Bus Structures and Hardware Interconnection with the Microprocessor

Chapter 4

INTRODUCTION

The microprocessor communicates with other parts of the microcomputer by sending information along groups of signal lines called buses. There are several different types of information to be transmitted between the various elements of the system:

(i) Program memory addresses
(ii) Instruction codes
(iii) Data memory addresses
(iv) Data from memory
(v) Data to memory
(vi) Input-output device addresses
(vii) Data to output devices
(viii) Data from input devices.

In a large computer system each type of information might be transmitted along its own bus. In a microcomputer, it is common for more than one type of information to share the same bus. Figure 4-1 shows the bus structure of the simple microcomputer used as an example in Chapter 1. It has two *shared buses* along which different types of information are transmitted at different times. The instruction codes from program memory and the data from data memory share the same input bus; the program memory and data memory addresses share the same address bus.

The need for introducing a shared bus system is a direct consequence of the limited number of external pin connections that are feasible on a standard integrated circuit package. Although the maximum pin count has increased from 16 in the early seventies

to 64 in the most recent microprocessor chips, some data multiplexing is always necessary to reduce the number of interconnections that are required. There are many possible ways of multiplexing the different types of information and a wide variety of bus structures are commonly used in microcomputers. In this chapter, the basic types of bus structures are explained, examples of the common bus systems are described and the methods used to interconnect memory and I/O devices to the microprocessor via the buses are discussed.

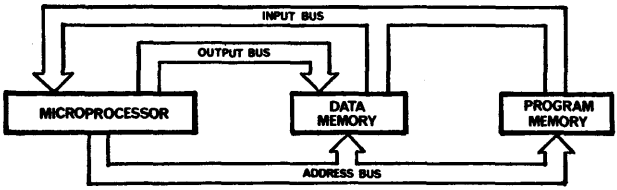

Figure 4-1. A simple shared bus microcomputer system

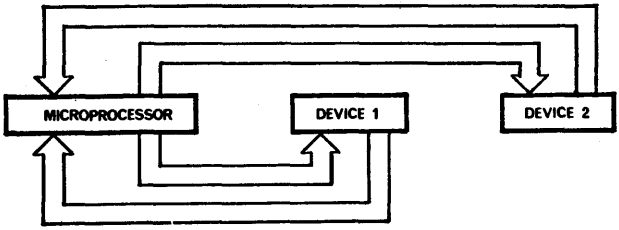

Figure 4-2a. Unique line bus structure

BASIC TYPES OF BUS STRUCTURES

The most fundamental bus structure is shown in Figure 4-2a. Each device in the system is linked to the microprocessor by unique signal lines. The structure is functionally simple, but expensive in external interconnections particularly on the microprocessor chip. There are two types of multiplexed bus structures based on the shared bus principle:

(i) *The daisy-chain structure.* As shown in Figure 4-2b, the information is transmitted through each element of the system in turn along a loop of *unidirectional buses* until it arrives at the correct device. Each device must act as a source and an acceptor of information on the bus.

(ii) *The party-line structure.* Each device of the system is linked directly to a single bus as shown in Figure 4-2c. When only one device acts as a source of information, the bus is unidirectional. More frequently the party-line structure is based on a *bidirectional bus* with information passing in either direction along the bus between a number of sources and acceptors of information.

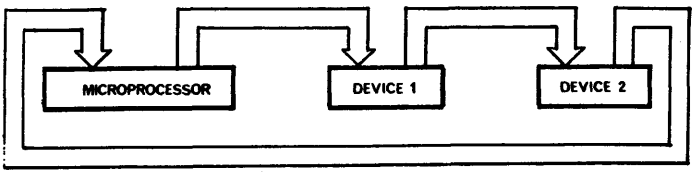

Figure 4-2b. Daisy-chain bus structure

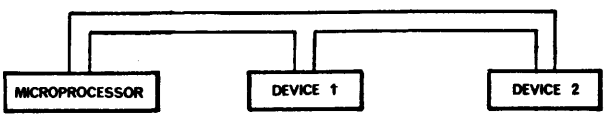

Figure 4-2c. Party-line bus structure

BUS CONTROL SIGNALS

In addition to controlling the rate of information flow on the bus, bus control signals are needed to indicate the direction of information flow on a bidirectional bus, to specify the type of information present on a shared bus and to define which element or elements of a microcomputer system are in control of the information flow on a party-line bus with more than one information source. There are two methods of controlling the rate of information flow on the bus:

(i) *Synchronous control* (Figure 4-3a). One element of the system, usually the microprocessor, generates all the bus control

and timing signals. The other elements in the system are required to synchronise to these signals when transmitting or receiving information on the bus.

