Digital Systems Design

D. G. Wong

Digital Systems Design

D. G. Wong

School of Electrical Engineering
University of Sydney

Edward Arnold

First published in Great Britain 1985 by
Edward Arnold (Publishers) Ltd
41 Bedford Square
London WC1B 3DQ

Edward Arnold, 300 North Charles Street
Baltimore, Maryland, USA

Edward Arnold (Australia) Pty Ltd
80 Waverley Road
Caulfield East, Victoria 3145
Australia

British Library Cataloguing in Publication Data

Wong, D. G.
Digital systems design.
1. Digital electronics 2. Logic design
I. Title
621.3815'3 TK7868.D5

ISBN 0 7131 3539 5

Printed in Great Britain by
Thomson Litho Ltd, East Kilbride, Scotland

PREFACE

The responses to a 1983 survey questionnaire from Nobel Prize winners in Science support the view that 'Computers, artificial intelligence and robots will be the technologies that will most strongly affect people's lives in the next century'. The characteristics which these technologies have in common are shared by 'Digital Systems' which involve the acquisition, transmission, storage, processing and utilisation of digital information.

The very rapid advances made in the technology of very large scale integrated circuits have caused a greater emphasis to be placed on digital systems (especially computers) within the curricula of secondary schools, universities and institutes of tertiary education.

The design of digital systems has been a university field of endeavour (teaching and research) for several decades. There are many books which cover various aspects of design methodology. However, because of the advances in the technology, there have been significant changes to the approaches and techniques which designers are using to achieve their objectives. There is, therefore, a need for books which provide an up-to-date overview of the field as well as a presentation of the necessary fundamental theory with design examples suitable for a university course. It is hoped that this book will help to satisfy this need.

Chapter 1 (Introduction to Digital Systems) provides a comprehensive overview of digital systems design. A unique feature of the book is the 'top-down' order of presentation of the material in this overview. The chapter starts with a review of applications of digital systems and discusses concepts relating to systems structure. The characteristics of some important systems components are then presented. Design methodology is summarised and the chapter ends with a description of the logical behaviour of fundamental logic gates.

Chapter 2 (Combinational Circuits) presents the basic theory and tools (Boolean algebra, Karnaugh maps etc.) for the analysis, minimisation and design of combinational circuits. These tools lead to methods for handling the universal logic elements (NOR- and NAND-gates) and larger combinational building blocks such as parity generator/detectors, adders, decoders and magnitude comparators.

Chapter 3 (Sequential Circuits) describes the characteristics of different types of flip-flops. Methods are then developed for the analysis and design of circuits (containing flip-flops) such as shift-registers and counters. Finally a systematic method of the design of the state transition logic of clocked sequential circuits is described.

Chapter 4 (Circuits for Arithmetic Operations) describes the two's complement, fixed-point representation of binary numbers and introduces the IEEE Standard (Standard 754) for floating-point binary numbers. The chapter describes serial and parallel implementations of circuits for the addition, subtraction, multiplication and division of numbers represented in two's complement, fixed-point binary notation. These circuits include iterative arrays for multiplication and division. Finally a full development of the carry lookahead adder is presented together with an analysis of the SN74181 and SN74182 integrated circuits.

Chapter 5 (Timing and Control Circuits) discusses the basic concepts and special needs of timing and control circuits. A number of examples of both synchronous and asynchronous timing are given. Finally the detailed design of a simple pushbutton and switch controlled timing unit is described.

Chapter 6 (Programmable-Logic-Arrays) describes the organisation of field-programmable-logic-arrays (FPLAs) and field-programmable-logic-sequencers. The detailed design is given of an arithmetic processor capable of the four arithmetic operations of addition, subtraction, multiplication and division. The implementation of the control signals for this arithmetic processor using field-programmable-logic-arrays is then described.

A set of questions is given in each of the Chapters 2 to 6. The last question of Chapter 6 concerns the PLA implementation of control signals for an arithmetic processor capable of sixteen operations. The standard of this question is representative of the levels of understanding and achievement which students should reach after studying this book.

This book is based on detailed notes prepared for undergraduate and post-graduate courses on Digital Systems given by the School of Electrical Engineering of the University of Sydney. The notes cover a wide range of material from 'combinational circuits' to 'microprocessor-based systems'. All this material could not be included in a single text and large sections on the following topics have been omitted: high-speed multipliers, floating-point arithmetic circuits, simulation of digital logic, memories, stored-program computers, microprocessors, and microprocessor-based systems.

Most of the material for Chapters 2 to 6 has been developed from 1977 onwards. Improvements have been made from time to time, and these chapters should now be in a very suitable form for first and second courses in Digital Systems at the undergraduate level. The overview of digital systems presented in Chapter 1 was written in 1983 and hence contains an introduction to some relatively new concepts and techniques such as 'systolic arrays' and 'data-flow computers'.

