ADVANCES IN COMPUTER ARCHITECTURE

Second Edition

GLENFORD J. MYERS

Advances in
Computer Architecture

Second Edition

OTHER BOOKS BY GLENFORD J. MYERS

Reliable Software Through Composite Design, 1975
Software Reliability: Principles and Practices, 1977
Composite/Structured Design, 1978
Advances in Computer Architecture, First Edition, 1978
The Art of Software Testing, 1979
Digital System Design with LSI Bit-Slice Logic, 1980

Advances in Computer Architecture

Second Edition

GLENFORD J. MYERS
Intel Corporation
Santa Clara, California

1807 175 YEARS OF PUBLISHING 1982

A WILEY-INTERSCIENCE PUBLICATION
JOHN WILEY & SONS
New York Chichester Brisbane Toronto Singapore

Library of Congress Cataloging in Publication Data:

Myers, Glenford J., 1946–
Advances in computer architecture.

"A Wiley-Interscience publication."
Includes bibliographical references and index.
1. Computer architecture. I. Title.
QA76.9.A73M93 1981 621.3819'52 81-11374
ISBN 0-471-07878-6 AACR2

Printed in the United States of America

10 9

TO MY PARENTS

Preface to the Second Edition

In the three years between the publication of the first edition of this book and the writing of this edition, important advances in computer architecture have occurred. These advances were motivated by a number of factors, including the arrival of VLSI (very-large-scale integration) design and manufacturing capabilities, renewed interest in programming-language improvements (e.g., Lisp, Ada), and increasing agreement with the issues and concepts discussed in the first edition of this book (e.g., tackling the software-development problem through improved computer architectures). These advances have appeared in such systems as Intel's iAPX 432 microprocessor and IBM's System/38, and in the implementation of the SWARD machine (an early version of which was described in the first edition).

I was motivated to write this second edition by the development of these new systems, new work in such other areas as data-flow and database machines, a desire to broaden and elaborate upon the ideas put forth in the first edition, and my personal experiences in the hardware and software implementation of the SWARD system.

The organization of the second edition parallels that of the first. Part I (Chapters 1–4) has been largely rewritten, and now contains more quantitative information and, I hope, more cogent arguments. Examples from the Motorola 68000 and IBM System/38 are used in Chapter 4. The case studies in Parts II, III, and IV have been retained from the first edition, although a number of inaccuracies were corrected. Part V, the case study of the SWARD machine, was completely rewritten to reflect the current state of this architecture. Part VI is a new case study, the Intel iAPX 432, a revolutionary microprocessor architecture. Part VII is an expansion of the material in the first edition on database machines. Also added was a chapter on data-flow machines. Finally, Part VIII contains some general information on the process of computer architecture.

I thank several individuals whose efforts contributed to this second edition, namely, George Cox, Justin Rattner, and Ken Aupperle of Intel and Dave Ditzel of Bell Laboratories.

GLENFORD J. MYERS

Santa Clara, California
October 1981

Preface to the First Edition

Since the 1950s, we have witnessed many advances in computing systems. The software field has advanced tremendously; for instance, we now have better tools, methodologies, and programming languages, software applications are more sophisticated, new algorithms have been invented, and the construction of such programs as operating systems and compilers is fairly well understood. The construction of physical computing devices has also advanced significantly; for example, circuit speeds and densities have increased by orders of magnitude, new storage technologies have been invented, better algorithms have been devised, and the microprogramming concept has been exploited. However, we have seen almost no advances at the hardware/software interface, the level of a system usually referred to as the *computer architecture*. To be fair, there have been some significant advances, but they have not received widespread attention and have not found their way into most conventional systems. For instance, if the instruction sets of most current large-scale systems, minicomputers, and microcomputers are examined, they will be found to be strikingly similar to those of machines designed in the 1950s.

The similarity of the architecture of today's systems to earlier systems can cause us to become complacent about the subject; we look around us and see tremendous software and hardware advances, but see that the architectures of current systems are virtually the same as those of earlier systems, so we might be inclined to assume that people in the 1940s and 1950s invented all there was to be invented in the area of computer architecture. The result would be that the architecture of future systems would remain the same. This attitude is my motivation for writing this book: to destroy this complacency by showing that there are serious problems in current computer architectures and discussing advanced architectural concepts that will solve these problems.

The intent of this book can also be expressed by examining two possible alternative titles that were considered. One title was *Fifth-Generation Computer Architectures*. The term "fifth" was selected because of the feeling that the fourth generation is already on the drawing board and that these systems would undoubtedly retain the architecture of earlier systems. This title was discarded, however, because it looked too "flashy." Also it would be misleading because some of the concepts discussed in the book arose in certain second-generation systems. Another possible title was *Second-Era Computer Architectures*, but this title was discarded because of the feeling that prospective readers would confuse "era" for "generation" and form the impression that the book is a historical survey of the IBM 1401, Burroughs 200, and other "second-generation" (discrete transistor) machines.

The chapters within this book are organized into six parts. Part I defines computer architecture, takes a critical look at current architectures, and discusses a set of properties needed in future computer architectures. Parts II, III, IV, and V are case studies; they discuss four advanced architectures having many or all of the desirable properties discussed in Part I. Part VI discusses other aspects of computer architecture, such as input/output considerations and the optimization or "tuning" of an architecture.

The book is intended for two audiences: for use as a text in a "second course" on computer architecture (where the "first course" would presumably cover conventional architectural concepts), and to spread some of these ideas to computer professionals in general. The reader is expected to have a good grasp of computer system fundamentals. In particular the reader should be knowledgeable of programming language concepts (e.g., the phrase "scope of names in a block-structured language" should be meaningful to the reader), have an understanding of the machine or assembly language of a conventional machine (e.g., S/370, PDP-10, CDC 6600), be familiar with operating system and compiler concepts (e.g., the term "reverse Polish notation" should be a familiar one), and have a grasp of the concept of microprogramming. A basic premise of this book is that this knowledge is prerequisite to the development of computer architectures.

I have found that the most effective way to understand an architecture is to do a mental compilation of a high-level language program to the architecture; many of the examples in the book were developed along this line. If the book is being used as a text, the student should be assigned a number of small PL/I, Cobol, or Fortran programs to be mentally compiled to each architecture.

Lastly, the opinions in the book are those of the author and do not necessarily represent the opinions, or future product directions, of the IBM Corporation.

GLENFORD J. MYERS

January 1978

Contents

PART I THE NEED FOR ARCHITECTURAL ADVANCES

PART V A SOFTWARE-ORIENTED ARCHITECTURE

PART VI AN OBJECT-ORIENTED MICROPROCESSOR