Written by
Nick Tredennick

DynamicSilicon

Published by
Gilder Publishing, LLC

The Investor's Guide to Breakthrough Micro Devices

Hardware Gets Softer

This is about a sea-change in semiconductors that is enabled by Moore's-law progress. Hardware is getting softer. Hardware is softening in its circuit specifications. Circuits, such as a microprocessor, were once sold only as chips. Now, circuits are so complicated that the instructions—recipes—for creating these circuits are as important as the circuits themselves. Complex software tools read the recipes to produce data that drives the equipment in a semiconductor fab. Each of these recipes is so complex, that designers can no longer come up with all new recipes when they plan a new "meal." However, each meal is a little different, so designers want to be able to customize their recipes to fit the occasion. For these reasons, hardware design has become much less physical. The emphasis is now on creating detailed *soft* descriptions of hardware. The hardware's valued form exists as information in computer files.

Moore's-law progress has shrunken the microprocessor to a tiny speck on even a modest-size chip. Today, a designer might prefer to license the microprocessor's "soft" description to incorporate it in a larger design. This change is disrupting the industry. The change began with the introduction of the microprocessor in 1971 and is not based on some miracle that occurred last week. My discussion will range across many types of semiconductors to explain the trend and its causes. I will explain, in what seems like too much detail, what has brought us to this point. It is a story of scarcities and abundances, of cycles of component proliferation and consolidation, and of the tug of war between designer productivity and circuit efficiency. Designer productivity is the efficiency of the engineer. Circuit efficiency can be power economy, performance, or chip size.

Before the microprocessor, IC macros dominated electronic systems design. The microprocessor brought the flexibility of computer programming to non-computer applications (embedded systems), making embedded hardware softer. The microprocessor consolidated the proliferation of IC macros. It also increased the productivity of the designer. Raising the level of abstraction from circuit design to programming increased the pool of eligible designers and it improved their productivity. Since a microprocessor-based design used a smaller set of standard components, it was cheaper than the equivalent set of IC macros. Because it was cheaper and had adequate performance, the microprocessor displaced IC macros in many systems. As the microprocessor proliferated, so did the software content of systems. Moore's law progress increased the complexity of microprocessors. As more transistors fit, the microprocessor's peripheral functions (e.g., serial channel, floating-point unit, MPEG decoder) moved onto the chip, creating the microcontroller. Different applications needed different peripherals, spawning a huge variety of microcontrollers.

For circuits with extreme efficiency requirements, application-specific integrated circuits (ASICs) displaced IC macros. An ASIC is a custom integrated circuit built by a system manufacturer for a specific application. Moore's law progress increased the complexity of ASICs.

The programmable logic device (PLD), invented in 1966, got off to a slow start. It started slowly, because there weren't enough transistors on an IC in 1966 to enable meaningful applications and because the microprocessor dominated electronics applications. Beginning with programmable array logic (PAL) in 1978, programmable logic devices began to invade territory held by IC macros. Since the system maker and not the chip maker customizes the PLD (or PAL) by "programming" it, it is softer than the custom ICs it displaces. PLDs are still too slow and too expensive for many applications. But their speed and capacity improve with Moore's law.

Programmable logic will invade market segments held by microprocessors and microcontrollers and it will invade markets held by ASICs. It will do so because it wastes the abundance (transistors) to conserve the scarcity (the design-

In This Issue:

Vol. 1, No. 6
June 2001

er's time). The microprocessor, its peripherals, and any custom logic will move from being chips to being licensed design files. This transition will transform the industry and it will change the companies that dominate the industry. Here are the details.

IC macros

Fairchild Semiconductor introduced commercial integrated circuits (ICs, circuits on a chip) in 1961. This accelerated the proliferation of electronic systems. Families of compatible chips—so-called "IC macros" like registers, flip-flops, AND gates, and arithmetic units—decreased the cost of electronic systems, encouraging wider application. IC macros raised the level of abstraction in system design, improving the designer's productivity at the cost of decreased circuit efficiency. Assembling a system from compatible IC macros is quicker and is simpler than designing the same system by optimizing individual transistors, resistors, and capacitors. With IC macros, the engineer is no longer concerned with current flows and bias voltages. For the next ten years, the family of IC macros grew and improved. Chips for old IC macros got smaller, faster, and cheaper. New IC macros ran faster and grew more complex. The "gate," representing a two-input, one-output logic function, displaced the transistor as the unit of circuit complexity.

