



B01

EOF1DZQMCSEQ

00010000

770225

PDP10 411

HDR1DFKTGBSEQ

00010000

770225

.REM \*

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DFKTG-B-D  
PRODUCT NAME: 11/34 MEMORY MANAGEMENT EXERCISER  
PRODUCT DATE: JANUARY 1977  
MAINTAINER: DIAGNOSTIC PROGRAMMING  
AUTHOR: DIAGNOSTIC ENGINEERING

COPYRIGHT (C) DIGITAL EQUIPMENT CORPORATION  
1975, 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 THE MEMORY MANAGEMENT PORTION OF A PDP 11/34. 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 MEMORY MANAGEMENT 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-11/34 STANDARD COMPUTER  
TELETYPE OR EQUIVALENT

#### 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

SET DESIRED MEMORY MANAGEMENT OPTION SWITCHES  
(IN LOC. 174, MMOPT) (SEE SECTION 4.2)  
ALL ZERO FOR WORST CASE TESTING  
SET DESIRED SWITCH REGISTER BITS.  
(USE LOC. 176 FOR SOFTWARE SWITCH REGISTER  
IF NECESSARY). (SEE SECTION 4.3 AND 5.1.2)  
LOAD ADDRESS 200 AND START.

## 4.1 NORMAL STARTING PROCEDURE (CONTINUED)

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 BEFORE STARTUP DETERMINE THE WAY IN WHICH MEMORY IS MAPPED AND EXERCISED:

MMOPT BIT0=1---INHIBIT MEMORY MGMT. (SR0<0> WILL NOT BE SET AT ALL)  
MMOPT BIT1=1---INHIBIT USE OF USER MODE.

(ALSO INHIBITS 4K AS 32K)

MMOPT BIT2=1---INHIBIT 4K AS 32 K (ALSO INHIBITED IF EITHER SW0 OR SW1 IS SET)-SEE SECTION 5.3.1 FOR EXPLANATION

MMOPT BIT5=1---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 IN THE SWITCH REGISTER (USE LOC. 176 FOR SOFTWARE SW. REG. IF NECESSARY) ALSO SEE SEC. 5.1.2. 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)

## 4.4 RESTART PROCEDURE

USING RESTART ADDRESS 310 THE SWITCH REGISTER SETTINGS GIVEN PREVIOUSLY ARE USED (FOR BOTH MEMORY MANAGEMENT SELECTION AND DEVICE SELECTION).

## 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

NOTE: IF NO HARDWARE SWITCH REGISTER IS AVAILABLE, THE PROGRAM WILL AUTOMATICALLY USE THE CONTENTS OF LOC. 176 AS THE SOFTWARE SWITCH REG. THE USER SHOULD SET THIS LOCATION BEFORE STARTING THE PROGRAM.

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 TCCK (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 LP1 (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 MMOPT BITS 0, 1, AND 2 ARE ALL ZERO (=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 MMOPT BITS 0, 1, OR 2 IS UP AND SWS IS ZERO 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 MMOPT BITS 0 AND 1 WERE BOTH ZERO 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 BIT0 WAS ZERO BUT BIT1 WAS ONE, 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 BIT0 WAS ONE, ORDINARY CORE EXPANSION IS RUN WITH NO SPECIAL MAPPING REQUIRED (MEMORY MGMT. IS TURNED OFF).

### 5.3.3 PROCESSOR TEST EXECUTION - BANK 0 ONLY

IF BITS 0, 1 OR 2 IS UP AND BITS IS UP AT STARTUP, ONLY BANK 0 IS UTILIZED. IN THIS CASE, IF BIT0 AND BIT1 WERE ZERO THE PROCESSOR TESTS ARE EXECUTED IN USER, WITH USER PAGE 0 MAPPED TO BANK 0. IF BIT0 WAS ZERO AND BIT1 WAS ONE, THE PROCESSOR TESTS ARE EXECUTED IN KERNEL, WITH KERNEL PAGE 0 MAPPED TO BANK 0. IF BIT0 WAS ONE, THE MEMORY MGMT. IS TURNED OFF AND THE PROCESSOR TESTS ARE EXECUTED IN KERNEL MODE OR USER MODE (DEPENDING ON BIT1) 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.

A PASS RUN WITH CORE EXPANSION AND 4K AS 32K RELOCATION BOTH INHIBITED TAKES LESS THAN 10 SECONDS (RUNNING NO I/O).

A PASS RUN WITH 4K AS 32K, IN CORE MEMORY WITH NO I/O, TAKES ABOUT 5 MINUTES PER 4K BANK. (AN ASTERIK IS PRINTED AT THE END OF A FULL PASS, AND THE BELL IS RUNG AT THE END OF EACH 4K BANK.

ACT11 WITH OPTIONS SET AS DESCRIBED IN SECTION 8.3:

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 8.3:

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

## 8.2 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.

## 8.3 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 MMOPT 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 MEMORY MANAGEMENT SETUP CODE. THE RF11 AND TC11 PRIMER CODE IS IN WITH THE MEMORY MANAGEMENT SETUP CODE SINCE THEY REQUIRE CERTAIN PARTS OF THE MEMORY MANAGEMENT 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.

\*

558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
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

.LIST ME,SEQ,BIN  
.NLIST MC,CND,MD,TOC  
.TITLE DFKTGB MAINDEC-11-DFKTG-B MEMORY MGMT. EXERCISER  
.ABS  
.DSABL ERFZ

; THIS PROGRAM IS A MODIFICATION OF THE 11/40 DIAGNOSTIC, DBKTG.  
; THIS TEST HAS BEEN MODIFIED TO PROVIDE SOFTWARE SWITCH CAPABILITY  
; AND TO ACCOUNT FOR ANY 11/34-11/40 DIFFERENCES.  
; THIS PROGRAM IS INTENDED FOR USE ON ONLY 11/34 PROCESSORS  
; \*\*\*\*\*  
; SBTTL OPERATING INSTRUCTIONS  
; \*\*\*\*\*  
; PDP11/34 SYSTEM EXERCISER, WITH MEMORY MGMT. --- 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  
; MEMORY MANAGEMENT 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, MMOPT (LOC. 174) SETTINGS ARE:  
; MMOPT BIT 0=1 OR UP --- RUN WITHOUT MEMORY MGMT.  
; MMOPT BIT 1=1 OR UP --- RUN ALL IN KERNEL MODE (INHIBITS RUNNING 4K AS 32K)  
; MMOPT BIT 2=1 OR UP --- INHIBIT RUNNING 28K USER MEMORY MGMT. FROM EVERY 4K  
; ;BANK (ALLOW NORMAL CORE EXPANSION)  
; MMOPT BIT 5=1 OR UP---INHIBIT VARIABLE CORE EXPANSION  
  
; SR (USE LOC. 176 IF NECESSARY), BIT 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

OPERATING INSTRUCTIONS

612  
613  
614  
615 000240  
616 104400  
617 000410  
618 000412  
619 177776  
620 104006  
621 104010  
622 000000  
623 000001  
624 000002  
625 000003  
626 000004  
627 000005  
628 000006  
629 000006  
630 000007  
631

```

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

```

Address	Hex	Dec	Op	Comment
632				*****
633			SBTTL	TRAP CATCHER
634				*****
635		000000	.=0	
636	000000	000002	.+2	:TRAP ENTRANCE
637	000002	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
638	000004	000006	.+2	:TRAP ENTRANCE
639	000006	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
640	000010	000012	.+2	:TRAP ENTRANCE
641	000012	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
642	000014	000016	.+2	:TRAP ENTRANCE
643	000016	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
644	000020	000022	.+2	:TRAP ENTRANCE
645	000022	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
646	000024	000026	.+2	:TRAP ENTRANCE
647	000026	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
648	000030	000032	.+2	:TRAP ENTRANCE
649	000032	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
650	000034	000036	.+2	:TRAP ENTRANCE
651	000036	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
652	000040	000042	.+2	:TRAP ENTRANCE
653	000042	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
654	000044	000046	.+2	:TRAP ENTRANCE
655	000046	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
656	000050	000052	.+2	:TRAP ENTRANCE
657	000052	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
658	000054	000056	.+2	:TRAP ENTRANCE
659	000056	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
660	000060	000062	.+2	:TRAP ENTRANCE
661	000062	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
662	000064	000066	.+2	:TRAP ENTRANCE
663	000066	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
664	000070	000072	.+2	:TRAP ENTRANCE
665	000072	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
666	000074	000076	.+2	:TRAP ENTRANCE
667	000076	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
668	000100	000102	.+2	:TRAP ENTRANCE
669	000102	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
670	000104	000106	.+2	:TRAP ENTRANCE
671	000106	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
672	000110	000112	.+2	:TRAP ENTRANCE
673	000112	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
674	000114	000116	.+2	:TRAP ENTRANCE
675	000116	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
676	000120	000122	.+2	:TRAP ENTRANCE
677	000122	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
678	000124	000126	.+2	:TRAP ENTRANCE
679	000126	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
680	000130	000132	.+2	:TRAP ENTRANCE
681	000132	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
682	000134	000136	.+2	:TRAP ENTRANCE
683	000136	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
684	000140	000142	.+2	:TRAP ENTRANCE
685	000142	000000	HALT	:TRAPPED TO PREVIOUS LOCATION
686	000144	000146	.+2	:TRAP ENTRANCE
687	000146	000000	HALT	:TRAPPED TO PREVIOUS LOCATION

DFKTGB MAINDEC-11-DFKTG-B MEMORY MGMT. EXERCISER  
DFKTGB.P11 01-SEP-76 12:57 TRAP CATCHER

688	000150	000152	.+2	; TRAP ENTRANCE
689	000152	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
690	000154	000156	.+2	; TRAP ENTRANCE
691	000156	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
692	000160	000162	.+2	; TRAP ENTRANCE
693	000162	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
694	000164	000166	.+2	; TRAP ENTRANCE
695	000166	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
696	000170	000172	.+2	; TRAP ENTRANCE
697	000172	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
698	000174	000176	.+2	; TRAP ENTRANCE
699	000176	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
700	000200	000202	.+2	; TRAP ENTRANCE
701	000202	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
702	000204	000206	.+2	; TRAP ENTRANCE
703	000206	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
704	000210	000212	.+2	; TRAP ENTRANCE
705	000212	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
706	000214	000216	.+2	; TRAP ENTRANCE
707	000216	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
708	000220	000222	.+2	; TRAP ENTRANCE
709	000222	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
710	000224	000226	.+2	; TRAP ENTRANCE
711	000226	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
712	000230	000232	.+2	; TRAP ENTRANCE
713	000232	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
714	000234	000236	.+2	; TRAP ENTRANCE
715	000236	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
716	000240	000242	.+2	; TRAP ENTRANCE
717	000242	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
718	000244	000246	.+2	; TRAP ENTRANCE
719	000246	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
720	000250	000252	.+2	; TRAP ENTRANCE
721	000252	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
722	000254	000256	.+2	; TRAP ENTRANCE
723	000256	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
724	000260	000262	.+2	; TRAP ENTRANCE
725	000262	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
726	000264	000266	.+2	; TRAP ENTRANCE
727	000266	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
728	000270	000272	.+2	; TRAP ENTRANCE
729	000272	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
730	000274	000276	.+2	; TRAP ENTRANCE
731	000276	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
732	000300	000302	.+2	; TRAP ENTRANCE
733	000302	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
734	000304	000306	.+2	; TRAP ENTRANCE
735	000306	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
736	000310	000312	.+2	; TRAP ENTRANCE
737	000312	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
738	000314	000316	.+2	; TRAP ENTRANCE
739	000316	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
740	000320	000322	.+2	; TRAP ENTRANCE
741	000322	000000	HALT	; TRAPPED TO PREVIOUS LOCATION
742	000324	000326	.+2	; TRAP ENTRANCE
743	000326	000000	HALT	; TRAPPED TO PREVIOUS LOCATION

```

744 000330 000332      .+2      ; TRAP ENTRANCE
745 000332 000000      HALT      ; TRAPPED TO PREVIOUS LOCATION
746 000334 000336      .+2      ; TRAP ENTRANCE
747 000336 000000      HALT      ; TRAPPED TO PREVIOUS LOCATION
748 000340 000342      .+2      ; TRAP ENTRANCE
749 000342 000000      HALT      ; TRAPPED TO PREVIOUS LOCATION
750 000344 000346      .+2      ; TRAP ENTRANCE
751 000346 000000      HALT      ; TRAPPED TO PREVIOUS LOCATION
752 000350 000352      .+2      ; TRAP ENTRANCE
753 000352 000000      HALT      ; TRAPPED TO PREVIOUS LOCATION
754 000354 000356      .+2      ; TRAP ENTRANCE
755 000356 000000      HALT      ; TRAPPED TO PREVIOUS LOCATION
756 000360 000362      .+2      ; TRAP ENTRANCE
757 000362 000000      HALT      ; TRAPPED TO PREVIOUS LOCATION
758 000364 000366      .+2      ; TRAP ENTRANCE
759 000366 000000      HALT      ; TRAPPED TO PREVIOUS LOCATION
760 000370 000372      .+2      ; TRAP ENTRANCE
761 000372 000000      HALT      ; TRAPPED TO PREVIOUS LOCATION
762 000374 000376      .+2      ; TRAP ENTRANCE
763 000376 000000      HALT      ; TRAPPED TO PREVIOUS LOCATION
764
765
766
767
768
769 000024 016370      .=24      ; POWER FAIL TRAP
770 000026 000240      PFAIL      ;
771
772 000030 015072      .=30      ; EMT CALLS
773 000032 000340      EMTSRV     ; HIGHEST PRIORITY
774
775 000034 014570      .=34      ; USER TRAP
776 000036 000000      SCOPEC     ;
777
778 000040 000000      .=40      ; LOAD MEDIUM INDICATOR.
779 000042 000042      0          ; LOADS AS 0.
780 000042 000000      .=42      ; AUTOMATIC MODE INDICATOR.(ACT11/XXDP).
781 000046 000046      0          ; ZERO AT LOAD TIME.
782 000046 015640      $ENDAD     ; POINTER TO LOGICAL END.
783
784 000052 040000      .=52      ; PROGRAM ATTRIBUTES WORD.
785 000052 000174      40000     ;
786 000174 000000      .=174     ;
787 000176 000000      MMOPT: 0   ; MEMORY MANAGEMENT OPTION SEL.
788
789
790
791
792
793
794
795 000200 000137 000664      .=200     ;
796 000300 000137 000664      JMP      @#START
797 000300 000137 000664      .=300     ;
798 000310 000310 000634      JMP      @#START
799 000310 000137 000634      .=310     ;
800 000310 000137 000634      JMP      @#RSTRT

```

```

;*****
;SBTTL LOAD VECTOR AREA
;*****

```

```

;*****
;SBTTL LOAD STARTING AREA
;*****

```

```

MMOPT: 0
SWREG: 0
SREG2=SWREG

```

LOAD STARTING AREA

800  
801  
802  
803  
804 000400 000400  
805 000400 000000  
806 000406 000406  
807 000406 177560  
808 000410 177564  
809 000412 177566  
810 000414 000064  
811 000416 000066  
812 000420 000000  
813 000422 000100  
814 000424 000102  
815 000426 177546  
816 000430 177514  
817 000432 177516  
818 000434 000200  
819 000436 000202  
820 000440 177470  
821 000442 177466  
822 000444 177462  
823 000446 177464  
824 000450 177460  
825 000452 177461  
826 000454 000204  
827 000456 000206  
828 000460 177413  
829 000462 177412  
830 000464 177406  
831 000466 177410  
832 000470 177404  
833 000472 177405  
834 000474 000220  
835 000476 000222  
836 000500 177572  
837 000502 177600  
838 000504 177602  
839 000506 177616  
840 000510 177640  
841 000512 177642  
842 000514 177656  
843 000516 172300  
844 000520 172302  
845 000522 172304  
846 000524 172316  
847 000526 172340  
848 000530 172342  
849 000532 172344  
850 000534 172356  
851  
852 000536 177600  
853 000540 177640  
854 000542 172300  
855 000544 172340

```

;*****
;SBTTL DATA AREA
;*****
      . =400
UBUFF: 0
      . = +4
TRCSR: 177560
TTCSR: 177564
TTDBR: 177566
TTPVC: 64
TTPST: 66
TTSAV: 0
KWLVC: 100
KWLST: 102
LKCSR: 177546
LPCSR: 177514
LPDBR: 177516
LPVC: 200
LPST: 202
RFDAR: 177470
RFDAR: 177466
RFCAR: 177462
RFCAR: 177464
RFCAR: 177460
RFCARH: 177461
RFVC: 204
RFST: 206
RKDAH: 177413
RKDAE: 177412
RKWC: 177406
RKBAR: 177410
RKCSR: 177404
RKCSRH: 177405
RKVC: 220
RKST: 222
SR0: 177572
UPDR0: 177600
UPDR1: 177602
UPDR7: 177616
UPAR0: 177640
UPAR1: 177642
UPAR7: 177656
KPDR0: 172300
KPDR1: 172302
KPDR2: 172304
KPDR7: 172316
KPAR0: 172340
KPAR1: 172342
KPAR2: 172344
KPAR7: 172356

IPDRTAB: 177600
        177640
        172300
IPDREND: 172340

```

```

;BUFFER FOR USER SP
;FOR STACK OVERRUN
;TTY READER STATUS REGISTER
;TTY PUNCH STATUS REGISTER

```

```

;DISK ADDRESS AND ERROR
;DISK ADDRESS REGISTER
;WORD COUNT REGISTER
;CURRENT ADDRESS REGISTER
;STATUS REGISTER
;HIGH BYTE ADDRESS OR CSR

```

```

;HIGH BYTE DISK ADR
;DISK ADDRESS REGISTER
;WORD COUNT REGISTER
;CURRENT ADDRESS REGISTER
;STATUS REGISTER
;HIGH BYTE OF CSR
;TRAP VECTOR

```

```

;MEMORY MANAGEMENT REGISTERS

```

```

856 000546 177570 SR: 177570 ;SWITCH REGISTER POINTER
857 000550 177571 SRH: 177571 ;HIGH BYTE OF SW. REG. POINTER
858 000552 177342 TCCM: 177342 ;CONTROL AND FUNCTION
859 000554 177340 TCST: 177340 ;GENERAL STATUS
860 000556 177350 TCDT: 177350 ;DATA
861 000560 177344 TCWC: 177344 ;WORD COUNT
862 000562 177346 TCBA: 177346 ;BUS ADDRESS
863 000564 000214 TCIV: 214 ;DECTAPE INTERRUPT VECTOR
864 000566 000216 TCSTA: 216
865 000570 000000 CURBNK: 0 ;SAF TO POINT TO CURRENT BANK
866 000572 000000 OLDBNK: 0
867 000574 000000 CURPAR: 0 ;ADDRESS OF CURRENT ISAR
868 000576 000000 CURPDR: 0
869 000600 000000 BNKSTR: 0 ; PC TO POINT TO BEGIN THRU CURRENT SEGMENT
870 000602 000000 TRPB: 0
871 ;THE NEXT TWO WORDS ARE THE MEMORY MAP. THE FIRST WORD REPRESENTS
872 ;0-64K WITH ONE BIT REPRESENTING A 4K CONTIGUOUS BLOCK. IF THE
873 ;BIT=1 THAT 4K BLOCK IS PRESENT. THE LSB REPRESENTS 0-4K, THE NEXT
874 ;SIGNIFICANT BIT REPRESENTS 4-8K ANS SO ON.
875 000604 177777 MEMO: 177777 ;0-64K
876 000606 077777 MEM1: 77777 ;64-124K
877 000610 000001 COREPT: 1
878 000612 000604 MEMUT: MEMO
879 000614 000000 TBANK: 0
880 000616 000000 REFF: 0
881 000620 000000 TEST: 0
882
883 ;*****
884 ;SBTTL FILLCT, ACT11, XXDP SOFTWARE SWITCHES
885 ;*****
886 000622 000014 FILLCT: 14 ;CONSOLE FILL COUNT.
887 000624 000040 ACTSW1: 40 ;NO CORE EXPANSION.
888 000626 000201 ACTSW2: 201 ;NO LP, NO TTY.
889 000630 000046 XDPSW1: 46 ;NO CORE EXPANSION, NO 4 AS 32, ETC.
890 000632 000001 XDPSW2: 1 ;NO TTY.
891
892 ;*****
893 ;SBTTL RESTART ADD USING INITIAL SR SETTINGS
894 ;*****
895 000634 012706 017760 RSTRT: MOV #KSTACK,R6
896 000640 012737 016370 000024 MOV #PFAIL,@#24
897 000646 005737 000042 TST @#42 ;IN AUTO MODE? (ACT11/XXDP)
898 000652 001077 BNE START2 ;BR IF YES.
899 000654 117737 177670 000177 MOVB @SRH,@#SREG2+1 ;UPDATE DYNAMIC SWITCH SETTINGS.
900 000662 000473 BR START2
901
902 ;*****
903 ;SBTTL START UP FOR MINI MONITOR - NORMAL START FROM LOC 000200
904 ;*****
905
906 000664 012706 017760 START: MOV #KSTACK,R6 ;SET UP STACK
907 000670 012737 000137 000200 MOV #137,@#200 ;RESTORE 200 IF START AT 300
908 000676 012737 000664 000202 MOV #START,@#202
909 000704 005067 177672 CLR TRPB ;NO TRACE IN FIRST PASS.
910 000710 005067 013772 CLR PASCNT ;CLEAR THE PASS COUNTER.
911 000714 013746 000004 MOV @#4,-(SP) ;SAVE ERROR VECTOR

