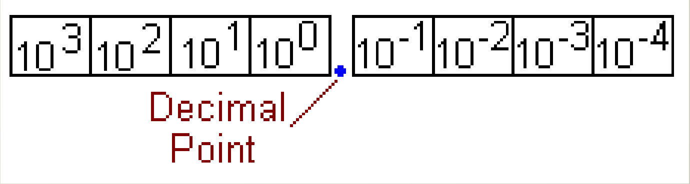
Number Systems, Operations, and Codes

Decimal Numbers

- The decimal number system has ten digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9
- The decimal numbering system has a base of 10 with each position weighted by a factor of 10:



Binary Numbers

- The binary number system has two digits: 0 and 1
- The binary numbering system has a base of 2 with each position weighted by a factor of 2:

POSITIVE POWERS OF TWO (WHOLE NUMBERS)									NEGATIVE POWERS OF TWO (FRACTIONAL NUMBER)					
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
256	128	64	32	16	8	4	2	1	1/2	1/4	1/8	1/16	1/32	1/64
									0.5	0.25	0.125	0.0625	0.03125	0.015625

Decimal-to-Binary Conversion

- Sum-of-weights method

Convert the following decimal numbers to binary:

(a) 12 **(b)** 25 **(c)** 58 **(d)** 82

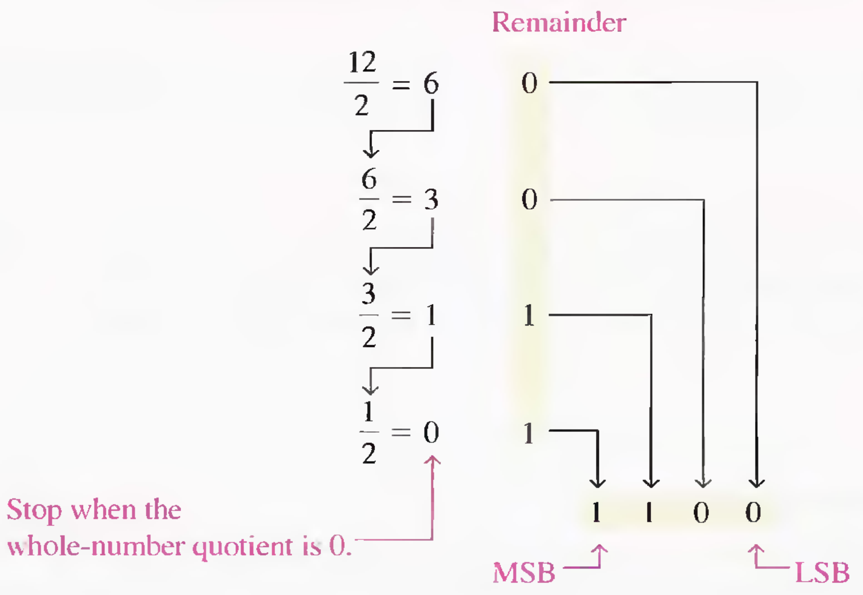
(a) $12 = 8 + 4 = 2^3 + 2^2 \longrightarrow$ **1100**

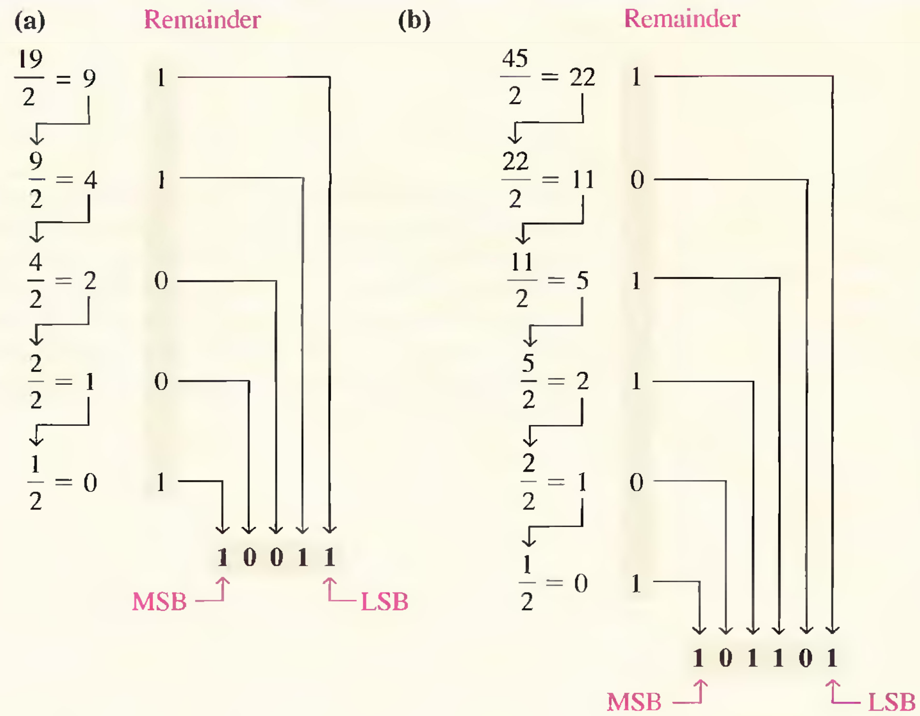
(b) $25 = 16 + 8 + 1 = 2^4 + 2^3 + 2^0 \longrightarrow$ **11001**

