Preface

Scope of the Book

This book presents a detailed description of the microarchitecture of a high-performance microprocessor and discusses the platform and systems issues related to designing and implementing microprocessor-based systems. We believe it is unique in its content and its approach. This is easily seen when contrasting it with other computer architecture books. We chose our particular content and approach for three important reasons:

1. *to appeal to three diverse audiences*—practitioners in the field, ranging from microarchitects to platform designers and implementers; university faculty and students in Computer Science, Computer Engineering, and Electrical Engineering Departments, as well as those in other technical and science disciplines; and technical and project management.
2. *to give an integrated treatment of microarchitecture, platform, and systems issues*. There are substantial sections of the book dealing with topics that surround the design, implementation, and use of a commercially deployed microprocessor. These issues are often ignored or only treated lightly in contemporary computer architecture books. For example, motherboard, chipset and BIOS[i] issues, design and implementation implications for desktop, entertainment, and mobile platforms, and a number of the VLSI and fabrication process challenges—all of which are fundamentally important today—are discussed in this book.
3. *to provide a valuable reference book*. There is a wealth of material on the CD-ROM that accompanies this book. It ranges from the complete text of tutorial and survey articles from professional society periodicals and conference proceedings to important industry standards and specifications. This and the other material on the CD-ROM, which will be described later, will prove a

[i] The BIOS, Basic Input/Output System, is low-level software for providing fundamental I/O service routines callable by other system software and low-level applications. After a system has been successfully powered up, the BIOS is always resident in memory.

valuable addition to your personal library. In the university context, the material on the CD-ROM can provide fertile ground for both assignments and projects. If the book is used by the practitioner or technical manager, this material can be used as a basis for self-study.

One of the themes discussed in several chapters is to examine how the ultimate uses of a microprocessor impact its design and implementation. In doing so, fundamental guidelines emerge that transcend the design, implementation, and use of a specific microprocessor. Thus, this book is worthy of study by those involved in computer architecture and computer systems-related research, development, implementation, or use.

Studying Real Machines

Much can be learned from studying the design and implementation of real machines. Consider the following quote from Harold Stone's book[ii]:

> *"One of the disappointing aspects of the existing literature in high-performance systems is that relatively few machines have been thoroughly documented in the published literature. Thornton's [1970] analysis of the CDC-6600 is a notable exception, and there are a few others, such as Organick [1972] on the Multics system and Organick [1973] on the Burroughs' B5700 and B6700 machines.*
>
> *High-speed implementations have grown far more complex since these books appeared, and there are many opportunities to increase that complexity to achieve greater performance by steadily increasing the average number of instructions completed per clock cycle. The literature of the 1970s and 1980s does not adequately reflect the actual state of the art in machine design, because in this field many advances are realized in physical machines and in the hands of users before the research and academic community have the opportunity to study them."*

Our book, in part, helps fill this gap. We use AMD's K6 3D microprocessor as the "case study" basis for discussing microarchitectural, platform, and systems issues. Each of our three target audiences can benefit from this approach:

1. *practitioners* can benefit from this book because of its detailed description of how a contemporary high-performance microprocessor works and of how to incorporate one into a number of different platforms.
2. *university faculty and students* can benefit from its treatment of

---

[ii] Harold S. Stone, *High-Performance Computer Architecture*, 3rd Edition, Addison-Wesley Publishing Company, 1993.

important technology issues (e.g., VLSI design and fabrication process-related issues) and the in-depth discussion of the system and platform issues involved in using microprocessors (e.g., bus structures, core logic chipset, and BIOS-related issues). Such technology, system, and platform issues are often lacking in contemporary texts on computer architecture.

3. *technical and product managers* can benefit from this book by understanding the evolving nature of microprocessors in general and what skill base is required to integrate them in various system platforms (e.g., 3D graphics platforms).

The book is divided into six main chapters:

1. an overview of issues that arise in the design and implementation of microprocessors and microprocessor-based systems.
2. a microarchitecture case study.
3. the K6 3D microarchitecture.
4. technology components of platform architectures.
5. platform memory technology.
6. platform optimization techniques and directions.

In general, each chapter starts with an overview meant for all three audiences. The discussion then proceeds into detail meant for both the practitioner and the university audiences. Various sections of the each chapter that we believe might be of specific interest to each of the audiences are identified in the chapter *road maps* which are introduced at the beginning of each chapter. The road maps help the reader pick out sections to read in-depth as well as sections to bypass. The six chapters fall roughly into two sections. The first section deals with microarchitecture and the second section deals with platform and systems issues. We will now examine each of these sections.

Introductory Notions: The Tools of the Trade

The basic underlying engineering approach to designing, implementing, testing, and fabricating processors—and integrating them into various platforms and systems—has not changed substantially since the computer industry began to take form in the mid-1960s. Computer architects, computer engineers, and platform designers employ now, as they did then, a variety of tools:

1. iterative design techniques based on performance and cost analyses and trade-offs. The trade-offs occur along a number of dimensions ranging from the complexity, engineering resource requirements, and impact on the software a particular solution

has, to the amount of area, power, and pins it requires.

