Contents

Foreword to the First Edition

Kai Hwang has introduced the issues in designing and using high performance parallel computers at a time when a plethora of scalable computers utilizing commodity microprocessors offer higher peak performance than traditional vector supercomputers. These new machines, their operating environments including the operating system and languages, and the programs to effectively utilize them are introducing more rapid changes for researchers, builders, and users than at any time in the history of computer structures.

For the first time since the introduction of Cray 1 vector processor in 1975, it may again be necessary to change and evolve the programming paradigm—provided that massively parallel computers can be shown to be useful outside of research on massive parallelism. Vector processors required modest data parallelism and these operations have been reflected either explicitly in Fortran programs or implicitly with the need to evolve Fortran (e.g., Fortran 90) to build in vector operations.

So far, the main line of supercomputing as measured by the usage (hours, jobs, number of programs, program portability) has been the shared memory, vector multiprocessor as pioneered by Cray Research. Fujitsu, IBM, Hitachi, and NEC all produce computers of this type. In 1993, the Cray C90 supercomputer delivers a peak of 16 billion floating-point operations per second (a Gigaflops) with 16 processors and costs about $30 million, providing roughly 500 floating-point operations per second per dollar.

In contrast, massively parallel computers introduced in the early 1990s are nearly all based on utilizing the same powerful, RISC-based, CMOS microprocessors that are used in workstations. These scalar processors provide a peak of ≈ 100 million floatingpoint operations per second and cost $20 thousand, providing an order of magnitude more peak per dollar (5000 flops per dollar). Unfortunately, to obtain peak power requires large-scale problems that can require $O(n^3)$ operations over supers, and this significantly increases the running time when peak power is the goal.

The multicomputer approach interconnects computers built from microprocessors through high-bandwidth switches that introduce latency. Programs are written in either an evolved parallel data model utilizing Fortran or as independent programs that communicate by passing messages. The book describes a variety of multicomputers including Thinking Machines' CM5, the first computer announced that could reach a teraflops using 8K independent computer nodes, each of which can deliver 128 Mflops utilizing four 32-Mflops floating-point units.

The architecture research trend is toward scalable, shared-memory multiprocessors in order to handle general workloads ranging from technical to commercial tasks and workloads, negate the need to explicitly pass messages for communication, and provide memory addressed accessing. KSR's scalable multiprocessor and Stanford's Dash prototype have proven that such machines are possible.

The author starts by positing a framework based on evolution that outlines the main approaches to designing computer structures. He covers both the scaling of computers and workloads, various multiprocessors, vector processing, multicomputers, and emerging scalable or multithreaded multiprocessors. The final three chapters describe parallel programming techniques and discuss the host operating environment necessary to utilize these new computers.

The book provides case studies of both industrial and research computers, including the Illinois Cedar, Intel Paragon, TMC CM-2, MasPar M1, TMC CM-5, Cray Y-MP, C-90, and Cray MPP, Fujitsu VP2000 and VPP500, NEC SX, Stanford Dash, KSR-1, MIT J-Machine, MIT *T, ETL EM-4, Caltech Mosaic C, and Tera Computer.

The book presents a balanced treatment of the theory, technology, architecture, and software of advanced computer systems. The emphasis on parallelism, scalability, and programmability makes this book rather unique and educational.

I highly recommend Dr. Hwang's timely book. I believe it will benefit many readers and be a fine reference.

C. Gordon Bell

Preface to the Second Edition

Technologies underlying computer architecture—*VLSI*, communication, storage, graphics and others—have undergone tremendous advances over the last two decades. High performance computing, which was earlier reserved for very large research projects, has today become far more affordable, and the range of applications of computer systems has expanded enormously. The digital revolution in communication technology, the *world wide web*, and increased user awareness of the huge power of computers have been major contributors to the enormous growth in computer applications.

Against the backdrop of this scenario, the revision of a well-accepted textbook on computer architecture is necessary. The first edition of *Advanced Computer Architecture* by Kai Hwang has been hugely successful with teachers and students—the basic reason being that it explains simply and effectively the key principles and techniques of high performance computing systems.

New to this Edition

The first edition of the book provides a faithful and very useful record of developments in computer architecture in its creative and formative years, beyond the original uniprocessor systems based on the von Neumann model. This was the exciting period when many innovative research ideas were tried out which have had a long-lasting impact on the field. This record of developments remains valuable even today—almost two decades later—because often the same innovative ideas and concepts reappear at a latter stage in a different form, on a more advanced technology platform.

