Behavior Research Methods & Instrumentation
1978, Vol. 10 (2), 254-258

A multi-user on-line 8080 microcomputer system

EDWARD K. CROSSMAN and JOSEPH G. WILLIAMS
Utah State University, Logan, Utah 84322

A laboratory control system based upon the Imsai 8080 microcomputer is described. This system is capable of programming separate events in each of five animal operant chambers and recording the resulting behavioral data. The interface between the computer and the chambers is compatible with the 28-V dc nature of these chambers and various peripheral devices, such as cumulative recorders. At present, system software is based on the 8080 assembly language, while the BASIC language is used for data analysis. Cost considerations and comparisons with minicomputers are discussed.

Prior to the introduction of the microcomputer system described in this paper, control equipment in the Psychology Animal Laboratory at Utah State University consisted exclusively of electromechanical 28-V dc modules, which programmed the various stimulus events inside rat and pigeon operant chambers. This network also recorded simple response events, while a PDP-8 minicomputer recorded more complex aspects of behavior and performed simple data analysis. In attempting to upgrade the PDP-8 system to take over the programming function, it was discovered that the cost of such an expansion exceeded available funds. About the same time several manufacturers announced the availability of low-cost microcomputers, and the decision was made to develop an entirely new system centering around the Imsai 8080 microcomputer. This particular machine was selected because of its substantial power supply and the ease with which it could be expanded.

The following criteria were established. The microcomputer system should be capable of programming and recording fairly complex events associated with at least five three-key operant chambers that contained IEE in-line projectors behind each key, and a maximum of six symbols per IEE projector should be capable of being illuminated by the microcomputer. Also, the microcomputer system should be compatible with the existing 28-V dc electromechanical network, and the speed of the system should be such that each of the five chambers could be sampled once every 10 msec.

These primary objectives were accomplished. However, inasmuch as electromechanical components are relatively easy to program, we hoped that programming the microcomputer system would be simple, so that graduate students and faculty would abandon relay programming in favor of computer programming. Initially, we thought the BASIC language would suit this purpose, but the slow BASIC programs necessitated use of more difficult assembly language programming. This discouraged some individuals from making the transition to the microcomputer system.

THE ESSENTIAL SYSTEM

Figure 1 shows the essential computer system and peripheral devices, excluding the interface; Table 1 shows associated costs.

