Digital

C O M P L E M E N T S O F N U M B E R S

- Two’s Complement 2 متمم
- ***Signed Binary Numbers***

Table 1.3
Signed Binary Numbers

Decimal	Signed-2’s Complement	Signed-1’s Complement	Signed Magnitude
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
−0	—	1111	1000
−1	1111	1110	1001
−2	1110	1101	1010
−3	1101	1100	1011
−4	1100	1011	1100
−5	1011	1010	1101
−6	1010	1001	1110
−7	1001	1000	1111
−8	1000	—	—

COMPLEMENTS OF NUMBERS

اعداد علامت دار در مبنای ۲

فرض کنید عددی را در مبنای ۲ با n بیت نشان می‌دهیم (مثلاً ۸ بیت) پر ارزش‌ترین بیت Msb (Most Significant bit) بیت علامت و بقیه بیت‌ها مقدار عدد است (۷ بیت دیگر به مقدار عدد اختصاص دارد)

دامنه و رنج عدد: اگر عدد بدون علامت باشد با ۸ بیت رنج عدد از $0 \ldots (2^8-1)$

$0 \ldots 255$

و اگر عدد علامت دار باشد رنج عدد از $(2^7-1) \ldots 0 \ldots (-2^7)$ خواهد بود

$(-128) \ldots 0 \ldots (127)$

مثلاً با طول عدد 4 بیت بدون علامت از $0 \ldots (2^4-1)$

$0 \ldots 15$

و اگر عدد علامت دار باشد $(-2^{4-1}) \ldots 0 \ldots (2^{4-1}-1)$

$-8 \ldots 0 \ldots 7$

بیت n ام اگر صفر باشد عدد را مثبت درنظر می‌گیریم و اگر بیت n ام 1 باشد عدد را منفی درنظر می‌گیریم

COMPLEMENTS OF NUMBERS

بدیهی است که در مثال 4 بیت اگر بخواهیم عدد مثلاً $+12$ را در فرمت اعداد علامت دار

بنویسیم امکان پذیر نیست چون همانطور که گفتیم رنج از $(+7 \ldots 0 \ldots -8)$ است

و باید جایی که برای ذخیره عدد در کامپیوتر در نظر می گیریم بیشتر از 4 بیت باشد مثلاً 5 بیت،

8 بیت، 16 بیت یا 32 بیت و یا بیشتر

برای اینکه با جمع اعداد علامت دار آشنا شویم. ۲ حالت زیر را در نظر می گیریم

الف - حالتی که مجموع دو عدد در قالب n بیت جای شود (امکان پذیر است)

ب - حالتی که مجموع دو عدد (حاصلجمع) در قالب n بیت جا نمی شود (امکان پذیر نباشد)

در حالت ب اصطلاحاً می گویند خطای سرریز Overflow رخ داده است

مثال حالت الف در اسلاید بعد نشان داده شده است

COMPLEMENTS OF NUMBERS

- Example:

$$\begin{array}{rr} +\ 6 & 00000110 \\ +13 & 00001101 \\ \hline +19 & 00010011 \end{array} \qquad \begin{array}{rr} -\ 6 & 11111010 \\ +13 & 00001101 \\ \hline +\ 7 & 00000111 \end{array}$$

$$\begin{array}{rr} +\ 6 & 00000110 \\ -13 & 11110011 \\ \hline -\ 7 & 11111001 \end{array} \qquad \begin{array}{rr} -\ 6 & 11111010 \\ -13 & 11110011 \\ \hline -19 & 11101101 \end{array}$$

COMPLEMENTS OF NUMBERS

مثال دوم : دو عدد $A = 84$, $B = 67$ را در نظر می‌گیریم و فرض می‌کنیم جایی که برای هر عدد در نظر می‌گیریم (به صورت اعداد علامت دار) 8 بیت باشد

واضح است که هر کدام از اعداد در رنج $(-128 \ldots +127)$ قرار دارند ولی جمع دو عدد 151 در این رنج قرار ندارد.

