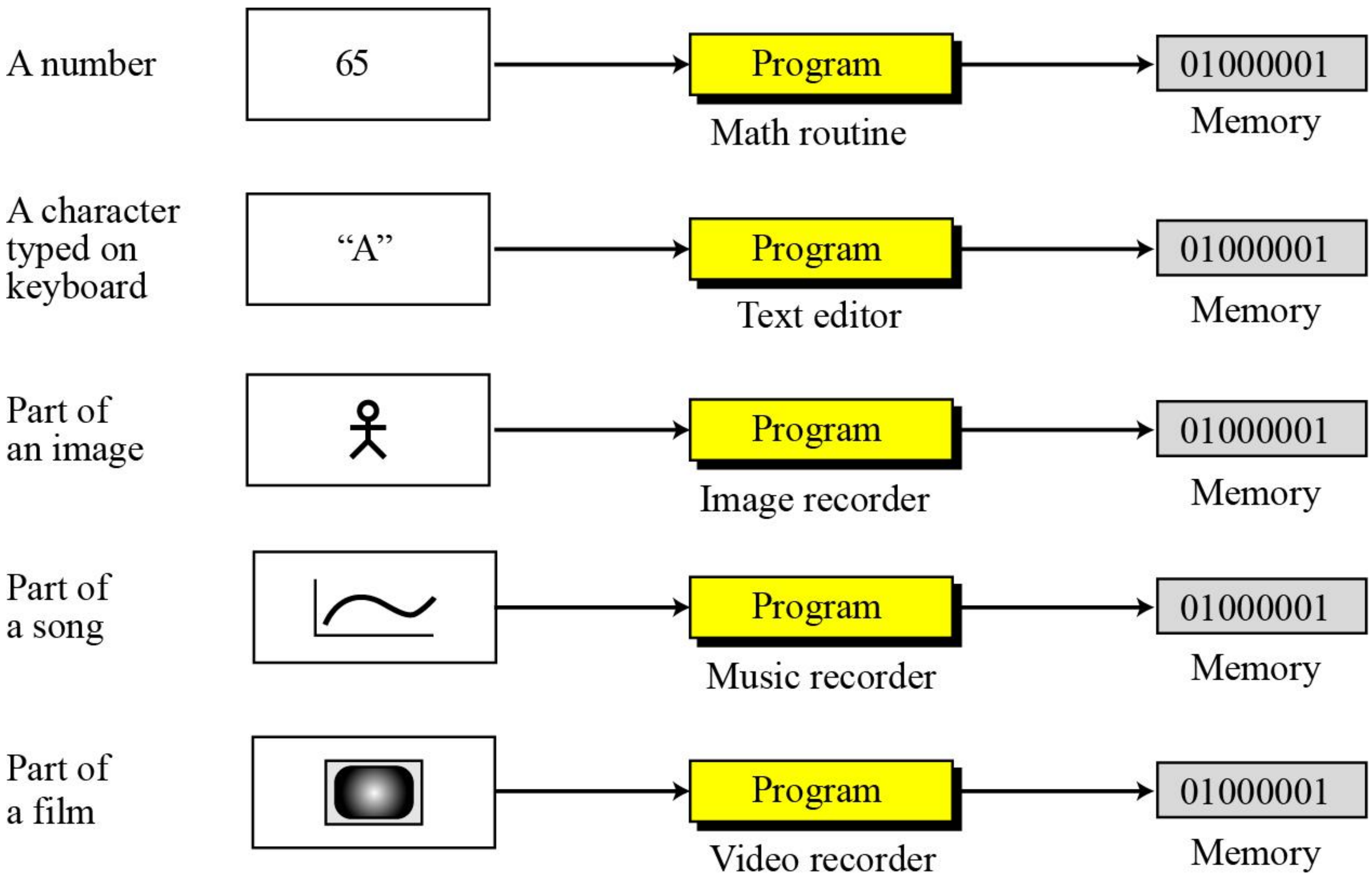
Introduction to Information Technology

CSC109

2019

By: Rajiv Raman Parajuli

Binary Data Representation



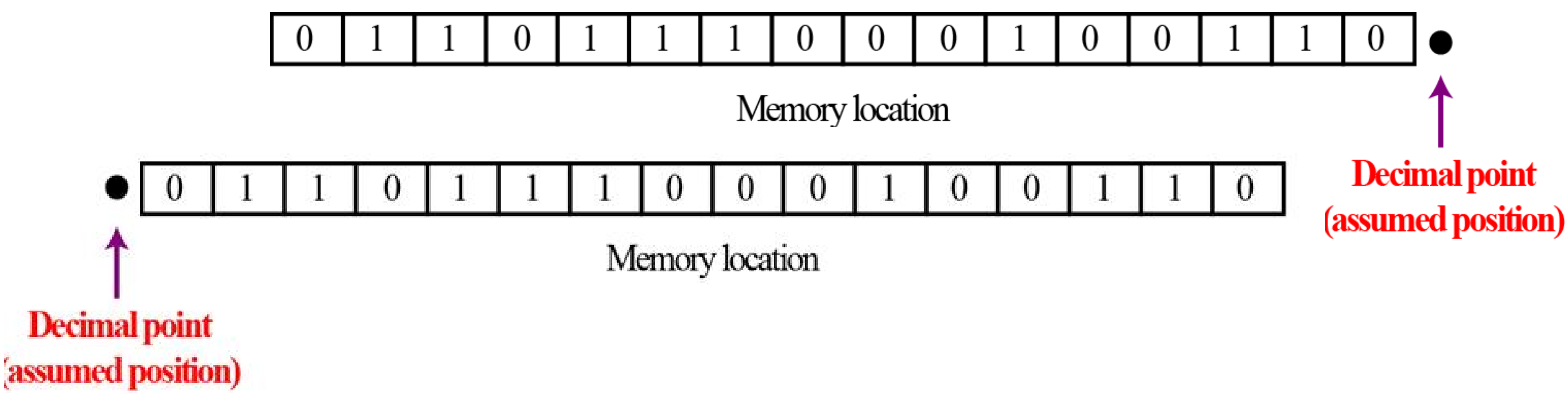
A number is changed to the binary system before being stored in the computer memory, as described in Chapter 2. However, there are still two issues that need to be handled:

1. How to store the sign of the number.
2. How to show the decimal point.

For the decimal point, computers use two different representations**: fixed-point and floating-point**.

Fixed-point representation

- binary point is fixed at one position
- extreme left make the number a fraction, or at the extreme right to make the number an integer
- the binary point is not stored in the register
- but the number is treated as a fraction or integer



Fixed Point
Unsigned representation

An **unsigned integer** is an integer that can never be negative and can take only 0 or positive values. Its range is between 0 and positive infinity.

$0 \rightarrow (2^n - 1)$

An input device stores an unsigned integer using the following steps:

1. The integer is changed to binary.
2. If the number of bits is less than *n*, 0s are added to the left.

Example 1

Store 7 in an 8-bit memory location using unsigned representation.

Solution

First change the integer to binary, $(111)_2$. Add five 0s to make a total of eight bits, $(00000111)_2$.

