1+1=1 a tale of genius

George Boole....

....was born in Lincoln, England on Nov. 2nd 1815. He inherited his father's passion for science and by the age of 14 could read Latin, Greek, French and German. But Boole's family fell on hard times, and he was forced find work to support them.

Boole discovered and taught himself mathematics while teaching in local schools. The papers that he published in the Cambridge Mathematical Journal earned him respect as a capable mathematician. In 1849, despite lacking a university degree, he was offered the first professorship of mathematics at Queen's College, Cork, in Ireland, where he taught until his death on Dec. 8th, 1864.

In 1854, Boole published his greatest and most influential work: "An Investigation Into the Laws of Thought, on Which are Founded the Mathematical Theories of Logic and Probabilities" in which he brilliantly combined algebra with logic. In 1937 Claude Shannon placed Boole's abstruse reasoning in an engineering context where it became instrumental in the development of the digital computer.

Boole was well liked and known to be extremely dedicated to his research, his students and his family. He is remembered as a personable, congenial, kind-hearted teacher and a brilliant mathematician. His papers are preserved in the archives of the Boole Library at University College, Cork. A lunar crater also bears his name.

Claude Elwood Shannon....

....was born in Petoskey, Michigan, on April 30th, 1916. He graduated from the University of Michigan in 1936 with bachelor's degrees in mathematics and electrical engineering. In 1940 he was awarded both a master's degree in electrical engineering and a Ph.D. in mathematics from the Massachusetts Institute of Technology (MIT).

Shannon joined the Mathematics Department at Bell Labs in 1941 with which he remained affiliated until 1972. He became a visiting professor at MIT in 1956, a permanent member of the faculty in 1958, and a professor emeritus in 1978.

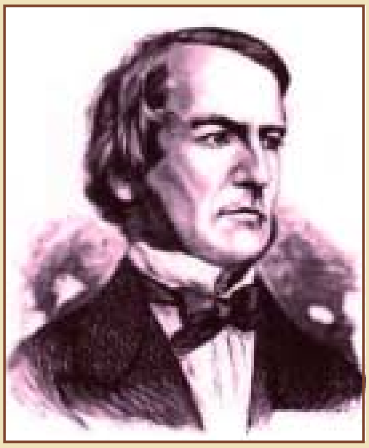
Shannon was renowned for his eclectic interests and capabilities. A favourite story describes him juggling while riding a unicycle down the halls at Bell Labs. He also designed and built chess playing, maze-solving, juggling and mind reading machines. These activities bear out Shannon's claim that he was motivated more by curiosity than usefulness. In his words "I just wondered how things were put together."

Another example of Shannon's diverse interests is his 1949 paper entitled "Communication Theory of Secrecy Systems", a work now generally credited with transform-ing cryptography from an art into a science.

Claude Shannon died on February 26th, 2001

"He looked, not like a professor writing a demonstration on a blackboard, but like an artist painting from a vision".

George Boole described by a student.



George Boole, 1815-1864

One of this journal's editorial policies - borrowed from that giant of radio broadcasting, John Reith - is "to inform, to educate and to entertain". Well, we try our best.

While chewing on my pencil wondering how I might achieve any of these objectives, I was compelled to pause to take in what I regard to be one of the most evocative pieces of music ever written. The recording I was playing (I'm ashamed to say for background listening) was of Franz Schubert's string quartet in D minor, popularly named "Death and the Maiden". In the slow movement the composer paints a picture of Death responding to a maiden's pleas to pass her by, gently assuring her that he comes to take her as a friend. And Death was soon to take Schubert, before he could enjoy the royalties that would later accrue to others or appreciate the extent of his musical legacy. Marx, Diesel, Braille, Goodyear, Hertz (as in "Hz"), Silver (the ubiquitous bar code) and Shakespeare are a few who spring to mind (apart, of course, from several notable religious prophets) that undoubtedly left this life with little idea of the impact that their work would have - for better or worse - on later generations.