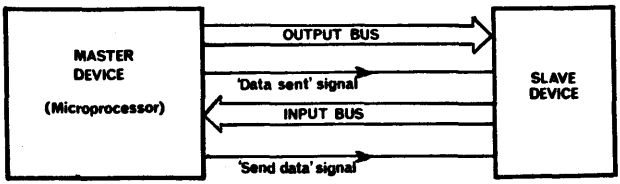

Figure 4-3a. Synchronous bus control

(ii) *Asynchronous control.* Bus control and timing signals are generated jointly by the source and the acceptor of the information transmitted on the bus. The *hand-shake* procedure shown in Figure 4-3b is often used. For example, the microprocessor might initiate the transmission of data from memory with a "send data" control signal, and then wait in a processor halt condition until it receives a response from memory. The response, a "data sent" control signal, is sent from the memory after it has placed its data on the bus. On receipt of the response, the microprocessor accepts the data from the bus and issues a "data accepted" control signal. The operation is finally completed by the memory which, after receiving this signal, removes its data from the bus and sends a "bus cleared" control signal.

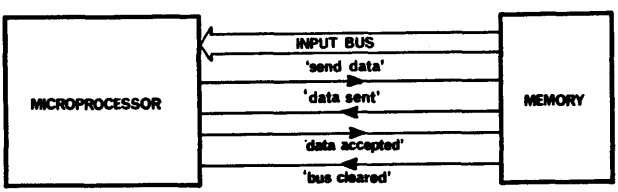

Figure 4-3b. Asynchronous bus control using a double hand-shake

The lines over which the various bus control signals are transmitted are frequently grouped together under one name – the *control bus.*

TYPICAL BUS SYSTEMS

The three basic bus structures described earlier form the basis of all microcomputer bus systems. All use the bidirectional party-line structure for the main data bus but the number and structure of the other buses differs from system to system.

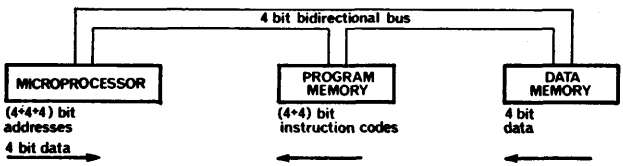

Figure 4-4a. Single 4-bit bus microcomputer

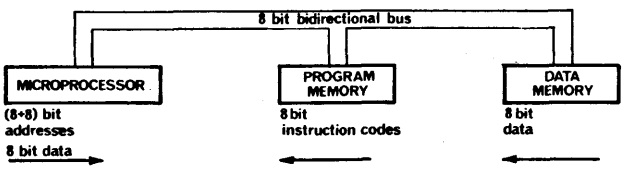

Figure 4-4b. Single 8-bit bus microcomputer

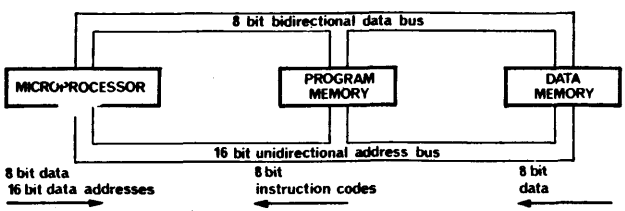

Figure 4-4c. Dual bus microcomputer

The single all-purpose bidirectional bus structure shown in Figure 4-4a was common in the first microprocessors where the number of external pin connections was severely limited (16 or 24 pins/chip). Frequently the bus was only 4 bits wide and instruction codes and memory addresses were split into 4-bit slices prior to transmission on the bus. The difficulties of demultiplexing the information on this type of bus are considerable. As larger numbers of pin connections have become feasible (40 or more pins/chip),

microprocessors have been introduced first using a single 8-bit wide bus (Figure 4-4b), and then using two separate buses (Figure 4-4c). In the latter case one is an unidirectional bus for addresses and the other is a bidirectional bus for data and instruction codes. The dual bus structure with a 12- or 16-bit address bus and an 8-bit data bus is now an accepted standard for 8-bit microprocessors. However, a variety of bus structures is used in 16-bit microprocessors. Some have a single 16-bit wide shared data/address bus (Figure 4-4d), whilst others have separate 16-bit buses – one for data and one for addresses.

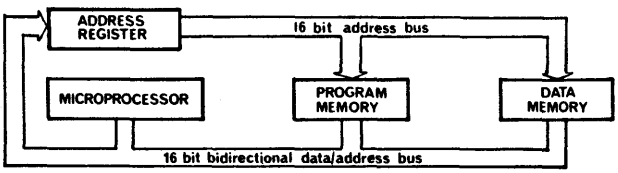

Figure 4-4d. Single 16-bit bus microcomputer with external address register

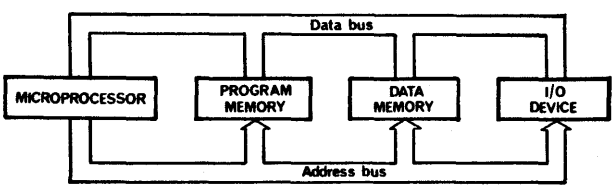

Figure 4-4e. Unified bus structure

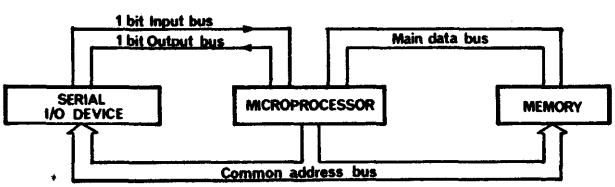

Figure 4-4f. Separate serial I/O bus structure

In most systems, communication with input-output devices is also made along the main bidirectional data bus as shown in Figure 4-4e (the unified bus approach), though several microprocessors have one or more separate I/O buses. Both serial and parallel I/O

buses are now common as shown in Figure 4-4f and Figure 4-4g. Almost all microprocessor systems, whether they have separate I/O instructions or use memory-mapped I/O, use the same address bus for both memory and I/O device addresses.

