Written by
Nick Tredennick

# DynamicSilicon

Published by
Gilder Publishing, LLC

The Investor's Guide to Breakthrough Micro Devices

# MEMS and Dynamic Logic: Why Now?

"You guys have been promising dramatic advances in robotics for fifty years—and you've never delivered! Why should I believe you this time?" I recently watched a presentation on soon-to-be-seen advances in robotics and this was the response from a skeptic in the audience. I don't remember the presenter's reply, but I remember thinking "There is a difference. Today we can put a million-transistor microprocessor in a robot's knee joint for two dollars."

A similarly skeptical remark might be made about the markets for MEMS (microelectromechanical systems) and for dynamic logic. MEMS sensors have been around since the '50s. Dynamic logic has been around since the mid-60s. What leads us today to conclude that the market for MEMS or for dynamic logic will soon grow rapidly? What's different? Everything. Let me walk you through my favorite milestones in semiconductor electronics.

Look at the size of the worldwide semiconductor marked in fig. 1. It's the backdrop for my milestones and their consequences.

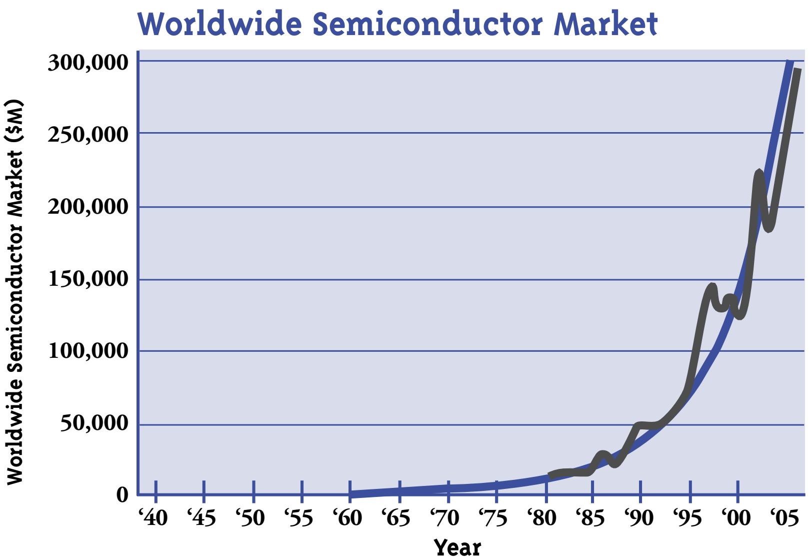


**Fig. 1.**
This is the growth of the worldwide semiconductor market (starting at zero when the transistor was invented in 1947). The smooth curve is a 16 percent compound annual growth rate beginning in 1959. The erratic curve plots the numbers and estimates published by the Semiconductor Industry Association.

According to the Semiconductor Industry Association (SIA), the worldwide semiconductor market in 2000 was about $226 billion. That's better than 1.5 times what it was in 1995 and more than four times its 1990 value. Despite its boom and bust cycles, the semiconductor industry has averaged 16 percent CAGR for forty years. At this growth rate, the semiconductor market doubles every five years.

That's one possible answer to the question of MEMS and dynamic logic growth: since it's part of the semiconductor market it'll double every five years. But that's not good enough. We're here

In This Issue:

to forecast dramatic changes, and something that doubles every five years, as delightful as that would be in most industries, is business as usual in semiconductors. First, I'll need to set the context for change. I'll do this by taking you through milestones in the electronic systems business and by offering a simple model of the electronic systems market. With the model, we will see which market segments have driven the market historically and, in the context of the milestones, which markets will emerge. Our destination: "MEMS and dynamic logic." The journey through this brief history should bring you to the conclusion that it makes sense.

**The computer**. The computer, invented about 1940, signified a breakthrough in problem-solving methods. Before the computer, engineers solved problems directly. That is, engineers solved problems by building special hardware. The hardware is a direct representation of a mathematical algorithm. Money limited the range of affordable solutions; if you couldn't afford the hardware, you couldn't solve the problem. The computer separated algorithms from hardware. The computer is a limited set of expensive, general-purpose computing resources. The algorithm, in the form of a program, resides in a cheaper attached memory. The computer can solve large problems by *iterating,* and since it has *general-purpose* resources, it can solve a range of problems. The computer amortizes the cost of expensive hardware either over time or across a range of problems.

The legacy of the computer's invention that is important to the worldwide semiconductor market, is that the computer changed the way engineers solved problems. Instead of building special hardware to solve a problem, engineers programmed the general-purpose hardware of a computer. Problem solving became programming.


DynamicSilicon

| | |
|---|---|
| Editors | Nick Tredennick |
| | Brion Shimamoto |
| Publisher | Lauryn Franzoni |
| Designer | Julie Ward |
| President | Mark T. Ziebarth |
| Chairman | George Gilder |

DynamicSilicon is published monthly by Gilder Publishing, LLC. Editorial and Business address: 291A Main Street, Great Barrington, MA 01230.


Editorial inquiries can be sent to: bozo@gilder.com Single-issue price: $50. For subscription information, call 800.229.2573, e-mail us at dynamicsilicon@gilder.com, or visit our website at www.dynamicsilicon.com

**The transistor**. Invention of the transistor in 1947 at Bell Labs heralded the end for the vacuum tube. The transistor was smaller and more reliable than the vacuum tube and it quickly displaced the vacuum tube in electronic systems. The transistor, because of its reliability, small size, and low-power consumption, made small portable electronic systems (like transistor radios) practical. The transistor accelerated the penetration of electronics into systems.

**The integrated circuit or "chip."** Engineers at Fairchild Semiconductor and at Texas Instruments invented the integrated circuit (IC) in 1959. Fairchild introduced the IC commercially in 1961. Since then, the IC and the semiconductor manufacturing process that supports it have been the magical engines of growth for the semiconductor industry. In 1965, Gordon Moore predicted that the number of transistors on a chip would double every year. For the first ten years or so, it did. Later the rate was modified to doubling every eighteen months (CAGR: 60 percent) and the formula entered folklore as "Moore's law." Moore's law says how many transistors fit on a chip; it says nothing about the performance of the chip though that's often the interpretation. Also, Moore's law isn't a law at all; it's a measure of the rate at which improvement is driven by competitive pressure. That is, the industry has decided to set its treadmill (consecutive design points) to run at this rate.

