SAND75-0613
Unlimited Release

TA533 Function Matrix and Status Interface Test Program

Thomas D. Sullivan

Prepared by Sandia Laboratories, Albuquerque, New Mexico 87115 and Livermore, California 94550 for the United States Energy Research and Development Administration under Contract AT (29-1) 789
Printed April 1976

Sandia Laboratories

SF 2900 Q(7-73)

When printing a copy of any digitized SAND Report, you are required to update the markings to current standards.

Issued by Sandia Laboratories, operated for the United States Energy Research and Development Administration by Sandia Corporation.

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

SAND75-0613
Unlimited Release
Printed April 1976

TA533 FUNCTION MATRIX AND STATUS INTERFACE TEST PROGRAM

Thomas D. Sullivan
Recording Systems Development Division 1115
Sandia Laboratories, Albuquerque, NM 87115

ABSTRACT

A Fortran program is described which allows testing of a TA533. Diagnostic information is printed by the program as an aid for localizing malfunctions in a TA533. Program control is under user input options.

Printed in the United States of America

Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22161
Price: Printed Copy $5.00; Microfiche $2.25

CONTENTS

	Page
Introduction	7
Interfacing to Computer	7
Group Numbers	8
Command Times	9
Operating Modes	10
Program Operation	11
Error Messages	16
Fortran Program and Listing	17
References	29

TA533 FUNCTION MATRIX AND STATUS INTERFACE TEST PROGRAM

Introduction

A Fortran test program has been written for the TA533.[1] This program was written to operate on a PDP-11 computer which utilizes an RT-11 operating system.[2] This test program is capable of testing only one-half of the operating features of the TA533. The features tested are status information about (1) the location of diode pins on the diode matrix band, (2) auxiliary functions, (3) monitoring of range timing, and (4) programmer timing and control of 30 command time functions. The half not tested is the output functions performed by the various timing signals distributed throughout the diode matrix board. This is due to the fact that no monitoring of functions was provided for in the TA533.

Interfacing to Computer

TA533 communications with a PDP-11 are accomplished by means of the MA155 recording Trailer Command Bus. This Command Bus is driven by the TA531 computer/command bus interface[3,4] which, in turn, is driven by a TA528 Digital I/0 card plugged into a PDP-11 small peripheral slot.

Because all communications with the computer are through a TA528, provisions have been made in the test program for the operator to inform the test program of the address assigned to the TA528. Provisions have also been made so that the operator can inform the test program of the device addresses assigned to the TA533 under test. Current address assignments for four different TA533's are:

Unit Number	Octal Address Range	Drawer Number
1	20 → 23	0
2	24 → 27	1
3	30 → 33	2
4	34 → 37	3

This information is entered into the test program by specifying the drawer number for the particular TA533 which is being tested.

Group Numbers

The test program was written to take advantage of the fact that a TA531 can store up to eight Phase-A commands in its memory. Therefore, whenever a status read request is made by the TA533 Test Program, eight sequential status registers are interrogated. Each set of eight addresses used by the TA533 test program has been assigned a unique group number. Group numbers versus TA533 status registers are listed in Table 1. Figure 1 is a pictorial view of the diode matrix board showing the correspondence between group numbers and controller/line numbers.

TABLE I

Group Number	Controller Number/Name	Line Numbers	Sub-Sub Address
1	1	0 and 1	0, 1, 2, and 3
2	1	2 and 3	0, 1, 2, and 3
3	1	4 and 5	0, 1, 2, and 3
4	1	6 and 7	0, 1, 2, and 3
5	2	0 and 1	0, 1, 2, and 3
6	2	2 and 3	0, 1, 2, and 3
7	2	4 and 5	0, 1, 2, and 3
8	2	6 and 7	0, 1, 2, and 3
9	3	0 and 1	0, 1, 2, and 3
10	3	2 and 3	0, 1, 2, and 3
11	3	4 and 5	0, 1, 2, and 3
12	3	6 and 7	0, 1, 2, and 3
13	AUXILIARY	0 and 1	0, 1, 2, and 3
14	AUXILIARY	2 and 3	0, 1, 2, and 3
15	TIME STATUS	RANGE/PROGRAMMER/COMMAND	0, 1, 2, and 3

Each line, auxiliary, and time status, requires four addresses. Therefore, by combining two lines, groups of eight addresses were formed. The four time status addresses were repeated in order to form one group of eight addresses. However, during program execution, only status information returned for the first four addresses of group 15 is used by the program. The information returned in the second half of group 15 is the same as that for the first half of group 15. From the foregoing it can be seen that in order to access all status registers of a TA533, 116 unique Phase-A read commands are required. These addresses are predefined within a TA533. An operator of the program need only concern himself with informing the program the drawer number of the TA533 being tested. The program will generate the 15 groups of eight addresses each after a drawer number has been entered by the operator.

