DECUS
PROGRAM LIBRARY

DECUS NO.	8-396
TITLE	MTS - 6/70 (MILLISECOND TIME-SHARING SYSTEM)
AUTHOR	Charles W. Snyder
COMPANY	Department of Psychology University of Notre Dame Notre Dame, Indiana
DATE	March 24, 1971
SOURCE LANGUAGE	PAL III

Although this program has been tested by the contributor, no warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related program material, and no responsibility is assumed by these parties in connection therewith.

MTS - 6/70 (MILLISECOND TIME-SHARING SYSTEM)

DECUS Program Library Write-up | DECUS NO. 8-396

A simple millisecond time-sharing program is described, along with input/output and control routines and a modification of experiment program coding in PAL III assembler language that allows simultaneous on-line data reduction and control of several experiments with no interaction.

Our present millisecond time-sharing system (MTS-6/70) began two years ago when an experimenter wished to print the data of one trial without missing any data inputs for the next trial. It was soon realized that computer time could be fully utilized up to the 1000 milliseconds in each second for many independent experiments at the same time. The primary limitations are size of core memory and number of output devices although experiments can be changed without interruption of others and experimental data can be sorted after cards are punched on another computer. An English interactive oscilloscope display and keyboard control are among the most useful features. The basic system of about 1400 core words also includes arithmetic, conversion, and service routines most commonly required in behavioral research.

MTS Schedule Control. The heart of the time-sharing system is a very simple subroutine of about 20 instructions (CONTRL), beginning at address Ø2ØØ with RC clock initialization. The experiment program indexing and timing are controlled by a multi-level decimal system that turns control of the computer for one millisecond at rates of 1, 10, 100 or 1000 times per second.

The control indexes require no counters since they are self-resetting. That is, when each index reaches its tenth position, it calls a brief subroutine that resets itself and loads into its variable, ninth position the next program address at the next slower level.

The remainder of this memory page is used for the few input routines (such as ADC) that may operate at 1000 Hz and for the 3 tables of program starting addresses (10 each) that are called at 100, 10, or 1 Hz. These tables consist of absolute starting addresses which are called indirectly through the indexing control program. In fact, the programs are called by a double indirect method for all schedules slower than 100 Hz. The illegal double indirect is avoided by having CONTRL call an indirect JMS to only the 100 Hz programs. After each millisecond share, the indirect JMS instruction itself is incremented. When it reaches the ninth position, one program from the 10 Hz level is already in that position from a preceding 100-Hz reset instruction. The 100-Hz index is reset after its tenth position and goes to the first program without a clock wait.

The 10-Hz and 1-Hz levels operate identically and any number of slower levels, such as 10-sec. or 60-sec. rates, could be added. In the present system, 1 Hz is the slowest schedule, so there is no variable program from a lower level at its ninth position. Instead this position is used to operate a 24-hour time-of-day clock (RELCLK).

<u>Real-Time Clock.</u> A 24-hour time-of-day clock can be displayed and

set from keyboard controls. Hour, minute, colon-space, and second can also be read in trimmed-packed ASCII (8 characters in 4 words) by calling JMS I READTI with the initial storage address-1 in the accumulator.

All message, keyboard, printer, and punch buffers store trimmed-packed ASCII with a ØØ end code identical to DEC MESAGE and MACRO-8 TEXT. (The % character is a special code, 45, for carriage return and the # character is a special code, 43, for line feed.)

Scope Display. The character display program is a modification of Digital's DSCPAK, limiting each message to one line of teletype characters (18 maximum) and of one size only, thereby eliminating all special characters. Its speed has been approximately doubled (to one full 4X6-dot character per millisecond) so that one scheduled rate of 100 Hz is satisfactory, although any unused 100-Hz schedules may be added to increase brightness and reduce flicker.

Up to 8 lines may be displayed by depositing the initial address of each message in MESAGØ to MESAG7 of Page Ø. MESAGØ is ordinarily reserved for time-of-day display, MESAG5 for the second core examine display (during debugging) or for programmed ID requests (during experiments), MESAG6 for keyboard buffer and echo (and for first core examine or change), and MESAG7 is reserved for keyboard command (CTRL) displays.

