

KT11-D

EXERCISER
MD-11-DBKTG-D

EP-DBKTG-D-DL-B
COPYRIGHT © 1976
FICHE 1 OF 1

DEC 1976
digital
MADE IN USA

[Frame 1: Data]	[Frame 2: Data]	[Frame 3: Data]	[Frame 4: Data]	[Frame 5: Data]	[Frame 6: Data]
[Frame 7: Data]	[Frame 8: Data]	[Frame 9: Data]	[Frame 10: Data]	[Frame 11: Data]	[Frame 12: Data]
[Frame 13: Data]	[Frame 14: Data]	[Frame 15: Data]	[Frame 16: Data]	[Frame 17: Data]	[Frame 18: Data]
[Frame 19: Data]	[Frame 20: Data]	[Frame 21: Data]	[Frame 22: Data]	[Frame 23: Data]	[Frame 24: Data]
[Frame 25: Data]	[Frame 26: Data]	[Frame 27: Data]	[Frame 28: Data]	[Frame 29: Data]	[Frame 30: Data]
[Frame 31: Data]	[Frame 32: Data]	[Frame 33: Data]	[Frame 34: Data]	[Frame 35: Data]	[Frame 36: Data]
[Frame 37: Data]	[Frame 38: Data]	[Frame 39: Data]	[Frame 40: Data]	[Frame 41: Data]	[Frame 42: Data]
[Frame 43: Data]	[Frame 44: Data]	[Frame 45: Data]	[Frame 46: Data]	[Frame 47: Data]	[Frame 48: Data]
[Frame 49: Data]	[Frame 50: Data]	[Frame 51: Data]	[Frame 52: Data]	[Frame 53: Data]	[Frame 54: Data]
[Frame 55: Data]	[Frame 56: Data]	[Frame 57: Data]	[Frame 58: Data]	[Frame 59: Data]	[Frame 60: Data]

[Small text or logo in the bottom right corner of the microfiche card]

B01

DBKTGD MAINDEC-11-DBKTG-D KT11-D EXERCISER
DBKTGD.P11 21-SEP-76 11:27

MACY11 27(1006) 21-SEP-76 11:30 PAGE 1

.REM *

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DBKTG-D-D
PRODUCT NAME: KT11-D EXERCISER
DATE RELEASED: FEBRUARY, 1977
MAINTAINER: DIAGNOSTIC PROGRAMMING
AUTHOR: DIAGNOSTIC ENGINEERING

COPYRIGHT (C) DIGITAL EQUIPMENT CORPORATION
1974, 1977

THE MATERIAL IN THIS DOCUMENT IS FOR INFORMATION
PURPOSES ONLY AND IS SUBJECT TO CHANGE WITHOUT NOTICE.
DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY
FOR THE USE OF SOFTWARE ON EQUIPMENT WHICH IS NOT
SUPPLIED BY IT.
DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY
FOR ANY ERRORS WHICH MAY APPEAR IN THE DOCUMENT.

1.0 ABSTRACT

THIS PROGRAM IS AN INTERACTIVE EXERCISER FOR A PDP-11/40 EQUIPPED WITH THE KT11-D OPTION. IT PERFORMS A TEST OF INSTRUCTIONS AND CONCURRENT OPERATIONS OF I/O EQUIPMENT WHILE RELOCATING THRU MEMORY. IT PROVIDES NUMEROUS MODES OF TESTING, FROM 4K EXECUTION WITH THE KT11-D TURNED OFF AND ONLY KERNEL MODE IN USE, TO 128K EXECUTION WITH EACH USER PAGE MAPPED SEQUENTIALLY TO EVERY 4K BANK OF MEMORY. THIS PROGRAM IS NOT TO BE CONSIDERED A TOTAL CHECK OF THE SYSTEM. IF AN ERROR IS DETECTED IN AN I/O DEVICE, IT WILL PROBABLY BE NECESSARY TO CORRECT THE MALFUNCTION WITH THE RESPECTIVE DIAGNOSTIC FOR THAT DEVICE.

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP-11/40 STANDARD COMPUTER
KT11-D MEMORY MANAGEMENT OPTION

2.1.1 OPTIONAL HARDWARE THAT THE PROGRAM WILL EXERCISE

MEMORY UP TO 124 KW OF MEMORY-DOES NOT HAVE TO BE CONTIGUOUS,
BUT BLOCKS OF LESS THAN 4KW WILL NOT BE USED
RF11 DISK
RK11 DISK
TC11 DECTAPE-TRANSPORT ONE(1)
KW11-L LINE CLOCK
KL11 ASR33 OR ASR35 TELEPRINTER
LP11 LINE PRINTER

2.2 STORAGE

THIS PROGRAM USES MEMORY FROM 00000 TO 17760.

3.0 LOADING PROCEDURE

PROCEDURE FOR NORMAL ABSOLUTE TAPES SHOULD BE FOLLOWED.

4.0 STARTING PROCEDURE AND SWITCH SETTINGS

4.1 NORMAL STARTING PROCEDURE

LOAD STARTING ADDRESS 200.
SET DESIRED MEMORY MANAGEMENT SELECTION SWITCHES (SEE 4.2)-ALL
DOWN FOR WORST CASE TESTING.
PRESS START.

4.1 NORMAL STARTING PROCEDURE (CONTINUED)

THE PROGRAM WILL IMMEDIATELY HALT. AT THE HALT, SET THE DESIRED DEVICE SELECTION SWITCHES (SEE 4.3) AND THE DESIRED DYNAMIC SWITCHES (SEE 5.1.2).

PRESS CONTINUE.

THE PROGRAM WILL RING THE BELL (UNLESS THE TTY OUTPUT IS SELECTED) AT THE END OF EACH BANK. IF SWITCHES 0, 1 AND 2 WERE ALL DOWN WHEN START WAS PRESSED (SELECTING THE USE OF 4K PHYSICAL ADDRESS SPACE AS 32K VIRTUAL ADDRESS SPACE-SEE 5.3.1) AN ASTERISK WILL BE TYPED AT THE END OF A FULL PASS THRU ALL MEMORY (UNLESS THE TTY OUTPUT IS SELECTED).

4.2 MEMORY MANAGEMENT SELECTION SWITCHES (INITIAL SWITCH REGISTER SETTINGS).

THE SWITCHES SET AT STARTUP DETERMINE THE WAY IN WHICH MEMORY IS MAPPED AND EXERCISED:

SW0=1 OR UP---INHIBIT THE KT11-D (SR0(0) WILL NOT BE SET AT ALL)
SW1=1 OR UP---INHIBIT USE OF USER MODE.

(ALSO INHIBITS 4K AS 32K)

SW2=1 OR UP---INHIBIT 4K AS 32 K (ALSO INHIBITED IF EITHER SW0 OR SW1 IS SET)-SEE SECTION 5.3.1 FOR EXPLANATION

SW5=1 OR UP---INHIBIT VARIABLE CORE EXPANSION
=0 OR DOWN-CORE EXPAND UNLESS SW0, 1 AND 2 ARE ALL DOWN
(IN WHICH CASE 4K AS 32K IS RUN INSTEAD)

4.3 DEVICE SELECTION SWITCHES

THE DEVICE SELECTION SWITCHES ARE SET AT THE FIRST (AND ONLY) HALT. EACH SWITCH, IF SET, INHIBITS A SINGLE I/O DEVICE FROM BEING EXERCISED. IF A DEVICE DOES NOT EXIST, THE CORRESPONDING INHIBIT SWITCH DOES NOT HAVE TO BE SET.

SW0=1 OR UP---INHIBIT TTY OUTPUT
SW3=1 OR UP---INHIBIT RK11 DISK
SW4=1 OR UP---INHIBIT LINE CLOCK
SW5=1 OR UP---INHIBIT RF11 DISK
SW6=1 OR UP---INHIBIT TC11 DECTAPE
SW7=1 OR UP---INHIBIT LINE PRINTER (USE SA310 IF LP11 IS SELECTED)

NOTE: WHEN THE PROGRAM IS LOADED VIA THE DECTAPE OR RK11 DISK (TCDP, RKDP), THE PROGRAM WILL NOT SELECT THE LOAD MEDIUM SUBSYSTEM EXERCISER CODE. THAT IS, IT WILL NOT EXERCISE IT. IF THE USER WISHES TO RUN THE LOAD MEDIUM, LOCATIONS 40 AND 41 MUST FIRST BE CLEARED MANUALLY.

4.4 RESTART PROCEDURE

USING RESTART ADDRESS 310 THE SWITCH REGISTER SETTINGS GIVEN PREVIOUSLY ARE USED (FOR BOTH MEMORY MANAGEMENT SELECTION AND DEVICE SELECTION). NO HALT OCCURS AFTER START IS PRESSED.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

5.1.1 BASIC SWITCH SETTINGS-STARTUP

SEE SECTIONS 4.2 AND 4.3 FOR THE BASIC SWITCH SETTINGS USED AT STARTUP. THOSE SWITCHES ARE NOT RECHECKED AFTER THEY ARE INITIALLY STORED.

5.1.2 DYNAMIC SWITCH SETTINGS

THE FOLLOWING SWITCHES ARE RECHECKED PERIODICALLY DURING PROGRAM EXECUTION:

SW15=1 OR UP---HALT ON ERROR
 SW14=1 OR UP---SCOPE LOOP
 SW13=1 OR UP---INHIBIT PRINT OUT
 SW12=1 OR UP---INHIBIT TRACE TRAPPING
 SW11=1 OR UP---INHIBIT SUB-PROGRAM ITERATION AND INHIBIT
 TESTS WHICH USE ALL COMBINATIONS OF
 NUMBERS
 SW10=1 OR UP---INHIBIT PROCESSOR TEST (ONCE SET, PROCESSOR
 TEST IS PERMANENTLY INHIBITED)

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED. IF A SCOPE LOOP IS REQUESTED, IT WILL JUMP TO THE START OF THE SUBTEST THAT THE SCOPE LOOP IS REQUESTED FOR. IF A SCOPE LOOP IS NOT REQUESTED, THERE WILL BE 256 ITERATIONS ON THAT SUBTEST BEFORE THE NEXT SUBTEST IS ENTERED. SWITCH 11 ON A 1 INHIBITS ITERATION OF SUBTESTS.

5.2.2 HLT

THIS EMT CALLS THE SUBROUTINE PRINT, WHICH PRINTS OUT THE LOCATION COUNTER AT THE TIME OF FAILURE, THE CONTENTS OF THE PROCESSOR STATUS REGISTER, AND THE CONTENTS OF THE CURRENT BANK COUNTER. NOTE THAT THE LOCATION COUNTER WILL BE THE VIRTUAL ADDRESS OF THE HLT PLUS TWO.

5.2.3 TRAPCATCHER

THIS IS A SERIES OF INSTRUCTIONS STARTING AT LOCATION 0 DESIGNED TO DETECT AND ISOLATE UNEXPECTED TRAPS AND INTERRUPTS TO THE TRAP AND INTERRUPT VECTOR AREA OF MEMORY.

EACH VECTOR ENTRANCE ADDRESS IS LOADED WITH THE ADDRESS OF THE NEXT LOCATION. THE NEXT LOCATION IS LOADED WITH A HALT (000000). THUS AN ILLEGAL TRAP OR INTERRUPT WILL CAUSE A HALT AT THE TRAP LOCATION PLUS TWO.

IF A HALT OCCURS IN THE TRAP OR INTERRUPT AREA EXAMINE KERNEL REGISTER SIX. IT WILL CONTAIN THE CURRENT STACK ADDRESS. THE CONTENTS OF THE CURRENT STACK ADDRESS IS THE VIRTUAL PC AT THE TIME THE TRAP OR INTERRUPT OCCURRED.

5.2.4 EMTSRV (EMT HANDLER)

THIS ROUTINE DECODES THE EMT CALLS AND PASSES CONTROL TO THE CORRECT SERVICE ROUTINE. THE ROUTINES HANDLED BY EMT CALLS ARE PRINT (HLT CALL) AND EOBSRV (EOB CALL).

5.2.6 EOBSRV (END OF BANK SERVICE)

THE VARIOUS EXECUTION OPTIONS FOR THIS EXERCISER REQUIRE SPECIAL HANDLING WHEN THE END OF THE PROCESSOR TESTS IS REACHED IN A BANK. THIS SERVICE ROUTINE PERFORMS THE VARIOUS MAPPING FUNCTIONS, DEPENDING UPON THE INITIAL SWITCH REGISTER SETTINGS.

5.2.7 BEGINX (CORE EXPANSION SPECIAL HANDLER)

WHEN CORE EXPANSION IS UTILIZED, A NUMBER OF SPECIAL ACTIONS MUST BE TAKEN AT THE BEGINNING OF EACH BANK. THE SCOPE ROUTINE VECTOR IS LOADED TO POINT TO THE NEW BANK, AND IF TC11 AND RF11 CODE AND BUFFER RELOCATION IS ALLOWED.

5.2.9 PFAIL (POWER FAIL)

IN THIS VERSION THE POWER FAIL ROUTINE IS NOT OPERABLE.

5.2.11 TYOUT (TTY OUTPUT)

THIS ROUTINE OUTPUTS A COUNT PATTERN IN THE INTERRUPT MODE TO THE TELEPRINTER.

5.2.12 RFSTART (RF11 DISK)

THIS ROUTINE PERFORMS A WRITE AND A WRITE CHECK OF THE DISK. THE DATA THAT IS WRITTEN ON THE DISK IS A PART OF THE TEST PROGRAM CODE THAT IS NEVER MODIFIED. THIS SEGMENT OF CORE IS WRITTEN IN CONTIGUOUS BLOCKS THRU THE DISK MEMORY. AFTER THE TOTAL DISK(S) HAS BEEN WRITTEN, A WRITE CHECK IS USED TO VERIFY THAT THE DATA HAS BEEN WRITTEN CORRECTLY ON THE DISK. NOTE THAT NO "DATI" ARE USED IN EXERCISING THE DISK (DATA IS NOT TRANSFERRED INTO MEMORY). THERE IS A LOCATION IN THE PROGRAM THAT IF MODIFIED WILL ALLOW EXERCISING UP TO EIGHT DISKS.

5.2.13 ENDZ (TC11 END ZONE HANDLER)

THIS ROUTINE IS PART OF THE TC11 SERVICE CODE. IT DRIVES THE DECTAPE INTO THE FORWARD OR REVERSE END ZONE, THEN REVERSES IT. IT ALSO DOES THE NECESSARY SETUP TO BEGIN READING OR WRITING THE TAPE.

5.2.14 REGEN (TC11 WRITE BUFFER REGENERATE ROUTINE)

THE TC11 CODE WRITES THE ENTIRE DECTAPE GOING FORWARD, THEN READS IT IN REVERSE. THE BUFFER IS REGENERATED BEFORE WRITING THE TAPE, AND IS CLEARED OUT ONCE THE ENTIRE TAPE HAS BEEN WRITTEN. THIS ROUTINE REGENERATES THE WRITE BUFFER.

5.2.15 RBN (TC11 READ BLOCK NUMBER SERVICE ROUTINE)

AT THE END OF EACH "BLOCK NUMBER FOUND" INTERRUPT, THIS ROUTINE IS ENTERED (UNLESS END ZONE IS BEING SEARCHED FOR). IT CHECKS FOR THE CORRECT SEQUENCE OF BLOCK NUMBERS, THEN SETS UP THE TC11 TO WRITE A BLOCK IF THE TAPE IS TRAVELLING FORWARD. IF IT IS GOING IN REVERSE, THE ROUTINE CHECKS TO SEE IF DATA IS STILL BEING CHECKED FROM A PREVIOUS READ. IF IT'S NOT, THE ROUTINE SETS UP TO READ A BLOCK. IF DATA IS STILL BEING CHECKED FROM BEFORE, IT SIMPLY DOES ANOTHER READ BLOCK NUMBER.

5.2.16 NXTBLK (TC11 READ BLOCK AND WRITE BLOCK SERVICE ROUTINE)

WHEN A READ BLOCK OR A WRITE BLOCK OPERATION IS COMPLETED, THIS ROUTINE IS ENTERED. IT CHECKS THE ERROR BIT, THEN SETS UP A CALL TO CHECK DATA IF DATA WAS JUST READ IN. THE ROUTINE ALSO SETS UP A READ BLOCK NUMBER OPERATION.

5.2.17 TCK (TC11 CHECK DATA ROUTINE)

WHEN A READ BLOCK OPERATION HAS BEEN COMPLETED, THIS ROUTINE IS CALLED VIA A PRIORITY INTERRUPT REQUEST AT LEVEL 3. THE ENTIRE BUFFER IS CHECKED, AND THE CONTENTS OF THE BUFFER IS ALTERED AS THE CHECK PROGRESSES. THUS, IF A READ BLOCK OPERATION DOES NOT ACTUALLY READ IN ANY DATA, THE DATA CHECK ROUTINE WILL FIND BAD DATA INSTEAD OF SEEING GOOD DATA FROM AN EARLIER READ.

5.2.18 LCLK (LINE CLOCK)

THIS TEST OF THE LINE CLOCK IS IN THE INTERRUPT MODE. IF OPERATING CORRECTLY THE SYSTEM I/O WILL RUN AT FULL SPEED FOR 55 SECONDS. AND THEN ALL I/O AT LEVEL FOUR OR LESS (AND THE PROCESSOR TESTS) WILL STALL FOR 5 SECONDS. TIMES GIVEN ARE BASED ON 60 CYCLES AS THE LINE FREQUENCY.

5.2.19 LPI (LINE PRINTER)

THIS ROUTINE OUTPUTS TO THE LINE PRINTER IN THE FLAG MODE WHILE FILLING THE BUFFER, AND IN THE INTERRUPT MODE WHILE THE BUFFER IS BEING PRINTED.

5.2.20 RKSTART (RK-11 DISK)

THIS ROUTINE PERFORMS A WRITE AND WRITE CHECK OF THE DISK. THE DATA THAT IS WRITTEN ON THE DISK IS PART OF THE TEST PROGRAM CODE THAT IS NEVER MODIFIED. THIS SEGMENT OF CORE IS WRITTEN IN CONTIGUOUS BLOCKS THRU THE DISK MEMORY. AFTER THE TOTAL DISK HAS BEEN WRITTEN, A WRITE CHECK IS USED TO VERIFY THAT DATA HAS BEEN WRITTEN CORRECTLY ON THE DISK.

5.2.22 CORE EXPANSION (DET1)

THIS ROUTINE IS CONTROLLED BY SWITCH 5. IF CALLED, THE PROCESSOR MAINLINE CODE WILL EXPAND TO THE MAXIMUM MEMORY THAT IS AVAILABLE (UP TO 28K). THE ROUTINE DETERMINES THE MAXIMUM MEMORY SIZE BY DOING A "DATO" TO A LOCATION IN EACH BANK. IF THE BANK DOES NOT EXIST, A TIMEOUT WILL OCCUR. AN IMAGE OF BANK 0 IS THEN TRANSFERRED TO EACH EXISTING BANK. THE CODE IN EACH BANK EXCEPT THE LAST IS MODIFIED TO CHANGE THE END OF BANK CALL TO A JUMP TO BEGINX (CORE EXPANSION SPECIAL HANDLER) IN THE NEXT BANK.

THE LISTING SHOWS ONLY THE CODE FOR BANK ZERO. WHEN AN ERROR OCCURS THAT IS NOT IN BANK ZERO, IGNORE THE BANK BITS OF THE PRINT OUT AND USE THE LISTING FOR BANK ZERO.

5.3 PROGRAM AND/OR OPERATOR ACTION

5.3.1 PROCESSOR TEST EXECUTION - 4K AS 32K

IF SWITCHES 0, 1, AND 2 ARE ALL DOWN (=0) AT STARTUP, THE PROCESSOR TEST WILL BE EXECUTED TREATING EACH 4K BANK AS 32K OF VIRTUAL ADDRESS SPACE. THE FOLLOWING DETAILS THIS MODE OF OPERATION.

USER PAGE 0 IS FIRST MAPPED RW, BANK 0, AND ALL OTHER USER PAGES ARE MAPPED NON-RESIDENT. THE PROCESSOR TESTS ARE EXECUTED IN USER THRU USER PAGE 0. WHEN DONE, USER PAGE 0 IS CHANGED TO NON-RESIDENT, AND USER PAGE 1 IS MAPPED RW, BANK 0. THE PC IS CHANGED TO ADDRESS THE START OF THE PROCESSOR TESTS THRU PAGE 1, AND ANOTHER PASS THRU THE PROCESSOR TESTS IS EXECUTED. AT THE END OF THIS PASS, USER PAGE 2 IS MAPPED RW, BANK 0, AND USER PAGE 1 IS MADE NON-RESIDENT. THE PC IS AGAIN CHANGED. THIS TIME TO ACCESS USER PAGE 2, AND THE PROCESSOR TESTS ARE EXECUTED THRU USER PAGE 2. THIS CYCLE IS REPEATED FOR THE REMAINING USER PAGES, MAPPING EACH IN TURN TO BANK 0 AND CHANGING THE PC TO EXECUTE THRU THE ONE CURRENTLY MAPPED. WHEN THE PASS USING USER PAGE 7 IS COMPLETED, A SEARCH IS MADE FOR THE NEXT 4K BANK OF MEMORY. WHEN A BANK IS FOUND, THE PROGRAM IS COPIED INTO THAT BANK FROM BANK 0. USER PAGE 0 IS MAPPED TO THE NEW BANK, AND THE PC IS CHANGED TO EXECUTE THRU USER PAGE 0. THE PREVIOUS CYCLE IS REPEATED, BUT THIS TIME EACH USER PAGE IS MAPPED IN TURN TO THE NEW BANK. ONCE EXECUTION THRU USER PAGE 7 IS COMPLETED, A SEARCH IS MADE FOR THE NEXT BANK. THE PREVIOUS BANK IS CLEARED (EXCEPT FOR THE LOADER), AND THE PROGRAM IS COPIED FROM BANK 0 INTO THE CURRENT BANK. THE CYCLE REPEATS UNTIL THE EXTERNAL BANK IS REACHED, AT WHICH POINT USER 0 IS MAPPED BACK TO BANK 0 AND THE PROCESS STARTS AGAIN.

5.3.2 PROCESSOR TEST EXECUTION - CORE EXPANSION

IF SWITCH 0, 1, OR 2 IS UP AND SW5 IS DOWN AT STARTUP, THE PROCESSOR TESTS WILL BE CORE EXPANDED THRU ALL AVAILABLE MEMORY UP TO 28K. THE ROUTINE DET1 DOES THIS CORE EXPANSION, COPYING BANK 0 INTO EACH OF THE OTHER BANKS. THE EMT CALL AT THE END OF EACH BANK (EOB) WHICH CALLS THE END OF BANK SERVICE ROUTINE IS CHANGED TO A JUMP TO BEGINX IN THE NEXT BANK. THE EOB CALL IN THE LAST BANK IS LEFT ALONE. IF SWITCHES 0 AND 1 WERE BOTH DOWN AT STARTUP, USER PAGES 0 THRU 6 ARE MAPPED SO THAT THE PHYSICAL AND VIRTUAL ADDRESSES CORRESPOND, AND THE PROCESSOR TESTS ARE THEN RUN IN USER. IF SW0 WAS DOWN BUT SW1 WAS SET, KERNEL PAGES 0-6 ARE MAPPED SO THAT THE PHYSICAL AND VIRTUAL ADDRESSES ARE THE SAME, AND THE PROCESSOR TESTS ARE THEN RUN IN KERNEL MODE. IF SW0 WAS SET, ORDINARY CORE EXPANSION IS RUN WITH NO SPECIAL MAPPING REQUIRED (KT11-D IS TURNED OFF).

5.3.3 PROCESSOR TEST EXECUTION - BANK 0 ONLY

IF SW0, 1 OR 2 IS UP AND SW5 IS UP AT STARTUP, ONLY BANK 0 IS UTILIZED. IN THIS CASE, IF SW0 AND SW1 WERE DOWN THE PROCESSOR TESTS ARE EXECUTED IN USER, WITH USER PAGE 0 MAPPED TO BANK 0. IF SW0 WAS DOWN AND SW1 WAS UP, THE PROCESSOR TESTS ARE EXECUTED IN KERNEL, WITH KERNEL PAGE 0 MAPPED TO BANK 0. IF SW0 WAS UP, THE KT11-D IS TURNED OFF AND THE PROCESSOR TESTS ARE EXECUTED IN KERNEL MODE OR USER MODE (DEPENDING ON SW1) IN BANK 0 ONLY.

6.0 ERRORS

6.1 ERROR PRINTOUT

PRINTOUTS ARE IN AN EXTENDED VERSION OF THE STANDARD FORMAT, USING THREE WORDS. THE FIRST WORD IS THE OCTAL VALUE OF THE VIRTUAL PC+2 OF THE DETECTED ERROR. THE SECOND WORD IS THE CONTENTS OF THE PROCESSOR STATUS REGISTER WHEN THE ERROR WAS DETECTED. THE THIRD IS THE TOP 12 BITS OF THE 18-BIT ADDRESS OF THE BANK BEING CURRENTLY USED FOR EXECUTION OF THE PROCESSOR TEST. THE FOURTH IS RETURN WHICH IS THE RETURN ADDRESS IN THE CURRENT BANK OF MEMORY. TO GET THE STARTING ADDRESS OF THE CURRENT BANK SIMPLY APPEND TWO ZEROS TO THE END OF THE OCTAL VALUE PRINTED OUT (I.E. 007400 INDICATES THE BANK BEGINNING AT PHYSICAL ADDRESS 740000).

6.2 ERROR RECOVERY

IN GENERAL, TEST FAILURES WILL PRINTOUT AN ERROR MESSAGE AND CONTINUE. IF THE "HALT ON ERROR" SWITCH IS SET, HITTING CONTINUE WILL RECOVER. IF THE PROGRAM HANGS UP IN A LOOP, THE ERROR IS LIKELY TO BE A SIGNAL WHICH WAS NEVER RECEIVED. IF A HALT OCCURS IN THE TRAP AND VECTOR AREA THE PROGRAM MUST BE RESTARTED. IF THE PROGRAM HALTS IN THE MAIN FLOW, CONSULT THE LISTING IF NO MESSAGE IS TYPED OUT. FOR TTY READER AND HSR, TAPE MUST BE REPOSITIONED TO LEADER BEFORE RESTARTING THE TEST.

