
Chapter 1

Digital Systems and Binary Numbers

Binary Numbers

11010.11 is 26.75.

$$\underline{1} \times 2^4 + \underline{1} \times 2^3 + \underline{0} \times 2^2 + \underline{1} \times 2^1 + \underline{0} \times 2^0 + \underline{1} \times 2^{-1} + \underline{1} \times 2^{-2}$$

Hexadecimal Numbers (A = 10, B = 11, C = 12, D = 13, E = 14, F = 15)

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

augend:	101101	minuend:	101101	multiplicand:	1011
addend:	<u>+100111</u>	subtrahend:	<u>-100111</u>	multiplier:	<u>× 101</u>
sum:	1010100	difference:	000110		1011
				partial product:	0000
					1011
				product:	<u>110111</u>

Table 1.1
Powers of Two

<i>n</i>	<i>2ⁿ</i>	<i>n</i>	<i>2ⁿ</i>	<i>n</i>	<i>2ⁿ</i>
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

Note that $2^{30} = 1,073,741,824$ (1G)

Number-base Conversions

	Integer Quotient		Remainder	Coefficient
$41/2 =$	20	+	$\frac{1}{2}$	$a_0 = 1$
$20/2 =$	10	+	0	$a_1 = 0$
$10/2 =$	5	+	0	$a_2 = 0$
$5/2 =$	2	+	$\frac{1}{2}$	$a_3 = 1$
$2/2 =$	1	+	0	$a_4 = 0$
$1/2 =$	0	+	$\frac{1}{2}$	$a_5 = 1$

Therefore, the answer is $(41)_{10} = (a_5a_4a_3a_2a_1a_0)_2 = (101001)_2$.

Number-base Conversions

เขียนแบบย่อๆ

Integer	Remainder
41	
20	1
10	0
5	0
2	1
1	0
0	1 101001 = answer

เขียนเลขฐาน 10 เป็นเลขฐาน 8

EXAMPLE 1.2

Convert decimal 153 to octal. The required base r is 8. First, 153 is divided by 8 to give an integer quotient of 19 and a remainder of 1. Then 19 is divided by 8 to give an integer quotient of 2 and a remainder of 3. Finally, 2 is divided by 8 to give a quotient of 0 and a remainder of 2. This process can be conveniently manipulated as follows:

$$\begin{array}{r|l} 153 & \\ 19 & 1 \\ 2 & 3 \\ 0 & 2 = (231)_8 \end{array}$$

The conversion of a decimal *fraction* to binary is accomplished by a method similar to that used for integers. However, multiplication is used instead of division, and integers instead of remainders are accumulated. Again, the method is best explained by example.



ส่วนที่เป็นทศนิยม

EXAMPLE 1.3

Convert $(0.6875)_{10}$ to binary. First, 0.6875 is multiplied by 2 to give an integer and a fraction. Then the new fraction is multiplied by 2 to give a new integer and a new fraction. The process is continued until the fraction becomes 0 or until the number of digits has sufficient accuracy. The coefficients of the binary number are obtained from the integers as follows:

เปลี่ยนหาร 2 เป็นคูณ 2	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	$a_{-4} = 1$

Therefore, the answer is $(0.6875)_{10} = (0. a_{-1} a_{-2} a_{-3} a_{-4})_2 = (0.1011)_2$.

Octal and Hexadecimal Numbers

Table 1.2
Numbers with Different Bases

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

ในสาขาคอมพิวเตอร์นิยมเขียนเลขฐาน 16
โดยเติม x หรือ 0x ข้างหน้า

เช่น 0x2FB5 หมายถึง $2FB5_{16}$

ແປລະຫຼາດ 2 ເປັນລະຫຼາດ 8

$$\begin{array}{ccccccccc} (10 & 110 & 001 & 101 & 011 & \cdot & 111 & 100 & 000 & 110)_2 & = & (26153.7406)_8 \\ 2 & 6 & 1 & 5 & 3 & & 7 & 4 & 0 & 6 \end{array}$$

ແປລະຫຼາດ 8 ເປັນລະຫຼາດ 2

$$\begin{array}{ccccccc} (673.124)_8 & = & (110 & 111 & 011 & \cdot & 001 & 010 & 100)_2 \\ & & 6 & 7 & 3 & & 1 & 2 & 4 \end{array}$$

ແປລະຫຼາດ 2 ເປັນລະຫຼາດ 16

$$10\ 1100\ 0110\ 1011\ .\ 1111\ 0000\ 0110 = (2C6B.F06)_{16}$$

2 C 6 B F 0 6

ແປລະຫຼາດ 16 ເປັນລະຫຼາດ 2

$$(1BB.2A)_{16} = 1\ 1011\ 1011\ .\ 0010\ 1010$$

1 B B 2 A

Complements of Numbers

Diminished Radix Complement

Given a number N in base r having n digits, the $(r - 1)$'s complement of N , i.e., its diminished radix complement, is defined as $(r^n - 1) - N$. For decimal numbers, $r = 10$ and $r - 1 = 9$, so the 9's complement of N is $(10^n - 1) - N$.

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

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

For binary numbers, $r = 2$ and $r - 1 = 1$, so the 1's complement of N is $(2^n - 1) - N$.

The 1's complement of 1011000 is 0100111.

