more complicated but efficient algorithms; but these circumstances have so far been rare. The bottleneck problems present a different story; here they need one of the modified algorithms for the classical case, and if these have to be written, the little extra effort to cover both types of assignment could be faced.

References

GROSS, O. A. (1959). "The Bottleneck Assignment Problem," Rand Report P-1630.
KUHN, H. W. (1955). "The Hungarian Method for the Assignment Problem," *Nav. Res. Log. Quart.*, Vol. 2, p. 83.
KUHN, H. W. (1956). "Variants of the Hungarian Method for Assignment Problems," *Nav. Res. Log. Quart.*, Vol. 3, p. 253.
PAGE, E. S. (1963). "A note on Assignment Problems," *The Computer Journal*, Vol. 6, p. 241.
PORSCHING, T. A. (1963). "Matrix Assignments and an Associated Min Max Problem," *Math. Comp.*, Vol. 17, p. 81.
VAJDA, S. (1961). *Mathematical Programming*, Addison-Wesley: Reading, Mass.

Book Reviews

Contd. from p. 303

puter design. It is becoming increasingly important that means should be found for reducing the dependence of the performance of a system on that of its components. For various reasons, components may be inaccessible for replacement or repair, or the system must operate continuously, allowing no opportunity for maintenance. The idea of using some form of redundancy to achieve this end is by no means new, even in the computer field. It cannot be claimed that redundancy offers a technique where, by using enough equipment, one can achieve an arbitrarily high reliability. On the contrary, as is shown in one paper in this book, in certain by no means unlikely circumstances, by using enough equipment one can achieve an arbitrarily high unreliability. Nevertheless, there are fields of application where the various techniques may have a beneficial effect, and one of the achievements of this book is to begin to define these fields. One fact brought home by a reading of this book is that all too little is known of the statistical behaviour of large assemblies of components, how this behaviour is influenced by the properties of the components, and even the properties of the components themselves. If the publication of this book has the effect of stimulating research into these problems, it will have performed a great service to the designers and users of future generations of computers.

J. B. STRINGER.

Machine Independent Computer Programming, by MAURICE H. HALSTEAD, 1962; 267 pages. (Washington, D.C.: *Spartan Books*, $6.50.)

While we have increasing interest in and discussion of many varieties of mathematical and commercial source languages little has been published on the actual process of translation of a language to machine code, particularly in a manner that the student or non-specialist programmer can readily understand. Dr. Halstead has provided us with a primer of compiling technique, and we must be glad that he has been able to strip away the cabalistic overtones of the process so neatly.

The subject of this book is the Nelliac language and the minimum compiler required to compile itself in that language. Appendices illustrate actual compilers, expressed in the source language, to run on three different machines, and a Nelliac program for one of these that will accept absolute octal machine code as input and will therefrom produce Nelliac statements—a decompiler.

Nelliac is claimed to be a dialect of ALGOL 58; the latter was doubtless the inspiration of the Nelliac effort but the notation and format of the language differ so much that it is perhaps better considered a separate language. Remembering that the first rudimentary Nelliac compiler began running in February 1959 this difference is hardly a fault. Most of the notational flexibility of ALGOL in such matters as recursion, block structure, variable arrays, compound conditionals, formal parameters and functions has disappeared in Nelliac, deliberately, to leave a language suitable for single-pass compilation to absolute machine code. Unfortunately the resulting notation is unnecessarily difficult to read, and probably to write, accurately, particularly in the case of subroutines and conditional expressions. The required niceties of punctuation lack the redundancy required for easy comprehension.

The Nelliac language is more slanted towards data processing than either ALGOL or FORTRAN, and the user is more aware that he is using a computing machine. An example of the latter is the very useful implicit definition of the machine working store as an array in the language. This feature would seem necessary in a self-compiling compiler. The reader should note that the terms "function" and "procedure" have very different meanings in Nelliac from those more widely accepted through the influence of ALGOL.

The whole family of Nelliac compilers uses the technique of "generators", where the occurrence of given successions of symbols in the source text triggers the inclusion of predefined blocks of machine code. The book describes the exact coding for one computer; from the acquisition of individual characters of the source text from the input unit to the siting of the compiled object program in the working store of the computer, ready to run. Because of this ultimate transformation to absolute code the book also serves as a valuable handbook on single-pass assembly schemes.

The decompilation program included in the appendices completes the circle of source text to machine code to source text. Moreover, it is shown that programs originally written in machine code can be decompiled successfully, and it is claimed that this process can be a powerful aid in "debugging" machine code programs. The fact that any meaningful sequence of machine instructions can be converted mechanically to a meaningful sequence of Nelliac text on at least one machine would seem to call for some studies of the properties of those machine/language pairs for which this relationship holds.

It is unfortunate that whole-hearted recommendation of the content of this book must be tempered by concern at the low quality of the typesetting. Not only are there misprints, and footnotes that have wandered from their proper page, but here and there whole lines would appear to have been omitted by the printer.

H. D. BAECKER.