Numbers and Representations
CS 0447

Thumrongsak Kosiyatrakul
tkosiyat@cs.pitt.edu

Numbers in Electronic Circuits

- How can we represent a number using electricity?
- Consider an outlet (**DO NOT TRY THIS ANYWHERE**)
- If you stick your finger in one of the hole of an electricity outlet, you either
  - get an electric shock, or
  - not get an electric shock.
- How to translate the electric shock feeling into a number?
  - getting an electric shock represents the number 1 (one), and
  - not getting an electric shock represents the number 0 (zero).
- An outlet can represent a number between 0 and 1.

Numbers in Electronics Circuits

- Some may interpret using the level of hurt when he/she get shock
  - not at all: 0,
  - hurt a little bit: 1,
  - hurt quite a lot: 2, or
  - hurt a lot: 3.
- Possible but hard to judge the level of hurt.
- Using volt meter may help but to get to 3, it has to go pass 1 and 2.
- Need to wait until the reading is stable before translating into a number.
- Not suitable if you want to translate to a number very very fast.
- 0 and 1 are simple to translate in electronic circuit.

Numbers

- Examples of Numbers:

$$5, 29, 253, 1455, \text{and } 20467$$

- A number consists of a series of digits (can have one digit)
- Each digit can be 0, 1, 2, 3, 4, 5, 6, 7, 8, or 9.
- The following examples are not numbers:
  - 1r2: The second digit (from right) is not a number between 0 to 9.
  - 12_45: The third digit (from right) is not a number.

Range of a Number

- Range of a number depends on the number of digits.
- Consider a three-digit number:
  - the smallest value is 000,
  - the largest value is 999.
- A three-digit number can be used to represent a value from 0 to 999.
- There are 1000 values.
- The range can be calculated by

$$\text{Range} = \text{Highest Value} - \text{Lowest Value} + 1.$$

- For example: a three-digit number:

$$999 - 0 + 1 = 1000.$$

- For example: a five-digit number:

$$99999 - 0 + 1 = 100000.$$

Value of a Number

- Consider a number 2013
- Its value is 2013 (two thousand and thirteen)
- Its value can be calculated as follows:

$$(2 \times 1000) + (0 \times 100) + (1 \times 10) + (3 \times 1) = 2013.$$

- Position of each digit effects the value (**positional number system**)
- Recalculating the value of 2013:

$$(2 \times 10^3) + (0 \times 10^2) + (1 \times 10^1) + (3 \times 10^0) = 2013.$$

- Consider the right most digit to be the digit 0
- The value of digit x is the number on that digit multiply by 10^x.
- We call this number system **base 10** or **decimal**.

Other Bases

- A base can be any number
- **Base 10** (decimal):
  - The value of digit x is the number on that digit multiply by 10^x.
  - Each digit can be a number between 0 to 9.
  - For example: 2013_{10}
- **Base 8** (octal):
  - The value of digit x is the number on that digit multiply by 8^x.
  - Each digit can be a number between 0 to 7.
  - For example: 2135_8
  - **NOTE**: 283_8 is not an octal number (the second digit cannot be the number 8).

Other Bases

- **Base 16** (hexadecimal):
  - The value of digit x is the number on that digit multiply by 16^x.
  - Each digit can be a number between 0 to 15 where a represents 10, b represents 11, and so on.
  - For example: $1a7c_{16}$
- **Base 2** (binary):
  - The value of digit x is the number on that digit multiply by 2^x.
  - Each digit can be either 0 or 1.
  - For example: 1011_2
- Use subscript to represent the base of the number (generally ignore when the number is base 10)

Binary Numbers

- The value of digit x is the number on that digit multiply by 2^x.
- Each digit can be either 0 or 1.
- A **bi**nary digi**t** is called a **bit**.
- Start from the bit zero (the right most bit).
- Examples:

