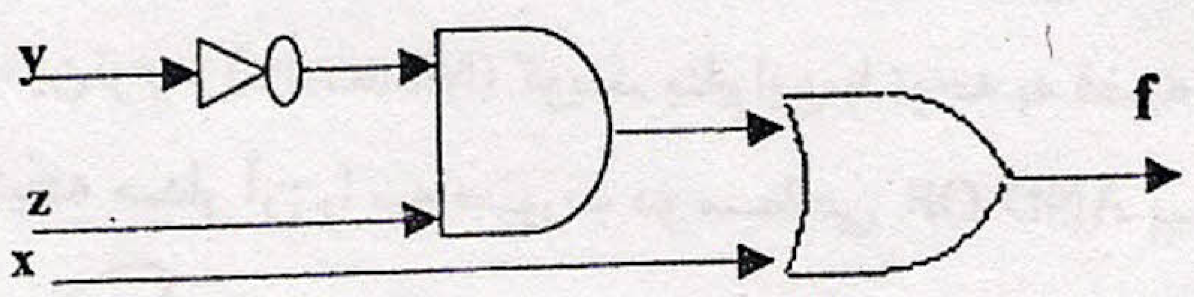
$\Rightarrow f = 1 \quad \text{if} \quad \overline{y}z = 1$

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.





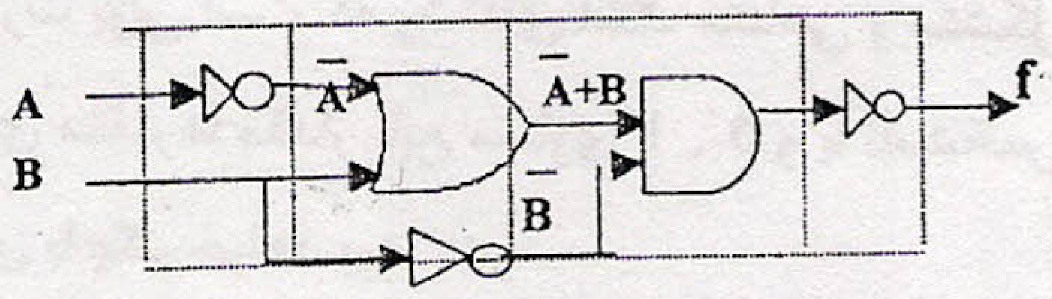
بنابراین از جبر بول دو مقداری می‌توان در تجزیه و تحلیل و طراحی مدارهای ترکیبی استفاده نمود و به کمک آن:

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.

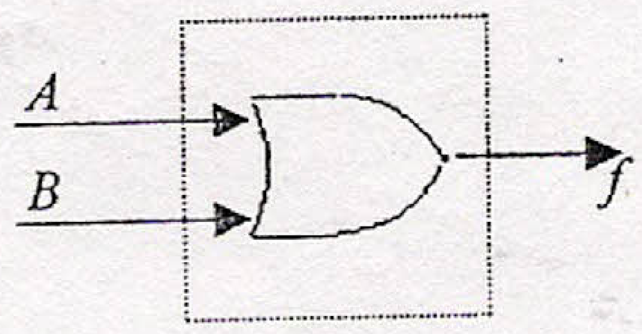
مثال: با فرض $f(A,B,C)=\overline{\left((A+B).\overline{B}\right)}$



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.

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



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.

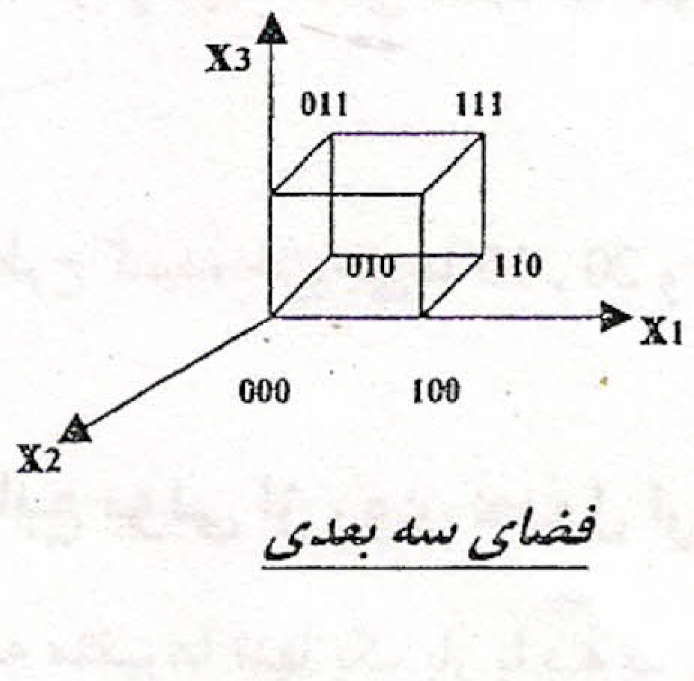
۱) ورودی‌ها را با متغیرهای دودویی مانند x_i نشان می‌دهیم.

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.

