Chapter 1

DATA REPRESENTATION

Number Systems

Number system	Base	Coefficients
Decimal	10	0 – 9
Binary	2	0 , 1
Octal	8	0 – 7
Hexadecimal	16	0 – 9, A – F

Binary Shop

Base conversion

- From any *base-r* to Decimal
- From Decimal to any *base-r*
- From Binary to either Octal or Hexadecimal
- From either Octal or Hexadecimal to Binary

Binary to octal or hexadecimal

Binary To Octal

Procedure:

- Partition binary number into groups of 3 digits

Example:

- $(10110001101011.11110000110)_2$ = $\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\__8$

Binary to octal or hexadecimal

$10\ 110\ 001\ 101\ 011\ .\ 111\ 100\ 000\ 110_2$

Binary to octal or hexadecimal

$$10\ 110\ 001\ 101\ 011\ .\ 111\ 100\ 000\ 110_2$$

$$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \qquad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$$

$$2\ \ 6\ \ 1\ \ 5\ \ 3\ .\ 7\ \ 4\ \ 0\ \ 6_8$$

Binary to octal or hexadecimal

Binary To Hexadecimal

Procedure:

- Partition binary number into groups of 4 digits

Example:

- $(101100011101011.11110010)_2$

  $= \underline{\hspace{7cm}}_{16}$

Binary to octal or hexadecimal

$10\ 1100\ 0110\ 1011\ .\ 1111\ 0010_2$

↓ ↓ ↓ ↓ ↓ ↓

Binary to octal or hexadecimal

$$10\ 1100\ 0110\ 1011\ .\ 1111\ 0010_{\ 2}$$

$$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$$

$$2\ \ C\ \ 6\ \ B\ .\ F\ \ 2_{\ 16}$$

Octal or hexadecimal to binary

Octal To Binary

Procedure:

- Each octal digit is converted to its 3-digit binary equivalent

Example: $(673.124)_8 = \_\_\_\_\__2$

Octal or hexadecimal to binary

$6 \quad 7 \quad 3 \;.\; 1 \quad 2 \quad 4_{8}$

$\downarrow \quad \downarrow \quad \downarrow \quad\;\; \downarrow \quad \downarrow \quad \downarrow$

Octal or hexadecimal to binary

$$
\begin{array}{ccccccccc}
6 & 7 & 3 & . & 1 & 2 & 4 & {}_8 \\
\downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow & \\
110 & 111 & 011 & . & 001 & 010 & 100 & {}_2
\end{array}
$$

Octal or hexadecimal to binary

Hexadecimal to Binary

Procedure:

- Each hexadecimal digit is converted to its 4-digit binary equivalent

Example: $(306.D)_{16}$ = ______ $_2$

Octal or hexadecimal to binary

$$3 \quad 0 \quad 6 \; . \; D_{16}$$

$$\downarrow \quad \downarrow \quad \downarrow \quad \quad \downarrow$$

Octal or hexadecimal to binary

$$
\begin{array}{ccccc}
3 & 0 & 6 & . & D_{16} \\
\downarrow & \downarrow & \downarrow & & \downarrow \\
0011 & 0000 & 0110 & . & 1101_{2}
\end{array}
$$

Any Other Number System

- In general, a number expressed in ***base-r*** has *r* possible coefficients multiplied by powers of *r*:

$$a_n r^n + a_{n-1} r^{n-1} + a_{n-2} r^{n-2} + \ldots + a_0 r^0 + a_{-1} r^{-1} + a_{-2} r^{-2} + \ldots + a_{-m} r^{-m}$$

where n = position of the coefficient

coefficients = 0 to r-1

Example

Base 5 number

coefficients: 0 to r-1 (0, 1, 2, 3, 4)

- $(324.2)_5$

Example

Base 5 number

coefficients: 0 to r-1 (0, 1, 2, 3, 4)

- $(324.2)_5$

$= 3 \times 5^2 + 2 \times 5^1 + 4 \times 5^0 + 2 \times 5^{-1}$

$= 89.4_{10}$

Example

Base 5 number

coefficients: 0 to r-1 (0, 1, 2, 3, 4)