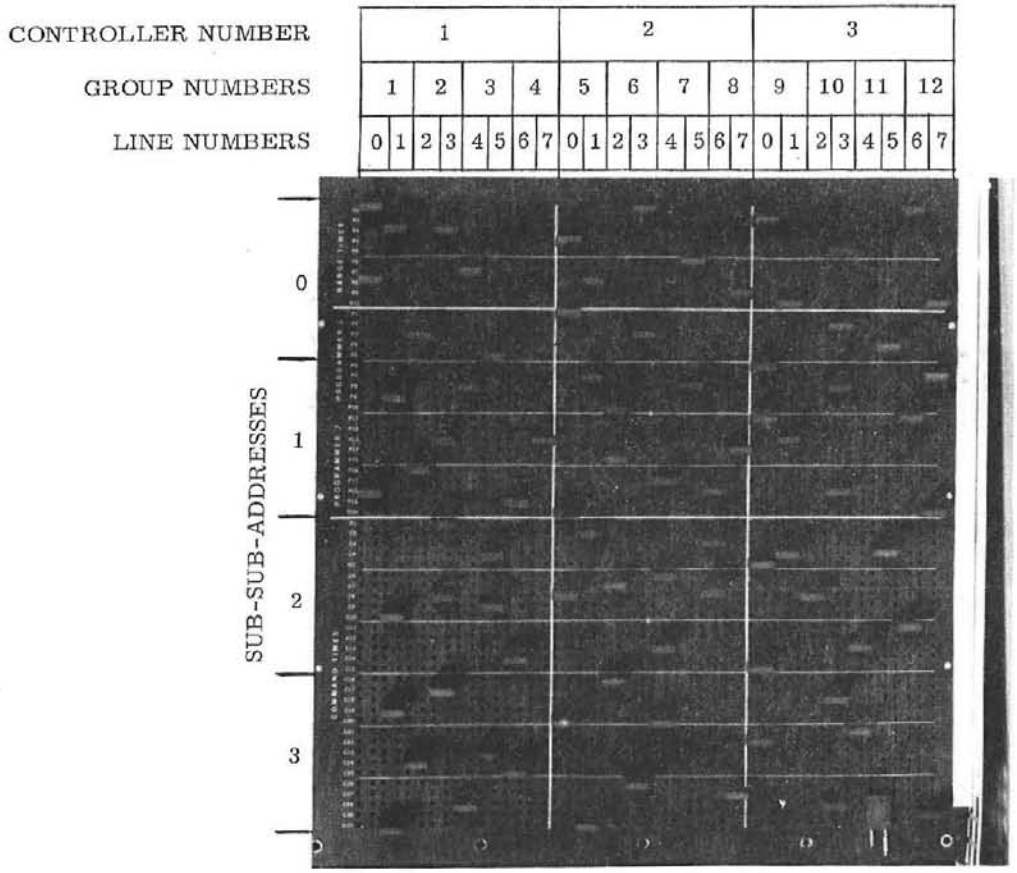

Figure 1. Correspondence between group numbers and controller/line numbers

Command Times

Besides requesting status information, the test program can also control all 30 command time functions of the TA533. These functions are controlled by means of a Phase-A write command and an associated Phase-B word. The design of a TA533 is such that any Phase-A write command with a device address in the range 20_8 to 37_8 will be accepted as valid. In order to control the 30 time functions, 61 unique Phase-B words are required. These Phase-B words can set or reset each of the individual command times. One additional Phase-B word is used as a master reset. To cause a command function to take place, all an operator need be concerned about is which function is to be activated. All required Phase-A and Phase-B words are predefined within the program.

Specifications for setting a given function are a plus sign followed by the function number For example, setting time function 5 is specified by "+5". Resets are specified by a minus sign followed by the function number. "-24" specifies a reset of function 24. Master reset is specified by a zero, "ϕ". An additional feature has been added to the test program for testing proper

operation of the Phase-B decoder. By specifying a "+31" (setting of nonexistent function 31), the program generates a random bit pattern Phase-B word. Testing of this random Phase-B word is done by the program to insure that it is not one of the 61 predefined words. If it is, another random Phase-B word is generated and again tested by the program. The random Phase-B word is sent over the MA155 Command Bus to the TA533. If the TA533 accepts this random Phase-B word as a valid command time control function, the program prints the action taken by the TA533.

In both the Phase-A and Phase-B words used for command time control, certain bits are defined as "don't cares." The test program inserts random bit patterns into these don't-care positions. The reason for this is to test the proper action of the decoders in the TA533. Don't-care bits should be ignored by the TA533. Also, the Phase-A address words for auxiliary and time status contain don't cares. Random bits are also inserted into these Phase-A words.

Operating Modes

In the status mode, the test program acquires the status of various sections of the TA533 which were specified by an operator through group numbers. The current status of the various groups are first stored in a table. The program then continuously interrogates sections of the TA533 and tests the new status against the status table. If any changes of status occurred, the program prints the status word that has changed along with the table status value and the current status. After printing, the status table is updated. In this way, an operator can manually move the diode pins or set auxiliary status inputs, set range timing/programmer timing inputs of the TA533, and monitor these changes. See Reference 1 for connector assignment for auxiliary and range-programmer timing inputs.

In the time function mode before each interrogation cycle of selected group number status information, a Phase-A write/Phase-B word is sent to the TA533. These Phase-B words control the setting or resetting of the 30 command time functions. Status information about command times are contained in the third and fourth words of group 15. Therefore, it is necessary to have selected group 15 before entering the time function operating mode. In this operating mode, only the status information of the abovementioned two words should change. Therefore, it is important for our operator not to manually change the status of the diode pins, auxiliary, range timing, programmer timing functions, while in the time function mode. If manual changes are made by an operator, the test program will report these changes as errors caused by the Phase-B command time words. Errors in setting and resetting the 30 command times are defined as failure to set when in the reset state, failure to reset when in the set state, and the wrong command function occurring. That is, if a "+5" causes the setting of function six, this is an error.

Program Operation

After ascertaining that all interconnecting cables are installed between the TA533-TA531-TA528-PDP-11 and all equipment is turned on, program execution can be initiated. The following commands are entered into the PDP-11 computer via an operator keyboard; after initializing data and time information for the RT-11 operating system.

```
R TA533↓
```

(Operator responses are underlined and terminate with a carriage return ↓)

Program responds with the following message:

```
ENTER OCTAL ADDRESS OF TA528 CONTROL AND STATUS
REGISTER. DEFAULT IS 167770 --- ↓
```

Response in the above example was a carriage return which signifies that the TA528 address being used is 167770. If the octal address entered is in the range of 0 to 77777, the program will use 167770, otherwise the entered address will be used. The program then prints the following prompting message:

```
ENTER OPTION CODE --- ↓
```

For a listing of options available, a response of carriage return is all that is required. The program then prints the following table:

```
ENTER OPTION CODE ---

THE OPTION CODES ARE :
     DR --- DRAWER NUMBER
     DM --- DIODE MATRIX
     AL --- ALL GROUPS
     GR --- GROUP SELECT
     WG --- WHAT GROUPS ARE IN EFFECT
     TB --- TABLE LIST
     CT --- CONTINUE
     TI --- TIME SET & RESET

ENTER OPTION CODE ---
```

Usually, the first option which is entered is DR. This allows an operator to specify the drawer number of the TA533 which is being tested.

```
ENTER OPTION CODE --- DR↓
ENTER DRAWER NUMBER --- 0↓
ENTER OPTION CODE ---
```

In the above example, drawer number zero was entered. Testing is done by the program to insure that drawer number is 0, 1, 2, or 3; if not the operator is requested to enter a correct drawer number.

The next option selected depends on which group numbers an operator wants selected. Option DM prepares the program to ascertain the status of the diode matrix board. From Table I, it can be seen that the DM option selects groups 1 through 12. The SL option selects groups 1 through 15. Normally, all groups are selected. However, if errors are occurring in only a portion of the TA533, option GR lets an operator select only a specific set of group numbers. By selecting only a few groups, program execution time is increased, thus allowing the use of an oscilloscope for troubleshooting the TA533. An example of each group selection option is shown below. Operator group number responses are tested by the program to make sure they are in the range of 1 to 15. If not in this range, an error message is printed and the correct response is solicited from the user.

```
ENTER OPTION CODE --- DM↓
ENTER OPTION CODE --- ALL↓
ENTER OPTION CODE --- GR↓
ENTER NUMBER OF GROUPS --- 3↓
ENTER  1ST GROUP NUMBER --- 1↓
ENTER  2ND GROUP NUMBER --- 15↓
ENTER  3RD GROUP NUMBER --- 13↓
```

Option WG allows an operator to get a listing of the groups that have been selected along with the current drawer number. As can be seen in the computer output below, drawer number is zero along with the three groups selected with the GR option above.

```
ENTER OPTION CODE --- WG↓

DRAWER NUMBER IS  0
   1ST GROUP NUMBER IS  1
   2ND GROUP NUMBER IS 15
   3RD GROUP NUMBER IS 13
ENTER OPTION CODE ---
```

Usually, option TB is selected after the group selection is made. This option prints the current status information of the groups selected and builds the current status table for the program. Since each group addresses eight status locations, only a portion of the table for the above three groups is shown.

```
ENTER OPTION CODE --- TB↓        MSB                         LSB

DR = 0 CT = 1 LN = 0 - 0         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                                 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DR = 0 CT = 1 LN = 0 - 1         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                                 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0

DR = 0 TS = 0 SA = 7 - 2         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                                 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DR = 0 TS = 0 SA = 5 - 3         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                                 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DR = 0 AX = 0 SA =10 - 1         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                                 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DR = 0 AX = 0 SA =11 - 2         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                                 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Format for the above table listing contains information about the section of the TA533 for which status is printed. Abreviations used are DR for drawer number, CT for controller, LN for line number, followed by a sub-sub-address. TS signifies time status change with sub-address (SA) followed by a random number between 0 and 7 and sub-sub-address. AX signifies auxiliary status followed by a sub-address (SA) which is randomly distributed in the range 10 to 17, followed by a sub-sub-address. The two lines of status information contain the table value of status on the first line followed by the current status on the second line. Draw zero, controller one, line 0 - 1 has status bit 11 set to the one state while all other status bits are in the zero state.

After printing of the table, a visual check can be made with the location of the diode pins. In this way, verification is made between actual and computer printed status. The program can now be placed in a fully automatic operating mode with option CT for continue.

