SESSION XII

MICROCOMPUTER-BASED LABORATORIES II

A. M. LESGOLD, *University of Pittsburgh, President*

A microcomputer-controlled laboratory: Hardware

RICHARD F. DILLON, BARRY MILLMAN, JO W. TOMBAUGH,
WILLIAM R. FERGUSON, and WILLIAM R. BEZANSON
Carleton University, Ottawa K1S 5B6, Canada

A microcomputer-controlled laboratory designed to incorporate the Motorola 6800 microprocessor unit is described. The microcomputer system is reliable, flexible, and expandable, as well as adaptable to the next generation of Motorola microprocessor components. A modular system design utilizes multiprocessing with separate microprocessor units dedicated to specific functions. This computer, when used with powerful, high-level software, provides a general-purpose psychology laboratory computer system that is easy to use.

In this and a following paper (Tombaugh, Dillon, Millman, & Bezanson, 1979), a large project to develop an easy-to-use and flexible microcomputer system adaptable to many laboratory settings is described. This project was designed and implemented by a team consisting of a systems engineer specializing in real-time applications, an electronic technologist experienced in design of microcomputer systems, and three psychologists who have previously developed laboratory computer system software including high-level languages (Dillon, Dumontier, Bezanson, & Brennan, 1978; Tombaugh, 1974; Millman, Note 1). This team was recruited at the request of physiological psychologists in our department who realized the advantages of a flexible microcomputer system if the programming problem could be minimized. The team developed the hardware configuration described in this paper and designed and programmed software to make the system easy to use (Tombaugh et al., 1979).

SELECTING A MICROCOMPUTER

The Decision to Build or to Buy

The sensational prices quoted for microprocessors,

This research was supported by NRC grants to W. E. Webster and Brian Tansley, and a grant from the Department of Psychology. Time on the central Carleton University computer was provided by a grant from the President's Computer Fund. G. Crichlow and D. Parlour contributed to a variety of phases of the project. N. Mikhail was instrumental in the design and fabrication of many of the prototype modules described in this paper. Complete plans for the hardware described in this paper and unpopulated printed circuit boards are available at nominal cost. Barry Millman is now at Data Kinetics, Ottawa, Canada. Requests for reprints should be addressed to Richard F. Dillon, Department of Psychology, Carleton University, Ottawa, Ontario K1S 5B6, Canada.

often less than \$100, are misleading since the microprocessor unit (MPU) must be accompanied by a number of components to produce a computer. Our first decision was whether to buy or to build a computer. Unfortunately, the systems commercially available in early 1978 failed to meet a number of important requirements.

The first problem was price. Microcomputers appropriate for laboratory use were available, but at a price that exceeded the projected budget. Affordable computers were oriented primarily at the hobbyist. The second problem was that we wanted a system that could easily be interfaced to a large number of peripheral devices and could be easily expanded. Because of an apparent emphasis on small size, most commercially available microcomputers in the desired price range were limited in this respect. Third, high-quality peripheral devices—reliable and capable of withstanding heavy use—were needed. The hobby computers had limited peripherals of an unknown quality. The prospect of buying a cassette-based system, or even a minifloppy disk system, from an unknown company was not appealing. Fourth, a structured high-level language was required for writing system software. BASIC, the dominant language in the hobby-computer field, is not an adequate systems language. Finally, it was not clear whether the commercial computers were designed to incorporate the next generation of microprocessors. With microprocessor technology advancing so rapidly, care must be taken to avoid equally rapid obsolescence. In view of these factors, the decision was made to use the expertise available on campus to design and fabricate a modular microcomputer system of industrial quality at a hobby-computer price.

Microprocessor Unit

Once the decision to design a computer system was

made, attention was focused on the choice of the most appropriate microprocessor unit. This decision was made on the basis of a number of criteria. First, cost (a total budget of less than \$8,000 for MPU, design costs, and peripherals) ruled out the most popular 16-bit MPUs. Second, although it was known that the three major manufacturers were planning to introduce second-generation MPUs, a decision was made to start development immediately using a MPU and support components that were readily available. Third, it was known in early 1978 that at least one manufacturer (Motorola) intended to introduce a second-generation MPU that would be hardware and software compatible with their existing MPU (Wiles, Musa, Ritter, Boney, & Gunter, 1978). We hoped to design a system around a MPU that could easily be replaced when the next generation became available and could use our software with minimum modification. Fourth, the plan was to start programming before the hardware was available. For this reason, a high-level cross-compiler, cross-assembler, and simulator were needed. Finally, a disk controller and interface and a convenient operating system were essential. It was not our intention to build the controller or to write the operating system. Consideration was given only to MPUs for which such hardware and software were available commercially.

Taking these criteria into consideration, the Motorola¹ M6800 MPU was selected. This MPU uses a straightforward hardware approach and has a convenient instruction set.

