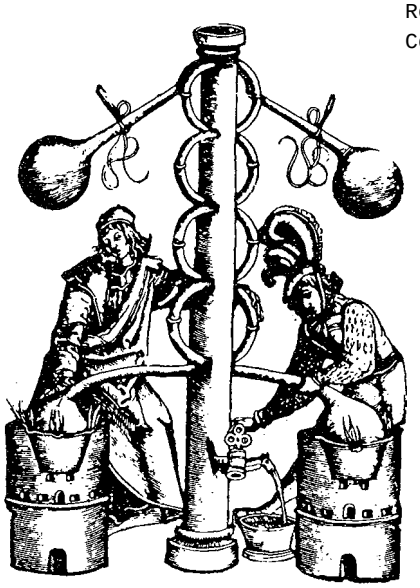
Reprinted with permission from American Laboratory, Vol. 11, No. 6, pp. 104-108.



Microcomputer interfacing

Direct memory access

By David G. Larsen, Peter R. Rony, Jonathan A. Titus, and Christopher A. Titus

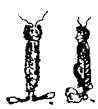
DIRECT MEMORY ACCESS is a technique that allows a computer user direct access to individual memory locations without first having to go through the computer's central processing unit. The direct memory access, or DMA, interfacing technique has a number of applications but few microcomputer users fully understand its use, and even fewer have fully implemented it for scientific data acquisition or control purposes.

Calculation shows that a very simple software routine, one that will input an eight-bit data point and transfer it to a memory location, will take many microseconds. Further, if additional points are to be taken and stored in some sort of a sequential file, additional software steps are needed for housekeeping tasks that count the number of data points, increment the address pointer, etc. When these additional software steps are introduced into the test program, an eight-bit data point may be acquired only every 30-40 usee. In many cases, however, higher data acquisition or data transfer rates are required. The idea of providing the computer user with a direct access to the memory section of the computer is not new. A direct memory access scheme was provided for in the Digital Equipment Corporation's PDP-8 series of minicomputers.

Most of the current microcomputers have a built-in DMA capability that few users ever consider. This capability allows small microcomputers to be used in high-speed data acquisition systems where the data transfer rates are higher than those that can be achieved under program control. This is possible because most microprocessor chips, such as the 8080, Z-80, and 8085, have data bus and address bus outputs that are three-state. Three-state devices can pass normal logic levels of one or zero, and they also can be forced into a third state, in which their outputs appear to be very high impedance loads on the bus. Thus, they no longer sink or source current and they are, in effect, disconnected from the bus. This third state allows external devices to obtain the use of the address and data bus lines when the computer's operations are suspended and its outputs effectively removed from the bus lines. It is easy to control the microcomputer's bus signals to force them into this third state. When the HOLD input on the 8080 is asserted at a logic one, it forces the 8080 to complete its current operation and then place its data bus and address bus outputs in their third state. A hand-shaking, hold-acknowledge (HLDA) signal is output by the 8080 to indicate that it has placed these signals in the high-impedance state.

External devices that wish to make use of the address and data bus lines are synchronized so that there will be no attempt to use the bus lines until the hold-acknowledge signal is received from the 8080. In general, these same devices have generated the bus-use request signal or hold signal that indicates to the 8080 that they wish to use the buses. In many microcomputer systems, additional buffers are used between external devices and the address bus and data bus connections to the microprocessor chip. If this is the case, it is important to ensure that these buffers sense the hold-acknowledge signal and also place their respective outputs in the third state. In the hold state, the central processor and its associated buffer circuitry all must be disconnected from the address and data buses.

External devices that are con-



Mr. Larsen, Department oj Chemistry, and Dr. Rony, Department of Chemical Engineering, are with the Virginia Polytechnic Institute and State University, Dr. J. Titus is Presidenl, and Dr. C. Titus is Consultant, Tychon, Inc.

figured for DMA transfers generally have their own registers that provide address information, usually 16 bits. These address registers also have three-state outputs that are normally disabled. They are only enabled, or turned on, when the 8080 is in the HOLD state. Then they can utilize the address bus to address the desired memory location to transfer data to or from. The data path between the external device and the memory chips also is placed in the third state. Therefore it can be used for the transfer of data, in this case the eight-bit value that is to be transferred between the external device and the location that is addressed by the external three-state address registers.

In most microcomputer systems, the reading-from and writing-to operations are controlled by two signals, memory read (MEMR or MR) and memory write (MEMW or MW). These signals mayor may not be available with three-state outputs. **In** some systems, where a control logic section has been constructed with discrete integrated circuits, these two signals probably will not have three-state outputs. When a system controller chip has been used, the outputs are probably three-state. **In** either case, it is easy to gate an 8080-generated read or write pulse with the corresponding one generated by the microprocessor chip. **In** this way, either the external hardware or the microprocessor can generate a read or a write pulse. Because the 8080 processor is not operating or executing program steps while it is in the HOLD mode, it cannot generate a read or a write signal. By using the HLDA signal to enable the read and the write pulses generated by the external device, the proper pulses will be generated by the 8080 in the normal mode and by the external device in the HOLD mode.