Decimal	Binary	Decimal	Binary
0	0000_2	8	1000_2
1	0001_2	9	1001_2
2	0010_2	10	1010_2
3	0011_2	11	1011_2
4	0100_2	12	1100_2
5	0101_2	13	1101_2
6	0110_2	14	1110_2
7	0111_2	15	1111_2

Bits, Bytes, and Words

- A binary digit is called a **bit**.
- A series of four binary digits is called a nibble.
- A series of eight binary digits is called a **byte**.
- A series of 16 or 32 binary digits is called a **word**.
- What is the range of a byte?
  - A byte consists of 8 bits.
  - Smallest value is 00000000_2 (0_{10}),
  - Largest value is 11111111_2 (255_{10}).
  - Thus, the range is 255 - 0 + 1 = 256.

Binary to Decimal Conversion

- Use the same method as in base 10.
- Recall that the value of bit (digit) x is the number on that bit (digit) multiply by 2^x.
- What is the value of 10110101_2 in decimal?

Bit	7	6	5	4	3	2	1	0
Bit Value	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Bit Value in Decimal	128	64	32	16	8	4	2	1

- Thus, the value of 10110101_2 is

$$(1\times2^7)+(0\times2^6)+(1\times2^5)+(1\times2^4)+(0\times2^3)+(1\times2^2)+(0\times2^1)+(1\times2^0)$$

or

$$(1\times128)+(0\times64)+(1\times32)+(1\times16)+(0\times8)+(1\times4)+(0\times2)+(1\times1)$$

or

$$128+32+16+4+1=181$$

Binary to Decimal Conversion

- How about 10011011_2?

$$(1 \times 2^7) + (1 \times 2^4) + (1 \times 2^3) + (1 \times 2^1) + (1 \times 2^0)$$

or

$$128 + 16 + 8 + 2 + 1 = 155$$

- How about 10011010_2?

$$(1 \times 2^7) + (1 \times 2^4) + (1 \times 2^3) + (1 \times 2^1)$$

or

$$128 + 16 + 8 + 2 = 154$$

- **Note**: If bit 0 is 0, it is an even number.
- How about $10110110101000111000000111110110_2$?
- We better write a program to do the conversion.

Powers of Two

- It will be good for you if you can remember some of powers of two at least up to 2^{10}

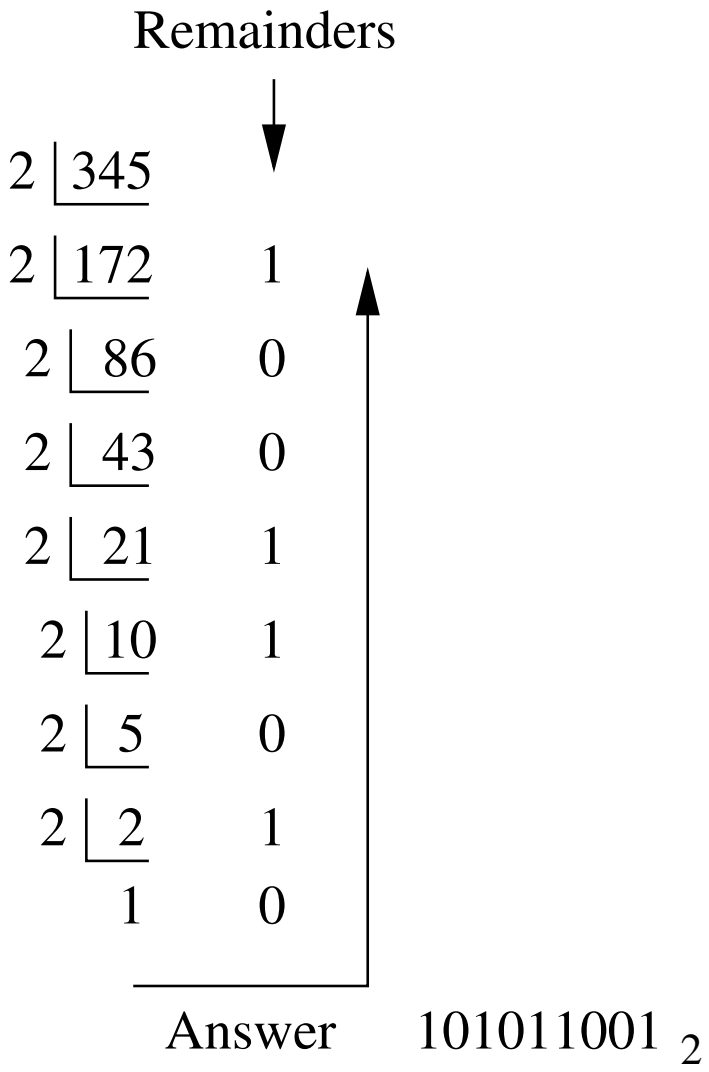
2^x	Decimal Value	2^x	Decimal Value
2^0	1	2^{11}	2048
2^1	2	2^{12}	4096
2^2	4	2^{13}	8192
2^3	8	2^{14}	16384
2^4	16	2^{15}	32768
2^5	32	2^{16}	65536
2^6	64	2^{17}	131072
2^7	128	2^{18}	262144
2^8	256	2^{19}	524288
2^9	512	2^{20}	1048576
2^{10}	1024	2^{21}	2097152