MICROCOMPUTER SYSTEM CONFIGURATION

A block diagram of the system is shown in Figure 1. At this stage, the system is fairly conventional. The modules and devices shown in Figure 1 provide maximum flexibility and reliability. In general, if high-quality modules were available at a reasonable price, they were purchased; if not, we designed our own.

Prototype versions of a number of our boards are illustrated in Figures 2-5. All modules have been tested and functioning for some time. Production versions will use high-quality printed circuit boards with plated-through holes and gold-plated contacts for maximum reliability. Individual boards were designed to provide maximum possible address flexibility, although our software attempts to use Southwest Technical Products² address conventions whenever possible.

Our design philosophy is to isolate functions on individual boards. Densely populated modules achieve small size at the expense of complexity. Avoiding multifunction boards facilitates hardware testing, maintenance, and troubleshooting.

Mother Board, Card Cage, and Chassis

The modules shown at the left of Figure 1 reside in a rack-mounted chassis with the front removable to

Figure 1. System block diagram.

provide easy access to all boards. A 19-slot mother board and card cage, shown in Figure 6, fit into the chassis. The card cage is 48.3 cm wide, 13.3 cm high, and 33.1 cm deep. The mother board provides access to the Motorola Exorciser bus, a convenient 86-pin connector bidirectional bus, at each slot. Since all addresses are provided to all slots, the maximum possible combination of random-access memory (RAM), erasable programmable read-only memory (EPROM), and special-function addresses is 65,000 bytes.

Power Supply

A Lambda³ triple-power supply provides 5 V at 12 A, and ± 12 V at 3 A. The power supply is rack mounted on a separate panel above the computer chassis.

Master 6800 MPU

The MPU with clock circuitry, data and address buffering, and direct memory access (DMA) circuitry is shown at the right in Figure 4. The 6800 A version MPU functions at a 1.5-MHz clock rate. Consistent with our desire to isolate functions on individual modules, there are no other functions on the MPU board, although there is a 128-byte RAM, convenient for testing the board, which is disabled for normal operation.

RAM

RAM in 8-kilobyte modules, was purchased from Creative Microsystems.⁴ This static memory with 250-nsec access time is switch selectable in 8-kilobyte blocks. Typically, three to six RAM modules are used, although modules can be added or removed as needed.

ROM

Switch-selectable EPROM modules, shown on the right in Figure 2, can hold up to 8 kilobytes of 2708-type EPROMs. One version of the EPROM board has been modified so that one socket contains a priority interrupt

Figure 2. Optically isolated parallel I/O board (left) and EPROM board (right).

controller, which provides eight vectored interrupts.

Among other programs, one EPROM board contains a monitor. At various times Motorola's Mikbug, Micro-ware's RT/68,⁵ and Southwest Technical Products' SWTbug and Diskbug monitors have been used. EPROMs are programmed using hardware and software that we developed. The very inexpensive Southwest Technical Products' EPROM programmer has also been satisfactory.

Parallel Input-Output

An optically isolated input-output (I/O) board, shown at the left in Figure 2, is populated with one peripheral interface adapter (PIA) to provide 20 bits of programmable parallel I/O. The optically isolated parallel board is used for control with input and output devices (lights, tones, shock sources, solenoids, response levers) that are outside the computer chassis. In electrically noisy environments, lines connecting such devices to the computer are doubly isolated at the computer and at the device. Twenty buttons and switches, interfaced through a 20-position keyboard encoder, use one of these boards. The encoder provides a unique

5-bit binary signal for each button, so all 20 buttons use the same PIA. Examination of the bit pattern indicates which button was pressed.

The block labeled "runtime control panel" in Figure 1 uses five of these parallel input buttons. The function of each of these special-purpose buttons in our run-time system is described in Table 1.

Another parallel I/O board, shown on the left in Figure 3, has three PIAs (60 bits) with no optical isolation. This board is used for logical control inside the computer chassis. Programming parallel input and output is very simple in the high-level language we provide.

Serial Ports

The left side of Figure 5 shows a serial board constructed from PIAs. This board, which will eventually be replaced with asynchronous communications interface adapter (ACIA) components, is used to interface the MPU with a variety of serial devices, often using the RS-232-C protocol. At various times, we have connected a DECwriter II and a number of cathode-ray tube display and keyboard units. One port, labeled "modem

Figure 3. Nonoptically isolated parallel I/O board (left) and dual-function timer board (right).

