C335
Computer Structures

Number Representation
Part #1

Dr. Liqiang Zhang

Department of Computer and Information Sciences

Adapted from Morgan Kaufmann and others

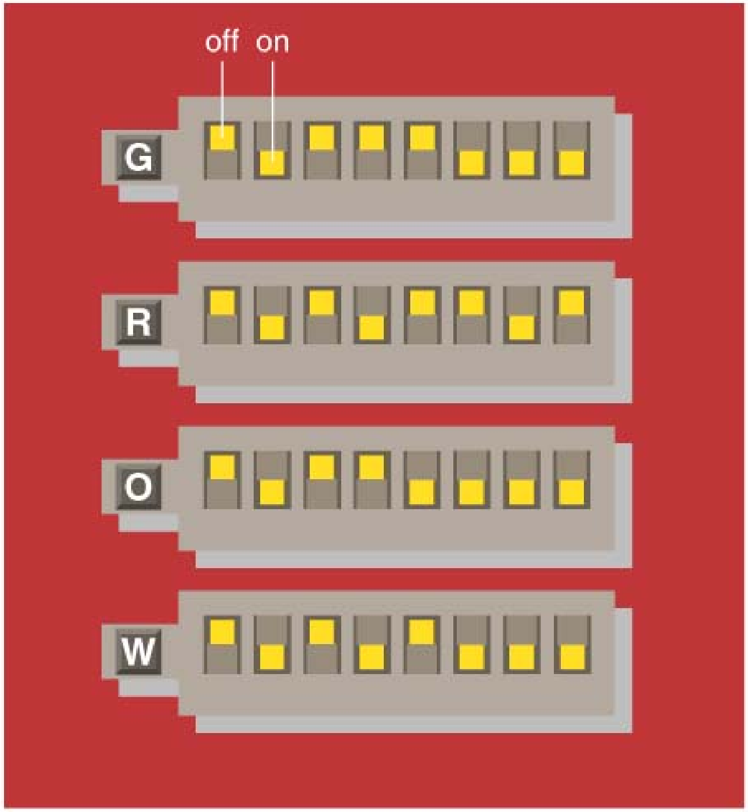
Outline

- Part 1: representation of unsigned numbers
  - Binary and hexadecimal
- Part 2: representation of signed numbers
  - 1’s complement
  - 2’s complement

Recall: The Binary (Numbering) System

The Binary System: Using On/Off Electrical States to Represent Data & Instructions

- The binary system has only two symbols - 0 and 1
- Bit - binary digit
- Byte - group of 8 bits used to represent one character, digit, or other value



Arithmetic with Binary Numbers

Binary Arithmetic for Whole Numbers (Integers)
(Counting Begins with 0, not 1)

Integer	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000

"There are 10 kinds of people—
those who understand binary and those who don't"

Decimal and Binary representation

- 257 (decimal)
  - $(2*10^2)+(5*10^1)+(7*10^0)$

position

base

digit

- 1100 (binary)
  - $(1*2^3)+(1*2^2)+(0*2^1)+(0*2^0)$
- Positional notation (base 10 or 2)

Numbers: positional notation

- Number Base B $\Rightarrow$ B symbols per digit:
  - Base 10 (Decimal): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
    Base 2 (Binary): 0, 1
- Number representation:
  - $d_{31}d_{30} \ldots d_1d_0$ is a 32 digit number
  - value = $d_{31} \times B^{31} + d_{30} \times B^{30} + \ldots + d_1 \times B^1 + d_0 \times B^0$
- Binary: 0,1 (In binary digits called "bits")
  - 0b11010 $= 1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0$
    $= 16 + 8 + 2$
    $= 26$
  - Here 5 digit binary # turns into a 2 digit decimal #

Convert Binary to Decimal

- 110001 (binary) → decimal?
  - 1 1 0 0 0 1
  - 2^5 2^4 2^3 2^2 2^1 2^0
  - $1*2^5 + 1*2^4 + 0*2^3 + 0*2^2 + 0*2^1 + 1*2^0$
  - 32 + 16 + 0 + 0 + 0 + 1
  - 49 (decimal)

Exercise

- Convert the following binary numbers to decimal
  - 100111
  - 1101101

Convert Decimal to Binary

- 49 (decimal) → binary?

