فصل دوم

جبر بول و گیت‌های منطقی

اصول اولیه جبر بول

جبر بول به وسیله‌ی مجموعه‌ای از عناصر، مجموعه‌ای از عملگرها، و تعدادی از اصول اثبات نشده یا بدیهیات تعریف می‌شود.

عملگر دودویی: قانونی است که به هر جفت از عناصر یک مجموعه مانند S، یک عنصر منحصربفرد از S را تخصیص می‌دهد.

مهمترین اصول مورد استفاده در ساختارهای جبری (جبر معمول و متداول):

- بسته بودن (Closure):
- قانون شرکت پذیری (Associative law):

$$(x*y)*z = x*(y*z) \text{ for all } x, y, z, \in S$$

- قانون جابجایی (Commutative law):

$$x*y = y*x \text{ for all } x, y \in S$$

- عضو خنثی یا همانی (Identity element): مجموعه‌ی S نسبت به عملگر * دارای عضو همانیِ $e \in S$ است هر گاه:

$$e*x = x*e = x \text{ for every } x \in S$$

مثال: نسبت به مجموعه‌ی اعداد صحیح و عملگر جمع +، عدد صفر یک عضو خنثی است.

- عضو خنثی: برای هر عضو $x \in S$ یک عضو $y \in S$ وجود داشته باشد طوری که

$$x*y = e$$

در مثال قبلی، عضو معکوس متناظر با a به صورت a- است.

- قانون توزیع‌پذیری (Distributive law): برای دو عملگر دودویی تعریف شده‌ی * و . روی مجموعه‌ی S عملگر * را روی عملگر . توزیع‌پذیر می‌گوییم هرگاه

$$x*(y\cdot z) = (x*y)\cdot(x*z)$$

جبر بول توسط هانتینگتون در سال ۱۹۰۴ تدوین و پایه‌ریزی شد. این جبر یک نوع ساختار جبری است که روی یک مجموعه از عناصر دودویی، B، و به کمک دو عملگر + و . تعریف می‌شود. اصول هانتینگتون عبارتند از:

- بسته بودن نسبت به عملگرهای + و .

- عضو خنثی برای عملگر + (0) و عضو خنثی برای عملگر . (1)

$$x + 0 = 0 + x = x$$
$$x \cdot 1 = 1 \cdot x = x$$

- برقراری خاصیت جابجایی نسبت به دو عملگر + و .

$$x + y = y + x$$
$$x \cdot y = y \cdot x$$

- توزیع‌پذیری + نسبت به . و توزیع‌پذیری . نسبت به +

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

- برای هر عضو $x \in B$ یک عضو $x' \in B$ (به نام متمم x) وجود دارد طوری که

$$x + x' = 1$$
$$x \cdot x' = 0$$

- در مجموعه‌ی B حداقل دو عضو مختلف $x \neq y$ وجود دارد.

جبر بول (هانتینگتون) تفاوتهایی با جبر معمولی (حوزه یا میدان اعداد حقیقی) دارد:

۱- جبر بول فاقد اصل شرکت‌پذیری است اما این اصل برای جبر بول معتبر و برای هر دو عملگر برقرار است.

۲- توزیع پذیری عملگر + روی عملگر . برای جبر بول معتبر اما برای جبر معمولی قابل قبول نیست.

۳- جبر بول دارای معکوس‌های جمع (+) و ضرب (.) نیست؛ بنابراین، عملگرهای تقسیم و تفریق وجود ندارند.

۴- اصل مربوط به وجود عضو متمم، در جبر بول وجود ندارد.

در این کتاب ما فقط با جبر بول دوارزشی یا دومقداره سر و کار داریم. این جبر روی مجموعه‌ی $B = \{0, 1\}$ و روی دو عملگر + و . که با کمک جدول زیر تعریف شده‌اند، کار می‌کند:

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

ملاحظه می‌کنید که عملگرهای + و . و ' به ترتیب متناظر با گیت‌های منطقیِ OR، AND و NOT هستند. از این مطلب بعدا در پیاده‌سازی عبارات بولی به کمک گیت‌های منطقی استفاده خواهیم کرد.

