Behavior Research Methods & Instrumentation
1982, Vol. 14(6), 534-538

Tachistoscopic timing on the TRS-80

ROBERT L. MAPOU
Emory University, Atlanta, Georgia 30322

A hardware/software system is described that enables the TRS-80 Model I microcomputer to be used as a tachistoscope. The system synchronizes stimulus presentation with the 16.7-msec scanning rate of the cathode-ray tube. Hardware and software configurations for reaction timing and stimulus presentation are presented and discussed.

The Radio Shack TRS-80 Model I is a popular microcomputer that has experienced substantial use in behavioral and perceptual-cognitive laboratories. The TRS-80 is a Z-80-based system that can be connected to a range of peripheral devices, including disk drives for data storage, external clocks for accurate timing, and subject response panels. This paper addresses three issues associated with application of the TRS-80 as a tachistoscope: hardware requirements and configuration, reaction timing, and stimulus presentation.

HARDWARE REQUIREMENTS AND CONFIGURATION

There are a number of ways in which the TRS-80 may be configured for perceptual-cognitive research. One configuration using the LVB Corporation interface has been described by Grice (1981). The system used in this laboratory is illustrated in Figure 1. It consists of a TRS-80 Model I with expansion interface, two disk drives, a serial printer, circuitry used to detect and control cathode-ray tube (CRT) operation, and several other peripheral devices, including an external clock and response panel as an input device for the subject. An HUH Mini 8100 is connected to the TRS-80 bus extension card edge on the expansion interface. The Mini 8100 is a peripheral interface that allows as many as four S-100 bus-compatible devices to be connected to the TRS-80. This approach increases the range of peripheral devices available and allows devices to be accessed via software routines. A 100,000-day clock that runs continuously and records time to .1-msec accuracy is plugged into one of the sockets on the Mini 8100.[1] Each of 15 addressable ports on the clock provides a single digit (0 to 6 or 9) that indicates the current time in the units denoted by that particular port. The first port provides the time in units of .1 msec. Subsequent ports provide the time in units that become progressively larger, up to units of 10,000 days. Port addresses that are referenced by the computer may be changed by a switch on the clock. Determining the time at any point involves reading the necessary ports and performing a sum-of-products operation.[2]

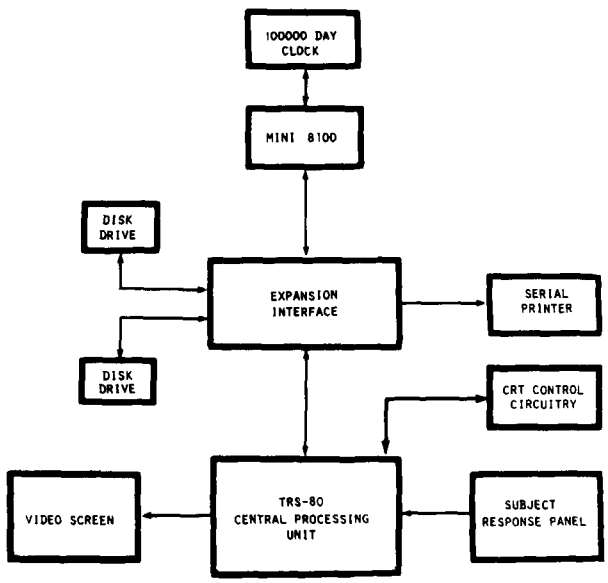

Figure 1. Block diagram of the system hardware configuration.

REACTION TIMING

Attempts to do reaction timing in BASIC (Perera, 1978, 1979) indicate that timing cannot be performed with the required accuracy, even with an external clock. This is due to the fact that BASIC is an interpreted language: Each individual program line is read by the computer, translated into machine language, and executed. The process is time-consuming and limits the accuracy of any timing method. Additionally, the start of the reaction time interval must also occur at the same time as stimulus onset, and this coordination is impossible in BASIC. With the system described in this paper, all timing is done in assembly language.

To determine a reaction time using the described hardware, a program must read the clock at the start of the reaction time interval, read the clock following a

A preliminary version of this paper was presented as part of a symposium at the 28th annual meeting of the Southeastern Psychological Association in March 1982. Thanks to L. D. Byrd and J. B. Richards for their comments during preparation of this paper. Address requests for reprints and additional information to Robert L. Mapou, Department of Psychology, Emory University, Atlanta, Georgia 30322.

