CME 2003
Digital Logic

Number Systems, Operations & Codes

Şerife SUNGUN

Outline

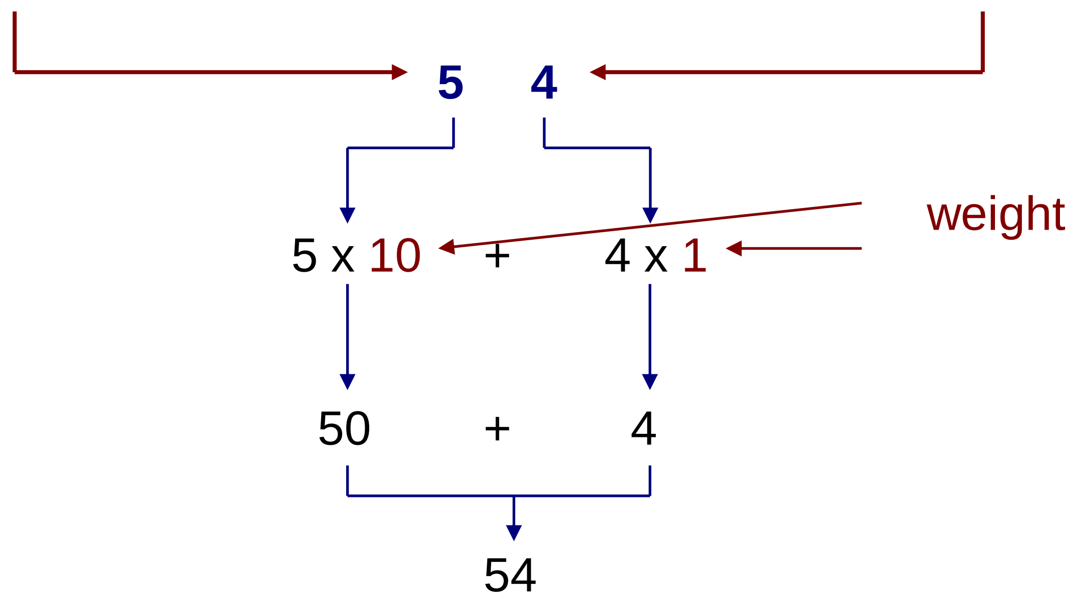
- Decimal Numbers
- Binary Numbers
- Decimal-to-Binary Conversion
- Binary Arithmetic
- 1's and 2's Complements of Binary Numbers
- Signed Numbers
- Arithmetic Operations with Signed Numbers
- Hexadecimal Numbers
- Octal Numbers
- Binary Coded Decimal (BCD)
- Digital Codes
- Error Detection and Correction Codes

Decimal Numbers

The position of each digit in a weighted number system is assigned a weight based on the **base** or **radix** of the system. The radix of decimal numbers is ten, because only ten symbols (0 through 9) are used to represent any number.

The digit 5 has a weight of 10 in this position.

The digit 4 has a weight of 1 in this position.



… Decimal Numbers

Powers of tens

… 10^{3} 10^{2} 10^{1} . 10^{-1} 10^{-2} 10^{-3} …

Decimal point

Example:

568.23 = $(5 \times 10^{2}) + (6 \times 10^{1}) + (8 \times 10^{0}) + (2 \times 10^{-1}) + (3 \times 10^{-2})$

= $(5 \times 100) + (6 \times 10) + (8 \times 1) + (2 \times 0.1) + (3 \times 0.01)$

= $500 + 60 + 8 + 0.2 + 0.03$

Binary Numbers

Binary has a **radix of two** and uses the digits 0 and 1 to represent quantities.

