

.REM *

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCKTG-D-D
PRODUCT NAME: KT11-C EXERCISER
DATE CREATED: MARCH 21, 1975
MAINTAINER: DIAGNOSTIC PROGRAMMING
AUTHOR: RICK FADDEN

COPYRIGHT (c) 1972, 1973, 1975
DIGITAL EQUIPMENT CORPORATION

"The material in this document is for information purposes only and is subject to change without notice. Digital Equipment Corporation assumes no responsibility for the use of software on equipment which is not supplied by it. Digital Equipment Corporation assumes no responsibility for any errors which may appear in the document."

1.0 ABSTRACT

THIS PROGRAM IS AN INTERACTIVE EXERCISER FOR A PDP-11/45 EQUIPPED WITH THE KT11-C OPTION. IT PERFORMS A TEST OF INSTRUCTIONS AND CONCURRENT OPERATIONS OF I/O EQUIPMENT WHILE RELOCATING THRU MEMORY. IT PROVIDES NUMEROUS MODES OF TESTING, FROM 4K EXECUTION WITH THE KT11-C TURNED OFF AND ONLY KERNEL MODE IN USE, TO 128K EXECUTION WITH EACH USER PAGE MAPPED SEQUENTIALLY TO EVERY 4K BANK OF MEMORY, TC11 AND RF11 BUFFER AND CODE RELOCATION THRU ALL MEMORY, AND SUPERVISOR MAPPING OF OTHER I/O DEVICES. 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-11/45 STANDARD COMPUTER
KT11-C MEMORY MANAGEMENT OPTION

2.1.1 OPTIONAL HARDWARE THAT THE PROGRAM WILL EXERCISE

MEMORY UP TO 128KW OF MEMORY-DOES NOT HAVE TO BE CONTIGUOUS,
BUT BLOCKS OF LESS THAN 4KW WILL NOT BE USED
RF11 DISK
TC11 DECTAPE-TRANSPORT ZERO
KW11-L LINE CLOCK
KL11 ASR33 OR ASR35 TELEPRINTER
LP11 LINE PRINTER

2.2 STORAGE

THIS PROGRAM USES MEMORY FROM 00000 TO 17760.

3.0 LOADING PROCEDURE

PROCEDURE FOR NORMAL ABSOLUTE TAPES SHOULD BE FOLLOWED.

4.0 STARTING PROCEDURE AND SWITCH SETTINGS

4.1 NORMAL STARTING PROCEDURE:

LOAD STARTING ADDRESS 200, (300 IF LP11 IS USED).
SET DESIRED MEMORY MANAGEMENT SELECTION SWITCHES (SEE 4.2)- ALL
DOWN FOR WORST CASE TESTING.
PRESS START.

(CONTINUED ON THE NEXT PAGE)

4.1 NORMAL STARTING PROCEDURE (CONTINUED)

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

THE PROGRAM WILL PRINT A \$ (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).

NOTE THAT IF TTY OUTPUT IS SELECTED, THE DOLLAR SIGN AND ASTERISK ARE NOT PRINTED.

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

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

SW0#1 OR UP---INHIBIT THE KI11-C (SR0<0> WILL NOT BE SET AT ALL)

SW1#1 OR UP---INHIBIT USE OF SUPERVISOR AND USER MODES
(ALSO INHIBITS 4K AS 32K)

SW2#1 OR UP---INHIBIT 4K AS 32 K (ALSO INHIBITED IF EITHER SW0 OR SW1 IS SET)-SEE SECTION 5.3,1 FOR EXPLANATION

SW3#1 OR UP---INHIBIT RELOCATION OF RF11 AND TC11 CODE AND BUFFERS-SEE SECTIONS 5.3.4 AND 5.3.5 FOR AN EXPLANATION

SW4#1 OR UP---INHIBIT CYCLING ACCESS KEY OF SUPERVISOR PAGE 7
-SEE SECTION 5.3.7 FOR AN EXPLANATION

SW5#1 OR UP---INHIBIT VARIABLE CORE EXPANSION

00 OR DOWN-CORE EXPAND UNLESS SW0, 1 AND 2 ARE ALL DOWN
(IN WHICH CASE 4K AS 32K IS RUN INSTEAD)

4.3 DEVICE SELECTION SWITCHES

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

SW0#1 OR UP---INHIBIT TTY OUTPUT

SW4#1 OR UP---INHIBIT LINE CLOCK

SW5#1 OR UP---INHIBIT LINE PRINTER

SW6#1 OR UP---INHIBIT RF11 DISK

SW7#1 OR UP---INHIBIT TC11 DECTAPE

4.4 RESTART PROCEDURE

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

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

5.1.1 BASIC SWITCH SETTINGS-STARTUP

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

5.1.2 DYNAMIC SWITCH SETTINGS

THE FOLLOWING SWITCHES ARE RECHECKED PERIODICALLY DURING PROGRAM EXECUTION:

