DECUS
PROGRAM LIBRARY

DECUS NO.	8-529
TITLE	OSCAR: AN OPERATING SYSTEM FOR COMPUTERIZED ANIMAL RESEARCH
AUTHOR	Dennis Kuch and John Platt
COMPANY	McMaster University Hamilton, Ontario, Canada
DATE	April 1972
SOURCE LANGUAGE	PAL III

OSCAR

An Operating System

for

Computerized Animal Research

Programmer's Manual

by

Dennis O. Kuch

Development of the earliest version of OSCAR was the work of Robert Norman. Later versions were developed by John Platt. Development of various versions of OSCAR were supported by grants to John Platt from the United States Public Health Service and the National Research Council of Canada.

Purpose

This manual was written to provide the OSCAR user with

(a) a ready reference to the facilities of the OSCAR system

(b) the background necessary to program the user-supplied subroutines required for the conduct of specific experiments.

Prerequisites

This manual assumes that the reader has a working knowledge of the PAL-III assembly language and is familiar with the PDP-8 family of computers. Information about PAL and the PDP-8 may be obtained from the Digital Equipment Corporation's "Introduction to Programming", chapters one through six. A working knowledge may be obtained at any teletype console plugged into a PDP-8 series computer.

Table of Contents

General Description

1.1 Introduction

The OSCAR system is designed to reduce the amount of programming needed for any specific class of experimental paradigms. In most cases less than four core pages of programming will be required from the user for his particular application. OSCAR oversees the servicing of the experimental stations (operant chambers, subjects' control panels, or other devices), manipulates and stores data for input or output, and provides subroutines for additional programming. OSCAR thus requires somewhat more programming than some available compiler languages but gives the user more flexibility in experimental design than these compilers. OSCAR also provides a degree of generality not available when the experimenter writes a completely new program for each new paradigm.

1.2 General functions

OSCAR's functions may be subdivided into six general categories as follows:

1) The Executive Monitor is the general overseer of the system and determines what gets done, when, and how. It is important to note that OSCAR controls the handling of the user's programming rather than the user's programming controlling OSCAR. That is, the Monitor is a complex interaction of subsystems which precludes the use of parts of OSCAR without engaging all of OSCAR.

2) The interrupt servicing routine determines system priorities in dealing with the input and output devices. First priority is given to a real-time monitor (initiated by pulses from the system clock) which ensures rapid servicing of the experimental stations. Other devices on the interrupt are the teletype reader/punch and the high speed reader/punch.

Inputs from the subject do not interrupt OSCAR but are scanned as required by the user's program.

3) Teletype routines provide operator-computer communication at all times.

4) Special input-output routines are provided for monitoring and control of experimental stations.

5) Mass data dumping may be accomplished by preprogrammed OSCAR routines.

6) OSCAR provides a set of subroutines for commonly needed functions. For example, a pseudorandom number generator, double precision arithmetic package and integer conversion routines are available for use by the user's subroutines.

1.3 User-supplied Subroutines

OSCAR requires four subroutines from the user which are called by the system during operation. These subroutines specify the particular paradigms, inputs, and outputs for classes of experiments and are described in detail in section 3. A brief description of each of the subroutines follows:

1) GETVAR: GETs VARiables (experimental parameters) from the operator.

2) START: initializes the experimental station and sets it up for the run by

(a) clearing and priming the run buffer, wherein are stored the experimental prarmeters and data for the station.

(b) turning on the output relays, which control the stimuli at the experimental station.

(c) putting a 1 in the first location of the run buffer which tells OSCAR to run the buffer.

3) RUN: is the principal and longest subroutine. It controls subject-environment interactions during the experiment proper and stores data.

4) OUTPUT: outputs data on command of the operator.

1.4 Teletype Commands to the System

The operator communicates with OSCAR through eleven numeric commands typed on the teletype: The operator types one of the following numbers on the teletype and OSCAR responds with the underlined message. The operator then types the number of the buffer to be manipulated followed by a period. If no period is given or an illegal buffer number is typed OSCAR ignores the command.

Command	Function
0:	initialize a station (executes GETVAR routine).
1:	start a station (executes START routine).
2:	stop a station.
3:	output data from a station (executes OUTPUT routine).
4:	not implemented (Mag tape dump in some versions).
5:	not implemented. (Calls library monitor in some versions).
6:	Produce ten inches of leader trailer on the high-speed punch.
7:	suppress teletype output, saving messages to the operator.
8:	examine station buffer. The operator types the buffer number followed by a period and then the buffer location he wants to look at. The contents of this location will be typed out by OSCAR. Both the location and its contents are in decimal.

Command	Function
9:	examine core location. The operator types the octal core location he wants to look at and OSCAR responds with the octal contents of that location. The operator then has the option of modifying the contents of the location. If he enters an octal number followed by a period this new number will replace the old contents. Only a period will effect modification of the contents and typing a space will skip the modification option.
::	restart the system.

1.5 OSCAR Core Allocation

OSCAR occupies 14 (decimal) pages of core, leaving 17 pages of core for the user-supplied program, the station run buffers, and the high-speed punch dump buffer. Core allocation is illustrated in Figure 1, which shows OSCAR occupying the lower nine pages and the upper five pages of core. The station run buffers, being above page zero, are said to reside in "upper core".

