

.REN *

IDENTIFICATION

PRODUCT CODE: M1NDEC-11-02TAF-C-0
PRODUCT NAME: TALK CASSETTE ABSOLUTE LOADER (TALDR)
DATE: 21 JANUARY, 1974
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: R. KOLLER

COPYRIGHT (C) 1973, 1974 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

TABLE OF CONTENTS

1. ABSTRACT

2. REQUIREMENTS

Requirements
Listed: SOFTWARE REQUIREMENTS
HARDWARE REQUIREMENTS
ENVIRONMENTAL REQUIREMENTS

3. USE PROCEDURE

Installation
Procedure:
1. LOAD FROM UNIT C. VERIFY ROM IS INSTALLED.
2. LOAD FROM UNIT D. VERIFY ROM IS INSTALLED.
3. LOAD FROM UNIT A. VERIFY ROM IS INSTALLED.
4. LOAD FROM UNIT B. VERIFY ROM IS INSTALLED.
5. PROGRAM AFTER EXECUTING PREVIOUSLY LOADED PROGRAM
DIAGNOSTIC MODE OPERATION

4. NOTES

5. DESCRIPTION

Part
No: BOOTSTRAP LOADER (CBOT)
TALM ABS LOADER

APPENDIX A. TALI BOOTSTRAP LOADER LISTING.

APPENDIX B. TALI BOOTSTRAP LOADER TOGGLE-IN LISTING

1. ABSTRACT

TALDR IS A LOADER PROGRAM DESIGNED TO LOAD MAINDEC-11 DIAGNOSTIC PROGRAMS FROM TALL CASSETTES.

THE TALDR LOADER IS THE FIRST FILE STORED ON A TALL MAINDEC-11 CASSETTE. IT IS LABELLED "TALDR.SYS". IT IS FOLLOWED BY ONE OR MORE MAINDEC-11 DIAGNOSTIC PROGRAMS STORED IN ABS LOADER FORMAT.

EACH TALL MAINDEC-11 CASSETTE IS PROVIDED WITH A DIRECTORY, THAT LISTS THE TALDR LOADER, AND ALL THE PROGRAMS STORED IN THE CASSETTE.

EACH FILE NAME IN THE CASSETTE IS NUMBERED WITH AN OCTAL SEQUENCE NUMBER. TALDR HAS THE SEQUENCE NUMBER OF 1. THE FILE NUMBER IS USED TO SPECIFY TO THE TALDR LOADER WHICH PROGRAM IS TO BE LOADED.

IN ORDER TO PREVENT ACCIDENTAL ERASURE OF CASSETTES, EACH CASSETTE TO BE LOADED FROM, SHOULD BE "WRITE-LOCKED". WRITE LOCKING OF A CASSETTE IS ACCOMPLISHED BY UNCOVERING THE SMALL HOLES AT THE BACK OF THE CASSETTE BY FLIPPING THE SMALL PLASTIC TABS OUT OF THE WAY.

ABILITY OF THE TALDR LOADER TO SUCCESSFULLY LOAD A PROGRAM DEPENDS ON THE SIZE OF THE PROGRAM TO BE LOADED. TALDR CAN NOT LOAD A PROGRAM THAT INFRINGES ON ITS OWN STORAGE AREA.

2. REQUIREMENTS

2.1 HARDWARE REQUIREMENTS

- A. A PDP11 PROCESSOR
- B. 8000-YH CASSETTE BOOTSTRAP ROM (MAY BE SIMULATED BY A TALL BOOTSTRAP LOADER PROGRAM)
- C. TALL TAPE CASSETTE CONTROL UNIT
- D. TALL TAPE CASSETTE TRANSPORT
- E. 8K MINIMUM STORAGE

2.2 SOFTWARE REQUIREMENTS

- A. MAINDEC-11 DIAGNOSTIC TAPE CASSETTES
- B. TALL BOOTSTRAP LOADER PROGRAM (IF 8000-YH IS NOT INSTALLED) (THIS PROGRAM MUST BE MANUALLY TOGGLED IN)

2.3 STORAGE REQUIREMENTS

ONCE LOADED AND RELOCATED, THE TALDR LOADER OCCUPIES THE UPPERMOST 288(10) WORDS OF AVAILABLE MEMORY, UP TO 28K.

3. USE PROCEDURE

3.1 LOADING FROM UNIT D, IBM®2814 ROM INSTALLED

IN THE STEPS THAT FOLLOW, ANY HALT AT OTHER THAN LOC X07014 IS AN ERROR. REFER TO SECTION 4, HALTS, FOR HALT INFORMATION.

- A. MOUNT WRITE-LOCKED CASSETTE IN UNIT D. (LEFT HAND DRIVE).
- B. LOAD ADDRESS 172300 (IBM®2814 START ADDRESS)
- C. SET SB WITH FILE NUMBER OF PROGRAM TO BE LOADED. REFER TO CASSETTE DIRECTORY. FILE NUMBER MUST BE GREATER THAN 1.
- D. PRESS START
- E. THE CASSETTE IS REWOUND. TADR IS LOADED, RELOCATED, AND PROGRAM HALTS AT LOC X07014.
- F. THE HALT AT LOC X07014 INDICATES THAT TADR HAS SUCCESSFULLY LOADED, AND IS READY TO LOAD A PROGRAM. SINCE YOU HAVE ALREADY SET THE SB TO DESIRED PROGRAM NUMBER, PRESS CONT TO LOAD THE PROGRAM.
- G. THE CASSETTE IS REWOUND. TADR SEARCHES FORWARD FOR THE DESIRED PROGRAM, AND LOADS IT.
- H. IF THE PROGRAM JUST LOADED IS SELF-STARTING, THE PROGRAM IS STARTED. IF NOT, THE PROGRAM HALTS AT LOC X07014, INDICATING A SUCCESSFUL LOAD AND READINESS TO LOAD ANOTHER PROGRAM. AT THIS POINT, THE PROGRAM JUST LOADED CAN BE STARTED MANUALLY (JUST PUSHC-1), PROGRAMS START AT LOC X02000), OR ANOTHER PROGRAM CAN BE LOADED.
- I. TO LOAD ANOTHER PROGRAM, SET THE SB TO FILE NUMBER OF DESIRED PROGRAM, AND PRESS CONT. GO TO STEP E.

3.2 LOADING FROM UNIT 1. 181792YH ROM INSTALLED

IN THE STEPS THAT FOLLOW, ANY HALT OTHER THAN AT LOC X07014 IS AN ERROR. REFER TO SECTION 4. HALTS.