(c) $58 = 32 + 16 + 8 + 2 = 2^5 + 2^4 + 2^3 + 2^1 \longrightarrow$ **111010**

(d) $82 = 64 + 16 + 2 = 2^6 + 2^4 + 2^1 \longrightarrow$ **1010010**

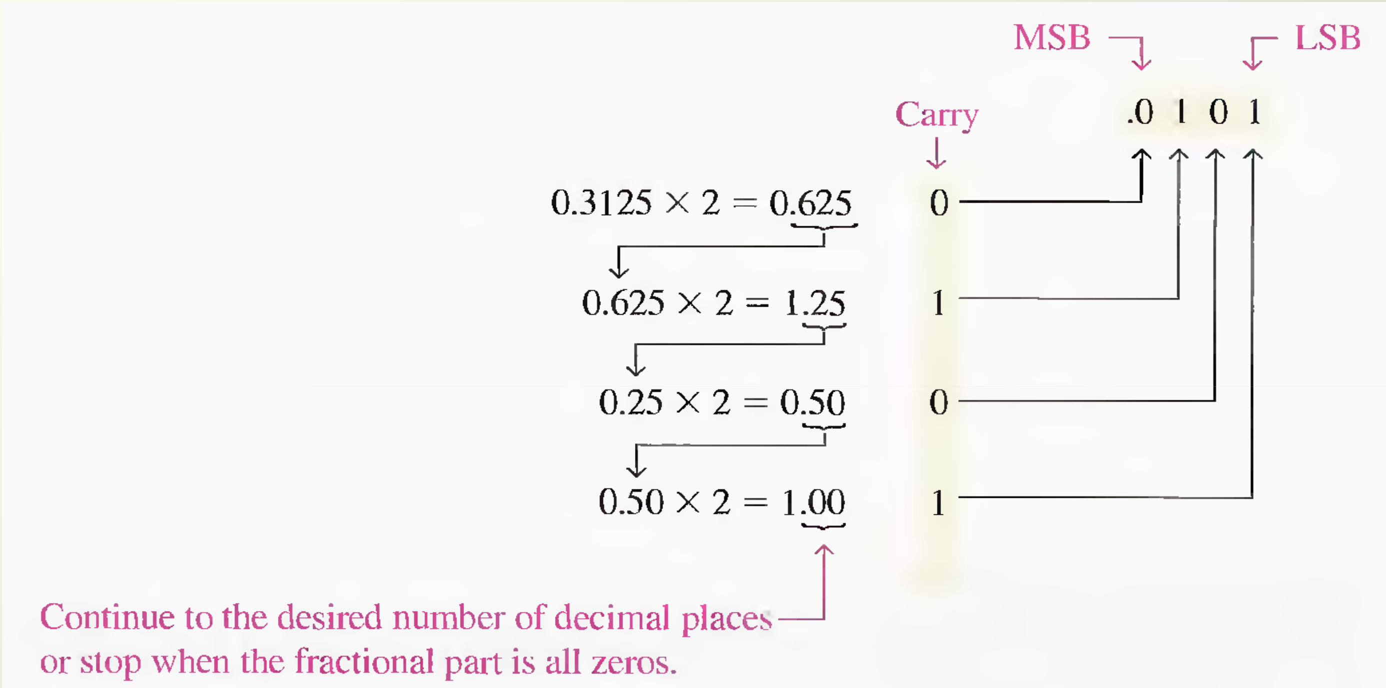
- Repeated division-by-2 method





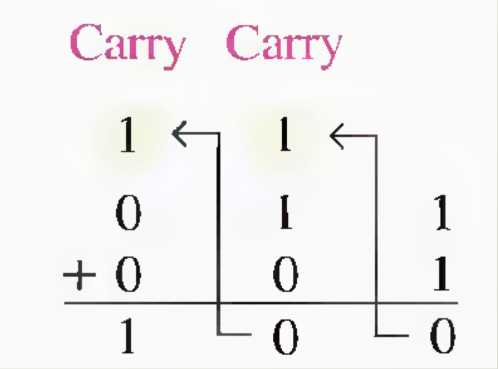
