Written by
Nick Tredennick

DynamicSilicon

Published by
Gilder Publishing, LLC

The Investor's Guide to Breakthrough Micro Devices

The PC Isn't the Microprocessor Market—Yet

Personal computer (PC) sales will decline in 2001. Shipments of the x86 microprocessors used in Windows PCs will therefore decline. The rest of the microprocessor market—embedded applications—is growing. Because of changes occurring in the microprocessor world, the x86 will come to dominate embedded applications. The surprise is that the x86 probably won't be Intel.

The invisible microprocessors

Mention the word microprocessor and most people think of Intel and of the PC. Indeed, the PC is the most visible use of microprocessors. Its microprocessor gets the most press. Manufacturers ship about 150 million PCs each year. Intel has become one of the world's most profitable corporations on the strength of its microprocessor shipments. But, when its unit sales are compared to the *number* of microprocessors shipped, Intel probably wouldn't make the top ten. If Intel was a car maker, its rank might be closer to Daewoo. So who are the GM and Ford of microprocessors? And why is Intel/Daewoo—and not the GM or Ford of microprocessors—the most profitable?

Intel charges more. Intel prices its Xeon processors, used in large servers, above $2000. Two-gigahertz Pentium 4s are $562. The Celeron, at 900 MHz, is $64. The average selling price (ASP) for an Intel microprocessor is about $200. Selling more than 100 million PC-bound microprocessors a year generates most of Intel's better than $30 billion annual revenue. This $30 billion, which represents less than two percent of microprocessor unit volumes, accounts for roughly half of the income generated by all types of microprocessors.

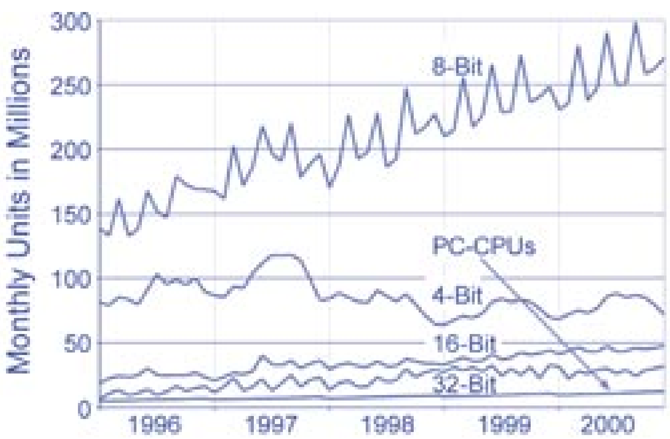

Fig. 1. Worldwide shipments of all types of microprocessors dwarf shipments of PC microprocessors. (Data courtesy of Jim Turley and the World Semiconductor Trade Statistics organization).

In contrast, microprocessors for the embedded market (washer/dryers, blenders, microwaves, hair dryers, clocks, anti-lock brakes, vending machines, etc.) may be had for a few dollars. How can this be? Are the differences sufficient to enable some microprocessors to command thousands of times the price of others?

It's as if some cars cost thousands of dollars while others cost millions. There *are* cars that cost more than a million dollars, but they are hand-assembled, while their cheap cousins are mass produced. All microprocessors are mass produced, so that's not the difference.

Intel can charge more for its microprocessors because the operating systems, web browsers, applications, and device drivers for Microsoft Windows-based personal computers depend on Intel's "x86" microprocessor instruction set for compatibility. Among its competitors, only Intel can afford enough of the high-end chip fabrication capacity needed to support the demand for x86 microprocessors. Intel's market position gives it the volumes it needs to justify the most expensive semiconductor processes. These processes keep its x86 chips on the leading edge in performance and in chip size (high yield), but this requires enormous expense in capital equipment. Intel's primary x86 competitor is Advanced Micro Devices (AMD); its secondary competitors are Transmeta (TMTA) and VIA Technologies (TSE2388).

