MODULE 2: Data Representation

Lecture 2.1
Unsigned Data Representation

Prepared By:
- Scott F. Midkiff, PhD
- Luiz A. DaSilva, PhD
- Kendall E. Giles, PhD

Electrical and Computer Engineering
Virginia Tech

Lecture 2.1 Objectives

- Explain how to represent any (unsigned) value, including integers and fractions, in any radix
- Convert between decimal, binary, octal, and hexadecimal representations
- Describe the importance of binary, octal, and hexadecimal representations in computer systems

Data Representation

- Computers and other digital devices represent information using only “1’s” and “0’s” (bits)
  - Voltage level
  - Current level
  - Electrical charge
  - Orientation of a magnetic field
  - Reflection or non-reflection of light
- Higher level information must be represented using sequences of 1’s and 0’s
  - Numerical, fixed, and floating point numbers
  - Non-numerical, e.g. alphabetic characters

Numerical Data

- Numerical data is usually processed and stored based on binary (base-2 or radix-2) representation
  - Two symbols: {0,1}
- Examples
  - $(13)_{10} = (1101)_2$
  - $(13.25)_{10} = (1101.01)_2$
- We use a “binary point” to separate the integer part of the number from the fraction part

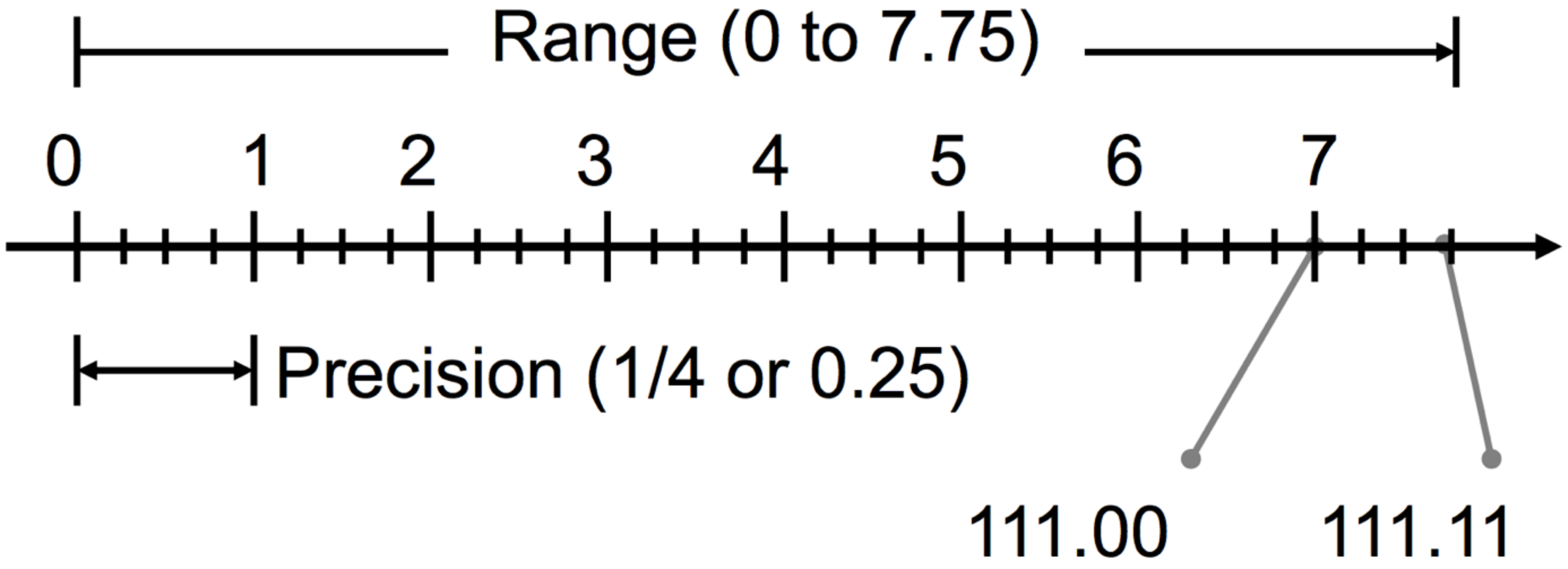
Fixed Point Numbers

- Use a fixed number of bits to represent any number
- Always assume that the decimal or binary point is in the same location (by convention, not explicitly stored)
- Examples
  - Decimal: 034.750, 543.201, 000.334
  - Binary: 1111.01, 0110.00, 0101.10
- The number of binary digits (bits) determines the precision and range of values that can be represented