The author was strongly influenced in the development of material for this book by Frankel's paper on the description of the LGP-30 computer (Frankel, S.P., 'The Logical Design of a Simple General Purpose Computer', *Trans. IRE*, PGEC, **EC-6**, No. 1, March 1957). In this paper, a detailed description of the LGP-30 computer was given in the form of Boolean equations in less than half a page of typescript. What is important in this approach is the understanding of the function performed by the logic corresponding to a single term in one equation or associated terms in a number of equations. This approach has been taken in this book. Sets of equations are given, for example, to define an 8-bit magnitude comparator, a 16-bit carry lookahead adder and an arithmetic processor capable of the four arithmetic operations of addition, subtraction, multiplication and division.

I would like to acknowledge the assistance given me throughout my career by Professor J. M. Bennett, Basser Department of Computer Science, University of Sydney, Professor H. K. Messerle, School of Electrical Engineering, University of Sydney and Professor M. W. Allen, School of Electrical Engineering and Computer Science, University of New South Wales. Their friendship and professional advice on numerous occasions has been greatly appreciated.

I would like to thank my colleagues and students for their interest, encouragement and assistance. I would like to thank Ms P. Attwater for her accurate typing of the manuscript and her invaluable assistance in producing this book. Ms N. Strevens typed some of the notes on which this book is based. Messrs C. H. Barratt, J. G. Rathmell, K. R. Rosolen and D. T. Thomson assisted with proof-reading. Ms M. S. Reed assisted with some of the line diagrams. Their help is gratefully acknowledged.

D. G. Wong
Sydney, Australia
1984

CONTENTS

1 INTRODUCTION TO DIGITAL SYSTEMS

An overview of digital systems covering applications, organisation, component characteristics, techniques and design methodology

1.1 INTRODUCTION

The main functions performed by digital systems are the acquisition, transmission, processing, storage and utilisation of digital information. The applications of such systems are diverse, and the many improvements in community services which they have provided have been accepted by society-at-large as a natural development of the electronic age of the twentieth century.

Domestic applications of digital systems are well known. They include digital watches and clocks, calculators, controllers for sewing machines, microwave ovens, washing machines and other electrical appliances, alarms and security systems, educational toys, video games, digital recordings, optical disks, videotex terminals and personal computers.

Commercial applications are also very familiar. At supermarkets digital techniques are used in product labelling, automatic product scanning for checkout and point-of-sales terminals for inventory control. Offices are being automated with the increasing use of word processors, electronic filing systems, data banks, electronic mailing systems and distributed computing systems interconnected by local area networks. In banking, access to central account files is possible from a large number of widely distributed bank terminals, and 'auto banking' is becoming a popular banking service. Computing systems are indispensible tools for modern accounting practices, and the computer generated invoice, statement of account, reminder notice and cheque are integral parts of today's 'computerised society'.

Digital systems have many applications in transport systems. World-wide, on-line reservation systems play a key role in the efficient operation of international carriers. 'Fault-tolerant' control computers are used for the control of high-performance trains, aircraft and space vehicles, and are essential components of some aircraft under development which have aerodynamically unstable air-frames. Air traffic control and navigation aids also rely on computers and digital techniques. To achieve optimum fuel economy, automobile manufacturers are utilising microprocessors with integrated sensors for monitoring and controlling the automobile engine and for providing improved dashboard displays.

The industrial applications of digital systems cover a very wide field. Automated factories make extensive use of industrial robots and numerically controlled machine tools. Industrial processes are controlled by 'direct digital control' computers which replace analogue controllers, and process 'optimisation' is often performed by computers at various levels in a hierarchical structure. In complex, interconnected systems, such as power utilities, the problem of system instability is of paramount concern, and on-line computers for 'security assessment' and control are often employed.

Many military systems for both defence and war are highly dependent on advanced technology. Many reported hostile engagements in recent years have clearly demonstrated the military advantages which can be gained by using sophisticated digital systems for weapons control. Systems for 'command, control and communications' are essential for gaining supremacy in any protracted engagement involving large numbers of opposing strategic weapons. Future systems will require micro-electronic devices with capabilities of several orders of magnitude greater than those in use today. Hence it is very likely that the technology of micro-electronic devices will continue to advance. Industrial and commercial applications of these devices will follow in due course.

Communications systems are making increasing use of digital techniques. Many switched communications networks are designed to handle digitally-encoded speech, and high data rates are available from transmission media such as fibre-optic cable. Advances in communications technology will certainly improve services such as world-wide voice, data, telex, facsimile, television, tele-conferencing, videotex and distributed computing.

The processing of numerical data has been the traditional area of application of digital computers. The need to solve large sets of non-linear algebraic and differential equations, e.g. for the improved simulation of complex dynamical systems for nuclear engineering and weather forecasting, has stimulated the development of 'super-computers'. Higher performance has been achieved not only by using improved circuits and packaging techniques but also by innovative features which exploit a high degree of hardware 'parallelism'.

The computer processing of pictorial data has been used in many applications including space exploration, weather prediction, mineral surveying, military surveillance and medical tomography. The requirements of image processing have led to novel solutions, some of which have utilised recent advances in very-large-scale-integration.