If the most common computer, the PC, is x86-based, doesn't it follow that x86 microprocessors might also dominate embedded applications? No. It's a matter of margins and of how the industry evolved. The paramount consideration in embedded applications is absolute cost (zero cost is the ideal for consumer products because even a low price is pure profit). Embedded microprocessors *had* to be the lowest-cost way—they were trying to replace some already-cheap way of doing things. Most embedded microprocessors have four- or eight-bit data paths and cost a few dollars (some are less than a dollar). These microprocessors are small chips and therefore do not require enormous capital expense to make. Intel was selling all the processors it could make for a hundred times as much. So Intel had little interest in using any of its x86-production capacity for what it saw as merely low-margin versions of the x86. The embedded market developed in the mid to late '70s and the '80s. The x86 wasn't efficient enough or cheap enough, so the embedded business went to Intel's (8-bit) 8051 microcontroller and its derivatives, to Motorola's (8-bit) 6805 and its derivatives, and to similar low-end microprocessors from a host of other manufacturers.

Intel is a niche player

At this month's Gilder-Forbes Telecosm Conference, Jim Turley, in a presentation titled "Intel has (approximately) 0% market share," broke down microprocessor shipments. Microprocessor shipments through June 2001 totaled four billion units worldwide. The 32-bit CPUs used in PCs make up a small fraction of that. *Each week*, manufacturers ship as many microprocessors as the PC's CPU makers ship in a whole year!

The x86 *is* in some embedded applications. National Semiconductor, ST Microelectronics, Intel, AMD, and Transmeta all build x86 microprocessors for embedded applications. But today's x86 uses constitute a tiny segment of the embedded market.

Embedded applications: the microcontrollers

Billions of microprocessors ship every year—invisibly embedded in applications from toothbrushes to transmissions. The microprocessor and a few standard peripheral chips displaced a much larger number of chips. The range of applications for the microprocessor and its few peripheral chips increased their production volume, which reduced their cost. This meant fewer part numbers, with broader uses. Decreasing the cost of electronic components increased their range of application—a virtuous cycle.

As new microprocessors improve in performance and in capability, they invade more demanding applications. As old microprocessors get smaller and cheaper, they invade more low-end applications. Moore's law says that the number of transistors on a chip doubles every eighteen months. Those transistors can be used to increase the performance and capability of the microprocessor or they can be used to integrate the microprocessor's peripheral functions. Manufacturers of PC CPUs have been using the additional transistors to augment the microprocessor's capability and to improve its performance. Other manufacturers, driven by cost-dominated consumer applications, reduce cost by using the additional transistors to integrate the microprocessor's peripheral functions—putting the microprocessor and its peripheral functions onto the same chip.

This combination of a microprocessor and its peripherals on one chip is called a *microcontroller*. The latest microcontrollers have the fancier designation "system on a chip" (SoC).

Microcontrollers, IP, and Triscend

The microprocessor is a physical block of logic on the microcontroller chip. Each peripheral function is a block of logic on the chip. Each of these blocks is called a "core." *Circuit* blocks designed for a particular semiconductor process are called "hard" cores; blocks designed to be portable so they can be implemented in any semiconductor process are called "soft" cores. Cores exist as intellectu-

DynamicSilicon

Editors	**Nick Tredennick**
	Brion Shimamoto
Publisher	**Lauryn Franzoni**
Web Editor	**Jorin Hawley**
Designer	**Julie Ward**
Subscription Services	**Melissa McNally**
Editorial Director	**Richard Vigilante**
President	**Mark T. Ziebarth**
Chairman	**George Gilder**

Dynamic Silicon 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.