  0005-7878/82/060534-05$00.75/0

subject's response, compute the two times, and subtract the first value from the second. Because a multiply operation was not readily available in assembly language for the TRS-80 at the time this system was developed,[3] the system was designed to record and store clock readings via an assembly language routine and compute reaction times in a main BASIC program.

STIMULUS PRESENTATION AND TIMING

Positions on the screen of the TRS-80 occupy a segment of memory from 3C00 to 3FFF,[4] known as video memory, which can be addressed directly from software. Stimuli are displayed by sending the appropriate ASCII codes for letters or graphics characters to video memory. Codes are transferred from video memory to the CRT by the video system. The first programs developed in this laboratory sent stimuli to the CRT using the BASIC command "POKE," which permits the user to send bytes of information directly to memory. Information composing a stimulus was sent directly to video memory. This proved to be unsatisfactory, because the slow execution time of BASIC prevented true tachistoscopic presentation. It was then decided to POKE information to memory locations other than video memory and have an assembly language routine send the stimulus to video memory at a much higher rate, allowing a closer simulation of a tachistoscope. By having several such areas, a number of stimuli could be stored and sent rapidly in succession to video memory. Unlike the Apple II microcomputer, which automatically allocates special areas of memory for this purpose (these are called "display pages," see Cavanagh & Anstis, 1980), the TRS-80 provides no such display areas. Thus, a portion of memory was set aside to simulate a display page for stimulus storage.

However, there is a serious complication inherent in this methodology. While it is possible to precisely determine the time during which a stimulus remains in video memory, this does not mean that the stimulus will appear on the CRT screen for the same time interval. This is due to the fact that the CRT cannot present a stimulus instantaneously but has a beam that scans the screen from left to right and top to bottom, effectively "painting" the contents of video memory on the screen once every 16.7 msec (the inverse of the 60-Hz scanning frequency; Grice, 1981; Lincoln & Lane, 1980; Reed, 1979). This has two ramifications for stimulus presentation. First, a stimulus will always appear on the screen for a multiple of 16.7 msec, since the stimulus will not be complete until the beam has passed through its last point and will not be totally erased until the beam again has passed through the same point. The time between passes of the beam through the same point on the CRT is always a multiple of the scanning rate. Second, transfer of information from video memory may occur at any point during the cycle. Thus, a stimulus may be drawn bottom first, followed by the top, should the beam be at the middle of the screen at the time information is sent to video memory. If reaction timing begins at the time that the information is sent to video memory, there is no guarantee that the interval onset will always coincide with the same point of stimulus onset. To overcome these problems, stimulus onset is synchronized with the vertical synchronization (synch) signal going to the CRT and stimulus display is timed by leaving the stimulus on the screen for a given number of synch pulses, as suggested by Grice (1981), Lincoln and Lane (1980), and Reed (1979). While this limits tachistoscopic presentation to multiples of 16.7 msec, it insures the timing accuracy necessary for this type of research.

SOFTWARE DESCRIPTION

Use of this system requires that a region of high memory be reserved and protected from overwriting by BASIC programs. Addresses referred to are those used with a 48-KB TRS-80 Model I and may be altered for smaller machines. It is assumed that this system will work on the newer TRS-80 Model II or Model III with minimum modification.

Software consists of three programs. A routine to clear information due to a previous stimulus from the display page and the tachistoscopic display and reaction time collection routine are written in assembly language; the main program is written in BASIC. The region of memory from F600 to FFFF is reserved for use by various assembly language routines, including nondisplay-related routines (F600 to F64F) and the display routines (F650 to F72B). Remaining memory is reserved for two display pages. The region beginning at FBFF is used as a 1,024-byte display page for the stimulus to be presented; the region beginning at F7FF contains 1,024 blank characters used to clear the display page.

Figure 2a presents a listing of the display page blank-

(a)

```
00010   ;*************************************************
00020   ;  Display Page Blanking Routine
00030   ;
00040   ;  Call using X=USR(D) where D is a dummy parameter
00050   ;
00060   ;
00070   ;*************************************************
00080           ORG     0F720H          ;63264 Decimal
00090   BSCREN  EQU     0F7FFH          ;Blank screen address
00100   VSCREN  EQU     0FBFFH          ;Display page address
00110   COUNT   EQU     400H            ;No. of bytes to xfer
00120           LD      DE,VSCREN       ;Transfer 1024 blanks
00130           LD      HL,BSCREN       ; to the display page
00140           LD      BC,COUNT        ; prior to sending the
00150           LDIR                    ; stimulus
00160           RET
00170           END
```

(b)

```
10  FOR I%=&HF720 TO &HF72B
20  READ J%: POKE I%,J%: NEXT I%
30  STOP
100 DATA 17, 255, 251, 33, 255, 247, 1, 0, 4, 237
110 DATA 176, 201
```

Figure 2. (a) Assembly language display page blanking routine; (b) BASIC program to POKE this routine into memory.

