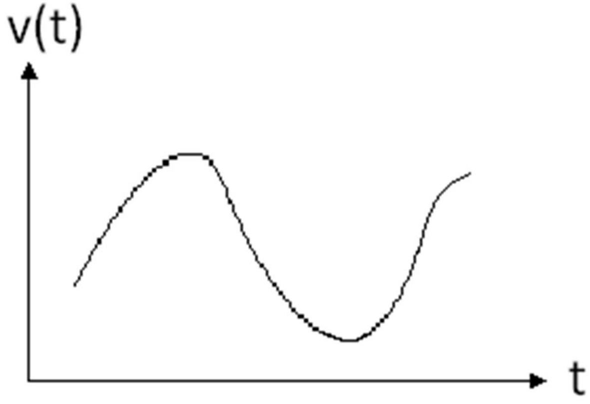
CS210201 Logic Design 數位邏輯設計

Digital Systems and Binary Numbers
Chapter 1

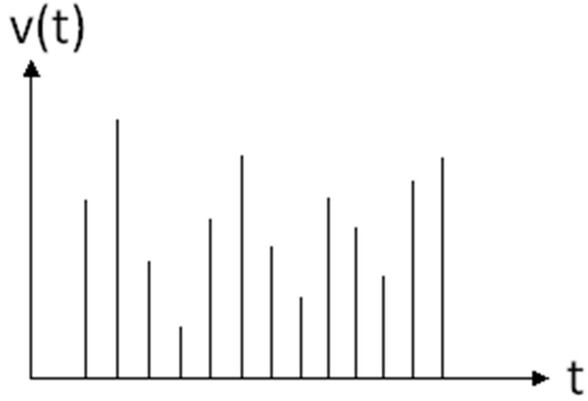
Chapter Objectives

- Understand binary number system.
- Know how to Convert between binary, octal, decimal, and hexadecimal numbers.
- Know how to take the complement and reduced radix complement of a number.
- Know how to form the code of a number.

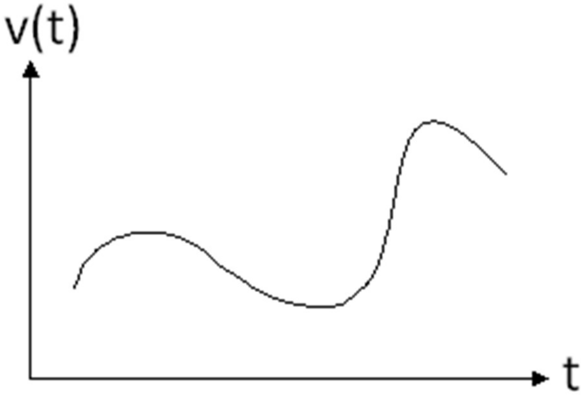
Digital Systems and Digital Signals



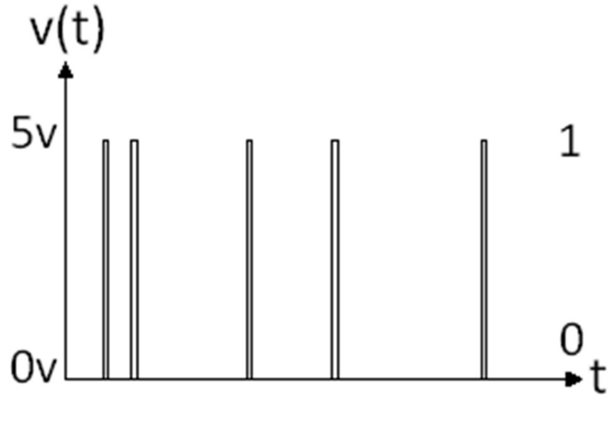
Continuous signal



Discrete signal



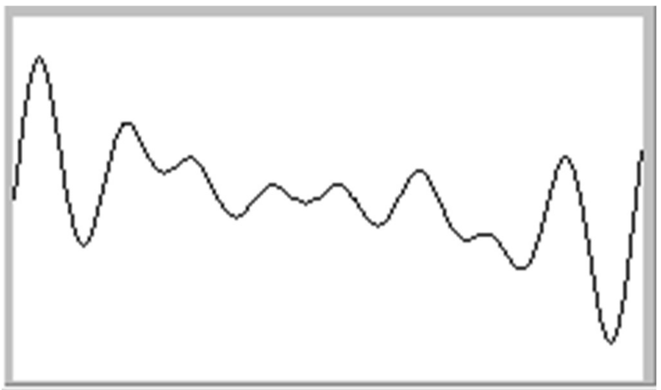
Analog signal



Digital signal

Digital vs Analog

- Analog: continuous values used (e.g.-8.0 to 12.0)
- Digital: a discrete set of values used (e.g. {0,1})
- PC is an example of a digital system
- The microphone & speaker of a sound system are analog devices (sound is analog in nature)



