# Single-chip Microcomputers

Edited by
PAUL F. LISTER

# Single-chip Microcomputers

Edited by
PAUL F. LISTER





**GRANADA**
London Toronto Sydney New York

Granada Technical Books
Granada Publishing Ltd
8 Grafton Street, London W1X 3LA

First published in Great Britain by
Granada Publishing 1984



*British Library Cataloguing in Publication Data*
Single-chip microcomputers.
1. Microcomputers 2. Microprocessors
I. Lister, Paul F.
621.3819'58 TK7888.3

ISBN 0-246-12106-8

Printed and bound in Great Britain by
William Collins Sons & Co Ltd

# Preface

Single-chip microcomputers, or microcontrollers as they are sometimes called, are of major significance to applications engineers. These more dedicated devices significantly outstrip general-purpose microprocessors in terms of dollar volume. Shipments of single-chip microcomputers have been predicted to grow at a compound rate of 77% per annum to a level of nearly 3 billion dollars in 1985 (source: Creative Strategies International). This rate is faster than for other integrated components.

Semiconductor manufacturers produce a variety of single-chip microcomputers ranging from simple, cheap, 4-bit microcontrollers to comparatively powerful devices with capabilities rivalling the most powerful general-purpose processor chips. The major impact of microelectronics on products is coming from single-chip microcomputers. These devices are used in products as diverse as hand-held games, washing machines, cars, televisions and computer peripherals.

The central objective of this book is to provide engineers, technicians and engineering managers with a broad view of the types of device available, their characteristics and some insight into the sorts of applications that different devices have been used in. Applications material is important because while the detail may be specifically directed to one device, central issues such as interfacing strategies are generally applicable. Many potential users of single-chip microcomputers have little electronics experience (a typical use of these devices is as timer/controllers to replace electromechanical components). It is hoped that the material presented here will go some way to raise the level of confidence of these potential users.

Chapter 1 provides an introduction to the world of single-chip microcomputers. Relatively little prior knowledge is assumed; a reader with some elementary familiarity with the architecture of popular microprocessors such as the Intel 8080, Zilog Z80 or Rockwell 6502 found in many personal computers and single-board educational systems, will have little difficulty with this chapter. A historical overview is followed by an overview of the structure of microcomputers. A section on special architecture microcomputers is included since devices of this type are of growing importance. The rest of this chapter is a discussion of the major issues that arise in designing a microcomputer into a product.

Chapters 2 to 6 are manufacturer specific chapters, each discussing one manufacturer's microcomputers. Each chapter presents an introduction to the major family members and discusses applications. Most include detailed application material that will prove valuable to applications engineers who will be designing microcomputer based products. This material along with much of the discussions of architecture, peripheral devices and programming issues, provides a broad background of insight and experience that will be valuable regardless of the specific microcomputer context in which an applications engineer is working.

Chapter 2 discusses the Motorola M6801 and M6805 families. The comprehensive presentation of the instruction set and hardware features provides a good general introduction for the reader who has little background in these areas. The case study of a speedometer/tachometer/odometer is presented in sufficient detail to be a valuable introductory example of applications engineering.

Chapter 3 presents an overview of the Texas Instruments TMS1000 and TMS7000 series. The TMS1000 is currently the most pervasive device, widely used in high volume low-cost applications.

Chapter 4 presents the relatively advanced Zilog Z8 in some detail along with variants such as the Z8671 with a BASIC interpreter in ROM and various Z8 family peripheral devices.

Chapter 5 presents the National Semiconductor COPS 400 microcontrollers. These devices are biased towards high volume, low-cost applications. A case study of a digital TV tuning system is discussed.

Chapter 6 presents an overview of the Mostek MK68200, an advanced 16-bit microcomputer based on the powerful architecture of the MK68000 microprocessor. The capabilities of this device in the context of robotic control systems is discussed.