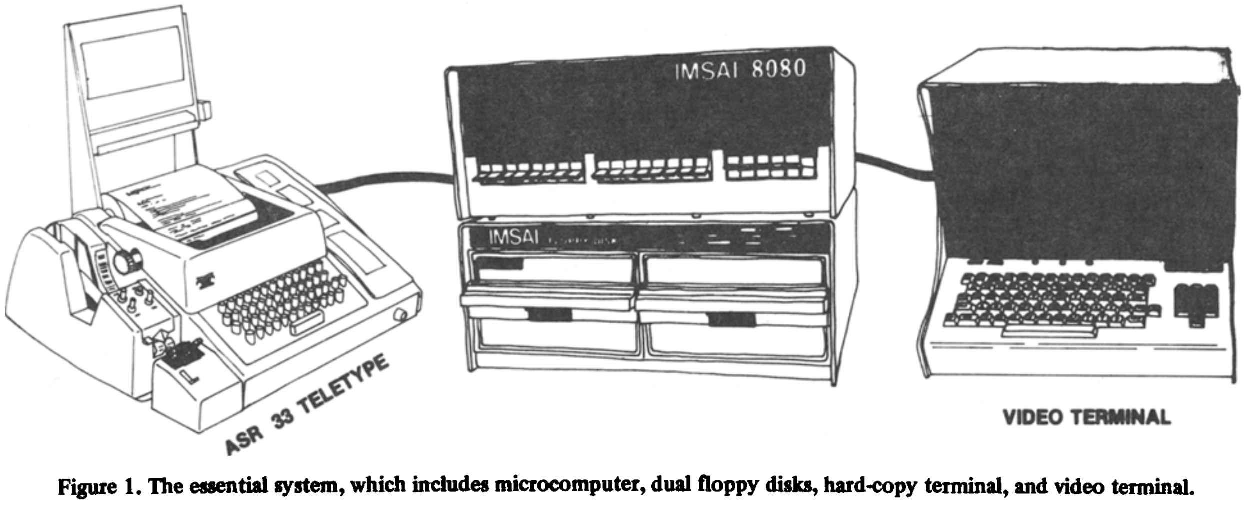

Figure 1. The essential system, which includes microcomputer, dual floppy disks, hard-copy terminal, and video terminal.

Table 1
The Essential System

1. Imsai 8080 Microcomputer (with 22-slot motherboard)	$ 860
2. 28k RAM Static Memory (500 nsec; various manufacturers)	1060
3. 3P+S Input/Output Board (Processor Technology)	190
4. PIC-8 Priority Interrupt and Clock (Imsai)	178
5. Dual-Drive Floppy Disks and Controller (Imsai)	3000
6. Disk Operating System (Software)	150
7. ASR-33 Teletype (Used)	700
8. Video Termial (Lear Siegler ADM-3 or comparable)	1000
Total (Includes Assembly Costs)	$7138

The heart of the system is the Imsai 8080 microcomputer, an 8-bit (1 byte) machine with a speed of 2-9 microsec to execute a single instruction. (This compares with the 1.5-3.5 microsec for a PDP minicomputer). The 8080 microprocessor chip, the heart of the Imsai, supports 78 basic instructions, about twice the number of instructions available to the user of PDP-8 minicomputers. However, the 8-bit microcomputer requires 2 bytes to address a memory location, whereas the PDP-8 minicomputer requires only one 12-bit word. On the other hand, the minicomputer cannot directly address all of memory, which necessitates additional instructions to handle indirect addressing. The microcomputer can access any part of memory and memory is much cheaper for the microcomputer [presently about $450 for a 16k static RAM (read and write memory) kit].

Another characteristic of the Imsai 8080 microcomputer is the S-100 bus located within the computer's mainframe. In our system (see Figure 1) the following cards (see Table 1 for costs) reside in the S-100 bus: (1) static RAM memory cards that provide a total of 28k of storage, (2) a PIC-8 card (priority interrupt) that contains a programmable clock capable of providing a signal every 1, 2, or 10 msec, which provides a time base for determining the length of time between two events, (3) a 3 P+S (three parallel and one serial port) card that provides the means for connecting either the Teletype, which provides typewritten copy, or the video terminal to the Imsai. The video terminal and keyboard are used to develop programs, examine data, etc; when a hard copy is desired, the ASR-33 Teletype is used. Another two cards interface (connect) the dual floppy disk system. One disk, capable of storing about 250k bytes, contains the 8080 assembler, BASIC compiler, and lab programs which run the various experiments. The second disk is used for data storage. Together the two units permit the copying of one disk from another. Finally, one card provides the connection between the Imsai and the laboratory interface. With the exception of this latter card, the cost for the essential system is $7,138 (Table 1), which includes the cost of assembling the Imsai and the various cards purchased in kit form.

The essential system described above is extremely flexible and powerful for many experimental situations which require the accurate measurement and temporal coding of digital data. However, as presently configured, the essential system is not particularly fast in comparison to comparable minicomputer systems. This is an important consideration, particularly in regard to software. Initially, we planned to use BASIC to control all aspects of the system, but the lack of speed forced us to program in 8080 assembly language, which is more difficult to teach and use. But it is possible to get around this problem in several ways. One is to replace the 8080 microprocessor with an Imsai 8085 microprocessor (speed is about 50% faster) and to replace the present 28k of memory with memory whose speed is compatible with the 8085. At present prices, this conversion costs about $1,900. The resultant instruction cycle time range is about 1-5 microsec, rather than the 2- to 9-microsec range of the present system. Another partially satisfactory solution is to await the development of a disk BASIC capable of linking up with assembly-language subroutines, which should be available in the near future. But even so, some assembly language programming will still be necessary for those applications in which speed is important.