6.3 FINDING WHICH PROCESSOR TEST WAS BEING EXECUTED WHEN AN ERROR OCCURRED

SOME ERRORS ARE DEPENDENT ON THE PROCESSOR TEST BEING RUN (SUCH AS LATENCY ERRORS WHICH ONLY SHOW UP IN WORST-CASE PROCESSOR TIMING). THE SCOPE ROUTINE CONTAINS A LOCATION CALLED "RETURN" WHICH STORES THE STARTING ADDRESS OF THE PROCESSOR TEST CURRENTLY BEING EXECUTED. NOTE THAT THE SCOPE ROUTINE IS EXECUTED IN USER MODE IF SW1 IS DOWN AT STARTUP, AND IS THEREFORE RELOCATED WITH THE PROCESSOR TESTS. THUS, TO DETERMINE WHICH PROCESSOR TEST WAS BEING EXECUTED WHEN A FAILURE OCCURRED, FIRST CHECK THE CONTENTS OF CURBNK IN BANK 0. THIS LOCATION CONTAINS THE ADDRESS OF THE CURRENT PHYSICAL BANK, SHIFTED RIGHT 6 PLACES. BY APPENDING 2 ZEROES TO IT, YOU HAVE THE 18-BIT ADDRESS OF THE CURRENT BANK OF MEMORY. ADD TO THIS THE ADDRESS OF RETURN IN BANK 0 AND YOU HAVE THE ADDRESS OF RETURN IN THE CURRENT BANK OF MEMORY. THE CONTENTS OF RETURN IN THE CURRENT BANK OF MEMORY IS THE VIRTUAL ADDRESS OF THE START OF THE CURRENT PROCESSOR TEST.

7.0 RESTRICTIONS

PROGRAM MUST BE LOADED INTO LOWER 4K OF MEMORY.

THE INHIBIT SWITCHES MUST ONLY BE SET FOR ALL DEVICES THAT ARE PART OF THE SYSTEM BUT WHICH YOU DO NOT WISH TO RUN.

IF THE LINE PRINTER IS USED, STARTING ADDRESS 310 MUST BE USED.

8.0 MISCELLANEOUS

8.1 EXECUTION TIME

EXECUTION TIME VARIES WITH THE AMOUNT OF MEMORY, THE TYPES OF MEMORY, AND THE OPTIONAL MODES OF EXECUTION USED.

THE FOLLOWING RUN TIMES ARE BASED ON A PDP-11/40 SYSTEM WITH 64K OF CORE STORAGE, AND WITH ONLY THE LINE CLOCK ENABLED.

STAND ALONE WITH ALL MEMORY MANAGEMENT OPTIONS ENABLED:

1ST PASS ABOUT 3 SECONDS.
2ND PASS ABOUT 70 SECONDS (TRACE MODE ON).
3RD PASS ABOUT 35 SECONDS (TRACE MODE OFF).

ACT11 WITH OPTIONS SET AS DESCRIBED IN SECTION 8.4:

1ST PASS ABOUT 3 SECONDS.
2ND PASS ABOUT 60 SECONDS (TRACE MODE ON).
3RD PASS ABOUT 30 SECONDS (TRACE MODE OFF).

XXDP WITH OPTIONS SET AS DESCRIBED IN SECTION 8.4:

1ST PASS 1 TO 2 SECONDS.
2ND PASS ABOUT 24 SECONDS (TRACE MODE ON).
3RD PASS ABOUT 12 SECONDS (TRACE MODE OFF).

9.2 STACK POINTERS

THE KERNEL STACK POINTER IS INITIALIZED TO 17760.

THE USER STACK POINTER IS INITIALIZED TO 400. IT IS RELOCATED THRU ALL USER PAGES AND TO EVERY 4K BANK IF THE 4K AS 32K MODE OF EXECUTION IS RUN.

9.3 MONITORING PHYSICAL AND VIRTUAL ADDRESSES

DURING EXECUTION OF 4K AS 32K, IT IS HELPFUL TO SET THE ADDRESS SELECTOR TO PROGRAM PHYSICAL AND THE DATA SELECTOR TO DATA PATHS. IF THIS IS DONE, THE ADDRESS LIGHTS WILL INDICATE THE CURRENT PHYSICAL ADDRESSES WHILE THE DATA LIGHTS WILL SHOW THE CURRENT VIRTUAL ADDRESSES (SINCE THEY ARE USED AS DATA A GREAT DEAL OF THE TIME).

9.4 ACT11/XXDP OPERATION

FOUR LOCATIONS ARE USED AS SOFTWARE SWITCHES TO CONTROL PROGRAM OPERATION DURING ACT11 OR XXDP CHAIN MODE OPERATION. THE SOFTWARE SWITCHES CONTENTS ARE USED TO SET SOFTWARE SWITCHES SREG1 AND SREG2, WHICH ARE THE LOCATIONS THAT ARE ROUTINELY CHECKED BY THE PROGRAM TO CONTROL ITS OPERATION.

THE ACT11/XXDP SOFTWARE SWITCHES ARE:

ACTSW1: 40 :NO CORE EXPANSION.
ACTSW2: 201 :INHIBIT LPT AND TTY DURING ACT11.

XDPSW1: 46 :INHIBIT KT11D, NO CORE EXPANSION, NO 4K AS 32
XDPSW2: 1 :INHIBIT TTY WHILE IN XXDP CHAIN MODE.

SWITCH XDPSW1 MUST ALWAYS BE LEFT WITH THE VALUE 46, AS IF CHANGED, THE PROGRAM WILL NOT FUNCTION UNDER CHAIN MODE.

ALL OTHER SWITCHES MAY BE CHANGED FREELY, ESPECIALLY THE DEVICE SELECTION SWITCHES XDPSW2 AND ACTSW2.

THE LOAD MEDIUM IS NOT EXERCISED BY THE PROGRAM WHEN LOADED VIA TCDP OR RKDP (THAT IS DECTAPE OR RK11 WILL NOT BE EXERCISED THEN).

9.0 PROGRAM DESCRIPTION

THIS MEMORY MANAGEMENT EXERCISER IS DESIGNED TO RUN BACKGROUND PROCESSOR TESTS AND FOREGROUND CONCURRENT I/O WITH MEMORY MANAGEMENT UTILIZED IN ANY OF SEVERAL DIFFERENT MODES. THE VARIOUS MODES AVAILABLE FOR UTILIZING MEMORY MANAGEMENT ARE INCLUDED TO AID IN FAULT ISOLATION BY PROVIDING A SERIES OF STEPS FROM SIMPLE TO COMPLEX. MEMORY MANAGEMENT CAN BE LEFT TURNED OFF AND THE PROCESSOR TESTS CAN STILL BE RUN IN 4K ONLY OR CORE EXPANDED UP TO 28K. WITH MEMORY MANAGEMENT ON, THE PROGRAM CAN BE RUN USING ONLY 4K, WITH EVERYTHING MAPPED IN KERNEL SPACE OR WITH USER AND KERNEL BOTH USED. AT THE NEXT LEVEL OF COMPLEXITY, CORE EXPANSION CAN BE RUN WITH MEMORY MANAGEMENT ON, USING KERNEL ONLY OR USING BOTH MODES AS DESIRED. FINALLY, ALL AVAILABLE MEMORY (IN 4K PIECES) CAN BE UTILIZED BY RUNNING 4K AS 32K.

THERE IS NO MONITOR IN THE CONVENTIONAL SENSE. EACH DEVICE THAT IS TO BE EXERCISED HAS ITS OWN STAND ALONE ROUTINE THAT OPERATES IN THE INTERRUPT MODE. THESE ROUTINES NEED NO SUPERVISION OR MONITORING AFTER THEY ARE INITIATED. THERE IS A PRIMER AREA THAT CHECKS THE SWITCH REGISTER TO SEE WHAT DEVICES ARE TO BE INITIATED. IT SETS THE INTERRUPT ENABLE BIT IN THE DEVICE STATUS REGISTER, INITIALIZES THE DATA PATTERN, AND INITIATES AN OPERATION TO RAISE DATA FLAGS ON DEVICES THAT CAN NOT INITIATE THEM THEMSELVES. THE PRIMER CODE THEN ENTERS THE KT11-D SETUP CODE.

THE RF11 AND TC11 PRIMER CODE IS IN WITH THE KT11-D SETUP CODE SINCE THEY REQUIRE CERTAIN PARTS OF THE KT11-D CODE TO BE RUN FIRST. AFTER MEMORY MANAGEMENT IS TURNED ON, EXECUTION OF THE BACKGROUND PROCESSOR TESTS BEGINS, AND THE I/O DEVICES ARE SERVICED WHEN THEY INTERRUPT.

D02

DBKTDG MAINDEC-11-DBKTDG-D KT11-D EXERCISER
DBKTDG.P11 21-SEP-76 11:27

MACY11 27(1006) 21-SEP-76 11:30 PAGE 16

000000
0070
071

:

REVISED BY: W.F. KELLICKER 25-FEB-74
R. KOLLER MARCH 1976

572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614

```

:*****
:SBTTL OPERATING INSTRUCTIONS
:*****
:PDP11/40 SYSTEM EXERCISER, WITH KT11-D --- TTY,PC11,KW11-L
:LP11,RF11,TC11
:TEST SIMULTANEOUS RUNNING OF I/O, WITH PROCESSOR INSTRUCTION TEST AND
:WITH TRACE BIT ENABLED TO BE CONSIDERED MAINLINE CODE

:I/O RUNS IN KERNEL MODE
:CPU TESTS RUN IN USER MODE UNLESS INHIBITED BY SR SETTINGS
:KT11-D IS UTILIZED

:(R6) IS THE STACK POINTER
:((R6)) IS THE PC+2 OF LOCATION WHERE THE TRAP ORIGINATED
:FOR NORMAL OPERATION RUN WITH ALL SWITCHES DOWN
:SA - 200
:RESTART - 310 (SR SETTINGS PREVIOUSLY MADE ARE USED)

:AT STARTUP, SR SETTINGS ARE:
:SR 0=1 OR UP --- RUN WITHOUT KT11-D
:SR 1=1 OR UP --- RUN ALL IN KERNEL MODE (INHIBITS RUNNING 4K AS 32K)
:SR 2=1 OR UP --- INHIBIT RUNNING 28K USER KT11-D FROM EVERY 4K
:                        :BANK (ALLOW NORMAL CORE EXPANSION)
:SR 5=1 OR UP---INHIBIT VARIABLE CORE EXPANSION

:AT HALT, SR SETTINGS ARE:
:SR 15=1 OR UP---HALT ON ERROR
:SR 14=1 OR UP---SCOPE LOOP
:SR 13=1 OR UP---INHIBIT PRINT OUT
:SR 12=1 OR UP---INHIBIT TRACE TRAPPING
:SR 11=1 OR UP---INHIBIT SUB-PROGRAM ITERATION AND INHIBIT TESTS WHICH
:                        :USE ALL COMBINATIONS OF NUMBERS
:SR 10=1 OR UP---INHIBIT PROCESSOR TEST

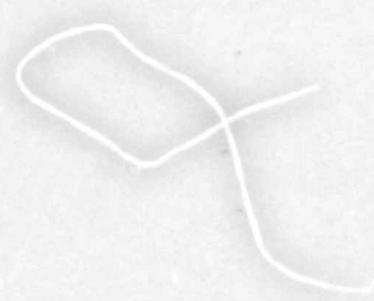
:SPECIAL DELETE SWITCHES-SET RESPECTIVE SWITCH TO A 1 TO INHIBIT
:INITIATION OF DEVICE
:SW 0=1 INHIBIT TTY OUTPUT
:SW 3=1 INHIBIT RK11 DISK
:SW 4=1 INHIBIT LINE CLOCK
:SW 5=1 INHIBIT RF11 DISK
:SW 6=1 INHIBIT TC11 DECTAPE
:SW 7=1 INHIBIT LINE PRINTER

```

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
000100

000240
104400
000410
000412
177570
177776
104006
104010
000000
000001
000002
000003
000004
000005
000006
000006
000007

```
*****
:SBTTL DEFINITIONS
*****
NOP=240                :SYSTEM NULL OPERATION
SCOPE=TRAP             :TRAP USED SCOPE LOOP AND ITERATION
TCSR=TTCSR
TDBR=TTDBR
SR=177570
PSR=177776
HLT=104006
EOB=104010
R0=%0
R1=%1
R2=%2
R3=%3
R4=%4
R5=%5
SP=%6
R6=SP
PC=%7
:ERROR PRINTOUT CALL
:END OF BANK CALL
```



```

636 :*****
637 :SBTTL TRAP CATCHER
638 :*****
639      000000      . = 0
640 000000 000002      . + 2      : TRAP ENTRANCE
641 000002 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
642 000004 000006      . + 2      : TRAP ENTRANCE
643 000006 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
644 000010 000012      . + 2      : TRAP ENTRANCE
645 000012 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
646 000014 000016      . + 2      : TRAP ENTRANCE
647 000016 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
648 000020 000022      . + 2      : TRAP ENTRANCE
649 000022 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
650 000024 000026      . + 2      : TRAP ENTRANCE
651 000026 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
652 000030 000032      . + 2      : TRAP ENTRANCE
653 000032 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
654 000034 000036      . + 2      : TRAP ENTRANCE
655 000036 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
656 000040 000042      . + 2      : TRAP ENTRANCE
657 000042 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
658 000044 000046      . + 2      : TRAP ENTRANCE
659 000046 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
660 000050 000052      . + 2      : TRAP ENTRANCE
661 000052 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
662 000054 000056      . + 2      : TRAP ENTRANCE
663 000056 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
664 000060 000062      . + 2      : TRAP ENTRANCE
665 000062 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
666 000064 000066      . + 2      : TRAP ENTRANCE
667 000066 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
668 000070 000072      . + 2      : TRAP ENTRANCE
669 000072 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
670 000074 000076      . + 2      : TRAP ENTRANCE
671 000076 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
672 000100 000102      . + 2      : TRAP ENTRANCE
673 000102 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
674 000104 000106      . + 2      : TRAP ENTRANCE
675 000106 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
676 000110 000112      . + 2      : TRAP ENTRANCE
677 000112 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
678 000114 000116      . + 2      : TRAP ENTRANCE
679 000116 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
680 000120 000122      . + 2      : TRAP ENTRANCE
681 000122 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
682 000124 000126      . + 2      : TRAP ENTRANCE
683 000126 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
684 000130 000132      . + 2      : TRAP ENTRANCE
685 000132 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
686 000134 000136      . + 2      : TRAP ENTRANCE
687 000136 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
688 000140 000142      . + 2      : TRAP ENTRANCE
689 000142 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION
690 000144 000146      . + 2      : TRAP ENTRANCE
691 000146 000000      HAL T      : TRAPPED TO PREVIOUS LOCATION

```

692	000150	000152	.+2	: TRAP ENTRANCE
693	000152	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
694	000154	000156	.+2	: TRAP ENTRANCE
695	000156	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
696	000160	000162	.+2	: TRAP ENTRANCE
697	000162	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
698	000164	000166	.+2	: TRAP ENTRANCE
699	000166	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
700	000170	000172	.+2	: TRAP ENTRANCE
701	000172	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
702	000174	000176	.+2	: TRAP ENTRANCE
703	000176	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
704	000200	000202	.+2	: TRAP ENTRANCE
705	000202	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
706	000204	000206	.+2	: TRAP ENTRANCE
707	000206	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
708	000210	000212	.+2	: TRAP ENTRANCE
709	000212	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
710	000214	000216	.+2	: TRAP ENTRANCE
711	000216	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
712	000220	000222	.+2	: TRAP ENTRANCE
713	000222	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
714	000224	000226	.+2	: TRAP ENTRANCE
715	000226	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
716	000230	000232	.+2	: TRAP ENTRANCE
717	000232	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
718	000234	000236	.+2	: TRAP ENTRANCE
719	000236	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
720	000240	000242	.+2	: TRAP ENTRANCE
721	000242	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
722	000244	000246	.+2	: TRAP ENTRANCE
723	000246	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
724	000250	000252	.+2	: TRAP ENTRANCE
725	000252	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
726	000254	000256	.+2	: TRAP ENTRANCE
727	000256	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
728	000260	000262	.+2	: TRAP ENTRANCE
729	000262	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
730	000264	000266	.+2	: TRAP ENTRANCE
731	000266	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
732	000270	000272	.+2	: TRAP ENTRANCE
733	000272	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
734	000274	000276	.+2	: TRAP ENTRANCE
735	000276	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
736	000300	000302	.+2	: TRAP ENTRANCE
737	000302	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
738	000304	000306	.+2	: TRAP ENTRANCE
739	000306	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
740	000310	000312	.+2	: TRAP ENTRANCE
741	000312	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
742	000314	000316	.+2	: TRAP ENTRANCE
743	000316	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
744	000320	000322	.+2	: TRAP ENTRANCE
745	000322	000000	HALT	: TRAPPED TO PREVIOUS LOCATION
746	000324	000326	.+2	: TRAP ENTRANCE
747	000326	000000	HALT	: TRAPPED TO PREVIOUS LOCATION

748	000330	000332	.+2	; TRAP ENTRANCE
749	000332	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
750	000334	000336	.+2	; TRAP ENTRANCE
751	000336	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
752	000340	000342	.+2	; TRAP ENTRANCE
753	000342	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
754	000344	000346	.+2	; TRAP ENTRANCE
755	000346	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
756	000350	000352	.+2	; TRAP ENTRANCE
757	000352	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
758	000354	000356	.+2	; TRAP ENTRANCE
759	000356	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
760	000360	000362	.+2	; TRAP ENTRANCE
761	000362	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
762	000364	000366	.+2	; TRAP ENTRANCE
763	000366	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
764	000370	000372	.+2	; TRAP ENTRANCE
765	000372	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
766	000374	000376	.+2	; TRAP ENTRANCE
767	000376	000000	HALT	; TRAPPED TO PREVIOUS LOCATION

```

768
769
770 ;*****
771 ;SBTTL LOAD VECTOR AREA
772 ;*****
773 000024 000024      =24
774 000026 016502    PFAIL      ;POWER FAIL TRAP
775 000030 000340    340
776 000030 015204    =30
777 000032 000340    EMTSRV     ;EMT CALLS
778 000034 000034    340         ;HIGHEST PRIORITY
779 000034 014702    =34
780 000036 000000    SCOPEC     ;USER TRAP
781 000040 000040    0
782 000040 000000    =40
783 000042 000042    0         ;LOAD MEDIUM INDICATOR.
784 000042 000000    =42
785 000046 000046    0         ;LOADS AS 0.
786 000046 015752    $ENDAD     ;AUTOMATIC MODE INDICATOR.(ACT11/XXDP).
787 000052 000052    =52
788 000052 040000    40000     ;ZERO AT LOAD TIME.
789
790
791 ;*****
792 ;SBTTL LOAD STARTING AREA
793 ;*****
794 000200 000200      =200
795 000200 000137 000666  JMP      @#START
796 000300 000300      =300
797 000300 000137 000666  JMP      @#START
798 000310 000310      =310
799 000310 000137 000636  JMP      @#RSTRT
800
801
802
803 ;*****
804 ;SBTTL DATA AREA
805 ;*****

```

804		000400		=400	
805	000400	000000	UBUFF:	0	;BUFFER FOR USER SP
806		000406		=. +4	;FOR STACK OVERRUN
807	000406	177560	TRCSR:	177560	;TTY READER STATUS REGISTER
808	000410	177564	TICSR:	177564	;TTY PUNCH STATUS REGISTER
809	000412	177566	TTDBR:	177566	
810	000414	000064	TTPVC:	64	
811	000416	000066	TTPST:	66	
812	000420	000000	TTSAV:	0	
813	000422	000100	KWLVC:	100	
814	000424	000102	KWLST:	102	
815	000426	177546	LKCSR:	177546	
816	000430	177514	LPCSR:	177514	
817	000432	177516	LPDBR:	177516	
818	000434	000200	LPVC:	200	
819	000436	000202	LPST:	202	
820	000440	177470	RFDAR:	177470	;DISK ADDRESS AND ERROR
821	000442	177466	RFDAR:	177466	;DISK ADDRESS REGISTER
822	000444	177462	RFWC:	177462	;WORD COUNT REGISTER
823	000446	177464	RFCAR:	177464	;CURRENT ADDRESS REGISTER
824	000450	177460	RFCSR:	177460	;STATUS REGISTER
825	000452	177461	RFCSRH:	177461	;HIGH BYTE ADDRESS OR CSR
826	000454	000204	RFVC:	204	
827	000456	000206	RFST:	206	
828	000460	177413	RKDAH:	177413	;HIGH BYTE DISK ADR
829	000462	177412	RKDAE:	177412	;DISK ADDRESS REGISTER
830	000464	177406	RKWC:	177406	;WORD COUNT REGISTER
831	000466	177410	RKBAR:	177410	;CURRENT ADDRESS REGISTER
832	000470	177404	RKCSR:	177404	;STATUS REGISTER
833	000472	177405	RKCSRH:	177405	;HIGH BYTE OF CSR
834	000474	000220	RKVC:	220	;TRAP VECTOR
835	000476	000222	RKST:	222	
836	000500	177572	SRD:	177572	;KT11-D REGISTERS
837	000502	177600	UPDR0:	177600	
838	000504	177602	UPDR1:	177602	
839	000506	177616	UPDR7:	177616	
840	000510	177640	UPAR0:	177640	
841	000512	177642	UPAR1:	177642	
842	000514	177656	UPAR7:	177656	
843	000516	172300	KPDR0:	172300	
844	000520	172302	KPDR1:	172302	
845	000522	172304	KPDR2:	172304	
846	000524	172316	KPDR7:	172316	
847	000526	172340	KPAR0:	172340	
848	000530	172342	KPAR1:	172342	
849	000532	172344	KPAR2:	172344	
850	000534	172356	KPAR7:	172356	
851	000536	000000	FTITLE:	0	;TITLE FLAG
852					
853	000540	177600	IPDRTAB:	177600	
854	000542	177640		177640	
855	000544	172300		172300	
856	000546	172340	IPDREND:	172340	
857	000550	000000	SREG1:	0	;STORES KT11-D SWITCH REGISTER SETTINGS
858	000552	000000	SREG2:	0	;STORES SR SETTINGS
859	000554	177342	TCCM:	177342	;CONTROL AND FUNCTION

860 000556 177340
861 000560 177350
862 000562 177344
863 000564 177346
864 000566 000214
865 000570 000216
866 000572 000000
867 000574 000000
868 000576 000000
869 000600 000000
870 000602 000000
871 000604 000000
872
873
874
875
876 000606 177777
877 000610 077777
878 000612 000001
879 000614 000606
880 000616 000000
881 000620 000000
882 000622 000000
883
884
885
886
887 000624 000014
888 000626 000040
889 000630 000201
890 000632 000046
891 000634 000001
892
893
894
895
896 000636 012706 017760
897 000642 012737 016502 000024
898 000650 005737 000042
899 000654 001143
900 000656 113737 177571 000553
901 000664 000537
902
903
904
905
906
907 000666 005767 177644
908 000672 001036
909 000674 023737 000042 000046
910 000702 001432
911 000704 004767 015506
912 000710 012700 000736
913 000714 105710
914 000716 001404
915 000720 112046

TCST: 177340
TCDT: 177350
TCWC: 177344
TCBA: 177346
TCIV: 214
TCSTA: 216
CURBNK: 0
OLDBNK: 0
CURPAR: 0
CURPDR: 0
BNKSTR: 0
TRPB: 0
MEMO: 177777
MEM1: 77777
COREPT: 1
MEMUT: MEMO
TBANK: 0
REFF: 0
TEST: 0

;GENERAL STATUS
;DATA
;WORD COUNT
;BUS ADDRESS
;DECTAPE INTERRUPT VECTOR
;SAF TO POINT TO CURRENT BANK
;ADDRESS OF CURRENT ISAR
; PC TO POINT TO BEGIN THRU CURRENT SEGMENT
;THE NEXT TWO WORDS ARE THE MEMORY MAP. THE FIRST WORD REPRESENTS
;0-64K WITH ONE BIT REPRESENTING A 4K CONTIGUOUS BLOCK. IF THE
;BIT=1 THAT 4K BLOCK IS PRESENT. THE LSB REPRESENTS 0-4K, THE NEXT
;SIGNIFICANT BIT REPRESENTS 4-8K ANS SO ON.
;0-64K
;64-124K

;SBTTL FILLCT, ACT11, XXDP SOFTWARE SWITCHES

FILLCT: 14 ;CONSOLE FILL COUNT.
ACTSW1: 40 ;NO CORE EXPANSION.
ACTSW2: 201 ;NO LP, NO TTY.
XDPSW1: 46 ;NO CORE EXPANSION, NO 4 AS 32, ETC.
XDPSW2: 1 ;NO TTY.