```



```

912 000720 013746 000006          MOV    @#6,-(SP)
913 000724 012767 000740 177052  MOV    #1$,4          ;SET UP TIME OUT VECTOR
914 000732 005777 177610          TST    @SR            ;TRY TO REFERENCE HARDWARE SW. REG.
915 000736 000404          BR     2$            ;BRANCH IF NO TIMEOUT TRAP OCCURS
916 000740 012767 000176 177600 1$:  MOV    #SWREG,SR     ;POINT TO SOFTWARE SW. REG.
917 000746 022626          CMP    (SP)+,(SP)+  ;RESTORE STACK
918 000750 016767 177572 177572 2$:  MOV    SR,SRH
919 000756 005267 177566          INC    SRH
920 000762 012637 000006          MOV    (SP)+,@#6    ;RESTORE ERROR VECTOR
921 000766 012637 000004          MOV    (SP)+,@#4
922 000772 005737 000042          TST    @#42
923 000776 001422          BEQ   STARTX        ;IN AUTOMATIC TEST MODE?
924 001000 023727 000042 015640  CMP    @#42,#SENDAD ;BR IF NOT IN AUTOMATIC MODE.
925 001006 001007          BNE   3$            ;IN ACT11 MODE?
926 001010 016737 177610 000174  MOV    ACTSW1,@#MMOPT ;BR IF NOT.
927 001016 016737 177604 000176  MOV    ACTSW2,@#SREG2 ;YES. SET MMOPT FROM ACTSW1.
928 001024 000412          BR     START1        ;SET SREG2 FROM ACTSW2.
929 001026 016737 177576 000174 3$:  MOV    XDPSW1,@#MMOPT ;XXDP MODE. SET MMOPT FROM XDPSW1.
930 001034 016737 177572 000176  MOV    XDPSW2,@#SREG2 ;SET SREG2 FROM XDPSW2.
931 001042 000403          BR     START1
932 001044          STARTX:
933 001044 017737 177476 000176  MOV    @SR,@#SREG2
934 001052          START1:
935 001052 004767 013752          START2: JSR    %7,NRALL
936 001056 012777 077406 177432  MOV    #77406,@KPDRO
937 001064 012777 007600 177442  MOV    #7600,@KPAR7   ;MAP PAGE 7 TO EXT BANK
938 001072 012777 077406 177424  MOV    #77406,@KPD7
939 001100 005067 177510          CLR    TBANK
940 001104 012767 177777 177472  MOV    #177777,MEMO   ;SET UP CORE MAPS
941 001112 012767 077777 177466  MOV    #77777,MEM1
942 001120 012767 000001 177462  MOV    #1,COREPT      ;SET UP 4K POINTER
943 001126 012767 000604 177456  MOV    #MEMO,MEMUT
944 001134 012777 077406 177360  MOV    #77406,@KPD2   ;BEING CHECKED FOR
945 001142 012737 001212 000004  MOV    #TMEMEX,@#4   ;SET UP FOR TIME OUTS
946 001150 005037 000006          CLR    @#6
947 001154 052777 000001 177316  BIS    #1,@SR0
948 001162 016777 177426 177342  MAP1:  MOV    TBANK,@KPAR2  ;MAP KERNEL PAGE 2 TO BANK
949 001170 005737 041000          TST    @#41000       ;1ST K PRESENT
950 001174 005737 045000          TST    @#45000       ;2ND K PRESENT
951 001200 005737 051000          TST    @#51000       ;3RD K PRESENT
952 001204 005737 055000          TST    @#55000       ;4TH K PRESENT
953 001210 000404          BR     MOVEPT        ;OK, FULL 4K BLOCK PRESENT
954 001212 046777 177372 177372  TMEMEX: BIC    COREPT,@MEMUT ;NO, BLOCK NOT PRESENT
955 001220 022626          CMP    (SP)+,(SP)+  ;ADJUST STACK POINTER
956 001222 062767 000200 177364  MOVEPT: ADD    #200,TBANK   ;UPDATE BANK POINTER
957 001230 006367 177354          ASL    COREPT
958 001234 103006          BCC   MAP2          ;THIS 1ST MEM WORD DONE
959 001236 012767 000001 177344  MOV    #1,COREPT
960 001244 012767 000606 177340  MOV    #MEM1,MEMUT
961 001252 022767 007600 177334  MAP2:  CMP    #7600,TBANK   ;EXTERNAL BANK YET
962 001260 001340          BNE   MAP1          ;NO, NOT YET?
963 001262 012767 000001 177320  MOV    #1,COREPT     ;RE-INIT
964 001270 012767 000604 177314  MOV    #MEMO,MEMUT
965 001276 042777 000001 177174  BIC    #1,@SR0
966 001304 012737 014570 000034  MOV    #SCOPEC,@#34
967 001312 005037 000036          CLR    @#36          ;INITIALIZE SCOPE CALL TO KERNEL STATUS

```

M02

DFKTGB MAINDEC-11-DFKTG-B MEMORY MGMT. EXERCISER  
 DFKTGB.P11 01-SEP-76 12:57

MACY11 27(1006) 29-OCT-76 15:23 PAGE 24  
 START UP FOR MINI MONITOR - NORMAL START FROM LOC 000200

968	001316	012737	015072	000030	MOV	#EMTSRV, @#30	
969	001324	012737	000340	000032	MOV	#340, @#32	
970	001332	012737	005542	014704	MOV	#BEGIN, @#RETURN	
971	001340	012737	000340	177776	MOV	#340, @#PSR	; LOCK OUT INTERRUPTS
972	001346	005037	016060		CLR	@#PRTON	; PRINT ROUTINE BUSY FLAG
973	001352	000005			RESET		
974	001354	012737	002404	000004	MOV	#NODEV, @#4	; RETURN FOR NO DEVICE
975	001362	005037	000006		CLR	@#6	
976					;*****		
977					.SBTTL TTY INIT		
978					;*****		
979	001366	005067	001464		CLR	DATA2	; BASE DATA FOR TTY TELEPRINTER
980	001372	033727	000176	000001	BIT	@#SREG2, #1	; INHIBIT TTY OUTPUT?
981	001400	001006			BNE	ST3	; YES, GO CHECK NEXT
982	001402	012777	003070	177004	MOV	#TYOUTR, @#TTPVC	; NO, SETUP INTERRUPT VECTOR
983	001410	052777	000100	176772	BIS	#100, @#TICSR	; START TTY OUTPUT
984					;*****		
985					.SBTTL RK11 INIT		
986					;*****		
987	001416	012700	000010		ST3:	MOV	#10, R0
988	001422	122737	000002	000041	CMPB	#2, @#41	; LOAD MEDIUM RK11?
989	001430	001432			BEQ	ST4	; BR IF YES. DON'T USE RK11 THEN.
990	001432	032737	000010	000176	BIT	#10, @#SREG2	; INHIBIT RK DISK
991	001440	001026			BNE	ST4	; YES, SKIP OVER
992	001442	005777	177022		TST	@#RKCSR	; PRESENT
993	001446	012777	003466	177020	MOV	#IRK, @#RKVC	; SETUP VECTOR RETURNS
994	001454	012777	000240	177014	MOV	#240, @#RKST	; PRIORITY 5 SERVICE.
995	001462	012767	043503	002040	MOV	#43503, @#RKFUNCT	
996	001470	005077	176766		CLR	@#RKDAE	; INIT
997	001474	016777	002170	176764	MOV	LLIMIT, @#RKBAR	; CORE BASE
998	001502	016777	002164	176754	MOV	WORDCT, @#RKWC	; TRANSFER LENGTH
999	001510	116777	002014	176752	MOVB	RKFUNCT, @#RKCSR	
1000					;*****		
1001					.SBTTL LINE CLOCK INIT		
1002					;*****		
1003	001516	006300			ST4:	ASL	R0
1004	001520	033727	000176	000020	BIT	@#SREG2, #20	; INHIBIT LINE CLOCK?
1005	001526	001015			BNE	ST5	; YES, GO CK NEXT
1006	001530	005777	176672		TST	@#LKCSR	; PRESENT
1007	001534	012777	003146	176660	MOV	#LK3, @#KWLVC	
1008	001542	012777	000300	176654	MOV	#300, @#KWLST	
1009	001550	005067	001466		CLR	TIME	; NO, INITIALIZE COUNT
1010	001554	052777	000100	176644	BIS	#100, @#LKCSR	; START LINE CLOCK
1011					;*****		
1012					.SBTTL RF11 INIT		
1013					;*****		
1014	001562	006300			ST5:	ASL	R0
1015	001564	033727	000176	000040	BIT	@#SREG2, #40	; TEST FOR INHIBITING RF11 DISK
1016	001572	001026			BNE	ST6	; SKIP IF SET
1017	001574	005777	176650		TST	@#RFCSR	; PRESENT?
1018	001600	012777	003562	176646	MOV	#IRF, @#RFVC	; SET UP TRAP RETURN
1019	001606	012777	000240	176642	MOV	#240, @#RFST	
1020	001614	012767	043503	002044	MOV	#43503, @#RFFUNCT	; WRITE CHECK/WRITE
1021	001622	105277	176624		INCB	@#RFCSR	; INITIALIZE DISK-DAR, DAE
1022	001626	016777	002040	176610	MOV	WORDCT, @#RFWC	; LENGTH OF TRANSFER
1023	001634	016777	002030	176604	MOV	LLIMIT, @#RFCAR	; CORE ADDRESS OF START OF TRANSFER

```

1024 001642 116777 002020 176600      MOVB   RFFUNCT,2RFCSR      ;START RF11 READ OR WRITE
1025                                     ;*****
1026                                     ;SBTTL TC11 INIT
1027                                     ;*****
1028 001650 006300                                     ST6:  ASL   R0
1029 001652 122737 000001 000041      CMPB   #1,2#41           ;LOAD MEDIUM DECTAPE?
1030 001660 001417                                     BEQ   ST7               ;BR IF YES. DON'T USE IT THEN.
1031 001662 033727 000176 000100      BIT    2#SREG2,#100     ;CHECK FOR INHIBITING TC11 DECTAPE
1032 001670 001013                                     BNE   ST7               ;SKIP IF SET
1033 001672 005777 176656                                     TST   2TCST            ;PRESENT?
1034 001676 012777 003702 176660      MOV    #FENDZ,2TCIV     ;GO TO END ZONE ON INTERRUPT
1035 001704 012777 000300 176654      MOV    #300,2TCSTA
1036 001712 012777 004503 176632      MOV    #R+IE+RB+DO,2TCCM ;START REVERSE READ BLOCK NUMBER
1037                                     ;*****
1038                                     ;SBTTL LINE PRINTER INIT
1039                                     ;*****
1040 001720 006300                                     ST7:  ASL   R0
1041 001722 033727 000176 000200      BIT    2#SREG2,#200     ;INHIBIT LINE PRINTER?
1042 001730 001032                                     BNE   ST8               ;YES GO CK NEXT
1043 001732 005777 176472                                     TST   2LPCSR           ;PRESENT?
1044 001736 012737 002016 000004      MOV    #ST8,2#4         ;DON'T CHANGE 200 IF NO SUCH DEVICE
1045 001744 012767 000137 001274      MOV    #137,SOLPAT      ;RESET FOR START OF LINE PATTERN
1046 001752 012767 000117 001360      MOV    #79,CLINCT       ;LINE COUNT
1047 001760 012767 000137 001262      MOV    #137,CURPAT
1048 001766 012777 000014 176436      MOV    #14,2LPDR        ;LINE FEED TO POSITION BUFFER
1049 001774 012777 003270 176432      MOV    #LPINTR,2LPVC    ;INTERRUPT ENABLE
1050 002002 012777 000200 176426      MOV    #200,2LPST       ;PROCESSOR LEVEL 4
1051 002010 012777 000100 176412      MOV    #100,2LPCSR     ;INTERRUPT ENABLE
1052                                     ;*****
1053                                     ;SBTTL PRE-PASS SETUP
1054                                     ;*****
1055 002016 005037 000006                                     ST8:  CLR   2#6           ;CHANGE ADDRESS ERROR VECTOR TO CAUSE
1056 002022 012737 000006 000004      MOV    #6,2#4           ;HALT ON A TRAP TO 4
1057 002030 004767 000370                                     JSR   %7,DET1           ;CHECK FOR CORE EXPANSION
1058 002034 032737 000001 000174      BIT    #1,2#MMOPT       ;INHIBIT MEMORY MGMT?
1059 002042 001106                                     BNE   MODE              ;YES - GO SETUP USER
1060 002044 004767 012760                                     JSR   %7,NRALL          ;NO - MAKE ALL SEGMENTS INITIALLY NON-RESIDENT
1061 002050 012777 077406 176446      MOV    #77406,2KPDR7
1062 002056 012777 007600 176450      MOV    #7600,2KPAR7
1063 002064 032737 000006 000174      BIT    #6,2#MMOPT       ;INHIBIT USER/KERNEL OR 4K AS 32K?
1064 002072 001415                                     BEQ   SEGM1             ;NO - BRANCH
1065 002074 012701 000007                                     MOV    #7,R1            ;YES - MAP KERNEL ASR'S 0-6 TO PA
1066 002100 016702 176422                                     MOV    KPAR0,R2
1067 002104 005003                                     CLR   R3
1068 002106 010312                                     SETEX: MOV  R3,2R2
1069 002110 012762 077406 177740      MOV    #77406,-40(R2)
1070 002116 005722                                     TST   (R2)+
1071 002120 062703 000200                                     ADD   #200,R3
1072 002124 077110                                     SOB   R1,SETEX
1073 002126 012777 077406 176362      SEGM1: MOV  #77406,2KPDR0 ;MAP KERNEL 0 TO BANK 0, RW
1074 002134 032737 000004 000174      BIT    #4,2#MMOPT       ;INHIBIT RUNNING 4K AS 32K?
1075 002142 001416                                     BEQ   USEALL            ;NO, SETUP FOR RUNNING 4K AS 32K
1076 002144 012701 000010                                     MOV    #10,R1           ;YES, MAP ALL USER ASR'S TO PA
1077 002150 016702 176334                                     MOV    UPAR0,R2
1078 002154 005003                                     CLR   R3
1079 002156 010312      SETUSE: MOV  R3,(R2)

```



```

1136 002456 012737 002542 000004 DET4:  MOV    #DET2, R#4      ;TRAP VECTOR SETUP
1137 002464 012737 000340 000006      MOV    #340, R#6      ;TRAP STATUS SETUP
1138 002472 000241      CLC
1139 002474 005537 037770      EIGHT:  ADC    R#37770      ;CHECK FOR 8K
1140 002500 000240      NOP
1141 002502 005537 057770      ADC    R#57770      ;CHECK FOR 12K
1142 002506 000240      NOP
1143 002510 005537 077770      ADC    R#077770     ;CHECK FOR 16K
1144 002514 000240      NOP
1145 002516 005537 117770      ADC    R#117770     ;CHECK FOR 20K
1146 002522 000240      NOP
1147 002524 005537 137770      ADC    R#137770     ;CHECK FOR 24K
1148 002530 000240      NOP
1149 002532 005537 157770      ADC    R#157770     ;CHECK FOR 28K
1150 002536 000240      NOP
1151 002540 000437      BR     STRT28
1152 002542 012602      DET2:  MOV    (R#)+, R#2    ;RETRIEVE TRAP PC
1153 002544 005726      TST    (R#)+        ;DISCARD TRAP STATUS WORD
1154 002546 062702 000074      ADD    #STRT4-EIGHT-4, R2
1155 002552 000112      JMP    R2
1156
1157 002554 005000      MOVE:  CLR    R#0      ;SET UP MAIN CORE POINTER
1158 002556 010102      MOV    R#1, R#2
1159 002560 062702 015030      ADD    #0+2, R#2    ;SET UP MAX CORE MOVE
1160 002564 012021      MOV    (R#0)+, (R#1)+ ;MOVE WORD
1161 002566 020201      CMP    R#2, R#1     ;MOVE COMPLETE?
1162 002570 001375      BNE    R#-4         ;MOVE ANOTHER WORD
1163 002572 000207      RTS    R#7         ;MOVE COMPLETE
1164 002574 000521      STRT4:  BR     DET3
1165 002576 000240      NOP
1166 002600 000240      NOP
1167 002602 004767 000110      JSR    R#7, XFER8   ;START 8K TRANSFER
1168 002606 000506      BR     MOD4         ;START 4K MODIFY
1169 002610 004767 000072      JSR    R#7, XFER12  ;START 12K TRANSFER
1170 002614 000475      BR     MOD8         ;START 8K MODIFY
1171 002616 004767 000054      JSR    R#7, XFER16  ;START 16K TRANSFER
1172 002622 000464      BR     MOD12        ;START 12K MODIFY
1173 002624 004767 000036      JSR    R#7, XFER20  ;START 20K TRANSFER
1174 002630 000453      BR     MOD16        ;START 16K MODIFY
1175 002632 004767 000020      JSR    R#7, XFER24  ;START 24K TRANSFER
1176 002636 000442      BR     MOD20        ;START 20K MODIFY
1177 002640 004767 000002      STRT28: JSR    R#7, XFER28 ;START 28K TRANSFER
1178 002644 000431      BR     MOD24        ;START 24K MODIFY
1179 002646 012701 140000      XFER28: MOV    #140000, R#1 ;SET UP MOVE START LOCATION
1180 002652 004767 177676      JSR    R#7, MOVE    ;GO TO MOVE SUBROUTINE
1181 002656 012701 120000      XFER24: MOV    #120000, R#1
1182 002662 004767 177666      JSR    R#7, MOVE
1183 002666 012701 100000      XFER20: MOV    #100000, R#1
1184 002672 004767 177656      JSR    R#7, MOVE
1185 002676 012701 060000      XFER16: MOV    #60000, R#1
1186 002702 004767 177646      JSR    R#7, MOVE
1187 002706 012701 040000      XFER12: MOV    #40000, R#1
1188 002712 004767 177636      JSR    R#7, MOVE
1189 002716 012701 020000      XFER8:  MOV    #20000, R#1
1190 002722 004767 177626      JSR    R#7, MOVE
1191 002726 000207      RTS    R#7         ;RETURN FROM TRANSFERS
    
```

1192 002730 012767 000137 131570  
1193 002736 012767 145510 131564  
1194 002744 012767 000137 111554  
1195 002752 012767 125510 111550  
1196 002760 012767 000137 071540  
1197 002766 012767 105510 071534  
1198 002774 012767 000137 051524  
1199 003002 012767 065510 051520  
1200 003010 012767 000137 031510  
1201 003016 012767 045510 031504  
1202 003024 012767 000137 011474  
1203 003032 012767 065510 011470  
1204 003040 005031 000006  
1205 003044 012737 000006 000004  
1206 003052 000207  
1207  
1208  
1209  
1210  
1211 003054 005027 000000  
1212 003056  
1213 003060 016777 177772 175324  
1214 003066 000002  
1215 003070 017767 175314 175322  
1216 003076 105767 175316  
1217 003102 100401  
1218 003104 104006  
1219 003106 005267 177744  
1220 003112 022767 000400 177736  
1221 003120 001755  
1222 003122 000756  
1223  
1224  
1225  
1226  
1227 003124 005037 003242  
1228 003130 052777 000100 175270  
1229 003136 052737 000100 177776  
1230 003144 000002  
1231 003146 105777 175254  
1232 003152 100401  
1233 003154 104006  
1234 003156 042777 000200 175242  
1235 003164 005237 003242  
1236 003170 022737 006344 003242  
1237 003176 103362  
1238 003200 042777 000100 175220  
1239 003206 042737 000100 177776  
1240 003214 022737 007020 003242  
1241 003222 001740  
1242 003224 105777 175176  
1243 003230 100375  
1244 003232 042777 000200 175166  
1245 003240 000751  
1246 003242 000000  
1247

MOD24: MOV #137,DONE+120000  
MOV #BEGINX+140000,DONE+120002  
MOD20: MOV #137,DONE+100000  
MOV #BEGINX+120000,DONE+100002  
MOD16: MOV #137,DONE+60000  
MOV #BEGINX+100000,DONE+60002  
MOD12: MOV #137,DONE+40000  
MOV #BEGINX+60000,DONE+40002  
MOD8: MOV #137,DONE+20000  
MOV #BEGINX+40000,DONE+20002  
MOD4: MOV #137,DONE  
MOV #BEGINX+20000,DONE+2  
DET3: CLR #6  
MOV #6,#4  
RTS %7

\*\*\*\*\*  
:SBTTL TTY TRANSMITTER PRINT VALUES 0 TO 377  
\*\*\*\*\*

TYOUT: CLR #0 ;INITAL DATA  
DATA2=-2  
TYOUT1: MOV DATA2,#TTDBR ;OUTPUT TO DEVICE  
RTI ;RETURN TO MAINLINE\*\*  
TYOUTR: MOV #TTCSR,TTSV  
TSTB TTSV ;TEST FOR DONE  
BMI .+4 ;BRANCH IF FLAG FOUND  
HLT ;FALSE INTERRUPT RETURN  
INC DATA2 ;INCREMENT DATA  
CMP #400,DATA2 ;TEST DATA FOR UPPER LIMIT  
BEQ TYOUT ;AT UPPER LIMIT START OVER  
BR TYOUT1 ;FINISH REST OF DATA

