

COMPUTER
ARCHITECTURE
A
QUANTITATIVE
APPROACH



JOHN L HENNESSY
&
DAVID A PATTERSON

**Computer
Architecture
A
Quantitative
Approach**

David A. Patterson
UNIVERSITY OF CALIFORNIA AT BERKELEY

John L. Hennessy
STANFORD UNIVERSITY

With a Contribution by
David Goldberg
Xerox Palo Alto Research Center

MORGAN KAUFMANN PUBLISHERS, INC.
SAN MATEO, CALIFORNIA

Sponsoring Editor Bruce Spatz
Production Manager Shirley Jowell
Technical Writer Walker Cunningham
Text Design Gary Head
Cover Design David Lance Goines
Copy Editor Linda Medoff
Proofreader Paul Medoff
Computer Typesetting and Graphics Fifth Street Computer Services

Library of Congress Cataloging-in-Publication Data

Patterson, David A.

Computer architecture : a quantitative approach / David A.

Patterson, John L. Hennessy

p. cm.

Includes bibliographical references

ISBN 1-55880-069-8

1. Computer architecture. 2. Electronic digital computers

--Design and construction. I. Hennessy, John L. II. Title.

QA76.9.A73P377 1990

004.2'2--dc20

89-85227

CIP

Morgan Kaufmann Publishers, Inc.

Editorial Office: 2929 Campus Drive, San Mateo, CA 94403

Order from: P.O. Box 50490, Palo Alto, CA 94303-9953

©1990 by Morgan Kaufmann Publishers, Inc.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, recording, or otherwise—without the prior permission of the publisher.

All instruction sets and other design information of the DLX computer system contained herein is copyrighted by the publisher and may not be incorporated in other publications or distributed by media without formal acknowledgement and written consent from the publisher. Use of the DLX in other publications for educational purposes is encouraged and application for permission is welcomed.

ADVICE, PRAISE, & ERRORS: Any correspondence related to this publication or intended for the authors should be addressed to the editorial offices of Morgan Kaufmann Publishers, Inc., Dept. P&H APE. Information regarding error sightings is encouraged. Any error sightings that are accepted for correction in subsequent printings will be rewarded by the authors with a payment of \$1.00 (U.S.) per correction. Electronic mail can be sent to bugs@vsop.stanford.edu.

INSTRUCTOR SUPPORT: For information on classroom software and other instructor materials available to adopters, please contact the editorial offices of Morgan Kaufmann Publishers, Inc.

Contents

| | | |
|----------|---|--------------|
| | Foreword | ix |
| | by C. GORDON BELL | |
| | Preface | xvii |
| | Acknowledgements | xxiii |
| 1 | Fundamentals of Computer Design | 2 |
| | 1.1 Introduction | 3 |
| | 1.2 Definitions of Performance | 5 |
| | 1.3 Quantitative Principles of Computer Design | 8 |
| | 1.4 The Job of a Computer Designer | 13 |
| | 1.5 Putting It All Together: The Concept of Memory Hierarchy | 18 |
| | 1.6 Fallacies and Pitfalls | 21 |
| | 1.7 Concluding Remarks | 22 |
| | 1.8 Historical Perspective and References | 23 |
| | Exercises | 28 |
| 2 | Performance and Cost | 32 |
| | 2.1 Introduction | 33 |
| | 2.2 Performance | 35 |
| | 2.3 Cost | 53 |
| | 2.4 Putting It All Together: Price/Performance of Three Machines | 66 |
| | 2.5 Fallacies and Pitfalls | 70 |
| | 2.6 Concluding Remarks | 76 |
| | 2.7 Historical Perspective and References | 77 |
| | Exercises | 81 |
| 3 | Instruction Set Design: Alternatives and Principles | 88 |
| | 3.1 Introduction | 89 |
| | 3.2 Classifying Instruction Set Architectures | 90 |
| | 3.3 Operand Storage in Memory: Classifying General-Purpose Register Machines | 92 |
| | 3.4 Memory Addressing | 94 |
| | 3.5 Operations in the Instruction Set | 103 |
| | 3.6 Type and Size of Operands | 109 |
| | 3.7 The Role of High-Level Languages and Compilers | 111 |
| | 3.8 Putting It All Together: How Programs Use Instruction Sets | 122 |
| | 3.9 Fallacies and Pitfalls | 124 |
| | 3.10 Concluding Remarks | 126 |
| | 3.11 Historical Perspective and References | 127 |
| | Exercises | 132 |

| | | |
|----------|--|------------|
| 4 | Instruction Set Examples and Measurements of Use | 138 |
| 4.1 | Instruction Set Measurements: What and Why | 139 |
| 4.2 | The VAX Architecture | 142 |
| 4.3 | The 360/370 Architecture | 148 |
| 4.4 | The 8086 Architecture | 153 |
| 4.5 | The DLX Architecture | 160 |
| 4.6 | Putting It All Together: Measurements of Instruction Set Usage | 167 |
| 4.7 | Fallacies and Pitfalls | 183 |
| 4.8 | Concluding Remarks | 185 |
| 4.9 | Historical Perspective and References | 186 |
| | Exercises | 191 |
| 5 | Basic Processor Implementation Techniques | 198 |
| 5.1 | Introduction | 199 |
| 5.2 | Processor Datapath | 201 |
| 5.3 | Basic Steps of Execution | 202 |
| 5.4 | Basic Steps of Execution | 204 |
| 5.4 | Hardwired Control | 208 |
| 5.5 | Microprogrammed Control | 214 |
| 5.6 | Interrupts and Other Entanglements | 220 |
| 5.7 | Putting It All Together: Control for DLX | 238 |
| 5.8 | Fallacies and Pitfalls | 240 |
| 5.9 | Concluding Remarks | 241 |
| 5.10 | Historical Perspective and References | 244 |
| | Exercises | 244 |
| 6 | Pipelining | 250 |
| 6.1 | What Is Pipelining? | 251 |
| 6.2 | The Basic Pipeline for DLX | 252 |
| 6.3 | Making the Pipeline Work | 255 |
| 6.4 | Making the Pipeline Work | 257 |
| 6.4 | The Major Hurdle of Pipelining—Pipeline Hazards | 278 |
| 6.5 | What Makes Pipelining Hard to Implement | 284 |
| 6.6 | Extending the DLX Pipeline to Handle Multicycle Operations | 290 |
| 6.7 | Advanced Pipelining—Dynamic Scheduling in Pipelines | 290 |
| 6.8 | Advanced Pipelining—Taking Advantage of More Instruction-Level Parallelism | 314 |
| 6.9 | Putting It All Together: A Pipelined VAX | 328 |
| 6.9 | Putting It All Together: A Pipelined VAX | 334 |
| 6.10 | Fallacies and Pitfalls | 337 |
| 6.11 | Concluding Remarks | 338 |
| 6.12 | Historical Perspective and References | 343 |
| | Exercises | 343 |

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.