;SBTTL RESTART ADD USING INITIAL SR SETTINGS

RSTRT: MOV #KSTACK,R6
MOV #PFAIL,#24
TST #42 ;IN AUTO MODE? (ACT11/XXDP)
BNE START2 ;BR IF YES.
MOVB #SR+1,#SREG2+1 ;UPDATE DYNAMIC SWITCH SETTINGS.
BR START2

;SBTTL START UP FOR MINI MONITOR - NORMAL START FROM LOC 000200

START: TST FTITLE ;HAS TITLE BEEN PRINTED?
BNE STARTA ;YES, SKIP TITLE
CMP #42,#46 ;ARE WE IN ACT11 AUTOMATIC MODE?
BEQ STARTA ;YES, SKIP TITLE
JSR PC,CRLF
MOV #TITLE,R0 ;GET MESSAGE ADDRESS
1\$: TSTB (0) ;ARE WE THROUGH YET?
BEQ 2\$;YES
MOVB (0)+,-(SP)


```

916 000722 004767 015534
917 000726 000772
918 000730 004767 015462
919 000734 000415
920 000736 052113 030461 042055
921 000744 042440 042530 041522
922 000752 051511 051105 020040
923 000760 041104 052113 026507
924 000766 000104
925
926 000770 012706 017760
927 000774 012737 000137 000200
928 001002 012737 000666 000202
929 001010 005067 177570
930 001014 005067 014000
931 001020 005737 000042
932 001024 001450
933 001026 012737 000002 000006
934 001034 012700 000003
935 001040 000261
936 001042 005737 177772
937 001046 005600
938 001050 000261
939 001052 105737 177777
940 001056 005600
941 001060 005037 177700
942 001064 005037 000006
943 001070 022700 000002
944 001074 001402
945
946
947 001076 104006
948 001100 000776
949
950
951 001102 023727 000042 015752
952 001110 001007
953 001112 016737 177510 000550
954 001120 016737 177504 000552
955 001126 000416
956 001130 016737 177476 000550
957 001136 016737 177472 000552
958 001144 000407
959 001146 013737 177570 000550
960 001154 000000
961 001156 013737 177570 000552
962 001164
963 001164 004767 013752
964 001170 012777 077406 177320
965 001176 012777 007600 177330
966 001204 012777 077406 177312
967 001212 005067 177400
968 001216 012767 177777 177362
969 001224 012767 077777 177356
970 001232 012767 000001 177352
971 001240 012767 000606 177346

```

```

                JSR    PC,CHROUT
                BR     1$
2$:             JSR    PC,CRLF
                BR     STARTA
TITLE:         .ASCIZ /KT11-D EXERCISER DBKTG-D/

                .EVEN
STARTA:        MOV     #KSTACK,R6           ;SET UP STACK
                MOV     #137, @#200        ;RESTORE 200 IF START AT 300
                MOV     #START, @#202
                CLR     TRPB                ;NO TRACE IN FIRST PASS.
                CLR     PASCNT              ;CLEAR THE PASS COUNTER.
                TST     @#42                ;IN AUTOMATIC TEST MODE?
                BEQ     STARTX              ;BR IF NOT IN AUTOMATIC MODE.
                MOV     #RTI, @#6           ;IS THIS AN 11/40
                MOV     #3, R0
                SEC
                TST     @#177772            ;RO=3 IF 11/45
                SBC     R0                  ;RO=2 IF 11/40
                SEC                          ;RO=1 IF 11/20
                TSTB    @#PSR+1             ;RO=0 IF 11/05
                SBC     R0
                CLR     @#177700
                CLR     @#6
                CMP     #2, R0              ;WELL, IS IT 11/40?
                BEQ     2$                  ;BR IF YES.

;????????????????????????????????????????????????????????
1$:             HLT
                BR     1$
;????????????????????????????????????????????????????????

2$:             CMP     @#42, #SENDAD       ;IN ACT11 MODE?
                BNE     3$                  ;BR IF NOT.
                MOV     ACTSW1, @#SREG1     ;YES. SET SREG1 FROM ACTSW1.
                MOV     ACTSW2, @#SREG2     ;SET SERG2 FROM ACTSW2.
                BR     START1
3$:             MOV     XDPSW1, @#SREG1     ;XXDP MODE. SET SREG1 FROM XDPSW1.
                MOV     XDPSW2, @#SREG2     ;SET SREG2 FROM XDPSW2.
                BR     START1
STARTX:        MOV     @#SR, @#SREG1        ;STORE KT11-D SWITCHES
                HALT
                MOV     @#SR, @#SREG2

START1:        JSR     %7, NRALL
START2:        MOV     #77406, @KPDR0
                MOV     #7600, @KPAR7       ;MAP PAGE 7 TO EXT BANK
                MOV     #77406, @KPDR7
                CLR     TBANK
                MOV     #177777, MEM0       ;SET UP CORE MAPS
                MOV     #77777, MEM1
                MOV     #1, CORÉPT          ;SET UP 4K POINTER
                MOV     #MEM0, MEMUT

```

```

972 001246 012777 077406 177246      MOV      #77406, @KPCR2      ;BEING CHECKED FOR
973 001254 012737 001324 000004      MOV      #TMEMEX, @#4      ;SET UP FOR TIME OUTS
974 001262 005037 000006      CLR      @#6
975 001266 052777 000001 177204      BIS      #1, @SR0
976 001274 016777 177316 177230      MAP1:  MOV      TBANK, @KPAR2      ;MAP KERNEL PAGE 2 TO BANK
977 001302 005737 041000      TST      @#41000          ;1ST K PRESENT
978 001306 005737 045000      TST      @#45000          ;2ND K PRESENT
979 001312 005737 051000      TST      @#51000          ;3RD K PRESENT
980 001316 005737 055000      TST      @#55000          ;4TH K PRESENT
981 001322 000404      BR       MOVEPT          ;OK, FULL 4K BLOCK PRESENT
982 001324 046777 177262 177262      TMEMEX: BIC      COREPT, @MEMUT      ;NO, BLOCK NOT PRESENT
983 001332 022626      CMP      (SP)+, (SP)+      ;ADJUST STACK POINTER
984 001334 062767 000200 177254      MOVEPT: ADD      #200, TBANK      ;UPDATE BANK POINTER
985 001342 006367 177244      ASL      COREPT
986 001346 103006      BCC      MAP2            ;THIS 1ST MEM WORD DONE
987 001350 012767 000001 177234      MOV      #1, COREPT
988 001356 012767 000610 177230      MOV      #MEM1, MEMUT
989 001364 022767 007600 177224      MAP2:  CMP      #7600, TBANK      ;EXTERNAL BANK YET
990 001372 001340      BNE      MAP1            ;NO, NOT YET?
991 001374 012767 000001 177210      MOV      #1, COREPT      ;RE-INIT
992 001402 012767 000606 177204      MOV      #MEM0, MEMUT
993 001410 042777 000001 177062      BIC      #1, @SR0
994 001416 012737 014702 000034      MOV      #SCOPEC, @#34
995 001424 005037 000036      CLR      @#36            ;INITIALIZE SCOPE 'CALL TO KERNEL STATUS
996 001430 012737 015204 000030      MOV      #EMTSRV, @#30
997 001436 012737 000340 000032      MOV      #340, @#32
998 001444 012737 005654 015016      MOV      #BEGIN, @RETURN
999 001452 012737 000340 177776      MOV      #340, @PSR      ;LOCK OUT INTERRUPTS
1000 001460 005037 016172      CLR      @#PRTON        ;PRINT ROUTINE BUSY FLAG
1001 001464 000005      RESET
1002 001466 012737 002516 000004      MOV      #NODEV, @#4      ;RETURN FOR NO DEVICE
1003 001474 005037 000006      CLR      @#6
1004      ;*****
1005      .SBTTL  TTY INIT
1006      ;*****
1007 001500 005067 001464      CLR      DATA2          ;BASE DATA FOR TTY TELEPRINTER
1008 001504 033727 000552 000001      BIT      @#SREG2, #1      ;INHIBIT TTY OUTPUT?
1009 001512 001006      BNE      ST3            ;YES, GO CHECK NEXT.
1010 001514 012777 003202 176672      MOV      #TYOUTR, @TTPVC  ;NO, SETUP INTERRUPT VECTOR
1011 001522 052777 000100 176660      BIS      #100, @TTCR      ;START TTY OUTPUT
1012      ;*****
1013      .SBTTL  RK11 INIT
1014      ;*****
1015 001530 012700 000010      ST3:  MOV      #10, R0
1016 001534 122737 000002 000041      CMPB     #2, @#41        ;LOAD MEDIUM RK11?
1017 001542 001432      BEQ      ST4            ;BR IF YES. DON'T USE RK11 THEN.
1018 001544 032737 000010 000552      BIT      #10, @#SREG2    ;INHIBIT RK DISK
1019 001552 001026      BNE      ST4            ;YES, SKIP OVER
1020 001554 005777 176710      TST      @RKCSR          ;PRESENT
1021 001560 012777 003600 176706      MOV      #IRK, @RKVC     ;SETUP VECTOR RETURNS
1022 001566 012777 000240 176702      MOV      #240, @RKST     ;PRIORITY 5 SERVICE.
1023 001574 012767 043503 002040      MOV      #43503, RKFUNCT
1024 001602 005077 176654      CLR      @RKDAE          ;INIT
1025 001606 016777 002170 176652      MOV      LLIMIT, @RKBAR  ;CORE BASE
1026 001614 016777 002164 176642      MOV      WORDCT, @RKWC   ;TRANSFER LENGTH
1027 001622 116777 002014 176640      MOVB    RKFUNCT, @RKCSR

```

```

1028 ;*****
1029 ;SBTTL LINE CLOCK INIT
1030 ;*****
1031 001630 006300 ST4: ASL RO
1032 001632 033727 000552 000020 BIT @#SREG2,#20 ;INHIBIT LINE CLOCK?
1033 001640 001015 BNE ST5 ;YES, GO CK NEXT
1034 001642 005777 176560 TST @LKCSR ;PRESENT
1035 001646 012777 003260 176546 MOV #LK3,@KWLVC
1036 001654 012777 000300 176542 MOV #300,@KWLST
1037 001662 005067 001466 CLR TIME ;NO, INITIALIZE COUNT
1038 001666 052777 000100 176532 BIS #100,@LKCSR ;START LINE CLOCK
1039 ;*****
1040 ;SBTTL RF11 INIT
1041 ;*****
1042 001674 006300 ST5: ASL RO
1043 001676 033727 000552 000040 BIT @#SREG2,#40 ;TEST FOR INHIBITING RF11 DISK
1044 001704 001026 BNE ST6 ;SKIP IF SET
1045 001706 005777 176536 TST @RFCSR ;PRESENT?
1046 001712 012777 003674 176534 MOV #IRF,@RFVC ;SET UP TRAP RETURN
1047 001720 012777 000240 176530 MOV #240,@RFST
1048 001726 012767 043503 002044 MOV #43503,RFFUNCT ;WRITE CHECK/WRITE
1049 001734 105277 176512 INCB @RFCSRH ;INITIALIZE DISK-DAR,DAE
1050 001740 016777 002040 176476 MOV WORDCT,@RFWC ;LENGTH OF TRANSFER
1051 001746 016777 002030 176472 MOV LLIMIT,@RFCAR ;CORE ADDRESS OF START OF TRANSFER
1052 001754 116777 002020 176466 MOV RFFUNCT,@RFCSR ;START RF11 READ OR WRITE
1053 ;*****
1054 ;SBTTL TC11 INIT
1055 ;*****
1056 001762 006300 ST6: ASL RO
1057 001764 122737 000001 000041 CMPB #1,@#41 ;LOAD MEDIUM DECTAPE?
1058 001772 001417 BEQ ST7 ;BR IF YES. DON'T USE IT THEN.
1059 001774 033727 000552 000100 BIT @#SREG2,#100 ;CHECK FOR INHIBITING TC11 DECTAPE
1060 002002 001013 BNE ST7 ;SKIP IF SET
1061 002004 005777 176546 TST @TCST ;PRESENT?
1062 002010 012777 004014 176550 MOV #FENDZ,@TCIV ;GO TO END ZONE ON INTERRUPT
1063 002016 012777 000300 176544 MOV #300,@TCSTA
1064 002024 012777 004503 176522 MOV #R+IE+RB+DO,@TCCM ;START REVERSE READ BLOCK NUMBER
1065 ;*****
1066 ;SBTTL LINE PRINTER INIT
1067 ;*****
1068 002032 006300 ST7: ASL RO
1069 002034 033727 000552 000200 BIT @#SREG2,#200 ;INHIBIT LINE PRINTER?
1070 002042 001032 BNE ST8 ;YES, GO CK NEXT
1071 002044 005777 176360 TST @LPCSR ;PRESENT?
1072 002050 012737 002130 000004 MOV #ST8,@#4 ;DON'T CHANGE 200 IF NO SUCH DEVICE
1073 002056 012767 000137 001274 MOV #137,SOLPAT ;RESET FOR START OF LINE PATTERN
1074 002064 012767 000117 001360 MOV #79,CLINCT ;LINE COUNT
1075 002072 012767 000137 001262 MOV #137,CURPAT
1076 002100 012777 000014 176324 MOV #14,@LPDBR ;LINE FEED TO POSITION BUFFER
1077 002106 012777 003402 176320 MOV #LPINTR,@LPVC ;INTERRUPT ENABLE
1078 002114 012777 000200 176314 MOV #200,@LPST ;PROCESSOR LEVEL 4
1079 002122 012777 000100 176300 MOV #100,@LPCSR ;INTERRUPT ENABLE
1080 ;*****
1081 ;SBTTL PRE-PASS SETUP
1082 ;*****
1083 002130 005037 000006 ST8: CLR @#6 ;CHANGE ADDRESS ERROR VECTOR TO CAUSE

```

1084	002134	012737	000006	000004		MOV	#6, @#4	:HALT ON A TRAP TO 4
1085	002142	004767	000370			JSR	%7, DET1	:CHECK FOR CORE EXPANSION
1086	002146	032737	000001	000550		BIT	#1, @#SREG1	:INHIBIT KT11-D?
1087	002154	001106				BNE	MODE	:YES - GO SETUP USER
1088	002156	004767	012760			JSR	%7, NRALL	:NO - MAKE ALL SEGMENTS INITIALLY NON-RESIDENT
1089	002162	012777	077406	176334		MOV	#77406, @KPAR7	
1090	002170	012777	007600	176336		MOV	#7600, @KPAR7	
1091	002176	032737	000006	000550		BIT	#6, @#SREG1	:INHIBIT USER/KERNEL OR 4K AS 32K?
1092	002204	001415				BEQ	SEGM1	:NO - BRANCH
1093	002206	012701	000007			MOV	#7, R1	:YES - MAP KERNEL ASR'S 0-6 TO PA
1094	002212	016702	176310			MOV	KPAR0, R2	
1095	002216	005003				CLR	R3	
1096	002220	010312			SETEX:	MOV	R3, @R2	
1097	002222	012762	077406	177740		MOV	#77406, -40(R2)	
1098	002230	005722				TST	(R2)+	
1099	002232	062703	000200			ADD	#200, R3	
1100	002236	077110				SOB	R1, SETEX	
1101	002240	012777	077406	176250	SEGM1:	MOV	#77406, @KPAR0	:MAP KERNEL 0 TO BANK 0, RW
1102	002246	032737	000004	000550		BIT	#4, @#SREG1	:INHIBIT RUNNING 4K AS 32K?
1103	002254	001416				BEQ	USEALL	:NO, SETUP FOR RUNNING 4K AS 32K
1104	002256	012701	000010			MOV	#10, R1	:YES, MAP ALL USER ASR'S TO PA
1105	002262	016702	176222			MOV	UPAR0, R2	
1106	002266	005003				CLR	R3	
1107	002270	010312			SETUSE:	MOV	R3, (R2)	
1108	002272	062703	000200			ADD	#200, R3	
1109	002276	012762	077406	177740		MOV	#77406, -40(R2)	
1110	002304	005722				TST	(R2)+	
1111	002306	077110				SOB	R1, SETUSE	
1112	002310	000425				BR	SETSEG	
1113	002312	012777	077406	176162	USEALL:	MOV	#77406, @UPDR0	:MAP USER ASR0 TO BANK 0, RW
1114	002320	012737	000000	000572		MOV	#0, @#CURBNK	:CURRENT SAR CONTENTS
1115	002326	012767	000001	176256		MOV	#1, COREPT	:INIT MAP POINTERS
1116	002334	012767	000606	176252		MOV	#MEMO, MEMUT	
1117	002342	016767	176142	176226		MOV	UPAR0, CURPAR	:CURRENT SEGMENT REGISTER ADDRESSES
1118	002350	016767	176126	176222		MOV	UPDR0, CURPDR	
1119	002356	012767	005654	176216		MOV	#BEGIN, BNKSTR	:CURRENT STARTING PC
1120	002364	052777	000001	176106	SETSEG:	BIS	#1, @SR0	:SET KT11-D BIT
1121	002372	005767	176206		MODE:	TST	TRAP	:USE TRACE MODE?
1122	002376	001406				BEQ	1\$:BR IF NOT.
1123	002400	012737	015774	000014		MOV	#TRTRP, @#14	:SET UP TRACE TRAP VECTOR.
1124	002406	012746	000020			MOV	#20, -(SP)	:ALLOW TRACE MODE.
1125	002412	000406				BR	2\$	
1126	002414	012737	000016	000014	1\$:	MOV	#16, @#14	:NO TRACE MODE . RESET THE VECTOR.
1127	002422	005037	000016			CLR	@#16	
1128	002426	005046				CLR	-(SP)	:INSURE NO TRACE WILL BE ENABLED.
1129	002430	012746	002436		2\$:	MOV	#3\$, -(SP)	:CONTINUE AT 3\$.
1130	002434	000002				RTI		:DO IT NOW.
1131	002436	032737	000002	000550	3\$:	BIT	#2, @#SREG1	:INHIBIT USER/KERNEL?
1132	002444	001016				BNE	MAIN+2	:YES - SKIP OVER
1133	002446	052737	140000	000036		BIS	#140000, @#36	:SET USER BIT IN SCOPE STATUS
1134	002454	012746	000400			MOV	#UBUFF, -(R6)	
1135	002460	052737	030000	177776		BIS	#30000, @#PSR	
1136	002466	006606				MTP1	SP	:SET UP USER STACK
1137	002470	012737	140000	177776		MOV	#140000, @#PSR	:CHANGE TO USER
1138	002476	000401				BR	+.4	
1139	002500	000001			MAIN:	WAIT		

```

1140 002502 033727 000552 002000 BIT 2#SREG2, #2000 ;INHIBIT PROCESSOR TEST
1141 002510 001373 BNE MAIN
1142 002512 000167 003136 JMP BEGIN
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
002516 050037 000552
002522 162716 000006
002526 042766 000017 000002
002534 000002

002536 012767 104010 012074
002544 032767 000007 175776

002552 001001
002554 000207
002556 032737 000040 000550
002564 001401
002566 000207
002570 012737 002654 000004 DET4:
002576 012737 000340 000006 MOV
002604 000241 CLC
002606 005537 037770 EIGHT: ADC
002612 000240 NOP
002614 005537 057770 ADC
002620 000240 NOP
002622 005537 077770 ADC
002626 000240 NOP
002630 005537 117770 ADC
002634 000240 NOP
002636 005537 137770 ADC
002642 000240 NOP
002644 005537 157770 ADC
002650 000240 NOP
002652 000437 BR
002654 012602 DET2: MOV
002656 005726 TST
002660 062702 000074 ADD
002664 000112 JMP

002666 005000 MOVE: CLR
002670 010102 MOV
002672 062702 015142 ADD
002676 012021 MOV
002700 020201 CMP
002702 001375 BNE
002704 000207 RTS
002706 000521 STRT4: BR
002710 000240 NOP
002712 000240 NOP
002714 004767 000110 JSR

```

```

:*****
.SBTTL NON-EXISTING DEVICE SERVICE
:*****
NODEV: BIS R0, 2#SREG2 ;SET INHIBIT BIT
SUB #6, (SP) ;ALTER PC RETURN
BIC #17, 2(SP) ;CLEAR Z BIT ON STACK
RTI

```

```

:*****
.SBTTL PDP-11 MEMORY DETERMINATION AND SETUP
:*****
:USE WITH VARIABLE CORE QUANTITY SYSTEMS/

```

```

DET1: MOV #EOB, DONE ;RESTORE INITIAL CODE
BIT #7, SREG1 ;INHIBIT RUNNING 4K AS 32K USER?
;OR INHIBIT SEGMENTATION?
BNE .+4 ;YES - ALLOW CORE EXPANSION
RTS %7 ;NO - INHIBIT CORE EXPANSION
BIT #40, 2#SREG1 ;CHECK VARIABLE CORE SWITCH
BEQ DET4 ;USE VARIABLE CORE ROUTINE
RTS %7 ;4K ONLY (SWITCH SET)
DET4: MOV #DET2, 2#4 ;TRAP VECTOR SETUP
MOV #340, 2#6 ;TRAP STATUS SETUP
EIGHT: ADC 2#37770 ;CHECK FOR 8K
NOP
ADC 2#57770 ;CHECK FOR 12K
NOP
ADC 2#077770 ;CHECK FOR 16K
NOP
ADC 2#117770 ;CHECK FOR 20K
NOP
ADC 2#137770 ;CHECK FOR 24K
NOP
ADC 2#157770 ;CHECK FOR 28K
BR STRT28
DET2: MOV (6)+, %2 ;RETRIEVE TRAP PC
TST (6)+ ;DISCARD TRAP STATUS WORD
ADD #STRT4-EIGHT-4, R2
JMP 2R2
MOVE: CLR %0 ;SET UP MAIN CORE POINTER
MOV %1, %2
ADD #0+2, %2 ;SET UP MAX CORE MOVE
MOV (0)+, (1)+ ;MOVE WORD
CMP %2, %1 ;MOVE COMPLETE?
BNE .-4 ;MOVE ANOTHER WORD
RTS %7 ;MOVE COMPLETE
STRT4: BR DET3
NOP
NOP
JSR %7, XFER8 ;START 8K TRANSFER

```

```

1196 002720 000506 BR MOD4 ;START 4K MODIFY
1197 002722 004767 000072 JSR %7,XFER12 ;START 12K TRANSFER
1198 002726 000475 BR MOD8 ;START 8K MODIFY
1199 002730 004767 000054 JSR %7,XFER16 ;START 16K TRANSFER
1200 002734 000464 BR MOD12 ;START 12K MODIFY
1201 002736 004767 000036 JSR %7,XFER20 ;START 20K TRANSFER
1202 002742 000453 BR MOD16 ;START 16K MODIFY
1203 002744 004767 000020 JSR %7,XFER24 ;START 24K TRANSFER
1204 002750 000442 BR MOD20 ;START 20K MODIFY
1205 002752 004767 000002 STRT28: JSR %7,XFER28 ;START 28K TRANSFER
1206 002756 000431 BR MOD24 ;START 24K MODIFY
1207 002760 012701 140000 XFER28: MOV #140000,%1 ;SET UP MOVE START LOCATION
1208 002764 004767 177676 JSR %7,MOVE ;GO TO MOVE SUBROUTINE
1209 002770 012701 120000 XFER24: MOV #120000,%1
1210 002774 004767 177666 JSR %7,MOVE
1211 003000 012701 100000 XFER20: MOV #100000,%1
1212 003004 004767 177656 JSR %7,MOVE
1213 003010 012701 060000 XFER16: MOV #60000,%1
1214 003014 004767 177646 JSR %7,MOVE
1215 003020 012701 040000 XFER12: MOV #40000,%1
1216 003024 004767 177636 JSR %7,MOVE
1217 003030 012701 020000 XFER8: MOV #20000,%1
1218 003034 004767 177626 JSR %7,MOVE
1219 003040 000207 RTS ;RETURN FROM TRANSFERS
1220 003042 012767 000137 131570 MOD24: MOV #137,DONE+120000
1221 003050 012767 145622 131564 MOV #BEGINX+140000,DONE+120002
1222 003056 012767 000137 111554 MOD20: MOV #137,DONE+100000
1223 003064 012767 125622 111550 MOV #BEGINX+120000,DONE+100002
1224 003072 012767 000137 071540 MOD16: MOV #137,DONE+60000
1225 003100 012767 105622 071534 MOV #BEGINX+100000,DONE+60002
1226 003106 012767 000137 051524 MOD12: MOV #137,DONE+40000
1227 003114 012767 065622 051520 MOV #BEGINX+60000,DONE+40002
1228 003122 012767 000137 031510 MOD8: MOV #137,DONE+20000
1229 003130 012767 045622 031504 MOV #BEGINX+40000,DONE+20002
1230 003136 012767 000137 011474 MOD4: MOV #137,DONE
1231 003144 012767 025622 011470 MOV #BEGINX+20000,DONE+2
1232 003152 005037 000006 DET3: CLR #6
1233 003156 012737 000006 000004 MOV #6,#4
1234 003164 000207 RTS %7
1235
1236 ;*****
1237 .SBTTL TTY TRANSMITTER PRINT VALUES 0 TO 377
1238 ;*****
1239 003166 005027 000000 TYOUT: CLR #0 ;INITAL DATA
1240 003170 016777 DATA2=-2
1241 003172 016777 177772 175212 TYOUT1: MOV DATA2,@TTDBR ;OUTPUT TO DEVICE
1242 003200 000002 RTI ;RETURN TO MAINLINE**
1243 003202 017767 175202 175210 TYOUTR: MOV @TTCSR,TTSV
1244 003210 105767 175204 TSTB TTSV ;TEST FOR DONE
1245 003214 100401 BMI .+4 ;BRANCH IF FLAG FOUND
1246 003216 104006 HLT ;FALSE INTERRUPT RETURN
1247 003220 005267 177744 INC DATA2 ;INCREMENT DATA
1248 003224 022767 000400 177736 CMP #400,DATA2 ;TEST DATA FOR UPPER LIMIT
1249 003232 001755 BEQ TYOUT ;AT UPPER LIMIT START OVER
1250 003234 000756 BR TYOUT1 ;FINISH REST OF DATA
1251

