21ST CENTURY TECH

CHAPTER 13 How Microchips Will Work

In the '60s of the late century, Gordon Moore made a prediction: The number of transistors that can fit on a single chip would double every 18 months. The observation has been enthroned as Moore's Law. Not to be confused with a law of nature, such as Newton's Laws, Moore's Law was just an uncanny bit of forecasting. Uncanny because at the time, Moore was unlikely to have foreseen some of the ways that would be developed to keep his prediction on schedule. But then Moore went on to co-found the microchip giant Intel. So he probably wasn't taking a stab in the dark. For the most part, the prediction has come true. The Intel 8088 chip, which came on the first IBM PC, contained 29,000 transistors. The latest Intel Pentium III has 9.5 million.

But in 1997 Moore made a revision in his law. He predicted that transistor miniaturization would hit a wall in 2017 or thereabouts. The wall is a literal one, the atoms stacked between adjacent *traces* in a microchip.

Traces are microchips' equivalent of wires that carry electrical current. Narrow beams of light are used to draw the pattern of the traces on a disk of silicon coated with a photosensitive coating. A chemical bath dissolves the film where the light struck the disk, and it's replaced with aluminum. With current microchip technology, the narrowest trace we can make is about .10 micros, roughly the distance across several hundred atoms. (A micro is 1 millionth of a meter.) And that's just theory. In practice, the Pentium III uses .25 micro-technology, and specialized IBM copper processors hold the smallest traces—.18 microns across—Intel plans to have .13 micron copper traces by 2001.

Chip makers have experimented with X-rays to draw traces instead of light because X-rays are much narrower than light waves. But as traces grow smaller, the problems increase. Gaps in the metal increase electrical resistance, blocking the flow of electrons. Copper, whose greater conductivity makes it a more efficient metal than aluminum at such Liliputian scales, is subject to corrosion and scratching. At some point the distance between traces becomes as important as the width of the traces themselves. If traces are too close to each other, the electromagnetic fields accompanying their flows of electrons begin to sap and distort the signals carried by adjacent traces. IBM recently introduced its *silicon-on-insulator* (*SOI*) technology that protects the transistors from stray electromagnetism. But even if you get all the kinks worked out—a way of drawing and manufacturing traces that carry current reliably and efficiently enough that a chip doesn't melt from it's own heat. Even if you do all that, you reach a point where the walls that separate the traces are only about five atoms across. At that scale, the insulating walls tend to spring electrical leaks as electrons spill out the sides on narrow turns. Then you have reached an absolute, physical limit to the number of transistors that will fit on a chip.

If there are limits to the capacity of microchips, couldn't we just create a sort of super microprocessor by joining several maxed-out microprocessors? Yes, we can, and we have. The arrangement is call *parallel processing* and it's currently the easiest, cheapest way to multiply computing power. Depending upon the microprocessor, anywhere from two to eight processors can be joined in parallel. Pieces of a calculation are broken into their constituent pieces and the pieces are fed to different processors at the same time. The output from the processors is joined together to produce the final calculation. It works, but the problem with parallel processing is that there a lot more overhead involved in handling the traffic among the processors, making sure that each processor produces results in time to join with the other results smoothly and with no lags. Microchips can only go so far along that path.

Why Silicon?

Of course, these limits are with silicon. We so closely associate the purified form of sand with microelectronics that they seem inseparable. But there are other ways to build microchips. One method, described by Ralph Merkle of Xerox Palo Alto Research Center, envisions microchips built of diamonds, complex, stable crystals that make it possible in theory to create the logical structures that make up transistors out of just a few atoms. Such a processor would contain more than a hundred billion billion bytes in a volume the size of a sugar cube.

Or we could give up on electricity entirely as a way to control the logic gates that make transistors. One scheme envisions light in place of electricity to carry signals and store data. Instead of transistors, the optical chip would consist of tiny towers of glass filled with layers of semiconductors. One laser beam strikes a tower and the light's energy releases electrons that become trapped in the semiconductors. The stronger the laser beam, the more opaque the glass becomes. When a second, stronger laser is aimed at the tower, the beam is either absorbed or reflected depending whether the tower is translucent or opaque. The reflected beam can represent either a 0 or 1, on or off, just as a transistor's switch does. Imagine several such layers of tower-studded glass with their attendant lasers on a single glass plate. Then imagine several plates stacked together. Data would be fed into one end of the stacks by shining multiple laser beams into one side and the results of the computation would be read as the pattern of lights that wind up coming out the other side.

Even the optical microchip would be built with several atoms making up each tower and its layers of semiconductors. But Danish scientists have reported creating a chip in which a single atom jumping back and forth could generate a computer's binary code. Using a scanning-tunneling microscope, the microelectronics center of the Danish University of Technology was able to remove, from a hydrogen layer surface on a silicon chip, one of a pair of hydrogen atoms attached to one silicon atom, leaving the remaining hydrogen atom jumping back and forth between the two layers. According to the leader of the experiment, Dr. Francois Grey, the information stored today on 1 million CD-ROMs could be stored on a single disk if bits were recorded at the atomic level. A practical application of the experiment is still at least a couple of decades away.

If the atomic level isn't small enough for you, there's one step left. The quantum computer takes us into the subatomic world in which the rules we consider the normal laws of science break down, where a subatomic particle can be thought of as existing in several different states at the same time. Only the act of measuring the particle forces it into one specific state. (If you think this at worst is impossible, at best incomprehensible, you're not alone.) Scientists at NEC wanted to find out if quantum states, rather than voltage levels, could be used to encode and process information. The team observed the wave behavior of quantum particles—in the process of merely observing somehow effecting the state of the particles—and found two waves that could correspond to two quantum states, establishing the binary on/off computing uses. But there's also an important difference from the usual binary concept.

The state of a quantum bit, or "qubit," is determined only when its value is observed. That means during a computation, each of the qubits in a string has potentially either a 0 or 1 value. Physically, they exist in an ambiguous "superposition of states" so a string of processor operations has the potential to simultaneously represent all possible strings of bits. Only when the calculation's value is observed, or read, does the string of bits take on a specific value. NEC scientist Jun'ichi Sone observed, "It takes one trillion years to factorize a 200-digit number with present supercomputers, but it would take only one hour or less with a quantum computer."

A microprocessor needn't be based on the binary bit technology that's the basis of all these concepts. An alternative is using the molecules that make up DNA. DNA normally exists as two long strings of only a few types of molecules called *bases* arranged in various combinations. Experimenters at the University of Southern California separated the two strings and arranged the molecules so that the strands would then be free to join with strands that have a complementary arrangement of molecules. Only certain combinations will join successfully. If different values are assigned to different bases, the successful joining represents a solution to a certain type of problem. The molecular soup microprocessor doesn't sound like it will be the smallest or swiftest of processors, but it's a good reminder that there are technologies as yet unimagined and untapped.

All these concepts are truly at the experimental level. Practical applications may never be achieved, and even if they are, they'll require new concepts for software and input/output. In the meantime, if Moore is right and his law runs out in 2017, by that time a single chip will have in the neighborhood of 19 trillion transistors, twenty times the number of neurons in the human brain.