Microprocessors

By 1971, the pieces of a central processing unit (CPU)—decoder, state sequencer, registers, and an arithmetic and logic unit—could fit on a single chip. Intel introduced the first commercially available CPU on a chip, or microprocessor. The microprocessor brought the computer's problem-solving method to embedded systems. The microprocessor raised the level of abstraction—the engineer wrote programs for the microprocessor's standardized hardware instead of creating custom hardware with IC macros. The microprocessor's instructions took the place of electrical signals that would have been produced by the IC macro implementation. Raising the level of abstraction increased the designer's productivity and it increased the pool of designers, further accelerating the propagation of electronics. My 1971 Texas Instruments book lists 181 "7400-series TTL IC macros." The 7400-series TTL (transistor-transistor logic) was the most popular of several families of IC macros. A few standard components, microprocessors, ROM, RAM, and peripherals, displaced large numbers of IC macros, simplifying design choices for the engineer. Complex systems could fit on a standard circuit board.

The microprocessor reduced the number of ICs in a system, making the system smaller, cheaper, and more reliable. More importantly the microprocessor relieved the engineer of the responsibility to design a state sequencer, decoder, registers, and arithmetic and logic unit for each application. It also brought the power and flexibility of programming to the design of embedded systems. The microprocessor began the softening of hardware.

The microprocessor and other standard components did not eliminate IC macros. The microprocessor contributed a decoder, state sequencer, registers, and a general-purpose arithmetic and logic unit. The ROM held the program and reference data. The RAM provided working space. Peripheral components helped interface the microprocessor to the outside world. But these standardized components were *never enough* for practical systems; IC macros built special functions, converted inputs and outputs to the form expected by the microprocessor and its peripherals, and implemented odds and ends needed to complete the system. IC macros perform the functions commonly called "glue logic."

DynamicSilicon

Editors	Nick Tredennick Brion Shimamoto
Publisher	Lauryn Franzoni
Associate Editor	Keri Geiger
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. Copyright 2001, Gilder Publishing, LLC. Editorial inquiries can be sent to: bozo@gilder.com Single-issue price: $50. For subscription information, call 800.229.2573, e-mail us at dynamic@gilder.com, or visit our website at www.dynamicsilicon.com

Programmable Array Logic

In 1978 Monolithic Memories (MMI) introduced programmable array logic. "PALs" may have been the first commercial market for programmable logic devices. The inside cover of the first edition of MMI's *PAL Handbook* (1978) says: "These 15 parts will functionally replace up to 90% of 7400S/LS series functions." MMI's goal was to consolidate IC macros into fifteen programmable chips. MMI claimed a reduction in chip count of four to one over IC macro implementations.

A PAL is a chip that contains certain standard logic circuits and has several inputs and outputs. A PAL10H8 chip, for example, provided 10 inputs and 8 outputs. Each of these 10 signals is connected by a fuse to each of 1,024 logic circuits. *Each* of these 1,024 logic circuits is connected by a fuse to *each* of 8 logic circuits. PALs ship with all fuses intact. Standard electronics equipment called "a PROM programmers" blows the fuses necessary to produce the desired logical behavior. The PAL exploits Moore's law: thousands of logic circuits and fuses on the chip can be wasted to build functions that would use only a handful of custom logic circuits. The PAL trades efficiency for flexibility.

PLDs take the low end. Over time, the most popular PALs came to be the 16v8 (sixteen inputs and eight outputs) and the 22v10 (twenty-two inputs and ten outputs). These PALs mopped up the miscellaneous logic needed in a microprocessor- or microcontroller-based application. Derivatives are still available. You can buy today's version of the 22v10 through

Arrow Electronics (www.arrow.com) for about two dollars. But the 22v10 and 16v8 are pad-limited chips (March 2001 *Dynamic Silicon*), so larger-capacity PLDs from **Altera** (ALTR) and **Xilinx** (XLNX) meet their cost and performance. For example, Altera's EPM7032, a PLD with about three times the capacity of the 22v10 and also sold through Arrow Electronics, is two dollars. The low-end PLD from Xilinx, the XC9536, is available through Avnet Electronics (www.avnetmarshall.com) for a little over three dollars. Low-end, chips from the PLD makers are becoming the industry's glue logic.