  1 | 4 9
  0 | 2 4
  0 | 1 2
  0 | 6
  1 | 3
  1

- 110001 (binary)

Exercise

- Convert the following decimal numbers to binary
  - 25
  - 106

Can we find a base that converts to binary easily?

Hexadecimal Numbers: Base 16

4 Bits (Base 2)*	Decimal (Base 10)	Hexadecimal (Base 16) Symbol
0000	0	**0 hex**
0001	1	1 hex
0010	2	2 hex
0011	3	3 hex
0100	4	4 hex
0101	5	5 hex
0110	6	6 hex
0111	7	7 hex

Begin Counting at Zero

- With 4 bits, there are 2^4=16 possible symbols.

Hexadecimal Numbers: Base 16

4 Bits (Base 2)	Decimal (Base 10)	Hexadecimal (Base 16) Symbol
1000	8	8 hex
1001	9	9 hex
1010	10	**A hex**
1011	11	**B hex**
1100	12	**C hex**
1101	13	**D hex**
1110	14	**E hex**
1111	15	**F hex**

After 9, Count A Through F

Hexadecimal Numbers: Base 16

- Hexadecimal:
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
  - Normal digits + 6 more from the alphabet
  - Often written as 0x… (e.g., 0xFAB5)
- Conversion: Binary⇔Hex
  - 1 hex digit represents 16 decimal values
  - 4 binary digits represent 16 decimal values
  - ⇒1 hex digit replaces 4 binary digits
- One hex digit is a "nibble". Two is a "byte"
- Example:
  - 1010 1011 0111 (binary) = 0x______ ?

Decimal vs. Hexadecimal vs. Binary

Examples:

1010 1100 0011 (binary)

=

10111 (binary)

= 0001 0111 (binary)

=

0x3F9

=

00	0	0000
01	1	0001
02	2	0010
03	3	0011
04	4	0100
05	5	0101
06	6	0110
07	7	0111
08	8	1000
09	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Which base do we use?

- Decimal: great for humans, especially when doing arithmetic
- Binary: what computers use;
  - To a computer, numbers always binary no matter how we write them:
  - 32_{ten} == 32_{10} == 0x20 == 100000_2 == 0b100000
  - Will learn how computers do +, -, *, /
  - And design circuits to do them
- Hex?
  - Easy to convert from binary to hex, and vice versa
  - Shorter then binary, making programs more succinct
  - Great for programmers!

Binary Encoding for Alternatives

Bits	Alternatives	Examples
1	$2^1=2$	Male = 0, Female = 1
2	$2^2=4$	Spring = 00, Summer = 01, Autumn = 10, Winter = 11
8	$2^8=256$	Keyboard characters for U.S. keyboards. Space=00000000, etc. ASCII code

Binary Encoding for Alternatives

Encoding Alternatives
(Product number, region, gender, etc.)
(N bits can represent 2^N Alternatives)

Number of Bits In Field (N)	**Number of Alternatives That Can be Encoded with N bits**
1	2 (2^1)
2	4 (2^2)
3	8 (2^3)
4	16 (2^4)
8	256 (2^8)
16	65,536 (2^{16})
…	…

Each added bit doubles the number of alternatives that can be represented

BIG IDEA: Bits can represent anything!!

- Characters?
  - ASCII and Unicode
- Text?
- Logical values?
  - 0 $\Rightarrow$ False, 1 $\Rightarrow$ True
- Colors ?
- Images?
- Audio?
- Locations / addresses? commands?

- MEMORIZE: N bits $\Leftrightarrow$ at most 2^N things

What to do with representations of numbers?

- Just what we do with numbers!
  - Add them
  - Subtract them
  - Multiply them
  - Divide them
  - Compare them

	1	**0**	**1**	**0**
+	**0**	**1**	**1**	**1**
1	**0**	**0**	**0**	**1**

- Example: 10 + 7 = 17
  - …so simple to add in binary that we can build circuits to do it!
  - subtraction just as you would in decimal
  - Comparison: How do you tell if X > Y ?