A book such as this is only possible through the efforts of the contributing authors, Graham Livey, Chris Moller, Brian Jasper, Reinhold Hohol, Alan Gant, Patrick McGehearty, Denise Burrows and Peter Vinson.

I wish to thank the microcomputer manufacturers, Motorola, Texas Instruments, Zilog, Intel, National Semiconductor and Mostek for their co-operation and permission to reproduce various figures. Dr Livey would like to thank M. Ritchie, M. Catherwood, R. Bettelheim and J. R. Livey for their help with Chapter 2. Thanks are also due to Bernard Watson of Granada for his assistance and to my wife, Muriel Lister, for typing and secretarial services.

Paul F. Lister
*University of Sussex*

# Contents

*Reprinted with permission from *Electronic Design*, vol. 31, No. 11; copyright Hayden Publishing Company, Inc., 1983.
†No longer at Mostek.

CHAPTER 1

# Microcomputer Fundamentals

*By Paul F. Lister*

## 1.1 EVOLUTION OF THE MICROCOMPUTER

The single-chip microcomputer is the culmination of both the development of the digital computer and the integrated circuit, arguably the two most significant inventions of the century.

Digital computers have developed from a historical background of calculating machine evolution spanning several hundred years. It is generally accepted that the major milestone in the development of the first computer was the Analytical Engine designed by Charles Babbage in around 1837. This mechanical design, although never fully implemented, had all the major characteristics of a modern computer. It was, however, some hundred years later that the developments that led directly to the modern digital computer took place.

Howard Aiken pioneered the development of a calculating machine based on relays, the Mark 1 built for IBM in 1944, at Harvard. The first electronic computer ENIAC (Electronic Numerical Integrator and Calculator) followed in 1946 developed by John Mauchly. This slow, primitive machine was based on some 18,000 valves (vacuum tubes) and weighed 30 tons. At this time a computer called EDVAC (Electronic Discrete Variable Automatic Computer) was being developed by John von Neumann and others at Princeton; it was completed in 1951. The major concept introduced by von Neumann was that of combining the separate program and data stores of the Aiken machine and hence storing both program and data in the same format so that machine instructions could be processed by the machine itself. This ability for the machine to manipulate its own instructions has facilitated the development of higher-level languages for computer programming.

These two types of architecture are found in single-chip microcomputers. Some employ the split program/data memory of the Harvard architecture, shown in Fig. 1.1, others follow the philosophy, widely adopted for general-purpose computers and microprocessors, of making no logical distinction between program and data memory as in the Princeton architecture, shown in Fig. 1.2.

The second generation of computers followed the development of the transistor in 1948. As larger machines evolved through the 1960s with more sophisticated operating systems, the third generation evolved.

