Microcomputer design

1—Introduction to digital hardware based on a microprocessor

by Phil Pittman, B.Sc. *in association with NASCO Ltd*

The low cost computing power of the microprocessor is now being used to replace not only other forms of digital electronics but also analogue electronics and electromechanical and pure mechanical control systems. It is not unreasonable to assume that within the next five years or so there will be hardly any companies engaged in electronics which are not using microprocessors in one area or another. One implication of this technology is that engineers skilled in the design of more conventional electronic circuits and systems now have to acquire new disciplines — those of digital computer system design and programming. This series of articles will present the theory and application of microcomputers by reference to a particular commercially available microprocessor, and to its use in a particular microcomputer system available to amateur experimenters as a kit (see panel). This low-cost kit includes memory, input/output circuits and a keyboard, and can be used in the home with a domestic television set as a display unit and an audio cassette recorder for permanent storage of programmes. The first article examines the hardware components and principles of operation of such a general purpose computer system. Future articles will explore programming languages, the organization of the central processing unit, and practical design techniques for both the hardware and software of microprocessor-based systems.

In its most general form a digital computer system has the structure shown in Fig. 1. The central processing unit (c.p.u.), memory and input and output units are the essential hardware blocks which any computer must have. The c.p.u. does the work, manipulating data as directed by a programme stored in the memory. The memory may also be used for storing data. Information is transferred to and from the outside world by the c.p.u. via the input and output units.

The c.p.u. being the most complex part and the heart of all operations in the system, will be examined first. It may be viewed as two parts. One part, called the arithmetic and logic unit, actually does the work, while another part controls the sequence in which the various functions are performed. For the moment our main attention will be given to the arithmetic and logic unit (a.l.u.).

Any digital computer, including a microprocessor-based system, performs its data manipulation operations by utilising various combinations of the basic Boolean logic functions AND, OR, NAND, NOR etc. Of course, in a processor system many of the operations are often compounded from these basic functions to provide more complex operations. Programme instructions are used to selectively activate the various logic and arithmetic functions of the processing unit in order to achieve the required result. Consequently, a processor may be viewed as a programmable, general purpose logic block.