می‌خواهیم محاسبات زیر را انجام دهیم

$$\begin{array}{r} +84 \; + \\ +67 \\ \hline ? \end{array} \qquad \begin{array}{r} +84 \; + \\ -67 \\ \hline ? \end{array} \qquad \begin{array}{r} -84 \; + \\ +67 \\ \hline ? \end{array} \qquad \begin{array}{r} -84 \; + \\ -67 \\ \hline ? \end{array}$$

برای اینکار ابتدا اعداد $+84$, -84 , $+67$, -67 را در مبنای ۲ با استفاده از سیستم متمم ۲ نمایش می‌دهیم (هر کدام با ۸ بیتی که فرض کرده‌ایم)

	علامت	64	32	16	8	4	2	1
+84	0	1	0	1	0	1	0	0
-84	1	0	1	0	1	1	0	0

	علامت	64	32	16	8	4	2	1
+67	0	1	0	0	0	0	1	1
-67	1	0	1	1	1	1	0	1

COMPLEMENTS OF NUMBERS

قبل از این که محاسبات را انجام دهیم. فرض کنید اگر سئوال شود با فرض اعداد در سیستم متمم ۲ و علامت دار با 8bit

مثلاً عدد 10111101 چه عددی است؟

ابتدا به بیت هشتم نگاه می کنیم می بینیم این عدد 1 است پس نتیجه می گیریم عدد منفی است

ولی اینکه این چه عددی را نشان می دهد چون در سیستم متمم 2 هستیم باید ابتدا متمم ۲ آن را

به دست آوریم که می شود $\begin{array}{cccccccc} & 64 & & & & & 2 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}$

حال این عدد مثبت است چون بیت هشتم صفر است (و یا ما متمم ۲ عددی منفی را به دست آوردیم

پس قطعاً عددی مثبت است) و مقدار عدد با استفاده از ارزش مکانی بیتها برابر با $64+2+1$

یعنی عدد $+67$ را نشان می دهیم پس متمم ۲ آن قطعاً عدد -67 است که جواب

سئوال داده شده است

COMPLEMENTS OF NUMBERS

a محاسبه

$+84 + (-67)$

```
        1  1 | 1 1 1 1 0 0 0    ← carry
+84     0  0 | 1 0 1 0 1 0 0
-67     0  1 | 0 1 1 1 1 0 1   +
        ---------------------
     1  0    | 0 0 1 0 0 0 1   = +17
                   16      1
```

توضیح ؟

b محاسبه

$-84 + (+67)$

```
        0  0 | 0 0 0 0 0 0 0
-84     0  1 | 0 1 0 1 1 0 0
+67     0  0 | 1 0 0 0 0 1 1   +
        ---------------------
        0  1 | 1 1 0 1 1 1 1
```

عدد منفی است

متمم 2

```
+
0 0 0 1 0 0 0 1 = +17
      16      1
```

پس عدد حاصل 17- است

ابتدا دو حالتی را در نظر گرفتیم که مختلف العلامت هستند. بدیهی است که عدد حاصل از لحاظ اندازه از هر یک از دو عدد 84, 67 کوچکتر خواهد بود و بنابراین در قالب 8bit جا می‌شود یا به عبارتی خطای سرریز یا overflow را نداریم

COMPLEMENTS OF NUMBERS

		0	1	0	0	0	0	0	0
+84	0	0	1	0	1	0	1	0	0
+67	0	0	1	0	0	0	0	1	1
		1	0	0	1	0	1	1	1

نشان دهنده منفی بودن

+84 + +67 می‌شه

C

در عدد حاصل بیت هشتم نشان می دهد که نتیجه منفی است
درحالیکه دو عدد مثبت را جمع کردیم به عبارت دیگر خطایی اتفاق افتاده است
اگر فرض کنیم که هر دو عدد با جای ۱۶ بیت و یا حتی ۹ بیت و یا بیشتر نشان داده شده باشند
آنگاه بیت شانزدهم نشان دهنده علامت و پانزده بیت دیگر مقدار را نشان می دهد
(در حالت ۱۶ بیت شکل این است که در بیت هرکدام از عددها از ابتدا ۸ تا صفر قرار دهیم)