While the DMA interface described so far does not seem to be too useful in transferring only one eight-bit data word to a single location, the interface does illustrate the simplicity with which a DMA interface may be constructed. **In** most situations, however, a block of data values will be transferred between the memory and an external device. Note that the transfer may be in either direction, i.e., from a high-speed data acquisition device to the computer's memory, or from the computer's memory to a high-speed data storage device, such as a floppy disk.

In one application with which the authors are familiar, SN-74193 programmable up-down counters were used to provide the 16-bit address information during the DMA transfer, and a first-in first-out (FIFO) buffer memory was used to provide a block of data to the microcomputer's memory. Some simple control logic generated a write pulse (MEMW) and then incremented the address (count) present at the SN74193 counters. A series of switches allowed the user to preset the starting address at which the block of data was to be stored. Once a complete block of data was accumulated in the 256-byte-long FIFO memory, external circuitry was used to detect the FIFO-FULL condition and to initiate the request for use of the bus, asserting the HOLD line to the 8080. When the 8080 acknowledged the hold condition with the HLDA signal, the data were transferred from the FIFO memory to the microcomputer's main read/write memory at a rate of 500 kbytes/sec, or one byte every 2 μsec. This data transfer rate could have been increased at least four-fold (it was unnecessary to do so in this particular situation). A block diagram of the

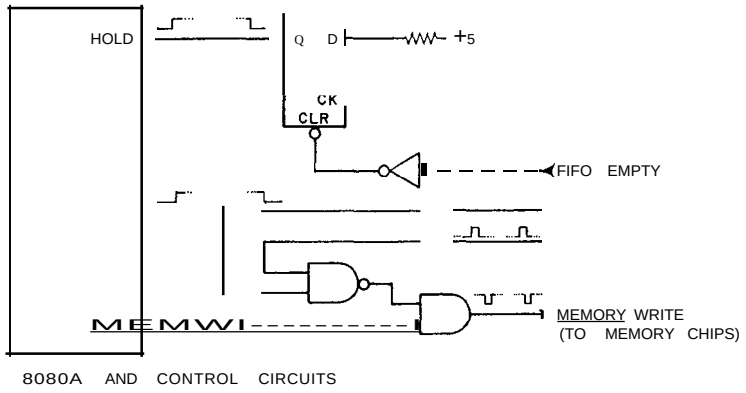


Figure 1 *Diagram of the interface used to control data transfers.*

interface that was used to control the DMA data transfers is shown in *Figure* 1.

Another use of the DMA interfacing technique is found in front panel control circuits. Control panels are often provided on microcomputers so that memory locations may be addressed and their contents examined by the operator. Through the use of a front panel the operator may elect to deposit new data in one or more memory locations. A simple front panel control system is constructed easily by the DMA technique because a 16-bit three-state buffer may be used to supply the address and eight indicators may be used to show the state of each bit in the memory location addressed. Eight logic switches are used to lead new data into a location and two control switches, EXAMINE and DEPOSIT, are configured to generate the memory-read and memory-write control pulses, respectively. Additional embellishments may be added to such a circuit to increment or decrement automatically the 16-bit memory address, etc.

There are, however, some notes of caution when the use of DMA is contemplated. The 8080-type processors are locked out of performing any program steps while they are in the HOLD state. This means that they cannot respond to interrupts while they are being controlled by an external DMA-based device. The DMA-based devices themselves stop the processor's operation, so if they are used, it is often difficult to predict when they will demand the use of the buses, thus halting complex and perhaps time-dependent operations. Some processors, such as the MC6800-series devices, require that the DMA-based devices relinquish control of the bus on a regular basis. The three-state-control (TSC) line in 6800-based systems can only be asserted for periods of 5 μsec at a time. Likewise, in systems that use dynamic read-write memory devices, the address and data buses cannot be in continuous use by DMA-based devices for more than approximately a millisecond because the external control logic must be able to use the address and data buses to refresh the dynamic memory cells on a regular basis.

The direct memory access interfacing technique is often overlooked by microcomputer users because it is not understood and when it is understood, it is often considered difficult to implement. We have concluded from our experience that DMA devices provide a readily accessible alternative to other interfacing schemes when high-speed data transfers are required. New DMA controller chips such as the Intel Corporation's 8257 provide much of the needed control circuitry in a small integrated circuit package. While such devices often simplify the required circuitry, they need additional control software to perform initialization steps, etc. The authors suggest that anyone who wishes to design a DMA controller start with a discrete integrated circuit approach to the interfacing task.