DIGITAL SIGNAL PROCESSOR TRENDS

ADVANCEMENTS IN DIGITAL SIGNAL PROCESSING TECHNOLOGY ARE ENABLING ITS USE FOR INCREASINGLY WIDESPREAD APPLICATIONS. DEVELOPERS WILL BE CHALLENGED TO USE THIS PROCESSING POWER TO ITS UTMOST, WHILE CREATING NEW APPLICATIONS AND IMPROVING EXISTING ONES.

Gene Frantz
Texas Instruments

During the past decade digital signal processors (DSPs) have hit critical mass for high-volume applications (Figure 1). Today, the entire digital wireless industry operates with DSP-enabled handsets and base stations. The mass-storage industry depends on DSPs to produce hard-disk drives and digital versatile disc players. Ever-increasing numbers of digital subscriber line and cable modems, line cards, and other wired telecommunications equipments are based on DSPs. Digital still cameras, hearing aids, motor control, consumer audio gear such as Internet audio are just some of the many mass market applications in which DSPs are routinely found today. More specialized DSP applications include image processing, medical instrumentation, navigation, and guidance.

With the growing importance of DSPs and their applications, it seems appropriate to look at the changes occurring in these devices and to hazard a few guesses about where DSP innovations will lead in the opening decades of the new century. The continued growth of DSP-enabled applications will depend on developments in several areas of technology: the underlying manufacturing processes, the DSP core and chip architectures, and the software for development and applications. An additional factor, and the most difficult one to anticipate, is innovation. In a few years, designers will be dealing with DSPs that integrate hundreds of millions of on-chip transistors and deliver performance measured in trillions of instructions per second. (See 1999 IEDM short course on system on a chip by author, available for sale at http://shop.ieee.org/store.) Determining how to use that processing power effectively will require imagination that goes beyond conventional engineering methodologies.

Why have DSPs done so well in the last few years? The DSP phenomenon is part of the overall microprocessor success story, and it must be seen in that light. Like the high-end reduced instruction set computing (RISC) engines used in computers and the medium-range RISC microcontrollers in embedded systems, DSPs are becoming increasingly differentiated, designed to handle the processing tasks of specific types of applications. This trend will continue with all microprocessors in the years ahead, and it will be responsible for much of the future success of DSPs.

A specialized architecture

Although DSPs are similar to RISC engines in some respects, they're fundamentally different in other ways. These differences date from the earliest microprocessor architectures, and they'll continue to influence the development of DSPs and their applications in the years ahead. Essentially, DSPs are designed for

0272-1732/00/$10.00 © 2000 IEEE

number crunching. Early computer theorists realized that many interesting mathematical functions could be performed by a series of high-speed multiplications and additions.[1,2] Since many of these math functions are useful for transforming and manipulating analog signals in the digital realm, a machine that would perform them efficiently would be extremely valuable as a DSP. Accordingly, certain microprocessor architects designed their processors around hardware dedicated to performing multiply-accumulate functions, and DSPs were born.[3,4]

Initial DSP designs borrowed another idea from early computer research as well. The first microprocessors,[5] like the computer central processing units that preceded them, employed a von Neumann architecture,[6] with a single bus and a unified address space for both data and instructions. However, at one time a research team at Harvard, in designing Eniac, had proposed a different architecture that used separate buses and address spaces for data and instructions. DSP designers seized on the Harvard architecture, with its separate buses, but they used the idea in a novel way. In addition to adding a bus for instructions, designers provided separate buses for each multiply-accumulate operand. Thus, data and instructions could be loaded and a complete multiply-accumulate performed during every cycle. Since designers accepted the value of unified address space, they didn't split instructions from data in the main memory, though caching schemes introduced later often keep small amounts of data and code separate in on-chip memory. This modified Harvard architecture has been an integral part of DSPs ever since, even though today's architectures may include a number of functions that the original computer researchers couldn't have imagined in their wildest fantasies.[7]

Deterministic operation

Since DSPs are used for processing continuous signals that come from, and often go back into, the real world, they're constrained to operate in real time. This constraint is another key difference between DSPs and other microprocessors, not only in application, but also in the underlying architecture. Every signal-processing task operating on a DSP must be deterministic. That is, the time it requires to finish must be determined exactly, or it runs the risk of breaking up the signal processing. Any function that can disrupt the determinism must be eliminated from the architecture, or modified so that it's not disruptive.

