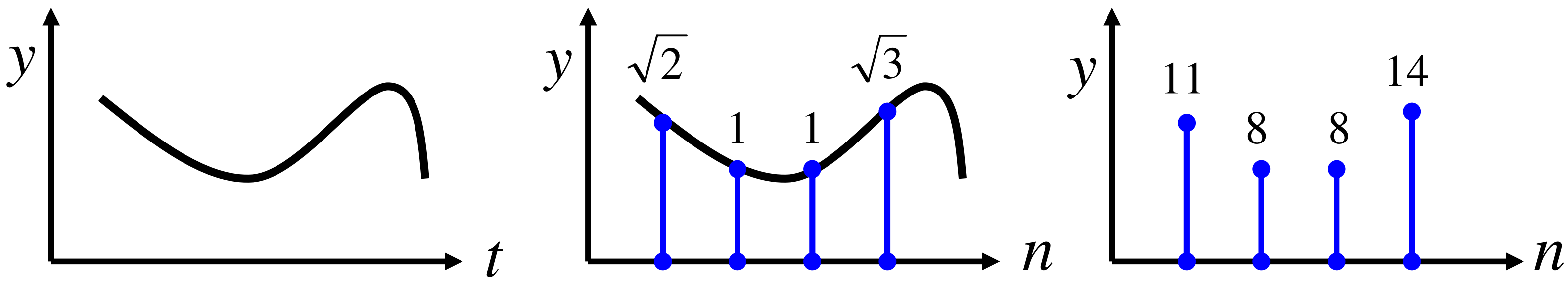
Binary Numbers

Chun-Jen Tsai
National Chiao Tung University
09/20/2012

Signals (1/2)

- An signal is a function of time (or space)
  - The range of the function represents physical quantity of information
- All signals are **analog** by nature (i.e. both range and domain are real numbers); however, for digital systems, we only process digital signals



Analog Signal   Discrete-Time Signal   Digital Signal

Signals (2/2)

- For digital systems, the variable (i.e. output of the function) takes on discrete values
  - Two-level (binary) values are the most prevalent
- Binary values are represented abstractly by:
  - digits 0 and 1
  - symbols False (F) and True (T)
  - symbols Low (L) and High (H)
  - states On and Off

Digital Number Systems

- A digital number system only allows discrete numbers (e.g. integers)
  - The base (or radix) of the number system can be any positive integers
- If the base is r (represented as a decimal number), we can convert any base-r number to a **decimal number** (base-10 number) as follows:

$$\ldots\ a_4 a_3 a_2 a_1 a_0\,.\, a_{-1} a_{-2}\ \ldots$$

$$= \ldots\ r^4 \cdot a_4 + r^3 \cdot a_3 + r^2 \cdot a_2 + r^1 \cdot a_1 + r^0 \cdot a_0 + r^{-1} \cdot a_{-1} + r^{-2} \cdot a_{-2} + \ldots$$

Examples of Digital Numbers

- Base-2 number (aka. binary number):

$$(11010.11)_2 = (26.75)_{10}$$
$$= 1\times 2^4 + 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 0\times 2^0 + 1\times 2^{-1} + 1\times 2^{-2}$$

- Base-5 number:

$$(4021.2)_5$$
$$= 4\times 5^3 + 0\times 5^2 + 2\times 5^1 + 1\times 5^0 + 2\times 5^{-1} = (511.5)_{10}$$

- Base-16 number:

$$(\mathrm{B65F})_{16} = 11\times 16^3 + 6\times 16^2 + 5\times 16^1 + 15\times 16^0 = (46{,}687)_{10}$$

Binary Arithmetic

Addition

```
Augend:     101101
Addend:    +100111
           -------
Sum:       1010100
```

Subtraction

```
Minuend:          101101
Subtrahend:      -100111
                 -------
Difference:       000110
```

Multiplication

```
Multiplicand:          1011
Multiplier:          ×  101
                     ------
Partial Products:      1011
                      0000-
                     1011--
                     ------
Product:             110111
```