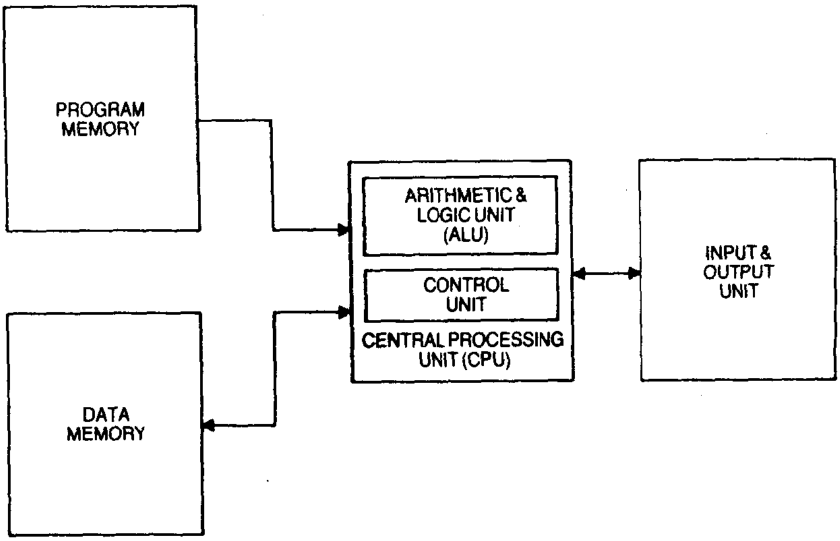


**Fig. 1.1** A Harvard type computer.

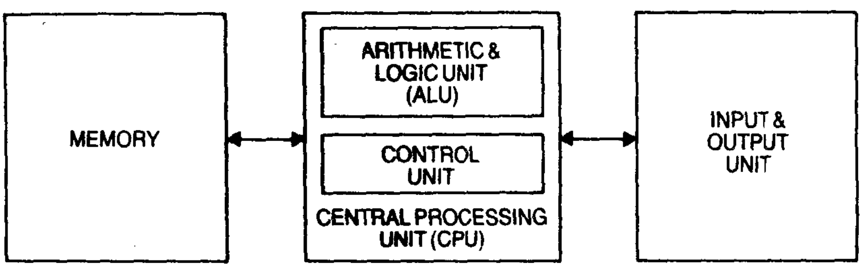


**Fig. 1.2** A conventional Princeton computer.

The integrated circuit, involving the placing of more than one transistor on a piece of semiconductor, was the cornerstone of this development. These early integrated circuits had only a few (~20) transistors connected to form a module. However, as fabrication techniques developed it was possible to place more and more components in an integrated circuit. Small-scale integrated (SSI) circuits led to medium-scale integrated (MSI) devices (a few hundred transistors) and then around 1970 to large-scale integrated (LSI) circuits with a few thousand transistors in a circuit. The application of LSI devices to computer design has given rise to a fourth generation of computers. The next generation, the fifth, based on genuine very-large-scale integrated (VLSI) circuits and artificial intelligence methods, is currently the subject of intense worldwide development stimulated by a Japanese initiative.

| | |
|---|---|
| 1971 | Intel 4004 4-bit processor |
| 1972 | Intel 8008 8-bit processor |
| | Rockwell PPS4 4-bit processor |
| 1974 | Intel 8080 8-bit processor |
| | RCA 1802 8-bit processor (CMOS) |
| | Motorola 6800 |

| | **(Microcomputers)** | **(Microprocessors)** |
|---|---|---|
| | Texas Instruments TMS 1000 | |
| 1976 | Intel 8048 | Zilog Z80 |
| | | Intel 8085 |
| | National Semiconductor COP 400 | Texas TMS 9900 |
| 1978 | Intel 8051 | Intel 8086 |
| | Motorola MC6801 | |
| | Zilog Z8 | Zilog Z8000 |
| 1980 | | Motorola 68000 |
| 1982 | | National Semiconductor 16032 |
| | Intel 2920 | Intel iAPX 432 |
| | Texas Instruments TMS 320 | |
| | | Zilog Z80,000 |

**Fig. 1.3** Microcomputer evolution.

The LSI technology of the early 1970s enabled a small computer to be built using just a few integrated circuits. This was recognised by Marcian E 'Ted' Hoff Jr. who was working at Intel on a programmable calculator project and who is generally credited as being the inventor of the microprocessor. This led to the development of the Intel 4004 CPU – a 4-bit processor implemented using about 2,300 transistors. This and the development of the EPROM (erasable programmable read only memory) were the pivotal developments in the microprocessor revolution.

The Intel 8008, an 8-bit processor followed in 1972. In 1974 Intel introduced the 8080 which, along with the Zilog Z80 (introduced in 1976), has been widely adopted in personal computers. Around this time Texas Instruments introduced their 4-bit single-chip microcomputer, the

TMS 1000, which has become widely used in consumer products such as toys and games; devices such as this that are widely used as logic replacement are often called microcontrollers. This type of device reflects the beginning of the dichotomy that separates single-chip microcomputers from microprocessors. A manufacturer of large integrated circuits is faced with the problem of how best to use the circuit complexity available. At a given state of fabrication technology it is possible to design a relatively powerful processor or a less powerful processor and include the other functional units necessary to provide a complete microcomputer. Figure 1.3 illustrates this with some approximately contemporary examples.

