Design at the system level with VLSI CMOS

by R. F. Sechler
G. F. Grohoski

This paper explores high-performance central processing unit (CPU) design with VLSI CMOS. Workstations are the focus, because they were first to apply the synergism of CMOS, VLSI, and reduced-instruction-set computing (RISC). But the advances of CMOS now encompass all computing system design, and extend to newly created environments. We discuss CMOS extendibility in the highest-performance areas.

Introduction

A workstation is usually thought of as the most powerful office computer, running under the UNIX® operating system. However, its hardware is similar to, and at one extreme priced close to, that of personal computers. At the other extreme, its performance approaches that of mainframes in scientific and large commercial applications. How it achieves high performance has been attributed to the operating system, RISC architecture, system organization, and, ultimately, CMOS circuits.

It is surprising how recently CMOS was not recognized as a major determinant of workstation performance. Possibly its early application to personal computers diminished performance expectations, but by the mid-1980s an IBM Research program had demonstrated that amazingly powerful machines could be designed by optimization in VLSI with CMOS. Within a few years CMOS would unseat bipolar emitter-coupled logic, the undisputed leader of the 30-year history of high-performance computing.

The theme of this paper is that the *foundation* of high performance in workstations is CMOS. To develop this theme, we review its high-performance history in IBM. We begin with an outline of the properties of CMOS which make it inherently and uniquely advantageous in VLSI. How can system design take advantage of these properties? We also review the principles of RISC design and how VLSI and CMOS facilitate their implementation.

We then follow chronologically workstation design in IBM. How were these systems organized to maximize the potential of CMOS technology? Two generations of workstations are described, similarly partitioned with separate cache and processing units, and designed between 1986 and 1992. Following that, we discuss the VLSI design paradigm of the 1990s.

To serve as the highest-performance technology, CMOS must have unlimited extensions in operating frequency. As clock frequencies approach 50 MHz, CMOS-specific interchip circuit design issues appear; these are analyzed in the sections on interchip signal connection and power distribution.

We conclude with a discussion of a few future possibilities. Can system performance enhancements exceeding 50% per year continue? What are the limits on the continuing advance of VLSI CMOS technology?

0018-8646/95/$3.00 © 1995 IBM

The emergence of VLSI, CMOS, and RISC

• *The CMOS–VLSI synergism*

Several reasons have been offered for the preeminence of CMOS in VLSI. Low cost resulting from the compatibility of the semiconductor process with high-volume DRAM is one. Energy efficiency resulting from scaling, the reduction of power supply voltage as semiconductor processes improve, is another. But it is efficient use of operating current that accounts for its original processor applications. We review this property below.

A central problem facing the circuit designer is delay minimization (frequency maximization). For any choice of circuit or design technique, there is a hyperbolic relationship between circuit delay and operating current. (This results from the equations defining inductance and capacitance, such as $i = C \times dV/dT$, thus $i \times dT = C \times dV$.) However, power dissipation is directly proportional to operating current in a circuit that draws current while it is idle (quiescent-current circuit). Therefore, a power/delay hyperbola can be constructed to represent a circuit's minimum-delay capability.

Contrary to its quiescent-current predecessors, a CMOS circuit draws current only while switching. This dynamic current results from the well-known combination of high input impedance and complementary output devices. Several advantages accrue to power-limited design from *two* effects of CMOS dynamic current.

First, in a typical computing system we realize an improvement in operating current efficiency approaching two orders of magnitude. For example, a circuit may have a 10% probability of changing state and a 10% time allocation for doing so within a cycle. Thus, the circuit draws its operating current during that 1% of the total time in which it switches, dramatically reducing power dissipation.

Second, power dissipation is explicitly dependent upon operating frequency, not operating current as in quiescent-current circuits. (Maximum operating frequency is, as for all circuits, proportional to operating current.) As a result, circuit power dissipation is usually expressed by the well-known energy-derived equation, $P = SCFV^2/2$, where S is the circuit-switching probability, C the capacitance charged, F the operating frequency, and V the signal swing.

Designers recognized the advantage of high current efficiency in the mid-80s, as VLSI advanced to the level at which power dissipation seriously affected design. CMOS allowed processor circuit counts to increase to whatever the state of the art could provide.

It appears now that designers did not at first recognize the advantages resulting from the second effect. On the contrary, because of it, there was some concern initially that CMOS would be an interim VLSI technology. Would not its power advantage diminish with frequency, at which point its disadvantages would dominate? This argument can be disproved. The current efficiency of CMOS is frequency-independent, and no one has ever discovered a circuit whose maximum performance is not at least linearly dependent upon its power dissipation.

To illustrate the advantages, hypothesize a quiescent-current circuit processor approaching, by material or topology, CMOS current efficiency. Its *average* circuit operating current would be 1% of CMOS. However, large operating currents are instrumental in providing other desired characteristics. One example is noise tolerance; a second is a less sensitive custom design problem.

Considering noise tolerance, static random access memory (SRAM) is the prime example in which design above minimum current is advantageous for memory cell stability. CMOS SRAM operating currents can be chosen nearly independently of power dissipation. But design above the minimum current supporting the desired performance would produce unacceptably high power in an SRAM drawing quiescent current.

Also, dynamic operating current maximizes voltage noise tolerance. Since a switched CMOS circuit achieves a zero-current rest state, the rest voltages have the maximum levels.

Second, considering custom design sensitivity, power-limited design is a double-ended problem. To maximize frequency, we minimize delay (maximize current), and to minimize power we minimize current (maximize delay). For quiescent-current circuits, power dissipation is directly dependent upon operating current. For CMOS the dependence is the lesser effect of current upon transistor capacitance. Also, in many known areas a relatively low switching factor (S) dilutes the effect of non-minimum currents on CMOS power dissipation. Areas having a high switching factor, such as clock distribution, can be identified for power control. Therefore, because its operating current is dynamic, CMOS has a less sensitive custom design problem at the VLSI level.

Therefore, it is not just energy efficiency that produces the ascendance of CMOS in VLSI, but the manner in which it is achieved. No doubt all CMOS advantages contribute to its success, but the manner in which CMOS uses operating current is the basis for the revolution across the computing spectrum. Low power dissipation is one of several crucial advantages to VLSI design.

• *Processor organization*

We define microarchitecture as the organization of processing units, local memory, and pipeline structure comprising the CPU. Its optimization is influenced by circuit and VLSI chip characteristics. To introduce RISC principles, we first review the trade-offs required for

microarchitecture optimization and how VLSI CMOS applies to them.

We begin with the following axiomatic system performance equation:

Compute time = [path length in number of instructions]

× [cycles per instruction (*CPI*)]

× [cycle time (T_c)].

Overall, the product of the first two terms is most influenced by the instruction set architecture and compiler. The hardware and microarchitecture contribution to performance is determined by the product of the latter two terms, $CPI \times T_c$, where $T_c = 1/F$ (frequency).

To minimize this product, several design approaches might be considered: 1) Signal parallelism, where the shortest path determines the time to compute a result; 2) Logical parallelism, where separate computing resources process independent operations; 3) Pipeline depth, where extended calculations are broken up into multiple cycles in order to preserve minimum cycle time.

How would VLSI CMOS apply to these approaches? CMOS is a technology for which the incremental "cost" of a circuit, in ease of VLSI implementation, is minimized. Thus, past strategies which were limited by circuit counts might be expanded, but strategies which minimize circuit count and play upon individual circuit strengths might be less advantageous in VLSI CMOS.

The first two approaches listed have been used successfully, with the potential for future expansion via multiprocessing. For example, superscalar designs (separate resources for separate functions within a uniprocessor) have developed rapidly during the past few years. Parallelism in signal processing is also widespread. Both approaches tend to be self-limiting by growth in circuit count; that is, circuit counts are increased to improve performance until the marginal cost of an additional circuit exceeds its utility.

The advent of VLSI CMOS creates opportunities for expansion of both logical and signal parallelism in system design.

The third approach uses circuits efficiently, but has had limited past success. Interruptions in program execution increase CPI more significantly for deeper pipelines, vitiating the advantage of shorter cycle times. Thus, superpipelined designs, as they are often called, have had limited applications.

To understand how deepened pipelines might apply to CMOS, we reconsider the basis for CMOS energy efficiency. Computer circuits are actually used (switched) rarely, and CMOS consumes power only during that switching time interval. A design approach which increases the switching activity of individual circuits—as does superpipelining—does not exploit the relative advantage of CMOS.

This argument does not constitute a case against deep pipelines in CMOS. In fact, RISC architecture facilitates pipelined design because of its instruction simplicity. But the characteristics of CMOS do not *enhance* superpipelined design. The choice of pipeline depth involves considerations separate from circuit migration to CMOS.

- *RISC design in CMOS*

The principles comprising RISC are well known. Comprehensive discussions are available in References [1–3]. The effect of the RISC revolution was that powerful CPUs could be designed with a much smaller hardware cost and reduced cycle time compared to otherwise equivalent complex-instruction-set computing (CISC) machines. In the early 1980s, however, CMOS was still in its infancy, and there was no clear understanding of the RISC–CMOS synergism.

IBM studied RISC implementations in several technologies. The first RISC prototypes were built in discrete TTL. Subsequent design studies were undertaken using bipolar emitter-coupled logic (ECL) gate-array technologies used by IBM System/370™ mainframes. Designs using n-MOS were investigated once MOS reached interesting integration and performance levels. Subsequently, CMOS emerged as the choice because of its clear advantage in the emerging era of power-limited VLSI.

ECL RISC and CISC machines of the early 1980s suffered from a packaging mismatch. A typical ECL gate-array chip had approximately 2500 circuits and 90 useful signal I/Os. The partitioning problems at this level of integration failed to demonstrate overwhelming advantages for RISC design concepts.