Interrupts are the most notable example of a disruption. Signal-processing tasks simply cannot be set aside while the processor performs system functions. High-performance RISC engines cannot manage more than a light load with digital signal processing because they're interrupt driven. Fortunately, today's DSPs offer so much performance overhead that they can handle deterministic signal-processing tasks during regularly scheduled periods, then deal with interrupts and other non-real-time tasks during the intervals between these periods.

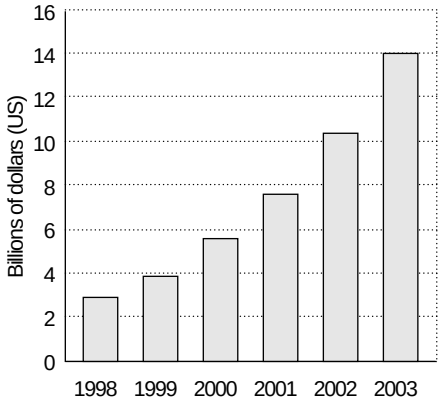


Figure 1. DSP market size (source: Forward Concepts).

Architectural changes

Increasingly, DSPs and other types of microprocessors have borrowed structures from each other, so that the line sometimes seems blurred where one type of processor leaves off and another begins. DSPs have become more supportive of the types of functions traditionally performed by microcontrollers and high-end RISC microprocessors. Interrupt support, which is critical to multitasking in embedded control systems, is now a regular feature of many DSPs that are meant to combine control and signal-processing functionality in a single device. Direct memory access control and various types of input/output peripherals are also routinely integrated into DSPs to provide the system-level support needed in a single- or satellite-processor application.

Two-level cache memories have been adapted from high-end RISC engines for the special requirements of DSPs. The two-level cache architecture makes a relatively small on-chip memory look like a much larger one to the core—enabling extremely fast DSPs to operate without outstripping the data available at a given time. At the same time, the cache design, coupled with the sheer speed of the DSP, provides enough configuration flexibil-

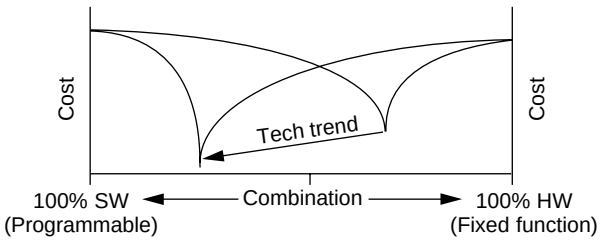


Figure 2. Combining software and hardware for the lowest cost system design. Cost can be defined in terms of financing, design cost, manufacturing cost, opportunity cost, power dissipation, time to market, weight, size.

ity and performance overhead that system designers can maintain the determinism they need for critical signal-processing tasks.

Greater parallelism

The most far-reaching recent innovation, though, is the introduction of very-long-instruction-word (VLIW) architectures to DSP cores. VLIW architectures are inherently parallel, providing multiple data paths for performing multiply-accumulates and other operations simultaneously. The introduction of Texas Instruments' TMS320C6000 core in 1997, the first DSP core based on a VLIW architecture, immediately raised the performance ceiling for DSPs by an order of magnitude. Top-flight DSP performance was no longer measured in hundreds of millions of instructions per seconds (MIPS), but in thousands of MIPS. A similar jump also occurred to million multiply-accumulates per second, the critical benchmark for number crunching.

When a VLIW architecture is supported by a carefully tuned C compiler, the powerful performance of the DSP engine becomes both highly efficient and easy to use. Programmers who have little familiarity with DSPs can then write code quickly without becoming familiar with the instruction set and underlying mechanics of the processor. A two-level cache memory also enhances ease of use by eliminating the need to micromanage the movement of data on and off chip. Since DSP assembly code is often seen as intimidating by noninitiates, the availability of straightforward compilers designed to use the underlying hardware most efficiently has made DSP development much more approachable for the vast pool of C programmers.