```

E03

DBKTGD MAINDEC-11-DBKTG-D KT11-D EXERCISER
DBKTGD.P11 21-SEP-76 11:27

MACY11 27(1006) 21-SEP-76 11:30 PAGE 30
TTY TRANSMITTER PRINT VALUES 0 TO 377

```

1252 :*****
1253 :SBTTL TEST OF LINE CLOCK, INTERRUPT FOR 55 SECONDS THEN STALL FOR 5 SECONDS.
1254 :*****
1255 003236 005037 003354 LK1: CLR @TIME ;CLEAR LINE CLOCK TIMER
1256 003242 052777 000100 175156 BIS #100,@LKCSR
1257 003250 052737 000100 177776 BIS #100,@PSR
1258 003256 000002 LK2: RTI
1259 003260 105777 175142 LK3: TSTB @LKCSR
1260 003264 100401 BMI .+4
1261 003266 104006 HLT ;FALSE INTERRUPT
1262 003270 042777 000200 175130 BIC #200,@LKCSR
1263 003276 005237 003354 LK4: INC @TIME ;HERE ON INTERRUPTS
1264 003302 022737 006344 003354 CMP #3300.,@TIME ;55 SEC YET?
1265 003310 103362 BHIS LK2 ;BR IF NOT
1266 003312 042777 000100 175106 BIC #100,@LKCSR
1267 003320 042737 000100 177776 BIC #100,@PSR ;LOWER PRIORITY
1268 003326 022737 007020 003354 CMP #3600.,@TIME ;ONE MINUTE YET
1269 003334 001740 BEQ LK1 ;YES RESET TIMER
1270 003336 105777 175064 TSTB @LKCSR ;NO, SKIP TILL MINUTE UP
1271 003342 100375 BPL -4
1272 003344 042777 000200 175054 BIC #200,@LKCSR ;CLEAR FLAG
1273 003352 000751 BR LK4
1274 003354 000000 TIME: 0
1275
1276 :*****
1277 :SBTTL LINE PRINTER SERVICE
1278 :*****
1279 :LINE PRINTER SHOULD RAISE PROCESSOR PRIORITY TO LEVEL OF LINE PRINTER/
1280 :INTERRUPT VECTOR IS 200/
1281 003356 012727 000000 000000 LP1: MOV #0,#0 ;START OF LINE TO CURRENT
1282 003362 CURPAT=-2 ;CHARACTER BEING PRINTED
1283 003360 SOLPAT=-4 ;START OF LINE CHARACTER
1284 003364 016777 177772 175040 LP2: MOV CURPAT,@LPDBR ;CURRENT PATTERN TO LINE PRINTER
1285 003372 105777 175032 TSTB @LPCSR
1286 003376 100420 BMI LP6
1287 003400 000002 RTI ;RETURN TO MAIN LINE
1288 003402 105777 175022 LPINTR: TSTB @LPCSR ;TEST FOR FLAG
1289 003406 100414 BMI LP6
1290 003410 005737 000042 TST @#42 ;MONITOR LOAD
1291 003414 001410 BEQ LP7 ;NO ERROR
1292 003416 032777 100000 175004 BIT #100000,@LPCSR ;YES, IS ERROR SET
1293 003424 001404 BEQ LP7 ;NO ERROR
1294 003426 042777 000100 174774 BIC #100,@LPCSR ;DIS ABLE INTERRUPT
1295 003434 000002 RTI
1296 003436 104006 HLT ;FALSE RETURN FROM MAIN LINE
1297 003440 026727 000006 000117 LP6: CMP CLINCT,#79. ;TEST FOR END OF LINE
1298 003446 001415 BEQ LP4 ;GO GENERATE CR/LF
1299 003450 005227 000000 INC #0 ;INCREMENT LINE POSITION COUNT
1300 003452 CLINCT=-2 ;POSITION OF LINE
1301 003454 026727 177702 000137 CMP CURPAT,#137 ;TEST FOR MAXIMUM PATTERN
1302 003462 001403 BEQ LP3 ;YES - GO TO LP3 AND RESET
1303 003464 005267 177672 INC CURPAT ;NO - INCREMENT TO NEXT PATTERN
1304 003470 000735 BR LP2 ;GO SEND IT TO LINE PRINTER
1305 003472 012767 000040 177662 LP3: MOV #40,CURPAT ;RESET PATTERN AND SEND TO PRINTER
1306 003500 000731 BR LP2 ;SENT TO LINE PRINTER
1307 003502 005067 177744 LP4: CLR CLINCT ;RESET LINE COUNT

```

```

1308 003506 012777 000012 174716      MOV      #12, @LPDBR      ;LINE FEED
1309 003514 105777 174710      TSTB    @LPCSR
1310 003520 100375      BPL     -4
1311 003522 026727 177632 000137      CMP     SOLPAT, #137    ;START OF LINE PATTERN
1312 003530 001403      BEQ     LPS
1313 003532 005267 177622      INC     SOLPAT          ;INCREMENT START OF LINE
1314 003536 000707      BR     LPI
1315 003540 012767 000040 177612 LPS:    MOV     #40, SOLPAT     ;RESET START OF LINE
1316 003546 000703      BR     LPI             ;PRINT
1317
1318 ;*****
1319 .SBTTL RK11 SERVICE
1320 ;*****
1321 ;RK11 DISK TEST INTERRUPT LEVEL 5, 2000 WORD TRANSFERS
1322 003550 005077 174706      RKSTART: CLR      @RKDAE ;INIT
1323 003554 013777 004002 174704      RK1:    MOV     @LLIMIT, @RKBAR ;CORE BASE
1324 003562 013777 004004 174674      MOV     @WORDCT, @RKWC    ;TRANSFER LENGTH
1325 003570 113777 003642 174672      MOV     @RKFUNCT, @RKCSR ;WRITE OR WRITE CK TO DISK
1326 003576 000002      RTI
1327 003600 032777 100200 174662      IRK:    BIT     #100200, @RKCSR ;INTERRUPT RETURN
1328 003606 003002      BGT     +6
1329 003610 104006      HLT
1330 003612 000756      BR     RKSTART
1331 003614 032777 000037 174640      BIT     #37, @RKDAE      ;DISK AT UPPER LIMIT?
1332 003622 001354      BNE     RK1
1333 003624 122777 000031 174626      CMP     #31, @RKDAH
1334 003632 001350      BNE     RK1
1335 003634 000337 003642      SWAB   @RKFUNCT        ;CHANGE COMMAND
1336 003640 000743      BR     RKSTART        ;RESTART NEW TRANSFER OF DISK
1337 003642 000000      RKFUNCT: 0
1338
1339 ;*****
1340 .SBTTL RF11 DISK
1341 ;*****
1342 003644 105277 174602      RFSTART: INCB    @RFCSRH    ;INITIALIZE DISK - DAR-DAE
1343 003650 013777 004002 174570      RF1:    MOV     @LLIMIT, @RFCAR ;CORE BASE
1344 003656 013777 004004 174560      MOV     @WORDCT, @RFCWC    ;LENGTH OF TRANSFER
1345 003664 113777 004000 174556      MOV     @RFFUNCT, @RFCSR  ;WRITE OR WRITE CHECK TO DISK
1346 003672 000002      RTI
1347 003674 105777 174550      IRF:    TSTB   @RFCSR      ;RETURN TO MAINLINE CODE
1348 003700 100402      BMI     +6              ;INTERRUPT VECTOR POINTS HERE
1349 003702 104006      HLT
1350 003704 000757      BR     RFSTART        ;RF11 READY NOT UP
1351 003706 005777 174536      TST     @RFCSR
1352 003712 100012      BPL     ERROK          ;ERROR SET?
1353 003714 032777 020000 174526      BIT     #20000, @RFCSR   ;BRANCH IF NOT
1354 003722 001404      BEQ     ERRSET        ;YES-WRITE CHECK ERROR?
1355 003724 104006      HLT
1356 003726 000337 004000      SWAB   @RFFUNCT        ;NO-BRANCH
1357 003732 000744      BR     RFSTART        ;YES-RF11 WRITE CHECK ERROR
1358 003734 104006      ERRSET: HLT
1359 003736 000742      ERROK:  TST     RFSTART  ;CHANGE COMMAND TO DO WRITE
1360 003740 005777 174500      ERROK:  TST     @RFCWC    ;RF11 ERROR SET-NOT WRITE CHECK
1361 003744 100002      BPL     +6
1362 003746 104006      HLT
1363 003750 000735      BR     RFSTART

```


1364	003752	122777	000003	174460	CMPB	#3,@RFDAR	:DISK AT UPPER LIMIT? 7=2, 17=4, 37=8
1365	003760	001333			BNE	RF1	:NO
1366	003762	027727	174454	174000	CMP	@RFDAR,#174000	:AS FAR ON DISK AS WE CAN GO
1367	003770	101727			BLOS	RF1	:NO
1368	003772	000337	004000		SWAB	@#RFFUNCT	:CHANGE COMMAND
1369	003776	000722			BR	RFSTART	:RESTART NEW TRANSFER OF DISK
1370	004000	000000			RFFUNCT:	0	:DISK COMMAND
1371	004002	005654			LLIMIT:	BEGIN	:FIRST CORE ADDRESS OF TRANSFER
1372	004004	176000			WORDCT:	-2000	:LENGTH OF TRANSFER

```

:*****
:SBTTL TC11 DIAGNOSTIC ROUTINE
:*****
:DECTAPE DIAGNOSTIC ROUTINE. THE TAPE IS FIRST DRIVEN TO THE FORWARD
:END ZONE. THE DESIRED DATA IS THEN GENERATED IN THE DECTAPE BUFFER AREA
:AND DATA IS WRITTEN ONTO ALL BLOCKS FROM THE BLOCK NUMBER IN TCFRST
:THRU THE BLOCK NUMBER IN TCLAST. BLOCK NUMBERS ARE ALSO CHECKED FOR
:BEING IN ORDER. AFTER THE BLOCK NUMBER IN TCLAST IS WRITTEN, TAPE IS
:DRIVEN INTO THE REVERSE END ZONE.
:THE TAPE IS THEN STARTED IN REVERSE, AND WHEN THE CLOSEST BLOCK THAT
:WAS WRITTEN (TCLAST) IS FOUND, IT IS READ INTO THE DECTAPE BUFFER AREA.
:THE PROGRAM INTERRUPT REQUEST FACILITY IS THEN USED TO BOOK A REQUEST
:FOR CHECKING THE DATA AT LEVEL 3, AND NO FURTHER DATA IS READ IN
:UNTIL THAT DATA HAS BEEN CHECKED. AFTER IT IS CHECKED, THE DATA IS
:SCRAMBLED TO GUARANTEE THAT NEW DATA IS REALLY READ IN NEXT TIME. WHILE
:THIS IS GOING ON, BLOCK NUMBERS ARE CHECKED FOR BEING IN ORDER AS THE
:TAPE TRAVELS TOWARD THE FORWARD END ZONE.ONCE THE DATA IS FULLY CHECKED
:THE NEXT BLOCK THAT COMES UP IS READ IN AND THE PROCESS REPEATED. ONCE
:THE BLOCK WHOSE NUMBER IS IN TCFRST HAS BEEN READ, THE TAPE IS DRIVEN
:INTO THE FORWARD END ZONE AND THE WHOLE SEQUENCE IS REPEATED.

```

```

:FUNCTION VALUES IN CSR
:DT11 DEC TAPE
RD=4
WD=14
RB=2
IE=500
DO=1
R=4000

```

```

:READ DATA
:WRITE DATA
:INTERRUPT ENABLE+UNIT 1
:DO - THE FUNCTION
:REVERSE

```

1404	004006	000000			TCFIRST:	0	:FIRST BLOCK TO BE SEARCHED FOR
1405	004010	001101			TCLAST:	577.	:LAST BLOCK TO BE SEARCHED FOR
1406	004012	000000			TCEXPE:	0	:THE BLOCK THAT IS EXPECTED
1407							
1408					:GO TO FORWARD END ZONE		
1409	004014	012777	004014	174544	FENDZ:	MOV #FENDZ,@TCIV	:END ZONE VECTOR SETUP
1410	004022	005777	174530			TST @TCST	:TEST FOR END ZONE
1411	004026	100403				BMI FEND1	:AT END ZONE?
1412	004030	105277	174520			INCB @TCCM	:SET DO - NO DELAY
1413	004034	000002				RTI	:NO - WAIT SOME MORE
1414	004036	012777	004066	174522	FEND1:	MOV #TCF1,@TCIV	:YES - NEW VECTOR
1415	004044	042777	104000	174502		BIC #104000,@TCCM	:SEARCH BLOCK FOWARD
1416	004052	016767	177730	177732		MOV TCFIRST,TCEXPE	:COUNT WHEN THIS BLOCK IS FOUND
1417	004060	105277	174470		TCF1A:	INCB @TCCM	:SET DO
1418	004064	000002				RTI	:RETURN ON NEXT BLOCK
1419	004066	032777	100200	174460	TCF1:	BIT #100200,@TCCM	:ANY ERROR ON READ?

```

1420 004074 100001          BPL      .+4
1421 004076 104006          HLT
1422 004100 001001          BNE      .+4
1423 004102 104006          HLT
1424 004104 027767 174450 177700  CMP      @TCDT,TCEXPE
1425 004112 002762          BLT      TCF1A
1426 004114 001401          BEQ      TCF2
1427 004116 104006          HLT
1428
1429 004120 012777 004134 174440 TCF2:  MOV      #TCF3,@TCIV
1430 004126 105277 174422          INCB    @TCCM
1431 004132 000002          RTI
1432
1433          ;FIND SEQUENTIAL BLOCK AT FOWARD DIRECTION
1434 004134 032777 100200 174412 TCF3:  BIT      #100200,@TCCM
1435 004142 100001          BPL      .+4
1436 004144 104006          HLT
1437 004146 001001          BNE      .+4
1438 004150 104006          HLT
1439 004152 027767 174402 177630  CMP      @TCDT,TCLAST
1440 004160 001414          BEQ      RENDZ
1441 004162 005267 177624          INC      TCXPE
1442 004166 027767 174366 177616  CMP      @TCDT,TCXPE
1443 004174 001401          BEQ      .+4
1444 004176 104006          HLT
1445 004200 000427          BR      TCWBK
1446 004202 105277 174346          TCF4:  INCB    @TCCM
1447 004206 000002          RTI
1448 004210 000701          XFENDZ: BR      FENDZ
1449
1450          ;MOVE TAPE TO REVERSE END ZONE
1451 004212 012777 004212 174346 RENDZ:  MOV      #RENDZ,@TCIV
1452 004220 016767 177564 177564  MOV      TCLAST,TCXPE
1453 004226 005777 174324          TST     @TCST
1454 004232 100403          BMI     REND1
1455 004234 105277 174314          INCB    @TCCM
1456 004240 000002          RTI
1457 004242 012777 004503 174304 REND1:  MOV      #R+IE+RB+DO,@TCCM
1458 004250 012777 004340 174310  MOV      #TCR1,@TCIV
1459 004256 000002          RTI
1460          ;WRITE FORWARD ALL BLOCKS EXCEPT 0
1461
1462 004260 012777 004312 174300 TCWBK:  MOV      #TCWB1,@TCIV
1463 004266 012777 177400 174266  MOV      #-400,@TCWC
1464 004274 012777 004622 174262  MOV      #TCWBUF,@TCBA
1465 004302 1.2777 000515 174244  MOVB    #IE+WD+DO,@TCCM
1466 004310 000002          RTI
1467 004312 005777 174236          TCWB1:  TST     @TCCM
1468 004316 100001          BPL      .+4
1469 004320 104006          HLT
1470 004322 012777 004134 174236  MOV      #TCF3,@TCIV
1471 004330 112777 000502 174216  MOVB    #IE+RB,@TCCM
1472 004336 000721          BR      TCF4
1473
1474 004340 032777 100200 174206 TCR1:  BIT      #100200,@TCCM
1475 004346 100001          BPL      .+4

```

```

;TC ERROR SET - FORWARD READ BLOCK
;DONE FLAG UP?
;FALSE INTERRUPT
;IS THIS OUR BLOCK FOR SYNC
;NO-READ SOME MORE BLOCKS
;YES
;WE PASSED THE BLOCK

```

```

;VECTOR FOR SEQUENTIAL READS
;SET DO
;RETURN AND TEST SEQENTIAL BLOCKS

```

```

;FIND SEQUENTIAL BLOCK AT FOWARD DIRECTION
;TEST ERROR AND READY

```

```

;FOWARD READ ERROR TC-11

```

```

;FALSE INTERRUPT ON TC-11
;HAVE WE TESTED ALL BLOCKS
;YES DRIVE UNIT IN END ZONE TO START OVER
;NO-INCREMENT EXPECTED COUNT
;IS CURRENT BLOCK CORRECT

```

```

;FAILED IN FOWARD READ TO FIND NEXT BLOCK
;THIS ROUTINE WRITES A BLOCK
;SET DO

```

```

;INDIRECT LINK

```

```

;END ZONE VECTOR SETUP
;SET UP FOR REVERSE SEARCH
;IN END ZONE
;YES - START TO TURN UNIT AROUND
;SET DO
;NO - WAIT TILL WE ARE
;FUNCTION = READ BLOCK, REVERSE AND GO
;SET UP NEW INTERRUPT VECTOR

```

```

;INTERRUPT VECTOR FOR WRITE
;ONE BLOCK
;THE WRITE BUFFER ADDRESS
;WRITE THE BLOCK
;RETURN WHEN BLOCK IS WRITTEN
;ANY ERRORS

```

```

;SEARCH BLOCK VECTOR
;READ BLOCK
;FIND THE NEXT BLOCK

```

```

;TEST FOR ERROR AND READY

```

```

1476 004350 104006 HLT ;DECTAPE ERROR ON READ BLOCK REVERSE
1477 004352 001001 BNE .+4
1478 004354 104006 HLT ;FALSE INTERRUPT FROM DECTAPE
1479 004356 027767 174176 177426 CMP @TCDT,TCEXPE ;IS IT OUR FIRST BLOCK
1480 004364 001406 BEQ TCR2 ;YES - GO TEST THE REST
1481 004366 002002 BGE TCR1A ;NO - HAVE WE PASSED THE BLOCK
1482 004370 104006 HLT ;WE PASS OUR BLOCK
1483 004372 000707 BR RENDZ ;GO TO END ZONE AND TRY AGAIN
1484 004374 105277 174154 TCR1A: INCB @TCCM ;SET DO
1485 004400 000002 RTI ;WE FOUND OUR FIRST BLOCK
1486 004402 012777 004416 174156 TCR2: MOV #TCR3,@TCIV ;SET UP INTERRUPT TO TEST ALL BLOCKS
1487 004410 105277 174140 INCB @TCCM ;SET DO
1488 004414 000002 RTI ;WAIT FOR NEXT BLOCK TO INTERRUPT
1489
1490 ;FIND SEQUENTIAL BLOCK IN REVERSE DIRECTION
1491 004416 032777 100200 174130 TCR3: BIT #100200,@TCCM ;TEST FOR READ AND ERROR
1492 004424 100001 BPL .+4
1493 004426 104006 HLT ;ERROR READING SEQUENTIAL BLOCK IN REVERSE
1494 004430 001001 BNE .+4
1495 004432 104006 HLT ;FALSE DECTAPE INTERRUPT
1496 004434 026777 177346 174116 CMP TCFIRST,@TCDT ;DID WE DO ALL THE BLOCKS
1497 004442 001662 BEQ XFENDZ ;YES - GO TO END ZONE TO RESTART
1498 004444 005367 177342 DEC TCXPE ;NO - DECREMENT BLOCK NUMBER
1499 004450 027767 174104 177334 CMP @TCDT,TCXPE ;TEST SEQUENTIAL BLOCK IN REVERSE
1500 004456 001401 BEQ .+4
1501 004460 104006 HLT ;TEST SEQUENTIAL READ BLOCK IN REVERSE FAILED
1502 004462 000403 BR TCRBK ;THIS ROUTINE READ A BLOCK
1503 004464 105277 174064 TCR4: INCB @TCCM ;SET DO
1504 004470 000002 RTI ;LETS TRY A NEW BLOCK
1505
1506 ;READ REVERSE ALL BLOCK EXCEPT BLOCK 1101
1507 004472 012777 004530 174066 TCRBK: MOV #TCRB1,@TCIV ;SET UP INTERRUPT VECTOR
1508 004500 012777 177400 174054 MOV #-400,@TCWC ;READ ONE BLOCK
1509 004506 012777 004622 174050 MOV #TCRBUF,@TCBA ;WHERE BUFFER IS
1510 004514 112777 000505 174032 MOVB #IE+RD+DO,@TCCM ;READ THE BLOCK
1511 004522 004767 000030 JSR %7,TC1 ;CHECK DATA BUFFER
1512 004526 000002 RTI ;EXIT - RETURN WHEN BLOCK IS READ
1513 004530 005777 174020 TCRB1: TST @TCCM ;AND ERRORS
1514 004534 100001 BPL .+4
1515 004536 104006 HLT ;DECTAPE ERROR
1516 004540 012777 004416 174020 MOV #TCR3,@TCIV ;NEW VECTOR FOR BLOCK SEARCH
1517 004546 112777 000502 174000 MOVB #IE+RB,@TCCM ;READ BLOCK FUNCTION
1518 004554 000743 BR TCR4 ;RETURN TO BLOCK SEARCH
1519
1520 ;THIS ROUTINE CHECKS THE READ DATA BUFFER TC11
1521 ;BY DOING A CHECK SUM ON THE DATA
1522 004556 010146 TC1: MOV %1,-(6) ;SAVE THESE ON THE STACK
1523 004560 010246 MOV %2,-(6)
1524 004562 010346 MOV %3,-(6)
1525 004564 005003 CLR %3 ;SUM OF DATA
1526 004566 012701 004622 MOV #TCRBUF,%1 ;ADDRESS OF READ BUFFER
1527 004572 012702 005622 MOV #TCRBUF+1000,%2 ;END OF READ BUFFER
1528 004576 062103 TC2: ADD (1)+,%3 ;EVEN ADD
1529 004600 062103 ADD (1)+,%3 ;ODD ADD -2'S COMPLIMENT
1530 004602 001401 BEQ .+4
1531 004604 104006 HLT ;DATA ERROR TC-11
    
```

```

1532 004606 020102      CMP      %1,%2      ;AT END OF BUFFER?
1533 004610 001372      BNE      TC2        ;NO - SUM THE REST
1534 004612 012603      MOV      (6)+,%3    ;RESTORE THE REGISTERS
1535 004614 012602      MOV      (6)+,%2
1536 004616 012601      MOV      (6)+,%1
1537 004620 000207      RTS      %7        ;EXIT
    
```

: THIS WRITE BUFFER LOOK THE SAME FORWARD OR REVERSE

```

1538
1539
1540 004622      TCWBUF:
1541 004622      TCRBUF:
1542           000001      N=1
1543 004622 000001      N          ;DECTAPE WRITE BUFFER
1544 004624 177777      -N
1545           000002      N=N+1
1546 004626 000002      N          ;DECTAPE WRITE BUFFER
1547 004630 177776      -N
1548           000003      N=N+1
1549 004632 000003      N          ;DECTAPE WRITE BUFFER
1550 004634 177775      -N
1551           000004      N=N+1
1552 004636 000004      N          ;DECTAPE WRITE BUFFER
1553 004640 177774      -N
1554           000005      N=N+1
1555 004642 000005      N          ;DECTAPE WRITE BUFFER
1556 004644 177773      -N
1557           000006      N=N+1
1558 004646 000006      N          ;DECTAPE WRITE BUFFER
1559 004650 177772      -N
1560           000007      N=N+1
1561 004652 000007      N          ;DECTAPE WRITE BUFFER
1562 004654 177771      -N
1563           000010      N=N+1
1564 004656 000010      N          ;DECTAPE WRITE BUFFER
1565 004660 177770      -N
1566           000011      N=N+1
1567 004662 000011      N          ;DECTAPE WRITE BUFFER
1568 004664 177767      -N
1569           000012      N=N+1
1570 004666 000012      N          ;DECTAPE WRITE BUFFER
1571 004670 177766      -N
1572           000013      N=N+1
1573 004672 000013      N          ;DECTAPE WRITE BUFFER
1574 004674 177765      -N
1575           000014      N=N+1
1576 004676 000014      N          ;DECTAPE WRITE BUFFER
1577 004700 177764      -N
1578           000015      N=N+1
1579 004702 000015      N          ;DECTAPE WRITE BUFFER
1580 004704 177763      -N
1581           000016      N=N+1
1582 004706 000016      N          ;DECTAPE WRITE BUFFER
1583 004710 177762      -N
1584           000017      N=N+1
1585 004712 000017      N          ;DECTAPE WRITE BUFFER
1586 004714 177761      -N
1587           000020      N=N+1
    
```

1588	004716	000020	N	;DECTAPE WRITE BUFFER
1589	004720	177760	-N	
1590		000021	N=N+1	
1591	004722	000021	N	;DECTAPE WRITE BUFFER
1592	004724	177757	-N	
1593		000022	N=N+1	
1594	004726	000022	N	;DECTAPE WRITE BUFFER
1595	004730	177756	-N	
1596		000023	N=N+1	
1597	004732	000023	N	;DECTAPE WRITE BUFFER
1598	004734	177755	-N	
1599		000024	N=N+1	
1600	004736	000024	N	;DECTAPE WRITE BUFFER
1601	004740	177754	-N	
1602		000025	N=N+1	
1603	004742	000025	N	;DECTAPE WRITE BUFFER
1604	004744	177753	-N	
1605		000026	N=N+1	
1606	004746	000026	N	;DECTAPE WRITE BUFFER
1607	004750	177752	-N	
1608		000027	N=N+1	
1609	004752	000027	N	;DECTAPE WRITE BUFFER
1610	004754	177751	-N	
1611		000030	N=N+1	
1612	004756	000030	N	;DECTAPE WRITE BUFFER
1613	004760	177750	-N	
1614		000031	N=N+1	
1615	004762	000031	N	;DECTAPE WRITE BUFFER
1616	004764	177747	-N	
1617		000032	N=N+1	
1618	004766	000032	N	;DECTAPE WRITE BUFFER
1619	004770	177746	-N	
1620		000033	N=N+1	
1621	004772	000033	N	;DECTAPE WRITE BUFFER
1622	004774	177745	-N	
1623		000034	N=N+1	
1624	004776	000034	N	;DECTAPE WRITE BUFFER
1625	005000	177744	-N	
1626		000035	N=N+1	
1627	005002	000035	N	;DECTAPE WRITE BUFFER
1628	005004	177743	-N	
1629		000036	N=N+1	
1630	005006	000036	N	;DECTAPE WRITE BUFFER
1631	005010	177742	-N	
1632		000037	N=N+1	
1633	005012	000037	N	;DECTAPE WRITE BUFFER
1634	005014	177741	-N	
1635		000040	N=N+1	
1636	005016	000040	N	;DECTAPE WRITE BUFFER
1637	005020	177740	-N	
1638		000041	N=N+1	
1639	005022	000041	N	;DECTAPE WRITE BUFFER
1640	005024	177737	-N	
1641		000042	N=N+1	
1642	005026	000042	N	;DECTAPE WRITE BUFFER
1643	005030	177736	-N	

