Introduction to Computer Architecture

Neil Willis
and Jon Kerridge

Introduction to Computer Architecture

Neil Willis
and **Jon Kerridge**

Department of Computer Studies
Sheffield City Polytechnic

Pitman

Preface

The aim of this book is to provide a student text which presents the ideas and concepts of computer architecture in a well organized and straightforward manner. It develops the fundamental concepts from the register transfer level upwards, not concerning itself with the logic level at all.

It does not assume any prerequisite knowledge of computers although we feel that the material is best taught in parallel with a course in assembler level or machine code programming. Indeed, the assembler level programming might well form the practical or tutorial part of such a course. Such practical sessions will certainly reinforce the understanding of Chapters 4, 5, 6, 7 and 8.

The choice of an illustrative computer for a text such as this is a much debated topic. Our basic premise is that to use examples of real machines is ideal and consequently examples are taken throughout the text from a variety of machines such as an IBM 370, PDP-11, ICL 2900, Hewlett Packard 2000 and HP 3000, Prime 750, Harris 800 and Motorola M6800.

These examples will complement the lecturer's illustrations of facilities available on the computer being used for the practical work

However, if it is not possible to illustrate a particular technique on the computer available, Appendix 2 of the text describes a computer system simulator This is a software package written in Pascal and is available to lecturers who would like to use it, at a nominal materials charge. Some of the problems in the text relate to the use of the simulator although they can obviously be solved on a real computer.

The general structure of the book is as follows. Chapter 1 introduces all the basic concepts of computing systems giving definitions of processors, memory, peripherals, backing store, communications and distributed systems. Because a distributed system is a collection of single processors, this topic is discussed further in Chapter 9 after the details of a single processor system have been described

Chapter 2 introduces descriptions of basic peripheral devices and gives the reader an appreciation of the physical characteristics, relative costs and flexibility of a wide range of peripheral devices.

Chapter 3 describes a range of backing store devices such as magnetic disc, drum, tape, floppy disc, bubble store, charge couple devices, content addressable file store.

Chapter 4 provides a functional description of memory together with a brief description of the technologies used to implement memory. The functional description includes interleaving and associative techniques as well as the Von Neumann structure.

Chapter 5 gives an outline of the structure and requirements of machine code and a review of addressing techniques

Chapter 6 is a functional description of a processing unit including synchronous and asynchronous operation Discussion of information flow in the processing unit leads naturally to the requirements for micro instructions and to descriptions of hardware and software micro instruction techniques.

Chapter 7 introduces the need for interrupts and discusses situations in which they may occur A review of interrupt handling techniques follows.

Chapter 8 is concerned with the transfer of information to and from peripheral devices. As well as input/output to 'close' devices the chapter includes a review of data transmission techniques.

Chapter 9 shows how computers may be linked to form a distributed system, from both the hardware and software aspects

Chapter 10 describes architectural developments which have been used to increase processor throughput.

Appendix 1 is a brief description of binary, octal, decimal and hexadecimal number systems. It describes the representation of characters and fixed and floating point numbers.

Appendix 2 is a specification of a computer system simulator together with a number of systems which can be used for teaching purposes.

This book has been based on a lecture course forming part of a BSc in Computer Studies and much of the text and many of the problems have been 'class tested'. We are very grateful to many students who have commented on their difficulties and for suggestions they have made for improvements to the material.

We also wish to express our thanks to all the people who have helped during the preparation of this book. In particular we are very grateful to Dan Simpson for his contribution of Chapter 9 and to Doug Bell for his patient reading of the entire text and his detailed and very helpful comments and suggestions.

Finally we wish to thank Mrs J M Kerridge and Mrs M Wilson for their very careful typing.

Neil Willis
Jon Kerridge
Sheffield, 1982

Contents

1 Basic concepts of computer systems

Computer systems are constructed from many different component parts and by connecting these parts together in different ways computer systems with different facilities can be built. However, there is within this variation much that is common. This chapter introduces you to the basic concepts that are common to all computing machines. Having acquired this fundamental knowledge you will then be able to go on and expand upon the framework provided.