```
ENTER OPTION CODE --- CT↓
```

In operating mode CT, the status of the selected groups is continuously read from the TA533 and compared with the table value of status. Any changes in status are printed by the program. Shown below is the effect of moving a diode pin. When the pin was removed (bit 1 of controller 1), line 0 - 3 changed from a one to a zero. Inserting the diode pin caused bit 0 of the same location to change to a one. Manually changing the status of groups not selected before the CT option are

```
DR = 0 CT = 1 LN = 0 - 3         0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
                                 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DR = 0 CT = 1 LN = 0 - 3         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                                 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
```

not reported by the program. Once the CT option is selected, program execution can be halted by typing two control C's in rapid succession. This action causes a return to RT-11 system monitor. Execution of the test program can be restored in two ways. Method one uses the

R TA531 Command. This causes all previous operator input to be reset to zero, and it will be necessary for operator to reenter TA528 address, drawer number, group numbers, and table building option. However, the RT-11 reenter command leaves all program variables in the state they were in prior to the control C exit.

```
^C

^C

RE ↓

ENTER OPTION CODE --- WG↓

DRAWER NUMBER IS  0
   1ST GROUP NUMBER IS  1
   2ND GROUP NUMBER IS 15
   3RD GROUP NUMBER IS 13
ENTER OPTION CODE ---
```

As can be seen by the above example, all information entered has been preserved. The second operating mode of the program can be selected with TI, the time set and reset command. When TI option is selected for a program test, insure that group 15 has been previously chosen. If not selected, an error message is printed informing the operator to select group 15. In this example, group 15 had been selected. As before, a listing of the various time option codes is printed in response to a carriage return.

```
ENTER OPTION CODE --- TI↓

ENTER TIME OPTION CODE --- ↓

THE TIME OPTIONS ARE:
    SE --- SELECT FUNCTIONS
    SR --- SET/RESET SEQUENCE
    CF --- CURRENT FUNCTIONS
    RN --- RANDOM TEST SEQUENCE
    EX --- EXECUTE SE  SR  SEQUENCE

ENTER TIME OPTION CODE ---
```

Option SE allows an operator to specify the sequence in which time functions are to be performed. The number of function wanted must be in the range of 1 to 61. If outside of this range, the program will print an error message and request the operator to enter a correct response. Likewise, time command functions must be in the range of -30 to +31. Commands not in this range are rejected by the program and the operator is requested to enter a correct command. An example of entering five commands is shown below. This sequence sets command time 5, sets time function 20, and reset function 5 and 20. The specifying of function 31 is the manner of specifying a no operation.

```
ENTER TIME OPTION CODE --- SE↓
ENTER NUMBER OF FUNCTIONS WANTED --- 5↓
ENTER FUNCTION  1 --- +5↓
ENTER FUNCTION  2 --- +20↓
ENTER FUNCTION  3 --- -5↓
ENTER FUNCTION  4 --- +31↓
ENTER FUNCTION  5 --- -20↓
```

The command SR generates a predetermined set of command time functions. This set has 61 commands. The first is a master reset followed by a sequential setting and resetting of command time functions 1 through 30.

Option code CF allows an operator to list the command functions which have been selected.

```
ENTER TIME OPTION CODE --- CF↓

FUNCTION  1 IS A   5
FUNCTION  2 IS A  20
FUNCTION  3 IS A  -5
FUNCTION  4 IS A  31
FUNCTION  5 IS A -20
```

Execution of the above time function commands along with status gathering of the selected groups is performed with the EX command.

```
ENTER TIME OPTION CODE --- EX↓

NUMBER OF TESTS = 0.0000E 00          13:58:27     11-NOV-75
NUMBER OF TESTS = 0.1000E 04          14: 0:40     11-NOV-75
NUMBER OF TESTS = 0.2000E 04          14: 2:52     11-NOV-75
```

The number of tests referenced in the computer printout is the total number of individual set/reset/don't care time function commands sent to the TA533.

Program execution can be halted by typing two control C's in rapid succession. This action returns the program to the RT-11 system monitor. Reentry into the program is accomplished with the Reenter Command as explained previously. All previously selected groups and time function commands are still in effect.

A fully random test sequence is entered with the option RN. This test generates the sixty-one allowed command time Phase-B words along with a given percentage of Don't Cares. The Don't Cares correspond to a +31 time function command. In this example, one-fourth of the command time functions are Don't Cares.

```
ENTER TIME OPTION CODE --- RN↓

ENTER PERCENTAGE OF DON'T CARES --- 25↓

NUMBER OF TESTS = 0.0000E 00          14:10:29     11-NOV-75
NUMBER OF TESTS = 0.1000E 04          14:12:55     11-NOV-75
```

Program execution is again halted by typing two control C's.

Listed below are a few typical diagnostic error messages indicating that there is a problem with the TA533's command time functions.

```
NUMBER OF TESTS = 0.0000E 00          15:19:40     11-NOV-75


FUNCTION  -20 WAS CAUSED BY   31, PHASE-B =    140767

FUNCTION  -20 WAS CAUSED BY   -5, PHASE-B =     15713

CANNOT SET FUNCTION  5, PHASE-B =    167012

DR = 0 CT = 1 LN = 0 - 3      0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
CHANGED BY  20 PB=   5750     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0


DR = 0 CT = 1 LN = 0 - 3      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
CHANGED BY  -5 PB= 101713     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
```

As listed above, command time function +31 or a don't care caused a resetting of function twenty (-20). Also listed is the Phase-B word which was used as the don't care. The leading seven bits of each Phase-B time function commands are random. The message "cannot set function 5" indicates that command +5 is not functioning. Listed with this message is the Phase-B word associated with +5 command. Also shown is the setting and resetting of some diode matrix status caused by some of the time function commands.

Error Messages

Besides testing the TA533, this program also tests for proper operation of the TA531. If an error is detected in the operation of the TA531, a message is printed informing the operator of this condition. After printing the error message, program execution is halted. Before resuming testing of the TA533, the operator should test the TA531 with its own test program.[4] Typical error messages and section of program which prints them are listed below.