Multi-processor systems containing high-performance microprocessors are providing very cost-effective solutions to many computational problems. Microprocessor-based systems in general are finding applications which affect many facets of our everyday life. Some of these have profound social implications.

1.2 THE ORGANISATION OF DIGITAL SYSTEMS

1.2.1 Components and functions

Digital systems contain components which perform the functions of:

(i) acquisition of new or previously recorded data,
(ii) transmission of digital information from one component to another,
(iii) processing of numerical and non-numerical data for data reduction and for management of system resources,
(iv) storage of data, and
(v) utilisation of old or computed data.

The organisation shown in Fig. 1.2.1 is representative of the following broad classes of digital systems:

(i) a large computer system with components structured around a *high-speed communication channel*,
(ii) a distributed system based on a *local area network*,
(iii) a *bus-structured* computer system,
(iv) a *switched communication network*, and
(v) a dedicated node computer.

The '*front-end computers*' of the large computer system provide an interface between the various input–output peripherals including user terminals and the powerful '*worker computers*' which take the form of large *main-frame* computers whose resources are managed by a *multi-processor operating system*. A front-end computer also handles the *communications protocols* pertaining to the switched communications network which allow the processing power of the large computer system to be distributed to a large number of remote user terminals.

A dedicated, high-speed communications link between the large computer system and a *node computer* is controlled by '*communications interface units*' (shown as CIUs in Fig. 1.2.1). This provides an effective method of handling a concentration of user terminals.

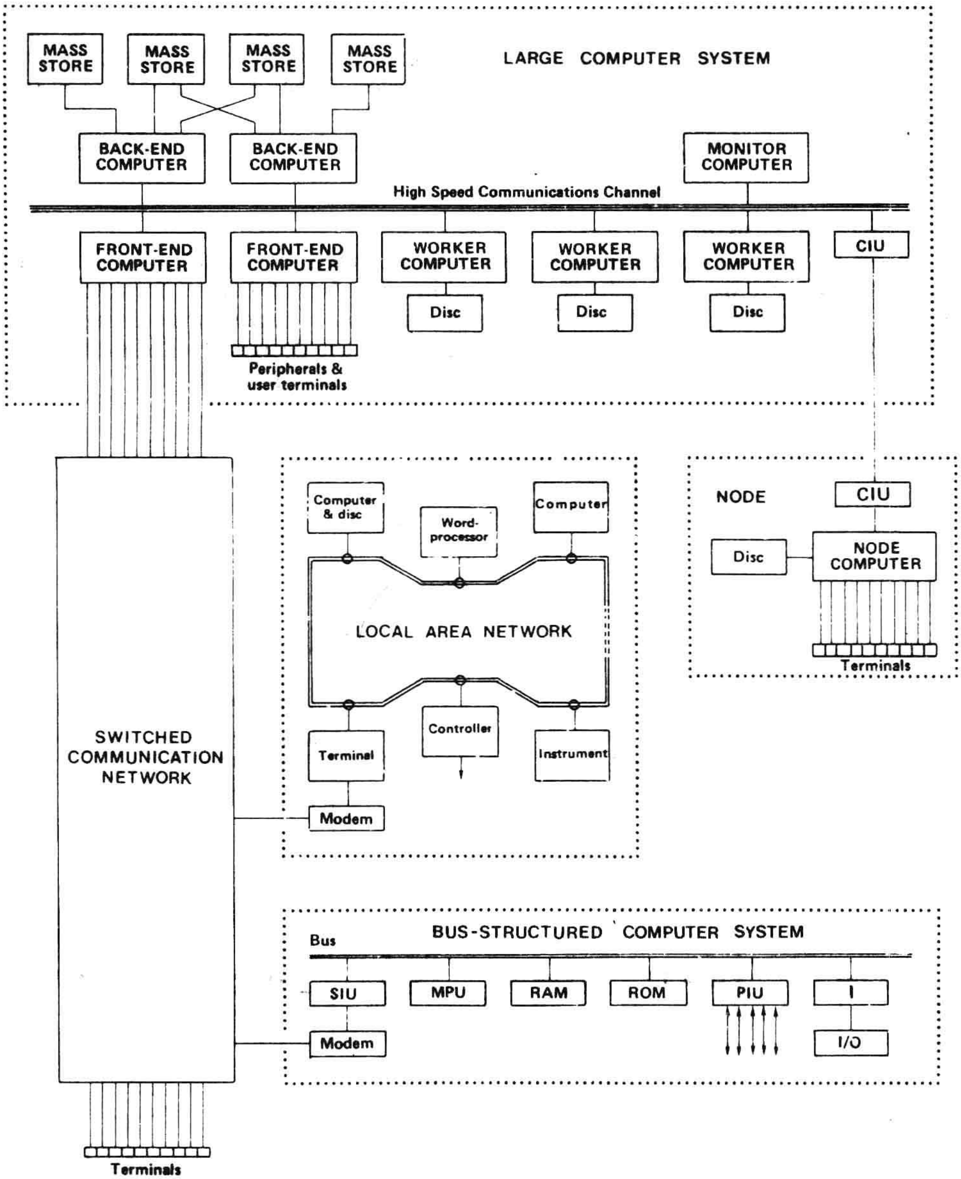

Fig. 1.2.1 Organisation of some digital systems

The '*back-end computers*' of Fig. 1.2.1 manage the expensive mass storage units which are shared by all worker computers. The integrity of the data-bases held in the mass storage units is guaranteed by a duplication of back-end computers and some mass storage units.