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

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

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.

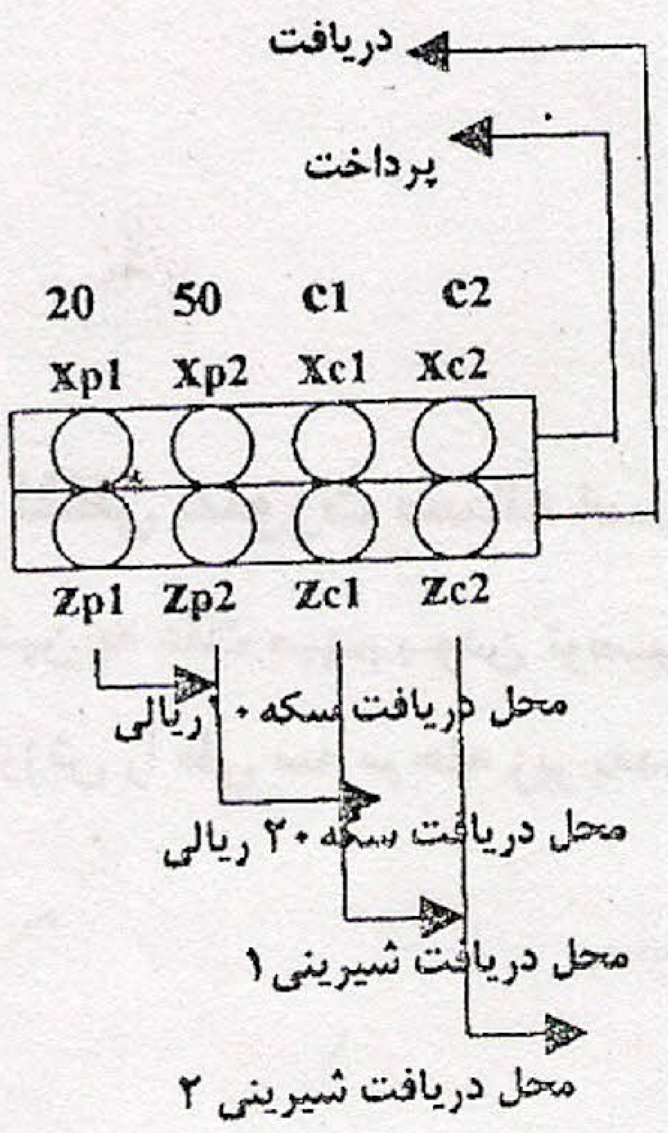
مثال ۳: ماشین خودکار برای توزیع شیرینی و باز پس دادن پول باقی‌مانده به شرح زیر طرح کنید:

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	$\overline{x}_1\overline{x}_2$	$\overline{x}_1 x_2$	$x_1\overline{x}_2$	$x_1 x_2$
m 0	0	0	1	0	0	0
m 1	0	1	0	1	0	0
m 2	1	0	0	0	1	0
m 3	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.

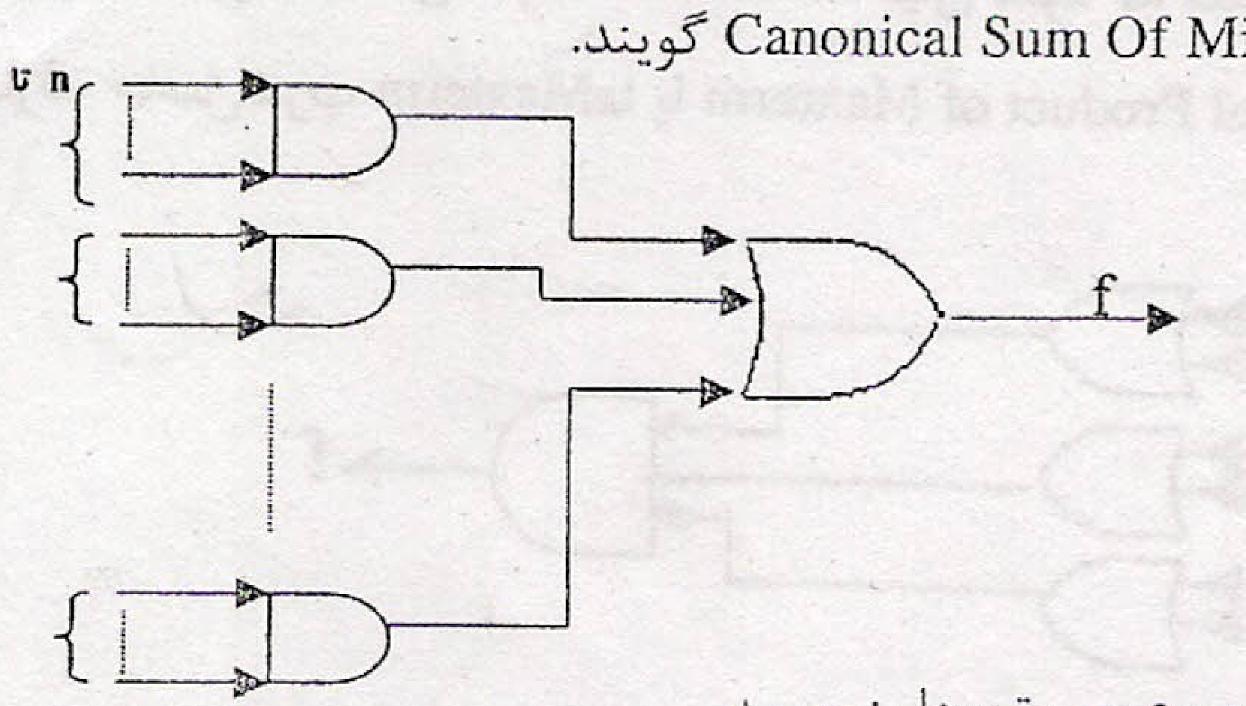
به طور کلی اگر n متغیر متمایز وجود داشته باشد، آن‌گاه 2^n مین‌ترم متمایز وجود خواهد داشت که آن‌ها را به ترتیب با $m_0, m_1, \ldots, m_{(2^n-1)}$ نشان می‌دهند. $x_1, x_2, x_3 \Rightarrow n=3 \Rightarrow 2^n = 2^3 = 8 \Rightarrow m_0, m_1, \ldots, m_7$

