LIKE THE IDEA OF A HANDHELD DEVICE that can be any of about 10 different gizmos, depending on your mood? You could soon have one if the ideas described in this two-part report become reality. In this article, Nick Tredennick and Brion Shimamoto of the *Gilder Technology Report* tell why only programmable logic devices could do the job. On p. 41, Diederik Verkest of the Interuniversity Microelectronics Center in Leuven, Belgium, describes a chameleon-like handheld being developed.

G RECONFIGURE

Programmable logic devices will give us a handheld that does everything—well

A viewpoint by
Nick Tredennick
& Brion Shimamoto

YOU WANT ONE DO-IT-ALL DEVICE. Something the size of an iPod that's a PDA, a cellphone, a GPS receiver, an MP3 player, an e-book reader, a digital camera, a portable television, a satellite radio, and a game player. It communicates with any wireless network it encounters, without prompting from you. It has lots of processing horsepower; it upgrades automatically; and it goes for days, not hours, on a battery charge. You might call it a universal digital assistant.

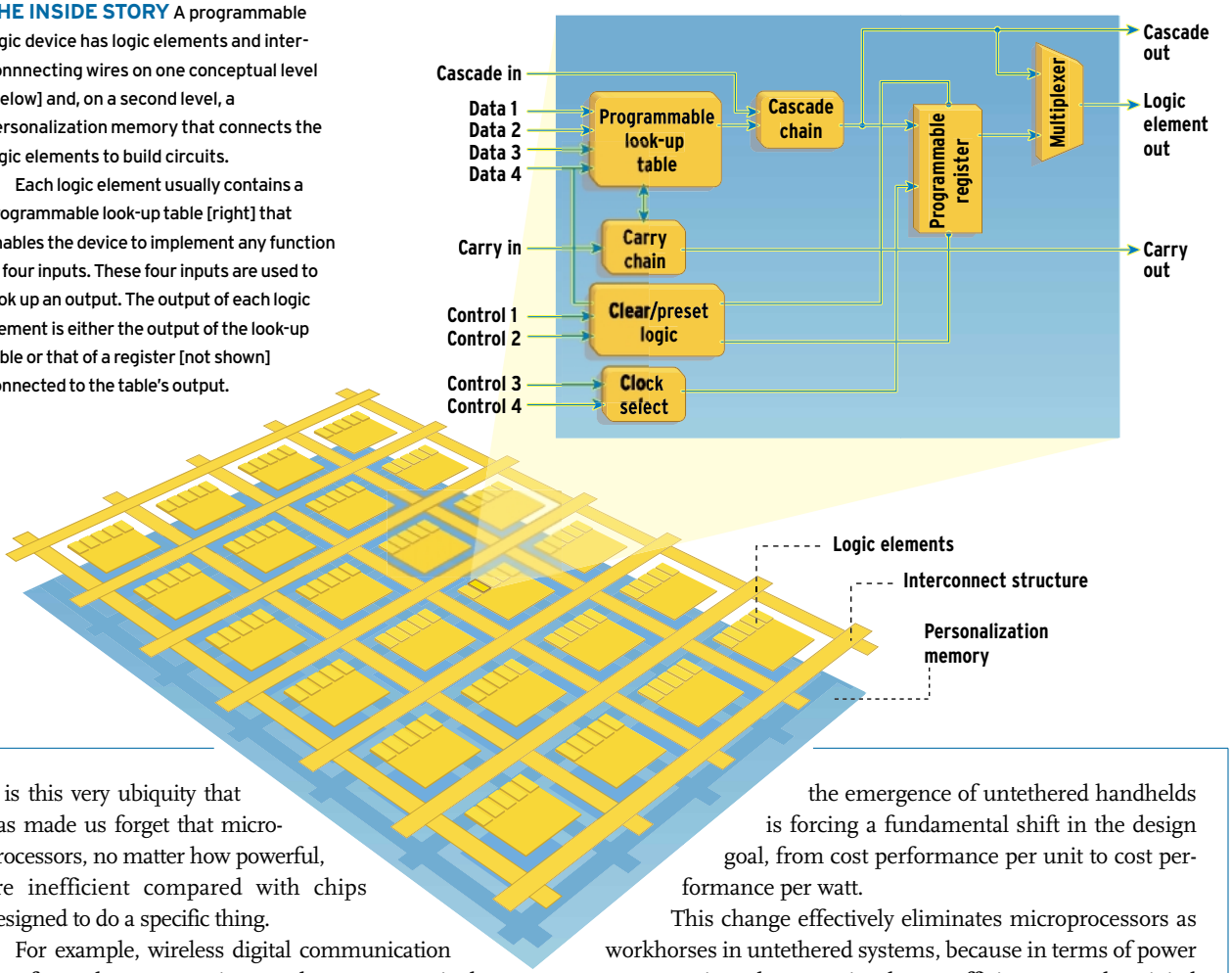
What you've got now is more separate devices than you have pockets, with an equal number of chargers, cables, and nonstandard lithium-ion batteries that never last long enough and that die at the worst moment.