Although the cost of the essential system, $7,138, is still too expensive for many investigators, it is considerably less expensive than a comparable disk-based minicomputer system. Is it possible to reduce the cost of the essential system without destroying its effectiveness? The answer is a partial yes. At current prices the assembled Imsai 8080 (or comparable) and 28k of static RAM memory still costs about $1,900, but the memory cost is dropping rapidly. A low-speed used printer, such as the ASR-33, costs about $500-750, but prices should diminish for acceptable printers. Savings can be achieved with substitutions for the video terminal and floppy disks. A used black and white (B/W) television set can be purchased and converted to a video terminal for about $125. A video terminal interface board (e.g., Polymorphics VTI), costing about $200 assembled, is necessary to interface both the TV set and a used ASCII-encoded keyboard (about $100) to the Imsai. Finally, a programmable high-speed dual-cassette drive (about $900) could replace the floppy disks. It would be necessary to retain the PIC-8 board for its clock function. Thus, the total cost of this alternative system

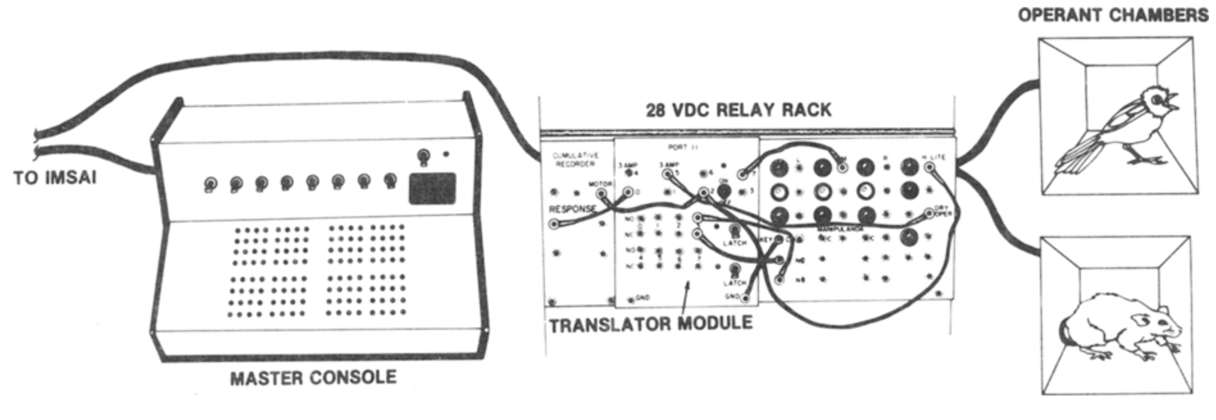

Figure 2. The interface system for connecting the essential system to the animal operant chambers.

is about $4,028, approximately half that of the system described in Table 1. Assembling programs with the digital cassette system are much slower than with the floppy disks, but, otherwise, computer performance would not be changed. Some costs should be anticipated for technical consulting by an electronics technician.

THE INTERFACE

Figure 2 shows the components used to interface the essential system to the 28-V dc animal chambers. Table 2 shows the associated costs. A 15-conductor cable runs from the computer to a single translator module located on a 28-V dc relay rack. Additional modules (up to 15) can then be daisy-chained to this initial module. The number of modules that can be handled depends on system speed and the complexity of the experiments. At present, we can run a maximum of five experiments at any one time. Eight of the 15 conductors are for transmitting 1 byte of data and two are for control, indicating whether data is to be sent to the translator modules (output from the computer) or from the translator modules (input to the computer). Four conductors are used to indicate the particular translator module that is either sending or receiving information; one ground wire is provided.

