DECUS
PROGRAM LIBRARY

DECUS NO.	8-210
TITLE	A REAL-TIME MULTIPLE TASK EXECUTIVE PROGRAM WITH A BUILT-IN CONSOLE UTILITY PACKAGE FOR PDP-8/S AND PDP-8 COMPUTERS
AUTHOR	C. D. Martin, Jr. and R. L. Simpson
COMPANY	Oak Ridge National Laboratory Oak Ridge, Tennessee
DATE	Submitted August 29, 1969
SOURCE LANGUAGE	PAL III

A REAL-TIME MULTIPLE TASK EXECUTIVE PROGRAM WITH A BUILT-IN CONSOLE UTILITY PACKAGE FOR PDP-8/S AND PDP-8 COMPUTERS[1]

C. D. Martin, Jr. R. L. Simpson
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37830

ABSTRACT

An executive routine was developed for the PDP-8/S and PDP-8 computers. This routine schedules process control tasks in real-time and establishes operating priorities. The program (including the utility package) occupies about one-third of a 4096-word memory block and accommodates eight major control tasks. The only hardware addition to the standard computer configuration required by the executive is a real-time interrupt.

INTRODUCTION

If a digital computer is to be used effectively for process control, a means must be provided for scheduling various programs that the computer is to execute and for assigning an operational priority to each program. As an example of the diversity of the programs, the program having the highest priority in a system might be one for scanning the analog inputs to the computer, digitizing them, checking them against prestored limits, initiating the printout of messages alerting operators that signals are out of limits, and storing the digitized values in locations accessible to other programs. The program having the next highest priority might be a control algorithm for examining the digitized analog signals, comparing them against desired values, and initiating a control output to minimize the difference. These might be followed by other programs such as one for printing periodic logs and one for writing the digitized data on magnetic tape for further processing by a larger computer.

Scheduling the execution of the programs, maintaining priorities, and coordinating the use of input-output equipment by several programs are accomplished by a program called a real-time executive. Various actions within the system are triggered by hardware interrupts from input-output equipment and from a real-time clock. A real-time clock gives the time of day while an internal computer clock synchronizes the computer operations.

MEMORY REQUIREMENTS AND SYSTEM OVERHEAD

The real-time executive was developed for the PDP-8 and 8/S computers to schedule the execution of eight different computer programs, or tasks, at desired time intervals. Input-output functions are coordinated so as to eliminate conflicts between tasks. The system can be expanded to accommodate more tasks. An initial limit of eight tasks was selected, because the computer used for developing the system had a memory of only 4K words (see memory map, Fig. 1). The executive system now requires about 576_{10} or 1100_8 words of memory, and the on-line--off-line utility package occupies about 640_{10} or 1200_8 words of memory, leaving the remainder of the memory for system tasks.

The system operates in response to a 60-hertz interrupt in the PDP-8 computer and a 10-hertz interrupt in the PDP-8/S computer. System overhead for the PDP-8/S computer is about 8%, leaving 92% of the available time for task execution. The overhead for the PDP-8 computer is about 3%.

REAL-TIME SCHEDULING

Real time is maintained in the system in two memory words, the cycles counter and the minute counter. In response to a clock interrupt, the cycles counter, which is set initially with the negative number of cycles per minute, is incremented by one count. When the count becomes zero, the minute counter, which is set initially with the negative number of minutes in a day, is incremented. By use of these counters the time of day can be calculated for log purposes. Every minute, when the cycles counter becomes zero, the counter is initialized with the negative number of cycles in a minute. At midnight, when the minute counter becomes zero, the minute counter is reset with the negative number of minutes in a day.

Several tables are used for internal control and status indication of the system. In every table each entry corresponds to a given task. The tasks are numbered 0 to 7 in decreasing order of priority; that is, task number 0 has the highest priority, and task number 7 has the lowest priority. All tables are arranged in this manner to make it easier to index all tables with a single pointer, which is called the task count. The task count can be added to the starting address of any table to form the address of the table entry for the particular task. If no task is being executed the task count is 10_8. The tables are as follows:

1. The status table indicates the status of a task at a particular time.
2. The location table contains the entry point of each task.
3. The period table contains the negative number of minutes between consecutive executions of a task.
4. The offset table controls executions of tasks to optimize scheduling.

[1]Research sponsored by the U.S. Atomic Energy Commission under contract with the Union Carbide Corporation.