اصل دوگانگی (Duality): اگر یک عبارت جبری برقرار و صحیح باشد، دوگان آن نیز برقرار و صحیح خواهد بود. برای به دست آوردن دوگان یک عبارت کافی است عملگر + را به عملگر . (و بالعکس) تبدیل کرده و مشابهاً عضو ۱ را با صفر (و بالعکس) تعویض کنیم.

برخی تئوری‌ها و اصول جبر بول: (اصول را بدیهی فرض کرده و می‌پذیریم اما تئوری‌ها را از روی اصول اثبات می‌کنیم)

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)	$x(x + y) = x$

روش اول اثبات برخی اتحادها یا تئوری‌ها

استفاده از اصول و روابط استخراج شده‌ی قبلی؛ **مثال:**

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)

روش دوم: استفاده از جدول صحت (Truth table)

مثال: اثبات تئوری جذب

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

مثال دوم: تئوری دمورگان

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

پیاده‌سازی توابع با گیت‌های منطقی و اهمیت استفاده از خواص و قوانین جبر بول در ساده‌سازی عبارات

مثال اول:

$$F_1 = x + y'z$$

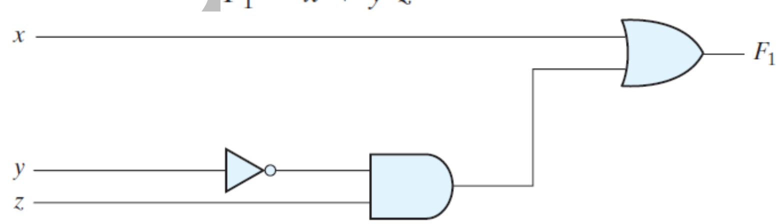


FIGURE 2.1
Gate implementation of $F_1 = x + y'z$

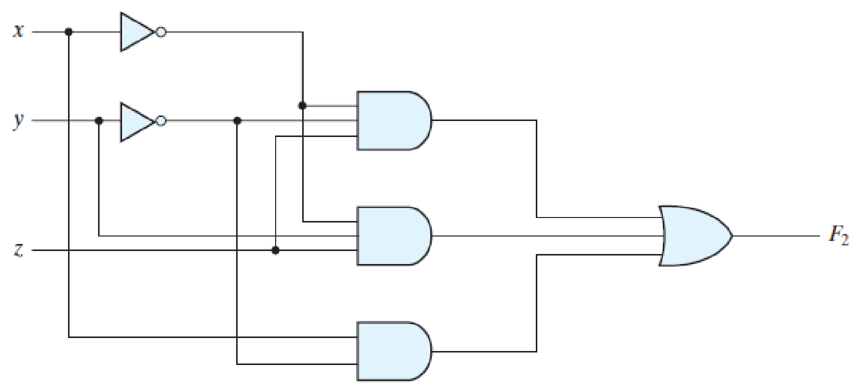
مثال دوم:

$$F_2 = x'y'z + x'yz + xy'$$

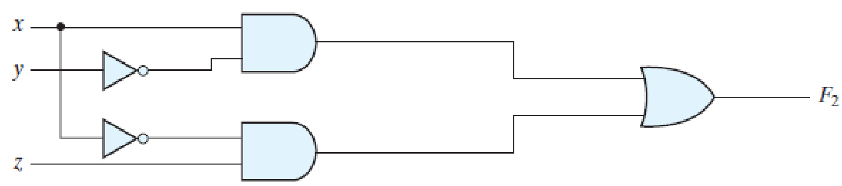
ساده‌سازی عبارت F_2:

$$F_2 = x'y'z + x'yz + xy' = x'z(y' + y) + xy' = x'z + xy'$$

مقایسه‌ی پیاده‌سازی با گیت‌های منطقی قبل و بعد از ساده‌سازی:



(a) $F_2 = x'y'z + x'yz + xy'$



(b) $F_2 = xy' + x'z$

FIGURE 2.2
Implementation of Boolean function F_2 with gates

در شکل (الف) در بالا، **می‌گوییم تابع F_2 دارای سه جمله و ۸ لیترال** است. مشابهاً در شکل (ب)، تابع F_2 دارای دو جمله و ۴ لیترال است.