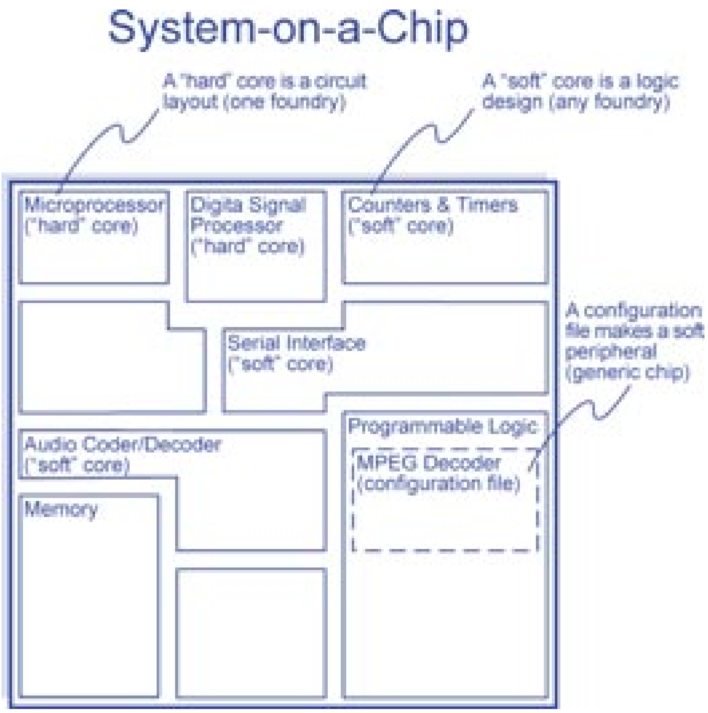

Fig. 2. A system-on-a-chip (SoC) is a collection of cores. Use of one or more "hard" cores ties the chip to a particular semiconductor process (foundry).

al property (IP) descriptions: modular entries in a chip design database. IP descriptions are designs packaged for reuse. Hard core IP is a *circuit layout* in a particular process. Hard core IP becomes a circuit block on a physical chip. Soft core IP is a *logic design*. It's an IP description that is mapped into a particular semiconductor process before it becomes a circuit block on a physical chip.

Different embedded applications want different peripheral functions, different physical packages, different microprocessors, etc. The microprocessor and peripheral functions for an MP-3 player won't be ideal for automotive engine control, so specialized microcontrollers serve each market. The number of embedded applications caused microcontroller varieties to skyrocket. This explosion of microcontroller types fragments production volumes and works against reducing cost in a cost-sensitive market. Triscend recognizes that the microcontroller proliferation converts the chip market into an IP market. Instead of building a huge variety of custom microcontrollers and fragmenting its chip manufacturing, Triscend builds a few standard components that include programmable logic. The user customizes the chip for an embedded application by putting the peripheral functions into the chip's programmable logic. Triscend is consolidating the microcontroller market.

Just a moment! Microcontrollers serve the zero-cost market. That market's as cost-sensitive as it gets—and I'm telling you that putting programmable logic on a microcontroller is a good strategy? Everyone knows that programmable logic is slow and that it has more overhead than the U.S. government. How can that move be competitive? The microcontroller market has been fragmenting into designs for ever more specific applications. Fragmenting manufacturing increases the cost of chip production. Triscend consolidates manufacturing by building identical microcontroller chips that are personalized, with programmable logic, *after they come off the production line*. That doesn't address the "big and slow" problems of programmable logic, but Moore's law does.

Moore's law again

Here's how. Consumers buy millions of coffee makers, toasters, clock radios, irons, waffle makers, camcorders, DVD players, and electronic games every year. And the performance demands of these embedded microprocessor applications don't change rapidly, so they use the same microcontrollers year after year. Normally, if Moore's law isn't improving the microcontroller's performance, it would be shrinking the chip's size, and, therefore, its cost. But Moore's law has already shrunk most of these chips as far as they can go—they are "pad limited." The microcontroller cost bottleneck is no longer in the size of the chip. The bottleneck is that there are too many varieties to manufacture. For most applications, the microcontroller doesn't need more transistors and it doesn't need more performance. Triscend reengages Moore's law, investing new transistors in flexibility (in the form of programmable logic) to consolidate chip manufacturing. The hard cores of the original microcontroller's peripherals become configuration files for the Triscend chip's programmable logic. This is splitting the market for custom microcontrollers into a market for generic microcontrollers and a market for IP configuration files representing the microcontroller's peripherals.