1644		000043	N=N+1	
1645	005032	000043	N	;DECTAPE WRITE BUFFER
1646	005034	177735	-N	
1647		000044	N=N+1	
1648	005036	000044	N	;DECTAPE WRITE BUFFER
1649	005040	177734	-N	
1650		000045	N=N+1	
1651	005042	000045	N	;DECTAPE WRITE BUFFER
1652	005044	177733	-N	
1653		000046	N=N+1	
1654	005046	000046	N	;DECTAPE WRITE BUFFER
1655	005050	177732	-N	
1656		000047	N=N+1	
1657	005052	000047	N	;DECTAPE WRITE BUFFER
1658	005054	177731	-N	
1659		000050	N=N+1	
1660	005056	000050	N	;DECTAPE WRITE BUFFER
1661	005060	177730	-N	
1662		000051	N=N+1	
1663	005062	000051	N	;DECTAPE WRITE BUFFER
1664	005064	177727	-N	
1665		000052	N=N+1	
1666	005066	000052	N	;DECTAPE WRITE BUFFER
1667	005070	177726	-N	
1668		000053	N=N+1	
1669	005072	000053	N	;DECTAPE WRITE BUFFER
1670	005074	177725	-N	
1671		000054	N=N+1	
1672	005076	000054	N	;DECTAPE WRITE BUFFER
1673	005100	177724	-N	
1674		000055	N=N+1	
1675	005102	000055	N	;DECTAPE WRITE BUFFER
1676	005104	177723	-N	
1677		000056	N=N+1	
1678	005106	000056	N	;DECTAPE WRITE BUFFER
1679	005110	177722	-N	
1680		000057	N=N+1	
1681	005112	000057	N	;DECTAPE WRITE BUFFER
1682	005114	177721	-N	
1683		000060	N=N+1	
1684	005116	000060	N	;DECTAPE WRITE BUFFER
1685	005120	177720	-N	
1686		000061	N=N+1	
1687	005122	000061	N	;DECTAPE WRITE BUFFER
1688	005124	177717	-N	
1689		000062	N=N+1	
1690	005126	000062	N	;DECTAPE WRITE BUFFER
1691	005130	177716	-N	
1692		000063	N=N+1	
1693	005132	000063	N	;DECTAPE WRITE BUFFER
1694	005134	177715	-N	
1695		000064	N=N+1	
1696	005136	000064	N	;DECTAPE WRITE BUFFER
1697	005140	177714	-N	
1698		000065	N=N+1	
1699	005142	000065	N	;DECTAPE WRITE BUFFER

1700	005144	177713	-N	
1701		000066	N=N+1	
1702	005146	000066	N	;DECTAPE WRITE BUFFER
1703	005150	177712	-N	
1704		000067	N=N+1	
1705	005152	000067	N	;DECTAPE WRITE BUFFER
1706	005154	177711	-N	
1707		000070	N=N+1	
1708	005156	000070	N	;DECTAPE WRITE BUFFER
1709	005160	177710	-N	
1710		000071	N=N+1	
1711	005162	000071	N	;DECTAPE WRITE BUFFER
1712	005164	177707	-N	
1713		000072	N=N+1	
1714	005166	000072	N	;DECTAPE WRITE BUFFER
1715	005170	177706	-N	
1716		000073	N=N+1	
1717	005172	000073	N	;DECTAPE WRITE BUFFER
1718	005174	177705	-N	
1719		000074	N=N+1	
1720	005176	000074	N	;DECTAPE WRITE BUFFER
1721	005200	177704	-N	
1722		000075	N=N+1	
1723	005202	000075	N	;DECTAPE WRITE BUFFER
1724	005204	177703	-N	
1725		000076	N=N+1	
1726	005206	000076	N	;DECTAPE WRITE BUFFER
1727	005210	177702	-N	
1728		000077	N=N+1	
1729	005212	000077	N	;DECTAPE WRITE BUFFER
1730	005214	177701	-N	
1731		000100	N=N+1	
1732	005216	000100	N	;DECTAPE WRITE BUFFER
1733	005220	177700	-N	
1734		000101	N=N+1	
1735		000100	N=N-1	
1736	005222	177700	-N	
1737	005224	000100	N	;DEC TAPE WRITE BUFFER
1738		000077	N=N-1	
1739	005226	177701	-N	
1740	005230	000077	N	;DEC TAPE WRITE BUFFER
1741		000076	N=N-1	
1742	005232	177702	-N	
1743	005234	000076	N	;DEC TAPE WRITE BUFFER
1744		000075	N=N-1	
1745	005236	177703	-N	
1746	005240	000075	N	;DEC TAPE WRITE BUFFER
1747		000074	N=N-1	
1748	005242	177704	-N	
1749	005244	000074	N	;DEC TAPE WRITE BUFFER
1750		000073	N=N-1	
1751	005246	177705	-N	
1752	005250	000073	N	;DEC TAPE WRITE BUFFER
1753		000072	N=N-1	
1754	005252	177706	-N	
1755	005254	000072	N	;DEC TAPE WRITE BUFFER

1756		000071	N=N-1	
1757	005256	177707	-N	
1758	005260	000071	N	;DEC TAPE WRITE BUFFER
1759		000070	N=N-1	
1760	005262	177710	-N	
1761	005264	000070	N	;DEC TAPE WRITE BUFFER
1762		000067	N=N-1	
1763	005266	177711	-N	
1764	005270	000067	N	;DEC TAPE WRITE BUFFER
1765		000066	N=N-1	
1766	005272	177712	-N	
1767	005274	000066	N	;DEC TAPE WRITE BUFFER
1768		000065	N=N-1	
1769	005276	177713	-N	
1770	005300	000065	N	;DEC TAPE WRITE BUFFER
1771		000064	N=N-1	
1772	005302	177714	-N	
1773	005304	000064	N	;DEC TAPE WRITE BUFFER
1774		000063	N=N-1	
1775	005306	177715	-N	
1776	005310	000063	N	;DEC TAPE WRITE BUFFER
1777		000062	N=N-1	
1778	005312	177716	-N	
1779	005314	000062	N	;DEC TAPE WRITE BUFFER
1780		000061	N=N-1	
1781	005316	177717	-N	
1782	005320	000061	N	;DEC TAPE WRITE BUFFER
1783		000060	N=N-1	
1784	005322	177720	-N	
1785	005324	000060	N	;DEC TAPE WRITE BUFFER
1786		000057	N=N-1	
1787	005326	177721	-N	
1788	005330	000057	N	;DEC TAPE WRITE BUFFER
1789		000056	N=N-1	
1790	005332	177722	-N	
1791	005334	000056	N	;DEC TAPE WRITE BUFFER
1792		000055	N=N-1	
1793	005336	177723	-N	
1794	005340	000055	N	;DEC TAPE WRITE BUFFER
1795		000054	N=N-1	
1796	005342	177724	-N	
1797	005344	000054	N	;DEC TAPE WRITE BUFFER
1798		000053	N=N-1	
1799	005346	177725	-N	
1800	005350	000053	N	;DEC TAPE WRITE BUFFER
1801		000052	N=N-1	
1802	005352	177726	-N	
1803	005354	000052	N	;DEC TAPE WRITE BUFFER
1804		000051	N=N-1	
1805	005356	177727	-N	
1806	005360	000051	N	;DEC TAPE WRITE BUFFER
1807		000050	N=N-1	
1808	005362	177730	-N	
1809	005364	000050	N	;DEC TAPE WRITE BUFFER
1810		000047	N=N-1	
1811	005366	177731	-N	

1812	005370	000047	N	;DEC TAPE WRITE BUFFER
1813		000046	N=N-1	
1814	005372	177732	-N	
1815	005374	000046	N	;DEC TAPE WRITE BUFFER
1816		000045	N=N-1	
1817	005376	177733	-N	
1818	005400	000045	N	;DEC TAPE WRITE BUFFER
1819		000044	N=N-1	
1820	005402	177734	-N	
1821	005404	000044	N	;DEC TAPE WRITE BUFFER
1822		000043	N=N-1	
1823	005406	177735	-N	
1824	005410	000043	N	;DEC TAPE WRITE BUFFER
1825		000042	N=N-1	
1826	005412	177736	-N	
1827	005414	000042	N	;DEC TAPE WRITE BUFFER
1828		000041	N=N-1	
1829	005416	177737	-N	
1830	005420	000041	N	;DEC TAPE WRITE BUFFER
1831		000040	N=N-1	
1832	005422	177740	-N	
1833	005424	000040	N	;DEC TAPE WRITE BUFFER
1834		000037	N=N-1	
1835	005426	177741	-N	
1836	005430	000037	N	;DEC TAPE WRITE BUFFER
1837		000036	N=N-1	
1838	005432	177742	-N	
1839	005434	000036	N	;DEC TAPE WRITE BUFFER
1840		000035	N=N-1	
1841	005436	177743	-N	
1842	005440	000035	N	;DEC TAPE WRITE BUFFER
1843		000034	N=N-1	
1844	005442	177744	-N	
1845	005444	000034	N	;DEC TAPE WRITE BUFFER
1846		000033	N=N-1	
1847	005446	177745	-N	
1848	005450	000033	N	;DEC TAPE WRITE BUFFER
1849		000032	N=N-1	
1850	005452	177746	-N	
1851	005454	000032	N	;DEC TAPE WRITE BUFFER
1852		000031	N=N-1	
1853	005456	177747	-N	
1854	005460	000031	N	;DEC TAPE WRITE BUFFER
1855		000030	N=N-1	
1856	005462	177750	-N	
1857	005464	000030	N	;DEC TAPE WRITE BUFFER
1858		000027	N=N-1	
1859	005466	177751	-N	
1860	005470	000027	N	;DEC TAPE WRITE BUFFER
1861		000026	N=N-1	
1862	005472	177752	-N	
1863	005474	000026	N	;DEC TAPE WRITE BUFFER
1864		000025	N=N-1	
1865	005476	177753	-N	
1866	005500	000025	N	;DEC TAPE WRITE BUFFER
1867		000024	N=N-1	

1868	005502	177754	-N	
1869	005504	000024	N	;DEC TAPE WRITE BUFFER
1870		000023	N=N-1	
1871	005506	177755	-N	
1872	005510	000023	N	;DEC TAPE WRITE BUFFER
1873		000022	N=N-1	
1874	005512	177756	-N	
1875	005514	000022	N	;DEC TAPE WRITE BUFFER
1876		000021	N=N-1	
1877	005516	177757	-N	
1878	005520	000021	N	;DEC TAPE WRITE BUFFER
1879		000020	N=N-1	
1880	005522	177760	-N	
1881	005524	000020	N	;DEC TAPE WRITE BUFFER
1882		000017	N=N-1	
1883	005526	177761	-N	
1884	005530	000017	N	;DEC TAPE WRITE BUFFER
1885		000016	N=N-1	
1886	005532	177762	-N	
1887	005534	000016	N	;DEC TAPE WRITE BUFFER
1888		000015	N=N-1	
1889	005536	177763	-N	
1890	005540	000015	N	;DEC TAPE WRITE BUFFER
1891		000014	N=N-1	
1892	005542	177764	-N	
1893	005544	000014	N	;DEC TAPE WRITE BUFFER
1894		000013	N=N-1	
1895	005546	177765	-N	
1896	005550	000013	N	;DEC TAPE WRITE BUFFER
1897		000012	N=N-1	
1898	005552	177766	-N	
1899	005554	000012	N	;DEC TAPE WRITE BUFFER
1900		000011	N=N-1	
1901	005556	177767	-N	
1902	005560	000011	N	;DEC TAPE WRITE BUFFER
1903		000010	N=N-1	
1904	005562	177770	-N	
1905	005564	000010	N	;DEC TAPE WRITE BUFFER
1906		000007	N=N-1	
1907	005566	177771	-N	
1908	005570	000007	N	;DEC TAPE WRITE BUFFER
1909		000006	N=N-1	
1910	005572	177772	-N	
1911	005574	000006	N	;DEC TAPE WRITE BUFFER
1912		000005	N=N-1	
1913	005576	177773	-N	
1914	005600	000005	N	;DEC TAPE WRITE BUFFER
1915		000004	N=N-1	
1916	005602	177774	-N	
1917	005604	000004	N	;DEC TAPE WRITE BUFFER
1918		000003	N=N-1	
1919	005606	177775	-N	
1920	005610	000003	N	;DEC TAPE WRITE BUFFER
1921		000002	N=N-1	
1922	005612	177776	-N	
1923	005614	000002	N	;DEC TAPE WRITE BUFFER

004

DBKTGD MAINDEC-11-DBKTG-D KT11-D EXERCISER MACY11 27(1006) 21-SEP-76 11:30 PAGE 42
DBKTGD.F11 21-SEP-76 11:27 TC11 DIAGNOSTIC ROUTINE

1924 000001
1925 005616 177777
1926 005620 000001
1927

N=N-1
-N
N

;DEC TAPE WRITE BUFFER

1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983

005622 010701
005624 042701 017777
005630 042737 160000 000034
005636 050137 000034
005642 000301
005644 006201
005646 006201
005650 010137 000572

005654
005654 005000
005656 066700 007210
005662 066700 007206
005666 001006
005670 012767 001233 007174
005676 012767 007622 007170
005704 005067 007104
005710 010767 007102
005714 062767 000040 007074
005722 016767 007064 007060
005730 005737 000042
005734 001407
005736 023737 000042 000046
005744 001403
005746 016767 007050 007034
005754

005754 012700 177770
005760 026027 015106 125252
005766 001401
005770 104006
005772 104400

005774 012700 000010
006000 022760 052525 015106
006006 001401
006010 104006
006012 104400

006014 012700 177770
006020 026060 015106 015106
006026 001401
006030 104006
006032 104400

006034 012700 000010
006040 026060 015106 015106
006046 001401
006050 104006

```
*****  
:SBTTL MAIN ROUTINE: CPU BACKGROUND TESTS  
*****  
BEGINX: MOV PC,R1 ;SET UP R1 TO SELECT CURBNK  
BIC #17777,R1  
BIC #160000,#34 ;SET SCOPE RET TO CURRENT BANK  
BIS R1,#34  
SWAB R1  
ASR R1  
ASR R1  
MOV R1,#CURBNK
```

:BINARY INSTRUCTIONS
:INDEX, AND INDIRECT TEST OF PDP-11
BEGIN:

```
CLR R0 ;CHECK RANDOM NUMBER GENERATOR SEEDS.  
ADD RP1,R0 ;AND RESTORE IF ZEROED.  
ADD RP2,R0  
BNE 1$ ;BR IF NOT ZEROED.  
MOV #1233,RP1 ;RESTORE RP1 SEED.  
MOV #7622,RP2 ;RESTORE RP2 SEED.  
1$: CLR SCOPEF  
MOV PC,RETURN ;FOR SCOPING - SETUP ADDRESS OF BEGIN1 IN  
ADD #40,RETURN ;THIS BANK THRU CURRENT ASR  
MOV $ICNT,ICOUNT ;ITERATION COUNT  
TST #42 ;AUTO MODE?  
BEQ 2$ ;BR IF NOT.  
CMP #42,#46 ;XXDP CHAIN MODE?  
BEQ 2$ ;BR IF NOT.  
MOV XDPCNT,ICOUNT ;USE XXDP CHAIN ITERATION COUNT.  
2$:
```

:SBTTL TEST COMPARE INSTRUCTION INDEXED

```
MOV #-10,%0 ;MINUS 10 TO REG 0  
CMP A(0),#125252 ;(A INDEX BY MINUS 10) TO #125252  
BEQ .+4  
HLT ;COMPARE WITH INDEX FAILED  
SCOPE  
  
MOV #10,%0  
CMP #052525,A(0)  
BEQ .+4  
HLT  
SCOPE  
  
MOV #-10,%0  
CMP A(0),A(0)  
BEQ .+4  
HLT  
SCOPE  
  
MOV #+10,%0  
CMP A(0),A(0)  
BEQ .+4  
HLT
```

1984	006052	104400			SCOPE
1985					
1986	006054	012700	177774		MOV #-4,%0
1987	006060	012701	000010		MOV #+10,%1
1988	006064	026061	015106	015106	CMP A(0),A(1)
1989	006072	001401			BEQ .+4
1990	006074	104006			HLT
1991	006076	104400			SCOPE
1992					
1993	006100	012700	177774		MOV #-4,%0
1994	006104	012701	000010		MOV #10,%1
1995	006110	026160	015106	015106	CMP A(1),A(0)
1996	006116	001401			BEQ .+4
1997	006120	104006			HLT
1998	006122	104400			SCOPE
1999					
2000					
2001					
2002					
2003					
2004	006124	012700	177770		MOV #-10,%0
2005	006130	016067	015106	006772	MOV A(0),TEMP
2006	006136	026727	006766	125252	CMP TEMP,#125252
2007	006144	001401			BEQ .+4
2008	006146	104006			HLT
2009	006150	104400			SCOPE
2010					
2011	006152	012700	177770		MOV #-10,%0
2012	006156	012760	125252	015130	MOV #125252,TEMP(0)
2013	006164	023727	015120	125252	CMP @#C,#125252
2014	006172	001401			BEQ .+4
2015	006174	104006			HLT
2016	006176	104400			SCOPE
2017					
2018					
2019					
2020					
2021	006200	012767	177777	006722	MOV #-1,TEMP
2022	006206	012700	177770		MOV #-10,%0
2023	006212	046067	015106	006710	BIC A(0),TEMP
2024	006220	026727	006704	052525	CMP TEMP,#052525
2025	006226	001401			BEQ .+4
2026	006230	104006			HLT
2027	006232	104400			SCOPE
2028					
2029	006234	012700	177770		MOV #-10,%0
2030	006240	012767	177777	006652	MOV #-1,TEMP-10
2031	006246	042767	052525	006644	BIC #052525,TEMP-10
2032	006254	026727	006640	125252	CMP TEMP-10,#125252
2033	006262	001401			BEQ .+4
2034	006264	104006			HLT
2035	006266	104400			SCOPE
2036					
2037	006270	012737	125252	015130	MOV #125252,@#TEMP
2038	006276	012700	177770		MOV #-10,%0
2039	006302	166760	006570	015140	SUB B,TEMP+10(0)

:SBTTL TEST MOVE INSTRUCTION FOR INDEX

:SBTTL TEST BIC INSTRUCTION FOR INDEXING

```

2040 006310 001401 BEQ .+4
2041 006312 104006 HLT
2042 006314 104400 SCOPE
2043
2044 006316 012737 052525 015130 MOV #052525,@#TEMP
2045 006324 012700 000010 MOV #10,%0
2046 006330 166760 006562 015120 SUB A+10,C(0)
2047 006336 001401 BEQ .+4
2048 006340 104006 HLT
2049 006342 104400 SCOPE

```

```

:*****
.SBTTL TEST UNARYS INDEXED
:*****

```

```

2055 006344 012737 177777 015130 MOV #-1,@#TEMP
2056 006352 012700 000010 MOV #+10,%0
2057 006356 005060 015120 CLR C(0)
2058 006362 005737 015130 TST @#TEMP
2059 006366 001401 BEQ .+4
2060 006370 104006 HLT
2061 006372 104400 SCOPE

```

```

2063 006374 012737 177777 015130 MOV #-1,@#TEMP
2064 006402 012700 000010 MOV #10,%0
2065 006406 005160 015120 COM C(0)
2066 006412 005737 015130 TST @#TEMP
2067 006416 001401 BEQ .+4
2068 006420 104006 HLT
2069 006422 104400 SCOPE

```

```

2071 006424 012737 177777 015130 MOV #-1,@#TEMP
2072 006432 012700 177770 MOV #-10,%0
2073 006436 005260 015140 INC D(0)
2074 006442 005737 015130 TST @#TEMP
2075 006446 001401 BEQ .+4
2076 006450 104006 HLT
2077 006452 104400 SCOPE

```

```

2079 006454 012737 000001 015130 MOV #1,@#TEMP
2080 006462 012700 177770 MOV #-10,%0
2081 006466 005360 015140 DEC D(0)
2082 006472 005737 015130 TST @#TEMP
2083 006476 001401 BEQ .+4
2084 006500 104006 HLT
2085 006502 104400 SCOPE

```

```

2087 006504 012737 000001 015130 MOV #1,@#TEMP
2088 006512 012700 000010 MOV #10,%0
2089 006516 005360 015120 DEC C(0)
2090 006522 005737 015130 TST @#TEMP
2091 006526 001401 BEQ .+4
2092 006530 104006 HLT
2093 006532 104400 SCOPE

```

```

2095 006534 012737 000001 015130 MOV #1,@#TEMP

```

2096	006542	012700	177770		MOV	#-10,%0
2097	006546	005460	015140		NEG	D(0)
2098	006552	022737	177777	015130	CMP	#-1,@#TEMP
2099	006560	001401			BEQ	.+4
2100	006562	104006			HLT	
2101	006564	104400			SCOPE	
2102						
2103	006566	012737	000001	015130	MOV	#1,@#TEMP
2104	006574	012700	000010		MOV	#+10,%0
2105	006600	005460	015120		NEG	C(0)
2106	006604	022737	177777	015130	CMP	#-1,@#TEMP
2107	006612	001401			BEQ	.+4
2108	006614	104006			HLT	
2109	006616	104400			SCOPE	
2110						
2111	006620	012737	177777	015130	MOV	#-1,@#TEMP
2112	006626	012700	177770		MOV	#-10,%0
2113	006632	000261			SEC	
2114	006634	005560	015140		ADC	D(0)
2115	006640	005737	015130		TST	@#TEMP
2116	006644	001401			BEQ	.+4
2117	006646	104006			HLT	
2118	006650	104400			SCOPE	
2119						
2120	006652	012737	177777	015130	MOV	#-1,@#TEMP
2121	006660	012700	000010		MOV	#+10,%0
2122	006664	000261			SEC	
2123	006666	005560	015120		ADC	C(0)
2124	006672	005737	015130		TST	@#TEMP
2125	006676	001401			BEQ	.+4
2126	006700	104006			HLT	
2127	006702	104400			SCOPE	
2128						
2129	006704	012737	000001	015130	MOV	#1,@#TEMP
2130	006712	012700	177770		MOV	#-10,%0
2131	006716	000261			SEC	
2132	006720	005660	015140		SBC	D(0)
2133	006724	005737	015130		TST	@#TEMP
2134	006730	001401			BEQ	.+4
2135	006732	104006			HLT	
2136	006734	104400			SCOPE	
2137						
2138	006736	012737	000001	015130	MOV	#1,@#TEMP
2139	006744	012700	000010		MOV	#+10,%0
2140	006750	000261			SEC	
2141	006752	005660	015120		SBC	C(0)
2142	006756	005737	015130		TST	@#TEMP
2143	006762	001401			BEQ	.+4
2144	006764	104006			HLT	
2145	006766	104400			SCOPE	
2146						
2147						
2148						
2149						
2150	006770	010700			MOV	%7,%0
2151	006772	062700	000010		ADD	#10,%0

```

;*****
;SBTTL TEST JMP INDIRECT
;*****

```

```

2152 006776 000110      JMP      @%0
2153 007000 104006      HLT
2154 007002 000240      NOP
2155 007004 104400      SCOPE
2156
2157 007006 010700      MOV      %7,%0
2158 007010 062700 000010      ADD      #10,%0
2159 007014 000110      JMP      @%0
2160 007016 104006      HLT
2161 007020 000240      NOP
2162 007022 104400      SCOPE
2163
2164 ;*****
2165 ;SBTTL TEST INDIRECT ADDRESSINGTEST COMPARE INSTRUCTION
2166 ;*****
2167 007024 023727 015076 125252      CMP      @#B,#125252
2168 007032 001401      BEQ      .+4
2169 007034 104006      HLT
2170 007036 104400      SCOPE
2171
2172 007040 022737 125252 015076      CMP      #125252,@#B
2173 007046 001401      BEQ      .+4
2174 007050 104006      HLT
2175 007052 104400      SCOPE
2176
2177 007054 023737 015076 015076      CMP      @#B,@#B
2178 007062 001401      BEQ      .+4
2179 007064 104006      HLT
2180 007066 104400      SCOPE
2181
2182 ;*****
2183 ;SBTTL TEST MOVE INSTRUCTIONS
2184 ;*****
2185 007070 013700 015076      MOV      @#B,%0
2186 007074 022700 125252      CMP      #125252,%0
2187 007100 001401      BEQ      .+4
2188 007102 104006      HLT
2189 007104 104400      SCOPE
2190
2191 007106 012737 125252 015130      MOV      #125252,@#TEMP
2192 007114 023737 015076 015130      CMP      @#B,@#TEMP
2193 007122 001401      BEQ      .+4
2194 007124 104006      HLT
2195 007126 104400      SCOPE
2196
2197 007130 013737 015076 015120      MOV      @#B,@#C
2198 007136 023737 015076 015120      CMP      @#B,@#C
2199 007144 001401      BEQ      .+4
2200 007146 104006      HLT
2201 007150 104400      SCOPE
2202
2203 ;*****
2204 ;SBTTL TEST BIC INSTRUCTION INDIRECT
2205 ;*****
2206 007152 012700 177777      MOV      #-1,%0
2207 007156 043700 015076      BIC      @#B,%0

