Binary and Digital Logic

{
ELTN 130
Tom Thoen – Teacher / Student / Hobbyist / Inventor

Computers at the very most basic level represent information as sets of *on* and *off* values (this is referred to as *digital*). This is somewhat conceptual, but we can also call describe these values with words or values such as:

Low	High
False	True
0	1

Why is this? Computer data is represented in *binary*, which means we use base 2 instead of base 10 (sometimes called "decimal"). In electronic systems, voltage levels are used to represent these different values. Fairly common voltages used are:

$$0V = 0$$
$$5V = 1$$

Counting numbers:

If we count in decimal (base 10), we start at zero, and then count up by ones:

0	Start at zero
1	0 + 1 = 1
2	1 + 1 = 2
3	2 + 1 = 3
4	
…	
8	7 + 1 = 8
9	8 + 1 = 9
10	9 + 1 = 10

However, <u>when we get to a value larger than 9</u> we need to "reset" that column to zero <u>and carry a 1 into the next column</u>. This is because in decimal we have only 10 symbols (0 – 9) to represent a value in a single column.

Counting in binary:

First we must understand some basic binary math:

0 + 1 = 1
1 + 0 = 1
1 + 1= 10   (if this doesn't make sense, look at the previous slide!
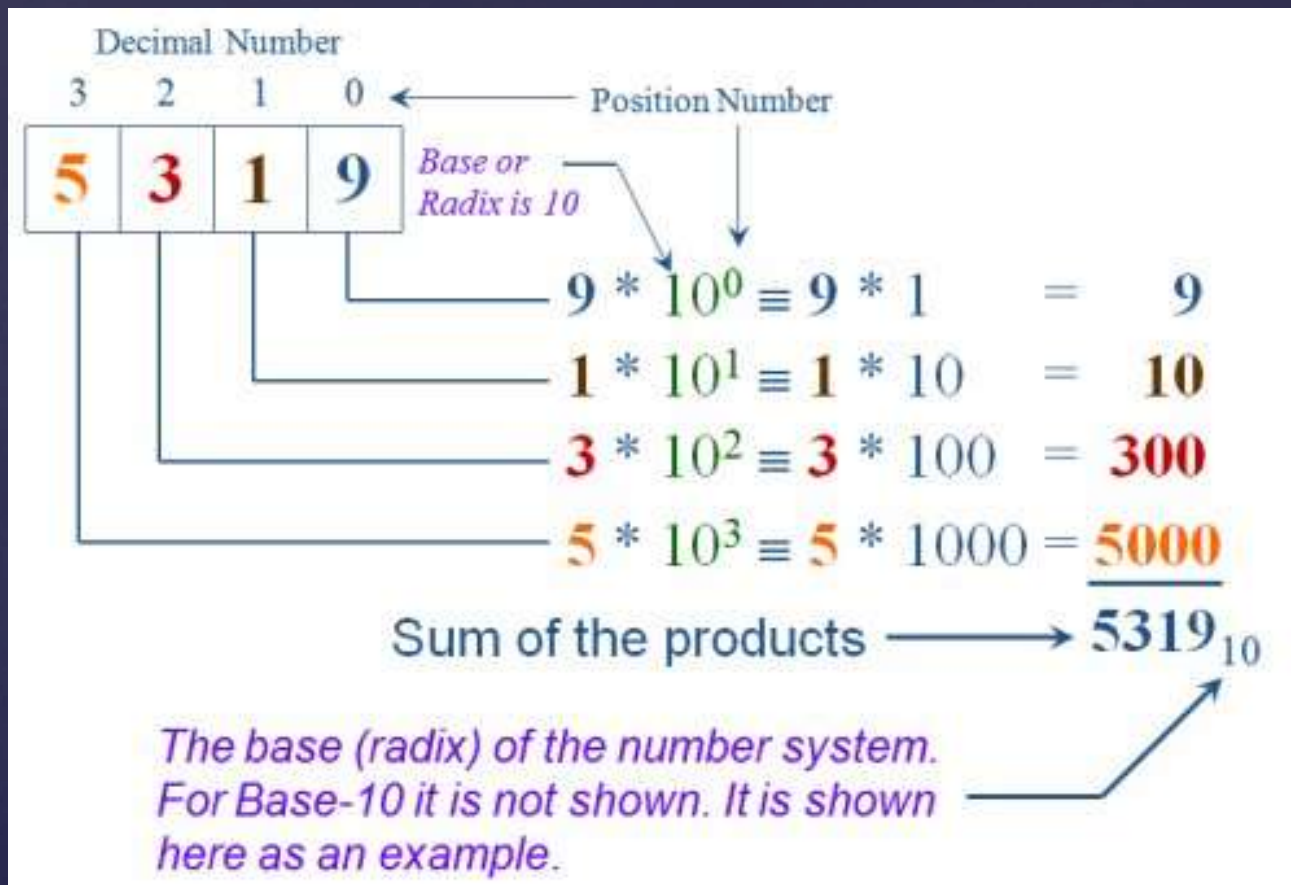   In binary, The largest number in a single column is 1)