```
' TA531 NOT SET FOR REMOTE OPERATION'
' ERROR--CANNOT LOAD ALL COMMANDS,TA531 FULL'
' ERROR--NUMBER OF COMMANDS .NE. TO 8'
' COMMANDS IN TA531 DONOT VERIFY'
' REQ A .NE. 1 AFTER COMMANDS TO BUS'
' ERROR IN NUMBER OF COMMANDS AFTER BUS EXECUTE'
' DEVICE ABLE-COMMAND NOT ACCEPTED BY TA533'
' CANNOT LOAD PHASE-A WRITE , REQ-A IS LOW'
' ERROR AFTER LOADING PHASE-A WRITE'
' PHASE-A IN ERROR BEFORE EXECUTE'
' PHASE-B IN ERROR BEFORE EXECUTE'
' REQ-A .NE.1 AFTER WRITE COMMAND TO BUS'
' ERROR IN NUMBER OF WORDS IN TA531 AFTER WRITE EXECUTE'
' DEVICE ABLE -WRITE COMMAND NOT ACCEPTED BY TA533'
' ERROR IN TA531'
```

By checking the program listing, the problem which caused each of the above error messages can be ascertained.

Fortran Program and Listing

The program consists of a main program and 10 subroutines. The main section of the program is used for operator interaction and for controlling execution of the subroutines. Listed below (Table II) are the names of the subroutines and the functions they perform; also listed (Table III) are the variable names used in the test programs and the function they serve. A complete listing of the test program follows.

TABLE II

Subroutine Name	Function
READ	Read status of selected groups
GETB	Driver routine for loading TA531 with 8 Phase-B read words, and return 8 Phase-B status words
PRINT	Print status changes of group status words
OTB	Octal to binary conversion of status words
DECOD	Decodes Phase-A read words into Drawer, Controller, line numbers
JUMBL	Insert random bits into Phase-A words of groups 13, 14, and 15
DATA	Predefined Phase-A address for TA533
TABLE	Predefined Table of Command Time function Phase-B words
TIMFUN	Operator interaction with selecting and execution time function commands
CHECK	Test for correct operation of command time functions
OUTPUT	Driver routine for leading TA531 with 1 Phase-A write command and associated Phase-B time function command

TABLE III

Symbol		Function
		Program MAIN
IBIT	COMMON/ICOM/	Time function number
IFUNC	COMMON/ICOM/	1 for set, -1 for reset of time function
IWA	COMMON/ICOM/	Phase-A write command for time function
IWB	COMMON/ICOM/	Phase-B function command for time function
ITFLG	COMMON/ICOM/	1 when in time function mode, else 0
NGP	COMMON/ISEL/	Number of groups
IGRP	COMMON/ISEL/	List of group number
ICSR	COMMON/ADDR/	Octal address of TA528 control & station register
IODR	COMMON/ADDR/	Octal address of TA528 output register
INDR	COMMON/ADDR/	Octal address of TA528 input register
IER	COMMON	Flag to control printing of status
IA	COMMON	Phase-A read words
IB	COMMON	Current Phase-B status words
ITBL	COMMON	Table Phase-B status words
IBASE	COMMON	Base address for groups 13, 14 and 15
INUMB		Character for printing
IQES		Flag to suppress initial dialog after reenter
M1		Test bit
IDR		Drawer number
J, I		Index
IC		Control listing of only 1st 4 status words of group 15
IK		Control printing of INUMB
		Subroutine READ
IFLG		Used to suppress printing of changes in command time status when in time function mode
J, I		Index
K		A group number
IC		Control listing of only 1st 4 status words of group 15
		Subroutine GETB
K		A group number
I		Index
M1		Request A or device able test bit
IO		TA531 status word
A1		Working storage
J		Number of commands in TA531
I1		Phase-A word returned from TA531
I2		Phase-B word returned from TA531
II, Z		Working storage
		Subroutine PRINT
N		Word pointer for Kth group number
K		Group number
IARY		Unitary status words for table status
IARR		Unitary status words for current status
IPT		Characters for printing identification of status words
IDR		Drawer number
ICT		Controller number
ISA		Sub-address
ISS		Sub-sub-address
NN		Index for IPT
ITFLG		Control printing of changes in status caused by time functions
IPRT		Command time function number and type
		Subroutine OTB
IIN		Octal input word
IRAY		Array of binary digits in IIN
IT		Working storage
Z		Working storage
		Subroutine DECODE
K		Group number
IIA		Phase-A input word
I1		Drawer number returned
I2		Controller number returned

Symbol		Function
		Subroutine DECODE (cont.)
I3		Sub-address returned
I4		Sub-sub-address returned
N		Index value for group numbers
IC		Working storage
Z		Working storage
II		Working storage
		Subroutine JUMBL
L	COMMON/RANDOM/	Seed value for random number generator
M	COMMON/RANDOM/	Seed value for random number generator
I		Index
J		Index
IRR		Random digit
		Subroutine DATA
I2		Base address for Phase-A words
INDX		Index value
II		Index value
IJ		Index value
IK		Index value
		Subroutine TABLE
ISET		Predefined Phase-B words for setting command time
IRSET		Predefined Phase-B words for resetting command time
ISA		Random sub-address
ISS		Random sub-sub-address
IDA		Random device address
IBX		Particular Phase-B word
IX		Random bit pattern
IB15		Random bit pattern
		Subroutine TIMFUN
ISEQ		Array of command time functions
NFUNC		Number of functions in ISEQ
I1	COMMON/ICK/	Binary equivalent array of time function status
I2	COMMON/ICK/	Binary equivalent array of time function status
IBT	COMMON/ICK/	Current status of time functions 1 or ϕ
IBTB	COMMON/ICK/	Table status of time functions 1 or ϕ
ICODE		User input option
NCYCLE		Cycle counter
TOTAL		Cycle totalizer
ITIME		Flag to control printing of TOTAL
NN, I		Index
IRFLG		1 if in random mode, else ϕ
CENT		Percent of no-ops
RANGE		Used in generation of no-ops
		Subroutine CHECK
ICH		0 if no change, 1 if change occurred
I		Index
ICW		-1, 0, 1, 2, Table status + IFUNC
IWH		What happened 1 if set, -1 for reset
IPT		Bit which changed -30 to +30
IPC		Command time function -30 to +31
		Subroutine OUTPUT
IPA		Phase-A write word
IPB		Phase-B command word
M1		Request A or device able test bit
IO		TA531 status word
A1		Working storage
J		Number of Phase-A words in TA531
I1		Phase-A word from TA531
I2		Phase-B word from TA531
II		Working storage
Z		Working storage