```



```

2208 007162 020027 052525      CMP      %0, #052525
2209 007166 001401                BEQ      .+4
2210 007170 104006                HLT
2211 007172 104400                SCOPE
2212
2213 007174 012737 177777 015130      MOV      #-1, @#TEMP
2214 007202 042737 125252 015130      BIC      #125252, @#TEMP
2215 007210 022737 052525 015130      CMP      #052525, @#TEMP
2216 007216 001401                BEQ      .+4
2217 007220 104006                HLT
2218 007222 104400                SCOPE
2219
2220 007224 012737 177777 015120      MOV      #-1, @#C
2221 007232 043737 015076 015120      BIC      @#B, @#C
2222 007240 023727 015120 052525      CMP      @#C, #52525
2223 007246 001401                BEQ      .+4
2224 007250 104006                HLT
2225 007252 104400                SCOPE
2226

```

```

;*****
;SBTTL TEST SUBTRACT INSTRUCTION
;*****

```

```

2230 007254 012700 125252      MOV      #125252, %0
2231 007260 163700 015076      SUB      @#B, %0
2232 007264 020027 000000      CMP      %0, #0
2233 007270 001401                BEQ      .+4
2234 007272 104006                HLT
2235 007274 104400                SCOPE
2236
2237 007276 012737 125252 015130      MOV      #125252, @#TEMP
2238 007304 166737 005566 015130      SUB      B, @#TEMP
2239 007312 001401                BEQ      .+4
2240 007314 104006                HLT
2241 007316 104400                SCOPE
2242
2243 007320 012767 125252 005602      MOV      #125252, TEMP
2244 007326 163767 015076 005574      SUB      @#B, TEMP
2245 007334 005767 005570                TST      TEMP
2246 007340 001401                BEQ      .+4
2247 007342 104006                HLT
2248 007344 104400                SCOPE
2249

```

```

;*****
;SBTTL TEST ADD INDIRECT
;*****

```

```

2253 007346 005000                CLR      %0
2254 007350 063700 015076      ADD      @#B, %0
2255 007354 022700 125252      CMP      #125252, %0
2256 007360 001401                BEQ      .+4
2257 007362 104006                HLT
2258 007364 104400                SCOPE
2259
2260 007366 005037 015130      CLR      @#TEMP
2261 007372 062737 125252 015130      ADD      #125252, @#TEMP
2262 007400 022737 125252 015130      CMP      #125252, @#TEMP
2263 007406 001401                BEQ      .+4

```

```

2264 007410 104006
2265 007412 104400
2266
2267 007414 012737 125252 015130
2268 007422 067737 005466 015130
2269 007430 023727 015130 177777
2270 007436 001401
2271 007440 104006
2272 007442 104400
2273
2274
2275
2276
2277 007444 012737 177777 015130
2278 007452 005037 015130
2279 007456 005737 015130
2280 007462 001401
2281 007464 104006
2282 007466 104400
2283
2284 007470 012737 125252 015130
2285 007476 005137 015130
2286 007502 022737 052525 015130
2287 007510 001401
2288 007512 104006
2289 007514 104400
2290
2291 007516 005037 015130
2292 007522 005237 015130
2293 007526 022737 000001 015130
2294 007534 001401
2295 007536 104006
2296 007540 104400
2297
2298 007542 005037 015130
2299 007546 005377 005360
2300 007552 023727 015130 177777
2301 007560 001401
2302 007562 104006
2303 007564 104400
2304
2305 007566 012737 000001 015130
2306 007574 005437 015130
2307 007600 022737 177777 015130
2308 007606 001401
2309 007610 104006
2310 007612 104400
2311
2312
2313
2314
2315 007614 027727 005260 125252
2316 007622 001401
2317 007624 104006
2318 007626 104400
2319

```

```

HLT
SCOPE
MOV #125252,@TEMP
ADD @A+6,@TEMP
CMP @TEMP,#-1
BEQ .+4
HLT
SCOPE

```

```

;*****
.SBTTL TEST UNARYS INDIRECT
;*****

```

```

MOV #-1,@TEMP
CLR @TEMP
TST @TEMP
BEQ .+4
HLT
SCOPE
MOV #125252,@TEMP
COM @TEMP
CMP #052525,@TEMP
BEQ .+4
HLT
SCOPE

```

```

CLR @TEMP
INC @TEMP
CMP #1,@TEMP
BEQ .+4
HLT
SCOPE

```

```

CLR @TEMP
DEC @TEMP+2
CMP @TEMP,#-1
BEQ .+4
HLT
SCOPE

```

```

MOV #1,@TEMP
NEG @TEMP
CMP #-1,@TEMP
BEQ .+4
HLT
SCOPE

```

```

;*****
.SBTTL TEST INDIRECT ADDRESSING WITH INDEXING, TEST COMPARE INSTRUCTION
;*****

```

```

CMP @B+2,#125252
BEQ .+4
HLT
SCOPE

```

2320	007630	022777	125252	005242	CMP	#125252,@B+2
2321	007636	001401			BEQ	.+4
2322	007640	104006			HLT	
2323	007642	104400			SCOPE	
2324						
2325	007644	027777	005230	005226	CMP	@B+2,@B+2
2326	007652	001401			BEQ	.+4
2327	007654	104006			HLT	
2328	007656	104400			SCOPE	
2329						

:SBTTL TEST MOVE INSTRUCTIONS

2333	007660	017700	005214		MOV	@B+2,%0
2334	007664	022700	125252		CMP	#125252,%0
2335	007670	001401			BEQ	.+4
2336	007672	104006			HLT	
2337	007674	104400			SCOPE	
2338						
2339	007676	012777	125252	005226	MOV	#125252,@TEMP+2
2340	007704	023737	015076	015130	CMP	@#B,@#TEMP
2341	007712	001401			BEQ	.+4
2342	007714	104006			HLT	
2343	007716	104400			SCOPE	
2344						

2345	007720	017777	005154	005174	MOV	@B+2,@C+2
2346	007726	023737	015076	015120	CMP	@#B,@#C
2347	007734	001401			BEQ	.+4
2348	007736	104006			HLT	
2349	007740	104400			SCOPE	
2350						

:SBTTL TEST BIC INSTRUCTION INDIRECT WITH INDEXING

2354	007742	012700	177777		MOV	#-1,%0
2355	007746	047700	005126		BIC	@B+2,%0
2356	007752	020027	052525		CMP	%0,#52525
2357	007756	001401			BEQ	.+4
2358	007760	104006			HLT	
2359	007762	104400			SCOPE	
2360						
2361	007764	012737	177777	015130	MOV	#-1,@#TEMP
2362	007772	042777	125252	005132	BIC	#125252,@TEMP+2
2363	010000	022737	052525	015130	CMP	#52525,@#TEMP
2364	010006	001401			BEQ	.+4
2365	010010	104006			HLT	
2366	010012	104400			SCOPE	
2367						
2368	010014	012737	177777	015120	MOV	#-1,@#C
2369	010022	047777	005052	005072	BIC	@B+2,@C+2
2370	010030	026737	005062	015120	CMP	A+10,@#C
2371	010036	001401			BEQ	.+4
2372	010040	104006			HLT	
2373	010042	104400			SCOPE	
2374						
2375	010044	012700	125252		MOV	#125252,%0

M04

2376	010050	167700	005024		SUB	@B+2,%0
2377	010054	020027	000000		CMP	%0,%0
2378	010060	001401			BEQ	+.4
2379	010062	104006			HLT	
2380	010064	104400			SCOPE	
2381						
2382	010066	012737	125252	015130	MOV	#125252,@TEMP
2383	010074	166777	004776	005030	SUB	B,@TEMP+2
2384	010102	001401			BEQ	+.4
2385	010104	104006			HLT	
2386	010106	104400			SCOPE	
2387						
2388	010110	012737	125252	015130	MOV	#125252,@TEMP
2389	010116	167777	004756	005006	SUB	@B+2,@TEMP+2
2390	010124	005737	015130		TST	@TEMP
2391	010130	001401			BEQ	+.4
2392	010132	104006			HLT	
2393	010134	104400			SCOPE	

```

;*****
;SBTTL TEST ADD INDIRECT WITH INDEXING
;*****

```

2394						
2395						
2396						
2397						
2398	010136	005000			CLR	%0
2399	010140	067700	004734		ADD	@B+2,%0
2400	010144	022700	125252		CMP	#125252,%0
2401	010150	001401			BEQ	+.4
2402	010152	104006			HLT	
2403	010154	104400			SCOPE	
2404						
2405	010156	005037	015130		CLR	@TEMP
2406	010162	062777	125252	004742	ADD	#125252,@TEMP+2
2407	010170	022737	125252	015130	CMP	#125252,@TEMP
2408	010176	001401			BEQ	+.4
2409	010200	104006			HLT	
2410	010202	104400			SCOPE	
2411						
2412	010204	012737	125252	015130	MOV	#125252,@TEMP
2413	010212	067777	004676	004712	ADD	@A+6,@TEMP+2
2414	010220	023727	015130	177777	CMP	@TEMP,#-1
2415	010226	001401			BEQ	+.4
2416	010230	104006			HLT	
2417	010232	104400			SCOPE	

```

;*****
;SBTTL TEST UNARYS INDIRECT WITH INDEXING
;*****

```

2418						
2419						
2420						
2421						
2422	010234	012737	177777	015130	MOV	#-1,@TEMP
2423	010242	005077	004664		CLR	@TEMP+2
2424	010246	005737	015130		TST	@TEMP
2425	010252	001401			BEQ	+.4
2426	010254	104006			HLT	
2427	010256	104400			SCOPE	
2428						
2429	010260	012737	125252	015130	MOV	#125252,@TEMP
2430	010266	005177	004640		COM	@TEMP+2
2431	010272	022737	052525	015130	CMP	#052525,@TEMP

2432	010300	001401			BEQ	.+4
2433	010302	104006			HLT	
2434	010304	104400			SCOPE	
2435						
2436	010306	005037	015130		CLR	@TEMP
2437	010312	005277	004614		INC	@TEMP+2
2438	010316	022737	000001	015130	CMP	#1,@TEMP
2439	010324	001401			BEQ	.+4
2440	010326	104006			HLT	
2441	010330	104400			SCOPE	
2442						
2443	010332	005037	015130		CLR	@TEMP
2444	010336	005377	004570		DEC	@TEMP+2
2445	010342	023727	015130	177777	CMP	@TEMP,#-1
2446	010350	001401			BEQ	.+4
2447	010352	104006			HLT	
2448	010354	104400			SCOPE	
2449						
2450	010356	012737	000001	015130	MOV	#1,@TEMP
2451	010364	005477	004542		NEG	@TEMP+2
2452	010370	022737	177777	015130	CMP	#-1,@TEMP
2453	010376	001401			BEQ	.+4
2454	010400	104006			HLT	
2455	010402	104400			SCOPE	
2456						
2457	010404	012737	177777	015130	MOV	#-1,@TEMP
2458	010412	000261			SEC	
2459	010414	005577	004512		ADC	@TEMP+2
2460	010420	005737	015130		TST	@TEMP
2461	010424	001401			BEQ	.+4
2462	010426	104006			HLT	
2463	010430	104400			SCOPE	
2464						
2465	010432	012737	000001	015130	MOV	#1,@TEMP
2466	010440	000261			SEC	
2467	010442	005677	004464		SBC	@TEMP+2
2468	010446	005737	015130		TST	@TEMP
2469	010452	001401			BEQ	.+4
2470	010454	104006			HLT	
2471	010456	104400			SCOPE	
2472						
2473						
2474						
2475						
2476	010460	012700	177772		MOV	#-6,%0
2477	010464	027027	015106	125252	CMP	@A(0),#125252
2478	010472	001401			BEQ	.+4
2479	010474	104006			HLT	
2480	010476	104400			SCOPE	
2481						
2482	010500	012700	177772		MOV	#-6,%0
2483	010504	022770	125252	015106	CMP	#125252,@A(0)
2484	010512	001401			BEQ	.+4
2485	010514	104006			HLT	
2486	010516	104400			SCOPE	
2487						

```

;*****
;SBTTL TEST OF COMBINED INDEXING AND INDIRECT
;*****

```

010520 012700 177772
010524 012701 000002
010530 027071 015106 015106
010536 001401
010540 104006
010542 104400

MOV #-6,%0
MOV #+2,%1
CMP @A(0),@A(1)
BEQ .+4
HLT
SCOPE

:SBTTL TEST BIC INSTRUCTION

010544 012700 000006
010550 012767 177777 004352
010556 047067 015106 004344
010564 022767 125252 004336
010572 001401
010574 104006
010576 104400

MOV #+6,%0
MOV #-1,TEMP
BIC @A(0),TEMP
CMP #125252,TEMP
BEQ .+4
HLT
SCOPE

010600 012700 177772
010604 012737 177777 015120
010612 042770 125252 015130
010620 023727 015120 052525
010626 001401
010630 104006
010632 104400

MOV #-6,%0
MOV #-1,@#C
BIC #125252,@TEMP(0)
CMP @#C,#052525
BEQ .+4
HLT
SCOPE

010634 012737 177777 015120
010642 012700 177772
010646 012701 177772
010652 047071 015106 015130
010660 022737 052525 015120
010666 001401
010670 104006
010672 104400

MOV #-1,@#C
MOV #-6,%0
MOV #-6,%1
BIC @A(0),@TEMP(1)
CMP #052525,@#C
BEQ .+4
HLT
SCOPE

:BINARY INSTRUCTIONS
:INDEX, AND INDIRECT TEST OF PDP-11

:SBTTL TEST COMPARE INSTRUCTION INDEXED

010674 012700 177770
010700 126027 015106 000252
010706 001401
010710 104006
010712 104400

MOV #-10,%0 :MINUS 10 TO REG 0
CMPB A(0),#000252 : (A INDEX BY MINUS 10) TO #125252
BEQ .+4
HLT :COMPARE WITH INDEX FAILED
SCOPE

010714 012700 177770
010720 122760 000252 015106
010726 001401
010730 104006
010732 104400

MOV #-10,%0 :FOR INDEX
CMPB #000252,A(0) :A INDEXED
BEQ .+4
HLT
SCOPE

010734 012700 000010
010740 126027 015106 000125
010746 001401
010750 104006

MOV #10,%0 :INDEX
CMPB A(0),#000125
BEQ .+4
HLT

010752 104400
010754 012700 000010
010760 122760 000125 015106
010766 001401
010770 104006
010772 104400
010774 012700 177770
011000 126060 015106 015106
011006 001401
011010 104006
011012 104400
011014 012700 000010
011020 126060 015106 015106
011026 001401
011030 104006
011032 104400
011034 012700 177770
011040 012701 000004
011044 126061 015106 015106
011052 001401
011054 104006
011056 104400
011060 126160 015106 015106
011066 001401
011070 104006
011072 104400
011074 012700 177774
011100 012701 000010
011104 126061 015106 015106
011112 001401
011114 104006
011116 104400
011120 012700 177774
011124 012701 000010
011130 126160 015106 015106
011136 001401
011140 104006
011142 104400
011144 012700 177770
011150 116067 015106 003752
011156 126727 003746 000252
011164 001401
011166 104006
011170 104400

SCOPE
MOV #10,%0
CMPB #000125,A(0)
BEQ .+4
HLT
SCOPE
MOV #-10,%0
CMPB A(0),A(0)
BEQ .+4
HLT
SCOPE
MOV #+10,%0
CMPB A(0),A(0)
BEQ .+4
HLT
SCOPE
MOV #-10,%0
MOV #+4,%1
CMPB A(0),A(1)
BEQ .+4
HLT
SCOPE
CMPB A(1),A(0)
BEQ .+4
HLT
SCOPE
MOV #-4,%0
MOV #+10,%1
CMPB A(0),A(1)
BEQ .+4
HLT
SCOPE
MOV #-4,%0
MOV #10,%1
CMPB A(1),A(0)
BEQ .+4
HLT
SCOPE
:*****
:SBTTL TEST MOVE INSTRUCTION FOR INDEX
:*****
MOV #-10,%0
MOVB A(0),TEMP
CMPB TEMP,#000252
BEQ .+4
HLT
SCOPE

2600	011172	012700	000010		MOV	#+10,%0
2601	011176	116067	015106	003724	MOVB	A(0),TEMP
2602	011204	126727	003720	000125	CMPB	TEMP,#000125
2603	011212	001401			BEQ	.+4
2604	011214	104006			HLT	
2605	011216	104400			SCOPE	
2606						
2607	011220	012700	177770		MOV	#-10,%0
2608	011224	112760	125252	015130	MOVB	#125252,TEMP(0)
2609	011232	123727	015120	125252	CMPB	2#C,#125252
2610	011240	001401			BEQ	.+4
2611	011242	104006			HLT	
2612	011244	104400			SCOPE	
2613						
2614	011246	012700	000010		MOV	#+10,%0
2615	011252	112760	052525	015130	MOVB	#052525,TEMP(0)
2616	011260	123727	015140	052525	CMPB	2#TEMP+10,#052525
2617	011266	001401			BEQ	.+4
2618	011270	104006			HLT	
2619	011272	104400			SCOPE	
2620						
2621						
2622						
2623						
2624	011274	012767	177777	003626	MOV	#-1,TEMP
2625	011302	012700	177770		MOV	#-10,%0
2626	011306	146067	015106	003614	BICB	A(0),TEMP
2627	011314	126727	003610	177525	CMPB	TEMP,#177525
2628	011322	001401			BEQ	.+4
2629	011324	104006			HLT	
2630	011326	104400			SCOPE	
2631						
2632	011330	012767	177777	003572	MOV	#-1,TEMP
2633	011336	012700	000010		MOV	#10,%0
2634	011342	146067	015106	003560	BICB	A(0),TEMP
2635	011350	126727	003554	007652	CMPB	TEMP,#007652
2636	011356	001401			BEQ	.+4
2637	011360	104006			HLT	
2638	011362	104400			SCOPE	
2639						
2640	011364	012737	177777	015140	MOV	#-1,2#TEMP+10
2641	011372	012700	000010		MOV	#10,%0
2642	011376	142760	125252	015130	BICB	#125252,TEMP(0)
2643	011404	123727	015140	002525	CMPB	2#TEMP+10,#2525
2644	011412	001401			BEQ	.+4
2645	011414	104006			HLT	
2646	011416	104400			SCOPE	
2647						
2648	011420	012700	177770		MOV	#-10,%0
2649	011424	012767	177777	003466	MOV	#-1,TEMP-10
2650	011432	142767	052525	003460	BICB	#052525,TEMP-10
2651	011440	126727	003454	125252	CMPB	TEMP-10,#125252
2652	011446	001401			BEQ	.+4
2653	011450	104006			HLT	
2654	011452	104400			SCOPE	
2655						

```

:*****
:SBTTL TEST BIC INSTRUCTION FOR INDEXING
:*****

```

2624	011274	012767	177777	003626	MOV	#-1,TEMP
2625	011302	012700	177770		MOV	#-10,%0
2626	011306	146067	015106	003614	BICB	A(0),TEMP
2627	011314	126727	003610	177525	CMPB	TEMP,#177525
2628	011322	001401			BEQ	.+4
2629	011324	104006			HLT	
2630	011326	104400			SCOPE	
2631						
2632	011330	012767	177777	003572	MOV	#-1,TEMP
2633	011336	012700	000010		MOV	#10,%0
2634	011342	146067	015106	003560	BICB	A(0),TEMP
2635	011350	126727	003554	007652	CMPB	TEMP,#007652
2636	011356	001401			BEQ	.+4
2637	011360	104006			HLT	
2638	011362	104400			SCOPE	
2639						
2640	011364	012737	177777	015140	MOV	#-1,2#TEMP+10
2641	011372	012700	000010		MOV	#10,%0
2642	011376	142760	125252	015130	BICB	#125252,TEMP(0)
2643	011404	123727	015140	002525	CMPB	2#TEMP+10,#2525
2644	011412	001401			BEQ	.+4
2645	011414	104006			HLT	
2646	011416	104400			SCOPE	
2647						
2648	011420	012700	177770		MOV	#-10,%0
2649	011424	012767	177777	003466	MOV	#-1,TEMP-10
2650	011432	142767	052525	003460	BICB	#052525,TEMP-10
2651	011440	126727	003454	125252	CMPB	TEMP-10,#125252
2652	011446	001401			BEQ	.+4
2653	011450	104006			HLT	
2654	011452	104400			SCOPE	
2655						


```

*****
:SBTTL TEST UNARYS INDEXED
*****
2656 011454 012737 177777 015130 MOV      #-1,@TEMP
2657 011462 012700 177770 MOV      #-10,%0
2658 011466 105060 015140 CLRB     D(0)
2659 011472 105737 015130 TSTB    @TEMP
2660 011476 001401 BEQ      .+4
2661 011500 104006 HLT
2662 011502 104400 SCOPE
2663
2664 011504 012737 177777 015130 MOV      #-1,@TEMP
2665 011512 012700 177770 MOV      #-10,%0
2666 011516 105060 015140 CLRB     D(0)
2667 011522 023727 015130 177400 CMP      @TEMP,@177400
2668 011530 001401 BEQ      .+4
2669 011532 104006 HLT
2670 011534 104400 SCOPE
2671
2672 011536 012737 177777 015130 MOV      #-1,@TEMP
2673 011544 012700 177771 MOV      #-7,%0
2674 011550 105060 015140 CLRB     D(0)
2675 011554 023727 015130 000377 CMP      @TEMP,@000377
2676 011562 001401 BEQ      .+4
2677 011564 104006 HLT
2678 011566 104400 SCOPE
2679
2680 011570 012737 177777 015130 MOV      #-1,@TEMP
2681 011576 012700 000010 MOV      #+10,%0
2682 011602 105060 015120 CLRB     C(0)
2683 011606 105737 015130 TSTB    @TEMP
2684 011612 001401 BEQ      .+4
2685 011614 104006 HLT
2686 011616 104400 SCOPE
2687
2688 011620 012737 177777 015130 MOV      #-1,@TEMP
2689 011626 012700 177770 MOV      #-10,%0
2690 011632 105160 015140 COMB    D(0)
2691 011636 105737 015130 TSTB    @TEMP
2692 011642 001401 BEQ      .+4
2693 011644 104006 HLT
2694 011646 104400 SCOPE
2695
2696 011650 012737 177777 015130 MOV      #-1,@TEMP
2697 011656 012700 000010 MOV      #+10,%0
2698 011662 105260 015120 INCB    C(0)
2699 011666 105737 015130 TSTB    @TEMP
2700 011672 001401 BEQ      .+4
2701 011674 104006 HLT
2702 011676 104400 SCOPE
2703
2704 011700 012737 000001 015130 MOV      #1,@TEMP
2705 011706 012700 177770 MOV      #-10,%0
2706 011712 105360 015140 DECB    D(0)
2707 011716 105737 015130 TSTB    @TEMP
2708 011722 001401 BEQ      .+4

```

```

712 011724 104006 HLT
713 011726 104400 SCOPE
714
715 011730 012737 000001 015130 MOV @1,@TEMP
716 011736 012700 000010 MOV @+10,%0
717 011742 105460 015120 NEGB C(0)
718 011746 023727 015130 000377 CMP @TEMP,#377
719 011754 001401 BEQ .+4
720 011756 104006 HLT
721 011760 104400 SCOPE
722
723 011762 012737 177777 015130 MOV #-1,@TEMP
724 011770 012700 177770 MOV #-10,%0
725 011774 000261 SEC
726 011776 105560 015140 ADCB D(0)
727 012002 023727 015130 177400 CMP @TEMP,#177400
728 012010 001401 BEQ .+4
729 012012 104006 HLT
730 012014 104400 SCOPE
731
732 012016 012737 000001 015130 MOV @1,@TEMP
733 012024 012700 000010 MOV @+10,%0
734 012030 000261 SEC
735 012032 105660 015120 SBCB C(0)
736 012036 005737 015130 TST @TEMP
737 012042 001401 BEQ .+4
738 012044 104006 HLT
739 012046 104400 SCOPE
740
741 *****
742 :SBTTL TEST INDIRECT ADDRESSING, TEST COMPARE INSTRUCTION
743 *****
744 012050 123727 015076 000252 CMPB @B,#000252
745 012056 001401 BEQ .+4
746 012060 104006 HLT
747 012062 104400 SCOPE
748
749 012064 122737 125252 015076 CMPB #125252,@B
750 012072 001401 BEQ .+4
751 012074 104006 HLT
752 012076 104400 SCOPE
753
754 *****
755 :SBTTL TEST MOVE INSTRUCTIONS
756 *****
757 012100 113700 015076 MOVB @B,%0
758 012104 122700 000252 CMPB #000252,%0
759 012110 001401 BEQ .+4
760 012112 104006 HLT
761 012114 104400 SCOPE
762
763 012116 112737 125252 015130 MOVB #125252,@TEMP
764 012124 126737 002746 015130 CMPB B,@TEMP
765 012132 001401 BEQ .+4
766 012134 104006 HLT
767 012136 104400 SCOPE

```

```

2768
2769
2770
2771
2772 012140 012737 177777 015130
2773 012146 105037 015130
2774 012152 023727 015130 177400
2775 012160 001401
2776 012162 104006
2777 012164 104400
2778
2779 012166 012737 125252 015130
2780 012174 105137 015131
2781 012200 022737 052652 015130
2782 012206 001401
2783 012210 104006
2784 012212 104400
2785
2786 012214 005037 015130
2787 012220 105237 015131
2788 012224 022737 000400 015130
2789 012232 001401
2790 012234 104006
2791 012236 104400
2792
2793 012240 005037 015130
2794 012244 105377 002662
2795 012250 023727 015130 000377
2796 012256 001401
2797 012260 104006
2798 012262 104400
2799
2800 012264 005037 015130
2801 012270 112737 000001 015131
2802 012276 105437 015131
2803 012302 022737 177400 015130
2804 012310 001401
2805 012312 104006
2806 012314 104400
2807
2808
2809
2810
2811
2812 012316 122777 125252 002554
2813 012324 001401
2814 012326 104006
2815 012330 104400
2816
2817 012332 127777 002542 002540
2818 012340 001401
2819 012342 104006
2820 012344 104400
2821
2822
2823