SW15=1 OR UP---HALT ON ERROR
SW14=1 OR UP---SCOPE LOOP
SW13=1 OR UP---INHIBIT ERROR PRINT OUT
SW12=1 OR UP---INHIBIT TRACE TRAPPING
SW11=1 OR UP---INHIBIT SUBTEST ITERATION AND INHIBIT TESTS WHICH USE ALL COMBINATIONS OF NUMBERS (SEE 5.2.1)
SW10=1 OR UP---INHIBIT PROCESSOR TEST (ONCE SET, PROCESSOR TEST IS PERMANENTLY INHIBITED)
SW09=1 OR UP---INHIBIT CYCLING SUPERVISOR MAPPING (SUPERVISOR PAGE CURRENTLY MAPPED WILL BE USED UNTIL THE SWITCH IS SET =0)
SW08=1 OR UP---INHIBIT CYCLING SUPERVISOR PAGE 7 ACCESS KEY (THE KEY CURRENTLY IN USE WILL REMAIN IN USE UNTIL THE SWITCH IS SET =0)

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, THE CONTENTS OF THE CURRENT BANK COUNTER, AND THE STARTING ADDRESS (VIRTUAL) OF THE BACKGROUND PROCESSOR SUBTEST BEING EXECUTED AT THE TIME OF FAILURE. 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 KERNEL STACK ADDRESS. THE CONTENTS OF THE CURRENT KERNEL 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), EOBSRV (EOB CALL), AND RTIKEX (RTIK CALL).

5.2.5 RTIKEX (RTI VIA KERNEL)

SINCE I/O EXECUTION IN SUPERVISOR IS DONE BY MANY OF THE ROUTINES IN THIS EXERCISER, A MEANS IS REQUIRED TO RETURN FROM THE INTERRUPTS TO THE MODE THAT WAS IN EFFECT WHEN THE INTERRUPT OCCURRED. THE CALL RTIK IS USED TO DO THIS.

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, THEIR POINTERS AND BUFFER ADDRESSES MUST BE CHANGED. THIS ROUTINE PERFORMS THESE OPERATIONS.

5.2.8 ABSRV (KT11-C ABORT AND TRAP HANDLER)

IF CYCLING OF THE ACCESS KEY FOR SUPERVISOR PAGE 7 IS ALLOWED, I/O REFERENCES TO THE EXTERNAL BANK WILL SOMETIMES ABORT OR TRAP. THIS ROUTINE HANDLES THESE ABORTS AND TRAPS.

5.2.9 PFAIL (POWER FAIL)

THE POWER FAIL ROUTINE JUMPS TO THE RESTART ROUTINE ON POWER UP, AFTER PRINTING AN ERROR PRINTOUT.

5.2.10 TYOUT (TTY OUTPUT)

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

5.2.11 RFASTART (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 ON THE DISK. AFTER THE TOTAL DISK HAS BEEN WRITTEN, A WRITE CHECK IS USED TO VERIFY THAT THE DATA HAS BEEN WRITTEN CORRECTLY ON THE DISK. NOTE THAT NO "DATA" 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.

5.2.12 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.

5.2.13 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, IT IS CALLED BY A PRIORITY INTERRUPT REQUEST ON LEVEL 2,

5.2.14 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.

5.2.15 NXTBLK (TC11 HEAD 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,

5.2.16 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.

5.2.17 LCLK (LINE CLOCK)

THIS TEST OF THE LINE CLOCK IS IN THE INTERRUPT MODE, IF OPERATING CORRECTLY THE ROUTINE WILL OUTPUT A MINUTES ELAPSED COUNT TO THE HIGH BYTE OF THE DISPLAY REGISTER, THE MINUTE COUNT IS BASED ON A 60 CYCLE LINE FREQUENCY.

5.2.18 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.

5.2.19 CORE EXPANSION (DET1)

THIS ROUTINE IS CONTROLLED BY SWITCH 5 AT STARTUP, IF CALLED, THE PROCESSOR BACKGROUND CODE WILL BE EXPANDED THRU AVAILABLE MEMORY (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.

SEE 5.3.2 FOR A FURTHER EXPLANATION OF MODES OF EXECUTION USING CORE EXPANSION.

5.2.20 PIRSRV (PRIORITY INTERRUPT REQUEST SERVICE ROUTINE)

TC11 DATA BUFFER GENERATION AND CHECKING LENDS ITSELF TO HANDLING AT A LEVEL LOWER THAN THAT OF THE TC11 INTERRUPT, THE PRIORITY INTERRUPT REQUEST PROVIDES THE ABILITY TO DO THIS. THUS, SOFTWARE REQUESTS ARE USED IN THE DECTAPE CODE TO CALL THESE ROUTINES, THE DATA CHECK IS CALLED BY A LEVEL 3 REQUEST, AND THE BUFFER GENERATION ROUTINE IS CALLED BY A LEVEL 2 REQUEST, THE PIRSRV ROUTINE HANDLES THESE TWO PIRQ INTERRUPTS,

5.2.21 TRTRP

THE TRACE BIT IS NORMALLY SET ON ALTERNATE PASSES THRU THE BACKGROUND PROCESSOR TESTS (IT IS SET DURING THE EVEN NUMBERED PASSES). IF THE TRACE BIT IS SET, THIS ROUTINE'S ADDRESS IS LOADED IN THE TRACE TRAP VECTOR, IT SIMPLY DOES AN RTI BACK TO THE PROCESSOR TESTS, SWITCH 12 SET TO A ONE WILL INHIBIT THE USE OF THE TRACE BIT WITH THE BACKGROUND PROCESSOR TESTS.