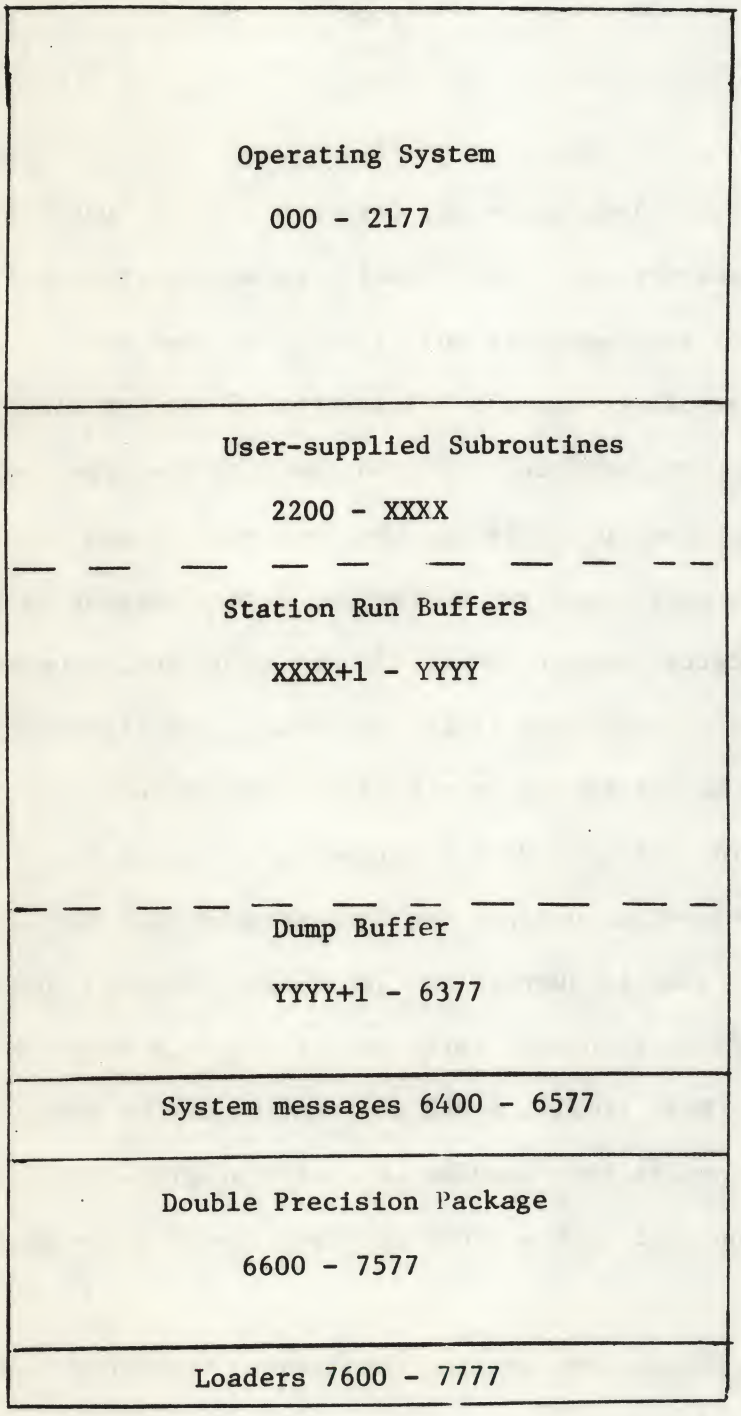
1.6 Run Buffers and the Clock Service Routine

Each station has associated with it a buffer storage area in upper core which contains experimental constants and variables for the particular station. On each clock pulse OSCAR transfers the first part of the buffer--the constants and variables to be used by the RUN subroutine--to page zero so that subroutines of the system can readily access it.

The first part of each run buffer is called a "run table" and contains the station number, subject identification, and experimental parameters. Two locations on page zero specify the size of the run table (TABSIZ is minus the number of locations in the run table) and the number of nonmodifiable constants (ABUMP specifies the number of locations which are not returned to the run buffer and are subsequently erased on page zero). The user must include values for these locations in his program. For example:

Figure 1

OSCAR CORE ALLOCATION



LOCATION	NAME	CONTENTS
0126	TABSIZ,	-5 /Length of run table.
0127	ABUMP,	3 /Number of constants.
0130	RUNTAB,	0 /These three constants.
0131	IDNO,	0 /are required by OSCAR
0132	BOXN,	0 /
0133	NSR,	0 /Variables (or constants, then
0134	FR,	0 /variables) follow.

With each clock pulse and for each station, OSCAR goes through the following "clock service" routine which is noninterruptable:

1) Get the run table and bring it to page zero.

2) Execute the user's RUN routine if the run flag is set.

3) Return variables from run table to the upper core run buffer.

4) Get the run table for the next station and repeat.

In the upper core run buffer the first location is the station status word and indicates whether the station is to be run, stopped or ignored; and whether it has any messages to the operator. The station run flag in this location must be set by the user's START subroutine.

2. OSCAR Subroutines for User Programming

The following routines are addressed on page zero and may be accessed by an indirect jump to subroutine (JMS I XXXX) through these addresses. For example, a decimal-to-binary (DTB) conversion on a number may be affected by a JMS I 0107. More simply, A DTB instruction may be defined by DTB=JMS I 0107, reducing the complexity of coding in a user program. Such shorthand codes are usually included in the OSCAR address directory assembled with the user program.

These subroutines are not reentrant. Since the user supplied subroutines RUN and START may be executed as the result of a clock pulse which has interrupted GETVAR or OUTPUT, any routine called by GETVAR or OUTPUT can not be called by RUN or START, and vice-versa. The functions of the various system subroutines are such that the user should not be tempted to

violate this restriction.

There are four main classes of routines available in OSCAR and they will be discussed in the following order:

(1) Teletype character output

- (a) TYPET & TYPETS: character string type out
- (b) CRLF: carriage return and line feed
- (c) TAB: tabulate

(2) Function routines

- (a) RN: random number generator
- (b) PACK: stores data in dump buffer
- (c) GETI: a combination of TYPETS and DTB
- (d) CLEAR: sets run buffer variables to zero
- (e) HSPARM: reads in ASCII-coded parameters from the high-speed reader

(3) Instructions for OSCAR-station interactions

- (a) RELAY: output to station
- (b) READ: input from station

(4) Arithmetic routines

- (a) DTB: decimal-to-binary conversion
- (b) BTD: binary-to-decimal conversion
- (c) Single precision arithmetic
  - (i) SMULT: multiply
  - (ii) DIVIDE: divide
  - (iii) SQUARE: square
- (d) Double precision arithmetic

2.1 Teletype Character Output Routines

2.11 TYPET & TYPETS

TYPET is a character string output routine which types out ASCII-

coded characters on the teletype. Characters are stored starting at a location whose address must follow the TYPET subroutine call as in:

```
        TYPET           /subroutine call
        MESSAD          /address of message
```

The characters are stored two to a core location in six-bit (stripped of the beginning "2" or "3") ASCII code. The end of the character string is designated by a "00". The message "HELLO" which is composed of the ASCII characters 310, 305, 314, 314 and 317 would thus be stored at MESSAD as follows: MESSAD, 1005; 1414; 1700. TYPET always types a carriage return and line feed before typing the character string.

TYPETS is a subsection of the TYPET subroutine and does not require a message address when called. It uses an address established by the preceding TYPET or TYPETS call. In this manner the TYPET routine may be called to set the message address and succeeding TYPETS called to type out successive character strings. For example a "HELLO....GOODBYE" sequence with intervening programming could be coded as follows:

```
        TYPET           /set the message address
        MESSAD
        TYPETS          /type "HELLO"
          :             /intervening programming
        TYPETS          /type "GOODBYE"
MESSAD, 0000            /a zero sets the TYPETS pointer to
                         MESSAD
        1005;1414;1700  /"HELLO" with 00 terminating the
                          message
        0717;1704;0231  /"GOODBYE"
        0500
```

2.12 CRLF

Calling CRLF will execute a carriage return and line feed on the teletype.