These changes have initiated a shift in DSP system development from hardware to software, a trend that will continue as DSP performance rises to much higher levels, and software tools become easier to use and familiar to larger numbers of programmers. Developers are finding that they can get more performance out of their systems earlier in the development cycle by using high-level languages than by doggedly handcrafting every routine in assembly to squeeze the last possible drop of performance from the DSP engine. Development time is already more valuable than MIPS, and the ratio is rising (Figure 2).

VLIW architectures have been criticized for enlarging programs by adding parallel instructions, but new DSP designs incorporate features that keep down code size. These features include single-instruction, multiple-data instructions and variable-length instructions that enable multiple instructions to be packed into the same stored word. Like performance, though, memory array sizes continue to increase geometrically, so the issue of code storage space will become less critical over time even though it will always be important.

Scalable increases in performance

VLIW architectures demonstrate that it's possible to continue to increase DSP performance by adding more multiply-accumulate data paths. Essentially, VLIW parallelism builds on the two structures—multiply-accumulates and multiple buses—that distinguished DSPs from other microprocessors from the very beginning. As long as the memory subsystem is designed to keep up with the core in throughput, and as long as the compiler is sophisticated enough to handle the complexities of a massively parallel pipeline efficiently, architects can keep adding extra multiply-accumulates and supporting buses to increase performance. Although core designs are far too complex to append data paths as merely modular additions, the overall effect is similar to just snapping on more pieces. Future DSP architectures will make use of this scalability as a straightforward approach to increasing performance.

Experts like to speculate about what new structure will introduce a performance boost comparable to the one provided by multiply-accumulates and multiple buses twenty years ago. Right now, though, there's no new, alto-

gether different architecture based on a new or rediscovered logic structure that suggests itself as the source of the next processing revolution. So added parallelism, with corresponding modifications in memory and code, will continue to be the main architectural technique to increase performance for some time to come.

Process advances

All of these architectural innovations indicate that DSPs are becoming more differentiated as the technology matures and new application areas are discovered. These considerations bring us back to my earlier question: Why have DSPs done so well in recent years? One part of the answer is that somewhere in the late 1980s, IC technology began to catch up with the potential offered by DSP architectures, just as it had begun to catch up with the potential of other types of processors a few years earlier.

Some numbers are revealing here. In 1982, a 50,000-transistor DSP offered 5 MIPS for $150 and consumed 150 milliwatts (mW) per MIPS. A decade later, a 500,000-transistor DSP capable of 40 MIPS operated on just 12.5 mW/MIPS and cost $15 (Table 1). These numbers show that, in the 1990s, DSPs were entering the realm of price, performance, and power consumption making them appropriate for high-volume applications. At the same time, markets appeared that demanded high signal-processing performance to open up more wireless channels, speed Internet delivery, and perform other needed services. It was a classic instance of the right technology arriving at the right time for the right applications.

Table 1. Two decades of DSP market integration (typical DSP figures).

	1982	1992	2002
Die size (mm)	50	50	50
Technology size (microns)	3	0.8	0.18
MIPS	5	40	5,000
MHz	20	80	500
RAM (words)	144	1,000	16,000
ROM (words)	1,500	4,000	64,000
Price (dollars)	150	15	1.50
Power dissipation (mW/MIPS)	150	12.5	0.1
Transistors	50,000	500,000	5 million
Wafer size (inches/mm)	3 / 75	6 / 150	12 / 300