ASCII codes 334, 337, and 241-251 ("!" to ")") cannot be displayed since these locations in the character table are taken by program instructions.

Inputs/Outputs on Interrupt. All output data is printed or punched and program loading is carried out under peripheral control of the interrupt rather than under scheduled (synchronous) control. In an earlier system, it was discovered that the high-speed reader does not operate properly even at 100 Hz (one-third of maximum) and the teletype will not operate up to 10 Hz at the end of a long line or during carriage return. The solution to these problems yielded more efficient use of free moments of computer time and freed schedule index positions for experiment programs.

In the MTS control program, the next scheduled program is not called until the 1000-Hz RC clock pulse occurs. During this wait, the computer interrupt is enabled and any peripheral I/O device may interrupt the clock-wait loop. During an interrupt, the computer takes an automatic JMS to location ØØØØ, checks I/O flags to determine which device caused the interrupt, and jumps to the appropriate program to service the peripheral device.

If a device causes an interrupt indicating it is ready to proceed, but no operation is required of it, the service routine simply clears the flag of that device. This means that these inputs/outputs must be initiated by separate subroutines that start the device (in addition to resetting buffer indexes, printing carriage return, etc.) in order to get the interrupt flag operating again for that device.

The interrupt system also yielded a simple and efficient means of leaving scheduled positions blank when not in use. The scheduled position is merely cleared to ØØØØ at any unneeded location. Then when this point is reached in the control schedule, the computer goes to location ØØØØ, checks the I/O flags once with interrupt disabled and then returns to CONCLK to wait for the next clock pulse with the interrupt on.

The input/output subroutines are more or less standard except there are several modifications to adapt these to on-line experiment time-sharing. Most important is the teletype line buffer which allows one line of output from only one experiment (with ID number) at a time. This prevents data from different experiments from coming out mixed.

When an experiment trial is finished and data are ready to be calculated and printed/punched, the initial location of a subroutine (TYBREQ, teletype buffer request), must be checked after TAD I TYBUFI to see if the teletype line buffer is busy. If it is busy with another experiment, its initial location is non-zero (it contains the return address of the calling experiment). If it is not busy, the initial location is zero (cleared at the end of line output) and this same subroutine can be called (JMS I TYBUFI) to initialize the teletype flag with a carriage return, initialize the buffer output with a line-feed, and to set auto-index 14 to load the teletype buffer with data. The line of data into the buffer must be terminated by ØØ end code (DCA I 14). Teletype lines are double-spaced by adding an extra line feed (TAD T43ØØ; DCA I 14). So far, all necessary calculations, including simple division and multiplication (single precision integer) and binary-to-decimal conversions (single or double precision without decimal point), have been carried out during the actual teletype line output, since the buffer loading stays ahead of output if the data calculations are done at 10 Hz or faster.

However, the high-speed punch is fast enough to "catch up" with its buffer loading at 10 Hz. But so far, the complex experiments that use the high-speed punch for data output also required separate output programs that were enabled from the main experiment but run on separate MTS schedules at 100 Hz. HPBUFI corresponds to TYBUFI for busy test and initializing punch and separate buffer output but does not set any auto-index for buffer loading.

A third device operating on interrupt is the high-speed tape reader. Presently, this is used only for loading new experiment programs from keyboard command, although any other tapes in BIN format could be read in for combining old data with on-line data from the same subject or for controlling an experiment under an extremely long series of conditions, such as a random sequence of time intervals. (A BIN tape of any data or information can be made by using the PAL III editor and assembler.)

[illegible] modification is made to the standard BIN LOADER program, [illegible] the last page of memory, for loading new programs during uninterrupted MTS operation. When the subroutine, HSRBIN, is initiated by CTRL B from the keyboard (or programmed by depositing the CTRL B code in location KBCTRL in Page Ø), the high-speed reader flag check in BIN is changed to return to MTS control and the final TAD CHKSUM instruction preceding HLT in BIN is changed to JMP 76ØØ. At this location a special BIN termination routine begins, taking unused locations preceding BIN. This routine checks for a tape reader checksum error and issues a scope message of "TAPE IN"; or "ERROR; RELOAD" if an error has occurred. The HLT in original BIN is thus bypassed and no halt of the computer occurs. After the reader and BIN are initiated, the interrupt flag check jumps directly to the read-fetch instruction in BIN each time the reader is ready and the interrupt is enabled. The original BIN is restored at the end of the special termination routine so that the BIN LOADER can be operated normally.