2.13 TAB

Calling TAB causes the teletype to space over the number of times indicated in the location following the call as in

```
TAB
 3                          /the teletype will type 3 spaces
```

2.2 Function Routines

2.21 RN

The RN routine returns with a pseudo-random positive octal fraction between 0000 and 3777 in the accumulator.

2.22 GET1

This subroutine is a combination of the TYPETS and DTB subroutines, typing out a message and then inputting a number from the teletype. Use of the GET1 code saves space for this frequently occurring sequence of subroutines.

2.23 CLEAR

CLEAR sets all nonconstant locations in the run buffer to zero. It is normally used in the START routine to initialize the buffer.

2.24 PACK

This subroutine may be called by the RUN routine to store data in the high-speed punch buffer for later dumping on tape. The subroutine call requires three locations as follows:

```
PACK                       /subroutine call
XXXX                       /address of data to be stored
YYYY                       /number of words to be stored
```

If XXXX were the location DATA and YYYY indicated that three data words were to be packed in the dump buffer, the call would be

```
     PACK                  /call
     DATA                  /data address
      3                    /3 words to be packed
      :                    /intervening programming
DATA,0                     /data address
      0                    /
      0                    /
```

2.25 HSPARM

This subroutine reads in ASCII-coded parameters from the high-speed reader. It is normally called by GETVAR when a program requires a long

variable sequence of parameters. These parameters are stored sequentially beyond the run table of the appropriate run buffer. The number of parameters to be read must be in the accumulator when the subroutine is called. A dollar sign on the parameter tape designates the end of the parameter list. An end of file (EOF) message is typed if the dollar sign is encountered before enough parameters have been read.

2.3 Instructions for OSCAR Station Interactions

2.31 RELAY

This subroutine transfers the contents of the accumulator to the relay output buffer of an experimental station. How the image is transferred to different experimental stations will depend upon the particular system configuration.

RELAY is called with the desired output image in the accumulator and returns with a cleared accumulator. A calling sequence would be:

```
TAD STIM
RELAY
STIM, 4040                    /simulus configuration
```

2.32 READ

Calling this subroutine transfers the contents of an input buffer from an experimental station to the accumulator. A standard subroutine returns with an image of the response inputs in the low order bits of the accumulator, although this will depend upon the particular system configuration.

2.4 Arithmetic Routines

2.41 DTB

This decimal-to-binary conversion routine is called with a cleared accumulator and returns with the converted, binary number in the accumulator. The number is taken from the teletype and must be single precision (less than 4096 in decimal). A space terminates the nput number and a slash -- / --

will erase the full number if typed prior to the terminator.

This subroutine is generally used to pick up parameters at the start of the experiment in the GETVAR routine. It may also be accessed in combination with TYPETS through a GET1 call.

2.42 BTD

The binary-to-decimal conversion routine is called with the number to be outputted in the accumulator. BTD converts this number to decimal and outputs it on the teletype in a four character, integer format. Leading zeros are suppressed and spaces are typed out instead. As BTD uses the double precision interpretive package, it can not be called from the RUN routine.

Single Precision Arithmetic

Because the RUN routine can not use the double precision interpretive package, OSCAR has three single precision arithmetic routines available.

Two locations on page zero are used as registers for these single-precision subroutines. SMACL (Single-precision Multiply ACcumulator, Low-order) is used for the low-order half of a number and SMACH for the high-order half. As this arithmetic is single-precision, SMACH is used only (1) for the result of a multiplication or square if the result exceeds 4095 (decimal) (2) for the dividend in a DIVIDE call if the dividend is double-precision.

2.43 SMULT

This multiply routine is entered with the multiplier in the accumulator and the multiplicand in the location following the subroutine call. The subroutine returns with the low-order part of the product in the accumulator and the high-order part in the page zero location SMACH. A typical calling sequence to multiply 1023 by 2 is:

```
        TAD B                       /Put the multiplicand after the call
        DCA .+3                     /
        TAD A                       /Get the multiplier in the accumulator
        SMULT                       /Call the subroutine
        Ø                           /Multiplicand is now here

A, 1023
B,    2
```

2.44 DIVIDE

To divide A by B this subroutine is entered with B in the accumulator and A in the page zero registers SMACL and SMACH. The subroutine returns with the quotient in the accumulator and the remainder in SMACH. A typical calling sequence is:

```
        TAD A
        DCA SMACL
        TAD A + 1                   /If A is single precision, SMACH must
        DCA SMACH                   /be cleared anyway
        TAD B
        DIVIDE                      /quotient will be in the accumulator
A,      1234
        5670
B,      5231
```

2.45 SQUARE

This routine is entered with the to-be-squared number in the accumulator. It returns with the double-precision square in SMACL and SMACH. The low-order part (SMACL) will be in the accumulator. A typical calling sequence to square A is:

```
        TAD A
        SQUARE                      /A squared will be in the accumulator and
A,      1234                        /any overflow in SMACH
```

Double Precision Interpretive Package

Arithmetic Routines

The double-precision arithmetic routines are called by the user's GETVAR and OUTPUT subroutines to manipulate, store and type out parameters and data.

The double-precision package uses locations 40, 41, 42 and 43 on page

zero as an arithmetic register. These locations are labeled DPAC1, DPAC2, DPAC3, and DPAC4 respectively as they constitute the Double Precision ACcumulator for the package. The low order part of a number is always stored in DPAC1 and higher order parts are stored in the three successive locations.

2.46 Special Instructions

(a) The double-precision package is entered by calling ENTER (or JMS I 7). All succeeding coding is interpreted by the double-precision interpreter and must therefore consist only of the following double-precision instructions. Enter with a cleared accumulator.

(b) Control returns to the main program when an EXIT (0000) instruction is encountered.

2.47 Arithmetic Memory Reference Instructions

(a) DPADD A

The call DPADD A causes the contents of locations A and A + 1 to be added to DPAC1 and DPAC2. The result is in DPAC1 and DPAC2.

(b) DPSUB A

DPSUB A causes the contents of locations A and A + 1 to be subtracted from DPAC1 and DPAC2. The result is in DPAC1 and DPAC2.

(c) DPMUY A

The call DPMUY A multiplies the contents of locations A and A + 1 by the contents of DPAC1 and DPAC2. The four precision answer is left in DPAC1 through DPAC4.

(d) DPDIV A

DPDIV is called with the divisor in DPAC1 and DPAC2. DPAC3 and DPAC4 should first be cleared by a DPMUY X instruction, prior to the division call. A typical calling sequence to divide the contents of B and B+1 by the contents

of A and A+1 is:

```
        TAD B
        DCA DPAC1
        TAD B+1
        DCA DPAC2
        ENTER
        DPMUY ONE                       /Clear DPAC3 & DPAC4
        DPDIV A
        EXIT
    B,  1234
        5670
    A,  3210
        0012
  ONE,    1
          0
```

The resulting quotient will be in DPAC1 and DPAC2 while the remainder will be in DPAC3 and DPAC4.