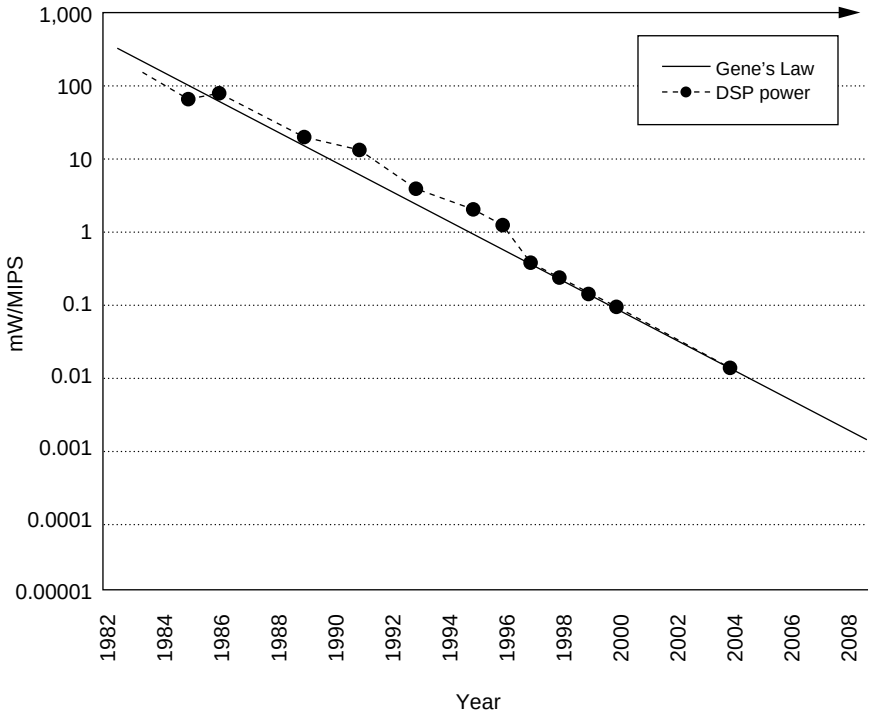


Figure 3. Power dissipation trends. The Gene's Law (named by the author) trendline follows that of Moore's Law in that DSP power dissipation per MIPS halves every 18 months.

Obviously, these trends are continuing. Current projections by Texas Instruments are that by 2002, a 5-million-transistor DSP that provides 5,000 MIPS will be priced at just $1.50 and will consume 0.1 mW/MIPS. Ten years later, a DSP with 50 million transistors capable of achieving 50,000 MIPS will cost just 15 cents and run on 1 nanowatt (nW) per MIPS (Figure 3). During this time, operating frequencies are predicted to zoom to more than 10 gigahertz. These figures seem incredible, even in an industry accustomed to breathtakingly rapid changes.

Table 2. DSP integration through the years (typical device capabilities).

	1980	1990	2000	2010
Die size (mm)	50	50	50	5
Technology (micrometers)	3	0.8	0.1	0.02
MIPS	5	40	5,000	50,000
MHz	20	80	1,000	10,000
RAM (bytes)	256	2,000	32,000	1,000,000
Price (dollars)	150	15	5	0.15
Power (mW/MIPS)	250	12.5	0.1	0.001
Transistors	50,000	500,000	5 million	50 million
Wafer size (inches)	3	6	12	12

Process challenges

In some ways, of course, these figures are indeed incredible. Road maps don't indicate the sweat, and sometimes panic, involved in going from one technology node to the next. Today, advanced DSP cores are manufactured with 0.15-micron transistor gate widths, and core operating voltages are at 1.5 V. Soon, gate widths will reach 100 nanometers (or 0.1 microns) and core voltages of around 1 V. According to data from Texas Instruments, road map projections call for gate sizes to diminish to 20 nm in a decade and core operating voltages to 0.2 V (Table 2 and Figure 4).

It's not yet clear how gates smaller than 50 nm will be made, since unwanted electron migrations through barriers at that scale are still a problem. Similarly, there are extremely complex problems to be addressed in the multiple layers of interconnect overlying the silicon. The capacitance and inductance caused by six or seven layers of metal conducting signals at hundreds of megahertz—soon gigahertz—is a big problem. The changeover to copper from aluminum interconnects has bought a few generations of security for on-chip continuity, though even the greater density of copper will not conduct reliably indefinitely as interconnect traces become thinner and thinner. These are only a few of the manufacturing challenges facing DSP suppliers as they look at the generations of technology ahead of them. Yet the physical limit of IC technology has always appeared to be about five years, or two process generations, in front of us. Chip technologists take it on faith that physicists will solve the materials problems by then. So far, their faith has been rewarded.

