

# Boolean castles in the air



“...In next month's *EW* I hope the editor will let me tell you more...”

Ivor Catt, *Letter to the Editor*,  
*EW* June 2004, p56 – he has

**F**irst I will give a cameo to illustrate how difficult innovation is, in an attempt to get back to the kind of ambience that would have existed during earlier eras, leading to today's multiple dysfunction in high technology.

## Cameo

Sir Clive Sinclair set up a company, Anamartic, to develop my Wafer Scale Integration invention, 'Catt Spiral'. It had previously been developed by UNISYS in Scotland, whose chief engineer told me he used hijacked money without the permission of UNISYS HQ in the USA. Then he moved to do the same work for Sinclair. Even though he had been a maverick, he thought it important that I should not be well informed as to the detail of the work being done to develop my invention. One day I gave him the slip, and came across a machine which could do "stitch bonding" across the face of a wafer. The engineer told me about its yield (reliability). This led later to my next invention, Kernel, which obsoleted Catt Spiral. The company, which had fired me, hired me back because of Kernel. This shows how important they thought it was.

Without reliable enough stitch, it was impossible to deliver both the electrical current needed for distributed processing across a wafer, and the global 100Mb serial data streams needed. The electrical resistance of the conventional Al conductors on a chip surface was too great. Decades before bonding, I had solved the obverse problem of heat extraction using on-

chip liquid cooling.

We can map this separation onto the situation in Bletchley and later Manchester, where there was an apartheid between mathematicians and engineers. This would have made it impossible for cryptographers like Turin, however brilliant, to contribute to the development of the general purpose computer, which was engineering hardware. The article by Lee Sallows, *A curious new result in switching theory* in *EW* May 2004, p32, prompted me to expostulate about logic design, the field that earned me my salary for decades. My article in *EW* in June 2003 is a cry over the unwarranted limit placed on computing power by allowing only one processor per machine. Now, in the case of Sallows, we can see a restriction at a much more basic level, similarly due to temporary historical engineering tradeoffs which were thought to have permanent significance.

In 1959, when I graduated in Engineering from Cambridge University, the head-hunters from Ferranti encouraged me to try working in the new digital computer industry. I joined the late Gordon Scarrott's labs in Ferranti Manchester, which was tied to Manchester University. The other important computer place was Elliot Bros., Borehamwood, which was tied to Cambridge University. Nowhere else mattered.

The Ferranti/Manchester team beat Cambridge, because the structure of their linkage between industry and university was better than the Cambridge structure. Apparently, in

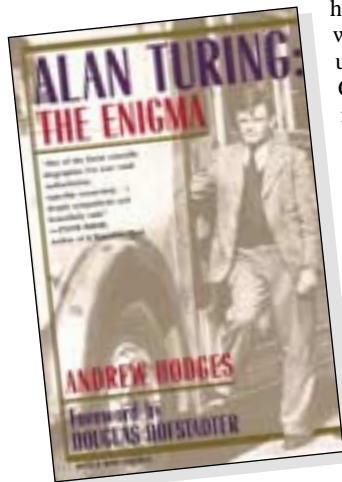
the Cambridge case, the university remained too much in control, which stunted them. The Ferranti/Manchester Atlas computer sold for a couple of million pounds, and competed with the big fast IBM machine, which was officially called 'Stretch'. Its name boasted that all its technologies had been stretched to their limit. When it failed to function, we called it 'Twang', and the world market came to our Atlas.

I did not hear the name 'Turing' until decades later, yet I now read that Alan Turing worked at Ferranti/Manchester University until he killed himself in 1954. His name was not on any of the documents I read and used, even though I did some of the design for the Ferranti Atlas. Why was there never any mention of Turing if, as we now persistently read, Turing was the genius who made massive contribution to our work designing and building computers? We did have a 'resident genius', but his nickname was 'Yanto', E T Warburton, not Turing. I never even heard of the 'Turing Machine' until decades later, though I worked in the Ferranti labs for three years. This is a useful sidelight on the bizarre schism down the middle of the subject called 'logic', or 'logic design'.

When you do a Google search for Turing + logic, you end up reading about Oxford Logic, about which more later. If Turing was the brains behind my work, but I find Oxford logicians name-dropping him, did Turing have a foot in both camps? The answer is probably that he had a foot in neither, but, like my hero T E Lawrence, his history has to be falsely rewritten now for PC reasons

which will be obvious to you.