1.1 Computing machines and computer systems

The first computing machines were built in the late 1940s. In retrospect they were somewhat limited in that they were designed to solve one particular problem or a small class of problems. Throughout the 1950s and 1960s developments were made which allowed the same computing machine to be used for many different classes of problem. During this time there were technological changes from the valve to the transistor which allowed more complex and more reliable computing machines to be built. It was also during this period that software was developed which opened up the use of the computer to the less specialized users. This was achieved by using programming languages which did not require the user to have intimate knowledge of the construction of the computing machine. There were also developed software systems which made the computing machine more flexible and this gave rise to the term computer system.

During the 1970s there were large developments in the electronics technology industry. These developments enabled the price of computing machines to fall dramatically. Together this meant that it was possible to build computer systems which were constructed from more than one computing machine.

1.1.1 *What is a computing machine?*

A computing machine is a piece of electronic equipment which, when given some data will process that data in some pre-defined way to produce the required results. It is fundamental to the concept that the operations which

take place do so in a finite time. A computing machine requires some means of accepting data, somewhere to store the sequence of operations to be carried out, some means of performing the calculations required and finally a method of communicating the results. These facilities are common to any computing machine regardless of the task to which it will be put.

1.1.2 *The difference between computing machines and computer systems*

A computer system takes the basic computing machine and by the addition of software makes the computer system more flexible so that it can be used to solve many different types of problem.

In the simplest case there is a one to one correspondence between a computing machine and a computer system. That is, the computer system uses only one computing machine.

However, the tendency is to develop computer systems that consist of several computing machines. There are two important reasons for this. Firstly, the nature of the environment—may best be supported by providing many localized computing machines—which also provide flexibility by communicating with each other. Secondly, there may be a security aspect which requires certain information held by the computer system to be maintained separate from other parts of the system. Such systems are called distributed systems. An example of such an environment is provided by a medical information system for a hospital. Each ward requires rapid information about patients on that ward but when a patient is moved within the hospital the information re ating to that patient should still be available. Also by distributing patient information it is inherently more secure.

These variations of requirement arise, for example, when many computing machines form a computer system but where some of the machines are general purpose and others are dedicated machines.

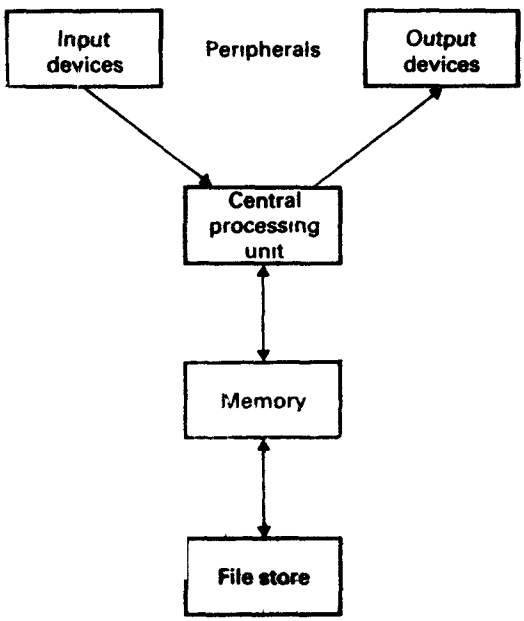
A general purpose computing machine is one which can be used to solve many types of problem. A dedicated computer is one used for one task only: for example, controlling a process in manufacturing industry. In a computer system constructed from such components the general purpose computing machines could be used to extract information maintained by the dedicated machine.

