However for two or three channel systems, though the trend is about the same, the previous remarks (for one channel) are less valid.

As far as the throughput rate is concerned, we noticed that both methods give higher throughput rates throughout. Comparison of the throughput rates for these two methods led to similar conclusions to those of the CPU utilisation.

### 6.1 *Limitations*

Both methods suffer their limitations. If in the software method, for example, the block size used for a single I/O transfer with a disc is already as big as the maximum physical block size for this disc, then this method cannot be applied. Also, if the increase in the block size reduces the degree of multiprogramming by 25% or more, then this may well lead to a degradation in system performance. The hardware method suffers from similar limitations; if the discs already connected to the system are the most efficient available, then there is no hope of applying this method.

From the implementation point of view, the hardware method needs substantial funds. The software needs some reorganisation of the disc file store within the operating system, again an expensive and time consuming process, though this method, once achieved, can subsequently be implemented in many similar installations at a very low cost and may also need program changes.

## 7. Conclusions

For the environment studied, it has been shown that valuable gains in CPU utilisation and rate of throughput may be obtained by either increasing the I/O block size or by employing faster I/O channels. The performance of the system is dominated by the value of the job's characteristic $R(j)$. This characteristic can be improved by either method with valuable improvements in all cases. Even more gains can be expected if the circumstances permit a combination of the two methods to be applied. From the tabulated simulation results, it is clear that the replacement of an I/O channel by a faster one will not necessarily lead to an equivalent increase in system performance—similarly for the software method. Therefore the decision to replace an existing channel(s) by a faster one or even adding a new channel or applying the software method should, in practice, be based on some knowledge of the dominant factor $R(j)$ of the jobs running in the system.

**Acknowledgements**

I wish to express my gratitude to the referee for his valuable comments. Thanks are also due to Professor Peter Jarratt and J. R. Yandle of the University of Birmingham.

## Appendix; Proof of Eqn (5)

CPU utilisation is generally defined as

$$U_0 = \frac{C}{E} \tag{A1}$$

where $C$ is total CPU time and $E$ elapsed time. In a computer system with more than one I/O channel, we define the utilisation of the $i$th channel as

$$U_i = \frac{T_C}{E} \tag{A2}$$

where $T_C$ is the total I/O time for the $i$th channel. Combining Eqns (A1) and (A2) and rearranging gives:

$$U_i = U_0 \frac{T_C}{C} \tag{A3}$$

If, during the time the system was in operation (time $E$), there were $n$ I/O requests for all I/O channels and if the probability of an I/O request going to channel $i$ was $P_i$, then the total I/O time for this channel would be $T_C = nP_im_i$ where $m_i$ is the average service time for the channel. Similarly the total CPU time can be expressed as $C = n'm_0$ where $m_0$ is the average CPU time between I/O calls and $n'$ the number of CPU requests. Because $n$ and $n'$ are approximately equal, then ($C$) becomes

$$U_i = U_0 \frac{nP_im_i}{n'm_0} = U_0 \frac{P_im_i}{m_0}$$

**References**

Denning, P. J. and Buzen, J. P. (1978). The operational analysis of queuing network models, *ACM Computing Surveys*, Vol. 10 No. 3, pp. 225–261.

Hellerman, H. Smith, H. R. Jr (1970). Throughput analysis of some idealized input, output and compute overlap configurations, *Computing Surveys*, Vol. 2 No. 2.

Lazos, C. and Vafiadis, A. (1979). A method to estimate the I/O buffer size in a computer system, *The Computer Journal*, Vol. 22, No. 4, pp. 323–327.

Lazos, C. and Yandle, J. R. (1979). Improving CPU utilisation in a multiprogramming system, *The Computer Journal*, Vol. 22 No. 3, pp. 203–205.

Price, T. G. (1976). A comparison of queuing network models and measurements of a multiprogrammed computer system, *Performance Evaluation Review*, Vol. 5 No. 4, pp. 39–62.

# Book reviews

*Integrated Office Systems*, Edited by N. Naffah, 1980; 275 pages. (*North-Holland*, $41·50)

An international workshop was held in Versailles, November 1979. The topics under discussion were basic hardware design, distributed system control and architecture, ergonomics, organisation and social aspects, office modelling and networking. This book is a collection of the papers presented and the results of the working group sessions. It has been published as an aid to people doing research in office systems. However, it is not intended as an introduction to the topics discussed. It consists of a series of papers, grouped by topic, followed by the results of the working groups. There has been little attempt to present the information in an easily assimilated format. This is unfortunate as the book contains a great deal of potentially useful information. Anyone doing research in office systems will find this book well worth studying.

R. E. Evans (Coventry)

*Elementary Computer Programming in Fortran IV* Second Edition, by Boris W. Boguslavsky, 1980; 482 pages. (*Prentice Hall*, £9·70)

This book is an introductory text to learning FORTRAN IV on the IBM 360 computer with the IBM E, G or H compilers and the WATFIV compiler. The whole approach is very much tied to the use of IBM systems. Particular examples are the use of arithmetic operations to do bit manipulation, packing and unpacking, the DUMP and DUMP debugging aids, the use of hexadecimal numbers, and a 10 pp. appendix on the IBM 29 card punch.

The second edition has been simplified by the removal of chapters on graphics, magnetic tapes and discs. While much remains that is dated, the exercises are everlasting. For example, try printing the number of cycles per second produced by every key on a standard piano keyboard, or try normalising the hexadecimal number, B.OFF.

P. A. Clarke (Harpenden)