The Boolean Logic Tax[1]

Erik P. DeBenedictis, Sandia National Laboratories

Moore's law relies on device size reduction for progress. But an energy tax due to Boolean logic properties could block this progress. There are alternatives.

Moore's law, which predicts a continuous rise in the number of transistors on a computer chip due to a continuous reduction in feature size [1], has been associated with worldwide economic growth. There are technical indications that Moore's law in its current form will falter in about a decade, raising questions as to whether a new growth driver will appear and, if so, what it might be.

Although progressive shrinking of device size underlies Moore's law, energy efficiency is the new limiting factor. Energy efficiency will no longer be determined by device size but instead by limits of the Boolean logic idiom used universally in system design. The future will need an alternative to Boolean logic for design in order for new device-level innovations to provide benefit.

FROM THE EDITOR

There's growing consensus that Moore's law, the technological driver behind computer hardware progress, will slow down or shift on a decadal timescale. In response, IEEE launched Rebooting Computing (http://rebootingcomputing.ieee.org), a multisociety initiative led by Tom Conte and Elie Track to "reboot" the computer industry through revolutionary ideas and approaches that question deeply rooted assumptions or defeat conventional wisdom.

As part of this important initiative, I'll be editing a bimonthly Computer column of the same name. The column will offer viewpoints on the maturing of Moore's law along with options for sustained growth in computer performance. Growth had been driven primarily by advances in the underlying technology (CMOS) and by software and applications enabled by more powerful hardware. However, for continued growth, research on materials and devices must be combined with a new computing approach that incorporates algorithm and software innovations. This column will draw on ideas from the full range of the technology stack.

In this inaugural column, I show how the ubiquitous design approach in computing—Boolean logic—imposes an energy tax on all circuits constructed with it. As a result, advances in device physics will become less effective over time if devices are used solely to create Boolean logic gates. Switching to non-Boolean gates could enable continued growth, but there are other approaches. —Erik DeBenedictis

Moore's Law

Let's review how Moore's law predicted semiconductor progress based on shrinking device dimensions. Chips obtain their functional

[1] Sandia National Laboratories approved for unlimited unclassified release SAND2016-3160 O
Published in DeBenedictis, Erik P. "The Boolean Logic Tax." *Computer* 49.4 (2016): 79-82. **DOI:** 10.1109/MC.2016.103. SAND2016-3160 O.

identity—such as being a microprocessor versus a graphics driver—by the specific Boolean logic network manufactured into the chip's layers. A Boolean logic network is a flow diagram showing how 0s and 1s, or false and true values, flow on wires between logic gates. Each gate computes a simple function of the values on its left-side inputs and transmits the computed value to the wire on its right side. For example, the NAND gates shown in Figures 1a and 1b produce a 0 output when both inputs are 1; otherwise the output is 1. A blue wire between the gates (Figure 1a) conveys the signal from one gate to the next, with the surface of a real chip containing many wires.

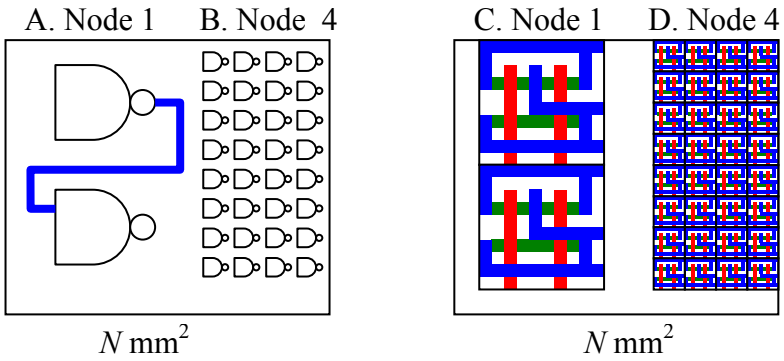

Figure 1: Illustration of Moore's law. (a) and (b) NAND gates produce a 0 output when both inputs are 1; otherwise the output is 1. (c) Simplified implementation of a NAND gate structured on the surface of a chip. (d) Scaling of linear dimension by four semiconductor generations shrinks line width by 4× (area by 16×), allowing the same area to accommodate 32 gates