5.3 PROGRAM AND/OR OPERATOR ACTION

5.3.1 PROCESSOR TEST EXECUTION - 4K AS 32K

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

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

5.3.2 PROCESSOR TEST EXECUTION - CORE EXPANSION

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

5.3.3 PROCESSOR TEST EXECUTION - BANK 0 ONLY

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

5.3.4 RF11 CODE AND BUFFER RELOCATION

IF SW3 IS DOWN AT STARTUP AND THE RF11 IS NOT INHIBITED, THE RF11 CODE WILL BE EXECUTED FROM THE SAME BANK THAT THE PROCESSOR TESTS ARE RUNNING IN. THE PHYSICAL MEMORY ADDRESS USED BY THE RF11 FOR WRITE AND WRITE CHECK OPERATIONS WILL ALSO BE IN THE SAME BANK AS THE PROCESSOR TESTS ARE RUNNING IN. THUS THE EXTENDED MEMORY ADDRESS BITS WILL GET SET IF THE PROCESSOR TESTS ARE RUNNING ABOVE 32K. THIS RELOCATION OCCURS IF CORE EXPANSION IS SELECTED (EVEN IF THE KT11-C IS INHIBITED) OR IF THE PROCESSOR TESTS ARE BEING RUN IN THE 4K AS 32K MODE.

NOTE THAT IF THE 4K AS 32K MODE IS USED, THE RF11 WILL BE MAPPED TO KERNEL PAGE 1, AND KERNEL PAGE 1 IS MAPPED TO THE CURRENT BANK. OTHERWISE, RELOCATION IS DONE BY MAPPING THE RF11 TO THE CURRENT BANK DIRECTLY (IN WHICH CASE VIRTUAL AND PHYSICAL ADDRESSES ARE MAPPED EQUAL IF KT11-C IS ON).

5.3.5 TC11 CODE AND BUFFER RELOCATION

IF SW3 IS DOWN AT STARTUP AND IF THE TC11 IS NOT INHIBITED, THE TC11 CODE WILL BE EXECUTED FROM THE SAME BANK THAT THE PROCESSOR TESTS ARE RUNNING IN. THE READ AND WRITE BUFFERS FOR THE TC11 WILL ALSO BE LOCATED IN THE BANK IN WHICH THE PROCESSOR TESTS ARE RUNNING. DUE TO THE COMPLEXITY OF RELOCATING THE TC11 CODE AND BUFFERS, RELOCATION IS DONE ONLY WHEN THE DECTAPE IS AT THE FORWARD END ZONE. THUS, BEFORE THE BANK IN WHICH THE PROCESSOR TESTS ARE RUNNING IS CHANGED, THE CODE WAITS FOR THE DECTAPE TO FINISH A PASS OF WRITING AND READING THE TAPE. A PASS THRU ALL OF MEMORY RUNNING TC11 RELOCATION WILL REQUIRE THE DECTAPE TO MAKE AS MANY PASSES AS THERE ARE BANKS OF MEMORY.

NOTE THAT IF THE 4K AS 32K MODE IS USED, THE TC11 WILL BE MAPPED TO KERNEL PAGE 1, AND KERNEL PAGE 1 IS MAPPED TO THE CURRENT BANK. OTHERWISE, RELOCATION IS DONE BY MAPPING THE TC11 TO THE CURRENT BANK DIRECTLY (IF KY11-C IS ON, VIRTUAL ADDRESSES ARE MAPPED TO EQUAL THE PHYSICAL ADDRESSES).

5.3.6 EXECUTION OF I/O CODE IN SUPERVISOR

IF SW1 IS DOWN AT STARTUP, THE FOLLOWING DEVICES ARE SERVICED IN SUPERVISOR MODE: TELETYPE OUTPUT, LINE CLOCK, AND LINE PRINTER. IF MEMORY MANAGEMENT IS NOT INHIBITED, THESE DEVICES' SERVICE ROUTINES EXECUTE THRU SUPERVISOR PAGE 0, THEN PAGE 1, AND SO ON, SEQUENTIALLY UP TO SUPERVISOR PAGE 6. THE PAGE BEING ACCESSED IS CHANGED EACH TIME THE END OF BANK CODE (EOBSRV ROUTINE) IS EXECUTED, BY CHANGING THE ADDRESSES STORED IN THE DEVICES' INTERRUPT VECTORS. WHEN SUPERVISOR PAGE 6 WAS THE ONE JUST USED, THE CYCLE STARTS OVER WITH PAGE 0. PAGE 7 IS USED FOR ACCESSING THE EXTERNAL BANK. A DYNAMIC SWITCH (SW09) IS PROVIDED TO INHIBIT CYCLING THE SUPERVISOR MAPPING. IF THIS SWITCH IS SET, THE SUPERVISOR CODE WILL REMAIN MAPPED TO THE PAGE IT WAS MAPPED TO WHEN THE SWITCH WAS SET.