Decimal-to-Binary Conversion

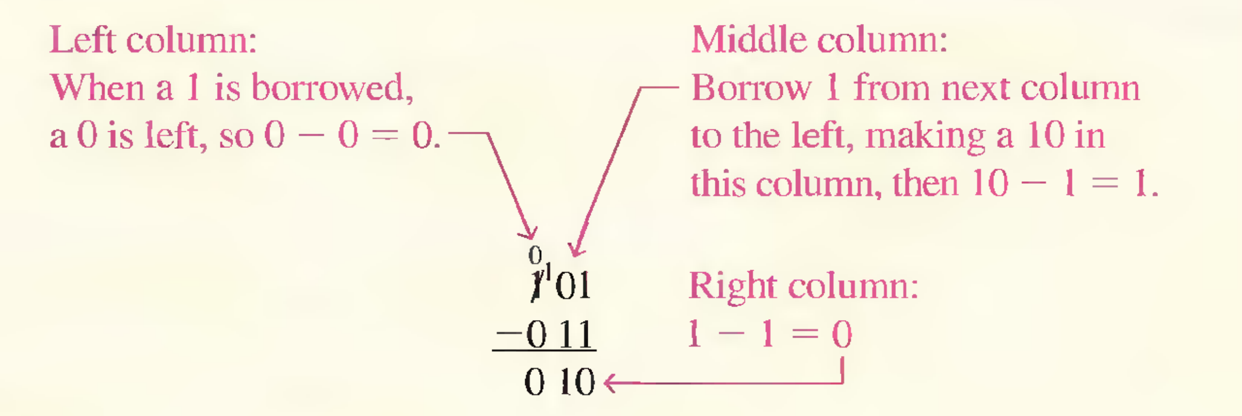
Conversion of decimal fractions to binary



Binary Arithmetic

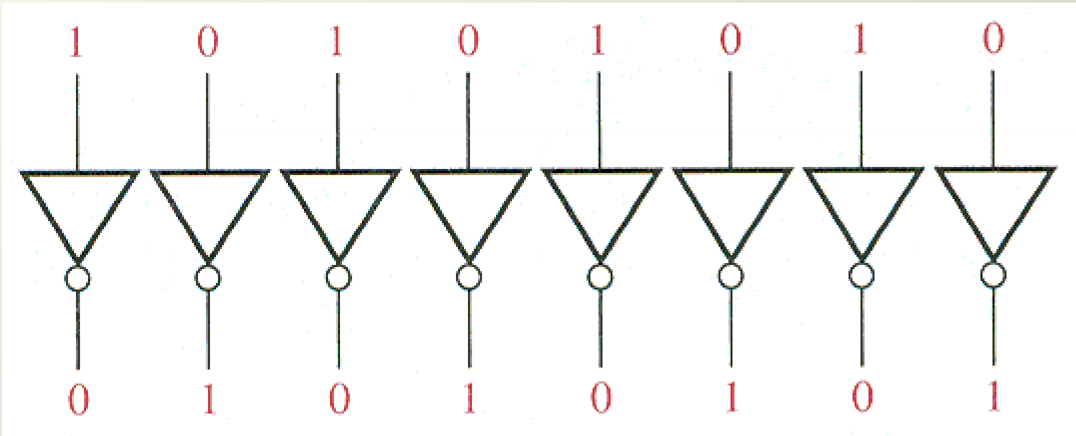
- Binary addition
- Binary subtraction
- Binary multiplication
- Binary division