2. simulators, emulators, prototyping systems, and models and related analysis tools used, first, to verify behavioral and functional specifications, compatibility requirements, and adherence to standards, and, second, to explore and analyze alternative solutions, such as impact of the static or dynamic binding of a specific aspect of the processor. Trace-driven detailed performance models have become an increasingly important component of tools in this category.
3. measurement tools to collect data gathered from the simulators, emulators, prototype systems or models to determine what systems actually do when executing either real or synthetic programs, (e.g., data from address and instruction traces that yields information about the frequency of occurrence of branches and other types of instructions, address modes, exception conditions, and interrupts).
4. formal analysis methods that attempt to predict one or more aspects of a specific system component or of overall system behavior such as performance, critical speed paths, latency, hit-rates and line-sizes, bandwidth requirements, or availability.

The disciplines of computer architecture and platform design both involve balancing a large collection of highly dependent trade-offs.

Indeed, the discipline of computer architecture is one of balancing a large collection of often highly dependent trade-offs. The same statement is equally true of platform design and implementation. What has changed in the intervening years are the tools, the design solutions, and the underlying semiconductor manufacturing process. The dramatic densities achieved by advances in the semiconductor manufacturing process continue seemingly unabated, as do their impact. Additionally, the range and quality of tools that computer architects, computer engineers, and platform designers have at their disposal for undertaking the above tasks have also improved substantially.

One can credibly argue that without the improvements in the tools, the industry could not have taken advantage of the advances in semiconductor technology. Moreover, the range of solutions to resolve architectural performance problems and provide increased functionality has also increased significantly—both in the form of enhancements or modifications of older solutions as well as the development of entirely new approaches. This observation is true not only because of the options created by advances in technology but also because of the options that emerge based on what we learn by studying existing processors and platforms. As with most disciplines, we advance by learning from and building upon past results.

Technology and Applications

technology pull, application push

The reason processor and system platform architectures continue to rapidly evolve stems from the dual impact of technology and applications:

1. *technology pull*—the relentlessly increasing density of circuits that can be included on a chip and the concurrent reductions in the costs of the chips forces us to continually rethink how they can be exploited to achieve significant increases in functionality and performance at substantially reduced price points; other important aspects of this pull that are important are the ways in which the circuits can be grouped, how the chips are packaged, and their power and cooling requirements.
2. *application push*—the rapid growth of the diverse ways in which computer systems are used; for example,
    a. accessing and manipulating streams of rich, multimedia data such as full-motion video and high-resolution, highly textured graphics.
    b. relying on integrated, high-speed communications to local networks, the Internet, and multiple, geographically dispersed intranets.
    c. exploiting the evolution of highly interactive human-machine interfaces such as virtual reality and real-time, speaker-independent speech input.
    d. providing the central functions for a wide variety of consumer electronics devices (e.g., cameras, VCRs, high-fidelity and home entertainment systems, set-top boxes, security systems, cellular phones, pagers, and other wireless devices, and home appliances, as well as in automobiles, boats, trains, airplanes, geo-positioning systems, and production lines in factories) has placed different requirements on the underlying processor architecture, core logic chipset, bus and memory architectures, motherboard functionality, and overall platform.

It is worth noting in passing that there have also been rather dramatic shifts in the way processors and platforms are marketed and distributed. These shifts have had a profound impact on the products and services supplied and on the growth of the computer industry. However, they are not the topic of this book.

Architecture, Microarchitecture, and Platform and System Literature

There are many fine textbooks, periodicals, and conference proceedings that deal with various aspects of designing, implementing, and testing high-performance processors and microprocessors. However, this base of

literature typically does not deal with the platform and systems related issues such as core logic chipset and BIOS design, bus and memory architectures, motherboard functionality, and overall platform packaging that are associated with integrating processors into various types of systems.

Suggested Readings

Technology Pull and Application Push

A reading of the articles in the September 1997 issue of *Computer* magazine (Vol. 30, No.9) will readily verify that the field is ripe with discussion on various approaches to address both the *technology pull* and the *application push*. A partial listing of these articles is:

1. D. Matzke, *Will Physical Scalability Sabotage Performance Gains?*, pp. 37-39.
2. J. A. Fisher, *Walk-Time Techniques: Catalyst for Architectural Change*, pp. 40-42.
3. K. Diefendorff and P.K. Dubey, *How Multimedia Workloads Will Change Processor Design*, pp. 43-45.
4. Y. N. Patt, S.J. Patel, M. Evers, D.H. Friendly and J. Stark, *One Billion Transistors, One Uniprocessor, One Chip*, pp.51-57
5. M.H. Lipasti and J.P. Shen, *Superspeculative Microarchitecture for Beyond* AD 2000, pp. 59-66.
6. J. E. Smith, and S. Vajapeyam, *Trace Processors: Moving to Fourth-Generation Microarchitectures*, pp. 68-74.
7. L. Hammond, B.A. Nayfeh and K. Olukotun, *A Single-Chip Multiprocessor*, pp. 79-85.