George Boole is another. A little-known professor of mathematics, Boole left behind a curious form of algebra, of interest to his peers but of no known practical value: until, that is, it was stumbled across many years later, outside the realm of pure mathematics and almost by accident. For Boole's "algebra of logic" and its associated laws were to become fundamental to the design of digital circuits. While it is untrue to say that digital computing and communications would not have existed but for Boole's work, it's difficult to imagine how, without it, complex binary circuits could operate reliably.

According to Boole

As is often the case with invention, Boole was not the first to investigate the problem. But in contrast to earlier attempts at "symbolic logic", Boole's exploration resulted in equations and techniques that make possible a scientific treatment of logic in which logical relationships can be expressed as formulae, free from vagueness and ambiguity. Although "Boolean logic" cannot be applied to the many everyday situations that involve speculation or uncertainty, it can be applied to the factual statements that form the basis of digital computing.

Boole argued that we tend to select things from within a boundary containing all possible choices. If asked to select the large black balls from a bowl containing black and while balls of two sizes, our selection criteria would probably be *large* AND *black*. Conversely, if asked to exclude all large black balls from our selection using the same operands, our criteria would become *black* NOT *large*. In either case we exclude all the white balls because they're not of the correct category. If, however, we needed all the large black and all the small white balls, our selection criteria would be (*black* AND *large*) OR (*white* NOT *large*).

Boolean logic implements this type of reasoning, an approach we now apply when using an Internet search engine to perform a "Boolean search". For example, searching for *George* AND *Boole* returns references in which both the words *George* and *Boole* appear, while searching for *George* OR *Boole* lists those in which either word appears. In fact we're combining and manipulating our search criteria using a binary 'true-false' (or 'open-shut', 'zero-one', 'yes-no', 'on-off', etc.) approach. A more recent development, "fuzzy logic", can handle the concept of partial truth - values that lie between "completely true" and "completely false" - but that's another story.

In Boolean logic, the symbols used - for example '***p***' and '***q***' - are not variables in the same sense that '***x***' and '***y***' are often used to represent numbers in conventional algebra. Boole defined a set of rules that specify the result of the permitted operations on the symbols, but without any regard to what they actually represent. The symbols can of course be interpreted, for example in terms of the black and white balls mentioned above, but logic that results in accepting the equation "1+ 1 = 1" is certainly not true of conventional algebra!

The three basic "operators" in Boolean algebra are '**AND**', '**OR**' and '**NOT**'.

Fig 1

pq

1. Both ***p*** and ***q*** can be true;
2. ***p*** can be true and ***q*** false;
3. ***p*** can be false and ***q*** true;
4. Both ***p*** and ***q*** can be false.

The 'AND' operator

In Boolean algebra, the AND operator is signified as a '.' although the dot is often dropped, ***p.q*** simply being written as ***pq***. Four possibilities can be derived from combining the symbols ***p*** and ***q*** (fig.1). Returning to the analogy of black and white balls, ***p*** could be interpreted as the property of being black and ***q*** of being large. In this case pq represents being *black* AND being *large*. According to Boole, if both operands are true the overall value is 'true' (fig.2); but if ***p*** is false (not black), or ***q*** is false (not large), or if both operands are false, then ***pq*** is false. In other words ***pq*** can only be true when both its operands ***p*** and ***q*** are true (i.e. 1.1 = 1).

An easier way to represent the possible combinations is to use a "truth table" (fig.3), each row of which shows the value of ***pq*** for given values of ***p*** and ***q***, but using 1s and 0s to represent true and false.

The 'OR' operator

In Boolean algebra '**+**' (not to be confused with the '+' used in arithmetic) signifies the OR operator. In our analogy of black and white balls, ***p* + *q*** represents the selection of items in the bowl that have the property of being black OR being large (OR both). According to Boole if either or both operands are true, then the overall result is true and we finish up with all the black balls, regardless of their size, and the large white balls. And, as is illustrated by the truth table at fig. 4, the equation 1 + 1 = 1.