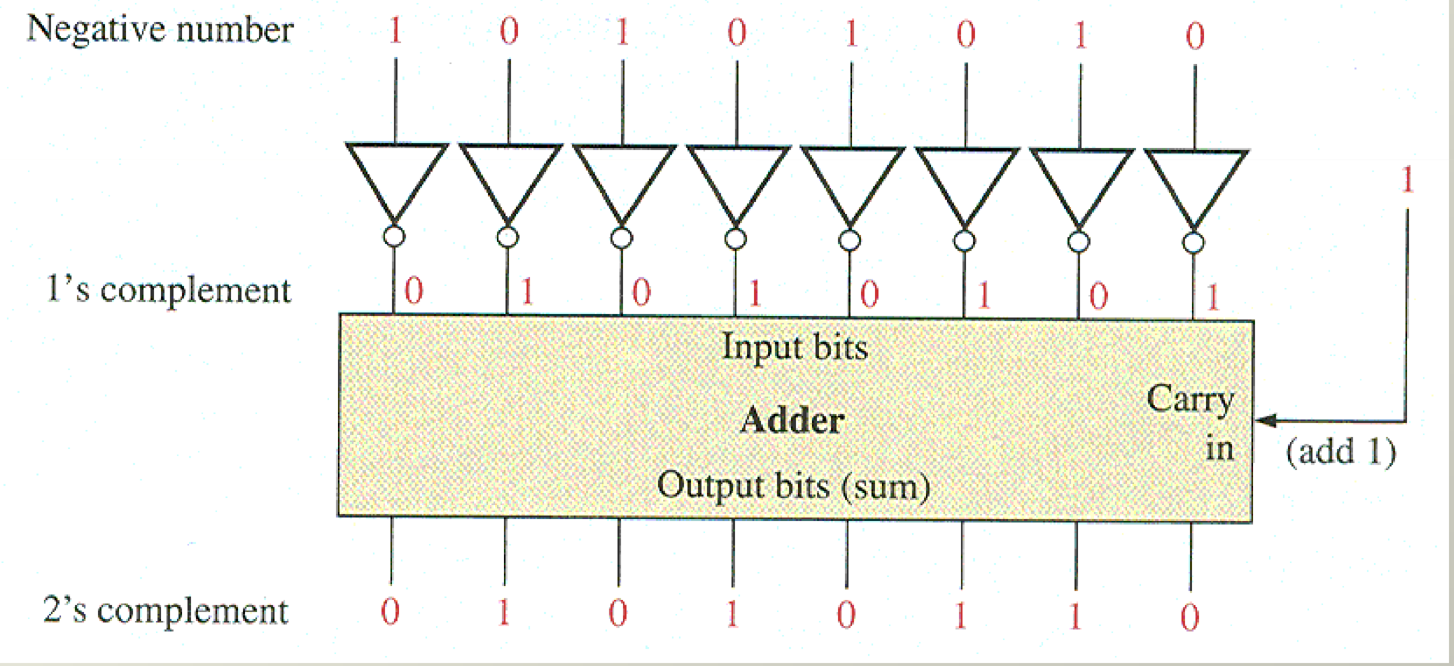
Complements of Binary Numbers

- 1's complement



Complements of Binary Numbers

- 2’s complement

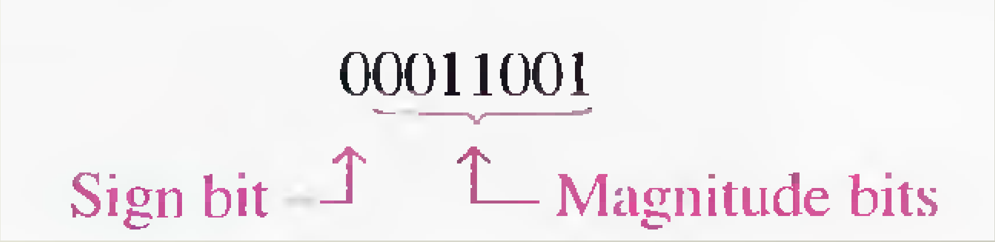


Signed Numbers

- Signed-magnitude form
- 1's and 2's complement form
- Decimal value of signed numbers
- Range of values
- Floating-point numbers

Signed Numbers

- Signed-magnitude form
  - The sign bit is the left-most bit in a signed binary number
  - A 0 sign bit indicates a positive magnitude
  - A 1 sign bit indicates a negative magnitude



Signed Numbers

- 1's complement form
  - A negative value is the 1's complement of the corresponding positive value
- 2's complement form
  - A negative value is the 2's complement of the corresponding positive value

Signed Numbers

- Decimal value of signed numbers
  - Sign-magnitude
  - 1's complement
  - 2's complement
- Example : 11001010
  - Sign-mag -> -74
  - 1's -> (-00110101) -> -53
  - 2's -> (-00110110) -> -54

Signed Numbers

- Range of Values

  Sign-magnitude:

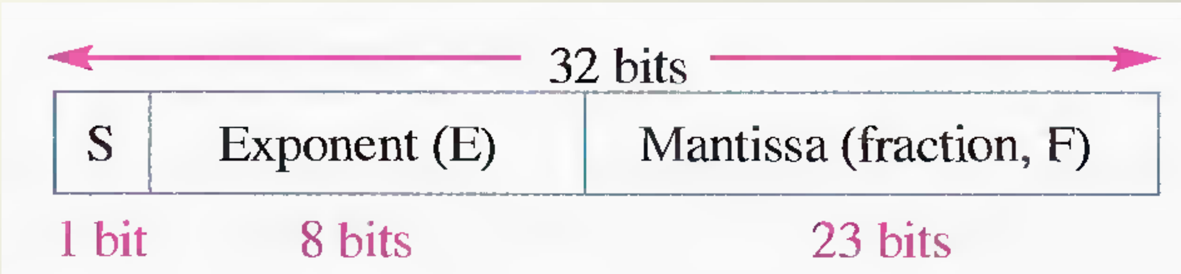
  $$-(2^{n-1}-1) \text{ to } +(2^{n-1}-1)$$