Even a conservative estimate of this growth rate, using microprocessors from the 2,300-transistor Intel 4004 in 1971 to the 40 million transistor microprocessors of last year yields doubling every two years (CAGR: 40 percent). How? Smaller circuit geometries primarily (halving the feature size quadruples the number of transistors), but larger chips and better circuit designs also contribute.

IC improvements mean (a) more capability and more performance for a fixed cost, or (b) lower cost and lower power dissipation for the same function. As fig. 2 illustrates on the next page, the cost to implement the same function drops by half every two years (historically costs have dropped faster than this). Similarly, the amount of function implemented at a fixed cost doubles every two years. Both curves extend the IC's application domain. If last year's microprocessor was just powerful enough for optimal tooth brushing algorithms, but was too expensive for the cost-sensitive consumer electronic toothbrush market, the same processing capability will be half as much next year. If that still isn't cheap enough, it'll be half as much again in two more years. Eventually the microprocessor will invade the electronic toothbrush, as indeed it has.

Before the IC, engineers designed systems with discrete components: resistors, capacitors, inductors, transistors, and diodes. The IC brought macro components—so-called macro functions—to the engineer, greatly improving engineering productivity at some cost in design efficiency. Macro functions are adders, shifters, registers, and multiplexers. For the most efficient design, each transistor would be sized for its position in the circuit. As the engineer places macro functions, efficiency is lost in sizing individual transistors (output transistors, for example, are designed to drive ten loads) and efficiency is lost in the exact fit of macro functions (there may be unused gates in a logic block). Families of compatible macro functions enabled the engineer to design systems with the electronic equivalent of Lego blocks. IC macro functions quickly displaced the transistor and other discrete components.

**The PLD**. The programmable logic device (PLD) was the next important breakthrough in semiconductors. Sven Whalstrom, whose fundamental patent (#3,473,160, *Electronically Controlled Microelectronic Cellular Logic Array*) was filed in 1966, invented the programmable logic device. The PLD did for IC macro functions what the computer did for electronic hardware design: it allowed the engineer to *program* connections among macro functions. Here's how it works. Conceptually, the PLD is a two-layer device. One layer is an array of logic blocks and interconnected segments (wires). The second layer is memory that specifies the connections between logic blocks and wires that build arbitrary circuits. The engineer, through a software interface, specifies the content of the memory and, thereby, the function implemented by the PLD.

Unfortunately for Sven, his invention came too early in the life of the IC to be of immediate use—there just weren't enough transistors on an IC for cost-effective application. The PLD had little effect on the semiconductor market in 1966, but thirty-five years of progress in semiconductor manufacturing will finally make Sven's breakthrough important.

**The microprocessor**. IC macro functions grew in complexity until the microprocessor became practical. Lee Boysel's AL1, the first microprocessor in a commercial product, appeared in a data terminal from Four Phase Systems in 1969. Boysel and Murphy describe the AL1 in an article in the April 1970 issue of *Computer Design* magazine ("Four-phase LSI logic offers new approach to computer designer"). In 1971, Intel introduced its 4004, the first commercially available microprocessor.

The microprocessor was invented for embedded applications. That is, it was invented as an invisible component in an electronic system for which the main system function was *not* general-purpose computing. Microprocessors were the ultimate integrated circuit. The microprocessor allowed engineers to apply the conceptual breakthrough of the computer to electronic system design. Using the microprocessor and a few standard "peripherals" in the form of ROM, RAM, and input/output chips, engineers built electronic systems and wrote programs to implement the algorithms. The microprocessor let engineers trade design efficiency for engineering productivity. Increasing the level of abstraction (designing with C or C++ rather than with discrete components) raises engineers' productivity but lowers the efficiency of the implementation. Lowering the efficiency of the design is OK if there is sufficient power for the circuits and enough time to complete the task. Most embedded systems, it turned out, were sensitive to cost, but were not sensitive to time or to the availability of power.

Because microprocessor-based systems were programmable, they worked across a broad range of applications. This led to high volumes for the microprocessor and its peripherals, which drove down the cost of these components. Lower cost led to more

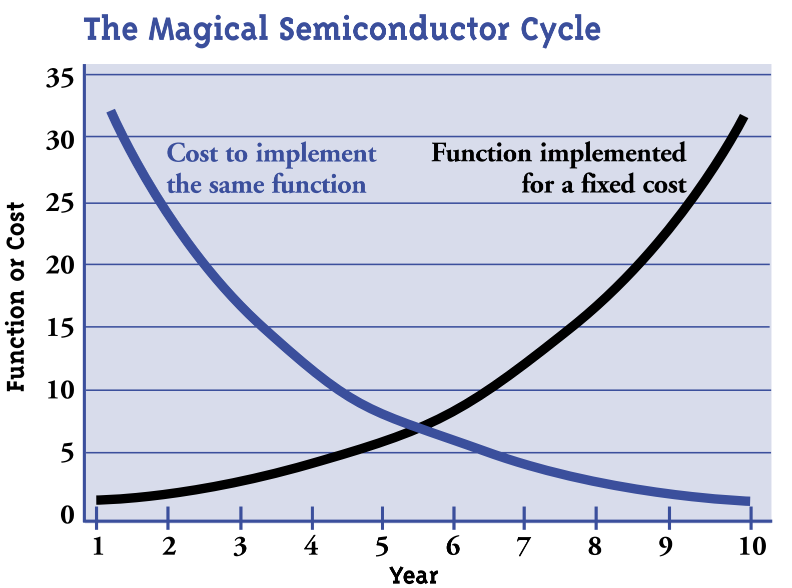


**Fig. 2.** The cost to implement the same function drops by half every two years. The amount of function implemented for a fixed cost doubles every two years.

applications. Microprocessor-based systems displaced designs based on IC macro functions. Fig. 3 shows that microprocessor shipments have grown to truly astounding unit volumes. For the microprocessor, I fitted the curve backward to 1971 from known numbers in 1995 and 1996 and then conservatively projected forward at the recent growth rate for personal computers. For the personal computers, I projected forward based on estimates for the years 1995 through 2000 and fitted the curve through known numbers in 1981 and 1995 for the intervening years (not that it shows). Last year, manufacturers shipped about seven billion microprocessors. That's more than one microprocessor for every person on the planet. Moreover, in the next three years, manufacturers will ship more microprocessors than the total shipped since its commercial introduction thirty years ago.