- A. MOUNT WRITE-LOCKED CASSETTE ON UNIT 1. (RIGHT HAND DRIVE).
- B. DEPOSIT 177500 INTO RD (SEE NOTE BELOW)
- C. DEPOSIT 00040D INTO LOCATION 177500 (SELECTS UNIT 1) (SEE NOTE BELOW)
- D. DEPOSIT 173306 IN PC (R7) (181792YH START ADDRESS+6)
- E. SET SR WITH FILE NUMBER OF PROGRAM TO BE LOADED. REFER TO CASSETTE DIRECTORY. FILE NUMBER MUST BE GREATER THAN 1.
- F. PRESS CONT
- G. THE CASSETTE IS REMOVED, TADR IS LOADED, RELOCATED, AND PROGRAM HALTS AT LOC X07014
- H. THE HALT AT LOC X07014 INDICATES THAT TADR HAS SUCCESSFULLY LOADED, AND IS READY TO LOAD A PROGRAM. SINCE YOU HAVE ALREADY SET THE SR WITH THE DESIRED FILE NUMBER (STEP E), PRESS CONT TO LOAD PROGRAM.
- I. THE CASSETTE IS REMOVED. TADR SEARCHES FORWARD FOR THE DESIRED PROGRAM, AND LOADS IT.
- J. IF THE PROGRAM JUST LOADED IS SELF-STARTING, THE PROGRAM IS STARTED. IF NOT, THE PROGRAM HALTS AT LOC X07014, INDICATING A SUCCESSFUL LOAD, AND READINESS TO LOAD ANOTHER PROGRAM. AT THIS POINT, THE PROGRAM JUST LOADED CAN BE STARTED MANUALLY (PUSH PAIRED-1) PROGRAMS START AT LOC 000200), OR ANOTHER PROGRAM CAN BE LOADED.
- K. TO LOAD ANOTHER PROGRAM, SET THE SR TO FILE NUMBER OF DESIRED PROGRAM, AND PRESS CONT. GO TO STEP G.

NOTE

- STEP B. FOR TALI WITH NON-STANDARD ADDRESS, DEPOSIT NON-STANDARD ADDRESS IN RD.
STEP C. FOR TALI WITH NON-STANDARD ADDRESS, DEPOSIT 00040D IN NON-STANDARD ADDRESS.

3.3 LOADING FROM UNIT D. NO BY785VH ROM

IN THE STEPS THAT FOLLOW, ANY HALT AT OTHER THAN LOC XX7D14 IS AN ERROR. REFER TO SECTION 4, HALTS.

- A. MOUNT WRITE-LOCKED CASSETTE ON UNIT D. (LEFT HAND DRIVE).
- L. TOGGLE-IN (DEPOSIT) THE 28 WORD SOFTWARE BOOT SHOWN IN APPENDIX B, STARTING AT LOCATION XX7D06.
- C. LOAD ADDRESS XX7706
- D. SET SR WITH THE FILE NUMBER OF PROGRAM TO BE LOADED. REFER TO CASSETTE DIRECTORY. FILE NUMBER MUST BE GREATER THAN 1.
- E. PRESS START
- F. THE CASSETTE IS REWOUND, TALDR IS LOADED, RELOCATED, AND PROGRAM HALTS AT LOC XX7D14.
- G. THE HALT AT LOC XX7D14 INDICATES THAT TALDR WAS SUCCESSFULLY LOADED, AND IS READY TO LOAD A PROGRAM. SINCE YOU HAVE ALREADY SET THE SR TO THE DESIRED PROGRAM NUMBER, PRESS CONT TO LOAD THE PROGRAM.
- H. THE CASSETTE IS REWOUND, TALDR SEARCHES FORWARD FOR THE DESIRED PROGRAM, AND LOADS IT.
- I. IF THE PROGRAM JUST LOADED IS SELF STARTING, THE PROGRAM IS STARTED. IF NOT, THE PROGRAM HALTS AT LOC XX7D14, INDICATING A SUCCESSFUL LOAD AND READINESS TO LOAD ANOTHER PROGRAM. AT THIS POINT, THE PROGRAM JUST LOADED CAN BE STARTED MANUALLY (NOT DRINDEC-1), PROGRAMS START AT LOC D02200), OR ANOTHER PROGRAM CAN BE LOADED.
- J. TO LOAD ANOTHER PROGRAM, SET THE SR TO FILE NUMBER OF DESIRED PROGRAM, AND PRESS CONT. GO TO STEP H.

3.4 LOADING FROM UNIT 1. NO BROADCAST ROM

IN THE STEPS THAT FOLLOW, ANY HALT AT OTHER THAN AT LOC XXXD14 IS AN ERROR. REFER TO SECTION 4, HALTS.

- A. MOUNT WRITE-LOCKED CASSETTE ON UNIT 1. (RIGHT HAND DRIVE).
- B. TOGGLE-IN (DEPOSIT IN CORE) THE 26 WORD SOFTWARE BOOT SHOWN IN APPENDIX B. STARTING AT LOC XXX776
- C. DEPOSIT 177500 INTO RO (SEE NOTE BELOW)
- D. DEPOSIT 000400 INTO LOCATION 177500 (SELECTS UNIT 1) (SEE NOTE BELOW)
- E. DEPOSIT XXX714 IN PC (R7)
- F. SET SR WITH FILE NUMBER OF PROGRAM TO BE LOADED. REFER TO CASSETTE DIRECTORY. FILE NUMBER MUST BE GREATER THAN 1.
- G. PRESS CONT
- H. THE CASSETTE IS REWIND. TALKER IS LOADED AND RELOCATED, AND THE PROGRAM HALTS AT LOC XXXD14.
- I. THE HALT AT LOC XXXD14 INDICATES THAT TALKER WAS LOADED SUCCESSFULLY, AND IS READY TO LOAD A PROGRAM. SINCE YOU HAVE ALREADY SET THE SR WITH THE DESIRED PROGRAM NUMBER, PRESS CONT TO LOAD THE PROGRAM.
- J. THE CASSETTE IS REWIND. TALKER SEARCHES FORWARD FOR THE DESIRED PROGRAM, AND LOADS IT.
- K. IF THE PROGRAM JUST LOADED IS SELF-STARTING, THE PROGRAM IS STARTED. IF NOT, THE PROGRAM HALTS AT LOC XXXD14, INDICATING A SUCCESSFUL LOAD AND READINESS TO LOAD ANOTHER PROGRAM. AT THIS POINT THE PROGRAM JUST LOADED CAN BE STARTED MANUALLY (NOT CHINOOK-11 PROGRAMS START AT LOC 000200), OR ANOTHER PROGRAM CAN BE LOADED.
- L. TO LOAD ANOTHER PROGRAM, SET THE SR TO FILE NUMBER OF DESIRED PROGRAM, AND PRESS CONT. GO TO STEP J.

NOTE

STEP C. FOR TALK WITH NON-STANDARD ADDRESS, DEPOSIT NON-STANDARD ADDRESS IN RO.

STEP D. FOR TALK WITH NON-STANDARD ADDRESS, DEPOSIT 000400 IN NON-STANDARD ADDRESS.