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

Decimal Number	Binary Number
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
10	1 0 1 0
11	1 0 1 1
12	1 1 0 0
13	1 1 0 1
14	1 1 1 0
15	1 1 1 1

Counting in Binary

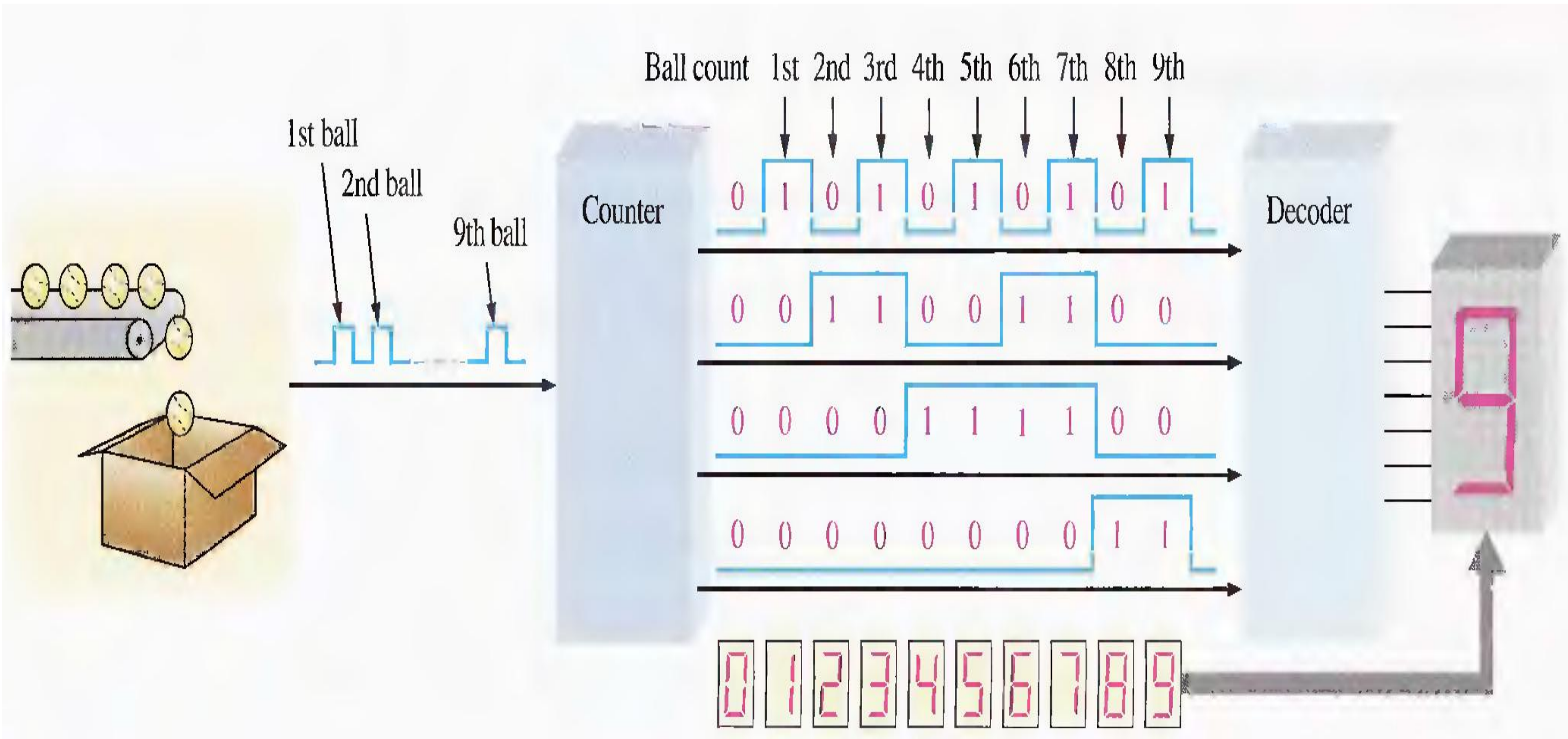
- with **n** bits you can count up to a number equal to 2^{n} - 1.
- Largest decimal number = 2^{n} - 1