One function of the translator module is to convert the computer's 5-V dc signals into negative 28-V dc signals. These signals are then directed, via snap leads, to connection panels which are wired to the operant chambers, to cumulative recorder panels, etc. For this purpose, eight studs are located on the face of each translator module; two of these are capable of driving a 3-amp load, leaving six outputs from the computer to drive key lights, house lights, etc., inside the operant chamber. Through the use of relays, these six binary outputs can be decimal coded so that it is possible with a single translator module to control 31 separate events inside the operant chamber. Below the eight studs are eight pairs of studs (input to the computer) labeled NO (normally open) and NC (normally closed). The NO and NC contacts of the microswitch behind the manipulandum are wired via the connection panel to these studs. Thus, the computer recognizes that an event has occurred when the 28-V dc ground changes from the NC to the NO stud. This method provides protection against contact bounce from the manipulandum. If, because there are a large number of simultaneous users on the system, it is impossible for the computer to read a translator module with 10 msec, two latch switches are provided, one for each 4 bits of the 1-byte input to the computer. In the "on" position, these switches hold the input byte until the computer reads it. Accompanying LEDs indicate switch status. Finally, an on-off power switch can isolate a particular module from the system.

The other major component of the interface is the master console (see Figure 2), which has two functions. Sixteen rows of eight lights (LEDs) each monitor information sent from the computer to the modules. On the vertical surface of the console are located eight switches and a two-digit thumbwheel switch, above which is a switch and accompanying LED. In the "on" position this switch indicates to the computer that it

Table 2
Interface

1. Engineering Design, Parts, and Construction of Master Console and One Translator Module	$1200
2. Parts for Seven Translator Modules	870
3. Construction Cost for Translator Modules	400
4. Cable (35 m, 15 conductor)	150
Total	$2620

should read the status of the eight switches and interpret this information as input from the particular translator module whose number appears in the thumbwheel switches. This permits a single user to simulate the operation of his/her own translator module to determine if the computer is responding appropriately. Because the computer recognizes only the particular module indicated by the thumbwheel switches as being in a test condition, other users can run simultaneously without being disturbed.

Cost of the system is listed in Table 2. A low-cost alternative, which does not include the master console and the latching feature of the translator module, consists of an opto-isolator relay control board that plugs directly into the S-100 bus of the Imsai. This board consists of eight optically isolated reed relays that can be operated by the computer (output) or can accept an 8-bit word from the external environment (input). Such a board costs about $120 in kit form, and can be purchased at a local computer store. Having no experience with this alternative, we cannot say with certainty whether it is a completely satisfactory substitute for the described interface.

THE SOFTWARE

A considerable amount of time was spent learning the 8080 instruction set. In the opinion of several programmers who have programmed both the PDP-8 and the 8080, the assembly language associated with the 8080 is as easy (or easier) to use as PDP-8 assembly language. Advantages of the 8080 language include the flexibility afforded by a large instruction set and direct addressing of every location in memory. Even so, it is no easy task to motivate students and faculty to use the 8080, or any other, assembly language.

The Imsai disk system is an advanced floppy disk system called CP/M. This software includes a powerful text editor, assembler, and debug program, as well as utility programs to transfer, rename, and copy disk files. These capabilities greatly speed up program writing.

The laboratory control program consists of a group of executive programs designed to simplify the programming task. This executive monitor allows users to request that control be transferred to their own subprogram on the basis of a change in the input port assigned to them (e.g., animal subject input) or on the basis of the passage of time. Thus, the user's program need only place these requests and provide routines in his/her own subprogram to handle their occurrence.

