NOERR, P. L. and WHITE, M. C. (1976). Merlin, design of a national bibliographic database, in *Systems for Large Data Bases*, ed. by P. C. Lockemann and E. J. Neuhold, North-Holland, p. 211.

MCCUTCHEN, C. W. (1965). Random code numbers for universal identification of documents, *American Documentation*, Vol. 16, No. 2, pp. 91-96.

NUGENT, W. R. (1968). Compression word coding techniques for information retrieval, *Journal of Library Automation*, Vol. 1/4, pp. 250-260.

PLAISTER, J. and others. (1973). Conversion of a regional union catalogue into machine-readable form and its use as a basis for local catalogue conversion, *Laser*.

PULSIFER, J. S. (1977). The special problems of serials, *Library Trends*, January 1977, pp. 685-702.

RATHER, L. J. (1977). Exchange of bibliographic information in machine-readable form, *Library Trends*, January 1977, pp. 625-643.

SCHWARTZ, E. S. and KLEIBEOMER, A. J. (1967). A language element for compression coding, *Information and Control*, Vol. 10, pp. 315-333.

UNISIST, International Serials Data System (ISDS). (1973). *Guidelines for ISDS*, Unesco.

WILLIAMS, M. E. (1976). Data element statistics for the MARC II data base, *Journal of Library Automation*, Vol. 9/2, pp. 89-100.

WILLIAMS, M. E. and MACLAURY, K. D. (1977). *A state-wide union catalog feasibility study*, Final Report on Project III-FY, Information Retrieval Research Lab., University of Illinois, October 1977.

Book reviews

Logic and Data Bases edited by Herve Gallaire and Jack Minker, 1978; 454 pages. *Plenum*, £18·58

This book is a collection of 14 papers given at a conference in 1977. However it is rather better than the usual conference proceedings in that the authors have had a chance to revise their papers and achieve a high level of readability and to include comments on how they relate to other papers in the book. Also there is an excellent introduction and overview by the editors providing background material on logic and tying together the main points of the different papers under various themes.

Outside of researchers in logic the book is principally of interest to theorists and those wanting a clearer understanding of the nature of programs and data. If your predicate calculus is rusty you are advised to brush it up before reading the book and the authors suggest it as a graduate computer science text for a specialist option. However this is not to say that the book will not have applications and those interested in data bases should not be put off. They will have to forsake their favourite jargon of Schema, Subschema and DDL, and learn new words such as Horn Clause and Intensional Data Base. The effort though is worthwhile and it gives a fresh insight into the whole activity of programming and searching. There are stimulating papers by Tarnlund and by Kowalski which show that when we thought we were writing programs we were actually producing proof derivations! A paper by Pirotte shows how all the popular relational query languages can be modelled in first order logic and their deductive powers compared.

However it is too easy to take an Olympian view that there is no difference between programs and data; we have known this since LISP, but it is when you come to implement it on a real machine in reasonable time that different representations rear their ugly head. The paper by Futo, *et al* is a fascinating example of this. After initial experience they have produced an interactive system with 2 or 3 second response to queries on pesticides and their side effects in six man months. Their system uses an explicit data base of a few thousand records, and an implicit data base of a hundred or so general rules, and is written in the theorem-proving language PROLOG. However into the pure programming style have crept pseudo-predicates like FAIL in order to make a depth-first theorem prover behave breadth-first, rather reminiscent of the introduction of GOTO into LISP. Further, the clauses of the explicit data bases are held in a special representation on backing store with flags to speed up pattern matching.

Papers by Reiter, Kellog and others face up to this problem by providing a division of labour between the theorem prover which works on general axioms that are inefficient to store explicitly (e.g. '*x* sells objects if they are red or green') and which transforms the clauses in a complex query into more basic relational queries, and the relational data base software that can answer these queries efficiently. However papers by Minker, Kowalski and others disagree on this separation. Ultimately a reconciliation may be found along the lines of Tarnlund who formalises the search problem for binary trees and explicitly considers the efficiency problem, a point too long neglected by logicians.

There is not room here to review all the papers but if you are interested in logic or data bases this is a book worth buying for the library. It is significant that data base workers after long concentration on efficiency of data representations are now talking about 'models of the real world'. It is high time they talked with the logicians who have long been concerned with such problems and this book may serve to start the debate.

P. M. D. GRAY (Aberdeen)

The Logical Processing of Digital Signals by S. L. Hurst, 1979; 580 pages. (*Edward Arnold*, £27·50)

The great majority of logic design is at present done by intuitive methods with a little backing from mathematical logic and Boolean Algebra. It is far more an art than a science. It is, however, getting every day more difficult as the sophistication and complexity of logic devices increases on silicon chips. In the future we need new and better techniques which are rigorous and sufficiently well defined to be programmed. Many people have been working in this field and it has proved a recalcitrant one, full of dead ends and great difficulties.

Stan Hurst, the author of this book, is an engineer with an unusual mathematical flair and has devoted years of his time to exploring the less obvious techniques of logic design. He has now produced this invaluable book which must be an enormous help to everybody who is interested in the development of new logic design techniques. Most of us who read the technical literature find ourselves baffled by the technicalities of threshold logic, Chow parameters, spectral techniques and their like. There are masses of references to wade through and most of us have lost hope of ever catching up. Hurst has written a book, starting from first principles, which deftly guides the reader through these very complicated fields. He writes very much as a teacher should, carefully avoiding the kind of intellectual obstacles that can so easily stump a reader working on his own. The book is long, 575 pages of text, and covers very thoroughly the more recondite topics of threshold logic, logic function classification, matrix transforms and spectral techniques, in particular. Much of the theory of these topics is due to Hurst himself and his colleagues and there is nobody better qualified to write about them.

It is regrettable that such a master work cannot appeal to a great number of people and can provide easy reading to very few. It is a book, however, that everyone who is seriously concerned with the development of logic design, and every teacher of logic design, should have on an available bookshelf. Nobody can forecast what will be the future of logic design: it may well be that the roots of it are here.

B. S. WALKER (Reading)