Example:

n=4

largest decimal number : $2^{4} - 1 = 15$

A Simple Binary Counting Application



Structure of Binary Numbers

$$2^{n - l} \; . \; . \; . \; 2^{3} \; 2^{2} \; 2^{1} \; 2^{0} \; . \; 2^{-1} \; 2^{-2} \; . \; . \; . \; 2^{-n}$$

MSB **Binary point** **LSB**

LSB (least significant bit): The right-most bit

MSB (most significant bit): The left-most bit

Binary Weights

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

Binary-to-Decimal Conversion

- **The decimal value of any binary number can be found by adding the weights of all bits that are 1 and discarding the weights of all bits that are 0s.**

EXAMPLE

Convert the binary whole number 1101101 to decimal.

Weight:	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Binary number:	1	1	0	1	1	0	1

$$
\begin{aligned}
1101101 &= 2^6 + 2^5 + 2^3 + 2^2 + 2^0 \\
&= 64 + 32 + 8 + 4 + 1 = \mathbf{109}
\end{aligned}
$$

… Binary-to-Decimal Conversion

EXAMPLE

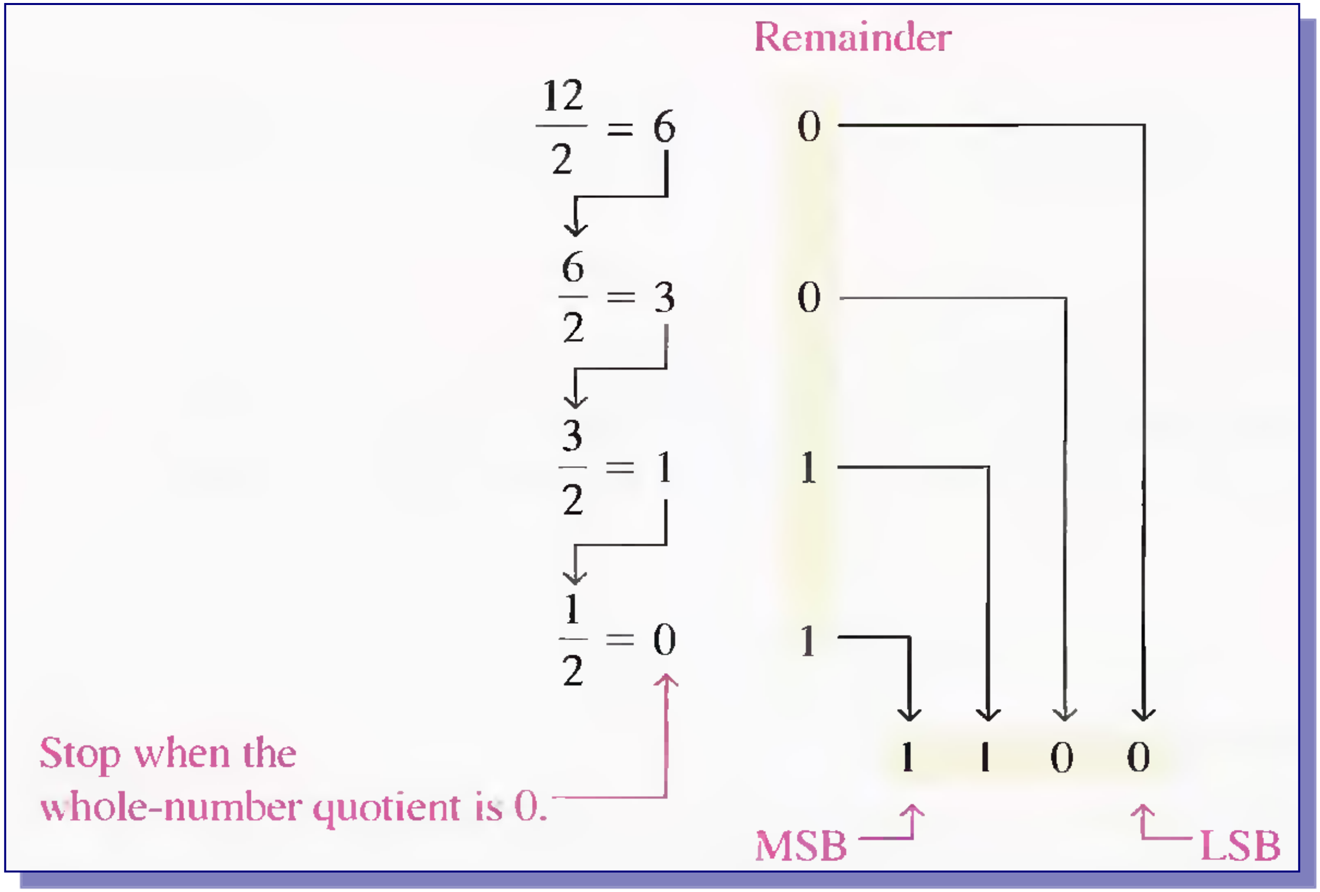
Convert the binary whole number 0.1011 to decimal.