Keeping this factor in mind, the required revisions have been made in chapters 1 to 11 of the first edition. Some revisions are necessitated by changes in technology. For example, the successful *Scalable Coherent Interface* (SCI) and InfiniBand standards grew out of the IEEE Futurebus+ standard, while the latter itself failed to take off. Thus it became necessary to describe this development in the revised edition of the book, since it has had an impact on the evolution of system interconnect technology.

The topic of *Instruction Level Parallelism* has been discussed in a self-contained manner in the newly introduced **Chapter 12**. The basic concepts and techniques of instruction-level parallelism have been described, followed by a discussion of the relevant system design and performance issues which often place a practical upper limit on its successful exploitation.

The trends and advances in underlying technologies have been discussed in another newly added **Chapter 13**, which also touches upon issues such as the design trade-offs involved in multi-core processors. The basic concepts of *parallel algorithms* and *stream processing* have been described. This chapter also includes several specific case-studies of recently introduced processors, systems, multi-core *systems-on-a-chip*, function libraries, and also the parallel programming language Chapel being developed at Cray.

Issues related to *instruction level parallelism, processor clock speed* and *power consumption* have defined the recent direction of development of processor design. These issues have been discussed as appropriate in both the newly introduced chapters.

Chapter Summary has been added to all the chapters. In the interests of usability and topical relevance, **Chapter 12** of the first edition and bibliographic notes provided at the end of each chapter in the first edition have been moved to the website associated with the book.

To summarize, the new features are as follows:

- Two new chapters on *Instruction Level Parallelism* and *Trends in Parallel Systems*
- Topical inclusions—
  - Pipelining hazards, data hazards, control hazards
  - PCI bus and PCI Express
  - Cluster computing, interconnection networks and clusters
  - MPI, openMP, PVM, Pthreads
  - Multi-core processors
  - Impact of technology
  - Stream processing
  - Programming language Chapel
  - Introduction to VLSI computing structures
- Updated coverage of recent processors and systems—Intel Pentium IV, UltraSparc, Blue Gene (from IBM), Cray XT series, XT5 and XMT
- Chapters 1 to 11 revised to better reflect the contributions of earlier systems to the development of computer architecture
- *Chapter Summary* added in each chapter
- Strong pedagogical features:
  - 360 Illustrations
  - 114 Solved Examples
  - 251 Exercise Problems
  - 6 (new) Case Studies

Chapter Organization

The book is divided into five parts and 13 chapters.

Part I is on the *Theory of Parallelism* and contains chapters 1 to 3.

Chapter 1 on Parallel Computer Models describes the state of computing, multiprocessors, multicomputers, multivector and SIMD computers, and PRAM and VLSI models. It also gives an overview of architectural development tracks.

Chapter 2 discusses Program and Network Properties. Conditions of parallelism, program partitioning and scheduling, program flow mechanisms and system interconnect architectures are explained in depth.

Chapter 3 deals with the Principles of Scalable Performance. This chapter covers the topics on performance metrics and measures, parallel processing applications, speedup performance laws and scalability analysis and approaches.

Part II deals with *Hardware Technologies* and contains chapters 4 to 6.

Chapter 4 provides coverage on Processors and Memory Hierarchy. Advanced processor technology, memory hierarchy technology and virtual memory technology are discussed in detail.

Chapter 5 presents the concepts of Bus, Cache and Shared Memory. It treats the topics of backplane bus systems, cache memory organizations, shared memory organizations and sequential and weak consistency models in depth.

Chapter 6 is on Pipelining and Superscalar Techniques. Linear and nonlinear pipeline processors; and instruction, arithmetic and superscalar design are explained in detail.

Part III is on *Parallel and Scalable Architectures* and contains chapters 7 to 9.

Multiprocessors and Multicomputers are taken up in **Chapter 7**. Multiprocessor system interconnects, cache coherence and synchronization mechanisms, three generations of multicomputers and message-passing mechanisms are covered in detail.

Chapter 8 is on Multivector and SIMD Computers. This deals with vector processing principles, multivector multiprocessors, compound vector processing and SIMD computer organizations. An overview of the Connection Machine CM-5 is also given here.

Chapter 9 is on Scalable, Multithreaded and Dataflow Architecture. This chapter takes up the concepts of latency-hiding techiniques, principles of multithreading, fine-grain multicomputers, scalable, multithreaded, dataflow and hybrid architectures.