The integer is stored in the memory location

Change 7 to binary	→						1	1	1
Add five bits at the left	→	0	0	0	0	0	1	1	1

Fixed Point
Signed representation

In this method, the available range for unsigned integers (0 to $2^n - 1$) is divided into two equal sub-ranges. The first half represents **positive integers**, the second half, **negative integers.**

0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	1	2	3	4	5	6	7	–0	–1	–2	–3	–4	–5	–6	–7

The leftmost bit "MSB" defines the sign of the integer. If it is 0, the integer is positive. If it is 1, the integer is negative.

Example 2

Store +28 in an 8-bit memory location using sign-and-magnitude representation.

Change 28 to 7-bit binary		0	0	1	1	1	0	0
Add the sign and store	**0**	0	0	1	1	1	0	0

Example 3

Store -28 in an 8-bit memory location using sign-and-magnitude representation.

Change 28 to 7-bit binary		0	0	1	1	1	0	0
Add the sign and store	**1**	0	0	1	1	1	0	0

Example 4

Retrieve the integer that is stored as 01001101 in sign-and-magnitude representation.

Example 5

Retrieve the integer that is stored as 10100001 in sign-and-magnitude representation.

One’s Complement

Original pattern	0	0	1	1	0	1	1	0
After applying one’s complement operation	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**1**

Two’s Complement

Method 1: Find One’s Complement and add 1 to LSB.