Partitioning problems were alleviated by emerging VLSI MOS technology. RISC required fewer circuits than CISC machines. By 1982, 8–10-mm MOS chips contained 20–30 000 usable transistors, allowing RISC CPUs to be packaged on one chip. The cycle time advantage of a MOS RISC design over a MOS CISC design became evident. Signal I/O limits became less restrictive, because less communication bandwidth is required at the boundary than within a processor.

Furthermore, the invariably slower cycle time of MOS as opposed to ECL circuitry provided for simpler interchip circuit design. Therefore, with the exception of CPU–cache communication, VLSI MOS technology solved many processor partitioning problems for RISC.

RISC designs still suffered from the limitations of off-chip SRAM cache, more than their CISC counterparts, given identical packaging technology. A RISC machine generally requires higher average instruction fetching bandwidth than a CISC machine of equivalent

Table 1 Comparison of average CISC/RISC memory bandwidth.

	CISC	*RISC*
CPI	3.0	1.2
Instruction length (bytes)	2.5	4
Branch instruction frequency	0.25	0.25
Memory reference instruction frequency	0.3	0.3
Memory reference length (bytes)	4	4
Required memory bandwidth (bytes per cycle)	$(1.25 \times 2.5 + 0.3 \times 4)/3 = 1.4$	$(1.25 \times 4 + 0.3 \times 4)/1.2 = 5.1$

performance. This arises because the RISC machine executes instructions at a higher rate than the CISC machine, and the instruction length of most RISC machines is fixed and larger than most CISC instructions. This can be illustrated by the two hypothetical designs in **Table 1**.

This cursory analysis ignores many factors which, taken together, could double these bandwidth estimates. A CISC machine could achieve its performance goals with a combined instruction/data cache bandwidth of 8 bytes per cycle [4], while a RISC design could not. Doubling the bandwidth of a combined cache to 16 bytes per cycle created other problems, such as address fan-out to a greater number of cache chips. Separating the cache into data and instruction caches was the most effective way for RISC machines to eliminate this bottleneck, but it aggravated packaging limitations.

Also, the normally larger instruction code space occupied by a RISC program compared to its CISC equivalent requires larger instruction caches to hold the same effective program fraction. Finally, realizing the potential for higher operating frequency of a pipelined RISC design meant that the RISC engine was more sensitive to off-chip delays in accessing the cache.

For these reasons, RISC engines benefited from technologies which supported SRAM effectively. As described earlier, CMOS is an ideal SRAM technology because of its ease of design for high current (stability) and low power. SRAM technology is also compatible with CMOS logic. The six-transistor CMOS SRAM cell is essentially a (four-gate) logic circuit. A critical level of integration was soon achieved—say 10 000 logic circuits and a KB of SRAM—where a single-chip RISC processor with on-chip data and instruction caches was feasible. With this configuration, packaging was far less limiting, because on-chip caches reduce interchip bandwidth requirements.

The effect of these considerations is illustrated by the following design points. By 1984, single-chip CMOS RISC CPUs could be designed with 50- to 100-ns cycle times and split on-chip instruction and data caches of 2 KB each. Allowing for a cache miss time of five cycles, this resulted in a native 5–10 MIPS. This was within one generation of ECL gate-array, System/370 mainframe performance at a fraction of the cost.

• *Development of RISC superscalar CPUs*

As the advantages of a RISC design became clear, researchers were motivated to further improve its performance. The ease of adding transistors to VLSI CMOS led to a reexamination of design options first considered in the 1960s [5]. One IBM Research project sought to demonstrate that CPUs which simultaneously issued multiple instructions were practical [6, 7]. Properly engineered, such a "superscalar" machine could compete with single-pipeline vector machines, at a much lower cost. Subsequently, other researchers proposed superscalar designs to speed integer performance [8, 9].

In the early IBM superscalar designs, the primary emphasis was high floating-point performance within a budget of several CMOS chips. By limiting the CPU to a few chips, it was possible to make effective use of signal pin limitations at chip boundaries. **Figure 1** illustrates the RISC System/6000® workstation partition. The main CPU occupied eight chips. The ICU contained the instruction cache (originally 8 KB), and instruction-issuing and branch-processing logic. The FXU contained the integer unit and the data cache directory and controls. The FPU contained a double-precision floating-point unit using a multiply–add pipeline as its dataflow. The SCU contained memory controls. The four-chip DCU contained I/O and memory buffers as well as the data cache.

A superscalar design like this would have been far more costly in mid-1980s ECL technology. This resulted from the technology cost of the required chips, the packaging expense of the TCM (thermal conduction module) [10] technology due to the high power density of the bipolar design, and the additional inefficiency of partitioning a design among several hundred ECL gate-array chips. Studies indicated that ECL RISC engines, even with less instruction-level parallelism, would occupy seven 100-chip TCMs, at a cost of $100 000, compared to a technology cost of perhaps $2000 for the eight CMOS chips. The performance difference between the two designs was less than a factor of three.

Yet the superiority of VLSI CMOS over ECL and n-MOS for superscalar designs was most apparent from microarchitectural requirements. Perhaps no advantage is more important than the massive on-chip wiring bandwidth of VLSI. A superscalar design aggravates the instruction-fetching bandwidth requirements of a RISC design. The IBM RISC System/6000® CPU required a sustained fetch bandwidth of four instructions per clock cycle, regardless of address alignment, to avoid limiting the pipeline performance in floating-point loops. The pin and latency implications of an interleaved off-chip cache would have increased both clock cycle and cycles per instruction.

Another example which demonstrates the advantages of CMOS over quiescent-current circuits is the FPU. Since most general-purpose computer applications use integer data exclusively, the FPU is usually idle. Yet a premium FPU is required to compete with vector machines. VLSI CMOS afforded the system a separate FPU, which provided high performance on demand without burdening all applications with additional static power consumption. Hundreds of thousands of circuits were added to the CPU at a small incremental cost over that of the base integer processing units. This principle applied within the FPU as well. A specialized 116-bit partial adder (the leading-zero anticipator) [11] shortened the multiply–add latency by one cycle relative to a traditional implementation.

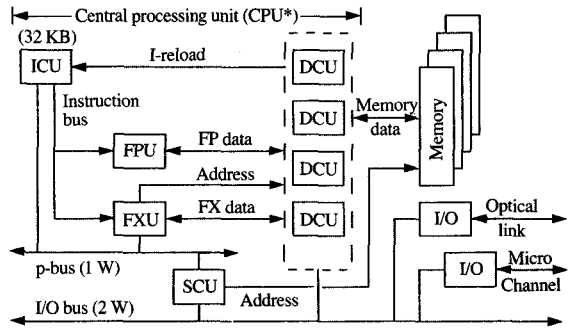
By 1990, the advantages of CMOS for superscalar designs had become widely known. By 1992, nearly every major microprocessor manufacturer had either shipped a superscalar CMOS microprocessor or was developing one [12–14]. Now, owing to the rapid improvements in CMOS, even the lowest-power parts use multiple-integer-unit superscalar designs [15].

Yet for all the RISC–CMOS synergism, integration levels of the late 1980s allowed sophisticated CISC engines to be designed in VLSI CMOS. One IBM study demonstrated that a System/370 CPU comparable to the RISC System/6000 CPU would have occupied an additional VLSI CMOS chip. As late as 1990, however, the industry rumor mill placed bipolar in high-performance workstations rather than CMOS in high-performance mainframes. It was nearly 1992 before the advantages of VLSI CMOS were recognized across the system design community [16].

- *Revolution in perspective*

Why did a revolution which swept the microprocessor industry take so long to migrate throughout the computer industry? One answer is that several technologies converged concurrently. Another is that even by the standards of this industry, this revolution was swift. What might we say in retrospect?

First, CMOS is a uniquely advantageous VLSI circuit. Its full potential was not originally appreciated. Even



POWER (1)	System (generation)	POWER2 (2)
62.5	Maximum frequency (MHz)	70+
256	Maximum chip I/O	512
	Bus widths	
2 W	Inst, I-reload	4 W
2 W	FP data	8 W
1 W	FX data	2 W
4 W	Memory data	8 W
64 KB	DCU	256 KB
1	FX/FP units	2
1–1.5	I/F SPECmark/MHz	2–3.5
	Semiconductor	
←	0.7-μm, Si-gate n-well CMOS →	
	0.45-μm channel, 12 nm gate oxide, 3.6 V	
3	Levels of metal	5
	(2.4, 3.2, $\cdots$ μm pitch)	
Single chip	8-chip CPU modules	Multichip (512 signal)

(*) 8-chip CPU: Instruction cache, floating point, fixed point, storage control, data cache units (4).

Figure 1

POWER/POWER2 chip logic partition.

VLSI CMOS designers investigated alternative high-performance technologies into 1991.

Second, RISC represented a VLSI-dependent advance of computing design concepts. It was relatively easy to implement a new architecture in a new technology, especially given the synergism between them. RISC required fewer circuits and was able to take advantage of VLSI partitions earlier. Later, RISC demanded large local caches for maximum effect, an ideal application for CMOS. Still later, superscalar functions could be incrementally added to RISC processors as VLSI CMOS design technologies advanced. Functions prohibitively costly in power dissipation, and I/O and wiring bandwidth at lower levels of integration, became feasible in advanced VLSI CMOS processes. Finally, until recent CMOS advances, RISC architecture was essential to achieve performance approaching that of advanced bipolar CISC.

Table 2 POWER products performance evolution.

