is varied. The user selects a point on the curve, inputs a corresponding parameter value to the system which then produces a mixed code working version.

3. *Normal running*

The inclusion of this as a phase in software production is deliberate: one should face the fact that, in the vast majority of cases, a substantial piece of software will still contain bugs and other deficiencies after it goes into productive use. It is here, perhaps, that the mixed code approach shows most promise. As interpretive code is comparatively rarely obeyed, one can afford to leave in some monitoring and debugging facilities in the production version without incurring excessive run-time penalties. Judging from our experience with MITEM, most of the major control paths of a program tend to remain in interpretive code, so that information obtained by monitoring interpretive code execution is much more complete than one might think. Such information could be used to monitor usage of different facilities of the software, tailor the code mix to actual usage and track down bugs when they occur.

Thus, as well as savings in storage requirements, the mixed code approach offers a good deal of promise in the area of debugging and monitoring. A further advantage may be in the area of segmentation or paging where the grouping of heavily used sections of code into a single segment has obvious advantages; in the MITEM project all direct code was, in fact, grouped together to simplify base register allocation.

References

CALDERBANK, V. J., and CALDERBANK, M. (1971). *LSD Manual* CLM-PDN 9/71, UKAEA Culham Laboratory, Abingdon, Berkshire.

NEWEY, M. C., POOLE, P. C., and WAITE, W. M. (1972). Abstract Machine Modelling to Produce Portable Software—a Review and Evaluation, *Software*, Vol. 2, pp. 107-136.

POOLE, P. C. (1971). *Draft MITEM Preliminary Users' Manual*, UKAEA Culham Laboratory, Abingdon, Berkshire.

WAITE, W. M. (1970). The Mobile Programming System: STAGE2, *CACM*, Vol. 13, pp. 415-421.

WAITE, W. M., and POOLE, P. C. (1972). Portability and Adaptability (printed lecture notes), *Advanced Course on Software Engineering*, Technical University of Munich, March 1972.

Book reviews

Sparse Matrices and their Applications, edited by D. L. Rose and R. A. Willoughby, 1972; 215 pages. (*Plenum Press*, $17.50)

This book consists of the proceedings of a symposium held at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York in September 1971. As is inevitable in a book of this kind, the individual contributions vary enormously in quality, and although the editors have written a long introduction in which they attempt to link the papers together into a coherent whole, there is really no disguising the fact that each has been written separately. Nevertheless it is a valuable book, principally because of its surveys of various aspects of the sparse matrix problem. Erisman explains how known zeros in the right-hand side vector b and a requirement for only part of the solution x may be exploited and he has a good compromise idea for avoiding the storage penalty of generating loop-free code and the execution speed penalty of working from integer arrays; Gustavson surveys basic techniques for the sparse matrix problem in a paper that is not altogether easy to read; Hachtel explains (perhaps not in as simple a way as he might) about the important idea of variability typing; Hellerman and Rarick explain their powerful 'partitioned preassigned pivot procedure' for factorising linear programming basis matrices; Tomlin explains about his use of LU factorisation for linear programming and gives some figures obtained from three large test examples showing performance much superior to that of the standard product-form algorithm; George considers elimination methods for matrices arising from finite-element discretisation of two-dimensional partial differential equations and has some sound practical advice as well as the rather surprising result that the equations arising from a $q \times q$ rectangular grid can be solved in $0(q^3)$ operations, although the high numerical factor means that q has to be very large for the gain to be substantial; Widlund gives a clear survey of direct methods of fast Fourier transform type; Cuthill surveys algorithms for reducing band-width, profile or wavefront.

The book is well produced with virtually no typographical errors. The references are collected together into an extensive bibliography with papers being ordered by year. Such a bibliography is valuable but (perhaps I am too conventional) I missed being able to see at a glance what papers each author referenced and could not find papers as easily as if they had been grouped alphabetically by authors' names.

J. K. REID (Harwell)

Introduction to Digital Computer Technology, by Louis Nashelsky 1973; 576 pages. (*John Wiley and Sons Ltd.*, £5·55)

This book is an enlarged and revised version (in hardback form) of the author's earlier text on 'Digital Computer Theory', and embraces (as did the previous book) the entire field of computer technology. Chapters are included on Number systems and codes, Machine language programming, Boolean algebra with combinational logic design, Computer circuits such as gates, bistables, counters, etc. and Computer units covering control, memory, arithmetic and input/output functions. The text is well written in a clear descriptive style with a large number of diagrams and photographs. Numerous worked examples are included in the text, and each chapter is concluded with a set of tutorial problems, the answers to selected problems being given at the end of the book.

The book is ideally suited for an introductory course in digital circuit techniques, but the treatment is too hardware orientated to be a good integrated text for computer systems engineers. For example, there is no mention of systems software or its effect on hardware structures as for instance in virtual memory systems.

Though the text is fundamentally sound, with remarkably few typographical errors, there are however a number of notable omissions. For instance, though a hardware approach is adopted, the use of read-only memory elements (ROM's) or any other complex integrated circuits is not mentioned. A more serious criticism is the lack of any discussion of interface requirements and their design; this is becoming one of the major activities of any digital systems engineer.

In fact the book has a somewhat dated flavour with its descriptions of discrete logic circuits, AC operated flip-flops, and the design of ripple counters using delays and feedback techniques. Modern methods of counter design, using sequential circuit theory and K-maps, are barely mentioned, and little attention is paid to circuit delays and hazards, and the application of synchronous counters with coherent outputs.

Irrespective of the somewhat critical comments, the book is nevertheless good value for money and is well worth recommending to students following an introductory course in digital techniques. Moreover, the presentation is such that non-specialists requiring a basic knowledge of computers would find the book extremely useful.

D. W. LEWIN (Uxbridge)