The 1's complement of 0101101 is 1010010.

เปลี่ยน 0 เป็น 1
เปลี่ยน 1 เป็น 0

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 $N = 0$. Comparing with the $(r - 1)$'s complement, we note that the r 's complement is obtained by adding 1 to the $(r - 1)$'s complement, since $r^n - N = [(r^n - 1) - N] + 1$.

the 10's complement of 012398 is 987602

the 10's complement of 246700 is 753300

the 2's complement of 1101100 is 0010100

the 2's complement of 0110111 is 1001001

Subtraction with Complements

The subtraction of two n -digit unsigned numbers $M - N$ in base r can be done as follows:

1. Add the minuend M to the r 's complement of the subtrahend N . Mathematically, $M + (r^n - N) = M - N + r^n$.
2. If $M \geq N$, the sum will produce an end carry r^n , which can be discarded; what is left is the result $M - N$.
3. 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 $(N - M)$. To obtain the answer in a familiar form, take the r 's complement of the sum and place a negative sign in front.

EXAMPLE 1.5

Using 10's complement, subtract $72532 - 3250$.

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

Note that M has five digits and N has only four digits. Both numbers must have the same number of digits, so we write N as 03250. Taking the 10's complement of N produces a 9 in the most significant position. The occurrence of the end carry signifies that $M \geq N$ and that the result is therefore positive.



EXAMPLE 1.6

Using 10's complement, subtract $3250 - 72532$.

$$M = 03250$$

$$10\text{'s complement of } N = + 27468$$

$$\text{Sum} = 30718$$

There is no end carry. Therefore, the answer is $-(10\text{'s complement of } 30718) = -69282$.

Note that since $3250 < 72532$, the result is negative. Because we are dealing with unsigned numbers, there is really no way to get an unsigned result for this case. When subtracting with complements, we recognize the negative answer from the absence of the end carry and the complemented result. When working with paper and pencil, we can change the answer to a signed negative number in order to put it in a familiar form.

Subtraction with complements is done with binary numbers in a similar manner, using the procedure outlined previously.



EXAMPLE 1.7

84

67

Given the two binary numbers $X = 1010100$ and $Y = 1000011$, perform the subtraction **(a)** $X - Y$ and **(b)** $Y - X$ by using 2's complements.

$$\begin{aligned} \text{(a)} \quad & X = 1010100 \\ & 2\text{'s complement of } Y = + \underline{0111101} \\ & \text{Sum} = 10010001 \\ & \text{Discard end carry } 2^7 = - \underline{10000000} \\ & \text{Answer: } X - Y = 0010001 \end{aligned}$$

$$\begin{aligned} \text{(b)} \quad & Y = 1000011 \\ & 2\text{'s complement of } X = + \underline{0101100} \\ & \text{Sum} = 1101111 \end{aligned}$$

There is no end carry. Therefore, the answer is $Y - X = -(2\text{'s complement of } 1101111) = -0010001$.



Table 1.3

Signed Binary Numbers

ปกติเขียน 2'com ล้วน ๆ ก็คือ signed 2'complement (ไม่มี unsigned 2'complement)

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	—	—

บวกกันได้ทันที

ไม่ว่าเลขบวกหรือลบ
ใช้วงจรบวกเท่านั้น

ข้อเสีย

มี +0 กับ -0

ข้อเสีย

ออกแบบวงจรบวกยาก ต้อง
แปลงเลขลบเป็น 2's
complement ก่อน

ในภาษา C (limits.h)

INT_MIN -2,147,483,648

INT_MAX +2,147,483,647

ในระบบ 2's complement บวกกันได้ตรง ๆ เลย ไม่ว่าจะเป็นเลขบวกหรือลบ

$$+ 6 \quad 00000110$$

$$+13 \quad 00001101$$

$$+19 \quad 00010011$$

อ่านค่าแบบ **binary number** ได้เลย

0001 0011 = 19

$$+ 6 \quad 00000110$$

$$-13 \quad 11110011$$

$$- 7 \quad 11111001$$

$$- 6 \quad 11111010$$

$$+13 \quad 00001101$$

$$+ 7 \quad 00000111$$

อ่านค่าแบบ **binary number** ได้เลย

0000 0111 = 7

$$- 6 \quad 11111010$$

$$-13 \quad 11110011$$

$$-19 \quad 11101101$$

MSB = 1 แสดงว่าผลบวกมีค่าเป็นลบ ถ้าต้องการรู้ว่ามีค่าลบเท่าใด ให้หา **2's complement** เช่น

2's complement ของ **1111 1001** คือ **0000 0110 + 1 = 0000 0111** หรือ **7** ดังนั้นผลบวกคือ **-7**

2's complement ของ **1110 1101** คือ **0001 0010 + 1 = 0001 0011** หรือ **19** ดังนั้นผลบวกคือ **-19**

ตัวเลขที่เกิน **n** บิต จะถูกโยนทิ้งไป เพราะไม่มีที่เก็บ ไม่ต้องสนใจ **n** อาจจะเป็น **8, 16, 32, 64** ขึ้นกับจำนวนบิตของ **CPU** เช่น เป็น **CPU** แบบ **32** หรือ **64** บิต