مثالی از ساده‌سازی توابع بولی با کمترین تعداد لیترال‌ها:

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.

متمم کردن یک تابع:

روش اول به کمک تئوری دمورگان،

$$(A + B + C + D + \cdots + F)' = A'B'C'D' \dots F'$$
$$(ABCD \dots F)' = A' + B' + C' + D' + \cdots + F'$$

روی دوم، کافی است ابتدا جای عملگرهای + و . را عوض کرده (**یعنی «دوگانِ» تابع را به دست آوریم**) و سپس هر لیترال را متمم کنیم.

مثال: متمم کردن دو تابعِ $F_1 = x'yz' + x'y'z$ و $F_2 = x(y'z' + yz)$ به کمک استفاده‌ی مکرر از تئوری دمورگان:

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

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

مثال: متمم کردن توابع F_1 و F_2 مثال قبلی به کمک استفاده از روش دوگان کردن و متمم کردن:

1. $F_1 = x'yz' + x'y'z$.
The dual of F_1 is $(x' + y + z')(x' + y' + z)$.
Complement each literal: $(x + y' + z)(x + y + z') = F_1'$.

2. $F_2 = x(y'z' + yz)$.
The dual of F_2 is $x + (y' + z')(y + z)$.
Complement each literal: $x' + (y + z)(y' + z') = F_2'$.

تعریف «مینترم» یا «جمله‌ی ضرب استاندارد»:

به فرض وجود n متغیر بولی، 2^n ترکیب یا عبارت بولی مختلف میتوان نوشت که هر عبارت شامل فقط عملگر ضرب (.) بوده و از تمام متغیرهای بولی یک بار (و فقط یک بار) استفاده کرده باشد. چنین عبارتی را یک «مینترم» یا «جمله‌ی ضرب استاندارد» می‌گوییم.

تعریف «ماکسترم» یا «جمله‌ی جمع استاندارد»:

به فرض وجود n متغیر بولی، 2^n ترکیب یا عبارت بولی مختلف میتوان نوشت که هر عبارت شامل فقط عملگر جمع (+) بوده و از تمام متغیرهای بولی یک بار (و فقط یک بار) استفاده کرده باشد. چنین عبارتی را یک «ماکسترم» یا «جمله‌ی جمع استاندارد» می‌گوییم.

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

توصیف توابع بولی بر حسب مینترم‌ها یا ماکسترم‌ها و تبدیل آنها به یکدیگر:

توصیف بر حسب مینترم: یک تابع را می‌توان برحسب ردیف‌هایی از جدول صحت که تابع در این ردیف‌ها مقدار ۱ به خود می‌گیرد، توصیف کرد. در این توصیف از عملگر + (معادل با OR) استفاده می‌کنیم (بدلیل تعریف ذاتی این عملگر و مفهوم جدول صحت).

مثال:

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

بنابراین:

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

مشابهاً برای تابع دوم:

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

توصیف بر حسب ماکسترم:

روش معادل اول) یک تابع را می‌توان برحسب ردیف‌هایی از جدول صحت که تابع در این ردیف‌ها مقدار صفر به خود می‌گیرد، توصیف کرد. در این توصیف از عملگر . (معادل با AND) استفاده می‌کنیم (بدلیل تعریف ذاتی این عملگر و مفهوم جدول صحت).

مثال: همان تابع f_1 قبلی:

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

روش معادل دوم)

مشابه با حالت قبلی که مربوط به توصیف تابع برحسب مینترم‌ها بود، عمل می‌کنیم. در این جا این طور می‌گوییم که اگر تابعی در برخی سطرهای جدول صحت خود مقدار صفر به خود بگیرد، متمم آن تابع در این سطرها مقدار ۱ می‌گیرد. پس در این جا ابتدا متمم تابع را از روی سطرهایی از جدول صحت که تابع مقدار صفر به خود می‌گیرد، بر حسب مینترم‌ها توصیف می‌کنیم. سپس طرفین را متمم می‌گیریم تا به خود تابع برسیم. در این حالت با توجه به تئوری دمورگان، عبارتی که برای تابع به دست می‌آید برحسب ضرب ماکسترمها خواهد بود.