Desk side system	*Date* (M/Y)	*I/F '92**	*Frequency* (MHz)	*I/D-Cache* (KB)	*Semiconductor*	
					V_{DD} (V)	*Gate length* (μm)
1/540	7/90	24/60†	30	8/64	5	1
1/550	4/91	35/84†	41.5	8/64	3.6	0.8
1/560	1/92	44/105	50	8/64	3.6	0.8
1/580	10/92	73/135	62.5	32/64	3.6	0.5
2/590	10/93	117/242	66.7	32/256	3.6	0.5

*SPECint92™/SPECfp92™ (performance relative to VAX™ 11/780).
†Estimated SPECint92/SPECfp92 equivalents derived from SPECmark89™.

By then, many technologies had converged—semiconductor processing, logic and system design, system design verification, high-performance CMOS circuit design—to facilitate advanced VLSI design with CMOS. These differed dramatically from the design technologies known in high-performance bipolar design. By the time migration was necessary, the chasm between technologies had become immense.

Workstation hardware development

Here we demonstrate the influence of VLSI CMOS and RISC design principles on workstation products. POWER, POWER2™, and PowerPC™ designs are well documented [17–19]. Interested readers can obtain detailed information on hardware and architecture.

IBM's line of RISC-based products spans a range from notebooks to supercomputers. Each design is developed to satisfy a specific market demand, and each places unique requirements on the underlying technologies. We focus on the highest-performance products, but it is significant that *all* products utilize VLSI CMOS and RISC. Until a few years ago, high-performance machines were expected to migrate from bipolar to GaAs. Support of multiple technologies would have increased system and chip design complexity enormously.

The flexibility of VLSI CMOS eliminated these concerns.

We review three product generations. The first two, POWER and POWER2, were initiated in 1986 and completed between 1990 and 1994. The third, the PowerPC line of products, first became available in late 1993.

- *POWER products*

POWER systems CPU products are illustrated in Figure 1. Both designs share a similar partition of processing units and cache. POWER2 achieves improved CPI by enhanced instruction processing, multiple execution units with wider buses to support them, and larger data cache and address translation buffers.

The eight-chip CPU of Figure 1 represents our high-performance design point of the technology available in the late 1980s. A lower-cost single-chip POWER architecture machine (not shown) was also developed.

Both product generations employed fairly standard pipeline designs. The pipelines were optimized for commercial and scientific tasks. Although RISC simplifies pipeline design, the expected benefit is shortened cycle time within a fixed pipeline depth. As previously explained, CMOS itself did not suggest any changes.

There is a five-cycle pipeline from instruction address generation and branch unit decode to data retrieval from the DCU. One feature already noted is the FPU data flow. Owing to CMOS efficiency, widely paralleled logic compresses the multiply–add operation to two cycles.

For both systems, high performance was sought by a balanced emphasis on CPI and cycle time (frequency). The basis for reduced early stress on frequency alone was multifaceted:

1. Packaging to support high frequencies was costly in the mid-1980s. However, within IBM VLSI lithography and high-pin-count packaging allowed significant on-chip and interchip parallelism.
2. CMOS levels of integration had not advanced to the point at which performance-limiting interchip communication could be suppressed.
3. System design concepts and design verification capability were sufficiently developed in mainframes that IBM was confident a VLSI superscalar design could be successful.

Additionally, product evolution plans favored equal stress on CPI and operating frequency in early phases of the program. Initial designs without a very high-frequency bias allowed concentration on digital design verification. This was particularly the case due to the high inherent stability (noise tolerance) of CMOS circuits. Thus, a 30-MHz initial design issuing up to four instructions was chosen. Later, with system designs in place, CPU clock frequencies could be increased. Design mapping by technology migration improved performance without extensive redesign. The performance evolution of our workstation is shown in **Table 2**.

Within 30 months, system performance increased by a factor of 3, while frequency increased slightly more than a factor of 2. The higher frequency and cache sizes resulted from simple technology mapping. Part of the increased performance resulted from compiler advances. Still, we managed performance growth exceeding 2× per 18 months by mapping a completed design. The introduction of POWER2 in late 1993 added a substantial boost to performance. However, superscalar design migration played a significant role here. POWER2 design success benefited dramatically from the superscalar simulation and design verification base of the original POWER systems.

The simulation and design verification history of the POWER systems is well documented [18]. Two principles were considered essential to the program schedules. The first was that simulation, design verification, and test methodology be able to ensure first-pass chips adequate to boot the operating system, and to find any latent errors during subsequent testing. Second, to provide fully correct chips prior to first delivery, a level of design-verification testing exceeding prior VLSI simulations by orders of magnitude was needed. The actual achievement exceeded the first IBM workstation (the RT PC®) by a factor approaching a million, with more than one billion verification vectors simulated before final specification of production chips. The success of this program established the base for rapid advances in future VLSI design.

The primary hardware difference between generations is packaging of the eight-chip CPU. The first generation uses pin grid array (PGA) single-chip modules (SCM). POWER2 uses a PGA multilayer ceramic multichip module (MLC MCM) [20].

Each POWER SCM is a metallized ceramic 36-mm substrate; 300 staked pins (44 power/ground) are arranged on a 2.5-mm grid having added interstitial locations. Systems exceeding 50 MHz use an additional substrate plane to distribute a low-inductance ground contact.

The POWER2 MCM has 512 signal pins and 224 power/ground. The 64-mm, nine-chip-site substrate has 624 pins on a 2.5-mm pitch, and an additional 112 on an interstitial 1.25-mm pitch. The module has 44 layers of ceramic signal, power, and ground layers for chip connections. Each chip site has four capacitor sites for power-supply decoupling.

In both systems the printed circuit board (PCB) contains main memory as well as CPU and support parts. The PCB has six layers of signal wiring and four power. Two buried signal layers are used for preferential wiring of asynchronous controls and clock signals. PCB dimensions are 30 by 45 cm. All synchronous CPU signals, including extremely long and heavily loaded lines to main memory, are wired at the CPU frequency.

For example, 4–8-W main memory data are wired across the entire PCB, and each bit accesses up to four memory cards. The result can be 40 cm of wire with four-card accesses of 15–25 pF each. Also, a few paths have significant logic on the source chip, subtracting from interchip delay allocation. These include address and select paths from processor (FXU) to DCU chips. To wire these paths within the machine cycle of 14–16 ns (71.5–62.5 MHz), discretionary wiring rules were established. Specific long "critical paths" were identified and wired with the shortest possible distances. Less critical paths were then wired around the blockages caused by the "critical" paths. **Figure 2** is a diagram of the POWER2 planar.

Each of the three packaging technologies represented mainstream products of the IBM Microelectronics Division. As a cost-driven approach, this corresponds to the use of VLSI CMOS. A packaging technology is chosen from a high-volume product base to control costs; its design is then customized to optimize performance in our application.

- *PowerPC systems*

The PowerPC project was initiated by IBM, Apple, and Motorola in 1991. The subject of interest here is the high-performance CPU design direction taken with integration levels of contemporary VLSI CMOS. All PowerPC products encompass the entire CPU within a single VLSI chip.

Differences between processor performances are determined by operating frequency, CPI, address width (32, 64b), etc. An example of design differences is shown in **Table 3**. The execution unit taxonomy is the number of branch units, the number of load/store units, the number of integer units, and the number of floating-point units. The 601 load/store unit and integer unit are the same.

In each case technology mapping to advanced CMOS processes provides the high-frequency extensions.

The advent of single-chip CPUs appears to alleviate the interchip performance problems seen in POWER systems, which are discussed in depth in succeeding sections. Actually, the problems merely move to another set of networks. In **Figure 3**, we can see as an example three classes of nets for a typically configured high-performance PowerPC system.

First, to satisfy bandwidth demands, an additional level of memory (L_2) is inserted in the system. The existence of on-chip first-level cache (L_1) alleviates, but does not eliminate, interchip performance issues. Although the absolute performance of L_2 data and address may not have to match the processor, processor frequencies are rising rapidly. As a result, interchip performance constraints in advanced PowerPC systems actually exceed those of POWER2 systems.

Also, the synchronous I/O bus introduces new performance issues. This bus accommodates multiple processors and memory, aggravating design problems for high bandwidth.

Nevertheless, two packaging platforms are envisioned for PowerPC products: The first is a carrier for the VLSI chip, fully accommodating its demands in I/O, operating frequency, power dissipation, and power distribution. The second must accommodate the full range of connection types, from PGA to ball grid array (BGA), with area or peripheral array connectors.