Fig 2

pq

For the case when:

1. ***p*** is true (black) and ***q*** is true (large) then ***pq*** is true (i.e. a large black ball);
2. ***p*** is true (black) and ***q*** is false (not large), then ***pq*** is false (i.e. a small black ball);
3. ***p*** is false (not black) and ***q*** is true (large), then **pq** is false (i.e. a large white ball);
4. ***p*** is false (not black) and ***q*** is false (not large), then ***pq*** is false (i.e. small white ball).

The NOT operator

The NOT operator (fig.5) has just one operand, which it negates or "inverts"; in other words it transforms true into false, and vice versa. It is represented either by placing an inverted comma behind the inverted symbol (NOT ***p*** is written ***p***') or by placing a bar over it (NOT ***p*** being written $\overline{p}$).

A bright idea

With the exception of students of symbolic logic, Boole's work was to remain largely unknown and unused for over 80 years after his death until a research student at the Massachusetts Institute of Technology, who just happened to have studied both logic and electrical engineering, applied it to the construction of switching circuits. Reflecting on events some 50 years later, Claude Shannon's comment "*it just happened that no one else was familiar with both fields at the same time*" portrays commendable modesty.

Shannon was recruited by Vannevar Bush[1] to work on the maintenance of Bush's large analogue computer. Analogue machines no longer occupy a place in mainstream IT, so it's worth saying a few words about their role in solving complex scientific and engineering problems before the age of the digital computer.

Fig 3

p	***q***	***p* + *q***
1	1	1
1	0	0
0	1	0
0	0	0

Truth table for the 'AND' operator

Fig 4

p	***q***	***p* + *q***
1	1	1
1	0	1
0	1	1
0	0	0

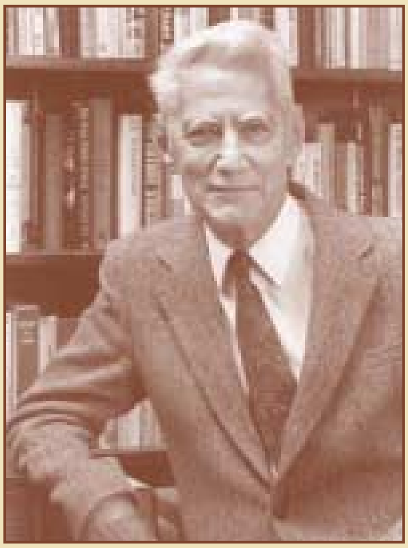
Truth table for the 'OR' operator

Fig 5

p	***p'***
1	0
0	1

Truth table for the 'NOT' operator

1 Professor Bush was to become a key figure is 20th Century American scientific development



Claude Shannon, 1916-2001
Photo: Bell Labs

The adjective analogue can be defined as "*of a circuit or device having an output that is proportional to the input*". The concept behind analogue computing is that instead of computing with discrete numbers, a physical model of the system to be investigated is built and its characteristics measured under different input conditions.

Before the coming of the digital computer, analogue machines were the only computational aides available for attacking such complex problems as current flows in the newly emerging national power networks, where the mathematical equations involved were too difficult to solve manually, assuming that they could be defined! Laboratory models or "analogues" were constructed from the various resistive, capacitive and inductive elements exhibited by such networks to study their real-world behaviour. And some were extremely complex; the AC Network Calculator constructed at the Massachusetts Institute of Technology during the 1920s to study current flow in grid systems took up an entire room.

Bush's *Differential Analyser* was a further example. Unlike other analogue computers, which were generally single purpose devices, the Differential Analyser was designed to attack a range of scientific and engineering problems that could be specified in terms of differential equations. It performed its calculations in decimal, rather than in binary, and like the slide rule and the clockwork watch, these were based on measurements of movement and distance. The machine used shaft movement to represent variables, gears to multiply and divide, and differential gears to add and subtract. It could calculate up to 18 independent variables, while integration was achieved using a sharply edged wheel spinning at variable radius on a round rotating table.

Just as Charles Babbage had planned to power his computer with a steam engine a century before, the only part that electricity played in the *Differential Analyser* was to drive its shafts. But despite looking back to the Babbage era, Bush's brainchild was in its time a marvel of scientific engineering, and several examples were built.