The final interrupt program in the present system is the keyboard input. The keyboard buffer routine includes an initial control character which is displayed on the bottom line and deposited in location KBCTRL to give commands for program or experiment control (as user programmed), or to tell the computer what to do with the following keyboard information. The subsequent keyboard line, up to 18 characters, is echoed only on the scope display, leaving the teletype printer and punch completely free for experiment data outputs.

A variety of keyboard instructions in English and/or octal numbers may be added to meet the demands of all or particular experiments. Some routines are used by every experiment, e.g. to enter or change the ID number. In addition, a standard routine is included for starting or stopping particular experiments by name (which must be 6 characters). This [illegible] routine, EXPALT, alters the control schedule for a certain experiment. A keyboard CTRL I initiates a subroutine to search for the experimental "name" in a table of name codes, schedule addresses, and starting addresses. When the name is found, its starting address is deposited in its schedule address and the experiment program routinely starts with a displayed request for ID keyboard entry. If the name is not found, the message "NOT LOADED" is displayed. When an experiment program tape is loaded, 3 locations in the experiment table (starting at EXPTBL) are filled in: the name code (=octal sum of the first 2 core words of the experiment name, i.e. the first 4 characters in trimmed code); the MTS schedule address; and the experiment's starting address. The 1-Hz schedules are addresses Ø31X, 1Ø-Hz schedules are Ø33X, and 1ØØ-Hz schedules are Ø35X. Since some of the service routines use these schedules, refer to the listing or use only those containing ØØØØ.

A CTRL O along with keyboard entry of its name will clear the control address to turn off the schedule for that experiment and also automatically reinitializes the experiment program itself to begin with the ID request. The experiment program remains in core memory, of course, ready to be returned to operation until another experiment is loaded over it and take its place in the experiment table. An experiment's schedule or schedules may still be running when replaced by another experiment. To avoid halting the computer, each experiment tape must begin by clearing all schedules that it uses.

Since every experiment requires an ID number which is printed/punched at the beginning of each data line output (and each card subsequently punched), a convention in ID entry programming has been adopted. Rather than using a separate control schedule for ID entry, which must be available and yet must be disabled after the entry, stage programming is used which requires only one starting address for each experiment (see later section).

Input Conversions. Except for the initial command character following a carriage return, keyboard entries are immediately converted to trimmed/ packed ASCII code before storing and displaying in its buffer. Therefore after line termination, the line may be transferred if desired by setting an index to its initial address, KBUFL, found in Page Ø at KBUFA.

However, if a keyboard entry of an experiment ID is made with 8 characters ending in a 2-digit initial trial number, a service routine (IDTRAN) may be called by JMS I IDTRI after setting the accumulator to the ID storage address-1. The location preceding the actual ID storage must be free to accept a decimal divider which is set by the subroutine. This is necessary because trial numbers are incremented up to a possible 99 and the trial number stored in trimmed code (2 characters per core word) cannot be simply incremented over Ø9. At the end of each trial, another service routine is called by JMS I TRLNI which increments the first place trial number every time except every tenth time when it resets the first place, increments the second place, and resets the divider. This JMS is also entered with the accumulator containing the ID storage address-1.

When the service routine for analog-to-digital conversions (ADFOUR) is called by JMS I ADFORI, the entering accumulator must contain the initial multiplex channel number (in octal, from Ø to 14, or 34 for AXØ8 potentiometers. The resulting values for the four channels are stored in locations Ø27Ø plus channel number. This ADC buffer has space for 16 channels and the indirect pointer (initial ADC buffer address) must be placed on the experiment page since it is not available from Page Ø. ADFOUR requires 0.12 msec. so it is usually desirable to convert a single channel with a separate routine. The ADFOUR routine is required primarily when channels Ø to 3 must be converted at 1000 Hz in the shortest possible time, in which case the routine must be enabled by replacing NOP in address CONKHZ (Ø21Ø) with JMS ADFOUR (4246).

