Digital Operations and Computations
Course Notes

Winter 2011
Professor H. Louie
Department of Electrical & Computer Engineering
Seattle University

01-Number Systems
Text: Unit 1

ECEGR/ISSC 201
Digital Operations and Computations

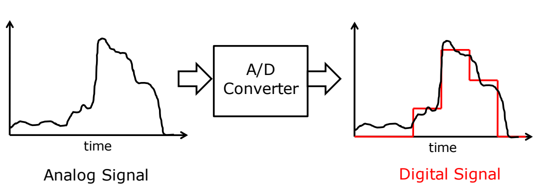
Overview

- Introduction to digital systems
- Decimal system
- Non-decimal systems
- Conversion to decimal
- Conversion from decimal
- Binary Codes

What is a Digital System?

- Digital: something taking on a discrete set of values
  - True, False
  - On, Off
  - -5 Volts, 0 Volts, 5 Volts
- Analog: something that can assume a continuous range of values
  - Temperature
  - 0-5 Volts

What is a Digital System?



Examples of Digital Systems

- Alphabet
- Abacus
- Morse Code
- Computer
- Light Switch




Decimal System

- What are the characteristics of our (decimal) numbering system?
  - Valid digits: 0, 1, 2, …, 9
  - Positional notation
    - 100 is ten times greater than 10, which is ten times greater than 1, which is ten times greater than 0.1, etc.
  - Positive, negative and zero can be represented
- We use a base 10 (decimal) system

Decimal System

- $10^1 = 10$
- $10^2 = 10 \times 10 = 100$
- $10^3 = 10 \times 10 \times 10 = 1000$
- What about:
  - $10^0 = ?$
  - $10^{-1} = ?$
  - $10^{-2} = ?$

Decimal System

- $10^1 = 10$
- $10^2 = 100$
- $10^3 = 1000$
- What about:
  - $10^0 = 1$ (any value to 0^{th} power = 1)
  - $10^{-1} = 1/10 = 0.1$
  - $10^{-2} = 1/10/10 = 1/100 = 0.01$

Example

- Find a_2, a_1, a_0, and a_{-1} (values between 0 and 9) such that:

$13 = a_2 \times 10^2 + a_1 \times 10^1 + a_0 \times 10^0 + a_{-1} \times 10^{-1}$

Example

- Find a_2, a_1, a_0, and a_{-1} such that:

$13 = a_2 \times 10^2 + a_1 \times 10^1 + a_0 \times 10^0 + a_{-1} \times 10^{-1}$

- Solution

$13 = 0 \times 10^2 + 1 \times 10^1 + 3 \times 10^0 + 0 \times 10^{-1}$

013.0

Decimal System

- Another example

$953.7 = 9 \times 10^2 + 5 \times 10^1 + 3 \times 10^0 + 7 \times 10^{-1}$

- Try a negative number

$-205 = -2 \times 10^2 - 0 \times 10^1 - 5 \times 10^0$

Other Numbering Systems

- What other numbering systems do you know of?
  - Binary (base 2)
  - Ternary (base 3)
  - Quintal (base 5)
  - Octal (base 8)
  - Hexadecimal (base 16)
- The base is also known as the *radix*
- Is there a base 1? base 0?
  - No!

Binary

- We are especially interested in binary
  - electronic systems
  - digital logic (true, false)

Binary

- Valid digits: 0, 1 (called bits)
- Some examples
  - $0_2 = 0_{10}$ (subscript refers to the base)
  - $1_2 = 1_{10}$
  - $01_2 = 1_{10}$
  - $10_2 = 2_{10}$
  - $11_2 = 3_{10}$
  - $100_2 = 4_{10}$

Binary

- What is the decimal equivalent of 101_2?
  - Recall that $953 = 9 \times 10^2 + 5 \times 10^1 + 3 \times 10^0$
  - Similarly, $101_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
    $= 4 + 0 + 1 = 5_{10}$

Example

- What is the decimal equivalent of 1011.11_2?

Example

- What is the decimal equivalent of 1011.11_2?
  - $1011.11_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
    $+ 1 \times 2^{-1} + 1 \times 2^{-2}$
    $= 8 + 0 + 2 + 1 + 0.5 + 0.25 = 11.75_{10}$

Binary

- In binary conversion, knowing powers of 2 becomes very important:
  - $2^0 = 1$
  - $2^1 = 2$
  - $2^2 = 4$
  - $2^3 = 8$
  - $2^4 = 16$
  - $2^5 = 32$
  - $2^6 = 64$
  - $2^7 = 128$
  - $2^8 = 256$
  - and so on