Digital Advantages

- Digitizing e.g. convert audio signal (analog) into digital form for storage on a CD.

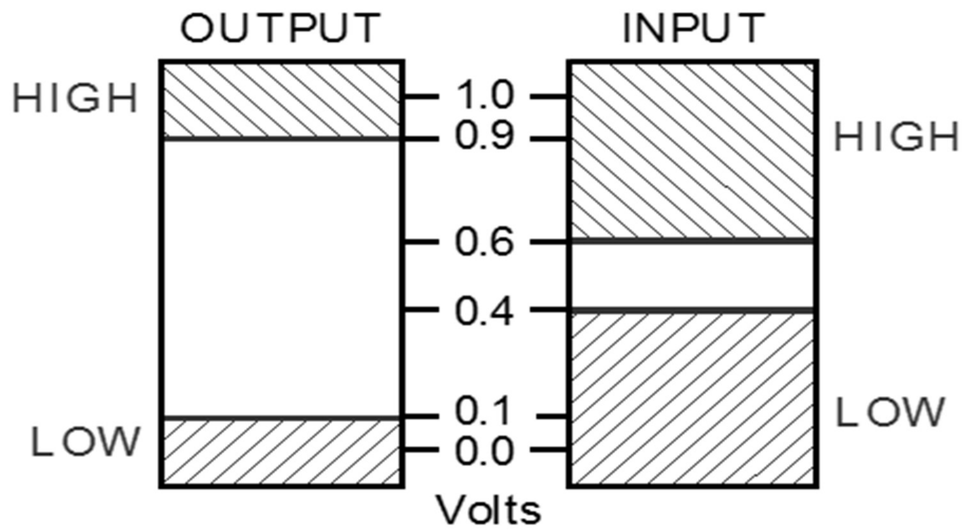
- Digital data can be stored compactly.
- Digital data can be processed and transmitted more efficiently and reliably than analog data.
- Noise does not affect digital data as much as analog data.
- Digital systems are easier to design & more reliable.

Signal

- For digital systems, the variable takes on discrete values
  - Two level, or binary values are the most prevalent values
- Binary values are represented abstractly by:
  - digits 0 and 1
  - words (symbols) False (F) and True (T)
  - words (symbols) Low (L) and High (H)
  - and words On and Off.
- Binary (two) values are represented by values or ranges of values of physical quantities

Binary Logic and Digital Circuit

- Binary logic
  - uses only two values: 1, 0 (True, False)
- We can use Voltage Range to define logic 1 and logic 0 in electronic devices.
- Usually, we associate HIGH voltage with 1 (True) and LOW voltage with 0 (False).



Digital System

- A digital system is an interconnection of digital modules. To understand the operation of each digital module, it is necessary to have a basic knowledge of **digital circuits** and their **logical function**.

- A major trend in digital design methodology is the use of a **HDL** to describe and simulate the functionality of a digital circuit, it is important that students become familiar with an HDL-based design methodology.

Binary Numbers

- Decimal number

$$\ldots\ a_5 a_4 a_3 a_2 a_1 . a_{-1} a_{-2} a_{-3} \ldots$$

Decimal point

a_j

Base or radix

Power

$$\boxed{\cdots + 10^5 a_5 + 10^4 a_4 + 10^3 a_3 + 10^2 a_2 + 10^1 a_1 + 10^0 a_0 + 10^{-1} a_{-1} + 10^{-2} a_{-2} + 10^{-3} a_{-3} + \cdots}$$

Example:

$$7{,}329 = 7 \times 10^3 + 3 \times 10^2 + 2 \times 10^1 + 9 \times 10^0$$

- General form of base-r system

$$\boxed{a_n \cdot r^n + a_{n-1} \cdot r^{n-1} + \cdots + a_2 \cdot r^2 + a_1 \cdot r^1 + a_0 + a_{-1} \cdot r^{-1} + a_{-2} \cdot r^{-2} + \cdots + a_{-m} \cdot r^{-m}}$$