The 3 trigger inputs and 8 digital inputs (with XR option of the LAB-8) are ordinarily carried out and tested for branching in the experiment program. However, a service routine CONTAC is available for simulating the action of electromechanical counters (SODECO) operating from electromechanical pulse-formers. Normally this routine runs continuously on a separate 100-Hz schedule (0351) from relay or pulse-former contact inputs (filtered) to CØ, C1, C2,and C3 "contingency" inputs to the AX-Ø8 Laboratory Peripheral. The routine increments a software flag or counter in Page Ø (CØFLAG, C1FLAG, C2FLAG, or C3FLAG) only after the input has been on for 20 msec. after having been off for at least 30 msec. The software flag must be cleared by a branch in an experiment program after each input or after counts have been allowed to accumulate. The resulting count of responses in this way agrees with the external SODECO counter.

Output Conversions. Experimental variables are stored in binary form and then converted to decimal and loaded into the teletype output buffer one line (trial) at a time. Since most experimenters request varying number of digits and spaces between columns of data and have no requirements for negative numbers or decimal points, the DEC single-precision (DECPRT) and double-precision (UDPRNT) programs were so modified. The result was a flexible compromise between readable printing and conservation of space (ultimately on punch cards).

Three modifications convert binary data in the accumulator to 2, 3, or 4 digits plus following space by calling JMS I DECP2I, DECP3I, or DECP4I. The digits are immediately trimmed and packed and deposited through auto-index 14. This index must previously be initialized to the teletype buffer by a JMS I TYBUFI. However, the very first thing before beginning an output line is to test to see if the teletype is already busy with TAD I TYBUFI; SZA CLA. The skip will occur if the teletype is free since the entering location of buffer request subroutine is cleared at the end of each line output.

In rare cases, a decimal scope display may be needed, in which case the TAD I TYBUFI; SZA CLA test must still be made followed by setting auto-index 14 to initial address-1 of an experiment display buffer, TAD data into accumulator, JMS I DECP4I (e.g.), and finally displaying the decimal result by depositing the display initial address (not -1) in MESAG1, MESAG2, MESAG3, or MESAG4 (cooperatively shared by other experiments).

The modified double-precision decimal conversion is called by JMS I UDPRTI with the accumulator containing the negative octal number of digits desired (from 1 to 7). If the accumulator is clear, 4 digits will be output. As usual, the location following the JMS must contain the address of the high-order data (not the data itself). The actual high-order data is somewhere else immediately followed by the low-order data. This conversion also adds a space after each score and loads trimmed/packed code through auto-index 14.

The high-speed punch may also use these conversions when the teletype is not. If TAD I TYBUFI is zero, teletype line outputs can be held up by a ISZ I TYBUFI and freed at the end of high-speed buffer loading by DCA I TYBUFI with accumulator clear. Auto-index 14 must be set to HPBUFA (Page Ø) before calling the conversion subroutine. The high-speed punch and buffer output are initiated with a carriage return and line-feed by calling JMS I HPBUFI. As in all buffers, the end code is ØØ (half or full word) obtained in these cases by a DCA I 14 with the accumulator clear at the end of buffer loading.

The capacity of both output line buffers is 72 characters.

Arithmetic Subroutines. Single-precision integer division and multiplication are carried out by programs identical to DEC-Ø8-FMCB and DEC-Ø8-FMBA. DIVIDE is called by JMS I DIVIDI with the accumulator containing the high-order dividend, the location following the JMS containing the low-order dividend, and the next location containing the divisor. All three of these are data, not addresses. Return is to the location following the divisor with the quotient in the accumulator.

MULT is called by JMS I MULTI with the multiplier in the accumulator and the location following the JMS containing the address of the multiplicand. Return is to the next location with the high-order product in the accumulator and the low-order product in MP1, obtained by TAD I MPI.

All indirect pointers (or links) to these addresses are in Page Ø.

Octal conversion is also available (OCTCON) but has never been used for output of experimental data.

