SANDY BAY PARK RESIDENTS ASSOCIATION (SBPRA)

COMPUTER BASICS 1

The Binary Number System

How computers work!
A basic understanding of
the numbering system
used in computers.

Kelvin Dimond

Computer Basics – The Binary Number System

Fundamental to the operation of computers, tablets, mobile phones, flat screen televisions , and in fact critical to virtually every digital device in use today.

You don't have to be an expert on the binary number, this information provides some background on the way it makes devices work.

What is binary?

Binary describes a numbering scheme in which there are only two possible values for each digit -- 0 or 1 -- and is the basis for all binary code used in computing systems. These systems use this code to understand operational instructions and user input and to present a relevant output to the user.

The term **binary** also refers to any digital encoding/decoding system in which there are exactly two possible states. In digital **data memory**, **storage**, **processing** and **communications**, the 0 and 1 values are sometimes called *low* and *high*, respectively. In transistors, 1 refers to a flow of electricity, while 0 represents no flow of electricity.

The importance of binary code

The binary number system is the base of all computing systems and operations. It enables devices to store, access and manipulate all types of information directed to and from the CPU or memory. This makes it possible to develop applications that enable users to do the following:

- view websites,
- create and update documents,
- play games,
- view streaming video and other kinds of graphical information,
- access software,
- perform calculations and data analyses.

The binary scheme of digital 1s and 0s offers a simple and elegant way for computers to work. It also offers an efficient way to control logic circuits and to detect an electrical signal's true (1) and false (0) states.

So, what does this all mean for you?

In mathematics and in computing systems, a binary digit, or bit, is the smallest unit of data. Each bit has a single value of either 1 or 0, which means it can't take on any other value.

Computers can represent numbers using binary code in the form of digital 1s and 0s inside the central processing unit (CPU) and Random Access Memory (RAM). These digital numbers are electrical signals that are either on or off inside the CPU or RAM.

How binary numbers work

The binary system is the primary language of computing systems. Inside these systems, a binary number consists of a series of eight bits. This series is known as a byte In the binary system, the position of each digit determines its decimal value. Thus, by understanding the position of each

bit, a binary number can be converted into a decimal number.

In decimal numbers, each additional place is multiplied by 10 as we move from right to left (first place, 10th place, 100th place, etc.). But, in binary numbers, each additional place while moving from right to left is multiplied by two.

Example

Here's how the decimal values are calculated for an 8-bit (byte) binary number 01101000.

In this number, the first digit is at the far right, while the eighth digit is at the far left. The second (0) to the seventh (1) digits are read from right to left.

Bit position	**8th Digit**	**2nd Digit**	**3rd Digit**	**4th Digit**	**5th Digit**	**6th Digit**	**7th Digit**	**1st Digit**
Bit	**0**	**0**	**0**	**1**	**0**	**1**	**1**	**0**
Binary-to-decimal calculation (exponent)	2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7
Decimal value (x2)	1	2	4	**8**	16	32	**64**	128

As the bit position increases from one to eight, the previous decimal value is multiplied by two. That's why the first bit has a value of 1, the second bit has a value of 2, the third bit has a value of 4 and so on.

The final value of the decimal number is calculated by adding the individual values from the above table. However, only those values where the bit equals 1 should be added. These values represent the "on" position. The 0s represent the "off" position, so they are not counted in the decimal value calculation.

So, for the binary number **01101000**, the decimal value is calculated as the following: **8 + 32 + 64 = 104**

Here's how the decimal values are calculated for the binary number **11111111**.

In this binary number, every bit has a value of 1, so all the individual values are added.

So, for this number**,** the decimal value is the following:

1 + 2 + 4 + 8 + 16+ 32 + 64 +128 = 255

As mentioned earlier, the binary numbering system only works with 1s and 0s. However, the position of just these two digits can represent many more numbers.

The examples show how any decimal number from 0 to 255 can be represented using binary numbers.

Numbers larger than 255 can also be represented by adding more bits to an 8-bit binary number. You may have heard of the terms 16 Bit, 32 Bit and 64 Bit computers, this is where

extra digits are added to allow bigger numbers to be processed and allow the computer to run faster and process more applications at the same time.

What actually goes inside your computer?

Obviously just presenting a series of numbers on a screen won't help you compose an email, create a word document, defend earth from alien invaders or access the internet.

Binary numbers can be translated into text characters using American Standard Code for Information Interchange (ASCII) codes to store information in the computer's RAM or CPU. ASCII-capable applications, like word processors, can read text information from the RAM or CPU.

They can also store text information that can then be retrieved by the user at a later time. ASCII codes are stored in the ASCII table, which consists of 128 text or special characters. Each character has an associated decimal value, see table on next page.

There are also some non-printed ASCII codes that control how a document is formatted, where the return is, line spacing, etc.

By working in a similar way with a computer's graphics card, sound card, etc, the use of binary numbers is used to control, colours and intensity of pictures/videos and audio reproduction.

ASCII Control Characters

BINARY	DECIMAL	HEXADECIMAL	SYMBOL	BINARY	DECIMAL	HEXADECIMAL	SYMBOL
010 0000	32	20	SPACE	101 0000	80	50	P
010 0001	33	21	!	101 0001	81	51	Q
010 0010	34	22	"	101 0010	82	52	R
010 0011	35	23	#	101 0011	83	53	S
010 0100	36	24	$	101 0100	84	54	T
010 0101	37	25	%	101 0101	85	55	U
010 0110	38	26	&	101 0110	86	56	V
010 0111	39	27	'	101 0111	87	57	W
010 1000	40	28	(	101 1000	88	58	X
010 1001	41	29	)	101 1001	89	59	Y
010 1010	42	2A	*	101 1010	90	5A	Z
010 1011	43	2B	+	101 1011	91	5B	[
010 1100	44	2C	,	101 1100	92	5C	\
010 1101	45	2D	-	101 1101	93	5D	]
010 1110	46	2E	.	101 1110	94	5E	^
010 1111	47	2F	/	101 1111	95	5F	_
011 0000	48	30	0	110 0000	96	60	`
011 0001	49	31	1	110 0001	97	61	a
011 0010	50	32	2	110 0010	98	62	b
011 0011	51	33	3	110 0011	99	63	c
011 0100	52	34	4	110 0100	100	64	d
011 0101	53	35	5	110 0101	101	65	e
011 0110	54	36	6	110 0110	102	66	f
011 0111	55	37	7	110 0111	103	67	g

Those ones and zeros might not look like anything to you, but in **binary code** the numbers are actually saying "Hello!"

Acknowledgements: TechTarget & ScienceFriday