With a variety of packaging options and component suppliers, it is desirable to allow independent on- and off-chip frequency optimization. To ensure the highest performance on-chip, counters are used at VLSI chip

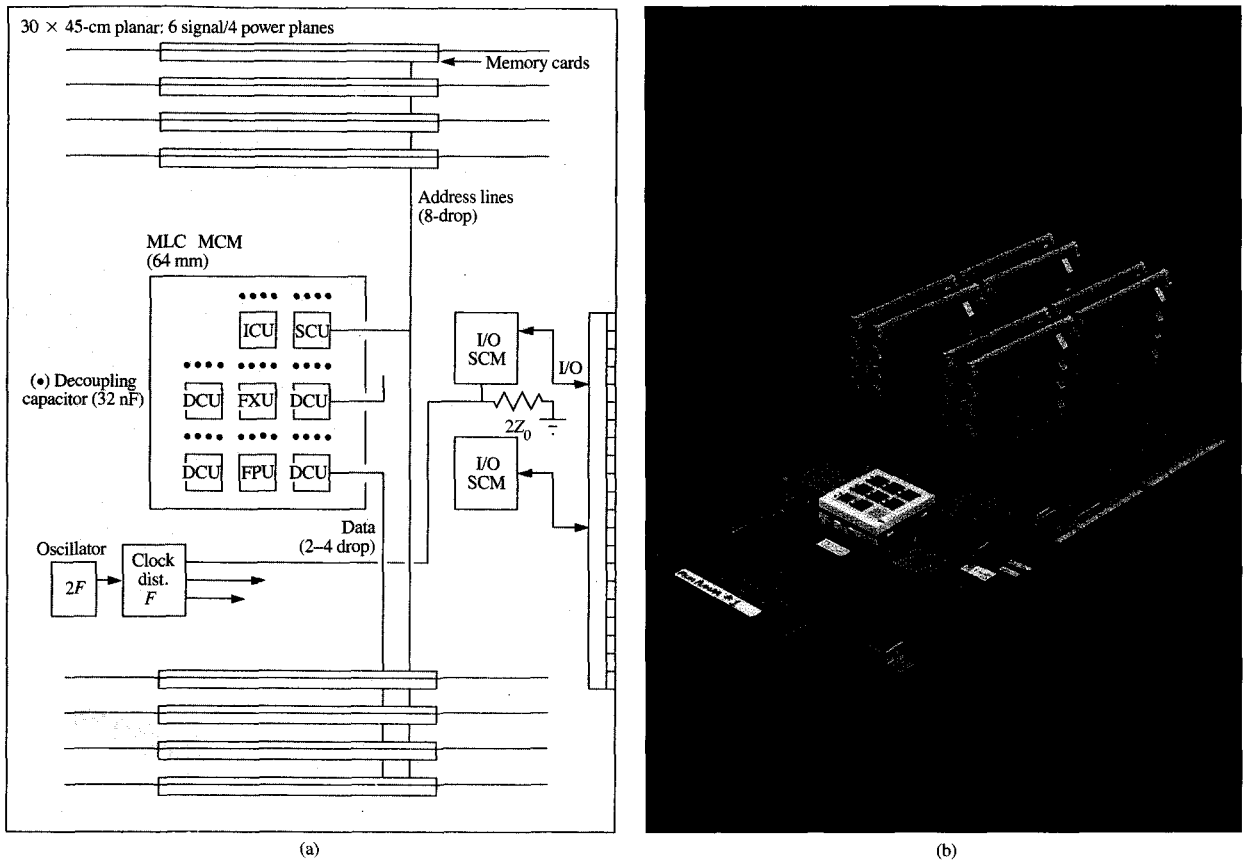


Figure 2

POWER2 CPU package; interchip design to 70+ MHz on planar; MCM design to support 80+ MHz: (a) schematic diagram; (b) photograph of populated planar (with heat sink removed from the MCM).

Table 3 High-performance PowerPC CPU chip sample.

PowerPC chip	*Address width* (bits)	*Execution units* (B/L/X/F)	*Frequency range* (MHz)	*Cache size I/D* (KB)	*Date*
601	32	1/–/1/1	60–120	32	1993
604	32	1/1/3/1	80–140	16/16	1994
620	64	1/1/3/1	120–150	32/32	1995

boundaries to select interchip frequency depending upon configuration and packaging.

Several new low-cost and/or high-performance SCMs are becoming available to support VLSI CPU chips. One is a very low-cost quad flat pack (QFP); another is BGA. All accommodate flip-chip attach, a desirable growth path in the era of high-I/O, high-frequency processors. Ceramic ball grid array (CBGA) offers the prospect of extremely high performance and I/O at competitive costs. High performance is realized through area array connection at chip and module, and power planes to provide low power distribution impedance. This SCM represents the top of the line for high-performance VLSI CMOS processors. In the future, nonceramic BGA topologically similar to CBGA may be available, further reducing costs at the highest performance.

An MCM was employed for the CPU of POWER2 to ensure that cache (L_1) access sustained the highest frequency attained on-chip. To constrain costs and design schedule, the module selected came from an established

high-volume base within IBM. Even so, some higher costs were considered acceptable for the presumed performance. The PowerPC packaging strategy does not exclude MCMs, but we anticipate stricter cost/performance criteria in the era of single-chip CPUs.

Signal interconnection design

VLSI CMOS electrical properties demand special design considerations at the chip interface level. Design problems at this level in POWER systems became apparent at frequencies above 40 MHz (25-ns cycle). (At this point, semiconductor process scaling allowed the power supply to be reduced from 5 to 3.6 volts; we assume the 3.6-V power supply in the following discussion.)

System signals can be divided into three classes: synchronous data, clocks, and asynchronous controls. Their requirements demand separate interconnect designs. Of these, synchronous data are the most influenced by VLSI CMOS; a companion paper [21] addresses their design issues in depth. Here we discuss briefly the POWER and PowerPC systems design approach for synchronous data and clocks.

Any high-frequency long-line signal distribution system must provide a means of controlling reflections on the transmission line [22, 23]. Resistive loss to control reflections can be provided by a source (series) or far-end (parallel) load, as shown in **Figure 4**. The system shown, with a single source, requires a single terminator. Multiple sources generally require multiple terminators: one at each source if series-terminated, and one at each of the two line extremities if parallel-terminated. In either case, series termination has a power dissipation advantage because of its low quiescent current.

It is a widely held view that CMOS circuits are inadequate driving transmission lines at high frequency. The presumed inadequacy is based on experience with parallel-terminated ECL circuits. Comparatively, CMOS demands a large signal swing with resulting high currents, and has a high driver-output impedance. However, these CMOS disadvantages are largely offset by ease of series termination. The complementary transistors of CMOS provide a source impedance that is symmetrical during either transition, and within the range needed for termination. Also, process advances are rapidly reducing any disadvantages in signal swing and transistor impedance. In fact, we will see that reduced currents with 3.6-V V_{DD} do allow limited introduction of parallel termination.

Our view, then, is that CMOS circuits are well suited to transmission-line drivers and may operate at high frequency. Series termination is our design preference because of its ease of implementation and its low power dissipation.

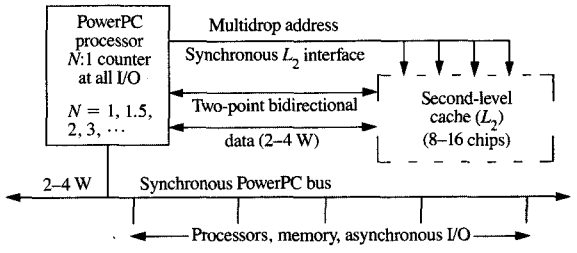
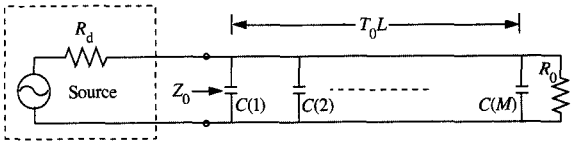


Figure 3

PowerPC processing configuration.



Z_0, T_0: Characteristic impedance, delay
Series-terminated: $R_d \approx Z_0$, $R_0 \gg Z_0$.
Parallel-terminated: $R_d \ll Z_0$, $R_0 \approx Z_0$.

Figure 4

Transmission-line circuit interconnection network.

- *Synchronous signal control*

Synchronous-signal delay and noise control are the most massive problems confronted in interconnect design. Switching must be ensured within delay constraints for the hundreds of paths which cross chip boundaries [24], and transmission-line reflections resulting from reactive and resistive discontinuities must be managed within circuit noise tolerances. The magnitude of these problems has generated a substantial literature devoted to interconnect wiring and delay rules.

To determine whether series-terminated networks are adequate to our problem, we must deal with two issues: the extreme delay we can expect in the system, and the characteristic of the switching signal.

On the first issue, series-terminated networks are generally known to have performance inferior to that of parallel-terminated networks. To illustrate, we compare approximate extreme delays from Figure 4:

Delay ≤ transmission time + reactive time constants

Series-terminated (Source → R_1): $T_d \leq 2T_0L + MZ_0C$, (1a)

Parallel-terminated (Source → R_M): $T_d \leq T_0L + MZ_0C/2$. (1b)

Each of M receiver circuits is approximated as a capacitor, C. The parallel-terminated system provides an initial signal which switches receivers. The series-terminated circuit provides a 50% initial signal, and depends upon the reflected signal to complete switching, hence the common terms "incident" and "reflected" signal switching.

Although the series-terminated system appears to have twice the delay of the parallel-terminated, a significant disadvantage, there are compensating factors. First, in many configurations the source drives receivers at the other extremity of the transmission line. In the limiting case, series termination has the same transmission time as parallel. Also, when series-terminated transmission time actually approaches $2T_0L$, a larger effective switching signal is always present. The initial signal, which generates half the total, appears early, producing an effective time constant well below twice that of parallel termination.

Therefore, astute series-terminated wiring can approach the delay of parallel. The usual approach is to wire nets to maximum symmetry. In effect, the longest path is minimized by wiring all paths to equal delay. POWER and PowerPC synchronous data networks are wired this way wherever possible.

The second issue is the nature of series-terminated switching signals. We can observe from the expected signals in Figure 4 that series-terminated differs from parallel-terminated in one crucial respect. Whereas parallel-terminated can in principle provide monotonic signals, series-terminated cannot. An indeterminate state is inevitable following the incident signal and prior to the primary reflection.

However, unlike clocks or asynchronous controls, synchronously clocked signals need not be monotonic. This property results from the synchronization of data with clock signals. Whereas clock signals are defined at all times, data validity is necessary only for narrow setup and hold timings around the clock transition [25]. Outside this narrow region, synchronous signal levels are irrelevant.