Note: Indirect addressing may be used with these memory reference instructions. However, an auto-indexing register will not be incremented by the call.

2.48 Operate Instructions

(a) DPSNA

This instruction performs a Skip on Nonzero ACcumulator where the "accumulator" referred to is the double precision accumulator; DPAC1 and DPAC2.

(b) SQROOT

Calling this routine takes the square root of the contents of DPAC1 and DPAC2 and leaves the result in DPAC1 and DPAC2.

(c) DPNEG

This instruction negates the contents of DPAC1 and DPAC2, leaving the result in DPAC1 and DPAC2.

(d) DPNOP

This is a double precision no operation and does nothing.

(e) DPIF

This instruction is followed by three addresses and performs an

arithmetic IF on the double precision accumulator DPAC1 and DPAC2. If the double precision accumulator is (1) negative, a jump to the first address is executed (2) zero, a jump to the second address or (3) greater than zero, a jump to the third address is executed. A typical calling sequence is:

```
DPIF
XXXX                    /where to go if DPAC1 & DPAC2 is negative
YYYY                    /DPAC is zero; go to XXYY
ZZZZ                    /DPAC is greater than zero; go to ZZZZ
```

2.49 Input/Output and Data Transfer Instructions

(a) DPOUT

This instruction performs a binary-to-decimal conversion on the double-precision accumulator and types out the result in the integar format prescribed by the call. For example DPOUT 4 outputs the number with 4 decimal places. The maximum number of digits is 7.

(b) DPGET A

The call DPGET A transfers the contents of A to DPAC1 and the contents of A + 1 to DPAC2. The contents of A and A + 1 are unchanged.

(c) DPPUT A

The call DPPUT A transfers the contents of DPAC1 and DPAC2 to locations A and A + 1. The contents of DPAC1 and DPAC2 are unchanged.

3. Writing the User-Supplied Subroutines

3.1 OSCAR Address Directory and Page Zero Constants

The user's source program (the symbolic listing in PAL-III code) is prefaced by a directory of the addresses of referenced OSCAR subroutines and registers. This is a standard listing which may be appended to all system programs. Using this directory the programmer need simply code a "TYPET" call when he wants a character string typed out and the assembler will translate this into a machine-language JMS I TYPE on the binary tape.

Another advantage of this directory, beyond parsimony in coding, is that changes in OSCAR require only a reassembling of the user's program with the new address directory and not a rewriting of the program.

The programmer must also supply page zero addresses for those locations of the run buffer which are to be referenced by the RUN routine. These locations constitute the run table. Two addresses, IDENT and BOX, are already provided by OSCAR and the user appends those constants and variables which are to be moved from the upper core run buffer to page zero during servicing of the RUN routine. Constants are listed first and followed by variables. The distinction in function between constants and variables is determined by the fact that only "variables" will be returned from page zero to the run buffer after use by the RUN routine. "Constants" therefore remain unmodified in the run buffer. How many parameters are moved from the run buffer to page zero and which parameters are treated as constants is determined by two locations on page zero. The location TABSIZ contains the two's complement of the number of locations in the run table and determines how many parameters are moved from upper core to page zero. The location ABUMP contains the number of constants in the run table (including RUNTAB, IDENT, and BOX) and determines those parameters which will not be modified by the RUN routine. Constants must be listed first, followed by variables.

One variable frequently included in the run table is a location the contents of which points to the end of the run table in upper core. This pointer may then be used by the RUN routine to store data in the remaining locations of the run buffer.

3.2 GETVAR

The GETVAR subroutine is used to obtain experimental parameters from the operator. These should be restricted to parameters which vary among the treatments and could not be handled by program modifications at intervals

of weeks or months. Otherwise, the teletype paper expenditure becomes prohibitive and much operator time is wasted. Other parameters may be stored in the program and modified through use of the "9" command if and when necessary.

OSCAR requests four parameters in response to the operator's 0 (initialize) command:

```
INITIALIZE BUFFER #_____
IDNO _____
BOXN _____
DUMP?_____
```

The buffer number may be any number from 1 to the maximum number of buffers set up at system initialization. If the buffer is currently being run, OSCAR will mention this fact and ignore the command. Although the buffer number need not correspond to the station (or box) number --- BOXN -- it is usually less confusing to maintain this correspondence. The IDNO is an identification number for the subject to be run at this station. The station number is given in response to BOXN. The reply to DUMP? is 1 if you desire a mass data dump as specified by your RUN routine or 0 if you do not. This information tells OSCAR whether or not to accept data for dumping. The dumping routine itself (PACK) must be called by the user in his RUN subroutine as described below.

After OSCAR executes this dialog it jumps to the user's GETVAR subroutine. Included in this subroutine will be text strings requesting frequently changed parameters from the operator. This may be easily accomplished by calling:

```
TYPET
GMES
```

which will set the TYPET routine to the address of GETVAR messages GMES. Now to output text and input parameters on the teletype one calls

```
GET1
```

which executes TYPETS and DTB, leaving the parameter in the accumulator. These parameters may then be stored in the run table by

```
DCA I AUT7
```

where AUT7 (autoindex register 7, location 17 on page zero) has been set to point to the location after BOX in the upper core run table. OSCAR takes care of this address setting of AUT7. A complete GETVAR routine to pick up two parameters from the operator could be as simple as:

```
        GETVAR, HLT
                TYPET
                GMES
                TAD M2
                DCA CTR
        GLOOP,  GET1
                DCA I AUT7
                ISZ CTR
                JMP GLOOP
                JMP I GETVAR
M2,             -2
CTR,             Ø
GMES,            Ø
                4322; 0511; 1606            /"#REINF."
                7240; 0000
                2201; 2411; 1740            /"RATIO SIZE"
                2311; 3205; 7240
                0000
```

the appropriate run table would be

```
                TABSIZ,  -7
                ABUMP,    5
                RUNTAB,   Ø
                IDENT,    Ø
                BOX,      Ø
                NSR,      Ø                 /CONSTANTS
                RATIO,    Ø                 /
                VAR1,     Ø                 /VARIABLES
                VAR2,     Ø                 /
```

The GETVAR routine may be (and often is) more complicated than the above example. Complications usually involved manipulations of the parameters prior to storage in the run table. A more elaborate GETVAR routine is provided by the sample program GDT in the appendix.

3.3 START

After the buffer has been initialized, a "1" command will start the indicated buffer by calling the user's START subroutine. START should

(a) clear the run buffer variables.

(b) initialize all variables in the run table.

(c) start the session by loading the appropriate stimulus configuration in the station output buffer.