ใช้วงจร
บวกเท่านั้น แปลง subtrahend ให้เป็น
2's complement ก่อน

$$\begin{aligned}(\pm A) - (+B) &= (\pm A) + (-B); \\(\pm A) - (-B) &= (\pm A) + (+B).\end{aligned}$$

Binary-Coded Decimal Code

Table 1.4
Binary-Coded Decimal (BCD)

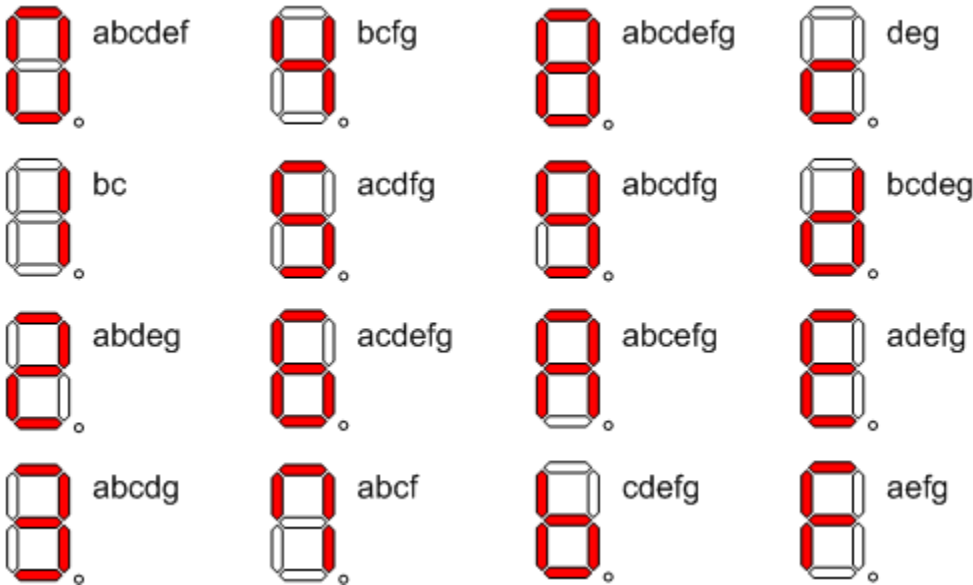
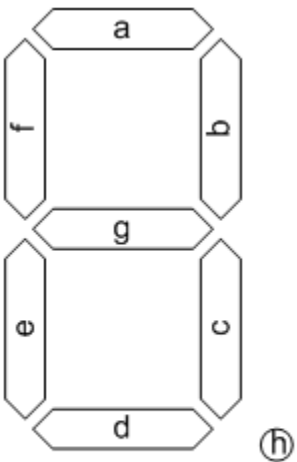
Decimal Symbol	BCD Digit
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

เช่น 365 เขียนแบบ BCD ได้ดังนี้

0011 0110 0101

3 6 5

แปลงจำนวนบิต $365_{10} = 1\ 0110\ 1101$ ใช้เนื้อที่แค่ 9 บิต แต่ BCD ใช้เนื้อที่ 12 บิต แต่ข้อเสียของ binary number คือแสดงผลเป็นเลขฐานสิบยาก



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
	7	6	0	

ถ้าผลบวกในหลักใดมากกว่าหรือเท่ากับ 1010 ให้บวก 0110 เพิ่มเข้าไปในหลักนั้น

Decimal Arithmetic

$$(+375) + (-240) = +135$$

$$\begin{array}{r} 0 \ 375 \\ +9 \ 760 \\ \hline 0 \ 135 \end{array}$$

มี 4 digits (ปกติเขียนติดกันไปเลย)
BCD 1 digit ใช้เนื้อที่ 4 บิต

ใช้ 10's complement แทนเลขลบ (คอมพิวเตอร์นิยมทำแบบนี้มากกว่าใช้ signed bit) Leading 9 บอกว่าเลขนั้นมีค่าเป็นลบ

คุณ กับ หาส เลข BCD ทำยังไง? หนังสือไม่ได้บอก

Other Decimal Code

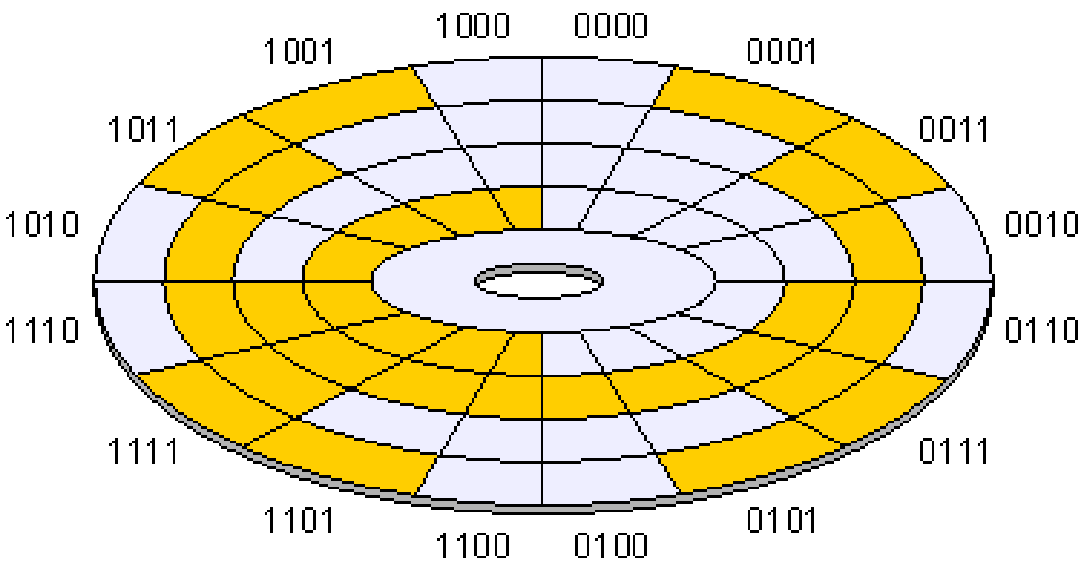
Table 1.5
Four Different Binary Codes for the Decimal Digits