3.6 LOADING A PROGRAM AFTER EXECUTING PREVIOUSLY LOADED PROGRAM

IT IS POSSIBLE TO LOAD A PROGRAM FROM CASSETTE AFTER A PREVIOUSLY LOADED PROGRAM HAS EXECUTED, PROVIDED THAT THE PREVIOUS PROGRAM HAS NOT DESTROYED THE TALDR LOADER. TO LOAD A PROGRAM:

- A. MOUNT WRITE-LOCKED CASSETTE ON SAME UNIT AS PREVIOUSLY USED. (TALDR SAVED THE TAII ADDRESS, AND UNIT NUMBER USED IN PREVIOUS LOAD).
- B. LOAD ADDRESS X07776
- C. PRESS START
- D. PROGRAM SHOULD HALT AT LOC X07014. IF ANYTHING ELSE HAPPENS, TALDR HAS BEEN DESTROYED. USE ONE OF THE LOADING PROCEDURES DESCRIBED IN SECTIONS 3.1 THROUGH 3.4
- E. SET SH WITH FILE NUMBER OF PROGRAM TO BE LOADED. REFER TO CASSETTE DIRECTORY. FILE NUMBER MUST BE GREATER THAN 1.
- F. PRESS CONT
- G. THE CASSETTE IS REWINDING, TALDR SEARCHES FORWARD FOR DESIRED PROGRAM AND LOADS IT.
- H. IF THE PROGRAM JUST LOADED IS SELF-STARTING THE PROGRAM IS STARTED. IF NOT, THE PROGRAM HALTS AT LOC X07014. INDICATING A SUCCESSFUL LOAD AND READINESS TO LOAD ANOTHER PROGRAM. AT THIS POINT THE PROGRAM JUST LOADED CAN BE STARTED MANUALLY, OR ANOTHER PROGRAM CAN BE LOADED.

3.6 DIAGNOSTIC MODE OPERATION

TALDR HAS A DIAGNOSTIC MODE, DESIGNED TO AID THE MAINTENANCE ENGINEER IN DIAGNOSING TALI PROBLEMS. DIAGNOSTIC MODE OPERATES DURING LOADING AND RELOCATING OF THE TALDR LOADER, AND IS ACTIVATED BY SETTING THE SR TO 00000. INSTEAD OF THE FILE NUMBER OF A PROGRAM TO BE LOADED, THE SEQUENCE THAT FOLLOWS ASSUMES USE OF THE SH795VM, TALI CONTROL AT STANDARD ADDRESS, AND USE OF UNIT 0.

- A. MOUNT WRITE-LOCKED CASSETTE ON UNIT 0. (LEFT HAND DRIVE).
- B. LOAD ADDRESS 1733C0
- C. SET SR TO 000000 (INDICATES DIAGNOSTIC MODE).
- D. PRESS START
- E. CASSETTE REWINDS, TAPE MOVES FORWARD AND STOPS. PROGRAM HALTS AT LOC 000010. THIS IS THE "POST BOOT" HALT. AT THIS POINT THE FIRST 64 WORDS OF THE TALDR LOADER SHOULD BE STORED IN MEMORY. STARTING AT LOC 000020, THESE 151 64 WORDS ARE REFERRED TO AS THE "PRE-LOADER". THE PRE-LOADER CONTENTS CAN BE VERIFIED BY EXAMINING CORE STARTING AT LOC 000000, AND REFERENCING THE TALDR LISTING. AFTER VERIFICATION, (OR IF NOT VERIFYING), PRESS CONT.
- F. THE CASSETTE REWINDS, MOVES FORWARD, AND THEN HALTS AT LOC 000132. THIS OPERATION IS PERFORMED BY THE "PRE-LOADER". THE 64 WORDS OF CODE READ IN DURING STEP E. IF THE CASSETTE ACTS WING (DOES NOT STOP, DOES NOT EVEN GET STARTED), THE PRE-LOADER CODE SHOULD BE EXAMINED FOR CORRECT OPERATION. WHEN THE PROGRAM HALTS AT LOC 000132, THE ENTIRE TALDR LOADER HAS BEEN READ INTO LOWER CORE. THE HALT OCCURS JUST PRIOR TO RELOCATING THE TALDR LOADER TO UPPER MEMORY. RV SHOULD CONTAIN THE VALUE X06714, WHICH IS THE INITIAL ADDRESS WHERE TALDR IS TO BE RELOCATED. PRESS CONT.
- G. TALDR IS RELOCATED TO UPPER MEMORY AND THE PROGRAM HALTS AT LOC X07014, WHICH IS THE "READY TO LOAD" HALT. AT THIS POINT A PROGRAM NUMBER MAY BE SET IN THE SR TO LOAD A PROGRAM. RD SHOULD CONTAIN THE ADDRESS OF TALI CONTROL (IN THIS CASE 177500). IF LOADING A PROGRAM, PRESS CONT.

4. HALTS

IN THE HALT DESCRIPTIONS THAT FOLLOW, HALTS ARE IDENTIFIED BY THE LOCATION OF THE HALT IN MEMORY, AND NOT BY ANY CONSOLE DISPLAY LIGHTS. IN ORDER TO CORRECTLY IDENTIFY A HALT ON THE VARIOUS PDP-11 PROCESSORS, USE THE TABLE BELOW:

<u>CPU USED</u>	<u>ACCESS LIGHTS DISPLAY</u>
11/20	HALT LOCATION
11/05	HALT LOCATION+2
11/45	HALT LOCATION+2
11/40	HALT LOCATION+2
LOC 000016	POST BOOT HALT. OCCURS AFTER 1ST BLOCK OF TALI HAS BEEN READ BY HARDWARE OR SOFTWARE BOOT AND BRAC (DIAGNOSTIC MODE OPERATION). PRESS CONT TO PROCEED WITH TALI LOAD OPERATION.
LOC 000132	TALI PRE-RELOCATION HALT. OCCURS IN DIAGNOSTIC MODE ONLY. INDICATES THAT THE ENTIRE TALI LOADER HAS BEEN READ INTO CORE AND THE PROGRAM IS READY TO RELOCATE THE TALI LOADER TO UPPER CORE. PRESS CONT TO ACCOMPLISH RELOCATION.
LOC 000172	TALI HARDWARE ERROR DETECTED IN PRE-LOADER SECTION OF TALI LOADER (1ST 54 WORDS OF THE LOADER). PRESS CONT TO TRY AGAIN.
LOC X07014	GOOD LOAD HALT. READINESS TO LOAD A PROGRAM. THE TALI LOADER RESIDES IN UPPER MEMORY AND IS READY TO LOAD A PROGRAM. THIS HALT ALSO OCCURS AFTER TALI HAS SUCCESSFULLY LOADED A PROGRAM, AND IS READY TO LOAD ANOTHER PROGRAM.
LOC X07026	ERROR. THE USER HAS SPECIFIED A PROGRAM FILE NUMBER THAT IS LESS THAN 2. THE .61 FILE (TALI) IS NOT A VLD PROGRAM FOR LOADING, AS IT IS NOT IN ABSOLUTE FORMAT. ALSO IF THE BR IS SET TO 0, TALI DOES NOT KNOW WHAT TO DO ABOUT THAT. SET CORRECT FILE NUMBER IN SR AND PRESS CONT TWICE.
LOC X07060	ERROR. SENTINEL FILE FOUND. THE END OF WRITTEN TAPE HAS BEEN FOUND BEFORE THE FILE SPECIFIED BY USER. INCORRECT FILE NUMBER HAS BEEN SPECIFIED. SET THE CORRECT FILE NUMBER IN SR AND PRESS CONT TWICE.