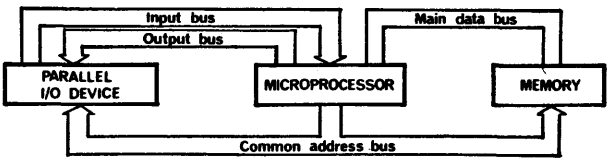

Figure 4-4g. Separate parallel I/O bus structure

INTERCONNECTING SEVERAL SOURCES OF INFORMATION TO THE SAME BUS

When information is supplied to a bus from more than one source, special bus driving circuits must be used to select and control the outputs from the sources so as to prevent interference between them.

The *bus drivers* perform three basic functions:

(i) They ensure electrical compatibility between each of the sources connected to the bus.

(ii) They control the connection and disconnection of sources from the bus.

(iii) They multiplex the information onto the bus.

Three different types of bus drivers are commonly used in microprocessor systems:

(i) *The logical-OR driver.* The driver includes a digital multiplexer which, as shown in Figure 4-5, is realised using conventional logic gates. Since one or another of the sources is always connected to the bus, this type of driver cannot be used with a bidirectional bus. The logical-OR driver is inconvenient in systems which have geographically distant sources and is difficult to modify if additional sources are to be added to the bus.

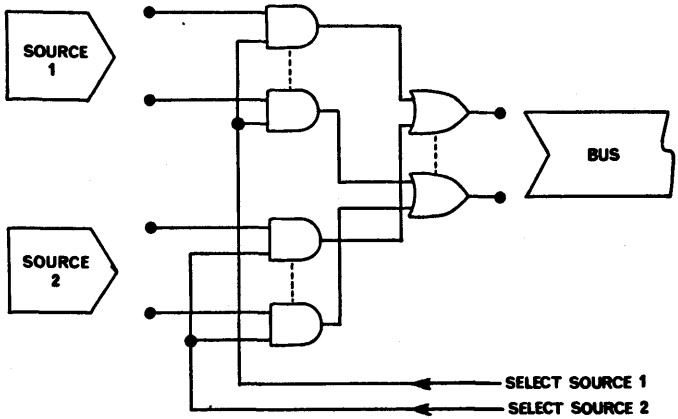

Figure 4-5. The logical-OR bus driver

(ii) *The wired-OR driver.* In this driver the multiplexing is achieved by using open-collector logic gates at the output of each source. The gate outputs are wired together directly onto the bus as shown in Figure 4-6. The open-collector gates have the property that any one output transistor, if switched on, will override all other transistors which are switched off and so determine the condition of the common output line. Since the output transistors of sources not selected are turned off, the condition of the output transistors of the selected source will determine the data on the bus. The wired-OR driver can be used with a bidirectional bus since it is possible to select none of the sources.

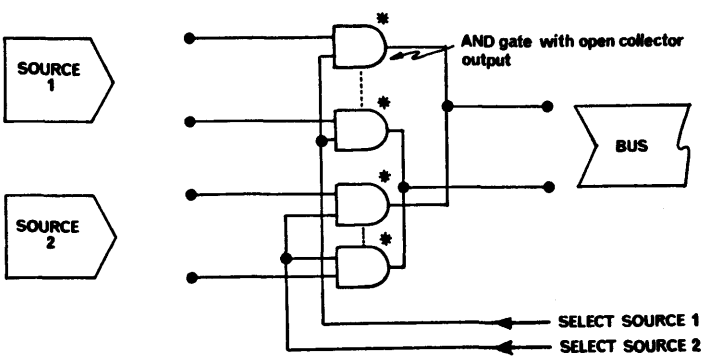

Figure 4-6. Wired-OR bus driver

(iii) *The three-state driver.* Special *three-state logic gates* which have active pull-up and pull-down output stages are used in the digital multiplexer. The output of the gates can be set in one of three states:

(a) Logical 1
(b) Logical 0
(c) High-impedance.

When the gate is enabled, the output is determined by the input to the gate. When the gate is disabled, the output is set in the high-impedance state in which the output line floats, acting as neither a source nor a sink of current. As shown in Figure 4-7, sources not selected are disabled and thus disconnected from the bus. The three-state driver can also be used on a bidirectional bus.

Since all sources of information must be connected to the bus via the same type of driver, the type of driver to be used on a particular bus will be determined by the microprocessor. The bus is often named according to the type of driver (e.g., open-collector bus, three-state bus).

