Using Base 2 (Binary) in Computers and Logic

Base 2 uses only two digits, a 0 and a 1. Each column is worth twice the value of the column to its right:

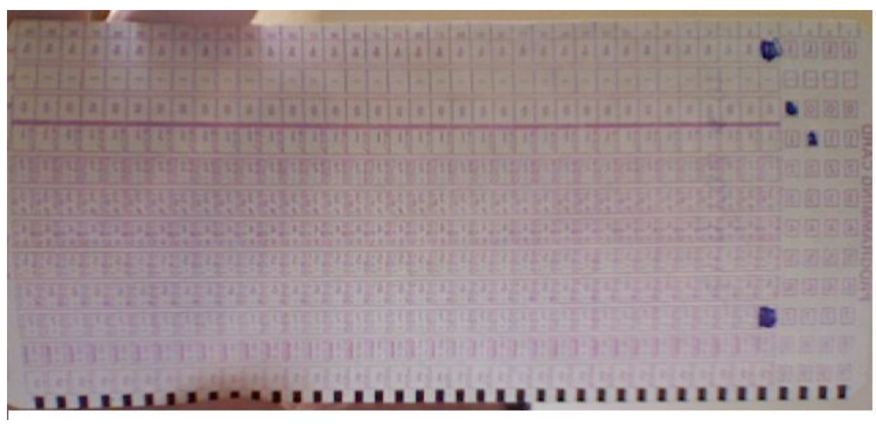
Column value	$32 = 2^5$	$16 = 2^4$	$8 = 2^3$	$4 = 2^2$	$2 = 2^1$	$1 = 2^0$
Number	1	0	1	1	0	1
Expanded	1 x 32 = 32	0 x 16 = 0	1 x 8 = 8	1 x 4 = 4	0 x 2 = 0	1 x 1 = 1

Thus, the number above is 101101_2 (the subscript means base 2) is equal to 32 + 0 + 8 + 4 + 0 + 1 = 45_{10} or 45.

Here are some highlights of the binary system:

(1) Only 2 digits, so any number can be represented by anything that can take two forms. Such as a light on, and light off; a candle lit, or a candle not lit (great for old guys birthday cakes, see below); a hole in a card, or not a hole (used in the Jacquard loom, see below); a mark on a card, or not a mark on a card (see below); something magnetized or not; a blip on a CD spot or not; and so on.

The birthday cake on the right illustrates my last birthday. A lighted candle represents a "1" and not lighted represents a "0" in binary. Hence I was 111101_2 years old. This makes the lighting of a cake less of a fire hazard as you grow older. Only those who can work in base 2 will know your actual age as well! Did you ever see the T-shirt with the following slogan on it? "There are 10 types of people in the world, those that understand binary and those that don't". Of course by not writing the 10 as 10_2 does make this slogan a bit of head shaker because if the base subscript is left out, it is assumed to be in base 10.



To the right is a picture of a computer card from the 70's that my students used to write programs. A mark on the card meant a "1" and not a mark was a "0"

Another example is from the late 1900's in the Jacquard loom. Here is a link to read more on it: http://www.ideafinder.com/history/inventions/jacquard.htm

What it was, was a loom that allowed a knitting needle to pass through if there was a hole in it, or not pass through if there was no hole in it. Thus a pattern could be recorded and repeated over and over.

(2) Base 2 has a really different counting system as illustrated in the following table:

Decimal	Binary	Decimal	Binary	Decimal	Binary	Decimal	Binary
1	1	6	110	11	1011	16	10000
2	10	7	111	12	1100	17	10001
3	11	8	1000	13	1101	18	10010
4	100	9	1001	14	1110	19	10011
5	101	10	1010	15	1111	20	10100

(3) Every even number ends in a "0", every odd number ends in a "1"

(4) The addition and multiplication tables are very easy to learn, you can see them below:

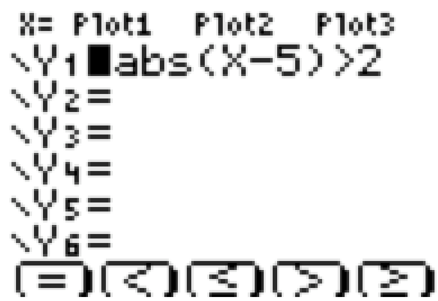
Add	0	1		Multiply	0	1
0	0	1		0	0	0
1	0	10		1	0	1

Think, for instance, that 1 plus 1 is 1 group of 2 and 0 groups of 1, hence we write 10_2

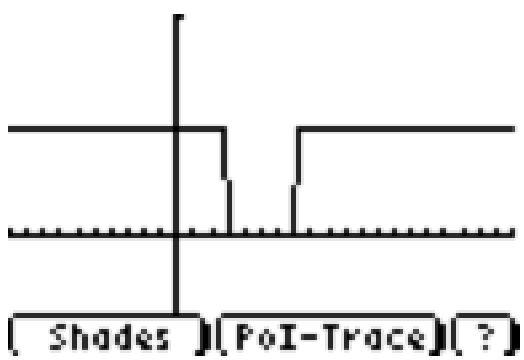
(5) George Boole noticed how similar the above tables are to the truth tables below:

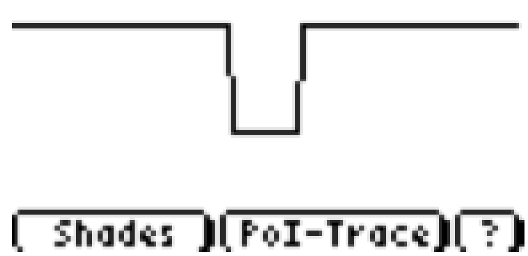
OR	False	True		ANS	False	True
False	False	True		False	False	False
True	False	True		True	False	True

By letting 0 represent False and 1 represent True, a new algebra of logic, called Boolean algebra, was invented. The only change that has to be made is have 1 + 1 = 1 and not 10_2, since you cannot get more True than True. Computers and calculators can use this in their logic circuits and in some examples:



In the example to the right, I have put the following equation: $|x-5|>2$ into the calculator as $y=|x-5|>2$. I am using a TI-84 graphing calculator. The abs in found in MATH NUM 1, and the ">" sign is found by pressing 2nd MATH (test) 3. What the calculator will do is to work out the truth value of each of the x-values from -10 to +20. If, when substituting in the x-value the right hand side is "True", it will graph a 1, if it is false, it will graph a "0" (ignore the vertical lines). Algebraically the solution to the equation $|x-5|>2$ is either x < 3 or x > 7. You can see from the graphs below that only when x<3 or x>7 do the graphs show a "1". In between x = 3 and x = 7, it graphs a zero. The left hand graph below shows the graph from x = -10 to +20 and y = -1 to +2. The right hand graph below shows the same graph without the axis in to illustrate that it is actually graphing y = 0 in between x = 3 and x = 7 (ignore vertical lines)





Another example of this was a computer program I wrote that computed provincial sales tax at 6%, and GST of 7%. The statement went something like this:
Total_price = sell_price + 0.06 x sell_price + 0.07 x sell_price

Then, on a certain date (let's say April 1, 2007) the sales tax dropped to 5%. I wanted an efficient way to handle this, plus keep the original value in as well. Suddenly George Boole came to the rescue. Now I wrote the same line like this:

Total_price = sell_price + 0.06 x sell_price x (date<2007.04.01) + 0.05 x sell_price x (date>2007.03.31) + 0.07 x sell_price

If the date was before April 1, 2007, the first bracket would be “true” and be recorded as a “1”. While the second bracket would be “false” and recorded as a “0”, so only 6% would be added for sales tax.

If the date was after March 31, 2007, the first bracket would be “false” and be recorded as a “0”. While the second bracket would be “true” and recorded as a “1”, so only 5% would be added for sales tax.

I have also used the same idea in Excel spreadsheets.

I will add to this whenever I come across any other uses of base 2. Tomorrow (Saturday April 18th), I will show you a math trick where base 2 is useful in understanding why it works. On Sunday April 19th, I will give you a Math game where base 2 can be useful, however you’ll have to wait a week to see how!!!