

منطق دودویی و جبر بول

فصل: ۲

منطق دودویی



منطق دودویی

- منطق دودویی با متغیرهایی که دوارزش جدا از هم دارد سروکار دارد
- منطق دودویی با عملیاتی که مفهوم منطقی دارد سروکار دارد
- دوازدهی که متغیرها در منطق دودویی می‌گیرند با اسامی مختلف نامگذاری می‌شوند مانند
 - درست و نادرست، بلی و خیر، high, low و ...
- برای مقصود ما مناسب است که آنها بصورت بیت در نظر گرفته شود و به آنها مقادیر 0 و 1 تخصیص دهیم
- منطق دودویی که بمنظور تشریح و دستکاری و پردازش اطلاعات دودویی بفرم ریاضی بکار می‌رود جبر بول نام دارد
- منطق دودویی شامل متغیرهای دودویی و عملیات منطقی است



منطق دودویی

- منطق دودویی شامل متغیرهای دودویی و عملیات منطقی است
- متغیرهای دودویی به وسیله حروف الفبا شامل $A, B, C, \dots$ مشخص می شوند
- هر متغیر فقط می تواند دو مقدار 0 یا 1 داشته باشد

□ عملیات منطقی در جبر بول

□ AND:

- نمایش به وسیله یک نقطه (.)
- عمل and به اینصورت تفسیر می شود که خروجی وقتی یک است اگر و فقط اگر هر دو ورودی یک باشد
- $Z = x.y$



منطق دودویی

OR □

□ نمایش به وسیله + (+)

□ عمل Or به اینصورت تفسیر می شود که خروجی وقتی صفر است اگر و فقط اگر هر دو ورودی صفر باشد

$$Z=x+y \quad \square$$

NOT □

□ نمایش به وسیله ' (-)

□ عمل not به اینصورت تفسیر می شود که خروجی برعکس ورودی باشد

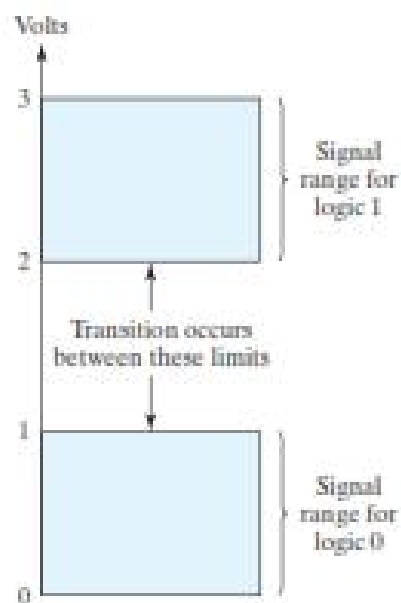
$$Z=x' \quad \square$$



مفهوم منطقی صفر و یک

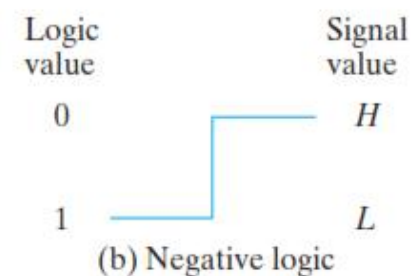
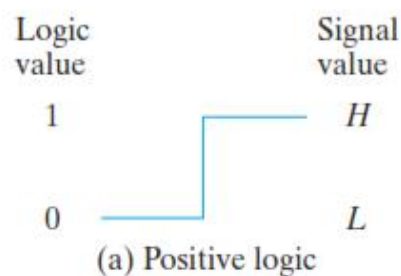
- در مدارهای دیجیتالی صفر و یک منطقی نشان دهنده یک سطح ولتاژ می باشد که متناسب با ولتاژ تعریف شده در تکنولوژی تعریف می شود
- در سطح سخت افزار هر عبارت منطقی با یک سیگنال الکتریکی نشان داده می شود.

مثال:



سطح ولتاژ بالا ← عبارت درست است.

سطح ولتاژ پائین ← عبارت نادرست است.



جبر بول

□ جبر بول مانند هر سیستم ریاضی بوسیله مجموعه ای از عناصر، مجموعه ای از عملگرها و تعدادی از اصول تعریف نشده تعریف می شود

➤ عملگرهای جبر بول: And , Or , Not

➤ تابع بولی: شامل متغیرهای دودویی (حروف الفبا) و عملگرهای اصلی و فرعی و پرانتز و علامت مساوی مانند: $f=xyz$

➤ اصول و قوانین جبر بول

✓ بسته بودن: اگر عملگری روی دو مجموعه جبر بول عمل کند برآیند حاصل در این مجموعه است



اصول و قوانین جبر بول

□ اصول و قوانین جبر بول

□ جابجایی: $x.y = y.x$ و $x+y = y+x$

□ شرکت پذیری: $(x+y)+z = x+(y+z)$ و $(x.y).z = x.(y.z)$

□ عضو خنثی $x+0 = 0+x = x$ و $x.1 = 1.x = x$

□ عضو معکوس $x + x' = 1$ و $x \cdot x' = 0$

□ توزیع پذیری $x + yz = (x + y)(x + z)$ و $x(y + z) = xy + xz$



اصول و قوانین جبر بول

□ اصول و قوانین جبر بول

□ تکرار پذیری

$$x \cdot x = x$$

$$x + x = x$$

□ دموورگان

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

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

$$x \cdot 0 = 0$$

$$x + 1 = 1$$

□

$$(x')' = x$$

□



اصول و قوانین جبر بول

□ اصول و قوانین جبر بول

$$x(x + y) = x \qquad x + xy = x \qquad \square$$

$$(a) \ 0 + 0 = 0 \qquad 0 + 1 = 1 + 0 = 1; \qquad \square$$

$$(b) \ 1 \cdot 1 = 1 \qquad 1 \cdot 0 = 0 \cdot 1 = 0. \qquad \square$$

□

$$(a) \ ab + a'c + bc = ab + a'c$$

$$(b) \ (a + b)(a' + c)(b + c) = (a + b)(a' + c)$$



قضیه گسترش شانون:

$$(a). f(x_1, x_2, \dots, x_n) = x_1 f(1, x_2, \dots, x_n) + (x_1)' f(0, x_2, \dots, x_n)$$

$$(b). f(x_1, x_2, \dots, x_n) = [x_1 + f(0, x_2, \dots, x_n)] [(x_1)' + f(1, x_2, \dots, x_n)]$$