\*\*\*\*\*  
:SBTTL TEST OF LINE CLOCK, INTERRUPT FOR 55 SECONDS THEN STALL FOR 5 SECONDS.  
\*\*\*\*\*

LK1: CLR #TIME ;CLEAR LINE CLOCK TIMER  
BIS #100,#LKCSR  
BIS #100,#PSR  
LK2: RTI  
LK3: TSTB #LKCSR  
BMI .+4  
HLT ;FALSE INTERRUPT  
LK4: BIC #200,#LKCSR  
INC #TIME ;HERE ON INTERRUPTS  
CMP #3300.,#TIME ;55 SEC YET?  
BHS LK2 ;BR IF NOT  
BIC #100,#LKCSR  
BIC #100,#PSR ;LOWER PRIORITY  
CMP #3600.,#TIME ;ONE MINUTE YET  
BEQ LK1 ;YES RESET TIMER  
TSTB #LKCSR ;NO, SKIP TILL MINUTE UP  
BPL .-4  
BIC #200,#LKCSR ;CLEAR FLAG  
BR LK4  
TIME: 0

E03

DFKTGB MAINDEC-11-DFKTG-B MEMORY MGMT. EXERCISER  
DFKTGB.P11 01-SEP-76 12:57

MACY11 27(1006) 29-OCT-76 15:23 PAGE 29  
TEST OF LINE CLOCK, INTERRUPT FOR 55 SECONDS THEN STALL FOR 5 SECONDS.

```

1248 ;*****
1249 .SBTTL LINE PRINTER SERVICE
1250 ;*****
1251 ;LINE PRINTER SHOULD RAISE PROCESSOR PRIORITY TO LEVEL OF LINE PRINTER/
1252 ;INTERRUPT VECTOR IS 200/
1253 003244 012727 000000 000000 LP1: MOV #0,#0 ;START OF LINE TO CURRENT
1254 003250 ;CHARACTER BEING PRINTED
1255 003246 SOLPAT=-4 ;START OF LINE CHARACTER
1256 003252 016777 177772 175152 LP2: MOV CURPAT, @LPDDBR ;CURRENT PATTERN TO LINE PRINTER
1257 003260 105777 175144 TSTB @LPCSR
1258 003264 100420 BMI LP6
1259 003266 000002 RTI ;RETURN TO MAIN LINE
1260 003270 105777 175134 LPINTR: TSTB @LPCSR ;TEST FOR FLAG
1261 003274 100414 BMI LP6
1262 003276 005737 000042 TST @#42 ;MONITOR LOAD
1263 003302 001410 BEQ LP7 ;NO, ERROR
1264 003304 032777 100000 175116 BIT #100000, @LPCSR ;YES, IS ERROR SET
1265 003312 001404 BEQ LP7 ;NO, ERROR
1266 003314 042777 000100 175106 BIC #100, @LPCSR ;DIS ABLE INTERRUPT
1267 003322 000002 RTI
1268 003324 104006 HLT ;FALSE RETURN FROM MAIN LINE
1269 003326 026727 000006 000117 LP6: CMP CLINCT, #79. ;TEST FOR END OF LINE
1270 003334 001415 BEQ LP4 ;GO GENERATE CR/LF
1271 003336 005227 000000 INC #0 ;INCREMENT LINE POSITION COUNT
1272 003340 CLINCT=-2 ;POSITION OF LINE
1273 003342 026727 177702 000137 CMP CURPAT, #137 ;TEST FOR MAXIMUM PATTERN
1274 003350 001403 BEQ LP3 ;YES - GO TO LP3 AND RESET
1275 003352 005267 177672 INC CURPAT ;NO - INCREMENT TO NEXT PATTERN
1276 003356 000735 BR LP2 ;GO SEND IT TO LINE PRINTER
1277 003360 012767 000040 177662 LP3: MOV #40, CURPAT ;RESET PATTERN AND SEND TO PRINTER
1278 003366 000731 BR LP2 ;SENT TO LINE PRINTER
1279 003370 005067 177744 LP4: CLR CLINCT ;RESET LINE COUNT
1280 003374 012777 000012 175030 MOV #12, @LPDDBR ;LINE FEED
1281 003402 105777 175022 TSTB @LPCSR
1282 003406 100375 BPL -4
1283 003410 026727 177632 000137 CMP SOLPAT, #137 ;START OF LINE PATTERN
1284 003416 001403 BEQ LP5
1285 003420 005267 177622 INC SOLPAT ;INCREMENT START OF LINE
1286 003424 000707 BR LP1
1287 003426 012767 000040 177612 LP5: MOV #40, SOLPAT ;RESET START OF LINE
1288 003434 000703 BR LP1 ;PRINT
1289
1290 ;*****
1291 .SBTTL RK11 SERVICE
1292 ;*****
1293 ;RK11 DISK TEST INTERRUPT LEVEL 5, 2000 WORD TRANSFERS
1294 003436 005077 175020 RKSTART: CLR @RKDAE ;INIT
1295 003442 013777 003670 175016 RKI: MOV @LLIMIT, @RKBAR ;CORE BASE
1296 003450 013777 003672 175006 MOV @WORDCT, @RKWC ;TRANSFER LENGTH
1297 003456 113777 003530 175004 MOVB @RKFUNCT, @RKCSR ;WRITE OR WRITE CK TO DSK
1298 003464 000002 RTI ;RETURN TO MAINLINE
1299 003466 032777 100200 174774 IRK: BIT #100200, @RKCSR ;INTERRUPT RETURN
1300 003474 003002 BGT +6
1301 003476 104006 HLT
1302 003500 000756 BR RKSTART
1303 003502 032777 000037 174752 BIT #37, @RKDAE ;DISK AT UPPER LIMIT?

```

1304 003510 001354  
 1305 003512 122777 000031 174740  
 1306 003520 001350  
 1307 003522 000337 003530  
 1308 003526 000743  
 1309 003530 000000  
 1310  
 1311  
 1312  
 1313  
 1314 003532 105277 174714  
 1315 003536 013777 003670 174702  
 1316 003544 013777 003672 174672  
 1317 003552 113777 003666 174670  
 1318 003560 000002  
 1319 003562 105777 174662  
 1320 003566 100402  
 1321 003570 104006  
 1322 003572 000757  
 1323 003574 005777 174650  
 1324 003600 100012  
 1325 003602 032777 020000 174640  
 1326 003610 001404  
 1327 003612 104006  
 1328 003614 000337 003666  
 1329 003620 000744  
 1330 003622 104006  
 1331 003624 000742  
 1332 003626 005777 174612  
 1333 003632 100002  
 1334 003634 104006  
 1335 003636 000735  
 1336 003640 122777 000003 174572  
 1337 003646 001333  
 1338 003650 027727 174566 174000  
 1339 003656 101727  
 1340 003660 000337 003666  
 1341 003664 000722  
 1342 003666 000000  
 1343 003670 005542  
 1344 003672 176000

BNE RK1  
 CMPB #31, JRKDAH  
 BNE RK1  
 SWAB @RKFUNCT ;CHANGE COMMAND  
 BR RKSTART ;RESTART NEW TRANSFER OF DISK  
 RKFUNCT: 0  
 \*\*\*\*\*  
 .SBTTL RF11 DISK  
 \*\*\*\*\*  
 RFSTART: INCB @RFCRSH ;INITIALIZE DISK - DAR-DAE  
 RF1: MOV @LLIMIT, @RFCAR ;CORE BASE  
 MOV @WORDCT, @RFWC ;LENGTH OF TRANSFER  
 MOVB @RFFUNCT, @RFCRSH ;WRITE OR WRITE CHECK TO DISK  
 RTI ;RETURN TO MAINLINE CODE  
 IRF: TSTB @RFCRSH ;INTERRUPT VECTOR POINTS HERE  
 BMI .+6  
 HLT ;RF11 READY NOT UP  
 BR RFSTART  
 TST @RFCRSH ;ERROR SET?  
 BPL ERROK ;BRANCH IF NOT  
 BIT #20000, @RFCRSH ;YES-WRITE CHECK ERROR?  
 BEQ ERRSET ;NO-BRANCH  
 HLT ;YES-RF11 WRITE CHECK ERROR  
 SWAB @RFFUNCT ;CHANGE COMMAND TO DO WRITE  
 BR RFSTART  
 ERRSET: HLT ;RF11 ERROR SET-NOT WRITE CHECK  
 BR RFSTART  
 ERROK: TST @RFWC  
 BPL .+6  
 HLT ;RF-11 WORD COUNT NOT ZERO  
 BR RFSTART  
 CMPB #3, @RFDAR ;DISK AT UPPER LIMIT? 7=2, 17=4, 37=8  
 BNE RF1 ;NO  
 CMP @RFDAR, #174000 ;AS FAR ON DISK AS WE CAN GO  
 BLOS RF1 ;NO  
 SWAB @RFFUNCT ;CHANGE COMMAND  
 BR RFSTART ;RESTART NEW TRANSFER OF DISK  
 RFFUNCT: 0 ;DISK COMMAND  
 LLIMIT: BEGIN ;FIRST CORE ADDRESS OF TRANSFER  
 WORDCT: -2000 ;LENGTH OF TRANSFER

1345  
 1346  
 1347  
 1348  
 1349  
 1350  
 1351  
 1352  
 1353  
 1354  
 1355  
 1356  
 1357  
 1358  
 1359

\*\*\*\*\*  
 .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



```

1360 ;SCRAMBLED TO GUARANTEE THAT NEW DATA IS REALLY READ IN NEXT TIME. WHILE
1361 ;THIS IS GOING ON, BLOCK NUMBERS ARE CHECKED FOR BEING IN ORDER AS THE
1362 ;TAPE TRAVELS TOWARD THE FORWARD END ZONE.ONCE THE DATA IS FULLY CHECKED
1363 ;THE NEXT BLOCK THAT COMES UP IS READ IN AND THE PROCESS REPEATED. ONCE
1364 ;THE BLOCK WHOSE NUMBER IS IN TCFRST HAS BEEN READ, THE TAPE IS DRIVEN
1365 ;INTO THE FORWARD END ZONE AND THE WHOLE SEQUENCE IS REPEATED.
1366
1367 ;FUNCTION VALUES IN CSR
1368 ;DT11 DEC TAPE
1369 RD=4 ;READ DATA
1370 WD=14 ;WRITE DATA
1371 RB=2
1372 IE=500 ;INTERRUPT ENABLE+UNIT 1
1373 DO=1 ;DO - THE FUNCTION
1374 R=4000 ;REVERSE
1375
1376 TCFRST: 0 ;FIRST BLOCK TO BE SEARCHED FOR
1377 TCLAST: 577. ;LAST BLOCK TO BE SEARCHED FOR
1378 TCEXPE: 0 ;THE BLOCK THAT IS EXPECTED
1379
1380 ;GO TO FORWARD END ZONE
1381 FENDZ: MOV #FENDZ,@TCIV ;END ZONE VECTOR SETUP
1382 TST @TCST ;TEST FOR END ZONE
1383 BMI FEND1 ;AT END ZONE?
1384 INCB @TCCM ;SET DO - NO DELAY
1385 RTI ;NO - WAIT SOME MORE
1386 FEND1: MOV #TCF1,@TCIV ;YES - NEW VECTOR
1387 BIC #104000,@TCCM ;SEARCH BLOCK FOWARD
1388 MOV TCFRST,TCEXPE ;COUNT WHEN THIS BLOCK IS FOUND
1389 TCF1A: INCB @TCCM ;SET DO
1390 RTI ;RETURN ON NEXT BLOCK
1391 TCF1: BIT #100200,@TCCM ;ANY ERROR ON READ?
1392 BPL .+4
1393 HLT ;TC ERROR SET - FORWARD READ BLOCK
1394 BNE .+4 ;DONE FLAG UP?
1395 HLT ;FALSE INTERRUPT
1396 CMP @TCDT,TCEXPE ;IS THIS OUR BLOCK FOR SYNC
1397 BLT TCF1A ;NO-READ SOME MORE BLOCKS
1398 BEQ TCF2 ;YES
1399 HLT ;WE PASSED THE BLOCK
1400
1401 TCF2: MOV #TCF3,@TCIV ;VECTOR FOR SEQUENTIAL READS
1402 INCB @TCCM ;SET DO
1403 RTI ;RETURN AND TEST SEQENTIAL BLOCKS
1404
1405 ;FIND SEQUENTIAL BLOCK AT FOWARD DIRECTION
1406 TCF3: BIT #100200,@TCCM ;TEST ERROR AND READY
1407 BPL .+4
1408 HLT ;FOWARD READ ERROR TC-11
1409 BNE .+4
1410 HLT ;FALSE INTERRUPT ON TC-11
1411 CMP @TCDT,TCLAST ;HAVE WE TESTED ALL BLOCKS
1412 BEQ RENDZ ;YES DRIVE UNIT IN END ZONE TO START OVER
1413 INC TCEXPE ;NO-INCREMENT EXPECTED COUNT
1414 CMP @TCDT,TCEXPE ;IS CURRENT BLOCK CORRECT
1415 BEQ .+4

```

```

1416 004064 104006          HLT          ;FAILED IN FOWARD READ TO FIND NEXT BLOCK
1417 004066 000427          BR          TCWBK      ;THIS ROUTINE WRITES A BLOCK
1418 004070 105277 174456   TCF4:      INCB      @TCCM    ;SET DO
1419 004074 000002          RTI
1420 004076 000701          XFENDZ:    BR          FENDZ    ;INDIRECT LINK
1421
1422          ;MOVE TAPE TO REVERSE END ZONE
1423 004100 012777 004100 174456   RENDZ:    MOV        #RENDZ,@TCIV ;END ZONE VECTOR SETUP
1424 004106 016767 177564 177564   MOV        TCLAST,TCEXPE ;SET UP FOR REVERSE SEARCH
1425 004114 005777 174434          TST        @TCST      ;IN END ZONE
1426 004120 100403          BMI        REND1     ;YES - START TO TURN UNIT AROUND
1427 004122 105277 174424          INCB      @TCCM      ;SET DO
1428 004126 000002          RTI          ;NO - WAIT TILL WE ARE
1429 004130 012777 004503 174414   REND1:    MOV        #R+IE+RB+DO,@TCCM ;FUNCTION = READ BLOCK, REVERSE AND GO
1430 004136 012777 004226 174420   MOV        #TCR1,@TCIV ;SET UP NEW INTERRUPT VECTOR
1431 004144 000002          RTI
1432          ;WRITE FORWARD ALL BLOCKS EXCEPT 0
1433
1434 004146 012777 004200 174410   TCWBK:    MOV        #TCWB1,@TCIV ;INTERRUPT VECTOR FOR WRITE
1435 004154 012777 177400 174376   MOV        #-400,@TCWC ;ONE BLOCK
1436 004162 012777 004510 174372   MOV        #TCWBUF,@TCBA ;THE WRITE BUFFER ADDRESS
1437 004170 112777 000515 174354   MOVB      #IE+WD+DC,@TCCM ;WRITE THE BLOCK
1438 004176 000002          RTI          ;RETURN WHEN BLOCK IS WRITTEN
1439 004200 005777 174346          TCWB1:    TST        @TCCM      ;ANY ERRORS
1440 004204 100001          BPL        .+4
1441 004206 104006          HLT
1442 004210 012777 004022 174346   MOV        #TCF3,@TCIV ;SEARCH BLOCK VECTOR
1443 004216 112777 000502 174326   MOVB      #IE+RB,@TCCM ;READ BLOCK
1444 004224 000721          BR          TCF4      ;FIND THE NEXT BLOCK
1445
1446 004226 032777 100200 174316   TCR1:    BIT        #100200,@TCCM ;TEST FOR ERROR AND READY
1447 004234 100001          BPL        .+4
1448 004236 104006          HLT
1449 004240 001001          BNE        .+4 ;DECTAPE ERROR ON READ BLOCK REVERSE
1450 004242 104006          HLT
1451 004244 027767 174306 177426   CMP        @TCDT,TCEXPE ;FALSE INTERRUPT FROM DECTAPE
1452 004252 001406          BEQ        TCR2      ;IS IT OUR FIRST BLOCK
1453 004254 002002          BGE        TCR1A     ;YES - GO TEST THE REST
1454 004256 104006          HLT          ;NO - HAVE WE PASSED THE BLOCK
1455 004260 000707          BR          RENDZ     ;WE PASS OUR BLOCK
1456 004262 105277 174264          TCR1A:    INCB      @TCCM      ;GO TO END ZONE AND TRY AGAIN
1457 004266 000002          RTI          ;SET DO
1458 004270 012777 004304 174266   TCR2:    MOV        #TCR3,@TCIV ;WE FOUND OUR FIRST BLOCK
1459 004276 105277 174250          INCB      @TCCM      ;SET UP INTERRUPT TO TEST ALL BLOCKS
1460 004302 000002          RTI          ;SET DO
1461          ;WAIT FOR NEXT BLOCK TO INTERRUPT
1462
1463 004304 032777 100200 174240   TCR3:    BIT        #100200,@TCCM ;FIND SEQUENTIAL BLOCK IN REVERSE DIRECTION
1464 004312 100001          BPL        .+4 ;TEST FOR READ AND ERROR
1465 004314 104006          HLT
1466 004316 001001          BNE        .+4 ;ERROR READING SEQUENTIAL BLOCK IN REVERSE
1467 004320 104006          HLT
1468 004322 026777 177346 174226   CMP        TCFIRST,@TCDT ;FALSE DECTAPE INTERRUPT
1469 004330 001662          BEQ        XFENDZ    ;DID WE DO ALL THE BLOCKS
1470 004332 005367 177342          DEC        TCXPE     ;YES - GO TO END ZONE TO RESTART
1471 004336 027767 174214 177334   CMP        @TCDT,TCXPE ;NO - DECREMENT BLOCK NUMBER
                                ;TEST SEQUENTIAL BLOCK IN REVERSE
    
```

```

1472 004344 001401          BEQ      .+4
1473 004346 104006          HLT
1474 004350 000403          BR      TCRBK      ;TEST SEQUENTIAL READ BLOCK IN REVERSE FAILED
1475 004352 105277 174174      TCR4:  INCB     @TCCM ;THIS ROUTINE READ A BLOCK
1476 004356 000002          RTI      ;SET DC
1477                                     ;LETS TRY A NEW BLOCK
1478                                     ;READ REVERSE ALL BLOCK EXCEPT BLOCK 1101
1479 004360 012777 004416 174176  †TCRBK: MOV     #TCRB1,@TCIV ;SET UP INTERRUPT VECTOR
1480 004366 012777 177400 174164      MOV     #-400,@TCWC ;READ ONE BLOCK
1481 004374 012777 004510 174160      MOV     #TCRBUF,@TCBA ;WHERE BUFFER IS
1482 004402 112777 000505 174142      MOVB   #IE+RD+DO,@TCCM ;READ THE BLOCK
1483 004410 004767 000030          JSR     %7,TC1      ;CHECK DATA BUFFER
1484 004414 000002          RTI      ;EXIT - RETURN WHEN BLOCK IS READ
1485 004416 005777 174130      TCRB1: TST     @TCCM ;AND ERRORS
1486 004422 100001          BPL     .+4
1487 004424 104006          HLT
1488 004426 012777 004304 174130      MOV     #TCR3,@TCIV ;DECTAPE ERROR
1489 004434 112777 000502 174110      MOVB   #IE+RB,@TCCM ;NEW VECTOR FOR BLOCK SEARCH
1490 004442 000743          BR      TCR4      ;READ BLOCK FUNCTION
1491                                     ;RETURN TO BLOCK SEARCH
1492                                     ;THIS ROUTINE CHECKS THE READ DATA BUFFER TC11
1493                                     ;BY DOING A CHECK SUM ON THE DATA
1494 004444 010146      TC1:  MOV     %1,-(6) ;SAVE THESE ON THE STACK
1495 004446 010246      MOV     %2,-(6)
1496 004450 010346      MOV     %3,-(6)
1497 004452 005003          CLR     %3
1498 004454 012701 004510          MOV     #TCRBUF,%1 ;SUM OF DATA
1499 004460 012702 005510          MOV     #TCRBUF+1000,%2 ;ADDRESS OF READ BUFFER
1500 004464 062103      TC2:  ADD     (1)+,%3 ;END OF READ BUFFER
1501 004466 062103      ADD     (1)+,%3 ;EVEN ADD
1502 004470 001401          BEQ     .+4 ;ODD ADD -2'S COMPLIMENT
1503 004472 104006          HLT
1504 004474 020102      CMP     %1,%2 ;DATA ERROR TC-11
1505 004476 001372      BNE     TC2 ;AT END OF BUFFER?
1506 004500 012603      MOV     (6)+,%3 ;NO - SUM THE REST
1507 004502 012602      MOV     (6)+,%2 ;RESTORE THE REGISTERS
1508 004504 012601      MOV     (6)+,%1
1509 004506 000207          RTS     %7 ;EXIT
1510
1511                                     ;THIS WRITE BUFFER LOOK THE SAME FORWARD OR REVERSE
1512 004510      †CWBUF:
1513 004510      TCRBUF:
1514                                     N=1
1515 004510 000001          N ;DECTAPE WRITE BUFFER
1516 004512 177777          -N
1517                                     N=N+1
1518 004514 000002          N ;DECTAPE WRITE BUFFER
1519 004516 177776          -N
1520                                     N=N+1
1521 004520 000003          N ;DECTAPE WRITE BUFFER
1522 004522 177775          -N
1523                                     N=N+1
1524 004524 000004          N ;DECTAPE WRITE BUFFER
1525 004526 177774          -N
1526                                     N=N+1
1527 004530 000005          N ;DECTAPE WRITE BUFFER
    
```