The last paragraph is explained by my journey into the past while writing this article. In a letter dated 26 October 1948 the Ministry of Supply placed an order with Ferranti Ltd. "to construct an electronic calculating machine to the instructions of Professor F. C. Williams." The line between "mathematicians" and "engineers" was demarcated very clearly, and if not quite an Iron Curtain, it was a barrier as awkward as the MacMahon Act. This would never be Alan Turing's machine. This explains both why I never heard of Turing the mathematician, and also why we can ignore Turing when tracing the history of the development of logic gates. He had no access to those who were developing computer logic design. This also explains something I never before understood, which is why, when I arrived from Cambridge, I was met with such hostility by the engineers. By ignoring the fact that my degree was in engineering, and that the Cambridge physicist Ken Johnson was already there out-performing them, they would



have feared that I was yet another unpractical Cambridge mathematician like Turing trying to do engineering design. It also explains why they reversed the deflection plates on Ken Johnson's oscilloscope, and were gleeful when it took him a

week to find out what was wrong. The autumn of 1949 saw Alan's only titbit of hardware design for a Ferranti machine. His own electronic knowledge stopped short of the necessary practical detail.

In 1959, when I started doing logic design for the Ferranti Sirius Computer, I asked my boss, the late Charlie Portman, what books I should read. He replied that there were none. He said we were doing something totally new. So much for the influence of "Oxford Logic", an academic discipline which had a pedigree of centuries. Stargazers tell me that Sirius



Alan Turing

is a dog's leg. Our Sirius was the size and shape of an upright piano. Total main memory was 40,000 bits, so software was minimal. This article's text would just about fill our memory. We had an assembler from Machine Code into Assembler, which would then be printed onto punched tape, to be used as input when the programme was run. We

had no (real time) interpreter. A cabinet with three times more add-on memory cost what I would earn in ten years. One logic gate cost £5, half a week's pay. I did some of the logic design, including the 'Divide' instruction, which we added to entice

### ***All oranges are purple. It is purple. Therefore it is an orange. True of false?***

the reluctant customer to our £25,000 machine. Start with Dividend and Divider, and end with Quotient and Remainder. I did 'divide' by successive subtraction. Take the divider away again and again until what is left of the Dividend changes sign. Add one Divider back, and subtract one from the count of how many times, which becomes the Quotient. What is then left of the Dividend is the Remainder. You might think it sad that, correctly, nobody told me to look into what I call 'Oxford Logic'. I have never, ever, found useful overlap between Oxford Logic and my decades of salaried work doing logic design of digital systems.

When I do a Google search for Logic + implication, which latter is the only "function" that I remember from their world, I find hits for Turing! When I do a Google search for Turing + Boole, I get the book, *The Universal Computer: The Road from Leibniz to Turing* by Martin Davis.

What is "The Universal Computer"? Is it our kind of computer, or some confection of Oxford Logic? Mark Johnson, reviewing the book, writes:

"The first major advance came when George Boole developed an algebra of logic. His system was able to capture a fair amount of what might be called

everyday reasoning, but it still had limitations. Gottlob Frege was able to address these limitations, and in so doing, created essentially the system of first-order logic which we use today."

I have never heard of "first-order logic", although I designed computer systems for decades. "First-order logic which we use today"! Who uses it? So Turing is behind first order logic etc., and Turing is the genius behind the digital computers I helped to design. And I never heard of Turing until years later, and I never heard of "first order logic" until today.

Mark Johnson ends:

"Read this book. Have your friends read it. And remember both the logicians and the engineers the next time you boot up your universal computer."

Does he mean logic designers like me who designed your computer, or the Oxford logicians who bend the brains of their students?

I outline the nature of Oxford Logic as follows:

All oranges are purple.

It is purple.

Therefore it is an orange.

True of false?

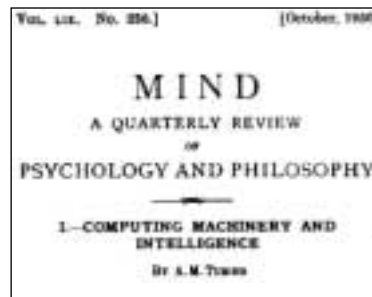
Since I usually earned my living doing logic design, or teaching it to students, my mind resists going through more than half a page of their stuff. Do the students who get sucked into their logic then run away, and become pinstripes in the City earning fat salary trading currencies? Or do they go on to teach younger victims about purple oranges? Arnold Lynch says that Colossus was not a computer, and it lacked memory. He also says that Turing was involved with a simpler machine and had nothing to do with Colossus.

I have concluded that since even in my time there was virtually no software in our computers because of the cost of memory, it followed that up until then,

although

mathematicians might have done brilliant work using their primitive computers, they would not have been able to influence computer hardware. In much the same way, however brilliant I

proved in my use of a hand calculator in 1980, I would not have had much influence on its design, particularly if I had little knowledge of its engineering. Last week, Arnold Lynch said that in the case of cracking German codes with Colossus, 80 or 90% of the challenge was in the hardware design



and construction. Turing, who Arnold says was probably the best mathematician at Bletchley, could only have influenced the other 10 or 20%, that is, developing procedures to solve problems using any available computing machines. Colossus was specified by Max Newman, who had Turing as a student in Cambridge. Arnold says that Bletchley rejected Colossus because of their lack of technical knowledge about valve reliability, and it was built by Flowers at Dollis Hill after its rejection by Bletchley. Here we see that lack of technical knowledge caused mathematician/cryptographers to obstruct architectural advance even towards special purpose computer systems dedicated to their own problem. As with stitch bonding, state of the art technical knowledge is indispensable, even for apparently special purpose machines, let alone general purpose.