Table 1
Console Switches and Their Functions in a
Signal-Averaging Experiment

Switch Label	Function
START	This switch is used to start an experiment.
RESTART	This switch results in all parameters being reset to the values that they had when START was first pressed. Useful to check electrodes, etc., before starting actual run.
EXCLUDE DATA	Data currently displayed on display system will not be recorded on disk. Used to reject "noisy" data.
NEXT AVERAGE	A researcher programs some number of trials to be included in an average. This button overrides the predetermined number.
STOP	This button terminates the experimental session.

to Sigma 9" in Figure 1, is connected to an acoustic coupler that provides access to an extremely useful peripheral device, the central campus computer. We attempt to dedicate the laboratory computer to running experiments and perform as many functions as possible on the central computer.

Interval Timers

Figure 3 shows a multiple-function timer (on the right), which provides sophisticated timing capabilities. An interval timer function is programmed by storing 2 consecutive bytes containing the desired interval at the timer address. The timer sends an interrupt to the MPU when the interval is complete. The second function does not interrupt the MPU. The time from the start of an experiment can be determined by loading an accumulator with the contents of the addresses used by this function. The interval timer is used to control predetermined intervals; the interrogation timer is used

Figure 4. Early prototype dual timer board (left) and MPU board (right).

to, for example, measure response latencies. This is accomplished by noting the time when a response is first permitted, noting the time when the response occurs, and calculating the difference. These two functions replace, and give more convenient operation than, a real-time clock.

Floppy Disk System

In addition to the mass storage provided on the central computer, a floppy disk system is available. A DMAF-1 disk system was purchased from Southwest Technical Products in kit form. This reasonably priced system uses two Calcomp 143M double-sided 8-in. disk drives to provide a total of over 1.2 million bytes of on-line storage. Data are transferred between the disk and memory using DMA without MPU intervention. The disk controller uses the "halt-burst" mode of DMA, which halts the MPU, transfers a 256-byte sector at the

fastest rate possible, and then starts the MPU again. The interval timers continue to operate during the transfer, and any interrupt request is serviced as soon as the data are transferred. However, a user should request data recording at times when concurrent processing is minimal, such as during an intertrial interval or during a 3-sec reinforcement period.

The disk controller uses the SS-50 bus rather than the Motorola Exorciser bus. Manufacturers of M6800 computer systems seem to be using the SS-50 bus as a standard. A special Exorciser board with standard SS-50 male connectors makes it possible to plug SS-50 modules into the Exorciser bus. A similar board (see the right module in Figure 6) makes it possible to use Southwest Technical Products I/O boards, which use 20-pin connectors, in the Exorciser bus. These conversion boards, together with the Exorciser bus, provide access to almost all of the major M6800 modules available commercially.

Figure 5. Parallel-to-serial board with baud-rate generator (left) and bus conversion board (right).

Arithmetic Processing Unit

It is not practical to do real-time signal-averaging research using a microcomputer if arithmetic must be performed with software. To provide fast arithmetic processing, a module using the Advance Micro Devices⁶ AM9511 arithmetic chip was developed. While this board was being developed, we used the Southwest Technical Products calculator board. The calculator board is very slow, but provides convenient access to numerical functions at a reasonable price. The first version of our AM9511 board uses straightforward interfacing techniques without interrupts or DMA transfers. The need for more elaborate interfacing is being evaluated at this time.

Cassette Controller

Much of the software available commercially comes on cassettes using the Kansas City recording standard. A PerCom⁷ CIS-30 cassette controller has been used to transfer data from cassette onto a floppy disk.

Display Subsystem

The device labeled "scope display" in Figure 1 interfaces to the microcomputer through a serial port. This subsystem, still in the design stage, will be used to display analog data, either in raw form or after being processed. At this stage it appears that the subsystem will be a complete microcomputer with its own power supply, MPU, RAM, EPROM, and a graphics processing unit connected to an inexpensive video monitor. The plan is to modularize the display function by allowing the master MPU to transfer data to the display subsystem through a standard RS-232-C interface at a high baud rate. The display unit, after receiving data from the master MPU, will be responsible for refreshing the monitor and moving whole curves up, down, left, and right at the command of the user. The master program can transfer data at noncritical times and then devote its attention to more important time-critical events. Not only does this approach modularize the display hardware, but it permits modularizing the

Figure 6. Card cage and 19-slot mother board.

software as well. Modifications can be made to the display subsystem hardware and software without influencing the master system in any way. In fact, use of a standard interface means that this inexpensive unit can be used as a peripheral device with any computer.

A/D and D/A Conversion

The blocks labeled "A/D" and "D/A" on the right of Figure 1 are currently used in the master MPU system. A Datel⁸ Sinetrac ST-6800 board provides 16 channels of multiplexed 12-bit A/D conversion with 20-microsec conversion time, and two 12-bit D/A converters. If needed, the number of channels can be expanded substantially by purchasing expander boards. Four Datel 12-bit D/A boards are available for additional D/A conversion.