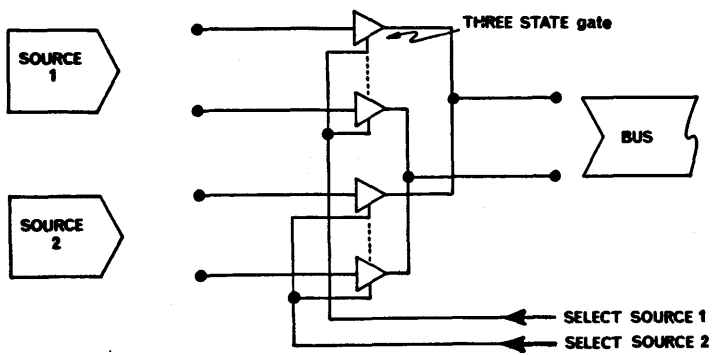

Figure 4-7. Three-state bus driver

INTERCONNECTING SEVERAL ACCEPTORS OF INFORMATION TO THE SAME BUS

When the information on the bus can be sent to either one of several acceptors, bus receiving circuits are needed to connect each acceptor to the bus. The *bus receivers* perform two basic functions:

(i) They ensure electrical compatibility between the acceptor and the sources on the bus.

(ii) They selectively gate the flow of information from the bus to the acceptor.

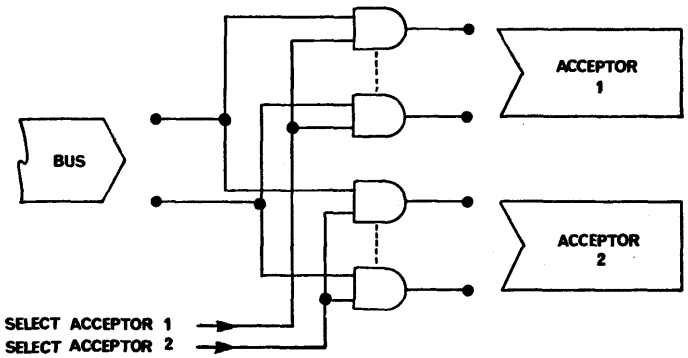

Figure 4-8. A simple bus receiver

Figure 4-8 illustrates the use of a simple receiver. The input gates, which are connected directly to the bus, are conventional logic gates of any logical family compatible with the bus. The maximum current loading on the bus is determined by the bus driver circuits. Most microprocessors have drivers capable of driving the equivalent of one standard TTL load on data or address buses. The total capacitive loading on the bus is also important since for a given source resistance, it will determine the bandwidth of the bus and limit the maximum speed of operation of the microcomputer. A maximum capacitive bus loading of approximately 100pF is typical for a microprocessor with a 2 μs cycle time. Bus receivers using special input gates can reduce the maximum bus loading, since the input lines to these circuits can be set in the high impedance state when the gate is disabled and the receiver is then electrically disconnected from the bus. The total bus loading is dependent on the maximum number of receivers which are enabled simultaneously. Some bus receivers use clocked latches in place of the input gates to incorporate a temporary information store (a buffer register) in the receiver. Figure 4-9 shows a 4-bit bus system with two sources and two acceptors using three-state drivers and latched receivers.

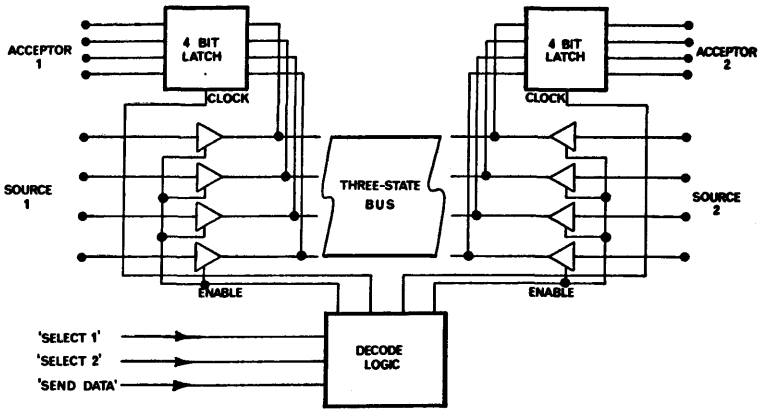

Figure 4-9. Bidirectional bus system

THE THREE-STATE BUS

Almost all microprocessor systems have one or more three-state buses. A number of special-purpose three-state devices are available to simplify the problem of interconnection with this type of bus. The *three-state bus transceiver* (a pair of interconnected bus receivers and drivers – see Figure 4-10) is often used as the basis of an interface circuit to link an I/O device to the bus. The *three-state latch* (a 4- or 8-bit data register with three-state output drivers – see Figure 4-11) is of general use in the design of bus interfaces. The *three-state bidirectional driver* (a quad of back-to-back bus drivers connected in parallel with complimentary driving directions – see Figure 4-12) is used as a bus buffer to boost the driving capability of a microprocessor which is connected to a heavily loaded bus system.