Number-Base Conversion (1/3)

- Convert decimal $(41)_{10}$ to binary:

	Quotient		Remainder	Coefficient
$41/2 =$	20	$+$	$1/2$	$a_0 = 1$
$20/2 =$	10	$+$	0	$a_1 = 0$
$41/2 =$	20	$+$	0	$a_2 = 0$
$41/2 =$	20	$+$	$1/2$	$a_3 = 1$
$41/2 =$	20	$+$	0	$a_4 = 0$
$41/2 =$	20	$+$	$1/2$	$a_5 = 1$

$$\rightarrow (41)_{10} = (a_5a_4a_3a_2a_1a_0)_2 = (101001)_2$$

Number-Base Conversion (2/3)

- The conversion process can be done conveniently in a tabular form
  - Example: convert decimal $(153)_{10}$ to octal. The base r is 8.

Integer	Remainder
153	
19	1
2	3
0	2

$= (231)_8$

Number-Base Conversion (3/3)

- Convert $(0.6875)_{10}$ to binary:

	Integer		Fraction	Coefficient
$0.6875 \times 2 =$	1	+	0.3750	$a_{-1} = 1$
$0.3750 \times 2 =$	0	+	0.7500	$a_{-2} = 0$
$0.7500 \times 2 =$	1	+	0.5000	$a_{-3} = 1$
$0.5000 \times 2 =$	1	+	0.0000	$a_{-4} = 1$

$$\rightarrow (0.6875)_{10} = (0.a_{-1}a_{-2}a_{-3}a_{-4})_2 = (0.1011)_2$$

- To convert a decimal fraction to a number of base r, multiplication should be by r instead of 2.

Octal and Hexadecimal Numbers

- Why do we use octal or hex notations?
  - Easy to read
  - Easy to convert

Decimal (base 10)	Binary (base 2)	Octal (base 8)	Hexadecimal (base 16)
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Signed Numbers

- Leopold Kronecker (according to H. Weber, 1893):

  "God made the natural numbers; all else
  is the work of man."

- How do we represent singed binary numbers?
  - Complements of numbers
  - Sign-magnitude notation
  - Excess notation†

† Check your textbook of Intro. To Computer Science.

Complements of Numbers (1/2)

- Two types of complements for base-r system: the radix complement and diminished radix complement.
  - The first one is called the r's complement and the second one is called the $(r - 1)$'s complement
- Diminished Radix Complement
  - The $(r - 1)$'s complement of an n-digit number N in base r is defined as $(r^n - 1) - N$.
- Example: base-10 complement

The 9's complement of 546700 is 999999 – 546700 = 453299.

The 9's complement of 012398 is 999999 – 012398 = 987601.

Complements of Numbers (2/2)

- Radix Complement
  - The r's complement of an n-digit number N in base r is defined as $r^n - N$ for $N \neq 0$ and as 0 for $N = 0$.
  - The r's complement is obtained by adding 1 to the $(r-1)$'s complement, since $r^n - N = [(r^n - 1) - N] + 1$.
- Example: Base-10 complement

  The 10's complement of 012398 is 987602
  The 10's complement of 246700 is 753300
- Example: Base-2 complement

  The 2's complement of 1101100 is 0010100
  The 2's complement of 0110111 is 1001001

Subtraction with Complement

- The subtraction of two n-digit unsigned numbers $M - N$ in base r can be done as follows:
  - Add the minuend M to r's complement of the subtrahend N. Note that $M + (r^n - N) = M - N + r^n$.
  - If $M \geq N$, the sum will produce and end carry r^n, which can be discarded; what is left is the result $M - N$.
  - If M < N, the sum does not produce an end carry and is equal to $r^n - (N - M)$, which is the r's complement of $(M - N)$.

