Reed Electronics Group

MICROPROCESSOR REPORT

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

VIEWPOINT

THE MYTHOLOGY OF MOORE'S LAW

Why Such a Widely Misunderstood 'Law' Is So Captivating to So Many

By Tom R. Halfhill {12/13/04-02}

Moore's law gets more attention all the time. Google finds 223,000 hits for the term on the Internet, remarkable for something as arcane as semiconductor chip manufacturing. People who can't tell a silicon wafer from a compact disc don't hesitate to name-drop Moore's law at business lunches and parties, usually in the context of whether Intel stock is a good buy. Not since a falling apple led Sir Isaac Newton to discover universal gravitation have so many people been so captivated by a scientific law.

Yet Moore's law isn't really a law in the formal sense, and it isn't scientific. Indeed, it barely works for its intended purpose: describing the progress of component integration on affordable silicon chips. But that doesn't stop news reporters, commentators, analysts, and almost anybody with a calculator from applying Moore's law to things as disparate as microprocessor clock frequency, microprocessor power consumption, general computer-system performance, disk storage capacity, network bandwidth, digital camera resolution, or—in the most egregious example I've seen—the business fortunes of Netscape, a software company. Two years ago, the widespread and growing misapplication of Moore's law prompted me to define *Moron's law*: "The number of ignorant references to Moore's law doubles every 12 months."

As we approach the 40th anniversary of Moore's law in 2005, it's time to set the record straight. The facts are these: Moore's law is a narrow observation of a general manufacturing trend, not a law of physics; it wasn't clearly defined in the first place; its definition has been significantly changed over the years, both by its author and by trespassers, to make it better fit the actual data; and past performance is no guarantee of future results.

Understand that I'm not attacking Moore's law itself or its author, Intel cofounder Dr. Gordon E. Moore. My purpose is to counter the growing misconceptions about an interesting observation that, since 1965, has acquired a strange life of its own. Indeed, I believe there's something romantic about Moore's law that has propelled it into popular mythology.

The Origin of Moore's Law

One thing about Moore's law everyone agrees on: it dates back to the April 19, 1965, issue of *Electronics* magazine, in which Gordon Moore wrote an article entitled "Cramming More Components Onto Integrated Circuits." At the time, Moore was director of Fairchild Semiconductor's Research and Development Laboratories. Only four years earlier, Fairchild and Texas Instruments had introduced the first commercially available planar integrated circuits. And three years after publishing his landmark article, Moore helped found Intel. Moore defined his law at a crucial moment in the semiconductor industry.

Moore's 1965 article is widely available on the Internet, including Intel's website (see the "For More Information" box). It's only three and a half pages long, with two hand-drawn graphs, a picture of Moore, and an artist's cartoon of a future salesman hawking "handy home computers," which in 1965 were still in the realm of science fiction. Reading the article will be a revelation for those who enjoy offhandedly quoting Moore's law, because nowhere in the text does Moore explicitly state the law. In fact, the word "law" never appears in the article. The term "Moore's law" was coined years later by Carver Mead, a professor at the California Institute of Technology (Caltech). The closest Moore comes to expounding the law in his 1965 article is within a three-paragraph section under the subheading "Costs and Curves."

In that section, Moore discusses the component counts of integrated circuits manufactured at the most economical point on the semiconductor cost curve. In 1965, he notes, the most economical chips integrate about 50 components. By 1970, he speculates, the most economical chips will have about 1,000 components. And by 1975, he predicts, the most economical chips will contain about 65,000 components. Moore comes closest to actually stating the law when he writes, "The complexity for minimum component costs has increased at a rate of roughly a factor of two per year." His graph is the now-classic log-base-two plot, showing component counts doubling every 12 months.

In retrospect, it's possible to extrapolate Moore's law from that single sentence and graph, but it's definitely not the succinct version of the law popularly quoted today. After the article appeared, it remained for others to slice up Moore's careful engineering language and boil down his observation into a more concise statement: "The number of transistors on a silicon chip doubles every 12 months."

That rewrite is certainly more concise, but it's also less precise. To begin with, it severs the relationship between component *counts* and component *costs*. Moore was observing a trend in component counts for the most economical integrated circuits, which is quite different from describing the maximum number of components it's possible to cram on a chip at any given time. Early on, there began a trend of simplifying and generalizing Moore's law. Soon, it began slipping out of Moore's control and entering the public domain.