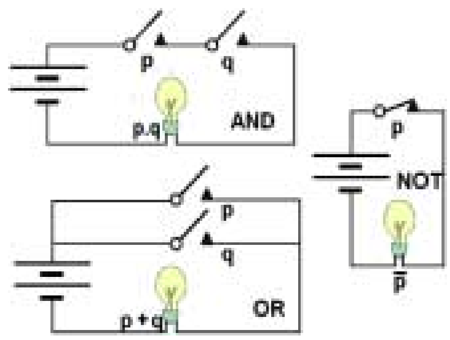
The *Differential Analyser* required a lot of maintenance. Its gears had to be manually configured to specific ratios before it could process a problem and Shannon was put to work on this monotonous task, while at the same time being encouraged by Bush to base his master's thesis on the machine's logical operations. Perhaps an inevitable consequence was that Shannon considered ways to improve the existing arrangements by replacing the purely mechanical parts with electric circuits laid out using the Boolean principles that he'd learned as an undergraduate.

Shannon completed his thesis in 1937 and in the following year published a paper based on it - "*A Symbolic Analysis of Relay and Switching Circuits*" - in which he demonstrated how to build logic circuits from electromechanical relays. The paper was hailed as brilliant and the ideas put forward were almost immediately applied to the design of automatic telephone switching systems.

Gates for channelling logic

Shannon was concerned with representing the Boolean operators 'AND', 'OR' and 'NOT' in terms of electro-mechanical circuits. He accomplished this by configuring relay contacts to conduct current (true) or not (false) according to which relays were energised (i.e. 'shut') or released (i.e. 'open'). For example, a simple 'AND' circuit (fig.6) requires both its relays (p and q) to be shut to light the lamp. If only one relay (or neither) is shut, the lamp will not light; these events correspond to the truth table at fig.3.

Fig 6



Similarly in an 'OR' circuit the lamp will light if either (or both) of its relays is shut, corresponding to the truth table at fig.4. In the 'NOT' circuit, relay p is fitted with a 'break', rather than a 'make' contact. The lamp is therefore lit (true) when the relay is open (false), and is extinguished (false) when the relay is shut (true - see fig.5). The analogy is rather like a succession of gates opening and shutting.

Electro-mechanical relays are useful for illustrating Boolean operators, but logic circuit designers now use standard symbols (fig.7) to represent the "logic gates", as they're called, used to compute Boolean functions. In common with the relay circuits in fig. 6, all have inputs and outputs that are limited to two values, 1 (true) and 0 (false); or in electrical terms, to set voltages (e.g. +5V and 0V).

Fig 7

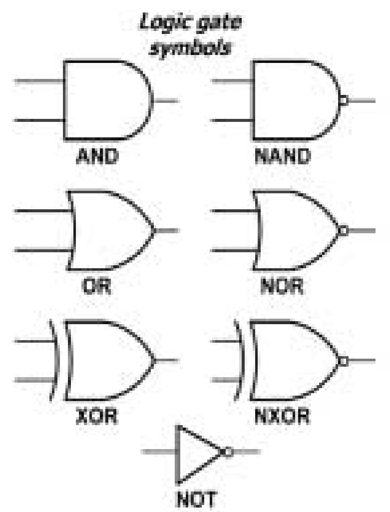
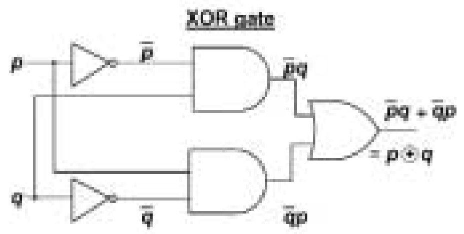


Fig 8



Although there are symbols for more than the Boolean operators AND, OR and NOT, all can be derived from these three. The NAND gate, for example, is simply an AND plus a NOT gate in tandem; likewise the NOR gate is an OR plus a NOT gate in tandem, whilst the exclusive OR gate - XOR - shown in fig. 7 is made up from the gates shown in fig.8. Unlike a conventional OR gate, XOR gives a true value if either, but not both, of its inputs are true (fig.9). One of its applications is in the circuitry used to add binary numbers in a computer's arithmetic logic unit. The symbol for the XOR operator is a circle containing a '+' sign, ⊕

Simplifying the problem

Simply stringing gates together to perform a logical function would lead to complexity and the wasteful use of components. The rules of Boolean logic can be used to avoid creating these problems by allowing complex logical functions derived from truth tables to be greatly simplified.