Figure 3 shows the sequence of these events. The program begins and accepts time requests (actual time intervals plus the addresses in the user's subprogram where control should be transferred once the executive has determined that the intervals have expired). The time requests are then placed by the executive in a buffer, TIMBUF (not shown in Figure 3). At the same time the executive collects input requests from each user. These take the form of the port identification number of a particular user and the address in each user's subprogram to which the executive should transfer control upon receiving an input. These requests are then placed in another buffer called INBUF (not shown in Figure 3). At this point, the executive returns to the background program which simply examines a third buffer, ADDBUF (not shown in Figure 3) to determine whether any user addresses (through which the executive transfers control to the user) are in the buffer. Assuming that there are none, the executive remains in a loop examining ADDBUF.

Suppose at this point that a clock tick occurs; that is, the clock is programmed to interrupt every 10 msec and, as Figure 3 shows, the executive monitor begins check-

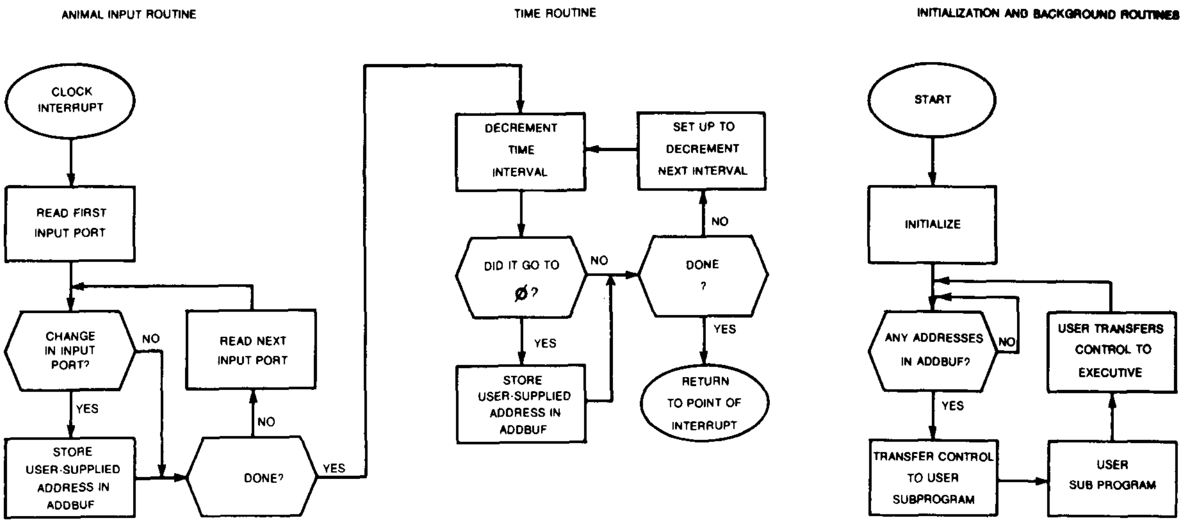
ANIMAL INPUT ROUTINE
TIME ROUTINE
INITIALIZATION AND BACKGROUND ROUTINES
CLOCK INTERRUPT
READ FIRST INPUT PORT
CHANGE IN INPUT PORT?
NO
YES
READ NEXT INPUT PORT
STORE USER-SUPPLIED ADDRESS IN ADDBUF
DONE?
NO
YES
DECREMENT TIME INTERVAL
SET UP TO DECREMENT NEXT INTERVAL
DID IT GO TO Ø?
NO
YES
DONE ?
NO
YES
STORE USER-SUPPLIED ADDRESS IN ADDBUF
RETURN TO POINT OF INTERRUPT
START
INITIALIZE
ANY ADDRESSES IN ADDBUF?
NO
YES
USER TRANSFERS CONTROL TO EXECUTIVE
TRANSFER CONTROL TO USER SUBPROGRAM
USER SUB PROGRAM

Figure 3. Flowchart of executive monitor software and its interaction with user subprogram.

ing the animal input ports to see if an event has been transmitted by one of the animal chambers. It accomplishes this by comparing the previous status of a particular port to its current status. If such an event has occurred, the executive, by examining INBUF, determines to which user the event belongs and the user-supplied address associated with this particular input event. This address is then placed at the top of ADDBUF and the search for events coming in through other ports continues until all currently active ports have been examined.