```
C
C       TA533 TEST PROGRAM
C       WRITTEN BY T.D.SULLIVAN
C       DIVISION - 1115
C
        COMMON/ICOM/IBIT,IFUNC,IWA,IWB,ITFLG
        COMMON/ISEL/NGP,IGRP(15)
        COMMON/ADDR/ICSR,IODR,INDR
        COMMON IER,IA(8,15),IB(8,15),ITBL(8,15),IBASE(8,3)
        DIMENSION INUMB(4)
        DATA INUMB/"52123,"42116,"42122,"44124/
        DATA ITBL/120*0/
        DATA IQES/0/
        IF(IQES.NE.0) GO TO 27
        TYPE 15
15      FORMAT(' ENTER OCTAL ADDRESS OF TA528 CONTROL AND STATUS'/
        1' REGISTER. DEFAULT IS 167770 --- '$)
        ACCEPT 20,ICSR
20      FORMAT(O6)
        IF(ICSR.GE."0) ICSR="167770
        IODR=ICSR+"2
        INDR=ICSR+"4
        IQES=1
27      CALL PUT("2,ICSR) ! CSR1=1
        CALL PUT("50000,IODR) ! INHIBIT & STATUS
        CALL BITT(11,INDR,M1) ! READ STATUS BIT 11
        CALL PUT("0,ICSR) ! CSR1=0
        IF(M1.EQ.1) GO TO 52
        STOP ' TA531 NOT SET FOR REMOTE OPERATION'
52      TYPE 50
50      FORMAT(' ENTER OPTION CODE --- '$)
        ITFLG=0
        ACCEPT 55,ICODE
55      FORMAT(A2)
        IF(ICODE.EQ."44524) GO TO 350 ! TI
        IF(ICODE.EQ."51104) GO TO 100 ! DR
        IF(ICODE.EQ."52103) GO TO 150 ! CT
        IF(ICODE.EQ."41124) GO TO 200 ! TB
        IF(ICODE.EQ."51107) GO TO 250 ! GR
        IF(ICODE.EQ."46101) GO TO 300 ! AL
        IF(ICODE.EQ."46504) GO TO 400 ! DM
        IF(ICODE.EQ."43527) GO TO 500 ! WG
        TYPE 60
60      FORMAT(' THE OPTION CODES ARE :'/5X,' DR ---- DRAWER NUMBER'/
        15X,' DM ---- DIODE MARTRIX'/
        15X,' AL ---- ALL GROUPS'/5X,' GR ---- GROUP SELECT'/
        35X,' WG ---- WHAT GROUPS ARE IN EFFECT'/
        25X,' TB ---- TABLE LIST'/5X,' CT ---- CONTINUE'/
        35X,' TI ---- TIME SET & RESET'/)
        GO TO 52
100     CONTINUE
        TYPE 110
110     FORMAT(' ENTER DRAWER NUMBER --- '$)
        ACCEPT 255,IDR
        IF(IDR.GE.0.AND.IDR.LE.3) GO TO 125
        TYPE 120
120     FORMAT(' DRAWER NUMBERS ARE: 0 , 1 , 2 , OR  3'/)
        GO TO 100
125     CALL DATA(IDR)
        GO TO 52
150     CALL READ(0)
        GO TO 150
```

```
200     CALL JUMBL
        DO 210 J=1,NGP
210     CALL GETB(IGRP(J))
        DO 220 J=1,NGP
        IC=8
        IF(IGRP(J).EQ.15) IC=4
        DO 220 K=1,IC
220     CALL PRINT(K,IGRP(J))
        GO TO 52
250     TYPE 252
252     FORMAT(' ENTER NUMBER OF GROUPS --- '$)
        ACCEPT 255,NGP
255     FORMAT(I2)
        IF(NGP.GT.0.AND.NGP.LT.16) GO TO 260
        TYPE 258
258     FORMAT(' NUMBER OF GROUPS MUST BE 1 TO 15'/)
        GO TO 250
260     DO 270 I=1,NGP
        IK=I
        IF(I.GT.4) IK=4
263     TYPE 265,I,INUMB(IK)
265     FORMAT(' ENTER',I3,A2,' GROUP NUMBER --- '$)
        ACCEPT 255,IGRP(I)
        IF(IGRP(I).GT.0.AND.IGRP(I).LT.16) GO TO 270
        TYPE 268
268     FORMAT(' GROUP NUMBERS ARE 1 TO 15 '/)
        GO TO 263
270     CONTINUE
        GO TO 52
300     NGP=15
        DO 310 I=1,NGP
310     IGRP(I)=I
        GO TO 52
350     CALL TIMFUN
        GO TO 52
400     NGP=12
        DO 410 I=1,NGP
410     IGRP(I)=I
        GO TO 52
500     TYPE 505,IDR
505     FORMAT(' DRAWER NUMBER IS',I3)
        DO 520 I=1,NGP
        IK=I
        IF(I.GT.4) IK=4
520     TYPE 525,I,INUMB(IK),IGRP(I)
525     FORMAT(I5,A2,' GROUP NUMBER IS',I3)
        GO TO 52
        STOP ' END OF PROGRAM'
        END
C
C       READ IN ALL SELECTED GROUPS
C
        SUBROUTINE READ(IFLG)
        COMMON/ISEL/NGP,IGRP(15)
        COMMON IER,IA(8,15),IB(8,15),ITBL(8,15),IBASE(8,15)
        CALL JUMBL
        DO 33 J=1,NGP
33      CALL GETB(IGRP(J))
        IF(IER.EQ.1) GO TO 25
```

```
35      DO 45 J=1,NGP
        K=IGRP(J)
        IC=8
        IF(K.EQ.15) IC=4
        DO 45 I=1,IC
        IF(K.EQ.15.AND.IFLG.EQ.1.AND.I.GT.2) GO TO 45
        IF(IB(I,K).NE.ITBL(I,K)) CALL PRINT(I,K)
45      CONTINUE
        RETURN
25      TYPE 29
29      FORMAT(//)
        IER=0
        GO TO 35
        END
C
C       READ PHASE B WORDS OF GROUP K
C
        SUBROUTINE GETB(K)
        COMMON/ADDR/ICSR,IODR,INDR
        COMMON IER,IA(8,15),IB(8,15),ITBL(8,15),IBASE(8,3)
        CALL PUT("2,ICSR) ! CSR1=1
        CALL PUT("60000,IODR) ! INHIBIT & DISARM
        CALL PUT("0,ICSR) ! CSR1=0
        DO 10 I=1,8
        CALL BITT(7,ICSR,M1) ! GET REQ-A
        IF(M1.EQ.1) GO TO 20
        STOP ' ERROR--CANNOT LOAD ALL COMMANDS,TA531 FULL'
20      CALL PUT(IA(I,K),IODR) ! LOAD TA531 WITH PHASE-A (IA(I,K))
10      CONTINUE
        CALL PUT("2,ICSR) ! CSR1=1
        CALL PUT("50000,IODR) ! INHIBIT & STATUS
        CALL GET(INDR,IO) ! GET STATUS WORD AS IO
        A1=IO/32.
        J=16.*(A1-IFIX(A1)) ! J=# OF PHASE-A IN TA531
        IF(J.EQ.8) GO TO 30
        TYPE 35,IO,J
35      FORMAT(' STATUS WORD IS ',O6,' # OF COMMANDS =',I3)
        CALL PUT("0,ICSR) ! CSR1=0
        STOP ' ERROR--NUMBER OF COMMANDS .NE. TO 8'
30      CALL PUT("0,ICSR) ! CSR1=0
        DO 45 I=1,8
        CALL GET(INDR,I1) ! READ PHASE-A FROM TA531
        CALL GET(INDR,I2) ! READ PHASE-B FROM TA531
        IF(I1.EQ.IA(I,K)) GO TO 45 ! VERIFY PHASE-A
        TYPE 50,I,IA(I,K),I1
50      FORMAT(' ERROR IN PHASE-A ',I3,2O10)
        STOP ' COMMANDS IN TA531 DONOT VERIFY'
45      CONTINUE
        CALL PUT("1,ICSR) ! CSR0=1
        CALL BITT(7,ICSR,M1) ! GET REQ-A
        IF(M1.EQ.1) GO TO 60
        STOP ' REQ A .NE. 1 AFTER COMMANDS TO BUS'
60      CALL PUT("2,ICSR) ! CSR1=1
        CALL PUT("150000,IODR) ! ODD DATA,INHIBIT & STATUS
        CALL GET(INDR,IO) ! READ STATUS AS IO
        IO=IO-"100000
        A1=IO/32.
        J=16.*(A1-IFIX(A1)) ! J=# OF PHASE-A IN TA531
        IF(J.EQ.8) GO TO 65
        TYPE 35,IO,J
        CALL PUT("0,ICSR) ! CSR1=0
        STOP ' ERROR IN NUMBER OF COMMANDS AFTER BUS EXECUTE'
```