ENHANCEMENTS

Thus far, this has been a description of a microcomputer system that provides the capability of a low-level minicomputer. The right side of Figure 1 shows planned changes that will allow the system to outperform many minicomputers. We believe that the real potential of microprocessors lies in multiprocessing with special-purpose software and hardware providing specific functions for each processor. From the be-

ginning of this project, the plan was to have a master MPU, a slave MPU dedicated to collecting analog data, and another MPU (the display subsystem) for displaying analog data. In such a configuration, software would be modular and straightforward. Although the master MPU and the display subsystem communicate through a standard serial interface, the master MPU and A/D subsystem share a block of memory.

Multiprocessing plans were postponed in the fall of 1978 to wait for the introduction of the Motorola M6809 MPU in early 1979. The 6809 is faster than the 6800 and has an improved instruction set, increased stack and index capabilities, and a remarkable array of addressing capabilities. These changes make high-level language programming more efficient (Forsyth & Howard, 1978). Because we programmed in a high-level language, the software can be easily modified to run even more efficiently on the 6809. It is only necessary to recompile our existing code on the new MPL compiler.

The 6809 MPU, since it uses the same bus as the 6800, will fit into our system with only minor modifications to the 6800 MPU board. Indications are that most, if not all, of our modules will work with the 6809. Because of good system design, we will be able to move up to the next generation of microprocessors with minimum effort and at a very reasonable cost.

Multiprocessing plans were postponed because the 6809 will include improved hardware and software to facilitate multiprocessing. As illustrated in Figure 1, the master and slave units are both complete microcomputers sharing a segment of RAM. Under control of the master, the slave MPU collects A/D data and deposits them in the common memory. The master MPU follows behind the slave, accesses the data in common memory, and performs signal averaging, and so on.

No decision has been made concerning the best way to allow the two MPUs to access the common memory. We have seen an experimental multiprocessing system using two synchronized 6800 MPUs sharing common memory on opposite phases of the system clock. With this system the master MPU could process data one-half cycle after the A/D system collects them. In a second approach, tri-state switches, under control of the master MPU, can be used to select the MPU that has access to the address bus for the common memory. In this technique, the slave MPU collects data, deposits them in a 2-kilobyte block of memory, then turns the block over to the master MPU for processing while the next 2-kilobyte block is filled with data. We are experimenting with the two systems so that we will know which will serve us best when the 6809 becomes available. Both techniques result in signal-averaging capabilities approaching those obtained with the best signal-averaging devices. The transition to multiprocessing with the 6809 should be relatively easy because, even now, the software for A/D data collection is modular and is called as if there were two MPUs.

In summary, the existing microcomputer system

provides a reliable solution to many laboratory control problems. With our high-level language, and with the addition of multiprocessing using the next generation 6809 microprocessors, we will have a system with the unique combination of power and ease of use.

REFERENCE NOTE

1. Millman, B. *PSYPAL: A computer language for the control of psychological experiments*. Calgary, Alberta: University of Calgary, Department of Psychology Technical Report, 1971.

REFERENCES

- DILLON, R. F., DUMONTIER, L. R., BEZANSON, W. R., & BRENNAN, R. A high-level language for control of psychology experiments. *Behavior Research Methods & Instrumentation*, 1978, **10**, 315-324.
- FORSYTH, C. H., & HOWARD, R. J. Compilation and PASCAL on the new microprocessors. *Byte*, April 1978, pp. 50-61.
- TOMBAUGH, J. W. A user's language and operating system for behavioral research. *Proceedings of the Digital Equipment Corporation Users' Society*, Ottawa, Canada, 1974.
- TOMBAUGH, J. W., DILLON, R. F., MILLMAN, B., & BEZANSON, W. R. A microcomputer-controlled laboratory: Software. *Behavior Research Methods & Instrumentation*, 1979, **11**, 301-310.
- WILES, M., MUSA, F., RITTER, T. F., BONEY, J., & GUNTER, T. Compatibility cures growing pains of microcomputer family. *Electronics*, 1978, **51**(3), 95-103.

NOTES

1. Motorola Semiconductor Products, Inc., Box 20912, Phoenix, Arizona 85036.
2. Southwest Technical Products Corporation, 219 West Rhapsody, San Antonio, Texas 78216.
3. Lambda Electronics, 515 Broad Hollow Road, Melville, New York 11746.
4. Creative Micro Systems, 6773 Westminster Avenue, Westminster, California 92683.
5. Microware Systems Corporation, P. O. Box 954, Des Moines, Iowa 50304.
6. Advanced Micro Devices, 901 Thompson Place, Sunnyvale, California 94036.
7. PerCom Data Company, Inc., Department B, 310 Barnes, Garland, Texas 75042.
8. Datel Systems, Inc., 1020 Turnpike Street, Canton, Massachusetts 02021.