Powers of Two

- Kilo, Mega, and Giga
  - 2^{10} is called Kilo,
  - 2^{20} is called Mega,
  - 2^{30} is called Giga.
- For example 4 kilobyte is $4 \times 2^{10} = 4096$ bytes.
- **Note**: Sometime people use Kilo, Mega, and Giga to represent 10^3, 10^6, and 10^9 respectively.
- For example 1 Terabyte harddrive means 1×10^{12} bytes
- When you check the property of your harddrive in your computer, it may say that the capacity is 932 Gigabyte ($932 \times 2^30 = 1000727379968$ bytes)
- In exams, we will use Kibi (2^{10}), Mebi (2^{20}), Gibi (2^{30}) for binary prefix multipliers.

Decimal to Binary Conversion

- Convert 345_{10} to binary.



Decimal to Binary Conversion

- Suppose we have a four-bit binary number $x_3x_2x_1x_0$, where x_n is either 0 or 1.
- The value of $x_3x_2x_1x_0$ in base 10 is

$$(x_3 \times 2^3) + (x_2 \times 2^2) + (x_1 \times 2^1) + (x_0 \times 2^0).$$

- The above value is equal to

$$(x_3 \times 2^3) + (x_2 \times 2^2) + (x_1 \times 2^1) + x_0.$$

- The above value is also equal to

$$\Big[2 \times \big((x_3 \times 2^2) + (x_2 \times 2^1) + x_1\big)\Big] + x_0.$$

- What is the remainder of the above value divided by 2?
- Obviously the left term of the above equation is divisible by 2. (two times anything is divisible by 2)
- Thus, the remainder of the above value divided by 2 is x_0.

Decimal to Binary Conversion

- The result of the above value divided by 2 (without decimal places) is
$$(x_3 \times 2^2) + (x_2 \times 2^1) + x_1.$$
- The above value is equal to
$$\Big[2 \times \big((x_3 \times 2^1) + x_2\big)\Big] + x_1.$$
- Again, the remainder of the above value divided by 2 is x_1.
- The result of the above value divided by 2 (without decimal places) is
$$(x_3 \times 2^1) + x_2.$$
- **Note that**:
  - x_0 is the remainder of the decimal value divided by 2
  - x_1 is the remainder of the result of the decimal value divided by 2, divided by 2.
  - x_2 is ....

Decimal to Binary Conversion

- To convert a decimal value to binary, start with `value` be a decimal value and x be 0.
  1. Bit x is the remainder of `value` divided by 2.
  2. Replace `value` by `value`/2 without decimal places.
  3. Increase x by 1.
  4. Go back to step 1 until `value` is 0.
- Let's work on some examples and verify the result.
  - $24_{10} = 11000_2$
  - $351_{10} = 101011111_2$

Representations

How to represent numbers/data in a computer?