1528	004532	177773	-N	
1529		000006	N=N+1	
1530	004534	000006	N	;DECTAPE WRITE BUFFER
1531	004536	177772	-N	
1532		000007	N=N+1	
1533	004540	000007	N	;DECTAPE WRITE BUFFER
1534	004542	177771	-N	
1535		000010	N=N+1	
1536	004544	000010	N	;DECTAPE WRITE BUFFER
1537	004546	177770	-N	
1538		000011	N=N+1	
1539	004550	000011	N	;DECTAPE WRITE BUFFER
1540	004552	177767	-N	
1541		000012	N=N+1	
1542	004554	000012	N	;DECTAPE WRITE BUFFER
1543	004556	177766	-N	
1544		000013	N=N+1	
1545	004560	000013	N	;DECTAPE WRITE BUFFER
1546	004562	177765	-N	
1547		000014	N=N+1	
1548	004564	000014	N	;DECTAPE WRITE BUFFER
1549	004566	177764	-N	
1550		000015	N=N+1	
1551	004570	000015	N	;DECTAPE WRITE BUFFER
1552	004572	177763	-N	
1553		000016	N=N+1	
1554	004574	000016	N	;DECTAPE WRITE BUFFER
1555	004576	177762	-N	
1556		000017	N=N+1	
1557	004600	000017	N	;DECTAPE WRITE BUFFER
1558	004602	177761	-N	
1559		000020	N=N+1	
1560	004604	000020	N	;DECTAPE WRITE BUFFER
1561	004606	177760	-N	
1562		000021	N=N+1	
1563	004610	000021	N	;DECTAPE WRITE BUFFER
1564	004612	177757	-N	
1565		000022	N=N+1	
1566	004614	000022	N	;DECTAPE WRITE BUFFER
1567	004616	177756	-N	
1568		000023	N=N+1	
1569	004620	000023	N	;DECTAPE WRITE BUFFER
1570	004622	177755	-N	
1571		000024	N=N+1	
1572	004624	000024	N	;DECTAPE WRITE BUFFER
1573	004626	177754	-N	
1574		000025	N=N+1	
1575	004630	000025	N	;DECTAPE WRITE BUFFER
1576	004632	177753	-N	
1577		000026	N=N+1	
1578	004634	000026	N	;DECTAPE WRITE BUFFER
1579	004636	177752	-N	
1580		000027	N=N+1	
1581	004640	000027	N	;DECTAPE WRITE BUFFER
1582	004642	177751	-N	
1583		000030	N=N+1	

1584	004644	000030	N	;DECTAPE WRITE BUFFER
1585	004646	177750	-N	
1586		000031	N=N+1	
1587	004650	000031	N	;DECTAPE WRITE BUFFER
1588	004652	177747	-N	
1589		000032	N=N+1	
1590	004654	000032	N	;DECTAPE WRITE BUFFER
1591	004656	177746	-N	
1592		000033	N=N+1	
1593	004660	000033	N	;DECTAPE WRITE BUFFER
1594	004662	177745	-N	
1595		000034	N=N+1	
1596	004664	000034	N	;DECTAPE WRITE BUFFER
1597	004666	177744	-N	
1598		000035	N=N+1	
1599	004670	000035	N	;DECTAPE WRITE BUFFER
1600	004672	177743	-N	
1601		000036	N=N+1	
1602	004674	000036	N	;DECTAPE WRITE BUFFER
1603	004676	177742	-N	
1604		000037	N=N+1	
1605	004700	000037	N	;DECTAPE WRITE BUFFER
1606	004702	177741	-N	
1607		000040	N=N+1	
1608	004704	000040	N	;DECTAPE WRITE BUFFER
1609	004706	177740	-N	
1610		000041	N=N+1	
1611	004710	000041	N	;DECTAPE WRITE BUFFER
1612	004712	177737	-N	
1613		000042	N=N+1	
1614	004714	000042	N	;DECTAPE WRITE BUFFER
1615	004716	177736	-N	
1616		000043	N=N+1	
1617	004720	000043	N	;DECTAPE WRITE BUFFER
1618	004722	177735	-N	
1619		000044	N=N+1	
1620	004724	000044	N	;DECTAPE WRITE BUFFER
1621	004726	177734	-N	
1622		000045	N=N+1	
1623	004730	000045	N	;DECTAPE WRITE BUFFER
1624	004732	177733	-N	
1625		000046	N=N+1	
1626	004734	000046	N	;DECTAPE WRITE BUFFER
1627	004736	177732	-N	
1628		000047	N=N+1	
1629	004740	000047	N	;DECTAPE WRITE BUFFER
1630	004742	177731	-N	
1631		000050	N=N+1	
1632	004744	000050	N	;DECTAPE WRITE BUFFER
1633	004746	177730	-N	
1634		000051	N=N+1	
1635	004750	000051	N	;DECTAPE WRITE BUFFER
1636	004752	177727	-N	
1637		000052	N=N+1	
1638	004754	000052	N	;DECTAPE WRITE BUFFER
1639	004756	177726	-N	

1640		000053	N=N+1	
1641	004760	000053	N	;DECTAPE WRITE BUFFER
1642	004762	177725	-N	
1643		000054	N=N+1	
1644	004764	000054	N	;DECTAPE WRITE BUFFER
1645	004766	177724	-N	
1646		000055	N=N+1	
1647	004770	000055	N	;DECTAPE WRITE BUFFER
1648	004772	177723	-N	
1649		000056	N=N+1	
1650	004774	000056	N	;DECTAPE WRITE BUFFER
1651	004776	177722	-N	
1652		000057	N=N+1	
1653	005000	000057	N	;DECTAPE WRITE BUFFER
1654	005002	177721	-N	
1655		000060	N=N+1	
1656	005004	000060	N	;DECTAPE WRITE BUFFER
1657	005006	177720	-N	
1658		000061	N=N+1	
1659	005010	000061	N	;DECTAPE WRITE BUFFER
1660	005012	177717	-N	
1661		000062	N=N+1	
1662	005014	000062	N	;DECTAPE WRITE BUFFER
1663	005016	177716	-N	
1664		000063	N=N+1	
1665	005020	000063	N	;DECTAPE WRITE BUFFER
1666	005022	177715	-N	
1667		000064	N=N+1	
1668	005024	000064	N	;DECTAPE WRITE BUFFER
1669	005026	177714	-N	
1670		000065	N=N+1	
1671	005030	000065	N	;DECTAPE WRITE BUFFER
1672	005032	177713	-N	
1673		000066	N=N+1	
1674	005034	000066	N	;DECTAPE WRITE BUFFER
1675	005036	177712	-N	
1676		000067	N=N+1	
1677	005040	000067	N	;DECTAPE WRITE BUFFER
1678	005042	177711	-N	
1679		000070	N=N+1	
1680	005044	000070	N	;DECTAPE WRITE BUFFER
1681	005046	177710	-N	
1682		000071	N=N+1	
1683	005050	000071	N	;DECTAPE WRITE BUFFER
1684	005052	177707	-N	
1685		000072	N=N+1	
1686	005054	000072	N	;DECTAPE WRITE BUFFER
1687	005056	177706	-N	
1688		000073	N=N+1	
1689	005060	000073	N	;DECTAPE WRITE BUFFER
1690	005062	177705	-N	
1691		000074	N=N+1	
1692	005064	000074	N	;DECTAPE WRITE BUFFER
1693	005066	177704	-N	
1694		000075	N=N+1	
1695	005070	000075	N	;DECTAPE WRITE BUFFER

1696	005072	177703	-N	
1697		000076	N=N+1	
1698	005074	000076	N	;DECTAPE WRITE BUFFER
1699	005076	177702	-N	
1700		000077	N=N+1	
1701	005100	000077	N	;DECTAPE WRITE BUFFER
1702	005102	177701	-N	
1703		000100	N=N+1	
1704	005104	000100	N	;DECTAPE WRITE BUFFER
1705	005106	177700	-N	
1706		000101	N=N+1	
1707		000100	N=N-1	
1708	005110	177700	-N	
1709	005112	000100	N	;DEC TAPE WRITE BUFFER
1710		000077	N=N-1	
1711	005114	177701	-N	
1712	005116	000077	N	;DEC TAPE WRITE BUFFER
1713		000076	N=N-1	
1714	005120	177702	-N	
1715	005122	000076	N	;DEC TAPE WRITE BUFFER
1716		000075	N=N-1	
1717	005124	177703	-N	
1718	005126	000075	N	;DEC TAPE WRITE BUFFER
1719		000074	N=N-1	
1720	005130	177704	-N	
1721	005132	000074	N	;DEC TAPE WRITE BUFFER
1722		000073	N=N-1	
1723	005134	177705	-N	
1724	005136	000073	N	;DEC TAPE WRITE BUFFER
1725		000072	N=N-1	
1726	005140	177706	-N	
1727	005142	000072	N	;DEC TAPE WRITE BUFFER
1728		000071	N=N-1	
1729	005144	177707	-N	
1730	005146	000071	N	;DEC TAPE WRITE BUFFER
1731		000070	N=N-1	
1732	005150	177710	-N	
1733	005152	000070	N	;DEC TAPE WRITE BUFFER
1734		000067	N=N-1	
1735	005154	177711	-N	
1736	005156	000067	N	;DEC TAPE WRITE BUFFER
1737		000066	N=N-1	
1738	005160	177712	-N	
1739	005162	000066	N	;DEC TAPE WRITE BUFFER
1740		000065	N=N-1	
1741	005164	177713	-N	
1742	005166	000065	N	;DEC TAPE WRITE BUFFER
1743		000064	N=N-1	
1744	005170	177714	-N	
1745	005172	000064	N	;DEC TAPE WRITE BUFFER
1746		000063	N=N-1	
1747	005174	177715	-N	
1748	005176	000063	N	;DEC TAPE WRITE BUFFER
1749		000062	N=N-1	
1750	005200	177716	-N	
1751	005202	000062	N	;DEC TAPE WRITE BUFFER

1752		000061	N=N-1	
1753	005204	177717	-N	
1754	005206	000061	N	;DEC TAPE WRITE BUFFER
1755		000060	N=N-1	
1756	005210	177720	-N	
1757	005212	000060	N	;DEC TAPE WRITE BUFFER
1758		000057	N=N-1	
1759	005214	177721	-N	
1760	005216	000057	N	;DEC TAPE WRITE BUFFER
1761		000056	N=N-1	
1762	005220	177722	-N	
1763	005222	000056	N	;DEC TAPE WRITE BUFFER
1764		000055	N=N-1	
1765	005224	177723	-N	
1766	005226	000055	N	;DEC TAPE WRITE BUFFER
1767		000054	N=N-1	
1768	005230	177724	-N	
1769	005232	000054	N	;DEC TAPE WRITE BUFFER
1770		000053	N=N-1	
1771	005234	177725	-N	
1772	005236	000053	N	;DEC TAPE WRITE BUFFER
1773		000052	N=N-1	
1774	005240	177726	-N	
1775	005242	000052	N	;DEC TAPE WRITE BUFFER
1776		000051	N=N-1	
1777	005244	177727	-N	
1778	005246	000051	N	;DEC TAPE WRITE BUFFER
1779		000050	N=N-1	
1780	005250	177730	-N	
1781	005252	000050	N	;DEC TAPE WRITE BUFFER
1782		000047	N=N-1	
1783	005254	177731	-N	
1784	005256	000047	N	;DEC TAPE WRITE BUFFER
1785		000046	N=N-1	
1786	005260	177732	-N	
1787	005262	000046	N	;DEC TAPE WRITE BUFFER
1788		000045	N=N-1	
1789	005264	177733	-N	
1790	005266	000045	N	;DEC TAPE WRITE BUFFER
1791		000044	N=N-1	
1792	005270	177734	-N	
1793	005272	000044	N	;DEC TAPE WRITE BUFFER
1794		000043	N=N-1	
1795	005274	177735	-N	
1796	005276	000043	N	;DEC TAPE WRITE BUFFER
1797		000042	N=N-1	
1798	005300	177736	-N	
1799	005302	000042	N	;DEC TAPE WRITE BUFFER
1800		000041	N=N-1	
1801	005304	177737	-N	
1802	005306	000041	N	;DEC TAPE WRITE BUFFER
1803		000040	N=N-1	
1804	005310	177740	-N	
1805	005312	000040	N	;DEC TAPE WRITE BUFFER
1806		000037	N=N-1	
1807	005314	177741	-N	



1808	005316	000037	N	;DEC TAPE WRITE BUFFER
1809		000036	N=N-1	
1810	005320	177742	-N	
1811	005322	000036	N	;DEC TAPE WRITE BUFFER
1812		000035	N=N-1	
1813	005324	177743	-N	
1814	005326	000035	N	;DEC TAPE WRITE BUFFER
1815		000034	N=N-1	
1816	005330	177744	-N	
1817	005332	000034	N	;DEC TAPE WRITE BUFFER
1818		000033	N=N-1	
1819	005334	177745	-N	
1820	005336	000033	N	;DEC TAPE WRITE BUFFER
1821		000032	N=N-1	
1822	005340	177746	-N	
1823	005342	000032	N	;DEC TAPE WRITE BUFFER
1824		000031	N=N-1	
1825	005344	177747	-N	
1826	005346	000031	N	;DEC TAPE WRITE BUFFER
1827		000030	N=N-1	
1828	005350	177750	-N	
1829	005352	000030	N	;DEC TAPE WRITE BUFFER
1830		000027	N=N-1	
1831	005354	177751	-N	
1832	005356	000027	N	;DEC TAPE WRITE BUFFER
1833		000026	N=N-1	
1834	005360	177752	-N	
1835	005362	000026	N	;DEC TAPE WRITE BUFFER
1836		000025	N=N-1	
1837	005364	177753	-N	
1838	005366	000025	N	;DEC TAPE WRITE BUFFER
1839		000024	N=N-1	
1840	005370	177754	-N	
1841	005372	000024	N	;DEC TAPE WRITE BUFFER
1842		000023	N=N-1	
1843	005374	177755	-N	
1844	005376	000023	N	;DEC TAPE WRITE BUFFER
1845		000022	N=N-1	
1846	005400	177756	-N	
1847	005402	000022	N	;DEC TAPE WRITE BUFFER
1848		000021	N=N-1	
1849	005404	177757	-N	
1850	005406	000021	N	;DEC TAPE WRITE BUFFER
1851		000020	N=N-1	
1852	005410	177760	-N	
1853	005412	000020	N	;DEC TAPE WRITE BUFFER
1854		000017	N=N-1	
1855	005414	177761	-N	
1856	005416	000017	N	;DEC TAPE WRITE BUFFER
1857		000016	N=N-1	
1858	005420	177762	-N	
1859	005422	000016	N	;DEC TAPE WRITE BUFFER
1860		000015	N=N-1	
1861	005424	177763	-N	
1862	005426	000015	N	;DEC TAPE WRITE BUFFER
1863		000014	N=N-1	

1864	005430	177764	-N	
1865	005432	000014	N	;DEC TAPE WRITE BUFFER
1866		000013	N=N-1	
1867	005434	177765	-N	
1868	005436	000013	N	;DEC TAPE WRITE BUFFER
1869		000012	N=N-1	
1870	005440	177766	-N	
1871	005442	000012	N	;DEC TAPE WRITE BUFFER
1872		000011	N=N-1	
1873	005444	177767	-N	
1874	005446	000011	N	;DEC TAPE WRITE BUFFER
1875		000010	N=N-1	
1876	005450	177770	-N	
1877	005452	000010	N	;DEC TAPE WRITE BUFFER
1878		000007	N=N-1	
1879	005454	177771	-N	
1880	005456	000007	N	;DEC TAPE WRITE BUFFER
1881		000006	N=N-1	
1882	005460	177772	-N	
1883	005462	000006	N	;DEC TAPE WRITE BUFFER
1884		000005	N=N-1	
1885	005464	177773	-N	
1886	005466	000005	N	;DEC TAPE WRITE BUFFER
1887		000004	N=N-1	
1888	005470	177774	-N	
1889	005472	000004	N	;DEC TAPE WRITE BUFFER
1890		000003	N=N-1	
1891	005474	177775	-N	
1892	005476	000003	N	;DEC TAPE WRITE BUFFER
1893		000002	N=N-1	
1894	005500	177776	-N	
1895	005502	000002	N	;DEC TAPE WRITE BUFFER
1896		000001	N=N-1	
1897	005504	177777	-N	
1898	005506	000001	N	;DEC TAPE WRITE BUFFER
1899				

```

1900 ;*****
1901 .SBTTL MAIN ROUTINE: CPU BACKGROUND TESTS
1902 ;*****
1903 005510 010701 BEGINX: MOV PC,R1 ;SET UP R1 TO SELECT CURBNK
1904 005512 042701 017777 BIC #1777,R1
1905 005516 042737 160000 000034 BIC #160000,#34 ;SET SCOPE RET TO CURRENT BANK
1906 005524 050137 000034 BIS R1,#34
1907 005530 000301 SWAB R1
1908 005532 006201 ASR R1
1909 005534 006201 ASR R1
1910 005536 010137 000570 MOV R1,#CURBNK
1911
1912 ;BINARY INSTRUCTIONS
1913 ;INDEX, AND INDIRECT TEST OF PDP-11
1914 BEGIN:
1915 005542 005000 CLR R0 ;CHECK RANDOM NUMBER GENERATOR SEEDS.
1916 005544 066700 007210 ADD RP1,R0 ;AND RESTORE IF ZEROED.
1917 005550 066700 007206 ADD RP2,R0
1918 005554 001006 BNE IS ;BR IF NOT ZEROED.
1919 005556 012767 001233 007174 MOV #1233,RP1 ;RESTORE RP1 SEED.
1920 005564 012767 007622 007170 MOV #7622,RP2 ;RESTORE RP2 SEED.
1921 005572 005067 007104 IS: CLR SCOPEF
1922 005576 010767 007102 MOV PC,RETURN ;FOR SCOPING - SETUP ADDRESS OF BEGINI IN
1923 005602 062767 000042 007074 ADD #42,RETURN ;THIS BANK THRU CURRENT ASR
1924 005610 016767 007064 007060 MOV $ICNT,ICOUNT ;ITERATION COUNT
1925 005616 005737 000042 ST #42 ;AUTO MODE?
1926 005622 001407 BEQ ZS ;BR IF NOT.
1927 005624 023737 000042 000046 CMP #42,#46 ;XXDP CHAIN MODE?
1928 005632 001403 BEQ ZS ;BR IF NOT.
1929 005634 016767 007050 007034 MOV XDPcnt,ICOUNT ;USE XXDP CHAIN ITERATION COUNT.
1930 ZS:
1931 ;*****
1932 .SBTTL TEST COMPARE INSTRUCTION INDEXED
1933 ;*****
1934 005642 012700 177770 MOV #10,%0 ;MINUS 10 TO REG 0
1935 005646 026027 014774 125252 CMP A(0),#125252 ;(A INDEX BY MINUS 10) TO #125252
1936 005654 001401 BEQ .+4
1937 005656 104006 HLT ;COMPARE WITH INDEX FAILED
1938 005660 104400 SCOPE
1939
1940 005662 012700 000010 MOV #10,%0
1941 005666 022760 052525 014774 CMP #052525,A(0)
1942 005674 001401 BEQ .+4
1943 005676 104006 HLT
1944 005700 104400 SCOPE
1945
1946 005702 012700 177770 MOV #10,%0
1947 005706 026060 014774 014774 CMP A(0),A(0)
1948 005714 001401 BEQ .+4
1949 005716 104006 HLT
1950 005720 104400 SCOPE
1951
1952 005722 012700 000010 MOV #+10,%0
1953 005726 026060 014774 014774 CMP A(0),A(0)
1954 005734 001401 BEQ .+4
1955 005736 104006 HLT