```
65      II=IO/1024.
        Z=II/2.
        I2=Z
        M1=(Z-I2)*2.  ! EXTRACT BIT 10 FROM STATUS-DEVICE ABLE
        IF(M1.EQ.0) GO TO 70
        STOP ' DEVICE ABLE-COMMAND NOT ACCEPTED BY TA533'
70      CALL PUT("0,ICSR) ! CSR1=0
        DO 80 I=1,8
80      CALL GET(INDR,IB(I,K)) ! READ PHASE-B FROM TA531
        CALL PUT("2,ICSR) ! CSR1=1
        CALL PUT("44000,IODR) ! INHIBIT & DISPLAY ENABLE
        CALL PUT("0,ICSR) ! CSR1=0
        RETURN
        END
C
C       PRINT CHANGES AND BUILD TABLE FOR GROUP K  ITEM N
C
        SUBROUTINE PRINT(N,K)
        DIMENSION IARY(16),IARR(16),IPT(2,3)
        COMMON IER,IA(8,15),IB(8,15),ITBL(8,15),IBASE(8,3)
        COMMON/ICOM/IBIT,IFUNC,IWA,IWB,ITFLG
        DATA IPT/"52103,"47114,"51524,"40523,"54101,"40523/
        CALL OTB(ITBL(N,K),IARY)
        CALL OTB(IB(N,K),IARR)
        CALL DECOD(K,IA(N,K),IDR,ICT,ISA,ISS,NN)
        TYPE 10,IDR,IPT(1,NN),ICT,IPT(2,NN),ISA,ISS,IARY
        IF(ITFLG.EQ.1) GO TO 20
        TYPE 15,IARR
10      FORMAT(' DR =',I2,1X,A2,' =',I2,1X,A2,' =',O2,' -',I2,5X,16O2)
15      FORMAT(30X,16I2/)
        GO TO 30
20      IPRT=IFUNC*IBIT
        TYPE 25,IPRT,IWB,IARR
25      FORMAT(' CHANGED BY',I4,' PB=',O7,4X,16O2/)
30      IER=1
        ITBL(N,K)=IB(N,K)
        RETURN
        END
C
C       CONVETR OCTAL NUMBER IIN TO 16 BINARY DIGITS AND STORE IN IRAY
C
        SUBROUTINE OTB(IIN,IRAY)
        DIMENSION IRAY(1)
        IT=IIN
        IRAY(1)=0
        IF(IT.GE.0) GO TO 10
        IT=IT-"100000
        IRAY(1)=1
10      DO 20 I=2,16
        Z=IT/2.
        IT=Z
        IRAY(18-I)=(Z-IT)*2.
20      CONTINUE
        RETURN
        END
C
C       GET ADDRESS INFORMATION FROM PHASE A WORD IIA
C       AND GROUP K .  OUTPUT DRAWER #-I1 , CONTROLLER #-I2
C       SUBADDRESS-I3 , SUBSUB-I4 , AND N TITLE NUMBER
```

```
C
      SUBROUTINE DECOD(K,IIA,I1,I2,I3,I4,N)
      IC=IIA+"100000
      Z=IC/256.
      II=Z
      IK=(Z-II)*256.
      Z=IK/4.
      I1=Z
      I2=(Z-I1)*4.
      Z=II/16.
      II=Z
      I3=(Z-II)*16.
      Z=II/4.
      II=Z
      I4=(Z-II)*4.
      N=1
      IF(K.EQ.13.OR.K.EQ.14) N=3
      IF(K.EQ.15) N=2
      I1=I1-4
      RETURN
      END
C
C     INSERT DONT CARES INTO PHASE-A WORDS OF SUX & TIME STATUS
C
      SUBROUTINE JUMBL
      COMMON/RANDOM/L,M
      COMMON IER,IA(8,15),IB(8,15),ITBL(8,15),IBASE(8,3)
      DO 10 I=1,3
      DO 10 J=1,8
      IRR=8.*RAN(L,M)
      IRR=IRR*256
10    IA(J,12+I)=IBASE(J,I)+IRR
      RETURN
      END
C
C     GIVEN DRAWER NUMBER J ,GENERATE 116 UNIQUE PHASE A WORDS
C     AND STORE AS 15 GROUPS OF 8 ITEMS (TIME STAT IS REPEATED FOR
C     A TOTAL OF 120 WORDS)
C
      SUBROUTINE DATA(J)
      DIMENSION IZ(8,5)
      COMMON IER,IA(8,15),IB(8,15),ITBL(8,15),IBASE(8,3)
      DATA IZ/"100021,"110021,"120021,"130021,"100421,"110421,
     5        "120421,"130421,
     1        "101021,"111021,"121021,"131021,"101421,"111421,
     6        "121421,"131421,
     2        "102021,"112021,"122021,"132021,"102421,"112421,
     7        "122421,"132421,
     3        "103021,"113021,"123021,"133021,"103421,"113421,
     8        "123421,"133421,
     4        "104020,"114020,"124020,"134020,"104021,"114021,
     9        "124021,"134021/
      INDX=1
      DO 30 II=1,3
      DO 20 IJ=1,4
      DO 10 IK=1,8
10    IA(IK,INDX)=IZ(IK,IJ)+II-1+J*4
      INDX=INDX+1
20    CONTINUE
30    CONTINUE
      DO 40 IK=1,4
      IBASE(IK,3)=IZ(IK,1)-1+J*4
```