In this concept lies one of the reasons

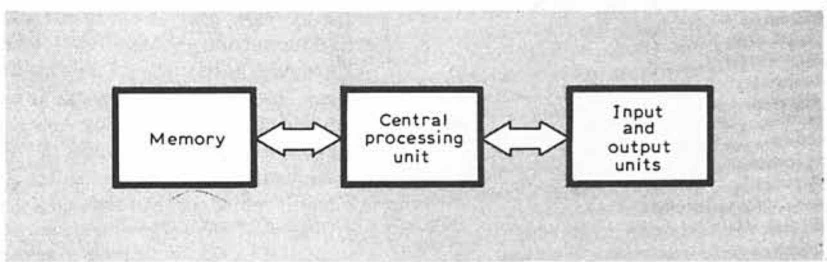

Fig 1

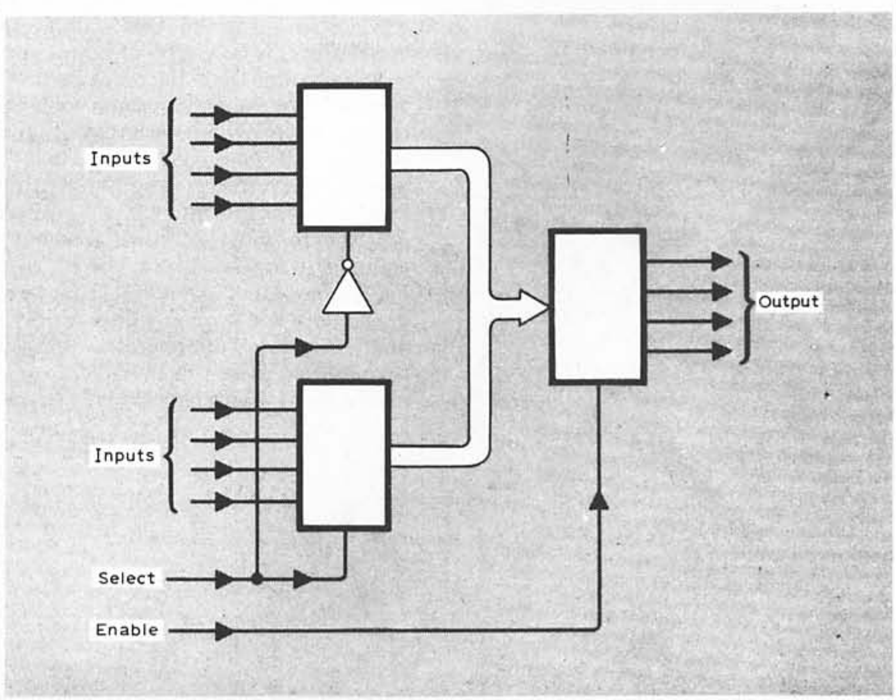

Fig 2

Fig. 1. Basic structure of a digital computer.
Fig. 2. A typical standard logic block, considered in the article as a step on the way to programmable logic.

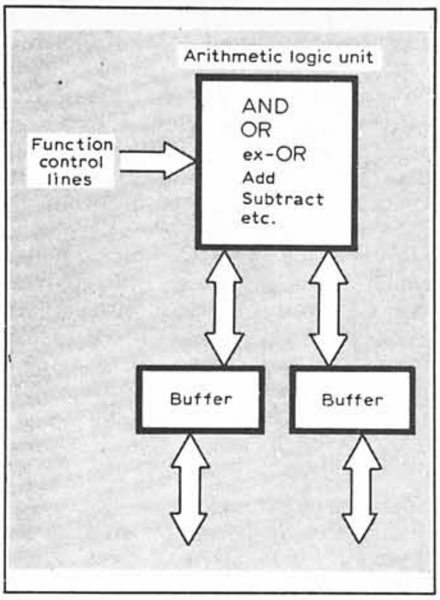

Fig. 3. A general purpose arithmetic/logic block.

for the use of the microprocessor, not only for the more usual computer type applications of data processing, but also for dedicated controllers and logic replacement devices. The low cost of microprocessors has now made them an economic solution for countless applications.

It is worth exploring this programmable logic concept further in order to understand more fully the operation and application of the central processing unit. Consider a standard quad 2-input multiplexer logic circuit as shown in Fig. 2. Here we see a standard logic block having a number of data input and output lines. The functions performed within the block are implemented with conventional logic gates. The exact function performed is dependent on the state of other inputs to the "system". The source of input data is determined by the state of the "select" line and the data is transmitted to the output under control of the "enable" line. Extending this concept further, we can arrive at the example of Fig. 3. The logic block has now been enhanced to include binary arithmetic functions. Also, there must be several more control lines available to select both the a.l.u. function and the data source and destination.

This programmable logic system now looks remarkably similar to the a.l.u. of a computer or microprocessor. In the processor the function control lines are derived, usually via decoding logic within the processor, from the binary instruction words which form the computer's programme. In order to control the sequential fetching and execution of the control codes, or instructions, from the programme memory the processor has a counter called a programme counter. If we now add a memory unit to the system of Fig. 3, together with additional a.l.u. operations to transfer data to and from the memory, the resulting structure is virtually identical to the general purpose computer of Fig.1.

Internally the c.p.u. processes information as parallel rows of bits (binary digits) and so information usually flows in and out in the same format. The multiplexer circuit example given in Fig. 2 receives and issues data in 4-bit parallel form. Common microprocessor organisations are based on 4-, 8-, 12- and 16-bit word lengths. Of these, the 4-bit devices were the earliest types to appear commercially, partly because they were useful in calculators operating with binary coded decimal data, but also because the a.l.u., being only 4 bits "wide," was less complex, this allowing more circuit functions for a given cost on a semiconductor chip of given area.