Microcontrollers

Increasing complexity fostered the *microcontroller*. A microcontroller is a chip with a microprocessor, ROM, RAM, and peripheral functions all on the same chip. Motorola introduced its MC6801 microcontroller in 1978. A 16-bit timer, serial interface, 2 kB of ROM, 256 bytes of RAM, and 31 programmable input/output lines joined the 8-bit 6800 microprocessor on the chip. A little under a quarter-inch on a side, the MC6801 chip consisted of two layers in a 3.5-micron semiconductor process. A redesign of the MC6801 in today's 0.13-micron semiconductor process would probably fit under one of its forty bonding pads. A direct shrink of the original design would fit 725 copies of the circuit on a quarter-inch chip. The shrunken circuit would occupy only about 5 percent of a smaller chip, whose area would be determined by the size of the bonding pads.

Most of the billions of microcontrollers shipping each year go into low-end consumer products (the "zero-cost segment"). They don't need much processing power and they don't need fancy peripherals. They run microwave ovens, washing machines, toothbrushes, toasters, and hair dryers. They don't strain either the performance or the capacity of semiconductor processes. Some of these microcontrollers have been shipping for more than twenty years. Motorola shipped its first MC6805 microcontroller in 1979, for example. By 1981, there were already six variations. As I write this, Motorola's web site (www.motorola.com) lists *thirty-nine* variations in the M68HC05 family (not counting the various packages, speed grades, and other options). Once a design has shrunk to the point that it is pad limited, there's little incentive to continue shrinking the circuit portion, since the cost to make the chip won't decrease once the chip stops shrinking (limited by the ring of bonding pads). While 0.13-micron processes yield leading-edge microprocessors, most microcontrollers probably come from fully amortized 0.5-micron production lines. Microcontrollers don't need leading-edge performance, they need low cost.

Microcontroller variations occur as their range of application expands. Each application's requirements are a little different. An automatic transmission will want a certain set of counters, timers, serial and parallel interfaces, input/output lines, interrupts, and a host of other functions in its microcontroller. The Furby, the digital camera, the remote control, and the refrigerator will all want their own microcontroller variations. The variety of microcontrollers has burgeoned into the thousands from dozens of manufacturers, including Intel, NEC, Motorola, Toshiba, TI, Hitachi, Samsung, Philips, STMicroelectronics, and Siemens. Engineers at each of these companies design custom processors and custom peripherals for each custom microcontroller.

Suppose you are the design engineer on your company's "Einstein" smart camera project. Your first significant challenge will be to track down the right microcontroller for the application. There won't be a microcontroller that's a perfect fit. No microcontroller will combine exactly the right set of peripherals with exactly the processor you need in exactly the right physical package. You might beg a few manufacturers to build what you need. No way; they're too busy to build a custom microcontroller for your application unless you want millions of units a year. You don't have the time or resources to design your own custom microcontroller. It won't be long before you've dubbed your project the "Frankenstein" camera. That's the opening **Triscend** (www.triscend.com) is driving its business plan through.

Hard cores and soft cores

An ARM7TDMI 32-bit processor core, implemented in a 0.18-micron semiconductor process occupies less than 0.6 square millimeters. To give you an idea of scale, it would fit inside any "o" in this sentence. This ARM processor is called a "hard" core because it's designed as a fixed circuit that cannot be changed. When Triscend moves to 0.13 microns, it will have to obtain a redesigned processor core for its foundry's 0.13-micron process. The new ARM7 core will occupy far less than one percent of a modest-size 170 square millimeter chip. For most microcontroller applications, transistors are abundant in 2001. Add the functions everyone will need, such as memory, memory interface unit, and power-on reset, and there is still plenty of space left on the chip. Triscend allocates this area to programmable logic and to programmable interconnect, termed the Configurable System Logic (CSL) Matrix and the Configurable System Interconnect (CSI) Bus, respectively. This lets the engineer build an exact-fit microcontroller by selecting and connecting peripherals from a library of "soft" cores. These peripheral cores are "soft" because they exist as configuration files for the chip's programmable logic and programmable interconnect. When Triscend moves from 0.18 microns to 0.13 microns, the configuration files for these "soft" cores will still work. These "soft" peripherals will be faster and they will use less space without any investment in redesign.

The theme here is as follows:

- Separating function from a standard physical design.
- Supporting many functions with one physical design.
- Letting the standard physical design improve with Moore's law.

Triscend's "FastChip Soft IP Library" contains modules for serial communication, logic functions, memory, display drivers, controllers, input/output, and interface functions.

There's even a module for triple-DES encryption. Third parties can build modules to license through Triscend. If there's a special function you need that Triscend doesn't have, you can build it as a soft core.