Part IV discusses *Software for Parallel Programming*. This part contains chapters 10 and 11.

Chapter 10 covers Parallel Models, Languages and Computers. Parallel programming models, languages and compilers are discussed first. Dependence analysis of data arrays and code optimization and scheduling are taken up thereafter. Finally, the chapter ends with detailed section on loop parallelization and pipelining.

Parallel Program Development and Environments are explained in **Chapter 11.** Here, parallel program environments, synchronization and multiprocessing modes, shared-variable program structures, message-passing program development and mapping programs onto multicomputers are dealt with in depth.

Part V deals with the *Instruction and System Level Parallelism*. This part contains the two newly introduced chapters 12 and 13.

Chapter 12 is on Instruction Level Parallelism. Design issues are brought out using the basic problem definition in the context of a typical processor model. Specific techniques discussed include reorder buffer, register renaming, operand forwarding, branch prediction, and Tomasulo's algorithm. Compiler-detected instruction level parallelism and thread level parallelism are also explained, along with the practical limitations encountered in exploiting instruction level parallelism.

Chapter 13 elucidates the current trends in parallel systems. A brief overview of relevant semiconductor, display, storage, interconnect and network technologies is taken up first, followed by forms of parallelism, some relevant concepts of parallel algorithms, and several case studies of current processors and systems. In the latter part of the chapter, parallel programming models and the parallel programming language Chapel are discussed.

All chapters are supplemented with exhaustive pedagogical features like summary, solved examples and exercise problems. Moreover, case studies are given at relevant places.

Issues related to *instruction level parallelism, processor clock speed* and *power consumption* have defined the recent direction of development of processor design. These issues have been discussed as appropriate in both the newly introduced chapters.

Chapter Summary has been added to all the chapters. In the interests of usability and topical relevance, **Chapter 12** of the first edition and bibliographic notes provided at the end of each chapter in the first edition have been moved to the website associated with the book.

To summarize, the new features are as follows:

- Two new chapters on *Instruction Level Parallelism* and *Trends in Parallel Systems*
- Topical inclusions—
  - Pipelining hazards, data hazards, control hazards
  - PCI bus and PCI Express
  - Cluster computing, interconnection networks and clusters
  - MPI, openMP, PVM, Pthreads
  - Multi-core processors
  - Impact of technology
  - Stream processing
  - Programming language Chapel
  - Introduction to VLSI computing structures
- Updated coverage of recent processors and systems—Intel Pentium IV, UltraSparc, Blue Gene (from IBM), Cray XT series, XT5 and XMT
- Chapters 1 to 11 revised to better reflect the contributions of earlier systems to the development of computer architecture
- *Chapter Summary* added in each chapter
- Strong pedagogical features:
  - 360 Illustrations
  - 114 Solved Examples
  - 251 Exercise Problems
  - 6 (new) Case Studies

Chapter Organization

The book is divided into five parts and 13 chapters.

Part I is on the *Theory of Parallelism* and contains chapters 1 to 3.

Chapter 1 on Parallel Computer Models describes the state of computing, multiprocessors, multicomputers, multivector and SIMD computers, and PRAM and VLSI models. It also gives an overview of architectural development tracks.

Chapter 2 discusses Program and Network Properties. Conditions of parallelism, program partitioning and scheduling, program flow mechanisms and system interconnect architectures are explained in depth.

Chapter 3 deals with the Principles of Scalable Performance. This chapter covers the topics on performance metrics and measures, parallel processing applications, speedup performance laws and scalability analysis and approaches.

V S Shankar [illegible]
Birla Institute of Technology, Mesra
Ranchi, Jharkhand

A P Shanthi
College of Engineering, Anna University
Chennai, Tamil Nadu

K V Madhu Murthy
College of Engineering, S V University
Tirupati, Tamil Nadu

N Subrahmanyam
National Institute of Technology (NIT-W)
Warangal, Andhra Pradesh

Sincere thanks are due to the editorial and production team at Tata McGraw-Hill who have worked on the book—and specifically to Surbhi Suman, Surbhi Shukla, Anjali Razdan and Sohini Mukherjee.

Naresh Jotwani

Feedback

We hope that teachers and students will find the revised edition of this book useful, as they set out to explore the fascinating world of high performance computer architecture. Please send your comments, views and suggestions to tmh.csefeedback@gmail.com, mentioning the title and author name in the subject line. Please, also do report any piracy of the book spotted by you.