Method 2: This operation is done in two steps. First, we copy bits from the right until a 1 is copied; then, we flip the rest of the bits.

Original integer	0	0	1	1	0	1	0	0
	↓	↓	↓	↓	↓	↓	↓	↓
Two’s complementing once	**1**	**1**	**0**	**0**	**1**	1	0	0

we get the original integer if we apply the one's complement operations twice.

Original pattern	0	0	1	1	0	1	1	0
One's complementing once	1	1	0	0	1	0	0	1
One's complementing twice	0	0	1	1	0	1	1	0

we always get the original integer if we apply the two's complement operation twice.

Original integer	0	0	1	1	0	1	0	0
	↓	↓	↓	↓	↓	↓	↓	↓
Two's complementing once	1	1	0	0	1	1	0	0
	↓	↓	↓	↓	↓	↓	↓	↓
Two's complementing twice	0	0	1	1	0	1	0	0

Storing an integer in two’s complement format:

- The integer is changed to an n-bit binary.
- If it is positive or zero, it is stored as it is.
- If it is negative, take the two’s complement and then stores it.

Retrieving an integer in two’s complement format:

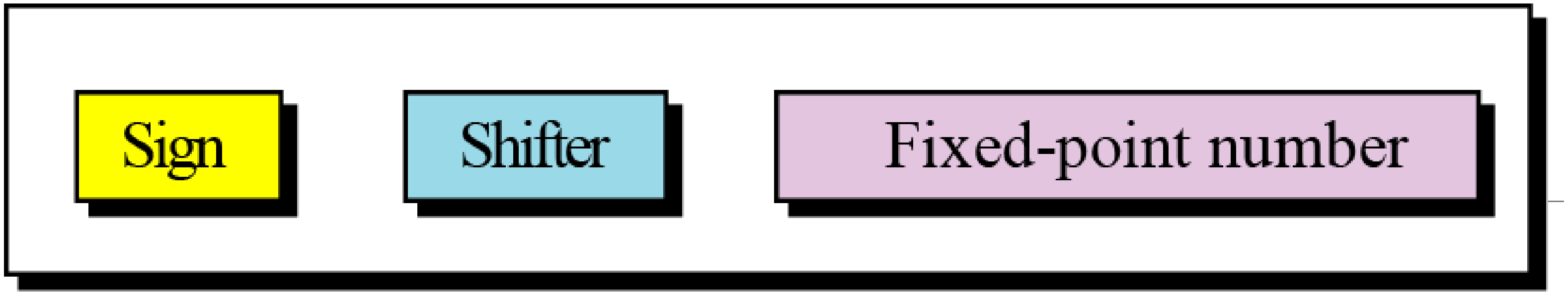
- If the leftmost bit is 1, the computer applies the two’s complement operation to the integer.
- If the leftmost bit is 0, no operation is applied.
- The computer changes the integer to decimal.

Floating-point representation

- For maintaining accuracy or precision floating-point representation is use.
- uses two registers
- The first register stores the number without the binary point
- The second register stores a number that indicates the position of the binary point

For example, 23.7 is a real number—the integral part is 23 and the fractional part is 7/10.

A floating point representation of a number is made up of three parts: a sign, a shifter and a fixed-point number.



Floating-point pepresentation

Floating-point representation is used in science to represent very small or very large decimal numbers. In this representation called scientific notation, the fixed-point section has only one digit to the left of point and the shifter is the power of 10.

Example 6

The following shows the decimal number
7,452,000,000,000,000,000,000.00
in scientific notation (floating-point representation).

Actual number	→	+	7,425,000,000,000,000,000,000.00
Scientific notation	→	+	7.425×10^{21}

The three sections are the sign (+), the shifter (21) and the fixed-point part (7.425).
Note that the shifter is the exponent.
Some programing languages and calculators shows the number as +7.425E21

Example 7

Show the number
−0.0000000000000232
In floating-point representation.