Example of Subtraction

- Using 10’s complement, subtract 72532 – 3250

$$
\begin{array}{rrr}
 & M = & 72532 \\
\text{10's complement of} & N = & \underline{+\,96750} \\
 & \text{Sum} = & 169282 \\
 & \text{Discard end carry } 10^5 = & \underline{-\,100000} \\
 & \text{Answer} = & 69282
\end{array}
$$

- Using 10’s complement, subtract 3250 – 72532

$$
\begin{array}{rrr}
 & \mathrm{M} = & 03250 \\
\text{10's complement of} & \mathrm{N} = & \underline{+\,27468} \\
 & \text{Sum} = & 30718
\end{array}
$$

Subtraction with End Carry

- Compute $X - Y$ where $X = 1010100$ and $Y = 1000011$ using 2's complement:

$$
\begin{array}{rr}
X = & 1010100 \\
\text{2's complement of } Y = & \underline{+0111101} \\
\text{Sum} = & 10010001 \\
\text{Discard end carry } 2^7 = & \underline{-10000000} \\
\text{Answer. } X - Y = & 0010001
\end{array}
$$

Subtraction by $(r-1)$'s complement

- In previous example, compute (a) $X-Y$ and (b) $Y-X$ using 1's complement:

(a) $X - Y = 1010100 - 1000011$

$$
\begin{array}{rr}
X = & 1010100 \\
\text{1's complement of } Y = & \pm\, 0111100 \\
\text{Sum} = & 10010000 \\
\text{End-around carry} = & \underline{+ \qquad 1} \\
\text{Answer. } X - Y = & 0010001
\end{array}
$$

(b) $Y - X = 1000011 - 1010100$

$$
\begin{array}{rr}
Y = & 1000011 \\
\text{1's complement of } X = & \underline{+\, 0101011} \\
\text{Sum} = & 1101110
\end{array}
$$

Signed Binary Numbers

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	—	—

In all forms, a 1 in MSB represents negative numbers

Binary Codes

- Binary numbers are easy for computer to manipulate, but, when we have to represents “things” that are **non-binary** in nature, we have to design a new coding rule:
  - BCD code
  - Gray code
  - ASCII code
  - Error-detecting code

BCD Code (1/3)

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

1. A number with *k* decimal digits will require 4*k* bits in BCD. Decimal 396 is represented in BCD with 12bits as 0011 1001 0110.

2. A decimal number in BCD is the same as its equivalent binary number only when the number is between 0 and 9.

3. The binary combinations 1010 through 1111 are not used and have no meaning in BCD.

BCD Code (2/3)

- Example:

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

- BCD Addition:

$$\begin{array}{rr}
4 & 0100 \\
+5 & +0101 \\
\hline
9 & 1001
\end{array}
\qquad
\begin{array}{rr}
4 & 0100 \\
+8 & +1000 \\
\hline
12 & 1100 \\
 & +0110 \\
\cline{2-2}
 & 10010
\end{array}
\qquad
\begin{array}{rr}
8 & 1000 \\
+9 & +1001 \\
\hline
17 & 10001 \\
 & +0110 \\
\cline{2-2}
 & 10111
\end{array}$$

BCD Code (3/3)

- Carry processing in BCD addition:
  - 184 + 576 = 760

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

Other Decimal Codes

- Weighted codes and self-complementing 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
	1010	0101	0000	0001
Unused	1011	0110	0001	0010
bit	1100	0111	0010	0011
combi-	1101	1000	1101	1100
nations	1110	1001	1110	1101
	1111	1010	1111	1110

Gray Code

- Gray code is designed in a way that only 1-bit difference is present between two neighboring numbers

- Great for encoding continually varying numbers

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

ASCII Code (1/2)

American Standard Code for Information Interchange (ASCII)

$b_4b_3b_2b_1$ \ $b_7b_6b_5$	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

ASCII Code (2/2)

- American Standard Code for Information Interchange
- A popular code used to represent information sent as character-based data.
- It uses 7-bits to represent:
  - 94 Graphic printing characters.
  - 34 Non-printing characters
- Some non-printing characters are used for text format (e.g. BS = Backspace, CR = carriage return)
- Other non-printing characters are used for record marking and flow control (e.g. STX and ETX start and end text areas).