## 1.2 STRUCTURE OF A MICROCOMPUTER

In general terms a single-chip microcomputer is characterised by the incorporation of all the units of a computer, as shown in Figs 1.1 or 1.2, into a single device. Figure 1.4 illustrates the principal features found in many microcomputers. The characteristics of these features are discussed in outline in the following sections as a general introduction to the aspects of particular facilities that are discussed in subsequent chapters.

### 1.2.1 Read only memory (ROM)

ROM is usually for the permanent, non-volatile storage of an applications program. Many microcomputers and microcontrollers are intended for high-volume applications and hence the economical manufacture of the devices requires that the contents of the program memory be committed permanently during the manufacture of the chips. Clearly, this implies a rigorous approach to ROM code development (see section 1.5.5) since changes cannot be made after manufacture. This development process may involve emulation using a sophisticated development system with a hardware emulation capability as well as the use of powerful software tools.

Some manufacturers provide additional ROM options by including in their range devices with (or intended for use with) user programmable memory. The simplest of these is usually a device which can operate in a microprocessor mode by using some of the input/output lines as an address and data bus for accessing external memory. This type of device can behave functionally as the single-chip microcomputer from which it is derived albeit with restricted I/O and a modified external circuit. The use of these ROMless devices is common even in production circuits where the volume does not justify the development costs of custom on-chip ROM; there can still be a significant saving in I/O and other chips compared to a conventional microprocessor based circuit. More exact

