Microprocessor Application: A Less Sophisticated Approach

James T. Arnold

There is a widespread appreciation for the impact of data processing by digital computers, which profoundly touches so many aspects of life today. The rapid development and application of the digital computer has indeed been one of the most dramatic achievements of technology. Digital methods of management and manipulation of information at very high speed with digital computer operating systems give access to solutions of problems so vast and so intricate as to have been simply inconceivable before their advent.

The unique capability of the digital computer operating system to solve complex problems in a timely way justifies the necessarily enormous expenditures for its development and its services. These high costs have largely restricted the application of computer methods to tasks which could justify the fees by efficient use of the full capability of the computer operating system.

The microprocessor offers the means to relieve this cost constraint in the use of digital methods for problems of smaller scope. It is a revolutionary tool by virtue of the fact that it incorporates the potential performance of at least the lower end of the minicomputer line and yet, as a component, it has a cost so low (*1*) that its use can be justified in applications requiring only a small fraction of its capability. Moreover, while the application of complete computer systems, not to mention their design and construction, has been largely and properly left in the hands of a professional cadre of specialists, the microprocessor, intelligently used, can be an effective tool in the hands of the reasonably talented amateur.

This suggestion is not to denigrate the need for professional attentions in major computer problems. (The microprocessor has grown out of computer technology, and it reflects the complexity and sophistication of that field.) It is rather to state that the microprocessor offers, incrementally and at low cost, digital management and manipulative capability which can be understood and implemented, for simple problems, by nonspecialists. Applications in which use of the full range of even the minicomputer is contemplated are probably still served best by the engagement of professional technologists. The rare amateur who succeeds in such an application will, himself, have become an expert, at the expense of a considerable effort.

In this article I view uses of microprocessors as elements of instrument systems or to facilitate individual experiments. Of course, the instrument or experiment contemplated must succeed on its own merits; designs including microprocessors do not automatically guarantee superior performance. Indeed, because of the programmed nature of microprocessor operation, the experiment or instrument designer must exercise considerable care in planning and specification to avoid constraints that would be inconvenient in later operation.

Another Description of the Microprocessor and Its Operation

All descriptions of the microprocessor inevitably tend to illuminate its role as a potential element of a computer system. The computer role is deliberately deemphasized here in favor of such descriptors as controller and data manager.

The microprocessor is a digital electronic device of considerable complexity in which electrical current pathways are suceptible to controlled switching. It is designed to operate cyclically in such a way that during a cycle, data are (electrically) abstracted from one or more sources, manipulated by an arithmetic logic unit, and then distributed to one or more destinations, these steps of the cycle being under control of a prescribed control regimen called an instruction (Fig. 1). The data are represented in this process as ordered sequences of binary statements. The binary nature of each statement, commonly called a bit, is conventionally conceptualized by word pairs such as false, true; off, on; or zero, one; and the ordered sequences are connoted as words or as (binary) numbers. The data themselves can be as diverse as definition allows.

Implicit in the utility of the data is some means for their temporary or permanent storage. Storage can be accommodated in registers, either external or internal to the microprocessor. A single register is a component possessing an ordered number of cells, each of which is capable of reflecting a binary statement in such a way that the statement can be entered or abstracted at will. The order of the cells in the register and the identity, or address, of the register itself preserve the identity of the data.

The microprocessor system, either internally or externally, is provided with several special purpose registers in addition to general purpose registers for data storage. The control regimen or instruction for each cycle of operation is impressed on a first special register, called the instruction register, in the form of an ordered sequence of binary statements—that is, an instruction word. A second special register is called the program register. This register contains a binary number which identifies the location from which an instruction, or occasionally other information, is to be drawn.

In addition to executing a cycle prescribed by the instruction word, the microprocessor system facilitates the execution of a sequence of instructions by adjusting the content of its program register, thereby calling for another instruction. Figure 2 illustrates this operation, showing three successive cycles. The set of instructions, called the program, shown in the left-hand column of Fig. 2, actually governs the events taking place in the microprocessor. These events, shown in the center column, include the step of adjusting the program register which points to the address of the next instruction. The complete collection of instructions for operating the microprocessor is called the program. The microprocessor cycles shown are numbered in the sequence P, $P + 1$, $P + 2$. It is quite natural to assign addresses in the program as sequential numbers as well, and to interpret the content of the program register as a multiple-digit binary number. The most usual adjustment of the program register then, is to in-

The author is a senior scientist in the Corporate Research Laboratories, Varian Associates, Palo Alto, California 94303.

crement this number by one. (It should be noted that an instruction may direct the microprocessor to treat the next program address as the location of a data source rather than an instruction, and another may direct the entry of a nonserial address pointer in the program register.)