When writing his article in *EW* May 2004, Sallows enters a murky world where political correctness has encouraged much rewriting of history, aided by the heavy secrecy surrounding Bletchley Park. However, even without the present urge to erase any achievements by white heterosexual males from history, he would have been misled. Sallows' "remarkably simple, highly intriguing, probably useless, but undeniably fundamental new result in switching theory" is to get two inverters, aided by numerous other Boolean logic gates, to perform the function of three inverters. Perhaps this challenge derives from the era when the transistor inverted, and the transistor was expensive. In contrast, I am concerned about very useful but suppressed aspects of logic design which Sallows tends to obscure even more. The reason is that Boolean functions are not fundamental, as I showed in my article published in February 1968, see [www.ivorcatt.com/47.htm](http://www.ivorcatt.com/47.htm), where I prove that the basic set of logic functions with one or two inputs totals four, the Inverter, the AND, the OR and the Exclusive-OR. Starting with a gate with one input, we find that one type only, the Inverter, is possible. Moving on to gates with two inputs where the inputs are treated the same, I show that the three basic gates are AND, OR and Exclusive-OR. All other possibilities are the inverse of my three, plus output stuck at 0 and output stuck at 1.

Lacking the Exclusive-OR, nothing which builds on Boole can be fundamental. This is not the fault of Boole, who intellectualised about his

kind of logic in 1850, not about the basics of the digital electronic computer in 1950 and 2000. Even in his own period he was at fault for missing the Exclusive OR, but not seriously so considering his objective, to clarify reasoning. In stark contrast, our billion-dollar industry wants to serve humankind without subjecting them to intellectual activity. The computer designer wants to get the hole in the wall to deliver cash to you aided by minimal thought and action from you, and without your having to

---

***Truth as a semantic concept.  
I should like to propose  
the name "the semantic  
conception of truth" for the  
conception of truth which  
has just been discussed.***

---

consider the nature of Truth, which is irrelevant.

Computer science did not emerge into view as a separate discipline from a cluster of related topics. Logic design emerged as part of digital hardware design when engineers strove to build practical machines. They came to think that short-term engineering convenience was based on fundamentals which for a time happened to reinforce the gap in Boole's set of logic functions.

I have checked back to find that circuitry was so expensive and small in number that machines like Colossus had virtually no logic design content. A few years later on, mechanical relays could most easily implement AND, OR and INVERT. The next generation of logic, using resistors and very expensive triodes, later expensive transistors, could most economically implement AND, OR, INVERT. The Exclusive OR remained more expensive to build.

Although I went on a training course to programme the last machine to use triodes, the Ferranti Pegasus, I did virtually no logic design with valves, beyond a three bit counter. My main logic design began with discrete diodes and transistors. A transistor cost £2, about a day's pay, while a diode was much cheaper at seven shillings. The ruling logic gate used a bank of diodes for AND or OR, and a restandardising transistor which insisted on inverting while doing so. This series of accidents caused the incompleteness of Boole's set to be overlooked. The Exclusive OR required two transistors, and so was ruled it out of the set for reasons of cost.

By 1965, the cost of transistors had

fallen enough to justify building the exclusive OR, but virtually nobody did. Its design relied on the fact that in order to conduct, a transistor's emitter and base must be at different voltages. One transistor would conduct for A and NOT B, while the other transistor would conduct for B and NOT A. Collector OR-ing gave the complete Exclusive OR. Only one person, the logic board designer in Data Products Corp., Culver City, noticed the engineering opportunity. I found it very useful, and this helped me to escape from the conceptual trap everyone had fallen into, starting with Boole and deepening because of short-term engineering tradeoffs with relays and valves.

### **Oxford Logic**

I went to see my co-author Dr Arnold Lynch this week and audiotaped him for two hours on his design work on the Bletchley Park Colossus, which Lynch said was not a computer and had no memory. Ninety years old on June 2, 2004, he is one of only two survivors from those who helped to design and build the machine, see *Electronics World*, June 2004, page 16. There were "need to know" secrecy barriers within the design team, but after the war, Lynch heard a lecture by the key designer, Thomas H. Flowers. Flowers said there was no mathematical symbolism in the matter of the machine's logic, whereupon Lynch suggested to him that he read Tarski, not knowing that Tarski was "Oxford Logic" (purple oranges). A single quote from Tarski will suffice: Truth as a semantic concept. I should like to propose the name "the semantic conception of truth" for the conception of truth which has just been discussed.