Nowadays the technology has developed such that a complex 8-bit processor or even a complete microcomputer can be built on a single chip, resulting in the fact that hardly any manufacturers are introducing new 4-bit designs now. Eight bits has proved to be the most popular word length for microprocessing since the majority of applications can conveniently be dealt with by 8-bit quantities. Also, 8 bits represents the best cost/performance trade-off compared with other word lengths. Current 16-bit microprocessors offer surprisingly little increase in performance over 8-bit machines, even for applications requiring the manipulation of 16-bit quantities. Using a 16-bit processor where an 8-bit one will suffice will also inevitably incur higher system hardware costs.

Microprocessor systems

By adding more detail to the diagram of Fig. 1 we can evolve a block diagram of a practical microprocessor system. In this case it represents part of the commercial microcomputer kit referred to above, which uses a Mostek Z80 microprocessor, and is shown in Fig. 4.

A microcomputer system is merely an l.s.i. (large scale integration) implementation of the basic computer structure. The c.p.u., or microprocessor, is usually a single integrated circuit containing the a.l.u. plus programme sequence control and instruction decoding logic. The internal structure of the Mostek Z80 c.p.u. is shown in Fig. 5. The memory may consist of anything from one to a great many components of various types of memory. The input and output circuits may transfer data in serial or parallel fashion, the number of bits in a transfer being determined by the design, or more specifically the word length, of the microprocessor. Broadly speaking, the majority of input/output circuits (commonly abbreviated to i/o) in a microprocessor system will be of a parallel nature. Each i/o block is commonly called a port, where again the number of bits constituting a port is given by the microprocessor's word length. Each i/o port may be a complete integrated circuit, although many microprocessor families, including the Z80, have circuits containing a number of i/o functions.

All microprocessors use some form of clock circuit as a basic timing reference for instruction executions, memory and i/o operations. In the case of the Z80 a single-phase square wave has to be supplied to the c.p.u. component. This is provided by a simple t.t.l. circuit in the kit.

In order to form a working system these components must, of course, be suitably interconnected. Herein lies the elegance of microcomputer hardware. The microprocessor has a number of external connexions which may be used directly or indirectly to provide three categories of information for the remainder of the system. These information "buses", as they are called (shown by broad arrows in the diagrams), are connected in a standard manner to the memory and i/o devices regardless of the end application and regardless of the number of memory and i/o components to be used. Consequently a microprocessor system

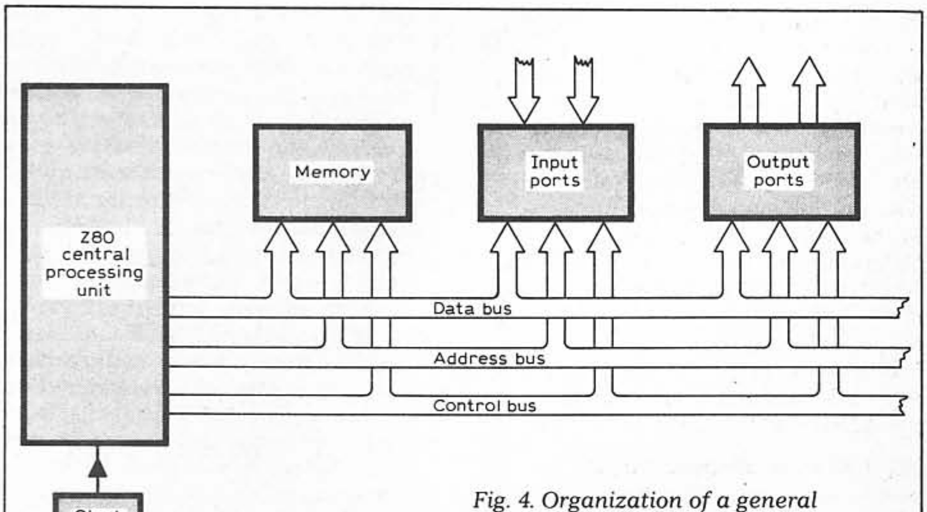

Fig. 4. Organization of a general three-bus microcomputer system.