(4. CONT'D)

- LOC X07164 CHECKSUM ERROR. TADLR HAS DETECTED A CHECKSUM ERROR WHILE READING THE PROGRAM. TO TRY AGAIN PRESS CONT. TADLR REVERSES TO START OF PROGRAM AND TRIES AGAIN. IF AFTER SEVERAL TRIES THE MFLY STILL OCCURS, THE PROGRAM IS NOT LOADABLE. GO TO SECTION 3.5 TO LOAD ANOTHER PROGRAM IF DESIRED.
- LOC X07210 TADLR LOADER OVERRUN. THE PROGRAM IN PROCESS OF BEING LOADED IS ENTERING INTO THE TADLR STORAGE SPACE. DUE TO ITS SIZE, THE PROGRAM CAN NOT BE LOADED COMPLETELY WITH THE SYSTEM'S CURRENT STORAGE CAPACITY.
- LOC X07462 TALI ERROR. CASSETTE ERROR HAS BEEN DETECTED BY THE TADLR LOADER WHILE PERFORMING CASSETTE OPERATIONS. EXAMINE CONTENTS OF TALI CONTROL REGISTER TO DETERMINE CAUSE OF ERROR. TO TRY AGAIN, PRESS CONT TWICE.
- LOC X00776 SOFTWARE BOOT TALI ERROR. HARDWARE ERROR DETECTED BY SOFTWARE BOOT. OR 1ST BYTE OF THE BLOCK READ BY THE BOOT IS NOT A 2ND. INDICATION: THAT THE FIRST PROGRAM IN THE CASSETTE IS NOT THE TADLR LOADER. IF HARDWARE ERROR (EXAMINE TALI CSR), PRESS CONT TO TRY AGAIN. IF NOT HARDWARE ERROR, LOAD THE PROPER CASSETTE, AND PRESS CONT.
- LOC 173360 TALI ERROR, OR 1ST BYTE OF DATA READ FROM CASSETTE'S 1ST BLOCK NOT A 2ND. DETECTED BY INTERRUPT ROM. IF HARDWARE ERROR (EXAMINE TALI CSR), PRESS CONT TO TRY AGAIN. IF NOT HARDWARE ERROR, LOAD THE CORRECT CASSETTE, AND PRESS CONT.

E. DESCRIPTION

E.1 BOOTSTRAP LOADER (CBOOT)

THIS PROGRAM IS SPECIFICALLY DESIGNED TO READ IN THE FIRST BLOCK OF THE TALKER ABSOLUTE LOADER PROGRAM. VERIFY THAT THE BLOCK READ IN IS THE 1ST BLOCK OF TALKER, AND TRANSFER PROGRAM CONTROL TO THE TALKER PROGRAM. IT IS AVAILABLE IN TWO VERSIONS: 1) HARD WIRED INTO THE 80732-YH RCT OR 2) AS A SOFTWARE COPY THAT MAY BE DEPOSITED BY THE OPERATOR. REFER TO APPENDIX A FOR A COPY OF THE BOOT LOADER PROGRAM.

REGISTER USAGE:

- RO - CONTAINS THE ADDRESS OF THE TALKER CONTROL AND STATUS REGISTER
- R1 - USED AS A POINTER TO THE COMMAND TABLE THAT HOLDS THE REQUIRED TALKER CASSETTE COMMANDS
- R2 - USED AS A DATA BUFFER POINTER AND DATA FLAG
- R3 - HOLDS BIT TEST MASK REQUIRED TO QUERY THE TALKER CONTROL AND STATUS REGISTER

PROGRAM SEQUENCE:

- CBOOT - THE FIRST INSTRUCTION LOADS RO WITH THE TALKER ADDRESS, FOLLOWED BY A CLEAR OF THE TALKER TO INSURE SELECTING UNIT D.
- RESRTR - THE MOVE FOLLOWED BY THE ADD SETS UP R1 TO POINT TO THE FIRST BYTE IN THE COMMAND TABLE. THESE TWO INSTRUCTIONS RELATE THE LOCATION OF THE COMMAND TABLE TO THE PC WHICH INSURES THAT THE BOOT PROGRAM IS RELOCATABLE. FOLLOWING THIS, R2 IS LOADED WITH A CONSTANT OF 375(8) WHICH WILL FIRST SERVE TO DETECT THE BEGINNING OF THE DATA XFR AND THEN BE USED AS A DATA BUFFER POINTER TO LOAD 128 BYTES INTO CORE STARTING AT LOC. D. R3 IS LOADED WITH THE FIRST BYTE IN THE COMMAND TABLE (A 801(8) WHICH ALIGNS TESTING THE "READY" AND "TRANSFER" FLAG IN THE TALKER) AND R1 AUTOMATICALLY UPDATES TO POINT TO THE SECOND BYTE. AFTER AN ERROR HALT WILL RESTART THE LOADER AT "RESRTR".

(5.1 CONT'D)

LOOP1 - THE NEXT COMMAND IN THE COMMAND TABLE IS LOADED INTO THE TAPDR AND R1 UPDATED TO POINT TO THE NEXT COMMAND. THIS LOOP IS EXECUTED FIVE TIMES TO FETCH THE FOLLOWING COMMANDS:

- 1) RETURN TO BEGINNING OF TAPE
- 2) SKIP FORWARD OVER HEADER BLOCK
- 3) READ FWD ONE BLOCK 1128 BYTES TRANSFERRED INTO CORE LOC 0 TO 177)
- 4) CHECK THE CRC BYTES
- 5) TERMINATE (HIGH ORDER BIT OF COMMAND BYTE IS SET)

AFTER LOADING EACH COMMAND, A BIT1 TESTS FOR THE TERMINATOR AND TRANSFERS CONTROL TO "DONE" IF THE COMMAND BYTE IS NEGATIVE. IF NOT NEGATIVE THE PROGRAM FALLS THROUGH TO "LOOP2".

