

University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

Faculty Publications, Department of Psychology

Psychology, Department of

1979

An economical microcomputer design for the generation and display of timed visual stimuli: The TDL VDB

John H. Flowers

University of Nebraska-Lincoln, jflowers1@unl.edu

Mark H. Murphy

University of Nebraska-Lincoln

Follow this and additional works at: <http://digitalcommons.unl.edu/psychfacpub>



Part of the [Psychiatry and Psychology Commons](#)

Flowers, John H. and Murphy, Mark H., "An economical microcomputer design for the generation and display of timed visual stimuli: The TDL VDB" (1979). *Faculty Publications, Department of Psychology*. 458.

<http://digitalcommons.unl.edu/psychfacpub/458>

This Article is brought to you for free and open access by the Psychology, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Faculty Publications, Department of Psychology by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

An economical microcomputer design for the generation and display of timed visual stimuli: The TDL VDB

JOHN H. FLOWERS and MARK H. MURPHY
University of Nebraska, Lincoln, Nebraska 68588

Recent developments in microprocessor-based technology should be of particular interest to psychologists concerned with perceptual and cognitive processes. This paper describes the advantages of a Z-80-based system that uses the Technical Design Labs video display board (VDB) for the generation and control of visual stimuli.

Microprocessor-based technology has brought within the reach of individual investigators in small institutions the power of experimental control and data acquisition operations previously associated with large-scale minicomputer-based research laboratories. For the experimental psychologist interested in perceptual and cognitive processes, the generation and temporal control of visual displays in conjunction with the capacity to measure response choice and latency are fundamental laboratory requirements. Currently existing microcomputer systems are ideally suited for such purposes and in many instances may be less cumbersome and more flexible (not to mention cheaper by orders of magnitude) than the more elaborate combinations of minicomputers and CRT displays. What is even more striking, however, is that microcomputer systems offer the possibility of totally replacing nearly all the functions served by traditional stimulus display equipment such as memory drums and tachistoscopes for less total cost, a fact that some traditional equipment manufacturers may have been slow in recognizing.

The present paper describes a system recently assembled in our laboratory utilizing the Technical Design Labs (TDL) video display board (VDB), which is an exceptionally versatile component for generating and displaying the types of visual fields that are useful to experimental psychologists. We believe our experience illustrates some typical problems, compromises, and solutions in implementing this type of system that may be useful to others.

BASIC DESIGN

Some desirable characteristics of a microcomputer-based visual information processing laboratory have

The project described in this report was funded in part by the Happold funds from the University of Nebraska Foundation and by a summer fellowship grant to the first author from the University of Nebraska Teaching Council.

been described by Santa and Streit (1978). An important objective of such a system is ease of operation by persons having no sophisticated knowledge of computers. (Santa and Streit's norm of usability by undergraduates after one or two brief training sessions is a good one.) Also, the system must be flexible enough to perform a wide variety of tasks with minimal modification. To achieve these objectives, Santa and Streit chose to create a library of highly general software subroutines and a master program that effectively form a highly general, custom-designed experimental monitor system. We have adopted a fundamentally different approach that relies almost entirely upon the selection of flexible hardware components (e.g., the TDL VDB) and commercial software packages [e.g., the TDL Zapple monitor, the North Star disk operating system (DOS), and North Star BASIC] for obtaining simplicity of operation. Generality is obtained by the availability of a large number of small but simple programs, each designed for a specific experiment. These can be immediately accessed and run from disk by one or two simple keyboard commands. Given the simplicity of the interfacing and ease of control afforded by our choice of hardware (in particular, the VDB), most of our experiment-specific routines share common structures and subroutines and are sufficiently straightforward that programming time associated with setting up a "new" experiment is trivial. In addition, this type of arrangement provides an excellent opportunity for interested undergraduate and graduate students to gain experience with BASIC and assembly language programming in a relatively painless manner, since they are surrounded by large numbers of documented examples. However, this approach makes a sharp distinction between being able to operate the system (e.g., create stimulus sequences and run preexisting programs), which requires no contact with programming, and the actual creation of new programs, which requires at least an open mind toward learning the fundamentals of assembly language (and familiarity with BASIC).

