

Lecture 2: Number Systems

◆ Logistics

- Webpage is up! <http://www.cs.washington.edu/370>
- HW1 is posted on the web in the calendar --- due 10/1 10:30am
- Third TA: Tony Chick chickt@cs.washington.edu
- Email list: please sign up on the web.
- Lab1 starts next week: sections MTW --- show up to pick up your lab kit

◆ Last lecture

- Class introduction and overview

◆ Today

- Binary numbers
- Base conversion
- Number systems
 - ✦ Twos-complement
- A/D and D/A conversion

CSE370, Lecture 2

CSE 370 – AUTUMN 2008

YOKY MATSUOKA

WITH VINCE ZANELLA AND BRIAN DELLON



Organization:

[Lecture Times and Office Hours](#)

[Textbook](#)

[Academic Accommodations](#)

Coursework:

[Course Goals and Syllabus](#)

[Course Structure, Policies and Guidelines](#)

[Calendar](#)

Software Tools:

[Computing Labs and Tools](#)

[Active-HDL Tutorials](#)

WELCOME TO THE AUTUMN 2008 EDITION OF

CSE 370 - INTRODUCTION TO DIGITAL DESIGN

There are 10 kinds of people in the world. Those who understand binary... and those who don't.

This page contains essential (and useful) information for the class. Keep in mind that this document is not static, and that new information will be added over the entire quarter. Make sure to check the class e-mail archive frequently. Some links may be inactive until later in the quarter. If you have any problems with this document or the CSE 370 web, please send mail to the course webmaster so that we may address the issue or solve the problem. You can also send Mail about anything related to the course to the instructor and/or the TAs, or see us during office hours.

The Calendar page has links to all course materials including Lecture Slides, Homework and Solutions.

To get in touch with course staff, add yourself to the class email list: [click here](#)

Portions of the CSE 370 Web may be reprinted or adapted for academic nonprofit purposes, providing the source is accurately quoted and duly credited.

The CSE 370 Web: ©1996-2008, Department of C&SE, University of Washington.

CSE370, Lecture 2

The "WHY" slide

- ◆ Binary numbers
 - All computers work with 0's and 1's so it is like learning alphabets before learning English
- ◆ Base conversion
 - For convenience, people use other bases (like decimal, hexadecimal) and we need to know how to convert from one to another.
- ◆ Number systems
 - There are more than one way to express a number in binary. So 1010 could be -2, -5 or -6 and need to know which one.
- ◆ A/D and D/A conversion
 - Real world signals come in continuous/analog format and it is good to know generally how they become 0's and 1's (and visa versa).

CSE370, Lecture 2

Digital

- ◆ Digital = discrete
 - Binary codes (example: BCD)
 - Decimal digits 0-9
- ◆ Binary codes
 - Represent symbols using binary digits (bits)
- ◆ Digital computers:
 - I/O is digital
 - ↪ ASCII, decimal, etc.
 - Internal representation is binary
 - ↪ Process information in bits

<u>Decimal Symbols</u>	<u>BCD Code</u>
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

CSE370, Lecture 2

The basics: Binary numbers

◆ Bases we will use

- Binary: Base 2
- Octal: Base 8
- Decimal: Base 10
- Hexadecimal: Base 16

◆ Positional number system

- $101_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
- $63_8 = 6 \times 8^1 + 3 \times 8^0$
- $A1_{16} = 10 \times 16^1 + 1 \times 16^0$

◆ Addition and subtraction

$$\begin{array}{r} 1011 \\ + 1010 \\ \hline 10101 \end{array} \qquad \begin{array}{r} 1011 \\ - 0110 \\ \hline 0101 \end{array}$$

CSE370, Lecture 2

Binary → hex/decimal/octal conversion

◆ Conversion from binary to octal/hex

- Binary: 10011110001
- Octal: 10 | 011 | 110 | 001 = 2361_8
- Hex: 100 | 1111 | 0001 = $4F1_{16}$