Weight:	2^{-1}	2^{-2}	2^{-3}	2^{-4}
Binary number:	0 . 1	0	1	1

$$0.1011 = 2^{-1} + 2^{-3} + 2^{-4}$$
$$= 0.5 + 0.125 + 0.0625 = \mathbf{0.6875}$$

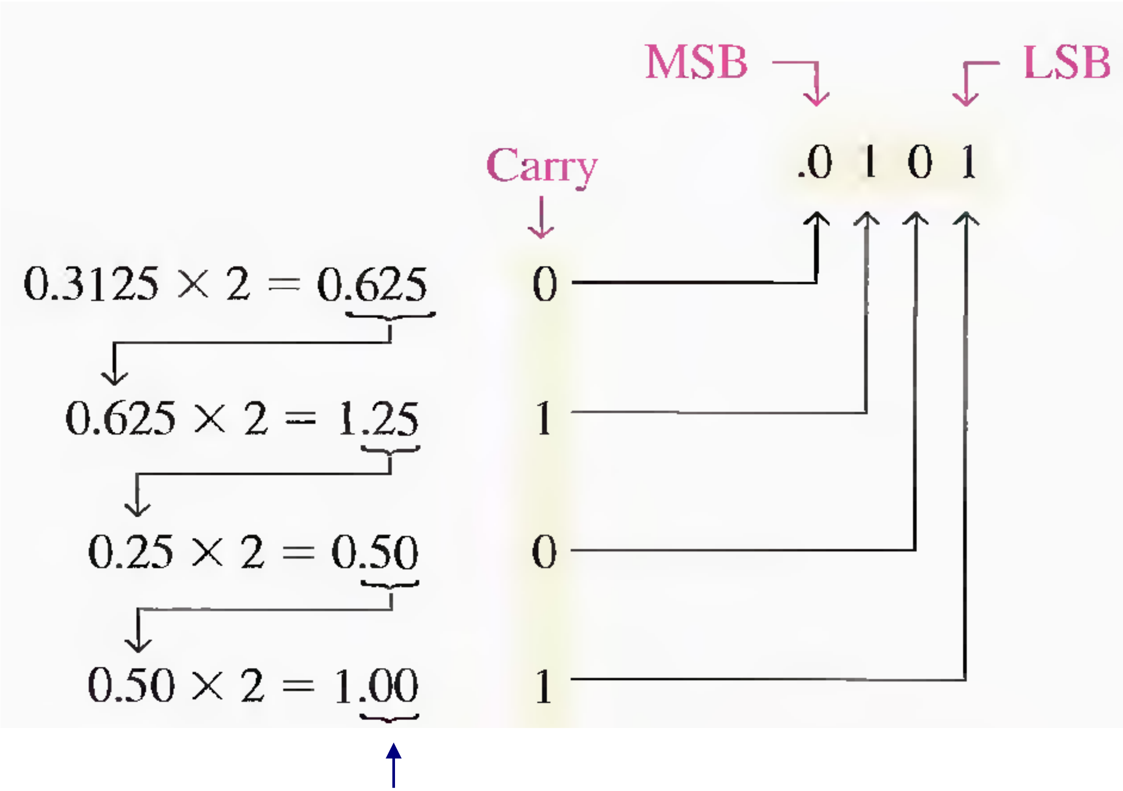
Decimal-to-Binary Conversion

- Repeated Division-by-2 Method



Converting Decimal Fractions to Binary

- Repeated Multiplication by 2



Continue to the desired number of decimal places
or stop when the fractional part is all zeros.