:□ □□

- $f(A, B, C) = AB + AC' + A'C$

- $f(A, B, C) = AB + AC' + A'C = A f(1, B, C) + A' f(0, B, C)$

- $= A(1 \times B + 1 \times C' + 1' \times C) + A'(0 \times B + 0 \times C' + 0' \times C) = A(B + C') + A'C$

- $f(A, B, C) = A(B + C') + A'C = B[A(1 + C') + A'C] + B'[A(0 + C') + A'C]$

- $= B[A + A'C] + B'[AC' + A'C] = AB + A'BC + AB'C' + A'B'C$

- $f(A, B, C) = AB + A'BC + AB'C' + A'B'C$

- $= C[AB + A'B \times 1 + AB' \times 1' + A'B' \times 1] + C'[AB + A'B \times 0 + AB' \times 0' + A'B' \times 0]$

- $= ABC + A'BC + A'B'C + ABC' + AB'C'$

اصول و قوانین جبر بول

Table 2.1
Postulates and Theorems of Boolean Algebra

Postulate 2	(a)	$x + 0 = x$	(b)	$x \cdot 1 = x$
Postulate 5	(a)	$x + x' = 1$	(b)	$x \cdot x' = 0$
Theorem 1	(a)	$x + x = x$	(b)	$x \cdot x = x$
Theorem 2	(a)	$x + 1 = 1$	(b)	$x \cdot 0 = 0$
Theorem 3, involution		$(x')' = x$		
Postulate 3, commutative	(a)	$x + y = y + x$	(b)	$xy = yx$
Theorem 4, associative	(a)	$x + (y + z) = (x + y) + z$	(b)	$x(yz) = (xy)z$
Postulate 4, distributive	(a)	$x(y + z) = xy + xz$	(b)	$x + yz = (x + y)(x + z)$
Theorem 5, DeMorgan	(a)	$(x + y)' = x'y'$	(b)	$(xy)' = x' + y'$
Theorem 6, absorption	(a)	$x + xy = x$	(b)	$x(x + y) = x$



THEOREM 1(a): $x + x = x$.

Statement	Justification
$x + x = (x + x) \cdot 1$	postulate 2(b)
$= (x + x)(x + x')$	5(a)
$= x + xx'$	4(b)
$= x + 0$	5(b)
$= x$	2(a)

THEOREM 1(b): $x \cdot x = x$.

Statement	Justification
$x \cdot x = xx + 0$	postulate 2(a)
$= xx + xx'$	5(b)
$= x(x + x')$	4(a)
$= x \cdot 1$	5(a)
$= x$	2(b)

THEOREM 6(a): $x + xy = x$.

Statement	Justification
$x + xy = x \cdot 1 + xy$	postulate 2(b)
$= x(1 + y)$	4(a)
$= x(y + 1)$	3(a)
$= x \cdot 1$	2(a)
$= x$	2(b)



ساده سازی

منظور از ساده سازی اینست که با قوانین کاری کنیم که تعداد عملگرها یا کلیدهای منطقی تا حد امکان کم شود

مثال: ☐

$$ABC + AB'C = AC$$

مثال: ☐

$$\begin{aligned}(a(b+z(x+a'))))' &= a' + (b+z(x+a'))' \\ &= a' + b'(z(x+a'))' \\ &= a' + b'(z' + (x+a'))' \\ &= a' + b'(z' + x'a'') \\ &= a' + b'(z' + x'a)\end{aligned}$$

مثال: ☐

$$\begin{aligned}(W' + X' + Y' + Z')(W' + X' + Y' + Z)(W' + X' + Y + Z')(W' + X' + Y + Z) &= \\ (W' + X' + Y')(W' + X' + Y + Z')(W' + X' + Y + Z) &= (W' + X' + Y')(W' + X' + Y) = \\ (W' + X')\end{aligned}$$



ساده سازی

EXAMPLE 2.1

Simplify the following Boolean functions to a minimum number of literals.

1. $x(x' + y) = xx' + xy = 0 + xy = xy.$
2. $x + x'y = (x + x')(x + y) = 1(x + y) = x + y.$
3. $(x + y)(x + y') = x + xy + xy' + yy' = x(1 + y + y') = x.$
4.
$$\begin{aligned} xy + x'z + yz &= xy + x'z + yz(x + x') \\ &= xy + x'z + xyz + x'yz \\ &= xy(1 + z) + x'z(1 + y) \\ &= xy + x'z. \end{aligned}$$
5. $(x + y)(x' + z)(y + z) = (x + y)(x' + z),$ by duality from function 4.

$$F_1' = (x'yz' + x'y'z)' = (x'yz')'(x'y'z)' = (x + y' + z)(x + y + z')$$

$$\begin{aligned} F_2' &= [x(y'z' + yz)]' = x' + (y'z' + yz)' = x' + (y'z')'(yz)' \\ &= x' + (y + z)(y' + z') \\ &= x' + yz' + y'z \end{aligned}$$



ساده سازی

مثال: ☐

$$\begin{aligned}wy' + wx'y + wxyz + wxz' &= wy' + wx'y + wxy + wxz' \\&= wy' + wy + wxz' = w + wxz' = w\end{aligned}$$

مثال: ☐

$$(a + bc)' = (a + (bc))' = a'(bc)' = a'(b' + c') = a'b' + a'c'$$

مثال: ☐

$$\begin{aligned}(a(b + z(x + a')))' &= a' + (b + z(x + a'))' \\&= a' + b'(z(x + a'))' \\&= a' + b'(z' + (x + a'))' \\&= a' + b'(z' + x'(a'))' \\&= a' + b'(z' + x'a) \\&= a' + b'(z' + x')\end{aligned}$$

مثال: ☐

$$(a(b + c) + a'b)' = (ab + ac + a'b)' = (b + ac)' = b'(ac)' = b'(a' + c')$$



ساده سازی

$$AB + A'CD + BCD = AB + A'CD$$

مثال: 

$$(a + b')(a' + c)(b' + c) = (a + b')(a' + c)$$

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



جدول صحت

جدول صحت یا درستی :

برای هر ترکیبی از مقادیر X, Y مقدار معینی برای Z وجود دارد که این مقدار پس از تعریف عمل منطقی روی X, Y مشخص می شود این تعریف را می توان بصورت خلاصه و با استفاده از جدول صحت فهرست کرد

یک جدول صحت، جدولی است متشکل از تمام ترکیبات ممکن متغیر ها و بیانگر ارتباط بین مقادیر آنها و نتایج حاصل از عمل مربوطه روی آنها ست

توجه داشته باشید برای هر متغیر بولی می توان دو ارزش در نظر گرفت که عبارتند از صفر و یک و بنابراین برای n متغیر می توان ترکیبی از دو ارزش $0, 1$ در نظر گرفت که شامل جدولی است با 2^n سطر و n ستون.