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

Gray Code

Table 1.6
Gray Code

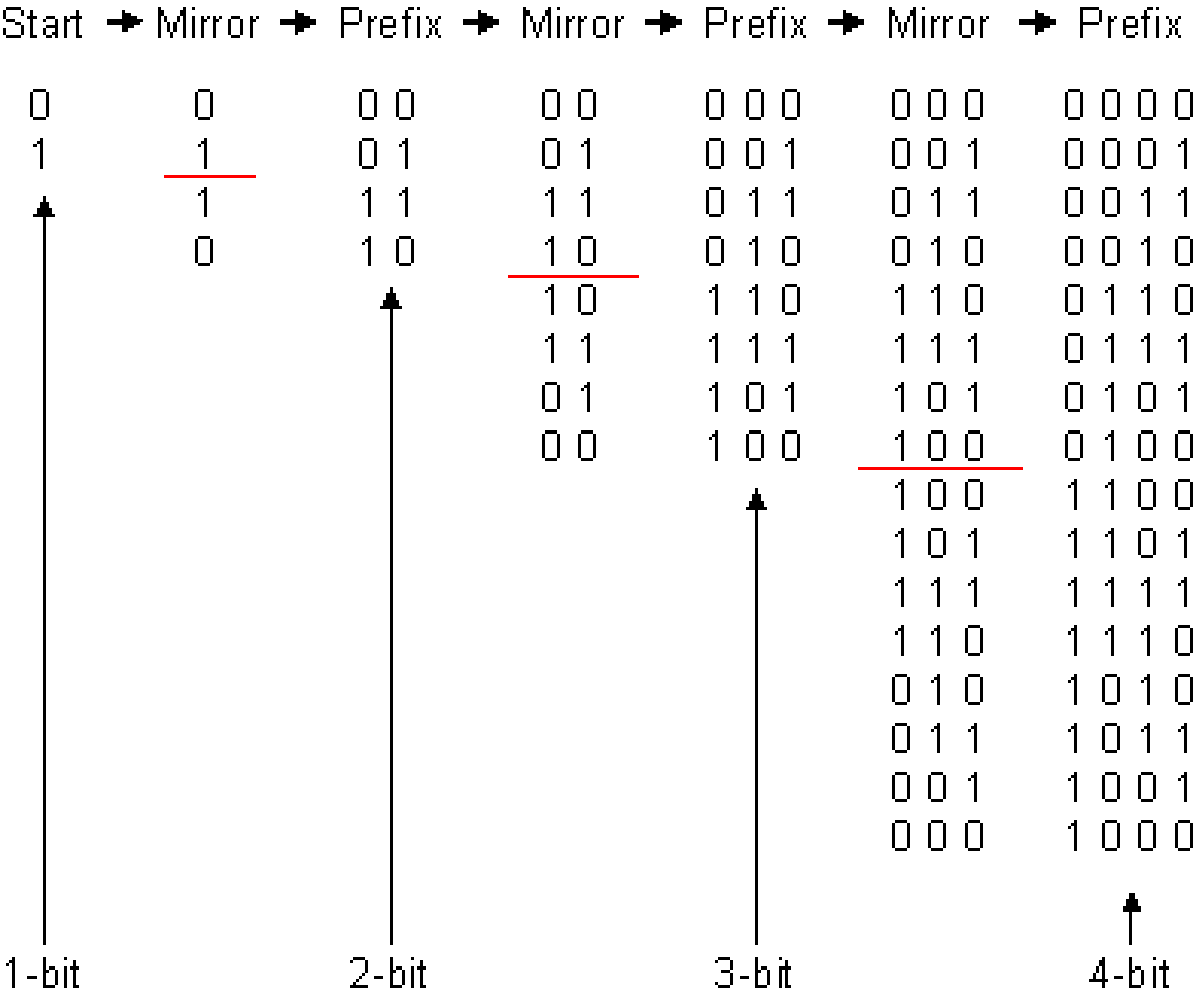
Gray Code	Decimal Equivalent
0000	0
0001	1
0011	2
0010	3
0110	4
0111	5
0101	6
0100	7
1100	8
1101	9
1111	10
1110	11
1010	12
1011	13
1001	14
1000	15



Gray code used for shaft-angle encoding

Using a Gray code sequence to define the conducting and non-conducting areas ensures that no intermediate values are generated as the shaft rotates.