```
40      IBASE(IK+4,3)=IZ(IK,1)-1+J*4
        DO 50 IK=1,8
        IBASE(IK,1)=IZ(IK,5)+J*4
50      IBASE(IK,2)=IZ(IK,5)+2+J*4
        RETURN
        END
C
C       GET PHASE B WORD FOR IBIT
C       GENERATE RANDOM PHASE-A
C
        SUBROUTINE TABLE
        COMMON /RANDOM/L,M
        COMMON/ICOM/IBIT,IFUNC,IWA,IWB,ITFLG
        DIMENSION ISET(30),IRSET(30)
        DATA  ISET/"1302,"1504,"0606,"0310,"1012,"1614,"0516,"1620,
        1          "0522,"0324,"1026,"1530,"0632,"0034,"1336,"1440,
        2          "0742,"0144,"1246,"1750,"0452,"0254,"1156,"0260,
        3          "1162,"1764,"0466,"0170,"1272,"1474/
        DATA IRSET/"1403,"1205,"0107,"0411,"1713,"1115,"0217,"1121,
        1          "0223,"0425,"1727,"1231,"0133,"0735,"1437,"1341,
        2          "0043,"0645,"1547,"1051,"0353,"0555,"1657,"0561,
        3          "1663,"1065,"0367,"0671,"1573,"1375/
        ISA=16.*RAN(L,M)
        ISS= 4.*RAN(L,M)
        IDA=16.*RAN(L,M)
        IWA=IDA+256*ISA+4096*ISS+"40020
        IF(IBIT.GT.30) GO TO 10
        IF(IBIT.EQ.0) GO TO 30
        IF(IFUNC.EQ.-1) GO TO 40
        IBX=ISET(IBIT)
15      IX=32.*RAN(L,M)
        IB15="0
        IF(RAN(L,M).GT..5) IB15="100000
        IWB=IB15+IX*1024+IBX
        RETURN
10      IBIT=31
        IBX=1024.*RAN(L,M)
        DO 20 K=1,30
        IF(IBX.EQ.ISET(K).OR.IBX.EQ.IRSET(K)) GO TO 10
20      CONTINUE
        IF(IBX.EQ."0701) GO TO 10
        GO TO 15
30      IBX="0701
        GO TO 15
40      IBX=IRSET(IBIT)
        GO TO 15
        END
C
C       EXECUTE SET & RESET OF COMMAND TIMES
C       CHECK OPERATION OF THESE FUNCTIONS
C
        SUBROUTINE TIMFUN
        COMMON/RANDOM/L,M
        COMMON/ISEL/NGP,IGRP(15)
        COMMON IER,IA(8,15),IB(8,15),ITBL(8,15),IBASE(8,3)
        COMMON/ICOM/IBIT,IFUNC,IWA,IWB,ITFLG
        COMMON/ICK/ I1(16),I2(16),IBT(30),IBTB(30)
        DIMENSION ISEQ(61)
        DATA NFUNC,ISEQ/62*0/
        DO 1 I=1,NGP
        IF(IGRP(I).EQ.15) GO TO 3
```

```
1       CONTINUE
        TYPE 2
2       FORMAT(' GROUP 15 MUST BE SELECTED FOR TIME FUNCTIONS'/)
        RETURN
3       CONTINUE
        ITFLG=1
        CALL GETB(15)
        CALL OTB(IB(3,15),I1)
        CALL OTB(IB(4,15),I2)
        DO 5 I=1,15
        IBT(I)=I1(17-I)
5       IBT(I+15)=I2(17-I)
        DO 8 I=1,30
C       BUILD TABLE OF CURRENT STATE
8       IBTB(I)=IBT(I)
        IRFLG=0 ! RANDOM TESTING NOT INVOKED
10      TYPE 15
15      FORMAT(' ENTER TIME OPTION CODE --- '$)
        ACCEPT 20,ICODE
20      FORMAT(A2)
        IF(ICODE.EQ."42523) GO TO 100 ! SE
        IF(ICODE.EQ."54105) GO TO 200 ! EX
        IF(ICODE.EQ."43103) GO TO 300 ! CF
        IF(ICODE.EQ."51123) GO TO 400 ! SR
        IF(ICODE.EQ."47122) GO TO 500 ! RN
        TYPE 25
25      FORMAT(' THE TIME OPTIONS ARE:'/
        15X,'SE --- SELECT FUNCTIONS'/,
        25X,'SR --- SET/RESET SEQUENCE'/,
        35X,'CF --- CURRENT FUNCTIONS'/,
        45X,'RN --- RANDOM TEST SEQUENCE'/,
        55X,'EX --- EXECUTE SE  SR  SEQUENCE'/)
        GO TO 10
100     TYPE 110
110     FORMAT(' ENTER NUMBER OF FUNCTIONS WANTED --- '$)
        ACCEPT 115,NFUNC
115     FORMAT(I5)
        IF(NFUNC.GT.0.AND.NFUNC.LT.62) GO TO 130
        TYPE 120
120     FORMAT(' NUMBER OF FUNCTIONS MUST BE 1 TO 61'/)
        GO TO 100
130     DO 150 I=1,NFUNC
135     TYPE 140,I
140     FORMAT(' ENTER FUNCTION',I3,' --- '$)
        ACCEPT 115,ISEQ(I)
        IF(ISEQ(I).GT.-31.AND.ISEQ(I).LT.32) GO TO 150
        TYPE 145
145     FORMAT(' FUNCTIONS ARE IN THE RANGE -30 TO +31'/)
        GO TO 135
150     CONTINUE
        GO TO 10
200     CONTINUE
        NCYCLE=0
        TOTAL=0
        ITIME=2
        GO TO 600
```

```
205     CONTINUE
        DO 250 I=1,NFUNC
        IBIT=IABS(ISEQ(I))
        IFUNC=ISIGN(1,ISEQ(I))
        CALL TABLE
        CALL OUTPUT(IWA,IWB)
        CALL READ(1)
        CALL OTB(IB(3,15),I1)
        CALL OTB(IB(4,15),I2)
        DO 210 J=1,15
        IBT(J)=I1(17-J)
210     IBT(J+15)=I2(17-J)
C       UPDATE OCTAL TABLE
        DO 220 NN=1,8
220     ITBL(NN,15)=IB(NN,15)
        CALL CHECK
260     NCYCLE=NCYCLE+1
        IF(NCYCLE.EQ.1000) GO TO 600
250     CONTINUE
        IF(IRFLG.EQ.1) GO TO 540
        GO TO 205
300     DO 310 NN=1,NFUNC
310     TYPE 320,NN,ISEQ(NN)
320     FORMAT(' FUNCTION',I3,' IS A',I5)
        GO TO 10
400     NFUNC=61
        DO 410 I=2,60,2
410     ISEQ(I)=I/2
        DO 420 I=1,61,2
420     ISEQ(I)=(1-I)/2
        GO TO 10
500     ITIME=1
        NFUNC=61
        IRFLG=1 ! RANDOM TEST INVOKED
        TYPE 510
510     FORMAT(' ENTER PERCENTAGE OF DON''T CARES --- '$)
        ACCEPT 520,CENT
520     FORMAT(F10.0)
        IF(CENT.GE.0.0.OR.CENT.LE.99.0) GO TO 540
        TYPE 530
530     FORMAT(' % MUST BE 0.0 TO 99.0 '/)
        GO TO 500
540     DO 570 I=1,61
        RANGE=61./(1.-.01*CENT)
        ISEQ(I)=IFIX(RANGE*RAN(L,M))-30
570     CONTINUE
        GO TO (200,205) ITIME
600     TOTAL=TOTAL+NCYCLE
        NCYCLE=0
        TYPE 610,TOTAL
610     FORMAT(' NUMBER OF TESTS = 'E10.4,5X,$)
        CALL CLOCK
        GO TO 250
        RETURN
        END
C
C       CHECK FOR CORRECT OPERATION OF SET & RESET FUNCTIONS
```