Alphanumeric messages are stored in trimmed-packed ASCII form so they can be loaded directly into an output buffer (via auto-index 14)using a service routine (TRANSF) which transfers all full words up to but not including a full-word ØØØØ end code (loading either a ØØ half-word or full-word zero would prematurely terminate the output line). This subroutine is called by JMS I TRANSI with the initial data address-1 in the accumulator. Character output to the teletype or punch buffers ordinarily involves the experiment ID at the beginning of each line (which also uniquely identifies each card if punched). It is expedient to insert a double space (4Ø4Ø) code as a permanent addition to the stored ID.

Scope messages are also stored in trimmed and packed ASCII code with either ØØ or ØØØØ end code. A message is displayed by depositing the initial address of it directly into MESAGØ to 5 (Page Ø). There is no priority system of display and only up to eight may be in effect at a moment so a high-priority message must be repeatedly displayed or must reserve a line to override others. A message will be displayed until the address in MESAGØ to 7 is changed to another. Any change of MESAGØ will override the time-of-day display which may be returned (with no loss in time) by a keyboard command CTRL T. A change of MESAG6 will temporarily override the keyboard line display until the next key is pressed. MESAG7 is entirely reserved for keyboard command CTRL displays and is cleared by each carriage return from keyboard.

There are usually several stages or sequences of conditions in an experiment involving not only keyboard entry of ID number but also changes in trial inputs, data calculation, data output, etc. This programming convention is described in detail in the next section.

Stage Programming. In a time-sharing system, obviously the computer cannot loop for many seconds waiting for an ID keyboard entry. For that matter, it cannot actually wait for anything, such as a rat to jump a hurdle or a human to show a skin resistance response. It can only "wait" (with interrupt on) after one program turn is finished in less than one millisecond and before the next clock pulse occurs. It would be quite inefficient to use a separate control schedule for each phase in an experiment; a separate set of instructions would be necessary to end each phase and begin the next. Fortunately, a simple solution to this problem also allows segmentation of old programs to fit into the time-sharing system.

Regardless of whether the phases of an experiment are to be scheduled, enabled by inputs, reversed, or even placed in random order, they can all be combined on one schedule and divided into sections running less than one millisecond by a simple programming device in each experiment program. The method is somewhat difficult to symbolize since each time a scheduled program is entered, it goes to the "exit" to determine where to "enter" this time. But the instructions consist of the most powerful set that have been encountered yet in the PDP-8. They do not require the indexing of program pointers or, even more inefficiently, the storing and transferring changes in entrance instructions. An experiment program for MTS must limit itself to less than one millisecond (since it is not run on clock interrupt) by a simple segmentation with the following instructions (the address-tag symbols are illustrative only):

```
ENTER,   Ø               /STARTING ADDRESS OF PROGRAM CALLED BY MTS
         JMP I EXIT
EXIT,    ID              /INITIAL STAGE, CHANGED BY EACH JMS EXIT
         JMP I ENTER     /RETURN TO MTS CONTROL
ID,      .               /DISPLAY REQUEST FOR ID ENTRY
         .
         .
         .
         .
         JMP I ENTER     /RETURN TO MTS WITHOUT CHANGING STAGE
         .
         .
         .
         .               /THIS STAGE FINISHED
         JMS EXIT        /CHANGE STAGE AND RETURN TO MTS
NEXT,    .               /REENTERED HERE, BYPASSING PRECEDING STAGES
         .
         .
         .
         (ETC.)
```

It is important to note that a stage may be passed through only once, such as the ID request or a transfer of ID from keyboard buffer to experiment storage if the stage includes no JMP I ENTER instruction. When such a stage is finished by a JMS EXIT instruction, the address of the immediately following location ("NEXT," above) is deposited in location EXIT before the program returns to MTS by taking the JMP I ENTER of the fourth line (ENTER contains the MTS return address since MTS called it with a JMS to that location initially). It can now be understood how the service routine for command CTRL 0 reinitializes the experiment program for restarting at the initial stage (ID) when it turns the experiment off. The CTRL 0 routine simply contains instructions to deposit the address of the fifth location ("ID" above) in the third location ("EXIT") of any experiment it is turning off.