- $(324.2)_5$

  $= 3 \times 5^2 + 2 \times 5^1 + 4 \times 5^0 + 2 \times 5^{-1}$

  $= 89.4_{10}$
- $(4021)_5$

Example

Base 5 number

coefficients: 0 to r-1 (0, 1, 2, 3, 4)

- $(324.2)_5$

  $= 3 \times 5^2 + 2 \times 5^1 + 4 \times 5^0 + 2 \times 5^{-1}$

  $= 89.4_{10}$
- $(4021)_5$

  $= 4 \times 5^3 + 0 \times 5^2 + 2 \times 5^1 + 1 \times 5^0$

  $= 511_{10}$

Fixed-Point Representation

Unsigned Number

- leftmost bit is the most significant bit
- Example:
  - 01001 = 9
  - 11001 = 25

Fixed-Point Representation

Unsigned Number

- leftmost bit is the most significant bit
- Example:
  - 01001 = 9
  - 11001 = 25

Signed Number

- leftmost bit represents the sign
- Example:
  - 01001 = +9
  - 11001 = - 9

Systems Used to Represent Negative Numbers

Signed-Magnitude Representation

- A number consists of a magnitude and a symbol indicating whether the magnitude is positive or negative.

Examples:

$+85 = 01010101_2$ $\quad$ $-85 = 11010101_2$

$+127 = 01111111_2$ $\quad$ $-127 = 11111111_2$

Systems Used to Represent Negative Numbers

Signed-Complement System

- This system negates a number by taking its complement as defined by the system.
- Types of complements:
  - Radix-complement
  - Diminished Radix-complement

Complements

- Diminished Radix Complement

  General Formula:

  $(r-1)$'s C of N $= (r^n - r^{-m}) - N$

  where

  n = # of digits (integer)

  m = # of digits (fraction)

  r = base/radix

  N = the given # in base-r

- Radix Complement

  General Formula:

  r's C of N $= r^n - N$

  where

  n = # of bits

  r = base/radix

  N = the given # in base-r

Complements

Examples

- 9's C
  - 012390 = 987609
  - 54670.5 = 45329.4
- 10's C
  - 012390 = 987610
  - 54670.5 = 45329.5

Complements

Examples

- 9's C
  - 012390 = 987609
  - 54670.5 = 45329.4
- 10's C
  - 012390 = 987610
  - 54670.5 = 45329.5

Examples

- 1's C
  - 1101100 = 0010011
  - 0110111 = 1001000
- 2's C
  - 1101100 = 0010100
  - 0110111 = 1001001

Binary Codes

Code – a set of n-bit strings in which different bit strings represent different numbers or other things.

Binary codes are used for:

- Decimal numbers
- Character codes

Binary codes for decimal numbers

At least four bits are needed to represent ten decimal digits.

Some binary codes:

- BCD (Binary-coded decimal)
- Excess-3
- Biquinary

Binary codes for decimal numbers

BCD

- straight assignment of the binary equivalent
- weights can be assigned to the binary bits according to their position

Excess-3 Code

- unweighted code
- BCD + 3

Biquinary Code

- seven-bit code with error detection properties
- each decimal digit consists of 5 0’s and 2 1’s

Binary codes for the decimal digits

Decimal Digit	BCD 8421	Excess-3	84-2-1	2421	Biquinary 5043210
0	0000	0011	0000	0000	0100001
1	0001	0100	0111	0001	0100010
2	0010	0101	0110	0010	0100100
3	0011	0110	0101	0011	0101000
4	0100	0111	0100	0100	0110000
5	0101	1000	1011	1011	1000001
6	0110	1001	1010	1100	1000010
7	0111	1010	1001	1101	1000100
8	1000	1011	1000	1110	1001000
9	1001	1100	1111	1111	1010000

Differences between Binary and BCD

- BCD is not a number system
- BCD requires more bits than Binary
- BCD is less efficient than Binary
- BCD is easier to use than Binary

Coding vs Conversion

Conversion

- bits obtained are binary digits
- Example: 13 = 1101

Coding

- bits obtained are combinations of 0's and 1's
- Example: 13 = 0001 0011

Character Code

American Standard Code for Information Exchange

- 7-bit code
- contains 94 graphic chars and 34 non-printing chars

Extended Binary Coded Decimal Interchange Code

- 8-bit code
- last 4 bits range from 0000-1001

Gray code

- It is a binary number system where two successive values differ in only one digit, originally designed to prevent spurious output from electromechanical switches.