ASCII Design Properties

- ASCII has some interesting properties:
  - Digits 0 to 9 span Hexadecimal values $(30)_{16}$ to $(39)_{16}$
  - Upper case A - Z span $(41)_{16}$ to $(5A)_{16}$
  - Upper case a - z span $(61)_{16}$ to $(7A)_{16}$
    - Lower to upper case conversion (and vice versa) occurs by flipping bit 6
  - Delete (DEL) is all bits set, a carryover from when punched paper tape was used to store messages.
    - Punching all holes in a row erased a mistake!

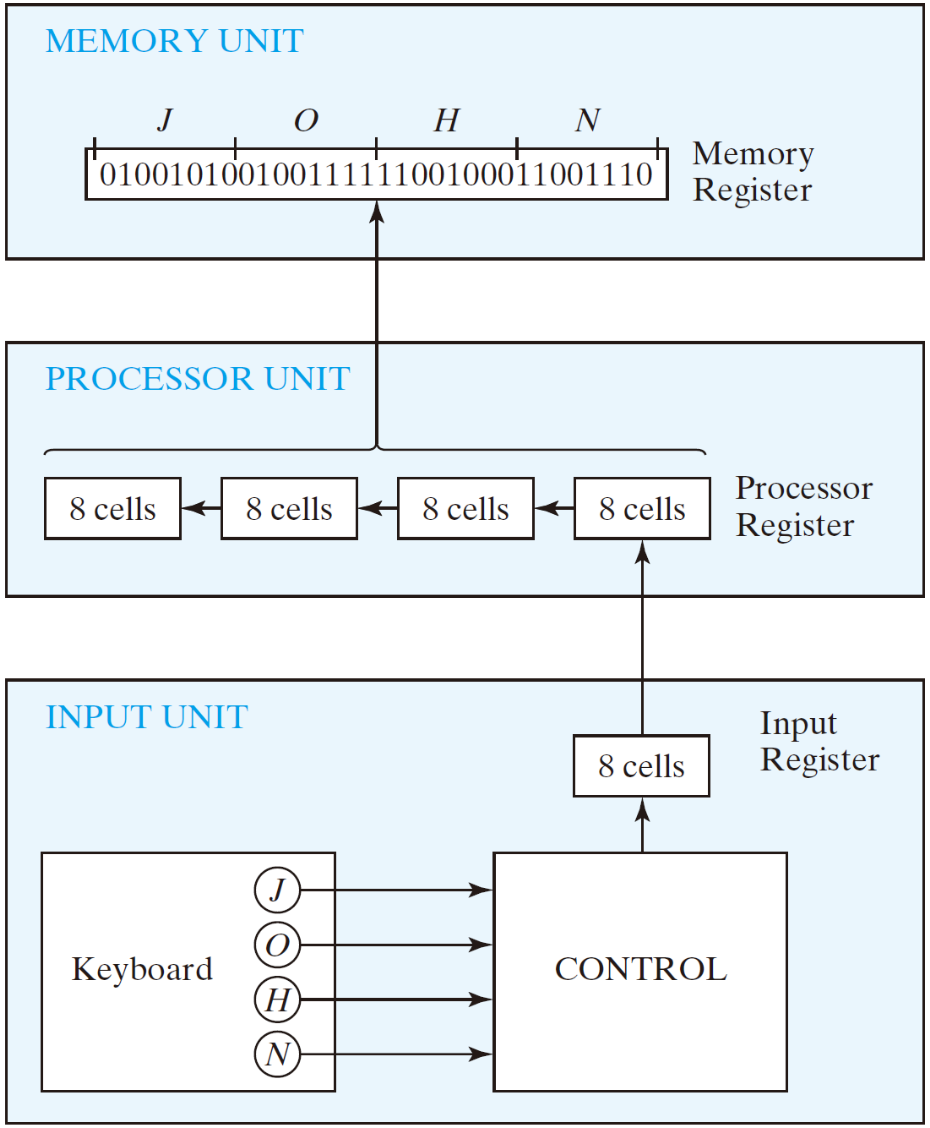
Error-Detecting Code

- To detect errors in data communication and processing, redundant bits are added to code words to detect and/or correct errors.
- A simple form of error-detecting code is parity coding. A parity bit is an extra bit added to a code word to make the total number of 1's either even or odd.
- For error-correcting codes, “code distance” between any two code word should be large enough so that bit errors will not cause ambiguity in decoding.