### Growth of the Microprocessor

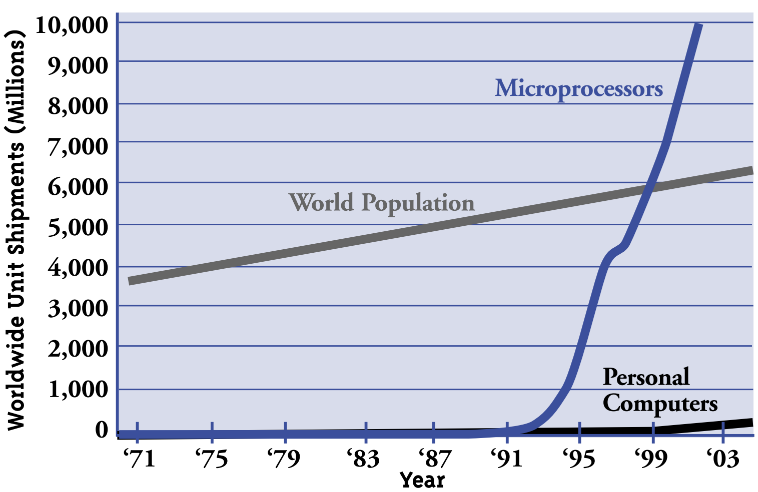


**Fig. 3.** Since its introduction in 1971, annual worldwide shipments of microprocessors have grown to exceed the population of the planet. By contrast, the personal computer will ship about 150 million units in 2001.

The microprocessor has invaded everything from toothbrushes to transmissions. It's everywhere. Further, the culture of microprocessor-based design is entrenched in the engineering design community, in the educational system, and in commercial manufacturing interests—a point I will return to shortly. The market for the microprocessor, which here also includes microcontrollers and digital signal processors (DSPs), will continue to thrive and to grow rapidly.

If you think of the personal computer when you think of microprocessors, you are sort-of right and sort-of wrong. The personal computer dominates the world electronics market, it dominates press coverage, and its CPUs dominate microprocessor revenues. CPUs for x86-based personal computers collect almost 90 percent of worldwide microprocessor revenues. But CPUs for personal computers are a nit, in unit volumes, at less than 2 percent of the total.

**The IBM PC**. Microprocessor-based computers had been around since 1974, but were viewed as objects for nerd hobbyists until IBM introduced its Personal Computer in 1981. IBM sold 15,000 units the first year, launching the market for personal computers.

The personal computer is important because it changed the microprocessor's design focus from low-cost to performance.

**The million-transistor IC**. IBM announced the 1 Mb DRAM in 1984. A million transistors on an integrated circuit. That's a milestone because it signaled the onset of practical PLDs. Sven's programmable logic devices had been all but forgotten in the avalanche of microprocessor-based systems. About the time Sven's patent expired, Altera (ALTR) (1983) and Xilinx (XLNX) (1984) started to exploit the potential market for PLDs.

**The one-micron process**. About 1988, the CMOS semiconductor process arrived at 1-micron geometries. In its continuing march to finer geometries, the semiconductor industry paid for the development of 1-micron processing equipment and then left it behind to pursue the next generation. Semiconductor manufacturers donated processing equipment to universities or sold it at bargain rates to make room for the next generation. The ready availability of 1-micron processing equipment greatly aided the development of micro-electromechanical systems (MEMS).

**The Palm Pilot**. The Palm Pilot, introduced in 1996, became the first highly successful, personal digital assistant. The Palm Pilot signaled the beginning of high-growth for compute-intensive portable devices.

### Putting it all together

In fig. 4 on the next page, you can view these key developments against the backdrop of the worldwide semiconductor market. Here's where we are. The computer split engineering design into two domains, direct implementations and programmed implementations. The transistor displaced the vacuum tube, made portable systems practical, reduced system cost and power consumption, and significantly improved reliability. The integrated circuit displaced discrete components in systems because it enabled IC macro functions, which improved designer productivity. The PLD was a conceptual breakthrough that had no practical consequence at the time it was invented. The microprocessor displaced IC macro functions and brought the computer's programming methods to embedded systems problems. The IBM PC changed the microprocessor's design focus from low-cost to performance. The million-transistor IC signaled practical utility for PLDs. The industry's advance beyond the 1-micron semiconductor process left the equipment needed for MEMS experimentation and development. Finally, introduction of the Palm Pilot signaled the bifurcation of the market into tethered and untethered devices.

### Workstations and personal computers

Fig. 3 plotted the growth of the microprocessor market. It also showed the personal computer market. By my calculation, manufacturers will ship more than eight billion microprocessors in 2001, but only a little more than 150 million personal computers. Unit volumes for the PC won't even be noticed among eight billion microprocessors. Still, we might be tempted to forecast the market for electronic systems by looking at the PC since it dominates revenue. That would be a mistake. We need to identify trends. To do that, we'll need a supply-side view of electronic systems applications. We need a taxonomy for the applications those eight billion microprocessors disappear into. I use a simple zeroes model, which characterizes electronic systems applications by their dominant characteristics. The zeroes model, shown in fig. 5, is four overlapping segments defined by their design requirements.

