

must supply its ID as part of the message it sends on x . Thus, in this case, messages take the form: $\langle x, \text{sender's ID}, \text{message} \rangle$. The IDs are unique numbers generated by the compiler when mapping the logical network into the grid and are available to the programmer as read-only variables. By treating each pair $\langle x, \text{ID} \rangle$ as a new channel name, we can implement the shared channel, x , as N distinct private channels $\langle x, \text{ID}_1 \rangle, \dots, \langle x, \text{ID}_N \rangle$.

Process Dischead

Input Channel : Request, Release;

Output Channel : Grant[*];

Type Entry = **Record** upward: Boolean; cyl#: 0 .. max;
id: Integer **End**;

Var Q : **Queue** of Entry; (* ordered by first two fields and arranged as a circular list; $Q.\text{length}$ gives current length of Q *)

t, c : ↑Entry; (* c is a pointer to the head entry of Q and t is temp *)

Repeat

```

When INPUT(Request;  $i$ ; cyl) Do New( $t$ );
   $t\uparrow.\text{upward} :=$  If  $Q.\text{length} = 0$  Then True
    Else ( $c\uparrow.\text{upward}$  And  $c\uparrow.\text{cyl\#} \leftarrow \text{cyl}$ ) Or
      (Not  $c\uparrow.\text{upward}$  And  $c\uparrow.\text{cyl\#} > \text{cyl}$ );
   $t\uparrow.\text{cyl\#} :=$  If  $t\uparrow.\text{upward}$  Then cyl Else max-cyl;
   $t\uparrow.\text{id} := i$ ;

```

References

- ARNOLD, R. C. and PAGE, E. W. (1976). A Hierarchical Restructurable Multi-Microprocessor Architecture, Proceedings of the Third Annual Symposium on Computer Architecture, pp. 40-45.
- BRINCH HANSEN, P. (1975). The Programming Language Concurrent Pascal, *IEEE TSE* 1.2, pp. 199-207.
- BRINCH HANSEN, P. (1978). Distributed Processes: A Concurrent Programming Concept, *CACM*, Vol. 21 No. 11, pp. 934-941.
- DIJKSTRA, E. W. (1972). Hierarchical Ordering of Sequential Processes, In *Operating Systems Techniques*, Academic Press, New York.
- DIJKSTRA, E. W. (1975). Guarded Commands, Non-determinacy and Formal Derivation of Programs, *CACM*, Vol. 18 No. 8, pp. 453-457.
- HOARE, C. A. R. (1978). Communicating Sequential Processes, *CACM*, Vol. 21 No. 8, pp. 666-677.
- MAHJOUR, A. (1979). Analysis of Response Time in Real-Time Systems, Ph.D Thesis, Department of Computer Science, SUNY, Stony Brook.
- PARKE, N. (1975). Hierarchical Systems of Microprocessors, IEEE International Symposium on Circuits and Systems, Boston.
- QUAYNOR, N. (1977). A Multi Microprocessor Architecture for Level Structured Systems, Ph.D Thesis, Department of Computer Science, SUNY at Stony Brook.
- STROVINIK, E. F. (1978). Compilation Strategies for Multiprocessor Message Passing Systems, Seventh Texas Conf. on Computing Systems, Nov. 1978.
- SWAN, R. *et al.* (1976). The implementation of the CM* Multi-Microprocessor, Carnegie-Mellon University Technical Report.
- WALDEN, D. C. (1972). A System for Interprocess Communication in a Resource Sharing Computer Network, *CACM*, Vol. 15 No. 4.
- WIDDOES, L. C. Jr. (1976). The Minerva Multiprocessor, Proc. 3rd Annual Symposium on Computer Architecture, pp. 34-39.

Book review

Microprocessors: Theory and Applications by G. A. Streitmatter and Vito Fiore, 1979; 456 pages. (Reston, £11)

In the last two years a number of microprocessor texts have been published which aim at introducing readers to the concepts required for an understanding of microprocessors and their applications. The book by Streitmatter and Fiore follows a well established pattern for books in this field; a review of number systems, digital logic, microprocessor hardware and an introduction to software.

In the first three chapters of their book, Streitmatter and Fiore present a lucid introduction to microprocessor concepts and digital electronics. These subjects are treated in a general way where the text does not rely on one specific microprocessor family for examples. Following this general introduction Streitmatter and Fiore continue with chapters on 'instructions' and 'an introduction to software'. At this stage the text introduces the instruction sets for the Intel 8080 and the Motorola 6800 microprocessors. Next follow chapters on bus structure, I/O concepts and memory devices. Throughout these hardware chapters, examples of Intel and Motorola products are used to illustrate the text.

Insert Entry(t) into Q according to the ordering on the first two fields.

```

If  $Q.\text{length} = 1$  Then Begin  $c := t$ ; OUTPUT(Grant [ $c\uparrow.\text{id}$ ]) End;
End;

```

When INPUT(release) **Do**

```

   $t := c$ ;  $c :=$  next of  $c$ ; Delete Entry( $t$ ) from  $Q$ ;
  If  $Q.\text{length} > 0$  Then OUTPUT(Grant [ $c\uparrow.\text{id}$ ]);

```

End

Forever

End Dischead;

Similarly when a channel, y , has N receivers, the sender of a message on y must specify the ID of the intended receiver. In this case the sender must have had a previous communication with that receiver and obtained its ID. Here also channel y can be implemented as N distinct channels $\langle y, \text{ID}_1 \rangle, \dots, \langle y, \text{ID}_N \rangle$. At run time two additional checks are needed:

1. It is possible that the channel name specified in a communication may not exist as a result of wrong indexing. In this case a run time error must be returned to the user.
2. In the algorithms of the INPUT statement buffers for all channels $\langle \text{channel name}, \text{ID} \rangle$ must be checked and upon a match the sender's ID must be transferred to the corresponding input variable. These are relatively minor checks.

One further chapter is devoted to 'advanced software' which includes assembly code programming and program debugging using a simple monitor. The remaining sections of the book are dedicated to an indepth treatment of the operation of both the Intel 8080 and the Motorola 6800 microprocessors. The treatment in the latter chapters relies heavily on manufacturers material. Finally at the end of the book five appendices are provided which present reference tables covering number systems, ASCII code, available integrated circuits, standard logic symbols and instruction sets. The appendix on instruction sets is particularly long and presents details from the 8080 and 6800 manufacturers literature.

This book presents a solid introduction to microprocessors and their applications, concentrating on the Intel 8080 and Motorola 6800 devices. The text follows a treatment similar to the majority of books aimed at introducing readers to microprocessors. The book is well written and presented with a large number of clear diagrams. Students will find the questions at the end of each chapter helpful in testing their understanding of the concepts presented.

M. E. BRINSON (London)