Moore added to the confusion, because the numbers in his article don't add up. Starting from a baseline of 50 components in 1965, if the component count doubles every 12 months, a chip manufactured in 1970 should have 1,600 components, not 1,000 components as Moore predicted in his text. His first forecast fell short of his own mathematical projection by 38%. Likewise, by 1975, a chip should have 51,200 components, not the 65,000 Moore predicted in the text. His second forecast overshot his mathematical projection by 27%.

Like the divorce between component counts and costs, these inconsistencies also started a trend. Since 1965, the relatively few people who have tried reconciling the forecasts of Moore's law with actual data have found the task frustrating—which is why the law has been restated in at least three different ways. There's something so fascinating about Moore's law that people want to make it work, whether it really works or not. Even Intel plays the game. The Moore's law page on Intel's website has the graph shown in Figure 1, which charts the actual progress of Intel's microprocessors, not the progress predicted by the law.

Of course, Moore wasn't defining a scientific "law" to be taken literally. Other people saddled his casual observation with that burden. Moore was merely pointing out a four-year trend and speculating where it might lead in the five- to ten-year future. Because Moore's law is an observation, not a law of physics, it cannot be held to strict accountability. It's not like Newton's law of gravitation, which identifies a universal constant. It's more like Bode's law, an observation by early astronomers that each planet in our solar system is roughly twice as far from the sun as the planet in the next inner orbit. Modern astronomers don't expect the distances between planets to add up exactly, and they don't expect other solar systems to conform to the same rule.

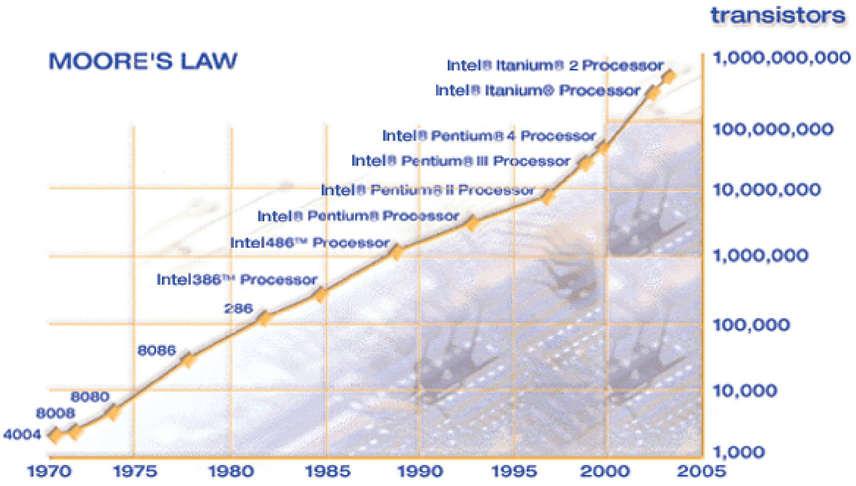

Figure 1. Intel uses this graph on its website (see the "For More Information" box) to illustrate Moore's law. But the graph tracks the steady progress of transistor integration for Intel's actual microprocessors—from the 4004 in 1971 to the Itanium 2 in 2004—not the progress of integration predicted by Moore's law.

The Evolution of Moore's Law

Since 1965, Moore's law has been revised twice in its most common form—once by its author, and again by others who keep struggling to make it fit the data. In 1975, at an IEEE meeting, Moore amended his law, which had already become famous within the semiconductor industry. Moore stretched the period for doubling components from the original 12 months to a more conservative 24 months. This significant modification reflected the slower progress of integration as engineers encountered new manufacturing challenges, which are most definitely governed by the laws of physics.

When progress accelerated again a few years later, Moore's law required additional tinkering to maintain it as a reasonably accurate description of reality. Moore didn't seem interested in continuing to bend the law, so other people jumped in. They resorted to a sophisticated mathematical method known as "splitting the difference." After looking at the original 12-month period for doubling component counts and comparing it with Moore's revised 24-month period, they settled on a compromise of 18 months. Since then, the 18-month period has become a widely quoted version of Moore's law, perhaps *the* most quoted version, even though Moore had nothing to do with it.

The unauthorized redefinition of Moore's law provided more flexibility for those who enjoy using the law to prove their points, whatever those points may be. With three time intervals to choose from—12, 18, and 24 months—people who knew a little more about arithmetic than about semiconductor manufacturing could pick the interval that best described their actual data or desired forecast. Given the nearly ubiquitous use of this technique, it's a wonder Microsoft hasn't built the various permutations of Moore's law into *Excel* as predefined functions—along with macros to automatically generate hockey-stick graphs.