5. The cycles and minutes rundown tables are counters for the time between executions of tasks.

6. The printer queue table contains the address of the first ASCII character to be printed for each task.

7. The floating-point-package queue table is used to store the return address to the task after the floating point package has been used.

STATUS TABLE

The information bits in each word of the status table are assigned as follows:

1. If bit 0 is a 1, the task has been actuated for execution.

2. If bit 1 is a 1, the task is waiting for the completion of an input-output function.

3. If bit 2 is a 1, the task has been interrupted.

4. If bit 3 is a 1, the task is waiting for the floating-point package.

5. If bits 4 through 11 are all a zero, the task is not scheduled to be actuated by the clock.

If a task is to be scheduled by the clock, at least one of these bits must be set at a 1.

A task can be actuated for immediate execution by setting bit 0 to a 1 by the keyboard utility package or by another task. The execution of a task is begun or the execution of a previously interrupted task is resumed as a result of a scan of the status table which is scanned after every clock, keyboard, and printer interrupt, and after completion of a task. The scan is always started with the entry corresponding to the highest priority task (task 0). The first task found whose bit 0 is a 1 and whose bit 2 is a zero is executed from the address in the location table. Bit 2 will always be a 1 if either bit 1 or bit 3 is a 1, but not vice versa; that is, a task waiting for the completion of an input-output function or the floating-point package must have been interrupted. If bit 2 is a 1 and bit 3 is a zero, the registers are restored and execution is resumed from the interrupted address. If bit 3 is a 1 (waiting for the floating-point package), the floating-point-package busy flag is checked; if the package is busy, the task count is incremented and the next task is checked. If not busy, the busy flag is set, the registers are restored, and execution is resumed from the interrupted address.

OFFSET, MINUTES RUNDOWN, AND CYCLES RUNDOWN TABLES

The offset table indicates the period in negative number of cycles for a task having a period less than 1 min.

Every entry in the cycles rundown table is incremented every clock cycle to keep the next execution time current with the clock. When an entry in this table becomes zero, the corresponding entry for the same task in the minutes rundown table is incremented. If the corresponding entry too becomes zero, the task is actuated (bit 0 in the status table is set), and the minutes rundown table is reset from the corresponding entry in the period table. The entry in the cycles rundown table is reset to the corresponding entry in the offset table if the period for that task is less than 1 min, otherwise the entry in the cycles rundown table is set to the negative number of cycles per minute.

PRINTER CONTROL

The printer queue table has an entry for each task. The pointer to the task that is using the printer is called the printer busy flag, and it is loaded with the task number. When a task calls for a printer output, the location of the first ASCII character to be printed is stored in the printer queue table entry for that task. Once the output for that task is started, the address is incremented every time a character is printed, and the contents of the incremented address are examined for a negative number. When a negative number is found, the entry in the table is zeroed, bit 1 in the status word for that task is cleared, and the table is scanned for a nonzero entry to start another output on the printer. If no requests are in the printer queue table, the printer busy flag equals 10_8.

INTERRUPT HANDLER

The following list describes the interrupts that the real-time executive is set up to handle (Fig. 2):

1. Low power--The response to this interrupt is to store the contents of the accumulator, of the link, and of the program counter (return address), to store a "jump to a restart routine" instruction in cell zero, and to halt to wait for the power to be restored. When power is restored, the computer is automatically started at cell zero. The "jump to a restart routine" instruction is executed, and the restart routine restores the contents of the accumulator and the link, and jumps to the next instruction in the program that had been interrupted.

2. Clock--The response to this interrupt updates the time-of-day clock (cycles and minutes counters), checks for zero in the minutes counter (indicates midnight), and increments the cycles rundown entry in each task. Since this incrementing could result in the actuation of a higher priority task (having a higher priority than the task interrupted by the clock), the contents of the registers for the interrupted task are saved and are restored before execution of the interrupted task is resumed.

3. Keyboard--This interrupt occurs when there is a character in the teletype output buffer awaiting transfer to the accumulator. This character is checked to determine if it is an "Alt Mod" (alternate mode) character. If it is, the utility package is actuated for immediate execution. If it is not, the character is stored, and the "input-output in progress" bit in the status word for the utility package is cleared so that execution of the task will be resumed after the status

table has been scanned and no task is found for execution that has a higher priority.