When we count in <u>binary</u>, we start at zero, and then add one each time (just like decimal). Since 1 is the <u>largest number</u>, we have to carry left into the next column:

|	
0	zero
1	0 + 1 = 1
10	1 + 1 = zero with a carry of one
11	10 + 1 = 11
100	11 + 1 = zero with two carries…

Weighted numbering systems:

Decimal is considered a "weighted" system. As we move to the left, each column is 10 times greater than the previous:

Decimal Number

| 3 | 2 | 1 | 0 | ← Position Number |

| 5 | 3 | 1 | 9 | Base or Radix is 10 |

$$9 * 10^0 \equiv 9 * 1 = 9$$
$$1 * 10^1 \equiv 1 * 10 = 10$$
$$3 * 10^2 \equiv 3 * 100 = 300$$
$$5 * 10^3 \equiv 5 * 1000 = 5000$$

Sum of the products ⟶ 5319_{10}

The base (radix) of the number system. For Base-10 it is not shown. It is shown here as an example.

Weighted numbering systems:

In binary, the same concept applies, except when we move to the left we are increasing our values by a power of 2 instead of 10. As we move to the left, *each column is 2 times greater than the previous*:
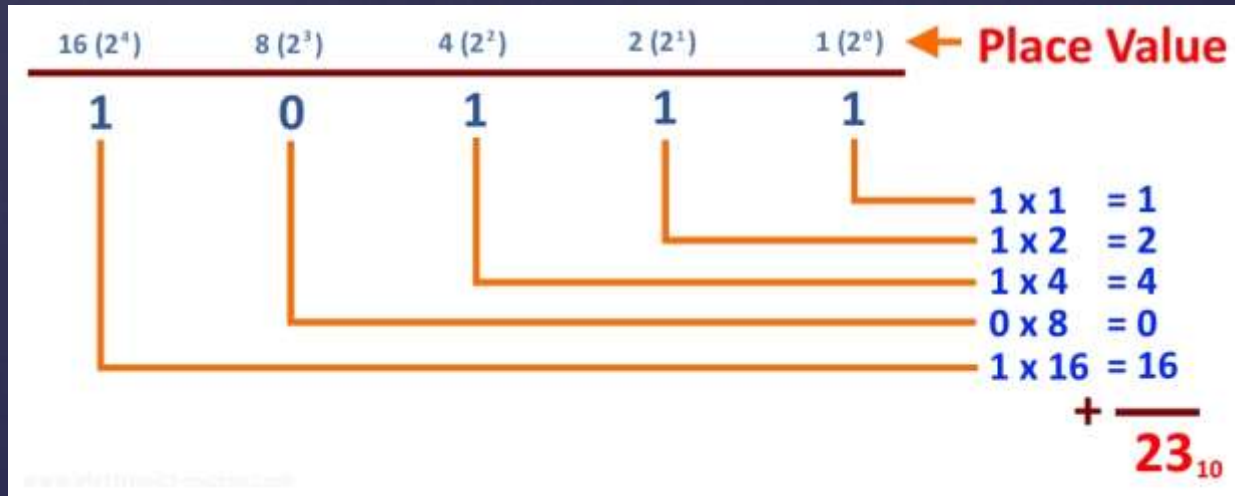
Converting numbers from decimal to binary

Good link: http://www.wikihow.com/Convert-from-Binary-to-Decimal

Converting decimal to binary

Look at step 2 at the following link:

http://www.wikihow.com/Convert-from-Decimal-to-Binary

Grouping sets of binary numbers

A single binary value can be either a zero or a one. This is called a *bit*, which is the contraction of *b*inary dig*it*.

Example: One bit: 0 or 1

Four bits are called a *nibble*   0110
Eight bits are called a *byte*   0110 1001
Sixteen bits are called a *word*   0110 1001 1100 0010

Hexadecimal

A more "compact" way of representing information is to use Hexadecimal. This allows us to represent a four bit value (i.e. from 0000 to 1111) with a single character.