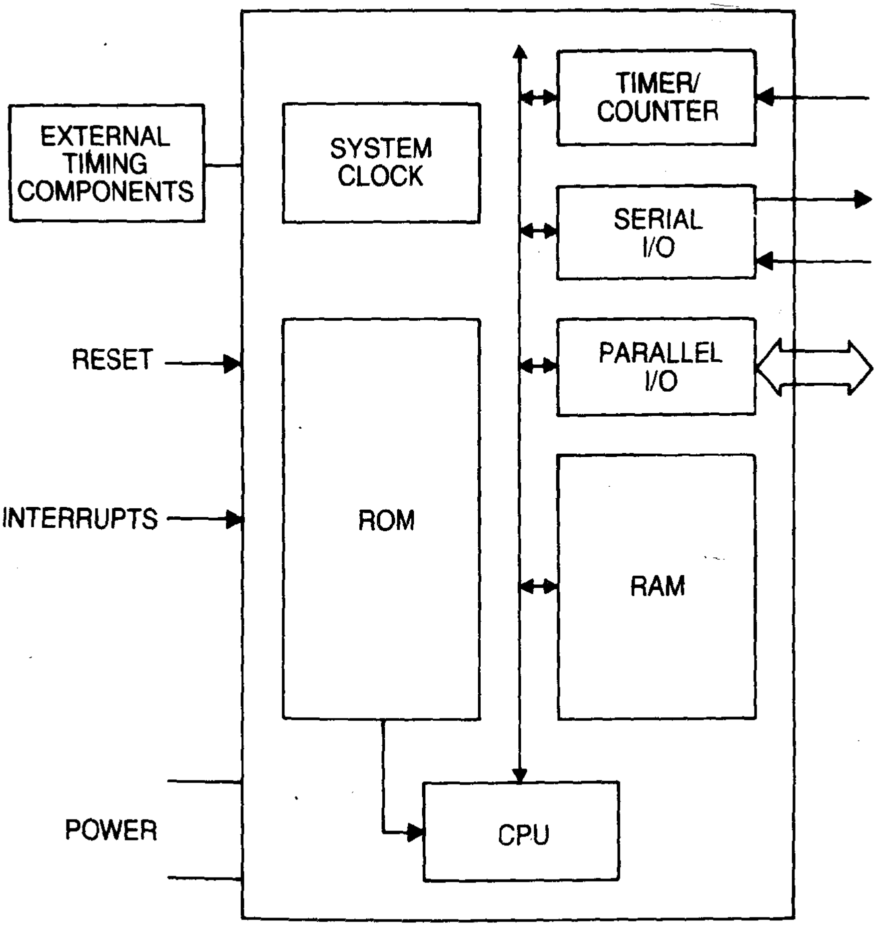


**Fig. 1.4** Principal features of a microcomputer.

replacements for ROM devices can be obtained in the form of variants with 'piggy-back' EPROM (Erasable Programmable ROM) sockets or devices with EPROM instead of ROM. These devices are naturally more expensive than the equivalent ROM device, but do provide complete circuit equivalents. EPROM based devices are also extremely attractive for low-volume applications where they provide the advantages of a single-chip device, in terms of on-chip I/O, etc., with the convenience of flexible user programmability.

### 1.2.2 Read/write memory (RAM)

RAM is for the storage of working variables and data used during program execution. The size of this memory varies with device type but it has the same characteristic width (4, 8, 16 bits, etc.) as the processor. Special function registers, such as a stack pointer or timer register are often logically incorporated into the RAM area. It is also common in Harvard type microcomputers to treat the RAM area as a collection of registers; it is unnecessary to make a distinction between RAM and processor registers as is done in the case of a microprocessor system since RAM and registers are not usually physically separated in a microcomputer.

### 1.2.3 Central processing unit (CPU)

The CPU is much like that of any microprocessor. Many applications of microcomputers and microcontrollers involve the handling of binary-coded decimal (BCD) data (for numerical displays, for example), hence it is common to find that the CPU is well adapted to handling this type of data. It is also common to find good facilities for testing, setting and resetting individual bits of memory or I/O since many controller applications involve the turning on and off of single output lines or the reading of a single line. These lines are readily interfaced to two-state devices such as switches, thermostats, solid-state relays, valves, motors, etc.

### 1.2.4 Parallel input/output

Parallel input and output schemes vary somewhat in different microcomputers; in most a mechanism is provided to at least allow some flexibility of choosing which pins are outputs and which are inputs. This may apply to all or some of the ports. As an example of this, Fig. 1.5 shows the internal circuit arrangement for an Intel 8048/8051 bidirectional I/O pin. Writing a '1' to the pin causes Q2 to turn on for a short time providing fast totem-pole TTL compatible pull-up of the output. However, the pin is held up by R1 which is about 50 kilohms; large enough to allow an external output connected to the pin to pull it down if the pin is an input. Hence to use a pin as an input simply requires that a '1' be written out to it to ensure that Q1 is off.

Some I/O lines are suitable for direct interfacing to, for example, fluorescent displays, or can provide sufficient current to make interfacing to other components straightforward.

Some devices allow an I/O port to be configured as a system bus to allow off-chip memory and I/O expansion. This facility is potentially useful as a product range develops, since successive enhancements may become too big for on-chip memory and it is undesirable not to build on the existing software base.