4. Printer--This interrupt occurs when the printer has typed a character and is ready to type another. The response procedure was explained in the description of the printer queue table.

5. Parity--Only the PDP-8/S computer is equipped with this interrupt as a standard hardware feature. Since the result of any parity error will soon become evident, the parity errors are counted by incrementing a parity counter and an attempt is made to resume execution of the interrupted task.

The analog-to-digital converter (ADC) interrupt was removed from the interrupt buss on the PDP-8/S computer. All analog-to-digital conversion is done in a task, because a check for an ADC flag requires 38 μsec, but conversion (using the AD8S) requires only 20 μsec. Also, since contents of the registers are saved after every interrupt, the PDP-8/S computer requires more than 1180 μsec and the PDP-8 computer requires more than 75 μsec to save registers. Thus the removal of the ADC interrupt results in a significant time saving.

REGISTER SAVE

To maintain continuity during execution of a task, the contents of eight "registers" are saved when execution of a task is interrupted, because it might happen that a task with a higher priority than the one in execution will be actuated. When execution of the interrupted task is resumed, the task must first be restored to its previous state. The registers saved are the AC, the link, the return address in the interrupted task, and core memory locations 0016, 0017, 0020, 0021.

GENERAL CONSIDERATIONS

When a task is completed, the task return to the executive must be through a task completion routine which clears bits 0 through 3 in the status word for that task and then scans the status table for another task awaiting execution or waits for an interrupt.

All input-output operations must be scheduled by the executive program. When scheduled, all messages are output intact; that is, once a character string is started, all of that string will be printed before another character string will be started. All keyboard input is processed through the utility package.

The floating-point package was changed so that it could be used by a task through the real-time executive.

CONSOLE UTILITY PACKAGE

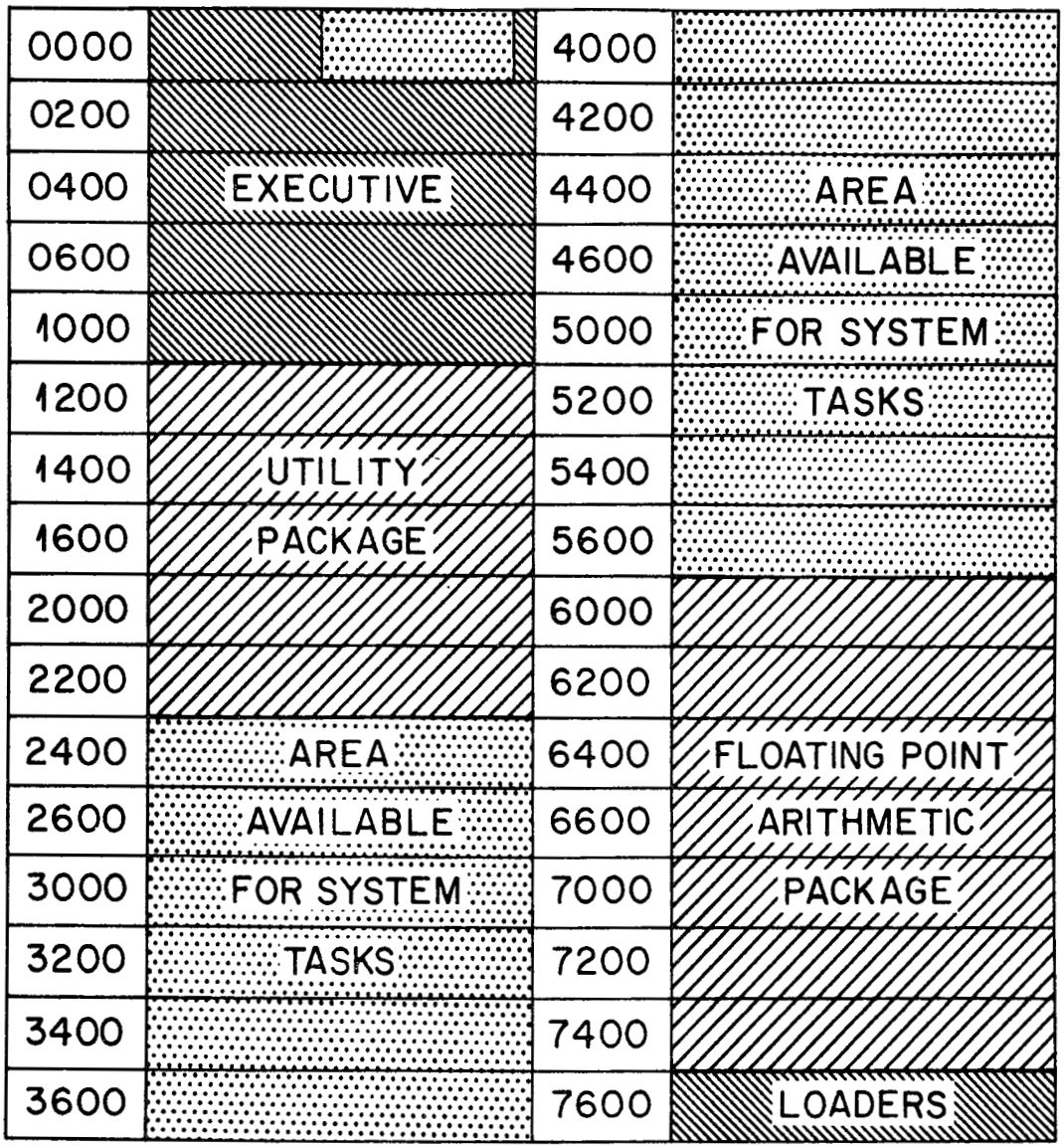
By use of the console utility package the operator or programmer can communicate with the computer either on-line or off-line; that is, he can type a mnemonic code which will set up the following operation: store into the memory from the keyboard, read and punch binary paper tape, obtain an octal dump, actuate a program to be executed only once ("one shot"), clear defined parts of the memory, set defined parts of memory to a specific bit configuration, and disable the keyboard. Before any instructions or information can be stored in the memory from the keyboard or paper tape, the limits of the storage must have been specified. This prevents arbitrary storage and destruction of any program outside the "legal" limits.