LOOP2 - A BIT1 FOLLOWED BY A BEQ TESTS FOR SUCCESSFUL COMPLETION OF THE CURRENT COMMAND. THE PROGRAM BRANCHES ON "LOOP2" UNTIL EITHER "READY" OR "TRANSFER READY" IS ASSERTED. IF ONE OR BOTH OF THESE FLAGS ARE ASSERTED THE PROGRAM FALLS THROUGH AND INCREMENTS R2 (NOTE THAT R2 IS INCREMENTED WITH A BYTE INSTRUCTION AND STARTED AT 205) UNTIL THE "READ" COMMAND IS FETCHED. THE PROGRAM LOOPS ON "LOOP1" VIA A BIT1. AFTER THE READ COMMAND IS LOADED R2 INCREMENTS TO 000000 (HIGH ORDER CARRIES 1) PROPAGATED BEYOND BIT 7) AND THE PROGRAM FALLS THROUGH TO THE MOVE THAT LOADS DATA INTO CORE. IF THE FIRST CHARACTER READ IS NOT A 20018 (INDICATING THE TAPDR PROGRAM) THE PROGRAM BRANCHES TO A HALT AT "STOP". IF THE FIRST BYTE IS A 20018) THE PROGRAM LAPS ON "LOOP2" UNTIL 128 BYTES ARE TRANSFERRED INTO CORE AT WHICH TIME R2 GOES TO 00018) (BECOMES NEGATIVE AGAIN). AT THIS TIME THE PROGRAM RETURNS TO "LOOP1" TO FETCH THE 1128 COMMAND FOLLOWED BY A TERMINATOR COMMAND.

(8.1 CONT'D)

- STOP - PROGRAM BRANCHES TO A HALT HERE IF THE 1ST
BYTE TRANSFERRED IS 0 CODE IS NOT A 0010
ADDRESSING CONTINUE CAUSES PROGRAM TO BRANCH TO
"RESTRI".
- DONE - PROGRAM BRANCHES HERE AFTER FETCHING COMMAND
TABLE TERMINATOR. IT THEN TESTS FOR CASSETTE
FOR ERRORS AND CLEARS THE PC TO TRANSFER
CONTROL TO LOCATION 8. IF NO ERRORS IF AN ERROR
IS DETECTED THE PROGRAM BRANCHES TO "STOP" AND HALTS.

TABLE - COMMAND TABLE:

TABLE=0	/	0000	TEST FCR READY OR XFR REG
•1	/	0001	RESTRI
•2	/	0002	SPACE FWD BLOCK
•3	/	0003	READ
•4	/	0004	TEST - TEST CRC
•5	/	0005	TERMINATOR
•6	/	0006	TWO WORDS OF FILLER
•7	/	0007	
•8	/	0008	
•9	/	0009	
•10	/	0010	
•11	/	0011	

- VECTOR - IN SOME OF THE PDP11 PROCESSORS THE POWER FAIL VECTOR
IS DESIGNED TO POINT TO A HARDWARE ROM. THE LAST
16 BYTES OF THE SCOT LOADER POINT THE PC TO
THE START OF THE SCOT LOADER AND SET THE
PROCESSOR PRIORITY TO 7 IF THE POWER FAIL VECTOR
IS DESIGNED TO POINT TO 77320. THESE TWO WORDS
AND THE TWO FILLER WORDS IN THE COMMAND TABLE
ARE NOT REQUIRED IN THE SOFTWARE VERSION OF THE
SCOT LOADER.

5.2 TADR TAPE CASSETTE ABSOLUTE LOADER

TADR IS DESIGNED TO LOAD "ABS" FORMAT PROGRAMS FROM MAINDEC-11 TALI CASSETTES. THE PROGRAM CONSISTS OF TWO SECTIONS. FIRST, A 128 BYTE (164 WORDS) "PRE-LOADER" WHOSE FUNCTION IS TO READ IN THE ENTIRE TADR LOADER INTO LOWER CORE AND THEN RELOCATE IT TO THE TOP OF MEMORY. SECOND, THE CASSETTE LOADER ITSELF, WHOSE FUNCTION IS TO LOAD AN ABS FORMAT PROGRAM FROM CASSETTE WHOSE FILE NUMBER HAS BEEN SET IN THE SR.

5.2.1 PRE-LOADER PROGRAM SEQUENCE

- GO: THE PROGRAM TESTS THE SWITCH REGISTER. IF SR = 0, THE PROGRAM HALTS AND THE OPERATOR MUST DEPRESS CONTINUE TO PROCEED TO "GO1". IF THE SR IS NOT ZERO THE PROGRAM FALLS THROUGH TO "GO1".
- GO1: THE STACK POINTER IS SET UP AND THE CORE MEMORY IS SIZED TO DETERMINE THE HIGHEST POSSIBLE MEMORY ADDRESS. LOCATION 4 IS SET TO POINT TO "DATCH" WHEN A BUS TIMEOUT OCCURS (NON-EX MEM). WHEN THE PROGRAM EXITS THE SIZER ROUTINE RW CONTAINS AN ADDRESS TWO GREATER THAN THE HIGHEST POSSIBLE MEMORY ADDRESS.
- GO*COR: THE PROGRAM NOW BACKSPACES ONE FILE, SPACES FORWARD ONE BLOCK TO SKIP THE HEADER (32 BYTES), AND THEN LOADS THE ENTIRE TADR PROGRAM (6 BLOCKS - 758 BYTES) INTO CORE STARTING AT LOCATION 0. THE FIRST BLOCK WHICH HAD BEEN READ IN BY THE BOOTSTRAP LOADER GETS OVERLAPED. IF ANY ERRORS OCCUR WHILE READING, THE PROGRAM HALTS AT LOC 000172. IF NO ERRORS OCCUR, THE PROGRAM TESTS THE SR AND IF IT IS EQUAL TO 000000, HALTS AFTER LOADING; OTHERWISE THE PROGRAM FALLS THROUGH TO "RELOC".
- RELOC: THIS ROUTINE RELOCATES THE TADR CODE TO THE TOP OF MEMORY STARTING AT "PSTART". AFTER RELOCATION CONTROL IS TRANSFERRED TO "PSTART" AT LOC X16714, THE PROGRAM THEN HALTS AT LOC X17014 TO AWAIT OPERATOR INPUT.
- WAITA: SUBROUTINE CALLED TO ALLOW WAITING FOR THE TAPE CASSETTE TO TERMINATE ITS COMMANDS AND ALSO TO TEST FOR CASSETTE ERRORS.
- ERRORA: PROGRAM BRANCHES TO A HALT HERE IF IT DETECTS ANY CASSETTE ERRORS. INCLUDES BRANCH TO "C" TO RESTART AFTER ERRORS WHEN CONTINUE IS DE-PRESSED.