```

1956	005740	104400			SCOPE
1957					
1958	005742	012700	177774		MOV #-4,%0
1959	005746	012701	000010		MOV #+10,%1
1960	005752	026061	014774	014774	CMP A(0),A(1)
1961	005760	001401			BEQ .+4
1962	005762	104006			HLT
1963	005764	104400			SCOPE
1964					
1965	005766	012700	177774		MOV #-4,%0
1966	005772	012701	000010		MOV #10,%1
1967	005776	026160	014774	014774	CMP A(1),A(0)
1968	006004	001401			BEQ .+4
1969	006006	104006			HLT
1970	006010	104400			SCOPE
1971					
1972					*****
1973					.SBTTL TEST MOVE INSTRUCTION FOR INDEX
1974					*****
1975					
1976	006012	012700	177770		MOV #-10,%0
1977	006016	016067	014774	006772	MOV A(0),TEMP
1978	006024	026727	006766	125252	CMP TEMP,#125252
1979	006032	001401			SEQ .+4
1980	006034	104006			HLT
1981	006036	104400			SCOPE
1982					
1983	006040	012700	177770		MOV #-10,%0
1984	006044	012760	125252	015016	MOV #125252,TEMP(0)
1985	006052	023727	015006	125252	CMP @#C,#125252
1986	006060	001401			BEQ .+4
1987	006062	104006			HLT
1988	006064	104400			SCOPE
1989					
1990					*****
1991					.SBTTL TEST BIC INSTRUCTION FOR INDEXING
1992					*****
1993	006066	012767	177777	006722	MOV #-1,TEMP
1994	006074	012700	177770		MOV #-10,%0
1995	006100	046067	014774	006710	BIC A(0),TEMP
1996	006106	026727	006704	052525	CMP TEMP,#052525
1997	006114	001401			BEQ .+4
1998	006116	104006			HLT
1999	006120	104400			SCOPE
2000					
2001	006122	012700	177770		MOV #-10,%0
2002	006126	012767	177777	006652	MOV #-1,TEMP-10
2003	006134	042767	052525	006644	BIC #052525,TEMP-10
2004	006142	026727	006640	125252	CMP TEMP-10,#125252
2005	006150	001401			BEQ .+4
2006	006152	104006			HLT
2007	006154	104400			SCOPE
2008					
2009	006156	012737	125252	015016	MOV #125252,@#TEMP
2010	006164	012700	177770		MOV #-10,%0
2011	006170	166760	006570	015026	SUB B,TEMP+10(0)

2012	006176	001401			BEQ	.+4
2013	006200	104006			HLT	
2014	006202	104400			SCOPE	
2015						
2016	006204	012737	052525	015016	MOV	#052525,@#TEMP
2017	006212	012700	000010		MOV	#10,%0
2018	006216	166760	006562	015006	SUB	A+10,C(0)
2019	006224	001401			BEQ	.+4
2020	006226	104006			HLT	
2021	006230	104400			SCOPE	

```

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

```

2022						
2023						
2024						
2025						
2026						
2027	006232	012737	177777	015016	MOV	#-1,@#TEMP
2028	006240	012700	000010		MOV	#+10,%0
2029	006244	005060	015006		CLR	C(0)
2030	006250	005737	015016		TST	@#TEMP
2031	006254	001401			BEQ	.+4
2032	006256	104006			HLT	
2033	006260	104400			SCOPE	
2034						
2035	006262	012737	177777	015016	MOV	#-1,@#TEMP
2036	006270	012700	000010		MOV	#10,%0
2037	006274	005160	015006		COM	C(0)
2038	006300	005737	015016		TST	@#TEMP
2039	006304	001401			BEQ	.+4
2040	006306	104006			HLT	
2041	006310	104400			SCOPE	
2042						
2043	006312	012737	177777	015016	MOV	#-1,@#TEMP
2044	006320	012700	177770		MOV	#-10,%0
2045	006324	005260	015026		INC	D(0)
2046	006330	005737	015016		TST	@#TEMP
2047	006334	001401			BEQ	.+4
2048	006336	104006			HLT	
2049	006340	104400			SCOPE	
2050						
2051	006342	012737	000001	015016	MOV	#1,@#TEMP
2052	006350	012700	177770		MOV	#-10,%0
2053	006354	005360	015026		DEC	D(0)
2054	006360	005737	015016		TST	@#TEMP
2055	006364	001401			BEQ	.+4
2056	006366	104006			HLT	
2057	006370	104400			SCOPE	
2058						
2059	006372	012737	000001	015016	MOV	#1,@#TEMP
2060	006400	012700	000010		MOV	#10,%0
2061	006404	005360	015006		DEC	C(0)
2062	006410	005737	015016		TST	@#TEMP
2063	006414	001401			BEQ	.+4
2064	006416	104006			HLT	
2065	006420	104400			SCOPE	
2066						
2067	006422	012737	000001	015016	MOV	#1,@#TEMP

G04

2068	006430	012700	177770		MOV	#-10,%0
2069	006434	005460	015026		NEG	D(0)
2070	006440	022737	177777	015016	CMP	#-1,2#TEMP
2071	006446	001401			BEQ	+.4
2072	006450	104006			HLT	
2073	006452	104400			SCOPE	
2074						
2075	006454	012737	000001	015016	MOV	#1,2#TEMP
2076	006462	012700	000010		MOV	#+10,%0
2077	006466	005460	015006		NEG	C(0)
2078	006472	022737	177777	015016	CMP	#-1,2#TEMP
2079	006500	001401			BEQ	+.4
2080	006502	104006			HLT	
2081	006504	104400			SCOPE	
2082						
2083	006506	012737	177777	015016	MOV	#-1,2#TEMP
2084	006514	012700	177770		MOV	#-10,%0

2085	006520	000261			SEC	
2086	006522	005560	015026		ADC	D(0)
2087	006526	005737	015016		TST	@#TEMP
2088	006532	001401			BEQ	+.4
2089	006534	104006			HLT	
2090	006536	104400			SCOPE	
2091						
2092	006540	012737	177777	015016	MOV	#-1,@#TEMP
2093	006546	012700	000010		MOV	#+10,%0
2094	006552	000261			SEC	
2095	006554	005560	015006		ADC	C(0)
2096	006560	005737	015016		TST	@#TEMP
2097	006564	001401			BEQ	+.4
2098	006566	104006			HLT	
2099	006570	104400			SCOPE	
2100						
2101	006572	012737	000001	015016	MOV	#1,@#TEMP
2102	006600	012700	177770		MOV	#-10,%0
2103	006604	000261			SEC	
2104	006606	005660	015026		SBC	D(0)
2105	006612	005737	015016		TST	@#TEMP
2106	006616	001401			BEQ	+.4
2107	006620	104006			HLT	
2108	006622	104400			SCOPE	
2109						
2110	006624	012737	000001	015016	MOV	#1,@#TEMP
2111	006632	012700	000010		MOV	#+10,%0
2112	006636	000261			SEC	
2113	006640	005660	015006		SBC	C(0)
2114	006644	005737	015016		TST	@#TEMP
2115	006650	001401			BEQ	+.4
2116	006652	104006			HLT	
2117	006654	104400			SCOPE	
2118						
2119						
2120						*****
2121						:SBTTL TEST JMP INDIRECT
2122	006656	010700				*****
2123	006660	062700	000010		MOV	%7,%0
2124	006664	000110			ADD	#10,%0
2125	006666	104006			JMP	@%0
2126	006670	000240			HLT	
2127	006672	104400			NOP	
2128					SCOPE	
2129	006674	010700				
2130	006676	062700	000010		MOV	%7,%0
2131	006702	000110			ADD	#10,%0
2132	006704	104006			JMP	@%0
2133	006706	000240			HLT	
2134	006710	104400			NOP	
2135					SCOPE	
2136						*****
2137						:SBTTL TEST INDIRECT ADDRESSINGTEST COMPARE INSTRUCTION
2138						*****
2139	006712	023727	014764	125252	CMP	@#B,#125252
2140	006720	001401			BEQ	+.4

2141	006722	104006			HLT	
2142	006724	104400			SCOPE	
2143						
2144	006726	022737	125252	014764	CMP	#125252, @#B
2145	006734	001401			BEQ	.+4
2146	006736	104006			HLT	
2147	006740	104400			SCOPE	
2148						
2149	006742	023737	014764	014764	CMP	@#B, @#B
2150	006750	001401			BEQ	.+4
2151	006752	104006			HLT	
2152	006754	104400			SCOPE	
2153						
2154						*****
2155						.SBTTL TEST MOVE INSTRUCTIONS
2156						*****
2157	006756	013700	014764		MOV	@#B, %0
2158	006762	022700	125252		CMP	#125252, %0
2159	006766	001401			BEQ	.+4
2160	006770	104006			HLT	
2161	006772	104400			SCOPE	
2162						
2163	006774	012737	125252	015016	MOV	#125252, @#TEMP
2164	007002	023737	014764	015016	CMP	@#B, @#TEMP
2165	007010	001401			BEQ	.+4
2166	007012	104006			HLT	
2167	007014	104400			SCOPE	
2168						
2169	007016	013737	014764	015006	MOV	@#B, @#C
2170	007024	023737	014764	015006	CMP	@#B, @#C
2171	007032	001401			BEQ	.+4
2172	007034	104006			HLT	
2173	007036	104400			SCOPE	
2174						
2175						*****
2176						.SBTTL TEST BIC INSTRUCTION INDIRECT
2177						*****
2178	007040	012700	177777		MOV	#-1, %0
2179	007044	043700	014764		BIC	@#B, %0
2180	007050	020027	052525		CMP	%0, #052525
2181	007054	001401			BEQ	.+4
2182	007056	104006			HLT	
2183	007060	104400			SCOPE	
2184						
2185	007062	012737	177777	015016	MOV	#-1, @#TEMP
2186	007070	042737	125252	015016	BIC	#125252, @#TEMP
2187	007076	022737	052525	015016	CMP	#052525, @#TEMP
2188	007104	001401			BEQ	.+4
2189	007106	104006			HLT	
2190	007110	104400			SCOPE	
2191						
2192	007112	012737	177777	015006	MOV	#-1, @#C
2193	007120	043737	014764	015006	BIC	@#B, @#C
2194	007126	023727	015006	052525	CMP	@#C, #52525
2195	007134	001401			BEQ	.+4
2196	007136	104006			HLT	



```

2197 007140 104400 SCOPE
2198
2199 ;*****
2200 .SBTTL TEST SUBTRACT INSTRUCTION
2201 ;*****
2202 007142 012700 125252 MOV #125252,%0
2203 007146 163700 014764 SUB @#B,%0
2204 007152 020027 000000 CMP %0,%0
2205 007156 001401 BEQ .+4
2206 007160 104006 HLT
2207 007162 104400 SCOPE
2208
2209 007164 012737 125252 015016 MOV #125252,@#TEMP
2210 007172 166737 005566 015016 SUB B,@#TEMP
2211 007200 001401 BEQ .+4
2212 007202 104006 HLT
2213 007204 104400 SCOPE
2214
2215 007206 012767 125252 005602 MOV #125252,TEMP
2216 007214 163767 014764 005574 SUB @#B,TEMP
2217 007222 005767 005570 TST TEMP
2218 007226 001401 BEQ .+4
2219 007230 104006 HLT
2220 007232 104400 SCOPE
2221
2222 ;*****
2223 .SBTTL TEST ADD INDIRECT
2224 ;*****
2225 007234 005000 CLR %0
2226 007236 063700 014764 ADD @#B,%0
2227 007242 022700 125252 CMP #125252,%0
2228 007246 001401 BEQ .+4
2229 007250 104006 HLT
2230 007252 104400 SCOPE
2231
2232 007254 005037 015016 CLR @#TEMP
2233 007260 062737 125252 015016 ADD #125252,@#TEMP
2234 007266 022737 125252 015016 CMP #125252,@#TEMP
2235 007274 001401 BEQ .+4
2236 007276 104006 HLT
2237 007300 104400 SCOPE
2238
2239 007302 012737 125252 015016 MOV #125252,@#TEMP
2240 007310 067737 005466 015016 ADD @A+6,@#TEMP
2241 007316 023727 015016 177777 CMP @#TEMP,#-1
2242 007324 001401 BEQ .+4
2243 007326 104006 HLT
2244 007330 104400 SCOPE
2245
2246 ;*****
2247 .SBTTL TEST UNARYS INDIRECT
2248 ;*****
2249 007332 012737 177777 015016 MOV #-1,@#TEMP
2250 007340 005037 015016 CLR @#TEMP
2251 007344 005737 015016 TST @#TEMP
2252 007350 001401 BEQ .+4

```

```

2253 007352 104006 HLT
2254 007354 104400 SCOPE
2255
2256 007356 012737 125252 015016 MOV #125252,@#TEMP
2257 007364 005137 015016 COM @#TEMP
2258 007370 022737 052525 015016 CMP #052525,@#TEMP
2259 007376 001401 BEQ .+4
2260 007400 104006 HLT
2261 007402 104400 SCOPE
2262
2263 007404 005037 015016 CLR @#TEMP
2264 007410 005237 015016 INC @#TEMP
2265 007414 022737 000001 015016 CMP #1,@#TEMP
2266 007422 001401 BEQ .+4
2267 007424 104006 HLT
2268 007426 104400 SCOPE
2269
2270 007430 005037 015016 CLR @#TEMP
2271 007434 005377 005360 DEC @TEMP+2
2272 007440 023727 015016 177777 CMP @#TEMP,#-1
2273 007446 001401 BEQ .+4
2274 007450 104006 HLT
2275 007452 104400 SCOPE
2276
2277 007454 012737 000001 015016 MOV #1,@#TEMP
2278 007462 005437 015016 NEG @#TEMP
2279 007466 022737 177777 015016 CMP #-1,@#TEMP
2280 007474 001401 BEQ .+4
2281 007476 104006 HLT
2282 007500 104400 SCOPE
2283
2284
2285 ;*****
2286 ;SBTTL TEST INDIRECT ADDRESSING WITH INDEXING, TEST COMPARE INSTRUCTION
2287 ;*****
2287 007502 027727 005260 125252 CMP @B+2,#125252
2288 007510 001401 BEQ .+4
2289 007512 104006 HLT
2290 007514 104400 SCOPE
2291
2292 007516 022777 125252 005242 CMP #125252,@B+2
2293 007524 001401 BEQ .+4
2294 007526 104006 HLT
2295 007530 104400 SCOPE
2296
2297 007532 027777 005230 005226 CMP @B+2,@B+2
2298 007540 001401 BEQ .+4
2299 007542 104006 HLT
2300 007544 104400 SCOPE
2301
2302
2303 ;*****
2304 ;SBTTL TEST MOVE INSTRUCTIONS
2305 ;*****
2305 007546 017700 005214 MOV @B+2,%0
2306 007552 022700 125252 CMP #125252,%0
2307 007556 001401 BEQ .+4
2308 007560 104006 HLT

```

2309	007562	104400			SCOPE
2310					
2311	007564	012777	125252	005226	MOV #125252,@TEMP+2
2312	007572	023737	014764	015016	CMP @#B,@#TEMP
2313	007600	001401			BEQ .+4
2314	007602	104006			HLT
2315	007604	104400			SCOPE
2316					
2317	007606	017777	005154	005174	MOV @B+2,@C+2
2318	007614	023737	014764	015006	CMP @#B,@#C
2319	007622	001401			BEQ .+4
2320	007624	104006			HLT
2321	007626	104400			SCOPE
2322					
2323					
2324					
2325					
2326	007630	012700	177777		MOV #-1,%0
2327	007634	047700	005126		BIC @B+2,%0
2328	007640	020027	052525		CMP %0,#52525
2329	007644	001401			BEQ .+4
2330	007646	104006			HLT
2331	007650	104400			SCOPE
2332					
2333	007652	012737	177777	015016	MOV #-1,@#TEMP
2334	007660	042777	125252	005132	BIC #125252,@TEMP+2
2335	007666	022737	052525	015016	CMP #52525,@#TEMP
2336	007674	001401			BEQ .+4
2337	007676	104006			HLT
2338	007700	104400			SCOPE
2339					
2340	007702	012737	177777	015006	MOV #-1,@#C
2341	007710	047777	005052	005072	BIC @B+2,@C+2
2342	007716	026737	005062	015006	CMP A+10,@#C
2343	007724	001401			BEQ .+4
2344	007726	104006			HLT
2345	007730	104400			SCOPE
2346					
2347	007732	012700	125252		MOV #125252,%0
2348	007736	167700	005024		SUB @B+2,%0
2349	007742	020027	000000		CMP %0,#0
2350	007746	001401			BEQ .+4
2351	007750	104006			HLT
2352	007752	104400			SCOPE
2353					
2354	007754	012737	125252	015016	MOV #125252,@#TEMP
2355	007762	166777	004776	005030	SUB B,@TEMP+2
2356	007770	001401			BEQ .+4
2357	007772	104006			HLT
2358	007774	104400			SCOPE
2359					
2360	007776	012737	125252	015016	MOV #125252,@#TEMP
2361	010004	167777	004756	005006	SUB @B+2,@TEMP+2
2362	010012	005737	015016		TST @#TEMP
2363	010016	001401			BEQ .+4
2364	010020	104006			HLT

```

;*****
.SBTTL TEST BIC INSTRUCTION INDIRECT WITH INDEXING
;*****

```

```

2365 010022 104400          SCOPE
2366
2367 ;*****
2368 .SBTTL TEST ADD INDIRECT WITH INDEXING
2369 ;*****
2370 010024 005000          CLR      %0
2371 010026 067700 004734  ADD      @B+2,%0
2372 010032 022700 125252  CMP      #125252,%0
2373 010036 001401          BEQ      .+4
2374 010040 104006          HLT
2375 010042 104400          SCOPE
2376
2377 010044 005037 015016  CLR      @#TEMP
2378 010050 062777 125252 004742  ADD      #125252,@TEMP+2
2379 010056 022737 125252 015016  CMP      #125252,@#TEMP
2380 010064 001401          BEQ      .+4
2381 010066 104006          HLT
2382 010070 104400          SCOPE
2383
2384 010072 012737 125252 015016  MOV      #125252,@#TEMP
2385 010100 067777 004676 004712  ADD      @A+6,@TEMP+2
2386 010106 023727 015016 177777  CMP      @#TEMP,#-1
2387 010114 001401          BEQ      .+4
2388 010116 104006          HLT
2389 010120 104400          SCOPE
2390
2391 ;*****
2392 .SBTTL TEST UNARYS INDIRECT WITH INDEXING
2393 ;*****
2394 010122 012737 177777 015016  MOV      #-1,@#TEMP
2395 010130 005077 004664          CLR      @TEMP+2
2396 010134 005737 015016          TST      @#TEMP
2397 010140 001401          BEQ      .+4
2398 010142 104006          HLT
2399 010144 104400          SCOPE
2400
2401 010146 012737 125252 015016  MOV      #125252,@#TEMP
2402 010154 005177 004640          COM      @TEMP+2
2403 010160 022737 052525 015016  CMP      #052525,@#TEMP
2404 010166 001401          BEQ      .+4
2405 010170 104006          HLT
2406 010172 104400          SCOPE
2407
2408 010174 005037 015016  CLR      @#TEMP
2409 010200 005277 004614          INC      @TEMP+2
2410 010204 022737 000001 015016  CMP      #1,@#TEMP
2411 010212 001401          BEQ      .+4
2412 010214 104006          HLT
2413 010216 104400          SCOPE
2414
2415 010220 005037 015016  CLR      @#TEMP
2416 010224 005377 004570          DEC      @TEMP+2
2417 010230 023727 015016 177777  CMP      @#TEMP,#-1
2418 010236 001401          BEQ      .+4
2419 010240 104006          HLT
2420 010242 104400          SCOPE

```

```

2421
2422 010244 012737 000001 015016
2423 010252 005477 004542
2424 010256 022737 177777 015016
2425 010264 001401
2426 010266 104006
2427 010270 104400
2428
2429 010272 012737 177777 015016
2430 010300 000261
2431 010302 005577 004512
2432 010306 005737 015016
2433 010312 001401
2434 010314 104006
2435 010316 104400
2436
2437 010320 012737 000001 015016
2438 010326 000261
2439 010330 005677 004464
2440 010334 005737 015016
2441 010340 001401
2442 010342 104006
2443 010344 104400
2444
2445
2446
2447
2448 010346 012700 177772
2449 010352 027027 014774 125252
2450 010360 001401
2451 010362 104006
2452 010364 104400
2453
2454 010366 012700 177772
2455 010372 022770 125252 014774
2456 010400 001401
2457 010402 104006
2458 010404 104400
2459
2460 010406 012700 177772
2461 010412 012701 000002
2462 010416 027071 014774 014774
2463 010424 001401
2464 010426 104006
2465 010430 104400
2466
2467
2468
2469
2470 010432 012700 000006
2471 010436 012767 177777 004352
2472 010444 047067 014774 004344
2473 010452 022767 125252 004336
2474 010460 001401
2475 010462 104006
2476 010464 104400