The utility program checks all characters typed from the keyboard to determine that the characters are the proper type (alphabetic or numeric) for their position in the instruction. If they are not, the teletype bell will ring, and the character will not be accepted. For example, letters are accepted only as the first two characters, then a comma, and then only octal numbers. Usually, the utility package is run as the task of highest priority, but it can also run at any other priority level.

The console utility package functions with their mnemonic commands are given in the following list:

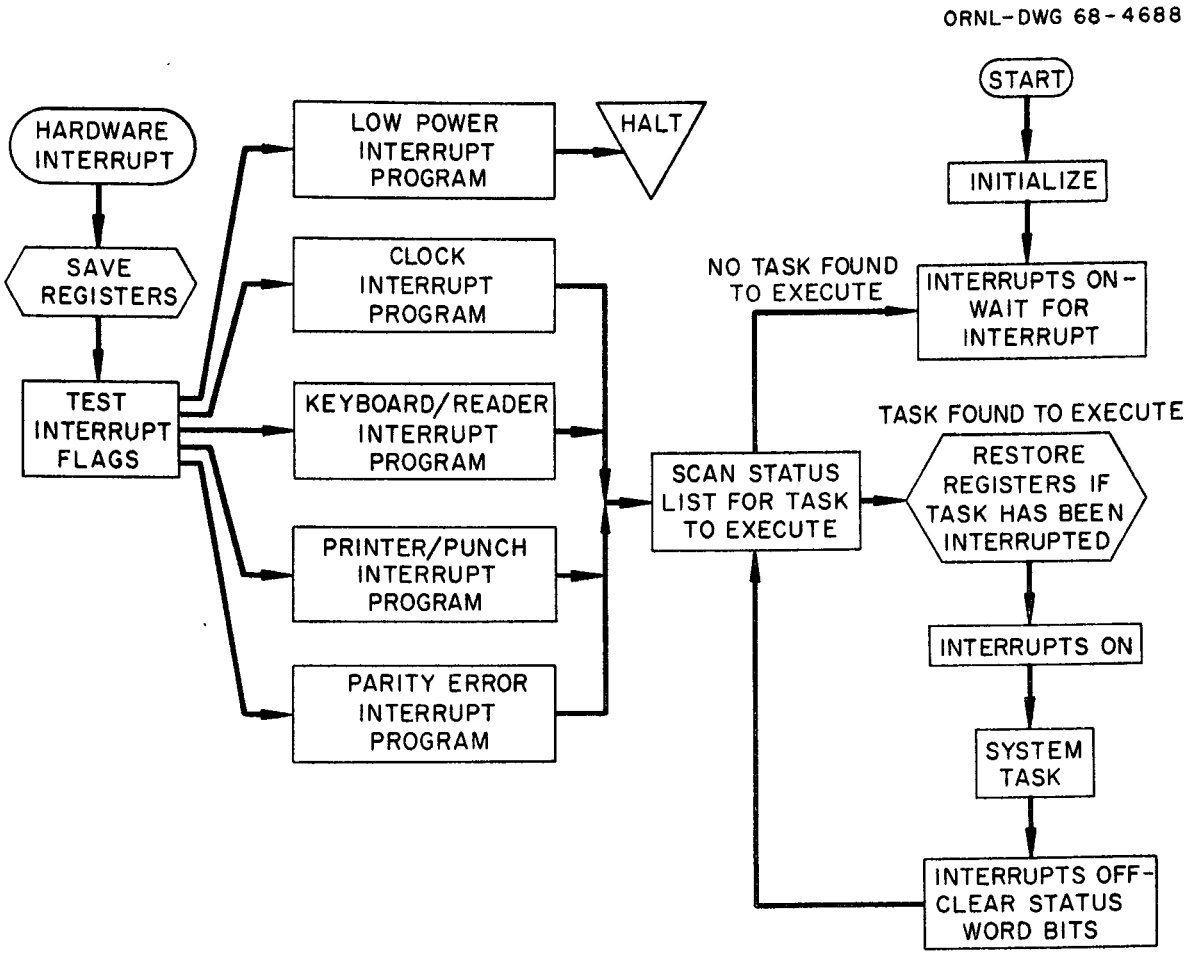
AR,XXXX,YYYY	Defines the first and last addresses of the legal core storage area.
CL,XXXX,YYYY	Clear to zero from XXXX to YYYY.
GO,XXXX	Go to XXXX and execute as lowest priority (one shot) task.
MW,XXXX,YYYY, ZZZZ	Sets ZZZZ into the core from XXXX to YYYY.
PT,XXXX,YYYY	Punches binary tape of core from XXXX to YYYY (inhibits all other teletype message outputs from tasks).
RT,XXXX	Reads binary tape; location can be offset by $XXXX_8$ words (integer number of pages).
TI,XXXX,YYYY	Store into location XXXX the the contents YYYY.
TO,XXXX,YYYY	Type out the contents of XXXX through YYYY.
FI	Disables the keyboard input to all characters except "ALT MOD" and makes all core locations illegal.

ORNL-DWG 68-4687



Memory Map for 4K PDP-8 Real-Time Multiple

Figure 1 Memory Map for a 4K PDP-8 Real-Time Multiple Task System



Schematic Flow Chart of Real-Time Multiple Task Executive for PDP-8/PDP-8S Computers.

Figure 2

OPERATING INSTRUCTIONS:

1. Use the standard D.E.C. binary loader to load the paper tape program.