### Worldwide Semiconductor Market

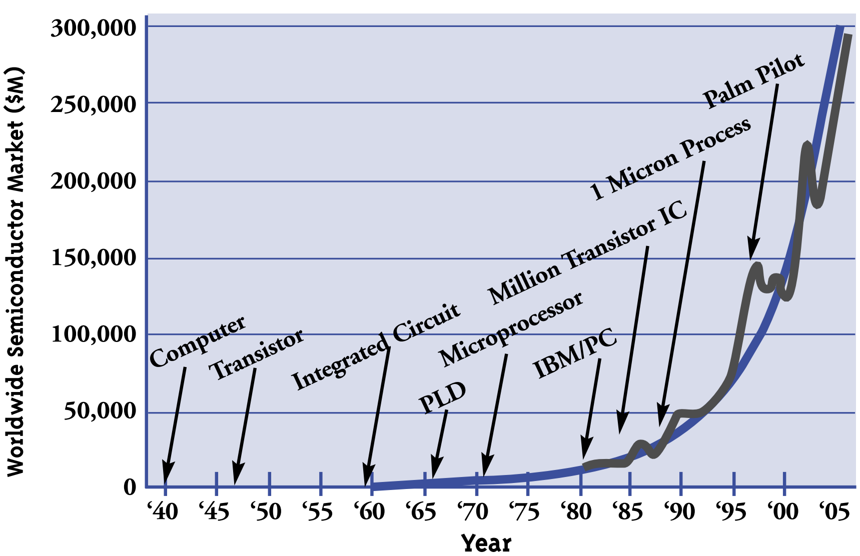


**Fig. 4.** Semiconductor market milestones plotted against the backdrop of the worldwide semiconductor market.

Zero cost, zero power, zero delay, and zero volume are the four overlapping segments of the electronic systems market. Most applications fall within the zero-cost segment, which is by far the largest segment. Virtually all consumer applications fall within the zero-cost segment. Because consumer markets are price competitive, cost is always a concern for these products. The zero-power and zero-delay segments overlap substantially with the zero-cost segment. The zero-volume segment overlaps with the zero-delay segment, but is disjoint from the zero-cost segment.

### Electronic Systems Market Segments

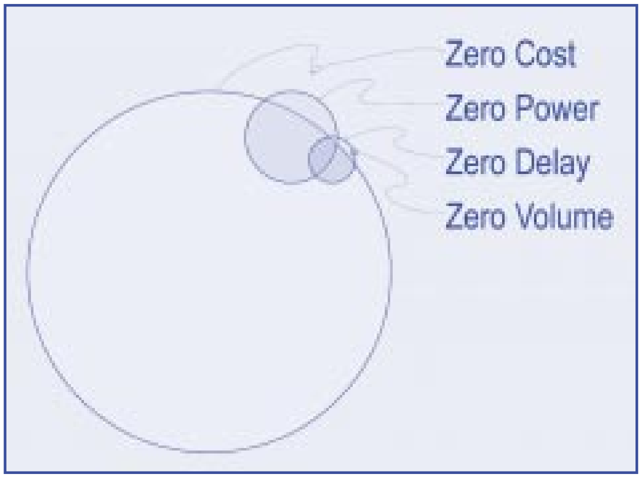


**Fig. 5.** Dominant design characteristics define electronic systems market segments.

*The zero-cost segment*, which to a first approximation represents all of the electronic systems market, is the segment for which low-cost is the overriding consideration. Most microprocessors go into consumer appliances (microwave ovens, electric razors, blenders, radios, toasters, and washing machines) that have minimal processing needs. These are commodity markets: that means they sell in high volumes (millions of units to tens of millions of units). These markets are characterized by intense price competition, so substantial effort goes into reducing production cost. The ideal would be zero cost to implement.

*The zero-power segment*, which to a first approximation is a few percent of the electronic systems market, is the segment for which zero-power dissipation is the ideal. These applications are mostly consumer items such as smoke detectors, cellular phones, pagers, pacemakers, hearing aids, MP3 players, and pocket calculators. Consumers want them to run forever on a single button-size battery or on ambient light. As with all consumer applications, minimum product cost remains a concern.

*The zero-delay segment*, which to a first approximation is little more than zero percent of the electronic systems market, is the segment for which zero delay from data in to result out is the ideal. These applications are also mostly consumer items such as personal computers, printers, scanners, copiers, and fax machines for which processing power and throughput are important—at minimum product cost, of course.

*The zero-volume segment*, which to more than a first approximation, is zero percent of the electronic systems market, is the segment for which the application potential is nearly zero. If the sales potential is close to zero, then production units and profits will be close to zero. Public relations and a leading-edge image motivate support for the zero-volume segment.

Before IBM introduced the Personal Computer in 1981, microprocessor design focused on zero cost. I chose the PC's introduction as a milestone because it changed the microprocessor's design focus to performance. The desktop computer segment, illustrated in fig. 6, is the overlap between the zero-cost segment and the zero-delay segment. The PC is a consumer item, so it belongs to the zero-cost segment (where the design goal says "if the cost is zero the price is all profit"). Marketing for the PC is based on performance, so it also belongs to the zero-delay segment. Since desktop computers are plugged into wall sockets, power isn't a primary design constraint, so the desktop computer segment doesn't overlap the zero-power segment. Before the PC's introduction, microprocessor design focused on the zero-cost segment; afterward design focused on performance. Intel, Motorola, Advanced Micro Devices, VIA Technologies, Inc., Transmeta (TMTA), and IBM design microprocessors for the desktop segment.

Because workstations and servers are not consumer items, microprocessors for workstations and servers are in the zero-delay segment outside the zero-cost segment. Because they are plugged into the wall, zero power is not a design goal, so workstations and servers are outside the zero-power segment. In fig. 7 this performance-oriented segment lies in the zero-delay segment and outside both the zero-cost and zero-power segments. Hewlett-Packard, Sun, Compaq, IBM, and Intel design microprocessors for the performance segment.

