Number Systems

Reading:

Chapter 3 – Number Systems (except 3.7)

Base of a Number System

- *Base*: the number of different digits including zero in the number system
  - Example: Base 10 has 10 digits, 0 through 9
- Decimal or base 10 number system
  - Origin: counting on the fingers
  - “Digit” from the Latin word *digitus* meaning “finger”
- *Binary* or *base 2*
  - *Bit* (binary digit): 2 digits, 0 and 1
- *Octal* or *base 8:* 8 digits, 0 through 7
- *Hexadecimal or base 16:*
  16 digits, 0-9, followed by A-F
- Examples: $\mathbf{10}_{10} = \mathbf{A}_{\mathbf{16}}$; $\mathbf{11}_{10} = \mathbf{B}_{\mathbf{16}}$

Sometimes a number is shown with the base of the system

List consecutive numbers in base 2, base 10, and base 16

Base or Radix

- Base:
  - The number of different symbols required to represent any given number
- The *larger* the base, the *more* numerals are required
  - Base 10: 0,1, 2,3,4,5,6,7,8,9
  - Base 2: 0,1
  - Base 8: 0,1,2, 3,4,5,6,7
  - Base 16: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Why Binary?

- Early computer design was decimal
  - Mark I and ENIAC
- John von Neumann proposed binary data processing (1945)
  - Simplified computer design
  - Used for both instructions and data
- Natural relationship between on/off switches and calculation using Boolean logic

	
On	Off
True	False
Yes	No
1	0

Keeping Track of the Bits

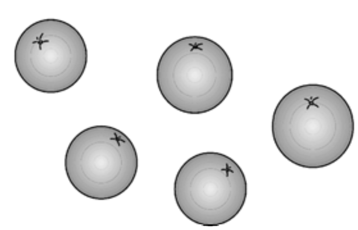
- Bits commonly stored and manipulated in groups
  - 8 bits = 1 *byte*
  - 4 bytes = 1 word (in many systems)
- Number of bits used in calculations
  - Affects accuracy of results
  - Limits size of numbers manipulated by the computer

Consider some primitive types used in Java: short, int, and long

3-5

Numbers: Physical Representation

- Different numerals, same number of oranges
  - Cave dweller: IIIII
  - Roman: V
  - Arabic: 5
- Different bases, same number of oranges
  - 5_{10}
  - 101_2
  - 12_3



3-6

Number System

- Roman: position *independent*
- Modern: based on positional notation (place value)
  - Decimal system: system of **positional** notation based on powers of 10.
  - Binary system: system of **positional** notation based powers of 2
  - Octal system: system of **positional** notation based on powers of 8
  - Hexadecimal system: system of **positional** notation based powers of 16

Positional Notation: Base 10

$527 = 5 \times 10^2 + 2 \times 10^1 + 7 \times 10^0$

100's place 10's place 1's place

Place	10^2	10^1	10^0
Value	100	10	1
Evaluate	5 x 100	2 x 10	7 x1
Sum	500	20	7

Positional Notation: Octal

624_8 = 404_{10}

64's place | 8's place | 1's place

Place	8^2	8^1	8^0
Value	64	8	1
Evaluate	6 x 64	2 x 8	4 x 1
Sum for Base 10	384	16	4

Positional Notation: Hexadecimal

6704_{16} = $26{,}372_{10}$

4,096's place | 256's place | 16's place | 1's place

Place	16^3	16^2	16^1	16^0
Value	4,096	256	16	1
Evaluate	6 x 4,096	7 x 256	0 x 16	4 x 1
Sum for Base 10	24,576	1,792	0	4

In Web color notation, #0000FF is pure blue

Positional Notation: Binary

1101 0110$_2$ = 214_{10}

Place	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Value	128	64	32	16	8	4	2	1
Evaluate	1 x 128	1 x 64	0 x 32	1 x16	0 x 8	1 x 4	1 x 2	0 x 1
Sum for Base 10	128	64	0	16	0	4	2	0

Range of Possible Numbers

- R = B^K where
  - R = range
  - B = base
  - K = number of digits
- Example #1: Base 10, 2 digits
  - R = $10^{2\ =}$ 100 different numbers (0…99)
- Example #2: Base 2, 16 digits
  - R = 2^{16} = 65,536 or 64K
  - 16-bit PC can store 65,536 different number values

How many colors can be represented in a 24-bit RGB color scheme? Each base color (r, g, and b) uses 8 bits.

Decimal Range for Bit Widths

Bits	Digits	Range
1	0+	2 (0 and 1)
4	1+	16 (0 to 15)
8	2+	256
10	3	1,024 (1K)
16	4+	65,536 (64K)
20	6	1,048,576 (1M)
32	9+	4,294,967,296 (4G)
64	19+	Approx. 1.6×10^{19}
128	38+	Approx. 2.6×10^{38}

3-13

Counting in Base 2

Binary Number	Equivalent				Decimal Number
	8's (2^3)	4's (2^2)	2's (2^1)	1's (2^0)	
0				0×2^0	0
1				1×2^0	1
10			1×2^1	0×2^0	2
11			1×2^1	1×2^0	3
100		1×2^2			4
101		1×2^2		1×2^0	5
110		1×2^2	1×2^1		6
111		1×2^2	1×2^1	1×2^0	7
1000	1×2^3				8
1001	1×2^3			1×2^0	9
1010	1×2^3		1×2^1		10

3-14

Binary Arithmetic

	1	1	1	1	1			
		1	1	0	1	1	0	1
+				1	0	1	1	0
	1	0	0	0	0	0	1	1

3-15

Boolean Logic

- System for logical operations
- Named after George Boole
- Components
  - Elements as members of a set
  - Operations (not, and, or, exclusive or)
  - Subsets and supersets
- Set membership and boolean operations are often implemented with binary numbers and operations

3-16

Binary Arithmetic: Boolean Logic

- Boolean logic without performing arithmetic
  - *EXCLUSIVE-OR*
    - Output is “1” only if either input, but ***not*** both inputs, is a “1”
  - *AND*
    - Output is “1” if and only both inputs are a “1”

	1	**1**	**0**	**1**	**1**	**0**	**1**
and			**1**	**0**	**1**	**1**	**0**
	0	**0**	**0**	**0**	**1**	**0**	**0**

Sometimes referred to as a masking operation

Converting from Base 10

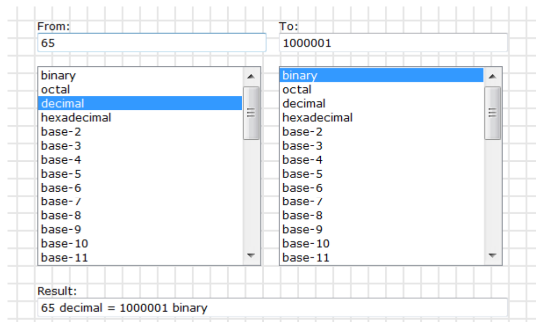
- Powers Table

Power / Base	8	7	6	5	4	3	2	1	0
2	256	128	64	32	16	8	4	2	1
8				32,768	4,096	512	64	8	1
16					65,536	4,096	256	16	1

Conversion Calculator

- To quickly convert numbers, use

www.translatorscafe.com/cafe/units-converter/numbers/calculator/



 3-19

From Base 16 to Base 2

- The nibble approach
  - Hex easier to read and write than binary

Base 16	1	F	6	7
Base 2	0001	1111	0110	0111

  - Why hexadecimal?
    - Modern computer operating systems and networks present variety of troubleshooting data in hex format

 3-20