fields so that a computer could be constructed or, if one had a computer, it could be used efficiently.

Sections 1 through 9 cover analog computers, while sections 10 through 21 are concerned with digital computers.

Covered under the subject of analog computers are terminology, design of analog-computer building blocks and systems, and the organization and maintenance of analog computers. Also covered are the applications of analog computers to control systems (such as process-control), random-process studies, and their application to mathematical solutions (such as algebraic-equation solvers), the solution of partial differential equations, and linear programming methods.

There is also a section devoted to solid-state analog computer components, which describes many new solid-state circuits.

The author describes many techniques that are not too familiar to many engineers, such as network-type analogies for fields and structures. These are illustrated by mechanical, electro-mechanical, hydrodynamic, and heat-transfer problems.

The remainder of the book deals with digital computers, the solid-state components used with computers and typical circuits such as emitter followers, shift registers, and adders. Also, input-output devices such as magnetic drums and tape handlers are covered thoroughly.

The only omission noted in the book was in the section dealing with logical elements. The use of stroke functions [1] in the reduction of logical systems was not mentioned as a means of determining minimal nets corresponding to a given set of logic functions.

The section covering programming and coding is thorough—indeed, probably too detailed in its description of ALGOL-60, in contrast to its abbreviated discussion of other languages.

The last section covers special-purpose computers, with particular emphasis on the digital differential analyzer, which the author states can be substituted in many cases for an analog computer in the solution of ordinary differential equations. The former provides a higher order of accuracy, since the precision of operations is limited only by a register size rather than by the tolerance of components of an analog computer. This type of special-purpose computer could be used to generate and supply continuous-control variables or other functions to a general-purpose computer.

In the opinion of the reviewer, this handbook offers much valuable information, even to the most experienced computer personnel.

WILLIAM SEALS

Applied Mathematics Laboratory
David Taylor Model Basin
Washington, D. C.

1. N. T. GRISAMORE, "Logical design using stroke functions," IRE *Transactions on Electronic Computers*, Vol. EC-7, No. 2, June 1958.

**69[Z].**—ALLEN NEWELL, ET AL., *Information Processing Language-V Manual*, Second Edition, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1964, xxxvi + 267 pp., 27 cm. Price $7.95 (paperbound).

The size of this book needs to be somewhat discounted because most of it is set in pica type, at four lines per vertical inch, and the pages are provided with generous

margins. It is well laid out, clearly written, and has a bibliography of 62 pertinent references. It consists of two parts; namely, a learning manual and a reference manual. The learning manual does not require prior knowledge of any form of computer programming. It contains a section, "Organizing Complete Tasks," which would be of value to persons learning other forms of programming. The learning manual contains exercises and solutions for some of them and is suitable for self-teaching. There is an index, which applies to the reference manual only.

For those already acquainted with IPL-V, the differences between the first and second editions are:

1. Conventions and definitions in the initial manual have been changed at only six minor points, all associated with loading and monitoring. These may impose minor modifications to existing programs. About 30 basic processes have been added. All the changes and extensions are listed in four pages of the manual and are also incorporated in their appropriate places in Part Two, the Reference Manual.

2. Part One, the section for those learning the language, has been largely re-written. Exercises, with sample solutions for some of them, have been added.

For those meeting IPL-V for the first time, if you have wondered about such expressions as list-processing, list structures, Newell-Shaw-Simon, push-down, pop-up, and recursion, this is a good source of relevant information. IPL-V, which was developed mainly by the RAND Corporation, is designed for non-arithmetic processing of symbolic information, especially information which can be organized in the form of lists. A list element consists of a data portion and the location of the next list element. Programs for computers having a "plus-one" instruction format, such as IBM 650, are lists of this type. The introduction gives the developments in programming which have led to IPL, explains the basic idea of lists, and gives some areas in which IPL has been used. Theorem proving, game playing, natural language processing, and artificial intelligence are examples thereof.

IPL-V is one of several list-processing languages now in use. The "V" indicates the fifth version in its series. IPL-V is a pseudo-code; that is, it is similar to a machine code, and it is processed by an interpretive program, which can be considered as an IPL-V computer. IPL-V is a lower level language than a symbolic assembly is, in some respects. For example, IPL symbolic names consist of only one alphabetic character followed by up to four numbers, allowing very slight mnemonic possibilities. The power of the language lies in the large number of basic processes or subroutines which are available, processes which are useful in list-processing and in organizing and debugging a program. There are provisions for tracing, dumping, saving for restart, and dividing a program into more than one memory load.

Thirteen IPL-V implementations for 12 different computers are listed as available at the time of publication. Input-output conventions are peculiar to the different machine systems and are not covered in the IPL-V manual. They would have to be looked up before actual running of an IPL program. System questions, such as how to add new processes, are also peculiar to the particular implementation and are not covered in this manual.

NORA M. TAYLOR

Applied Mathematics Laboratory
David Taylor Model Basin
Washington, D. C.