(a)

```
00160           ORG   0F650H      ;63056 Decimal
00170 VSCREN    EQU   0FBFFH      ;Display page address
00180 BSCREN    EQU   0F7FFH      ;Blank page address
00190 SCREEN    EQU   3C00H       ;Video memory address
00200 KEYBD     EQU   3810H       ;Keyboard buffer address
00210 COUNT     EQU   400H        ;No. of bytes to xfer
00220 PORT      EQU   250         ;Sync/blank port
00230           DI                ;Disable interrupts
00240           LD    IY,KEYBD    ;For keyboard input
00250           LD    A,1
00260           OUT   (PORT),A    ;Blank the screen
00270           LD    HL,(CHAR)   ;Get bad char.
00280           LD    H,0         ;Clear upper byte
00290           PUSH  HL          ;Save for later
00300           CALL  0A7FH       ;Get delay loop count
00310           PUSH  HL          ;Save for later
00320           LD    HL,(DISP)   ;Get top of screen delay
00330           PUSH  HL          ;Save for later
00340           LD    DE,SCREEN   ;Clear video memory
00350           LD    HL,BSCREN
00360           LD    BC,COUNT
00370           LDIR
00380           LD    DE,SCREEN   ;Send stimulus from
00390           LD    HL,VSCREN   ; display page to video
00400           LD    BC,COUNT    ; memory
00410           LDIR
00420 SYNC1     IN    A,(PORT)    ;Wait for vertical sync
00430           AND   1           ;Test bit 1
00440           JR    Z,SYNC1     ;JP if not set
00450           POP   DE          ;Retrieve delay count
00460 LOOP      DEC   DE          ;Delay for beam to reach
00470           LD    A,E         ; top of screen
00480           OR    D
00490           JR    NZ,LOOP
00500 TIME1     LD    HL,CLK1     ;Start time storage adr
00510           LD    B,37        ;Transfer time digits
00520           LD    C,B         ; in units of tens of
00530           INI               ; seconds down thru
00540           LD    C,B         ; milliseconds
00550           INI
00560           LD    C,B
00570           INI
00580           LD    C,B
00590           INI
00600           LD    C,B
00610           INI
00620           LD    HL,0001H    ;For use below
00630           POP   DE          ;Get display count
00640           LD    A,H
00650           OUT   (PORT),A    ;Turn CRT back on
00660 STIN      IN    A,(PORT)    ;Wait for vert sync
00670           AND   L
00680           JR    Z,STIN      ;Continue till set
00690           LD    B,75
00700 WAIT      NOP               ;Wait for end of sync
00710           DJNZ  WAIT        ; pulse
00720           DEC   E           ;Countdown no. of pulses
00730           JR    NZ,STIN     ;JP if not done
00740 CLEAR     LD    A,L
00750           OUT   (PORT),A    ;Blank the CRT
00760           LD    A,H
00770           POP   DE          ;Get bad char.
00780 INP       ADD   A,(IY)      ;Check keyboard
00790           JR    Z,INP       ;JP if done
00800           CP    E           ;Bad character?
00810           JR    NZ,TIME2    ;JP if no
00820           LD    A,H
00830           JP    INP
00840 TIME2     LD    HL,CLK2     ;Stop time storage adr
00850           LD    B,37        ;Transfer time as above
00860           LD    C,B
00870           INI
00880           LD    C,B
00890           INI
00900           LD    C,B
00910           INI
00920           LD    C,B
00930           INI
00940           LD    C,B
00950           INI
00960           LD    DE,SCREEN   ;Clear video memory
00970           LD    HL,BSCREN
00980           LD    BC,COUNT
00990           LDIR
01000           LD    L,A         ;Save response
01010           LD    H,0
01020           LD    A,0
01030           OUT   (PORT),A    ;Turn on CRT
01040           EI                ;Enable interrupts
01050           JP    0A9AH       ;Return to BASIC
01060 ;
01070 ;
01080 CLK1      DEFB  0           ;Start times - 10X sec
01090           DEFB  0           ; Sec
01100           DEFB  0           ; 1/10 Sec
01110           DEFB  0           ; 1/100 Sec
01120           DEFB  0           ; 1/1000 SEC
01130 CLK2      DEFB  0           ;End times - 10X sec
01140           DEFB  0           ; Sec
01150           DEFB  0           ; 1/10 Sec
01160           DEFB  0           ; 1/100 Sec
01170           DEFB  0           ; 1/1000 Sec
01180 CHAR      DEFB  0           ;Char code not allowed
01190 DISP      DEFW  203         ;Delay to top of screen
01200           END
```