برای مثال جدول $\text{and}, \text{or}, \text{not}$

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

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

x	x'
0	1
1	0



جدول صحت

Truth Tables for F_1 and F_2

x	y	z	F_1	F_2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	0
1	1	1	1	0

مثال: 

x	y	$x + y$	$(x + y)'$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

x'	y'	$x'y'$
1	1	1
1	0	0
0	1	0
0	0	0



جدول صحت

مثال: 

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

$y + z$	$x \cdot (y + z)$
0	0
1	0
1	0
1	0
1	0
0	0
1	1
1	1
1	1

$x \cdot y$	$x \cdot z$	$(x \cdot y) + (x \cdot z)$
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	1	1
1	0	1
1	1	1



گیت منطقی

گیت یا دروازه های منطقی المان های الکترونیکی هستند که می توانند متناسب با ورودی عمل خاصی را انجام دهند

And:

AND



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

OR:

OR



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



گیت منطقی

□ buffer:

Buffer



x	F
0	0
1	1

□ Not:

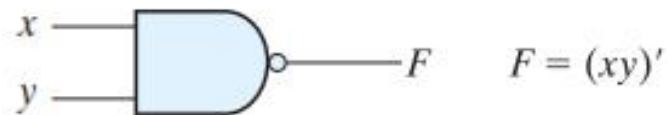
Inverter



x	F
0	1
1	0

□ Nand:

NAND



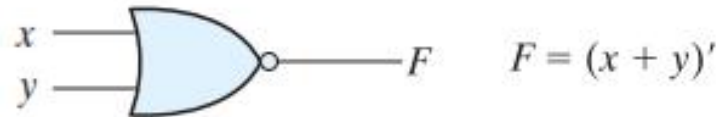
x	y	F
0	0	1
0	1	1
1	0	1
1	1	0



گیت منطقی

□ Nor:

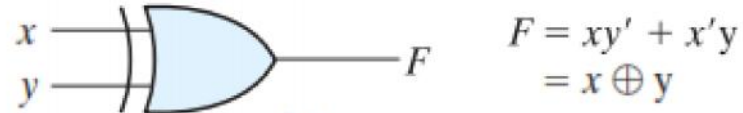
NOR



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

□ Xor:

Exclusive-OR
(XOR)

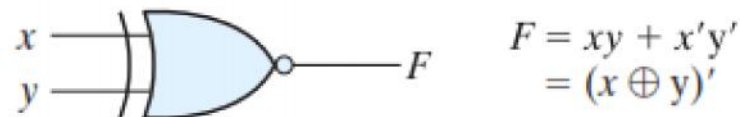


$$\begin{aligned} x \oplus 0 &= x \\ x \oplus 1 &= x' \\ x \oplus x &= 0 \\ x \oplus x' &= 1 \\ x \oplus y' &= x' \oplus y = (x \oplus y)' \end{aligned}$$

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

□ Xnor:

Exclusive-NOR
or
equivalence



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



گیت منطقی

x	y	z
L	L	L
L	H	L
H	L	L
H	H	H

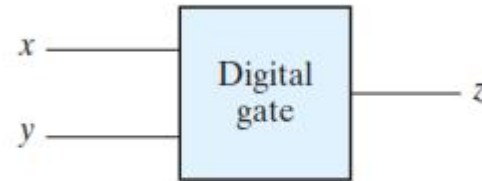
(a) Truth table with H and L

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

(c) Truth table for positive logic

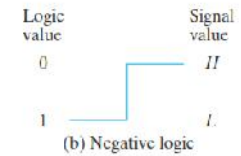
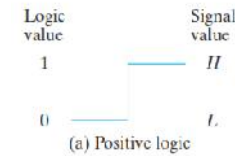
x	y	z
1	1	1
1	0	1
0	1	1
0	0	0

(e) Truth table for negative logic

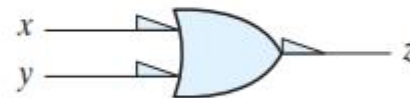


(b) Gate block diagram

مفهوم low , high active



(d) Positive logic AND gate

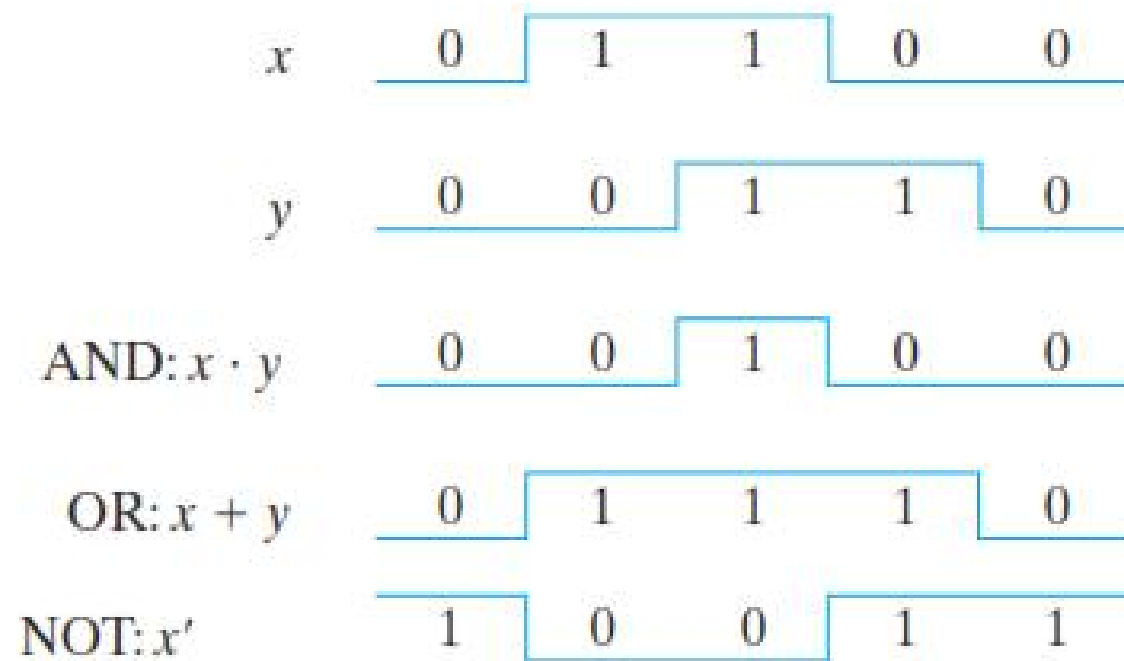


(f) Negative logic OR gate



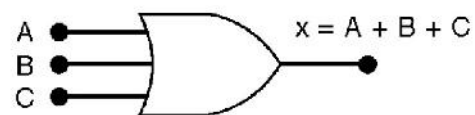
گیت منطقی

مفهوم 0,1 در عمل: 