The value of the microcontroller has *been* in its silicon. Moore's law advances are moving value *from the silicon to the IP files* for microprocessor and peripheral cores. These IP files can be hard cores, soft cores, or configuration files for programmable logic.

DSPs join the act

The microprocessor is a set of *resources* and a *state sequencer*. The resources include registers, arithmetic units, bit manipulators, and comparators. The state sequencer decides, overall, what to do next. It sequences each instruction through fetch (getting the instruction from memory), decode (finding out what the instruction's bit pattern means), and execute (doing the instruction's tasks). The microprocessor's state sequencer is vital to the embedded system. If the embedded application was built without the microprocessor, the engineer would have to design a state sequencer for the particular system's logic. For applications that require intensive computation, a digital signal processor (DSP) may be more efficient than a microprocessor. A microprocessor is good for executing a lot of instructions. In contrast, a DSP is good for operating on a lot of data with relatively few, calculation-intensive, instructions. DSPs are "data-flow" engines; that is, they are typically built to flow two streams of input operands through the arithmetic unit into a result stream. DSPs can perform a huge number of arithmetic operations very fast. Microprocessors are good for choosing among alternatives; DSPs are good for crunching through lots of data.

Texas Instruments (TXN) is the leader in high-performance DSPs. Motorola (MOT), Agere (AGR/A, a spinout of Lucent), Analog Devices (ADI), and a host of other companies also manufacture DSP chips. But, in the same way that Moore's law is turning the microprocessor chip business into an IP business, it will turn the DSP chip business into an IP business. Just as peripherals migrated onto the same chip with the microprocessor, the DSP will want to be on the same chip with the microprocessor and peripherals. The DSP won't displace the microprocessor core on the chip because the strength of the microprocessor is in its decision-making and the strength of the DSP is in calculation. Whether TI remains the leader in DSPs will depend on its ability to transform its chip business into an *IP business* as it moves from *selling chips* to *licensing* DSP *cores*.

Tethered devices, untethered devices, and IP

The world is dividing into tethered and untethered devices. Tethered devices are the computers, access ports, data transport networks, and storage systems that get power from a wall socket and constitute the global information grid. Untethered devices do not plug into wall sockets. Attributes of untethered devices include: (wireless) communication, power conservation, sensors, actuators, performance, versatility, and adaptability. Wireless communication connects the untethered device to the global information grid (which can connect it to other untethered devices). Power conservation gives it the operational duration to compete in the zero-power segment of the market (where the system's design objective is minimum power dissipation). Sensors and actuators connect the untethered device to its environment (and will lead to the proliferation of microelectromechanical systems). Performance, versatility, and adaptability differentiate the device from its competitors and place it in the zero-delay segment (where the system's design objective is minimum time to service the user's request). Since the largest market for untethered devices is consumers, these devices are also in the zero-cost segment (where the system's design objective is lowest possible cost).

Integrating the untethered device's functions onto one chip drives standardization. The functions are becoming soft IP. Even small companies can create valuable IP. Standard IP functions, designed once, can be licensed throughout the industry.

The electronics industry is developing interfaces that let software and hardware functions connect in standard ways. This encourages the licensing of common hardware and software functions. Motorola, IBM, Hitachi, Intel, and TI are among the giants of the microprocessor and DSP business. These companies are integrated device manufacturers (IDMs). They develop, manufacture, and sell microprocessors and DSPs. They are developers *and* manufacturers. But the rise of standards and the continuing Moore's-law driven integration will *split* development and manufacturing. The business models of the IDMs *consider the value to be in the physical chip*. In the future, the physical chips will be *generic*. They will assume their value when they have been personalized with IP files. *The value will be in the IP*. It will be a tough transition for the IDMs. Separating developers from manufacturers brings more developers into play. Functions, operating systems, application programming interfaces, and application software all become commodities, which lowers cost and fosters proliferation.

