Edwards, C. R. (1975). The application of the Rademacher-Walsh transform to Boolean function classification and threshold-logic synthesis, *IEEE Trans*, Vol. C.24, pp. 48-62.

Edwards, C. R. (1978). Some improved designs for the digital-summation threshold-logic (DSTL) gate, *The Computer Journal*, Vol. 21, No. 1, pp. 73-78.

Eves, J. M. (1976). Multioutput digital summation threshold logic, B.Sc. Final Year project report, Ref. EE 12/76, School of Electrical Engineering, University of Bath.

Halligan, J. (1974). Using Majority logic blocks, *Electronic Equipment News*, Vol. 16, Nov. 1974, p. 105.

Hampel, D. (1973). Multifunction threshold gates, *IEEE Trans*, Vol. C.22, pp. 197-203.

Haring, D. R. and Diephuis, R. J. (1967). A realization procedure for multi-threshold threshold elements, *IEEE Trans*, Vol. EC.16, pp. 828-835.

Hurst. S. L. (1969). An introduction to threshold logic: a survey of present theory and practice, *The Radio and Electronic Engineer*, Vol. 37, pp. 339-351.

Hurst, S. L. (1973). Digital-summation threshold-logic gates: a new circuit element, *Proc. IEE*, Vol. 120, pp. 1301-1307.

Hurst, S. L. (1976). Application of multioutput threshold-logic gates to digital-network design, *Proc. IEE*, Vol. 123, pp. 297-306.

Hurst, S. L. (1977). *The logical processing of digital signals*, New York: Crane-Russak.

Hurst, S. L. and Edwards, C. R. (1976). Preliminary consideration of the design of combinatorial and sequential digital systems under symmetry methods, *International Journal of Electronics*, Vol. 40, No. 5, pp. 499-507.

Kautz, W. H. (1971). Programmable cellular logic, in *Recent Developments in Switching Theory*, ed. A. Mukhopadhay, Academic Press.

Muroga, S. B. (1971). *Threshold logic and its applications*, New York: Wiley Interscience.

Reddy, V. C. V. P. and Swamy, P. S. N. (1974). Note on digital-summation threshold-logic gates, *Proc. IEE*, Vol. 121, pp. 1085-1086.

Robinson, J. P. and Hoffner, C. W. (1975). Easily-tested three-level gate networks for *T* or more of *N* symmetric functions, *IEEE Trans*, Vol. C.24, pp. 331-335.

Tabloski, T. E. and Mowle, F. J. (1976). A numerical expansion technique and its applications to minimal multiplexer circuits, *IEEE Trans*, Vol. C.25, pp. 684-702.

Winn, G. C. E. (1975). A digital approach to the efficient synthesis of threshold gates, *The Computer Journal*, Vol. 18, pp. 239-242.

Wooley, B. A. and Baugh, C. R. (1974). An integrated *m*-out-of-*n* detection using threshold logic, *IEEE Trans*, Vol. SC. 9, pp. 297-306.



# Book reviews

*Software Metrics*, by Tom Gilb, 1977; 282 pages. (*Prentice-Hall*, £11·50)

This is a curate's egg of a book if ever there was one. Those who read Tom Gilb's articles in the computer press in late 1975 will already be familiar with some of the ideas to be found in this book, such as 'bebugging' (the deliberate seeding of errors in a program) and 'dual coding' (two versions of a program written completely independently).

Gilb's aim is the cost-effective design, construction and maintenance of software. He seeks to apply certain 'metrics' in order to monitor the quality of software. 'Bebugging', for instance, may be used to discover the total number of bugs in a program. You sow the program with a fixed number of bugs and set someone to find bugs. From among the bugs discovered you then find out how many were deliberately implanted. This gives you some idea of the number of bugs actually present before you started. It is, in fact, similar to the method of finding the total number of fish in a lake by introducing a certain number of tagged fish, and then going fishing.

Such a simplistic view of things does not, of course, take into account the fact that bugs (unlike fish) may be astonishingly different from one another in form. Two cards in the wrong order may be responsible for one bug, but another may be due to a mismatch in the interface between modules, and a third may only show up as an inadequate fix-up after a rare combination of invalid data. If you think that the author concerns himself with such awkward complications, you are wrong. He wants a metric, and therefore to him a bug is a bug is a bug.

In fact, sometimes even bigger corners are cut in the search for something which he can call a 'metric'. In Fig. 32 (p. 69) the metric corresponding to 'EFFECTIVENESS' is 'Transactions per cost-unit' (which is very reasonable), but the metric corresponding to 'ROBUSTNESS' is 'garbage in does not lead to G. out' which is hardly something one can measure (not to speak of the inadequacies in punctuation).

There are without doubt nuggets of wisdom hidden in this book, but the form of the book and its style tend to keep them hidden. Perhaps it is hardly surprising that one who disdains structured programming should produce a book with so little structure. The chapters are not numbered, and no page has a running heading, so one has to resort to a minute comparison of type sizes to determine when a new chapter is indeed starting. One half of the book (Part II) reads like a collection of notes and jottings. Many diagrams appear to be untouched reproductions of foils for an overhead projector.

As far as the style is concerned, the text seems to have been 'bebugged'. On p. 60, for instance, the 'Hawthorne effect' is mentioned twice, but the term is only explained on its second occurrence, not the first. P. 60 also refers to 'lines of source code of PLS (abbreviated as LOC in the figure)'. What is PLS? (The meaning of APL is given on p. 86, but that of PLS is nowhere explained.) And who in his right mind would use LOC as an abbreviation for PLS? After no small amount of research, it dawned that PLS is a language and LOC is short for 'lines of code'.

These are typical of the obstacles with which the reader must contend in order to dig out the nuggets. Perhaps what is needed is for someone else to take the information and to write it up in a well-structured, easy-to-understand manner. Surely it would be fitting to apply to Gilb's own work one of his own concepts—that of dual coding.

Colin Day (London)

*Digital Signal Processing in FORTRAN*, by F. Taylor and S. L. Smith, 1976; 402 pages. (*Lexington Books*, £13·25)

My first reaction to the title of this book was negative. Here was another title in the already overburdened market for books about FORTRAN and its applications. However, a closer study of the text showed that this was a much more interesting book than usual. It describes a comprehensive package, written by the authors (in FORTRAN) providing digital filtering algorithms within an easy to use, user-oriented, software system called SPECTRUM IV. Full details are given of the structure and use of the package, and each section is accompanied by a thorough treatment of the mathematical and information-theoretic background of the algorithms in question.

The software components include features for spectral analysis (fast Fourier transform, autocorrelation and power spectral density), bivariate spectral transforms, and various digital filters. These sections are clearly presented with examples of the use of the programs, and the graphical output produced.

On the whole, however, I feel the book falls between two stools. It contains too much background to be a simple manual of SPECTRUM IV, but the reader seeking to understand digital filtering techniques is confused by the descriptions of card-decks required for its use.

The package was written initially for the CDC 3100, but was subsequently transferred to an IBM 360. The authors generously offer copies for general use at nominal cost, and an application form is included in the back of the book.

S. J. Goldsack (London)