The battles between RISC (reduced instruction set computer) and CISC (complex instruction set computer) microprocessor designs were fought in the zero-delay segment. RISC microprocessors were in the performance segment (fig. 7) and attempted to invade the desktop computer segment (fig. 6), which was held by CISC microprocessors. RISC advocates assumed that if they designed for performance, volume would follow. CISC designers had to support the desktop computer segment,

## The Desktop Computer Segment

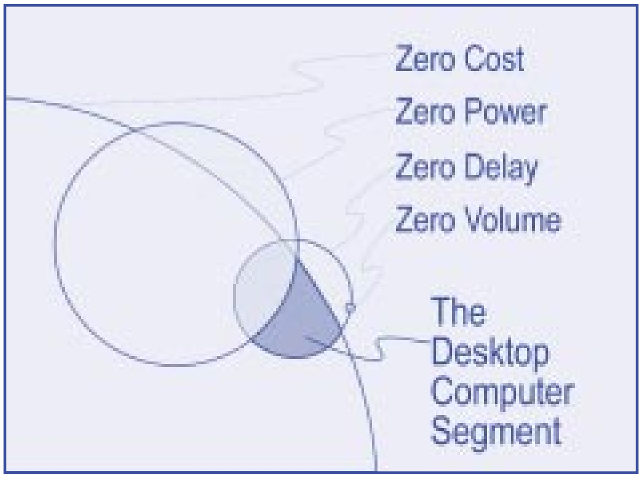


**Fig. 6.** The desktop computer segment includes the personal computer.

## The Performance Segment

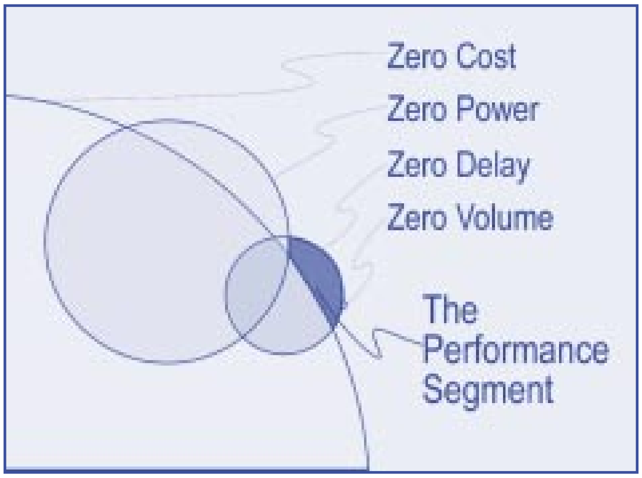


**Fig. 7.** The performance segment includes workstations and servers.

which meant low cost and high volume. Back to Moore's law: design for volume and performance will follow. The strategy of designing for high volume and riding Moore's law to performance beat the strategy of designing for performance alone. Workstations attempted to invade the desktop computer segment from their performance niche. Instead, the PC spilled over from the desktop computer segment into the performance segment and forced a retreat by workstation manufacturers.

We are indebted to the RISC advocates. The evangelism, competitive effort, and money that they spent on RISC drove Intel to designs with higher performance than necessary. The PC, using Intel's x86-compatible microprocessors, has been the beneficiary. In a final irony, however, Intel, whose volume strategy for CISC so thoroughly defeated the RISCs, got caught in the RISC fad and is now busily working with HP on its own RISC derivative called EPIC (explicitly parallel instruction computing).

PC manufacturers will ship about 150 million units worldwide in 2001. By contrast, Sun, the dominant company in the workstation market, will ship fewer workstations in a year than Dell ships in a week. In manufacturing volume, the Holy Grail of the semiconductor business, Sun will take all year to build as many microprocessors as Intel builds in a single six-hour period.

The desktop computer segment and the performance segment are here to stay and will continue to grow. Companies such as Sun and Intel that dominate these segments should do well as they build tethered computers for the global information grid. Improvements in desktop computers mean continued retreat by workstation and server manufacturers, but even the performance segment will grow in absolute terms. But these are traditional and not emerging segments. We have to look elsewhere for disruptive trends.

The Palm Pilot signaled the emergence of compute-intensive untethered devices. Cellular phones and GPS (global positioning system) receivers also fit this category. The personal digital assistant (PDA) is a consumer device (zero-cost segment), it is portable (zero-power segment), and it has high computational requirements (zero-delay segment). I call the overlap among these three segments the leading-edge wedge, as illustrated in fig. 8.

The world is splitting into tethered and untethered devices. Tethered devices provide computing, access ports, data transport, and storage for the global information grid. Designing microprocessor-based systems for the tethered world is now business as usual. Portable devices for the leading-edge wedge will be the collectors and consumers of data. These devices are a particular challenge because they combine requirements for low power and low cost with computationally intensive algorithms.

The major contributor to the microprocessor's improved performance has been increasing clock frequency. Generally, performance increases directly with megahertz. Doubling clock frequency doubles performance. The dilemma for the microprocessor is that while its performance increases directly with clock frequency, so does its power dissipation. As the microprocessor gets faster and more capable it consumes more power. The power dissipated by a microprocessor is proportional to the product of the capacitance, the frequency, and the square of the power supply voltage. In the recent past, lowering the supply voltage has been saving the microprocessor. Since the voltage term is squared, lowering the voltage by half permitted running the clock at four times the frequency for the same power dissipation. As a performance strategy, this has two problems. First, lowering the voltage also lowers the maximum frequency the IC can attain. Second, voltage can be lowered only so far before circuits quit working. Power supply voltage for the microprocessor has fallen from five volts to three-quarters of a volt and cannot go much lower.

## Dynamic logic

The microprocessor's dilemma is an opening for dynamic logic. The microprocessor's circuits don't change. The microprocessor runs programs on a fixed set of resources. Dynamic logic circuits, based on PLDs,

### The Leading-Edge Wedge

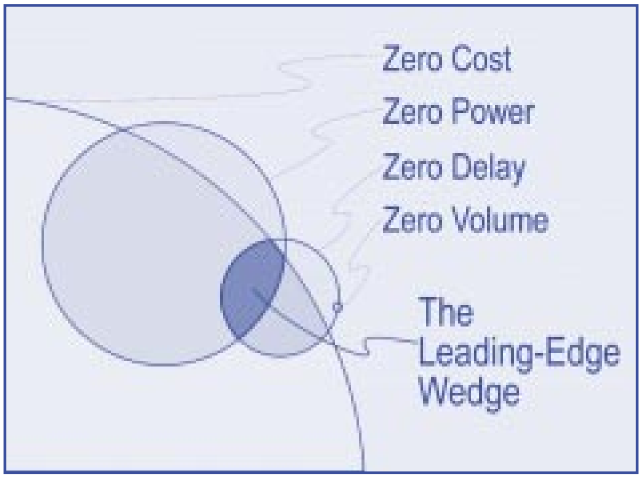


**Fig. 8.** The leading-edge wedge is the overlap among the zero-cost, zero-delay, and zero-power segments. These are portable, compute-intensive devices for the consumer market.