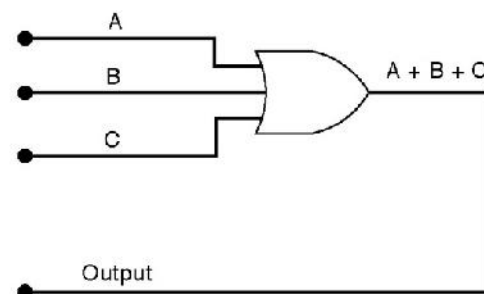
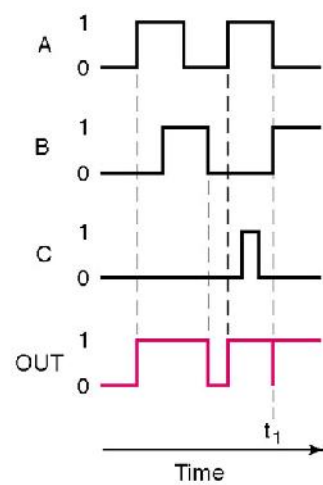


گیت منطقی

مفهوم 0,1 در عمل: 



A	B	C	$x = A + B + C$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



مکمل و دوگان

Not شده یک عبارت مکمل عبارت است ☐

$$F=(a+bc) \rightarrow F'=(a+bc)'=a' \cdot (bc)' = a' \cdot (b' + c')$$

مثال: ☐

$$F=(x'yz'+x'y'z) \rightarrow F'=(x+y+z)(x+y+z')$$

☐ ☐ ☐ ☐ :

☐ and $\rightarrow$ or

☐ or $\rightarrow$ and

مثال: ☐

$$F=x(y'z'+yz) \rightarrow F'=(x'+(y+z)(y'+z'))$$

$$\rightarrow F^*=(x'+(y'+z')(y+z))$$



روش های نمایش تابع بولی

□ هدف بدست آوردن معادله ای است که رابطه خروجی با متغیرهای ورودی را نشان می دهد و این کار برای پیدا کردن تابع بولی از روی جدول صحت می باشد

□ فرم استاندارد نمایش تابع بولی:

□ مجموع حاصلضرب یا مینترم: (SOP:Sum Of Product)

□ هدف از بدست آوردن فرم مجموع حاصلضرب ها بدست آوردن تابع بصورت جمله های حاصلضرب استاندارد است

□ یک جمله حاصلضرب استاندارد جمله and حاصل از تمام متغیرهای ورودی بصورت ساده یا مکمل شده است



روش های نمایش تابع بولی

□ حاصلضرب مجموع یا ماکسترم: (POS: Product Of Sum)

□ متغیر های یک جمله حاصل جمع استاندارد ، یک ردیف از جدول صحت است که مقدار تابع در آن ردیف 0 باشد را نشان می دهد

□ در جمله های حاصلضرب استاندارد متغیر های مکمل شده بیانگر ورودی 1 و متغیر های ساده بیانگر 0 است

□ به جمله های حاصلضرب استاندارد ماکسترم می گویند و با M نمایش می دهند

□ شکل کلی یک جمله حاصلضرب برای تابع سه متغیره f به وسیله ماکسترم بصورت زیر است

$$F(x,y,z)=(f_0+M_0) (f_1+M_1) (f_2+M_2) (f_3+M_3) (f_4+M_4) (f_5+M_5) (f_6+M_6) (f_7+M_7)$$



$$F(x,y,z)=\pi M(0,1,2,\dots,6,7)$$



روش های نمایش تابع بولی

□ فرم استاندارد نمایش تابع بولی:

□ حاصلضرب مجموع یا ماکسترم: (POS: Product Of Sum)

□ هدف از بدست آوردن فرم حاصلضرب مجموع بدست آوردن تابع بصورت جمله های حاصل جمع استاندارد است

□ یک جمله حاصل جمع استاندارد جمله or حاصل از تمام متغیر های ورودی بصورت ساده یا مکمل شده است



روش های نمایش تابع بولی

□ مجموع حاصلضرب یا مینترم : (SOP:Sum Of Product)

□ متغیر های یک جمله حاصلضرب استاندارد ، یک ردیف از جدول صحت است که مقدار تابع در آن ردیف 1 باشد را نشان می دهد

□ در جمله های حاصلضرب استاندارد متغیر های مکمل شده بیانگر ورودی 0 و متغیر های ساده بیانگر 1 است

□ به جمله های حاصلضرب استاندارد مینترم می گویند و با m نمایش می دهند

□ شکل کلی یک جمله حاصلضرب برای تابع سه متغیره f به وسیله مینترم بصورت زیر است

$$F(x,y,z)=f_0m_0+ f_1m_1+ f_2m_2+ f_3m_3+ f_4m_4+ f_5m_5+ f_6m_6+ f_7m_7$$

$$F(x,y,z)=\Sigma m(0,1,2,\dots,6,7)$$



روش های نمایش تابع بولی

Table 2.6

Truth Table for $F = xy + x'z$

x	y	z	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

Minterms

Maxterms



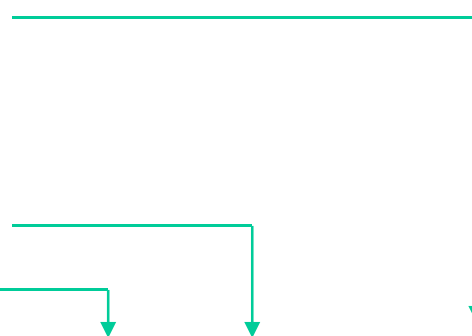
روش های نمایش تابع بولی

Table 2.3
Minterms and Maxterms for Three Binary Variables

			Minterms		Maxterms	
x	y	z	Term	Designation	Term	Designation
0	0	0	$x'y'z'$	m_0	$x + y + z$	M_0
0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
0	1	0	$x'yz'$	m_2	$x + y' + z$	M_2
0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
1	1	0	xyz'	m_6	$x' + y' + z$	M_6
1	1	1	xyz	m_7	$x' + y' + z'$	M_7