5.3.7 EXECUTION OF I/O CODE IN SUPERVISOR - USE OF SUPERVISOR PAGE 7

IF SW1 IS DOWN AT STARTUP, THE DEVICES LISTED IN 5.3.6 WILL BE SERVICED IN SUPERVISOR MODE, AS EXPLAINED PREVIOUSLY. IF SW4 IS ALSO DOWN AT STARTUP, SUPERVISOR PAGE 7 (MAPPED TO THE EXTERNAL BANK) WILL HAVE ITS ACCESS KEY CHANGED EACH TIME THE END OF BANK CODE (EQBSRV) IS EXECUTED. THE ACCESS KEY WILL BE SET FIRST TO 0, THEN 1, AND SO ON UP THRU 7. AFTER BEING SET TO 7, IT WILL BE RECYCLED TO 0 AND THE CYCLE WILL BE REPEATED. DEPENDING ON THE KEY IN EFFECT AT THE TIME, THE SUPERVISOR I/O MAY TRAP, ABORT, OR EXECUTE WHEN REFERENCING THE EXTERNAL PAGE. IF A TRAP OR ABORT OCCURS THE KERNEL ROUTINE ABSRV IS ENTERED, THIS ROUTINE PERFORMS CERTAIN CHECKS ON THE TRAP OR ABORT, AND IF IT WAS AN ABORT, IT REEXECUTES THE INSTRUCTION IN KERNEL MODE (MAPPED R/W), TO REGAIN CONTROL AFTER THE INSTRUCTION IS COMPLETED, THE ROUTINE SETS THE T-BIT AND CHANGES THE TRACE TRAP RETURN TO POINT TO A SECTION OF THE ABORT SERVICE CODE. AFTER THE INSTRUCTION IS COMPLETED, A TRACE TRAP TO THE KERNEL ABORT SERVICE CODE OCCURS. THE CODE RESTORES THE PREVIOUS TRACE-TRAP VECTOR, RESTORES THE T-BIT TO ITS PREVIOUS STATE, AND RETURNS TO THE SUPERVISOR DEVICE SERVICE CODE AT THE INSTRUCTION AFTER THE ONE WHICH ABORTED.

NOTE THAT SWITCH 0 UP WILL INHIBIT THE CYCLING OF THE ACCESS KEY DYNAMICALLY, LEAVING IT SET TO THE CURRENT VALUE.

6.0 ERRORS

6.1 ERROR PRINTOUT

PRINTOUTS ARE IN AN EXTENDED VERSION OF THE STANDARD FORMAT, USING FOUR 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 16 BIT ADDRESS OF THE BANK BEING CURRENTLY USED FOR EXECUTION OF THE PROCESSOR TESTS (AND R/11 AND T/11 CODE IF IT IS BEING RELOCATED), 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), THE FOURTH WORD IS THE PC (VIRTUAL) OF THE CURRENT PROCESSOR SUBTEST BEING EXECUTED IN BACKGROUND.

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.

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. THE CONTENTS OF RETURN IN THE CURRENT BANK IS THE FOURTH WORD OF THE ERROR PRINTOUT.

7.0 RESTRICTIONS

PROGRAM MUST BE LOADED INTO THE LOWER 4K OF MEMORY.

IF THE LINE PRINTER IS USED, STARTING ADDRESS 300 MUST BE USED INSTEAD OF SA 200.

8.0 MISCELLANEOUS

8.1 EXECUTION TIME

EXECUTION TIME VARIES WITH THE AMOUNT OF MEMORY, THE TYPES OF MEMORY, THE DEVICES RUN, 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, THRU 128K OF CORE MEMORY, WITH NO I/O TAKES ABOUT 15 MINUTES. (AN ASTERISK IS PRINTED AT THE END OF A FULL PASS, AND A DOLLAR SIGN IS PRINTED AT THE END OF EACH 4K BANK).

A PASS RUN WITH 4K AS 32K, WITH DECTAPE RELOCATION, TAKES ABOUT 1 MINUTE PER 4K BANK, IF ITERATIONS ARE INHIBITED (SW11 IS UP).

8.2 STACK POINTERS

THE KERNEL STACK POINTER IS INITIALIZED TO 17700.

THE SUPERVISOR STACK POINTER IS INITIALIZED TO 600. IT IS RELOCATED TO THE CURRENT SUPERVISOR PAGE.

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 MONITORING PHYSICAL AND VIRTUAL ADDRESSES

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

8.4 DISPLAY REGISTER

THE LOWER BYTE OF THE DISPLAY REGISTER SHOWS THE PASS COUNT. THE UPPER BYTE IS USED WHEN THE KW11-L LINE CLOCK IS RUN TO INDICATE MINUTES ELAPSED SINCE THE PROGRAM WAS STARTED (BASED ON A LINE FREQUENCY OF 60 CYCLES PER SECOND).

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. THESE LEVELS ARE AS FOLLOWS:

	KT11-C ON	CORE EXPAN (SEE 5.3.2)	4K AS 32 (TO 128K) (SEE 5.3.1)	USER/SUPER KERNEL	RF/TC RELOC (SEE 5.3.4 AND 5.3.5)	VARIABLE USE OF SUPER (SEE 5.3.6 AND 5.3.7)
1						
2				X		
3		X				
4		X		X		
5		X			X	
6		X		X	X	
7	X					
8	X			X		X
9	X	X				
10	X	X		X		X
11	X	X			X	
12	X	X		X	X	X
13	X		X	X		X
14	X		X	X	X	X

NOTE THAT WHERE VARIABLE USE OF THE SUPERVISOR SPACE IS ALLOWED,
THERE ARE FOUR POSSIBLE LEVELS FOR ITS USE:

1. ALL EXECUTION OCCURS THRU SUPERVISOR PAGE 0,
WITH PAGE 7 MAPPED RW FOR I/O REFERENCES.
2. EXECUTION OCCURS THRU SUPERVISOR PAGES 1
THRU 6, ONE AT A TIME, WITH PAGE 7 MAPPED
RW FOR I/O REFERENCES.
3. ALL EXECUTION OCCURS THRU SUPERVISOR PAGE 0
WITH PAGE 7 CYCLED THRU ALL VALUES OF THE
ACCESS KEY.
4. EXECUTION OCCURS THRU SUPERVISOR PAGES 1 THRU 6,
ONE AT A TIME, WITH THE ACCESS KEY FOR PAGE 7 CYCLED
THRU ALL VALUES.

