Number systems and logic

- **What is the decimal based number system**
- **How does the binary number system work**
- **Converting between decimal and binary**
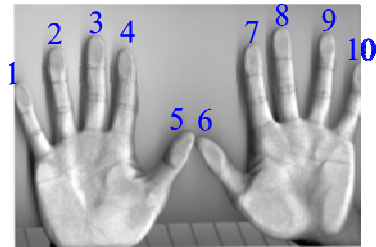- **Storing negative and real numbers**
- **Common logical operations**

What Is Decimal?

- Base 10
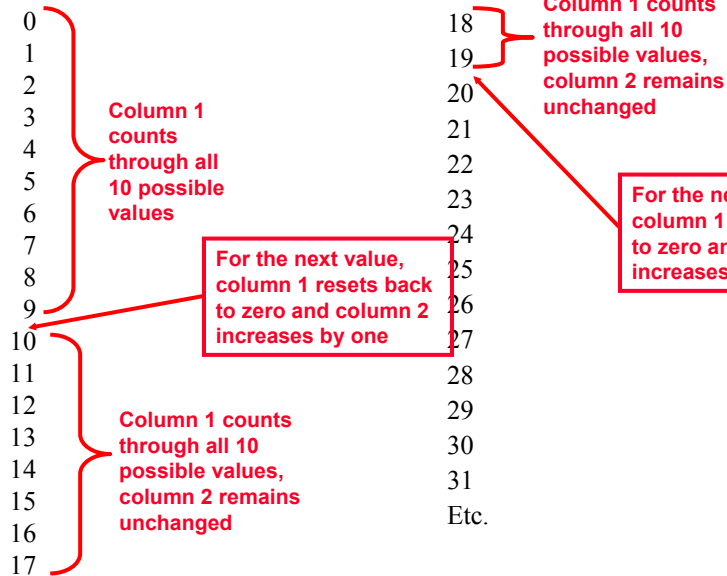 - 10 unique symbols are used to represent values

0
1
2
3
4
5
6
7
8
9
10
:

The number of digits is based on…the number of digits

How Does Decimal Work?

0
1
2
3
4
5
6
7
8
9

Column 1 counts through all 10 possible values

10
11
12
13
14
15
16
17

Column 1 counts through all 10 possible values, column 2 remains unchanged

For the next value, column 1 resets back to zero and column 2 increases by one

18
19
20
21
22
23
24
25
26
27
28
29
30
31
Etc.

Column 1 counts through all 10 possible values, column 2 remains unchanged

For the next value, column 1 resets back to zero and column 2 increases by one

James Tam

Decimal

• Base ten

• Employs ten unique symbols (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

• Each digit can only take on a value from 0 – 9
  – Once a column has traversed all ten values then that column resets back to zero (as does it's right hand neighbours) and the column to it's immediate left increases by one.

James Tam

Recall: Computers Don't Do Decimal!

- Most parts of the computer work in a discrete state:
  - On/off
  - True/false
  - Yes/No
  - Pitted / Smooth

- These two states can be modeled with the binary number system

Binary

- Base two

- Employs two unique symbols (0 and 1)

- Each digit can only take on the value 0 or the value 1
  - Once a column has traversed both values then that column resets back to zero (as does it's right hand neighbours) and the column to it's immediate left increases by one.
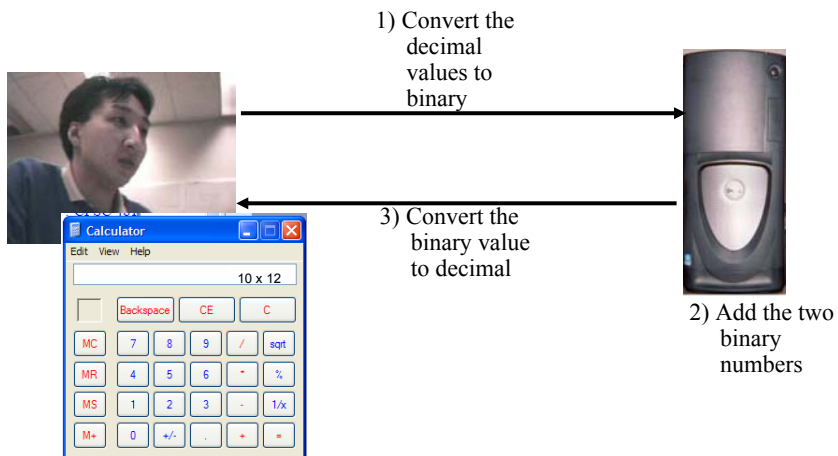
Counting In Binary

Decimal value	Binary value	Decimal value	Binary value
0	0000	8	1000
1	0001	9	1001
2	0010	10	1010
3	0011	11	1011
4	0100	12	1100
5	0101	13	1101
6	0110	14	1110
7	0111	15	1111

Computers Don't Do Decimal!