Another source of confusion is whether the "components" Moore refers to in his 1965 article are synonymous with "transistors." Today, nearly everyone assumes Moore was discussing the scale of transistor integration. Even Intel measures the progress of Moore's law by this benchmark. However, Moore's 1965 article refers to the larger-scale integration of resistors and diodes as well as transistors, so he's using "components" in a generic sense. This broader definition makes it even more difficult to track the accuracy of the law. Transistor counts for chips are relatively easy to come by, but counts of passive components are not.

If transistor counts are the accepted benchmark, the original law is a wildly inaccurate predictor. Starting from a baseline of 50 transistors in 1965 and assuming Moore's original doubling period of 12 months, an economical microprocessor in 2004 should have 27.4 trillion transistors. That's dramatically more transistors than are found in the latest-model Intel Pentium M (170 million). Even Intel's newest Itanium 2 (Madison 9M)—which at $4,226 is hardly an economical chip—falls far short of the target, having "only" 592 million transistors. (Some critics might argue that slapping 9MB of L3 cache on Madison is a cheesy way to chase Moore's law. But the law doesn't distinguish between logic transistors and memory transistors, and larger caches can be as useful for improving performance as additional logic is.)

In fairness to Moore, he didn't expect his curve to remain accurate beyond 1975 or so, which is why he later revised the doubling period to 24 months. That revision pushes the curve back toward reality. Starting from a baseline of 50 transistors in 1965, the 24-month curve predicts that an economical processor in 2004 should have 37 million transistors—just about right. In contrast, the 18-month split-the-difference curve predicts 3.3 billion transistors. (On a log-base-two curve, small differences add up quickly.) Table 1 shows the differences among various versions of Moore's law and the actual progress of microprocessors.

Still more confusion ensues when people try to link Moore's law with Intel's manufacturing schedule. Intel's aggressive goal is to introduce a new chip-fabrication process every two years. If it was 0.13 micron in 2001, then it must be 90nm in 2003, 65nm in 2005, 45nm in 2007, 32nm in 2009, and so on. Surely, note some observers, it can't be a coincidence that this two-year cycle exactly matches the interval of one of the three versions of Moore's law—and that the company committed to the schedule is Intel, which was cofounded by the author of the law!

As a result, some people are reverting to Moore's 1975 definition, which pegged the doubling interval at 24 months. Just because Intel is rolling out a new fabrication process every two years, however, doesn't mean the transistors on its chips are doubling at the same pace. Moore's law doesn't address the ability of CPU architects to effectively use the expanding transistor budgets that the law predicts.

The Greatly Exaggerated Death of Moore's Law

The latest trend isn't to modify Moore's law but to pronounce it dead. This is a particularly popular theme in the mainstream press, and even in the trade press. References to the imminent demise of the law have been doubling roughly every 12 months.

There are at least three reasons for these ominous predictions of impending doom. First, the industry has been coasting on the coattails of Moore's law for 40 years, and there's growing uneasiness that the carnival ride is about to end. Second, companies like Intel decline to predict what their engineers will be able to achieve more than eight or ten years in the

	Transistors (Baseline 1965, Predicted for 2004)	Transistors (Actual in 2004)
Moore's Law (2x - 12 Months)	27.4 trillion	—
Moore's Law (2x - 18 Months)	3.3 billion	—
Moore's Law (2x - 24 Months)	37 million	—
Intel Pentium 4 (Prescott)	—	125 million
Intel Pentium M (Dothan)	—	170 million
Intel Itanium 2 (Madison)	—	410 million
Intel Itanium 2 (Madison 9M)	—	592 million

Table 1. Three different versions of Moore's law predict very different results over the past 40 years. Moore's original 1965 law observed that component integration doubles every 12 months, but that curve was too aggressive to remain valid for long. Moore's revised 1975 law extended the doubling period to 24 months, which has turned out to be the most accurate curve. The widely quoted, but unofficial, 18-month version of the law is significantly off base, though it's better than the original 12-month version.