Binary Arithmetic

- Addition
- Subtraction
- Multiplication
- Division

Binary Addition

```
0 + 0 = 0   Sum of 0 with a carry of 0
0 + 1 = 1   Sum of 1 with a carry of 0
1 + 0 = 1   Sum of 1 with a carry of 0
1 + 1 =10   Sum of 0 with a carry of 1
```

… Binary Addition

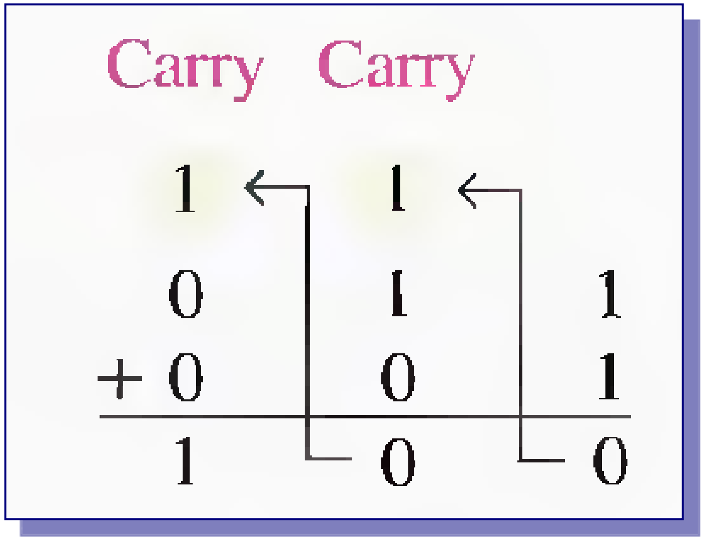
1 + 0 + 0 = **01** Sum of 1 with a carry of **0**

1 + 0 + 1 = **10** Sum of 0 with a carry of **1**

1 + 1 + 0 = **10** Sum of 0 with a carry of **1**

1 + 1 + 1 = **11** Sum of 0 with a carry of **1**

carry



Binary Subtraction

0 – 0 = 0

1 – 1 = 0

1 – 0 = 1

10 - 1 = 1 (0 - 1 with a borrow of **1**)

Left column:
When a 1 is borrowed,
a 0 is left, so $0 - 0 = 0$.

Middle column:
Borrow 1 from next column
to the left, making a 10 in
this column, then $10 - 1 = 1$.

$$
\begin{array}{r}
\overset{0}{\cancel{1}}\,{}^{1}0\,1 \\
\underline{-0\,1\,1} \\
0\,1\,0
\end{array}
$$

Right column:
$1 - 1 = 0$

Binary Multiplication

0 x 0 = 0

0 x 1 = 0

1 x 0 = 0

1 x 1 = 1

Example

$$\begin{array}{lr} & 111 \\ & \times\,101 \\ \hline & 111 \\ \text{Partial products} & 000 \\ & +111 \\ \hline & 100011 \end{array}$$

Binary Division

- Division in binary follows the same procedure as division in decimal.