  2's complement form:

  $$-(2^{n-1}) \text{ to } +(2^{n-1}-1)$$

Signed Numbers

- Floating-point numbers
  - Single-precision (32 bits)
  - Double-precision (64 bits)
  - Extended-precision (80 bits)



$$\text{Number} = (-1)^S(1 + F)(2^{E-127})$$

Floating-Point Number Example

Convert the decimal number 3.248×10^4 to a single-precision floating-point binary number.

Solution Convert the decimal number to binary.

$$3.248 \times 10^4 = 32480 = 111111011100000_2 = 1.11111011100000 \times 2^{14}$$

The MSB will not occupy a bit position because it is always a 1. Therefore, the mantissa is the fractional 23-bit binary number 11111011100000000000000 and the biased exponent is

$$14 + 127 = 141 = 10001101_2$$

The complete floating-point number is

0	**10001101**	**11111011100000000000000**

Arithmetic Operations with Signed Numbers

- Addition
- Subtraction
- Multiplication
- Division

Arithmetic Operations with Signed Numbers

Addition of Signed Numbers

- The parts of an addition function are:
  - Addend
  - Augend
  - Sum

	00001111	15
	+ 11111010	+ −6
Discard carry → 1	00001001	9

00000111	7
+ 00000100	+ 4
00001011	11

	11111011	−5
	+ 11110111	+ −9
Discard carry → 1	11110010	−14

00010000	16
+ 11101000	+ −24
11111000	−8

01111101	125
+ 00111010	+ 58
10110111	183

Sign incorrect

Magnitude incorrect

Arithmetic Operations with Signed Numbers

Four conditions for adding numbers:

- Both numbers are positive.
- A positive number that is larger than a negative number.
- A negative number that is larger than a positive number.
- Both numbers are negative.

Arithmetic Operations with Signed Numbers

Signs for Addition

- When both numbers are positive, the sum is positive.
- When the larger number is positive and the smaller is negative, the sum is positive. The carry is discarded.

Arithmetic Operations with Signed Numbers

Signs for Addition

- When the larger number is negative and the smaller is positive, the sum is negative (2's complement form).
- When both numbers are negative, the sum is negative (2's complement form). The carry bit is discarded.

Arithmetic Operations with Signed Numbers

Subtraction of Signed Numbers

- The parts of a subtraction function are:
  - Minuend
  - Subtrahend
  - Difference

Subtraction is addition with the sign of the subtrahend changed.

	00001000	Minuend (+8)
	+ 11111101	2’s complement of subtrahend (−3)
Discard carry →	**1 00000101**	Difference (+5)

Arithmetic Operations with Signed Numbers

Subtraction

- The sign of a positive or negative binary number is changed by taking its 2's complement
- To subtract two signed numbers, take the 2's complement of the subtrahend and add. Discard any final carry bit.

Arithmetic Operations with Signed Numbers

Multiplication of Signed Numbers

- The parts of a multiplication function are:
  - Multiplicand
  - Multiplier
  - Product

Multiplication is equivalent to adding a number to itself a number of times equal to the multiplier.

Arithmetic Operations with Signed Numbers

There are two methods for multiplication:

- Direct addition
- Partial products

The method of partial products is the most commonly used.

Arithmetic Operations with Signed Numbers

Multiplication of Signed Numbers

- If the signs are the same, the product is positive.
- If the signs are different, the product is negative.

Multiply the signed binary numbers: 01010011 (multiplicand) and 11000101 (multiplier).

Step 1: The sign bit of the multiplicand is 0 and the sign bit of the multiplier is 1. The sign bit of the product will be 1 (negative).

Step 2: Take the 2's complement of the multiplier to put it in true form.

$$11000101 \longrightarrow 00111011$$

Steps 3 and 4: The multiplication proceeds as follows. Notice that only the magnitude bits are used in these steps.