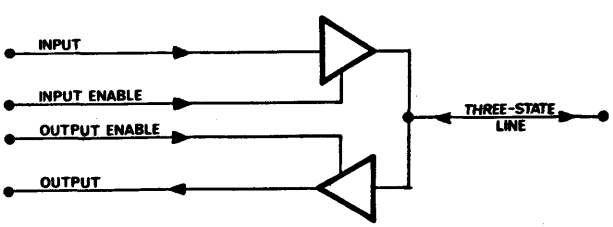

Figure 4-10. Three-state transceiver

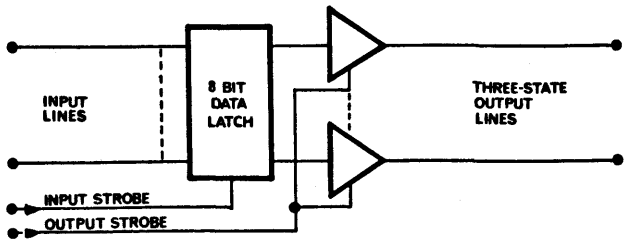

Figure 4-11. Three-state latch

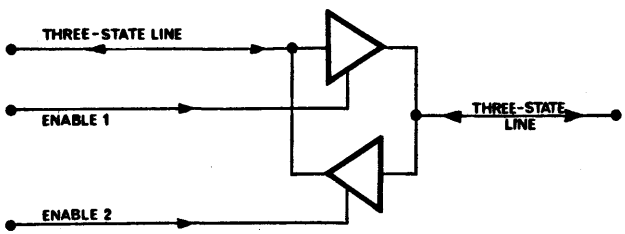

Figure 4-12. Three-state bidirectional driver

Specific examples of the use of these circuits are given later in Chapter 9.

THE MEMORY/BUS INTERFACE

The interface between the microprocessor and a memory array is required to perform three basic functions:

(i) Ensure electrical compatibility between the bus and the memory array.

(ii) Generate the memory-cell address.

(iii) Interact with the bus control signals to synchronise the data transfer to and from memory with the operation of the bus.

The complexity of the interface is determined by the type of memory array and structure of the bus system used in the microcomputer. Memory interfaces may be classified into three types:

(i) *The memory interface chip.* At the time of writing, manufacturers of solid-state general-purpose memory chips have

achieved some degree of standardisation in the organisation of the memory array (usually in 4-bit or 8-bit words) and, at least functionally, in the external connections. Chips with separate three-state data and address buses, and two control signals – chip select and mode control are common. Many microprocessor systems use a standard memory interface chip which is designed to link the general-purpose memory chips to their bus system. Figure 4-13 illustrates the use of a memory interface chip in a single bus microprocessor system with a single general-purpose read/write memory chip. Memory interface chips for dynamic memories also include automatic refresh control logic (see Chapter 6).

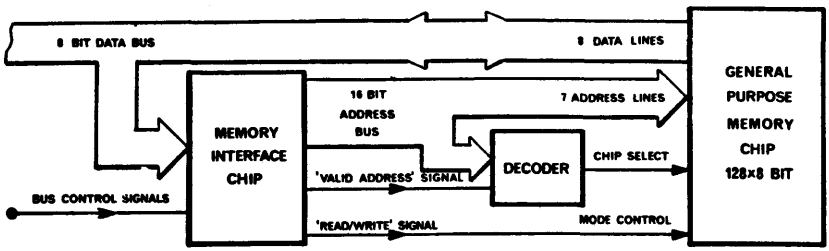

Figure 4-13. Memory interface chip

(ii) *The special-purpose memory array with on-chip bus interface.* Because the maximum number of external pin connections per chip was very limited in the early single bus microprocessor systems, the rather complex bus multiplexing and demultiplexing logic required in the memory interface was often included with the memory array on the same integrated circuit. The same approach has been adopted in many of the small low-cost microcomputers where a minimisation of the total chip count is important. Figure 4-14 illustrates a typical special-purpose chip for a single bus microprocessor system. In some cases, even the logic circuits associated with memory access operations which are normally located on the microprocessor chip (e.g., program counter, stack pointer, memory address register, address auto increment/decrement) are also included on the memory chip. The distinction between the microprocessor chip and the memory chip becomes less clear in these systems. It is also common for special-

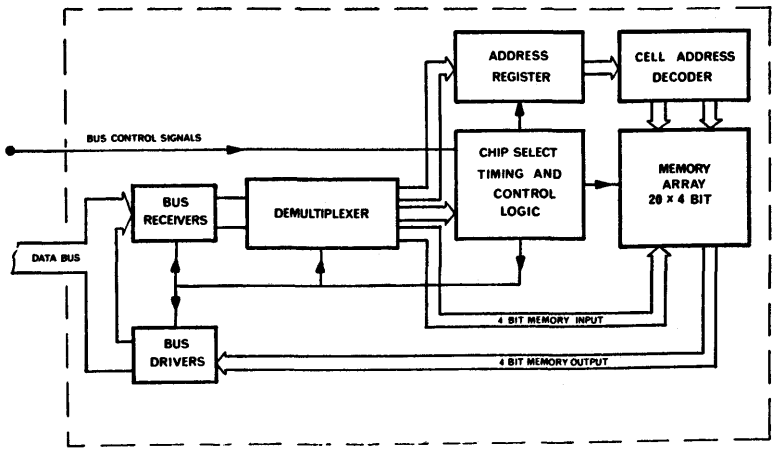

Figure 4-14. Special-purpose memory chip

purpose memory chips to have input/output lines, real-time clock circuits and interrupt control logic included on-chip.