For example, the truth table in fig.10 represents the Boolean function A'B'C + A'BC + AB'C + ABC' + ABC. This could be implemented using five AND gates, five NOT gates and an OR gate, as shown in fig.11, but this scheme can be simplified and the number of components reduced to achieve the same end. There are different methods for doing this, but in this particular case simply manipulating the algebra using one of the rules that Boole defined (i.e. x + x' = 1) results in……

ABC + A'BC + ABC' + A'B'C + AB'C
= BC(A + A') + B'C(A + A') + ABC'
= BC + B'C + ABC' =

C(B + B') + ABC' = C + ABC'

The final step can be further simplified to C + AB by using another Boolean identity, x +x'y = x + y. Thus the original circuit reduces to one AND gate plus one OR gate (Fig 12).

World's smallest logic gate

Whereas Shannon worked with relays, vacuum tube circuits soon followed (not necessarily smaller, but very much faster), then discrete transistors (much smaller) and finally microchips, which seem continually to break new barriers of miniaturisation - according to a recent IBM research notice, scientists are now building logic circuits at the molecular level.

What are claimed to be the world's smallest working computer circuits use an approach in which individual molecules move across an atomic surface like toppling dominoes. The new "molecule cascade" technique enables working logic circuits to be constructed some 260,000 times smaller than those in advanced microchips.

The circuits were made by creating a precise pattern of carbon monoxide molecules on a copper surface. Moving a single molecule initiates a cascade of molecule motions, just as toppling a single domino can cause a large pattern to fall in sequence. Tiny structures were then created to demonstrate the fundamental Boolean OR and AND functions, data storage and retrieval, and the "wiring" necessary to connect these components into a functioning computing circuit. The most complex circuit built was a 12 x 17 nanometre (a billionth of a meter; the length of five to 10 atoms in a line) three-input sorter, so small that 190 billion could fit on top of a standard pencil-top eraser 7mm in diameter.

Computation is possible because each cascade carries a single bit of information. By analogy, a toppled domino can be thought of as a logical "1", and an untoppled domino a logical "0". Similarly, a cascaded or non-cascaded molecular array can represent a logical "1" or "0," respectively. The logical AND and OR operations, and other features needed for complex circuits, are created by cleverly designing the intersections of two cascades. Molecular arrangements have