Mirroring Technique



ASCII Character Code

Table 1.7

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

1 char = 7 bits

Control Characters

NUL	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



ขึ้นบรรทัดใหม่

Windows ใช้ CR + LF

Linux ใช้ LF

Unicode

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
0010	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
0020		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
0030	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0040	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0050	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0060	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0070	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

@  0x0040
A  0x0041
B  0x0042
... ..

<http://unicode-table.com>
<https://symbl.cc/th/unicode-table/>

1 char = 2 - 3 bytes

Unicode

Code point ของ "ก" คือ 0E 01 ถ้าใช้ UTF-8 "ก" จะได้ E0 B8 81

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0E00		ก	ข	ฃ	ค	ฅ	ฆ	ง	จ	ฉ	ช	ซ	ฌ	ญ	ฎ	ฏ
0E10	ฐ	ฑ	ฒ	ณ	ด	ต	ถ	ท	ธ	น	บ	ป	ผ	ฝ	พ	ฟ
0E20	ภ	ม	ย	ร	ฤ	ล	ฦ	ว	ศ	ษ	ส	ห	ฬ	อ	ฮ	า
0E30	ะ	ั	า	ำ	ิ	ี	ึ	ุ								฿
0E40	เ	แ	โ	ใ	ไ	า	า	๕	ิ	๖	๗	๘	๙	๐	๑	๒

มีครบเกือบทุกภาษาเขียนในโลก

สรุป
first - last Code point ↔ UTF-8 conversion

First code point	Last code point	Byte 1	Byte 2	Byte 3	Byte 4
U+0000	U+007F	0xxxxxxx			
U+0080	U+07FF	110xxxxx	10xxxxxx		
U+0800	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx	
U+10000	U+10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

Code point ของ "ก" คือ 0E 01 ถ้าใช้ UTF-8 "ก" จะได้ E0 B8 81

11100000 10111000 10000001
E0 B8 81

Error Detecting Code

จะไปเรียนในวิชาอื่นๆ
เช่น Oper Sys, Data Comm

Binary Logic

Table 1.8
Truth Tables of Logical Operations

AND			OR			NOT	
x	y	$x \cdot y$	x	y	$x + y$	x	x'
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

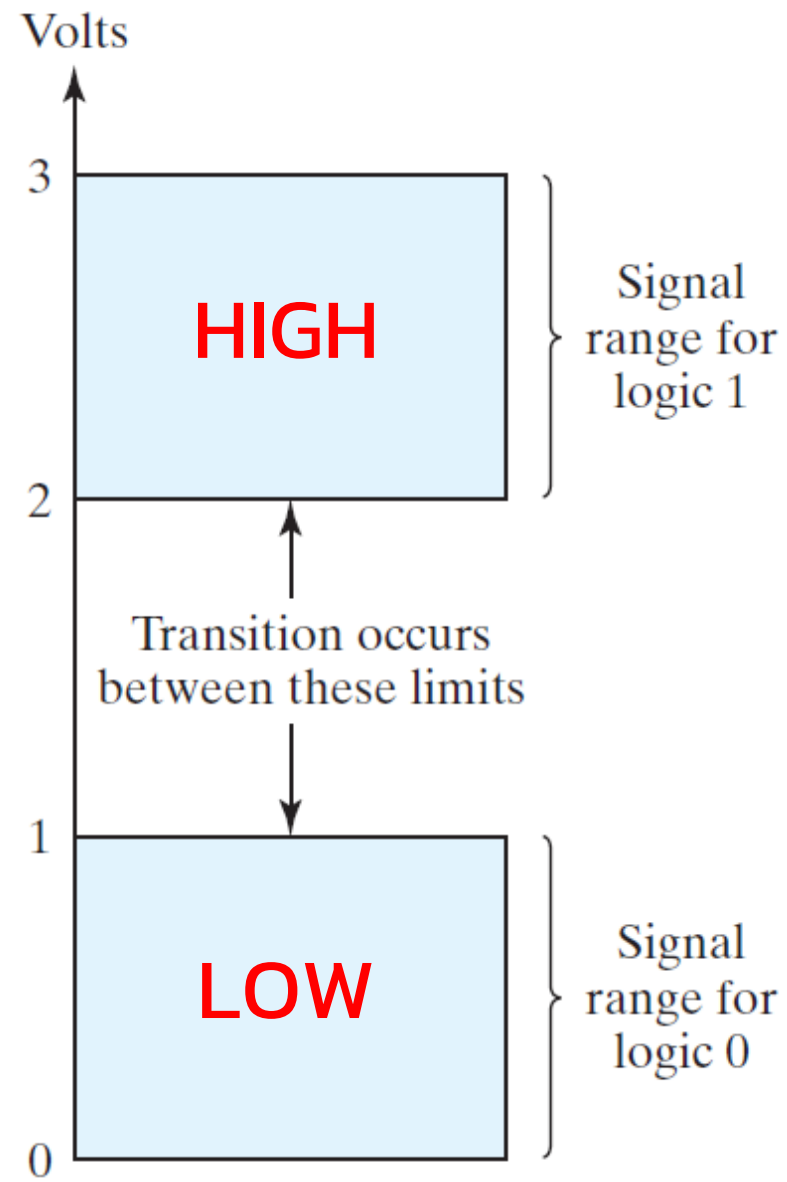
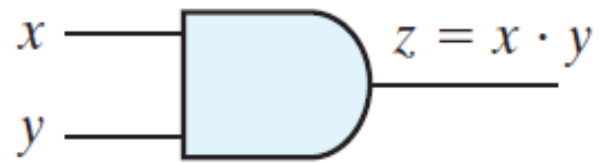
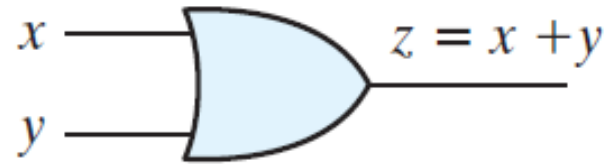