Therefore, reflected signals prior to timing completion can be treated as delay time constants. Only following timing completion do signal level constraints apply. This separation, into pre- and post-switching controls, is a desired design simplification for the large number of synchronous data lines. That the separation is essential to series-terminated design is not a deterrent to its use for synchronous data.

To provide the required delay and noise control, wiring rules and packaging constraints similar to those for parallel-terminated networks [22, 23] are applied. However, the change in timing [from Equation (1b) to (1a)] creates different transmission system constraints. These are covered in depth in a companion paper [21].

• *Clock distribution*

Numerous clock design techniques have been published; for background, we refer the reader to the literature on clock distribution and latching [26]. VLSI high-frequency designs tend to fall into one of two approaches. In the first, a central oscillator generates a high-quality system clock, which is distributed symmetrically throughout the system. In the second, a central clock signal is reconstituted into a system clock within a VLSI chip. POWER products employed the first approach [25], while advanced PowerPC systems plan to use the second.

We now discuss the unique attributes of these designs.

Figure 5 is a diagram of the POWER and POWER2 clock distribution system. Clocks are initiated by an oscillator of twice the system frequency. A central clock chip receives the oscillator signal and divides by two. Identical copies of this signal are delivered to each internal latch within all synchronous VLSI chips. As shown, two cascaded data latches, logically combined into a D-flip-flop, are used as storage elements. (All products utilize level-sensitive scan design, or LSSD, in which L1 and L2 are separately clocked latches for purposes of testability.)

The significance to cycle time of clock signal tolerances can be understood from timing considerations [26], applied to the figure. A positive clock C2 launches data from L2, and after logic evaluation the results are stored upon a negative signal C1 at the following L1. The next positive C2 signal initiates the succeeding cycle. Both separation between signals C1 and C2 and skew between the two C2 signals are included in cycle time. However, some separation may be necessary to protect against fast paths—the insertion of information from machine state $N + 1$ into machine state N. This separation must include the skew between any two identical clock signals. Therefore, a fast-path-protected design may include as much as twice the skew in cycle time.

Several design techniques are combined to minimize POWER system skew: The double-frequency oscillator minimizes mid-cycle skew by basing all transitions on the oscillator frequency stability. Interchip clock signals are isolated in buried tri-plate wiring with limited crossovers, to minimize machine-state-dependent noise coupling. And, within the constraint of equal length wiring, total delay from clock output to latch input is minimized.

Several additional design techniques were applied to minimize interchip skew, shown in Figure 5(b):

- At 50 MHz and above, clock lines are parallel-terminated. The use of external resistors improves the accuracy of the termination resistance, while parallel termination reduces transition by minimizing source impedance. Well-controlled signal quality is attained by the parallel termination of $2Z_0$ combined with a source resistance of $Z_0/2$.
- In the double-latch, separately clocked design shown, cross-chip skew can be removed from the interchip cycle by overlapping clocks. The overlap ensures that the clock signal (C2) is available by the time data arrive at the launch latch.
- Overlapped interchip clocks and skew create a fast-path exposure in interchip paths. The fast path must be circumvented by a delay pad. This pad is placed on the receiving chip, where it has beneficial delay tracking with clock path B. To a first approximation, a fast-path delay pad on the receiving chip adds its pad delay directly to the cycle time. Alternatively, a fast-path pad on the source chip would add twice the pad delay to the cycle, because of process variation between the source chip and the receiving chip.

We have estimated the effect of clock skew on interchip cycle time. We separate it into three terms: off-chip skew, source-to-receiver on-chip skew, and receiver pad delay. These approximately equal terms sum to 1.5 ns, or 11% of a 14-ns POWER2 cycle. Here we assign the entire buffer delay to clock distribution. However, the buffer has wiring advantages. A standardized high-noise-tolerance receiver buffer eases constraints on long-line interconnect wiring.

In evaluating clock distribution approaches for PowerPC systems, two considerations point to a change of strategy: First, rising frequencies lead to difficult and costly problems distributing system clocks among VLSI chips. Second, future products anticipate a variety of separately supplied chips and design techniques sharing synchronous buses, complicating interchip skew minimization. This may be inferred from the differing bus attachments of Figures 1 and 3.

These considerations have led to the popularity of phase-locked-loop (PLL) designs [26], usually in analog circuit form. PLLs compensate for skew between chips by providing signal timing independent of chip design and process variations. They also allow local frequency multiplication from a low-frequency central oscillator. (However, PLLs do not compensate for other tolerances. In particular, off-chip design tolerances are not alleviated, and must be controlled just as for POWER systems.)

Although POWER system clock tolerances compare favorably to other approaches, PLL designs are gaining favor for these reasons. Advanced PowerPC designs take advantage of PLLs to construct the system clock within the VLSI chip.

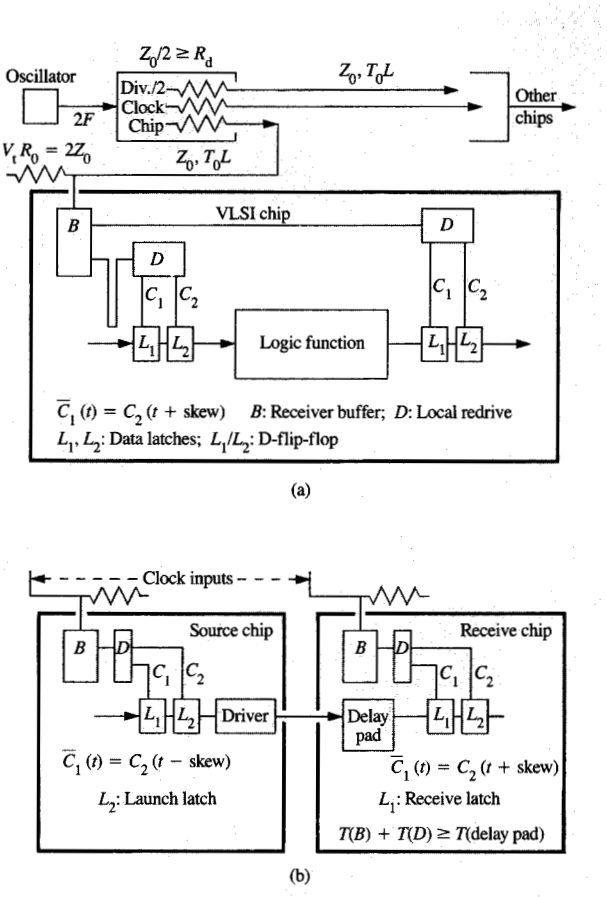


Figure 5

POWER system clock distribution: (a) within VLSI chips; (b) for chip-to-chip paths.

Power distribution

It is no surprise that a technology that handles current uniquely has unique power distribution problems [27]. Almost since its inception, CMOS has been theorized to have problems due to its low quiescent current, producing low-loss (underdamped) networks. Problems from underdamped networks may be manifested in one of two related ways. One is a slowly decaying oscillatory transient from a single input. A second, and potentially more dangerous, is resonance.

Resonance is a well-known phenomenon in nature. A periodic input to a system produces a multicycle response greater than that of the first cycle. Two conditions are essential to its occurrence: One is a low loss in the system; the second is a system natural frequency in close proximity to that of the source.

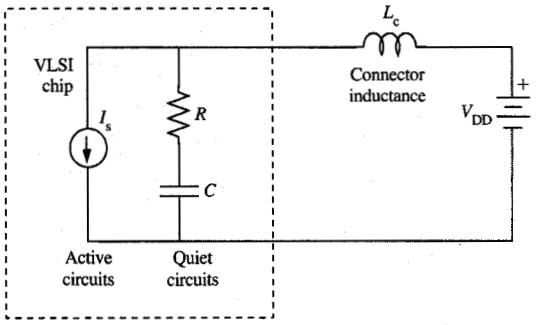


Figure 6

Power distribution.

Another consequence of CMOS properties magnifies power distribution problems. As opposed to the quiescent-current technologies that are being displaced by CMOS, virtually all current in CMOS is switched. If we take advantage of that property and apply power to CMOS circuits at the levels of quiescent-current circuits, total switched current is vastly increased. Any time the system is initiated from an inactive state, almost all current in the system is switched. Thus, in early generations of CMOS where power dissipation was very low, power distribution problems may not have surfaced. However, now that we are designing far higher-frequency, higher-circuit-count chips, power distribution problems assume major proportions.

For either transient or resonant noise, the network to be assessed is similar. We can observe the likelihood of a problem from the simple VLSI CMOS model of **Figure 6**. Switching circuits are represented as a time-varying current source, repeating at a harmonic of the operating frequency. Inactive circuits are a series RC circuit, and the power source is a simple inductance.

The current source has a wide variety of possibilities. A typical signature might be a current step during the first 15–25% of the cycle, followed by a decay toward zero, repeating in succeeding cycles. Also, in some applications it may have a significant repeating portion on the half cycle. During start-up after long quiescent periods, larger current steps may occur. The source therefore has harmonics well above or below the operating frequency, providing a variety of conditions in which resonant or transient problems may occur.