THERE IS NO MONITOR IN THE CONVENTIONAL SENSE. EACH DEVICE THAT IS
TO BE EXERCISED HAS ITS OWN STAND ALONE ROUTINE THAT OPERATES IN THE
INTERRUPT MODE. THESE ROUTINES NEED NO SUPERVISION OR MONITORING
AFTER THEY ARE INITIATED. THERE IS A PRIMER AREA THAT CHECKS THE
SWITCH REGISTER TO SEE WHAT DEVICES ARE TO BE INITIATED. IT SETS
THE INTERRUPT ENABLE BIT IN THE DEVICE STATUS REGISTER, INITIALIZES
THE DATA PATTERN, AND INITIATES AN OPERATION TO RAISE DATA FLAGS
ON DEVICES THAT CAN NOT INITIATE THEM THEMSELVES. THE PRIMER CODE
THEN ENTERS THE KT11-C SETUP CODE. THE RF11 AND TC11 PRIMER CODE
IS IN WITH THE KT11-C SETUP CODE SINCE THESE DEVICES, AS CODED
REQUIRE CERTAIN PARTS OF THE KT11-C 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.
ALL CYCLING FEATURES OCCUR AT THE END OF THE PROCESSOR TESTS IN EACH
BANK.

•

ICOPYRIGHT 1972,1973, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS. 01754
 IPDP-11/43 SYSTEM EXERCISER WITH KT11-C --- TTY, KW11-L
 ILP11,RF11,AND TC11

IRF11 AND TC11 CODE RUN IN KERNEL MODE IN SAME
 IPHYSICAL BANK AS BACKGROUND TESTS (IF ALL SWITCHES DOWN)
 IALL OTHER I/O IS RUN IN SUPERVISOR MODE, WITH EXTERNAL
 IBANK ACCESS KEY CYCLED THRU ALL VALUES (SUPERVISOR ONLY)
 ICPU TESTS RUN IN USER MODE UNLESS INHIBITED BY SR SETTINGS

ISA - 200 (300 IF LP11 IS USED)
 IRESTART - 310 (SR SETTINGS PREVIOUSLY MADE ARE USED)

IAT STARTUP, SR SETTINGS ARE:
 ISR 0=1 OR UP --- RUN WITHOUT KT11-C
 ISR 1=1 OR UP --- RUN ALL IN KERNEL MODE (INHIBITS RUNNING 4K AS 32K
 IAND INHIBITS RUNNING I/O IN SUPERVISOR)
 ISR 2=1 OR UP --- INHIBIT RUNNING 32K USER RELOCATION FROM EVERY 4K
 IBANK (ALLOW NORMAL CORE EXPANSION)
 ISR 3=1 OR UP --- INHIBIT RF AND TC RELOCATION OF CODE AND BUFFERS
 ISR 4=1 OR UP --- INHIBIT CYCLING ACCESS KEYS IN S7
 ISR 5=1 OR UP --- INHIBIT VARIABLE CORE EXPANSION

IAT HALT, SR SETTINGS ARE:
 ISR 13=1 OR UP---HALT ON ERROR
 ISR 14=1 OR UP---SCOPE LOOP
 ISR 13=1 OR UP---INHIBIT PRINT OUT
 ISR 12=1 OR UP---INHIBIT TRACE TRAPPING
 ISR 11=1 OR UP---INHIBIT SUB-PROGRAM ITERATION AND INHIBIT TESTS WHICH
 IUSE ALL COMBINATIONS OF NUMBERS
 ISR 10=1 OR UP---INHIBIT PROCESSOR TEST (ONCE SET, CLEARING IT WILL NOT
 ICAUSE PROCESSOR TEST TO CONTINUE)
 ISR 09=1 OR UP---STOP CYCLING SUPERVISOR PAGES
 ISR 08=1 OR UP---STOP CYCLING SUPERVISOR PAGE 7 ACCESS KEY
 ITHESE SWITCHES ARE ALSO CHECKED AT THE END OF EACH PASS

ISPECIAL DELETE SWITCHES-SET RESPECTIVE SWITCH TO A 1 TO INHIBIT
 IINITIATION OF DEVICE - IF SWITCH IS DOWN, DEVICE WILL BE STARTED
 IIF PRESENT
 ISW 0=1 INHIBIT TTY OUTPUT
 ISW 4=1 INHIBIT LINE CLOCK
 ISW 5=1 INHIBIT LINE PRINTER
 ISW 6=1 INHIBIT RF11 DISK
 ISW 7=1 INHIBIT TC11 DECTAPE

.SBTTL DATA

000240	NOP=240	ISYSTEM NULL OPERATION
104400	SCOPE=TRAP	ITRAP USED SCOPE LOOP AND ITERATION
177776	CC=177776	
177776	PSR=CC	
000616	TCSR=TTCSR	
000620	TDBR=TTDBR	