DSP optimization vectors

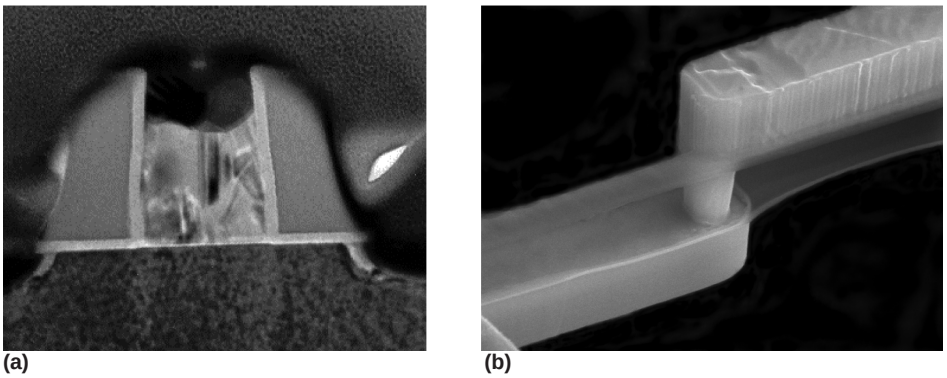
Manufacturing processes indirectly affect us all, but they aren't at the top of the list of concerns that a system developer has in evaluating a DSP for a specific design. While there are many considerations that enter into the evaluation process, the ones that matter the most are the three Ps: price, performance, and power consumption. System developers' requirements force DSP vendors to treat the three Ps as the key vectors of device optimization. Stated a different way, at any given process node, DSP vendors tend to optimize their products for low-cost, high-processing speed, or low-power operation depending on application needs. Taking any one of these vectors to an extreme means some degree of sacrifice from each of the other two.

For example, keeping costs down usually means keeping die sizes small by minimizing functional integration, which in turn tends to slow down throughput and hobble performance. Although a smaller chip may consume less power at a given time, if it takes longer to perform operations, it may consume more power overall than a larger chip. Another way of keeping costs down is by rescaling older, slower DSPs to gain the speed advantage of smaller transistors in a leading-edge process. But since simple rescaling doesn't optimize the design to take advantage of the new process node, performance, though improved, is not maximized, as it would be with a redesigned chip.

Optimizing for performance, power, consumption

The other two vectors, performance and power consumption, are inseparably linked at the transistor level. As CMOS process nodes advance, smaller transistors require less voltage to drive them, which means less power consumption. Lower voltages also tighten the gap between high and low state thresholds, enabling faster transitions that speed up switching and raise overall logic performance. In addition, since more small transistors can be packed in the same space than large ones, there's room on the chip for extra logic functions, larger memory arrays, additional buses, and so on that serve to increase performance.

The fastest transistors must achieve the absolute minimum in transition times between the on and off states. To accomplish this, the



(a) (b)

Figure 4. Silicon technology, the driver behind DSP process advances: a 0.1-micron transistor (a) and a copper interconnect (b).

transistors stay at the verge of entering their transition states. Just as a water faucet would be inclined to leak slightly if it were kept at the verge of opening all the time, the transistors are inclined to leak current slightly. These leakages are negligible in individual transistors, of course, but when multiplied by millions of transistors on a die, they become significant in aggregate. With modifications in the process, the same transistors can be designed to shut tight to conserve power. However, they take slightly longer to build up or bleed off the field-effect capacitance that gates the transistor to its off and on states. Again, these delays associated with capacitance are negligible for individual transitions, but they add up.

Architectural techniques for diminishing chip power consumption include distributing address control for memory accesses throughout the chip, turning core internal functions and peripherals off when they're not in use, and reducing the number of cycles required for data reads and instruction fetches. These and other techniques have shown excellent results for reducing power consumption and are thus extremely useful for battery-operated applications. Texas Instruments' TMS320C55x architecture, for instance, has used all of these techniques to achieve five times the performance per unit of power consumption over its earlier TMS320C54x architecture, which was already extremely power conservative.