Representations

- Suppose we have 8 outlets, we can represent an 8-bit binary number.
- Right most outlet represents bit 0 (**Least Significant Bit**)
- Left most outlet represents bit 7 (**Most Significant Bit**)
- In a small circuit, we can use a series of wires.
  - 8 wires (8-bit binary),
  - 16 wires (16-bit binary),
  - 32 wires (32-bit binary), or
  - 64 wires (64-bit binary).
- Each wire represents one bit of a binary number.
  - If a wire has high voltage, it represents 1.
  - If a wire has no voltage, it represents 0.

Representing Integers in a Computer

- Any positive integer number can be convert to a binary number.
- Simply convert a positive integer into a binary number in a logical way.
- Use series of wires with high/low voltages to represent a binary number.
- How about a negative number?
- A negative integer number (decimal) is represented by a minus sign in front of a number.
- For examples: -135, -2, -199, etc
- Yes, we can put a minus sign in front of a binary number.
- What to do in a computer? Put an extra wire?
- If that wire is 0, it is a positive number, and vice versa.

Sign-Magnitude

- Use the Most Significant Bit to represent a sign
  - if MSB is 0, the number is positive
  - if MSB is 1, the number is negative
- Use the rest of bits to represent a number
- Consider an 8-bit number, 123_{10} is 01111011_2.
- Using sign-magnitude, -123_{10} can be represented as 11111011_2.
- Range of an 8-bit sign-magnitude is 255 (from -127 to 127).
- There are two zeros 10000000_2 and 00000000_2.
- Easy to convert
- But $(-x) + x \neq 0$

One's Complement

- One's complement was implemented to solve $(-x) + x \neq 0$
- Simply perform a bit by bit invert.
- For example: 123_{10} is 01111011_2, -123_{10} can be represented by 10000100_2.
- Note that $01111011_2 + 10000100_2 = 11111111_2 = 0_{10}$.
- There are two zeros 00000000_2 and 11111111_2.
- But $x + 0 = x$ only works for positive zero (00000000_2).

Two's Complement

- Two's complement was implemented to solve two zeros.
- Simply perform a bit by bit invert and add 1.
- For example: 123_{10} is 01111011_2, after bit by bit invert is 10000100_2, after add 1 is 10000101_2.
- Note that $01111011_2 + 10000101_2 = 00000000_2 = 0_{10}$
- $(-x) + x = 0$
- There is only one zero
- $x + 0 = x$

Convert Two's Complement Number to Decimal

- Method 1:
  - If the MSB is 0, just perform simple translation from binary to decimal
  - If the MSB is 1,
    - perform bit by bit invert
    - add 1
    - perform simple translation from binary to decimal
    - negate the result
- Method 2: same as simple conversion except that the magnitude of MSB is negative.
  - Example (4-bit): 1101_2

$$-(1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$$

or

$$-8 + 4 + 1 = -3$$

Summary

- Examples of 3-bit binary numbers and their values in decimal:

Binary Number	Sign-Magnitude	One's Complement	Two's Complement
000	+0	+0	+0
001	+1	+1	+1
010	+2	+2	+2
011	+3	+3	+3
100	-0	-3	-4
101	-1	-2	-3
110	-2	-1	-2
111	-3	-0	-1

Representing Characters

- Simply perform one-to-one conversion (mapping).
- For examples:
  - 1100001_2 (96_{10}) represents the character a
  - 110110_2 (54_{10}) represents the character 6
- Use ASCII table for conversion

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000																
0001																
0010	Space	!	”	#	$	%	&	’	(	)	*	+	’	-	.	/
0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0110	‘	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
1000																

- From the above table, the left four bits binary representing the character 'A' come from the row heading (0100) and the right four bits come from the column heading (0001).
- From ASCII the character 'A' is represented by 01000001_2.