(b)

```
10 FOR  I%=&HF650 TO &HF6F7
20 READ J%: POKE I%,J%: NEXT I%
30 STOP
100 DATA  243, 253, 33, 16, 56, 62, 1, 211, 250, 42
110 DATA  246, 246, 38, 0, 229, 205, 127, 10, 229, 42
120 DATA  247, 246, 229, 17, 0, 60, 33, 255, 247, 1
130 DATA  0, 4, 237, 176, 17, 0, 60, 33, 255, 251
140 DATA  1, 0, 4, 237, 176, 219, 250, 230, 1, 40
150 DATA  250, 209, 27, 123, 178, 32, 251, 33, 236, 246
160 DATA  6, 37, 72, 237, 162, 72, 237, 162, 72, 237
170 DATA  162, 72, 237, 162, 72, 237, 162, 33, 1, 0
180 DATA  209, 124, 211, 250, 219, 250, 165, 40, 251, 6
190 DATA  75, 0, 16, 253, 29, 32, 243, 125, 211, 250
200 DATA  124, 209, 253, 134, 0, 40, 251, 187, 32, 4
210 DATA  124, 195, 182, 246, 33, 241, 246, 6, 37, 72
220 DATA  237, 162, 72, 237, 162, 72, 237, 162, 72, 237
230 DATA  162, 72, 237, 162, 17, 0, 60, 33, 255, 247
240 DATA  1, 0, 4, 237, 176, 111, 38, 0, 62, 0
250 DATA  211, 250, 251, 195, 154, 10, 0, 0, 0, 0
260 DATA  0, 0, 0, 0, 0, 0, 0, 203
```

Figure 3. (a) Assembly language tachistoscopic display and reaction time collection routine; (b) BASIC program to POKE this routine into memory.

ing routine. Line 80 establishes the origin of the program at F720, which may be changed by the user. Lines 90 and 100 set the addresses of the 1,024 blanks and the display page, respectively, both of which may also be changed. Line 110 sets the count of 1,024 bytes to be transferred. Lines 120-150 send the blanks to the display page, overwriting any previous stimulus. Control returns to BASIC at Line 160. A program for POKEing the routine into memory from BASIC appears in Figure 2b.

The tachistoscopic display and reaction time collection routine is listed in Figure 3a. Line 160 sets the starting address to F650, which may be changed as desired. Lines 170-220 set a number of constants required by the program, the first two of which may be changed to alter display page locations, as noted above. Port 250 is used for video synchronization. Information input from this port indicates the status of the vertical synch pulse: When the first bit is one, a synch pulse is occurring; when the first bit is zero, the system is between pulses. Sending a one to this port prevents information from being transferred from video memory to the CRT,

effectively blanking the screen; sending a zero to the port restores the information transfer.[5]

Following the subroutine entry from BASIC at Line 230, interrupts are disabled to insure accurate timing and the CRT is blanked. The locked-out keyboard character (described below), the number of 16.7-msec intervals for stimulus display, and the value for the top-of-screen delay loop (described below) are obtained and placed on the stack in Lines 270-330. Video memory is cleared and the stimulus is sent from the display page to video memory in Lines 340-410. Lines 420-440 wait for the next vertical snych pulse to occur. Upon its detection, a delay is begun, followed by input and storage of the onset time of the reaction time interval, which coincides with CRT beam positioning at the upper left corner of the CRT. The delay loop is necessary because the synch pulse occurs when the electron beam is at the lower right corner of the CRT and time must be provided to allow the beam to trace back to the upper left corner.[6] Once the delay has been completed and the time has been input, the communication between video memory and the CRT is restored and the first display frame of the stimulus begins. Although the time input lags the information transfer restoration by approximately 100 microsec, this was not seen as critical. Also, unless the stimulus begins on the first line of the CRT, there will be an additional lag. The lag is always constant and may be computed and subtracted from the final reaction time, if desired.[7]

Following the stimulus onset, the loop in Lines 660-730 detects each vertical synch pulse and counts down until the desired number of pulses have occurred, at which point the CRT is again blanked, in Lines 740-750. Note that a short delay loop of about 325 microsec is required immediately following synch pulse detection (Lines 690-710). This loop insures that the same synch pulse is not detected more than once, since the duration of the pulse is many times longer than the few microseconds required to decrement the pulse counter and wait for another pulse.