Like the architectural functions discussed previously, the three P vectors enable the differentiation of DSPs. Today, DSPs based on leading-edge processes are increasingly designed to meet the requirements of either high-performance systems such as multichannel base stations and line cards, or low-power systems such as wireless phones. Note that these high-performance applications still need to conserve power to whatever extent is possible to pack as many channels into the tightest space that the heat dissipated will allow. Similarly, the low-power applications still require a high level of performance. The important point here is not that either performance or power consumption is maximized, but that the device is optimized for the best trade-off between these vectors for the intended application. The C55x and C64x architectures serve as a good illustration that the same process node can be optimized to achieve either good performance with extremely low power consumption, or reasonable power consumption with outstanding performance, depending on the application requirement.

Greater integration of analog functions

While new processes can be optimized for low cost, the most cost-efficient solutions tend to trail by a process node or two. This is because when cost is the most important design criterion, mature technology usually offers solutions that are already available. Mature process nodes also lend themselves to the integration of analog functionality, such as power CMOS and bipolar drivers, that require extra mask layers in manufacture. Thus, older logic processes remain valuable to system designers because chip vendors increasingly exploit their capability to integrate more types of system functionality onto the same chip.

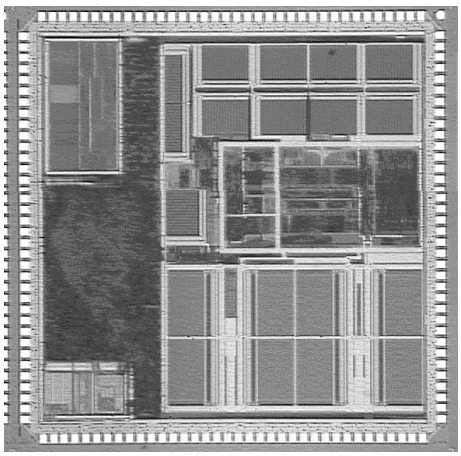


Figure 5. Cellular phone baseband system on a chip featuring a 100- to 200-MHz DSP plus a microcontroller unit, ASIC logic, dense memory, and analog functions.

This helps to reduce component counts, lower costs, and minimize space and weight. As a vector driving product differentiation, cost efficiency serves as both a spur and a limit to the integration of new types of functionality.

To date, analog integration capabilities have been used more in data converters and other supporting chips than in the DSPs themselves. But as future DSP performance outstrips the signal-processing needs of many relatively simple applications, mature CMOS nodes that provide possibilities for greater analog integration along with the processor will become increasingly important, especially for applications where space is at a premium. System-on-a-chip technologies already exist, of course, but in the future these will extend to include analog functions that have been impractical to integrate previously (Figure 5).

Future uses of DSPs

Shrinking process geometries are driving designers relentlessly toward larger, faster DSPs that cost less and consume less power per MIPS. Application requirements are forcing differentiation into architectures optimized for the three P value vectors. Assuming that the industry can achieve its manufacturing goals for the upcoming decade, what kind of applications will be using DSPs?

To begin with, current applications will become increasingly pervasive and gain increased functionality. For instance, hearing aids will appear that not only bring hearing back to normal but also raise it to a better than normal level. A hearing aid might let someone isolate a single conversation with another person in the midst of a crowded, loud room. Digital television will become highly interactive, letting viewers watching a commercial or show click on a product, actor, or whatever and get more detail about that thing or person. Portable electronic equipment will become smaller, lighter, and more personal, letting people hold video conversations and routinely access the Internet from anywhere, among other things.

The most exciting devices, of course, are the ones that haven't been invented yet, or that have been developed for one kind of use but haven't been applied in other areas yet. For example, speech recognition technology might enable handicapped people to drive cars or to operate other machines by voice alone. Powerful processors might be used as medical implants, replacing or supplementing neural tissue while running only on body heat. Since signal processing and control will be inexpensive enough to introduce into just about any kind of equipment, vending machines may do away with buttons and simply accept spoken requests.