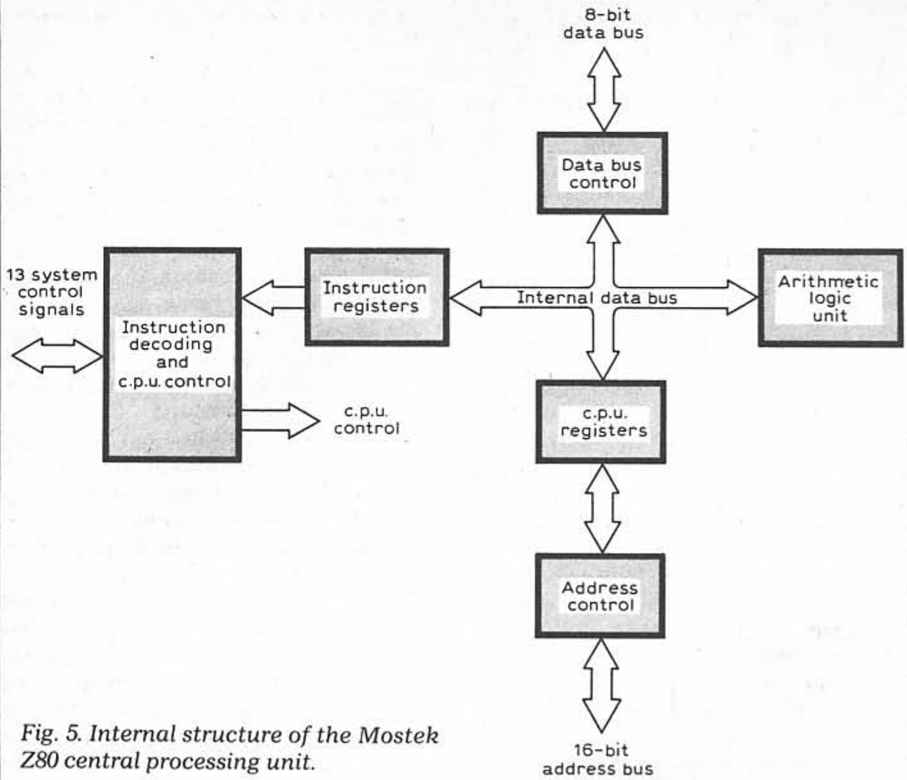

Fig. 5. Internal structure of the Mostek Z80 central processing unit.

may be usefully programmed and reprogrammed for a variety of applications. Similarly, a system with minimal hardware, if well designed, may be expanded at will to a more powerful configuration. This was the philosophy of the kit design.

Consider these three information buses which are shown in Fig. 4. They are the data bus, the address bus and the control bus. The data bus is a bidirectional one used for transfer of both data and instructions into and out of the c.p.u. The number of lines constituting the data bus is the same as the number of bits in the machine's word length. The Z80, being an 8-bit c.p.u., has a 8-bit data bus. This means that the system memory must also be organised as locations of 8 parallel bits. Similarly, i/o transfers will be to and from 8-bit ports. The data bus is connected around the components of the system such that all devices are placed on the common bus. All information transferred under programme control travels on this bus via the c.p.u.; for example, to transfer an item of data stored in the memory to a port, the data must first be fetched from the memory and passed into the c.p.u. and then sent from the c.p.u. to the output port.

Now let us consider the address bus. The number of bits constituting this bus has no direct relationship to the word length of the microprocessor. The address bus is used to select, or address, the location in the memory or the particular i/o port required for the current operation. The Z80 processor has an address bus of 16 bits, allowing the kit to be expanded to a maximum of $2^{16} = 65536$ memory locations (or 64K bytes, where 1K = 1024). The value placed on the bus by the processor depends on the operation being performed, e.g. at the beginning of the instruction cycle the processor must supply the address of the next instruction in sequence to be fetched from programme memory. Then, during the execution of the instruction, data may be required to be moved between the c.p.u. and either the memory or an i/o port. If this is the case then the data memory address or i/o port address must be placed on the address bus by the c.p.u.