ARM goes soft

ARM Limited (ARMHY) focuses on untethered devices. ARM is an IP company that, until recently, concentrated on hard-core implementations of its ARM microprocessor. In the future, ARM's cores will be soft. ARM is moving from supplying microprocessor cores to supplying SoC solutions. ARM offers its microprocessor core with three categories of peripherals, as part of its ARM PrimeXsys Wireless Platform. The first category is functions required to boot and run an operating system. It includes several timers, a real-time clock, general-purpose input/output functions, and power management functions.

The second category of ARM peripherals is functions that are based on established standards and that will be required for any 2.5/3G (two-and-a-half or third generation) wireless phone or PDA (personal digital assistant).

The third category is bus peripherals, like memory and display controllers, that evolve rapidly.

ARM's change in strategy from supplier of hard-core microprocessors to supplier of soft cores and SoC solutions makes ARM a Dynamic Silicon company. Dynamic Silicon companies Altera and Triscend are ARM licensees.

What's going on: standardization and differentiation

The enormous range of embedded applications works for standardization in some areas and works against it in others. Standards are both bane and blessing. Standards restrict one's ability to solve problems directly, since each problem solution must work within constraints imposed by the standard. But standards enable work partitioning and solution sharing.

Application programming interfaces (APIs) ban programmers' direct communication among program modules. This may reduce the program's efficiency, but APIs enable the application builder to employ program modules written by other developers. For example, APIs enable a single set of programs, called device drivers, supplied with the operating system, to support a wide range of peripherals across all application programs. Without APIs, *application programs* would have to include device drivers for all the peripherals (e.g., printers, disk drives, and displays) that they might use.

Soft cores allow a small design house in Hoboken to contribute to the system design for a transmission designed in South Bend and built in Mexico. Moore's law enables the transistor capacity for SoC integration. Standard interfaces and soft cores (the softening of hardware) enable the division of labor that empowers engineering teams outside the IDMs to build subsystems for designs around the world. Standard description languages for the chip design databases make the designs portable—manufacturable by a number of semiconductor processes. All this contributes to the proliferation of standard designs and to reducing the cost of producing SoCs. But SoCs assembled wholly from standard pieces don't offer the differentiation necessary for manufacturers to distinguish their products.

ARC Cores (ARK) and Tensilica, both Dynamic Silicon companies, sell soft-core microprocessors with a difference. ARC and Tensilica allow the designer to customize the microprocessor's instruction set and its performance. The engineer adjusts the size and capability of the microprocessor's resources and its instruction set to the requirements of the application. Rather than entrusting the instruction set design to a priesthood of "computer architects," ARC and Tensilica provide a core instruction set (a functional minimum). The core instruction set makes the soft-core microprocessor useful in applications where all that is needed is a simple controller. These controllers are needed so frequently that many of the SoC designs that employ an ARC or Tensilica core employ more than one. For more demanding instances, the engineer adds instructions tailored to the particular embedded application. The result doesn't waste transistors, power, or efficiency on unused features—features that exist in fixed instruction-set microprocessors to make them appeal to a broad range of applications.

It sounds great to be able to hand the power to add instructions and hardware resources to the engineer, but it isn't easy. If you add instructions, the operating system and the compilers have to know about them to take advantage of them. Simulators and debuggers have to know about hardware extensions and about custom instructions. Development tools for customizable processors are the big challenge. ARC and Tensilica address these points and have a head start of several years on potential rivals.

Today, Tensilica's Xtensa design environment follows the engineer's direction to create a microprocessor core with custom instructions and with hardware tailored to the application by the development engineer. Xtensa turns out a soft-core microprocessor and the corresponding custom software tools (compiler, assembler, simulator, and debugger) and the operating system extensions that are necessary to take advantage of the custom instructions.