(d) set the run flag to one in the first location of the run buffer (RUNTAB).

The following coding is an example of a complete START subroutine which would accomplish these four goals:

```
        START, HLT
(a)            CLEAR                    /CLEAR BUFFER
               TAD NSR                  /SET REINFORCEMENT COUNTER
               CIA
(b)            DCA VAR1
               TAD RATIO                /SET FIXED RATIO COUNTER
               CIA
               DCA VAR2
(c)            TAD STIM                 /SET RELAYS
               RELAY
(d)            IAC                      /SET RUN FLAG
               DCA I RUNTAB
               JMP I START
        STIM,  2024
```

The START routine in the appendix exemplifies a more complex example of this routine.

3.4 RUN

While the GETVAR, START, and OUTPUT subroutines are initiated by the operator through teletype commands, the RUN subroutine is called by OSCAR on every pulse from the system clock. If the run flag of a buffer is set to 1, OSCAR executes the RUN routine.

RUN controls all subject-environment interactions during the experiment. The user must provide for the following functions in his RUN sub-

routine.

(a) RUN must determine the current status of a station. OSCAR must know whether the station is currently in the middle of a trial, in a time out, whether reinforcement is being delivered, etc.

(b) depending upon what is happening at the station, RUN may then increment timers, modify stimuli, or check on the subject's responding.

(c) monitoring of the subject's responding then involves incrementing counters, checking timers, changing stimuli and taking other appropriate actions depending upon the contingencies.

(d) Data storage programming must be provided. At the end of a trial summed session data and/or trial data may be put away in the run buffer for later analysis by the OUTPUT subroutine.

(e) Mass data dumping may be required (see below).

(f) The RUN subroutine must stop the station by depositing 3 in RUNTAB.

If dumping of data on the high-speed paper-tape punch is desired, the subroutine PACK is used. As described previously (section 2.24) this subroutine call includes specification of the address of the data to be dumped and the number of data words to be dumped. The standard format punched on the tape is a record mark in the eighth channel of the first frame followed by successive data words split in half, six bits per tape frame. The low order half of the data word is punched first. Channel seven is always punched in all frames. OSCAR must be told to dump by replying with a "1" to the DUMP? question at station initialization or the PACK routine will be ineffective.

A complete RUN routine for the GDT program is presented in the appendix.

3.5 OUTPUT

The user's OUTPUT subroutine is accessed by OSCAR when the operator

types a "3" on the teletype and gives the desired buffer number. OUTPUT's functions may include

(a) typing out trial-by-trial statistics if such data were stored during the run.

(b) performing arithmetic operations on stored data and typing out summary session statistics.

As in the GETVAR routine text formatting is accomplished by the TYPET and TYPETS subroutines:

```
TYPET
OMES
```

where OMES is the first location of the data output text. The location ADRPNT on page zero is useful here as it points to the run table in upper core. By putting this address into an autoindexing register (AUT6) one may easily access the data stored in the upper core run buffer. A sample OUTPUT routine for the GDT program is provided in the appendix.

APPENDIX

A Sample User's Program:

GDT

a General Discrete Trials

program for behavioral research

A2

The GDT program will run a variety of procedures requiring response-contingent and/or time-contingent changes of stimuli (e.g., FR, VR, FT, VI). Six parameters are specified for each trial and these parameters are either read in on the high-speed reader during buffer initialization or are taken from the run buffer if they were previously stored there. The operator specifies the number of trials, the intertrial-interval stimulus (in decimal), the "hopper mode" (Ø if fixed time, 1 if head-in-hopper time is used), whether or not the overwrite option is used (described below) and whether new trial parameters are to be read in (1 if yes, 0 if using the previous parameters).

The six trial parameters are

1 2 3 4 5 6
Trial time limit, stimulus, magazine time, number of responses, time interval, ITI duration

1. The trial will be terminated, with reinforcement if such is specified by parameter #3, if the response and/or time criteria are not met by this time.

2. The trial stimulus depends upon the system configuration. It is the decimal equivalent of the buffer image to be output to the station.

3. Magazine time is the number of clock pulses during which the magazine will be energized.

4. The number of responses specifies the response criterion as in a FR or VR schedule.

5. The time interval (as in a VI or FI schedule) is specified in seconds.

6. This parameter specifies the duration of the intertrial interval in seconds. A zero indicates free operant.

Three other parameters may be specified for all stations during any

one run:

1. DROM determines what responses, if any, will reset the ITI if they occur during the ITI.

2. RMASK determines what responses will be looked at during the trial. GDT only runs single-response schedules.

3. REIN is the output buffer image for reinforcement.

The summary session statistics typed on the teletype in response to the "3" command are:

1. subject number
2. number of on-trial responses
3. session time
4. number of off-trial responses

There are two independent options available for mass data dumping:

1. If the operator replies "1" to the DUMP? interrogation during buffer initialization, the following data will be punched on the high-speed punch for each subject for each trial:

    a. subject number.
    b. stimulus and reinforcement/nonreinforcement.
    c. number of responses.
    d. time from start of trial to first response, double
    e. precision.
    f. time from first response to end of trial, double
    g. precision.

2. If the operator replies "1" to OVERWRITE during buffer initialization, the above trial-by-trial data will be stored over the trial parameters in the run buffer. This data can later be recorded on magnetic tape in some OSCAR configurations.

```
001                 / OSCAR ADDRESS DIRECTORY
002                 DO=JMS I 0
003                 TYPET=DO 101
004                 TYPETS=DO 102
005                 CRLF=DO 105
006                 TAB=DO 106
007                 DTB=DO 107
008                 BTD=DO 111
009                 SMULT=DO 112
010                 DIVIDE=DO 113
011                 SQUARE=DO 114
012                 RELAY=DO 115
013                 READ=DO 116
014                 RN=DO 117
015                 GET1=DO 120
016                 CLEAR=DO 121
017                 PACK=DO 122
018                 HSPARM=DO 123
019
020                 AUT0=10
021                 AUT1=11
022                 AUT2=12
023                 AUT3=13
024                 AUT4=14
025                 AUT5=15
026                 AUT6=16
027                 AUT7=17
028
029                 ADRPNT=77
030                 SMACL=75
031                 SMACH=76
032                 CONV=65
033
034                 ENTER=DO 7
035                 DPIF=1
036                 DPSNA=2
037                 DPOUT=100
038                 DPNEG=4
039                 SQROOT=3
040                 DPAC1=40
041                 DPAC2=41
042                 DPAC3=42
043                 DPAC4=43
044
045                 OGET=371
046                 ORUN=1510
047                 ODAT=1710
048                 OTAB=126
049                 OBEG=2400
050
051                 ACL=7701
052                 CAM=7621
053                 BSW=7002
```

```
054                 SWP=7521
055                 MQA=7501
056                 MQL=7421
057
058                 PAUSE
```