You can find the full text versions of articles 3, 4 and 5 on the companion CD-ROM.

Moreover, this same literature base does not treat fabrication related issues and how they impact what can and what cannot be included in the architecture, microarchitecture, platform, or system. For example, when producing a microprocessor, the microarchitectural design team and implementation team might have to work closely with those doing the floorplanning of the chip in order to know the ultimate size of specific resources on the chip, (e.g., the on-chip caches or associated TLBs). Such restrictions will have a great deal to do with the ability to reach a specific price/performance point and might steer the design group to choose other processor features (a larger prefetch buffer, for example) in an attempt to compensate for the restrictions. Few existing books deal with the numerous steps involved in taking a high-level design and translating it into silicon.

Architecture and Microarchitecture

Computer architecture and microprocessor books typically deal with processors at various levels of abstraction—the instruction set architecture level, the microarchitecture level, the logical design level, and the actual microprocessor chip itself. These levels are differentiated from one another in this book as follows:

> **DEFINITION**
>
> **Architecture**
>
> Architecture refers to the instruction set, resources, and features of a processor that are visible to software programs running on the processor. The architecture determines what software the processor can directly execute and essentially forms a specification for the microarchitecture.

The term *architecture* as defined here is often called the instruction set architecture.

instruction set architecture

> **DEFINITION**
>
> **Microarchitecture**
>
> Microarchitecture refers to the set of resources and methods used to realize the architecture specification. The term typically includes the way in which these resources are organized as well as the design techniques used in the processor to reach the target cost and performance goals. The microarchitecture essentially forms a specification for the logical implementation.

Sometimes in the computer architecture literature the *architecture* is called the “virtual machine” and the *microarchitecture* is called “host machine.” This definition of microarchitecture leads us to a definition for the *logical design or logical implementation* level.

virtual machine
host machine

> **DEFINITION**
>
> **Logical Design or Logical Implementation**
>
> Logical design or logical implementation refers to the actual logic and circuit designs used to realize the microarchitecture specifications. These designs essentially form a specification for the microprocessor chip itself.

And finally, this brings us to the physical implementation of the *microprocessor chip* itself,

DEFINITION

Microprocessor Chip

The microprocessor chip is the physical implementation of the logical design in a given semiconductor process technology.

Many contemporary architecture and microprocessor textbooks deal with a set of core issues in designing computers, such as:

1. the structure and control of basic building blocks (combinational logic, sequential logic, adders, shifters, decoders, multiplexers, etc.).
2. basic control mechanisms (finite state machines, field programmable gate arrays, microcode, state machines, etc.).
3. instruction set design (register and memory addressing, highly/minimally encoded opcodes, compiler issues, operating system issues, architecture state, etc.).
4. parallelism, synchronization, and consistency issues (pipelining, hazards, multiple functional units, instruction level parallelism, task switching, multi-computer and multi-processor support, etc.).
5. cache and memory design (cache organizations and protocols, memory hierarchies, burst and pipeline accesses, virtual memory, coherency, etc.).
6. input/output systems (interrupt, handlers, masking, channels, device controllers, etc.).
7. interconnection and network technology (buses, crossbars, switches, routers, hubs, etc.)

Introductory texts in the field deal with the more fundamental of these issues while the more advanced texts assume the fundamentals as background. Many of the books aid the reader in understanding specific issues by using examples of real architectures to demonstrate a particular solution to a given problem and discussing the implications of the specific solution. Almost none, except for the Thornton book and the two Organick books cited above, deal with a specific commercially successful architecture in depth, giving the reader a reasonably thorough understanding of how it works. The Golze text presents a thorough treatment of a non-commercially available microprocessor but is, nonetheless, instructive.

.

Suggested Readings

Books and Articles on Computer Architecture and Microprocessors

Some contemporary books of interest are:

1. Harold S. Stone, *High-Performance Computer Architecture*, 3rd Edition, Addison-Wesley, 1993.
2. Michael J. Flynn, *Computer Architecture: Pipelined and Parallel Processor Design*, Jones and Bartlett, 1995.
3. David A. Patterson, and John L. Hennessy, *Computer Architecture: A Quantitative Approach*, 2nd Edition, Morgan Kaufmann, 1996.