The bus interfacing problem is very much simpler in a microprocessor system with separate data and address buses. Here the "special-purpose" memory chips only differ from the general-purpose memory chips in the chip select logic. As shown in Figure 4-15, additional interface logic is included on-chip to decode the most significant of the address bus lines so that a chip select signal may be generated.

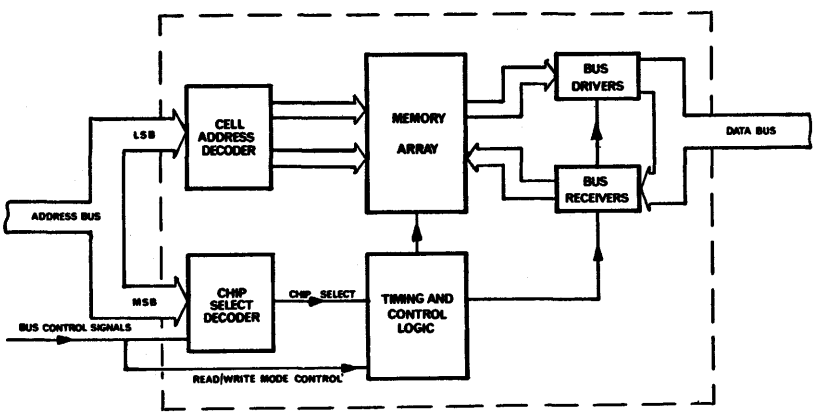

Figure 4-15. Modified general-purpose memory chip

(iii) *The special-purpose memory interface.* The memory interface will be relatively simple in a multiple bus microcomputer which has little or no multiplexing of information on the bus. In these systems the user can design his own memory interface, using standard random logic components to meet particular system requirements. Microcomputer systems which use large memory arrays or include direct-memory-access facilities are often more effectively designed using a special-purpose memory interface. The approach is unavoidable where the technology or structure of the memory array is fundamentally incompatible with that of the bus system. An example of the design and operation of a special-purpose memory interface is given in Chapter 6.

INPUT-OUPUT/BUS INTERFACES

Although the basic function of the I/O bus interface is similar to that of the memory interface, the design is often functionally more complex. There are two main reasons for the additional complexity. Firstly, the majority of I/O devices operate at relatively slow speeds in real-time and demand joint asynchronous control of the data flow through the interface. Secondly, many I/O devices are electro-mechanical in operation and cannot provide the degree of signal standardisation and bus compatibility found in semiconductor memory devices. In addition, some form of signal or code conversion is often required in the interface prior to transmission of data to or from the device.

There are many approaches to the design of I/O interfaces ranging from the use of special-purpose device-orientated input/output chips, where all the interface functions are performed on-chip in hardware, to designs based on simple general-purpose input/output circuits, where the special-purpose interface functions are realised in software. Five different types of I/O interfaces may be defined:

(i) *The parallel I/O port.* The I/O port is the most basic I/O interface and is little more than a simple I/O bus, either unidirectional or bidirectional, with latched three-state or open collector drivers on its outputs. Ports are usually 4 or 8 bits wide and are normally located on the microprocessor chip itself or on a special-purpose memory chip. Data transmission through the port is under

synchronous control of the microprocessor and there are often no bus control signals associated with a unidirectional port. To avoid external multiplexing hardware, several ports are frequently used in the design of a single I/O interface. Figure 4-16 illustrates the use of four ports in the design of a paper-tape reader interface. The timing and control of the reader is determined by software.