Binary

- Negative powers of two:
  - $2^{-1} = 1/2^1 = 1/2 = 0.5$
  - $2^{-2} = 1/2^2 = 1/4 = 0.25$
  - $2^{-3} = 1/2^3 = 1/8 = 0.125$
  - and so on

Conversion

- In general, when positional notation is used:
  - $N = (a_4a_3a_2a_1a_0.a_{-1}a_{-2})_R$
    $= a_4 \times R^4 + a_3 \times R^3 + a_2 \times R^2 + a_1 \times R^1 + a_0 \times R^0 + a_{-1} \times R^{-1} + a_{-2} \times R^{-2}$
  - Where a_i is a digit between 0 and R-1 (inclusive)
  - The result is the decimal equivalent of N

Hexadecimal

- $0_{16} = 0_{10}$ (subscript refers to the base)
- $1_{16} = 1_{10}$
- $2_{16} = 2_{10}$ and so on…
- $9_{16} = 9_{10}$
- $A_{16} = 10_{10}$
- $B_{16} = 11_{10}$ and so on…
- $F_{16} = 15_{10}$

Examples

- Convert the following to decimal
  - 10.11_2
  - 10.11_8
  - 10.11_{16}
  - AF_{16}

Examples

- Convert the following to decimal
  - $10.11_2 = 2.75$
  - $10.11_8 = 8.140625$
  - $10.11_{16} = 16.06640625$
  - $AF_{16} = 175$

Conversion from Decimal

- Convert 53_{10} to Binary
- Cannot directly apply the previous method
- First consider the case of a decimal integer

Conversion of Decimal Integers by Division

- Consider an integer
- $N = (a_n a_{n-1} \ldots a_2 a_1 a_0)_R$
  $= a_n R^n + a_{n-1} R^{n-1} + \ldots + a_1 R^1 + a_0$
- Dividing by R yields
  $$\frac{N}{R} = a_n R^{n-1} + a_{n-1} R^{n-2} + \ldots + a_2 R^1 + a_1 = Q_1$$
  - where the remainder is a_0

Conversion of Decimal Integers by Division

- Next divide the quotient Q_1 by R
  $$\frac{Q_1}{R} = a_n R^{n-2} + a_{n-1} R^{n-3} \ldots a_3 R^1 + a_2 = Q_2$$
  - where the remainder is a_1
- The process repeats until a_n is left

Conversion of Decimal Integers by Division

- Convert 53_{10} to Binary
- R = 2
  $2\overline{)53}$ quotient 26 remainder 1
  - $a_0 = 1$

  $2\overline{)26}$ quotient 13 remainder 0
  - $a_1 = 0$

  $2\overline{)13}$ quotient 6 remainder 1
  - $a_2 = 1$

  $2\overline{)6}$ quotient 3 remainder 0
  - $a_3 = 0$
  - $a_4 = 1$
  - $a_5 = 1$

Conversion of Decimal Integers by Division

- Putting the digits together
  $53_{10} = 110101_2$
- Now try converting 53_{10} to hexadecimal (base 16)

Conversion of Decimal Integers by Division

- Putting the digits together
  $53_{10} = 110101_2$
- Now try converting 53_{10} to hexadecimal (base 16)

  $16\overline{)53}$ quotient 3 remainder $5 = a_0$

  $16\overline{)3}$ quotient 0 remainder $3 = a_1$

- $53_{10} = 35_{16}$

Conversion of Decimal Fractions by Division

- $F = (.a_{-1}a_{-2} \ldots a_{-m})_R$
  $= a_{-1}R^{-1} + a_{-2}R^{n-2} + \ldots + a_{-m}R^{-m}$
- We know what R (base converting to) and F (decimal number) are
- Need to isolate a_{-1}

Conversion of Decimal Fractions by Division

- $F = (.a_{-1}a_{-2} \ldots a_{-m})_R$
  $= a_{-1}R^{-1} + a_{-2}R^{n-2} + \ldots + a_{-m}R^{-m}$
- We know what R (base converting to) and F (decimal number) are
- Need to isolate a_{-1}
- Multiply by R
  $FR = a_{-1} + a_{-2}R^{-1} + a_{-3}R^{-2} \ldots + a_{-m}R^{-m+1} = a_{-1} + F_1$
- Where a_{-1} is the integer part of the result and F_1 is the fractional part

Conversion of Decimal Fractions by Division

- Repeat by multiplying F_1 with R
  $F_1R = a_{-2} + a_{-3}R^{-2} \ldots + a_{-m}R^{-m+2} = a_{-2} + F_2$
- The process may not terminate, in which case the result is a repeating fraction