در اینصورت عدد حاصل	1	0	0	1	0	1	1	1
	128			16		4	2	1

به درستی عدد 151 است $128+16+4+2+1=151$

COMPLEMENTS OF NUMBERS

d) محاسبه

$$\begin{array}{r} -84 \; + \\ -67 \\ \hline \end{array}$$

	1	0	1	1	1	1	0	0	0
−84	0	1	0	1	0	1	1	0	0
−67	0	1	0	1	1	1	1	0	1
	1	0	1	1	0	1	0	0	1

نشان دهنده مثبت بودن

مشابه با حالت قبل در اینجا هم
دو عدد منفی جمع شده‌اند و حاصل به صورت مثبت نشان داده می‌شود
که دلالت بر وقوع خطا دارد.
در این دو حالت اخیر هر دو عدد متحدالعلامت بودند و حاصلجمع به صورت +151 و یا -151
است که در رنج (-128 ... 0 ... +127) نمی‌گنجد
حال فرض کنید که دو عدد مثلاً +84 و +67 از نتیجه یک سری محاسبات به دست آمده‌اند
و در دو ثبات A و B قرار داده شده‌اند کامپیوتر از کجا باید بداند که خطای سرریز دارد
و در حالتهایی که دو عدد همنشان در قالب ۸ بیت می‌گنجد و خطای سرریز رخ نمی‌دهد
بتواند این حالتها را از هم تفکیک نماید.

BINARY CODES

- ***Binary-Coded Decimal (BCD)***

It is important to realize that BCD numbers are decimal numbers and not binary numbers, although they use bits in their representation.

$$(185)_{10} = (0001\ 1000\ 0101)_{BCD} = (10111001)_2$$

Binary-Coded Decimal (BCD)

Decimal Symbol	BCD Digit
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

BINARY CODES

- **BCD Addition**

4	0100	4	0100	8	1000
+5	+0101	+8	+1000	+9	1001
9	1001	12	1100	17	10001
			+0110		+0110
			10010		10111

Consider the addition of 184 + 576 = 760 in BCD:

BCD	1	1		
	0001	1000	0100	184
	+0101	0111	0110	+576
Binary sum	0111	10000	1010	
Add 6		0110	0110	
BCD sum	0111	0110	0000	760

Binary-Coded Decimal (BCD)

Decimal Symbol	BCD Digit
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

BINARY CODES

-

Four Different Binary Codes for the Decimal Digits

Decimal Digit	BCD 8421	2421	Excess-3	8, 4, −2, −1
0	0000	0000	0011	0000
1	0001	0001	0100	0111
2	0010	0010	0101	0110
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	1011	1000	1011
6	0110	1100	1001	1010
7	0111	1101	1010	1001
8	1000	1110	1011	1000
9	1001	1111	1100	1111
Unused bit combinations	1010	0101	0000	0001
	1011	0110	0001	0010
	1100	0111	0010	0011
	1101	1000	1101	1100
	1110	1001	1110	1101
	1111	1010	1111	1110