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


$$xyz + xyz' + x'yz$$

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

$$G = xyz + xyz' + x'yz$$



$$G = m_7 + m_6 + m_3 = \Sigma(3, 6, 7)$$

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

$$G = xyz + xyz' + x'yz$$



$$G = m_7 + m_6 + m_3 = \Sigma(3, 6, 7)$$



$$G = M_0M_1M_2M_4M_5 = \Pi(0,1,2,4,5)$$

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



روش های نمایش تابع بولی

$$f_1 = x'y'z + xy'z' + xyz = m_1 + m_4 + m_7$$

$$f_2 = x'yz + xy'z + xyz' + xyz = m_3 + m_5 + m_6 + m_7$$

$$\begin{aligned} f_1 &= (x + y + z)(x + y' + z)(x' + y + z')(x' + y' + z) \\ &= M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6 \end{aligned}$$

$$\begin{aligned} f_2 &= (x + y + z)(x + y + z')(x + y' + z)(x' + y + z) \\ &= M_0 M_1 M_2 M_4 \end{aligned}$$



ایجاد تابع بولی استاندارد

Express the Boolean function $F = A + B'C$ as a sum of minterms. The function has three variables: A , B , and C . The first term A is missing two variables; therefore,

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

This function is still missing one variable, so

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

The second term $B'C$ is missing one variable; hence,

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

Combining all terms, we have

$$\begin{aligned} F &= A + B'C \\ &= ABC + ABC' + AB'C + AB'C' + A'B'C \end{aligned}$$

But $AB'C$ appears twice, and according to theorem 1 ($x + x = x$), it is possible to remove one of those occurrences. Rearranging the minterms in ascending order, we finally obtain

$$\begin{aligned} F &= A'B'C + AB'C + AB'C' + ABC' + ABC \\ &= m_1 + m_4 + m_5 + m_6 + m_7 \end{aligned}$$

When a Boolean function is in its sum-of-minterms form, it is sometimes convenient to express the function in the following brief notation:

$$F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$$



ایجاد تابع بولی استاندارد

EXAMPLE 2.5

Express the Boolean function $F = xy + x'z$ as a product of maxterms. First, convert the function into OR terms by using the distributive law:

$$\begin{aligned} F &= xy + x'z = (xy + x')(xy + z) \\ &= (x + x')(y + x')(x + z)(y + z) \\ &= (x' + y)(x + z)(y + z) \end{aligned}$$

The function has three variables: x , y , and z . Each OR term is missing one variable; therefore,

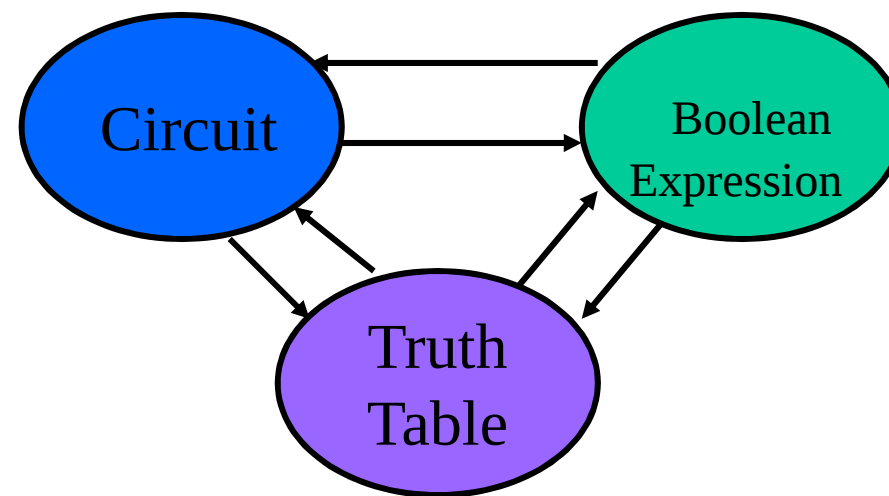
$$\begin{aligned} x' + y &= x' + y + zz' = (x' + y + z)(x' + y + z') \\ x + z &= x + z + yy' = (x + y + z)(x + y' + z) \\ y + z &= y + z + xx' = (x + y + z)(x' + y + z) \end{aligned}$$

Combining all the terms and removing those which appear more than once, we finally obtain

$$\begin{aligned} F &= (x + y + z)(x + y' + z)(x' + y + z)(x' + y + z') \\ &= M_0 M_2 M_4 M_5 \end{aligned}$$

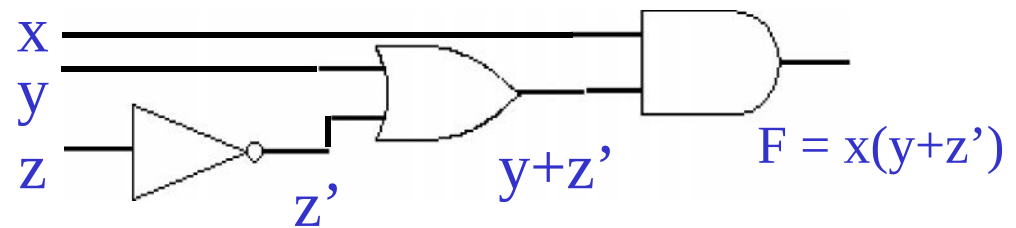


رابطه جبر بول و جدول صحت و مدار



ساخت مدار

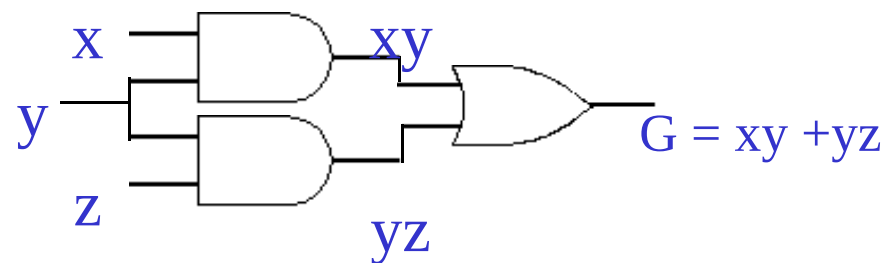
x	y	z	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1