5.2.2 TALKER PROPER - PROGRAM SEQUENCE

PSTART: START OF CODE THAT SETS UP POINTERS IN ORDER TO MAKE TALKER COMPLETELY POSITION INDEPENDENT.

START: THIS CODE USES POINTERS SET UP BY PSTART, IN PREPARATION TO LOADING A PROGRAM. IT THEN HALTS AT LOC X00014 INDICATING READINESS TO LOAD A PROGRAM.

GETNUM: START OF CODE TO GET FILE NUMBER OF PROGRAM TO BE LOADED. IT READS THE FILE NUMBER FROM THE TAPE. IF THE NUMBER IS LESS THAN TWO (2) AN ERROR HALT OCCURS, AS THE 1ST FILE ON THE CASSETTE (TALKER ITSELF) IS NOT LOADABLE VIA TALKER.

GETFIL: CODE TO GET TO DESIRED PROGRAM FILE. THE SEQUENCE IS:

- A. REWIND THE CASSETTE.
- B. SKIP THE NUMBER OF FILES REQUIRED TO GET TO THE FILE TO BE LOADED. AFTER A FILE IS SKIPPED, THE HEADER FOR THE NEXT FILE IS READ TO VERIFY THAT THE END OF WRITTEN TAPE HAS NOT BEEN REACHED (SENTINEL FILE). IF END OF TAPE IS REACHED, AN ERROR HALT OCCURS, INDICATING THAT AN INVALID FILE NUMBER HAS BEEN SPECIFIED.

LOADFL: THIS CODE IS THE ACTUAL PROGRAM LOADER. SEQUENCE FOLLOWS:

- A. LOAD READ BUFFER ("BUF") WITH ONE BLOCK OF DATA (JSR PC,STDATA).
- B. CLEAR THE CHECKSUM WORD.
- C. GET A BYTE OF DATA UNTIL ONE WITH VALUE OF 1 IS FOUND.
- D. GET ANOTHER BYTE. IT MUST BE 0. IF NOT GO TO B. IF YES, BLOCK START HAS BEEN DETECTED.
- E. GET THE BYTE COUNT. (NUMBER OF BYTES IN BLOCK). 1 WORD.
- F. PUT BYTE COUNT IN RS. DEDUCT 4 FROM IT TO ACCOUNT FOR 2 BYTES OF BLOCK START, AND 2 BYTES OF BYTE COUNT.
- G. SEE IF REMAINDER IS 2. IF 2, THE BLOCK IS ADDRESS TRANSFER BLOCK. GO TO STEP N. IF NOT, IT IS A DATA BLOCK. FALL THROUGH.
- H. GET THE LOAD ADDRESS. 1 WORD. LOAD ADDRESS IS THE ADDRESS WHERE DATA IN BLOCK IS TO BE STORED. LOAD ADDRESS IS IN RS.
- I. NOW GET A DATA BYTE. BRANCH IF BYTE COUNT COUNT IS NOT YET 0. GO TO STEP L.
- J. BYTE COUNT IS 0. TEST THE CHECKSUM. MUST BE 0. IF 0, GO TO STEP B. HALT IF NOT. CHECKSUM ERROR.

(5.2.2 CONT'D)

- K. UPON CONTINUE, REVERSE TO START OF FILE. SKIP THE HEADER, AND GO TO STEP A TO RE-START.
- L. CHECK THAT ADDRESS WHERE DATA IS TO BE STORED HAS NOT EXCEEDED ON TACOR CORE SPACE. IF YES, HALT. UPON CONTINUE, GO TO "START"
- M. STORE DATA BYTE. GO TO STEP I.
- N. GET THE TRANSFER ADDRESS. (SAVED IN R5).
- O. GET ANOTHER BYTE. TEST THE CHECKSUM. IF NOT 0, GO TO J.
- P. SET A 5 IN LOC 41, INDICATING THAT LOAD MEDIUM IS CASSETTE.
- Q. SEE IF TRANSFER ADDRESS (IN R5) IS ODD. IF ODD, THE PROGRAM IS NOT SELF-STARTING. GO TO "START". IF EVEN, PUT R5 IN PC TO START PROGRAM JUST LOADED.

RCFRM: GET NEXT SEQUENTIAL BYTE OF DATA SUBROUTINE. THE BYTE IS RETURNED IN R2. TAKEN FROM READ BUFFER.

RCBFRM: THIS SUBROUTINE ASSEMBLES 2 BYTES OF DATA INTO ONE WORD. THE WORD IS RETURNED IN R5.

RCORR: READ FILE HEADER SUBROUTINE. A FILE HEADER CONSISTS OF A DATA BLOCK OF 32 BYTES. THE HEADER CONTAINS INFORMATION IDENTIFYING THE FILE, SUCH AS NAME, DATE, BLOCK SIZE, ETC. FOR PURPOSES OF THIS, THE HEADER IS CHECKED ONLY TO INSURE THAT THE END OF TAPE HAS NOT BEEN REACHED (SENTINEL FILE; ALL ZERGES).

GTDATA: SUBROUTINE TO LOAD BUFFER WITH ONE BLOCK'S WORTH OF DATA (128 BYTES). FILE DATA IS STORED IN BLOCKS OF 128 BYTES. GTDATA READS 128 BYTES OF DATA INTO A READ BUFFER CALLED "BUF".

WAIT: WAIT SUBROUTINE. AWAITS COMPLETION OF PREVIOUSLY ISSUED CASSETTE FUNCTION AND CHECKS FOR ERRORS. IF AN ERROR OCCURS, A TALL HARDWARE ERROR OCCURS.