A dedicated computing machine is the same as a general purpose machine in its fundamental construction. It is only dedicated by the connections made between it and the environment in which it operates. An illustration would be a computerized railway system where there will be specialized sensors able to detect the presence of a train and others to check that a set of points have moved as well as the electronically controlled equipment which moves the points. The computer which controls such a system will be dedicated to the task of controlling the

movement of trains and will not be used for any other task. These specialized sensors and the equipment which causes mechanical operations to take place are called transducers.

1.2 Basic components of a computing machine

Computing machines, whether dedicated or general_purpose can be represented by the following block diagram (Fig. 1.1). Obviously in



rticular instances some of the components may assume greater impor-
ce

1.2.1 *Central processing unit*

The central processing unit (CPU) pr vides central control of the functions of the whole computing machine. In order to function correctly a computing machine requires all necessary data to be made available to the component parts of the machine at exactly the right time If this is not done the user of the computing machine would find that the results produced by the computing machine would be wrong. As well as providing control

functions the CPU also provides the arithmetic and logic capability of the computing machine.

1.2.2 *Memory*

The memory of a computing machine is used to hold two distinct types of information: first the data which is to be processed by the computing machine and secondly, the finite sequence of operations which will process that data to produce desired results. The sequence of operations is called an algorithm. The algorithm is said to operate upon a data structure, the data structure being a description of the form of the data rather than the particular data values. The combination of an algorithm and its associated data structure is called a program.

The memory of a computing machine is divided into many similar cells or locations each of which is individually addressable To say that a cell is addressable means that a CPU can uniquely identify a location within a program so that it can hold a specified data item value. Similarly the address can be used by the algorithmic part of a program to alter the sequence of execution of operations. Instructions are stored in the memory of the computer in sequence, one location after another. Computing machines are extremely good at carrying out repetitive sequences of operations because they are fast and accurate.

This situation is very similar to the way in which knitting patterns are described. The knitter is told the sequence of stitches required to produce a desired effect and then told to repeat that sequence until the desired length has been achieved.

In order that repetitions can be generated in algorithms it is necessary for the sequence to be able to go back over itself, thus generating the necessary repetitions. The addresses of the memory locations are utilized by some of the operations within the computing machine to achieve this effect.

Most memories are unable to distinguish from the contents of a location whether it contains data, algorithm or irrelevant information. It is up to the user of a computer to ensure that appropriate locations of the memory are filled with sensible data and an algorithm before commencing any processing.

1.2.3 *File store*

The file store holds bulk data and its storage capacity is very large. A single unit of file store may contain ten or a hundred times more locations than the memory itself. The file store can then be constructed from many such units. In order that this large amount of storage space can be more easily managed it is logically split into smaller sections called files. It is then usual

to associate some of the files with a particular program. However, in most computer systems there is nothing to stop any file being processed by any program provided the data is stored in the correct format according to the data structure in the program.

Memory is one of the more expensive components of a computer and it is, therefore, not an infinite resource. Hence, within one computing machine it would not take long before the memory was filled by programs which although required were not currently being executed. The file store provides a means of keeping copies of these programs available to the computing machine in a state of readiness. When such a stored program is required it can be brought from the file store and placed in memory so that it can be executed. On completion, the program will be returned to the file store until required again.

Probably a more important use of file store is for the recording of bulk data. The program describes the data structure upon which it will operate but not the actual data values to be manipulated. These data values have to be stored somewhere and presented to the algorithm as required. In the majority of data processing applications there is far too much data to be processed for it all to be in the memory at the same time. This bulk data is stored in the file store and then transferred to the program as necessary.

It is usual to have a means of identifying users of the computer system so that only those with appropriate authorization can process specific data files with particular programs.

1.2.4 *Peripherals*

The majority of data processed by a computer together with the algorithms has to be placed in the computer by some method of communication. A peripheral provides this communication with the outside world.

Peripherals can be divided into three classes; input devices, output devices and i/o devices which are capable of communicating in both modes. In a general purpose computer system there will be many different peripherals, so that the user has a degree of flexibility and the use of peripherals is thus tailored to the needs of the application.