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

ساخت مدار

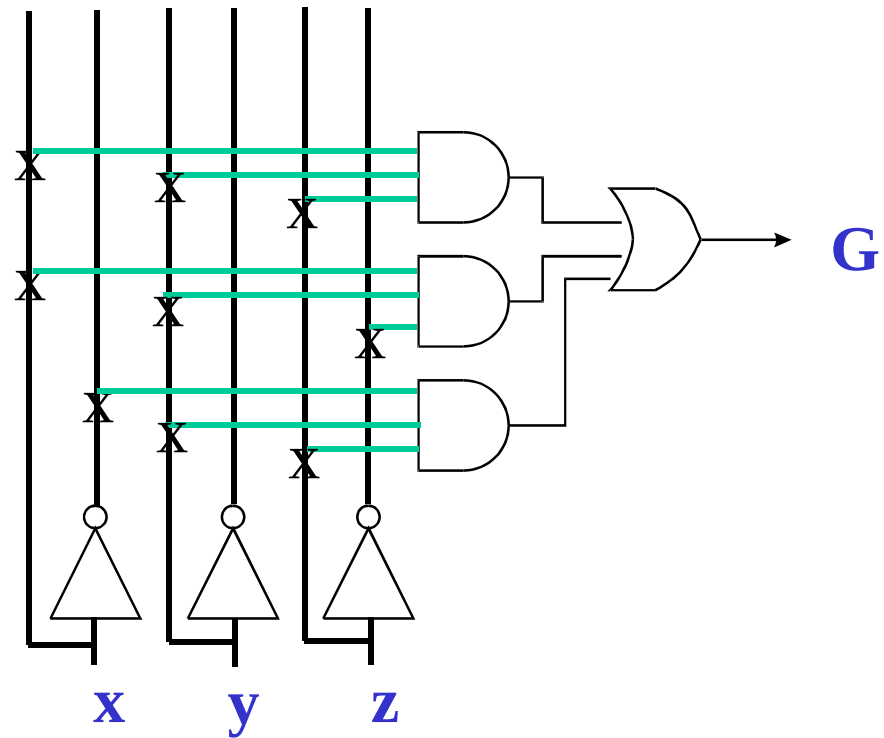
x	y	z	xy	yz	G
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	1	1
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	1	0	1
1	1	1	1	1	1



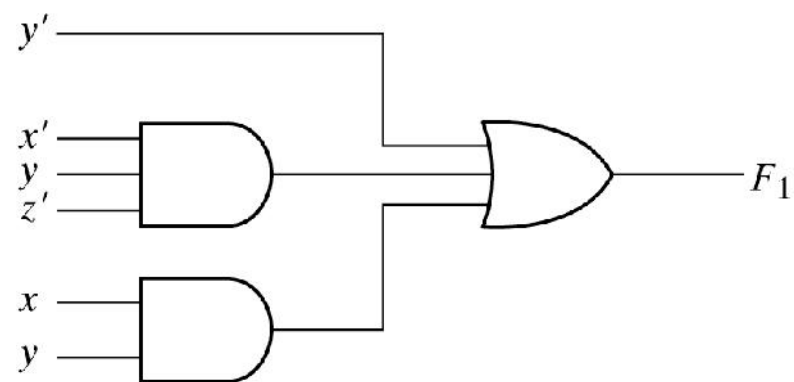
ساخت مدار

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

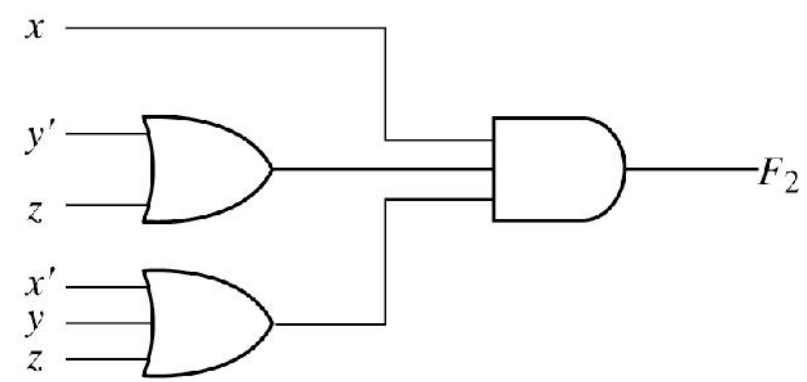
$$G = xyz + xyz' + x'yz$$



ساخت مدار



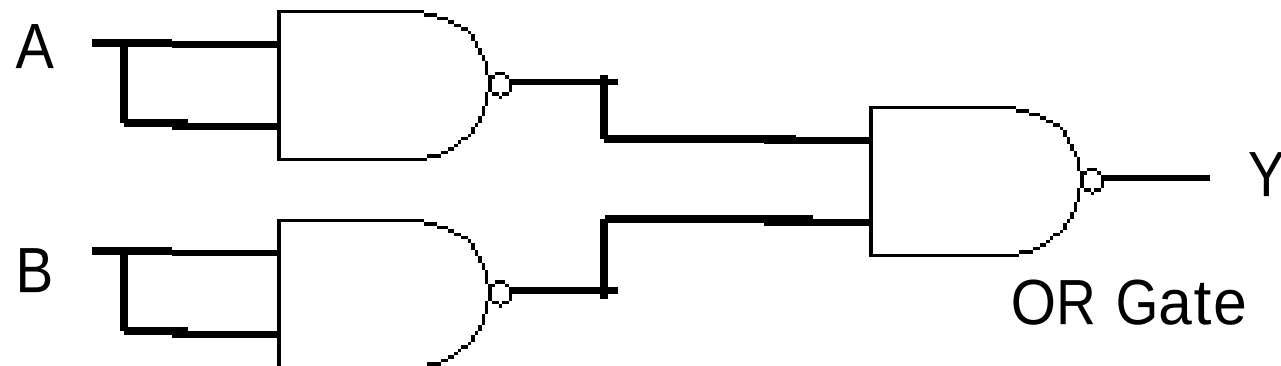
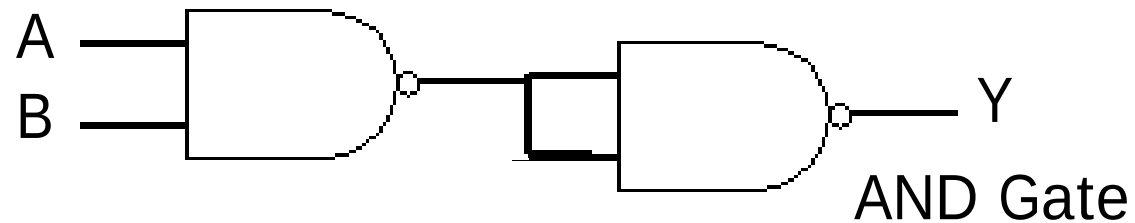
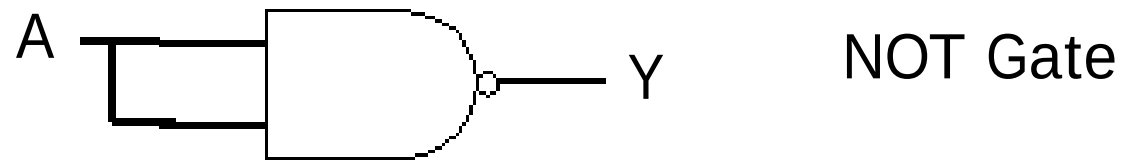
(a) Sum of Products