It is also important to note that a main stage of an experiment may be reentered at the same point many times, e.g., for input sampling, for repeated multiplication or division, or for repeated decimal conversions for output. There is time for only one multiplication, division, or decimal conversion per one-millisecond turn. In these cases, it is necessary to return to MTS control with the stage entrance unchanged. This is done by a JMP I ENTER instruction, which leaves the next reentering address contained in location EXIT unchanged. Since repeated calculations and output conversions are typically programmed in a loop which is repeated until the desired count is reached, it is necessary only to have a JMS EXIT just before the beginning of such a loop and to insert a JMP I ENTER instruction somewhere within the loop, where the contents of the accumulator and link are cleared. The accumulator is shared with all other programs and its contents are not saved by MTS. If necessary, the experiment program can save its accumulator and/or link before it self-interrupts. (The usual programming convention is followed in all MTS Coding: the accumulator is always left clear when not in use and the link is always cleared just before use. Following this convention, it is not necessary to clear the accumulator before each use. As an additional safeguard, each pass through MTS control leaves the accumulator and link clear.)

There have been cases of more complex program stage control, as when a program at one schedule rate must enable or disable another program operating on a different schedule. In most cases, the cross-schedule control is simply a matter of depositing a stored instruction or address in the other program (usually by on-page indirect addressing, since Page Ø space for pointers is very scarce). But it is also quite feasible to have a jump into and exit through the other program since all scheduled programs return to MTS control at a single location following the CONTRL JMS (namely, CONTIM). For example, the most efficient way for one program to change the stage of another is for it to return to MTS via a jump to a JMS EXIT instruction just preceding that stage in the second program. In a similar manner, one program can regress or reinitialize itself to an earlier stage by returning to MTS via a jump to a JMS EXIT preceding that stage. Another useful program control device has been the direct jump out of a subroutine. A subroutine is normally exited by a JMP I SUB which returns to the instruction following the JMS call, but it is sometimes very useful to jump to another sequence upon certain conditions within the subroutine. This could be confusing, however, if the programmer forgets that a return from such a JMS may not occur as usually assumed.

Assembly. Experiment programs coded in PAL III symbols may include parameter definitions of the MTS symbols for Page Ø constants and addresses used. Or the ASCII tape of PAL III PERMANENT SYMBOL TABLE MODIFICATION FOR MTS-6/7Ø may be assembled as the first tape of Pass 1. This tape of "EXPUNGE.........FIXTAB; PAUSE" includes the basic PDP-8/I memory and operate instructions, the IOT instructions for this system including those for the AX-Ø8 Laboratory Peripheral, along with all of the MTS constants and addresses in Page Ø. Use of this symbol tape eliminates the parameter assignments for the MTS symbols in each experiment program and also excludes these in the program symbol table and listing during assembly.

Program Editing and Debugging. The process of monitoring and debugging new programs is vastly facilitated by an actual display of address and contents of core memory along with octal keyboard control of both. The dynamic monitoring of variables during program operation is also valuable in checking and calibrating inputs and external trigger and analog devices. With a keyboard CTRL E (for examine) and keyboard entry of the core address in octal, the scope displays both address and contents. The displayed contents change when indexes or variables are monitored.

To change a core content, a keyboard CTRL C (for change core contents) is followed by keyboard entry of the desired core contents, including leading zeroes. A CTRL A will advance and display the address and present contents in octal, and the display of the preceding address and contents will move to the line above. With these three keyboard controls, debugging errors are usually quickly located and corrected since the effect of any changes can be monitored during uninterrupted operation. It also becomes quite feasible to program short routines directly in machine language from keyboard without the frustration of the switch register, to load variables in checkout routines, and to demonstrate machine function "live." The educational advantages alone are worth inclusion of this program which runs about one-half memory page, including the conversions between ASCII and octal codes which are used also for experiment program control (EXPALT).

Use of the dual dynamic display for examining two adjacent instructions or variables (including double-precision) or two variables or indexes not necessarily in successive addresses has made debugging almost a pleasure. However, location ØØØ4 is left free for the breakpoint of ODT or DDT if the programmer desires to use separate debugging programs.

Once a new program is debugged, a BIN copy of the core may be punched with DEC-8-5-U BIN PUNCH (HIGH) program, which is initially loaded over output buffer core by MTS-6/70 with starting address of 7465. This cannot be carried out without interrupting the time-sharing schedule, but it is not advisable to load or debug new programs while experiments are in progress since one error can destroy everything in memory. Since a corrected symbolic program or complete listing is necessary for future reference, the program should be edited and reassembled using high-speed reader and punch.