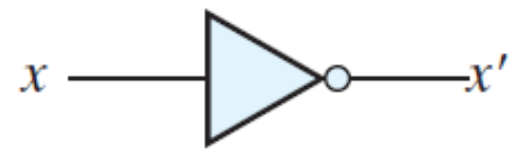
FIGURE 1.3
Signal levels for binary logic values



(a) Two-input AND gate



(b) Two-input OR gate



(c) NOT gate or inverter

FIGURE 1.4

Symbols for digital logic circuits

Timing Diagram แกนนอนคือเวลา

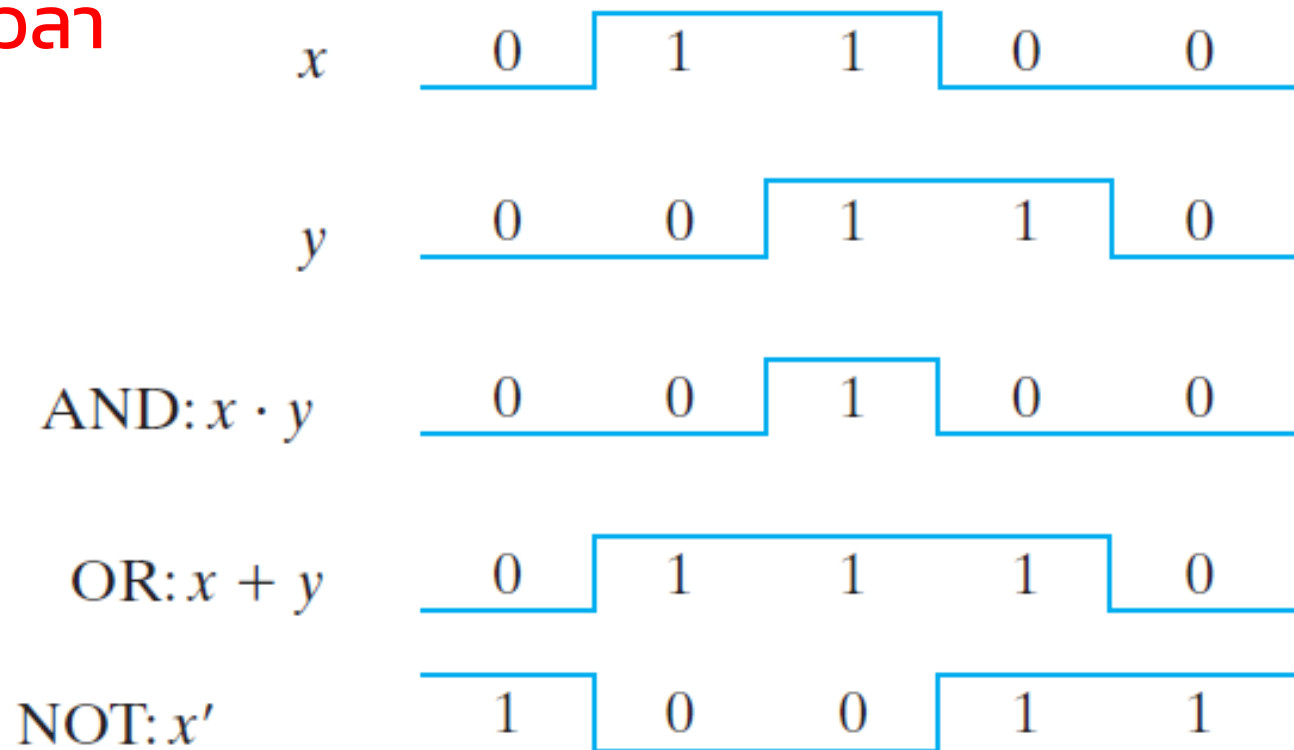
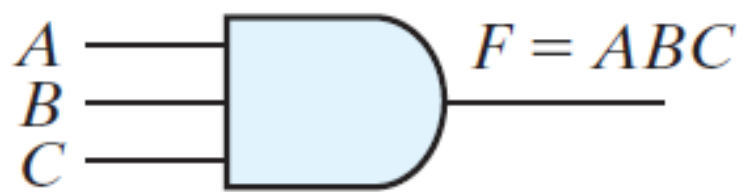
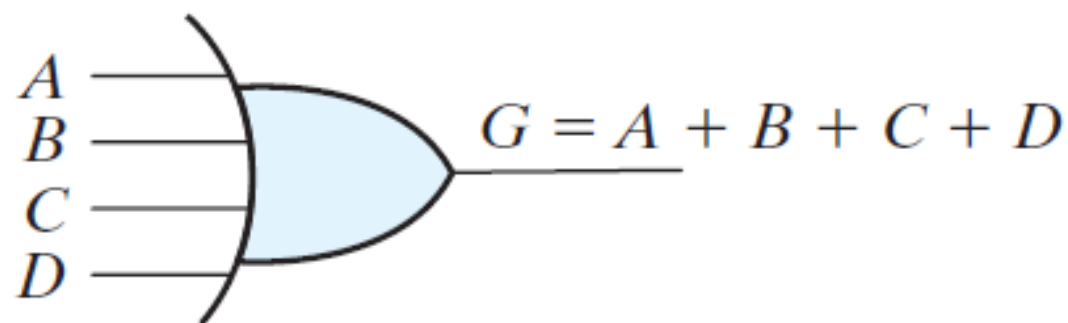