177576	SR=177576	
104006	HLT=104006	IEROR PRINTOUT CALL
104010	RTIK=104010	IRTI VIA KERNEL CALL
104012	EOB=104012	IEND OF BANK CALL
000000	R0=X0	
000001	R1=X1	
000002	R2=X2	
000003	R3=X3	
000004	R4=X4	
000005	R5=X5	
000006	SP=X6	
000006	R6=SP	
000007	PC=X7	
177776	STATUS=PSR	
001024	SEOREG=SREG1	
001026	NE61=SREG2	

ILOAD TRAP CATCHER IN LOCATIONS 0 THRU 577

ILOAD VECTOR AREA

000030	013226	000030	ENTSRV	IENT CALLS
000032	000340	000032	340	IHIGHEST PRIORITY
000034	013222	000034	.=34	
000036	000000	000036	SCOPE	IUSER TRAP
000046	015004	000046	.=46	
000052	000052	000052	LOGICAL	
000052	040000	000052	.=52	
000240	013460	000240	40000	
000242	000340	000242	.=240	
			P1RRV	
			340	

ILOAD STARTING AREA

000200	000200	001264	.=200	
	000137		JMP	##START
	000300		.=300	
000300	000137	001264	JMP	##START
	000310		.=310	
000310	000137	001250	JMP	##RSTMT

I DATA AREA

000400	000000	000400	.=400	
000600	000000	000600	UBUFF: 0	IBUFFER FOR USER SP
000600	000000	000600	.=600	
000600	000000	000600	SBUFF: 0	IBUFFER FOR SUPERVISOR STACK
000606	177560	000606	.=, +4	IFOR STACK OVERRUN
000610	177562	000610	TCSR: 177560	ITTY READER STATUS REGISTER
000612	000000	000612	TRBR: 177562	
000614	000000	000614	TTRVC: 00	
000616	177564	000616	TTRST: 02	
000620	177566	000620	TTCSR: 177564	ITTY PUNCH STATUS REGISTER
			TTDBR: 177566	

000622 000064 TTPVC: 64
000624 000066 TTPST: 66
000626 000000 TTSAV: 0
000630 177546 LKCSR: 177546
000632 000100 KMLVC: 100
000634 000102 KMLST: 102
000636 177514 LPOSR: 177514
000640 177516 LPDBR: 177516
000642 000200 LPVC: 200
000644 000202 LPST: 202
000646 177470 RFDAR: 177470
000650 177472 RFDOR: 177472
000652 177466 RFDAR: 177466
000654 177462 RFWC: 177462
000656 177464 RFCAR: 177464
000660 177460 RFOSR: 177460
000662 177461 RFOSRH: 177461
000664 000204 RFVC: 204
000666 000206 RFST: 206
000670 177572 SR0: 177572
000672 177574 SR1: 177574
000674 177576 SR2: 177576
000676 172516 SR3: 172516
000700 177600 UIPDR0: 177600
000702 177602 UIPDR1: 177602
000704 177616 UIPDR7: 177616
000706 177620 UDOR0: 177620
000710 177622 UDOR1: 177622
000712 177636 UDOR7: 177636
000714 177640 UIPAR0: 177640
000716 177642 UIPAR1: 177642
000720 177656 UIPAR7: 177656
000722 177660 UDPAR0: 177660
000724 177662 UDPAR1: 177662
000726 177676 UDPAR7: 177676
000730 172200 SIRDOR: 172200
000732 172220 SDPOR0: 172220
000734 172214 SIRDOR: 172214
000736 172256 SIPAR7: 172256
000740 172276 SDPAR7: 172276
000742 172216 SIPDR7: 172216
000744 172236 SDPOR7: 172236
000746 172300 KIPDR0: 172300
000750 172302 KIPDR1: 172302
000752 172304 KIPDR2: 172304
000754 172306 KIPDR3: 172306
000756 172316 KIPDR7: 172316
000760 172320 KDPDR0: 172320
000762 172322 KDPDR1: 172322
000764 172324 KDPDR2: 172324
000766 172326 KDPDR3: 172326
000770 172336 KDPDR7: 172336
000772 172340 KIPAR0: 172340
000774 172342 KIPAR1: 172342

STORE CONTENTS OF TTY PRINTER CSR HERE

IDISK ADDRESS AND ERROR
IDATA BUFFER REGISTER
IDISK ADDRESS REGISTER
IWORD COUNT REGISTER
ICURRENT ADDRESS REGISTER
ISTATUS REGISTER
IHIGH BYTE ADDRESS OR CSR

IKT11-C STATUS REGISTERS

000776 172344 KIPAR2: 172344
001000 172346 KIPAR3: 172346
001002 172356 KIPAR7: 172356
001004 172360 KDPAR0: 172360
001006 172362 KDPAR1: 172362
001010 172364 KDPAR2: 172364
001012 172366 KDPAR3: 172366
001014 172376 KDPAR7: 172376