The '*local area network*' shown in part of Fig. 1.2.1 provides an effective means of communication between components of a computer system distributed throughout a factory, a large office, a building, a research establishment etc. The components connected to the local area network include word processors, personal computers, discs, data-acquisition systems, instruments, controllers, microprocessor development systems, etc.

The '*bus-structured computer system*' represented in Fig. 1.2.1 typifies a small *micro-*

processor-based system in which all components are interconnected via standard '*data*', '*address*' and '*control*' lines of a system *bus*. The components which can be so connected include a microprocessor unit (MPU), random-access memory (RAM), read-only memory (ROM), serial interface unit (SIU), parallel interface unit (PIU) and a wide variety of peripheral input–output units (I/O) such as keyboards, visual display units, floppy discs, etc.

1.2.2 Hierarchical structure

A hierarchical structure is an effective approach to the efficient organisation of complex systems. In digital systems, hierarchy is exploited in a number of distinct areas such as in *multi-level control* of complex industrial processes, in *memory organisation* using devices of widely different costs, access times and capacities, and the design of VLSI (very-large-scale-integration) systems where complexity management is important.

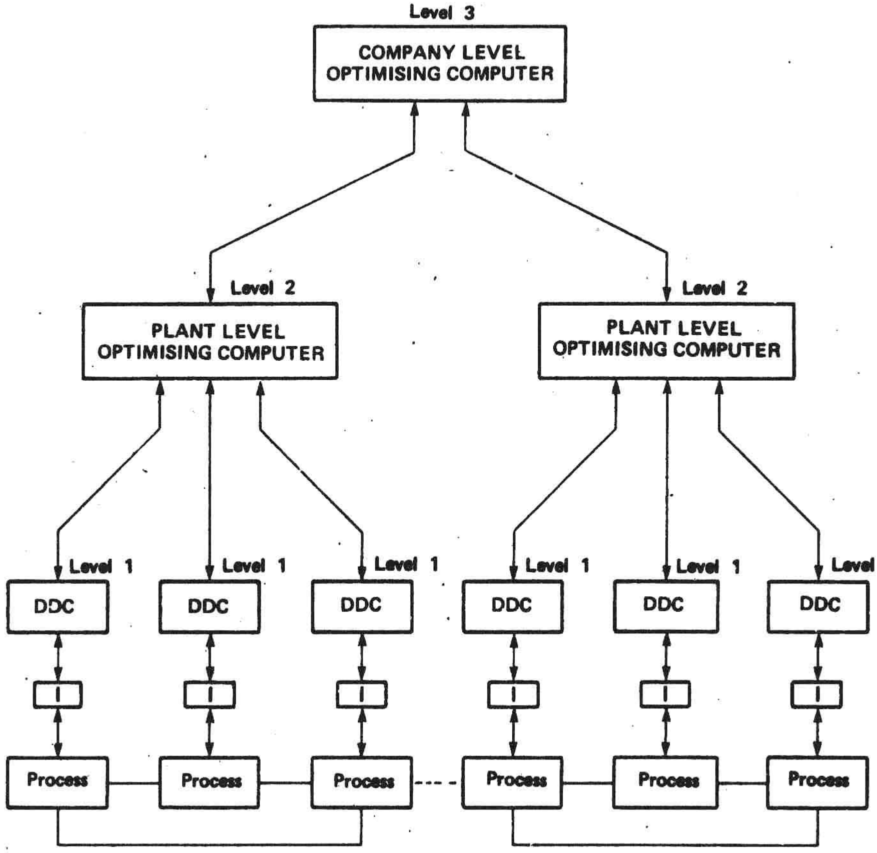
Level 3
COMPANY LEVEL OPTIMISING COMPUTER
Level 2
PLANT LEVEL OPTIMISING COMPUTER
Level 2
PLANT LEVEL OPTIMISING COMPUTER
Level 1
DDC
I
Process

Fig. 1.2.2 Block diagram of hierarchical computer control system (DDC = direct digital control computer, I = interface)

Some features of a hierarchical structure will be described in terms of Fig. 1.2.2 which is a block diagram of a *hierarchical computer control system*. In this diagram, the word 'Process' represents the different industrial processes in a complex of interconnected processes run by a large company. The symbol DDC refers to a *direct-digital-control* computer which performs the *real-time control* of a process. The symbol I refers to the *interface* between a DDC computer and a process. This interface would normally consist of sensors (e.g. for temperature, pressure or flow), analogue-to-digital (A/D) converters, parallel interface units, digital-to-analogue (D/A) converters and process actuators. Each DDC computer is capable of performing the function of a number of analogue controllers on a time-shared basis. This results in improved, centralised control. However, hardware and software reliability is very important as the failure of a DDC computer would result in the closure of a complete process.

