Using A/D and D/A Converters with Microcomputers

Chapter 8

INTRODUCTION

Since the world in which we live is largely an analog one, virtually every computer has some form of interface circuit for converting signals from digital to analog form or vice-versa. This chapter is devoted to a discussion of the interface techniques used to connect analog and digital circuits to each other.

In using microcomputers, the complete system design is subject to many constraints. Three of these are:

(i) The microcomputer operates at a speed dictated by its own clock frequency, and not by the sequence in which external events occur.

(ii) Generally it is desirable to reduce the number of integrated-circuit packages and the number of connections, so as to reduce the overall cost and to improve system reliability.

(iii) Microcomputers are relatively slow and, where possible, hardware should relieve the microcomputer of time consuming tasks.

These three key design parameters occur repeatedly in this and the succeeding chapter.

DIGITAL-TO-ANALOG CIRCUITS AND THE MICROCOMPUTER INTERFACE

One of the fundamental principles in using output devices with a microcomputer is that the output information is latched. The

information, which the microcomputer sends to the output port, is placed in latches and retained until the microcomputer chooses to change the information at some later time. The microcomputer only addresses the output port at sporadic intervals and between these intervals the output device must be self-sustaining. Applying this principle to digital-to-analog converters, it can be seen that the ideal D/A converter should include a set of latches so as to hold the specified value until it is updated. However, this apparently simple requirement is not easily achieved because the technology for fabricating D/A converters and that for fabricating logic is not always the same, and therefore it is difficult to include the two on the same integrated circuit. One solution is to build D/A converters in hybrid form by mounting separate latch, switch and resistor circuits on a common substrate and interconnecting them, but this method is expensive. Fortunately, two monolithic techniques do lend themselves fairly well to single chip fabrication of D/A converters: these are I^2L logic, which is a form of logic using bipolar transistors and CMOS logic. I^2L technology has the advantage that high quality analog circuits and high speed logic can be included on the same chip. CMOS, on the other hand, is not suited to sophisticated analog circuits, but has the advantage of excellent analog switches, good logic capability and low power consumption. It should be noted that it is desirable to keep on-chip power dissipation low for D/A and A/D converters so as to minimise errors due to local heating of the R-2R ladder.

Clearly, the simple D/A converter is not quite as simple as it would appear. The microcomputer interface problem is compounded by the fact that most microcomputers to date are 8-bit machines, so that data can only be transmitted to output ports in 8-bit bytes. However, this only provides a bipolar accuracy of about 1% and this is inadequate for many analog applications. Consequently most D/A converters have either 10- or 12-bit resolution, and this requires that the output be transmitted as two consecutive data bytes. In doing this, the two bytes have to be re-assembled at the D/A converter as a single word and then presented to the ladder circuit. It is not appropriate to re-assemble the word using the latches directly connected to the D/A ladder switches because this would cause extreme glitches on the D/A output during the period

when the transfer of the two data bytes is half completed. Thus in order to connect a 10- or 12-bit D/A converter to an 8-bit microcomputer, it is necessary to use the multiple latches method as shown in Figure 8-1. This arrangement is known as a double-buffered D/A.

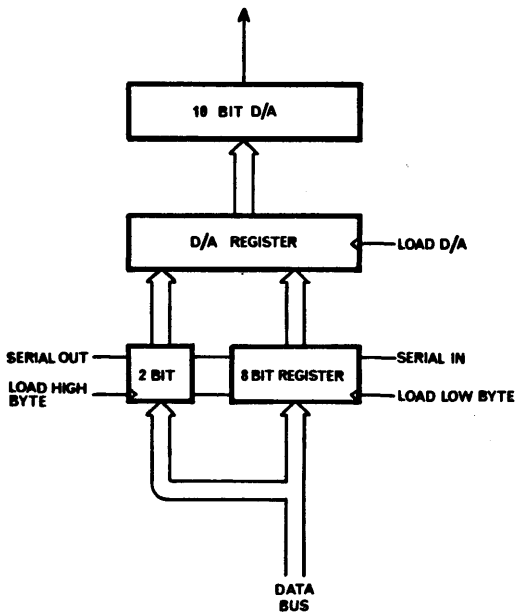

Figure 8-1. Double-buffered D/A converter (AD7522)

Separate clock inputs for the high and low byte latches are provided so that these can be strobed whenever the microcomputer sends information to them. A third clock input serves to load the latch set driving the D/A converter from the input latches, so that once a complete new word has been received it can be transferred "en bloc" to the main D/A circuit.