A need for ideas

In fact, with performance measured in trillions of instructions per second and miniscule power consumption, all costing just pennies, the biggest future challenge the industry faces may be how to use the potential of DSPs intelligently. In other words, will developers eventually run short of ideas? This may seem a silly question at first glance because developers have always been able to create something new with greater performance. But, assuming that the fundamental technology continues to advance as planned, the problem of where to get new ideas may loom sooner than anyone anticipates.

Consider what it will take to develop software for these new generations of DSPs. Today, integrated development environments, high-level-language support tools, and modular software from third-party vendors make it possible to develop DSP applications much more easily and quickly than ever before. So far, software development has focused on building systems on a task-by-task basis—with detailed analysis of how the code for each algorithm functions

in system operation, and reprogramming as needed. Today's DSPs give designers just enough performance to create a cellular phone or an asymmetric DSL modem comfortably, without very many MIPS left over.

What will happen when the DSP in a wireless handset offers enough performance for twenty cellular phones yet is inexpensive enough and draws little enough power that it is still the best choice for the system? What about a DSP in a refrigerator motor that could control a hundred such motors? Will developers know how to use all that potential, or will they let it go to waste?

New development methodologies needed

Whatever they do in the future, developers won't be writing software in the same way they do now because they won't need to painstakingly rewrite code to shave a few cycles here and there. By then, many of the algorithms that currently require major development efforts will be textbook stuff, as fixed as elementary logic and control structures are today. By then, designers will treat entire systems as the modular building blocks of new megasystems.

To some, a word like "megasystem" might connote a Rube Goldberg type of machine, connecting everything from a cellular phone to an intelligent ice crusher in a single system, and letting users call home from the freeway and have a martini ready by the time they get there. But if the performance will soon exist to accomplish such crazy but useless tasks, what kind of crazy but useful things will it also enable? Whatever the answer, the tools don't exist today that will enable building megasystems, and the engineering methodology isn't in place that will enable developers with visions of those megasystems to make them a reality.

All of this speculation comes back to a single central issue: how much imagination will it take to make use of the DSP processing power we will have available in the next ten to twenty years? Advances in manufacturing processes; architectural developments; software innovations; DSP differentiation for the optimization of performance, price, and power consumption are all factors driving DSPs to become so powerful as to make them ubiquitous in the future. DSPs could provide intelligence for every system that transforms one kind of input to another kind of output. With such processing power ahead of them, what will developers have to do to keep pace? MICRO

References

1. A. Oppenheim and R. Schafer, *Digital Signal Processing*, Prentice Hall, New Jersey, 1975.
2. L. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*, Prentice Hall, New Jersey, 1975.
3. S.S. Magar, E.R. Caudel, and A.W. Leigh, "A Microcomputer with Digital Signal Processing Capability," *Proc. Int'l Solid-State Circuits Conf. (ISSCC)*, IEEE, Piscataway, N.J., 1982, pp. 32-33.
4. P. Lapsley et al., *DSP Processor Fundamentals, Architectures and Features*, Berkeley Design Technology, Berkeley, Calif., 1996.
5. F. Faggin et al., "The History of the 4004," *IEEE Micro*, Vol. 16, No. 6, 1996, pp. 10-20.
6. H.H. Goldstein, *The Computer from Pascal to von Neumann*, Princeton Univ. Press, New Jersey, 1972.
7. K. Lin, G. Frantz, and R. Simar, "The TMS320 Family of Digital Signal Processors," *Proc. IEEE*, Vol. 75, No. 9, 1987, pp. 1,143-1,159.

Gene Frantz is DSP business development manager and a senior fellow at Texas Instruments. He is presently responsible for creating new businesses within Texas Instruments using digital signal processing technology. Frantz received a BSEE from the University of Central Florida, an MSEE from Southern Methodist University, and an MBA from Texas Tech University. Frantz is a senior member of the Institution of Electric and Electronics Engineers.

Readers with question regarding this article can contact Texas Instruments Product Information Center at 12500 TI Boulevard, M/S 8671, Dallas, TX 75243-3500; http://www.ti.com/sc/docs/general/hardsupt.htm.