Triscend also has chips with an 8051 8-bit, hard-core processor. The 8051 is the world's most popular 8-bit microcontroller architecture. It was specified by junior engineer John Wharton at Intel in 1980 and it was introduced the same year. Since its introduction, manufacturers have shipped more than 3 billion 8051s and variants. Twenty-one years after its introduction, it is shipping more than 300 million per year. It's a good first choice for a hard core. In addition to the ARM- and 8051-based chip offerings, Triscend collaborates with Hitachi. Hitachi will build Triscend-like chips using its SuperH microprocessor as the hard-core processor. Hitachi will manufacture the chips, aimed at telecommunications applications, in its own fabs.

Triscend has hard-core processors and soft-core peripherals. Triscend is a component supplier. It's great positioning for today's market: exploit the abundance of transistors to consolidate the microcontroller market. Triscend ships nine different chips into what would have been the market for thousands of microcontroller variations. In the semiconductor business, volume is king. Build one mask set, set up the production line, and run off a billion chips. That's cheaper than making a thousand mask sets and retooling the production line after each run of a million chips. It reduces inventory management for the manufacturer, for the distributor, and for the customer. Over time the portable soft-core library accumulates, broadening the range of applications and further consolidating the microcontroller market. In addition, semiconductor process improvements enable more complex cores and more cores per chip. This is a great strategy. *However*, the huge volumes and the great variety of microcontrollers come from the microcontroller's pervasive consumer applications. That is the zero-cost segment of the market, so the average selling price must be low. Triscend's long-term success depends more on market share than on short-term profit. Triscend should not make Apple's mistake of sacrificing market share for short-term profit.

It's a great story; you can get or can quickly create the peripherals you need on the main chip for the Einstein camera. But your camera needs special processing instructions that ARM, SuperH, and the 8051 processors don't have.

Configurable processors

ARC Cores' (ARK) configurable, soft-core processor is an answer. Just as Triscend's customers, using its software, "drag and drop" peripherals to build the perfect microcontroller, would-be computer architects, using ARC's "ARChitect" software, select instruction-set features for the Tangent-A4 processor core. You can download demonstration software from www.arccores.com. If you have a PC and Internet access, I recommend it. It compellingly demonstrates what it means to raise the level of abstraction in design. While hardware design languages such as Verilog and VHDL will look like hieroglyphics to you and me, this doesn't. Anyone who can read this paragraph can do it.

Simple drop-down menus offer choices in output format, instruction extensions, digital signal processing, cache configuration, memory, debug features, peripherals, and more. Counters in the corner of the screen tell you the clock speed and design size as you add features. In a half-hour or so, I created designs ranging from 8,444 gates to 155,644 gates. Even the largest of these designs fits easily in an Altera APEX II or a mid-range Xilinx Vertex II SRAM PLD. The design software's output is compatible with standard ASIC and PLD development tools. Run ARC's demo and see the future.

The engineer's expertise is applied in knowing whether, for example, to select saturating arithmetic or in choosing a 16x16 multiply-accumulate unit rather than a 24x24 multiply-accumulate unit. Engineers with special application requirements can add custom instructions, registers, condition codes, interfaces, and peripherals. When the engineer creates user-defined instructions, the software spits out a compiler that can generate the new instructions and a debugger that can recognize them. There's no need to spend weeks simulating the design—just download it into a PLD and run it to see how it works. It's easier to tune an engine by running it than it is to tune it by simulating it. You might develop an ARC-based custom design for the Einstein camera. You could prototype it in an Altera APEX II chip and run the camera's software to see how it works. Profiling the software tells where time is spent. Detailed study of bottlenecks in execution leads to performance-enhancing, user-defined instructions.

ARC's customers will want more peripherals. ARC could add third-party peripheral designs to its "Peripherals" drop-down menu and pass a portion of the license fee to the peripheral's supplier. Third parties can develop "plug-ins" for ARC in much the same way that programmers develop plug-ins for Photoshop and for Internet Explorer. ARC's competitors, vendors of hard and soft processor cores, include ARM, MIPS, Lexra, Motorola, IBM, TriMedia, picoTurbo, Lexra, and **Tensilica**. Some of these vendors, such as MIPS and IBM, offer configuration options. This month, for example, MIPS announced user-selectable configuration options for its MIPS32 4KE family cores and for the MIPS64 5Kf core. MIPS doesn't allow users to add custom instructions. Users can add customer-specific instructions to Lexra's LX4380 core. Tensilica offers user-configurable soft-core processors—and it does allow users to add custom instructions.