BINARY CODES

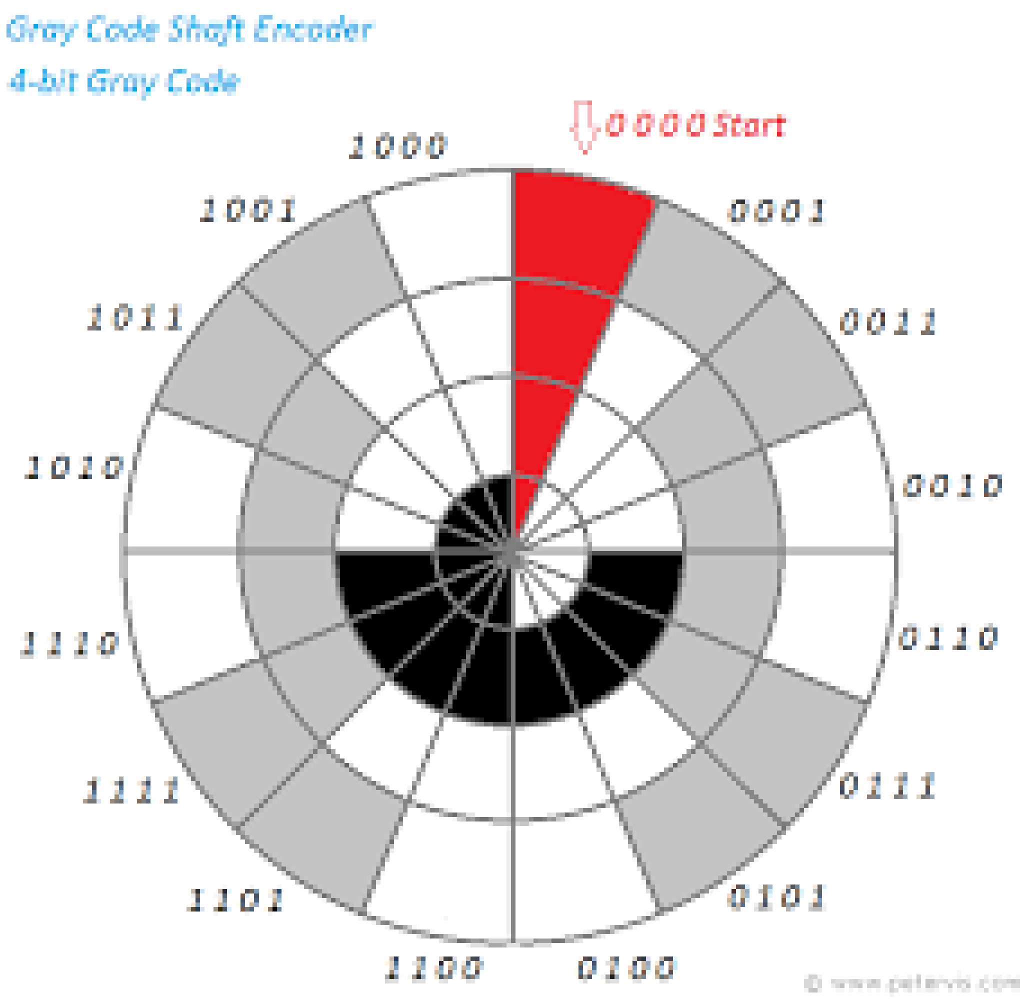
- self-complementing codes

The 2421 and the excess-3 codes are examples of self-complementing codes. Such codes have the property that the 9's complement of a decimal number is obtained directly by changing 1's to 0's and 0's to 1's (i.e., by complementing each bit in the pattern).

For example,
decimal 395
is represented in the excess-3 code as 0110 1100 1000.
The 9's complement of 604 is represented as 1001 0011 0111,
which is obtained simply by complementing each bit of the code
(as with the 1's complement of binary numbers).