```

```

MOV #1,@TEMP
NEG @TEMP+2
CMP #-1,@TEMP
BEQ .+4
HLT
SCOPE

```

```

MOV #-1,@TEMP
SEC
ADC @TEMP+2
TST @TEMP
BEQ .+4
HLT
SCOPE

```

```

MOV #1,@TEMP
SEC
SBC @TEMP+2
TST @TEMP
BEQ .+4
HLT
SCOPE

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

2477									
2478	010466	012700	177772		MOV	#-6,%0			
2479	010472	012737	177777	015006	MOV	#-1,@#C			
2480	010500	042770	125252	015016	BIC	#125252,@TEMP(0)			
2481	010506	023727	015006	052525	CMP	@#C,#052525			
2482	010514	001401			BEQ	+.4			
2483	010516	104006			HLT				
2484	010520	104400			SCOPE				
2485									
2486	010522	012737	177777	015006	MOV	#-1,@#C			
2487	010530	012700	177772		MOV	#-6,%0			
2488	010534	012701	177772		MOV	#-6,%1			
2489	010540	047071	014774	015016	BIC	@A(0),@TEMP(1)			
2490	010546	022737	052525	015006	CMP	#052525,@#C			
2491	010554	001401			BEQ	+.4			
2492	010556	104006			HLT				
2493	010560	104400			SCOPE				
2494									
2495									
2496									
2497									
2498									
2499									
2500	010562	012700	177770		MOV	#-10,%0			
2501	010566	126027	014774	000252	CMPB	A(0),#000252			
2502	010574	001401			BEQ	+.4			
2503	010576	104006			HLT				
2504	010600	104400			SCOPE				
2505									
2506	010602	012700	177770		MOV	#-10,%0			
2507	010606	122760	000252	014774	CMPB	#000252,A(0)			
2508	010614	001401			BEQ	+.4			
2509	010616	104006			HLT				
2510	010620	104400			SCOPE				
2511									
2512	010622	012700	000010		MOV	#10,%0			
2513	010626	126027	014774	000125	CMPB	A(0),#000125			
2514	010634	001401			BEQ	+.4			
2515	010636	104006			HLT				
2516	010640	104400			SCOPE				
2517									
2518	010642	012700	000010		MOV	#10,%0			
2519	010646	122760	000125	014774	CMPB	#000125,A(0)			
2520	010654	001401			BEQ	+.4			
2521	010656	104006			HLT				
2522	010660	104400			SCOPE				
2523									
2524	010662	012700	177770		MOV	#-10,%0			
2525	010666	126060	014774	014774	CMPB	A(0),A(0)			
2526	010674	001401			BEQ	+.4			
2527	010676	104006			HLT				
2528	010700	104400			SCOPE				
2529									
2530	010702	012700	000010		MOV	#+10,%0			
2531	010706	126060	014774	014774	CMPB	A(0),A(0)			
2532	010714	001401			BEQ	+.4			

;BINARY INSTRUCTIONS  
;INDEX, AND INDIRECT TEST OF PDP-11  
;\*\*\*\*\*  
.SBTL TEST COMPARE INSTRUCTION INDEXED  
;\*\*\*\*\*  
;MINUS 10 TO REG C  
;(A INDEX BY MINUS 10) TO #125252  
;COMPARE WITH INDEX FAILED  
;FOR INDEX  
;A INDEXED  
;INDEX

2533	010716	104006			HLT	
2534	010720	104400			SCOPE	
2535						
2536	010722	012700	177770		MOV	#-10,%0
2537	010726	012701	000004		MOV	#+4,%1
2538	010732	126061	014774	014774	CMPB	A(0),A(1)
2539	010740	001401			BEQ	+.4
2540	010742	104006			HLT	
2541	010744	104400			SCOPE	
2542						
2543	010746	126160	014774	014774	CMPB	A(1),A(0)
2544	010754	001401			BEQ	+.4
2545	010756	104006			HLT	
2546	010760	104400			SCOPE	
2547						
2548	010762	012700	177774		MOV	#-4,%0
2549	010766	012701	000010		MOV	#+10,%1
2550	010772	126061	014774	014774	CMPB	A(0),A(1)
2551	011000	001401			BEQ	+.4
2552	011002	104006			HLT	
2553	011004	104400			SCOPE	
2554						
2555	011006	012700	177774		MOV	#-4,%0
2556	011012	012701	000010		MOV	#10,%1
2557	011016	126160	014774	014774	CMPB	A(1),A(0)
2558	011024	001401			BEQ	+.4
2559	011026	104006			HLT	
2560	011030	104400			SCOPE	
2561						
2562						
2563						
2564						
2565	011032	012700	177770		MOV	#-10,%0
2566	011036	116067	014774	003752	MOV	A(0),TEMP
2567	011044	126727	003746	000252	CMPB	TEMP,#000252
2568	011052	001401			BEQ	+.4
2569	011054	104006			HLT	
2570	011056	104400			SCOPE	
2571						
2572	011060	012700	000010		MOV	#+10,%0
2573	011064	116067	014774	003724	MOV	A(0),TEMP
2574	011072	126727	003720	000125	CMPB	TEMP,#000125
2575	011100	001401			BEQ	+.4
2576	011102	104006			HLT	
2577	011104	104400			SCOPE	
2578						
2579	011106	012700	177770		MOV	#-10,%0
2580	011112	112760	125252	015016	MOV	#125252,TEMP(0)
2581	011120	123727	015006	125252	CMPB	TEMP,#125252
2582	011126	001401			BEQ	+.4
2583	011130	104006			HLT	
2584	011132	104400			SCOPE	
2585						
2586	011134	012700	000010		MOV	#+10,%0
2587	011140	112760	052525	015016	MOV	#052525,TEMP(0)
2588	011146	123727	015026	052525	CMPB	TEMP+10,#052525

```

;*****
;SBTTL TEST MOVE INSTRUCTION FOR INDEX
;*****

```

```

2589 011154 001401      BEQ      .+4
2590 011156 104006      HLT
2591 011160 104400      SCOPE
2592
2593 ;*****
2594 .SBTTL TEST BIC INSTRUCTION FOR INDEXING
2595 ;*****
2596 011162 012767 177777 003626      MOV      #-1,TEMP
2597 011170 012700 177770      MOV      #-10,%0
2598 011174 146067 014774 003614      BICB    A(0),TEMP
2599 011202 126727 003610 177525      CMPB    TEMP,#177525
2600 011210 001401      BEQ      .+4
2601 011212 104006      HLT
2602 011214 104400      SCOPE
2603
2604 011216 012767 177777 003572      MOV      #-1,TEMP
2605 011224 012700 000010      MOV      #10,%0
2606 011230 146067 014774 003560      BICB    A(0),TEMP
2607 011236 126727 003554 007652      CMPB    TEMP,#0C7652
2608 011244 001401      BEQ      .+4
2609 011246 104006      HLT
2610 011250 104400      SCOPE
2611
2612 011252 012737 177777 015026      MOV      #-1,@#TEMP+10
2613 011260 012700 000010      MOV      #10,%0
2614 011264 142760 125252 015016      BICB    #125252,TEMP(0)
2615 011272 123727 015026 002525      CMPB    @#TEMP+10,#2525
2616 011300 001401      BEQ      .+4
2617 011302 104006      HLT
2618 011304 104400      SCOPE
2619
2620 011306 012700 177770      MOV      #-10,%0
2621 011312 012767 177777 003466      MOV      #-1,TEMP-10
2622 011320 142767 052525 003460      BICB    #052525,TEMP-10
2623 011326 126727 003454 125252      CMPB    TEMP-10,#125252
2624 011334 001401      BEQ      .+4
2625 011336 104006      HLT
2626 011340 104400      SCOPE
2627
2628 ;*****
2629 .SBTTL TEST UNARYS INDEXED
2630 ;*****
2631 011342 012737 177777 015016      MOV      #-1,@#TEMP
2632 011350 012700 177770      MOV      #-10,%0
2633 011354 105060 015026      CLRB    D(0)
2634 011360 105737 015016      TSTB    @#TEMP
2635 011364 001401      BEQ      .+4
2636 011366 104006      HLT
2637 011370 104400      SCOPE
2638
2639 011372 012737 177777 015016      MOV      #-1,@#TEMP
2640 011400 012700 177770      MOV      #-10,%0
2641 011404 105060 015026      CLRB    D(0)
2642 011410 023727 015016 177400      CMP     @#TEMP,#177400
2643 011416 001401      BEQ      .+4
2644 011420 104006      HLT

```



2645	011422	104400			SCOPE
2646					
2647	011424	012737	177777	015016	MOV #-1,@#TEMP
2648	011432	012700	177771		MOV #-7,%0
2649	011436	105060	015026		CLRB D(0)
2650	011442	023727	015016	000377	CMP @#TEMP,#000377
2651	011450	001401			BEQ .+4
2652	011452	104006			HLT
2653	011454	104400			SCOPE
2654					
2655	011456	012737	177777	015016	MOV #-1,@#TEMP
2656	011464	012700	000010		MOV #+10,%0
2657	011470	105060	015006		CLRB C(0)
2658	011474	105737	015016		TSTB @#TEMP
2659	011500	001401			BEQ .+4
2660	011502	104006			HLT
2661	011504	104400			SCOPE
2662					
2663	011506	012737	177777	015016	MOV #-1,@#TEMP
2664	011514	012700	177770		MOV #-10,%0
2665	011520	105160	015026		COMB D(0)
2666	011524	105737	015016		TSTB @#TEMP
2667	011530	001401			BEQ .+4
2668	011532	104006			HLT
2669	011534	104400			SCOPE
2670					
2671	011536	012737	177777	015016	MOV #-1,@#TEMP
2672	011544	012700	000010		MOV #+10,%0
2673	011550	105260	015006		INCB C(0)
2674	011554	105737	015016		TSTB @#TEMP
2675	011560	001401			BEQ .+4
2676	011562	104006			HLT
2677	011564	104400			SCOPE
2678					
2679	011566	012737	000001	015016	MOV #1,@#TEMP
2680	011574	012700	177770		MOV #-10,%0
2681	011600	105360	015026		DECB D(0)
2682	011604	105737	015016		TSTB @#TEMP
2683	011610	001401			BEQ .+4
2684	011612	104006			HLT
2685	011614	104400			SCOPE
2686					
2687	011616	012737	000001	015016	MOV #1,@#TEMP
2688	011624	012700	000010		MOV #+10,%0
2689	011630	105460	015006		NEGB C(0)
2690	011634	023727	015016	000377	CMP @#TEMP,#377
2691	011642	001401			BEQ .+4
2692	011644	104006			HLT
2693	011646	104400			SCOPE
2694					
2695	011650	012737	177777	015016	MOV #-1,@#TEMP
2696	011656	012700	177770		MOV #-10,%0
2697	011662	000261			SEC
2698	011664	105560	015026		ADCB D(0)
2699	011670	023727	015016	177400	CMP @#TEMP,#177400
2700	011676	001401			BEQ .+4

```

2701 011700 104006 HLT
2702 011702 104400 SCOPE
2703
2704 011704 012737 000001 015016 MOV #1,@#TEMP
2705 011712 012700 000010 MOV #+10,%0
2706 011716 000261 SEC
2707 011720 105660 015006 SBCB C(0)
2708 011724 005737 015016 TST @#TEMP
2709 011730 001401 BEQ .+4
2710 011732 104006 HLT
2711 011734 104400 SCOPE
2712
2713 ;*****
2714 .SBTTL TEST INDIRECT ADDRESSING, TEST COMPARE INSTRUCTION
2715 ;*****
2716 011736 123727 014764 000252 CMPB @#B,#000252
2717 011744 001401 BEQ .+4
2718 011746 104006 HLT
2719 011750 104400 SCOPE
2720
2721 011752 122737 125252 014764 CMPB #125252,@#B
2722 011760 001401 BEQ .+4
2723 011762 104006 HLT
2724 011764 104400 SCOPE
2725
2726 ;*****
2727 .SBTTL TEST MOVE INSTRUCTIONS
2728 ;*****
2729 011766 113700 014764 MOVB @#B,%0
2730 011772 122700 000252 CMPB #000252,%0
2731 011776 001401 BEQ .+4
2732 012000 104006 HLT
2733 012002 104400 SCOPE
2734
2735 012004 112737 125252 015016 MOVB #125252,@#TEMP
2736 012012 126737 002746 015016 CMPB B,@#TEMP
2737 012020 001401 BEQ .+4
2738 012022 104006 HLT
2739 012024 104400 SCOPE
2740
2741 ;*****
2742 .SBTTL TEST UNARYS INDIRECT
2743 ;*****
2744 012026 012737 177777 015016 MOV #-1,@#TEMP
2745 012034 105037 015016 CLRB @#TEMP
2746 012040 023727 015016 177400 CMP @#TEMP,#177400
2747 012046 001401 BEQ .+4
2748 012050 104006 HLT
2749 012052 104400 SCOPE
2750
2751 012054 012737 125252 015016 MOV #125252,@#TEMP
2752 012062 105137 015017 COMB @#TEMP+1
2753 012066 022737 052652 015016 CMP #052652,@#TEMP
2754 012074 001401 BEQ .+4
2755 012076 104006 HLT
2756 012100 104400 SCOPE

```

```

2757
2758 012102 005037 015016 CLR @#TEMP
2759 012106 105237 015017 INCB @#TEMP+1
2760 012112 022737 000400 015016 CMP #400,@#TEMP
2761 012120 001401 BEQ .+4
2762 012122 104006 HLT
2763 012124 104400 SCOPE
2764
2765 012126 005037 015016 CLR @#TEMP
2766 012132 105377 002662 DECB @TEMP+2
2767 012136 023727 015016 000377 CMP @#TEMP,#377
2768 012144 001401 BEQ .+4
2769 012146 104006 HLT
2770 012150 104400 SCOPE
2771
2772 012152 005037 015016 CLR @#TEMP
2773 012156 112737 000001 015017 MOVB #1,@#TEMP+1
2774 012164 105437 015017 NEGB @#TEMP+1
2775 012170 022737 177400 015016 CMP #177400,@#TEMP
2776 012176 001401 BEQ .+4
2777 012200 104006 HLT
2778 012202 104400 SCOPE
2779

```

```

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

```

```

2784 012204 122777 125252 002554 CMPB #125252,@B+2
2785 012212 001401 BEQ .+4
2786 012214 104006 HLT
2787 012216 104400 SCOPE
2788
2789 012220 127777 002542 002540 CMPB @B+2,@B+2
2790 012226 001401 BEQ .+4
2791 012230 104006 HLT
2792 012232 104400 SCOPE
2793

```

```

;*****
;SBTTL TEST MOVE INSTRUCTIONS
;*****

```

```

2797 012234 117700 002526 MOVB @B+2,%0
2798 012240 122700 125252 CMPB #125252,%0
2799 012244 001401 BEQ .+4
2800 012246 104006 HLT
2801 012250 104400 SCOPE
2802
2803 012252 112777 125252 002540 MOVB #125252,@TEMP+2
2804 012260 126737 002500 015016 CMPB B,@#TEMP
2805 012266 001401 BEQ .+4
2806 012270 104006 HLT
2807 012272 104400 SCOPE
2808
2809 012274 117777 002466 002506 MOVB @B+2,@C+2
2810 012302 126737 002456 015006 CMPB B,@#C
2811 012310 001401 BEQ .+4
2812 012312 104006 HLT

```

TEST MOVE INSTRUCTIONS

Address	Op1	Op2	Op3	Op4	Instruction
2813	012314	104400			SCOPE
2814					
2815					*****
2816					.SBTTL TEST BIC INSTRUCTION INDIRECT WITH INDEXING
2817					*****
2818	012316	012700	177777		MOV #-1,%0
2819	012322	147700	002440		BICB @B+2,%0
2820	012326	120027	052525		CMPB %0,#52525
2821	012332	001401			BEQ .+4
2822	012334	104006			HLT
2823	012336	104400			SCOPE
2824					
2825	012340	012737	177777	015016	MOV #-1,@#TEMP
2826	012346	142777	125252	002444	BICB #125252,@TEMP+2
2827	012354	122737	052525	015016	CMPB #52525,@#TEMP
2828	012362	001401			BEQ .+4
2829	012364	104006			HLT
2830	012366	104400			SCOPE
2831					
2832	012370	012737	177777	015006	MOV #-1,@#C
2833	012376	147777	002364	002404	BICB @B+2,@C+2
2834	012404	126737	002374	015006	CMPB A+10,@#C
2835	012412	001401			BEQ .+4
2836	012414	104006			HLT
2837	012416	104400			SCOPE
2838					
2839					*****
2840					.SBTTL TEST UNARYS INDIRECT WITH INDEXING
2841					*****
2842	012420	012737	177777	015016	MOV #-1,@#TEMP
2843	012426	105077	002366		CLRB @TEMP+2
2844	012432	105737	015016		TSTB @#TEMP
2845	012436	001401			BEQ .+4
2846	012440	104006			HLT
2847	012442	104400			SCOPE
2848					
2849	012444	005037	015016		CLR @#TEMP
2850	012450	105277	002344		INCB @TEMP+2
2851	012454	122737	000001	015016	CMPB #1,@#TEMP
2852	012462	001401			BEQ .+4
2853	012464	104006			HLT
2854	012466	104400			SCOPE
2855					
2856	012470	005037	015016		CLR @#TEMP
2857	012474	105377	002320		DECB @TEMP+2
2858	012500	123727	015016	177777	CMPB @#TEMP,#-1
2859	012506	001401			BEQ .+4
2860	012510	104006			HLT
2861	012512	104400			SCOPE
2862					
2863	012514	012737	000001	015016	MOV #1,@#TEMP
2864	012522	105477	002272		NEGB @TEMP+2
2865	012526	122737	177777	015016	CMPB #-1,@#TEMP
2866	012534	001401			BEQ .+4
2867	012536	104006			HLT
2868	012540	104400			SCOPE

```

2869
2870 012542 012737 177777 015016      MOV      #-1,@TEMP
2871 012550 000261                    SEC
2872 012552 105577 002242              ADCB     @TEMP+2
2873 012556 022737 177400 015016      CMP      #177400,@TEMP
2874 012564 001401                    BEQ      .+4
2875 012566 104006                    HLT
2876 012570 105737 015016              TSTB    @TEMP
2877 012574 001401                    BEQ      .+4
2878 012576 104006                    HLT
2879 012600 104400                    SCOPE
2880
2881 012602 012737 000001 015016      MOV      #1,@TEMP
2882 012610 000261                    SEC
2883 012612 105377 002202              DECB    @TEMP+2
2884 012616 005737 015016              TST     @TEMP
2885 012622 001401                    BEQ      .+4
2886 012624 104006                    HLT
2887 012626 104400                    SCOPE
2888
2889
2890
2891
2892 012630 012700 177772              MOV      #-6,%0
2893 012634 127027 014774 125252      CMPB    @A(0),#125252
2894 012642 001401                    BEQ      .+4
2895 012644 104006                    HLT
2896 012646 104400                    SCOPE
2897
2898 012650 012700 177772              MOV      #-6,%0
2899 012654 012701 000002              MOV      #+2,%1
2900 012660 127071 014774 014774      CMPB    @A(0),@A(1)
2901 012666 001401                    BEQ      .+4
2902 012670 104006                    HLT
2903 012672 104400                    SCOPE
2904
2905
2906
2907
2908 012674 012700 000006              MOV      #+6,%0
2909 012700 012767 177777 002110      MOV      #-1,TEMP
2910 012706 147067 014774 002102      BICB    @A(0),TEMP
2911 012714 122767 125252 002074      CMPB    #125252,TEMP
2912 012722 001401                    BEQ      .+4
2913 012724 104006                    HLT
2914 012726 104400                    SCOPE
2915
2916 012730 012700 177772              MOV      #-6,%0
2917 012734 012737 177777 015006      MOV      #-1,@#C
2918 012742 142770 125252 015016      BICB    #125252,@TEMP(0)
2919 012750 123727 015006 000125      CMPB    @#C,#000125
2920 012756 001401                    BEQ      .+4
2921 012760 104006                    HLT
2922 012762 104400                    SCOPE
2923
2924 012764 012700 014766              MOV      #B+2,%0          ;ADDRESS OF ADDRESS OF B

```

2925	012770	023067	001770		CMP	a(0)+,B	
2926	012774	001401			BEQ	.+4	
2927	012776	104006			HLT		
2928	013000	104400			SCOPE		
2929							
2930	013002	012700	014770		MOV	#B+4,%0	
2931	013006	025067	001752		CMP	a-(0),B	
2932	013012	001401			BEQ	.+4	
2933	013014	104006			HLT		
2934	013016	104400			SCOPE		
2935							
2936	013020	012700	014770		MOV	#B+4,%0	
2937	013024	125067	001734		CMPB	a-(0),B	
2938	013030	001401			BEQ	.+4	
2939	013032	104006			HLT		
2940	013034	104400			SCOPE		
2941							
2942	013036	012700	015012		MOV	#C+4,%0	
2943	013042	012737	177777	015006	MOV	#-1,a#C	
2944	013050	105050			CLRB	a-(0)	
2945	013052	023727	015006	177400	CMP	a#C,#177400	
2946	013060	001401			BEQ	.+4	
2947	013062	104006			HLT		
2948	013064	104400			SCOPE		
2949							
2950	013066	012737	177777	015006	MOV	#-1,a#C	
2951	013074	012700	177772		MOV	#-6,%0	
2952	013100	012701	177772		MOV	#-6,%1	
2953	013104	147071	014774	015016	BICB	aA(0),aTEMP(1)	
2954	013112	022737	177525	015006	CMP	#177525,a#C	
2955	013120	001401			BEQ	.+4	
2956	013122	104006			HLT		
2957	013124	104400			SCOPE		
2958							
2959							
2960							
2961							
2962	013126	012700	052525		MOV	#52525,%0	;THIS IS CHECKED LATER IN PROGRAM
2963							
2964							
2965							
2966							
2967	013132	004767	000002		JSR	%7,TJSR2	;PLACE PC ON STACK
2968	013136	000405			TJSR1: BR	TJSR3	;RETURN HERE ON RTS %19
2969	013140	121627	013136		TJSR2: CMPB	a%6,#TJSR1	;CHECK FOR CORRECT PC ON STACK
2970	013144	001401			BEQ	.+4	
2971	013146	104006			HLT		;INCORRECT PC ON STACK
2972	013150	000207			RTS	%7	;RETURN TO IMST AFTER JSR
2973	013152	104400			TJSR3: SCOPE		
2974							
2975	013154	000257			CCC		
2976	013156	004717			JSR	%7,a%7	;INSTRUCTION UNDER TEST
2977	013160	121627	013160		CMPB	a%6,#TJSR3+6	;TEST THE STACK
2978	013164	001401			BEQ	.+4	
2979	013166	104006			HLT		;PC OF JSR DID NOT GO TO STACK
2980	013170	005726			TST	(6)+	;REPOSITION THE STACK