An alternative method of transferring information from a microcomputer to an output port is to do the data transfer in serial mode. Then only three interconnections are necessary: one line carries the output information, another provides the clock pulse for clocking the information into the output port which consists

of a shift register, and a third loads the shift register into the D/A register. This method has the advantage that it requires only three pins to connect any number of output ports to the microcomputer, and since this number is so low, it is economically viable to completely isolate the analog output from the digital circuits by means of optical isolators.

For multiple output devices the serial output mode can be utilised either by connecting all the output devices in a long string as shown in Figure 8-2 and then shifting in new information as one long word, or by using a demultiplexer to supply the clock and having common data and D/A register load strobes as shown in Figure 8-3.

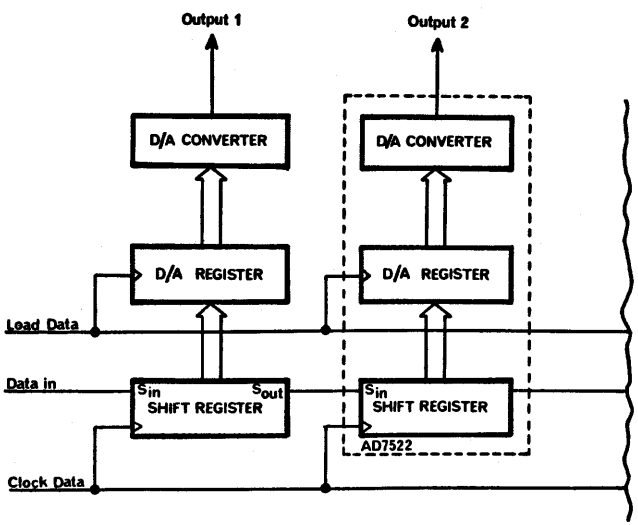

Figure 8-2. Loading multiple D/A converters in full serial mode

The circuit of Figure 8-2 has the disadvantage that a complete set of data has to be output each time one D/A converter is changed and the data transfer rate is consequently slow. With the circuit of Figure 8-3, data is shifted into the one device which is clocked from the demultiplexer, whilst all other devices remain unchanged thus allowing the overall data transfer rate to be higher. Since serial output to D/A converters uses few interconnections, it is

possible to obtain much higher packing density of D/A converters on printed circuit boards. The serial data link technique is also an important factor with single chip microcomputers because it minimises pin count, reduces overall systems costs and increases reliability.

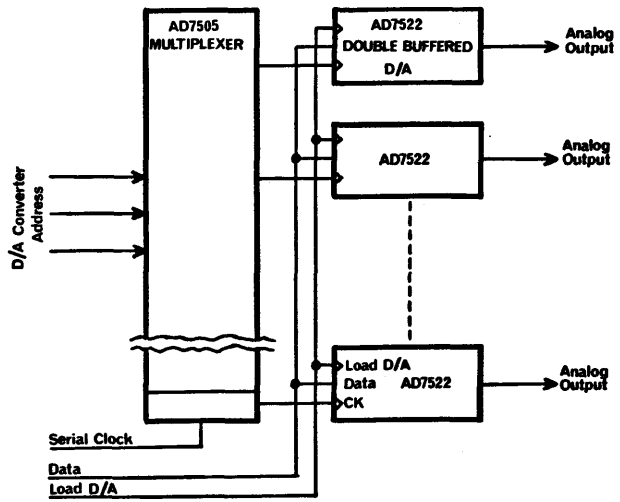

Figure 8-3. Loading multiple D/A converters in serial-parallel mode

MULTIPLE BYTE PARALLEL DATA TRANSFERS

The previous section referred to the fact that in order to parallel load a 10- or 12-bit D/A converter from an 8-bit microcomputer, the full word must be sent as two 8-bit bytes. The two bytes are re-assembled into a single word at the D/A converter. However, there are many ways in which 10 bits of information can be sent as two 8-bit bytes as shown below:

Word to be transmitted = $b_9b_8b_7b_6b_5b_4b_3b_2b_1b_0$

First Byte – Address A000	Second Byte – Address A001
(a) 0 0 0 0 0 0 b_9b_8	$b_7b_6b_5b_4b_3b_2b_1b_0$
(b) $b_7b_6b_5b_4b_3b_2b_1b_0$	0 0 0 0 0 0 b_9b_8
(c) $b_9b_8$0 0 0 0 0 0	$b_7b_6b_5b_4b_3b_2b_1b_0$
(d) $b_7b_6b_5b_4b_3b_2b_1b_0$	$b_9b_8$0 0 0 0 0 0

Figure 8-4 gives the connections to the AD7522 10-bit double-buffered D/A converter for two of these four cases.

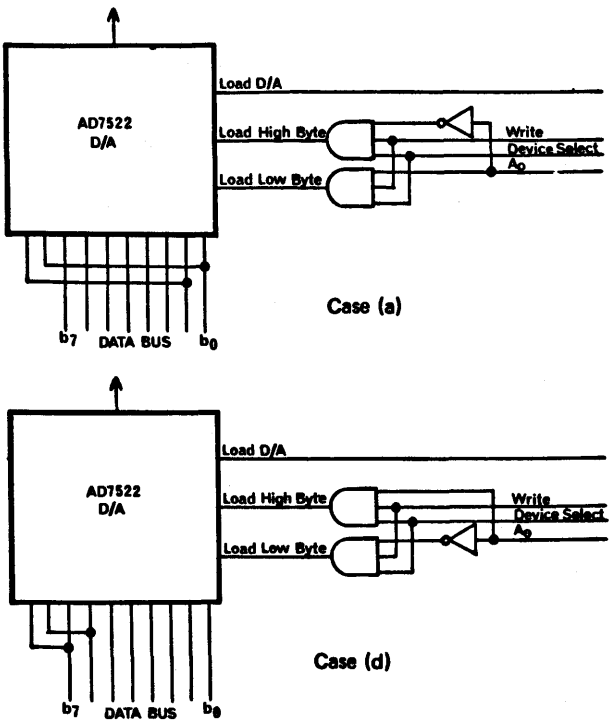

Figure 8-4. Multiple byte parallel loaded D/A converter

The various methods of transmitting multiple bytes arise as a result of the internal architecture of the microprocessor and the manner in which the data is generated and used within the microcomputer. Most 8-bit microprocessors have a single instruction which transfers a 16-bit data word as two 8-bit bytes. Unfortunately some transfer the data with the least significant byte first and others transfer the data with the most significant byte first – hence cases a and b above. In addition, some applications require the sign bit of the data (e.g., b_9) to be at the most significant end of the high data byte so that it can be easily shifted into the carry flag and examined with a "jump on carry" instruction – hence cases c and d.

FAST ANALOG-TO-DIGITAL CONVERTERS AND THE MICROCOMPUTER INTERFACE

All of the considerations discussed above also apply to A/D converters, but in addition, there is one other important system parameter – namely conversion time. The ideal A/D converter would yield a digital value immediately it is interrogated by the microcomputer. However, all A/D converters have a finite conversion time and this governs the way in which the converter can be used. Where the conversion time is short in relation to the instruction cycle time, the easiest method is to start the A/D conversion with one instruction, follow this by a number of dummy instructions (such as add zero to the accumulator) which allows time for the conversion to take place, and then read the result from the converter. Figure 8-5 shows the connections for connecting the AD7570 10-bit successive-approximations A/D converter to the Intel 8080 microprocessor. Address A000 (hex) is decoded and ANDed with the write signal, so that the instruction "Store accumulator A at A000 (hex)" is used to provide the