Microprocessors began as individual chips with instruction sets designed for broad market application. Peripherals moved onto the chip, splitting the market into microcontrollers and microprocessors. The microprocessor market sprouted a digital signal processor (DSP) segment. The microcontroller market will consolidate as microcontrollers move to standard designs with configurable soft-core peripherals. Triscend is leading this transition. Instead of being only chips, microprocessors and DSPs will become licensable, hard and soft cores. In addition, microprocessor and DSP instruction sets and programming models, rather than being designed for broad market acceptance will become malleable. Designers will mold malleable microprocessor and DSP cores to suit their application. ARC Cores and Tensilica are leading the transition to malleable cores. Many companies sell hard and soft cores. These transitions will not happen overnight. It will take years to retrain the industry to abandon entrenched practices.

Application specific integrated circuits

The application-specific integrated circuit is what it sounds like; it is a custom chip designed by a system manufacturer for a specific application. The ASIC consolidates custom hardware, such as a microprocessor, DSP, and custom peripherals, in a single chip. The ASIC differs from the microcontroller in that the ASIC is built for a specific application and may or may not include a microprocessor, while the microcontroller's central component is a microprocessor together with the peripherals for a range of applications. The single-chip solution is faster (wider connections and shorter wires) and is more energy efficient (on-chip drivers are smaller). If production volumes are high, such as in hard disks and in cell phones, it is cheaper. System companies, such as hard disk and cell phone designers, develop their custom chips through ASIC suppliers. ASIC suppliers, such as NEC, IBM, Fujitsu, LSI Logic, TI, VLSI Technology, and STMicroelectronics, provide development tools, circuit libraries, and chip manufacturing.

An ASIC is a custom design built by the supplier for a single customer. An application-specific standard product or ASSP is a custom design built by a supplier and sold to anyone. **Analog Devices'** (ADI) recently announced ADV-JP2000 digital camera chip is an ASSP, as is TI's TRF6150 direct-conversion radio chip. Leading ASSP suppliers include Analog Devices, Broadcom, Conexant, LSI Logic, TI, Qualcomm, STMicroelectronics, Infineon, and PMC Sierra. From the supplier's perspective, an ASSP has more customers and therefore higher volumes than an ASIC. From the customer's perspective, an ASSP is cheaper and it involves less development risk than an ASIC. The disadvantages of an ASSP are that it may not exactly fit your requirements and that it is available to competitors.

ASICs and ASSPs assemble hard and soft cores on a single chip. Functions may include peripherals, a microprocessor, a DSP, and custom logic. The ASSP and ASIC are typically well matched to the application, easy to use, and efficient. But ASICs suffer from long development time, high development cost, and difficulty in testing. Correcting an error in an ASIC can cost months of delay and millions of dollars. There are about 10,000 ASIC design starts every year, though according to Dataquest the number has been decreasing since its peak in 1997. Almost half of the design starts are for circuits requiring fewer than 250,000 gates. Two-thirds are below a million gates. At $12 billion (*Electronic News*, 4 December 2000), the ASIC market invites competition from PLD makers.

Programmable logic devices

Altera. Altera makes two kinds of PLDs that can be distinguished by how they store their configuration bits. One stores configuration bits in EEPROM (electrically-erasable programmable read-only memory), while the other stores them in SRAM (static random-access memory). EEPROM PLDs are often called complex PLDs or CPLDs.

Altera makes two families of CMOS EEPROM-based PLDs, the MAX 3000 family and the MAX 7000 family. MAX 3000 devices are low-cost PLDs ranging in size from 600 to about 5,000 logic gates. MAX 7000 devices range in size from 600 to about 10,000 logic gates. Distributor's unit prices for these components range from $1 to a little more than $100.

Altera has several families of CMOS SRAM PLDs: APEX, ACEX, FLEX, and Mercury. The APEX and ACEX, as the newest general-purpose device families, set Altera's strategic direction. APEX is the high end and ACEX is the low end.

APEX comes in two varieties, the APEX 20K series and the newer APEX II series. The logic element (groups of logic gates) and not the logic gate is the smallest incremental building block for constructing logic circuits, but here we are concerned with trends relative to other components, so we need a common measure. The low end of the APEX 20K series has 113,000 logic gates and 24 K RAM bits. The high end of the APEX II series, the EP2A90, has 7,000,000 logic gates and 1.5 M RAM bits. The newest APEX II components aren't yet available and. Unit prices for APEX 20K chips from distributor Arrow range from $336 to $900.