Figures 1c and 1d show a simplified implementation of a NAND gate structured on the surface of a chip. Blue, green, and red represent layers of metal, diffusion, and polysilicon in a CMOS representation that's been virtually unchanged since the 1980s. Say that at a point in time, some region of a chip can accommodate two gates in a vertical stack (Figure 1c); Moore's law predicts that scaling of linear dimension across four semiconductor generations shrinks line width by 4× (area by 16×), allowing that same region to accommodate 32 gates (Figure 1d). This dimensional scaling has continued for many years, allowing the number of gates to rise to the billions found in today's microprocessors.

A key feature of Moore's law is that energy efficiency per gate rises at nearly the same rate as the number of gates per chip, allowing almost unchanging chip power dissipation as the number of components rises dramatically. The main contributor to heat generation is the energy in the 0 and 1 signals stored on wires. Each wire in Figure 1 assumes the value 0 or 1 in each clock period, split about equally between 0s and 1s. Based on today's ubiquitous CMOS circuit design, every time a signal changes from one state to another and back again, CV^2 of energy is turned into heat, where C is the capacitance between the wire (blue in Figures 1c and 1d) and its environment that's at a fixed voltage (ground) and V is the power supply voltage. The critical scaling argument is that the total amount of wire capacitance, or the cumulative area of the blue metal layer in Figures 1c and 1d, doesn't change as a result of scaling. This principle is visible to the eye: the blue features in Figure 1c cover the same percentage of the total area as the smaller but more numerous features in Figure 1d. This is because the patterns are the same.

Moore's law is slowing down and could stop, but not for the reasons commonly understood. For many years, the popular literature joked that wires and layers couldn't shrink forever because eventually they would be less than one atom across or thick. Size is no longer the primary challenge—energy dissipation is the new limiting factor. Even today, industry can make chips with so many transistors that the chips will overheat if all the transistors are turned on at once.

Theoretical limits based on communications theory

The Boolean logic network in Figure 1 has an energy minimum based on communications theory that will gain importance as we move into the future. Each gate output is like a little radio transmitter, sending 0s and 1s to the inputs of other gates. Rather than being free-propagating radio waves, the 0s and 1s are quite similar to signals propagating in a wire or coax cable. The signals have energy ½ CV^2 (so a 1 signal followed by a 0 signal comprises CV^2 of energy as before)—equivalent to the per-bit energy of a radio transmitter. The gate inputs treat A key feature of Moore's law is input signals like radio receivers, processing the received signal correctly as long as the signal is strong enough to overcome background noise.

This alternative explanation doesn't change the minimum energy dissipation from the previous interpretation in Figure 1 but shows why Moore's law is in crisis. Although the number of gates grows at a constant power per chip, the signals conveying 0s and 1s inside the chip are becoming progressively weaker—approaching the point of unreliable reception. This situation is roughly analogous to reducing the power of cell phone radio signals: eventually the signals will acquire static and become unusable. Just as Moore's law has not increased the range of radio signals, new computing devices will not change the minimum energy of signals in a Boolean logic network.

How close to the end are we?

Computing the CV^2 energy on a single wire is straightforward, but assessing the average value in a large system yields a result that depends on the distribution of wire lengths and, hence, capacitance, which varies based on the circuit design. In an extensive study of devices either in production or experimentally proposed [2], Dmitri Nikonov and Ian Young estimated that the lowest energy for a NAND gate constructed of experimental tunneling transistors was 5×10^{-3} fJ = 1,200 kT (k being Boltzmann's constant and T being ambient temperature of 300 kelvin; signal energies are henceforth expressed in units of kT because they're the natural units for discussing signal reliability). These projections used only near-neighbor wires, whereas David Frank included wire length distribution in a representative microprocessor over various devices in his simulation [3]. Frank's simulation showed 20,000 to 100,000 kT per signal; although these energies are higher than in Nikonov and Young, they were estimated for a more realistic environment [2].