Decimal	Binary code	Gray code
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Number Representations

- No representation method is capable of representing all real numbers
- Most real values must be represented by an approximation

- Various methods can be used:
  - Fixed-point number system
  - Rational number system
  - Floating point number system
  - Logarithmic number system

Fixed-point representation

- It is a method used to represent integer values.
- Disadvantages
  - Very small real numbers are not clearly distinguished
  - Very large real numbers are not known accurately enough

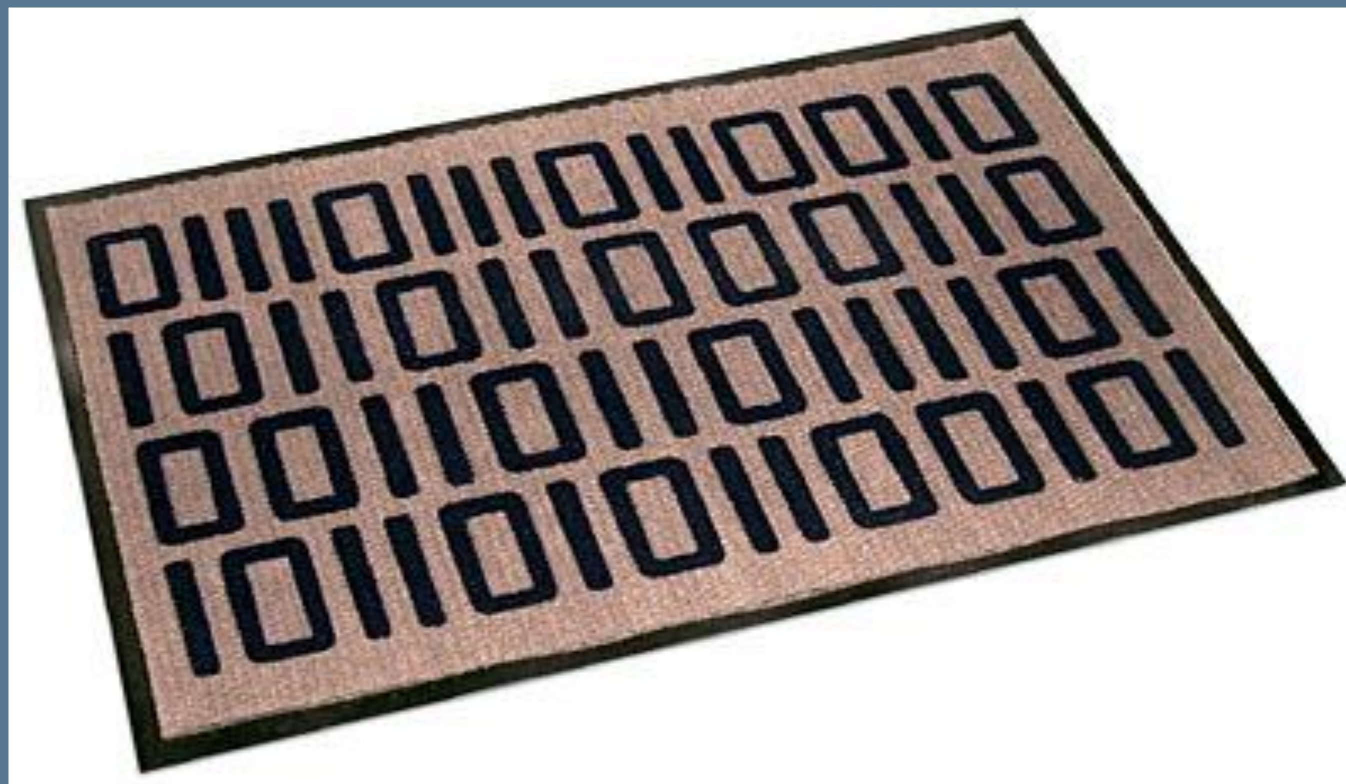
Floating-point Representation

- It is a method used to represent real numbers
- Notation:
  - Mantissa x $\text{Base}^{\text{exponent}}$
- Example