Tensilica: compiled hardware

In a presentation at the Microprocessor Forum in October 2001, Tensilica's CEO, Dr. Chris Rowen, described a generation of processor design tools that is in development. This software accepts programming language source code, sample data profiles, and chip size constraints as input. Rather than have the development engineer specify hardware extensions and custom instructions, this next-generation design software runs through thousands of possible solutions looking for an optimized solution. It spits out a custom soft-core microprocessor with optimized instructions, the necessary software tools, the operating system extensions, *and* the application code (binary). In Dr. Rowen's view, these "extensible processors" are the new transistors. Designers will employ from dozens to thousands of extensible processors on each chip. These custom processors are the building blocks for system-on-a-chip designs. These "extensible processors" won't be thought of as processors, however; they're just custom functions created by the application's requirements.

Dr. Rowen has got it right. I know it's right because Tensilica's processor generator raises the level of abstraction yet again. There won't be a need for computer architects because a program does the work. Programmers are being handed the power to create custom soft-core functions. Raising the level of abstraction raises designer productivity and it expands the pool of designers. It's always the right decision.

Intel and the x86 business

Intel dominates the market for x86 microprocessors. For our purposes, I'll divide the x86 market into three segments: workstations and servers, desktop computers, and notebook computers.

Historically, Intel's x86 strategy was a single family of chips spanning a range of performance. Intel periodically introduced a new microprocessor at the high end, pushing the microprocessor's family members down in price and in ranking and dropped the lowest-performance chip from the range. This strategy maintained a family of chips that grew in performance over time. The family also grew in price/performance over time, since Intel introduced each new higher-performance microprocessor at about the same $1000 price. The problem with this strategy was that Intel wanted to maintain high margins for high-performance microprocessors but also wanted to offer low enough prices to maintain its market share at the low end of the range. This eventually led to an *eight-to-one ratio in price supported by only a two-to-one ratio in performance* across the product family. That got to be a problem, because as microprocessor clock frequencies continued to rise, performance across the family compressed. How do you convince the customer to pay eight times as much for twice the performance?

A few years ago, Intel split its x86 chip offerings into separate product lines serving each market segment. Intel now offers Xeon microprocessors for workstations and servers, Pentiums for midrange desktops, and a separate line of Pentiums for notebook computers. Intel differentiated microprocessor families by options, such as the speed of the memory bus and the size of on-chip memory. That differentiated product families by more than performance and allowed Intel better control over its margins.

Intel and AMD duke it out in high-end microprocessors that power workstations and servers. Intel and AMD are also the main competitors in desktop computers. VIA Technologies competes in the "value" segment of the desktop computer market. Intel and Transmeta compete in notebook computers. Transmeta competes with Intel and with AMD for web and file servers in giant server farms, where space and electrical power are more important than raw performance (microprocessor performance isn't the bottleneck in serving web pages and files). RLX, for example, makes a Transmeta-based "blade" server that squeezes 24 servers into a 5.25-inch-high space of a 19-inch-wide rack. Because Transmeta's chip doesn't require its own cooling fan, RLX can pack 324 servers in an industry-standard rack that normally houses 42 servers. As if to validate Transmeta's strategy, Intel announced in March and unveiled at this November's Comdex, low-power chips for the server market. These chips are based on the notebook version of the Pentium III. Intel has also announced the 830 motherboard chipset that supports error-correcting memory, a popular feature in servers.

That's a snapshot of today's x86 market minus two important pieces. First, where is this market headed? Second, what is the x86's position in the enormous embedded market?

Whither PCs?

In the second quarter of 2001, the worldwide market for PCs declined from 30.4 million in 2Q2000 to 29.8 million in 2Q2001. Data from IDC (www.idc.com) shows that growth of the worldwide market for PCs has been slowing in recent years (and will be *negative* for 2001). The PC market is maturing. As the market matures, the performance needs of more users are satisfied by cheaper computers (*Dynamic Silicon*, Vol. 1, No. 3). In areas, such as the United States, where the PC has achieved high market penetration, sales will be dominated by purchasers replacing older PCs, so sales growth will be slow. In today's new high-growth markets, such as China and other rapidly developing regions, "value" desktops will dominate sales, not the high-end computers with high-margin microprocessors. It's good news for VIA Technologies, which builds the best microprocessors for value markets, and it's bad news for Intel and for AMD.