Finally, the debugged program should be checked out for overtime. The MTS control program includes a continuous check and if the next clock pulse has already occurred when control is returned to MTS, the address of that experiment is deposited in location OVRTIM. This may be examined while running by the CTRL E keyboard command followed by the address of OVRTIM, Ø323. The content on display is the absolute starting address of the overtime program. If the content is changing or flickering, more than one program is overtime and the display should be frozen with a RETURN to read and then restarted with another CTRL E until all overtime programs can be identified.

MTS-6/7Ø Operation

Minimum Hardware: 4K PDP-8I, ASR33 Teletype, AX-Ø8 Laboratory Peripheral, with oscilloscope, XR, XM, and XC options up to 16 analog channels (LAB-8 system); PCØ8 high-speed reader/punch.

Program: BIN tape, symbolic ASCII tape, PAL III listing

Starting Address: Ø2ØØ

Core Space: Program is 11-plus pages, including I/O buffers:

ØØ2ØØ to Ø377 - control, schedule tables, ADC
52ØØ to 7611 - service, math, conversion, display routines, buffers, and experiment tables

Fixed Page Ø:

ØØØØ to ØØØ3 - interrupt
ØØØ5 to ØØØ6 - variable and constant
Ø1Ø6 to Ø133 - constants (available to all programs)
Ø134 to Ø177 - variables and addresses of service subroutines

User's Core Space:

ØØ1Ø to ØØ11 - unreserved auto-indexes (available for 1 msec.)
ØØ12 to ØØ13 - may be reserved for specific experiments
ØØ15 to ØØ17 - may be reserved for specific experiments
ØØ14 - reserved for loading teletype buffer for one line of data output
ØØ2Ø to ØØ77 - Page Ø space reserved for specific experiments
Ø1ØØ to Ø1Ø5 - Page Ø unreserved indexes (available for 1 msec.)
Ø4ØØ to 5177 - 19 pages for experiment programs

Description:

Up to 18 experiments may run at the same time without interaction on one millisecond shares scheduled at 100, 10, or 1 per second. Inputs may also be sampled at 1,000 per second, but must be run in less than 0.2 msec. since this subtracts from the 1.0 msec. available to all other shares.

The time-sharing is strictly cooperative since only the teletype, reader, and punch are operated on interrupt between schedules. Programs may be checked for overtime returns.

Keyboard inputs and programmed messages are displayed on the oscilloscope (up to 8 lines of 18 characters each) allowing English interaction with programs. Special keyboard commands allow starting and stopping of experiments, swapping programs on BIN tape, and monitoring and changing of memory without interruption of computer operation. A 24-hour real-time clock is also displayed and available to experiment programs.

The program listings include all system and user service routines and one experiment program example that may be demonstrated without external experimental equipment or subject.

Operation:

1. Load MTS tape with BIN LOADER in core on high-speed reader:

    Place 7777 in switch register (SR). Press LOAD ADD. Change SR to 3777. Press START. Tape stops after short LAB-8 control section. Press CONT. Tape stops after LAB-8 Section II. Press CONT.

    Adjust TIMING controls to align crosshairs on scope. This sets the RC clock for 1000 Hz (1 millisecond rate).

    Press keyboard RETURN. Disregard "TRIGGER" on teletype. Press RETURN again. MTS tape will be read in.

    Place starting address, Ø2ØØ, in SR. Press LOAD ADD. Press START. Program name is displayed on scope.

2. Load experiment program with leader of tape in high-speed reader with keyboard command CTRL B (see 3.)

    Start experiment program with keyboard command CTRL I (see 3.).

    Stop experiment program with keyboard command CTRL O (see 3.).

3. Keyboard Commands: Precede each by RETURN and execute or terminate entry with a RETURN (only one RETURN necessary between commands).

CTRL B (for BIN load) - Load BIN experiment program on high-speed reader (left SR bit= Ø) with leader of tape in reader. Tape stops with "TAPE IN" displayed or "ERROR: RELOAD".

CTRL I (for In) - Follow with 6-character experiment name before pressing RETURN to start experiment in MTS schedule.