ACEX chips range from 10,000 logic gates and 12 K RAM bits to the EP1K100's 100,000 logic gates and 49 K RAM bits. Unit prices for ACEX chips from Arrow range from $10 to about $35.

Xilinx. Like Altera, Xilinx also makes EEPROM-based PLDs and SRAM PLDs. Xilinx has two families of EEPROM-based PLDs, the XC9500 family and the CoolRunner XPLA3 family. The CoolRunner family, acquired from Philips Semiconductors, is for low-power applications and has a range of capacities that matches Altera's MAX 7000 series. The XC9500 series, which has a lower maximum capacity, is similar but is built for performance.

Xilinx has several families of CMOS SRAM PLDs: Virtex, Spartan, and XC4000. Virtex and Spartan are the families of general-purpose devices that set Xilinx's strategic direction. Virtex is the high end and Spartan is the low end.

Virtex II is the newest series in the Virtex family. The low end of the Virtex II series has 40,000 logic gates and 78 K RAM bits. It also has four 18x18 multipliers. The high end of the Virtex II series, the XC2V10000, is announced, but is not yet available. It will have 10,000,000 logic gates, 192 18x18 multipliers, and more than 5 M RAM bits.

Spartan II is the newest series in the Spartan family. The low end of the Spartan II series has 15,000 logic gates and 3 K RAM bits. The high end XC2S200 has 200,000 logic gates and 64 K RAM bits. Unit prices for these chips from distributor Avnet range from $8 to $30.

SRAM PLDs. Xilinx and most of the rest of the industry call SRAM PLDs field-programmable gate arrays or FPGAs. SRAM PLDs are field programmable, but they are not arrays of gates. The smallest accessible logic element is equivalent to 150-200 gates (this varies among chip designs). Altera generally doesn't refer to its SRAM PLDs as FPGAs, preferring to call them PLDs or, SRAM PLDs, or "look-up-table devices." Differences in terms confuse observers, but Altera's APEX competes directly with Xilinx's Vertex for the high-end SRAM PLD market. Altera's ACEX competes directly with Xilinx's Spartan for the low-end SRAM PLD market. According to Altera, the PLD market was a little over $4 billion in 2000. Altera had 34 percent and Xilinx had 38 percent. A little over two-thirds of that revenue, for both companies, came from the communications sector. The primary uses for SRAM PLDs are in circuit prototyping and in applications that might otherwise use an ASIC.

As legend has it, "dog years" pass seven times faster than our calendar year. In three calendar years, your dog is twenty-one. ASIC vendors will tell you that PLD vendors measure their chips in "dog gates," meaning the effective gates you can get from a PLD design will be less than a third or fourth of the advertised number. That's OK. The numbers are big and they're growing with Moore's law. We're concerned with trends. Because of Moore's law, a factor of four is only three years.

All but the largest ASIC designs would fit in an existing or announced SRAM PLD. About a quarter of the ASIC starts would fit in a chip that's less than $30 (quantity one). In a year, *half* of ASIC design starts will fit in a $30 chip. PLD makers are adding hard cores to improve performance. Altera, for example, offers MIPS and ARM hard-core processors on "Excalibur" derivatives of some APEX chips. Xilinx has hard-core 18x18 multipliers on its Virtex II chips. LSI Logic, **Cypress** (CY), and Actel have countered by offering programmable logic on ASICs.

Altera and Xilinx are moving into soft-core IP licensing. Altera, for example, offers the soft-core Nios processor and a wide range of "MegaCore" soft-core IP for signal processing, communications, bus interfaces, and memory controllers (www.altera.com/ipmegastore). Xilinx has announced its MicroBlaze soft-core processor and it has a large library of soft-core IP (www.xilinx.com/ipcenter). Xilinx has at least two dozen IP "AllianceCORE" partners. Altera and Xilinx look like chip suppliers, but they have always been in the software business (PLD tools). Now they are soft-core IP brokers as well. If they weren't already dynamic silicon companies, I would add them for this.