Can this pocket-electronics menagerie be consolidated in one device that adapts to any network, changes to fit new standards, and upgrades its security system for the latest virus? It can and it will, but getting there will not be business as usual.

We are at a turning point in electronic design. We got to today's profusion of handheld gadgets by speeding down the road paved by the microprocessor. But the microprocessor won't take us to the end of the rainbow, where our universal digital assistant awaits. We'll need something else for that: the programmable logic device (PLD), a semiconductor containing a personalization memory and logic elements. Bits in the personalization memory set up temporary physical connections to build complex digital circuits and then can reset them at will.

The case against microprocessors

For more than 25 years, the microprocessor has been the heart and soul of electronic systems. Microprocessors are in everything from personal computers to washing machines, from digital cameras to toasters. But



THE INSIDE STORY A programmable logic device has logic elements and interconnnecting wires on one conceptual level [below] and, on a second level, a personalization memory that connects the logic elements to build circuits.

Each logic element usually contains a programmable look-up table [right] that enables the device to implement any function of four inputs. These four inputs are used to look up an output. The output of each logic element is either the output of the look-up table or that of a register [not shown] connected to the table's output.

it is this very ubiquity that has made us forget that microprocessors, no matter how powerful, are inefficient compared with chips designed to do a specific thing.

For example, wireless digital communication uses forward error correction to reduce error rates in the presence of noise. One manufacturer, Texas Instruments Inc. (Dallas), found in its design work that two of these forward error-correction techniques tied up as much as 90 percent of a digital signal processor's computing capabilities. So instead of processing forward error correction with the signal processor's instructions, Texas Instruments built custom logic circuits to handle that function in its TMS320C6416 digital signal processor. These logic circuits sped up the functions dramatically, in one case by a factor of 18 and in the other by a factor of four.

Until now, such inherent inefficiency mattered little. The microprocessor's signature application was in PCs, which were, traditionally, tethered: they plugged into an electric outlet. So to boost performance, engineers simply increased the microprocessor's speed and therefore its power consumption. Microprocessors have become dizzyingly fast, churning out billions of instructions per second at clock rates above 3000 MHz (more than 600 times the clock rate of the PC's first microprocessor) and consuming a fairly remarkable 80 W or more in the process. That's a power-density level, in watts per square centimeter, that is more than five times that of a stovetop cooking surface. But it hasn't been much of a problem, because the power has been coming from a wall socket.

The rise of the laptop computer in the 1980s and 1990s altered the equation somewhat, prompting manufacturers to at least begin paying attention to microprocessor power levels. But the emergence of untethered handhelds is forcing a fundamental shift in the design goal, from cost performance per unit to cost performance per watt.

This change effectively eliminates microprocessors as workhorses in untethered systems, because in terms of power consumption, they are simply not efficient enough. Digital signal processors—microprocessors optimized for continuous calculation on long data streams—are similarly unsuited for untethered systems. Instead, microprocessors will be kicked upstairs, to the role of supervisors. As overseers, they will manage systems' tasks, much as drivers manage their automobiles' tasks but don't do the work of propelling all that mass.

A role for ASICs?

The obvious place for the designers of untethered products to turn to was the area of application-specific integrated circuits (ASICs). These are special-purpose chips designed by one manufacturer for one product. Nokia Corp. (Keilahdentie, Finland) and LM Ericsson (Stockholm, Sweden), for example, have special ASICs that are the core of digital baseband processing in their cellphones. Baseband functions include the vocoders, codecs, user interface functions, peripheral controllers, and protocol processors.

An application-specific standard product (ASSP) is, like an ASIC, a special-purpose chip for a particular application, but it can be used by many system makers. Analog Devices, Infineon, Motorola, Qualcomm, and Texas Instruments, for example, make ASSP chip sets that anyone can buy to build cellphones. But ASIC buyers can design unique features into their phones; ASSP buyers can't.

ASICs and ASSPs are more efficient than ordinary microprocessors because they implement functions in hardware rather than in software. They can be hundreds or thousands of times more efficient, meaning lower cost and higher performance per

STEVE STANKIEWICZ

watt, than a microprocessor-based implementation. Image-processing functions that consider values in neighboring pixels, for example, are much more efficient in custom circuits of an ASIC or ASSP than in the instruction-based circuits of a microprocessor. An ASIC or ASSP can implement complex functions that act on blocks of pixels, while the microprocessor is restricted to simple functions on a serial stream of pixels.

