machines with a large central machine of 56K bytes of memory and small machines at remote sites with only 16K bytes of memory. The small machines use the modified task management facilities.

The time taken to produce machine and system level software was considerably influenced by hardware problems and as a result was difficult to estimate, but the time taken to produce application software was quite predictable. It is possible to estimate the number of fields that have to be defined in an application system from the system specification. Given this number and assuming that an application system requires one application table for every fifteen fields and that it takes one and a half weeks to produce a table, it is possible to estimate the total production time. In practice this estimate is found to be very reasonable.

3.2. *Performance*

3.2.1. *Testing*

The modular nature of the software meant that, in the development stage, bugs were usually easy to trace. Consequently the machine and system software quickly achieved good reliability.

The application handler is written so that SPG programs can be 'single shotted', one instruction at a time. With this facility, an application table can be thoroughly tested in two to three days.

3.2.2. *Efficiency*

A system in SPG, using the application handler as an interpreter, imposes a heavier load on the central processing unit than would an equivalent system written in compiled code, but in the application programs that have been produced so far, the system has always been peripheral bound, typically requiring two to three times more disc than processor time. As the handlers are designed to deal with the peripherals in an efficient manner, it is unlikely that any alternative software system could obtain a better performance from the same hardware.

Acknowledgement

Thanks are due to Capital Cities Computer Centres Ltd. for permission to publish this paper.

Reference

KNUTH, D. E. (1968). The Art of Computer Programming, Vol. 1, Addison-Wesley.

Book review

Design Methods for Digital Systems, by J. Chinal, 1973; 506 pages. (*Springer-Verlag*, US$36.10)

This book is a 1973 translation of a 1967 French publication. Unfortunately the material was not brought up-to-date prior to translation and this results in a book which, though immensely detailed in places, reflects the state of digital circuit design ten years ago. Very rapid advances have been made since then and the digital system designer will therefore find this book disappointing and unrelated to his present needs.

The title is a misnomer since the book relates to digital circuits rather than systems. The first 10 chapters (300 pages) relate to set theory, codes, boolean algebra and combinational logic which are presented in considerable detail, but with a heavy mathematical overview which often tends to obscure rather than enlighten. Practical applications are minimal, logic families and their properties are dismissed in six pages with no coverage of switching times, propagation delays or noise immunity, nor mention of post-1965 developments in TTL, MSI, ROM & RAM devices. A notable omission also is the topic of static hazards, which is of fundamental importance and could readily have been dealt with in the section on Karnaugh mapping.

The sequential circuit material in the second half of the book (chapters 11-16) is introduced with magnetic cores and flip-flops. Asynchronous sequential circuits are considered only very briefly in an analysis of secondary races. Huffman's design procedure is not presented and consequently there are no design examples for asynchronous systems, neither is there an adequate coverage of essential hazards and the consequent reasons for adopting synchronous methods. The major part of the sequential material relates to synchronous circuits, and a number of synthesis procedures are considered. The 'direct' method commences with obscure mathematical expressions and it is not made clear how the circuits are derived from them. The state diagram method is the most useful, and some helpful examples on sequence detectors are included. The regular expression approach is given a great deal of coverage, and this is one of the very few books to treat this topic in any detail. There is no doubt that state diagrams for synchronous systems, particularly sequence detectors, can be derived by using regular expressions, but the mathematical effort involved is considerable, and the designer will probably prefer a more intuitive approach, direct to the state diagram.

The characteristic equations of clocked bistables are briefly derived, but no examples are given of how they may be applied to derive practical circuits from state tables, no mention either of master/slave operation or integrated circuit realisations, and no mention of the relationships between sequential and iterative combinational circuits, even though Hennie's work is quoted in the bibliography.

In summary, this book does not relate to digital systems but rather provides a very detailed coverage of the mathematical framework underlying digital logic circuits. The practical implementation of this material into design methods is disappointing, largely because the book was written before integrated circuits became generally available. The publishers must be criticised for marketing an expensive new translation of an existing work in such a rapidly developing field without allowing the author to bring his material up-to-date.

B. V. HOWARD (Edinburgh)