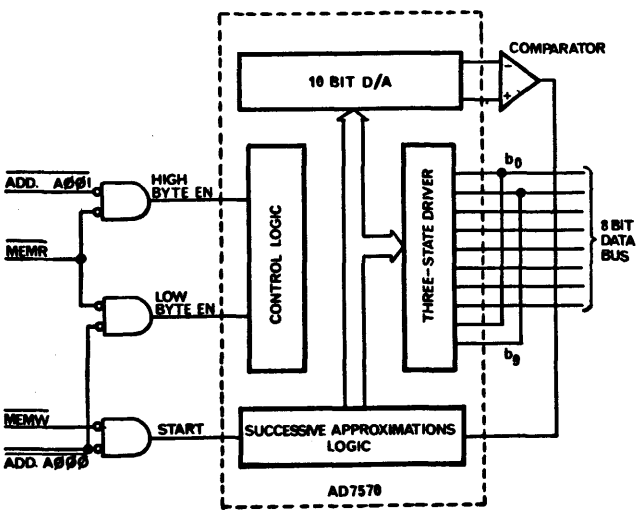

Figure 8-5. Successive-approximations converter connection to 8080 where dummy instructions are used to wait for the end of conversion

start signal. It does not matter that the microprocessor will attempt to write the accumulator contents into the A/D converter because the A/D converter will not accept them. The store instruction is followed by a number of "Clear accumulator" instructions which allow the AD7570 time to make the conversion, and finally the analog value is read from the A/D converter. An indirect addressing instruction is used to transfer the low order byte to the microprocessor, the index register is incremented to give address A001 and then the high order byte is transferred to the microprocessor where the two bytes are automatically assembled into a 16-bit word.

An alternative approach for using fast A/D converters with microcomputers is to connect the A/D converter so that the microprocessor sees it as a "slow memory." To start a conversion the microcomputer executes a read instruction at the address occupied by the A/D converter. This causes the converter to start conversion, and it issues a "busy" signal to the microcomputer. The microcomputer uses the busy signal to suspend operation until the signal disappears, whereupon the instruction cycle is resumed and the two bytes are transferred to the microprocessor as described above. The method of using the busy signal to temporarily halt the instruction cycle varies from one processor to another. Some (like the Intel 8080) insert dummy "do-nothing" sub-steps into the instruction cycle; others (like the National SC/MP) just stop and wait and some (like the Motorola 6800) stretch the microprocessor clock pulse. Unfortunately it is not always permissible to stretch the clock pulse sufficiently to use the method described above and it is important to check the microprocessor data sheet before using this technique. The "slow-memory" approach to A/D converter interface has the advantage that it uses a minimum of instructions and the value is returned to the microprocessor in the shortest possible time. The connections for using the AD7570 10-bit converter with an Intel 8080 are shown in Figure 8-6.

SLOW A/D CONVERTERS AND THE MICROCOMPUTER INTERFACE

If the methods described above for fast A/D converters are used with slow converters, then valuable computing time is lost whilst

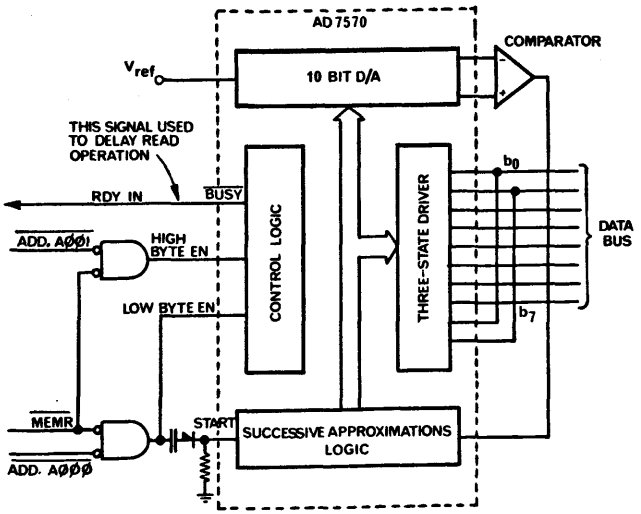

Figure 8-6. Using a successive-approximations converter as slow memory (connections shown for 8080)