1) Convert the decimal values to binary

3) Convert the binary value to decimal

Calculator
Edit View Help

10 x 12

	Backspace	CE	C
MC	7	8	9	/	sqrt
MR	4	5	6	*	%
MS	1	2	3	-	1/x
M+	0	+/-	.	+	=

2) Add the two binary numbers

Converting From Decimal To Binary

- Split up the integer and the fractional portions

1) For the integer portion:
   a. Divide the integer portion of the decimal number by two.
   b. The remainder becomes the first integer digit of the binary number (immediately left of the decimal).
   c. The quotient becomes the new integer value.
   d. Divide the new integer value by the target base.
   e. The new remainder becomes the second integer digit of the binary number (second digit to the left of the decimal).
   f. Continue dividing until the quotient is less than two (i.e., it's zero or one) and this quotient becomes the last integer digit of the binary number.

Converting From Decimal To Binary (2)

2) For the fractional portion:
   a. Multiply by two.
   b. The integer portion (if any) of the product becomes the first rational digit of the binary number (first digit to the right of the decimal).
   c. The rational portion of the product is then multiplied by two.
   d. The integer portion (if any) of the new product becomes the second rational digit of the binary number (second digit to the right of the decimal).
   e. Keep multiplying by two base until:
      a. either the resulting product equals zero,
      b. or you have the desired number of places of precision.

Converting From Decimal To Binary (3)

•e.g., 9_{10} to $???_2$ $1\ 0\ 0\ 1\ _2$

9 / 2: q = 4 r = 1

4 / 2: q = 2 r = 0

2/2: q = 1 r = 0

1
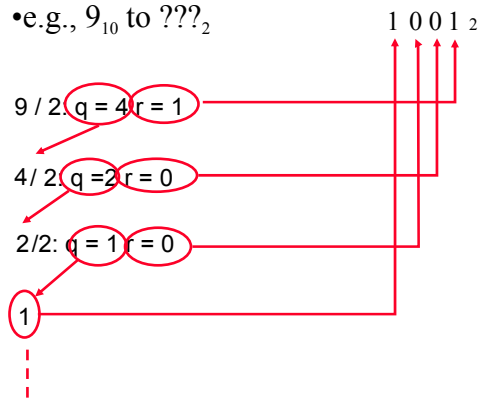
Stop dividing! (quotient is less than two)

Other Example Conversions: Decimal To Binary

Decimal value	Binary value (calculate to a maximum of four fractional digits)
0.5	???
0.1	???
35.25	???

Converting From Binary To Decimal

- Evaluate the expression: two raised to some exponent[1] multiply the resulting expression by the corresponding digit and sum the resulting products.

- **Example:**

  1 0 -1 ←———— Position of digits

  $1\ 1.\ 0_2$ ←———— Number to be converted

  Value in decimal = $(1 \times 2^1) + (1 \times 2^0) + (0 \times 2^{-1}) = (1 \times 2) + (1 \times 1) + 0 = 3$

- **General formula:**

  3 2 1 0 -1 -2 -3 ←———— Position of digits

  d7 d6 d5 d4. d3 d2 d1$_b$ ←———— Number to be converted

  Value in decimal = $(\text{digit7} \times 2^3) + (\text{digit6} \times 2^2) + (\text{digit5} \times 2^1) + (\text{digit4} \times 2^0) +$
  $(\text{digit3} \times 2^{-1}) + (\text{digit2} \times 2^{-2}) + (\text{digit1} \times 2^{-3})$

1 The value of this exponent will be determined by the position of the digit (superscript)

Other Example Conversions: Binary To Decimal

Binary value	Decimal value
0.1	???
0.01	???
10000	???
01111	???
10001	???

Representing Negative Numbers On The Computer

•**What you know:**
– With regular binary all the bits are used to indicate the size of the number
e.g., $100_2 = 4_{10}$
– This representation is *unsigned* because the bit pattern does not provide
information about the sign of the number.

•**What you will learn:**
– How the pattern of bits can be used to represent the sign of the number.