A6

```
001                 /GDT 770
002                 *ORUN
003   1510   2600           RUN
004   1511   2471           START
005                 *OGET
006   0371   2400           GETVAR
007   0372   3063           OUTPUT
008                 *OTAB
009   0126   7741   TABSIZ, -37
010   0127   0007   ABUMP,  7
011   0130   0000   RUNTAB, 0
012   0131   0000   IDENT,  0
013   0132   0000   BOX,    0
014   0133   0000   TRIAL,  0        /# TRIALS
015   0134   0000   ISTIM,  0        /ITISTIM
016   0135   0000   PCH,    0        /PHOTO CONTROL OF HOP TIME
017   0136   0000   OVERW,  0        /DATA OVERWRITES TRIAL VALUES
018   0137   0000   SESSR,  0
019   0140   0000           0
020   0141   0000   SESST,  0
021   0142   0000           0
022   0143   0000   ITR,    0        /INTERTRIAL RESPONSES
023   0144   0000           0
024   0145   0000   TLIM,   0        /TIME LIMIT (SEC)
025   0146   0000   STIM,   0        /TRIAL VALUES:
026   0147   0000   REINF,  0        /HOP TIME (.02 SEC)
027   0150   0000   RNES,   0        /RESP REQUIREMENT
028   0151   0000   TNES,   0        /TIME REQUIREMENT (SEC)
029   0152   0000   ITI,    0        /ITI (SEC9
030   0153   0000   RCNT,   0        /RESP CNT
031   0154   0000   STR,    0        /START TIME
032   0155   0000           0
033   0156   0000   RTIM,   0        /RUN TIME
034   0157   0000           0
035   0160   0000   RESP,   0        /CONTROL FLAGS:
036   0161   0000   SDIV,   0        /FREQ DIVIDER
037   0162   0000   SFLG,   0        /SECOND FLAG
038   0163   0000   TFLG,   0        /TRIAL COUNTER
039   0164   0000   IFLG,   0        /ITI FLAG
040   0165   0000   HFLG,   0        /HOPPER FLAG
041   0166   0000   DATAP,  0        /TRIAL POINTER
042
043                 *ODAT
044   1710   3160           END
045
046                 *OBEG
047   2400   7402   GETVAR, HLT
048   2401   4501           TYPET
049   2402   2434           GMES
050   2403   4520           GET1             /ASK TRIALS
051   2404   3231           DCA     TEM1
052   2405   1231           TAD     TEM1
053   2406   3417           DCA I   AUT7
```

```
054   2407   4520            GET1              /ASK ITI STIM
055   2410   3417            DCA I    AUT7
056   2411   4520            GET1              /ASK REINF CONTROL MODE
057   2412   3417            DCA I    AUT7
058   2413   4520            GET1              /ASK IF OVERWRITE
059   2414   3417            DCA I    AUT7
060   2415   4520            GET1              /ASK IF NEW TRIAL VALUE
061   2416   7650            SNA CLA
062   2417   5600            JMP I    GETVAR   /NO
063   2420   1231            TAD      TEM1     /YES
064   2421   3040            DCA      DPAC1
065   2422   3041            DCA      DPAC2
066   2423   4407            ENTER
067   2424   3232            DPMUY    D6
068   2425   0000            EXIT
069   2426   1040            TAD      DPAC1
070   2427   4523            HSPARM
071   2430   5600            JMP I    GETVAR
072   2431   0000  TEM1,     0
073   2432   0006  D6,       6
074   2433   0000            0
075   2434   0000  GMES,     0
076   2435   2422            2422              /TRIALS
077   2436   1101            1101
078   2437   1423            1423
079   2440   7240            7240
080   2441   4000            4000
081   2442   1124            1124              /ITISTIM
082   2443   1140            1140
083   2444   2324            2324
084   2445   1115            1115
085   2446   7200            7200
086   2447   1017            1017              /HOP MODE (1=PHOTO CONTROL)
087   2450   2040            2040
088   2451   1517            1517
089   2452   0405            0405
090   2453   7200            7200
091   2454   1726            1726              /OVERWRITE?
092   2455   0522            0522
093   2456   2722            2722
094   2457   1124            1124
095   2460   0577            0577
096   2461   4000            4000
097   2462   2001            2001              /PARAMETERS?
098   2463   2201            2201
099   2464   1505            1505
100   2465   2405            2405
101   2466   2223            2223
102   2467   7740            7740
103   2470   0000            0000
104
105   2471   7402  START,    HLT
106   2472   7240            STA               /CLEAR VARIABLE
107   2473   1320            TAD      STAB
```

```
108  2474  1127          TAD     ABUMP
109  2475  3010          DCA     AUT0
110  2476  1126          TAD     TABSIZ
111  2477  1127          TAD     ABUMP
112  2500  3317          DCA     TEM2
113  2501  3410          DCA I   AUT0
114  2502  2317          ISZ     TEM2
115  2503  5301          JMP     .-2
116  2504  1126          TAD     TABSIZ  /SET TRIAL POINTER
117  2505  7041          CIA
118  2506  1130          TAD     RUNTAB
119  2507  3166          DCA     DATAP
120  2510  1133          TAD     TRIAL   /SET TRIAL CNT
121  2511  7041          CIA
122  2512  3163          DCA     TFLG
123  2513  4321          JMS     NEW     /START NEW TRIAL
124  2514  7001          IAC             /SET RUN FLAG
125  2515  3530          DCA I   RUNTAB
126  2516  5671          JMP I   START
127  2517  0000  TEM2,    0
128  2520  0130  STAB,    RUNTAB
129
130  2521  7402  NEW,     HLT
131  2522  7240          STA             /GET TRIAL VALUES
132  2523  1166          TAD     DATAP
133  2524  3010          DCA     AUT0
134  2525  1360          TAD     M6
135  2526  3317          DCA     TEM2
136  2527  7240          STA
137  2530  1357          TAD     TLIMS
138  2531  3011          DCA     AUT1
139  2532  1410          TAD I   AUT0
140  2533  3411          DCA I   AUT1
141  2534  2317          ISZ     TEM2
142  2535  5332          JMP     .-3
143  2536  1360          TAD     M6      /CLEAR TRIAL DATA
144  2537  3317          DCA     TEM2
145  2540  3411          DCA I   AUT1
146  2541  2317          ISZ     TEM2
147  2542  5340          JMP     .-2
148  2543  1361          TAD     M62     /SET FREQ DIVIDER
149  2544  3161          DCA     SDIV
150  2545  1145          TAD     TLIM    /SET TIME LIMIT
151  2546  7041          CIA
152  2547  3145          DCA     TLIM
153  2550  1146          TAD     STIM    /LOAD RELAYS
154  2551  4515          RELAY
155  2552  1360          TAD     M6      /UPDATE TRIAL POINTE
156  2553  7041          CIA
157  2554  1166          TAD     DATAP
158  2555  3166          DCA     DATAP
159  2556  5721          JMP I   NEW
160  2557  0145  TLIMS,   TLIM
161  2560  7772  M6,      -6
```

```
162  2561  7716  M62,    -62
163
164
001
002
003                *OBEG+200
004  2600  7402  RUN,    HLT
005  2601  3162          DCA     SFLG    /FREQUENCY DIVIDE
006  2602  2161          ISZ     SDIV    /SEC?
007  2603  5207          JMP     .+4     /NO
008  2604  2162          ISZ     SFLG    /YES, SET FLAG
009  2605  1356          TAD     N62     /RESET DIVIDER
010  2606  3161          DCA     SDIV
011  2607  1165          TAD     HFLG    /HOPPER?
012  2610  7650          SNA CLA
013  2611  5236          JMP     CITI    /NO, CHECK FOR ITI
014  2612  1135          TAD     PCH     /YES, PHOTO CONTROL?
015  2613  7650          SNA CLA
016  2614  5222          JMP     .+6     /NO
017  2615  4516          READ            /YES, READ PHOTO CELL
018  2616  7112          CLL RTR
019  2617  7012          RTR
020  2620  7620          SNL CLA
021  2621  5600          JMP I   RUN
022  2622  2165          ISZ     HFLG    /HOP DONE?
023  2623  5600          JMP I   RUN     /NO
024  2624  1152  ITIGO,  TAD     ITI     /YES, ITI REQUEST?
025  2625  7450          SNA
026  2626  5261          JMP     NEWT    /NO, NEW TRIAL
027  2627  7041          CIA             /YES, START ITI
028  2630  3164          DCA     IFLG
029  2631  1356          TAD     N62     /SYNC DIVIDER
030  2632  3161          DCA     SDIV
031  2633  1134          TAD     ISTIM   /LOAD ITI STIM
032  2634  4515          RELAY
033  2635  5600          JMP I   RUN
034  2636  1164  CITI,   TAD     IFLG    /ITI IN PROGRESS?
035  2637  7650          SNA CLA
036  2640  5271          JMP     RCHK    /NO, RUN TRIAL
037  2641  4516          READ            /RESP?
038  2642  0363          AND     DROM    /SET FOR LEFT KEY OR LEVER
039  2643  7650          SNA CLA
040  2644  5254          JMP     .+10    /NO
041  2645  1152          TAD     ITI     /YES, RESET ITI
042  2646  7041          CIA
043  2647  3164          DCA     IFLG
044  2650  2143          ISZ     ITR
045  2651  7410          SKP
046  2652  2144          ISZ     ITR+1
047  2653  5600          JMP I   RUN
048  2654  1162          TAD     SFLG    /SEC?
049  2655  7650          SNA CLA
050  2656  5600          JMP I   RUN     /NO
051  2657  2164          ISZ     IFLG    /YES, ITI OVER?
```

```
052  2660  5600            JMP I   RUN     /NO
053  2661  4760  NEWT,     JMS I   DMPS    /DUMP TRIAL DATA
054  2662  2163            ISZ     TFLG    /ALL TRIALS?
055  2663  5267            JMP     .+4     /NO
056  2664  1357            TAD     C3      /YES, END RUN
057  2665  3530            DCA I   RUNTAB
058  2666  5600            JMP I   RUN
059  2667  4761            JMS I   NEWS    /START NEW TRIAL
060  2670  5600            JMP I   RUN
061  2671  2141  RCHK,     ISZ     SESST   /INC SESS TIME
062  2672  7410            SKP
063  2673  2142            ISZ     SESST+1
064  2674  1153            TAD     RCNT    /INC TRIAL TIME
065  2675  7640            SZA CLA
066  2676  5303            JMP     .+5
067  2677  2154            ISZ     STR     /LATENCY
068  2700  7410            SKP
069  2701  2155            ISZ     STR+1
070  2702  5306            JMP     .+4
071  2703  2156            ISZ     RTIM    /RUN TIME
072  2704  7410            SKP
073  2705  2157            ISZ     RTIM+1
074  2706  1162            TAD     SFLG    /SECOND?
075  2707  7650            SNA CLA
076  2710  5314            JMP     SRESP   /NO
077  2711  2145            ISZ     TLIM    /YES, TIME LIMIT?
078  2712  7410            SKP             /NO
079  2713  5346            JMP     HOP     /YES, END TRIAL
080  2714  4516  SRESP,    READ
081  2715  0364            AND     RMASK
082  2716  4762            JMS I   OFFS    /GET RESP OFFSETS
083  2717  7750            SPA SNA CLA     /RESP?
084  2720  5600            JMP I   RUN     /NO
085  2721  2153            ISZ     RCNT    /YES, ENOUGH RESPONSES?
086  2722  1150            TAD     RNES
087  2723  7041            CIA
088  2724  1153            TAD     RCNT
089  2725  7710            SPA CLA
090  2726  5600            JMP I   RUN     /NO
091  2727  7100            CLL             /YES, ENOUGH TIME
092  2730  1154            TAD     STR
093  2731  1156            TAD     RTIM
094  2732  3075            DCA     SMACL
095  2733  7204            GLK
096  2734  1155            TAD     STR+1
097  2735  1157            TAD     RTIM+1
098  2736  3076            DCA     SMACH
099  2737  1356            TAD     N62
100  2740  7041            CIA
101  2741  4513            DIVIDE
102  2742  7041            CIA
103  2743  1151            TAD     TNES
104  2744  7740            SMA SZA CLA
105  2745  5600            JMP I   RUN     /NO
```