BINARY CODES

- **Gray Code**

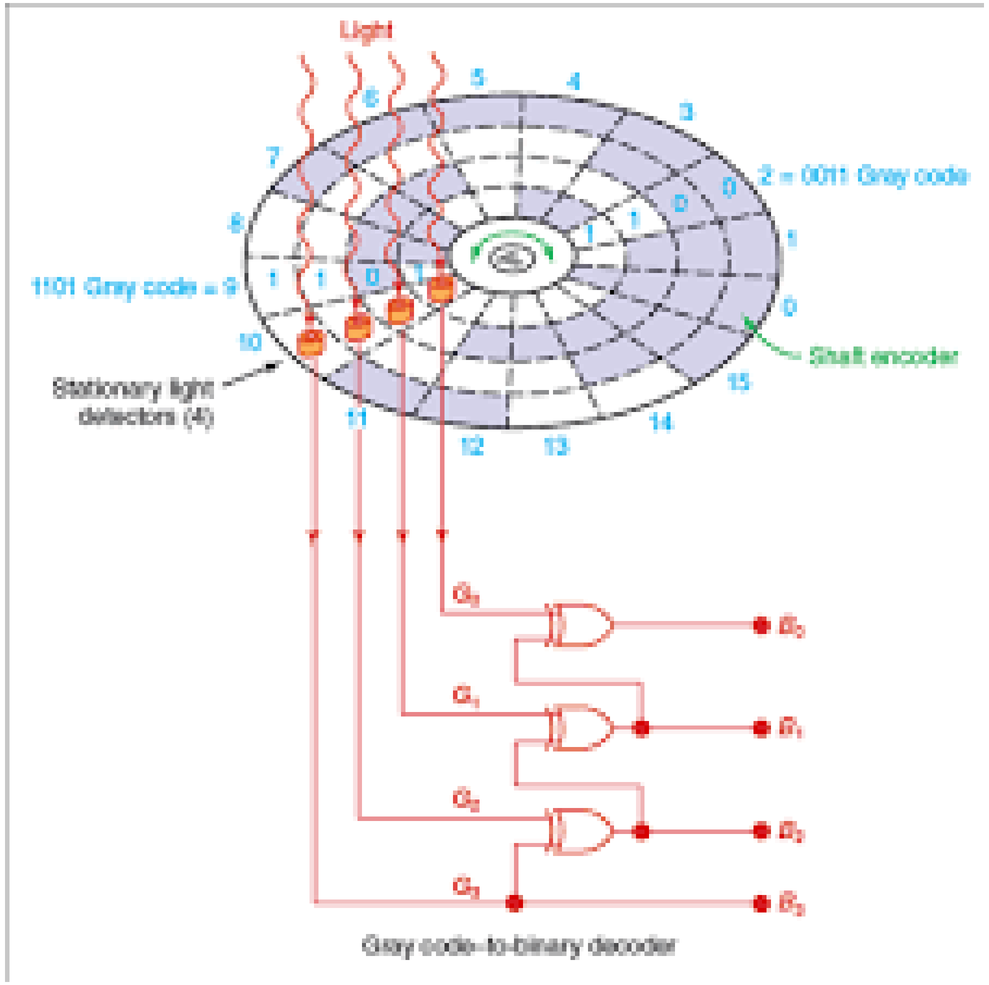


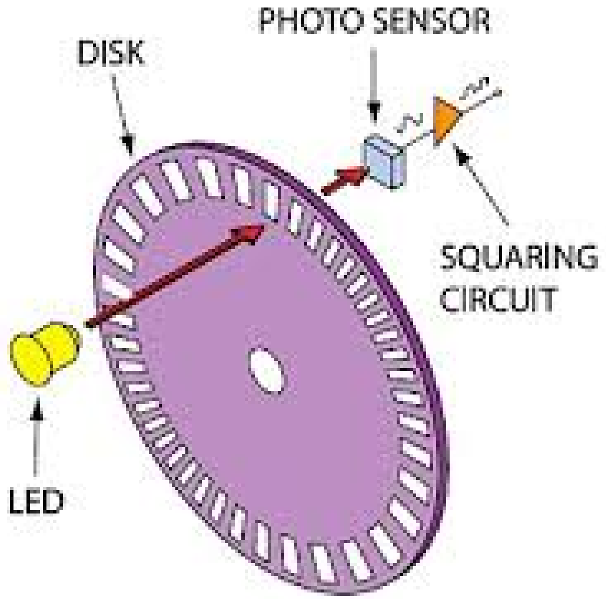
Gray Code

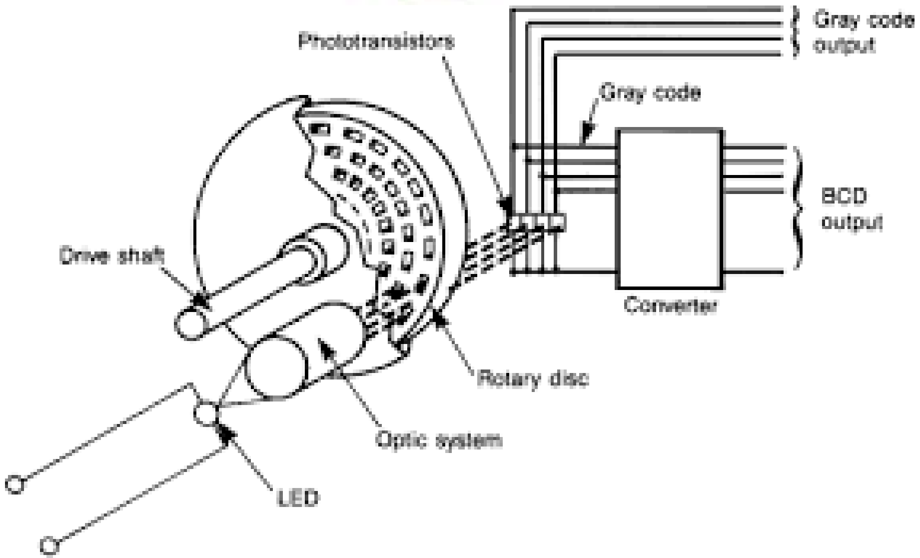
Gray Code	Decimal Equivalent
0000	0
0001	1
0011	2
0010	3
0110	4
0111	5
0101	6
0100	7
1100	8
1101	9
1111	10
1110	11
1010	12
1011	13
1001	14
1000	15