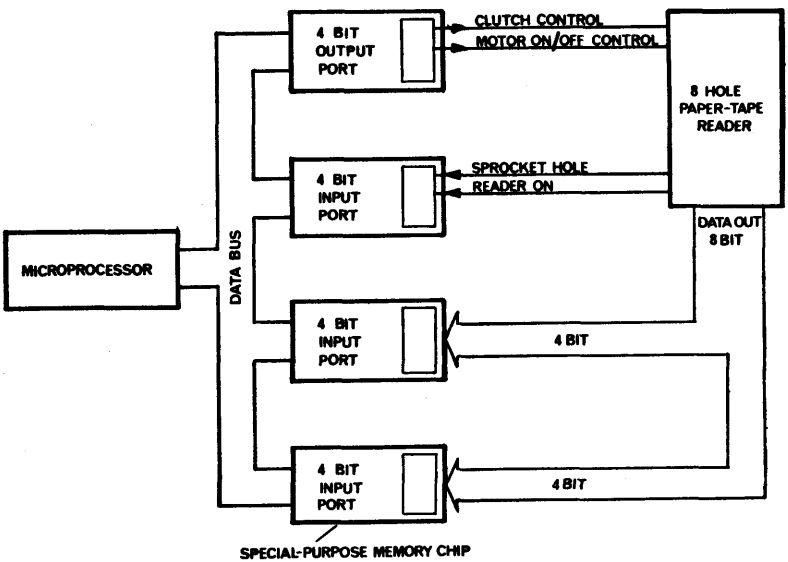

Figure 4-16. Simple paper-tape reader interface using multiple I/O ports

(ii) *The serial I/O port.* The port consists of two single-bit serial data buses linked to the serial input and serial output lines of a shift register (8 or 16 bits long). The shift register, which is usually located on the microprocessor chip, can be loaded or read via parallel input and output lines, or shifted one bit at a time, under program control.

As shown in Figure 4-17, the serial I/O port is often used as a simple asynchronous serial data communications interface for a teleprinter. The shift register performs the parallel-to-serial and serial-to-parallel code conversion under software control. Alternatively, as shown in Figure 4-18, the operation of the port can be synchronised to that of the address bus to provide a number of parallel I/O ports.

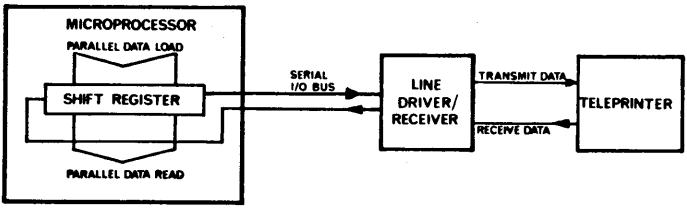

Figure 4-17. Serial I/O port used as a teleprinter interface

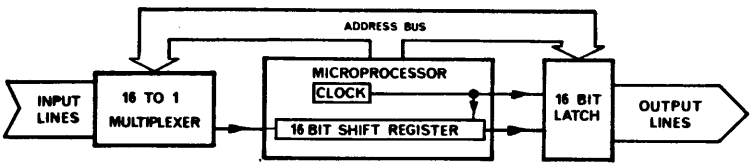

Figure 4-18. Parallel input and output using a serial I/O port

(iii) *The general-purpose bus interface.* There are three basic types:

The *non-programmable* input interface, shown in Figure 4-19, performs the basic bus interface functions and includes some interrupt request control logic. It is similar in operation to the parallel I/O port described earlier.

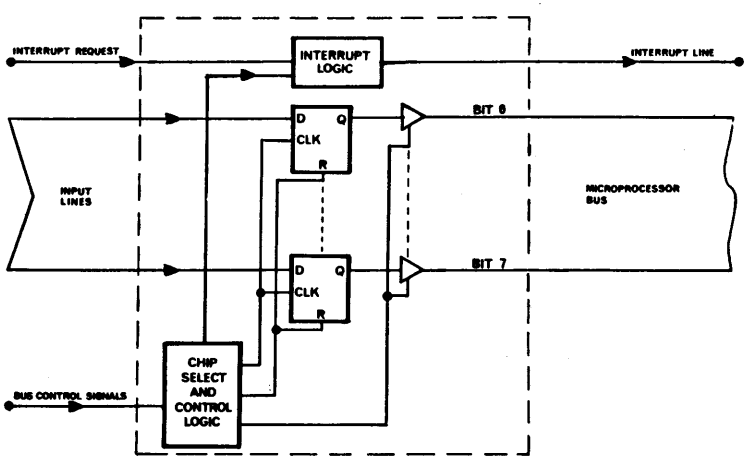

Figure 4-19. Non-programmable input interface

The *hardware programmable* interface, shown in Figure 4-20, typically includes decoding logic, several separately addressable parallel I/O ports and interrupt control circuits. External wiring determines the address, data direction and width of each port and controls the operation of the interrupt circuitry.