The network of Figure 6 is characterized by a second-order linear differential equation, for which the solution to a current step I_s is

$$V_n = I_s R e^{-bt}[\cos(\omega t) + B\sin(\omega t)], \quad (2)$$

where

$$\omega = [1/(L_c C) - (R/2L_c)^2]^{0.5},$$

$$B = [1/(RC) - R/(2L_c)]/\omega,$$

$$b = R/2L_c .$$

This equation can be simplified by determining the relative values of the internal expressions. To do so we need approximations to the values of R, L_c, and C for chips and modules in our workstations. An example from POWER is an SCM inductance of 0.75 nH, an inherent chip capacitance of 15 nF, and a resistance of the order of 30 milliohms. (Actually, the resistance is derived from the expected time constant of 500 ps.) With those inputs, the critical terms are

$$R = 30 \times 10^{-3}\ \Omega,$$

$$(L_c C)^{0.5} = 3.35 \times 10^{-9}\ \text{s},$$

$$2L_c/R = 50 \times 10^{-9}\ \text{s},$$

$$RC = 0.45 \times 10^{-9}\ \text{s}.$$

From these we can draw several simplifying conclusions. First, the inductive time constant is of little help in decoupling. Second, the natural frequency of the system is approximately equal to the resonant frequency, $1/(L_c C)^{0.5}$. A third conclusion is not immediately clear, but it can be shown that

$$RB \Rightarrow (L_c/C)^{0.5} = 220 \times 10^{-3}\ \Omega \gg R.$$

Therefore, the cosine term can be discarded, and our equation becomes

$$V_n = I_s(L_c/C)^{0.5}(e^{-bt})\sin[t/(L_c C)^{0.5}]. \quad (3)$$

We note that the damping provided by the exponential is small, and whatever damping occurs is enhanced by larger resistance. Capacitance magnitude alone therefore dominates decoupling. Since power supply ringing is inversely proportional to the square root of capacitance, large capacitance magnitudes may be required.

We can apply Equation (3) to explore power distribution transients. Before doing so, however, we take up the second and potentially more dangerous condition of resonance. To assess resonance, we write the network equation of Figure 6 in complex frequency s:

$$Z(s) = sL_c[R + 1/(sC)]/[sL_c + R + 1/(sC)].$$

For a seriously underdamped network, the approximate natural frequency is equal to the resonant frequency, $1/(L_c C)^{0.5}$. Then we can convert the equation into real frequency ($s = j\omega$), and simplify:

$$Z(j\omega) = j\omega L_c[R + 1/(j\omega C)]/[R + j(\omega L_c - 1/\omega C)],$$

$$|Z(j\omega_r)| = [L_c/C + (L_c/RC)^2]^{0.5} \geq L_c/RC. \quad (4)$$

The parameters for resonance can be summarized as follows:

$f_r = \omega_r/(2\pi) = 48$ MHz,

$|Z| = L_c/(RC) = 1.67\ \Omega$,

$|Z(L_c)| = \omega_r L_c = 0.22\ \Omega$ at 50 MHz.

This is a well-tuned circuit having a parallel impedance almost eight times the inductor at the resonant frequency. Since operating frequencies in the 50-MHz range are expected, we could expect serious problems in power distribution.

Comparing impedance magnitudes from Equations (3) and (4), we find an eight-times difference (0.22 compared to 1.67 Ω). Given our estimates of VLSI chip parameters, resonance may be the major problem.

How do we attack this problem? The best approach is lowered power distribution inductance, since that alleviates other problems as well. Unfortunately, lowered inductance has both design and packaging cost implications. The same is true for increased decoupling capacitance on module or chip. Initially, the approach taken was to control the time constant seen at the chip terminals.

At first this appears problematic, since the time constant results from the circuit switching speed. However, the internal time constant of the switching circuit and the external to the supply are partially separated by chip substrate resistance. The substrate is a common voltage contact on-chip which completes the attachment of most switched capacitance to the external supply. To a limited extent, its contacts can be designed to increase the external time constant. In effect, a second, paralleled R/C element is added to Figure 6. At the same total capacitance, a larger average time constant is created for the network.

This was the approach taken in POWER product chip designs. The actual designs realized time constants of the order of 1.5 ns, three times that of the circuits alone. This readily testable parameter can be designed to ensure that resonant peaks are suppressed.

With this design approach, the peak noise reverts to the single-cycle case when machine operation is initiated. The noise calculation is exceedingly difficult because of the intractability of inputs to Equation (3). A calculation of the voltage sag might be

$$V_n = I_s(L_c/C)^{0.5}(e^{-3bt})\sin(\omega t),$$

where the factor 3 represents the increased substrate resistance. Now, let $\sin(\omega t)$ = 1(1/4th cycle), and we can insert these parameter values:

$$V_n = (L_c/C)^{0.5} e^{-3bt} I_s = 0.22 \times 0.73 \times I_s = 0.16 \times I_s .$$

Now, what current, I_s, might be sustained for 5 ns out of 20? Steady-state chip power is of the order of 2.25 W (with V_{DD} = 3.6 V), but this power could peak at 3–3.5 W over a few cycles, yielding an average current of 0.9 A. How much could the current exceed the average for 25% of the cycle? Estimates of initialization of latches suggest up to a factor of 3:

$$V_n/V_{DD} = 0.16 \times 3 \times 0.9/3.6 = 0.12.$$

Not surprisingly, power distribution inductance of 0.75 nH appears excessive by 50 MHz. At frequencies above 50 MHz, we added a ground plane to our pinned ceramic modules, halving module inductance.

It also was observed that not all chip designs are free of resonant noise. The PowerPC 601™ chip has more ringing at 50 MHz than at 60, a clear indication of system resonance. However, total noise here was within our design guidelines.

What does the future hold? With rising integration levels and frequency, power distribution becomes one of the most complex problems confronting high-performance design. Increased decoupling capacitance will be required at all packaging levels, *including* the VLSI chip. And increased emphasis on package design to reduce inductance will be essential also. CMOS circuits offer immense advantages to VLSI design, but power distribution clearly is not among them.

Table 4 VLSI CMOS semiconductor process evolution.

	1992	*1995*	*1998*	*2001*
Memory	16Mb	64Mb	256Mb	1Gb
Minimum photolith/feature (μm)	0.7/0.5	0.5/0.35	0.35/0.25	0.25/0.18
Maximum chip size (mm^2)	200	300	400	
Power supply (V)	5–3.3	3.3–2.5	2.5	1.8
Performance (Integer SPECmarks)	62	250	1000	

The age of CMOS

Within the past few years, most have come to agree that the age of CMOS will extend well into the next decade. Projections of semiconductor process development driving continued system advances follow a trajectory typically like that shown in **Table 4**.

The performance projection is a generally accepted standard of four times per three-year period. [The performance growth, loosely $1/(CPI \times T_c)$, excludes architecture-independent compiler enhancements.] Can that growth be maintained? If so, how are problems with power dissipation, and power and signal distribution controlled?

Also, what has fueled the performance quadrupling per generation? Transistor performance typically improves

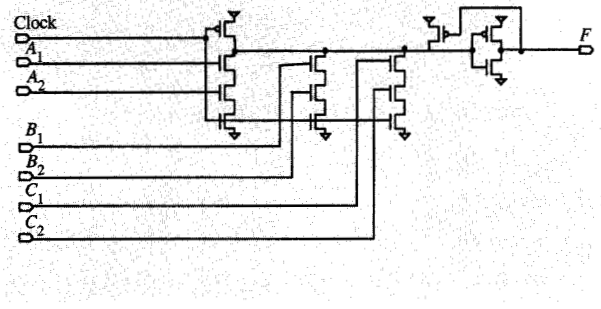


Figure 7

Dynamic 2 × 3 AND–OR gate.

30–40%, and wiring capacitance is reduced approximately by the reduction in feature dimension. The result is significantly less than a factor of 2 reduction in circuit delay. Thus, well over half the growth rate comes from terms not explicitly accounted for above. These include enhanced circuit and chip design, and microarchitectural improvements.

As an example of the problems resulting from geometric growth, we consider power dissipation. To a first approximation, CMOS power dissipation correlates directly to performance. (The usual term is *frequency*, but *instruction rate* is more inclusive.) The original energy-derived equation can be rewritten $P = SCIV^2$, for normalizing constant S, capacitance C, performance I, and power supply V.

We expect capacitance to decrease, at most, proportionately to the feature dimension. Across two generations, V_{DD} and C decrease by a factor of 2, yet power dissipation still increases by a factor of 2. (A more likely prospect is a voltage decrease by a factor of 2 across four generations.) This power may be contained in as few as a quarter of the chips. Since power dissipation problems were not trivial in 1992, we have a serious problem in our future. Similar projections apply to power and signal distribution, across VLSI chips and across the system.

We take up future design and performance issues in the following sections.

• *On-chip circuit performance*

The final frontier for VLSI CMOS is the highest-performance machines, until recently the province of bipolar ECL. It crosses that frontier not due to performance but due to cost/performance—the design point beyond which higher performance is not cost-justified.

In fact, CMOS enters the highest-performance realm with serious circuit disadvantages. For example, MOS transistor current follows a square law relationship, bipolar transistor current an exponential one. The sharper bipolar current produces faster switching. Also, bipolar transistor logic circuits are based on paralleled transistors, CMOS on paralleled *and* cascaded transistors. Finally, CMOS uses p-MOS/n-MOS transistor pairs. The current-to-capacitance ratio, the basis of MOS switching speed, is for p-MOS less than one half that of n-MOS.

The deficiencies of CMOS have spurred a growth industry in circuit creation. The product of that industry is precharged circuits; one is shown in **Figure 7**. (This circuit has a clocked restore signal, but ultimately may need self-timing so as not to interfere with path delay.) We call them dynamic as opposed to static CMOS, because of their dynamically defined voltage levels. Their primary concept is simple: First, they reduce the number of p-MOS transistors in the logic path. Second, to the maximum extent they parallel rather than cascade transistors to perform logic operations. For example, a static CMOS implementation of the AND–OR of Figure 7 would require two stacks of three cascaded p-MOS transistors above the n-MOS logic.

How does dynamic CMOS measure up to the advantages on which static CMOS has swept the industry? First, the basis for static CMOS VLSI design leverage, dynamic switching current, is maintained in dynamic circuits. However, the magnitude of the static CMOS power dissipation advantage is seriously compromised. To understand this, we reconsider actual CMOS current efficiency.