the microcomputer waits for the digital value to become available. Therefore it is better to interface a slow A/D converter to the microcomputer so that, once the microprocessor has initiated the A/D conversion process, it can return to the main-line program to do other calculations until the conversion is complete. This is achieved by connecting the A/D converter so as to behave as an interrupting input device. Figure 8-7 shows the AD7550 quad-slope converter (see Chapter 7) connected to operate in this way: all outputs of the AD7550 are three-state logic. The busy line of the converter is connected as an interrupt to the microcomputer so that, when a conversion is complete, the converter interrupts the microprocessor and causes a jump to the A/D handling routine. As long as the converter is idle and has a digital value available at its output, the busy line will attempt to cause an interrupt. It is therefore necessary to use the microcomputer to mask-off the converter interrupt so that an interrupt is only permitted at the completion of a conversion. The program flow-chart is then as shown in Figure 8-8. Many of the present microcomputers do not have the facility for selectively masking-off the I/O devices and it

may be necessary to build an interrupt mask in hardware as shown in the dotted portion of Figure 8-7. The interrupt latch is set on completion of the A/D conversion and is reset when the last byte is read from the converter.

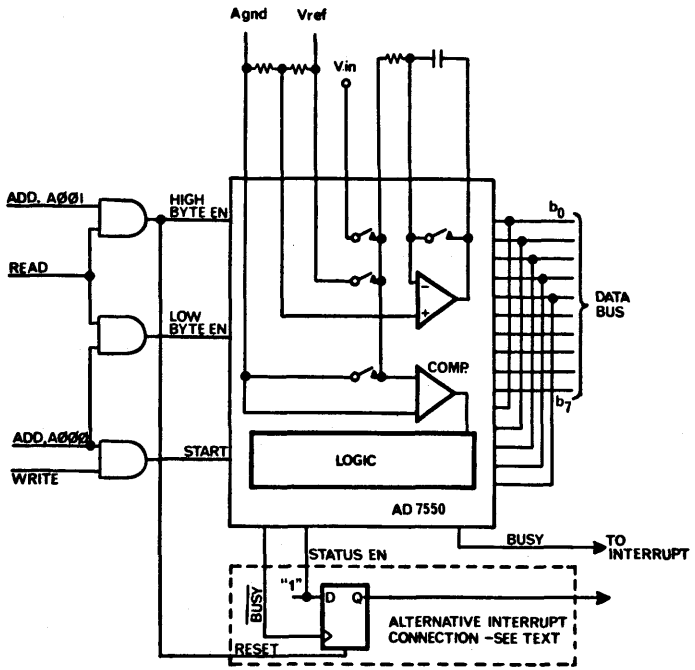

Figure 8-7. Connections for using an integrating A/D converter as an interrupting input device

It is of course possible to use the A/D converter with any of the I/O techniques described in Chapter 3. Program-controlled I/O is often used because it does not require a sophisticated interrupt structure and can be implemented with the minimum of extra hardware. Figure 8-9 shows the AD7550 connected for program-controlled I/O. Note that the two status lines "over-range" and "busy" are connected to the three-state data bus in the MSB and LSB positions so that they can easily be interrogated by reading the status word to the microprocessor accumulator and then shifting them into the carry flag.