More efficient though they are, ASICs and their ilk still do not carry us toward a universal device—one that will do a variety of chores—because they're not versatile. In fact, ASICs, as much as anything else, have contributed to the proliferation of single-purpose gadgets. A universal handheld would need dozens, one or more for each of its many functions. Meanwhile, rising mask costs put the price of starting production of a new ASIC at US $2 million and are making devices using them prohibitively expensive.

The case for PLDs

What is needed is something that combines the performance and efficiency of special-purpose hardware with the versatility of a programmable device such as a microprocessor. In other words, we need a programmable logic device. Some of these already exist. Conceptually, they are two-layer devices [see "The Inside Story," p. 38]. One layer contains configurable wiring and configurable logic elements. The second is the so-called personalization memory.

The PLDs commonly called field-programmable gate arrays (FPGAs) use static random-access memory (SRAM) as the personalization memory. SRAM is fast, but it's expensive (six transistors per memory cell), and it retains its contents only while the power is on. Bits in the personalization memory configure each logic element and specify how these elements interconnect to build a custom circuit that carries out the desired function—such as a 17-bit multiplier or a universal asynchronous receiver-transmitter (UART).

Today's PLDs are one-time programmable. Those using SRAM load personalization bits when the system initializes. A few PLDs are partially reconfigurable—that is, some logic elements and their connections can have their configuration changed while others remain fixed. An interface circuit might be adapted to a different protocol without changing the rest of the chip's functions. And a few PLDs are dynamically reconfigurable, allowing the configuration of any of the logic elements and their connections to be modified while on-chip circuits are operating.

The SRAM personalization memory is what makes PLDs reconfigurable. Think of your cellphone as a collection of transistors. Rearrange the transistors and it becomes a GPS receiver. While it isn't really practical to program a changing arrangement of transistors, it is practical to program a changing arrangement of logic elements: AND gates, OR gates, and so on. Changing the connections among individual transistors isn't feasible because each programmable connection requires at least seven transistors. With multiple possible input and output connections, the number of transistors needed as overhead to build connections among the logic element's transistors is far more than the number of transistors in the logic element.

Now imagine your cellphone as a collection of programmable logic elements. These could be programmed to form anything from a simple inverter to a multiplier. The universal, undifferentiated logic elements, after they're programmed, build a collection of specific logic functions. Rearrange the logic elements and the cellphone becomes a GPS receiver. Rearrange them again and it becomes an MP3 player. The system simply loads, or "pages," different hardware into the chip the same way an operating system pages programs into memory; functions that aren't needed aren't present, so they don't waste power. Someday soon, that is how our universal digital assistant will operate.

Before that happens, though, engineers will have to make programmable logic devices a lot faster and more powerful. Today's PLDs trade away too much of their efficiency for the sake of versatility. Although they deliver higher performance than microprocessors, they have a lot of setup overhead compared to ASICs, requiring as many as 20 transistors to accomplish what an ASIC does with just one. The ASIC implements custom functions, a custom interconnect structure, and custom inputs and outputs. The PLD has general-purpose logic elements, a general-purpose interconnect structure, and general-purpose (chip) inputs and outputs.

Transistors in the PLD's personalization memory are overhead that isn't there in an ASIC, as are all those that define interconnections. Personalization memory may account for 70 percent of the chip's transistors, and the programmed circuit configuration may waste (leave unused) most of the transistors in a logic element. A logic element configured as a simple AND gate, for instance, wastes almost all of its transistors. Indeed, in every configuration of a logic element, some of the transistors are unused. And long wires and transistor connections at wire intersections in the PLD slow circuit operation.

The bottom line is that, for complex functions, current PLDs are big, slow, power hungry, and expensive, and they come with high transistor overhead—fatal flaws for the core of an untethered system.

PLD makers are starting to address these problems, but they have a long way to go. Today, a high-end PLD can consume as much power as a microprocessor and can cost more than $1000. SRAM PLDs started in low-volume prototyping applications but are moving into consumer markets. Altera Corp. and Xilinx Inc. (both in San Jose, Calif.), for example, offer low-end SRAM PLD families—Cyclone and Spartan, respectively—priced below $5 for use in cost-oriented consumer applications. These companies, the current PLD market leaders, are extending the features of these devices to compete better with microprocessors and digital signal processors.

The latest PLDs, with high-speed input/output, memory blocks, multipliers, and on-chip microprocessors, compete well in high-end tethered applications. Unfortunately, these features increase the devices' power demand and make them

RICHARD TUSCHMAN

less, not more, suited to untethered operation. Our universal digital assistant isn't going to do us much good if we need to keep it plugged in to the end of a wire.