The point of this story is that the x86-compatible PC dominates the market. The Windows operating system, the x86 microprocessor, and the personal computer are the world's application development platform and, together with a web browser and its "plug-ins," its window to the World Wide Web.

I said that 150 million PCs is nothing compared to the number of microprocessors shipping in embedded applications every year. Isn't the PC inconsequential? No. The PC dominates Internet use and it is the platform used to develop most hardware and software.

Many of the web browser's plug-ins, such as the ones that allow sounds, animations, and videos to play on web pages, have been written specifically for x86-based computers. That's important because it ties convenient viewing of web pages to x86 microprocessors. The PC market has been driven to a single microprocessor standard. The efficiency of sharing subsystem components, device drivers, applications, operating systems, and web browsers drives standardization. The rise of the Internet and of the World Wide Web further solidify the x86's position as the standard. The x86 ties the PC to programs, data, and file formats in the network. No such equivalent exists either for the cell phone, which processes voice and has no data, or for appliances, which are just now connecting to the network. When cell phones and appliances connect to the Internet, the programs, data, and file formats already in the network will demand x86-compatibility.

Embedded x86

The PC market contrasts sharply with the market for embedded microprocessors. Since the introduction of the first commercial microprocessor in 1971, the state of embedded applications has been anarchy—any microprocessor or microcontroller might appear in any application. Three characteristics of the industry contributed to this situation. First, the enormous range of applications encouraged the proliferation of microprocessors and microcontrollers, to meet the special requirements of many market segments. Second, embedded applications tend to be self-sufficient. That is, they function autonomously; they may have to sense and act in the real world, but they generally don't interact with other digital systems. Third, thousands of small design shops build embedded systems; it doesn't require the resources of IBM or HP, with design teams of hundreds or thousands, to build embedded systems. Small teams select the microcontroller, write the software, and build the circuit boards to run the trash compactor, phone recorder, or talking picture frame. Thousands of independent design teams make one-time decisions for one-of-a-kind products. There's no reason for the brains of dishwashers from Maytag, KitchenAid, and General Electric to be the same microprocessor.

The proliferation of set-top boxes and of untethered devices such as cell phones and PDAs, gave manufacturers of other-than-x86 microprocessors high-volume opportunities. It looked to microprocessor manufacturers as if these applications were open to any instruction set. Set-top boxes might be based on a MIPS or on a PowerPC microprocessor, rather than on an x86 microprocessor. PDAs and cell phones based on ARM and on other microprocessors might dominate their market segments.

That's the way it looked for a few years. Wishful thinking. Along came the Internet and the World Wide Web. Your car may talk to your iron; your refrigerator may connect to the Internet. Soon the world's consumers will demand "always-on" connections to the Internet from untethered devices. This connection will drive the PC's standardization into the connected devices. Connecting devices to the Internet means connecting them to the PC. Browser plug-in compatibility, file system compatibility, application compatibility, and OS compatibility will enable and encourage x86 penetration in untethered devices. To paraphrase Transmeta's CTO Dave Ditzel: any other microprocessor is just a placeholder awaiting the x86's arrival.

The x86 will surge into embedded systems and particularly into the untethered devices that interact the most with PC applications, web pages, and databases. The x86 has the largest base of installed development systems and it has the largest base of programmers and developers.

Surprise ending

But, for power efficiency and design economy, untethered devices will require soft cores for their SoC designs. Intel and AMD are integrated device manufacturers; they have their own chip design teams, their own chip fabrication plants, and their own chip sales organizations. It will be difficult for IDMs to change themselves *from suppliers of chips to suppliers of soft-core x86 IP* because it requires a change in *culture*. Of the remaining x86 suppliers, only Transmeta and VIA Technologies have designs that are new enough to convert into useful soft cores.