CTRL O (for Out) - Follow with 6-character experiment name before pressing RETURN. Turns off experiment MTS schedule.

"NOT LOADED" is displayed to In and Out if experiment tape not loaded.

CTRL T - Read time of day on scope (to 2459: 59)

CTRL S - Set time of day before RETURN (enter in same format as display)

CTRL E - Examines octal contents of octal address, entered from keyboard. RETURN following entry stops the dynamic display of content changes. Two addresses may be dynamically examined by following first address by CTRL A, then back to CTRL E followed by change of second address, if desired, with RUB OUT.

CTRL C - Changes core contents to keyboard octal entry, displayed following "?" and executed by RETURN.

CTRL A - Advances to next memory address for examination and possible change. (Content display is static until followed by CTRL E.) (RETURN is not necessary before repeating or to change command to CTRL E or CTRL C.)

Notes:

a) Errors in keyboard entries may be corrected by pressing RUB OUT before RETURN and starting line over.

b) All control commands are made by pressing and holding down the CTRL key while the letter is typed.

If a CTRL command is made while another command is in effect (as displayed on bottom line), only result will be entry of a "?" in keyboard text. (Begin line over with RUB OUT.)

c) Any experiment may be stopped at any time with CTRL O and restarted from the beginning with CTRL I.

SHUTTL: Demonstration Experiment for MTS-6/7Ø

Minimum Hardware: 4K PDP-8I, ASR33 Teletype, AXØ8 Laboratory Peripheral with oscilloscope and XR option, PCØ8 high-speed reader.

Program: BIN tape, ASCII symbolic tape, PAL III listing

Starting Address: 16ØØ (under keyboard control)

Core Space: 16ØØ-1735 Experiment program
6330-6332 Name code, schedule, and starting address for MTS control

Other Programs Required: MTS-6/7Ø BIN tape

References: MTS-6/7Ø Operation instructions, writeup

Description: Counts number of inter-trial crossings and latency of shock escape and avoidance response of rat in a 2-way shuttle box. After each trial, the teletype prints/punches one line of data consisting of 8-character ID including current trial number (Ø1-99), number of pre-trial crossings (ØØ-99), and response latency (ØØ.ØØ-99.99 sec.)

The experiment is under complete control from external electro-mechanical or Solid-state programming equipment. The computer program is controlled by three "contingency" inputs to the AXØ8 Laboratory Peripheral: C3 for counting pre-trial crossings, C4 for trial start, and C5 for trial end. A crossing will be counted (using MTS service routine CONTAC) each time the C3 input goes to 0 V. for at least 20 msec. (milliseconds) after having been -3 V. for at least 30 msec., and can therefore be controlled by relay contacts of a pulse-former without erroneous counts from contact bounce. The latency clock (100 Hz) starts when C4 input goes from -3V. to 0 V. and is read when C5 input goes from -3V. to 0 V. (even though C4 is still on). The trial data is printed as soon afterwards as the teletype has finished a line from another experiment, if any. This program has data buffer storage for only one trial since it is assumed that trials are spaced at least 15 sec. apart (the maximum time for 2 teletype lines).

The experimental session is under keyboard/scope interactive control and is No. 5 in the experiment table (EXPTBL + (3x4))[10] of MTS-6/70.

Operation:

1. Load and start MTS-6/70 if not already running.

2. Load SHUTTL BIN tape with command CTRL B.

3. Start experiment program (with inputs off) with command CTRL I and experiment name, SHUTTL.

4. If scope displays NOT LOADED, correct experiment tape has not been properly loaded.

5. SHUTTL ID: should be displayed. Type 8-character (letters or digits) ID ending with 2-digit initial trial number (usually Ø1). Terminate with keyboard RETURN if correct on scope echo or correct whole line with RUB OUT key.

6. Following RETURN the experiment program is running and will count inter-trial crossings before the first trial begins.

7. The slow-speed PUNCH should be turned on if the data are to be punched on cards later.

8. The experimental program may be temporarily interrupted by typing letter T (no CTRL). This will display the initial request for new ID.

9. The program may be taken off schedule with command CTRL 0 followed by entry of its name, SHUTTL, plus RETURN key.

10. It may be restarted with step (3).

11. This program has no other programmed scope displays or messages.

— 2 —