Example:

Perform the following binary division: 110 / 10 = ?

```
  110 | 10
        ----
_ 10     11
-----
  010
_  10
-----
   00
```

Finding the 1's Complement

The 1's complement of a binary number is found by changing all 1s to 0s and all 0s to 1s,

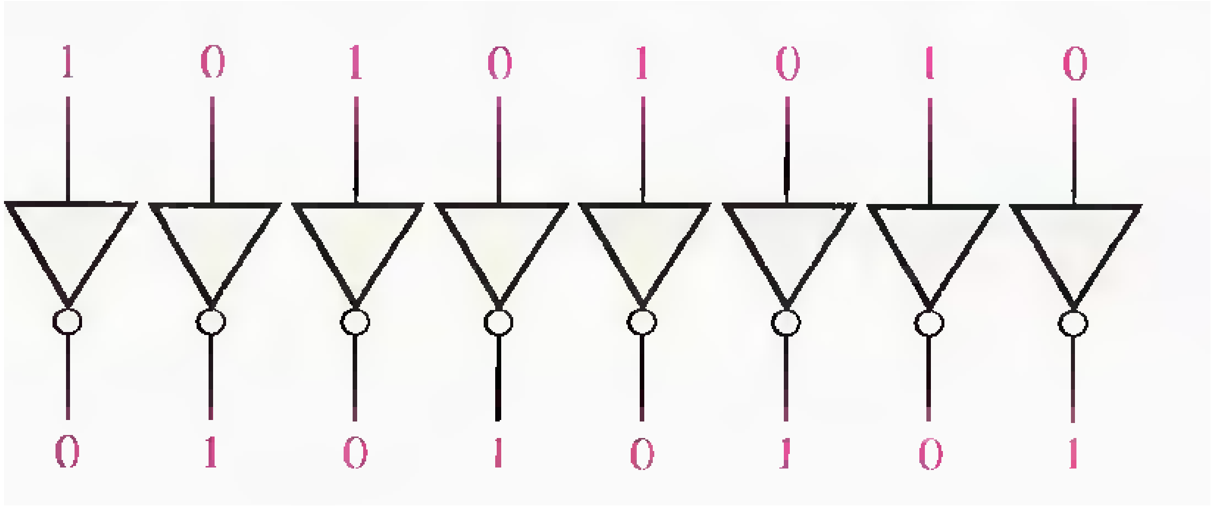
Example

1 0 1 1 0 0 1 0 Binary number

0 1 0 0 1 1 0 1 1 's complement

… Finding the 1's Complement

- The simplest way to obtain the 1's complement of a binary number with a digital circuit is to use parallel inverters (NOT circuits).



Finding the 2's Complement

The 2's complement of a binary number is found by adding 1 to the LSB of the 1's complement.

2's complement = (1's complement) + 1

<u>Example</u>

1	0	1	1	0	0	1	0	Binary number
↓	↓	↓	↓	↓	↓	↓	↓	
0	**1**	**0**	**0**	**1**	**1**	**0**	**1**	1 's complement
+							**1**	Add 1
0	**1**	**0**	**0**	**1**	**1**	**1**	**0**	2 's complement

An alternative method of finding the 2's complement of a binary number

1. Start at the right with the LSB and write the bits as they are up to and including the first 1.
2. Take the 1's complements of the remaining bits.

Change all bits to the left of the least significant 1 to get 2's complement.

Example

- Find the 2's complement of 10111000 using the alternative method.