FIGURE 1.5

Input-output signals for gates



(a) Three-input AND gate



(b) Four-input OR gate

FIGURE 1.6

Gates with multiple inputs

การเก็บจำนวนจุดลอยตัวแบบ IEEE 754

เนื่องจากจำนวนจริง (real number) มีเป็นจำนวนไม่จำกัดและอาจจะมีทศนิยมที่ไม่รู้จบ เช่น ค่าพาย (π) เป็นต้น ดังนั้นจึงเป็นไปได้ว่าคอมพิวเตอร์จะสามารถเก็บจำนวนจริงใดใดก็ได้ในหน่วยความจำที่มีจำกัด แต่คอมพิวเตอร์สามารถเก็บจำนวนจุดลอยตัว (floating-point) ได้ จำนวนจุดลอยตัวเป็นการประมาณค่าจำนวนจริง โดยใช้ “แมนทิสซา (mantissa)” และ “เอกซ์โพเนน (exponent)” ที่มีจำนวนบิตจำกัด เช่น 1.2345 แทนได้ด้วย 12345×10^{-4} แมนทิสซาคือ 12345 และเอกซ์โพเนนคือ -4 IEEE 754 คือมาตรฐานในการเก็บจำนวนจุดลอยตัว (floating point) แบบ single precision ใช้ที่ขนาด 32 บิต (แบบ double precision ใช้ที่ 64 บิต) 32 บิตของ IEEE 754 แบ่งเป็น 3 ส่วนดังนี้

- MSB ขนาด 1 บิตเรียกว่าบิตเครื่องหมาย (sign bit)
- Unsigned 8 บิตถัดมาคือเอกซ์โพเนน (exponent)
ขออย่าว่าเอกซ์โพเนนเป็น unsigned int และเป็นบวกเสมอ
- 23 บิตสุดท้ายคือแมนทิสซา (mantissa)

การแปลงเป็นเลขฐานสิบให้ทำดังนี้

- สร้างจำนวนแบบจุดตรึงขึ้นมาก่อน โดยนำแมนทิสซา ไปต่อท้าย 1. เช่น สมมติว่าแมนทิสซาคือ **01101** จะได้ **1.01101**
- นำจำนวนแบบจุดตรึงที่ได้ไปคูณกับ $2^{(\text{exponent} - 127)}$ สำหรับเลขฐานสองการคูณสองคือการเลื่อน (shift) บิตทั้งหมดไปทางซ้าย 1 หลัก การหารสองคือการเลื่อนบิตทั้งหมดไปทางขวา 1 หลัก
- ถ้าบิตเครื่องหมายเป็น 0 ให้ใส่เครื่องหมาย + กรณีเป็น 1 ให้ใส่ -

ลองดูตัวอย่าง single precision 32 บิต ต่อไปนี้

0 01111100 010000000000000000000000

sign bit = 0, exponent = 124, mantissa = 1.01 ดังนั้นจำนวนนี้มีค่าเท่ากับ $+1.01 \times 2^{(124-127)}$ หรือเลื่อน 1.01

ไปทางขวา 3 บิตจะได้ $0.00101 = 2^{-3} + 2^{-5} = 0.15625$ ในฐานสิบ

การแปลงเลขฐานสิบกลับไปเป็น IEEE 754 (single precision) อาจจะยากกว่าเล็กน้อย ขั้นแรกให้เขียนจำนวนแบบจุดตรึงของเลขฐานสิบให้ได้ก่อน เช่น $(85 + 85/128)$ เขียนแบบจุดตรึงจะได้ 1010101.1010101 ขั้นที่สองจัดให้ส่วนที่เป็นจำนวนเต็มเป็น 1 จะได้ 1.0101011010101 ส่วนที่เป็นทศนิยม “0101011010101” คือแมนทิสซา ถ้าไม่ครบ 23 บิตก็เติมศูนย์เข้าไปทางขวาจนครบ ขั้นที่สามคำนวณเอกซ์โพเนนจากสมการ

$$1.0101011010101 \times 2^{(\text{exponent} - 127)} = 1010101.1010101$$

จะได้เอกซ์โพเนนเท่ากับ 133 หรือ 10000101 (ถ้าไม่ครบ 8 บิตก็เติมศูนย์เข้าไปทาง MSB) ขั้นสุดท้ายเติมบิตสัญลักษณ์ให้ถูกต้อง ก็จะได้จำนวนตามมาตรฐาน IEEE754 (single precision) 32 บิตที่มี sign bit, exponent และ mantissa ดังนี้

0 10000101 010101101010100000000000

ปกติเราจะเขียนเป็นเลขฐานสิบหก เพราะเขียนเป็นเลขฐานสองมันยาว ก็ทำการแปลงทีละ 4 บิต