A11

```
106  2746  1147  HOP,     TAD      REINF    /YES, REINF
107  2747  7450           SNA
108  2750  5224           JMP      ITIGO
109  2751  7041           CIA
110  2752  3165           DCA      HFLG
111  2753  1365           TAD      REIN
112  2754  4515           RELAY
113  2755  5600           JMP I    RUN
114  2756  7716  N62,     -62
115  2757  0003  C3,      3
116  2760  3017  DMPS,    DMP
117  2761  2521  NEWS,    NEW
118  2762  3000  OFFS,    OFF
119                /FOLLOWING CONSTANTS ARE CHAMBER SPECIFIC
120  2763  0007  DROM,    7                 /RESP CODE DURING ITI
121  2764  0007  RMASK,   7                 /RESP CODE DURING TRIAL
122  2765  0003  REIN,    3                 /REINF CODE
123
001
002                *OBEG+400
003  3000  7402  OFF,     HLT
004  3001  3216           DCA      TEM3
005  3002  1160           TAD      RESP     /EXCLUSIVE OR
006  3003  0216           AND      TEM3
007  3004  7041           CIA
008  3005  7104           CLL      RAL
009  3006  1160           TAD      RESP
010  3007  1216           TAD      TEM3
011  3010  0160           AND      RESP     /AND WITH OLD IMAGE
012  3011  7421           MQL
013  3012  1216           TAD      TEM3
014  3013  3160           DCA      RESP
015  3014  7501           MQA
016  3015  5600           JMP I    OFF
017  3016  0000  TEM3,    0
018
019  3017  7402  DMP,     HLT               /SUMP TRIAL DATA
020  3020  1131           TAD      IDENT
021  3021  3145           DCA      TLIM
022  3022  4522           PACK
023  3023  0145           TLIM
024  3024  0013           13
025  3025  1136           TAD      OVERW    /OVERWRITE?
026  3026  7650           SNA CLA
027  3027  5250           JMP      RET      /NO
028  3030  1261           TAD      N7
029  3031  1166           TAD      DATAP
030  3032  3010           DCA      AUT0
031  3033  1147           TAD      REINF
032  3034  7640           SZA CLA
033  3035  7001           IAC
034  3036  1146           TAD      STIM
035  3037  3410           DCA I    AUT0
036  3040  1262           TAD      RCNTS
```

```
037  3041  3011          DCA      AUT1
038  3042  1260          TAD      N5
039  3043  3216          DCA      TEM3
040  3044  1411          TAD I    AUT1
041  3045  3410          DCA I    AUT0
042  3046  2216          ISZ      TEM3
043  3047  5244          JMP      .-3
044  3050  7100  RET,     CLL                /UPDATE RESP COUNT
045  3051  1153          TAD      RCNT
046  3052  1137          TAD      SESSR
047  3053  3137          DCA      SESSR
048  3054  7204          GLK
049  3055  1140          TAD      SESSR+1
050  3056  3140          DCA      SESSR+1
051  3057  5617          JMP I    DMP
052  3060  7773  N5,     -5
053  3061  7771  N7,     -7
054  3062  0152  RCNTS,  RCNT-1
055
056  3063  7402  OUTPUT, HLT
057  3064  1077          TAD      ADRPNT
058  3065  3016          DCA      AUT6
059  3066  4501          TYPET
060  3067  3116          OMES
061  3070  4502          TYPETS             /IDENT
062  3071  1416          TAD I    AUT6
063  3072  4511          BTD
064  3073  7240          STA
065  3074  1077          TAD      ADRPNT
066  3075  1127          TAD      ABUMP
067  3076  3016          DCA      AUT6
068  3077  1315          TAD      N3
069  3100  3314          DCA      TEM4
070  3101  4502  LOOP,   TYPETS
071  3102  1416          TAD I    AUT6
072  3103  3040          DCA      DPAC1
073  3104  1416          TAD I    AUT6
074  3105  3041          DCA      DPAC2
075  3106  4407          ENTER
076  3107  0107          0107
077  3110  0000          EXIT
078  3111  2314          ISZ      TEM4
079  3112  5301          JMP      LOOP
080  3113  5663          JMP I    OUTPUT
081  3114  0000  TEM4,    0
082  3115  7775  N3,     -3
083
084  3116  0000  OMES,    0
085  3117  2325          2325               /SUBJ
086  3120  0212          0212
087  3121  4000          4000
088  3122  1716          1716               /ON TRIAL RESPONSES
089  3123  4024          4024
090  3124  2211          2211
```