◆ Conversion from binary to decimal

- $101_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 5_{10}$
- $63.4_8 = 6 \times 8^1 + 3 \times 8^0 + 4 \times 8^{-1} = 51.5_{10}$
- $A1_{16} = 10 \times 16^1 + 1 \times 16^0 = 161_{10}$

CSE370, Lecture 2

Decimal→ binary/octal/hex conversion

<u>Binary</u>			<u>Octal</u>		
	<u>Quotient</u>	<u>Remainder</u>		<u>Quotient</u>	<u>Remainder</u>
56÷2=	28	0	56÷8=	7	0
28÷2=	14	0	7÷8=	0	7
14÷2=	7	0			
7÷2=	3	1			
3÷2=	1	1			
1÷2=	0	1			
			56 ₁₀ =111000 ₂		
			56 ₁₀ =70 ₈		

- ◆ Why does this work?
 - $N=56_{10}=111000_2$
 - $Q=N/2=56/2=111000/2=11100$ remainder 0
- ◆ Each successive divide liberates an LSB (least significant bit)

CSE370, Lecture 2

Number systems

- ◆ How do we write negative binary numbers?
- ◆ Historically: 3 approaches
 - Sign-and-magnitude
 - Ones-complement
 - Twos-complement
- ◆ For all 3, the most-significant bit (MSB) is the sign digit
 - 0 ≡ positive
 - 1 ≡ negative
- ◆ twos-complement is the important one
 - Simplifies arithmetic
 - Used almost universally

CSE370, Lecture 2

Sign-and-magnitude

- ◆ The most-significant bit (MSB) is the sign digit
 - 0 ≡ positive
 - 1 ≡ negative
- ◆ The remaining bits are the number's magnitude
- ◆ Problem 1: Two representations for zero
 - 0 = 0000 and also -0 = 1000
- ◆ Problem 2: Arithmetic is cumbersome

Add		Subtract			Compare and subtract		
4	0100	4	0100	0100	- 4	1100	1100
+ 3	+ 0011	- 3	+ 1011	- 0011	+ 3	+ 0011	- 0011
= 7	= 0111	= 1	≠ 1111	= 0001	- 1	≠ 1111	= 1001

CSE370, Lecture 2

Ones-complement

- ◆ Negative number: Bitwise complement positive number
 - 0011 ≡ 3₁₀
 - 1100 ≡ -3₁₀
- ◆ Solves the arithmetic problem

Add		Invert, add, add carry		Invert and add	
4	0100	4	0100	- 4	1011
+ 3	+ 0011	- 3	+ 1100	+ 3	+ 0011
= 7	= 0111	= 1	1 0000	- 1	1110
		add carry:	+1		
			= 0001		

- ◆ Remaining problem: Two representations for zero
 - 0 = 0000 and also -0 = 1111

CSE370, Lecture 2

Twos-complement

◆ Negative number: Bitwise complement **plus one**

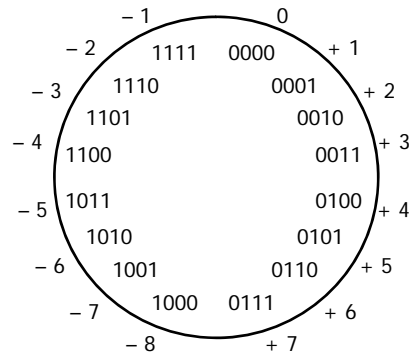
- $0011 \equiv 3_{10}$
- $1101 \equiv -3_{10}$

◆ Number wheel

◆ Only one zero!