1010011	Multiplicand
× 0111011	Multiplier
1010011	1st partial product
+ 1010011	2nd partial product
11111001	Sum of 1st and 2nd
+ 0000000	3rd partial product
011111001	Sum
+ 1010011	4th partial product
1110010001	Sum
+ 1010011	5th partial product
100011000001	Sum
+ 1010011	6th partial product
1001100100001	Sum
+ 0000000	7th partial product
1001100100001	Final product

Step 5: Since the sign of the product is a 1 as determined in step 1, take the 2's complement of the product.

$$1001100100001 \longrightarrow 0110011011111$$

Attach the sign bit

→ **1 0110011011111**

Arithmetic Operations with Signed Numbers

Division of Signed Numbers

- The parts of a division operation are:
  - Dividend
  - Divisor
  - Quotient

Division is equivalent to subtracting the divisor from the dividend a number of times equal to the quotient.

Arithmetic Operations with Signed Numbers

Division of Signed Numbers

- If the signs are the same, the quotient is positive.
- If the signs are different, the quotient is negative.

Hexadecimal Numbers

- Decimal, binary, and hexadecimal numbers

DECIMAL	BINARY	HEXADECIMAL
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Hexadecimal Numbers

- Binary-to-hexadecimal conversion
  1. Break the binary number into 4-bit groups
  2. Replace each group with the hexadecimal equivalent

Hexadecimal Numbers

- Hexadecimal-to-decimal conversion
  1. Convert the hexadecimal to groups of 4-bit binary
  2. Convert the binary to decimal
- Decimal-to-hexadecimal conversion
  – Repeated division by 16

Binary Coded Decimal (BCD)

Decimal and BCD digits

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

Convert each of the following decimal numbers to BCD:

(a) 35 (b) 98 (c) 170 (d) 2469

Solution

(a) $\underbrace{0011}_{3}\underbrace{0101}_{5}$ = **00110101**

(b) $\underbrace{1001}_{9}\underbrace{1000}_{8}$ = **10011000**

(c) $\underbrace{0001}_{1}\underbrace{0111}_{7}\underbrace{0000}_{0}$ = **000101110000**

(d) $\underbrace{0010}_{2}\underbrace{0100}_{4}\underbrace{0110}_{6}\underbrace{1001}_{9}$ = **0010010001101001**

Digital Codes

- Gray code

DECIMAL	BINARY	GRAY CODE
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001

Binary-to-gray

1. The most significant bit (left-most) in the Gray code is the same as the corresponding MSB in the binary number.
2. Going from left to right, add each adjacent pair of binary code bits to get the next Gray code bit. Discard carries.

For example, the conversion of the binary number 10110 to Gray code is as follows:

1	— + →	0	— + →	1	— + →	1	— + →	0	Binary
↓		↓		↓		↓		↓	
1		1		1		0		1	Gray

The Gray code is 11101.