## The Elements of Dynamic Silicon

**MEMS**

**Microelectromechanical systems (MEMS) are tiny motors, gears, moving mirrors, turbines, and other complicated mechanical structures. MEMS are integrated circuits incorporating electronics and moving or deformable parts and are built with semiconductor fabrication methods.**

**Dynamic Logic**

**Dynamic logic creates hardware functions on the chip as needed. Driven by processing demand, the resident logic circuits vary with time. (By contrast, microprocessors run programs on fixed logic circuits: adders, shifters, registers, etc.)**

change over time. PLDs do for logic what memory does for the microprocessor. The microprocessor's memory holds the programs that personalize the microprocessor-based application. The PLD's memory holds the bit patterns that personalize the PLD-based application (it also happens to contain all the logic as well). Dynamic logic implements functions directly and "pages" them as needed into its available resources. Microprocessor solutions translate the algorithm's equations into the microprocessor's instruction set. Dynamic logic implements the algorithm's equations directly (in the PLD's gates). The direct implementation is more efficient.

The microprocessor and the DSP tout speed as a good thing. Intel and AMD battle for bragging rights on who's delivering microprocessors with the highest clock frequency. When the 1,200-MHz microprocessor is available, it's difficult to sell last year's 600-MHz component. The microprocessor needs speed to make up for the inefficiency of its programming languages, translations, instructions, state sequencer, operating systems, and fixed resources.

Folklore says PLDs are slow, but in portable devices, that can be an advantage. If the PLD delivers the same result in the same time, but runs at 1 MHz instead of the microprocessor's 1,000 MHz, it will use substantially less power than the microprocessor. If it's slower, how can it get the answer at the same time? The dynamic logic implementation has two advantages over the microprocessor: efficiency and parallelism. Direct implementation of a function (implementing the transformation equations as circuits instead of programming them) is more efficient. Since functions are "paged" into the PLDs logic resources, it's possible to implement parallel structures to the limit of available resources. A microprocessor may have one to two 32- or 64-bit shifters and comparators. If it was important in solving the problem, the engineer could create fifty or a hundred 137-bit shifters and comparators in a single PLD.

If dynamic logic is so great, why hasn't it already happened? Dynamic logic had to wait for some semiconductor milestones (invention and gate capacity), but even then it has been delayed by the enormous popularity of microprocessor-based solutions. I see four barriers to proliferation of dynamic logic implementations.

- It's a new idea.
- Microprocessor-based design has been successful for decades and is entrenched in the engineering community.
- University education is teaching engineers microprocessor-based design and it is not teaching dynamic logic design.
- Huge commercial enterprises are betting their future on microprocessors and on microprocessor-based design and will not be willingly displaced by dynamic logic upstarts.

The microprocessor and the DSP are not efficient enough for portable devices of the future, however, so there will be a transition to dynamic logic. It'll happen with a single large-scale application. Once that application proves the concept, the flood gates will open and the industry will turn at the rate its engineering community can learn new design methods (about a year) and can provide the necessary development tools (longer, perhaps three years).

Eventually, dynamic logic will invade everything that today hosts a microprocessor or DSP because, when you can afford it, efficiency is important. Dynamic logic has advantages in thermal management, in battery life, and in performance.

### MEMS: Powerful Machines

As the world splits into tethered and untethered devices it opens the door for dynamic logic and for microelectromechanical systems (MEMS). Portable devices become the collectors and consumers of data.

These collectors and consumers of data will interact with the real world. To interact with the real world, portable devices will need sensors and actuators. They will need moving parts. They will have to hear and see and they will have to sense motion. To fit in portable consumer devices, the sensors and actuators will have to be small, cheap, sensitive, and rugged.

MEMS are amazing little machines, but for our purposes what is most interesting about them is that once again they solve problems in hardware for which a microprocessor plus software, or in some cases even a hardwired ASIC, would be too slow or couldn't do at all (since microprocessors and ASICs don't have moving parts).

MEMS are built using semiconductor fabrication methods, but they do not require the expensive fine-line process equipment required for commercially-competitive semiconductors. Today's leading-edge semiconductor fabs for CMOS ICs use 0.18-micron line widths and will soon move to 0.13 microns. By contrast, line widths for MEMS ICs need be only in the range of 0.5 to 1.0 microns – Stone Age tolerances by comparison. MEMS use equipment and processes whose development and accumulated expertise have already been paid for. Since MEMS fabrication rides the coattails of the semiconductor industry, it may achieve a growth rate that exceeds that of the semiconductor industry itself.

As a particular advantage over other emerging areas, MEMS advance the state of something we already know how to do. We know how to build motors, switches, inductors, and springs and we know how to build semiconductors. Contrast this state with, for example, the similarly advancing areas of biotechnology and nanotechnology (building systems with moving parts on an atomic scale). Developments in biotechnology and nanotechnology advance things we aren't already familiar with.

Semiconductors have been improving for fifty years; the integrated circuit has been improving for forty years. The industry has made astounding progress in the face of significant limitations. For example, semiconductors did not have moving parts and they were generally fabricated using only resistors, capacitors, and active elements (transistors and diodes). There has been no convenient way to fabricate a coil (inductor), a moving-plate capacitor, or a relay, for example. Any inductor required by the circuit had to be supplied as a discrete element attached to the integrated circuit. MEMS fabrication can build coils, moving-plate capacitors, transformers, electromechanical switches, and relays.

MEMS are attractive for portable devices partly because they are small, sensitive, reliable, and cheap. Batch fabrication, which produces hundreds or thousands of chips per wafer, reduces the cost of MEMS. Chemical, physical, and optical analyzers should cost only a few dollars soon. Electronic "noses and tongues" will be small and cheap and will completely change some businesses. The electronic lab-on-a-chip, from companies such as Aclara, Affymetrix, Caliper, Cepheid, and Genefluidics, will change chemical collection and processing businesses. The labs on a chip may be more reliable than current methods because the chips can have arrays of sensors that sample in parallel and correlate the results. In biomedical analysis applications, MEMS devices work with nanoliters (a millionth of a milliliter) instead of milliliters (thousandths of a liter). If someone wants my blood, sample size is important. Also, small samples are an advantage if thermal cycling is required.