Binary Storage and Registers

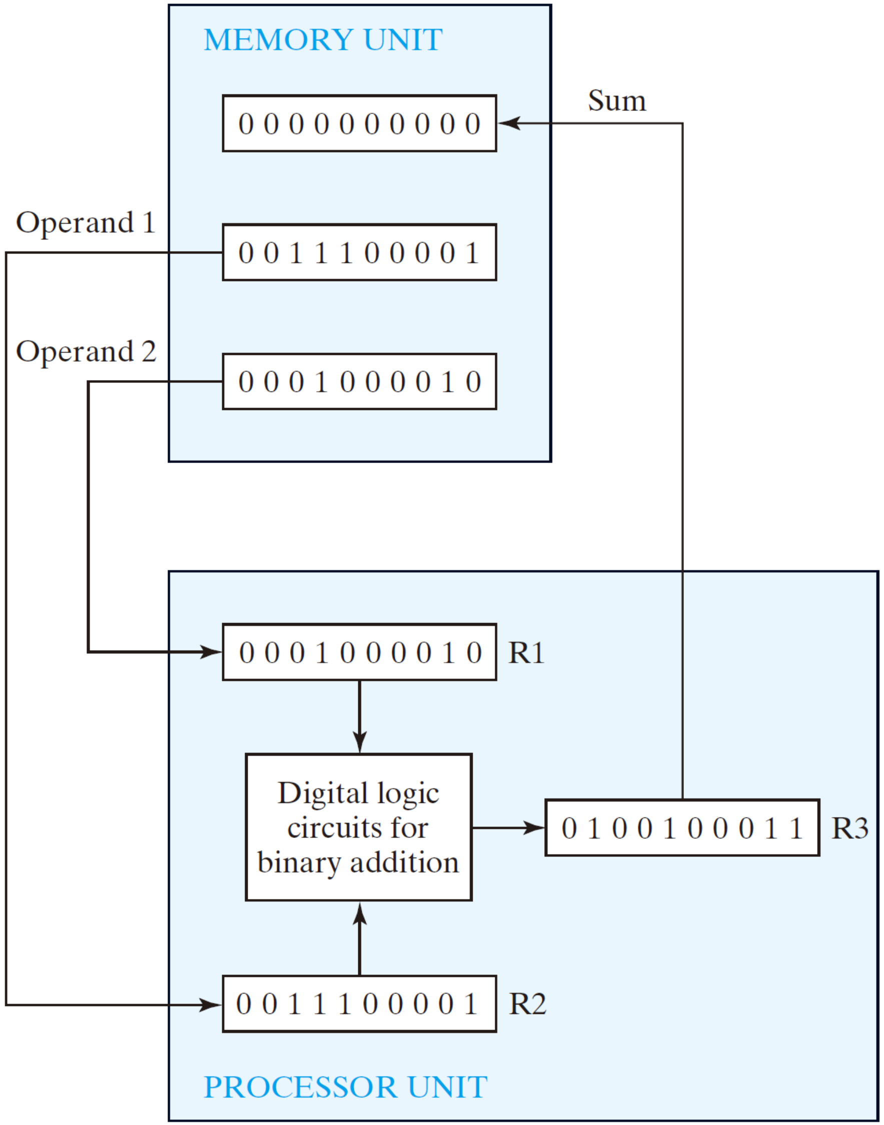
- Registers
  - A binary cell is a device that possesses two stable states and is capable of storing one of the two states.
  - A register is a group of binary cells. A register with n cells can store discrete quantity of information that contains n bits.
- Register Transfer
  - A transfer of the information stored in one register to another.
  - One of the major operations in digital system.