مثال:

در همان مثال قبلی،

$$f_1' = x'y'z' + x'yz' + x'yz + xy'z + xyz'$$

در نتیجه:

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

مشابهاً برای تابع دوم خواهیم داشت:

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

دو مثال مهم مربوط به توصیف رسمی توابع بولی بر حسب مجموع مینترم‌ها یا حاصل‌ضرب ماکسترم‌ها

EXAMPLE 2.4

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

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

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

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

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

$$F = A'B'C + AB'C + AB'C + ABC' + ABC$$
$$= m_1 + m_4 + m_5 + m_6 + m_7$$

و یا به صورت رسمی این طور می‌نویسیم (فرم «سیگما»):

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

توجه کنید که در نوشتن فوق، ترتیب نوشته شدن متغیرهای بولی (در این جا A و B و C) مهم است.

یک راه دیگر برای رسیدن به توصیف رسمی فوق، این است که ابتدا جدول صحت تابع را نوشته و سپس شماره‌ی مینترمهایی که تابع در آن مینترمها مقدار ۱ به خود می‌گیرد را محاسبه کنیم:

Table 2.5
Truth Table for $F = A + B'C$

A	B	C	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

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:

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

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

$$x' + y = x' + y + zz' = (x' + y + z)(x' + y + z')$$
$$x + z = x + z + yy' = (x + y + z)(x + y' + z)$$
$$y + z = y + z + xx' = (x + y + z)(x' + y + z)$$

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

$$F = (x + y + z)(x + y' + z)(x' + y + z)(x' + y + z')$$
$$= M_0M_2M_4M_5$$

و یا به صورت رسمی این طور می‌نویسیم (فرم «پای»):

$$F(x, y, z) = \Pi(0, 2, 4, 5)$$

با دقت در دو فرم «سیگما» و «پای» ملاحظه می‌شود که بین این دو نمایش رابطه وجود دارد. برای مثال:

$$F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$$
$$F'(A, B, C) = \Sigma(0, 2, 3) = m_0 + m_2 + m_3$$
$$F = (m_0 + m_2 + m_3)' = m_0' \cdot m_2' \cdot m_3' = M_0M_2M_3 = \Pi(0, 2, 3)$$

در حالت کلی بین هر جمله‌ی مینترم و جمله‌ی ماکسترم متناظر، رابطه‌ی زیر وجود دارد:

$$m_j' = M_j$$

فرم‌های استاندارد:

غیر از روش نمایش مینترم و ماکسترم، دو روش استاندارد نمایش توابع بولی به نامهای «جمع حاصل‌ضرب‌ها یا SOP» و «ضرب حاصل‌جمع‌ها یا POS» نیز وجود دارد که در این روشها دیگر محدودیت موجود در مینترم‌ها و ماکسترم‌ها (یعنی این محدودیت که هر جمله باید شامل تمام متغیرها باشد) وجود ندارد.

مثالی از فرم SOP:

$$F_1 = y' + xy + x'yz'$$

مثالی از فرم POS:

$$F_2 = x(y' + z)(x' + y + z')$$

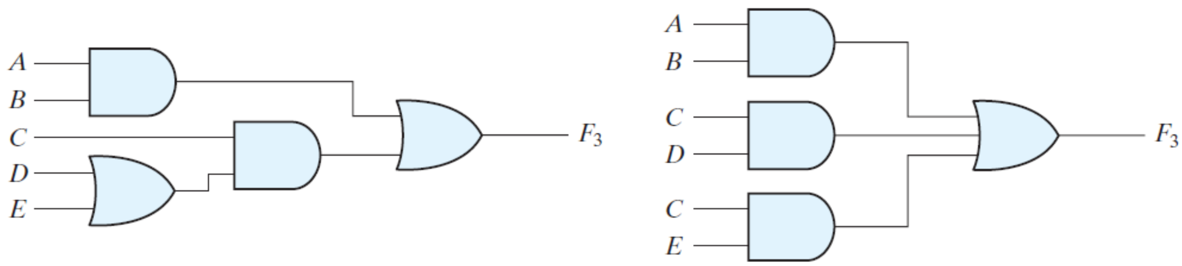
پیاده‌سازی فرم‌های SOP و POS همیشه «دو سطحی» بوده و نسبت به سایر فرم‌های غیراستاندارد ترجیح داده می‌شوند زیرا به هنگام انتشار ورودی‌ها به سمت خروجی‌ها، کمترین مقدار تاخیر را در گیت‌ها تولید می‌کنند. همچنین، در این فرم‌های استاندارد، پیش‌بینیِ تغییرات زمانی سیگنالها دقیق‌تر قابل انجام است.