Programming, programming languages, hardware description languages. "Programming" for PLDs and PALs loads a file of configuration bits to "personalize" the PLD or PAL. This personalization makes the PLD behave like a specific piece of custom logic. In the case of an SRAM PLD, a separate ROM chip typically stores the configuration bits. When the system boots, the PLD reads the configuration bits from the ROM and stores them in the SRAM locations that control the logic element functions and wire connections. "Programming" a PAL blows fuses. PLD or PAL "programming" uses a file of bits to configure the device. It has nothing to do with *running* programs or with "programming languages." Programming languages, such as C, C++, and Java, whose purpose is automatic symbol manipulation, are bit files that "run" on a computer. Hardware designers employ programming languages to solve analytical problems or to write executable behavioral specifications for hardware. "Hardware description languages," that look and sometimes act like programming languages, specify hard and soft cores. Hardware description languages—are for writing hardware "recipes"—they describe a circuit with enough detail to specify both its construction and its behavior. Hardware description languages drive simulations that verify the design and they are inputs to the development systems that produce the circuits.

Celoxica

Celoxica (www.celoxica.com) is a pre-IPO startup. Celoxica's DK1 software suite, which sports a customized version of the C programming language called Handel-C, maps algorithms into SRAM PLDs.

I visited reconfigurable computing expert Wayne Luk at Imperial College in London. He introduced me to Ian Page, Celoxica's CTO. I spent half a day with Ian. We're in agreement on the ascendance of dynamic logic, but I told him I didn't favor describing hardware with programming languages. Special languages, such as Verilog and VHDL, are designed to meet the needs of hardware description. Hardware is inherently parallel, while programming languages are inherently serial. Compromising the hardware description to fit a standard sequential language maintains the integrity of the programming language. That keeps the language's user base, but it forfeits information about the parallel nature of the problem. It's like asking a compiler to build an internal-combustion engine from a sequential listing of components. And compromising the programming language to accommodate parallel constructs renders the programming language unique. This reduces the user base and forces construction of new development tools.

Ian and other researchers spent eleven years working on it at Oxford University, so he thinks they've got the programming language right. I'm changing my view. First, using programming languages for hardware design isn't about design efficiency, it's about raising the level of abstraction, raising engineering productivity, and empowering more designers. Second, specifications are changing. Twenty years ago, specifications were written in English. Today, complex specifications are written in programming languages. Executable specifications simplify verification of the design's correctness. If the specification is written in Handel-C, there's no intermediate translation from an executable specification to a hardware description language. Simulation verifies the specification; the specification compiles into a netlist (how the circuits connect) that's compatible with PLD vendors' logic placement and wiring tools. As functions get more complex, they move toward software content anyway—MP3, TCP, voice over IP (VoIP), encryption, and compression.

So much for my objection to programming languages to describe hardware.

Celoxica is working with Wind River Systems (www.wrs.com) and with Xilinx to develop the application programming interfaces and protocols necessary to exchange information between the programs running on a board's microcontroller and its SRAM PLDs. The objective is to let application code run as SRAM PLD hardware. A major piece of Celoxica's strategy is soft-core IP (intellectual property). Soft cores are the future and have the necessary leverage. Since IP licensing is the major strategic direction for Celoxica, it is a dynamic silicon company.

The big picture

For many years the microprocessor was king. It was the central component in a board design. The digital signal processor (DSP) came along in the mid-1980s to share the limelight. Most designs that need a DSP use both; the microprocessor runs the decision functions and the DSP does math-intensive processing. Moore's law threatens their independent positions as peripheral functions move onto the chip and as the processors themselves soften and move from silicon chips to reusable recipes (cores).

An ASIC and PLD battle rages. ASIC vendors say PLDs are slow and expensive and have more overhead than the U.S. government. PLD vendors say ASICs are inflexible dinosaurs and that half of ASIC design starts never finish. This reminds me of the battle between assembly language programming and high-level language programming. The programming battle, which surges to life even today, was fought over memory efficiency and over execution performance of hand-written assembly programs versus high-level languages and compiled code. Programming languages and compilers won, but not because they achieved the efficiency and performance necessary to wipe out assembly language programming. Compilers and languages won because the real issue was not memory use and performance, it was designer productivity. The scarce resource was the designer's time. Moore's law made memory and performance abundant. It's easy to cite cases where memory or performance mandates special attention, but these are exceptions. High-level languages and compilers raised the level of abstraction. The programmer described algorithms in terms such as multiply, divide, square root, sine, and tangent—terms closer to the application—rather than in assembly language, which has terms, such as load, store, test, branch, shift, and add, that are closer to the hardware. Raising the level of abstraction raised the designer's productivity. Raising the level of abstraction also increased the pool of qualified workers. Computer access was once limited to a technical priesthood; software in the form of browsers, word processors, spreadsheets, and graphical user interfaces has raised the pool of computer users into the hundreds of millions.