APPENDIX A

T-11 BOOTSTRAP LOADER

```

          173300          .ABS
          173300          .W173300
          000000          BC=10
          000000          RI=11
          000000          R2=12
          000000          R3=13
          000000          PC=14
          000000          PC=15

773300 018700 177500 CBOOT: MOV #177500,R0 :R0 HOLDS ADDRESS OF T-11
773300 018700 177500 :MOV R0,R1 :R1 HOLDS ADDR OF COMMAND TABLE
773300 018700 177500 :MOV #0,R2 :R2 HOLDS ADDR OF DATA FLAG
773300 018700 177500 :MOV (R1)+,R3 :MOVE TEST BITS TO R3
773300 112110 :LOOP1: MOV (R1)+,(R0) :MOVE COMMAND FROM TABLE TO T-11
773300 112110 :BMS DONE :IF COMMAND IS NEGATIVE THEN EXIT
773300 112110 :LOOPE: BTR R3,(R0) :TEST READY AND TRANSFER REQUEST BITS IN TACS
773300 112110 :BEG LOOPE :BRANCH IF BITS ARE NOT SET
773300 112110 :INCB R2 :ADVANCE MEMORY POINTER
773300 112110 :BMS LOOP1 :IF ZERO, TRY ANOTHER TABLE COMMAND
773300 112110 000000 :MOV (R0),(R2) :READ DATA INTO MEMORY
773300 112110 000000 :CMB R3,R2 :FIRST BYTE READ SHOULD BE "END"
773300 112110 001767 :BEG LOOPE :IF EQUAL, GO READ ANOTHER BYTE

773300 000000 STOP: HALT ON ERROR
773300 000000 BR RESTART :RESTART ON CONTINUE

773300 005210 DONE: TST (R0) :CHECK FOR ERROR
773300 005210 :BMS STOP :BRANCH TO HALT ON ERROR
773300 005207 :CLA PC :JUMP TO 0

773300 :TABLE: HIGH BYTE
773300 017640 :.WORD 024*000 + 2ND :1155+END-00 LOW BYTE
773300 017640 :.WORD 025*000 + 1ST :READY+TRANSFER REQUEST
773300 017640 :.WORD 026*000 + 1ST :READY+00 SPACE FORWARD BLOCK-00
773300 017640 :.WORD 027*000 + 1ST :READY+1155+END TABLE READ-1155
773300 000000 :.WORD 0,C :TWO WORDS OF FILLER

773300 173300 ** VECTOR: CBOOT :POWER-UP VECTOR (P1)
773300 000040 ** :POWER-UP STATUS (P2)
  
```

** NOT INCLUDED IN THE SOFTWARE VERSION

APPENDIX B

TALK BOOTSTRAP LOADER - TOGGLE-IN LISTING

ALTHOUGH THE TOGGLE-IN BOOT IS POSITION INDEPENDENT, IT IS RECOMMENDED THAT IT BE STORED STARTING AT LOCATION XX7706

LOCATION CONTENTS

LOCATION	CONTENTS
XX7706	0128200
XX7707	0128200
XX7708	0128200
XX7709	0128200
XX7710	0128200
XX7711	0128200
XX7712	0128200
XX7713	0128200
XX7714	0128200
XX7715	0128200
XX7716	0128200
XX7717	0128200
XX7718	0128200
XX7719	0128200
XX7720	0128200
XX7721	0128200
XX7722	0128200
XX7723	0128200
XX7724	0128200
XX7725	0128200
XX7726	0128200
XX7727	0128200
XX7728	0128200
XX7729	0128200
XX7730	0128200
XX7731	0128200
XX7732	0128200
XX7733	0128200
XX7734	0128200
XX7735	0128200
XX7736	0128200
XX7737	0128200
XX7738	0128200
XX7739	0128200
XX7740	0128200
XX7741	0128200
XX7742	0128200
XX7743	0128200
XX7744	0128200
XX7745	0128200
XX7746	0128200
XX7747	0128200
XX7748	0128200
XX7749	0128200
XX7750	0128200
XX7751	0128200
XX7752	0128200
XX7753	0128200
XX7754	0128200
XX7755	0128200
XX7756	0128200
XX7757	0128200
XX7758	0128200
XX7759	0128200
XX7760	0128200
XX7761	0128200
XX7762	0128200
XX7763	0128200
XX7764	0128200
XX7765	0128200
XX7766	0128200
XX7767	0128200
XX7768	0128200
XX7769	0128200
XX7770	0128200
XX7771	0128200
XX7772	0128200
XX7773	0128200
XX7774	0128200
XX7775	0128200
XX7776	0128200
XX7777	0128200
XX7778	0128200
XX7779	0128200
XX7780	0128200
XX7781	0128200
XX7782	0128200
XX7783	0128200
XX7784	0128200
XX7785	0128200
XX7786	0128200
XX7787	0128200
XX7788	0128200
XX7789	0128200
XX7790	0128200
XX7791	0128200
XX7792	0128200
XX7793	0128200
XX7794	0128200
XX7795	0128200
XX7796	0128200
XX7797	0128200
XX7798	0128200
XX7799	0128200
XX7800	0128200

.TITLE DITAF-C TALKER TALK DIAGNOSTIC ABB LOADER

.ABB

..LIST 00-000000
..INCLST 00-000000

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

..RECORDS 0017760

```

..TRACE READ FUNCTION
..TRACE UNIT LAST BYTE SEQUENCE CODE.
..TRACE SPACE REV FILE AND GO.
..TRACE SPACE REV BLOCK AND GO.
..TRACE SPACE FORUMNO FILE AND GO.
..TRACE SPACE REV BLOCK.
..TRACE REMNO AND GO.

```

..PAGEEND-RESTART

..PAGEEND-2008177600

..STR16-00

..STR13-00

..STR08-00

..STR08-00

..STR13-00

..STR11-00

.....

.....

.....

000000	000000		BTTL	PRE-LOADER SECTION OF TALLER (128 BYTES)	
000000	000000		BR	GO	IF NOT FOR CS007.
000000	000000		WORD	R0TCOR,0	
000000	000000		BNE	GO1	BR = 0?
					BR IF NOT.
000014	000000		HALT		YES, POST BOOT HALT
000000	001600	001:	MOV	#BYTNT+100,SP	
000000	000000	18:	CLR	R4	SIZE CORE.
000000	000000		TEST	(4)	REFERENCE.
000000	001064	00TCOR:	BR	#0000,R4	
000000	000000		IS		
000000	000000		MOV	#R0TCOR,R4	DETERMINE RELOC ADDR.
000000	000000		MOV	#WAIT,R3	WILL GO TO WAIT VIA R3
000000	000000		PC	(0)	RETURN TO START OF FILE.
000000	000000		PC	(3)	
000000	000000		PC	(0)	SKIP 1ST BLOCK.
000000	000000		PC	(3)	
000000	001400		MOV	#BYTNT,R2	LOAD ADDRESS IS 0.
000000	000000	18:	CLR	R1	R2 = TOTAL BYTES TO READ.
000000	000000		MOV	#READ,(0)	START READING A BLOCK.
000000	000000		PC	(3)	ISSUE READ FUNCTION.
000000	000000	28:	MOV	#2(0)(5)+	STORE BYTE.
000000	000000		PC	(3)	WAIT FOR XFR REQUEST.
000000	000000		PC	(3)	HAVE READ 128 BYTES?
000000	000000		PC	(3)	BR IF NO.
000000	000000		MOV	#LBS,(0)	START END BLOCK SEQUENCE.
000000	000000		PC	(3)	
000000	000000		PC	(3)	ALL BYTES READ?
000000	000000		PC	(3)	BR IF NOT TO READ NEXT BLOCK.
000000	177400		BR	0?	
000000	001001		BNE	RELOC	BR IF NOT, SKIP DIAGNOSTIC HALT.
000128	000000		HALT		PRE-RELOCATION HALT
000124	001076	000176:	RELOC:	MOV #PSTART,R1	RELOCATE CODE STARTING AT "PSTART"
000124	000000			MOV R4,R2	
000124	001064	18:	MOV	#R0TCOR,R3	RELOCATE A BYTE.
000124	000000		MOV	#1+(2)+	DONE?
000124	000000		PC	(0)	BR IF NOT.
000124	000000		PC	(0)	GET STARTED.
000124	000240	WAIT:	MOV	#R4,PC	CHECK FOR ERROR/XFR REQUEST
000124	000000		BIT	#WAIT,(0)	
000124	000000		BEQ	WAITA	CHECK FOR ERROR.
000124	000000		BIT	#4	BR IF ERROR.
000124	000000		PC	AC	OK, EXIT.
000172	000000		HALT		HARDWARE ERROR
000174	000711		BR	GO1	GO TRY AGAIN.