Coefficient: $a_j = 0$ to $r - 1$

Binary Numbers

Example: Base-2 number

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

Example: Base-5 number

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

Example: Base-16 number

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

Binary Numbers

Example: Base-2 number

$$(110101)_2 = 32 + 16 + 4 + 1 = (53)_{10}$$

Special Powers of 2

- 2^{10} (1024) is Kilo, denoted "K"
- 2^{20} (1,048,576) is Mega, denoted "M"
- 2^{30} (1,073, 741,824) is Giga, denoted "G"

Powers of two

⇒ **Table 1.1**

Table 1.1
Powers of Two

n	2^n	n	2^n	n	2^n
0	1	8	256	16	65,536
1	2	9	512	17	131,072
2	4	10	1,024 (1K)	18	262,144
3	8	11	2,048	19	524,288
4	16	12	4,096 (4K)	20	1,048,576 (1M)
5	32	13	8,192	21	2,097,152
6	64	14	16,384	22	4,194,304
7	128	15	32,768	23	8,388,608

Arithmetic operation

Negative Numbers

- Sign-magnitude
  - (sign bit) + (positive magnitude)
  - Hard to implement in hardware
- 1's complement
  - $\overline{N} = (2^n - 1) - N$
- 2's complement
  - $N^* = 2^n - N$, for $n = 4$, $-N \Rightarrow 16-N$
  - e.g., $-3 \Rightarrow 16-3 = 13 = 1101_2$
- Easy way to form 1's complement
  - Complementing N bit-by-bit
- Easy way to form 2's complement
  - Complementing N bit-by-bit and then adding 1

Non-numeric Binary Codes

- Given *n* binary digits (called bits), a binary code is a mapping from a set of represented elements to a subset of the 2^n binary numbers
- E.g. Light On/Off: 1/0
- E.g. Eight different colors

000	White
001	Red
010	Blue
011	Yellow

100	Purple
101	Orange
110	Green
111	Black

Representing Information with Digital Signals

- Pick a representation based on operations required
- E.g. A color representation to support the operation of subtractive color mixing (e.g., red + blue = purple)
  - No colors – white
  - Primary colors – red, yellow, blue
  - Derived colors – orange, purple, green, black

Color	Code
White	000
Red	001
Yellow	010
Blue	100
Orange	011
Purple	101
Green	110
Black	111

Binary-Coded-Decimal (BCD)

- A common coding method to code each digit of a decimal number in terms of binary bits.
- Each digit is coded by its binary equivalent.
  - E.g. The digit “3” is encoded as 0011
  - E.g. The number 30 is encoded as 00110000
- BCD is NOT converting the number as a whole into binary.

BCD (Cont'd)

TABLE 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

Table 1-3 Binary-Coded Decimal (BCD)

Q. What is the BCD representation of the decimal no. 547?

Q. What does the BCD no. 000110011001 represent?

ASCII Character Code

- ASCII (American Standard Code for Information Interchange) is the standard binary code for the set of alphanumeric characters
- The character set includes:
  - numerals: 0, 1, …, 9
  - alphabets: a, b, …, z, A, B, …, Z
  - special printable characters: $, #, (, +, …
  - control characters (used for data transmission and printed text formatting)

TABLE 1-4
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	NULL	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

Control Characters:

NULL	NULL	DLE	Data link escape
SOH	Start of heading	DC1	Device control 1
STX	Start of text	DC2	Device control 2
ETX	End of text	DC3	Device control 3
EOT	End of transmission	DC4	Device control 4
ENQ	Enquiry	NAK	Negative acknowledge
ACK	Acknowledge	SYN	Synchronous idle
BEL	Bell	ETB	End of transmission block
BS	Backspace	CAN	Cancel
HT	Horizontal tab	EM	End of medium
LF	Line feed	SUB	Substitute
VT	Vertical tab	ESC	Escape
FF	Form feed	FS	File separator
CR	Carriage return	GS	Group separator
SO	Shift out	RS	Record separator
SI	Shift in	US	Unit separator
SP	Space	DEL	Delete

Table 1-4 American Standard Code for Information Interchange (ASCII)