We have estimated a static CMOS current efficiency 100 times that of quiescent-current circuits. This efficiency results from the probability of a circuit switching multiplied by the fraction of a cycle allocated to do so. At a fixed delay, one circuit has the same time to switch as another. The probability of a circuit switching is actually the product of two terms: One is the probability that the next machine state is a new state; the other is the probability that the new state has a logic level differing from the previous state.

For power dissipation calculations, taken over millions of cycles, the latter probability cannot exceed 50%. (For data it is 50% maximum, for controls considerably less.) The former probability results from the efficiency with which a machine resource is utilized. This varies considerably among resources, but on average seldom exceeds 33%. The combined probability is usually of the order of 10–15%.

Considering dynamic CMOS, the latter probability is identical to static. However, when a dynamic circuit switches, it does so twice. Also, if unused resources are not disabled, the former probability is 100% for dynamic circuits. This is simple in concept. Since the previous state

is preset by the circuit and not the machine, the next machine state is always a new state.

Therefore, dynamic CMOS may have more than five times the switching rate of static. The power increase is lower because of the lower capacitance of circuits with few p-MOS transistors. Also, dynamic CMOS power can be reduced by disabling circuits not in use. But we cannot realize a factor of 5 in savings.

What do these arguments suggest as to the future of dynamic CMOS? A design divergence is likely, in which the office environment will have few of the advanced circuits, while the highest-performance arena will see high utilization. Our latter judgment is based upon the two considerations on which CMOS now dominates VLSI:

- Dynamic CMOS maintains dynamic switching currents, that essential basis for ease of VLSI design.
- Dynamic CMOS uses current inefficiently compared to static, but is far superior to its quiescent-current predecessors. Even a CMOS circuit accessed every cycle draws current during only a small fraction of total time.

The foregoing argument suggests that precharge be used on demand, when no other solution suffices. And wherever power/performance is key, dynamic CMOS may have limited application. However, to supplant ECL in high-performance applications, dynamic CMOS offers the right balance between performance and feasibility of VLSI design.

- *Cross-chip performance*

Our second subject for consideration is signal delay management across large VLSI chips. Designers of high-performance CMOS are well aware of problems of interconnect transmission lines on-chip, approximated as R/C transmission lines.With resistance R_s and capacitance C_s per unit length, a line of length L has a time constant at its far end, $T_s = R_s C_s L^2/2$. We approximate delay as T_s. Clearly, controls are needed for long lines. How will cross-chip delay be affected by the technology improvements above?

The generally accepted position is that R/C delay is constant with technology advances, because R_s increases linearly with density, C_s holds constant, and L decreases with the square root. Therefore, to traverse a constant logical boundary, each generation sees the same R/C delay. However, to support the expected 33–40% improved transistor performance per generation, R/C delay should improve at least equally across a constant logical boundary. To do so, approximately the same delay per unit of physical length must be maintained across generations. Several proposals address the problem.

The first is to adjust the process to provide transmission approaching L/C lines. In concept, L/C lines, at 70 ps/cm delay, may approach ten times the performance of R/C. This can be accomplished within current materials by use of thick and wide wiring layers [28], with some impact on wirability and capacitance.

A second is improved materials, lower-dielectric-constant insulators, and lower-resistance metal.

At present, designers are concentrating on astute physical design to maintain constant R/C delays per unit length. By the use of repeater buffers for long-line cross-chip signals, the square law delay above can be linearized. The resulting uniform delay is dependent as much on the buffers as on R/C. Thus, the overall delay improves slightly between generations. Some advantages are still possible from the semiconductor process. Maintaining constant dielectric thickness allows for marginal reductions in capacitance across generations. And wiring critical paths above minimum pitch improves performance by lowering line-to-line coupling capacitance. In fact, all of these achieve improved performance by lowered capacitance, an important direction in future power-limited designs.

Whatever the approach, a successful design will provide cross-logic transmission delay matching the improvement in transistor performance. Even then, designers must manage system microarchitecture on-chip, as they did between chips in earlier generations. Reduced logic and added cycles of latency will be essential to cross-chip communication in the technologies beyond 1995.

- *Interchip circuits*

In our previous discussion, we took issue with a widely held industry view that CMOS circuits are inadequate for high-frequency interchip communication. We then proposed a series-terminated CMOS implementation, as opposed to higher-performance parallel termination. How do we reconcile these views?

The case for standard CMOS interconnect circuits has several bases. First, the standard circuit is always easier to implement, and therefore incurs the lowest cost. Second, series termination has obvious cost/performance advantages, which are the usual selection criteria. Additionally, though, CMOS circuits have a property which offers unique capability for providing high performance at low cost. This property, its symmetrical output impedance, allows for either series or parallel termination with moderate changes to the driver.

We introduced this property previously when we implemented a partially parallel-terminated ($2Z_0$) clock driver with reduced but symmetrical source impedance. Also, with a modest reduction of maximum source resistance, we could design for impedance matching. What other circuit, with minor modifications to the source, can be converted from series to parallel termination? Such an option would be far less effective, even when possible, in

nonsymmetrical ECL, or open-drain and push-pull n-MOS circuits.

In the present era of 3.3-V CMOS, relatively large switching currents naturally restrict the number of parallel-terminated networks, but in the coming era of lower voltages, beginning with 2.5 V, these restrictions will largely disappear. At that time, series or parallel termination will be a readily implemented CMOS interconnection.

Will system performance growth be enhanced by a change away from CMOS interconnect circuits? There is no certainty, but in our judgment it is unlikely [21].

• *Instruction set architecture and microarchitecture*

Superscalar design concepts may be approaching the limit of their potential. The controls required to manage further enhancements will increase latency, reducing the performance advantage of the additional units. In the near term, deeper pipelines may aid performance, if they facilitate continuing growth in frequency. Eventually, however, approaches making use of functional growth will be needed if performance trends are to continue. We describe a few possibilities below.

Current superscalar CPUs extract most of the instruction-level parallelism available given current compiler technology. An average performance increase of perhaps 10% would be gained by adding a third integer unit to POWER2 [29], not 50% as one might expect. Even this small gain has large hardware costs: additional register file ports, another result bus, more instruction-issuing bandwidth, additional instruction control and state repair circuits, and increased instruction and data buffering. The likely effects would be additional logic levels in control paths, increased gate load on critical paths, and increased wire length, exacerbating *RC* delays. Two factors militate against increasing the degree of superscalarity. The first is the nearly unmanageable complexity of current superscalar designs. The second is a likely reduction in operating frequency.

New architectures will likely be required to maintain recent gains in CPU performance. VLIW [30] and multiscalar [31] are two candidates. Both improve the utilization of a large number of functional units by requiring the compiler to extract global parallelism from programs. VLIW binds instruction position to a functional unit, simplifying instruction issue logic. Multiscalar designs distribute control-independent subsections of a program to several similar instruction execution units, each of which might be superscalar. Each promises a large performance increase (more than a factor of 2) over superscalar designs on integer codes. Much of the performance advantage can be exploited with hybrid superscalar designs.

Higher system performance may be achieved by increasing system throughput. Users launch many processes to improve their working efficiency. Providing multiple processors on a desktop allows performance-bound users to achieve higher performance. One can contemplate two to four superscalar CPUs per chip with upcoming VLSI. Current high-end CPUs require of the order of 500 signal I/O pins to communicate with a dedicated L2 cache and a system bus to access memory, graphics adapters, and I/O devices. Depending upon the size of on-chip caches, the utilization of these buses can approach 50%. MP-on-a-chip may be limited by inadequate off-chip communication bandwidth, highlighting the importance of the CMOS interchip communication studied in previous sections.

A barrier to higher performance is the wide mismatch between instruction execution time and memory access time. A 200-ns main memory access time is 67 cycles of the 300-MHz CPU clock; machines executing multiple instructions per clock cycle aggravate this mismatch. There are several approaches to offset ever-increasing memory latencies. The first is to use much of the on-chip transistor budget for a cache hierarchy and spend a relatively small number of transistors on prefetch techniques to reduce the apparent access time of the next memory hierarchy level. This approach works well in the era of power-constrained VLSI CMOS, since cache cell activity is lower than logic gate activity. Additional benefits are increased on-chip decoupling capacitance and a reduction in the area over which clock skew must be tightly controlled.

The second approach blends DRAM and SRAM structures on memory chips to reduce 30–50-ns DRAM access times to a few ns for a large fraction of accesses. This approach still suffers from memory packaging delays.

The third approach is to adopt a multithreaded architecture [32]. Multithreading provides a fraction of multiprocessor performance while also tolerating memory latencies. The primary states of a small number of processes (or threads of a process) are kept in on-chip registers. The registers share a set of execution units and on-chip caches. Multithreading relies upon capturing enough state on-chip to make significant forward progress between off-chip cache misses. Otherwise, activating a new thread can remove other threads' working sets from the on-chip cache, resulting in higher miss rates than a single-threaded design. Thus, it complements designs with on-chip cache hierarchies.

Perspective

Will CPU performance continue to quadruple every technology generation? It is of course impossible to know with certainty. In fact, there is no unanimous agreement on the *history* of performance improvements. At some times, if one includes compiler advances, performance growth has appeared to approach a factor of 8 per

generation. And the contributions of each technology are probably inseparable, therefore forever to be debated.

But the discussion here suggests that certain design paradigms must change to continue on this course. Microarchitecture, on-chip circuit design, and interchip bandwidths will be severely stressed in coming generations. This stress will come in some areas directly, as in the limits of superscalar CPI reduction. In other areas it will come from secondary considerations such as local power dissipation on dense chips.