The third bus is the control bus. This is slightly different from the other two buses in that it is really a collection of individual control lines for memory, i/o and c.p.u. control. For example, in the case of the Z80 the main control signals are a "read" strobe pulse used to strobe data on the data bus into the c.p.u. from memory or i/o, and a "write" strobe to indicate that valid data is on the data bus from the c.p.u. to memory or i/o. This may be used to strobe data into a port or memory. Also there is an "input/output request" signal to indicate that the address bus contains a valid i/o port address, rather than a memory address. Similarly, there is a "memory request" signal indicating a valid memory address on the address bus, rather than an i/o address. Other control signals include a "reset" to the c.p.u., interrupt control (a concept explained later) and signals for suspending c.p.u. operation and de-activating the buses (useful in more complex, e.g. multiprocessor, systems).

A complete microcomputer system has now been evolved which contains all the necessary functional blocks. In a future article the Z80 c.p.u. (Fig. 5) will be explained in more detail, along with the concepts which influence its design and use. For now it is sufficient to say that in addition to the a.l.u., the c.p.u. contains various register stores. In the case of the Z80 there are 18 eight-bit registers and 4 sixteen-bit registers which are accessible to the programmer via the various c.p.u. instructions. Some of these registers serve special functions and others are general purpose stores similar to the main memory locations.

Memory organisation

It has already been implied that the memory of a computer system is used for two things – remembering instruction sequences (the programme) and remembering data. Semiconductor memory components may be one of two basic types, i.e. fixed, non-alterable memory and alterable, read/write memory. In dedicated microprocessor applications it is desirable to have the applications programmes fixed in permanent memory so that they are not lost when electrical power is removed. When power is applied, automatic operation of the system is then to be guaranteed. Such memory is called "read only memory" (r.o.m.). A true r.o.m. as such generally has its information fixed in it during the manufacturing process according to the particular customer's requirements. Consequently there is a minimum manufacturing quantity for this "customising" which is typically in the region of 100 to 1000 units. A popular alternative for lower volume and prototyping applications is the p.r.o.m. This is a programmable r.o.m. where the information may be fixed by the user by an electrical process which still results in permanent storage. A further development of this is the eraseable p.r.o.m. or e.p.r.o.m.. These devices may have their data erased by exposure to short wavelength ultra-violet light, thereby enabling them to be reprogrammed.

Strictly speaking all these memories have the feature of being "random access". This means that any memory location may be reached, or "accessed", with equal ease, at random, by applying the appropriate address. However, the term "random access memory", or r.a.m., has commonly come to mean a read/write, or alterable, memory, e.g. the type used as a data storage element in a microprocessor system. In a general purpose computer system this r.a.m. may also be used as a programme store, thereby enabling different programmes to be loaded and executed at will.

The microcomputer kit referred to above employs a combination of the memories just described. In order to allow meaningful user communications with this system there is a 1024-location

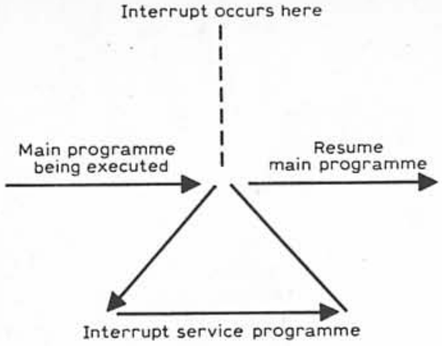

Fig. 6. Representation of the programme interrupt sequence.

(1K × 8 or 1K byte) e.p.r.o.m. containing a fixed programme. There is also a 2048-location (2K × 8 or 2K byte) r.a.m., a small amount of which is used for variable data required by this programme. However, the main purpose of this r.a.m. is to allow the system to function as a general purpose computer, i.e. the user may enter his own programmes to the r.a.m., via the i/o peripherals and under control of the e.p.r.o.m. programme, for subsequent execution.