K02

1) LOAD THE FILE.

LOAD PL1

RK20:

000140

000150

000160

000170

177720

25:

000000

000007

000122

000260

45:

000472

000000

000764

000130

177646

000005

```

LOAD THE FILE.
LOAD PL1
RK20:
000140
000150
000160
000170
177720
25:
000000
000007
000122
000260
45:
000472
000000
000764
000130
177646
000005 000041
    
```

```

: CLEAR BLOCK COUNTER.
: SEND PREV DATA BLOCK.
: CLEAR THE CHECKSUM. (NO IS CHVSUM)
: GET A BYTE.
: BYTE IN AC. IS IS A SYNC? (1).
: BR IF NOT.
: NEXT BYTE MUST BE 0.
: BR IF 0.
: BR IF NOT. NOT A BLOCK START.
: GET THE BYTE COUNT (WORD).
: BR IF 0.
: GET THE SYNC AND BYTE COUNT.
: IF 0 IT'S XFR BLOCK.
: BR IF YES.
: GET LOAD ADDR.
: GET BYTE COUNT.
: BR IF BYTE COUNT STILL NOT 0.
: CHECKSUM OK?
: BR IF YES.
    
```

```

*****
: NO. CHECKSUM ERROR.
*****
: ISSUE REVERSE FILE FUNCTION
: BR IF 0.
: SKIP HEADER.
: TRY AGAIN.
: REACHED LOADER START?
: BR IF NOT.
*****
: TELLER LOADER OVERRUN
*****
55:
: STORE BYTE.
65:
: GET TRANSFER ADDR.
: CHECK FOR CORRECT CHECKSUM.
: BR IF+.
: BR IF NOT.
: SET CASSETTE LOAD INDICATION.
: CHECK FOR 000 XFR ADDR.
: BR TO 000 ADDR.
: RESTORE XFR ADDR.
: GO SELF START PROGRAM.
    
```


.5BTL CBOOT COPY

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099

PC
R15
R14
R13
R12
R11
R10
R9
R8
R7
R6
R5
R4
R3
R2
R1
TACS

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099

;TA=11 CONTROL AND STATUS REG.

000000 177500
000001 177500
000002 177500
000003 177500
000004 177500
000005 177500
000006 177500
000007 177500
000008 177500
000009 177500
000010 177500
000011 177500
000012 177500
000013 177500
000014 177500
000015 177500
000016 177500
000017 177500
000018 177500
000019 177500
000020 177500
000021 177500
000022 177500
000023 177500
000024 177500
000025 177500
000026 177500
000027 177500
000028 177500
000029 177500
000030 177500
000031 177500
000032 177500
000033 177500
000034 177500
000035 177500
000036 177500
000037 177500
000038 177500
000039 177500
000040 177500
000041 177500
000042 177500
000043 177500
000044 177500
000045 177500
000046 177500
000047 177500
000048 177500
000049 177500
000050 177500
000051 177500
000052 177500
000053 177500
000054 177500
000055 177500
000056 177500
000057 177500
000058 177500
000059 177500
000060 177500
000061 177500
000062 177500
000063 177500
000064 177500
000065 177500
000066 177500
000067 177500
000068 177500
000069 177500
000070 177500
000071 177500
000072 177500
000073 177500
000074 177500
000075 177500
000076 177500
000077 177500
000078 177500
000079 177500
000080 177500
000081 177500
000082 177500
000083 177500
000084 177500
000085 177500
000086 177500
000087 177500
000088 177500
000089 177500
000090 177500
000091 177500
000092 177500
000093 177500
000094 177500
000095 177500
000096 177500
000097 177500
000098 177500
000099 177500

```

CBOOT: MOV #TACS,PC
        CLR (R0)
RESTART: PUL R1
        MOV #TABLE--,R1
        MOV #075,R2
        MOV (R1)+,R3
        ;SELECT UNIT NO
        ;USE FOR PC
        ;R1 HOLDS ADDR. OF COMMAND TABLE
        ;MEMORY PTR. AND DATA FLAG
        ;TEST BITS

LOOP1: MOVB (R1)+,(R0)
        BRN DONE
        ;COMMAND FROM TABLE TO TACS
        ;WHEN COMMAND CODE NEW, QUIT
LOOP2: BITB R3,(R0)
        ;TEST READY AND T-REG. BITS IN TACS
        ;LOOP 'TILL SOMETHING COMES UP
        ;ADVANCE MEMORY PTR.
        INCB R3
        ;IF MINUS, TRY NEXT COMMAND
        LOOP1
        ;READ DATA INTO MEMORY
        MOVB (R0),(R2)
        ;FIRST BYTE READ SHOULD BE 'END'
        ;IF O.K., GO READ ANOTHER BYTE
        MOVB R3,R0
        LOOP2
STOP: BRN T
        ;HALT ON ERROR
        ;RESTART ON CONTINUE
        RESTART

DONE: TST (R0)
        ;CHECK FOR ERROR
        BRN STOP
        ;HALT ON ERROR
        CLR PC
        ;" JMP 280"

TABLE: .LORG 27*40+24
        ;READY+Y-REG./ILBS+READY+60
        .LORG 24*40+16
        ;R2+00+R3+16
        .LORG 22*40+24
        ;R2+ILBS+R3+ILBS+E.O.TABLE

;RESTART CODE. NOT PART OF CBOOT CODE ABOVE.
BR REPEAT
FRGEND: REPEAT
        ;GO TO ABS LOADER.

.END

```

REC'D 08-SEP-76 09:31
COMMUNICATIONS SECTION
FEDERAL BUREAU OF INVESTIGATION
WASHINGTON, D.C. 20535

LINE NO.	DESCRIPTION	UNIT	RATE	AMOUNT	EXTENSION
1004*	STANDARD	0000	10.00	10.00	09
101*	STANDARD	0000	10.00	10.00	09
1016	STANDARD	0000	10.00	10.00	09
1009	STANDARD	0000	10.00	10.00	09
1003	STANDARD	0000	10.00	10.00	09

E03

Speaker on line 3 seconds 10 KFS 07 disk reads 3 disk writes 29 10000
#####