Additionally, the proceedings of the Annual International Symposium on Computer Architecture (ISCA) are extremely important in the evolution of the field. The following three books are important because of the collection (and analysis) of early processor designs which they contain. In our opinion, every student of computer architecture should examine them:

1. Daniel P. Siewiorek, C. Gordon Bell, and Allen Newell, *Computer Structures: Principles and Examples*, McGraw-Hill, 1982.
2. C. Gordon Bell, and Allen Newell, *Computer Structures: Readings and Examples*, McGraw-Hill, 1971.
3. Peter M. Kogge, *The Architecture of Pipelined Computers*, McGraw-Hill, 1981.

Other informative historical books, because of their in-depth treatment of interesting and important computers, are:

1. J. E. Thornton, *Design of a Computer: The Control Data 6600*, Foresman Press, 1970.
2. E. I. Organick, *The Multics System: An Examination of Its Structure*, MIT Press, 1972.
3. E. I. Organick, *Computer System Organization: the B5700/B6700 Series*, Academic Press, 1973.

A recent book, presenting a comprehensive treatment of the design and implementation of the University of Braunschweig's TOOBSIE microprocessor can be found in Ulrich Golze, *VLSI Chip Design with the Hardware Description Language Verilog: An Introduction Based on a Large RISC Processor Design*, Springer-Verlag, 1996.

PC Platforms and Systems

We integrate microprocessors and a number of other components to form *platforms* that address the needs of specific market segments, such as those required in the desktop, workstation, or laptop markets.

Definition

Platform

A platform consists of a number of key components and interconnections on a motherboard and typically includes a high-performance peripheral bus and ports, main memory, an I/O module, a processor module and appropriate BIOS code. The processor module typically includes the processor, processor local bus, optional external cache, and a controller for the peripheral bus and the main memory. The I/O module includes bus controllers and ports for standardized and optional peripherals.

Systems are built from platforms.

Definition

System

A system consists of a platform extended with a number of essential and optional peripherals, an operating system, device drivers including BIOS extensions, other configuration and power management software, and a basic set of applications software. Examples are a Microsoft Windows 95 desktop system and a Unix-based workstation system.

Platform and systems literature exist in an environment similar to that of architecture and microarchitecture: namely, there are many excellent existing information resources for platform and system design, including developer documentation, system architecture books, design guides, industry standards, conference proceedings, seminars, and workshops. Examples of these sources are shown in the *Suggested Readings* inset on the next page. While invaluable and recommended, these information sources do have shortcomings.

Conferences and seminars are relatively expensive, have limited geographic and calendar availability, and are essentially only accessible to a limited number of people. Currently available books, while generally affordable and easy to obtain, do not lend themselves to easy study by the computer engineer or computer scientist who are new to platform and system design. The system architecture books are in-depth reference materials on narrow facets of platform architecture and are suitable for advanced self-study by those already well versed in PC platforms. System design guides and standards documents conversely are comprehensive in

breadth, but have little depth, being essentially platform product specifications targeted for veteran platform designers.

Suggested Readings

Examples of Platform and Systems-Related Resources

Several Annual Conferences and Seminars are valuable, but have limited accessibility:

1. Microsoft's Windows Hardware Engineering Conference or WinHEC.
2. Micro Design Resources' Microprocessor Forum.

The following books are targeted at practitioners and have extensive details on narrow facets of system architecture:

1. Tom Shanley & Don Anderson, *PCI System Architecture,* Addison-Wesley, 3rd Edition, 1995.
2. Richard F. Ferraro, *Programmer's Guide to the EGA, VGA, and Super VGA Cards*, Addison-Wesley, 1994.

The next book is a systematic and all encompassing system specification for PC Platforms; it is broad, but has little depth:

1. *PC98 System Design Guide*, Microsoft Press, 1997.

The following WEB-sites include developer documentation, however the material tends to be disjoint and loosely organized collections of diverse documents:

1. http://www.microsoft.com/hwdev/
2. http://developer.intel.com/

(A note about URLs: URL links change frequently, especially at the lower levels of directory structures. If the links provided become invalid, try looking for the same topic at a higher level in the Web-site's directory hierarchy.)

PC Technical Reference guides that survey the entire PC platform are valuable for their breadth, but they are primarily targeted at PC "enthusiasts" and not practitioners. The following book is one of the best of this genre:

1. Winn L Rosch, *Hardware Bible*, Premiere Edition, SAMS, 1996.

Moreover, there are a number of PC-systems readings of historical interest that we can recommend to you:

1. "Perspective" Section articles from *Understanding x86 Microprocessors*, Ziff-Davis Press, 1987.
2. Lewis C. Eggebrecht, *Interfacing to the IBM Personal Computer*, Howard W. Sams & Co., 1986.
3. Edward Solari, *ISA & EISA Theory and Operation*, Annabooks, 1993.

Our book, in part, provides a needed survey and overview of the key concepts and essential standards of PC platform and systems. We hope you achieve a level of understanding and insight that will enable you to quickly make effective use of the other information resources such as those above and those referred to in later chapters.