Input/output organisation

Data may be transferred to and from a microprocessor system in several ways, some under programme control and some initiated by external events not under control of the processor. The most commonly used type of input/output operation is generally that which is initiated by programmed instructions. The Z80 can transfer an 8-bit value to or from an i/o port with an instruction taking 4 microseconds for its execution. Alternatively, a single instruction may be used to transfer a complete block of data at a rate of 8 microseconds per byte. The i/o instructions are used to transfer the state of data existing on the lines of an input port into the c.p.u. or, conversely, data from the c.p.u. to the lines of an output port. The circuits of the i/o ports frequently have the capability of temporarily storing the i/o data. That is, incoming data may be latched on the port by the peripheral device until it is accepted by the processor executing an input instruction. Similarly, data output from the processor is often latched on the port until required by the peripheral. A subsequent output to the same port could then change the state of the output lines.

Other ways of implementing input/output in a system are by direct memory access (d.m.a.) or by programme interrupts.

Programme interrupts are a method of initiating a data transfer independent of the normal programme flow. For example, suppose the processor has to interrogate several peripheral i/o ports to see if they have any data to be collected for processing. One way for the c.p.u. to handle such a situation is for the system programme to control a "polling" routine whereby each device is periodically examined in turn for data. The disadvantage of this is that a great deal of valuable processing time may be consumed by checking for valid data at a port when frequently there may be none ready. Interrupt operation overcomes this limitation. Now the c.p.u. does not have to periodically check for valid data: it is told, or interrupted, by the peripheral when this data is available. This interrupt, usually sent as a signal from a interrupt control circuit, has the effect of suspending the execution of the programme and then forcing the c.p.u. to a new programme which services the interrupting device. Upon completion of the service programme the c.p.u. is allowed to resume the previous execution of the programme. Fig. 6 shows the interrupt sequence diagrammatically. In a system with a number of separate interrupt sources it is usual to assign a priority to each one to ensure a sequence of servicing if multiple interrupts occur.

Direct memory access is generally a faster method of transferring data than may be achieved under programme control. D.m.a. transfers occur directly between the system's memory and the i/o device without involving the c.p.u. Consequently the speed of data transfer is limited essentially only by the speed of main memory. A typical d.m.a. transfer rate would be in excess of 1 megabyte/s. A special d.m.a. controller circuit initiates and controls the transfer in response to an external request. The c.p.u. operation must be suspended during the transfer and is allowed to resume operation when the transfer is complete. This is so that both c.p.u. and d.m.a. controller will not try to use the system buses simultaneously. D.m.a. is generally used in more complex systems when large blocks of data have to be transferred to or from peripherals at a speed greater than can be achieved by programme instructions. It is also possible to do single byte transfers under d.m.a., often without stopping the processor if the transfer can take place while the c.p.u. is not using the buses. This is often called "cycle stealing".

The actual i/o parts of a microprocessor may be implemented with standard t.t.l. logic circuits, e.g. 8-bit latches or buffers, or with the more integrated members of a manufacturer's l.s.i. microcomputer family.

Part 2 of this series will give a practical example of hardware and will also deal with software.

SPECIAL TERMINOLOGY

Microcomputer. A digital computer which uses a microprocessor as its central processing unit. The prefix "micro" does not mean literally "a millionth part of" but is derived from the word "microcircuit," an early name for the integrated circuit.

Microprocessor. A digital processing unit constructed as one or more integrated circuits, using l.s.i. manufacturing technology. Can be used as part of a microcomputer.

Instruction. An expression that defines a computer operation and identifies its operands.

Programme. A prepared list of instructions, written in a special "language" or code, to be carried out in sequence by a computer or other programmable device.

Bit. Abbreviated form of **binary digit.** The basic unit of binary coding (1 or 0) used to represent numbers, instructions or addresses.

Byte. Unit of binary information, normally consisting of eight bits.

Word. A group of binary digits representing a number, instruction or any other item of information. Often specified by its length, e.g. 16-bit word.

K. Abbreviation for 1024 (to be distinguished from the common lower-case prefix "k" that represents 1000).

Serial. Representation of binary information in which the binary digits occur in time sequence (e.g. on a single wire).

Parallel. Representation of binary information in which the binary digits occur simultaneously (e.g. 4 bits on 4 wires).

Bus. Abbreviated form of "bus-bar" derived from "omnibus". A group of conductors carrying words in parallel (one bit per conductor) in either direction; usually common to several devices and identified by function, e.g. address bus.