The literature on the theoretical minimum energy for Boolean logic—including studies specifically on transistor scaling [4] and other studies repurposing results from communications theory [5]—yield similar estimates. Just as a cell phone picks up static

incrementally as it moves farther from the transmitter tower, the theoretical minimum energy of a Boolean logic gate is a function of the acceptable error rate, $P_e = \exp(-CV^2/kT)$.

Due to the exponential error dependence, CV^2 must be greater than 40 to 70 kT to be useful in a computer. This corresponds to an error rate range of approximately $10^{-30} < Pe < 10^{-17}$, which is the reliability limit for systems ranging from consumer devices to supercomputers operating for a few years without spontaneous errors. These are theoretical figures assuming absolutely perfect devices with no device-to-device variance and equal length wires; thus, practical devices will inevitably fall short of the limit by at least a modest factor, the most common number cited being $CV^2 \geq 100\ kT$.

So, how close is current technology to the end of Moore's law? The answer to the question is controversial because each factor of two is equivalent to a semiconductor generation that adds trillions of dollars to the world's economy. Because the stakes are so high, estimates are commonly adjusted up or down to match some agenda. This article provides the raw data and references for determined readers to form their own estimates. You can determine the available improvement factor by dividing one of the energies per operation for current technology described earlier (100,000, 20,000, or 1,200 kT) by one of the theoretical minimum energies immediately just described (40, 70, or 100 kT), being careful to understand the assumptions associated with the numbers. A reasonable low end might be Nikonov and Young's 1,200 kT for tunnel FETs divided by the 100 kT limit, or 12×.2 A reasonable high end could be Frank's 20,000 kT, which includes microprocessor overheads divided by the same 100 kT, or approximately 200×. This translates to a little less than four to eight generations.

Even without relying on alternatives to Boolean logic, the available improvement factor is enough to sell hardware. Any company that develops an improved transistor that makes a cell phone's processor 12× to 200× more capable for the same battery life will gain a significant market advantage.

However, the story is different for the overall economy. Users expect software and applications to continuously improve but new functions that benefit users are only possible with advancing computer hardware. The ubiquitous word processor is one example; its feature set has expanded in concert with advancing semiconductor technology and has produced a continuous stream of sales revenue. At the traditional rate of Moore's law, there are only four to eight more generations of advancement left.

Alternatives

Do other design approaches have higher energy efficiency limits than Boolean logic? Reversible logic gate families do, in fact, have higher limits, both for classical computing (bits defined as 0s and 1s) and quantum computing (quantum bits [qubits] defined by a bit value, a phase from 0 to 2π, and including entanglement with other qubits). However, no reversible technology demonstrations have come close to Boolean logic in practicality. The crucial question is whether current Boolean logic approaches are ahead because billions upon billions of additional dollars have been spent on their optimization (a head

start that will dissipate eventually) or because reversible approaches have yet-to-be-discovered complications that will forever prevent their practical realization.

Adiabatic and reversible logic essentially recycle CV^2 signal energy many times before dissipating it as heat. If, say, 99 percent of the signal energy in the output of one logic gate is recycled to become the signal energy of the next logic gate, the energy drawn from the power supply could be reduced by 100×. Adiabatic and reversible logic have many subtleties [6], one of which is that recycling can't be performed for Boolean logic gates (such as NAND in Figure 1) because of their functional behavior. Different gates are required (see Figure 2).