Developer documentation generally takes the form of product support literature and is disjointed and scattered, rather than being organized in a

compact collection facilitating the study of platform and system design. We make use of an extensive set of references through the systems-based chapters. The following inset gives some examples showing the breadth of content and sources of platform and systems-related literature.

Given this summary of the architecture and microarchitecture and the platform and systems-related literature, we now proceed to the roles of this book.

The Dual Roles of the Chapters That Lie Ahead

Our book is not meant to replace any of the contemporary books in a course on computer architecture, computer engineering, or computer systems. Rather, it is intended to both complement and extend them by:

1. providing an in-depth look at the microarchitecture of a specific contemporary high-performance, high-volume microprocessor, AMD's K6 3D that employs many of the techniques discussed in conventional texts. Thus the book presents a detailed and coherent context to study and understand these techniques and their impact on the overall design of a microprocessor.
2. addressing the platform and systems issues identified above and examining the impact of these issues in designing different types of platforms—e.g., desktop systems, mobile systems, and entertainment systems.

Thus, our book is to be used in conjunction with a contemporary book on computer architecture or as a second book by those who have already taken a course in computer architecture. As mentioned earlier, the book is divided into two complementary sections; the first section describes the microarchitecture of the K6 3D microprocessor and the second deals with the platform and systems issues. We examine both of these roles—providing a detailed and coherent context for studying microarchitecture issues and providing a detailed and coherent treatment for studying platform and systems issues—sequentially in the following discussion.

Providing a Context for Studying Microarchitecture Issues

When used to provide a detailed and coherent context for studying microprocessors, an instructor can:

1. assign specific sections of this book as additional reading material corresponding to specific topic coverage in the course's textbook.
2. direct specific questions, as appropriate, at the end of the chapters in a conventional architecture text to the microarchitecture presented in this book.
3. frame entirely new questions related to design approaches

represented in the K6 3D's microarchitecture (e.g., evaluating a particular feature by modeling it or examining the implications of substituting a different design solution).

4. combine these approaches.

Assuming a prerequisite or concurrent course in computer architecture, this book can be used as a reference book or as a text in a graduate seminar course as a basis for studying and discussing issues in high-performance computing and instruction level parallelism. Whether used in conjunction with a text or as a reference volume, we believe that you will extend the range of solutions available to you in your own microprocessor-based projects.

Providing a Context for Studying Platform and Systems Issues

When used to provide a detailed and coherent treatment for studying platform- and systems-related issues, an instructor can:

1. assign specific sections of this book as additional reading material corresponding to specific topic coverage in the course's textbook.
2. develop individual or team projects to design and simulate one or more platform components (e.g., the Rambus memory channel and interface, which is discussed in detail in Chapter 5).
3. develop individual or team projects that center around the simulation, analysis, and prototyping of small platforms and systems using the platform- and systems-related chapters as appropriate background.
4. combine these approaches.

Again, assuming a prerequisite or concurrent course in computer architecture, the book can be used as a reference book or text in the study of platforms and systems by the practitioner or student who wants to understand the design and implementation of microprocessor-based systems. Indeed, the book could be used as a text in an undergraduate or graduate project or seminar course as a basis for studying system implementations. This will become clearer when we expand on our approach to discussing these issues later in this chapter.

The Amount of Detail and the Task at Hand

What must be done, of course, in a book that attempts to give the reader a substantive understanding of a specific microarchitecture and to systematically treat related platform and systems issues is to achieve the sometimes difficult balance of giving *enough* detail, but not *too much* detail.

On the microarchitecture side of the ledger, we try to achieve this balance by limiting our detailed discussions to three main aspects of the

K6 3D, namely, its out-of-order, speculative scheduler, its operation commit unit, and its register renaming scheme. To do this, we must burrow down into its microarchitecture to see how instructions are predecoded and then decoded, how multiple internal operations result from this decoding process, how these operations go through a substantial expansion process before they are loaded into its centralized scheduler, how its pipelines are controlled and what type of work they do at each stage, how the decoding of instructions and the execution of the resulting operations are decoupled from one another, and how and when predicted branches are ultimately resolved. Because explaining some of these concepts requires knowing a bit about the internal operation set of the microarchitecture and the internal representation of operations within the scheduler, we must present some detail of both.

On the platform and systems side of the ledger, our goals for the platform and systems chapters are complementary, but not without tension. We strive to maintain a balance between general applicability and commercial relevance. The PC platform is by far the most significant class of high-performance microprocessor-based system in terms of unit and dollar volume. Furthermore, it exerts tremendous influence in the design of PC-platform-compatible processors, such as the K6 3D. Thus, we have elected to complement the microarchitecture material by (a) providing insights into the PC-based systems environment in which high-performance microprocessors are currently designed, and (b) establishing a broadly applicable framework for microprocessor-based system design.