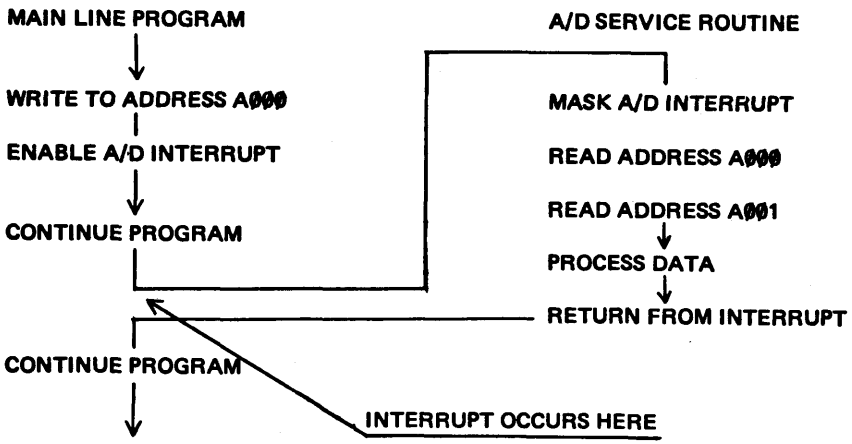

Figure 8-8. Program flow for handling interrupting A/D converter

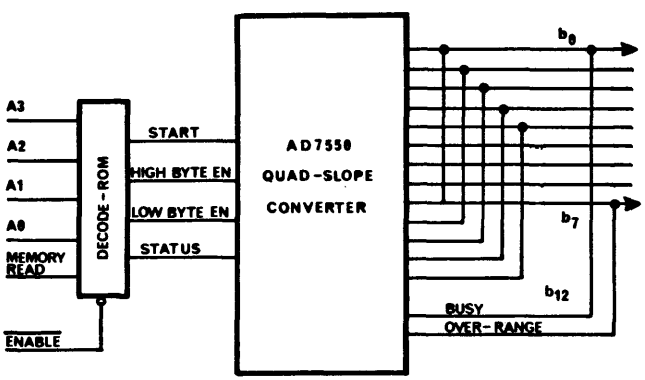

Figure 8-9.. Integrating converter connected for program controlled I/O

Most microcomputers which have analog interface elements also include a real-time clock so as to control the rate at which the analog environment is addressed by the microcomputer. It is a relatively simple matter to include the reading and restart of the A/D converter as part of the real-time clock handling routine provided that the A/D conversion time is slightly less than the real-time clock frequency.

MAKING A/D CONVERTERS APPEAR AS MEMORY

It was stated earlier that the ideal A/D should immediately respond to a read signal. This is not feasible for many reasons, but nevertheless there are many applications which require the A/D to have the most current value always available to the microprocessor. In this case the software is not constrained by the A/D conversion process and the A/D converter appears to the microprocessor as memory. For most A/D converters it is possible to achieve this requirement by inserting a buffer register between the converter and the microcomputer buses. The timing should be arranged so that the register is loaded by the A/D during a period when the microprocessor will not attempt a read operation – for example, whilst the program counter is being incremented. Figure 8-10 shows the AD7550 connected for this type of application.

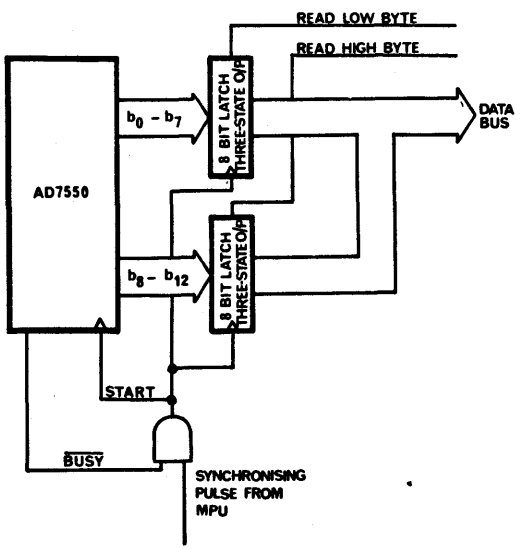

Figure 8-10. Continuously running A/D converter connected to appear to the MPU as memory

The buffer memory approach to A/D interfaces can be used to advantage to construct multi-channel A/D converters as shown in Figure 8-11. The A/D converter consists of a counter whose output feeds a D/A converter. The D/A output is compared with

the input signals and, at the point where the input and the D/A value are equal, the counter contents are written into the memory location associated with that particular comparator. The memory is a multi-port device which can be written into one address and read from another at the same time (e.g., SN74170). Clock pulses are continuously fed to the counter so that it scans through the whole range of analog values and repeatedly updates the memory contents. To obtain the analog value of an input channel the microcomputer simply addresses the memory. To avoid reading whilst writing, memory updates should take place at points when the microcomputer will not access memory. Note that this is a form of memory shared DMA as discussed at the end of Chapter 3.