In summary, the microprocessor is an example of a stored-program device; it can proceed autonomously through a sequence of operations, selecting, manipulating, and distributing data in accordance with a previously stored plan or program. It can perform its function rapidly; cycle times are typically from 1 to 20 μsec.

Evolution of a Simple System Using a Microprocessor Well Below Its Capacity

The following paragraphs describe the incorporation of a variety of functions into an imaginary instrument by the use of a microprocessor. Actual development of a system provided with such a set of functions should be undertaken as a whole rather than by the steps that are serialized here for illustrative purposes.

Keyboard and display. The imaginary instrument will involve interaction with an operator; it will require inputs from the operator, and it will display certain responses to these inputs and its own operation. The growing popularity of keyboard entry and alphanumeric display suggests selection of this method to interface the instrument to the operator. The first function of the microprocessor, under control of a stored program, is to accept keyboard entries and to output these, and possibly other information, to a display.

The operation of a microprocessor lends itself naturally to this function. The operator's keyboard can be treated as a data source and the display as a destination. Successive access to source and destination requires control by a repeated sequence of program instruction steps. Particular requirements of the display device may necessitate the use of manipulative as well as distributive instructions.

Implementation of the keyboard-to-display function is shown diagrammatically in Fig. 3. A word of explanation is required for the blocks in Fig. 3 marked "memory." Memories are generally external to the microprocessors and consist of large numbers of registers or other word storage devices, each having a unique address susceptible to call from the microprocessor's program register output. The read only memory, ROM,

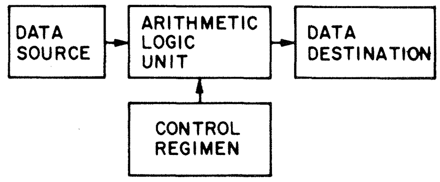

Fig. 1. Schematic representation of a data flow cycle in a microprocessor.

acts exclusively as a source of instructions and data, being loaded with the proper sequence of words. The random access memory, RAM, can act both as a data destination and as a source to recall data previously deposited.

Operation begins with an instruction in the ROM which directs the processor to

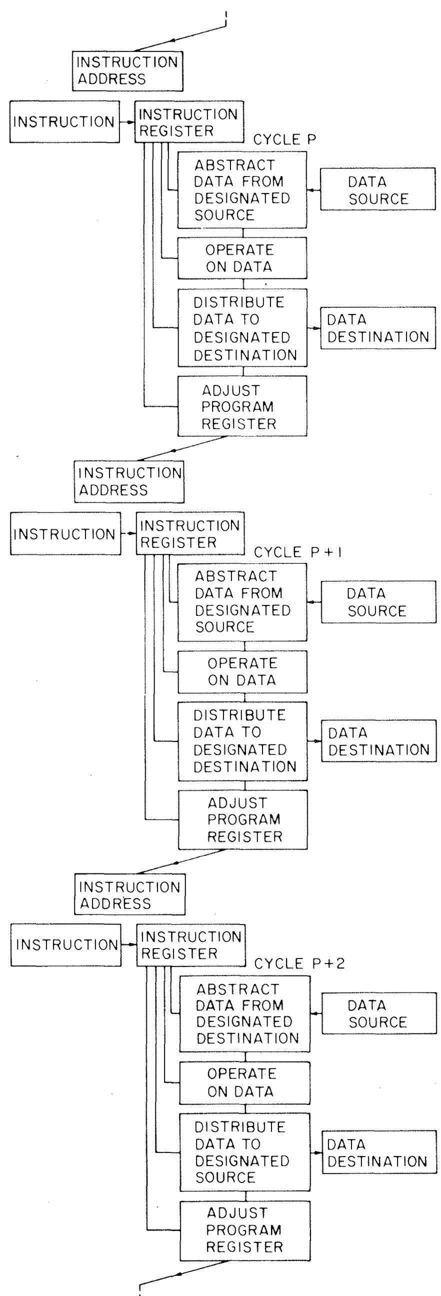

Fig. 2. Representative progression of microprocessor cycles.

"read" the first entry from the keyboard. When the entry is received, the microprocessor must perform the equivalent of the following steps.

1) Retain the entry in temporary storage, for example, the RAM.

2) Assign a location for the entry on the display.

3) Retain this assignment paired with the entry itself in temporary storage.