Hardware

The microcomputer is housed in a IMSAI 8080 mainframe, but the CPU is the Technical Design Labs ZPU (a Z-80 as opposed to an 8080 CPU). The system also contains the TDL system monitor board (SMB), the TDL video monitor board (VMB), a North Star disk drive with DOS, a custom-built real-time clock (based on the Intel 8253), and 16K of RAM. Externally, we use a Koyo video monitor, a Cherry ASCII-encoded keyboard interfaced to the VDB, and a custom-made interface containing banana jack inputs and outputs from six parallel ports and relays that can be driven to operate shutters, projectors, and other machinery. This custom made I/O device is interfaced through an IMSAI PIO/6. A Teletype from the Automated Data Systems 1800A computer in an adjoining laboratory is occasionally borrowed to obtain printed copy of programs or data.

The components of primary relevance to the present discussion are the TDL system monitor board (SMB) and North Star DOS (which together allow ease of operation by inexperienced students) and the TDL VDB, which in combination with the keyboard generates and controls the displays.

TDL System Monitor Board and North Star Disk Operating System

The TDL Zapple monitor, contained in ROM on the SMB, contains an extremely powerful set of keyboard commands for such tasks as examining, loading, substituting, or moving information in memory locations, in addition to commencing program execution, testing input and output ports, and so on. Obviously, Zapple is an extremely useful debugging aid for creating an assembly language program. Its most useful feature for the user who is simply running experiments is the ability to write ASCII characters directly from the keyboard into memory, including both upper- and lowercase alphanumeric and control characters. Another Zapple command allows the display of all printing (noncontrol) ASCII characters in memory in a particular block of memory (nonprinting characters appear as periods). Thus a list of alphanumeric stimuli can be easily scanned for errors prior to its being copied on disk or used in program execution.

The DOS, which is loaded from diskette upon system start-up by a bootstrap loader along with the VDB driver routines, is easily accessed from Zapple by the command G2028. A very simple North Star command set (which has mnemonics including LF, SF, CR, and DE for load file, save file, create file, and delete file, respectively) allows the operator to load programs and/or data into memory, or dump stored programs and/or data to a disk file. Files containing machine code can be designed for "load and go" operation. The statement GO POSNER, for example, might load and begin executing a letter-matching experiment contained on a disk file named POSNER.

North Star BASIC

One serious obstacle to setting up a microcomputer-based system for a psychology laboratory is that documentation of various components is sketchy at best, and certainly not oriented toward applications in psychology. A period of time "playing" with the system is absolutely essential for discovery of the best ways to accomplish tasks such as controlling the video board, obtaining data from subjects via the keyboard, and so on. Initially, we found that assembly language programming offered the most direct approach to this learning process. Furthermore, it seemed that the lack of speed and memory constraints afforded by a high-level language interpreter would probably make BASIC impractical for data acquisition and experimental control.

More recently, after turning our attention to the disk file handling capabilities of North Star BASIC and after having written assembly language subroutines for typical control and display functions, we found that writing programs in BASIC instead of assembler is remarkably efficient for most applications. We simply load a file of machine language subroutines for time-critical operations (e.g., erasing the screen, tachistoscopic display, and accessing the real-time clock) into an "out of the way" block of memory and call these routines from BASIC. Stimulus data can then be read from disk and response data can be written to disk as the program runs. Data analysis by BASIC routines is vastly simplified.

North Star BASIC appears to us to be a quite powerful version of BASIC in addition to its file handling capabilities. However, several manufacturers, including TDL, have recently marketed disk-based systems that may have equal or greater capabilities.

Video Display Board

Computer display devices are not designed to be tachistoscopes, and it is often very difficult to determine the suitability of a product for stimulus display by reading the literature and specifications provided by the manufacturer. The TDL VDB is clearly one product that is exceptionally well suited for the psychology laboratory; furthermore, its price (approximately \$400, assembled and tested, not including a video monitor) makes it a bargain compared to most CRT terminals. Among the features of the VDB that make it so versatile are the following.