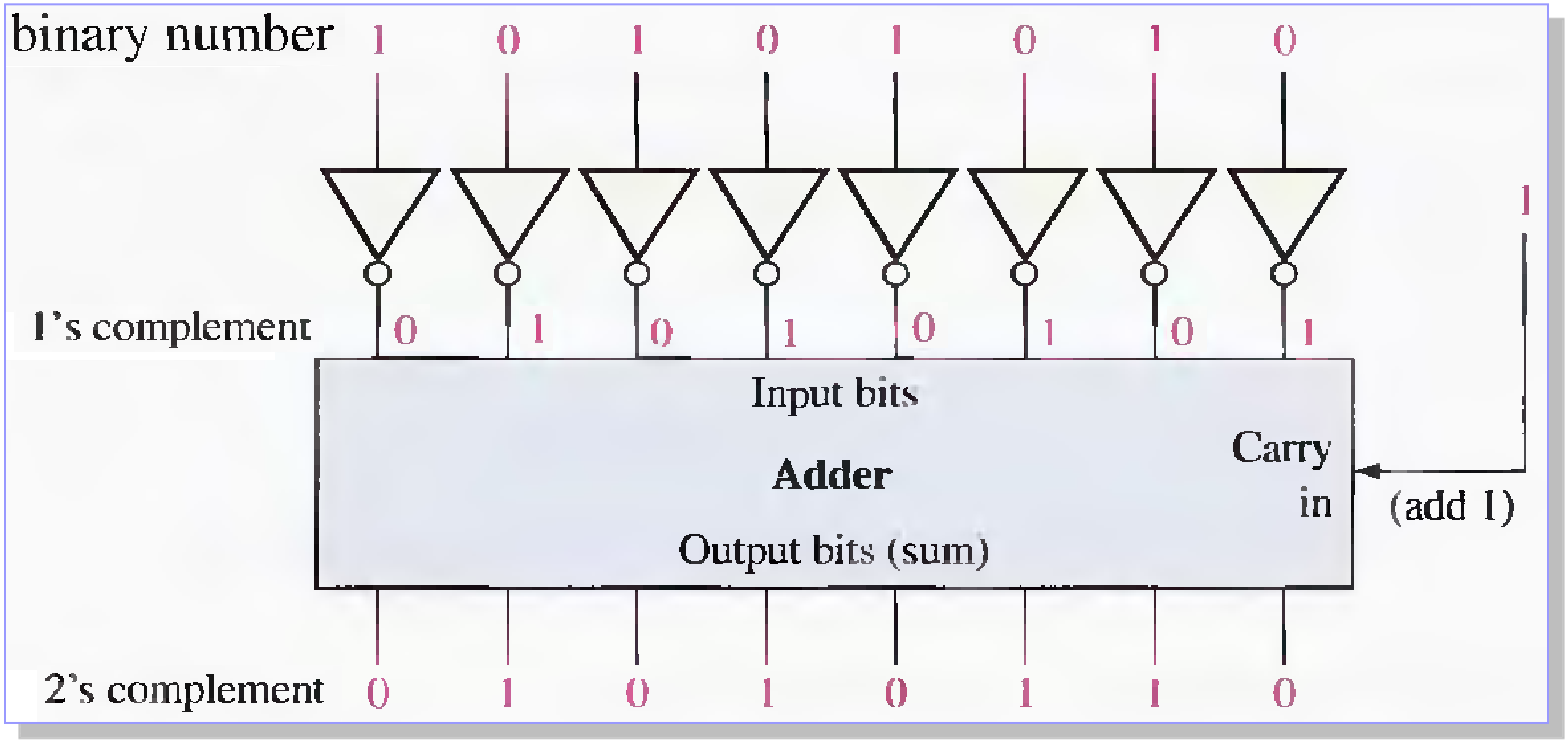
```
10111000

01001000
```

1's complements of original bits

These bits stay the same

Example



binary number
1 0 1 0 1 0 1 0
1
1's complement
0 1 0 1 0 1 0 1
Input bits
Adder
Output bits (sum)
Carry
in
(add 1)
2's complement
0 1 0 1 0 1 1 0

Signed Numbers

- A signed binary number consists of both **sign** and **magnitude** information.
- The sign indicates whether a number is positive or negative, and the magnitude is the value of the number.
  1. sign-magnitude
  2. 1's complement
  3. 2’s complement.

The Sign Bit

- The left-most bit in a signed binary number is the sign bit, which tells you whether the number is positive or negative:

  0 ➢ a positive number

  1 ➢ a negative number.

Sign-Magnitude Form

- A negative number has the same magnitude bits as the corresponding positive number but the sign bit is a 1 rather than a zero.

Example

+ 25 : 0 0 0 1 1 0 0 1

Sign bit   Magnitude bits

- 25 : 1 0 0 1 1 0 0 1

1' s Complement Form

- A negative number is the 1's complement of the corresponding positive number.