0100 0010 1010 1011 0101 0100 0000 0000

4 2 A B 5 4 0 0

IEEE 754 แบบ double precision ใช้ที่ 64 บิตก็ใช้หลักการเดียวกัน เพียงแต่เอกซ์โพเนนและแมนทิสซายาวขึ้น

IEEE 754 เป็นมาตรฐานของตัวแปร float และ double ในหลายๆ ภาษาโปรแกรม

สุดท้ายที่กล่าวมาทั้งหมดเกี่ยวกับ IEEE 754 มีข้อยกเว้น เมื่อเอกซ์โพเนนและแมนทิสซามีค่าดังต่อไปนี้

<u>Exponent</u>	<u>Mantissa</u>	<u>ความหมาย</u>
0	0	ค่าศูนย์
2^8-1 (11111111)	0	Infinity
2^8-1 (11111111)	non zero	NaN (not a number)
0	non zero	Denormalized number
1 to 2^8-2	any	Normalized number

Infinity	คือจำนวนที่มีคุณสมบัติเหมือน $\pm\infty$ ในคณิตศาสตร์
NaN	คือ not a number เช่น จำนวนที่หารด้วยศูนย์ เป็นต้น
Denormalized number	คือเวลาอ่าน <u>ไม่ต้องมี 1</u> . นำหน้าแมนทิสซา ให้เติมจุดทศนิยมหลังบิตแรก (MSB) ของแมนทิสซาเลย เช่น ถ้าแมนทิสซาคือ <u>01101000101</u> 000000000000 ให้เอา <u>0.1101000101</u> คูณกับ $2^{(\text{exponent} - 127)}$ คูณกับ $(-1)^{\text{sign bit}}$
Normalized number	คือ เวลาอ่านค่าต้องมี 1. นำหน้าแมนทิสซา เหมือนที่อธิบายไว้ในตอนแรก

IEEE 754 Converter (JavaScript), V0.22

	Sign	Exponent								Mantissa																						
Value:	+1	2^{53}								1.1102230548858643																						
Encoded as:	0	180								924618																						
Binary:	☐	☒	☐	☒	☒	☐	☒	☐	☐	☐	☐	☒	☒	☒	☐	☐	☐	☒	☒	☐	☒	☒	☒	☒	☐	☐	☒	☐	☒	☐		
You entered	10000000000000000 10<sup>16																															
Value actually stored in float:	10000000272564224 ห้ามใช้ float/double เก็บจำนวนเงิน																															
Error due to conversion:	272564224																															
Binary Representation	01011010000011100001101111001010																															
Hexadecimal Representation	0x5a0e1bca																															

IEEE 754 Converter (JavaScript), V0.22

	Sign	Exponent								Mantissa																						
Value:	+1	2^{-34}								1.7179869413375854																						
Encoded as:	0	93								6022911																						
Binary:	☐	☐	☒	☐	☒	☒	☒	☐	☒	☒	☐	☒	☒	☒	☒	☒	☒	☐	☐	☒	☒	☐	☒	☒	☒	☒	☒	☒	☒	☒		
You entered	1E-10																															
Value actually stored in float:	1.00000001335143196001808973960578441619873046875E-10																															
Error due to conversion:	1.335143196001808973960578442E-18																															
Binary Representation	00101110110110111110011011111111																															
Hexadecimal Representation	0x2edbe6ff																															

IEEE 754 เก็บจำนวนที่มีค่ามาก ๆ ได้ โดยยอมให้คลาดเคลื่อนได้ <https://www.h-schmidt.net/FloatConverter/IEEE754.html>

PROBLEMS

(Answers to problems marked with * appear at the end of the text.)

- 1.1** List the octal and hexadecimal numbers from 16 to 32. Using A and B for the last two digits, list the numbers from 8 to 28 in base 12.
- 1.2*** What is the exact number of bytes in a system that contains (a) 32K bytes, (b) 64M bytes, and (c) 6.4G bytes?
- 1.3** Convert the following numbers with the indicated bases to decimal:
- | | |
|-----------------|-------------------|
| (a)* $(4310)_5$ | (b)* $(198)_{12}$ |
| (c) $(435)_8$ | (d) $(345)_6$ |
- 1.4** What is the largest binary number that can be expressed with 16 bits? What are the equivalent decimal and hexadecimal numbers?
- 1.5*** Determine the base of the numbers in each case for the following operations to be correct:
- | | | |
|----------------|-----------------|----------------------|
| (a) $14/2 = 5$ | (b) $54/4 = 13$ | (c) $24 + 17 = 40$. |
|----------------|-----------------|----------------------|
- 1.6*** The solutions to the quadratic equation $x^2 - 11x + 22 = 0$ are $x = 3$ and $x = 6$. What is the base of the numbers?

จงเขียนจำนวนต่อไปนี้ตามมาตรฐาน IEEE754 (single precision) เขียนตอบเป็นเลขฐานสิบหก

ก. 0

ข. $777 + (777/1024)$

ค. 3.14

ง. +infinity

จ. NaN

ฉ. $2^5 + 2^2 + 2^1 + 2^{-2} + 2^{-3} + 2^{-10}$

ช. -infinity

ซ. $-78 - (78/128)$