For More Information

Dr. Gordon E. Moore's original 1965 article in *Electronics* magazine is available on Intel's website:
- *ftp://download.intel.com/research/silicon/moorespaper.pdf*
Intel also devotes a web page to Moore's law:
- *www.intel.com/research/silicon/mooreslaw.htm*

future, which some people misinterpret as an expiration date. Third, Intel's recent retreat from higher clock frequencies for its Pentium 4 processors and the worsening problem of power consumption have led some people to mistakenly conclude that Moore's law is almost ready to take its place in history alongside the Code of Hammurabi.

Fear mongering isn't wholly to blame for this rising dread of chipocalypse. Indeed, the first reason for anticipating the end of Moore's law is the least hyperventilated: the law can't last forever. *Duh!* Otherwise, silicon chips would eventually have more transistors than there are grains of sand with which to make them. But we've known that since 1965. Today, all we know for certain is we're 40 years closer to the end of Moore's law. We still don't know the distance to what I call Moore's *wall* (the reverse of Moore's law).

Reading doom into the reluctance of Intel and other companies to make far-out predictions isn't very useful, either. It's perfectly understandable that they can't anticipate what their engineers are capable of doing much beyond eight or ten years in the future. It's always been that way. We'll have to wait and see.

The third reason for declaring the death of Moore's law—the slowing pace of clock-speed inflation and the growing problem of power consumption—forgets that the law describes component integration, not clock frequency, dynamic power, or static current leakage. There's every reason to believe future multicore processors will continue integrating more transistors than today's processors do, even if their clock speeds don't climb at the heretofore feverish pace. And there's every reason to believe we'll continue finding better ways to control power consumption, which doesn't necessarily scale at the same rate as component integration.

In microprocessor design, what matters is throughput, not clock speed. Multicore processors in servers, supercomputers, and embedded systems have already proved that chip multiprocessing is as valid for improving performance as caffeinating the clock frequency is. Multicore designs are just another way of leveraging higher-scale integration—and remember, higher integration is the point of Moore's law.

The greatest shortcoming of Moore's law is that it doesn't acknowledge another informal edict: the law of diminishing returns. Moore's law is an infinite log-base-two curve without a shoulder, so it never reaches the inevitable plateau. More important, it doesn't suggest how close we are to the plateau, which would be useful to know. Dave Epstein, a long-time member of the *Microprocessor Report* editorial board, proposes a solution he modestly calls Epstein's amendment: "Starting in 1970 with the predicted doubling every 12 months, the interval will increase by six months every ten years."

In other words, assume that in 1970, Moore's law was still chugging along at a rate of 2× every 12 months. By 1980, it slowed to 2× every 18 months. By 1990, it was 2× every 24 months; by 2000, 2× every 30 months. Epstein's amendment adds a leveling factor that accounts for the law of diminishing returns. Although his factor doesn't quite fit the actual data, some juggling with a spreadsheet should whip it into shape. Maybe the interval increases by six months every eight years or eight months every six years. I'll leave this problem as an exercise for the obsessive reader.

The Popularity of Moore's Law

As we near the 40th anniversary of Moore's law, it's amazing to see how a cost-component curve plotted in 1965 to describe semiconductor manufacturing has so thoroughly penetrated popular culture. Moore's law isn't just for engineers any more. References keep appearing in more and more places with increasing ambiguity, sort of like crop circles in wheat fields. And, as with those wacky circles, everyone is eager to offer a personal interpretation. Today, Moore's law, in all its bastardized forms, belongs to everyone.

I believe Moore's law has entered popular mythology because it's so compelling. Who can resist a 40-year-old prediction that practically promises chips will get better on a regular schedule, as if by clockwork? To a great extent, the law is self-fulfilling, because it gives manufacturing engineers (or at least their whip-cracking managers) a target to aim at. We landed a man on the moon before 1970 because we said we would. We double the transistors on a chip every 24 months because we say we will. Nothing motivates like a deadline.

No other mathematical curve captures the popular imagination the way Moore's law does. It shoots skyward like a rocket, year after year. Population-growth curves are impressively steep, too, but the endpoints are frightening to contemplate. The zigzag charts of economists and accountants are unreliable predictors of the future and have too many sharp spikes to be user friendly. The gentle bell curves of psychologists and social scientists are as reliable as Moore's law, but they're often depressing, because they usually describe unfair distributions of the human condition.

Perhaps the most attractive attribute of Moore's law, then, is its time-tested optimism. At least *something* in the world is getting twice as good at regular intervals brief enough to perceive within our lifetimes. The hope inherent in Moore's law could be the best explanation for all the clumsy attempts to apply the law to things having nothing to do with component integration on chips of silicon. ◇