2. Start the computer at address 0001.
3. Press "ALT MOD" key on Teletype. The printer should then type out "OK."
4. Use the utility package and keyboard to set up a legal area in core for storing tasks. Type: "AR, 2400, 7577 ↵" (Do not type quotes). Computer will reply "OK."

   Place binary tape of task in paper tape reader (low speed) type "LT, ↵" and switch reader to "START." Computer will read tape and print "OK" when finished. Turn off reader at that time. If you assembled your task program using the Pal III assembler, it will have the address 0200 as the first program address. In this case you must define the area from 200 to 7577 and take your chances. An alternate approach is to load your tasks using the D.E.C. binary loader before starting the executive.
5. The utility package can be used to activate tasks when the system is running. Define a legal area of 1000 to 1077 and use the TI (Type In) function to set the starting address in the location table (LTABLE) which starts at address 1010. Note that the utility package uses the first entry in each table since it is now the HIGHEST PRIORITY TASK. This makes the utility package available at all times even in a "time bound" system. Set up the period and offset tables for periodic tasks, and enter the value 0001 in the STATUS TABLE for each periodic task you desire to run, and enter 7777 in the minutes table (1040-1047) and cycles table (1050-1057) for each of these tasks. If the real time clock is running, be sure the cycles table entry is made last. Otherwise turn the real time clock on after all entries are made.