```

```

:*****
:SBTTL TEST UNARYS INDIRECT
:*****
MOV      #-1,@TEMP
CLRB    @TEMP
CMP     @TEMP,#177400
BEQ     .+4
HLT
SCOPE

MOV     #125252,@TEMP
COMB   @TEMP+1
CMP    #052652,@TEMP
BEQ    .+4
HLT
SCOPE

CLR     @TEMP
INCB   @TEMP+1
CMP    #400,@TEMP
BEQ    .+4
HLT
SCOPE

CLR     @TEMP
DECB   @TEMP+2
CMP    @TEMP,#377
BEQ    .+4
HLT
SCOPE

CLR     @TEMP
MOVB   #1,@TEMP+1
NEGB   @TEMP+1
CMP    #177400,@TEMP
BEQ    .+4
HLT
SCOPE

:*****
:SBTTL TEST INDIRECT ADDRESSING WITH INDEXING, TEST COMPARE INSTRUCTION
:*****
CMPB   #125252,@B+2
BEQ    .+4
HLT
SCOPE

CMPB   @B+2,@B+2
BEQ    .+4
HLT
SCOPE

:*****
:SBTTL TEST MOVE INSTRUCTIONS

```

```

2824 ;*****
2825 012346 117700 002526          MOVB   @B+2,%0
2826 012352 122700 125252          CMPB   #125252,%0
2827 012356 001401          BEQ    .+4
2828 012360 104006          HLT
2829 012362 104400          SCOPE
2830
2831 012364 112777 125252 002540          MOVB   #125252,@TEMP+2
2832 012372 126737 002500 015130          CMPB   B,@#TEMP
2833 012400 001401          BEQ    .+4
2834 012402 104006          HLT
2835 012404 104400          SCOPE
2836
2837 012406 117777 002466 002506          MOVB   @B+2,@C+2
2838 012414 126737 002456 015120          CMPB   B,@#C
2839 012422 001401          BEQ    .+4
2840 012424 104006          HLT
2841 012426 104400          SCOPE
2842
2843 ;*****
2844 .SBTTL TEST BIC INSTRUCTION INDIRECT WITH INDEXING
2845 ;*****
2846 012430 012700 177777          MOV    #-1,%0
2847 012434 147700 002440          BICB   @B+2,%0
2848 012440 120027 052525          CMPB   %0,#52525
2849 012444 001401          BEQ    .+4
2850 012446 104006          HLT
2851 012450 104400          SCOPE
2852
2853 012452 012737 177777 015130          MOV    #-1,@#TEMP
2854 012460 142777 125252 002444          BICB   #125252,@TEMP+2
2855 012466 122737 052525 015130          CMPB   #52525,@#TEMP
2856 012474 001401          BEQ    .+4
2857 012476 104006          HLT
2858 012500 104400          SCOPE
2859
2860 012502 012737 177777 015120          MOV    #-1,@#C
2861 012510 147777 002364 002404          BICB   @B+2,@C+2
2862 012516 126737 002374 015120          CMPB   A+10,@#C
2863 012524 001401          BEQ    .+4
2864 012526 104006          HLT
2865 012530 104400          SCOPE
2866
2867 ;*****
2868 .SBTTL TEST UNARYS INDIRECT WITH INDEXING
2869 ;*****
2870 012532 012737 177777 015130          MOV    #-1,@#TEMP
2871 012540 105077 002366          CLRB  @TEMP+2
2872 012544 105737 015130          TSTB  @#TEMP
2873 012550 001401          BEQ    .+4
2874 012552 104006          HLT
2875 012554 104400          SCOPE
2876
2877 012556 005037 015130          CLR   @#TEMP
2878 012562 105277 002344          INCB  @TEMP+2
2879 012566 122737 000001 015130          CMPB  #1,@#TEMP

```

```

2880 012574 001401 BEQ .+4
2881 012576 104006 HLT
2882 012600 104400 SCOPE
2883
2884 012602 005037 015130 CLR @TEMP
2885 012606 105377 002320 DECB @TEMP+2
2886 012612 123727 015130 177777 CMPB @TEMP,#-1
2887 012620 001401 BEQ .+4
2888 012622 104006 HLT
2889 012624 104400 SCOPE
2890
2891 012626 012737 000001 015130 MOV #1,@TEMP
2892 012634 105477 002272 NEGB @TEMP+2
2893 012640 122737 177777 015130 CMPB #-1,@TEMP
2894 012646 001401 BEQ .+4
2895 012650 104006 HLT
2896 012652 104400 SCOPE
2897
2898 012654 012737 177777 015130 MOV #-1,@TEMP
2899 012662 000261 SEC
2900 012664 105577 002242 ADCB @TEMP+2
2901 012670 022737 177400 015130 CMP #177400,@TEMP
2902 012676 001401 BEQ .+4
2903 012700 104006 HLT
2904 012702 105737 015130 TSTB @TEMP
2905 012706 001401 BEQ .+4
2906 012710 104006 HLT
2907 012712 104400 SCOPE
2908
2909 012714 012737 000001 015130 MOV #1,@TEMP
2910 012722 000261 SEC
2911 012724 105377 002202 DECB @TEMP+2
2912 012730 005737 015130 TST @TEMP
2913 012734 001401 BEQ .+4
2914 012736 104006 HLT
2915 012740 104400 SCOPE
2916
2917
2918
2919
2920 012742 012700 177772 MOV #-6,%0
2921 012746 127027 015106 125252 CMPB @A(0),#125252
2922 012754 001401 BEQ .+4
2923 012756 104006 HLT
2924 012760 104400 SCOPE
2925
2926 012762 012700 177772 MOV #-6,%0
2927 012766 012701 000002 MOV #+2,%1
2928 012772 127071 015106 015106 CMPB @A(0),@A(1)
2929 013000 001401 BEQ .+4
2930 013002 104006 HLT
2931 013004 104400 SCOPE

```

```

;*****
:SBTTL TEST OF COMBINED INDEXING AND INDIRECT
;*****

```

```

;*****
:SBTTL TEST BIC INSTRUCTION
;*****

```

2936	013006	012700	000006		MOV	#+6,%0	
2937	013012	012767	177777	002110	MOV	#-1,TEMP	
2938	013020	147067	015106	002102	BICB	@A(0),TEMP	
2939	013026	122767	125252	002074	CMPB	#125252,TEMP	
2940	013034	001401			BEQ	.+4	
2941	013036	104006			HLT		
2942	013040	104400			SCOPE		
2943							
2944	013042	012700	177772		MOV	#-6,%0	
2945	013046	012737	177777	015120	MOV	#-1,@#C	
2946	013054	142770	125252	015130	BICB	#125252,@TEMP(0)	
2947	013062	123727	015120	000125	CMPB	@#C,#000125	
2948	013070	001401			BEQ	.+4	
2949	013072	104006			HLT		
2950	013074	104400			SCOPE		
2951							
2952	013076	012700	015100		MOV	#B+2,%0	;ADDRESS OF ADDRESS OF B
2953	013102	023067	001770		CMP	@(0)+,B	
2954	013106	001401			BEQ	.+4	
2955	013110	104006			HLT		
2956	013112	104400			SCOPE		
2957							
2958	013114	012700	015102		MOV	#B+4,%0	
2959	013120	025067	001752		CMP	@-(0),B	
2960	013124	001401			BEQ	.+4	
2961	013126	104006			HLT		
2962	013130	104400			SCOPE		
2963							
2964	013132	012700	015102		MOV	#B+4,%0	
2965	013136	125067	001734		CMPB	@-(0),B	
2966	013142	001401			BEQ	.+4	
2967	013144	104006			HLT		
2968	013146	104400			SCOPE		
2969							
2970	013150	012700	015124		MOV	#C+4,%0	
2971	013154	012737	177777	015120	MOV	#-1,@#C	
2972	013162	105050			CLRB	@-(0)	
2973	013164	023727	015120	177400	CMP	@#C,#177400	
2974	013172	001401			BEQ	.+4	
2975	013174	104006			HLT		
2976	013176	104400			SCOPE		
2977							
2978	013200	012737	177777	015120	MOV	#-1,@#C	
2979	013206	012700	177772		MOV	#-6,%0	
2980	013212	012701	177772		MOV	#-6,%1	
2981	013216	147071	015106	015130	BICB	@A(0),@TEMP(1)	
2982	013224	022737	177525	015120	CMP	#177525,@#C	
2983	013232	001401			BEQ	.+4	
2984	013234	104006			HLT		
2985	013236	104400			SCOPE		
2986							
2987							
2988							
2989							
2990	013240	012700	052525		MOV	#52525,%0	;THIS IS CHECKED LATER IN PROGRAM
2991							

```

;*****
;SBTTL SET UP TO TEST THAT R0 IS NOT DESTROYED BY FALSE SELECTION
;*****

```

K05

DBKTGD MAINDEC-11-DBKTG-D KT11-D EXERCISER
 DBKTGD.P11 21-SEP-76 11:27

MACY11 27(1006) 21-SEP-76 11:30 PAGE 62
 SET UP TO TEST THAT RO IS NOT DESTROYED BY FALSE SELECTION

```

2992 ;*****
2993 .SBTTL TEST JSR INSTRUCTION
2994 ;*****
2995 013244 004767 000002          JSR      %7, TJSR2          ;PLACE PC ON STACK
2996 013250 000405          TJSR1: BR      TJSR3          ;RETURN HERE ON RTS %19
2997 013252 121627 013250      TJSR2: CMPB   @%6, #TJSR1    ;CHECK FOR CORRECT PC ON STACK
2998 013256 001401          BEQ      .+4
2999 013260 104006          HLT
3000 013262 000207          RTS      %7          ;INCORRECT PC ON STACK
3001 013264 104400          TJSR3: SCOPE          ;RETURN TO IMST AFTER JSR
3002
3003 013266 000257          CCC
3004 013270 004717          JSR      %7, @%7          ;INSTRUCTION UNDER TEST
3005 013272 121627 013272      CMPB   @%6, #TJSR3+6      ;TEST THE STACK
3006 013276 001401          BEQ      .+4
3007 013300 104006          HLT          ;PC OF JSR DID NOT GO TO STACK
3008 013302 005726          TST     (6)+          ;REPOSITION THE STACK
3009 013304 104400          SCOPE
3010
3011 ;*****
3012 .SBTTL TEST NESTED SUBROUTINES
3013 ;*****
3014 013306 000257          CCC          ;CLEAR CONDITION CODES
3015 013310 004767 001360      JSR      %7, SUBR6
3016 013314 100401          BMI     .+4
3017 013316 104006          HLT          ;JSR OR RTS FAILED
3018 013320 001401          BEQ     .+4
3019 013322 104006          HLT          ;JSR OR RTS FAILED
3020 013324 102401          BVS    .+4
3021 013326 104006          HLT          ;JSR OR RTS FAILED
3022 013330 103401          BCS    .+4
3023 013332 104006          HLT          ;JSR OR RTS FAILED
3024 013334 104400          SCOPE
3025
3026 ;*****
3027 .SBTTL TEST ROTATE ODD BYTE
3028 ;*****
3029 013336 104400          SCOPE
3030 013340 000257          CCC          ;CLEAR "C"
3031 013342 012767 123456 001560  MOV     #123456, TEMP
3032 013350 106067 001555          RORB    TEMP+1          ;ROTATE ODD BYTE
3033 013354 103401          BCS    .+4
3034 013356 104006          HLT          ;C NOT SET
3035 013360 102401          BVS    .+4
3036 013362 104006          HLT          ;V NOT SET
3037 013364 022767 051456 001536  CMP     #051456, TEMP
3038 013372 001401          BEQ     .+4
3039 013374 104006          HLT          ;ROTATE FAILED
3040 013376 104400          SCOPE
3041
3042 013400 000277          SCC          ;SET C
3043 013402 012767 123456 001520  MOV     #123456, TEMP
3044 013410 106067 001515          RORB    TEMP+1
3045 013414 103401          BCS    .+4
3046 013416 104006          HLT          ;C NOT SET
3047 013420 102001          BVC    .+4
  
```

3048	013422	104006			HLT					;V NOT CLEARED
3049	013424	022767	151456	001476	CMP	#151456,TEMP				
3050	013432	001401			BEQ	.+4				
3051	013434	104006			HLT					;ROTATE FAILED
3052	013436	104400			SCOPE					
3053										
3054	013440	000257			CCC					
3055	013442	012767	123456	001460	MOV	#123456,TEMP				
3056	013450	106167	001455		ROLB	TEMP+1				
3057	013454	103401			BCS	.+4				
3058	013456	104006			HLT					;C NOT SET
3059	013460	102401			BVS	.+4				
3060	013462	104006			HLT					;V NOT SET
3061	013464	022767	047056	001436	CMP	#047056,TEMP				
3062	013472	001401			BEQ	.+4				
3063	013474	104006			HLT					;ROTATE BYTE FAILED
3064	013476	104400			SCOPE					
3065										
3066	013500	000277			SCC					;SET C
3067	013502	012767	123456	001420	MOV	#123456,TEMP				
3068	013510	106167	001415		ROLB	TEMP+1				
3069	013514	103401			BCS	.+4				
3070	013516	104006			HLT					;C NOT SET
3071	013520	102401			BVS	.+4				
3072	013522	104006			HLT					;V NOT SET
3073	013524	022767	047456	001376	CMP	#047456,TEMP				
3074	013532	001401			BEQ	.+4				
3075	013534	104006			HLT					;ROTATE ODD BYTE FAILED
3076	013536	104400			SCOPE					
3077										
3078	013540	000257			CCC					;CLEAR C
3079	013542	012767	177777	001360	MOV	#-1,TEMP				
3080	013550	106267	001355		ASRB	TEMP+1				
3081	013554	103401			BCS	.+4				
3082	013556	104006			HLT					;C NOT SET
3083	013560	102001			BVC	.+4				
3084	013562	104006			HLT					;V NOT CLEARED
3085	013564	026727	001340	177777	CMP	TEMP,#-1				
3086	013572	001401			BEQ	.+4				
3087	013574	104006			HLT					;SHIFT FAILED
3088	013576	104400			SCOPE					
3089										
3090	013600	000277			SCC					
3091	013602	012767	177777	001320	MOV	#-1,TEMP				
3092	013610	106367	001315		ASLB	TEMP+1				
3093	013614	103401			BCS	.+4				
3094	013616	104006			HLT					;C NOT SET
3095	013620	102001			BVC	.+4				
3096	013622	104006			HLT					;V NOT CLEARED
3097	013624	026727	001300	177377	CMP	TEMP,#177377				
3098	013632	001401			BEQ	.+4				
3099	013634	104006			HLT					;SHIFT BYTE FAILED
3100	013636	104400			SCOPE					
3101										
3102										
3103										

;*****

M05

DBKTGD MAINDEC-11-DBKTG-D KT11-D EXERCISER
 DBKTGD.P11 21-SEP-76 11:27

MACY11 27(1006) 21-SEP-76 11:30 PAGE 64
 TEST THAT RO WASN'T CLEARED BY FALSE SELECTION

```

3104 .SBTTL TEST THAT RO WASN'T CLEARED BY FALSE SELECTION
3105 ;*****
3106 013640 022700 052525          CMP      #52525,RO
3107 013644 001401                BEQ      .+4
3108 013646 104006                HLT
3109 013650 104400                SCOPE
3110
3111 ;*****
3112 .SBTTL TEST COMBINATIONS OF NUMBERS WITH COMPARE INSTRUCTION
3113 ;*****
3114 013652 004767 001146          JSR      PC,RNGEN          ;GET RANDOM NUMBER.
3115 013656 010001                MOV      RO,R1
3116 013660 020001          CMP1:  CMP      %0,%1          ;ARE THE EQUAL
3117 013662 001401                BEQ      .+4
3118 013664 104006                HLT
3119 013666 104400                SCOPE
3120
3121 ;*****
3122 .SBTTL TEST ROTATING RANDOM NUMBERS
3123 ;*****
3124 013670 004767 001130          JSR      PC,RNGEN          ;GET RANDOM NUMBER.
3125 013674 010067 164720          MOV      RO,REFF          ;PUT IN REF WORD.
3126 013700 016767 164714          ROTALL: MOV      REFF,TEST
3127 013706 006067 164710          ROR      TEST
3128 013712 006067 164704          ROR      TEST
3129 013716 006067 164700          ROR      TEST
3130 013722 006167 164674          ROL      TEST
3131 013726 006167 164670          ROL      TEST
3132 013732 006167 164664          ROL      TEST
3133 013736 100004                BPL      .+12
3134 013740 103007                BCC      .+20              ;Z=1
3135 013742 102013                BVC      .+30              ;Z=1, C=1
3136 013744 104006                HLT              ;Z=C, BUT V=1
3137 013746 000411                BR       .+24
3138 013750 103006                BCC      .+16              ;Z=0
3139 013752 102407                BVS      .+20              ;Z=0, C=1
3140 013754 104006                HLT              ;Z NOT EQUAL C, V=1
3141 013756 000405                BR       .+14
3142 013760 102404                BVS      .+12              ;Z=1, C=0
3143 013762 104006                HLT              ;Z NOT EQUAL C, V=1
3144 013764 000402                BR       .+6
3145 013766 102001                BVC      .+4              ;Z=0, C=0
3146 013770 104006                HLT              ;Z=C, BUT V=1
3147 013772 026767 164624          CMP      TEST,REFF
3148 014000 001401                BEQ      .+4
3149 014002 104006                HLT
3150 014004 104400                SCOPE
3151
3152          000620          REF=REFF
3153
3154 ;*****
3155 .SBTTL TEST ROTATING BYTE EVEN/ODD, RANDOM NUMBERS
3156 ;*****
3157 014006 004767 001012          JSR      PC,RNGEN          ;GET RANDOM NUMBER.
3158 014012 010067 164602          MOV      RO,REFF          ;PUT IN REF WORD.
3159 014016 004767 000006          JSR      %7,ROTBE
  
```

N05

```

3160 014022 004767 000110          JSR    %7,ROTBO
3161 014026 000503                    BR     ROTEN1
3162 014030 016767 164564 164564 ROTBE: MOV    REFF,TEST
3163 014036 106067 164560                    RORB   TEST          ;ROTATE BYTE EVEN
3164 014042 106067 164554                    RORB   TEST
3165 014046 106067 164550                    RORB   TEST
3166 014052 106167 164544                    ROLB   TEST
3167 014056 106167 164540                    ROLB   TEST
3168 014062 106167 164534                    ROLB   TEST
3169 014066 100004                    BPL    .+12
3170 014070 103007                    BCC    .+20          ;Z=1
3171 014072 102013                    BVC    .+30          ;Z=1, C=1
3172 014074 104006                    HLT
3173 014076 000411                    BR     .+24          ;Z=C, BUT V=1
3174 014100 103006                    BCC    .+16          ;Z=0
3175 014102 102407                    BVS    .+20          ;Z=0, C=1
3176 014104 104006                    HLT          ;Z NOT EQUAL C, V=1
3177 014106 000405                    BR     .+14
3178 014110 102404                    BVS    .+12          ;Z=1, C=0
3179 014112 104006                    HLT          ;Z NOT EQUAL C, V=1
3180 014114 000402                    BR     .+6
3181 014116 102001                    BVC    .+4          ;Z=0, C=0
3182 014120 104006                    HLT          ;Z=C, BUT V=1
3183 014122 026767 164474 164470    CMP    TEST,REFF
3184 014130 001401                    BEQ    .+4
3185 014132 104006                    HLT
3186 014134 000207                    RTS     %7
3187 014136 106067 164461          ROTBO: RORB   TEST+1          ;ROTATE BYTE ODD
3188 014142 106067 164455                    RORB   TEST+1
3189 014146 106067 164451                    RORB   TEST+1
3190 014152 106167 164445                    ROLB   TEST+1
3191 014156 106167 164441                    ROLB   TEST+1
3192 014162 106167 164435                    ROLB   TEST+1
3193 014166 100004                    BPL    .+12
3194 014170 103007                    BCC    .+20          ;Z=1
3195 014172 102013                    BVC    .+30          ;Z=1, C=1
3196 014174 104006                    HLT          ;Z=C, BUT V=1
3197 014176 000411                    BR     .+24
3198 014200 103006                    BCC    .+16          ;Z=0
3199 014202 102407                    BVS    .+20          ;Z=0, C=1
3200 014204 104006                    HLT          ;Z NOT EQUAL C, V=1
3201 014206 000405                    BR     .+14
3202 014210 102404                    BVS    .+12          ;Z=1, C=0
3203 014212 104006                    HLT          ;Z NOT EQUAL C, V=1
3204 014214 000402                    BR     .+6
3205 014216 102001                    BVC    .+4          ;Z=0, C=0
3206 014220 104006                    HLT          ;Z=C, BUT V=1
3207 014222 026767 164374 164370    CMP    TEST,REFF
3208 014230 001401                    BEQ    .+4
3209 014232 104006                    HLT
3210 014234 000207                    RTS     %7
3211 014236 104400          ROTEN1: SCOPE
  
```

```

3212
3213 ;*****
3214 .SBTTL ADD AND SUBTRACT RANDOM NUMBERS AGAINST FIXED NUMBERS
3215 ;*****
  
```

```

3216
3217 014324 011667 000054
3218 014324 004767 000554
3219 014324 010067 164344
3220 014324 004767 000002
3221 014324 000420
3222 014324 016767 164332 164332
3223 014324 066767 000024 164324
3224 014324 166767 000016 164316
3225 014324 026767 164310 164310
3226 014324 001401
3227 014324 104006
3228 014324 000207
3229 014324 000000
3230 014324 104400
3231
3232
3233
3234
3235 014324 004767 000474
3236 014330 010067 164266
3237 014334 012767 177777 164256
3238 014342 160067 164252
3239 014346 005167 164250
3240 014352 026767 164244 164240
3241 014360 001401
3242 014362 104006
3243
3244 014364 104400
3245
3246
3247
3248
3249 014366 004767 000432
3250 014372 010067 164224
3251 014376 012767 177777 164214
3252 014404 160067 164210
3253 014410 105167 164206
3254 014414 126767 164202 164176
3255 014422 001401
3256 014424 104006
3257
3258 014426 104400
3259
3260
3261
3262
3263 014430 004767 000370
3264 014434 010067 164162
3265 014440 012767 177777 164152
3266 014446 160067 164146
3267 014452 105167 164145
3268 014456 126767 164141 164135
3269 014464 001401
3270 014466 104006
3271

```

:A+B=C, C-A=B, BF SHOULD EQUAL BI

```

MOV 2%6, NUMA
JSR PC, RNGEN ;GET RANDOM NUMBER.
MOV RO, REFF ;PUT IN REF WORD.
JSR %7, ADSUB
BR ARIEND
ADSUB: MOV REFF, TEST
ADD NUMA, TEST
SUB NUMA, TEST
CMP REFF, TEST
BEQ .+4
HLT
RTS %7
NUMA: 0
ARIEND: SCOPE

```

```

:*****
.SBTTL TEST COMPLIMENTING RANDOM NUMBERS
:*****

```

```

JSR PC, RNGEN ;GET RANDOM NUMBER.
MOV RO, TEST ;PUT IN TEST.
MOV #-1, REFF ;PREDETERMINE RESULT IN REFF.
SUB RO, REFF
COM TEST ;DO THE COMPLEMENT.
CMP TEST, REFF ;EXPECTED RESULT?
BEQ .+4 ;BR IF YES.
HLT ;ERROR!!! COMPLEMENT OF TEST DID NOT
;MATCH EXPECTED RESULTS AS IN REFF.
SCOPE

```

```

:*****
.SBTTL TEST COMB (EVEN BYTE) RANDOM NUMBERS.
:*****

```

```

JSR PC, RNGEN ;GET RANDOM NUMBER.
MOV RO, TEST ;PUT IN TEST.
MOV #-1, REFF ;PREDETERMINE THE RESULT.
SUB RO, REFF
COMB TEST ;DO COMPLEMENT EVEN BYTE.
CMPB TEST, REFF ;EXPECTED RESULT?
BEQ .+4 ;BR IF YES. OK.
HLT ;ERROR!!! RESULT IN TEST DOES NOT MATCH
;EXPECTED RESULT IN REFF.
SCOPE

```

```

:*****
.SBTTL TEST COMB (ODD BYTE) RANDOM NUMBERS
:*****

```

```

JSR PC, RNGEN ;GET RANDOM NUMBER.
MOV RO, TEST ;PUT IN TEST.
MOV #-1, REFF ;PREDETERMINE RESULT.
SUB RO, REFF
COMB TEST+1 ;DO ODD BYTE COMPLEMENT.
CMPB TEST+1, REFF+1 ;EXPECTED RESULT?
BEQ .+4 ;BR IF YES. OK.
HLT ;ERROR!!! RESULT IN TEST DOES NOT MATCH
;EXPECTED RESULT IN REFF.