The ASIC and PLD battle rages over efficiency and performance, just as the battle between assembly language programming and high-level language programming did. It's also just as misguided. In the hardware domain, just as in the software domain, the fundamental scarcity is still the designer's time and the fundamental abundance is still transistors. Soft cores will displace hard cores because soft cores raise the level of abstraction. The soft core's representation is a program; the hard core's representation is a precise physical layout. Soft cores waste the abundance (transistors) and conserve the scarcity (the designer's time). Rather than having thousands of engineers independently designing hard-core peripherals for dozens of vendors of ASICs, ASSPs, and microcontrollers, there will be a few engineers at a few companies (dozens?) designing soft cores to be licensed to chip suppliers.

In the microcontroller market, Triscend is a chip supplier whose soft-core peripherals will take market share from its hard-core-based rivals. Triscend's rivals will either convert to soft cores, as Hitachi has begun to do, or they will lose market share. Triscend, which today mixes hard and soft cores on its chips, will be encouraged by the market to migrate to soft cores for all of its cores.

In the beginning of the PC business, the companies that built the systems had high margins. As the PC became a commodity, profits migrated from the system makers to the makers of the key enabling components. The PC's key enabling components are the microprocessor and the operating system. In the chip business, the key enabling components are manufactured silicon and soft-core IP. Altera, Xilinx, Triscend, ARC, Tensilica, and Celoxica are in the soft-core IP business.

Chips and Technologies was the first "fabless" semiconductor company. Up to its time, it was assumed that you had to build your own chips to be in the chip business. Today, even huge semiconductor producers such as Motorola contract chip production to foundries like **TSMC** (TSM) (www.tsmc.com) and **UMC** (UMC) (www.umc.com). Altera and Xilinx are fabless semiconductor companies. Most of Altera's chip production goes through TSMC, while Xilinx uses UMC as its principal foundry. Chartered (www.csminc.com) has a process-development agreement with Lucent and a multi-year production contract with Broadcom. TSMC, UMC, and Chartered own the foundries supplying manufactured silicon. Product supplied to Altera and Xilinx is so important to these foundries that programmable logic, rather than memory, is the principal process driver for the foundries. TSMC and UMC are dynamic silicon companies.

Nick Tredennick Brion N. Shimamoto

Nick Tredennick and Brion Shimamoto
June 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, several companies on this list may be startups. We will have much to say about these companies in future issues.

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.

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

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.

Technology Leadership	Company (Symbol)	Reference Date	Reference Price	5/31/01 Price	52-Week Range	Market Cap.
General Programmable Logic Devices (PLDs)	Altera (ALTR)	12/29/00	26.31	24.00	18.81 - 67.12	10.4B
Dynamic Logic for Mobile Devices	QuickSilver Technology, Inc. (none*)	12/29/00				
MEMS Foundry, Dynamic Logic	Cypress (CY)	12/29/00	19.69	21.15	13.72 - 55.75	2.9B
RF Analog Devices, MEMS, DSPs	Analog Devices (ADI)	12/29/00	51.19	44.55	30.50 - 103.00	15.3B
Configurable Microprocessors	ARC Cores (ARK**)	12/29/00	£3.34	£0.98	£0.82 - 4.29	£499M
Field Programmable Gate Arrays (FPGAs)	Xilinx (XLNX)	2/28/01	38.88	41.25	29.80 - 98.31	14.2B
Configurable Microcontrollers (Peripherals)	Triscend (none*)	2/28/01				
Silicon for Wireless RF, GPS	SiRF (none*)	12/29/00				
Microprocessor Instruction Sets	Transmeta (TMTA)	12/29/00	23.50	12.16	10.67 - 50.88	1.8B
Photonic Switches	Calient (none*)	3/31/01				
DKI Development Suite	Celoxica (none*)	5/31/01				
Design Environment Licensing for Configurable Soft Core Processors	Tensilica (none*)	5/31/01				
CMOS Semiconductor Foundry	Taiwan Semiconductor (TSM†)	5/31/01	19.86	19.86	16.13 - 39.88	47.7B
CMOS Semiconductor Foundry	United Microelectronics (UMC†)	5/31/01	10.16	10.16	7.06 - 15.19	21.8B

* Pre-IPO startup companies.

** ARK is currently traded on the London Stock Exchange

† Also listed on the Taiwan Stock Exchange

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.