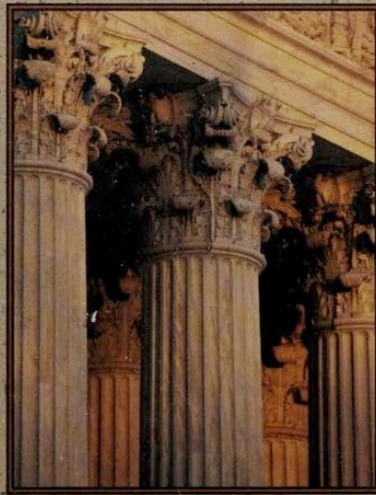


Computer Architecture

A Quantitative Approach

THIRD EDITION



John L. Hennessy
David A. Patterson

QA 26.9 . A78 P377 2003

Senior Editor Denise E. M. Penrose
Assistant Publishing Services Manager Edward Wade
Senior Production Editor Cheri Palmer
Editorial Coordinator Alyson Day
Cover Design Ross Carron Design
Cover Image Greg Pease / gettyimages
Text Design Rebecca Evans & Associates
Technical Illustration Lineworks, Inc.
Composition Nancy Logan
Copyeditor Ken DellaPenta
Proofreader Jennifer McClain
Indexer Ty Koontz
Printer Courier Corporation

Designations used by companies to distinguish their products are often claimed as trademarks or registered trademarks. In all instances in which Morgan Kaufmann Publishers is aware of a claim, the product names appear in initial capital or all capital letters. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

Morgan Kaufmann Publishers
An Imprint of Elsevier Science
340 Pine Street, Sixth Floor, San Francisco, CA 94104-3205, USA
www.mkp.com

© 1990, 1996, 2003 by Elsevier Science (USA)
All rights reserved

Published 1990. Third edition 2003
Printed in the United States of America
07 06 05 04 03 10 9 8 7 6 5 4 3 2 1

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

ADVICE, PRAISE, & ERRORS: Any correspondence related to this publication or intended for the authors should be addressed to ca3comments@mkp.com. Information regarding error sightings is also 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 upon availability of the new printing. Bugs can be sent to cabugs@mkp.com. (Please include your full name and permanent mailing address.)

Library of Congress Control Number: 2001099789

ISBN 1-55860-596-7 (cloth)
ISBN 1-55860-724-2 (paper)

This book is printed on acid-free paper.

UNIVERSITY OF TORONTO
LIBRARY

5.1 Introduction

Computer pioneers correctly predicted that programmers would want unlimited amounts of fast memory. An economical solution to that desire is a *memory hierarchy*, which takes advantage of locality and cost-performance of memory technologies. The *principle of locality*, presented in the first chapter, says that most programs do not access all code or data uniformly (see Section 1.6, page 47). This principle, plus the guideline that smaller hardware is faster, led to hierarchies based on memories of different speeds and sizes. Figure 5.1 shows a multilevel memory hierarchy, including typical sizes and speeds of access.

Since fast memory is expensive, a memory hierarchy is organized into several levels—each smaller, faster, and more expensive per byte than the next lower level. The goal is to provide a memory system with cost almost as low as the cheapest level of memory and speed almost as fast as the fastest level. The levels of the hierarchy usually subset one another. All data in one level are also found in the level below, and all data in that lower level are found in the one below it, and so on until we reach the bottom of the hierarchy.

Note that each level maps addresses from a slower, larger memory to a smaller but faster memory higher in the hierarchy. As part of address mapping, the memory hierarchy is given the responsibility of address checking; hence protection schemes for scrutinizing addresses are also part of the memory hierarchy.

The importance of the memory hierarchy has increased with advances in performance of processors. For example, in 1980 microprocessors were often designed without caches, while in 2001 many come with two levels of caches on the chip. As noted in Chapter 1, microprocessor performance improved 35% per year until 1986, and 55% per year since 1987. Figure 5.2 plots CPU performance projections against the historical performance improvement in time to access main memory. Clearly, there is a processor-memory performance gap that computer architects must try to close.

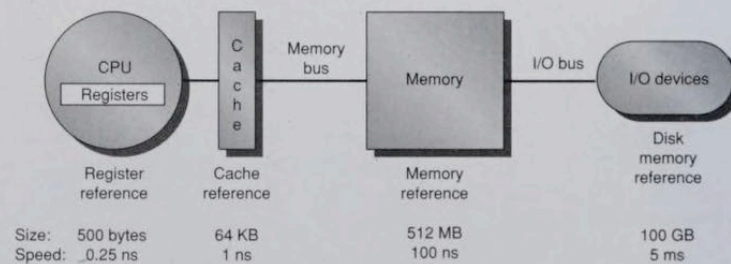


Figure 5.1 The levels in a typical memory hierarchy in embedded, desktop, and server computers. As we move farther away from the CPU, the memory in the level below becomes slower and larger. Note that the time units change by factors of 10—from picoseconds to milliseconds—and that the size units change by factors of 1000—from bytes to terabytes. Figure 5.3 shows more parameters for desktops and small servers.