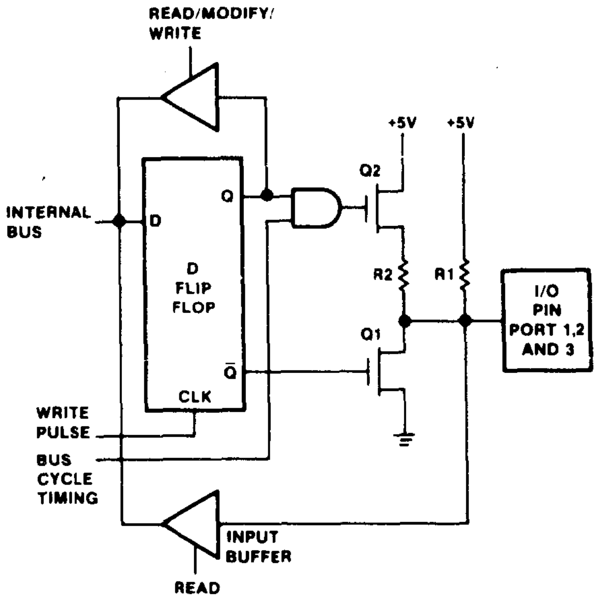


**Fig. 1.5** Intel pseudo-bidirectional I/O port circuitry (Source: Intel Corp.).

### 1.2.5 Serial input/output

Serial communication with terminal devices is a common means of providing a link using a small number of lines. This sort of communication can also be exploited for interfacing special function chips or linking several microcomputers together. Both the common asynchronous and synchronous communication schemes require protocols that provide framing (start and stop) information. This can be implemented as a hardware facility or U(S)ART (Universal (synchronous) asynchronous receiver/transmitter) relieving the processor (and the applications programmer) of this low-level, time-consuming, detail. It is merely necessary to select a baud-rate and possibly other options (number of stop bits, parity, etc.) and load (or read from) the serial transmitter (or receiver) buffer. Serialisation of the data in the appropriate format is then handled by the hardware circuit.

### 1.2.6 Timer/counter facilities

Many applications of single-chip microcomputers require accurate evaluation of elapsed real time. This can be determined by careful assessment of the execution time of each branch in a program but this rapidly becomes inefficient for all but the simplest programs. The preferred approach is to use a timer circuit that can independently count precise time increments and generate an interrupt after a preset time has elapsed. This type of timer is usually arranged to be preloadable with the required count. The timer then decrements this value producing an interrupt or setting a flag when the counter reaches zero. Better timers then have the ability to automatically reload the initial count value. This relieves the programmer of the responsibility of reloading the counter and assessing the elapsed time before the timer is restarted, which otherwise would be necessary if continuous precisely timed interrupts were required (as in a clock, for example). Sometimes associated with a timer is an event counter. With this facility there is usually a special input pin, that can drive the counter directly.

### 1.2.7 Timing components

The clock circuitry of most microcomputers requires only simple timing components. If maximum performance is required, a crystal must be used to ensure the maximum clock frequency is approached but not exceeded. Many clock circuits also work with a resistor and capacitor as low-cost timing components or can be driven from an external source. This latter arrangement is useful if external synchronisation of the microcomputer is required.

## 1.3 4-, 8- AND 16-BIT DEVICES

Microcomputers and computers in general are usually classified in terms of the characteristic width of the internal data paths. This width, 4, 8, 16 or 32 bits, has a great bearing on the performance of the device in specific applications. There are currently no 32-bit single-chip microcomputers, but this is perhaps only a matter of time; such devices would form the heart of high-performance, low-cost personal and small business computers.

The data path width inside a microcomputer implies that there is circuitry to manipulate that number of bits in parallel during an operation. Circuit complexity is therefore virtually a linear function of this width. It is usually more economical in chip area and hence yield and production costs to produce 4-bit devices rather than 8 or 16-bit ones. However, 4-bit devices, for example, are not suited to all types of application.

It is a feature of practically all reasonable general-purpose computer architectures that, regardless of data path width, any computational problem can be solved, in a logical sense, by any processor. The crucial issue is the time it takes and, to a lesser extent, the ease of programming it.

Four-bit devices are widely used in calculators and small controllers. This type of device can handle binary coded decimal (BCD) data one decimal digit at a time. Information involving more than one decimal digit must be processed in a digit serial manner. Individual bits can be manipulated readily for I/O purposes, but address processing with only 4-bits is difficult. These devices are therefore suited to low end applications involving relatively small amounts of data driven by modest scale programs operating within a comparatively relaxed time scale. Devices such as these satisfy many application requirements and they represent the largest single category of microprocessor type components. Texas Instruments claim to have shipped in excess of sixty million of their TMS1000 family 4-bit microcomputers.