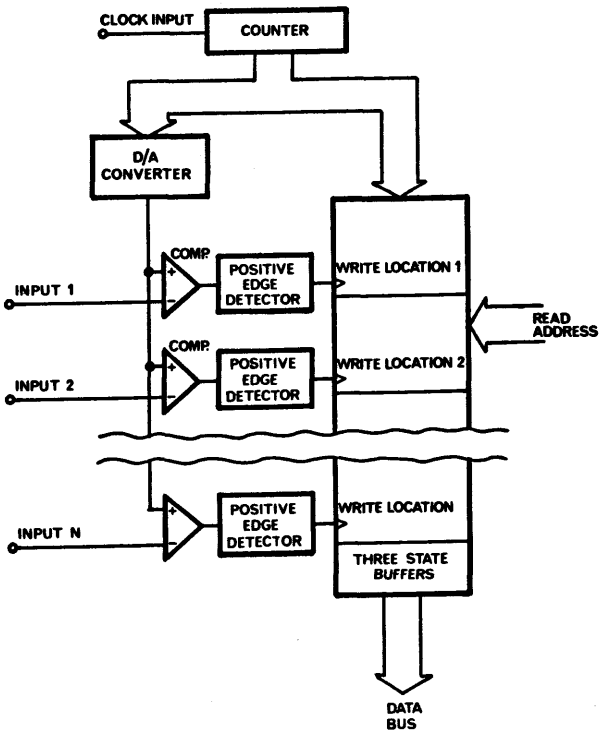

Figure 8-11. Memory shared DMA A/D converter

ANALOG CIRCUITS IN A DIGITAL ENVIRONMENT

The function of this small section is to offer some guidelines for connecting analog peripherals into a microcomputer system so as to reduce noise problems and aid system checks. There are three major noise sources in a microcomputer, namely high frequency clocks, noise on the power supply connections and noise generated by the power supply. Clock noises are best excluded by enclosing the analog circuits within a shield; hybrid and encapsulated converters often have this feature as standard. Where switching regulators are used to obtain the power supplies, the regulator itself should be enclosed in a shield and analog circuits should be as far away from the regulator as possible. Twisted pair wiring of analog signal lines helps to reduce magnetic coupling. Ground loops should be avoided at all costs and a strict ground current management procedure should be followed. Analog signals should not have long signal paths, nor should they run close to digital lines. Where possible, analog lines should be surrounded by ground with a ground plane on the opposite side of the board: this minimises stray capacitance and reduces the risk of coupling from other signals. If the supplies to the analog circuits are derived from the +5V logic supplies, a dc-to-dc converter which gives excellent isolation between the two grounds and rejects high frequency noise that may be present on the +5V supply line should be chosen.

Microcomputer systems are notoriously difficult to check and all designs should include some features which help the service engineer to trouble-shoot the system. One simple arrangement for checking the A/D function is to have an input signal multiplexer which has inputs connected to ground and $V_{REF/2}$. A small resident ROM uses the microcomputer to switch in the various inputs, exercise the A/D and present the results so that the function and calibration can be observed. Of course where a D/A converter is also present in the system, it can be used to provide the A/D input and thereby generate an overall system accuracy check.