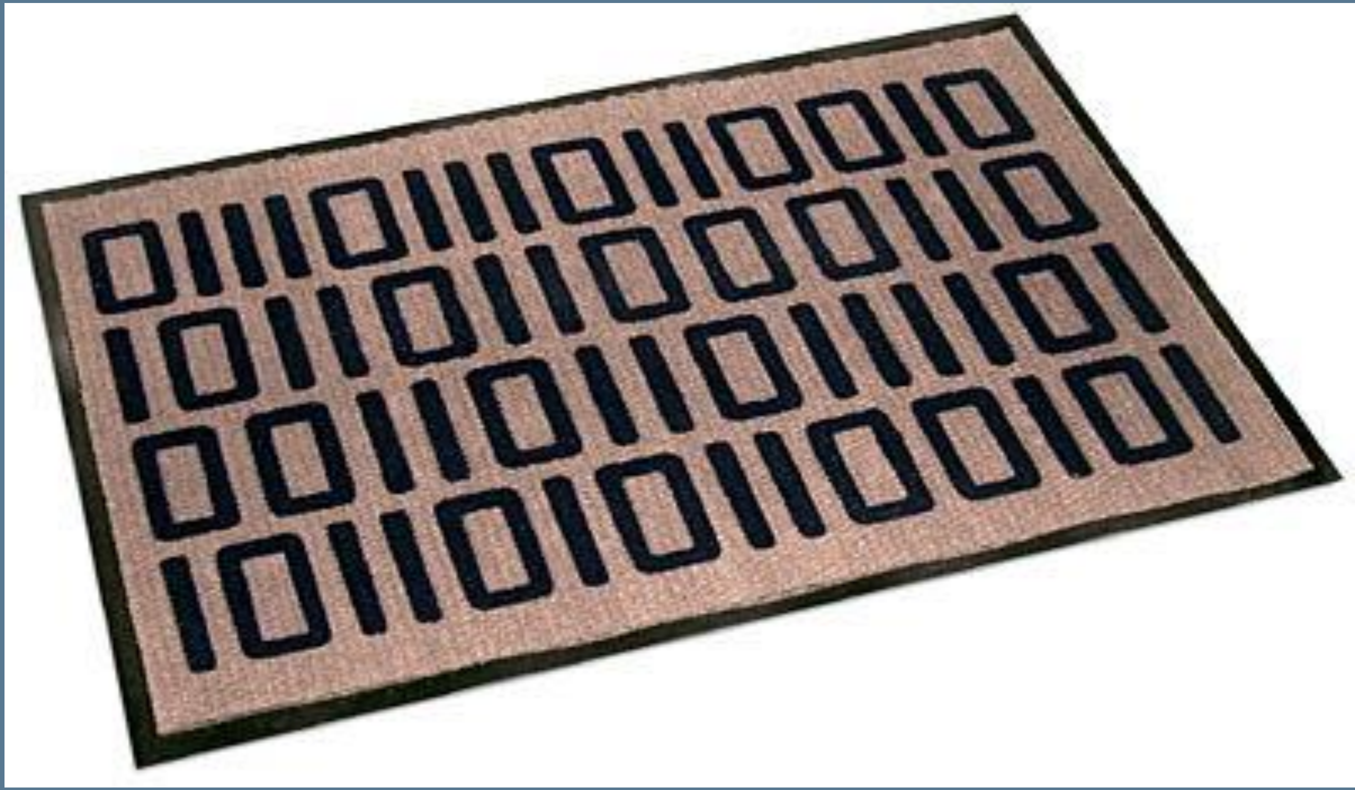
  1 1000011 00010010110000000000000000

  = - 300.0

Activity



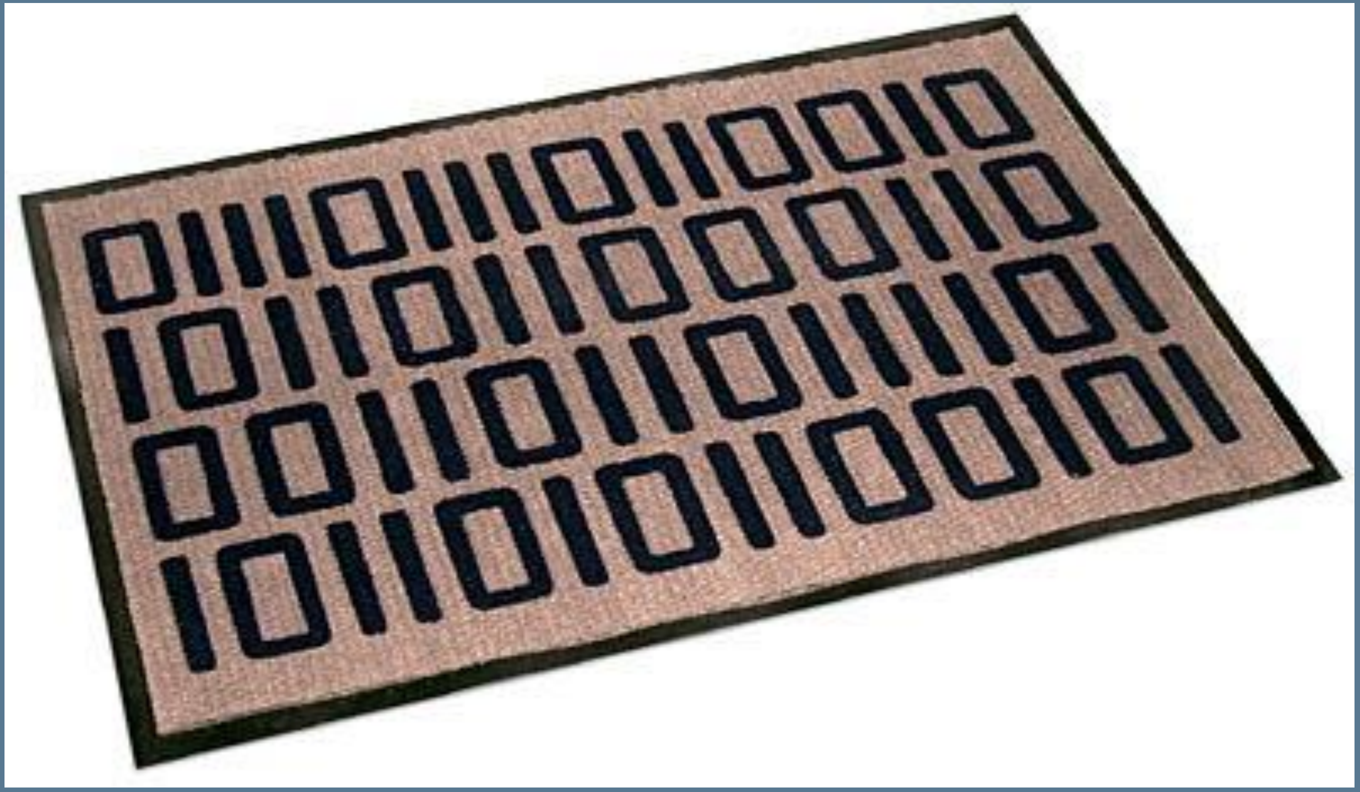
Activity



01110111 01100101 01101100 01100011 01101111 01101101 01100101

Activity

97 = ‘a’



01110111 01100101 01101100 01100011 01101111 01101101 01100101