4) Manipulate the entry to conform with a code appropriate to the display.

5) Deposit the coded entry in the assigned display location.

Many displays are volatile, and in this case the last operation must be repeated at a rate compatible with visual persistence (about 30 sec^{-1}). As the successive keyboard entries are directed to the display, each must be updated in its turn at this rate, and to accommodate this requirement the microprocessor must rapidly generate an elaborate sequence of data to deliver to the display.

Fortunately the system can be organized to do just this. One simple method is to have the microprocessor arrange a data matrix in a block of the RAM such that its addresses are in a one-to-one correspondence with locations on the display. This display "mirror" is built up relatively slowly by entries from the operator's keyboard, as required, and is rapidly scanned serially by the microprocessor to provide the continuous, complex (but almost mindlessly repetitive) data stream required by the display.

A cathode-ray tube with a conventional television raster is a popular display device. The data stream, encoded by the microprocessor according to a stored program in the ROM, can be made to accommodate the image dissection and timing requirements of the raster. Factory encoded character generating ROM's are even available, at low cost, for this purpose, relieving the instrument designer of the chore of generating the codes.

While the function described above does not significantly tax the operating sophistication of the microprocessor, and the rate of data entry from the keyboard is extremely slow by microprocessor standards, the continuous update of the display screen will burden the microprocessor to the point that no other functions can be accommodated. This difficulty can be relieved by augmenting the system of Fig. 3 with special purpose hardware which, standing alone, can convert the contents of the RAM display data matrix block into the data stream required by the display. This augmentation is shown schematically in Fig. 4. Access to the shared display RAM is invoked occasionally for data entry by the microprocessor and continuously by the dissected image generator for display updating. To prevent ambiguity, access to the shared RAM address is normally continuous from the dissected image generator, but is preempted by the microprocessor whenever the latter is required to enter data. This arrangement for priority of access permits completely asynchronous operation of microprocessor and display. With the hardware augmentation in the system of Fig. 4, the microprocessor is grossly underemployed, and other functions can be accommodated.

Instrument control and data acquisition. Since the function of writing to a display, by itself, is not an adequate operator-instrument interface, the excess capacity of the microprocessor can be turned to good use in performing the additional tasks of instrument control and data acquisition. Instrument operating programs and operating parameters can be stored in the ROM, and instrument output, converted to digital words, can be read and stored in an orderly way in the RAM. Use of the keyboard and display can be merged with these tasks. With the keyboard, the operator can access the microprocessor to select operating programs and parameters for a desired instrument cycle. Under control of the selected operating program, the microprocessor can also be directed to accept progressive and final results from the instrument, to format these results, and to enter selected results on the display.

An implementation of this more complicated operation is shown schematically in Fig. 5. Data paths to and from the instrument and an additional RAM have been added, and the ROM must be considerably expanded to accommodate the new tasks required of the system. The utility of the system is dependent on the care and imagination exercised in the hardware and operating program design, and the burdens and benefits of microprocessor use are most sharply focused in the design process.

In terms of their hardware needs, instruments require input intermittently or continuously, and they generate outputs which must be accepted to formulate results. The inputs and outputs may cover a variety of physical quantities, not necessarily electrical in expression, and generally analog in nature. Since the microprocessor is digital and electrical, transducers are generally required to convert data to appropriate form in either direction. The transducers adjacent to the instrument are likely to be analog in character and may relate electrical levels to other physical quantities. The ones adjacent to the microprocessor accommodate analog-to-digital and digital-to-analog conversion (ADC and DAC).

Devices to accomplish a wide variety of conversions are readily available on the commercial market, specified with reasonable intelligibility. In digital parlance, the precision of a device is usually specified in bits, each bit representing a power of 2. Thus 8-bit precision represents a resolution of 1 in 2^8 or 1 in 256. Other specifications include speed of conversion, absolute accuracy, range, and so forth.

Data transfers are negotiated one word at a time. The width of the word (in bits) is determined by the data paths; the data paths into and out of most microprocessors are 8-bits wide, although other widths may exist in other parts of the system. (This constraint does not limit the precision of the data, since more than one word may be used to transfer a particular element of information if greater precision is required.)

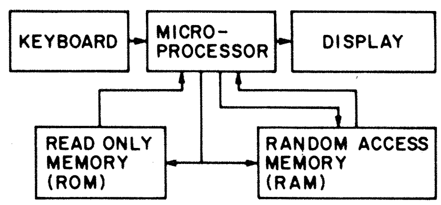

Fig. 3. Schematic implementation of the direct keyboard-to-display function with a microprocessor.

