16 February 2012
Mathematics & physics

From an amp to an atom

26 September 2003

Tony Hoare
A Shortcut Through Time

Tony Hoare takes a guided tour of a quantum-powered future.

The most powerful computers in the world are designed for simulating nuclear explosions. They are built from thousands of microprocessors, each of them switching millions of microscopic silicon latches in parallel. They consume megawatts of electricity and live in buildings that sprawl over acres of the New Mexico desert. Their random-access memory is measured in trillions of bytes, and they deliver raw computing power measured in trillions of arithmetic operations per second (a trillion is a thousand billion, or a million million). With continuing advances in silicon technology and new and larger networks of processors, the computing power provided by such systems goes up by a factor of ten every few years. Their size and power consumption goes up too, but fortunately not quite as fast.

To make a quantum leap ahead of these almost-conventional supercomputers is the prospect offered by the quantum computer. In a quantum computer, the size of the individual computing element could go down to just a single atom. The collection of atoms, perhaps strung along a single molecule, would be transformed by commands issued by a program, perhaps through the medium of a series of laser pulses. The speed and the power consumption of possible future quantum computers are unknown. But they are almost irrelevant in view of the most remarkable fact of all: that the capacity and raw computing power of a quantum computer can be doubled by just adding a single computing element, just one atom, to the assembly.

With a single register of just 64 active latches, the raw computing power of a quantum computer could far exceed any conceivable conventional silicon multi-computer system, even one that occupied the entire surface of the globe, oceans and all.

In the prologue to his book, George Johnson gives a vivid account of the nature and promise of quantum computing. The ultimate secret of its miraculous doubling power is that a quantum register of (say) 64 bits can store much more than the single 64-bit number that would be stored in a conventional silicon register of the same size. In fact, a quantum register can simultaneously store any (non-empty) set of such numbers, quadrillions and quintillions of them, all at the same time. And a program defined to calculate a single result from a single input value, when it is applied to a quantum register, will simultaneously compute separate results for all the quintillions of numbers that are held in the register. And if that isn't enough, just add a single atom, and you will double the storage and the computing power of the quantum computer.

So what is the problem? There are many problems, and Johnson explains them well in the remaining chapters of the book. The first and most dramatic problem is that of output. At the end of a quantum computation, it is possible to read out only a single result from the quantum register, just one out of all the quintillions that it may contain. Furthermore, you cannot choose which of the contents of the register will be given to you - the number will be selected at random from all those that are there, simply by the act of reading it. So the art of programming a quantum computer is gradually to reduce that size of the set of numbers held in the register, preferably down to just one - the one that you want. Then there is no randomness in the choice.

A second difficulty is that the program can only be straight-line code. It is not possible to make any test of the value in the register and take different actions in different cases. In fact, any test will usually be passed by some of the numbers in the register and failed by other numbers.

So it is random on any given occasion whether the test will pass or fail.

And the test will have a rather surprising side-effect: if the test succeeds, all the stored numbers that fail the test will disappear from the register, and similarly in the case that the test gives a negative result.

A clever program for a quantum computer exploits this fact in ingenious ways. Without such clever programs, quantum computers will deliver none of their promised raw computing performance.

So far, there are only two such clever programs known, and neither of them helps much in simulating nuclear explosions. One of them does an associative search of a large database, whose records are all held simultaneously in a single quantum register. And the other one factorises a large number. With a quantum register of a thousand bits, factorisation has possible practical applications in deciphering the most widely used secret codes of the present day, those based on public-key cryptography. The possibility of fast quantum programs also has great theoretical potential in exploring the ultimate limits of the speed of solution of whole classes of hard problems on a computer.

Apart from the difficulty of writing the software, no one yet knows how to construct the hardware of a quantum register. There are three different construction technologies under investigation, some involving high refrigeration or a near-vacuum environment. About half a dozen latches is the maximum size of register so far built, and even then the register works only for less than a second. Fortunately, quantum devices can in theory be made to correct their own errors, at the expense of perhaps a tenfold redundancy.

Johnson explains these problems well while maintaining his optimism and exciting the enthusiasm of his readers. His explanations range widely over many important and interesting topics in computer science - Boolean logic, Turing machines, cellular automata, public-key cryptography, error-correcting codes and computational complexity, with a little bit of atomic physics and molecular biology thrown in. The book is refreshingly free from historical anecdote and descriptions of the appearance and personality of the leading scientists involved. But it does start with an engaging account of the author's own personal scientific development. He first acquired an engineer's interest in how things work by examining the diagrams of a Ron's Fender Deluxe Reverb amplifier in the days when electric guitars were built from thermionic valves.

Herein lies my one criticism of the book. It introduces many interesting topics, with a good account of motivation and background, and a series of striking analogies that converge on a more complex truth. But the explanation sometimes seems to stop short, just before the point at which the reader will exclaim the "aha, I now see how it works". For example, there are several pictures in the second chapter of computing circuits built from Tinkertoys but they are connected by cogs and pulleys that happen to form a deadlocked combination. And any hint of a mathematical formula is studiously avoided. Surely even the most non-mathematical reader might be interested in the fact that the quantum-factoring program is based on the familiar algebraic rule that n 2 - 1 = (n + 1) (n - 1) Johnson makes a virtue of the fact that he is not an expert in the field and that he acquired his understanding of quantum computing by searching the web and barraging scientists with emails. He commends his book as "an exploration, a questioning, an introspection, [a record of] a mind at work, imagining, spinning, struggling to understand". The reader is invited to join the author in "seeking a foothold on the granite face of a new idea".

I can certainly recommend him as an excellent introductory guide, leading by the most accessible paths to the most spectacular views.

Professor Sir Tony Hoare joined Microsoft Research in 1999 after more than 30 years as professor of computing in Belfast and at Oxford. He had no more expertise in quantum computing before reading the book than the author had before writing it.

A Shortcut Through Time: The Path to the Quantum Computer

Author - George Johnson
ISBN - 0 224 06233 6
Publisher - Cape
Price - £17.99
Pages - 204