001016 177600 IPDRTAB: 177600
001020 172200 IPDRTAB: 172200
001022 172300 IPDRTAB: 172300
001024 000000 SREG1: 0
001026 000000 SREG2: 0
001030 000000 SECNT: 0
001032 177342 TCCH: 177342
001034 177340 TCST: 177340
001036 177350 TCCT: 177350
001040 177344 TCNC: 177344
001042 177346 TCBA: 177346
001044 000214 TCIV: 214
001046 000216 TCSTA: 216
001050 000000 CURBNK: 0
001052 000000 OLCBNK: 0
001054 000000 CURIPAR: 0
001056 000000 CURDPAR: 0
001060 000000 CURIPDR: 0
001062 000000 CURDPDR: 0
001064 000000 CSIPDR: 0
001066 000000 BNKSTR: 0
001070 000000 TRPB: 0
001072 000000 PPSAV: 0
001074 177770 UBRK: 177770
001076 177772 FIR: 177772
001100 000240 FIRDIV: 240
001102 000242 FIRBST: 242
001104 000000 DISPLY: 0

STORES KTI1-C SWITCH REGISTER SETTINGS
STORES SR SETTINGS
USED TO LOAD KTI1-C REGISTERS
CONTROL AND FUNCTION
GENERAL STATUS
IDATA
IWORD COUNT
IBUS ADDRESS
IDETAPE INTERRUPT VECTOR

ISAF TO POINT TO CURRENT BANK
ISAVES ADDRESS OF PREVIOUS BANK
IADDRESS OF CURRENT IPAR

IPC TO POINT TO BEGIN THRU CURRENT PAGE

IMICROBREAK REGISTER ADDRESS

IVALUE LOADED INTO DISPLAY REGISTER
ILOW BYTE SHOWS PASS COUNT
IHIGH BYTE SHOWS MINUTES ELAPSED IF RUNNING KW11

IPARITY TRAP SERVICE ROUTINE
001106 000077 177666 PARSRV: CLR *KIPAR3
001112 000077 177674 PARSRV: CLR *KDPAR3
001116 012777 077406 177636 HDV *77406,*KIPDR3
001124 012777 077406 177634 HDV *77406,*KDPDR3
001132 012737 001234 000114 HDV *PARFU,##114
001140 012737 001230 000004 HDV *PARTQ,##4
001146 000000 PAR1: CLR *6
001152 012700 PAR1: HDV *00000,R0
001156 012702 PAR1: HDV *10000,R2
001162 011001 PAR2: HDV *R0,R1
001164 000720 PAR2: TST (R0)+
001166 077203 PAR2: SOB R2,PAH2

ICHANGE PARITY TRAP RETURN
ISETUP TIMEOUT RETURN

IINITIALIZE R0 TO TEST BANK THRU KERNEL 3
IR2 IS USED AS A COUNTER
IREAD LOCATION AND SAVE CONTENTS
IHOME POINTER
IFLOOP UNTIL 4K HAS BEEN CHECKED

```

001170 062777 000200 177602 PAR3: ADD #200,0KIPAR3
001176 062777 000200 177606 ADD #200,0KDPAR3
001204 022777 007600 177566 CMP #7600,0KIPAR3
001212 003357 PAR1
001214 011600 MOV #SP,R0
001216 104006 HLT

001220 000000 HALT
001222 000009 RESET
001224 000167 000020 JMP RSTRT
001230 022626 PART0: CMP (SP)+,(SP)+
001232 000756 PAR3
001234 022626 PARF0: CMP (SP)+,(SP)+
001236 104006 HLT

001240 000000 HALT
001242 000009 RESET
001244 000167 000000 JMP RSTRT
  
```

ILOAD PC AT TIME OF ABORT INTO R0
 IPARITY TRAP OCCURRED BUT MEMORY
 ISCAN DID NOT DETECT A PARITY ERROR-
 IR0 CONTAINS PC AT TIME OF ABORT
 IHALT AFTER PARITY ERROR
 ICLEAR PARITY REGISTERS AND DEVICES
 IRESTART TEST
 ITIMEOUT OCCURRED- BANK NOT PRESENT
 IGO CHECK FOR ANOTHER BANK
 IBAD PARITY FOUND
 IPARITY ERROR OCCURRED AND MEMORY
 ISCAN FOUND A MEMORY LOCATION
 ICONTAINING BAD PARITY, R0 CONTAINS
 ITHE ADDRESS OF THE LOCATION
 IHAVING BAD PARITY
 I(VIRTUAL MAPPED THRU KERNEL PAGE 3)
 IHALT AFTER PARITY ERROR
 IF CONTINUED, CLEAR PARITY AND DEVICES
 IRESTART PROGRAM

IRESTART ADDRESS USING INITIAL SR SETTINGS

```

RSTRT: MOV #KSTACK,R6
MOV0B #SR+1,#SREG2+1
OR START1
  
```

IBTTL I/O CODE
 ISTART UP FOR MINI MONITOR

```

001264 012706 017760 START: MOV #KSTACK,R6
001270 009737 000042 TST #042
001274 001417 BEQ STARTX
001276 012737 014764 000004 MOV #LOGIC,#04
001304 009037 000006 CLR #06
001310 005777 177562 TST #PIN
001314 012737 000006 000004 MOV #6,#04
001314 009037 001024 CLR #SREG1
001322 009037 001026 CLR #SREG2
001326 000407 BR
001332 000407 BR
001334 013737 177570 001024 STARTX: MOV #SR,#SREG1
001342 000000 HALT
001344 013737 177570 001026 MOV #SR,#SREG2

001392 012737 000001 013344 START1: MOV #1,#ICOUNT
001360 009037 001104 CLR #DISPLY
001364 004767 013376 JSR #7,CRFP
001370 012737 013292 000034 MOV #SCOPC,#034
001376 009037 000036 CLR #036
001402 012737 006670 013350 CLR #BEGIN,#RETURN
001410 009037 013346 CLR #SCOPE
001414 012737 013326 000030 MOV #EMTSRV,#030
  
