

## Book Review

RICHARD R. SHIVELY, *Book Reviews Editor*

In this section, the IEEE Computer Society publishes reviews of books in the computer field and related areas. Readers are invited to send comments on these reviews for possible publication in the Correspondence section of this TRANSACTIONS. Please address your comments and suggestions to the Book Reviews Editor: Richard R. Shively, Bell Telephone Laboratories, Inc., Whippany, N. J. 07981. The Computer Society does not necessarily endorse the opinions of the reviewers.

**B71-6 System Structure in Data, Programs, and Computers**—Lyle R. Johnson. (Englewood Cliffs, N. J.: Prentice-Hall, 1970, 303 pp., \$12.50.)

This book represents an interesting attempt to take an overall view of a *data system* defined by the author as “the artifact that consists of a digital computer, a control program, and an accessible library of programs and data.” His intended audience are professionals in the computer industry whose pursuit of specialized vocations leads them away from systematic consideration of the data system as a whole. His avowed aim is “crispness in style and frugality in detail.” Consequently this book covers a good deal of ground—sometimes very briefly.

The first five chapters cover data structures and their processing. The author’s basic structure is the tree diagram and within this framework he classifies various types of tree organizations utilizing nomenclature and notation from Iverson’s work. There are brief descriptions and comparisons of various types of table searches and techniques such as hash addressing, inverted files, etc., in one chapter and similar descriptions and comparison of sorting methods in another.

Chapter 6 is called “Structure in Programs” and consists mainly of a brief introduction to Iverson’s APL together with a few examples, although Fortran, Cobol, Algol, and PL/1 are mentioned. The next three chapters cover machine organization, starting with gates and memory elements, through registers, decoders, arithmetic units, memories, etc., to a description of the Edvac because of “its conceptual clarity and simplicity.” There is also discussion of more modern techniques, instruction and carry look ahead, multiprogramming, multiprocessing, etc. Included here is a chapter on addressing modes (direct,

indirect, etc.), assembly languages, interpreters and compilers, and emulation and simulation—all in 18 pages!

A chapter entitled “Table Management” discusses the memory allocation problem and possible solutions, list processing and garbage collection, overlays, and paging. Also included is a discussion of representation of data elements, alphameric codes, number formats, BCD codes, and error detection and correction.

Finally the last chapter, “The Data System,” gives an overview plus a brief look at the IBM 360 and its operating system OS. There is a discussion of various design tradeoffs, tape versus disk, on-line versus off-line I/O, hardware versus software, etc., all in 11 pages!

Since the complete book is 280 pages exclusive of references, which are quite extensive, it is apparent that he has met his goal of frugality in detail. I believe he has also met it in crispness in style; I found it very readable. The book attempts to organize, classify, and codify an enormous mass of information. It is not, however, a textbook. If you already know something, then he presents you with a great deal of material in a small space. If you do not know anything, you probably cannot learn from this book.

I found the first five chapters more interesting than later chapters. This is probably because the information in these earlier chapters is less familiar to me and therefore I learned more from it, but he generally has a variety of interesting ways of looking at even familiar material.

T. R. BASHKOW  
Dep. Elec. Eng. and Comput. Sci.  
Columbia Univ.  
New York, N. Y. 10027