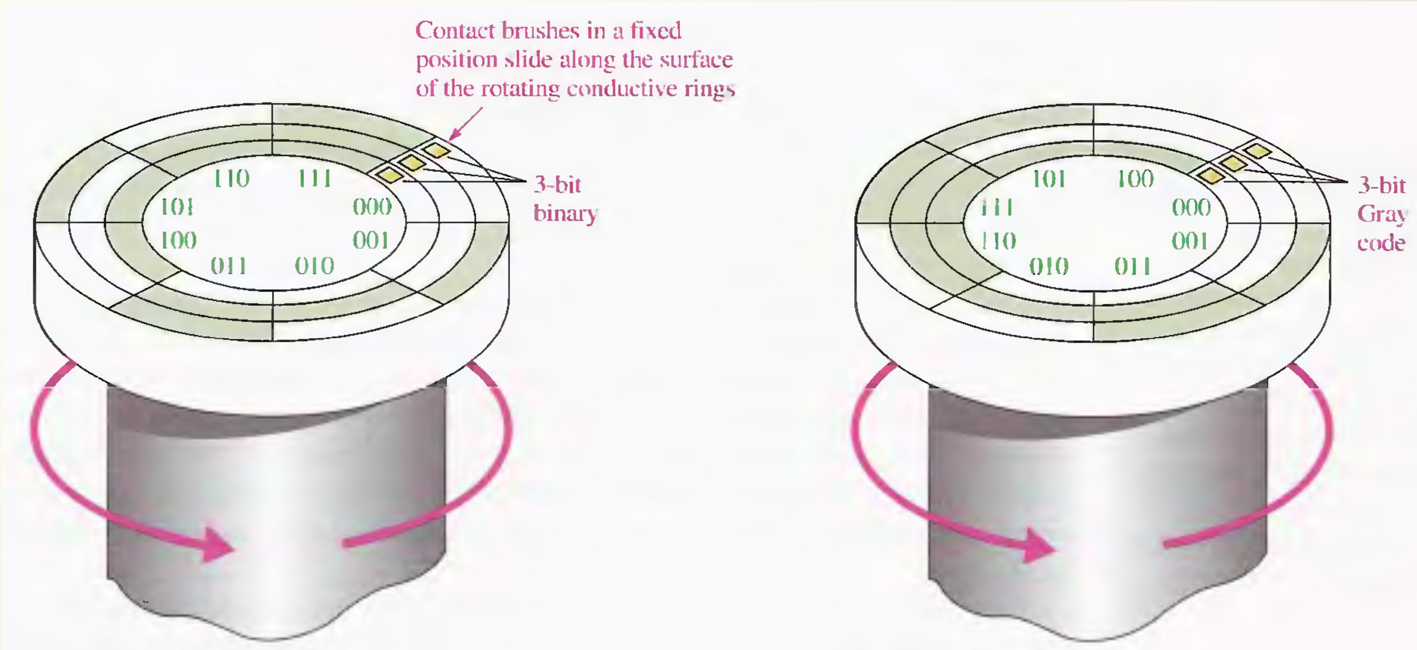
gray-to-Binary

1. The most significant bit (left-most) in the binary code is the same as the corresponding bit in the Gray code.
2. Add each binary code bit generated to the Gray code bit in the next adjacent position. Discard carries.

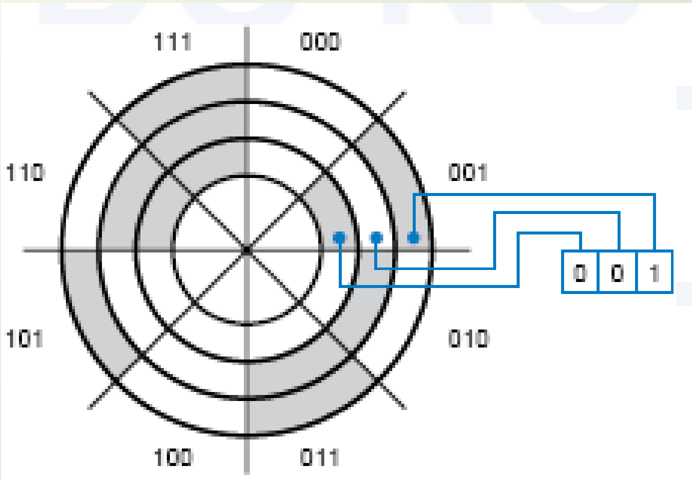
For example, the conversion of the Gray code word 11011 to binary is as follows:

1		1		0		1		1	Gray
↓	+ ↗	↓	+ ↗	↓	+ ↗	↓	+ ↗	↓	
1		0		0		1		0	Binary

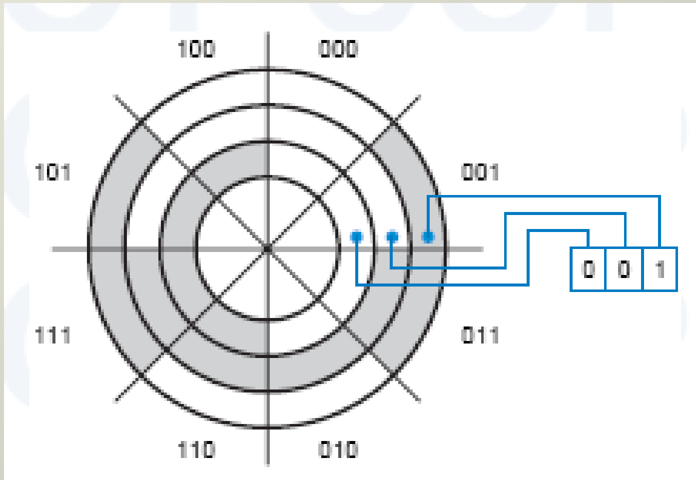
The binary number is 10010.



(a) Binary (b) Gray code



Binary



Gray

*Wakerly

Digital Codes

- ASCII code (control characters)

NAME	DEC	BINARY	HEX	NAME	DEC	BINARY	HEX
NUL	0	0000000	00	DLE	16	0010000	10
SOH	1	0000001	01	DC1	17	0010001	11
STX	2	0000010	02	DC2	18	0010010	12
ETX	3	0000011	03	DC3	19	0010011	13
EOT	4	0000100	04	DC4	20	0010100	14
ENQ	5	0000101	05	NAK	21	0010101	15
ACK	6	0000110	06	SYN	22	0010110	16
BEL	7	0000111	07	ETB	23	0010111	17
BS	8	0001000	08	CAN	24	0011000	18
HT	9	0001001	09	EM	25	0011001	19
LF	10	0001010	0A	SUB	26	0011010	1A
VT	11	0001011	0B	ESC	27	0011011	1B
FF	12	0001100	0C	FS	28	0011100	1C
CR	13	0001101	0D	GS	29	0011101	1D
SO	14	0001110	0E	RS	30	0011110	1E
SI	15	0001111	0F	US	31	0011111	1F