مثالی از فرم غیراستاندارد:

$$F_3 = AB + C(D + E)$$

پیاده‌سازی تابع فوق، «سه سطحی» است. اگر عبارت فوق را کمی دستکاری کنیم به پیاده‌سازی «دو سطحی» می‌رسیم:

$$F_3 = AB + C(D + E) = AB + CD + CE$$



(a) $AB + C(D + E)$ (b) $AB + CD + CE$

FIGURE 2.4
Three- and two-level implementation

دیگر اعمال منطقی:

Table 2.7
Truth Tables for the 16 Functions of Two Binary Variables

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

Table 2.8
Boolean Expressions for the 16 Functions of Two Variables

Boolean Functions	Operator Symbol	Name	Comments
$F_0 = 0$		Null	Binary constant 0
$F_1 = xy$	$x \cdot y$	AND	x and y
$F_2 = xy'$	x/y	Inhibition	x, but not y
$F_3 = x$		Transfer	x
$F_4 = x'y$	y/x	Inhibition	y, but not x
$F_5 = y$		Transfer	y
$F_6 = xy' + x'y$	$x \oplus y$	Exclusive-OR	x or y, but not both
$F_7 = x + y$	$x + y$	OR	x or y
$F_8 = (x + y)'$	$x \downarrow y$	NOR	Not-OR
$F_9 = xy + x'y'$	$(x \oplus y)'$	Equivalence	x equals y
$F_{10} = y'$	y'	Complement	Not y
$F_{11} = x + y'$	$x \subset y$	Implication	If y, then x
$F_{12} = x'$	x'	Complement	Not x
$F_{13} = x' + y$	$x \supset y$	Implication	If x, then y
$F_{14} = (xy)'$	$x \uparrow y$	NAND	Not-AND
$F_{15} = 1$		Identity	Binary constant 1