Range and Precision

- The “range” of a representation scheme is the distance from the smallest to largest value that can be represented
  - E.g., What’s the biggest number that can be represented?
- The “precision” of a representation scheme indicates the closeness to any actual value that can be represented
  - E.g., What’s the error in representing a number?
- Floating point representation allows variable trade-off between range and precision

Graphical View



- Assuming 3 bits for integer and 2 bits for fraction
  - 000.00, 000.01, 000.10, 000.11, 001.00, 001.01, 001.10, 001.11, 010.00, 010.01, ..., 110.11, 111.00, 111.01, 111.10, 111.11

CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can do the following:

- If the smallest three numbers in a fixed point format using four bits for the integer part and two bits for the fraction part are 0000.00 = 0, 0000.01 = .25, and 0000.10 = .5, and the largest three numbers in this format are 1111.01 = 15.25, 1111.10 = 15.5, and 1111.11 = 15.75, what are the range and precision of this number format?

CHECK POINT

Answer:

- The Range = 15.75 - 0 = 15.75, and the Precision = .25.

If you have any difficulties, please review the lecture video before continuing.

Radices

- Values are represented using some base or radix, e.g. radix-2 (binary) or radix-10 (decimal)
- In general, radix *r* has *r* symbols: { 0, ..., *r*-1 }
    - Binary: {0, 1}
    - Decimal: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
    - Radix-8: {0, 1, 2, 3, 4, 5, 6, 7}
    - Radix-16: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}
- Values are represented with sequences of these symbols, with each location in the sequence having an implicit weight that is a power of *r*

Radix-10

- Value of a decimal number where b_i is the decimal digit in position i from the decimal point

$$Value = \sum_{i=-m}^{n-1} b_i \times 10^i$$

- Example: 209.75
  - $b_2 = 2$, $b_1 = 0$, $b_0 = 9$, $b_{-1} = 7$, $b_{-2} = 5$
  - $2\times10^2 + 0\times10^1 + 9\times10^0 + 7\times10^{-1} + 5\times10^{-2}$
  - $2\times100 + 0\times10 + 9\times1 + 7\times0.1 + 5\times0.01 = 209.75$

Radix-2

- Value of a binary number where b_i is the binary digit in position i from the binary point

$$Value = \sum_{i=-m}^{n-1} b_i \times 2^i$$

- Example: 1101.01
  - $b_3 = 1$, $b_2 = 1$, $b_1 = 0$, $b_0 = 1$, $b_{-1} = 0$, $b_{-2} = 1$
  - $1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2}$
  - $1\times8 + 1\times4 + 0\times2 + 1\times1 + 0\times0.5 + 1\times0.25$
  - $8 + 4 + 0 + 1 + 0 + 0.25 = (13.25)_{10}$

Radix-*r*

For any radix *r*, value of a radix-*r* number where b_i is the digit in position *i* from the radix point

$$Value = \sum_{i=-m}^{n-1} b_i \times r^i$$

- Example for radix 8: 157.10
  - $b_2 = 1$, $b_1 = 5$, $b_0 = 7$, $b_{-1} = 1$, $b_{-2} = 0$
  - $1 \times 8^2 + 5 \times 8^1 + 7 \times 8^0 + 1 \times 8^{-1} + 0 \times 8^{-2}$
  - $1 \times 64 + 5 \times 8 + 7 \times 1 + 1 \times (1/8) + 0 \times (1/64)$
  - $64 + 40 + 7 + 0.125 = (111.125)_{10}$

CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can:

- Compute the decimal value of this number for radix 8: 147.11

CHECK POINT

Answer:

- $147.11_8 = 1\times8^2 + 4\times8^1 + 7\times8^0 + 1\times8^{-1} + 1\times8^{-2}$
  $= 64 + 32 + 7 + .125 + .015625$
  $= 103.140625_{10}$

If you have any difficulties, please review the lecture video before continuing.

Converting Between Radices

- Converting from any radix to decimal can be done, as just seen, using the “polynomial method”
- Converting a decimal value to any radix is done by separately converting the integer and fraction parts
  - Integer conversion using the “remainder method” with repeated division by the target radix
  - Fraction conversion using the “multiplication method” with repeated multiplication

Integer Conversion

- Divide value by *r* (where target is radix *r*) and remainder is next most significant digit
  - Note that remainder will always be less than *r*
- Example: convert $(25)_{10}$ to binary (radix-2)

$25 \div 2 = 12$	remainder 1	$b_0 = 1$
$12 \div 2 = 6$	remainder 0	$b_1 = 0$
$6 \div 2 = 3$	remainder 0	$b_2 = 0$
$3 \div 2 = 1$	remainder 1	$b_3 = 1$
$1 \div 2 = 0$	remainder 1	$b_4 = 1$