Example of Transfer of Information



Transfer Through Processing Unit

- A register transfer operation may pass trough an arithmetic unit so that the original information will be modified before it is stored back to another register.



Binary Logic

- Binary logic consists of binary variables and a set of logical operations.
- The variables are designated by letters of the alphabet, such as A, B, C, x, y, z, etc, with each variable having values 1 or 0.
- Three basic logical operations: AND, OR, and NOT.

Truth Tables of Logical Operations

AND

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

OR

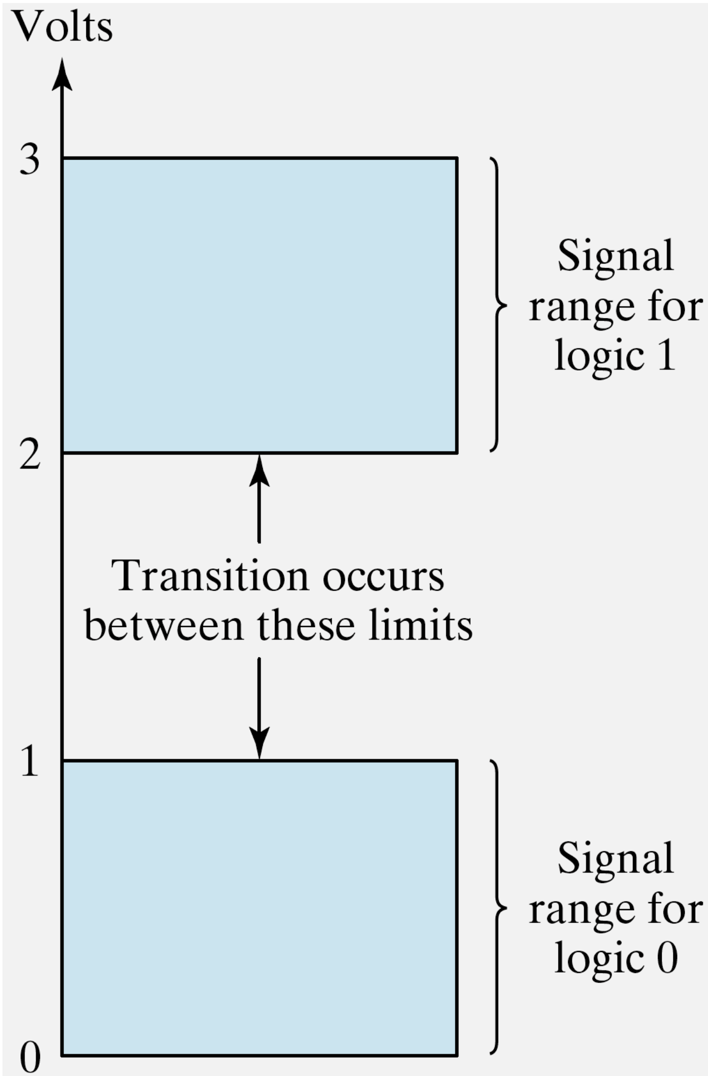
x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

NOT

x	x'
0	1
1	0

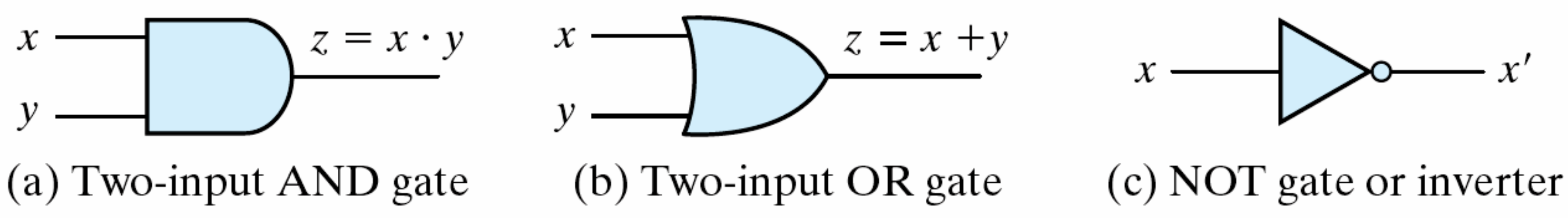
Electrical Form of Logic Values

- A binary logic signal can be represented by the voltage of the electrical current:

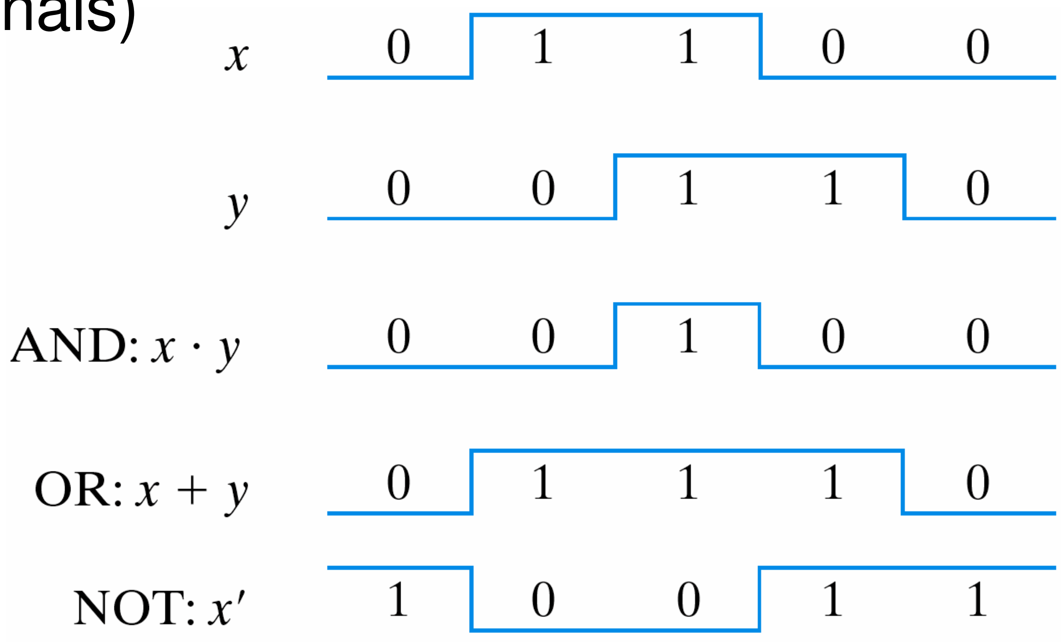


Logic Gates (1/2)

- Graphical symbols for logic gates:

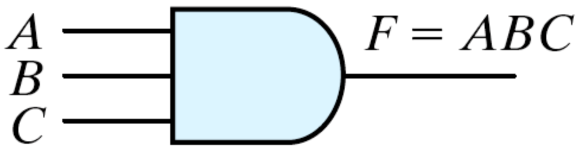


- Wave forms (signals) of binary logic operations

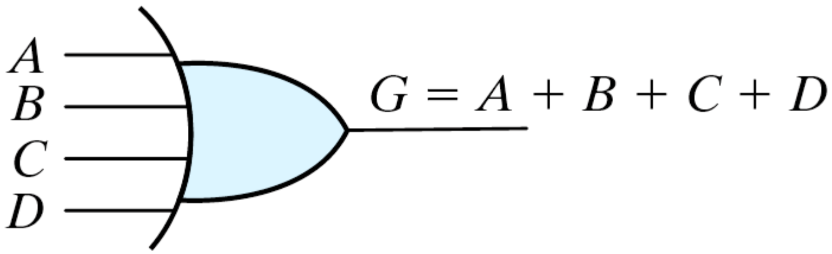


Logic Gates (2/2)

- A logic gate can have more than two inputs as well:



(a) Three-input AND gate



(b) Four-input OR gate