Conversion of Decimal Fractions by Division

- Convert 0.75_{10} to binary
  - R = 2, F = 0.75

Conversion of Decimal Fractions by Division

- Convert 0.75_{10} to binary
  - R = 2, F = 0.75
    $FR = a_{-1} + a_{-2}R^{-1} + a_{-3}R^{-2} \ldots + a_{-m}R^{-m+1} = a_{-1} + F_1$
    $1.5 = a_{-1} + F_1$
    $=> a_{-1} = 1$
    $=> F_1 = 0.5$

Conversion of Decimal Fractions by Division

- Convert 0.75_{10} to binary
  - R = 2, F = 0.75
    $FR = a_{-1} + a_{-2}R^{-1} + a_{-3}R^{-2} \ldots + a_{-m}R^{-m+1} = a_{-1} + F_1$
    $1.5 = a_{-1} + F_1$
    $=> a_{-1} = 1$
    $=> F_1 = 0.5$
- Next step
  $F_1R = a_{-2} + a_{-3}R^{-1} + \ldots + a_{-m}R^{-m+2} = a_{-2} + F_2$
  $1 = a_{-2} + F_2$
  $=> a_{-2} = 1$
  $=> F_2 = 0$
- stop

Conversion of Decimal Fractions by Division

- Putting it together:
  $0.75_{10} = 0.11_2$

Example

- Now try converting 0.70_{10} to binary

Example

- Now try converting 0.70_{10} to binary
  1) $RF=1.4=a_{-1}+F_1$
  $=>a_{-1}=1$
  $=>F_1=0.4$
  2) $RF_1=0.8=a_{-2}+F_2$
  $=>a_{-2}=0$
  $=>F_2=0.8$
  3) $RF_1=1.6=a_{-3}+F_3$
  $=>a_{-3}=1$
  $=>F_3=0.6$
  4) $RF_3=1.2=a_{-4}+F_4$
  $=>a_{-4}=1$
  $=>F_4=0.2$
  5) $RF_1=0.4=a_{-5}+F_5$
  $=>a_{-5}=0$
  $=>F_5=0.4$
  6) $RF_5=0.8=a_{-6}+F_6$
  $=>a_{-6}=0$
  $=>F_6=0.8$
- Repeating result: 0.1 0110 0

Conversion of Decimal Numbers

- If converting 11.75, do the integer division method for 11, and the decimal method for .75 and combine the result
- It is possible to use either method when converting from bases other than decimal, but this is difficult because it involves non-decimal based math
- Better idea is to convert to decimal, then convert to the desired base

Binary to Hexadecimal Conversion

- A few shortcuts exist
- Consider 1001101.010111_2 in hexadecimal
- Regroup as 0100 1101 . 0101 1100
- Convert each word
  - 0100 1101 . 0101 1100

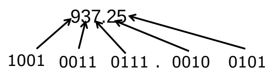
    4 D 5 C
- The result is $4D.5C_{16}$

Binary Codes

- Binary codes are a way of representing decimal numbers that are intended to be easier than converting the entire decimal number to binary

Binary Codes

- Binary codes are a way of representing decimal numbers that are intended to be easier than converting the entire decimal number to binary
- Example:



- This is known as 8-4-2-1 BCD (Binary Coded Decimal)

Binary Codes

Decimal Digit	8-4-2-1 Code	6-3-1-1 Code	Gray Code
0	0000	0000	0000
1	0001	0001	0001
2	0010	0011	0011
3	0011	0100	0010
4	0100	0101	0110
5	0101	0111	1110
6	0110	1000	1010
7	0111	1001	1011
8	1000	1011	1001
9	1001	1100	1000

Binary Codes

- 8-4-2-1 and 6-3-1-1 are weighted codes
- $N = w_3a_3 + w_2a_2 + w_1a_1 + w_0a_0$
- Decimal 8 in 8-4-2-1 is: 1000
  - 8(1) + 4(0) + 2(0) + 1(0)=8
- In 6-3-1-1 it is: 1011
  - 6(1) + 3(0) + 1(1) + 1(1)=8

Binary Codes

- Gray Code: only one digit changes
  - Useful in analog to digital conversion
  - More reliable than if two digits change (it is unlikely that two or more switches would occur simultaneously)

Gray Code
0000
0001
0011
0010
0110
1110
1010
1011
1001
1000

ASCII Code

- ASCII (American Standard Code for Information Interchange)
- Alphanumeric code
- 7-bits =>128 symbols
- See text page 22 for a partial list

Bits and Bytes

- In binary, the digits are known as bits (binary-digits)
- Collection of 8 bits is commonly known as a byte (or more specifically, as a octet)
  - Able to represent keyboard characters

Bits and Bytes

- kilobyte: 2^{10} bytes (1,024)
- megabyte: 2^{20} bytes (1,048,576)
- gigabyte: 2^{30} bytes (1,073,741,824)