Signed Representations

•The first bit on the left (also referred to as the 'most significant
bit' or the 'sign bit') is used to represent the sign of the number.
– If this bit is zero then the number is positive
– If this bit is one then the number is negative

•The remaining bits can then be used to size of the number:

S B B B B B

Size (magnitude)
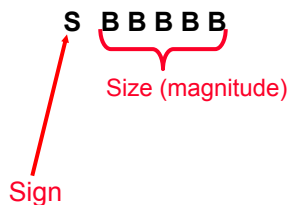
Sign

•*Because the sign bit cannot be used to represent the size, the
range of numbers that can be represented is reduced.*

Overflow: A Real World Example

•You can only represent a finite number of values
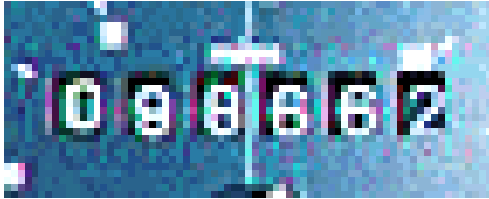
Overflow: Regular Binary

•Occurs when you don't have enough bits to represent a value
(*'wraps' –around to zero*)

Binary (1 bit)	Value
0	0
1	1
0	0
:	:

Binary (2 bits)	Value
00	0
01	1
10	2
11	3
00	0
:	:

Binary (3 bits)	Value
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7
000	0
:	:
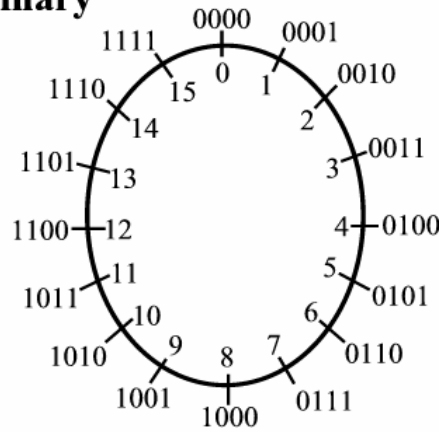
Graphical Representation: Unsigned Binary With 4 Bits

Binary

Graphical Representation: Unsigned Binary With 4 Bits

Binary

Overflow point
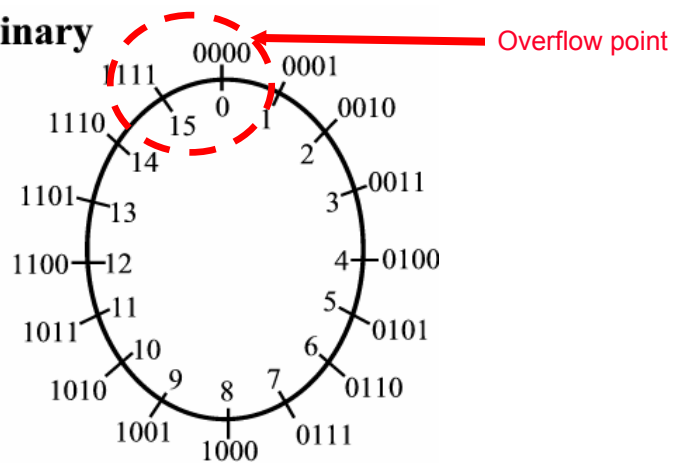
Overflow: Signed Representation

•Occurs when you don't have enough bits to represent a value but with signed overflow:
  – A positive number becomes a negative number!!!
  – A negative number becomes a positive number!!!

•Subtraction – subtracting two negative numbers results in a positive number.

  e.g. - 7
      - 1
      + 7

•Addition – adding two positive numbers results in a negative number.