◆ MSB is the sign digit

- $0 \equiv$ positive
- $1 \equiv$ negative



CSE370, Lecture 2

Twos-complement (con't)

◆ Complementing a complement \Leftrightarrow the original number

◆ Arithmetic is easy

- Subtraction = negation and addition
- Easy to implement in hardware

Add		Invert and add		Invert and add	
4	0100	4	0100	-4	1100
+ 3	+ 0011	- 3	+ 1101	+ 3	+ 0011
= 7	= 0111	= 1	1 0001	- 1	1111
		drop carry	= 0001		

CSE370, Lecture 2

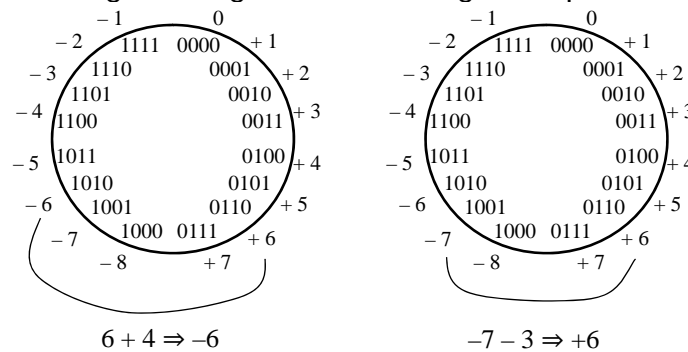
Miscellaneous

- ◆ Twos-complement of non-integers
 - $1.6875_{10} = 01.1011_2$
 - $-1.6875_{10} = 10.0101_2$
- ◆ Sign extension
 - Write +6 and -6 as twos complement
 - ✎ 0110 and 1010
 - Sign extend to 8-bit bytes
 - ✎ 00000110 and 11111010
- ◆ Can't infer a representation from a number
 - 11001 is 25 (unsigned)
 - 11001 is -9 (sign magnitude)
 - 11001 is -6 (ones complement)
 - 11001 is -7 (twos complement)

CSE370, Lecture 2

Twos-complement overflow

- ◆ Summing two positive numbers gives a negative result
- ◆ Summing two negative numbers gives a positive result



- ◆ Make sure to have enough bits to handle overflow

CSE370, Lecture 2

Gray and BCD codes

<u>Decimal Symbols</u>	<u>Gray Code</u>
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101

<u>Decimal Symbols</u>	<u>BCD Code</u>
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

CSE370, Lecture 2

The physical world is analog

- ◆ Digital systems need to
 - Measure analog quantities
 - ✦ Speech waveforms, etc
 - Control analog systems
 - ✦ Drive motors, etc
- ◆ How do we connect the analog and digital domains?
 - Analog-to-digital converter (A/D)
 - ✦ Example: CD recording
 - Digital-to-analog converter (D/A)
 - ✦ Example: CD playback

CSE370, Lecture 2

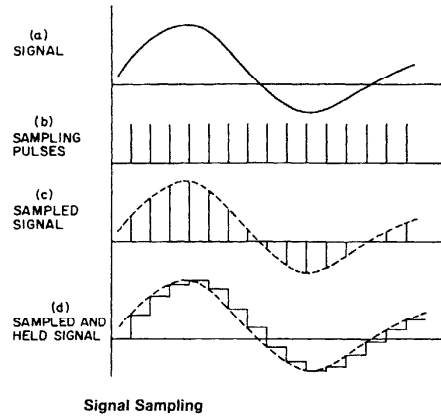
Sampling

◆ **Quantization**

- Conversion from analog to discrete values

◆ **Quantizing a signal**

- We sample it



Signal Sampling

Datel Data Acquisition and Conversion Handbook

CSE370, Lecture 2

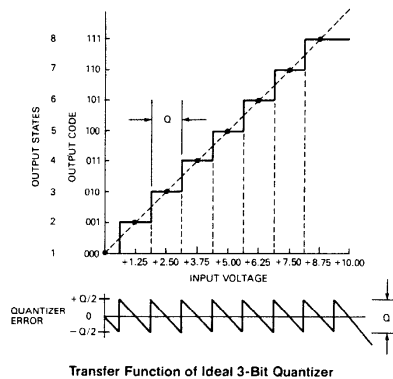
Conversion

◆ **Encoding**

- Assigning a digital word to each discrete value

◆ **Encoding a quantized signal**

- Encode the samples
- Typically Gray or binary codes



Transfer Function of Ideal 3-Bit Quantizer

Datel Data Acquisition and Conversion Handbook

CSE370, Lecture 2