```

2981 013172 104400          SCOPE
2982
2983 ;*****
2984 .SBTTL TEST NESTED SUBROUTINES
2985 ;*****
2986 013174 000257          CCC          ;CLEAR CONDITION CODES
2987 013176 004767 001360  JSR          %7, SUBR6
2988 013202 100401          BMI          .+4
2989 013204 104006          HLT          ;JSR OR RTS FAILED
2990 013206 001401          BEQ          .+4
2991 013210 104006          HLT          ;JSR OR RTS FAILED
2992 013212 102401          BVS          .+4
2993 013214 104006          HLT          ;JSR OR RTS FAILED
2994 013216 103401          BCS          .+4
2995 013220 104006          HLT          ;JSR OR RTS FAILED
2996 013222 104400          SCOPE
2997
2998 ;*****
2999 .SBTTL TEST ROTATE ODD BYTE
3000 ;*****
3001 013224 104400          SCOPE
3002 013226 000257          CCC          ;CLEAR "C"
3003 013230 012767 123456 001560  MOV          #123456, TEMP
3004 013236 106067 001555          RORB         TEMP+1          ;ROTATE ODD BYTE
3005 013242 103401          BCS          .+4
3006 013244 104006          HLT          ;C NOT SET
3007 013246 102401          BVS          .+4
3008 013250 104006          HLT          ;V NOT SET
3009 013252 022767 051456 001536  CMP          #051456, TEMP
3010 013260 001401          BEQ          .+4
3011 013262 104006          HLT          ;ROTATE FAILED
3012 013264 104400          SCOPE
3013
3014 013266 000277          SCC          ;SET C
3015 013270 012767 123456 001520  MOV          #123456, TEMP
3016 013276 106067 001515          RORB         TEMP+1
3017 013302 103401          BCS          .+4
3018 013304 104006          HLT          ;C NOT SET
3019 013306 102001          BVC          .+4
3020 013310 104006          HLT          ;V NOT CLEARED
3021 013312 022767 151456 001476  CMP          #151456, TEMP
3022 013320 001401          BEQ          .+4
3023 013322 104006          HLT          ;ROTATE FAILED
3024 013324 104400          SCOPE
3025
3026 013326 000257          CCC
3027 013330 012767 123456 001460  MOV          #123456, TEMP
3028 013336 106167 001455          ROLB         TEMP+1
3029 013342 103401          BCS          .+4
3030 013344 104006          HLT          ;C NOT SET
3031 013346 102401          BVS          .+4
3032 013350 104006          HLT          ;V NOT SET
3033 013352 022767 047056 001436  CMP          #047056, TEMP
3034 013360 001401          BEQ          .+4
3035 013362 104006          HLT          ;ROTATE BYTE FAILED
3036 013364 104400          SCOPE

```

```

3037
3038 013366 000277          SCC          ;SET C
3039 013370 012767 123456 001420  MOV      #123456,TEMP
3040 013376 106167 001415  ROLB    TEMP+1
3041 013402 103401          BCS     .+4
3042 013404 104006          HLT          ;C NOT SET
3043 013406 102401          BVS     .+4
3044 013410 104006          HLT          ;V NOT SET
3045 013412 022767 047456 001376  CMP     #047456,TEMP
3046 013420 001401          BEQ     .+4
3047 013422 104006          HLT          ;ROTATE ODD BYTE FAILED
3048 013424 104400          SCOPE
3049
3050 013426 000257          CCC          ;CLEAR C
3051 013430 012767 177777 001360  MOV     #-1,TEMP
3052 013436 106267 001355  ASRB   TEMP+1
3053 013442 103401          BCS     .+4
3054 013444 104006          HLT          ;C NOT SET
3055 013446 102001          BVC     .+4
3056 013450 104006          HLT          ;V NOT CLEARED
3057 013452 026727 001340 177777  CMP     TEMP,#-1
3058 013460 001401          BEQ     .+4
3059 013462 104006          HLT          ;SHIFT FAILED
3060 013464 104400          SCOPE
3061
3062 013466 000277          SCC
3063 013470 012767 177777 001320  MOV     #-1,TEMP
3064 013476 106367 001315  ASLB   TEMP+1
3065 013502 103401          BCS     .+4
3066 013504 104006          HLT          ;C NOT SET
3067 013506 102001          BVC     .+4
3068 013510 104006          HLT          ;V NOT CLEARED
3069 013512 026727 001300 177377  CMP     TEMP,#177377
3070 013520 001401          BEQ     .+4
3071 013522 104006          HLT          ;SHIFT BYTE FAILED
3072 013524 104400          SCOPE
3073
3074
3075
3076
3077

```

```

;*****
.SBTTL TEST THAT RO WASN'T CLEARED BY FALSE SELECTION
;*****

```

```

3078 013526 022700 052525  CMP     #52525,RO
3079 013532 001401          BEQ     .+4
3080 013534 104006          HLT
3081 013536 104400          SCOPE
3082

```

```

;*****
.SBTTL TEST COMBINATIONS OF NUMBERS WITH COMPARE INSTRUCTION
;*****

```

```

3086 013540 004767 001146  JSR     PC,RNGEN          ;GET RANDOM NUMBER.
3087 013544 010001          MOV     RO,R1
3088 013546 020001  CMP1:  CMP     %0,%1          ;ARE THE EQUAL
3089 013550 001401          BEQ     .+4
3090 013552 104006          HLT
3091 013554 104400          SCOPE          ;RO AND R1 DID NOT COMPARE
3092

```



```

3093 ;*****
3094 .SBTTL TEST ROTATING RANDOM NUMBERS
3095 ;*****
3096 013556 004767 001130 JSR PC,RNGEN ;GET RANDOM NUMBER.
3097 013562 010067 165030 MOV RO,REFF ;PUT IN REF WORD.
3098 013566 016767 165024 165024 ROTALL: MOV REFF,TEST
3099 013574 006067 165020 ROR TEST
3100 013600 006067 165014 ROR TEST
3101 013604 006067 165010 ROR TEST
3102 013610 006167 165004 ROL TEST
3103 013614 006167 165000 ROL TEST
3104 013620 006167 164774 ROL TEST
3105 013624 100004 BPL .+12
3106 013626 103007 BCC .+20 ;Z=1
3107 013630 102013 BVC .+30 ;Z=1, C=1
3108 013632 104006 HLT ;Z=C, BUT V=1
3109 013634 000411 BR .+24
3110 013636 103006 BCC .+16 ;Z=0
3111 013640 102407 BVS .+20 ;Z=0, C=1
3112 013642 104006 HLT ;Z NOT EQUAL C, V=1
3113 013644 000405 BR .+14
3114 013646 102404 BVS .+12 ;Z=1, C=0
3115 013650 104006 HLT ;Z NOT EQUAL C, V=1
3116 013652 000402 BR .+6
3117 013654 102001 BVC .+4 ;Z=0, C=0
3118 013656 104006 HLT ;Z=C, BUT V=1
3119 013660 026767 164734 164730 CMP TEST,REFF
3120 013666 001401 BEQ .+4
3121 013670 104006 HLT ;INITIAL NOT EQUAL TO FINAL
3122 013672 104400 SCOPE
3123
3124 000616 REF=REFF
3125
3126 ;*****
3127 .SBTTL TEST ROTATING BYTE EVEN/ODD, RANDOM NUMBERS
3128 ;*****
3129 013674 004767 001012 JSR PC,RNGEN ;GET RANDOM NUMBER.
3130 013700 010067 164712 MOV RO,REFF ;PUT IN REF WORD.
3131 013704 004767 000006 JSR %7,ROTBE
3132 013710 004767 000110 JSR %7,ROTBO
3133 013714 000503 BR ROTENI
3134 013716 016767 164674 164674 ROTBE: MOV REFF,TEST
3135 013724 106067 164670 RORB TEST ;ROTATE BYTE EVEN
3136 013730 106067 164664 RORB TEST
3137 013734 106067 164660 RORB TEST
3138 013740 106167 164654 ROLB TEST
3139 013744 106167 164650 ROLB TEST
3140 013750 106167 164644 ROLB TEST
3141 013754 100004 BPL .+12
3142 013756 103007 BCC .+20 ;Z=1
3143 013760 102013 BVC .+30 ;Z=1, C=1
3144 013762 104006 HLT ;Z=C, BUT V=1
3145 013764 000411 BR .+24
3146 013766 103006 BCC .+16 ;Z=0
3147 013770 102407 BVS .+20 ;Z=0, C=1
3148 013772 104006 HLT ;Z NOT EQUAL C, V=1

```

3149	013774	000405		
3150	013776	102404		
3151	014000	104006		
3152	014002	000402		
3153	014004	102001		
3154	014006	104006		
3155	014010	026767	164604	164600
3156	014016	001401		
3157	014020	104006		
3158	014022	000207		
3159	014024	106067	164571	
3160	014030	106067	164565	
3161	014034	106067	164561	
3162	014040	106167	164555	
3163	014044	106167	164551	

ROTBO:

BR	:+14
BVS	:+12
HLT	
BR	:+6
BVC	:+4
HLT	
CMP	TEST,REFF
BEQ	:+4
HLT	
RTS	%7
RORB	TEST+1
RORB	TEST+1
RORB	TEST+1
ROLB	TEST+1
ROLB	TEST+1

;Z=1, C=0  
;Z NOT EQUAL C, V=1

;Z=0, C=0  
;Z=C, BUT V=1

;ROTATE BYTE ODD

```

3164 014050 106167 164545
3165 014054 100004
3166 014056 103007
3167 014060 102013
3168 014062 104006
3169 014064 000411
3170 014066 103006
3171 014070 102407
3172 014072 104006
3173 014074 000405
3174 014076 102404
3175 014100 104006
3176 014102 000402
3177 014104 102001
3178 014106 104006
3179 014110 026767 164504 164500
3180 014116 001401
3181 014120 104006
3182 014122 000207
3183 014124 104400
3184
3185
3186
3187
3188
3189 014126 011667 000054
3190 014132 004767 000554
3191 014136 010067 164454
3192 014142 004767 000002
3193 014146 000420
3194 014150 016767 164442 164442
3195 014156 066767 000024 164434
3196 014164 166767 000016 164426
3197 014172 026767 164420 164420
3198 014200 001401
3199 014202 104006
3200 014204 000207
3201 014206 000000
3202 014210 104400
3203
3204
3205
3206
3207 014212 004767 000474
3208 014216 010067 164376
3209 014222 012767 177777 164366
3210 014230 160067 164362
3211 014234 005167 164360
3212 014240 026767 164354 164350
3213 014246 001401
3214 014250 104006
3215
3216 014252 104400
3217
3218
3219

```

```

ROLB TEST+1
BPL .+12
BCC .+20 ;Z=1
BVC .+30 ;Z=1, C=1
HLT ;Z=C, BUT V=1
BR .+24
BCC .+16 ;Z=0
BVS .+20 ;Z=0, C=1
HLT ;Z NOT EQUAL C, V=1
BR .+14
BVS .+12 ;Z=1, C=0
HLT ;Z NOT EQUAL C, V=1
BR .+6
BVC .+4 ;Z=0, C=0
HLT ;Z=C, BUT V=1
CMP TEST, REFF
BEQ .+4
HLT
RTS %7
ROTEN1: SCOPE
;*****
;SBTTL ADD AND SUBTRACT RANDOM NUMBERS AGAINST FIXED NUMBERS
;*****
;A+B=C, C-A=B, BF SHOULD EQUAL BI
MOV %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.

```

```

3220 ;*****
3221 014254 004767 000432 JSR PC,RNGEN ;GET RANDOM NUMBER.
3222 014260 010067 164334 MOV RO,TEST ;PUT IN TEST.
3223 014264 012767 177777 164324 MOV #-1,REFF ;PREDETERMINE THE RESULT.
3224 014272 160067 164320 SUB RO,REFF
3225 014276 105167 164316 COMB TEST ;DO COMPLEMENT EVEN BYTE.
3226 014302 126767 164312 164306 CMPB TEST,REFF ;EXPECTED RESULT?
3227 014310 001401 BEQ .+4 ;BR IF YES. OK.
3228 014312 104006 HLT ;ERROR!! RESULT IN TEST DOES NOT MATCH
3229 ;EXPECTED RESULT IN REFF.
3230 014314 104400 SCOPE
3231
3232 ;*****
3233 .SBTTL TEST COMB (ODD BYTE) RANDOM NUMBERS
3234 ;*****
3235 014316 004767 000370 JSR PC,RNGEN ;GET RANDOM NUMBER.
3236 014322 010067 164272 MOV RO,TEST ;PUT IN TEST.
3237 014326 012767 177777 164262 MOV #-1,REFF ;PREDETERMINE RESULT.
3238 014334 160067 164256 SUB RO,REFF
3239 014340 105167 164255 COMB TEST+1 ;DO ODD BYTE COMPLEMENT.
3240 014344 126767 164251 164245 CMPB TEST+1,REFF+1 ;EXPECTED RESULT?
3241 014352 001401 BEQ .+4 ;BR IF YES. OK.
3242 014354 104006 HLT ;ERROR!! RESULT IN TEST DOES NOT MATCH
3243 ;EXPECTED RESULT IN REFF.
3244 014356 104400 SCOPE
3245
3246 ;*****
3247 .SBTTL TEST COMPARE RANDOM VALUES EVEN BYTE WITH ODD
3248 ;*****
3249 014360 004767 000326 JSR PC,RNGEN ;GET RANDOM NUMBER.
3250 014364 110067 000426 MOVB RO,TEMP ;PUT IN LOW BYTE OF TEMP.
3251 014370 110067 000423 MOVB RO,TEMP+1 ;PUT IN HIGH BYTE.
3252 014374 126767 000416 000415 CMPB TEMP,TEMP+1 ;DO THE COMPARE.
3253 014402 001401 BEQ .+4 ;BR IF EQUAL. OK.
3254 014404 104006 HLT ;COMPARE FAILED.
3255 014406 002001 BGE .+4
3256 014410 104006 HLT ;V IS NOT = TO N
3257 014412 003401 BLE .+4
3258 014414 104006 HLT ;V IS SET
3259 014416 104400 SCOPE
3260
3261 ;*****
3262 .SBTTL TEST SWAB
3263 ;*****
3264 014420 012767 000200 164172 MOV #0200,TEST
3265 014426 000367 164166 SWAB TEST
3266 014432 100001 BPL .+4
3267 014434 104006 HLT
3268 014436 001401 BEQ .+4
3269 014440 104006 HLT
3270 014442 000367 164152 SWAB TEST
3271 014446 100401 BMI .+4
3272 014450 104006 HLT
3273 014452 001001 BNE .+4
3274 014454 104006 HLT
3275 014456 104400 SCOPE

```

3276  
3277  
3278  
3279  
3280  
3281  
3282  
3283  
3284  
3285  
3286  
3287  
3288  
3289  
3290

014460 004767 000226  
014464 010067 164126  
014470 105067 164123  
014474 116767 164116 164117  
014502 105067 164112  
014506 000367 164106  
014512 026767 164102 164076  
014520 001401  
014522 104006  
014524 104400

```

;*****
;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

```

```

3291 ;*****
3292 ;SBTTL END OF USER CODE IN BANK
3293 ;*****
3294 ;CALL KERNEL/
3295 ;ALTERED IN CORE EXPANSION/
3296 014526 104010 DONE: EOB ;THIS EMT CALL GOES TO EOBSRV
3297 014530 000240 NOP ;TO ALLOW CORE EXPANSION TO PATCH IN JMP
3298
3299 ;*****
3300 ;SBTTL GROUP OF NESTED SUBROUTINES/
3301 ;*****
3302 014532 000207 SUBR1: RTS %7 ;ONE INSTRUCTION
3303 014534 000277 SUBR2: SCC ;ONE DEEP
3304 014536 000207 RTS %7
3305 014540 004767 177770 SUBR3: JSR %7, SUBR2 ;TWO DEEP
3306 014544 000207 RTS %7
3307 014546 004767 177766 SUBR4: JSR %7, SUBR3 ;THREE DEEP
3308 014552 000207 RTS %7
3309 014554 004767 177766 SUBR5: JSR %7, SUBR4 ;FOUR DEEP
3310 014560 000207 RTS %7
3311 014562 004767 177766 SUBR6: JSR %7, SUBR5 ;FIVE DEEP
3312 014566 000207 RTS %7
3313
3314 ;*****
3315 ;SBTTL SCOPE AND/OR ITERATION LOOP FOR EACH TEST TIMES/
3316 ;*****
3317 014570 005737 000042 SCOPEC: TST @#42 ;IN AUTOMATIC MODE?
3318 014574 001017 BNE 2$ ;BR IF YES.
3319 014576 032767 002000 163372 BIT #2000, SREG2 ;INHIBIT PROCESSOR TESTS?
3320 014604 001403 BEQ 1$ ;NO
3321 014606 022626 CMP (SP)+, (SP)+ ;CLEAN UP EMT CALL FROM STACK.
3322 014610 000167 165552 JMP MAIN ;YES
3323 014614 032767 040000 163354 1$: BIT #40000, SREG2 ;TEST SR FOR SCOPE
3324 014622 001015 BNE SCOPEB ;YES SCOPE
3325 014624 032767 004000 163344 BIT #4000, SREG2 ;NO-TEST FOR ITERATION
3326 014632 001014 BNE SCOPEG ;INHIBIT ITERATION
3327 014634 005767 000046 2$: TST PASCNT ;IN FIRST PASS?
3328 014640 001411 BEQ SCOPEG ;BR IF YES. NO ITERATIONS ON 1ST PASS.
3329 014642 005267 000034 INC SCOPEF ;INCREMENT COUNT.
3330 014646 026767 000030 000022 CMP SCOPEF, ICOUNT ;COMPARE CURRENT COUNT TO MAX NUMBER
3331 014654 103003 BHIS SCOPEG ;EXIT-DONE
3332 014656 016716 000022 SCOPEB: MOV RETURN, @SP
3333 014662 000002 RTI
3334 014664 005067 000012 SCOPEG: CLR SCOPEF ;CLEAR COUNT
3335 014670 011667 000010 MOV @%6, RETURN ;SAVE SCOPE RETURN POINTER
3336 014674 000002 RTI ;RETURN INLINE-NEXT TEST
3337 014676 000000 ICOUNT: 0 ;ITERATION COUNT
3338 014700 000024 $ICNT: 20. ;STANDARD ITERATION COUNT.
3339 014702 000000 SCOPEF: 0 ;COUNT LOCATION FOR ITERATION LOOP
3340 014704 000000 RETURN: 0 ;ADDRESS OF LAST TEST
3341 014706 000000 PASCNT: 0 ;HOLDS PASS COUNT.
3342 014710 001750 XDPCNT: 1000. ;PASS COUNT TO USE WHEN IN XXDP CHAIN MODE.
3343
3344 ;*****
3345 ;SBTTL ROUTINE RNGEN
3346

```

```

3347
3348
3349 014712 016700 000042
3350 014716 006100
3351 014720 006100
3352 014722 066700 000034
3353 014726 010067 000026
3354 014732 006100
3355 014734 006100
3356 014736 066700 000020
3357 014742 006100
3358 014744 006100
3359 014746 010067 000010
3360 014752 016700 000002
3361 014756 000207
3362 014760 001233
3363 014762 007622
3364
3365
3366
3367
3368 014764 125252
3369 014766 014764
3370 014770 052525
3371 014774 014774
3372 014774 177777
3373 014776 015000
3374 015000 015000
3375 015000 125252
3376 015002 015004
3377 015004 052525
3378
3379
3380 015006 000000
3381 015010 015006
3382 015016 015016
3383 015016 000000
3384 015020 015016
3385 015024 015024
3386 015024 015026
3387 015026 000000
3388

```

```

;*****
;RANDOM NUMBER GENERATOR
RNGEN:  MOV   RP1,RO
        ROL   RO
        ROL   RO
        ADD   RP2,RO
        MOV   RO,RP1
        ROL   RO
        ROL   RO
        ADD   RP2,RO
        ROL   RO
        ROL   RO
        MOV   RO,RP2
        MOV   RP1,RO
        RTS   PC
;RETURN.