The PC-based systems environment is dominated by Microsoft Windows-based PC systems. Therefore, we explore this environment by providing a guide to the hardware architecture of PC platforms that support Windows. We examine how they are organized and optimized. This is done against a backdrop of examining the directions in which the PC industry is moving. The examples used throughout the systems-related chapters of the book are based on a PC platform targeted for desktop consumer 3D graphics applications. As 3D graphics continues to evolve, it presents a seemingly ever-growing demand on the bandwidth of the platform's buses, peripherals, and subsystems. However, in presenting this material, we must again deal with the issue of *too much* versus *not enough* detail.

Detail and Simulation

In the end, our criteria for *enough* detail was based on what we believe you need to know to help translate the passive words in this text into an active (and concrete) learning experience. We have given you enough detail to allow you to simulate important portions of the microarchitecture and platform components. You can gain a much greater understanding of how these chunks of a design actually work by doing simulations.

There are many important issues to consider when simulating something—issues arise for which there are no immediate answers and you have to make assumptions in order to get on with the work at hand. The knowledge gained from such an undertaking is important in understanding the cost and performance trade-offs involved in taking an abstract design of a microarchitecture and producing a working chip from it and, given the chip, the complexities of producing a platform from it. To aid you on this learning excursion, we have given "register transfer level"-like descriptions of key elements of the microarchitecture in Chapter 2. These descriptions provide a basis for the simulations which we are encouraging you to undertake. There are copies of VHDL and Verilog simulators and an interactive HDL simulation environment for both of these languages on the CD-ROM which accompanies this book for readers who do not have access to either type of simulator. In any case, you can hand translate the descriptions into the appropriate input form for whatever simulation system you want to use. It is quite obvious that we believe in the old axiom of the Dewey school of education, "Learn by doing."

We hope that our explanations turn out to facilitate your understanding and enjoyment of this material, and that we don't turn out to be like the authors that Don Norman comments on in the adjacent side panel, who would make this a difficult and uninteresting task.

We will start our journey into this text by examining the overall process of designing and implementing a microprocessor. We begin our journey into the platform and systems-related issues by giving a hardware overview of the design and implementation of a consumer-market 3D graphics PC platform. Both of these journey's begin in Chapter 1.

This concludes our introductory remarks regarding the technical content of the book. In closing, we briefly discuss the text insets, the companion CD-ROM, and give acknowledgments to those who helped us with this project.

"But These Are Complicated Topics. *One standard excuse of obscurantist authors is that the material in question is complex and technical, sometimes very abstract and refined. The fact that the writing is difficult to follow is unavoidable. The argument then gets turned around: The inability of readers like me to follow such complex thoughts reflects upon me, the reader, not upon the writing. If I really cared, I would do the work required to understand. And, if I still can't, well, I should just face up to the fact that my mind isn't sufficiently powerful. Complex ideas require complex writing, and then complex powerful minds to deal with them. Simple writing is for simple ideas, simple minds. Is there any case to be made for this? It sounds to me suspiciously like those folks who told me that if I made errors using the Unix computer system, why then I had no business using it. Clearly those who are incompetent to use something or to understand a text have no business trying to do so. Isn't this a great defense? You can cover up any kind of inelegant design or writing this way. Wonderful*."
Donald Norman, *Turn Signals are the Facial Expressions of Automobiles*, Addison-Wesley, 1992, pp.178-179.

Text Insets

From time-to-time, *text insets* appear in the body of the text and are labeled as either *Design Notes*, *Comparative Analysis*, *Historical Comments*, *Industry Standards*, *Pseudo-RTL Descriptions*, or *Suggested Readings*. The *Design Notes* sections are meant to specifically identify for the reader interesting, novel, or new design approaches that are implemented in the K6 3D. The *Comparative Analysis* sections show how the K6 3D distinguishes itself from other microprocessors and what the differences amount to in terms of cost, performance, or fabrication- and production-

related issues. The *Historical Comments* sections reestablish our connections to the first several decades of processor design, lest we think everything in microarchitecture and instruction level parallelism is newly developed. As is true with many areas of computer science and engineering, some results are often "re-discovered." The *Industry Standards* sections call your attention to specific industry standards and platform or system specifications that are important to the specific topic under discussion. The *Suggested Readings* sections, often appear with *Historical Comments*, suggest information sources that can prove valuable to both students and practitioners alike. The *Pseudo-RTL Description* sections, briefly introduced in Chapter 1 and treated extensively in Chapter 3, form the basis of simulations that we encourage you to perform.

Description, Content and Use of the CD-ROM

The complete text of the book, including all figures and tables, are on the companion CD-ROM and can be viewed using Adobe's Acrobat Reader. There are a substantial number of hyperlinks throughout the CD-ROM version of the book. For example, all entries in the Table of Contents, List of Figures, List of Tables, and the Glossary/Index[iii] are hyperlinked into the appropriate section of the book. All cross-references are hyperlinked and the *entire* book has been indexed so that arbitrary, free text, multiple-term Boolean queries can be used to identify sections of interest using the Adobe Acrobat Search Plug-In. Moreover, may of the key figures in the text are hyperlinked, i.e., clicking on an element within the figure will take you to the text that deals with that element.