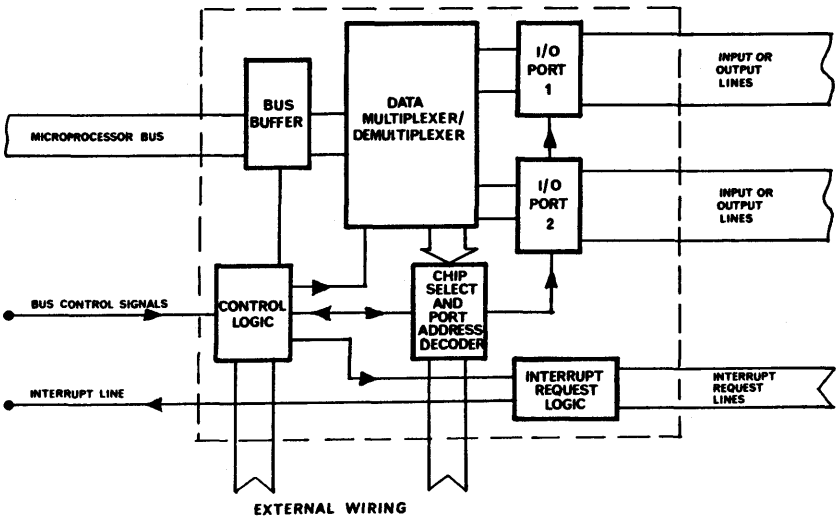

Figure 4-20. Hardware programmable I/O interface

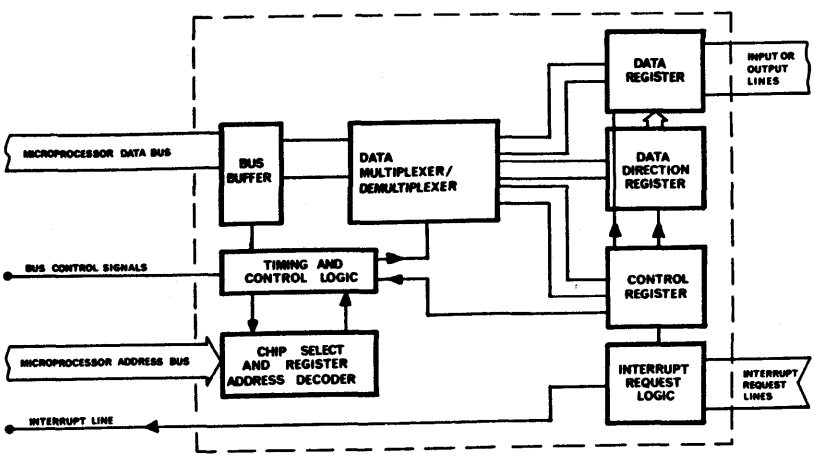

Figure 4-21. Software programmable I/O interface

The *software programmable* interface determines the structure and operation of the interface according to the contents of a control register which can be loaded by program. The interface shown in Figure 4-21 includes a second control register (the data direction register) which allows the function of individual I/O lines to be selected by software.

(iv) *The special-purpose bus interface.* The interface consists of one or more integrated circuits which are specially designed to link the microprocessor to a particular I/O device. In addition to the basic bus interface circuitry, the interface contains circuits to perform specific functions peculiar to the I/O device. For example, a keyboard interface would include matrix encoding and parity generation circuits; a lineprinter interface would include message formatting, character storage and print-head control and timing circuits; a communications interface would include serial-to-parallel/parallel-to-serial conversion, parity generation, message formatting and MODEM control circuits, etc. Unfortunately the lack of standardisation amongst the same type of I/O devices made by different manufacturers, and the high cost of developing special-purpose integrated circuits, has severely limited the number of I/O devices for which special-purpose interfaces are available.

(v) *The intelligent bus interface.* Intelligent (program-controlled) bus interface circuits were introduced to reduce the

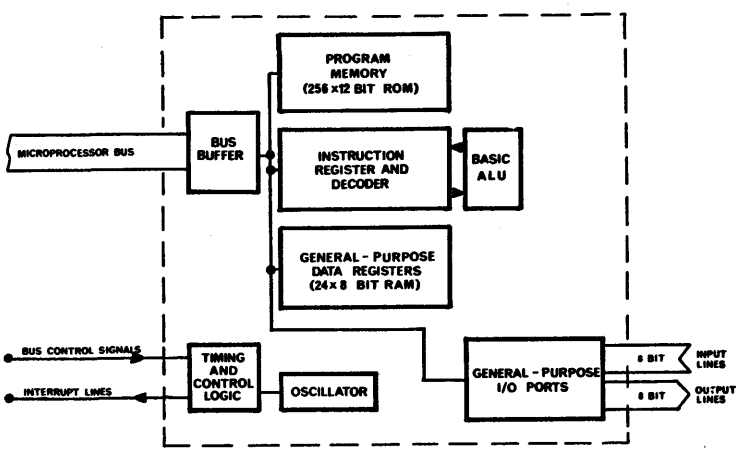

Figure 4-22. Intelligent interface

development costs associated with special-purpose interfaces. The intelligent interface, shown in Figure 4-22, is a simple single-chip microcomputer which has been specially designed with a limited instruction set to emphasise I/O control and interface operations instead of the usual general-purpose computing operations. The microcomputer has on one chip an external bus which is directly compatible with that of the host microprocessor system, extensive I/O facilities, and small data and program memories. The system is preprogrammed during manufacture to perform the interface functions required by a particular I/O device.

Off-the-shelf intelligent interfaces are available for a number of common peripherals (keyboards, magnetic tape cassette drives, printers etc.). General-purpose intelligent interfaces, which have user-programmable read-only memory on the microcomputer chip, can be used to meet the specific requirements of an I/O device made by a particular manufacturer.

SUMMARY

The bus structure of most microprocessors is determined by the integrated-circuit manufacturer and is fixed as far as the user is concerned. Nevertheless, it is important to understand the various types of bus structures that exist and their relative merits, because this plays a significant part in choosing a particular microprocessor and in designing the microcomputer system.