Fig 9

p	*q*	*p xor q*	*p nxor q*
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

Fig 10

A	B	C	Out
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

Fig 11 Simplifying logic circuits

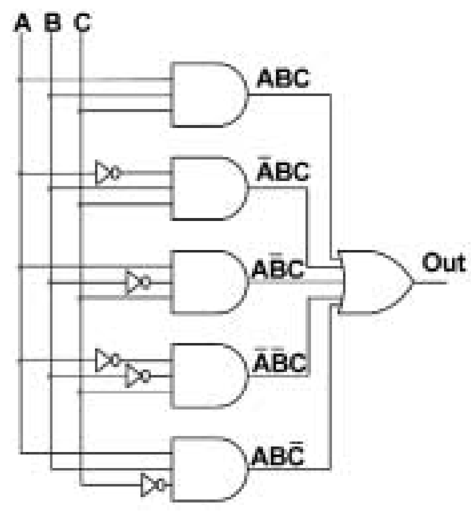
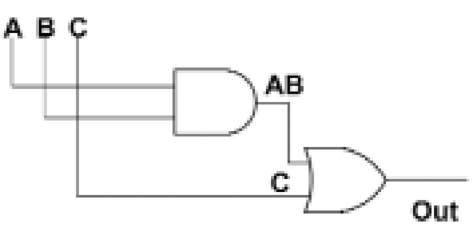
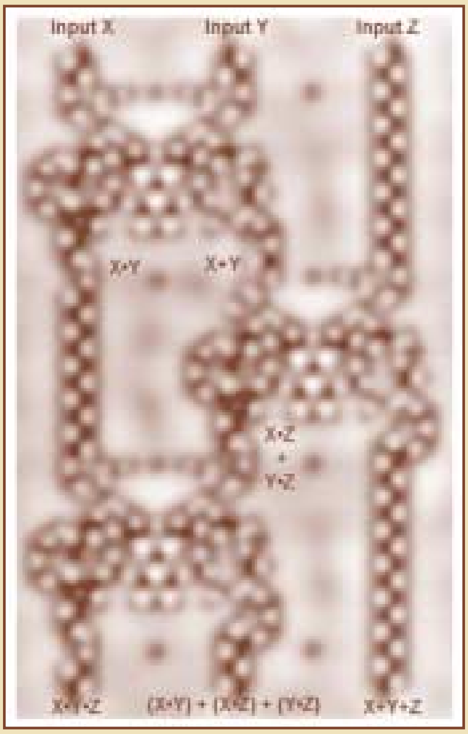


Fig 12 Simplifying logic circuits





World's smallest logic circuit - (Photo IBM)

been designed to act as crossovers (allowing two cascade paths to cross over each other) and fan-outs (splitting one cascade into two or more paths). Boole would surely be staggered by such developments!

Epilogue

Claude Shannon came to be widely regarded not just for his work on logic circuitry, but for solving technical and engineering problems within the telecommunications industry. After making the link between Boolean logic and switching circuits, he went on to undertake research at the Bell Telephone Laboratories on the problem of transmitting information more efficiently. In his paper "A Mathematical Theory of Communication" published in 1948, Shannon explained the communication of information in digital terms. The idea of transmitting pictures, words, sounds etc. as a stream of binary digits (1's and 0's) is something now taken for granted but at that time it had only been considered in analogue terms as the transmission of electromagnetic waves. The concept of digital transmission was fundamentally new. Although he went on to publish further research including important work on cryptography, Shannon's 1948 paper on digital transmission was to be the pinnacle of his achievement.

Shannon's work at Bell Labs led him to be regarded in his lifetime as the founding father of the digital communications age, but George Boole was less fortunate. He got soaked in a heavy rainstorm while walking from his home to college, where he then lectured in wet clothes before returning home to mark papers. Unsurprisingly George caught a cold. His wife Mary, believing that the remedy should resemble the cause, put him to bed, and since his illness had been caused by getting wet, poured buckets of water over him. Perhaps the inevitable consequence was that George contracted pneumonia from which he died, leaving behind the tools that would enable others to create applications of which he could never have dreamed. Taking account of the essential part played by digital circuitry in placing men on the Moon, it's a fitting tribute to George Boole that a lunar crater now bears his name.

Ian Petticrew

Postscript

Alicia Stott (1860-1940) was the third of George and Mary Boole's five daughters. Like her father she received no formal education in mathematics but this did not prevent her becoming well-known for her research in analytical geometry. In 1914 she was awarded an honorary doctorate by the University of Groningen in the Netherlands where her papers were published. A co-researcher described her thus: “The strength and simplicity of her character combined with the diversity of her interests to make her an inspiring friend.”

See also:

Freeware: software to construct and run your own digital circuit
http://www.spsu.edu/cs/faculty/bbrown/circuits/

Freeware: a software tool for simplifying Boolean functions using Karnaugh maps
http://puz.com/sw/karnaugh/index.htm

Working traffic light model - demonstrates each logic gate's switching function
http://users.senet.com.au/~dwsmith/beginners.htm

George Boole: "The Calculus of Logic"
http://www.maths.tcd.ie/pub/HistMath/People/Boole/CalcLogic/CalcLogic.pdf

Claude E. Shannon - "A Mathematical Theory of Communication"
http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html