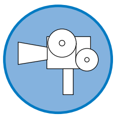
In reality, the CD-ROM provided with this book provides you with multiple books, not just one. A wide variety of items are included on it, such as complete data books, articles from journals and conference proceedings, manuscripts of important historical interest, industry standards, increasingly significant industry platform and system specifications, VHDL and Verilog simulators and a number of video and audio clips. The following icons are used in the text to identify relevant material on the CD-ROM that is related to the particular topic under discussion.

The "article" icon indicates that there is a copy of an article on the CD-ROM that appeared in either a periodical or conference proceedings sponsored by a professional society, such as the IEEE Computer Society.

[iii] Following the example in Stone's book, cited above, an extensive glossary appears within the book's index.

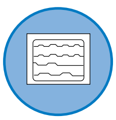
The "rubber stamp" icon with the word "STANDARD" indicates there is a copy on the CD-ROM of either (a) an IEEE Standard or (b) an industry document that has already emerged (or which we believe will emerge) as a "de facto" industry standard.

The "rubber stamp" icon with the word "REPORT" on it is used to identify all other technical articles and documents (i.e., those which do not appear in a professional society periodical or conference proceedings) on the CD-ROM. Some examples are: white papers; product data sheets; articles that appeared in industry newsletters; government or university reports; and industry developed specifications.

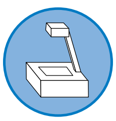


Additionally, there are several "video clips" on the CD-ROM. In these videos, various technical questions are presented to key members of the K6 3D design and implementations team.

There are also a number of "audio clips" on the CD-ROM that give overviews of specific technical topics. While some are directed at summarizing topics in the text, others expand on material given less detailed treatment.



There are a number of simulations on the CD-ROM that are identified by the "simulation screen" icon. We encourage the reader to run these simulations while studying the sections of the text in which they appear.



There are copies of a number of technical presentations on the CD-ROM covering important topics (e.g., the evolution of high-performance instruction set architectures, speculative execution, Socket 7, and Rambus technology). Technical presentations are identified by the "overhead projection" icon.

Adobe FrameMaker 5.5.2 and Adobe Acrobat 3.01 were used to produce this book in PostScript form for the printed version and in PDF form for the CD-ROM version. This allows you to use license-free readers/viewers (e.g., Adobe's Acrobat Reader and Apple's QuickTime viewer) as the vehicles for accessing material on the CD-ROM. See the CD-ROM User's Guide for relevant information on using the companion CD-ROM.

Web-site

Readers can submit questions, comments, detected errors, and suggestions at the Web-site that is associated with the book and its companion CD-ROM. They can receive a list of verified errata and corrections and a

variety of other information related to the book at this Web-site. The URL for the site is:

http://computer.org/books/anatomy.htm

Whenever additional material is made available, in any of the categories discussed earlier in the section titled, "Description, Content and Use of the CD-ROM," it will be made available at this Web-site. Examples of such material are: articles, reports, standards, suggested readings, and links to additional technical material; audio clips; book extensions, if and when they become available; exercises and problems posted by readers of the book, along with solutions when available; simulations of the various chunks of logic described by the pseudo-RTL descriptions of the architectural elements; and additional technical presentations as they become available and pointers to technical presentations

Acknowledgments

We give our sincerest thanks to Atiq Raza, Chief Technology Officer and Executive Vice President of AMD, who has been supportive of this project from its very inception; without his encouragement and support, it undoubtedly would not have been undertaken or completed. We would like to thank Harold Stone, NEC Research Laboratory, not only for his encouragement to proceed with this project after he had reviewed some early material regarding its content and form, but also for the numerous and substantive comments he made on the draft manuscript in an effort to improve its overall quality.

Similarly, we would like to thank Greg Favor, the chief architect of the K6 3D, and Amos Ben-Meir, the chief designer of the K6 3D, who gave substantial amounts of their time to not only describing the processor to us, but reviewing what we had written about it. We thank Bill Siegle, AMD's Chief Scientist, and Keith Schakel, Warren Stapleton, and Anu Mitra, members of AMD's technical staff, who were involved in this information transfer and review process.

We thank the IEEE CS Press reviewers, Tore Larsen, University of Tromso (Norway), Ed Gallizzi, Eckerd College, and Ron Hoelzeman, University of Pittsburgh, for their technical reviews of the draft version of this manuscript. They made numerous recommendations to improve the technical content and presentation of the material. We also thank Yale Patt, University of Michigan, who reviewed and commented on very early versions of material that eventually found its way into Chapters 1 and 2.