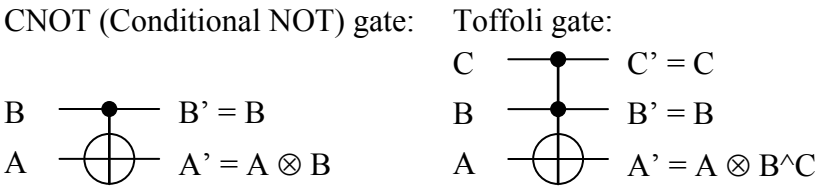

Figure 2: Reversible gates: (a) Conditional NOT (CNOT) and (b) Toffoli. These gates might be seen as a functional variant of Boolean logic gates by imagining that they convey signals from left to right, modifying the signals as shown by the equations

Figure 2 shows the Conditional NOT (CNOT) and Toffoli gates. These gates might be seen as a functional variant of Boolean logic gates by imagining that they convey signals from left to right, modifying the signals as shown by the equation. In English, the gates invert the main signal line on the bottom if, and only if, all the other lines are in a 1 state. They behave the same when running in reverse (signals moving right to left), which is part of their design's subtlety.

Quantum computing approaches use the higher computing capability of qubits compared to bits to solve problems in vastly fewer steps. Quantum computing requires reversible gates, but is otherwise beyond the scope of this discussion.

Spintronics represents a promising direction in energy-efficient computing but is still so raw and uncharacterized that it's difficult to categorize. Boolean and reversible logic discussions began with a concise mathematical theory of how to express computation through gate networks, and the mathematical theory ended up applying to many different physical realizations. However, computational structures in this new class have been proposed and, in some cases, experimentally demonstrated that achieve results beyond the limits of either Boolean or reversible logic. The problem is that no theory at present allows generalization of the specific examples into an overall approach to computing.

However, a common factor in these demonstrations is that devices contain memory (called state) and perform computation using that information as inputs to the computation. Because neither Boolean nor reversible gate sets include information in the gate, these demonstrations can't be represented by known gate sets. Nikonov and Young discuss a series of these approaches [2].

Until the last decade, students learned computer design in only one idiom: Boolean logic. Today, students learn alternatives including reversible computing, quantum computing, and neuromorphic computing. The Spintronics discussion indicates that even more new idioms can be added to the set. As with all science, certainties give way to new areas of research and progress.

It seems a split might be coming to the computer industry. The world's commercial enterprises have invested heavily in the technology stack comprising CMOS, Boolean logic, microprocessors, and software. This technology's energy efficiency might be limited by what we're calling a tax on energy imposed by Boolean logic idiom use, but the approach might nonetheless suffice for a vast number of applications and is likely to continue.

New computing applications are urgently needed but aren't yet practical because of computing power constraints. Some such applications might already exist in a less than optimal form, perhaps running offline on a server or supercomputer where the real impact would come from real-time execution in a self-driving car. A new computing approach will inevitably emerge at the logic level and join forces with new devices based on new materials, hopefully enabling new applications that would have the persistence of the word processor.

References

[1] Nikonov, Dmitri E., and Ian A. Young. "Overview of beyond-CMOS devices and a uniform methodology for their benchmarking." *Proceedings of the IEEE* 101.12 (2013): 2498-2533.

[2] Frank, David J., "Reversible Adiabatic Classical Computation – an Overview." *2nd IEEE Rebooting Computing Summit, Santa Cruz, CA* May 15, 2014, http://rebootingcomputing.ieee.org/images/files/pdf/7-rcs2-adiabatic-reversible-5-14-14.pdf

[3] Theis, Thomas N., and Paul M. Solomon. "In quest of the "next switch": prospects for greatly reduced power dissipation in a successor to the silicon field-effect transistor." *Proceedings of the IEEE* 98.12 (2010): 2005-2014.

[4] Bellamy, John C. *Digital Telephony (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2000.

[5] Feynman, Richard Phillips, J. G. Hey, and Robin W. Allen. *Feynman lectures on computation*. Addison-Wesley Longman Publishing Co., Inc., 1998.

[6] Moore, Gordon E. "Cramming more components onto integrated circuits, Electronics, 38: 8 (1965)." *URL: ftp://download. intel. com/research/silicon/moorespaper. pdf* 16 (2005).