  e.g.   7
      + 1
      - 8

Graphical Representation: Signed Binary With 4 Bits

Signed Binary

Graphical Representation: Signed Binary With 4 Bits

Signed Binary

Overflow points
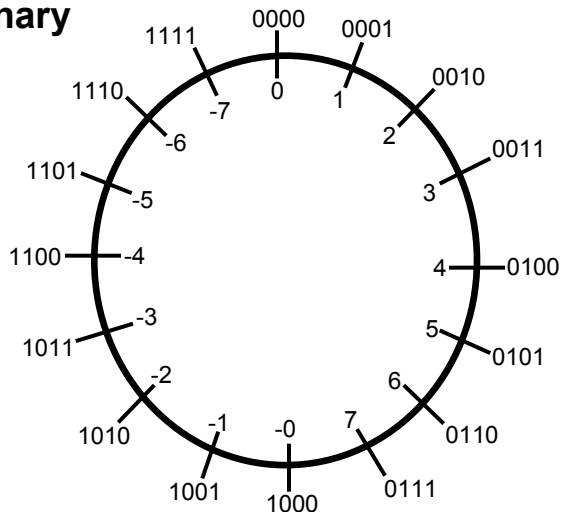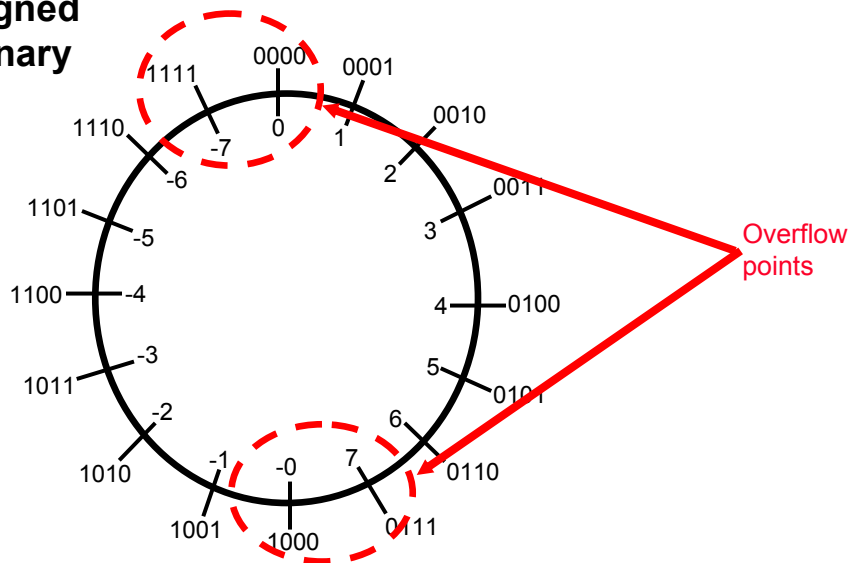
Representing Real Numbers Via Floating Point

• Numbers are represented through a sign bit, a mantissa and an exponent

Sign	Mantissa	Exponent

Examples with 5 digits used to represent the mantissa:
 – e.g. One: 123.45 is represented as $12345 * 10^{-2}$
 – e.g. Two: 0.12 is represented as $12000 * 10^{-5}$
 – e.g. Three: 123456 is represented as $12345 * 10^{1}$

• Using floating point numbers may result in a loss of accuracy!

Logic

•Refers to statements that are true or false:
  –2 + 3 = 5
  –2 + 3 = 4
  –The number 5

•Logic operations
  –AND
  –OR
  –NOT

Logic Applied In Software

•The use of logic is common in some computer programs

Logical AND

- Search for all documents that contain the name '*Bruce*' AND the name '*Lee*'

Logical AND

Truth table		
X	Y	X AND Y
False	False	False
False	True	False
True	False	False
True	True	True

Logical AND: An Example

	T	T	F	F	T	F
AND	F	T	F	T	T	F

Logical OR

•Search for all documents that contain the name '*Bruce Lee*' OR the name '*Little dragon*'

Logical OR

Truth table		
X	Y	X OR Y
False	False	False
False	True	True
True	False	True
True	True	True

Logical OR: An Example

		T	T	F	F	T	F
OR		F	T	F	T	T	F

Logical NOT

•Search for all documents that contain the name 'James Tam'
 NOT including those from the University of Calgary

Logical NOT

Truth table	
X	Not X
False	True
True	False

Logical NOT: An Example

	T	T	F	F	T	F
NOT						

Logical Operations Implemented In Hardware
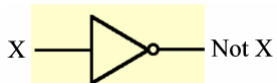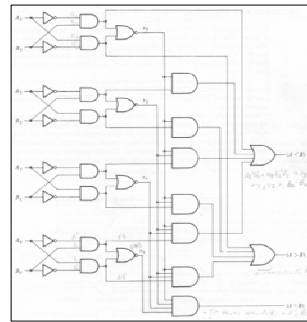
AND gate

X
Y
X AND Y

OR gate

X
Y
X OR Y

NOT gate

X
Not X
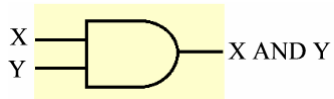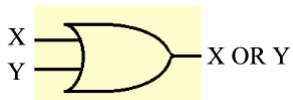
Reasons For Learning About Logic

•Software perspective:
– Logic is used in database (e.g., web) searches

•Hardware perspective
– The computer is built using logic circuits

You Should Now Know

•What is meant by a number base.

•How the binary number system works and what role it plays in the computer.

•How to convert to/from binary and decimal.

•How negative numbers are represented on the computer.
– What are implications of using signed vs. unsigned representations.
– What is signed and unsigned overflow.
– What is the magnitude of using each representation with a given number of bits.

•How are real numbers represented on the computer.
– How a real number vs. an integer number representation can result in a loss of precision.

You Should Now Know (2)

•The different types of logical operations that a computer may
 perform:
 – AND
 – OR
 – NOT

•The role of logic when using software

•How logic gates form an important part in the hardware of
 computers