Down the road

Flawed as it is, the PLD is our best chance to realize our dream handheld. In the next several years, manufacturers will reduce PLDs' input/output overhead and wiring overhead and will speed configuration (making them easier and quicker to customize for specific applications). They will also enable partial reconfigurability so that important functions remain resident and infrequently used circuits are paged in as needed. In addition to these improvements, today's logic elements will evolve into higher-level logic primitives derived from applications experience. Such changes will make PLDs more suitable for untethered applications.

As for companies making good progress—there aren't any. It seems that new PLDs should be here by now, but established PLD companies have their hands full. More interested in growing with their current customers, they'll be slow to move to radically changed chip designs. Start-ups are difficult to establish in chip markets, and they've had problems getting funding. A good example of vision for what is needed comes from QuickSilver Technology Inc. (San Jose, Calif.). But QuickSilver has had trouble deciding whether to offer development software or chips or licenses—and for which applications. Another start-up, Ascenium Corp. (Soquel, Calif.), builds a reconfigurable processor but is having difficulty getting funding, partly because chip development is too expensive.

Other small companies working on reconfigurable systems include Cradle Technologies, Elixent, FlexLogics, GateChange Technologies, IP Flex, Leopard Logic, MathStar, Morpho Technologies, picoChip Designs, Savion, and Stretch.

The host of large companies dabbling in reconfigurable systems include Intel, Motorola, NEC, Nokia, and Texas Instruments. There are also numerous companies working on the closely related software-defined radio, which uses software to control functions such as protocol, wave form, and frequency that are built into hardware in conventional radios. Many of these larger companies are misappropriating the term "reconfigurable."

Partial reconfiguration or dynamic reconfiguration are capabilities that multiply the chip's effective capacity by reusing the same logic elements and wires for numerous temporarily resident circuits. Most commercial PLDs today don't offer this capability, but there is no technical barrier to doing so. And efforts have begun. Start-up companies, like QuickSilver, are developing PLDs specifically for untethered applications. Elm Technology, IBM, Matrix Semiconductor, Tezzaron Semiconductor, Ziptronix, and others are working on chips with stacks of silicon layers that are connected internally by thousands of vertical wires. This shrinks the size of the transistors that amplify digital outputs so that the signals reach distant inputs quickly through short vertical wires. Shorter wires and smaller drive transistors reduce power draw and speed circuits.

Another improvement will come when PLD makers replace power-hungry SRAM with more efficient nonvolatile memory. Today's flash memory wears out and is too slow to keep up with the demands of reconfiguration. Already dozens of companies are working on advanced memories. Three leading candidates are magnetoresistive memory, ferroelectric memory, and ovonic unified memory. Each of these uses exotic materials with special magnetic, electrical, or phase-change properties to store bits compactly and without volatility [see "The New Indelible Memories," *IEEE Spectrum*, March, pp. 49–54].

What's needed is a new type of PLD—manufactured generically and customized in the field

Replacing the SRAM-based configuration memory with fast nonvolatile memory will improve PLDs' performance, circuit capacity, ease of use, and security. Security is improved because nonvolatile memory keeps personalization bits inside the chip. SRAM PLDs load personalization memory from off-chip storage on initialization.

So what's needed is a new type of PLD—manufactured generically and customized in the field—along with the development of systems that will enable the engineering base of programmers to design digital circuits instead of just writing programs.

Only programmable logic has the efficiency and versatility needed to enable a handheld device to be all things to all people. But the microprocessor isn't going away. Under its supervision, the next-generation PLD will be the workhorse of the utopian do-it-all consumer device. ●

TO PROBE FURTHER

QuickSilver Technology Inc.'s *ACM Technology Guide* provides a good description of reconfigurable systems concepts. It can be found at http://www.qstech.com/acm_tech_guide.htm.

One of the oldest conferences on reconfigurable computing is FCCM (FPGAs [Field-Programmable Gate Arrays] for Custom Computing Machines). The Web site for the conference is at http://www.fccm.org.

Steve Guccione has a searchable bibliography collection at http://www.io.com/~guccione/Bib/Bib.shtml.

For an overview, see "Reconfigurable Computing: A Survey of Systems and Software," by K. Compton and S. Hauck, *ACM Computing Surveys*, Vol. 34, no. 2, June 2002, pp. 171–210.

For reconfigurable systems research at the Interuniversity Microelectronics Center (IMEC) in Leuven, Belgium, see the next article [p. 41] and IMEC's Web site at http://www.imec.be/ovinter/static_research/reconfigurable.shtml.

Disclosure: Nick Tredennick has financial interests (public stock, investments, or stock options) in a number of companies developing reconfigurable systems, including Altera, Ascenium, QuickSilver Technology, and Xilinx.

RICHARD TUSCHMAN