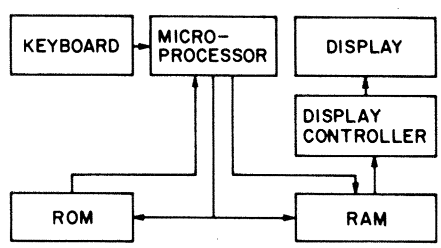

Fig. 4. Addition of an auxiliary display controller to relieve the microprocessor of repetitive display refreshment.

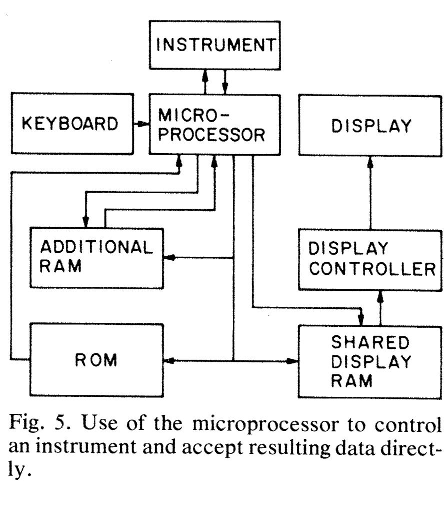

Fig. 5. Use of the microprocessor to control an instrument and accept resulting data directly.

Each of the functions to be controlled and each of the quantities to be monitored by the microprocessor must be assigned an individual address. Most microprocessors boast a number of input and output "ports" which represent the means to communicate with external devices. In addition, the addressing structure, usually designated for memory exchanges, can also be used to access a number of external registers for data communication to and from an instrument.

Equal in importance to the hardware in the use of a microprocessor in an instrument system is the organization of the operating programs. The designer must devise through these programs the means to utilize as much of the capability of the instrument as possible, to improve the efficiency of its use, and to reduce the errors and ambiguity of its results. To do this he must have a thorough understanding of the operating characteristics of the instrument and the methodology of the measurements made. Once a set of operating procedures has been established for a desired operating program, the procedures can be reduced to a chain of operating steps, each of which can be negotiated by the microprocessor under control of one or more instruction words.

The chain of operating steps can be as elaborate as the instrument requirements dictate. It can branch to alternative lines on the basis of incremental results. It can be keyed to external parameters such as real time. It can be interrupted temporarily, on demand, for the service of off-line requirements. It can sense anomalous events and return control to the operator, signaling a diagnosis of the anomaly it has acquired. The orchestration of this operation is by means of the flow of instruction words called from the stored operating program by the microprocessor. Although the reduction of the operating procedures to a set of instructions is necessarily complicated, it is negotiable, by hand, for simple problems, and writing the program can be facilitated by design and organizational aids such as flow diagrams which map the sequence of program steps.

The designer may choose, in simple cases, to construct the program in the instruction code directly understood by the microprocessor. In routines of greater extent, coding aids are available which can be accessed by an accumulation of program statements in a simple formalism called assembly language. The reduction of these program statements to microprocessor code requires the use of a program assembler routine residing in a modest computer operating system. Programming aids of even greater abstraction are available; however, their use entails access to specialists and large computer operating systems.

Most instrument operating routines are relatively simple and can be approached even at the level of microprocessor instruction codes. An example may be found in the frequently encountered case of observing a dependent variable quantity as a function of the sweep of a predetermined small interval of an independent variable, making observations at uniformly incrementing values of the independent variable. The procedure is as follows.

1) Begin, set initial and terminal values for independent variable in temporary registers.

2) Set memory (storage) pointer to the head of a one-column data matrix in the RAM.

3) Acquire the value for the dependent variable.

4) Deposit the value in the data matrix (RAM) address.

5) Increment data matrix address.

6) Increment independent variable.

7) Compare independent variable with its terminal value.

8) Exit if done; else

9) Loop back to step 3.

This small procedure is generally embedded in a larger routine which addresses, for example, the same measurement for a different interval of the independent variable. The larger routine itself may be embedded in a still larger program which (by keyboard selection) can access this or other routines. As long as the routines are simple, they may be individually coded with relative ease, and the collection of routines can be made accessible on selection by a supervisory portion of the program.

There is no rule other than good sense to proscribe hand coding as the problems grow larger. At some point—sometimes identified, in the end, by sheer frustration—the designer will discover that hand coding is too costly in time and effort. At that point a more formal approach, involving abstract methods of assembling microprocessor instructions with a computer operating system, will be clearly indicated.