The recent consensus seems to be that performance growth (separate from shared-memory multiprocessing) will not continue at quite the rate of the past 6–12 years. We might expect instead a factor of 2–3 per generation, still a very aggressive cumulative rate. And architecture-independent compiler advances will probably not add more than another 20% to that. Indeed, architectural and hardware performance improvements are themselves dependent upon significant compiler enhancements.

Still, the industry remains unpredictable. To forecast with no basis a continuation of present growth rates is like predicting the same weather tomorrow as today. That prediction is usually correct, but not when it is most interesting or most important. We cannot say with any certainty what will develop over the next decade, but we do know that the ultimate limits of VLSI CMOS technology are not yet in sight.

References

1. George Radin, "The 801 Minicomputer," *IBM J. Res. Develop.* **27,** 237–246 (May 1983).
2. J. Hennessy, N. Jouppi, F. Baskett, and J. Gill, "MIPS: A VLSI Processor Architecture," *Proceedings of the CMU Conference on VLSI Systems and Computations*, Computer Science Press, Rockville, MD, October 1981.
3. D. Patterson, "Reduced Instruction Set Computers," *Commun. ACM* **28,** No. 1, 8–21 (January 1985).
4. S. Tucker, "The IBM 390 System: An Overview," *IBM Syst. J.* **25,** No. 1, 4–19 (1986).
5. J. Cocke, "The Search for Performance in Scientific Processors" (1987 Turing Award Lecture), *Commun. ACM* **31,** No. 3, 249–253 (March 1988).
6. T. Agerwala and J. Cocke, "High Performance Reduced Instruction Set Processors," *Research Report RC-12434*, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, March 1987.
7. G. F. Grohoski, "Machine Organization of the IBM RISC System/6000 CPU," *IBM J. Res. Develop.* **34,** No. 1, 37–58 (January 1990).
8. W. W. Hwu and Y. N. Patt, "HPSm, a High Performance Restricted Data Flow Architecture Having Minimal Functionality," *Proceedings of the 13th Annual Symposium on Computer Architecture*, 1986, pp. 297–307.
9. M. Johnson, *Superscalar Microprocessor Design*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1990.
10. S. Oktay and H. C. Kammerer, "A Conduction-Cooled Module for High-Performance LSI Devices," *IBM J. Res. Develop.* **26,** No. 1, 55–66 (January 1982).
11. E. Hokenek and R. K. Montoye, "Leading-Zero Anticipator (LZA) in the IBM RISC System/6000 Floating-Point Execution Unit," *IBM J. Res. Develop.* **34,** No. 1, 71–77 (January 1990).
12. G. Blanck and S. Krueger, "The SuperSPARC Microprocessor," *Proceedings of the COMPCON Conference*, February 1992, pp. 136–141.
13. D. Dobberpuhl, R. Witek, R. Allmon, R. Anglin, S. Britton, L. Chao, R. Conrad, D. Dever, B. Gieseke, G. Hoeppner, J. Kowaleski, K. Kuchler, M. Ladd, M. Leary, L. Madden, E. McLellan, D. Meyer, J. Montanaro, D. Priore, V. Rajagopalan, S. Samudrala, and S. Santhanam, "A 200 MHz 64-bit Dual-Issue CMOS Microprocessor," *ISSCC Digest of Technical Papers*, pp. 106–107 (February 1992).
14. K. Diefendorff and M. Allen, "Organization of the 88110 Superscalar RISC Microprocessor," *IEEE Micro* **12,** No. 2, 40–63 (April 1992).
15. B. Burgess, M. Alexander, Y. W. Ho, S. P. Litch, S. Mallick, D. Ogden, S. H. Park, and J. Slaton, "The PowerPC 603 Microprocessor: A High Performance, Low Power, Superscalar RISC Microprocessor," *Proceedings of the COMPCON Conference*, February 1994, pp. 300–306.
16. A. Masaki, "Possibilities of CMOS Mainframe and Its Impact on Technology R&D," *Proceedings of the International Symposium on VLSI Technology*, May 1991, pp. 1–4.
17. *IBM J. Res. Develop.* **38,** No. 5 (1994); topical issue on POWER2 and PowerPC architecture and implementation.
18. *IBM RISC System/6000 Technology*, 1990, Austin Communications Dept.; Order No. SA23-2619; available through IBM branch offices.
19. *POWER2 and PowerPC: Technical Aspects of the New RISC System/6000*, 1994, Austin Communications Dept.; Order No. SA23-2737, available through IBM branch offices.
20. R. F. Sechler, "RISC System/6000 Central Processing Unit (CPU) MCM," *Proceedings of the 1993 International Conference and Exhibition on Multichip Modules*, Denver, April 1993, pp. 22–27.
21. R. F. Sechler, "Interconnect Design with VLSI CMOS," *IBM J. Res. Develop.* **39,** No. 1/2, 23–31 (1995, this issue).
22. R. F. Sechler, A. R. Strube, and J. R. Turnbull, "ASLT Circuit Design," *IBM J. Res. Develop.* **11,** No. 1, 74–85 (January 1967).
23. E. E. Davidson, "Electrical Design of a High Speed Computer Package," *IBM J. Res. Develop.* **26,** No. 3, 349–361 (May 1982).
24. H. B. Bakoglu, G. F. Grohoski, and R. K. Montoye, "The IBM RISC System/6000 Processor: Hardware Overview," *IBM J. Res. Develop.* **34,** No. 1, 12–22 (January 1990).
25. D. W. Terry, D. E. Stivers, K. W. Pennington, M. W. Riley, and H. C. Nguyen, "Packaging for High Performance," *IBM RISC System/6000 Technology*, pp. 110–113, 1990, Austin Communications Dept.; Order No. SA23-2619; available through IBM branch offices.
26. H. B. Bakoglu, *Circuit, Interconnections, and Packaging for VLSI*, VLSI System Series, Addison-Wesley Publishing Co., Reading, MA, Ch. 8.1–8.6.
27. R. F. Sechler, R. P. Masleid, and B. L. Krauter, "CPU Inter-Chip Communication," *IBM RISC System/6000*

Technology, pp. 106–109, 1990, Austin Communications Dept.; Order No. SA23-2619; available through IBM branch offices.
28. G. A. Sai-Halasz, "Performance Trends in High-End Processors," *Proc. IEEE* **83,** January 1995.
29. J. Barreh, S. Dhawan, T. Hicks, and D. Shippy, "The POWER2 Processor," *Proceedings of the COMPCON Conference*, February 1994, pp. 389–398.
30. J. A. Fisher, "Very Long Instruction Word Architectures and the ELI-512," *Proceedings of the 10th Annual Symposium on Computer Architecture*, June 1983, pp. 140–150.
31. M. Franklin and G. Sohi, "The Expandable Split-Window Paradigm for Exploiting Fine-Grain Parallelism," *Proceedings of 19th Annual International Symposium on Computer Architecture*, 1992, pp. 58–67.
32. A. Agarwal, J. Kubiatowicz, D. Kranz, B. H. Lim, D. Yeung, G. D'Souza, and M. Parkin, "Sparcle: An Evolutionary Design for Large Scale Multiprocessors," *IEEE Micro* **13,** No. 3, 48–61 (June 1993).

Received May 24, 1994; accepted for publication October 20, 1994

Robert F. Sechler *IBM Systems Technology and Architecture Division, 11400 Burnet Road, Austin, Texas 78758 (RSECHLER at AUSVM6).* Mr. Sechler received his B.S. degree in engineering physics from Lehigh University in 1961, and his M.S. degree in engineering from the University of Vermont in 1983. In 1961 he joined IBM in the newly formed Components Division in East Fishkill, New York. During the succeeding 16 years he worked on a variety of assignments in bipolar logic circuit development. From 1964 to 1966 he was assigned to circuit design for the high-speed ECL circuits used in the IBM System/360™ Model 91. From 1966 to 1971 he was a manager responsible for design of the ECL circuits used in System/370 and advanced scientific machines; he later managed design for bipolar LSI circuits used in the IBM 4341 and 3081 processors. In 1977 he transferred to Austin to work on the small-machine semiconductor applications. Since 1983 he has been a senior technical staff member working on the implementation of high-performance technologies in advanced workstations. Mr. Sechler has several filed or issued patents and publications on computer logic circuits. He co-authored a paper on System/360 Model 91 logic circuit design, and has published recent work as part of the IBM RISC System/6000 workstation announcements in Austin.

Gregory F. Grohoski *IBM Systems Technology and Architecture Division, 11400 Burnet Road, Austin, Texas 78758 (GROHOSKI at AUSVM6).* Mr. Grohoski received a B.S. with distinction in electrical engineering from Cornell University in 1980 and an M.S.E.E. from the University of Illinois at Urbana–Champaign in 1981. He joined IBM at the Thomas J. Watson Research Center, Yorktown Heights, New York, in 1981. During the succeeding five years he worked on advanced high-performance computer architecture and hardware, including RISC processor design and superscalar research projects. In 1986 he transferred to Austin to join the RISC System/6000 design team. Since 1990 he has worked on a variety of workstation hardware and architecture projects. He is currently employed as a senior engineer at the Somerset Design Center in Austin, developing a future PowerPC processor. Mr. Grohoski's work has been published as part of the announcement of RISC System/6000 products. In particular, he authored two papers in the January 1990 issue of the *IBM Journal of Research and Development* devoted to the subject of processor design. Also in 1990, he received an IBM Corporate Award for his contributions to RISC System/6000 architecture. He holds six U.S. patents and has achieved three IBM Invention Plateaus.