As my web pages show, my colleague Theocharis and I are very concerned about Truth, as his article in *Nature* proves. Our concerns do not map onto Oxford Logic. Further, both are orthogonal to my decades of work designing computers, where true and false are given, and never questioned. Tarski and Oxford Logic, and also my own concerns about Truth, have no place in digital computer hardware as it developed, and as it is today. My own suppressed article on Truth, and also Theocharis, can be found on my websites.

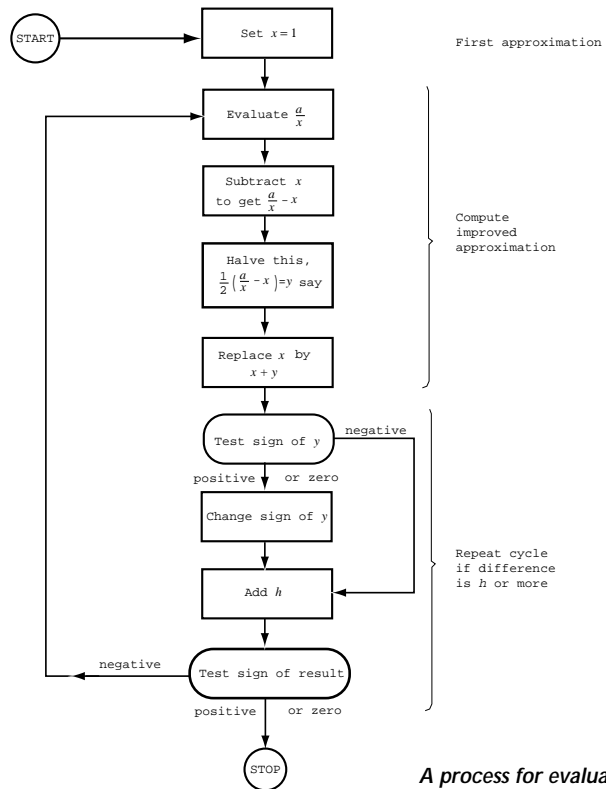
In your local bookstore, you can pay £40 for a book on Oxford Logic written by an Oxford Professor in 2002, presumably used as text in college courses on "Logic" to unsuspecting student victims. This will not give them access to the multi-billion dollar industry, digital

hardware, that my culture spawned, but which has admittedly now been driven abroad. After reading half a page of Oxford Logic, my head is spinning, and I stop. Oxford students must be chastened, deeply impressed by the Tarski tribe standing between them and their degree.

When I was Principal Lecturer in West Herts College and a member of the County Syllabus Committee, I tried hard to get rid of magnetic core memory from our Computer Hardware courses because I knew it had been obsolete for a quarter of a century. I failed, because all the other lecturers, although junior to me in status, succeeded in stopping me from removing what little they knew from the syllabus.

Oxford Logic has no relevance to the hardware behaving the way you want when you dialogue for money with a hole in the wall. My co-author David Walton, who later specialised in problems with large, complex arrays of software, may argue that it then has relevance, but that came much later when the cost of memory had fallen and made complex software possible.

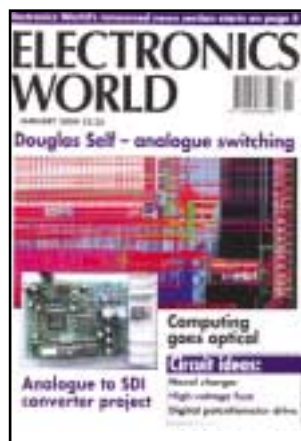
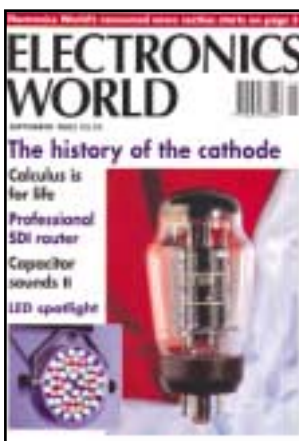
Sources for this article can be found at [www.ivorcatt.com/47.htm](http://www.ivorcatt.com/47.htm)



A process for evaluating  $\sqrt{a}$  ( $x$  is the result) from the Ferranti Pegasus programming manual 1962

# ELECTRONICS WORLD

## Back Issues



Back issues of Electronics World are available priced at £4 including p+p in the UK and £4 plus p+p elsewhere. Please send correct payment to:

Electronics World, Highbury Business, Media House, Azalea Drive, Swanley, Kent BR8 8HU

There are also a limited number of back issues from 2002 & 2003.