Following CRT blanking, the subroutine waits for the subject's response, on Lines 760-830.[8] The response is detected using the method described by Owings and Fiedler (1979). Responses are detected from one of three input switches, which are connected electrically in parallel with the "1," "2," and "3" keys on the keyboard. One input may also be declared as illegal, and the routine will not repsond should this input occur. The locked-out character code, in this case eight for the "3" key, is POKEd from the BASIC program to the location CHAR in Line 1180.

Once a legal response has occurred, the time of the occurrence is obtained and stored in Lines 840-950. Following this, video memory is cleared, the response is saved, communication between the video memory and the CRT is restored, and interrupts are enabled in Lines 960-1040. Control returns to BASIC on Line 1050.

The final lines of the program allocate storage space for a number of variables. Lines 1080-1120 allocate space for the five clock inputs required to determine the time in milliseconds at the start of the stimulus display; Lines 1130-1170 allocate space for the clock inputs made following the subject's response. Line 1180 allocates space for the locked-out character code, and Line 1190 allocates space for the top-of-screen delay loop value. Figure 3b contains a program that will POKE this routine into memory from BASIC.

The BASIC program listed in Figure 4 is a sample application of the tachistoscope system. It is assumed that the routines in Figures 2 and 3 have been assembled using the Radio Shack Editor/Assembler or equivalent assembler and have been loaded into memory or have been POKEd into memory by a BASIC program. The sample program displays the stimulus, in this case a large "T" composed of small "T"s, placed at the center of the screen, waits for the subject to respond, and records the reaction time. Lines 30-40 set the USR subroutine addresses for the tachistoscopic routine and display page blanking routine, respectively. These values must be the same as those in the ORG statements of the two programs. Line 50 sets the address for the time data retrieval and must be equal to a value one less than the address of the label CLK1 in Figure 3a. Line 60 is an offset that is added to a video memory address between 3C00 and 3FFF to produce a corresponding address on the display page beginning at FBFF. This value must be changed should the display page location be changed. Line 70 sets the "3" button as an illegal response to the stimulus by POKEing the value eight to the location addressed by the label CHAR in Figure 3a. Line 80 loads the region beginning at F7FF with 1,024 blanks. The value of G must be changed if this region is relocated. Lines 90-100 initialize an array containing the video memory addresses for the stimulus. Line 110 inputs the desired number of 16.7-msec intervals for

```
10 DIM T(10)
20 DEFINT G-S
30 DEFUSR0=&HF650
40 DEFUSR1=&HF720
50 RA=&HF6EB
60 SO=&HBFFF
70 POKE &HF6F6,8
80 G=&HF7FF:FOR I=G TO G+1023:POKE I,32:NEXTI
90 FOR I=1 TO 9:READ IS(I):NEXTI
100 DATA 15707,15709,15711,15713,15715,15775,15839,15903,15967
110 CLS:INPUT"DISPLAY THE STIMULUS FOR HOW MANY 16.7 MS INTERVALS";ST
120 X=USR1(0)
130 FOR I=1 TO 9:POKE IS(I)+SO%,84:NEXT I
140 CLS:PRINT@448,CHR$(23);"WHEN READY, PRESS THE NUMBER 3"
150 IF INKEY$="3" THEN CLS ELSE150
160 IX=USR0(ST)
170 FOR M=1 TO 10
180 T(M)=PEEK(RA+M)
190 T(M)=T(M)-INT(T(M)/16)*16
200 NEXT M
210 TIME=T(10)+10*(T(9)+10*(T(8)+10*(T(7)+10*T(6))))
220 TIME=TIME-(T(5)+10*(T(4)+10*(T(3)+10*(T(2)+10*T(1)))))
230 IF TIME<0 THEN TIME=TIME+60000
240 PRINT"REACTION TIME WAS";TIME,"BUTTON PRESSED WAS";IX/2
250 INPUT"CONTINUE (Y OR N)";A$
260 IF A$="Y" THEN 110 ELSE STOP
270 END
```

Figure 4. Sample BASIC program used to display a stimulus and compute reaction time.

stimulus display. Line 120 blanks the display page, and Line 130 sends the stimulus to the display page. Lines 140 and 150 prompt the subject to begin the trial. Line 160 calls the tachistoscopic routine and waits for a response. Lines 170-240 retrieve the clock values, compute the reaction time, and print the reaction time and button pressed.