Then the executive jumps to the time routine and begins to decrease, by one, each interval stored in TIMBUF. When any of these intervals are decremented to zero, the executive again selects the user-supplied address from TIMBUF and places it in ADDBUF. After all intervals in TIMBUF have been decremented, the executive returns to the background program and examines ADDBUF. If there are addresses in ADDBUF, the executive jumps to the first such address and turns program control over to the user at that point. Upon completion of the subprogram, the user, if appropriate, must reset the interval in TIMBUF and turn the control back to the executive, which continues its search for addresses in ADDBUF. When all user addresses have been serviced, the executive remains in the background routine, repeatedly examining ADDBUF until the next clock tick occurs. Of course, a clock tick could cause an interrupt at any point in the total sequence, but the executive keeps track of the location where the interrupt occurred and returns to that location. The portions of the executive program that check the status of the input ports (animal input routine) and decrement the time intervals in TIMBUF (time routine) are deliberately designed to require less than 10 msec so that the executive cannot be interrupted while performing these functions.

DATA STORAGE AND ANALYSIS

During the running of an experiment, data are stored in memory. Subsequently, the data are written on disk in either binary or ASCII format. ASCII files can be read directly by BASIC, as well as edited, typed, listed, etc., just as any other file in the system. BASIC greatly simplifies data analysis.

CONCLUSION

The microcomputer system described here presents an attractive alternative to electromechanical systems, minicomputers, etc., for laboratory control and recording of behavioral events. The system is based on the Imsai 8080 microcomputer with 28k of static RAM memory, dual floppy disks for program and data storage, a video terminal-keyboard, an ASR-33 Teletype for hard copy, and an interface to drive about five 28-V dc animal operant chambers. Programming is accomplished with 8080 assembly language. BASIC programs perform data analysis.

As with any new technology, a variety of problems required technical consultation. The computer store where the components were purchased, the Electrical Engineering Department at Utah State University, and several different electronic technicians in our laboratory all rendered invaluable assistance. But technical problems of design are gradually diminishing. The microcomputer industry and related publications have made significant strides toward: (1) informing the user as to the compatibility of this manufacturer's components with that manufacturer's mainframe; (2) discouraging kit building, where many users' problems appear in the first place; and (3) making available from a single source all of the components necessary to construct a system such as we have described. This latter benefit simplifies the decision-making process required for purchasing as well as servicing. Coupled with the rapidly decreasing costs of many of the components, particularly memory, these features make it unlikely that any other procedure of laboratory control, including relays, minicomputers, etc., can long compete.

Major drawbacks of this system are the relatively slow instruction cycle time and the lack of a high-level language that can be utilized by psychologists with a minimal amount of training. Slow instruction time can now be overcome by choosing a microcomputer that utilizes a faster microprocessor, such as the Imsai 8085. With increased speed, it is likely that BASIC will be fast enough to supplant assembly language programming for many applications. The widespread acceptance of BASIC means that many programs, some of which will benefit the experimental psychologist, will be written and available for a modest fee. The ideal solution to lab control is the development of a language specifically designed for psychologists, such as SCAT (Grason-Stadler), SKED (State Systems, Inc.), or INTERACT (BRS/LVE), which currently run on minicomputers.

Perhaps the ultimate system will consist of an animal chamber with its own microprocessor and a large amount of memory. During the experimental session, data can be stored in memory and subsequently dumped onto a mass storage device, such as a floppy disk, which plugs directly into the animal chamber. Data analysis can then be performed on a remote microcomputer that reads the floppy disk. This same computer can also be used to develop programs and place them on an erasable chip that can be plugged into the animal chamber and control chamber events. To some extent, such a system decentralizes the dependence on a single computer to handle all functions. If one of the chamber microcomputers fails, the entire system is not paralyzed.