Fig. 1.2.2 refers to the hierarchical computer control system of a large company which operates many plants consisting of interconnected processes. The computers of the control system are arranged in several levels. The DDC computers (in Level 1) handle the real-time (second-by-second) monitoring and control of every process. The Level 2 computers have a *supervisory function* as well as performing *optimisation* at the plant level. Level 3 computers perform optimisation at the company level.

Several features of the hierarchical computer control system of Fig. 1.2.2 deserve some comment. Firstly, communication between computers is essentially between computers in adjacent levels of the hierarchy. The lowest level (DDC) computers handle large amounts of real-time data but only the modest amount of data relevant to the overall performance of each process is transmitted between the DDC computers and computers in the next higher level. Secondly a larger proportion of the company-wide operations is handled by a computer which is higher in the hierarchy. Finally a larger '*component size*' is used for sub-system optimisation by a computer which is higher in the hierarchy.

1.2.3 Structure for a real-time application

Real-time systems have *response times* which are '*time-critical*'. By implication they are *on-line* systems which receive stimuli from an environment and respond effectively to these stimuli within a time period which is less than a specified response time. Response times vary from application to application. In an airline reservation system, a satisfactory response time measured from the time an enquiry is keyed in by an operator to the time the required information is displayed on a visual display unit would be about a second. Many industrial processes have '*time constants*' varying from fractions of a second to many minutes. Suitable response times of a computer system which controls these processes vary from milliseconds to many seconds.

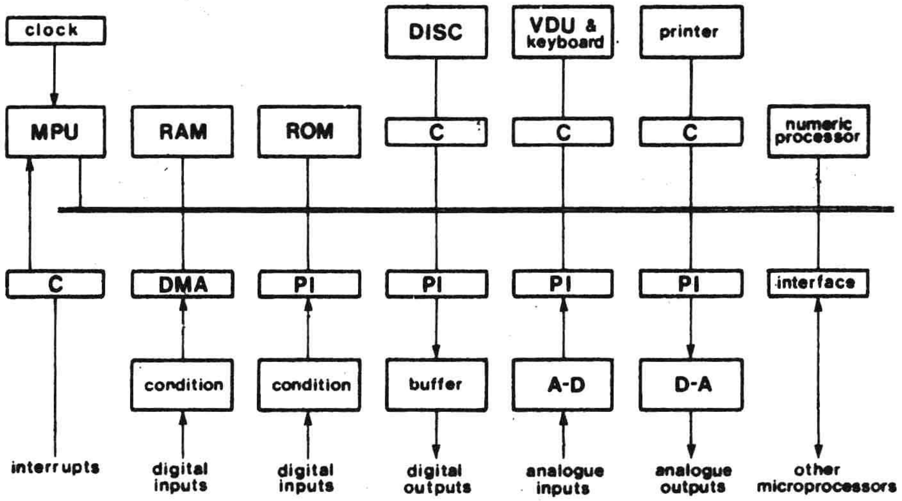

Fig. 1.2.3 Microprocessor-based system for data-acquisition and control (A–D = analogue-digital converter, D–A = digital-analogue converter, DMA = direct-memory-access channel, C = controller, PI = peripheral interface)

An example of a real-time system is the microprocessor-based system for data-acquisition and control shown in Fig. 1.2.3. The inputs to the system may be in either digital or analogue form. Digital input signals may require '*conditioning*' to transform voltages to levels acceptable to the microprocessor-based system. Analogue (voltage) inputs are converted to digital form by analogue-to-digital converters. The digital and analogue information acquired by the system is stored in the random-access memory and perhaps on disc. Output of this information may be via a visual display unit (VDU) or a printer.

The *control* function of the system of Fig. 1.2.3 is achieved by computing *feedback command signals* which take the form of digital outputs and analogue outputs from the microprocessor-based system. Digital outputs are *buffered* to provide the necessary electrical driving capability for relays, circuit breakers, etc. The analogue outputs produced by the digital-to-analogue converters are used by voltage-controlled actuators which control the process.

The '*direct-memory-access*' (DMA) channel provides a high-speed data-collection facility with data flowing from the monitored process directly to the random-access memory under the direction of the DMA controller and not the microprocessor (MPU).

The '*interrupt*' hardware provides a continuous monitoring of asynchronous events which may occur at unpredictable times but require action by the microprocessor within pre-defined time limits. *A priority interrupt* mechanism ensures that an interrupt signal corresponding to a more time-critical event is serviced with a response time which is less than that for a less time-critical (i.e. more patient) event.