```

```

3272 014470 104400
3273
3274
3275
3276
3277 014472 004767 000326
3278 014476 110067 000426
3279 014502 110067 000423
3280 014506 126767 000416 000415
3281 014514 001401
3282 014516 104006
3283 014520 002001
3284 014522 104006
3285 014524 003401
3286 014526 104006
3287 014530 104400
3288
3289
3290
3291
3292 014532 012767 000200 164062
3293 014540 000367 164056
3294 014544 100001
3295 014546 104006
3296 014550 001401
3297 014552 104006
3298 014554 000367 164042
3299 014560 100401
3300 014562 104006
3301 014564 001001
3302 014566 104006
3303 014570 104400
3304
3305
3306
3307
3308 014572 004767 000226
3309 014576 010067 164016
3310 014602 105067 164013
3311 014606 116767 164006 164007
3312 014614 105067 164002
3313 014620 000367 163776
3314 014624 026767 163772 163766
3315 014632 001401
3316 014634 104006
3317 014636 104400
3318

```

SCOPE

```

:*****
:SBTTL TEST COMPARE RANDOM VALUES EVEN BYTE WITH ODD
:*****

```

```

JSR PC,RNGEN ;GET RANDOM NUMBER.
MOVB RO,TEMP ;PUT IN LOW BYTE OF TEMP.
MOVB RO,TEMP+1 ;PUT IN HIGH BYTE.
CMPB TEMP,TEMP+1 ;DO THE COMPARE.
BEQ .+4 ;BR IF EQUAL. OK.
HLT ;COMPARE FAILED.
BGE .+4
HLT ;V IS NOT = TO N
BLE .+4
HLT ;V IS SET
SCOPE

```

```

:*****
:SBTTL TEST SWAB
:*****

```

```

MOV #0200,TEST
SWAB TEST
BPL .+4
HLT
BEQ .+4
HLT
SWAB TEST
BMI .+4
HLT
BNE .+4
HLT
SCOPE

```

```

:*****
:SBTTL TEST RANDOM COMBINATIONS OF SWAB
:*****

```

```

JSR PC,RNGEN ;GET RANDOM NUMBER.
MOV RO,REFF ;PUT IN REF.
CLRB REFF+1
MOVB REFF,TEST+1 ;PUT IN TEST HIGH BYTE.
CLRB TEST ;AND CLEAR THE LOW BYTE OF TEST.
SWAB TEST ;NOW DO THE SWAB!
CMP TEST,REFF ;REFF AND TEST MUST BE SAME.
BEQ .+4 ;BR IF SAME. OK.
HLT ;ERROR!! SWAB FAILURE.
SCOPE

```


015142
015144
015146
015150
015154
015160
015162
015164
015166
015172
015174
015176
015200
015202
015204
015210
015214
015220
015224
015230
015236
015242
015244
015246
015250
015252
015254
015256
015264
015266
015272
015300
015302
015310
015316
015320
015326

010146
010246
010346
012701 000540
012703 000010
012102
005022
077302
020127 000546
003770
012603
012602
012601
000207
162716 000002
006576 000000
012667 000022
062716 000002
105067 000013
062767 015244 000004
017707 000000
000000
000000
000000
000000
000000
000000
000000
000000
015776
015256 032767 000007 163264
001406
012716 015676
012766 000340 000002
000002
042737 000340 177776
026767 163262 163176
001444
062737 020000 000034
062767 020000 163246

```
*****  
:SBTTL ROUTINE SUBROUTINE TO INITIALIZE ALL PAGES TO NR, BANK 0, 1 PAGE, UP/  
*****  
NRALL: MOV R1, -(R6) ;SAVE REGISTERS  
MOV R2, -(R6)  
MOV R3, -(R6)  
MOV @IPDRTAB, R1 ;R1 HOLDS ADDRESS OF CURRENT POSITION  
;IN TABLE OF ADDRESSES  
NRLOOP: MOV @R3  
MOV (R1)+, R2 ;R3 USED AS COUNTER  
;R2 CONTAINS ADDRESS OF PDR OR  
;PAR TO BE CLEARED  
CLR (R2)+ ;CLEAR ALL ASR'S FOR THIS MODE  
SOB R3, -2  
CMP R1, @IPDREND ;CHECK FOR DONE  
BLE NRLOOP ;CLEAR ALL IN NEXT MODE IF NOT DONE  
MOV (R6)+, R3  
MOV (R6)+, R2  
MOV (R6)+, R1  
RTS %7  
  
*****  
:SBTTL EMT HANDLER/  
*****  
:FIRST 3 CALLS LEFT OPEN IN TABLE FOR EASY PATCHES/  
EMTSRV: SUB #2, @SP ;GET CALL  
MFPI @SP  
MOV (SP)+, EPC  
ADD #2, @SP  
CLRB EPC+1 ;SAVE OFFSET ONLY  
ADD @EMTAB, EPC ;POINT TO TABLE OF ADDRESSES  
MOV @EPC, PC ;JUMP TO DESIRED ROUTINE  
  
EPC: 0  
PATCH1=0  
PATCH2=0  
PATCH3=0  
  
EMTAB: PATCH1 ;PATCH IN ADDRESS OF ROUTINE  
PATCH2  
PATCH3  
PRINT ;ERROR PRINTOUT  
EOBSRV ;END OF BANK  
  
*****  
:SBTTL ROUTINE END OF BANK SERVICE  
*****  
EOBSRV: BIT #7, SREG1 ;KT11-D/USER-KERNEL/4KAS 32 INHIBITED?  
BEQ EOB2 ;NO - CONTINUE  
EOB1C: MOV #LOGIC, (SP) ;GO TO BEGIN  
MOV #340, 2(SP) ;WILL ASSUME PRIORITY 7.  
RTI  
EOB2: BIC #340, @PSR  
CMP CURPAR, UPAR7 ;LAST USER ASR DONE?  
BEQ NXTBNK ;YES - GO FIND NEXT BANK  
ADD #20000, @#34 ;UPDATE SCOPE VECTOR ADDRESS IN BANK 0  
ADD #20000, BNKSTR ;UPDATE BANK START TO REFERENCE CURRENT ASR
```

```

3473 015334 016716 163242      MOV      BNKSTR,(SP)
3474 015340 026767 163144 163230  CMP      UPARD,CURPAR
3475 015346 001404      BEQ      NXTSEG
3476 015350 005077 163222      CLR      @CURPAR          ;SET PREVIOUS ASR TO NR, BANK 0
3477 015354 005077 163220      CLR      @CURPDR
3478 015360 062767 000002 163210  NXTSEG:  ADD      #2,CURPAR          ;UPDATE POINTERS TO NEXT SEGMENT
3479 015366 062767 000002 163204      ADD      #2,CURPDR
3480 015374 012777 077406 163176      MOV      #77406,@CURPDR  ;SET NEXT SEGMENT RW, 4K
3481 015402 016777 163164 163166      MOV      CURBNK,@CURPAR  ;MAP NEXT SEGMENT TO CURRENT BANK
3482 015410 052737 030000 177776      BIS      #30000,@PSR     ;SET PREVIOUS MODE TO USER
3483 015416 006506      MFPI     R6              ;PICK UP USER STACK POINTER
3484 015420 062716 020000      ADD      #20000,@R6     ;MAP IT TO NEXT ASR
3485 015424 006606      MTPI     R6              ;PUT IT BACK
3486 015426 000002      RTI
3487 015430 012746 000400      NXTBNK: MOV      #UBUFF,-(SP)
3488 015434 052737 030000 177776      BIS      #30000,@PSR
3489 015442 006606      MTPI     R6
3490 015444 013737 000572 000574      MOV      @CURBNK,@OLDBNK ;SAVE PREV BANK ADDRESS
3491 015452 062767 000200 163112  BNKTST:  ADD      #200,CURBNK
3492 015460 006367 163126      ASL      COREPT
3493 015464 103006      BCC
3494 015466 012767 000001 163116      MOV      #1,COREPT
3495 015474 012767 000610 163112      MOV      #MEM1,MEMUT
3496 015502 022767 007600 163062  1$:      CMP      #7600,CURBNK   ;CHECK FOR EXTERNAL BANK
3497 015510 001666      BEQ      EOB1C          ;BR IF YES TO START ANOTHER PASS.
3498 015512 016777 163054 163012  EOB3:    MOV      CURBNK,@KPAR2  ;MAP KERNEL SEGMENT 2 TO BANK BEING LOOKED FOR
3499 015520 012777 077406 162774      MOV      #77406,@KPDR2
3500 015526 036777 163060 163060      BIT      COREPT,@MEMUT
3501 015534 001746      BEQ      BNKTST
3502 015536 042737 160000 000034      BIC      #160000,@#34   ;INITIALIZE SCOPE VECTOR ADDRESS
3503 015544 005001      CLR      R1              ;R1 ADDRESSES BANK 0 THRU KERNEL ASR0
3504 015546 012702 040000      MOV      #40000,R2      ;R2 ADDRESSES NEW BANK THRU KERNEL ASR2
3505 015552 012703 015140      MOV      #0,R3
3506 015556 006203      ASR      R3
3507 015560 012122      CORMOV: MOV      (R1)+,(R2)+
3508 015562 077302      SOB      R3,CORMOV
3509 015564 016767 162720 163004      MOV      UPARD,CURPAR   ;FIRST ASR CHECKED IS USER ASR0
3510 015572 016767 162704 163000      MOV      UPDR0,CURPDR
3511 015600 016777 162766 162770      MOV      CURBNK,@CURPAR
3512 015606 012777 077406 162764      MOV      #77406,@CURPDR
3513 015614 005077 162674      CLR      @UPAR7
3514 015620 005077 162662      CLR      @UPDR7
3515 015624 026727 162744 000000      CMP      OLDBNK,#0     ;PREV BANK = 0
3516 015632 001414      BEQ      EOB6          ;YES, DO NOT CLEAR
3517 015634 016777 162734 162670      MOV      OLDBNK,@KPAR2
3518 015642 012777 077406 162652      MOV      #77406,@KPDR2
3519 015650 012701 040000      MOV      #40000,R1
3520 015654 012703 007630      MOV      #7630,R3
3521 015660 005021      BNKLP:  CLR      (R1)+
3522 015662 077302      SOB      R3,BNKLP
3523 015664 012716 005654      EOB6:   MOV      #BEGIN,(SP)
3524 015670 011667 162706      MOV      (SP),BNKSTR
3525 015674 000002      RTI
;*****
.SBTTL  END OF PASS CODE STARTS HERE

```



```

3529
3530 015676 042777 000001 162574 LOGIC: BIC #1,DSRO ;TURN OFF KT11.
3531 015704 012737 000016 000014 MOV #16,D#14 ;RESET THE TRACE VECTOR.
3532 015712 005037 000016 CLR D#16
3533 015716 032737 000001 000552 BIT #1,D#SREG2 ;TTY OUT SELECTED
3534 015724 001404 BEQ 1$ ;YES, NO ASTERISK
3535 015726 004767 000444 JSR PC,BELL ;RING BELL TOO.
3536 015732 004767 000414 JSR PC,STAR ;TYPE ASTERISK.
3537 015736 005267 177056 1$: INC PASCNT ;INCREMENT PASS COUNT.
3538
3539 ;*****START OF "ACT11/XXDP EOP HOOKS"*****
3540 015742 013701 000042 MOV D#42,R1
3541 015746 001405 BEQ HERE
3542 015750 000005 RESET
3543 015752 004711 SENDAD: JSR %7,D#R1
3544 015754 000240 NOP
3545 015756 000240 NOP
3546 015760 000240 NOP
3547
3548 ;*****END OF "ACT11/XXDP EOP HOOKS"*****
3549 015762 000005 HERE: RESET ;ISSUE RESET TO HALT I/O.
3550 015764 005167 162614 COM TRPB ;COMPLEMENT THE TRACE SWITCH.
3551 015770 000137 000636 JMP D#RSTRT ;RESTART.

```

```

3552 ;*****
3553 .SBTTL RTT EXECUTED WHEN TRACE IS ON/
3554 ;*****
3555 015774 000006 TRTRP: RTT
3556
3557
3558 ;*****
3559 .SBTTL ROUTINE PRINT
3560 ;*****
3561 ;ENTERED WITH SYSTEM TRAP CALL (HLT)
3562 ;PRINT OUT THE ERROR PC+2, STATUS REGISTER, AND LOCATION IN BACKGROUND
3563 PRINT: TST PRTON ;CHECK PRINT ON FLAG
3564 BEQ .+4
3565 RTI ;IF ANOTHER HALT IS BEING PRINTED, SKIP THIS ONE
3566 INC PRTON
3567 MOV #340,PSR ;SET PRIORITY TO 7
3568 BIT SR,#20000 ;TEST FOR INHIBIT PRINT OUT
3569 BNE CK ;BR TO INHIBIT PRINT.
3570 MOV (6)+,SAVPC ;PC OF FAILING ROUTINE
3571 MOV (6)+,SAVPSR ;PSR OF ERROR CONDITION
3572 CMP -(6),-(6) ;RESTORE STACK
3573 MOV #200,PSR
3574 JSR %7,CRLF ;OUTPUT CARRIAGE RETURN AND LINE FEED
3575 MOV SAVPC,PTEMP1 ;LOAD WITH FAILING PC+2
3576 JSR %7,PROCT ;PRINT FAILING PC+2
3577 JSR %7,SPACE
3578 MOV SAVPSR,PTEMP1 ;LOAD PROCESSOR STATUS
3579 JSR %7,PROCT ;PRINT PROCESSOR STATUS
3580 JSR %7,SPACE
3581 MOV CURBNK,PTEMP1
3582 JSR %7,PROCT
3583 JSR %7,SPACE
3584 MOV RETURN,PTEMP1
3585 JSR %7,PROCT
3586 016140 023727 000042 015752 CK: CMP #42,#SENDAD ;IN ACT11?
3587 016146 001403 BEQ 1$ ;BR IF YES TO HALT.
3588 016150 005767 161414 TST SR ;CHECK SR FOR HALT SWITCH
3589 016154 100001 BPL .+4 ;BRANCH IF NOT SET
3590 016156 000000 1$: HALT ;HALT ON ERROR UP
3591 016160 005067 000006 CLR PRTON ;ROUTINE DONE - CLEAR FLAG
3592 016164 000002 RTI ;RETURN TO MAIN LINE
3593 SAVPC: 0
3594 SAVPSR: 0
3595 PRTON: 0
3596
3597
3598 ;*****
3599 .SBTTL ROUTINE PROCT
3600 ;*****
3601 ;SUBROUTINE TO PRINT OUT OCTAL NUMBER/
3602 016174 012727 000006 016200 PROCT: MOV #6,#PTEMP3 ;CLEAR R4 FOR COUNTING CHARACTERS OUTPUT
3603 016200 016200 PTEMP3=-2
3604 016202 005067 000136 CLR PRFLG ;INITIALIZE CARRY FLAG FOR ROTATES
3605 016206 012767 000060 000134 MOV #60,PTEMP2 ;SETUP R3
3606 016214 005767 000126 TST PTEMP1 ;CHECK BIT 15 OF NUMBER
3607 016220 100002 BPL .+6 ;BRANCH IF ZERO

```

```

3608 016222 005267 000122      INC      PTEMP2      ;INCREMENT R3 IF ONE
3609 016226 006167 000114      ROL      PTEMP1      ;ROTATE LEFT MOST OCTAL TO RIGHT END
3610 016232 006167 000110      ROL      PTEMP1
3611 016236 005567 000102      ADC      PRFLG        ;STORE CARRY
3612 016242 016746 000102      P.WAIT: MOV      PTEMP2,-(SP) ;OUTPUT THE CHARACTER
3613 016246 004767 000210      JSR      PC,CHR0UT   ;DO IT.
3614 016252 005367 177722      DEC      PTEMP3      ;COUNT
3615 016256 001001                BNE      P.CNT1      ;BRANCH IF NOT DONE
3616 016260 000207                RTS                ;BRANCH IF NOT DONE
3617 016262 000241      P.CNT1: CLC          ;CLEAR CARRY
3618 016264 005767 000054      TST      PRFLG       ;CHECK FOR PREVIOUS CARRY
3619 016270 001403                BEQ      .+10        ;BRANCH IF PREVIOUSLY ZERO
3620 016272 005067 000046      CLR      PRFLG       ;INITIALIZE FLAG
3621 016276 000261                SEC                ;SET CARRY
3622 016300 006167 000042      ROL      PTEMP1      ;ROTATE NEXT CHARACTER INTO RIGHT END OF REGISTER
3623 016304 006167 000036      ROL      PTEMP1
3624 016310 006167 000032      ROL      PTEMP1
3625 016314 005567 000024      ADC      PRFLG        ;STORE CARRY
3626 016320 016767 000022 000022  MOV      PTEMP1,PTEMP2 ;LOAD DATA INTO R3
3627 016326 042767 177770 000014  BIC      #177770,PTEMP2 ;CLEAR ALL BUT LOWEST OCTAL DIGIT
3628 016334 052767 000060 000006  BIS      #60,PTEMP2   ;SET TO ASCII EQUIVALENT
3629 016342 000737                BR        P.WAIT     ;LOOP
3630 016344 000000                PRFLG:  0
3631 016346 000000                PTEMP1: 0
3632 016350 000000                PTEMP2: 0

```

3633
3634
3635
3636
3637

```

;*****
.SBTL ROUTINE STAR
;*****
;SUBROUTINE TO OUTPUT ASTERISK.
STAR: JSR      PC,CRLF   ;OUTPUT CRLF.
      MOV      #52,-(SP) ;GO OUTPUT A *
      BR        BELL1

```

3642
3643
3644
3645
3646
3647

```

;*****
.SBTL ROUTINE SPACE
;*****
;SUBROUTINE TO ISSUE SPACE/
SPACE: MOV      #40,-(SP) ;OUTPUT SPACE.
      JSR      PC,CHR0UT ;DO IT.
      RTS                ;RETURN

```

3651
3652
3653
3654
3655

```

;*****
.SBTL ROUTINE BELL
;*****
;BELL ON PASS COMPLETE
BELL:  MOV      #7,-(SP) ;OUTPUT BELL.
BELL1: JSR      PC,CHR0UT ;DO IT.
      DEC      #0        ;SLIGHT DELAY.
      BNE      #-4
BELL2: RTS                ;7

```

3656
3657 016376 012746 000007
3658 016402 004767 000054
3659 016406 005327 000000
3660 016412 001375
3661 016414 000207
3662
3663

```

3664 ;*****
3665 .SBTTL ROUTINE CRLF
3666 ;*****
3667 :SUBROUTINE TO OUTPUT CARRIAGE RETURN AND LINEFEED/
3668 016416 012746 000015 CRLF: MOV #15, -(SP) ;OUTPUT CR.
3669 016422 004767 000034 JSR PC, CHROUT ;DO IT.
3670 016426 012746 000012 MOV #12, -(SP) ;OUTPUT LF.
3671 016432 004767 000024 JSR PC, CHROUT ;DO IT.
3672 016436 016746 162162 MOV FILLCT, -(SP) ;GET THE FILL COUNT.
3673 016442 001405 BEQ 2$ ;BR IF 0.
3674 016444 005046 1$: CLR -(SP) ;WILL OUTPUT NULLS FOR FILLERS.
3675 016446 004767 000010 JSR PC, CHROUT ;DO IT.
3676 016452 005316 DEC (SP) ;DONE?
3677 016454 001373 BNE 1$ ;BR IF NOT.
3678 016456 005726 2$: TST (SP)+ ;CLEAN UP STACK.
3679 016460 000207 RTS PC ;RETURN.
3680
3681 ;*****
3682 .SBTTL ROUTINE CHROUT
3683 ;*****
3684 :SUBROUTINE TO OUTPUT CHARACTER TO CONSOLE TTY.
3685 016462 016677 000002 161722 CHROUT: MOV 2(SP), @TDBR ;LOAD THE CONSOLE BUFFER REG.
3686 016470 105777 161714 1$: TSTB @TCSR ;READY?
3687 016474 100375 BPL 1$ ;BR IF NOT. WAIT.
3688 016476 012616 MOV (SP)+, (SP) ;SET UP FOR EXIT.
3689 016500 000207 RTS PC ;RETURN.
3690
3691 ;ENTER HERE ON POWER FAIL/
3692 PFAIL: MOV @#24, -(6)
3693 016502 013746 000024 MOV %6, SAVR6 ;STORE STACK POSITION
3694 016506 010667 000010 MOV #RESTR, @#24 ;HALT ON POWER DOWN NORMAL
3695 016512 012737 016524 000024 HALT ;STACK IS SAVED HERE
3696 016520 000000 SAVR6: 0 ;RESTORE STACK WHEN POWERING UP
3697 016522 000000 RESTRT: MOV SAVR6, %6
3698 016524 016706 177772 MOV (6)+, @#24
3699 016530 012637 000024 MOV (SP)+, (SP)+ ;RESTORE STACK
3700 016534 022626 CMP HLT ;POWER FAIL OCCURRED
3701 016536 104006 HLT
3702 016540 000167 162072 JMP RSTRT ;RETURN TO MAIN LINE
3703
3704
3705 016544 000207 USER: RTS %7 ;OVERLAY USER ROUTINE HERE IF 4KW
3706 ;USE BANK1 IF 8KW
3707 017760 .=17760
3708 017760 000000 KSTACK: 0
3709 000001 .END

```

A	015106	IE	= 000500	NRALL	015142	ROTALL	013700	TCF1	004066
ACTSW1	000626	IPDREN	000546	NRL00P	015154	ROTBE	014030	TCF1A	004060
ACTSW2	000630	IPDRTA	000540	NUMA	014320	ROTBO	014136	TCF2	004120
ADSUB	014262	IRF	003674	NXTBNK	015430	ROTEN1	014236	TCF3	004134
ARIEND	014322	IRK	003600	NXTSEG	015360	RP1	015072	TCF4	004202
B	015076	KPAR0	000526	OLDBNK	000574	RP2	015074	TCIV	000566
BEGIN	005654	KPAR1	000530	PASCNT	015020	RSTRT	000636	TCLAST	004010
BEGINX	005622	KPAR2	000532	PATCH1=	000000	R6	=%000006	TCRBK	004472
BELL	016376	KPAR7	000534	PATCH2=	000000	SAVPC	016166	TCRBUF	004622
BELL1	016402	KPDRO	000516	PATCH3=	000000	SAVPSR	016170	TCRB1	004530
BELL2	016414	KPDR1	000520	PFAIL	016502	SAVR6	016522	TCR1	004340
BNKLP	015660	KPDR2	000522	PRFLG	016344	SCOPE =	104400	TCR1A	004374
BNKSTR	000602	KPDR7	000524	PRINT	015776	SCOPEB	014770	TCR2	004402
BNKTST	015452	KSTACK	017760	PROCT	016174	SCOPEC	014702	TCR3	004416
C	015120	KWLST	000424	PRTON	016172	SCOPEF	015014	TCR4	004464
CHROUT	016462	KWLVC	000422	PSR	= 177776	SCOPEG	014776	TCSR	= 000410
CK	016140	LKCSR	000426	PTEMP1	016346	SECTION=	***** U	TCST	000556
CLINCT=	003452	LK1	003236	PTEMP2	016350	SEGM1	002240	TCSTA	000570
CMP1	013660	LK2	003256	PTEMP3=	016200	SETEX	002220	TCWBK	004260
COREPT	000612	LK3	003260	P.CNT1	016262	SETSEG	002364	TCWBUF	004622
CORMOV	015560	LK4	003276	P.WAIT	016242	SETUSE	002270	TCWB1	004312
CRLF	016416	LLIMIT	004002	R	= 004000	SOLPAT=	003360	TCWC	000562
CURBNK	000572	LOGIC	015676	RB	= 000002	SPACE	016364	TC1	004556
CURPAR	000576	LPCSR	000430	RD	= 000004	SR	= 177570	TC2	004576
CURPAT=	003362	LPDBR	000432	REF	= 000620	SREG1	000550	TDBR	= 000412
CURPDR	000600	LPINTR	003402	REFF	000620	SREG2	000552	TEMP	015130
D	015140	LPST	000436	RENDZ	004212	SRO	000500	TEST	000622
DATA2 =	003170	LPVC	000434	REND1	004242	STAR	016352	TIME	003354
DET1	002536	LP1	003356	RESTRT	016524	START	000666	TITLE	000736
DET2	002654	LP2	003364	RETURN	015016	STARTA	000770	TJSR1	013250
DET3	003152	LP3	003472	RFCAR	000446	STARTX	001146	TJSR2	013252
DET4	002570	LP4	003502	RFCAR	000450	START1	001164	TJSR3	013264
DO	= 000001	LP5	003540	RFCSRH	000452	START2	001164	TMEMEX	001324
DONE	014640	LP6	003440	RFDAE	000440	STR28	002752	TRCSR	000406
EIGHT	002606	LP7	003436	RFDAR	000442	STR4	002706	TRPB	000604
EMTAB	015244	MAIN	002500	RFFUNC	004000	ST3	001530	TRTRP	015774
EMTSRV	015204	MAP1	001274	RFST	000456	ST4	001630	TTCR	000410
EOB	= 104010	MAP2	001364	RFSTAR	003644	ST5	001674	TTDBR	000412
EOBSRV	015256	MEMUT	000614	RFVC	000454	ST6	001762	TTPST	000416
EOB1C	015266	MEMO	000606	RFWC	000444	ST7	002032	TTPVC	000414
EOB2	015302	MEM1	000610	RF1	003650	ST8	002130	TTSV	000420
EOB3	015512	MODE	002372	RKBAR	000466	SUBR1	014644	TYOUT	003156
EOB6	015664	MOD12	003106	RKCSR	000470	SUBR2	014646	TYOUTR	003202
EPC	015242	MOD16	003072	RKCSRH	000472	SUBR3	014652	TYOUT1	003172
ERROK	003740	MOD20	003056	RKDAE	000462	SUBR4	014660	UBUFF	000400
ERRSET	003734	MOD24	003042	RKDAH	000460	SUBR5	014666	UPAR0	000510
FENDZ	004014	MOD4	003136	RKFUNC	003642	SUBR6	014674	UPAR1	000512
FEND1	004036	MOD8	003122	RKST	000476	TBANK	000616	UPAR7	000514
FILLCT	000624	MOVE	002666	RKSTAR	003550	TCBA	000564	UPDRO	000502
FTITLE	000536	MOVEPT	001334	RKVC	000474	TCCM	000554	UPDR1	000504
HERE	015762	N	= 000001	RKWC	000464	TCDT	000560	UPDR7	000506
HLT	= 104006	NODEV	002516	RK1	003554	TCEXPE	004012	USEALL	002312
ICOUNT	015010	NOP	= 000240	RNGEN	015024	TCFIRS	004006	USER	016544

M06

DBKTGD MAINDEC-11-DBKTG-D KT11-D EXERCISER MACY11 27(1006) 21-SEP-76 11:30 PAGE 78
DBKTGD.P11 21-SEP-76 11:27 SYMBOL TABLE

WD = 000014
WORDCT 004004
XDPCNT 015022

XDPSW1 000632
XDPSW2 000634
XFENDZ 004210

XFER12 003020
XFER16 003010
XFER20 003000

XFER24 003770
XFER28 002760
XFER8 003030

SENDAD 015752
\$ICNT 015012
= 017762

. ABS. 017762 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DBKTGD,DBKTGD/SOL=DBKTGD.P11
RUN-TIME: 11 23 .9 SECONDS
RUN-TIME RATIO: 90/35=2.5
CORE USED: 5K (9 PAGES)