سمبلهای گرافیکی برخی از گیتهای منطقی

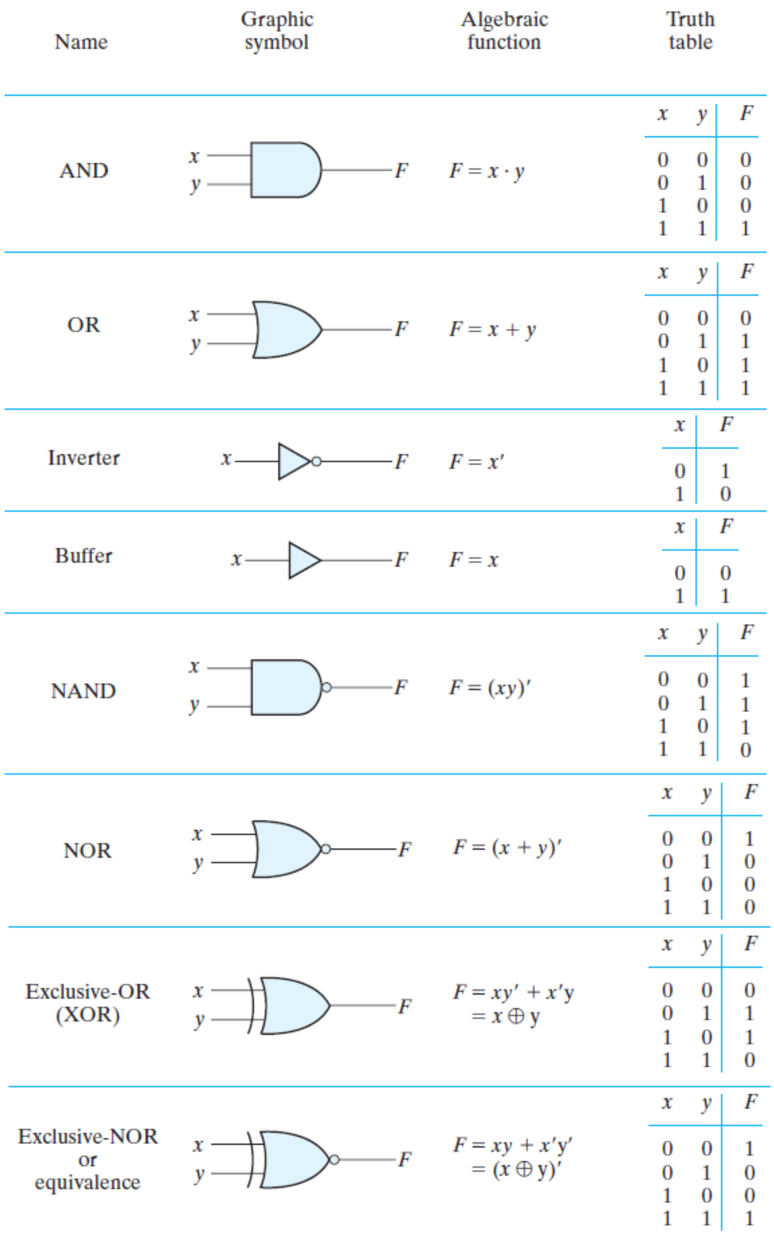


FIGURE 2.5
Digital logic gates

خانواده‌های مختلف منطق دیجیتال

یکی از معیارهای دسته‌بندی مدارات مجتمع دیجیتال (غیر از مواردی چون میزان پیچیدگی یا نوع عملیات منطقی)، تکنولوژی مداری خاص به کار رفته در این مدارات است. در هر خانواده‌ی منطقی، مدارات الکترونیکی پایه‌ای خاصی وجود دارند که از روی آنها، مدارات و عناصر دیجیتالی پیچیده‌تر ساخته می‌شوند. این مدارات پایه‌ای در هر تکنولوژی خاص (معمولاً) شامل NAND، NOR و NOT است. معمولاً از عناصر الکترونیکی استفاده شده برای ساخت (این

مدارات پایه در نامگذاری تکنولوژی استفاده می‌شود. برخی از متداولترین و معروف‌ترین خانواده‌های منطقی از مدارات مجتمع دیجیتال عبارتند از:

TTL	transistor–transistor logic;
ECL	emitter-coupled logic;
MOS	metal-oxide semiconductor;
CMOS	complementary metal-oxide semiconductor.

TTL is a logic family that has been in use for 50 years and is considered to be standard. ECL has an advantage in systems requiring high-speed operation. MOS is suitable for circuits that need high component density, and CMOS is preferable in systems requiring low power consumption, such as digital cameras, personal media players, and other handheld portable devices. Low power consumption is essential for VLSI design; therefore, CMOS has become the dominant logic family, while TTL and ECL continue to decline in use.

برخی از مهمترین پارامترهای متمایزکننده‌ی خانواده‌های منطقی مختلف عبارتند از:

گنجایش (یا ظرفیت) خروجی (Fan out): حداکثر تعداد بارهای استانداردی که خروجی یک گیت می‌تواند درایو کند بدون این که عملکرد معمول آن را دچار خدشه و آسیب کند. منظور از یک بار استاندارد، جریان مورد نیاز ورودیِ یک گیت مشابه با گیت مورد نظر ما است.

گنجایش (یا ظرفیت) ورودی (Fan in): تعداد ورودی‌های موجود در یک گیت (!).

توان مصرفی (Power dissipation): توان مصرفی شده توسط گیت که باید توسط منبع تغذیه فراهم شود.

تاخیر انتشار (Propagation delay): متوسط زمان تاخیر در انتقال یک سیگنال از ورودی به خروجی گیت

حاشیه‌ی نویز (یا حد پارازیت، Noise margin): بیشترین دامنه‌ی ولتاژ نویز خارجی که می‌تواند به سیگنال ورودی (به گیت منطقی) اضافه شود بدون این که تغییر نامطلوب یا ناخواسته در خروجی گیت ایجاد کند.

براي سلامتي رهبر انقلاب و تعجيل در ظهور حضرت ولي عصر (عج)

صلوات