Value = $(11001)_2$

Fraction Conversion

- Multiply by *r* (where target is radix *r*) and any integer part of the product is the next least significant bit
- Example: convert $(0.45)_{10}$ to binary (radix-2)

$0.45 \times 2 = 0.9$	integer = 0	$b_{-1} = 0$
$0.9 \times 2 = 1.8$	integer = 1	$b_{-2} = 1$
$0.8 \times 2 = 1.6$	integer = 1	$b_{-3} = 1$
$0.6 \times 2 = 1.2$	integer = 1	$b_{-4} = 1$
$0.2 \times 2 = 0.4$	integer = 0	$b_{-5} = 0$
$0.4 \times 2 = 0.8$	integer = 0	$b_{-6} = 0$
$0.8 \times 2 = 1.6$	integer = 1	$b_{-7} = 1$

Value = $(0.0111001100110011\ldots)_2$

Precision Revisited

- Many values cannot be represented exactly in a given radix *r*, even though they can be in decimal
- In a practical implementation, some finite number of bits will be used for the fraction
  - In the previous example, we might represent an approximation of $(0.45)_{10}$ as $(0.011100)_2$ or $(0.011101)_2$
- This limited number of bits for representation leads to an error in representation or a loss in precision

CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can:

- Convert $(4.25)_{10}$ to binary (radix-2)

CHECK POINT

Answer:

$(4.25)_{10}$

Integer part:

4 / 2 = 2 remainder 0

2 / 2 = 1 remainder 0

1 / 2 = 0 remainder 1 stop

Fraction part:

0.25 * 2 = 0.5 integer 0

0.5 * 2 = 1.0 integer 1

0.0 * 2 = 0.0 integer 0 stop

Therefore $4.25_{10} = 100.01_2$

If you have any difficulties, please review the lecture video before continuing.

Hexadecimal and Octal

- Hexadecimal (or "hex") and octal representation are of special value since they are based on radices that are powers of 2
  - Hexadecimal: radix-16
  - Octal: radix-8
- Useful as more compact representations of sequences of bits
- Since16 = 2^4, one hexadecimal digit represents four bits
- Since 8 = 2^3, one octal digit represents three bits

Hexadecimal Symbols

- Hexadecimal (radix-16) requires six new symbols
  - $(A)_{16} = (10)_{10} = (1010)_2$
  - $(B)_{16} = (11)_{10} = (1011)_2$
  - $(C)_{16} = (12)_{10} = (1100)_2$
  - $(D)_{16} = (13)_{10} = (1101)_2$
  - $(E)_{16} = (14)_{10} = (1110)_2$
  - $(F)_{16} = (15)_{10} = (1111)_2$

Conversion From Binary to Hexadecimal

- Conversion is based on the fact that one hexadecimal digit represents four binary digits
- Conversion
  - Form groups of four bits starting from the binary point
  - Each group becomes a hexadecimal digit
- Example: convert $(111010100.100)_2$ to hex

$$\underbrace{\mathbf{0}\,\mathbf{0}\,\mathbf{0}\,1}_{1}\;\underbrace{1\,1\,0\,1}_{D}\;\underbrace{0\,1\,0\,0}_{4}\,.\,\underbrace{1\,0\,0\,\mathbf{0}}_{8}$$

$$(111010100.100)_2 = (1D4.8)_{16}$$

Conversion From Hexadecimal to Binary

- Conversion
  - Convert each hexadecimal digit to the corresponding four binary digits, including all leading 0’s
- Example: convert $(3F.A)_{16}$ to binary

3 F . A

0011 1111 1010

$$(3F.A)_{16} = (00111111.1010)_2$$

CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can:

- Convert $(BC.F)_{16}$ to binary

CHECK POINT

Answer:

- $(BC.F)_{16}$ = 1011 1100 . 1111 = 10111100.1111_2

If you have any difficulties, please review the lecture video before continuing.

Summary

- Numerical values can be represented using any radix *r*
  - We're used to radix-10 or decimal
  - Radix-2 or binary is better suited to use in computers
  - Octal and hexadecimal are also convenient for notation
- Fixed point representation uses a fixed number of bits to represent values with an implied location for the binary point
  - Trade-off between range and precision
- Conversion between decimal and other radices is needed

MODULE 2: Data Representation

Lecture 2.1
Unsigned Data Representation

Prepared By:
- Scott F. Midkiff, PhD
- Luiz A. DaSilva, PhD
- Kendall E. Giles, PhD

Electrical and Computer Engineering
Virginia Tech