This is irony at its best. Everyone thinks of the PC as being synonymous with microprocessors. It isn't. In unit volumes, PC microprocessors are an inconsequential part of the microprocessor market. But as the world's myriad devices connect to the Internet, the PC's microprocessor standard will surge into embedded devices as the microprocessor on an SoC. As it does so, the PC's microprocessor *will* begin to represent the market for microprocessors. The x86 in these embedded devices will be a soft core that is *unlikely to come from Intel or from AMD*, but it could come from Transmeta or from VIA Technologies.

Nick Tredennick and Brion Shimamoto
November 26, 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, some companies on this list are startups.

Company (Symbol)	Technology Leadership	Reference Date	Reference Price	10/31/01 Price	52-Week Range	Market Cap.
Altera (ALTR)	General Programmable Logic Devices (PLDs)	12/29/00	26.31	20.45	14.66 - 42.00	7.9B
Analog Devices (ADI)	RF Analog Devices, MEMS, DSPs	12/29/00	51.19	38.20	29.00 - 64.00	13.8B
ARC Cores (ARK**)	Configurable Microprocessors	12/29/00	£3.34	£0.30	£0.25 - 3.90	£108M
ARM Limited (ARMHY***)	Microprocessor and System-On-A-Chip Cores	11/26/01	16.59		8.39 - 29.63	5.7B
Calient (none*)	Photonic Switches	3/31/01				
Celoxica (none*)	DKI Development Suite	5/31/01				
Chartered Semiconductor (CHRT)	CMOS Semiconductor Foundry	7/31/01	26.55	19.56	16.06 - 60.06	2.7B
Coventor (none*)	MEMS IP and Development Systems	7/31/01				
Cypress (CY)	MEMS Foundry, Dynamic Logic	12/29/00	19.69	19.75	13.72 - 39.69	2.4B
QuickSilver Technology, Inc. (none*)	Dynamic Logic for Mobile Devices	12/29/00				
SiRF (none*)	Silicon for Wireless RF, GPS	12/29/00				
Taiwan Semiconductor (TSM†)	CMOS Semiconductor Foundry	5/31/01	19.86	13.01	8.39- 19.02	43.8B
Tensilica (none*)	Design Environment Licensing for Configurable Soft Core Processors	5/31/01				
Transmeta (TMTA)	Microprocessor Instruction Sets	12/29/00	23.50	1.77	1.17 - 50.88	237M
Triscend (none*)	Configurable Microcontrollers (Peripherals)	2/28/01				
United Microelectronics (UMC†)	CMOS Semiconductor Foundry	5/31/01	10.16	5.73	4.25 - 12.23	13.1B
Wind River Systems (WIND)	Embedded Operating Systems	7/31/01	14.32	14.53	9.71 - 47.63	1.1B
Xilinx (XLNX)	General Programmable Logic Devices (PLDs)	2/28/01	38.88	30.75	19.52 - 91.94	10.3B

† Also listed on the Taiwan Stock Exchange
* Pre-IPO startup companies.
** ARK is currently traded on the London Stock Exchange
*** ARM is traded on the London Stock Exchange (ARM) and on NASDAQ (ARMHY)

NOTE: This list of Dynamic Silicon companies is not a model portfolio. It is a list of technologies in the Dynamic Silicon paradigm and of companies that lead in their application. Companies appear on this list only for their technology leadership, without consideration of their current share price or the appropriate timing of an investment decision. The presence of a company on the list is not a recommendation to buy shares at the current price. Reference Price is the company's closing share price on the Reference Date, the day the company was added to the table, typically the last trading day of the month prior to publication. The authors and other Gilder Publishing, LLC staff may hold positions in some or all of the companies listed or discussed in the issue.

Ask Nick: Don't forget, all subscribers have exclusive access to Nick on the DS Forum. Just enter the subscriber area of the site and log on with your questions or comments.