Specialized instrument controllers. The simple instrument operating routine whose nine steps were sketched above can be viewed as an advantageous use of a microprocessor in many cases. However, where the routine must be continuously exercised to operate the instrument, it may begin to strain the throughput capability of the microprocessor. In the interest of underutilization, this difficulty can be relieved by the interposition of an instrument controller, which accepts broader and less frequent control commands from the microprocessor and which accumulates results in such a fashion as to allow delivery to the microprocessor with the minimum of transmission consistent with integrity of the data.

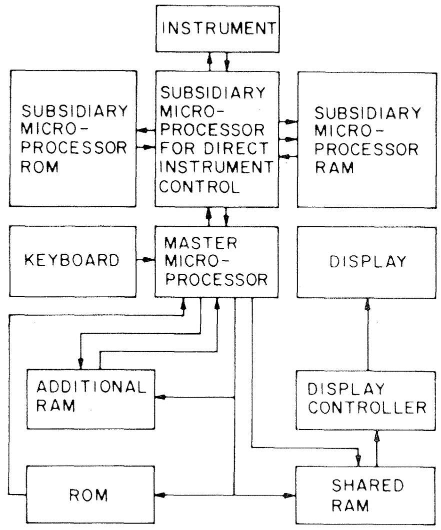

Fig. 6. Complete system for microprocessor instrument operation including the use of a subsidiary microprocessor, having its own ROM and RAM, for direct instrument control and data reception.

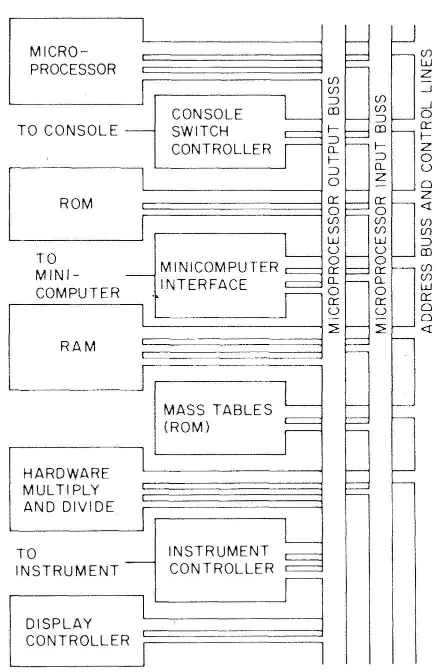

Fig. 7. System designed to operate a portable mass spectrometer with a microprocessor. The minicomputer interface allows the system to communicate with a minicomputer for checkout and program composition.

Such a system is shown in Fig. 6. In accordance with the example mentioned, the iterative part of the (independent variable) control routine can be managed by the controller with the resultant (dependent variable) data being accumulated temporarily within the controller itself. Transmission of the essentials of these data can then be called for by the microprocessor after each complete cycle of the routine. With such a controller, the microprocessor is relieved of continuous operation of the instrument and is required only to set the parameters defining the fiducial parts of the routine and to collect the relevant results.

The controller can frequently be so simple as to be amenable to design with hardwired digital logic, or even with analog circuits and switches. However, the reader will recognize that the controller, with no other obligations than to carry out a repetitive cycle of operation, may itself qualify for implementation with an underutilized microprocessor having its own ROM and RAM. This subsidiary microprocessor can then be operated as an instrument controller under the supervision of the primary microprocessor. Interconnection of the two through standard input/output data paths need not pose a difficult problem beyond the avoidance of possible timing conflicts which might lead to garbled results.

Data management and processing. With the system configured according to Fig. 6, where the primary microprocessor is underutilized in its function to respond to keyboard commands, to operate the instrument, to accept instrumental results, and to load the display, there will be capacity remaining to perform the function of more elaborate management and manipulation of data. The first task for the microprocessor in this area is perhaps simply one of arranging the data in a convenient format. Other tasks depend, of course, on the requirements of the instrument, and the operations described below are mentioned only by way of example.

The microprocessor can easily be programmed to sort data by a prescribed rule; for example, to segregate maximum and minimum values. It can also generate difference tables and locate turning points. The latter operations are arithmetic and invoke the computational capability of the microprocessor. Other simple arithmetic operations might include generation of moving averages or simple digital filtering of data.