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.

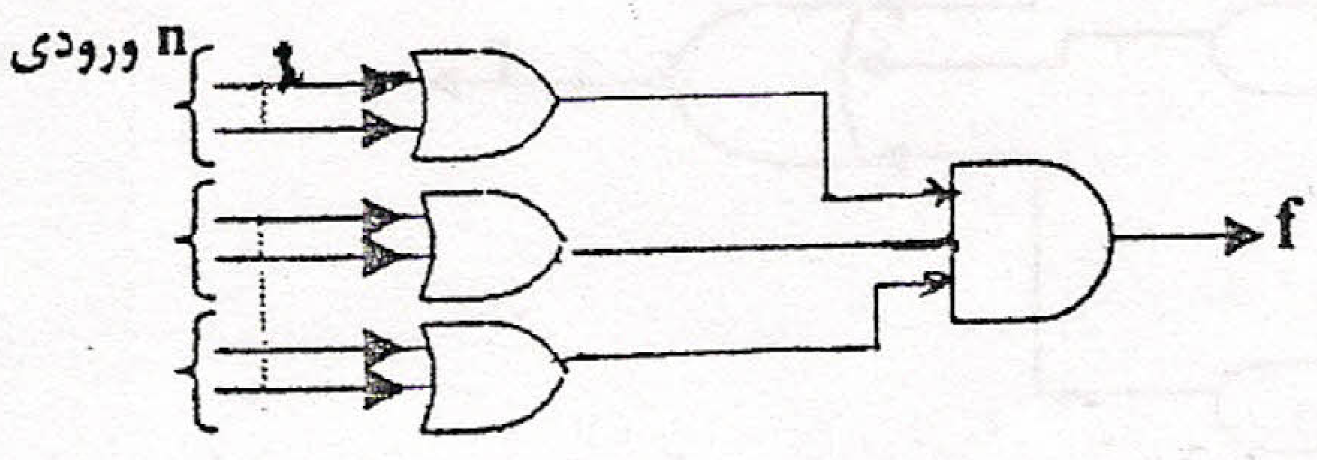
مثال: با دو متغیر x_1, x_2 چند جمله حاصل جمع کامل می‌توان نوشت:

	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 $M_0, M_1, \ldots\ldots, M_{(2^n-1)}$ 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 $f=\overline{x}_1\overline{x}_2x_3+x_1\overline{x}_2x_3+x_1x_2x_3$

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

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

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

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:

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

مثال: با فرض 3 متغیر x_1, x_2, x_3 جمله حاصل‌ضرب $\overline{x}_1x_3$ 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).

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

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
$$P = \overline{x}_2 \Rightarrow \begin{cases} x_1 & x_2 & x_3 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{cases}$$
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.

$$\Rightarrow P_2 = \overline{x}_1 x_2 \Rightarrow \begin{cases} 0 & 1 & 0 \\ 0 & 1 & 1 \end{cases} \Rightarrow \begin{cases} f(0\,1\,0) = 0 \\ f(0\,1\,1) = 1 \end{cases} \Rightarrow$$
P_2 یک ایجاب کننده نیست زیرا یک نقطه -0 تابع را می‌پوشاند.

$$\Rightarrow P_3 = x_1\overline{x}_2 \Rightarrow \begin{cases} 1 & 0 & 0 \\ 1 & 0 & 1 \end{cases} \Rightarrow \begin{cases} f(1\,0\,0) = - \\ f(1\,0\,1) = 1 \end{cases} \Rightarrow$$
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

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

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

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

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

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:

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

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

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

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:

(a) Binary Represention:

۱ـ در جدولی که ستون‌های آن با متغیرهای تابع یعنی $x_1, x_2, ..., x_n$ 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 $i = 1, 2, ..., n-1$ 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.