```
C
      SUBROUTINE CHECK
      COMMON/ICOM/IBIT,IFUNC,IWA,IWB,ITFLG
      COMMON/ICK/I1(16),I2(16),IBT(30),IBTB(30)
      IF(IBIT.EQ.0) GO TO 100 ! ALL CLEAR
      ICH=0
      DO 10 I=1,30
      IF(IBT(I).NE.IBTB(I)) GO TO 50
10    CONTINUE
      IF(ICH.EQ.1.OR.IBIT EQ.31) RETURN
      ICW=IBTB(IBIT)+IFUNC
      IF(ICW.EQ.2.OR.ICW.EQ.-1) RETURN
      IF(ICW.EQ.1) GO TO 40
      TYPE 35,IBIT,IWB
35    FORMAT(' CANNOT RESET FUNCTION',I3,', PHASE-B ='O10/)
      RETURN
40    TYPE 45,IBIT,IWB
45    FORMAT(' CANNOT SET FUNCTION',I3,', PHASE-B ='O10/)
      RETURN
50    IF(I.EQ.IBIT) GO TO 70
      IWH=IBT(I)-IBTB(I)
      IBTB(I)=IBT(I) ! UPDATE TABLE FOR BIT I.NE.IBIT
      IPT=I*IWH ! WHAT HAPPENED
      IPC=IBIT*IFUNC ! CAUSED BY
      TYPE 55,IPT,IPC,IWB
55    FORMAT(' FUNCTION',I5,' WAS CAUSED BY',I5,', PHASE-B =',O10/)
      GO TO 10
70    ICH=1 ! CORRECT BIT CHANGED
      IBTB(IBIT)=IBT(I) ! UPDATE TABLE
      GO TO 10
100   DO 110 I=1,30
      IF(IBT(I).NE.0) GO TO 120
110   IBTB(I)=IBT(I) ! UPDATE ENTIRE TABLE
      RETURN
120   TYPE 125,I,IWB
125   FORMAT(' FUNCTION',I3,' NOT RESET BY MASTER CLEAR, PHASE-B',
      1' =',O10/)
      GO TO 110
      END
C
C     LOAD PHASE-A WRITE & PHASE-B INTO TA531
C
      SUBROUTINE OUTPUT(IPA,IPB)
      COMMON/ADDR/ICSR,IODR,INDR
      CALL PUT("2,ICSR) ! CSR1=1
      CALL PUT("60000,IODR) ! INHIBIT & DISARM
      CALL PUT("0,ICSR) ! CSR1=0
      CALL BITT(7,ICSR,M1) ! GET REQ-A
      IF(M1.EQ.1) GO TO 20
      STOP ' CANNOT LOAD PHASE-A WRITE , REQ-A IS LOW'
20    CALL PUT(IPA,IODR)
      CALL PUT(IPB,IODR) ! LOAD PHASE-B
      CALL PUT("2,ICSR) ! CSR1=1
      CALL PUT("50000,IODR) ! INHIBIT & STATUS
      CALL GET(INDR,IO) ! GET STATUS WORD IO
      A1=IO/32.
      J=16.*(A1-IFIX(A1)) ! J=# OF PHASE-A IN TA531
      IF(J.EQ.1) GO TO 30
      TYPE 35,IO,J
```

```
35      FORMAT(' STATUS WORD IS ',O6,' # OF COMMANDS =',I3)
        CALL PUT("0,ICSR) ! CSR1=0
        STOP ' ERROR AFTER LOADING PHASE-A WRITE'
30      CALL PUT("0,ICSR) ! CSR1=0
        CALL GET(INDR,I1) ! READ PHASE-A FROM TA531
        CALL GET(INDR,I2) ! READ PHASE-B FROM TA531
        IF(I1.EQ.IPA) GO TO 40 ! VERIFY PHASE-A
        TYPE 45,IPA,I1
45      FORMAT(' LOADED PHASE-A AS',O9,' CAME BACK AS',O9)
        STOP ' PHASE-A IN ERROR BEFORE EXCUTE'
40      IF(IPB.EQ.I2) GO TO 60
        TYPE 55,IPB,I2
55      FORMAT(' PHASE-B LOADED AS',O9,' CAME BACK AS',O9)
        STOP ' PHASE-B IN ERROR BEFORE EXECUTE'
60      CALL PUT("1,ICSR) ! CSR0=1 EXECUTE
        CALL BITT(7,ICSR,M1) ! GET REQ-A
        IF(M1.EQ.1) GO TO 65
        STOP ' REQ-A .NE.1 AFTER WRITE COMMAND TO BUS'
65      CALL PUT("2,ICSR) !CSR1=1
        CALL PUT("50000,IODR) ! INHIBIT & STATUS
        CALL GET(INDR,IO) ! READ STATUS IO
        A1=IO/32.
        J=16.*(A1-IFIX(A1))
        IF(J.EQ.1) GO TO 70
        TYPE 35,IO,J
        CALL PUT("0,ICSR) 'CSR1=CSR0=0
        STOP ' ERROR IN NUMBER OF WORDS IN TA531 AFTER WRITE EXECUTE'
70      II=IO/1024.
        Z=II/2.
        I2=Z
        M1=(Z-I2)*2.  ! EXTRACT DEVICE ABLE(BIT-10) FROM STATUS
        IF(M1.EQ.0) GO TO 75
        STOP ' DEVICE ABLE -WRITE COMMAND NOT ACCEPTED BY TA533'
75      CALL PUT("0,ICSR) ! CSR1=0
        CALL GET(INDR,I1)
        CALL GET(INDR,I2) ! READ PHASE-B FROM TA531
        IF(I1.EQ.IPA) GO TO 80
        TYPE 85,IPA,I1
85      FORMAT(' AFTER WRITE EXECUTE PHA-A CHANGED FROM',O9,' TO',O9)
        STOP ' ERROR IN TA531'
80      IF(I2.EQ.IPB) GO TO 90
        TYPE 95,IPB,I2
95      FORMAT(' AFTER WRITE EXECUTE PHA-B CHANGED FROM,'O9,' TO',O9)
        STOP ' ERROR IN TA531'
90      RETURN
        END
*
```

References

1. R. G. Bradley, Function Matrix and Status Interface, TA533, Sandia Laboratories, Albuquerque, New Mexico 87115

2. RT-11 System Reference Manual, DEC-11-ORUGA-A-D, Digital Equipment Corporation, Maynard, Massachusetts, September 1973.

3. L. Van Blaricum, TA531 Computer/Command Bus Interface, Internal Memorandum, Division 1241, Sandia Laboratories, Albuquerque, New Mexico 87115.

4. T. D. Sullivan, TA531 Computer/Command Bus Interface Test Program, Sandia Laboratories, Albuquerque, New Mexico, SAND75-0589, February 1976.

DISTRIBUTION:

1115 J. L. Benson
1115 D. L. Caskey
1115 L. W. Ebinger
1115 G. J. Hansen
1115 R. G. Oliveira
1115 P. L. Wehrman (2)
1115 D. C. Browne
1241 L. Van Blaricum
1245 R. G. Bradley
1245 R. F. Glaser
1353 T. D. Sullivan (10)
8266 E. A. Aas (2)
3141 C. A. Pepmueller (Actg) (5)
3151 W. L. Garner (3)
For ERDA/TIC (Unlimited Release)
ERDA/TIC (25)
(R. P. Campbell, 3171-1)