Two-field capability. The VDB contains 4K of RAM, organized into two 2K pages. Each page corresponds to the 2,000 possible character positions in two different display fields, either of which may be instantly selected by a single OUT command from the CPU.

Blanking capability. Characters may be written to either page without displaying the page or the writing process. Either field can then be displayed with a single OUT command from the CPU. Blank intervals between stimulus fields can be used to write in new material.

Simplicity of control and writing. One VDB port, the control port, determines which of the two display fields is selected, whether that field is displayed (or the screen remains blank), whether a cursor is visible, and if so, whether it blinks. In addition, the control port locates the position of the cursor on the screen. The manufacturer's manual is not as clearly written as we would have liked, particularly in explaining the use of the control port to accomplish all of the above. Experimentation soon convinced us, however, that the procedure was relatively simple. Table 1 illustrates some of the more useful commands in both BASIC and assembly language for operation of the video control port. On our machine the address of this port is B0 in hex or 224 decimal (the user may readdress the port as desired). Note that the mnemonics for assembler commands are Intel-style and thus compatible with 8080 assembly language.

Once the page has been selected and the cursor located, writing to the screen is accomplished by outputting characters from memory using the VDB data port (addressed E1H or 225 decimal on our machine). This process is fast: about 2 microsec per character. If speed is not an issue, a simple PRINT command from BASIC may be used. The cursor position is stepped up one position after each character is output. A PRINT command causes a return to the leftmost position of the next line, unless a comma is the final character of the statement.

Display speed. Some applications require precise control of timing at very rapid durations (less than 10-15 msec). Obviously, a raster scan video monitor should not be considered for such purposes. However, the frame time on the VDB is the reciprocal of the line frequency, or about 16 msec. The P4 phosphor is reasonably fast in decay, and it is possible to produce tachistoscopic images suitable for demonstrations of iconic memory and masking phenomena as long as the researcher is not concerned with precise specification of the duration of the display. Technically, display duration will be constant in multiples of the raster scan cycle time. Many memory and reaction time tasks are not at all critical in terms of precise numeric control of very brief durations; thus, a video monitor is as suitable as any display device for such purposes.

Characters available. The full ASCII set of both upper- and lowercase alphanumerics as well as special characters can be displayed. The control, or nonprinting, characters can be used for graphics purposes and for creating special patterns, since they appear as a special set of graphics characters. While the flexibility for drawing patterns on the screen is not that of an analog-controlled CRT where individual points may be positioned with finer resolution, a 160 by 75 matrix of small rectangular locations is available; any of these can be either on or off. Patterns of considerable complexity can be created in this way. Since the raster scan refreshes the entire screen, the user does not have to worry about flicker when large numbers of points are plotted.

Access to the keyboard a response acquisition device. One very useful feature of the VDB is that it allows the use of an ASCII-encoded parallel keyboard as an input port and thus as a set of response buttons, each of which reads in a unique code. The command IN 0E3H will place the ASCII code of the key pressed into the accumulator. This value can then be compared with predetermined values to determine whether a response has been made and, if so, which one. In reaction time experiments, it is best to program such a sequence in assembly language or to include it as a machine language subroutine to be called from BASIC. For less time-critical applications, the BASIC statement R = INP(227) places the decimal equivalent of the ASCII value of the keyboard port in the variable "R." The keyboard may also be used in conjunction with the INPUT statement that requires a carriage return to be issued before an execution. Although it is useful for obtaining verbal responses from subjects, this procedure is not suitable for speeded tasks.

SOME USEFUL SUBROUTINES

The brief program segments shown in Table 2 illustrate some typical operations of the VDB and associated equipment. These may be included either as segments of an assembly language program or as machine language subroutines that can be called from BASIC. The subroutine LXY illustrates a fast means of positioning the cursor to the coordinates contained in register pair DE when the routine is called. BLANK