Example

+ 25 : 0 0 0 1 1 0 0 1

Sign bit 1's complement

- 25 : 1 1 1 0 0 1 1 0

2' s Complement Form

- A negative number is the 2's complement of the corresponding positive number.

Example

+ 25 : **0** 0 0 1 1 0 0 1

Sign bit 2's complement

- 25 : **1** 1 1 0 0 1 1 1

The Decimal Value of Signed Numbers

- **Example**

Determine the decimal value of this signed binary number expressed in **sign-magnitude**: 10010101.

The Decimal Value of Signed Numbers

- **Example**

Determine the decimal value of this signed binary number expressed in **sign-magnitude**: 10010101.

Solution

The seven magnitude bits and their powers-of-two weights:

Sign bit	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	0	1	0	1	0	1

Summing the weights where there are 1s,

$$16 + 4 + 1 = 21$$

The sign bit is 1; therefore, the decimal number is - 21.

Example

- Determine the decimal values of the signed binary numbers expressed in 1's complement:

a) 00010111

b) 11101000

Solution

(a) The bits and their powers-of-two weights for the positive number are as follows:

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	1	0	1	1	1

Summing the weights where there are 1s,

$$16 + 4 + 2 + 1 = \mathbf{+23}$$

(b) The bits and their powers-of-two weights for the negative number are as follows. Notice that the negative sign bit has a weight of -2^7 or -128.

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	0	1	0	0	0

Summing the weights where there are 1s,

$$-128 + 64 + 32 + 8 = -24$$

Adding 1 to the result, the final decimal number is

$$-24 + 1 = \mathbf{-23}$$

Example

- Determine the decimal values of the signed binary numbers expressed in 2's complement:

a) 01010110

b) 10101010

Solution

(a) The bits and their powers-of-two weights for the positive number are as follows:

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	0	1	0	1	1	0

Summing the weights where there are 1s,

$$64 + 16 + 4 + 2 = \mathbf{+86}$$

(b) The bits and their powers-of-two weights for the negative number are as follows. Notice that the negative sign bit has a weight of $-2^7 = -128$.

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0	1	0	1	0

Summing the weights where there are 1s,

$$-128 + 32 + 8 + 2 = \mathbf{-86}$$

Range of Signed Integer Numbers

For 2's complement signed numbers, the range of values for n-bit numbers

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

Example

n:8 => range = $-(2^7)$ to $+(2^7 - 1)$

-128 to $+127$

Floating-Point Numbers

- is capable of representing very large and very small numbers without an increase in the number of bits
- for representing numbers that have both **integer** and **fractional** components.
- known as a real number
- consists of two parts plus a sign:
  - mantissa : magnitude of the number.
  - exponent : number of places that the decimal point (or binary point) is to be moved.

… Floating-Point Numbers

Example 241,506,800

mantissa : 0.2415068

exponent : 9

$$0.2415068 \times 10^{9}$$

ANSI/IEEE Standard 754-1985

- Single-precision (32 bits)
- Double-precision (64 bits)
- Extended-precision. (80 bits)

Single-Precision Floating-Point

32 bits

S	Exponent (E)	Mantissa (fraction, F)
1 bit	8 bits	23 bits

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

S	E	F
1	10010001	10001110001000000000000

Example Single-Precision Floating-Point

S	E	F
1	10010001	10001110001000000000000

Exponent: 10010001 = 145

Number $= (-1)^{1}\,(1+0.10001110001)2^{(145-127)}$

$= (-1)\,(1.1\;0001110001)2^{(18)}$

$= -1100011100010000000$

This floating-point binary number is equivalent to

-407,688 in decimal.

Example Single-Precision Floating-Point

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*)

- Mantissa (23-bit binary): 11111011100000000000000
- Exponent : $14 + 127 = 141 = 10001101_{2}$

The complete floating-point number is

0	10001101	11111011100000000000000