**Since MEMS fabrication rides the coattails of the semiconductor industry, it may achieve a growth rate that exceeds that of the semiconductor industry itself.**

One particularly interesting application for MEMS devices is pseudo-batteries that burn hydrocarbon fuels. Alan Epstein at MIT is designing a microturbine, and Adam Cohen at USC/ISI (University of Southern California/Information Sciences Institute) is designing a small burner. Either of these devices could be fitted into the form factor of a conventional battery (most of the space would be given to the tank for the fuel). Hydrocarbon fuels have about a hundred times the energy density of the best of today's conventional batteries.

MEMS are essential to our vision. They make possible the intelligent mobile assistants of the future. MEMS are the eyes, ears (both for sound and for relative motion), noses, and tongues. They will measure pressure, temperature, motion, acceleration, humidity, and flow rates. They will be both the sensors and the actuators for the collectors and consumers of data.

## MEMS: Challenge and Promise

Microelectromechanical systems are hard; there are plenty of challenges to commercialization. The fabrication is similar to standard silicon semiconductor fabrication, but it isn't identical. The MEMS industry is in an experimental stage and hasn't settled on a single fabrication standard, just as the integrated circuit industry experimented before silicon CMOS emerged as the dominant standard. It may be difficult for MEMS to ever achieve a single dominant standard because the enormous range of application requirements may prevent standardization.

Fabrication, assembly, and packaging can be difficult. MEMS may have moving parts that have to be built in three dimensions using what is an essentially two-dimensional process. Each part that will move has to be released from its original position. MEMS devices can consist of numerous layers that must be assembled. Further, packaging can be a challenge, particularly for devices that sense reactive chemicals and gasses sampled from the world outside.

The Defense Advanced Research Projects Agency (DARPA) sponsors an interesting range of MEMS research projects (see http://www.darpa.mil/mto/ctareas.html). The challenge with DARPA sponsorship is in achieving continuity from the basic research, applied research, and advanced development to commercialization. Once DARPA has sponsored projects through advanced development it expects the military or other DoD organizations to move the results into fielded systems. Unfortunately, military and DoD budgets and priorities are such that the transitions mostly don't occur. Still, DARPA-sponsored research is a force to consider. During the time its name was ARPA, it sponsored the ARPANET, which became the Internet.

As was the case with dynamic logic, these challenges can be overcome in single instances where the case is compelling for the advantages conveyed by the MEMS device. One place this will occur is the cellular phone. The front end of the cellular phone contains too many discrete components. It has been impossible to displace these components because they are so cheap that a MEMS integration of them cannot compete. But MEMS will offer advantages that will give them a foothold. As volumes rise, costs fall, fostering larger markets.

## Halfway House: Transmeta

I have the world split into tethered and untethered devices. For the tethered devices, it's business as usual. Untethered devices will be the collectors and the consumers of data; they will need MEMS and dynamic logic. That's the picture I painted; but it's not today's truth. Microprocessors and DSPs are the heart of today's untethered devices. We have to get from today's microprocessor- and DSP-based devices to tomorrows dynamic logic implementations. In the transition there's room for something more efficient than today's microprocessor.

Transmeta (TMTA) isn't doing a dynamic logic implementation, but it is doing something that may provide a bridge to dynamic logic design methods. To explain how Transmeta is different (forget what its PR machine says about "code morphing" and "long-run technology") we'll start with an illustration from the automotive industry.

The automobile has been with us for over a hundred years. The typical automotive subsystem has used *open-loop* control (you set it, then it does what it does). The brakes and suspension were designed to balance comfort and performance across the range of users and driving conditions. Valve and spark timing, fuel flow, and suspension damping were set mechanically and were crudely controlled. Only recently have automobiles and their engines been outfitted with sensors and *feedback* mechanisms that allow them to *adapt* to driving conditions: accommodating changes in temperature, altitude, road conditions, driving habits, and fuel quality. *Closed-loop* control systems (you set it, you see what it does, you reset it, etc.) improve the performance and efficiency of the automobile.

With open-loop systems, you have to know what the system is doing; with closed-loop systems, you do not. Suppose, for example, you want to improve the performance of your automobile's open-loop suspension system. You have to learn what that means for your specific automobile and how you drive it and then you have to adjust or replace parts of the system. In a closed-

