Structured Computer Organization by Andrew S. Tanenbaum

A Great Introduction To Computer Hardware And Architecture

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.

My Personal Review:
This well-written and often humorous (in the good sense) book is targeted for a freshman or sophomore in computer science or computer engineering. Following a few introductory chapters which give both a historical, conceptual, and structural overview of the computer, Tanenbaum partakes upon a semi-detailed introduction to the various levels of a computer, beginning with the digital-logic level (the actual hardware) and working up to both the operating-systems and assembly-language levels.

With respect to the digital-logic level, I thought he did well in introducing the student to the essential components (e.g. registers, ALU, Flip-Flops) without overburdening the student with design techniques such as Karnaugh maps, finite-state machines, etc. However, by far the best part of the book seemed to be his explanation of the microarchitectural level, in which the relationship between memory, control, and datapath was fully explained. This chapter seemed to be where the rubber met the road in terms of showing the connection between programs and hardware. I would have preferred however if he had not introduced the IJVM language so early in this chapter, and had spent more time demonstrating microprograms. I know for a fact that many of my students seemed very confused about the difference between microinstructions and machine instructions. Moreover, a majority of them found microprogramming with the Mic-1 very difficult if not impossible. More microprogram examples would probably have helped.

Finally, where as the later chapters on instruction sets and assembly language seemed very adequate introductory overviews, the operating-systems level seemed overly broad and of not much use. Certainly, a student should read his other book on OS to fully appreciate this all-too-important aspect of the computer.
In closing, I should mention that the Mic1 software also accompanies this text, and helps provide the student with a well-rounded education, in that they get some hands-on experience. The software and the well-written text make it an irresistible choice to use a first course in computer organization.