The '*numeric processor*' (a 'co-processor' of the MPU) can perform arithmetic operations on *floating-point numbers* very rapidly, and hence significantly enhances the computational capability of the system.

1.2.4 Parallel structures

Multi-processing systems (i.e. systems involving concurrent processing by more than one processor) are structured to provide high performance. Examples of the functional partitioning of a system (e.g. front-end processor, back-end processor, worker computer, numeric processor, DMA channel) have been given in earlier sections. Many configurations of multi-processor systems have been used very successfully in the past, and with the availability of cheap, high-performance microprocessors and automated VLSI design procedures, interest in this area will certainly continue.

Processors in a multi-processing system may be *loosely coupled* or *tightly coupled*. Attributes of a loosely coupled system include geographically distributed processors and relatively low-speed communication links between processors. The objectives of interconnecting processors in such a system include the sharing of resources, the sharing of work load, the remote access to data-bases, data files and programs, improved processing availability and improved co-ordination of geographically distributed users contributing to a common computer project.

The attributes of a tightly coupled system include physically close processors and high-speed communication between processors. The objectives of designing such a system include high performance through *parallelism*, *fault-tolerance* and flexibility through a *reconfigurable* structure.

There are many problems, such as the real-time simulation of 3-dimensional systems defined by partial differential equations, which require computer capabilities far in excess of those of the fastest commercial computers. The architecture of the *super-computers* required to solve these problems is being actively studied. Many proposals are based on arrays of tightly coupled processors. The *interconnection networks* required to couple the processors take many forms such as *cross-point networks*, *ring networks*, *banyan networks*, *cube-connected networks*, *shuffle-exchange networks*, *tree-structured networks* and *systolic arrays*.

A simplified diagram representing the architecture of a super-computer based on tightly coupled processors is given in Fig. 1.2.4. In this diagram P represents a processor and M a memory unit. The concurrent operation of all processors is coordinated by the 'controller' (a special-purpose high-speed processor), and all elements of the system are interconnected via a high-speed interconnection network. The complexity of this interconnection network and its associated cost are very high. Hence compromises between *hardware complexity* and *generality of interconnections* are often made to provide cost-effective solutions.

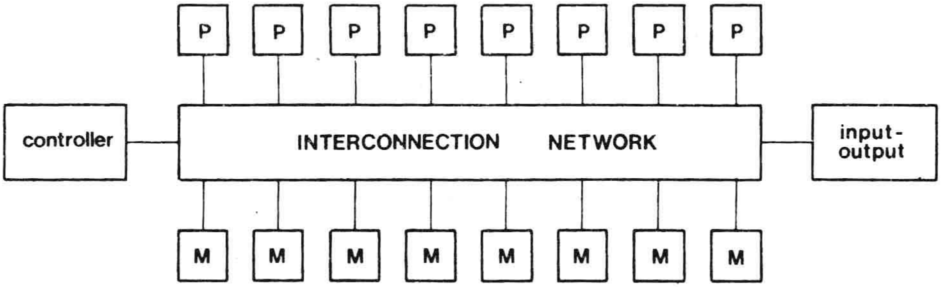

Fig. 1.2.4 Multiprocessor architecture for superfast computer (P = processor, M = memory unit)

The interconnection network which provides the highest generality of interconnection is the *cross-point switch*. This method of interconnection is shown in Fig. 1.2.5(a) for the case of eight processors (designated P) and eight memory units (designated M). Each switch of the network (designated by unlabelled squares) must be designed so that *any processor* may be connected to *any memory unit* and that *any combination* of the eight links between a processor and a memory unit is possible *simultaneously*.

The switches of the *banyan network* of Fig. 1.2.5(b) are shown with a *spread* of two and a *fan-out* of two. With this arrangement, any of the eight processors may be connected to any of the eight memory units. However every combination of links is not possible simultaneously. The banyan network has therefore a lower generality of interconnection than the cross-point switch. It has the advantage of less complexity and hence less cost. For n processors and n memory units the complexity of a cross-point switch varies as n^2 while that for a banyan network like Fig. 1.2.5(b) varies as $n \log n$.

An interconnection network which has very low generality of interconnection (i.e. high specialisation) is the *systolic array*. This method of interconnection becomes very cost-effective for specialised tasks. The concept of a systolic array is described by example in the next section.

Multi-microprocessor systems may be configured in many ways. Two of these are shown in Fig. 1.2.6 and Fig. 1.2.7. In the *shared memory* system of Fig. 1.2.6, all microprocessors communicate with one another by transmitting computed intermediate results to a shared memory. This memory is connected to the bus of one of the microprocessors called the *master* and it is accessed using an *address space* which has *global* significance, i.e. the *numeric addresses* used have the same meaning to every microprocessor in the system. For much of the time all microprocessors operate concurrently, and it is only when a microprocessor other than the master wishes to access the shared memory that *contention* of the bus associated with the master needs to be resolved by the master *bus arbitration hardware*.