A13

```
091   3125   0114            0114
092   3126   4022            4022
093   3127   0523            0523
094   3130   2017            2017
095   3131   1623            1623
096   3132   0523            0523
097   3133   4000            4000
098   3134   2305            2305          /SESSION TIME(.02)
099   3135   2323            2323
100   3136   1117            1117
101   3137   1640            1640
102   3140   2411            2411
103   3141   1505            1505
104   3142   4050            4050
105   3143   5660            5660
106   3144   6251            6251
107   3145   4000            4000
108   3146   1706            1706          /OFF TRIAL RESPONSES
109   3147   0640            0640
110   3150   2422            2422
111   3151   1101            1101
112   3152   1440            1440
113   3153   2205            2205
114   3154   2320            2320
115   3155   1716            1716
116   3156   2305            2305
117   3157   2300            2300
118   3160   0000   END,     0
```

```
ABUMP    0127
ACL      7701
ADRPNT   0077
AUT0     0010
AUT1     0011
AUT2     0012
AUT3     0013
AUT4     0014
AUT5     0015
AUT6     0016
AUT7     0017
BOX      0132
BSW      7002
BTD      4511
CAM      7621
CITI     2636
CLEAR    4521
CONV     0065
CRLF     4505
C3       2757
DATAP    0166
DIVIDE   4513
DMP      3017
DMPS     2760
DO       4400
DPAC1    0040
```

```
DPAC2    0041
DPAC3    0042
DPAC4    0043
DPIF     0001
DPNEG    0004
DPOUT    0100
DPSNA    0002
DROM     2763
DTB      4507
D6       2432
END      3160
ENTER    4407
GETVAR   2400
GET1     4520
GMES     2434
HFLG     0165
HOP      2746
HSPARM   4523
IDENT    0131
IFLG     0164
ISTIM    0134
ITI      0152
ITIGO    2624
ITR      0143
LOOP     3101
MQA      7501
MQL      7421
M6       2560
M62      2561
NEW      2521
NEWS     2761
NEWT     2661
N3       3115
N5       3060
N62      2756
N7       3061
OBEG     2400
ODAT     1710
OFF      3000
OFFS     2762
OGET     0371
OMES     3116
ORUN     1510
OTAB     0126
OUTPUT   3063
OVERW    0136
PACK     4522
PCH      0135
RCHK     2671
RCNT     0153
RCNTS    3062
READ     4516
REIN     2765
SAINF    0147
```

```
RELAY   4515
RESP    0160
RET     3050
RMASK   2764
RN      4517
RNES    0150
RTIM    0156
RUN     2600
RUNTAB  0130
SDIV    0161
SESSR   0137
SESST   0141
SFLG    0162
SMACH   0076
SMACL   0075
SMULT   4512
SQROOT  0003
SQUARE  4514
SRESP   2714
STAB    2520
START   2471
STIM    0146
STR     0154
SWP     7521
TAB     4506
TABSIZ  0126
TEM1    2431
TEM2    2517
TEM3    3016
TEM4    3114
TFLG    0163
TLIM    0145
TLIMS   2557
TNES    0151
TRIAL   0133
TYPET   4501
TYPETS  4502
```