There are technical contributions in Chapter 1 by Mario Barbacci, Carnegie-Mellon University, and by the Eckert-Mauchly Award Winners Robert Tomasulo, retired, Yale Patt, University of Michigan, Michael Flynn, Stanford University, Daniel Siewiorek, Carnegie-Mellon University, Harvey Cragon, University of Texas, Austin, and John Cocke, retired. Furthermore, Don Draper of AMD played an important role in the writing of

the sections on gate and circuit-level simulation, netlists and physical layout, and mask generation, wafer fabrication, and packaging in Chapter 1. We thank each of them for these contributions.

Special thanks are due to Ricky See and Keith Schakel, both of AMD, for helping with the pseudo-RTL descriptions found in Chapters 2 and 3, to Jeff Trull of AMD for discussions and the diagrams related to the scan chains in Chapters 2 and 3, and to Mike Yamamura of AMD for help with the die photographs and the floorplan diagrams in Chapter 1.

We would like to thank Louise Burnham, the copy-editor, and, additionally, Beverly Shriver, Eva Mae Connell, and Mary Lou Wenger for their efforts in improving the readability of the manuscript. We acknowledge the efforts of Elaine Rose and Crystal Jacobson, both of AMD, and Joanne Winningham for their help with the myriad administrative issues related to the publication of a book and CD-ROM. We owe special thanks to Lisa O'Conner of the IEEE Computer Society not only for all of her work scanning and converting a substantial number of the articles that appear on the CD-ROM so they could be electronically searched by the reader, but also for shepherding this project through the production process. We thank Tom Fink, IEEE Computer Society, for his work on the book's Web-site. We also thank Joseph Daigle of Studio Productions for his work on the book jacket, book cover, CD-ROM packaging, and CD-ROM label.

With respect to acquiring many of the technical articles and other platform- and system-related material on the CD-ROM and in the body of the book, we must acknowledge:

> Matt Loeb of the IEEE Computer Society for helping acquire permission to reprint material from the IEEE and the IEEE Computer Society (and, importantly, for his unfailing and enthusiastic support of this project).
>
> Judith Gorman of the IEEE for helping us with the permission associated with the IEEE Standards and to Susan Tatiner and Greg Kohn, both of the IEEE, for producing the associated PDF files.
>
> Satinder Singh, Jeff Ferris, Ned Finkle, Jerry Isaac, David Kyle, Levi Murray, and Jim Reilly, all of AMD, for helping us with K6-related documents and other special requests.
>
> Gene Hoffnagel of IBM for helping us obtain permission to reprint the articles from the *IBM Journal of Research and Development*.
>
> J.A.N. Lee of Virginia Tech for helping us locate a copy of the 1945 von Neumann article and to Michael Godfrey of Stanford University for providing us with a postscript version of it.
>
> Bill Aspray at the Computing Research Association for

helping us locate a copy of the Burks, Goldsine, and von Neumann article.

Dennis Allison of Stanford University and Linley Gwennap of MicroDesign Resources for their help in obtaining permission to reprint articles from the *Microprocessor Report*.

Additionally, we need to thank AMD, AMP, IBM, Molex, Rambus, SLDRAM and Standard Microsystems Corporation for allowing us to include numerous tables, figures, and/or other material in the hard-copy version of the book or on the CD-ROM.

The design and layout of both the book and the CD-ROM were collaborative efforts led by Bruce Shriver and involved Bennett Smith, Dennis Hays (Dennis Hays Information Design), Matthew Shriver (Fish & Neave), Lynn Edwards (Bookmark Media, in the early phases of the project), and feedback from Michael Elliott (IEEE Computer Society), and several reviewers. A number of graphics components were designed specifically for this book. Among them are the icons used in the hard copy of the book, the icons used on the CD-ROM, the CD-ROM packaging graphics, and the book jacket and CD-ROM images. We owe our thanks to the creative talents of Matthew Shriver who did all of these graphics and contributed to the project in additional ways. We also thank Alexander Bovone (composer) for the audio accompaniment to Matthew's animation on the CD-ROM.

Donna Mitchell of SynaptiCAD Inc., Elliot Mednick of Wellspring Solutions, Inc., and Dave Pellerin of Accolade Design Automation, Inc. gave their time, energy, and effort to not only make sure we had the most recent versions of their simulation systems and documentation but also to ensure that we had tutorials on their use.

We should acknowledge Marshall Eisen and Dan Geller of StoryLine Productions and Sandra Wheatly and Scott Allen of AMD for their help with filming, editing, and administrative issues related to the production if the video clips. Lastly, David Slone of Townsend, Townsend, & Crew, Bob Flynn, Ahmed Aslam, and Titus Chi of Polytechnic University, Mark Podlipec of Bay Networks, and Brent Hailpern and Hank Warren of IBM T. J. Watson Research Center all helped with the testing of the CD-ROM for cross-platform compatibility.

Enjoy,

Bruce Shriver, Ossining, New York
Bennett Smith, Palo Alto, California
1998