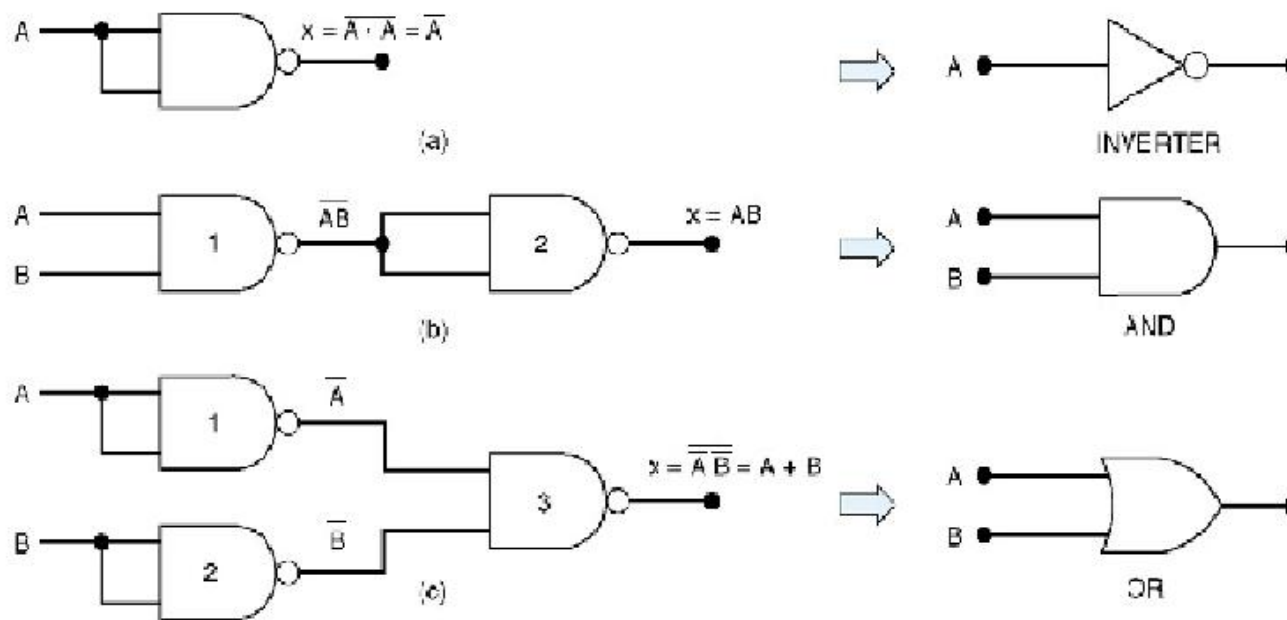
(b) Product of Sums

Fig. 2-3 Two-level implementation

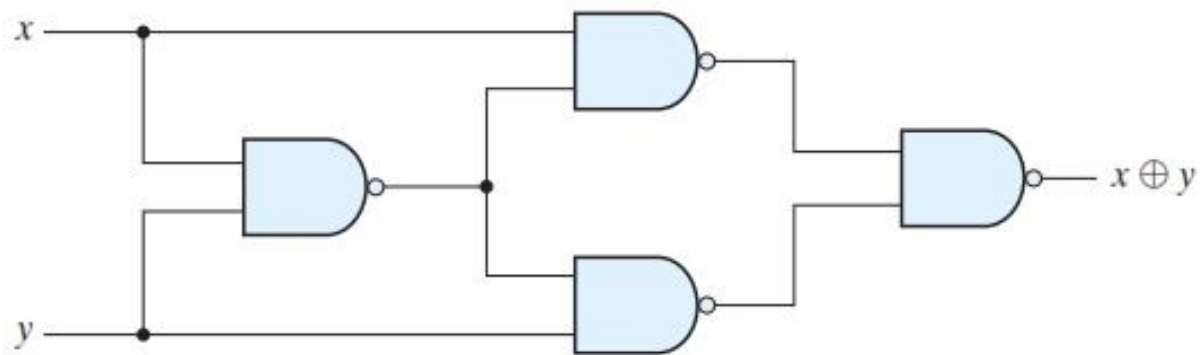
ساخت گیت های دیگر با گیت nand



ساخت گیت های دیگر با گیت nand



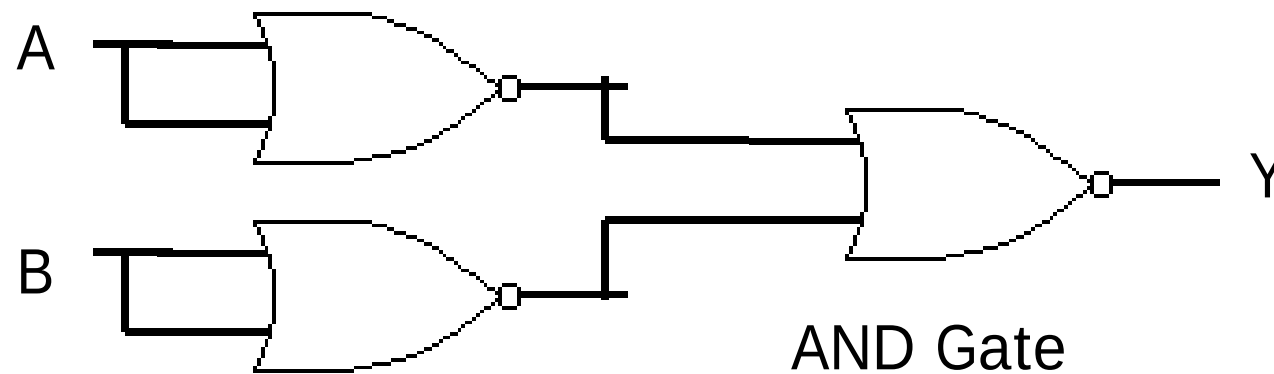
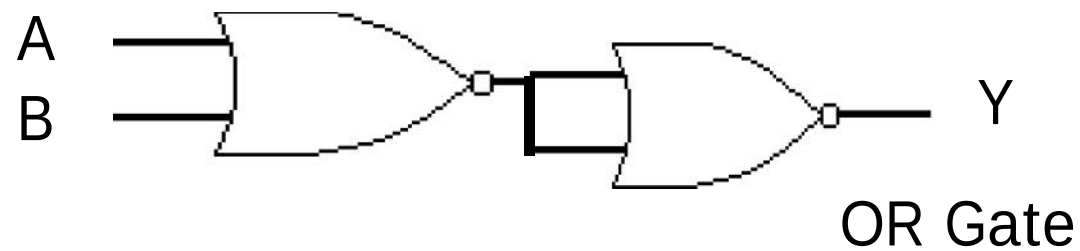
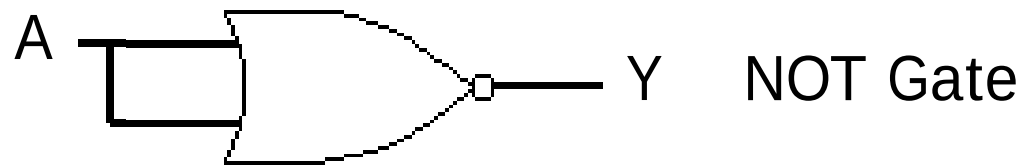
ساخت گیت های دیگر با گیت nand