Eight-bit microcomputers represent, for many more demanding applications, a sensible compromise in terms of performance and on-chip circuit utilisation. With 8 bits, two BCD characters can be manipulated simultaneously. Eight bits is necessary for the convenient handling of alphanumeric data in the ASCII code widely used in terminals, displays and character handling in general. Reasonable precision arithmetic is possible with 16-bit data which is not too onerous with an 8-bit processor.

This is also a convenient size for instruction op-codes and addresses can be specified with only one or two 8-bit words. Hence it is feasible to modify the strictly Harvard architecture to allow a measure of flexible interaction between program and data memories.

Sixteen-bit devices can offer the general-purpose computing capabilities of a small minicomputer and the regularity of architecture and instruction set found in microprocessors such as the Zilog Z8000, Motorola 68,000 or the National Semiconductor 16032. This size of data path permits alphanumerical data in ASCII, etc., to be packed at two characters per word or BCD packed at four digits per word.

Sixteen bits is much more appropriate for signal derived data. Digital representations of signals correspond (if properly scaled) to a signal-to-noise power ratio of approximately 6 dB per bit of representation. This means that 8 bits allows a maximum of around 50 dB signal-to-noise ratio. This is not very good in analogue circuit terms and there is no room for adding the signal samples together, for example, without continuously having to rescale. Twelve bits is more appropriate for most applications (~70 dB) and a 16-bit processor can handle this data directly with some headroom. Clearly, 12- or 16-bit data can be manipulated with an 8/bit (or 4-bit) processor, but with a time penalty. It should be borne in mind that double word length operations often involve more than just the required operation applied to each half of the data in turn. Intermediate carries have to be managed and appropriately different sign management undertaken for each part of the data. This can mean that double word length operations take significantly more than twice the execution time of a single length operation.

Thirty-two-bit processors offer the capability of handling a good resolution floating-point number as a single datum as well as the vast addressing and comprehensive processing capabilities normally associated with these microprocessors.

## 1.4 SPECIAL PURPOSE MICROCOMPUTERS

Some applications demand performance that is not available with a general-purpose architecture; this has given rise to a group of microcomputers with architectures adapted to a particular type of task. Signal processing is one such area and devices suited to this are finding application in telecommunications and instrumentation systems.

The digital processing of analogue signals is about performing operations on digital representations of analogue data. This invariably requires real-time processing involving operations that general-purpose architectures are not particularly optimised for. Specifically, multiplication is often central to the processing required.

For example, digital filters are usually implemented as a summation of previous stored values each weighted by a coefficient. Hence in the time interval between one input sample and the next many multiplications may be required.

### 1.4.1 Intel 2920

The Intel 2920 is a radical departure in the development of single-chip microcomputers. It is the first major deviation for a device of this type from a general-purpose architecture. As such it is surely the first of many such devices to come. The 2920 has an architecture specifically arranged to permit the fast, real-time, handling of analogue signals.

A block diagram of the 2920 is shown in Fig. 1.6 and Fig. 1.7 shows the analogue section in more detail. Four analogue inputs and eight analogue outputs are provided, all are multiplexed from a single 9-bit digital to analogue converter. Figure 1.8 illustrates the arithmetic unit and data memory organisation. This area bears a close resemblance to the ALU and scratchpad memory arrangement of a bit-slice processor such as the Advanced Micro Devices Am2901. Note, however, that data words are 25 bits long.

Control of the RAM, shifter, ALU and analogue sections is managed in parallel at the microprogram level. In fact the basic instructions of the 2920 are microinstructions and they do not decode to sequences of lower level steps. All arithmetic operations are performed in two's complement representation as are shifts (the sign-bit is brought down).