In a dedicated computer system it is often the peripherals which make the system dedicated. Up to this stage the computer could have been used for many different applications. The connection of certain peripherals in the form of specific transducers dedicates the computer system. Such dedication is found in process control.

In this case the computer system has to be able to respond rapidly to the requirements of the process being controlled. It would not be sufficient for this data to be transferred by a human operator. Thus it is necessary for the information collected from the parameter being sensed to be entered directly into the computer.

1.3 Putting it all together

Up to this stage it may have appeared that the only place for variation in computing machines and computer systems is in the nature of peripherals and in the amount of memory and file store available. This is not the case and it is the role of the computer architect to design a computing machine so that it will best respond to the environment in which it will be used. To this end, the computer architect has to make a great many decisions about the way in which the basic building blocks will be constructed and then how they will be connected together to form a unified solution for the problem environment. There is no one correct solution for a single environment because each architect has his own ideas and there is usually a financial compromise to be taken into account. Hence, given the same initial building blocks, it is possible to construct many different computing machines, some of which will be more appropriate to particular areas.

We will discuss briefly some of these design criteria at this stage but they will be expanded upon in later chapters.

1.3.1 *Interconnection*

The architect or designer has initially many ways in which he can connect the component parts of the computer together. By incorporating a large number of specific connections he can make the computing machine process extremely quickly. This approach is, however, costly because it requires a lot of expensive electronic components

Conversely the components of the computer can be connected together in a far more generalized way which saves on component costs, as not so many are used. In that event the computer system will process information more slowly because it is typical of such systems that the same components are used for many different data transfers. Hence some transfers may have to wait until a previous transfer has been completed. It is thus the job of the architect to choose the right compromise between speed and cost so that the computer system satisfies its design criteria

1.3.2 *Operating systems*

Besides producing the computing machine a computer manufacturer usually provides some means of controlling the complete computer system. In small systems this may be left to the purchaser The computing system control mechanism is provided by means of a program called the operating system The operating system is a program always available in the computer which allows the computing machine to be operated at its maximum efficiency. The system related to a particular environment provides the user with three basic resources name-

processing time, memory to store algorithms and data, and finally use of peripherals. One measure of operating system efficiency is to see how effectively it manages these resources so as to ensure that each resource is used to the maximum. One way is to keep several programs simultaneously in a partial state of execution within the computer. Each is given a share of the machine in turn. Obviously this requires a very complex operating system. Such an operating system will maintain a list of programs waiting to be executed for the first time. From this list the operating system will select that program which allows it to make most effective use of the resources.

An alternative approach is to initiate a program and then let it execute until processing is completed. This will obviously be wasteful of resources, but the operating system will be much simpler and quicker in operation.

The computer architect has to be aware of the sort of operating system to be used with the computing machine. He has, therefore, to design the computing machine and take account of the operating system. It is possibly true to say that there is more cost involved in the analysis, design and implementation of the operating system than there is of the underlying computing machine. Hence the computing machine and the operating system are normally designed in parallel.

1.3.3 *Distribution*

Increasingly the computer architect has had to become more aware of the problems caused by distributed computer systems.

As explained earlier, a distributed computer system is one which is constructed from several computing machines. Each of the computing machines needs to communicate with its immediate neighbours in the system. So that this may be done in an orderly fashion facilities have to be provided either by programs within the operating system or by the CPUs of the machines. Such facilities are called protocols.

If copies of the same data are stored on several different computing machines then there has to be a mechanism to ensure that all copies have been updated before any user has access to any copy of the data.

Another major problem arises when deciding what facilities of the computer system are to be distributed. The facilities to be considered are processing capability and data storage. The spectrum ranges from a distributed system which has a central file store with many computing machines connected, to a system which has processing and data storage capabilities uniformly distributed.

The computer architect has to ensure that either he constructs computing machines that can operate in all these modes or a range of computing machines each of which is suited to a particular role within the distributed system.