RP1:    1233
RP2:    7622

;*****
.SBTTL  FIXED VALUES FOR USE IN TEST/
;*****
B:      125252
;ADDRESS OF B

A:      -1
        A+4
        .=A+4
        125252
        A+10
        052525
;ADDRESS OF A+10

;FOR STORAGE
C:      0
        C
        .=C+10
TEMP:   0
        TEMP
        .=TEMP+6
        TEMP+10
;ADDRESS OF TEMP+10 OR "D"

D:      0

```

```

3389
3390 ;*****
3391 .SBTTL ROUTINE SUBROUTINE TO INITIALIZE ALL PAGES TO NR, BANK 0, 1 PAGE, UP/
3392 ;*****
3393 015030 010146 NRALL: MOV R1,-(R6) ;SAVE REGISTERS
3394 015032 010246 MOV R2,-(R6)
3395 015034 010346 MOV R3,-(R6)
3396 015036 012701 000536 MOV #IPDRTAB,R1 ;R1 HOLDS ADDRESS OF CURRENT POSITION
3397 ;IN TABLE OF ADDRESSES
3398 015042 012703 000010 NRLLOOP: MOV #8,R3 ;R3 USED AS COUNTER
3399 015046 012102 MOV (R1)+,R2 ;R2 CONTAINS ADDRESS OF PDR OR
3400 ;PAR TO BE CLEARED
3401 015050 005022 CLR (R2)+ ;CLEAR ALL ASR'S FOR THIS MODE
3402 015052 077302 SOB R3,-2
3403 015054 020127 000544 CMP R1,#IPDREND ;CHECK FOR DONE
3404 015060 003770 BLE NRLLOOP ;CLEAR ALL IN NEXT MODE IF NOT DONE
3405 015062 012603 MOV (R6)+,R3
3406 015064 012602 MOV (R6)+,R2
3407 015066 012601 MOV (R6)+,R1
3408 015070 000207 RTS %7
3409
3410 ;*****
3411 .SBTTL EMT HANDLER/
3412 ;*****
3413 ;FIRST 3 CALLS LEFT OPEN IN TABLE FOR EASY PATCHES/
3414 015072 162716 000002 EMTSRV: SUB #2,@SP ;GET CALL
3415 015076 006576 000000 MFPI @SP
3416 015102 012667 000022 MOV (SP)+,EPC
3417 015106 062716 000002 ADD #2,@SP
3418 015112 105067 000013 CLRB EPC+1 ;SAVE OFFSET ONLY
3419 015116 062767 015132 000004 ADD #EMTAB,EPC ;POINT TO TABLE OF ADDRESSES
3420 015124 017707 000000 MOV @EPC,PC ;JUMP TO DESIRED ROUTINE
3421 015130 000000 EPC: 0
3422 000000 PATCH1=0
3423 000000 PATCH2=0
3424 000000 PATCH3=0
3425 015132 000000 EMTAB: PATCH1 ;PATCH IN ADDRESS OF ROUTINE
3426 015134 000000 PATCH2
3427 015136 000000 PATCH3
3428 015140 015664 PRINT ;ERROR PRINTOUT
3429 015142 015144 EOBSRV ;END OF BANK
3430
3431 ;*****
3432 .SBTTL ROUTINE END OF BANK SERVICE
3433 ;*****
3434
3435 015144 032767 000007 163022 EOBSRV: BIT #7,MMOPT ;MEM. MGMT./USER-KERNEL/4KAS 32 INHIBITED?
3436 015152 001406 BEQ EOB2 ;NO - CONTINUE
3437 015154 012716 015564 EOB1C: MOV #LOGIC,(SP) ;GO TO BEGIN
3438 015160 012766 000340 000002 MOV #340,2(SP) ;WILL ASSUME PRIORITY 7.
3439 015166 000002 RTI
3440 015170 042737 000340 177776 EOB2: BIC #340,@#PSR
3441 015176 026767 163372 163310 CMP CURPAR,UPAR7 ;LAST USER ASR DONE?
3442 015204 001444 BEQ NXTBNK ;YES - GO FIND NEXT BANK
3443 015206 062737 020000 000034 ADD #20000,@#34 ;UPDATE SCOPE VECTOR ADDRESS IN BANK 0
3444 015214 062767 020000 163356 ADD #20000,BNKSTR ;UPDATE BANK START TO REFERENCE CURRENT ASR

```



```

3445 015222 016716 163352      MOV      BNKSTR,(SP)
3446 015226 026767 163256 163340  CMP      UPARD,CURPAR
3447 015234 001404      BEQ      NXTSEG
3448 015236 005077 163332      CLR      @CURPAR      ;SET PREVIOUS ASR TO NR, BANK 0
3449 015242 005077 163330      CLR      @CURPDR
3450 015246 062767 000002 163320  NXTSEG: ADD      #2,CURPAR      ;UPDATE POINTERS TO NEXT SEGMENT
3451 015254 062767 000002 163314      ADD      #2,CURPDR
3452 015262 012777 077406 163306      MOV      #77406,@CURPDR      ;SET NEXT SEGMENT RW, 4K
3453 015270 016777 163274 163276      MOV      CURBNK,@CURPAR      ;MAP NEXT SEGMENT TO CURRENT BANK
3454 015276 052737 030000 177776      BIS      #30000,@PSR      ;SET PREVIOUS MODE TO USER
3455 015304 006506      MFPI     R6      ;PICK UP USER STACK POINTER
3456 015306 062716 020000      ADD      #20000,@R6      ;MAP IT TO NEXT ASR
3457 015312 006606      MTPI     R6      ;PUT IT BACK
3458 015314 000002      RTI
3459 015316 012746 000400      NXTBNK: MOV      #UBUFF,-(SP)
3460 015322 052737 030000 177776      BIS      #30000,@PSR
3461 015330 006606      MTPI     R6
3462 015332 013737 000570 000572      MOV      @#CURBNK,@#OLDBNK      ;SAVE PREV BANK ADDRESS
3463 015340 062767 000200 163222  BNKTST: ADD      #200,CURBNK
3464 015346 006367 163236      ASL      COREPT
3465 015352 103006      BCC      1$
3466 015354 012767 000001 163226      MOV      #1,COREPT
3467 015362 012767 000606 163222      MOV      #MEM1,MEMUT
3468 015370 022767 007600 163172  1$:  CMP      #7600,CURBNK      ;CHECK FOR EXTERNAL BANK
3469 015376 001666      BEQ      EOB3C      ;BR IF YES TO START ANOTHER PASS.
3470 015400 016777 163164 163124  EOB3:  MOV      CURBNK,@KPAR2      ;MAP KERNEL SEGMENT 2 TO BANK BEING LOOKED FOR
3471 015406 012777 077406 163106      MOV      #77406,@KPD2
3472 015414 036777 163170 163170      BIT      COREPT,@MEMUT
3473 015422 001746      BEQ      BNKTST
3474 015424 042737 160000 000034      BIC      #160000,@#34      ;INITIALIZE SCOPE VECTOR ADDRESS
3475 015432 005001      CLR      R1      ;R1 ADDRESSES BANK 0 THRU KERNEL ASR0
3476 015434 012702 040000      MOV      #40000,R2      ;R2 ADDRESSES NEW BANK THRU KERNEL ASR2
3477 015440 012703 015026      MOV      #0,R3
3478 015444 006203      ASR      R3
3479 015446 012122      CORMOV: MOV      (R1)+(R2)+
3480 015450 077302      SOB      R3,CORMOV
3481 015452 016767 163032 163114      MOV      UPARD,CURPAR      ;FIRST ASR CHECKED IS USER ASR0
3482 015460 016767 163016 163110      MOV      UPDR,CURPDR
3483 015466 016777 163076 163100      MOV      CURBNK,@CURPAR
3484 015474 012777 077406 163074      MOV      #77406,@CURPDR
3485 015502 005077 163006      CLR      @UPAR7
3486 015506 005077 162774      CLR      @UPDR7
3487 015512 026727 163054 000000      CMP      OLDBNK,#0      ;PREV BANK = 0
3488 015520 001414      BEQ      EOB6      ;YES, DO NOT CLEAR
3489 015522 016777 163044 163002      MOV      OLDBNK,@KPAR2
3490 015530 012777 077406 162764      MOV      #77406,@KPD2
3491 015536 012701 040000      MOV      #40000,R1
3492 015542 012703 007630      MOV      #7630,R3
3493 015546 005021      BNKLP:  CLR      (R1)+
3494 015550 077302      SOB      R3,BNKLP
3495 015552 012716 005542      EOB6:  MOV      #BEGIN,(SP)
3496 015556 011667 163016      MOV      (SP),BNKSTR
3497 015562 000002      RTI

```

\*\*\*\*\*  
.SBTTL END OF PASS CODE STARTS HERE

3498  
3499  
3500

```

3501
3502 015564 042777 000001 162706
3503 015572 012737 000016 000014
3504 015600 005037 000016
3505 015604 032737 000001 000176
3506 015612 001404
3507 015614 004767 000444
3508 015620 004767 000414
3509 015624 005267 177056
3510
3511
3512 015630 013701 000042
3513 015634 001405
3514 015636 000005
3515 015640 004711
3516 015642 000240
3517 015644 000240
3518 015646 000240
3519
3520 015650 000005
3521 015652 005167 162724
3522 015656 000137 000634
3523

```

```

;*****
LOGIC: BIC #1,JSR0 ;TURN OFF MEMORY MANAGEMENT.
      MOV #16,@#14 ;RESET THE TRACE VECTOR.
      CLR @#16
      BIT #1,@#SREG2 ;TTY OUT SELECTED
      BEQ 1$ ;YES, NO ASTERISK
      JSR PC,BELL ;RING BELL TOO.
      JSR PC,STAR ;TYPE ASTERISK.
1$: INC PASCNT ;INCREMENT PASS COUNT.
;*****START OF "ACT11/XXDP EOP HOOKS"*****
      MOV @#42,R1
      BEQ HERE
SENDAD: JSR %7,@R1
      NOP
      NOP
      NOP
;*****END OF "ACT11/XXDP EOP HOOKS"*****
HERE: RESET ;ISSUE RESET TO HALT I/O.
      COM TRPB ;COMPLEMENT THE TRACE SWITCH.
      JMP @#RSTRT ;RESTART.

```

```

3524
3525
3526
3527 015662 000006
3528
3529
3530
3531
3532
3533
3534
3535 015664 005767 000170
3536 015670 001401
3537 015672 000002
3538 015674 005267 000160
3539 015700 012767 000340 162070
3540 015706 036727 162634 020000
3541 015714 001044
3542 015716 012667 000132
3543 015722 012667 000130
3544 015726 024646
3545 015730 012767 000200 162040
3546 015736 004767 000342
3547 015742 016767 000106 000264
3548 015750 004767 000106
3549 015754 004767 000272
3550 015760 016767 000072 000246
3551 015766 004767 000070
3552 015772 004767 000254
3553 015776 016767 162566 000230
3554 016004 004767 000052
3555 016010 004767 000236
3556 016014 016767 176664 000212
3557 016022 004767 000034
3558 016026 023727 000042 015640 CK:
3559 016034 001403
3560 016036 005767 162504
3561 016042 100001
3562 016044 000000 1$:
3563 016046 005067 000006
3564 016052 000002
3565 016054 000000 SAVPC: 0
3566 016056 000000 SAVPSR: 0
3567 016060 000000 PRTON: 0
3568
3569
3570
3571
3572
3573
3574 016062 012727 000006 016066
3575 016066 005067 000136
3576 016070 012767 000060 000134
3577 016074 005767 000126
3578 016102 100002
3579 016106

```

```

;*****
.SBTTL RTT EXECUTED WHEN TRACE IS ON/
;*****
TRTRP: RTT

;*****
.SBTTL ROUTINE PRINT
;*****
;ENTERED WITH SYSTEM TRAP CALL (HLT)
;PRINT OUT THE ERROR PC+2, STATUS REGISTER, AND LOCATION IN BACKGROUND
PRINT: TST PRTON ;CHECK PRINT ON FLAG
      BEQ .+4 ;IF ANOTHER HALT IS BEING PRINTED, SKIP THIS ONE
      RTI
      INC PRTON
      MOV #340,PSR ;SET PRIORITY TO 7
      BIT SR,#20000 ;TEST FOR INHIBIT PRINT OUT
      BNE CK ;BR TO INHIBIT PRINT.
      MOV (6)+,SAVPC ;PC OF FAILING ROUTINE
      MOV (6)+,SAVPSR ;PSR OF ERROR CONDITION
      CMP -(6),-(6) ;RESTORE STACK
      MOV #200,PSR
      JSR %7,CRLF ;OUTPUT CARRIAGE RETURN AND LINE FEED
      MOV SAVPC,PTEMP1 ;LOAD WITH FAILING PC+2
      JSR %7,PROCT ;PRINT FAILING PC+2
      JSR %7,SPACE
      MOV SAVPSR,PTEMP1 ;LOAD PROCESSOR STATUS
      JSR %7,PROCT ;PRINT PROCESSOR STATUS
      JSR %7,SPACE
      MOV CURBNK,PTEMP1
      JSR %7,PROCT
      JSR %7,SPACE
      MOV RETURN,PTEMP1
      JSR %7,PROCT
      CK: CMP #42,$SENDAD ;IN ACT11?
          BEQ 1$ ;BR IF YES TO HALT.
          TST SR ;CHECK SR FOR HALT SWITCH
          BPL .+4 ;BRANCH IF NOT SET
          HALT ;HALT ON ERROR UP
          CLR PRTON ;ROUTINE DONE - CLEAR FLAG
          RTI ;RETURN TO MAIN LINE

SAVPC: 0
SAVPSR: 0
PRTON: 0

```

```

;*****
.SBTTL ROUTINE PROCT
;*****
;SUBROUTINE TO PRINT OUT OCTAL NUMBER/
PROCT: MOV #6,#PTEMP3 ;CLEAR R4 FOR COUNTING CHARACTERS OUTPUT
      PTEMP3=-2
      CLR PRFLG ;INITIALIZE CARRY FLAG FOR ROTATES
      MOV #60,PTEMP2 ;SETUP R3
      TST PTEMP1 ;CHECK BIT 15 OF NUMBER
      BPL .+6 ;BRANCH IF ZERO

```

```

3580 016110 005267 000122      INC      PTEMP2      ;INCREMENT R3 IF ONE
3581 016114 006167 000114      ROL      PTEMP1      ;ROTATE LEFT MOST OCTAL TO RIGHT END
3582 016120 006167 000110      ROL      PTEMP1
3583 016124 005567 000102      ADC      PRFLG
3584 016130 016746 000102      P.WAIT: MOV     PTEMP2,-(SP) ;STORE CARRY
3585 016134 004767 000210      JSR      PC,CHROUT ;OUTPUT THE CHARACTER
3586 016140 005367 177722      DEC      PTEMP3      ;DO IT.
3587 016144 001001          BNE      P.CNT1      ;COUNT
3588 016146 000207          RTS
3589 016150 000241          P.CNT1: CLC
3590 016152 005767 000054      TST      PRFLG
3591 016156 001403          BEQ      .+10
3592 016160 005067 000046      CLR      PRFLG
3593 016164 000261          SEC
3594 016166 006167 000042      ROL      PTEMP1      ;CLEAR CARRY
3595 016172 006167 000036      ROL      PTEMP1      ;CHECK FOR PREVIOUS CARRY
3596 016176 006167 000032      ROL      PTEMP1      ;BRANCH IF NOT DONE
3597 016202 005567 000024      ADC      PRFLG
3598 016206 016767 000022 000022  MOV     PTEMP1,PTEMP2 ;STORE CARRY
3599 016214 042767 177770 000014  BIC     #177770,PTEMP2 ;LOAD DATA INTO R3
3600 016222 052767 000060 000006  BIS     #60,PTEMP2    ;CLEAR ALL BUT LOWEST OCTAL DIGIT
3601 016230 000737          BR      P.WAIT      ;SET TO ASCII EQUIVALENT
3602 016232 000000          PRFLG: 0
3603 016234 000000          PTEMP1: 0
3604 016236 000000          PTEMP2: 0
3605
3606
3607
3608 ;*****
3609 ;SBTTL ROUTINE STAR
3610 ;*****
3611 ;SUBROUTINE TO OUTPUT ASTERISK.
3612 STAR: JSR      PC,CRLF ;OUTPUT CRLF.
3613       MOV     #52,-(SP) ;GO OUTPUT A *
3614       BR      BELL1
3615
3616 ;*****
3617 ;SBTTL ROUTINE SPACE
3618 ;*****
3619 ;SUBROUTINE TO ISSUE SPACE/
3620 SPACE: MOV     #40,-(SP) ;OUTPUT SPACE.
3621       JSR      PC,CHROUT ;DO IT.
3622       RTS     %7 ;RETURN
3623
3624 ;*****
3625 ;SBTTL ROUTINE BELL
3626 ;*****
3627 ;BELL ON PASS COMPLETE
3628 BELL:  MOV     #7,-(SP) ;OUTPUT BELL.
3629       JSR      PC,CHROUT ;DO IT.
3630       DEC     #0 ;SLIGHT DELAY.
3631       BNE    BELL2
3632       BELL2: RTS     %7
3633
3634
3635

```

ROUTINE BELL

```

3636 ;*****
3637 ;SBTTL ROUTINE CRLF
3638 ;*****
3639 ;SUBROUTINE TO OUTPUT CARRIAGE RETURN AND LINEFEED/
3640 016304 012746 000015 CRLF: MOV #15,-(SP) ;OUTPUT CR.
3641 016310 004767 000034 JSR PC,CHROUT ;DO IT.
3642 016314 012746 000012 MOV #12,-(SP) ;OUTPUT LF.
3643 016320 004767 000024 JSR PC,CHROUT ;DO IT.
3644 016324 016746 162272 MOV FILLCT,-(SP) ;GET THE FILL COUNT.
3645 016330 001405 BEQ 2$ ;BR IF 0.
3646 016332 005046 1$: CLR -(SP) ;WILL OUTPUT NULLS FOR FILLERS.
3647 016334 004767 000010 JSR PC,CHROUT ;DO IT.
3648 016340 005316 DEC (SP) ;DONE?
3649 016342 001373 BNE 1$ ;BR IF NOT.
3650 016344 005726 2$: TST (SP)+ ;CLEAN UP STACK.
3651 016346 000207 RTS PC ;RETURN.
3652
3653
3654 ;*****
3655 ;SBTTL ROUTINE CHROUT
3656 ;*****
3657 ;SUBROUTINE TO OUTPUT CHARACTER TO CONSOLE TTY.
3658 016350 016677 000002 162034 CHROUT: MOV 2(SP),@TDBR ;LOAD THE CONSOLE BUFFER REG.
3659 016356 105777 162026 1$: TSTB @TCSR ;READY?
3660 016362 100375 BPL 1$ ;BR IF NOT. WAIT.
3661 016364 012616 MOV (SP)+,(SP) ;SET UP FOR EXIT.
3662 016366 000207 RTS PC ;RETURN.
3663
3664 ;ENTER HERE ON POWER FAIL/
3665 016370 013746 000024 PFAIL: MOV @#24,-(6) ;STORE STACK POSITION
3666 016374 010667 000010 MOV %6,SAVR6 ;HALT ON POWER DOWN NORMAL
3667 016400 012737 016412 000024 MOV #RESTRT,@#24 ;STACK IS SAVED HERE
3668 016406 000000 HALT ;RESTORE STACK WHEN POWERING UP
3669 016410 000000 SAVR6: 0
3670 016412 016706 177772 RESTRT: MOV SAVR6,%6
3671 016416 012637 000024 MOV (6)+,@#24
3672 016422 022626 CMP (SP)+,(SP)+ ;RESTORE STACK
3673 016424 104006 HLT ;POWER FAIL OCCURRED
3674 016426 000167 162202 JMP RSTRT ;RETURN TO MAIN LINE
3675
3676
3677 016432 000207 USER: RTS %7 ;OVERLAY USER ROUTINE HERE IF 4KW
3678 ;USE BANK1 IF 9KW
3679 017760 017760
3680 017760 000000
3681 017760 000001
KSTACK: 0
.END

```

















G07

MAIN	632#	1900													
ROUTIN	632#	3344	3389	3431	3529	3569	3606	3615	3624	3635	3653				
SECTIO	569	612	632#	765	791	801	883	892	902	976	984	1000	1011	1025	1037
	1052	1116	1124	1208	1224	1248	1290	1311	1346	1931	1972	1990	2023	2119	2136
	2154	2175	2199	2222	2246	2284	2302	2323	2367	2391	2445	2467	2497	2562	2593
	2628	2713	2726	2741	2780	2794	2815	2839	2889	2905	2959	2964	2983	2998	3075
	3083	3093	3126	3185	3204	3218	3232	3246	3261	3277	3291	3299	3314	3365	3410
	3499	3524													
STARS	569	571	612	614	632#	634	765	767	791	793	801	803	883	885	892
	894	902	904	976	978	984	986	1000	1002	1011	1013	1025	1027	1037	1039
	1052	1054	1116	1118	1124	1126	1208	1210	1224	1226	1248	1250	1290	1292	1311
	1313	1346	1348	1900	1902	1931	1933	1972	1974	1990	1992	2023	2025	2119	2121
	2136	2138	2154	2156	2175	2177	2199	2201	2222	2224	2246	2248	2284	2286	2302
	2304	2323	2325	2367	2369	2391	2393	2445	2447	2467	2469	2497	2499	2562	2564
	2593	2595	2628	2630	2713	2715	2726	2728	2741	2743	2780	2782	2794	2796	2815
	2817	2839	2841	2889	2891	2905	2907	2959	2961	2964	2966	2983	2985	2998	3000
	3075	3077	3083	3085	3093	3095	3126	3128	3185	3187	3204	3206	3218	3220	3232
	3234	3246	3248	3261	3263	3277	3279	3291	3293	3299	3301	3314	3316	3345	3347
	3365	3367	3390	3392	3410	3412	3432	3434	3499	3501	3524	3526	3530	3532	3570
	3572	3607	3609	3616	3618	3625	3627	3636	3638	3654	3656				
TNCV	3083#	3105	3141	3165											

. ABS. 017762 000

ERRORS DETECTED: 0  
DEFAULT GLOBALS GENERATED: 0

DSKZ:DFKTGB,DSKZ:DFKTGB.SEQ/SOL/CRF=DFKTGB  
RUN-TIME: 10 23 3 SECONDS  
RUN-TIME RATIO: 135/37=3.6  
CORE USED: 8K (15 PAGES)