APPLICATION

The system has been used in various stages of development for the past 2 years and has functioned well. Stimuli presented have included patterns composed of graphics characters, large letters composed of either letters from the internal character set, such as that in the example, or graphics characters, and sentences and letters generated using the internal character sets. Stimuli have been presented at various locations on the CRT, including locations that lateralized presentation to the left or right visual fields. Additional information regarding implementation of this system is available from the author.[9]

REFERENCES

CAVANAGH, P., & ANSTIS, S. M. Visual psychophysics on the Apple II: Getting started. *Behavior Research Methods & Instrumentation*, 1980, 12, 614-626.

GOLDBAND, S. Real-time clocks for microcomputers in behavioral research. *Behavior Research Methods & Instrumentation*, 1979, 11, 111-114.

GRICE, G. R. Accurate reaction time research with the TRS-80 microcomputer. *Behavior Research Methods & Instrumentation*, 1981, 13, 674-676.

LINCOLN, C. E., & LANE, D. M. Reaction time measurement errors resulting from the use of CRT displays. *Behavior Research Methods & Instrumentation*, 1980, 12, 55-57.

OWINGS, R. A., & FIEDLER, C. H. Measuring reaction time with millisecond accuracy using the TRS-80 microcomputer. *Behavior Research Methods & Instrumentation*, 1979, 11, 589-591.

PERERA, T. B. A versatile microcomputer-based multiple-field tachistoscope. *Behavior Research Methods & Instrumentation*, 1978, 10, 546-547.

PERERA, T. B. A comparison of BASIC language timing loops for the TRS-80 microcomputer. *Behavior Research Methods & Instrumentation*, 1979, 11, 592.

POST, T. Software control of reaction time studies. *Behavior Research Methods & Instrumentation*, 1979, 11, 208-211.

REED, A. V. Microcomputer display timing: Problems and solutions. *Behavior Research Methods & Instrumentation*, 1979, 11, 572-576.

NOTES

1. Approximate cost of the Mini 8100 and clock is $600. The clock was manufactured by Mountain Computers, Inc., 300 Harvey West Boulevard, Santa Cruz, California 95060. The Mini 8100 was manufactured by California Computer Systems, 250 Caribbean Drive, Sunnyvale, California 94085. Alternative methods for reaction timing for the TRS-80 Model I are detailed by Grice (1981), Lincoln and Lane (1980), and Owings and Fiedler (1979). General approaches to reaction timing with microcomputers have been presented by Goldband (1979) and Post (1979).

2. For example, suppose interval timing up to 9.999 sec in millisecond accuracy is desired. At a given point, one would need to read the seconds, hundreds of milliseconds, tens of milliseconds, and milliseconds ports. Assuming that values read from these ports are, respectively, 2, 1, 5, and 3, the time is obtained by computing 2(1,000) + 1(100) + 5(10) + 3 = 2,153 msec.

3. It later came to the attention of the author that assembly language multiplication routines were available in the BASIC read-only memory area. However, because the system was already developed and working successfully, it was decided to leave it as presented here.

4. All memory addresses are in hexadecimal.

5. The author is indebted to Frank M. Borowiec, who provided the design and implementation of hardware modifications necessary for synchronization. Modifications were based upon those offered by Grice (1981) and Reed (1979). Details of the circuitry are available from the author.

6. Determination of the delay loop value was done using a separate program. Details on loop calibration to provide CRT onset at screen locations other than the upper left corner are available from the author.

7. The CRT on the TRS-80 Model I displays 16 lines of information, each of which takes .758 msec to complete (Grice, 1981). Assuming that information transfer between video memory and the CRT begins at Line C, and a stimulus is complete following the completion of Line L, then the lag between the beginning of the reaction time interval and complete stimulus onset will be .758(L – C + 1) msec. This is a constant value that may be subtracted from the final reaction time computation.

8. Because the screen does not become totally blank until the beam has retraced through the last point of the stimulus, the user may wish to insert a small delay loop equal to 4.458 + .758(L) msec, with L defined as in Footnote 7. The loop should follow the CRT blanking command in Line 750 and will prevent the detection of any response that occurs prior to complete erasure of the stimulus.

9. Copies of all programs on a disk that is compatible with both TRSDOS and NEWDOS 80 for the TRS-80 Model I are available from the author.

(Received for publication June 1, 1982; revision accepted October 25, 1982.)