In the multi-microprocessor system of Fig. 1.2.7, inter-processor communication is achieved via small random-access memories with *dual ports*. One port of each 2-port RAM

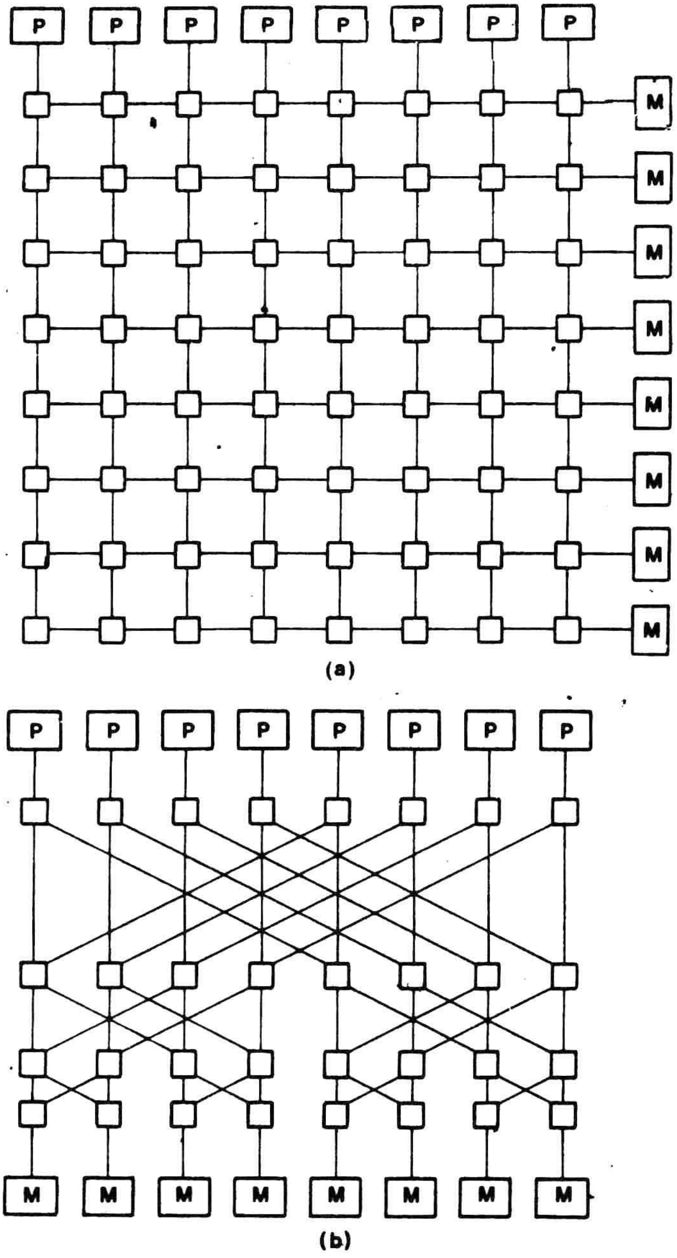

Fig. 1.2.5 Interconnection networks—(a)cross-point switch, (b) banyan network

is connected to the system bus and the second port is used exclusively by the associated microprocessor. An addressing scheme is used to allow any microprocessor to write-to or read-from any 2-port RAM. Again, for much of the time all microprocessors operate concurrently, and it is only when two or more microprocessors wish to use the bus simultaneously to access other 2-port RAMs that bus contention needs to be resolved by the arbitration hardware.

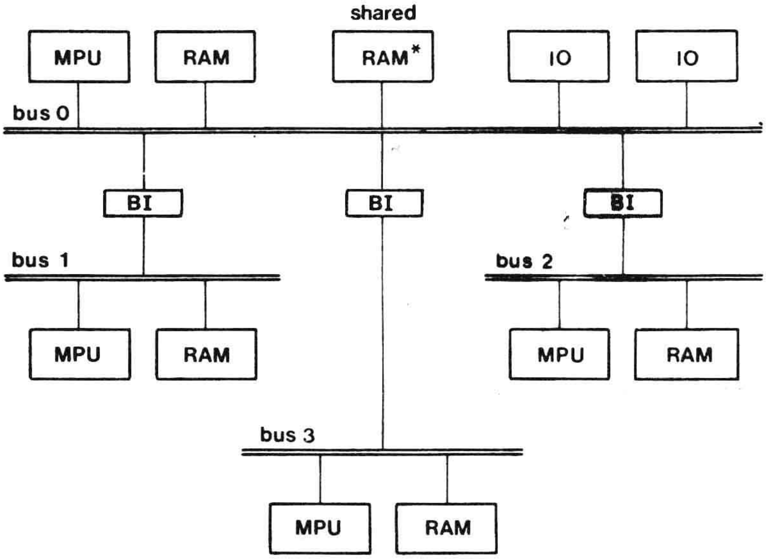

Fig. 1.2.6 Organisation of a multi-processor system with a shared memory (MPU = processor, RAM = local memory, RAM* = shared memory, BI = bus interface, IO = input-output peripheral)