Even more complex data processing is, of course, possible. However, elaborate data processing is difficult to program without the help of programming aids, and it is advisable to enlist the help of specialists and to review the suitability of the microprocessor (compared to a minicomputer operating system) where the application involves elaborate computations. The use of a microprocessor in complete imitation of a minicomputer operating system with major data handling peripherals is likely to be more costly and will certainly involve more labor than the use of its model. Notwithstanding this precaution, the microprocessor can be employed to remarkable advantage in systems which, superficially, appear to be very complex.

Judicious distribution of hardware assistance to the microprocessor can reduce its load, as has been shown by the examples above. In the computational area, two commonly required functions which, if relegated entirely to the microprocessor, can burden it excessively and consume inordinate amounts of time are multiplication and division. A hardware auxiliary for fast multiplication and division can be devised, either along classical calculator lines or with special purpose large-scale integration digital components. To use this auxiliary, the microprocessor simply distributes the multiplier (divisor) and multiplicand (dividend) to the auxiliary device, and then reads the product (quotient) resulting. The device can be made to operate at such a speed that there need be no hesitation in the entire process.

In the programming area, algorithms have been developed in the technology of computer application which can be borrowed for microprocessor use to speed up and simplify other operations. These are too numerous to permit a comprehensive list here; however they include such functions as extracting roots, double precision operations, floating decimal point management, arithmetic, trigonometric conversions, and exponentiation. In addition, where special functions or calibration corrections are required, they may be accessed by simple interpolation of tabular entries introduced in blocks of the ROM or RAM.

Concluding Remarks

Since the microprocessor has been designed, by and large, to provide the principal part of the central processing unit of a computer, there has been a tendency on the part of vendors and many users to view it exclusively in that function. Good computers have been and will be constructed with microprocessors. In the paragraphs above I have tried to convey an alternative view, which perceives the microprocessor as a component having many features that can be used individually, and to a considerably lower level than its capacity would allow, without requiring the expertise of the computer architect. The very low cost of today's microprocessors and their ability to perform a variety of operations on digital data in accordance with a preconceived program can justify underutilization of their fullest capability.

No specific mention has been made of circuit details or of detailed instruction formats or algorithms in programming. Detailed data sheets are available for most of the processors offered (*2*). These give sufficient descriptions of microprocessor operating cycles, instruction sets, and other features to allow the enterprising amateur to develop useful simple systems. Successful use of the devices will often involve their augmentation in the systems with specialized hardware auxiliaries. These auxiliaries serve to reduce the complexity of the tasks required of the microprocessor itself.

A final illustration, Fig. 7, shows the advantageous application of a microprocessor to a portable mass spectrometer (*3*) which was designed and constructed by an amateur oriented to the approach suggested in this article. The microprocessor used was a first-generation device with a relatively slow duty cycle, and the system could not have been operated without the several auxiliaries shown. The system accommodated several modes of operation based principally on the monitoring of a number of selected mass peaks and displaying a processed evolution of their amplitudes in the form of moving average values. Included in the program were an automatic self-calibration routine and one to allow sequential scans of the entire mass range of the instrument. The stored program occupied 1792 words of ROM, and 1024 words of RAM were provided, primarily to accommodate variable program and a data matrix for raw and processed data.

Most of the reference material for microprocessors as devices is in the form of product release descriptions and manufacturers' data sheets. A useful summary list of current offerings of manufacturers may be found in one of the electronic trade journals (*4*).

The decision to undertake the design and construction of a nontrivial system using a microprocessor is a commitment to a major effort and should be contemplated with a good deal of circumspection. However, if the system is reasonably confined in scope, the essay can be rewarding and fun.

References and Notes

1. Texas Instruments Inc., Houston, recently announced a price of $21.15 each in lots of 100 for the 8080 microprocessor. This 8-bit device is very competitive in specification and has excellent support in software.
2. For example, Intel Corp., Santa Clara, California: *Data Catalog 1975*; *MCS-8 Microcomputer Set Users Manual* (November 1972); *MCS-80 Microcomputer Systems Users Manual* (July 1975). Motorola Semiconductor Products Inc., Phoenix, Arizona: *M6800 Systems Reference and Data Sheets* (May 1975). National Semiconductor Corp., Santa Clara, California: *Pace Technical Description, 1975*. Rockwell International Corp., Anaheim, California: *Introduction and Description, Parallel Processing System (PPS8)* (Document No. 20164N40, 1974, 1975).
3. J. T. Arnold and P. P. Robbiano, *U.S. Army Tech. Rep. No. LWL-CR-03P68C* (June 1974).
4. R. H. Cushman, "Second annual microprocessor directory," *EDN* **20**, 42 (20 November 1975).