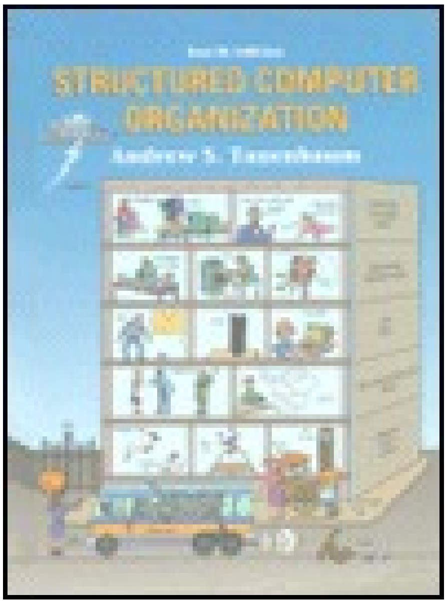

Structured Computer Organization

Andrew S. Tanenbaum

Structured Computer Organization

Andrew S. Tanenbaum

Structured Computer Organization Andrew S. Tanenbaum
Completely updated, this book explains how computer designers can follow the structured model to develop efficient hardware and software systems

New information has been included on UNIX, OS/2, INTEL 8088/80286/80386, Motorola 68000/68020/68030 and RISC machine. The operation of a typical IBM PC clone is now described in detail at the chip level.

Structured Computer Organization Details

Date	:	Published October 23rd 1998 by Prentice Hall (first published January 1st 1976)
ISBN	:	9780130959904
Author	:	Andrew S. Tanenbaum
Format	:	Hardcover 669 pages
Genre	:	Science, Computer Science, Nonfiction, Programming, Computers, Reference, Textbooks

Download and Read Free Online Structured Computer Organization Andrew S. Tanenbaum

From Reader Review Structured Computer Organization for online ebook

Nick says

read parts of this for school. interesting but awfully dry and requires some serious mental focus. that said, I now understand a little more what goes on under the hoods of these machines that I use everyday.

Joey Baker says

Im going to kill myself before I get through the whole thing. Binary numbers are fun though!

Mcebo Michael Metfula says

Great book. Very deep. Nice humor.

Russ says

This book is excellent, and covers a lot of fundamental computer architecture topics. It is, however, a bit dated at this point. The examples do the job, but so does Knuth v.1, so just go read that.

---

r says

The overall structure of the book makes perfect sense, but it can sometimes be difficult to follow due to the many dependencies. For example, it starts with a historical overview which largely presupposes an understanding of the terminology and technologies that are explained only later in the book. Also, for me, the inclusion of real-world examples throughout the text sometimes obscures, rather than clarifies, the conceptual exposition. Similarly for the many casual references to chips, machines, etc. - I often lacked the contextual knowledge to actually benefit from these (but I am sure they become more useful once one has delved deeper into the field).

Organisationally, it could perhaps also do with an acronym list of sorts, because I sometimes found it tedious to recall the meaning of e.g. a register acronym.

Despite my criticisms there is much to be gained from reading this text. For one thing, I now have a pretty good understanding of how I would go construct a functional processor out of a bag of transistors.

Kristine says

A+ for the use of humor! The IJVM was pretty cool.

David says

I read the 3rd edition of this book. The book reads well - it starts off assuming that the reader has little knowledge about computer architecture and gradually builds upon that. The book is not as detailed or technical as other books of its nature, but it does make for an easier read for newcomers to the topic. The complaint I have against this is that while the flow is good, the reading is not - the book's somewhat boring, but I guess that applies to books you have to read just to finish a university course.

Avinesh says

it is a good book

Nick Black says

The deeper processor design is covered just lightly enough to leave you utterly confused. Other than that, a solid introduction to architectural considerations.

Nelson Minar says

I've been hacking pretty deep on computers for more than half my life - I wrote my first 6502 assembly code at age 12, back in 1984 or so. But I never understood any of the details of what was going inside the box, I've never gone below the assembly code level before. I have to take the Computer Science GRE (bleah!) soon, though, so that was a good excuse to learn a little computer architecture. (I did fine on it, btw!).
Tanenbaum is a great book to learn that from. It's a bit simplistic (perfect sophomore CS textbook, I think), but the organization that he presents is excellent and it's fun to read. He starts from the bottom, a transistor (sadly, no quantum mechanics), and moves up through logic gates, LSI, simple computer design, VLSI, microcode, and then up to the software that runs on the hardware. Hardware makes sense to me now! Well, at least more so.

Paul says

I was disappointed in this text. While it provides brad coverage of computer organization, it does so at the cost of any significant depth.