Solution

We use the same approach as in the previous example—we move the decimal point after the digit 2, as shown below:

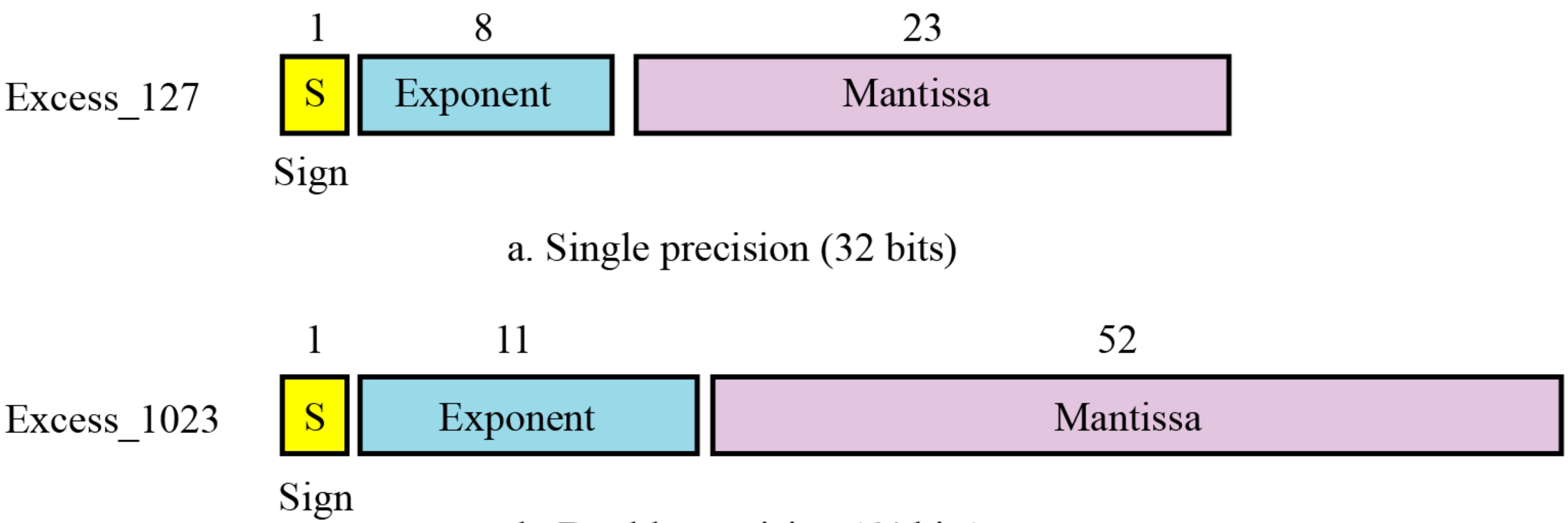
Actual number	→	–	0.0000000000000232
Scientific notation	→	–	2.32×10^{-14}

The three sections are the sign (–), the shifter (–14) and the fixed-point part (2.32). Note that the shifter is the exponent.

Exercise

1. $(101001000000000000000000000000000000.00)_2$
2. $-(0.0000000000000000000000000101)_2$
3. $(101001000000000000000000000000000000.00)_2$

IEEE Standard



a. Single precision (32 bits)

b. Double precision (64 bits)

Binary Coding Schemes

Three main coding systems that provide conversions of keyboard characters into binary:

- EBCDIC
- ASCII
- Unicode

EBCDIC

- EBCDIC stands for Extended Binary Coded Decimal Interchange Code.
- It is an extension of BCD which includes non-numeric characters, including all the keyboard characters and special characters.
- It is commonly used to encode data onto magnetic tape
- Uses 8 bits (4 bits for zone, 4 bits for digit)
- 256 unique symbols are represented by it
- EBCDIC codes are mainly used in Mainframe computers

ASCII

- ASCII stands for the American Standard Code for Information Interchange.
- It has been adopted as the industry-standard way of representing keyboard characters as binary codes.
- Every keyboard character is given a corresponding binary code.
- Two Types
- ASCII-7 ; 7 bit (3 bits for zone, 4 bits for digit), 128 characters
- ASCII-8; 8-bit (4 bits for zone, 4 bits for digit), code to provide 256 characters. Widely in use

Unicode

- UNICODE is the new standard to emerge that is replacing ASCII.
- Simple and efficient
- It has been adopted by many of the big businesses in the computing industry.
- It is designed to cover more of the characters that are found in languages across the world.
- It has become important due to the increased use of the Internet, as more data is being passed around globally.
- Check Unicode character encoding in MS–Word