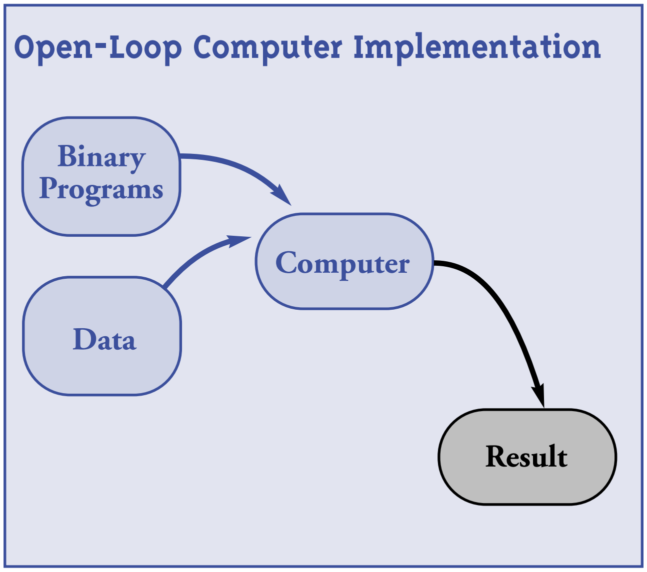


**Fig. 9:** An open-loop computer implementation balances ease of use and performance across applications.

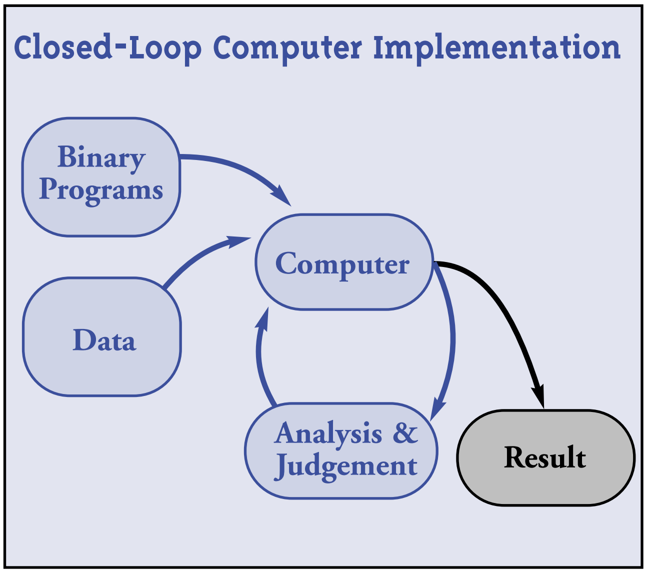


**Fig. 10.** A closed-loop computer implementation adapts to the problem.

loop suspension system, the system will measure the performance of its own components and it will measure the environment and your driving. It will optimize performance based on *its* measurements.

Computer architecture has been running open loop for more than fifty years. We design the instruction set, build the machines, and sell them to our customers. Once the machine is in the field, it does what it does and it never changes. We've built computers to balance comfort and performance across the range of applications, as shown in fig. 9 on page 10.

A closed-loop computer implementation analyzes feedback from sources within the system to adjust the system as it is running. In the Transmeta microprocessors, the system can adjust the clock frequency and voltage so that the microprocessor runs just fast enough to meet the computing requirements of the application. Running just fast enough greatly reduces power consumption relative to a microprocessor that runs at full speed and then idles for the balance of the time.

Closed-loop computer implementations, shown in fig. 10, can change the world because the computer adapts to the conditions of the problem. The computer budgets its performance and power to match the requirements of the problem. As techniques for implementing closed-loop computer systems improve, they will extend the battery life of mobile systems. Instead of designing computer instruction sets to match the application problems, "computer architects" will have to design computers that adapt well to their problems. Transmeta's closed-loop microprocessor is about half way between an open-loop microprocessor-based implementation and a dynamic logic implementation.

Transmeta will succeed in portable devices where x86 compatibility is important (many browser plug-ins are x86-specific) because its closed-loop microprocessor implementation is more efficient than the open-loop implementations of its competitors. Perhaps it will provide a pseudo-proof of concept for dynamic logic. Transmeta is better than DSPs and other microprocessors. We promoted it to our list of Dynamic Silicon Companies for its breakthrough as a closed-loop computer implementation.

Nick Tredennick and Brion Shimamoto
January 2001

# Dynamic Silicon Companies

**The world will split into the tethered fibersphere (computing, access ports, data transport, and storage) and the mobile devices that collect and consume data. Dynamic logic and MEMS will emerge as important application enablers to mobile devices and to devices plugged into the power grid. We add to this list those companies whose products best position them for growth in the environment of our projections. We do not consider the financial position of the company in the market. Since dynamic logic and MEMS are just emerging, several companies on this list may be startups. We will have much to say about these companies in future issues.**

**Altera and Xilinx** (ALTR http://www.altera.com) (XLNX http://www.xilinx.com)
Altera and Xilinx together dominate the programmable logic business, with almost seventy percent of the CMOS PLD market. Both companies are aggressive and competitive. Sixty-six percent of Altera's revenue comes from the rapidly growing communications segment (Telecosm companies) and an additional sixteen percent comes from the electronic data processing (EDP) segment. Altera and Xilinx are positioned to be major suppliers in tethered applications such as the base stations that support mobile devices.

**Analog Devices** (ADI http://www.analog.com)
Analog Devices is a leader in analog electronics for wireless RF and communication, MEMS for automotive applications (accelerometers, pressure sensors, transducers), and in DSPs.

**ARC Cores** (ARK (London) http://www.arccores.com)
ARC Cores makes configurable processor cores. Configurable processors allow the application engineer to adapt the processor's instruction set to the requirements of the problem. Conventional microprocessors have fixed instruction sets.

**Calient** (* http://www.calient.net)
Calient is a pre-IPO startup that builds photonic switches for the all-optical network core. It builds its own MEMS components. Calient has expertise in MEMS components in Ithica, NY through its acquisition of Kionix and through its own experts in Santa Barbara and San Jose, CA.

**Celoxica** (pre-IPO, www.celoxica.com) Celoxica supplies the DK1 development suite that maps program-level hardware descriptions to SRAM PLDs. Celoxica also offers design services and plans to become a supplier of soft-core IP.

**Cypress** (CY http://www.cypress.com)
Cypress Microsystems builds components for dynamic logic applications. Cypress also builds MEMS and is a foundry for MEMS.

**QuickSilver Technology, Inc.** (* http://www.qstech.com)
QuickSilver has the potential to dominate the world of dynamic logic for mobile devices (untethered). While many companies work on programmable logic and on "reconfigurable computing" for tethered applications, QuickSilver builds adaptive silicon for low power mobile devices.

**SiRF** (* http://www.SiRF.com)
SiRF builds RF GPS chips for the mobile market. It is a world leader in development of integrated GPS receivers.

**Tensilica** (pre-IPO, www.tensilica.com)
Tensilica provides a design environment and licensing for configurable soft-core processors.

**Transmeta** (TMTA http://www.transmeta.com)
Transmeta makes new generation microprocessors that use closed-loop control to adapt to problem conditions in an x86-compatible environment. This enables Transmeta's microprocessors to save power over conventional microprocessors from AMD and Intel. The base instruction set is not available to the application engineer.

**Triscend** (* http://www.triscend.com)
Triscend builds microcontrollers with configurable peripheral functions and with configurable inputs and outputs. Triscend helps consolidate the microcontroller market into high-volume, standard chips.

**TSMC** (TSM, www.tsmc.com) Taiwan Semiconductor Manufacturing Corp. TSMC is a leading independent CMOS semiconductor foundry and the principal supplier of chips to Altera.

**UMC** (UMC, www.umc.com) United Microelectronics Corp. UMC is a leading independent CMOS semiconductor foundry and the principal supplier of chips to Xilinx.



291A Main Street
Great Barrington, MA 01230
www.dynamicsilicon.com