Hex Number	4-bit Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A (10)	1010
B (11)	1011
C (12)	1100
D (13)	1101
E (14)	1110
F (15)	1111

Logic Gates

In order for electronic systems to perform operations on binary values, *logic gates* are used. Logic gates are the "molecules" of computer systems, which are composed of up to millions of gates. The three most basic gates are the AND, OR and NOT (Inverter) gates. The symbols and *truth tables* (rules for outputs based on inputs) are shown below:

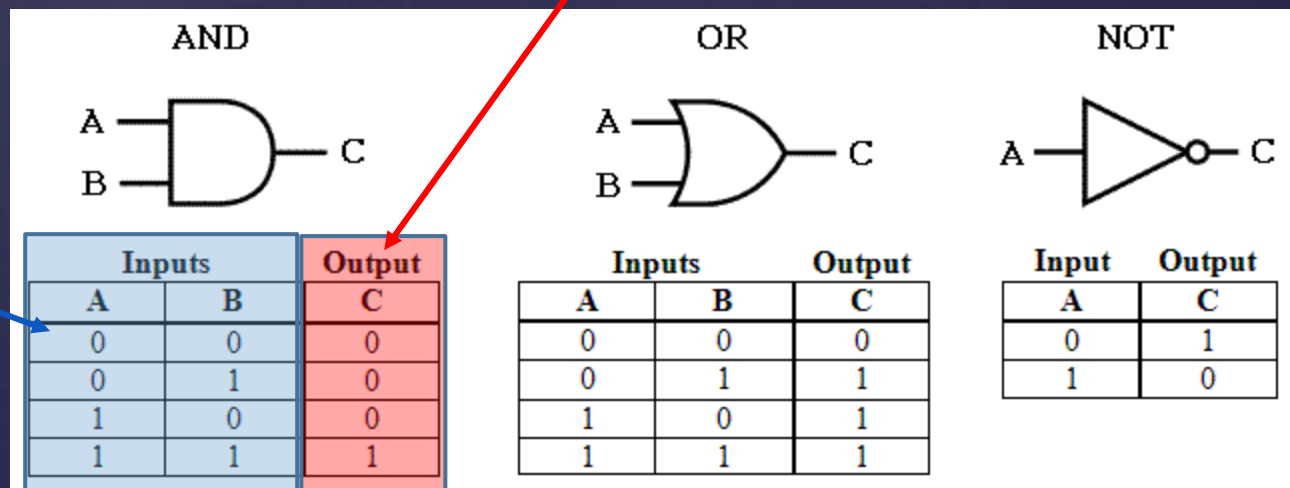| AND				OR				NOT	
Inputs		**Output**		**Inputs**		**Output**		**Input**	**Output**
A	B	C		A	B	C		A	C
0	0	0		0	0	0		0	1
0	1	0		0	1	1		1	0
1	0	0		1	0	1			
1	1	1		1	1	1			

Truth Tables

A truth table defines the output of a logic gate or circuit based on all possible inputs. The gates below only have two inputs, but gates can have 3,4, or even more inputs.

1 output

All possible inputs

AND			OR			NOT	
Inputs		**Output**	**Inputs**		**Output**	**Input**	**Output**
A	B	C	A	B	C	A	C
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

Logic Gates - Combinations

Logic Diagrams and Expressions

Truth Table	
X Y Z	$F = X + \overline{Y} \cdot Z$
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	1
1 1 0	1
1 1 1	1

Logic Equation

$$F = X + \overline{Y}\, Z$$

Logic Diagram
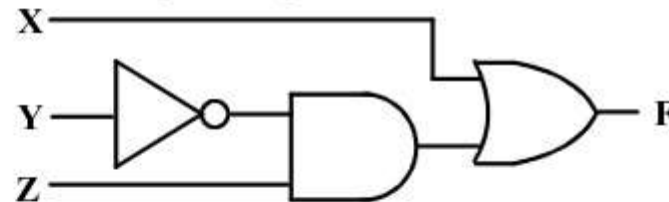
- Boolean equations, truth tables and logic diagrams describe the same function!
- Truth tables are unique, but expressions and logic diagrams are not. This gives flexibility in implementing functions.

Boolean Algebra

Logic Gates – input combinations

Logic gate inputs and outputs are connected to create more complex digital systems. We will be discussing logic circuits in the coming lectures.

Video links: Please watch to review!

Binary numbers:

https://www.youtube.com/watch?v=LpuPe81bc2w

Logic gates: Hardware connections on a breadboard:
https://www.youtube.com/watch?v=cdMJvFT-Afc

Logic gates and Boolean expressions – longer video, but goes into a lot of detail:
https://www.youtube.com/watch?v=Sf_Yx2z1fRk

And if you're interested in the origins of numbering systems and how they created calculators in the 1600's, check this out:

https://plus.maths.org/content/arithmetic-bones-and-counting

References

- https://wikihow.net
- https://www.slideshare.net