Digital Codes

- ASCII code (graphic symbols 20h – 3Fh)

SYMBOL	DEC	BINARY	HEX	SYMBOL	DEC	BINARY	HEX
space	32	0100000	20	0	48	0110000	30
!	33	0100001	21	1	49	0110001	31
"	34	0100010	22	2	50	0110010	32
#	35	0100011	23	3	51	0110011	33
$	36	0100100	24	4	52	0110100	34
%	37	0100101	25	5	53	0110101	35
&	38	0100110	26	6	54	0110110	36
'	39	0100111	27	7	55	0110111	37
(	40	0101000	28	8	56	0111000	38
)	41	0101001	29	9	57	0111001	39
*	42	0101010	2A	:	58	0111010	3A
+	43	0101011	2B	;	59	0111011	3B
,	44	0101100	2C	<	60	0111100	3C
—	45	0101101	2D	=	61	0111101	3D
.	46	0101110	2E	>	62	0111110	3E
/	47	0101111	2F	?	63	0111111	3F

Digital Codes

- ASCII code (graphic symbols 40h – 5Fh)

SYMBOL	DEC	BINARY	HEX	SYMBOL	DEC	BINARY	HEX
@	64	1000000	40	P	80	1010000	50
A	65	1000001	41	Q	81	1010001	51
B	66	1000010	42	R	82	1010010	52
C	67	1000011	43	S	83	1010011	53
D	68	1000100	44	T	84	1010100	54
E	69	1000101	45	U	85	1010101	55
F	70	1000110	46	V	86	1010110	56
G	71	1000111	47	W	87	1010111	57
H	72	1001000	48	X	88	1011000	58
I	73	1001001	49	Y	89	1011001	59
J	74	1001010	4A	Z	90	1011010	5A
K	75	1001011	4B	[	91	1011011	5B
L	76	1001100	4C	\	92	1011100	5C
M	77	1001101	4D	]	93	1011101	5D
N	78	1001110	4E	^	94	1011110	5E
O	79	1001111	4F	_	95	1011111	5F

Digital Codes

- ASCII code (graphic symbols 60h – 7Fh)

SYMBOL	DEC	BINARY	HEX	SYMBOL	DEC	BINARY	HEX
`	96	1100000	60	p	112	1110000	70
a	97	1100001	61	q	113	1110001	71
b	98	1100010	62	r	114	1110010	72
c	99	1100011	63	s	115	1110011	73
d	100	1100100	64	t	116	1110100	74
e	101	1100101	65	u	117	1110101	75
f	102	1100110	66	v	118	1110110	76
g	103	1100111	67	w	119	1110111	77
h	104	1101000	68	x	120	1111000	78
i	105	1101001	69	y	121	1111001	79
j	106	1101010	6A	z	122	1111010	7A
k	107	1101011	6B	{	123	1111011	7B
l	108	1101100	6C	|	124	1111100	7C
m	109	1101101	6D	}	125	1111101	7D
n	110	1101110	6E	~	126	1111110	7E
o	111	1101111	6F	Del	127	1111111	7F

Digital Codes

Extended ASCII code (80h – FFh)

- Non-English alphabetic characters
- Currency symbols
- Greek letters
- Math symbols
- Drawing characters
- Bar graphing characters
- Shading characters

Error Detection and Correction Codes

- Parity error codes

EVEN PARITY		ODD PARITY	
P	BCD	*P*	BCD
0	0000	1	0000
1	0001	0	0001
1	0010	0	0010
0	0011	1	0011
1	0100	0	0100
0	0101	1	0101
0	0110	1	0110
1	0111	0	0111
1	1000	0	1000
0	1001	1	1001

Error Detection and Correction Codes

- Hamming error codes
  - Hamming code words
  - Hex equivalent of the data bits
  - Example (7,4) Hamming
    - 4-bit Message $x_3x_2x_1x_0$
    - 3-bit Parity-check $c_2c_1c_0$
    - Hamming code = message + parity-check $(x_3x_2x_1x_0c_2c_1c_0)$
    - Here, $c_0 = x_2 + x_1 + x_0$,

    $c_1 = x_3 + x_2 + x_1$, $c_2 = x_3 + x_1 + x_0$

    ("+" here mean exclusive-or)

0000000	0
0001101	1
0010111	2
0011010	3
0100011	4
0101110	5
0110100	6
0111001	7
1000110	8
1001011	9
1010001	A
1011100	B
1100101	C
1101000	D
1110010	E
1111111	F