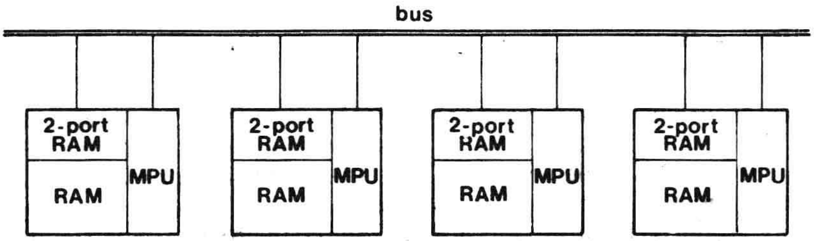

Fig. 1.2.7 Multi-microprocessor system in which any microprocessor can write into a (small) part of memory of any microprocessor connected to the bus

1.2.5 A systolic array

A *systolic array* is an array of special-purpose processors with a communication sti ucture tailored to a specific class of problem. The structure is such that each processor communicates with only those processors which are its *nearest neighbour*, and for most of the time data required by each processor is provided by the processor's nearest neighbours. The input data and computed result of each processor are *latched*. This allows a *simultaneous* transmission of data and computed results between processors. The operation of every processor in the array is *synchronised* by a system clock, and during each clock period, each processor receives input data and computes an output result. All processors in the array '*pulsate*' with activity synchronously, and hence the use of the term *systolic*.

A processor used in systolic arrays designed for operations on *vectors* and *matrices* is represented by Fig. 1.2.8. This processor has been called an '*inner product step processor*' as it has been structured to perform the inner product step utilised by algorithms for operations on vectors and matrices. The processor has three inputs, A_{IN}, B_{IN} and C_{IN} with

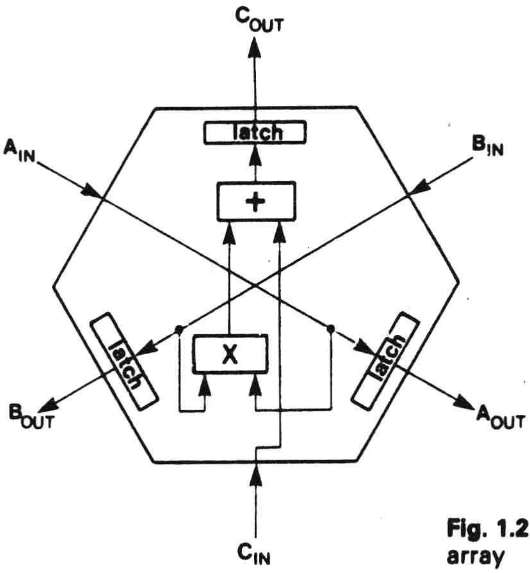

Fig. 1.2.8 Inner product step processor of systolic array

each input representing a number. The three outputs A_{OUT}, B_{OUT} and C_{OUT} are latched, and are related to the inputs by:

$$A_{OUT} = A_{IN}$$
$$B_{OUT} = B_{IN}$$
$$C_{OUT} = A_{IN} \times B_{IN} + C_{IN}$$

A systolic array which produces the product $C = (c_{ij})$ of two 3 × 3 matrices $A = (a_{ij})$ and $B = (b_{ij})$ is shown in Fig. 1.2.9. The array consists of nineteen *inner product step processors* defined by Fig. 1.2.8. The elements of matrix A are applied to processors 1, 2, 4, 9 and 14; the elements of matrix B are applied to processors 1, 3, 6, 11 and 16; and the elements of matrix C are produced by processors 4, 2, 1, 3 and 6. In order to understand how data and computed results flow through the array, each hexagon representing a processor in Fig. 1.2.9 is shown with a table of three columns and eleven rows. The three columns represent the three outputs A_{OUT}, C_{OUT} and B_{OUT} of the processor. The eleven rows represent eleven clock periods numbered 1 to 11 from top to bottom of the table. Blank entries represent zeros. Before clock period 1, all latches are cleared (i.e. set to zero).

The well-known product expansion for (c_{ij}) is reproduced below to facilitate the understanding of Fig. 1.2.9:

$$c_{11} = a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31}$$
$$c_{12} = a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32}$$
$$c_{13} = a_{11}b_{13} + a_{12}b_{23} + a_{13}b_{33}$$

$$c_{21} = a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31}$$
$$c_{22} = a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32}$$
$$c_{23} = a_{21}b_{13} + a_{22}b_{23} + a_{23}b_{33}$$

$$c_{31} = a_{31}b_{11} + a_{32}b_{21} + a_{33}b_{31}$$
$$c_{32} = a_{31}b_{12} + a_{32}b_{22} + a_{33}b_{32}$$
$$c_{33} = a_{31}b_{13} + a_{32}b_{23} + a_{33}b_{33}$$

In clock period 1, a_{11} is applied to processor 4 and b_{11} to processor 6. In clock period 2, these elements flow on to processors 7 and 8 respectively, and in clock period 3, they are