FUTURE TRENDS IN CONVERTERS

At the beginning of this chapter three design criteria were given

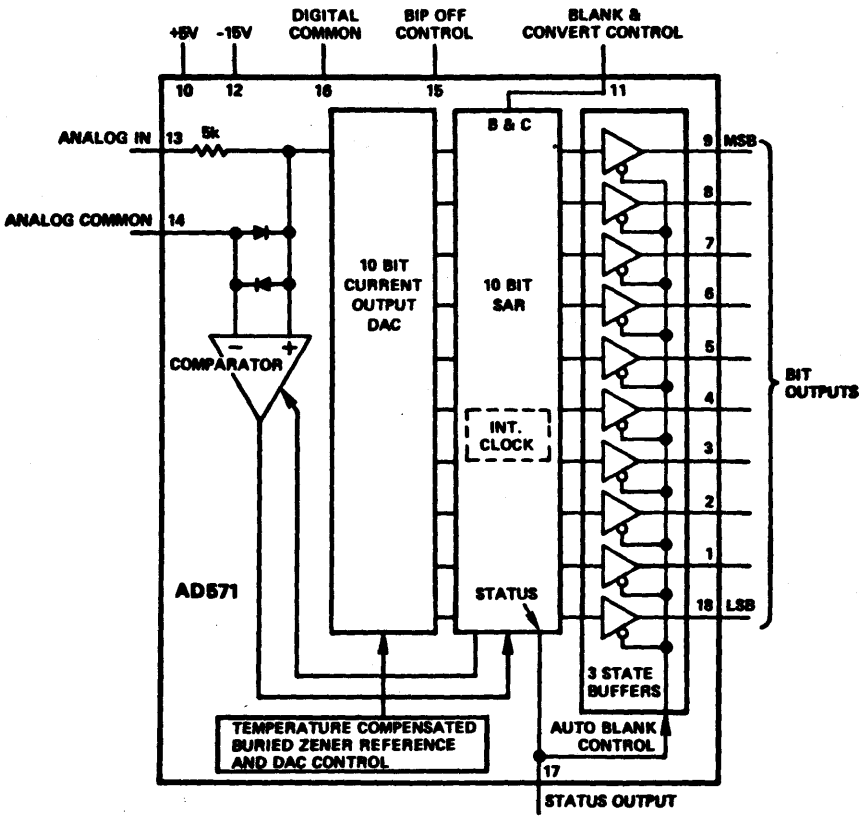

Figure 8-12. Minimum package size A/D converter

and, of those three, the most difficult one for the converter designer to meet is minimum package size. The number of pins available to an integrated circuit designer has always been a limitation, but with analog-to-digital circuits the problem is compounded by the necessity for separate analog and digital power and ground lines, and the large number of signal paths associated with analog circuits. As a result, there is a trend towards including more and more analog functions directly on the integrated-circuit. Figure 8-12 shows the AD571 successive-approximations converter which is built using I^2L technology. It has been possible to accommodate the complete A/D function into an 18-pin dual in-line package by including all functions such as voltage reference, clock and

comparator on the chip and by using minimum digital control pins. The device is ideally suited for use in the slow memory mode discussed earlier. For this application the blank and convert control input is connected to the read signal from the computer and the status output is used to put the processor in the wait mode until conversion is complete.

Another method of reducing pin count is to multiplex digital information onto a common bus and to use carefully encoded control signals. For example, by using careful design techniques it might be possible to build a 10-bit serial input D/A converter into a 10-pin package. Features such as improved on-chip address decoding are also desirable in future devices so as to reduce the number of address decoders in the overall system, but often this requirement tends to increase rather than decrease the pin count.

SUMMARY

It is difficult to completely cover analog interfaces to microcomputers in a single chapter. The above text has concentrated on system aspects rather than circuit design, and for full coverage of the latter the reader is referred to the companion text "Analog – Digital Conversion Notes." Also, the text has only considered the interface of single converters, and topics such as analog multiplexers and data acquisition systems have been omitted. These topics are an extension of the discussion to date and the reader should not have difficulty in extending the concepts given above to these cases. The importance of careful circuit layout cannot be overemphasised. A 2.5V change in 10 nanoseconds will couple a 25 millivolt spike to a 100 ohm impedance through only 1 picofarad. A 12-bit converter with 5 volts full-scale has a resolution of 1.2 mV, so that a spike as generated above would cause an error of 21 bits.