(b) Exclusive-OR with NAND gates

xor

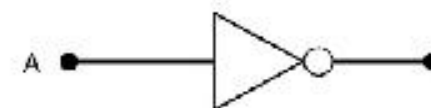
ساخت گیت های دیگر با گیت nor



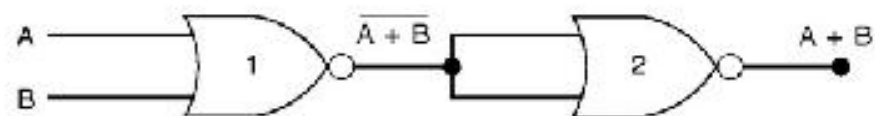
ساخت گیت های دیگر با گیت nor



(a)



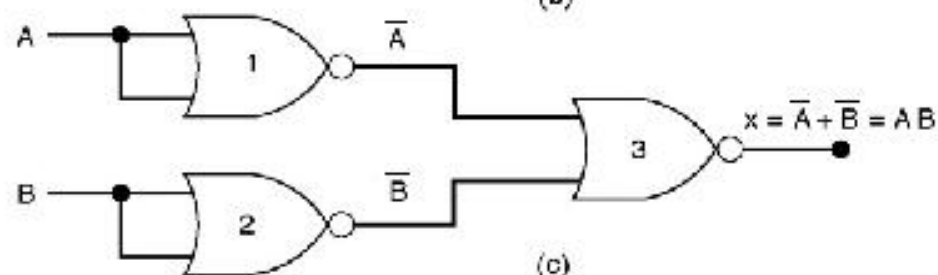
INVERTER



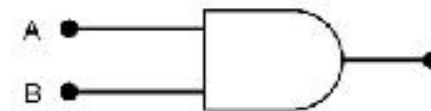
(b)



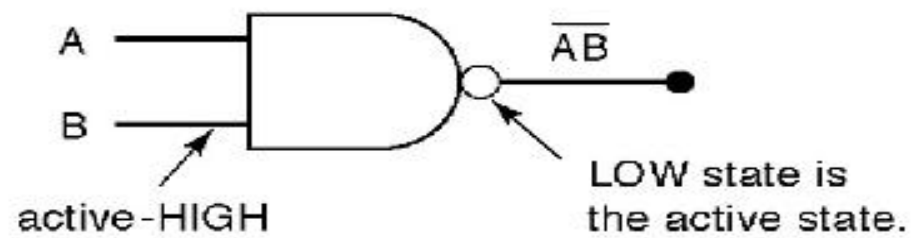
OR



(c)

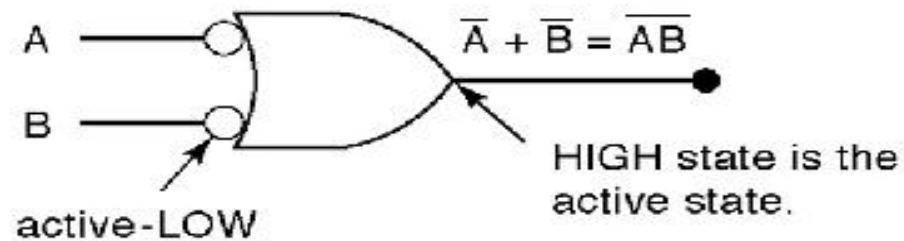


AND



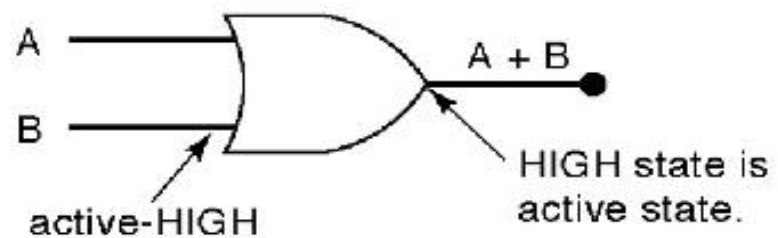
Output goes LOW only when all inputs are HIGH.

(a)



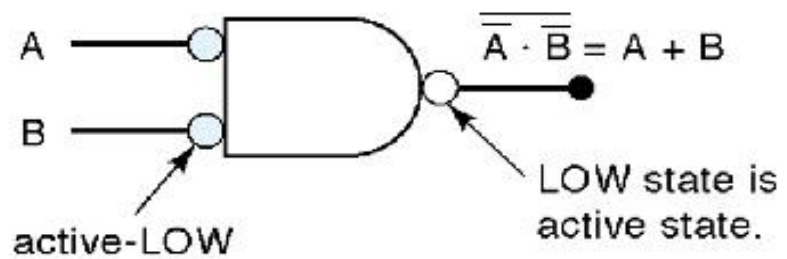
Output is HIGH when any input is LOW.

(b)



(a)

Output goes HIGH when any input is HIGH.



(b)

Output goes LOW only when all inputs are LOW.

تمرین

خروجی جدول صحت را برای تابع بالای جدول تکمیل کنید ☐

Table 2.5

Truth Table for $F = A + B'C$

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

تمرین

□ خروجی های جدول صحت زیر را بنویسید و ساده سازی کنید

Table 2.4
Functions of Three Variables

x	y	z	Function f_1	Function f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

تمرین

□ برای شکل های زیر خروجی را نوشته و ساده کنید