BINARY CODES

- Gray Code







BINARY CODES

- *American Standard Code for Information Interchange (ASCII)*

American Standard Code for Information Interchange (ASCII)

	$b_7b_6b_5$							
$b_4b_3b_2b_1$	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	“	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	‘	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	–	=	M	]	m	}
1110	SO	RS	.	>	N	∧	n	~
1111	SI	US	/	?	O	—	o	DEL

BINARY LOGIC

Truth Tables of Logical Operations

AND

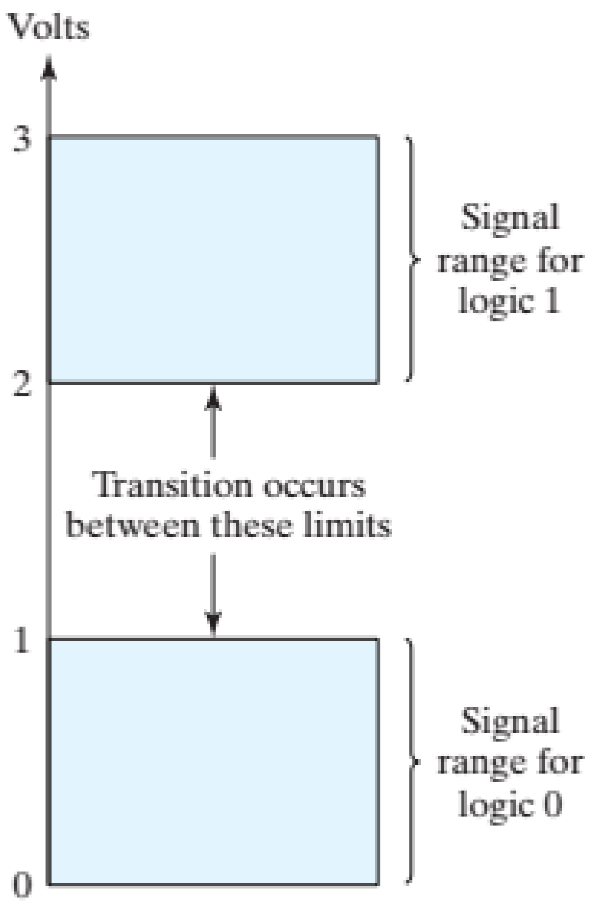
x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

OR

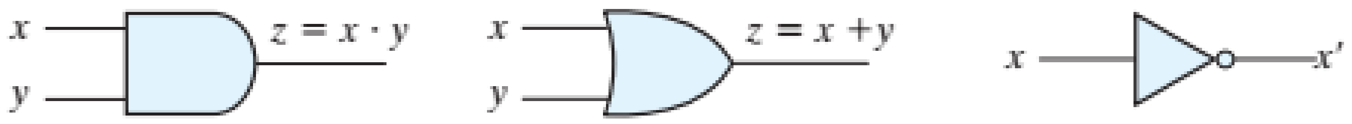
x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

NOT

x	x'
0	1
1	0



- **Logic Gates**



(a) Two-input AND gate (b) Two-input OR gate (c) NOT gate or inverter

B I N A RY LOGIC

- **Input–output signals for gates**