REAL TIME CLOCK

This program is designed to operate with a real time clock using the following device codes:

6311 SKIP ON CLOCK FLAG (SCLF)
6314 CLEAR CLOCK FLAG (CCLF)

The TIME OF DAY IN MINUTES SINCE MIDNIGHT is kept in address 0043. It must be manually synchronized with the time of day (if it is to be used as such). This is accomplished by calculating the time of day in minutes since midnight, converting this value to octal, and adding it to the octal number 5140. The resulting number is then entered in address 0043. This procedure is not required for the proper scheduling of tasks. It only provides an indication of time of day for logs, etc.

If a real time clock having a frequency other than 10 Hz. is used, it will be necessary to change the value of a minute in clock cycles, address 0044. To calculate the value of "minute" to enter in cell 0044, convert the number of clock cycles (interrupts) per minute to octal and subtract it from 10000 octal. For instance, a 60 Hz. clock would produce 60 x 60 = 3600 decimal = 7020 octal interrupts per minute. 10000 - 7020 = 0760, which is the number to enter in address 0044.

Clocks having frequencies higher than 60 Hz. are not recommended because system overhead in responding to higher frequency elapsed time interrupts would be excessive.

PROGRAMMING CONSIDERATIONS

1. All tasks should be terminated by the instruction JMP I Z 100 (5500). This insures that the status word for the task is properly manipulated to indicate that the task is no longer in execution but is ready to be executed from the starting address (contained in the location table) when next actuated by the clock. Failure to use this instruction to terminate a task will cause a system "HANG UP" or "BOMB OUT."

2. All printer output must go through the printer queue. To initiate a printer output use the following calling sequence:

```
        TAD     ADDR/AC MUST BE CLEAR
        IØF         /TURN OFF INTERRUPTS
        JMS I Z 101  /QUEUE THE PRINT REQUEST
        NEXT PROGRAM INSTRUCTION
                |
                V
        JMP I Z 100     /RETURN FROM TASK
ADDR,   ASCII
ASCII,  0317            /O
        0313            /K
        0215            /RETURN
        0212            /LINE FEED
        7777            /OUTPUT TERMINATOR
```

The above coding will cause the printer to type "OK" followed by a carriage return and line feed.

FLOATING POINT ARITHMETIC

Our experiences with floating pointing arithmetic on the PDP-8/S prompted us to rewrite our TASKS using fixed point arithmetic because the floating point is painfully slow. For this reason the floating point package is not included herewith. Do not try to use D.E.C.'s floating point package with this executive as your efforts will most assuredly meet with disaster. If your application requires the floating point arithmetic package, we can furnish an undocumented version which will run with the executive.

TASK ACTIVATION SUBTLETIES

A task which is not desired to be actuated periodically may be actuated for a "ONE-SHOT" execution by entering "4000" in the status table word for the particular task. This may be accomplished manually via the keyboard and utility package, or another task could deposit 4000 in the status word. This is the mechanism for actuating one task by another.

A task may modify any entry in any table. This means that a task may modify its entry point for successive executions. It may also modify its own (or another's) period of execution. This feature provides considerable flexibility in the scheduling of tasks.

DEBUGGING AIDE

During computer idle time the switch register is interrogated by the executive, the contents used as an address, and the contents of that address are displayed on the accumulator panel lights. This feature depends on interrupts for updating, so if your system has no real time clock, press a key on the teletype keyboard to update the display.

CAUTION

This program recognizes only the following interrupts and associated flags:

1. LOW POWER - 6102
2. ELAPSED TIME - 6311
3. ASR 33 KEYBOARD - 6031
4. ASR 33 PRINTER - 6041
5. PDP-8/S PARITY - 6101

Since the interrupt response routine checks flags for only the above devices, an interrupt from a source other than above will result in a system "HANG-UP" in the interrupt response routine.

USER ASSISTANCE

If you have questions on the use of this program, we will be happy to try to answer them. A telephone call may bring instant answers. Contact:

C. D. Martin Ext. 3-1343

R. L. Simpson Ext. 3-6592

R. K. Adams Ext. 3-6034

Oak Ridge National Laboratory (615) 483-8611

3-MODE CONTROLLER

A digital model of the conventional 3-mode controller using fixed point arithmetic and operating within the framework of this executive is available from R. K. Adams. A description of the program is scheduled for presentation at the New York ISA meeting (Session 27).