Table 1
Some Useful VDB Control Port Commands

Function	Assembler	BASIC
Display Page 0 with blinking cursor (normal mode)	MVI A, 80H	OUT 0E0
Display Page 1 with blinking cursor	MVI A, 81H	OUT 0E0
Display Page 0 without cursor (stimulus display)	MVI A, 90H	OUT 0E0
Display Page 1 without cursor (stimulus display)	MVI A, 91H	OUT 0E0
Select Page 0, but keep screen off	MVI A, 0B0	OUT 0E0
Select Page 1, but keep screen off	MVI A, 0B1	OUT 0E0
Move cursor position horizontally to the Xth column from the left ($0 \leq X \leq 79$)	MVI A, X	OUT 0E0
Move cursor position vertically to the Yth line from top ($0 \geq Y \geq 24$)	MVI A, Y	ORA 0C0 OUT 0E0

Table 2
Some Typical Program Segments Utilizing VDB Commands

	Purpose
BLANK: LXI B, \uparrow D2001 RPT: MVI A, 0 OUT 0E1 DCX B MOV A, B ORA C JRNZ RPT RET	Erase all characters in the selected page. Useful for quickly removing all material from a field before writing in a stimulus display.
LXY: MOV A, D OUT 0E0 MOV A, E ORA 0C0 OUT 0E0 RET	Quickly move cursor to coordinates stored in register pair DE.
STRT: MVI A, 91 OUT 0E0 CALL WAIT; delay loop MVI A, 90 OUT 0E0 CALL TIMER1; start timer	Turn on fixation field (Page 0) for specified time interval, then replace it with a stimulus field (Page 1), turn on a timing clock.
RKB: IN 0E3 CPI "F" JRZ SDATA CPI "J" JRNZ RKB	Scan keyboard for response (F or J). Store reaction time data, response choice, and return.
SDATA: CALL TIMER2; store latency CALL SRESP; store response RET	

simply erases all the characters contained in the field that has been selected by writing out 2,001 zeros to that page (zeros, like ASCII spaces, appear as blanks). BLANK is thus a good means for "starting with a clean slate" before constructing a new stimulus display. The final subroutine sequence is typical of a program segment that might be included in a reaction time experiment. It turns on a fixation field (which has previously been written into Page 0 of the VDB), calls a wait loop routine (not shown), then starts a timing-clock routine (also not shown), which turns on the stimulus field until the subject responds with either an "F" or a "J" on the keyboard. It then calls a routine to read the clock value, shuts off the display on the screen, and returns from whence it was called. This routine assumes that the subject has pressed a key other than the F or J between trials, perhaps as a means of initiating a trial. The clock and waiting routines are not included here since they depend on specialized components of our particular system that have not been discussed.¹

PROBLEMS

Our experience in developing this laboratory computer system has revealed two typical problems faced by nearly everyone undertaking a similar venture. These are insufficiencies of documentation (at least for psychologists) and the education of potential users. Given the documentation of the VDB in particular, it seems highly unlikely that persons unfamiliar with assembly language programming would be able to implement and operate the system. For potential users of the system, we have written a short practical introduction to operation principles, including specific programming examples. Since such a document by necessity includes details unique to a given system and requires constant revision as new components are added, it is probably of little use to users of similar equipment in other laboratories. We feel that, despite the seeming inefficiency of writing your own users guide, the practice is almost mandatory. Otherwise, operation procedures become "oral traditions" among only a few individuals, who find themselves bombarded by questions and problems from less knowledgeable individuals anxious to use the system.

Implementation problems are associated with assembling systems incorporating components from different manufacturers. None of the TDL software described here is designed to operate with a disk system. Interfacing the North Star DOS to the SMB and VDB requires the use of a software disk drive routine for accessing disk files when using the TDL test editor and microassembler. This routine is supplied by the dealer. Software interfacing problems can be minimized by obtaining as much of a system as possible from the same manufacturers. Potential buyers should investigate the newer TDL disk systems.

Another implementation problem involves wiring a keyboard and monitor to the VDB. It is less convenient to set up than a "ready to go" terminal, but the inconvenience is compensated by the VDB's low cost and suitability as an experimental display device.

REFERENCE

- SANTA, J. L., & STREIT, P. A microprocessor reaction time laboratory. *Behavior Research Methods & Instrumentation*, 1978, 10, 352-355.

NOTE

1. The authors would be happy to provide sample listings of programs that incorporate these features and subroutines upon request.