```

ISET UP KERNEL STACK
 IRUNNING UNDER A MONITOR?
 INO- BRANCH
 IYES- CHECK TO SEE IF THIS IS AN 11/45
 IF NO TRAP THIS IS AN 11/45
 IRUN WITH ALL SWITCHES DOWN IF UNDER A MONITOR
 ISTORE KTI1-C SWITCHES
 ISTORE DEVICE SELECTION SWITCHES AND
 IDYNAMIC SWITCHES
 IINITIALIZE ITERATION COUNT
 IINITIALIZE PASS COUNT
 IOUTPUT CARRIAGE RETURN, LINE FEED
 IINITIALIZE SCOPE TO KERNEL PAGE 0
 IINITIALIZE EMT CALL TO KERNEL PAGE 0

```

001422 012737 000340 000032 MOV #340,#032
001430 012737 006162 004206 MOV #NXTBLK,#LLIMIT

IIN-LINE ROUTINE TO SET ACTION ENABLE IN ALL MA11 OR MF11 PARITY
IREGISTER FOUND

001436 012737 001106 000114 PARSET: MOV #PARSRV,#0114
001444 012737 000340 000116 MOV #340,#0116
001452 012701 172000 MOV #172000,R1

ISET UP PARITY TRAP VECTOR
ICAUSE LOCK UP AFTER PARITY TRAP
IINITIALIZE R1 TO THE FIRST POSSIBLE
IPARITY REGISTER ADDRESS

001456 012737 000006 000004 MOV #6,#04
001464 012737 000002 000006 MOV #RTI,#06
001472 012721 000001 15: MOV #1,(R1)+

IIF NO TIMEOUT, PARITY PRESENT-
ISET PARITY ACTION ENABLE TO CAUSE
ITRAP TO 114 ON PARITY ERROR,
ITO INHIBIT SETUP OF PARITY MEMORY
ICHANGE THE #1 TO A ZERO
IDONE WITH PARITY ADDRESSES?
INO- BRANCH
IYES- RESTORE TRAPCATCHER
ILOCK OUT INTERRUPTS
IINITIALIZE PRINT ROUTINE BUSY FLAG
IINITIALIZE BANK POINTER
IINHIBIT USER/SUPERVISOR/KERNEL?
IYES - SKIP OVER
ISET USER BIT IN SCOPE STATUS
ISETUP SUPERVISOR STACK

001476 020127 172140 CMP R1,#172140
001502 103773 BLD 15
001504 009037 000006 PARDNI: CLR #06
001510 012737 000340 177776 MOV #340,#PSR
001516 009037 016534 CLR #PRTN
001522 012737 000000 001050 MOV #0,#CURBNK
001530 032737 000002 001024 BIT #2,#SREG1
001536 001017 BNE ST0
001540 052737 140000 000036 BIS #140000,#036
001546 012746 000600 MOV #SBUF,(R6)
001552 052737 010000 177776 BIS #10000,#PSR
001560 106606 MTPD SP
001562 012746 000400 MOV #UBUFF,(R6)
001566 052737 030000 177776 BIS #30000,#PSR
001574 106606 MTPD SP
001576 012737 003210 000004 ST0: MOV #NODEV,#04
001604 009037 000006 CLR #04
001610 012700 000001 MOV #1,R0
001614 009037 177776 CLR #PSR
001620 033700 001026 BIT #SREG2,R0
001624 001024 BNE ST1
001626 009777 176764 TST #TTCSR
001632 012777 000200 176764 MOV #200,#TTPST
001640 009037 005646 CLR #DATA2
001644 012777 005660 176790 #TYOUTR,#TTPVC
001652 032737 000002 001024 BIT #2,#SREG1
001660 001003 BNE #+10
001662 052777 040000 176734 BIS #40000,#TTPST
001670 052777 000100 176720 BIS #100,#TTCSR

001676 006300 ST1: ASL R0
001700 006300 ST2: ASL R0
001702 006300 ST3: ASL R0
001704 006300 ST4: ASL R0
001706 003700 001026 BIT #SREG2,R0
001712 001026 BNE ST5
001714 009777 176710 TST #LKCSN
001720 109037 001105 CLR #DISPLY+1
001724 012777 003716 176700 MOV #LK3,#KWLVC
001732 012777 000240 176674 MOV #240,#KWLST
  
```

ISET UP PARITY TRAP VECTOR
 ICAUSE LOCK UP AFTER PARITY TRAP
 IINITIALIZE R1 TO THE FIRST POSSIBLE
 IPARITY REGISTER ADDRESS
 IIF NO TIMEOUT, PARITY PRESENT-
 ISET PARITY ACTION ENABLE TO CAUSE
 ITRAP TO 114 ON PARITY ERROR,
 ITO INHIBIT SETUP OF PARITY MEMORY
 ICHANGE THE #1 TO A ZERO
 IDONE WITH PARITY ADDRESSES?
 INO- BRANCH
 IYES- RESTORE TRAPCATCHER
 ILOCK OUT INTERRUPTS
 IINITIALIZE PRINT ROUTINE BUSY FLAG
 IINITIALIZE BANK POINTER
 IINHIBIT USER/SUPERVISOR/KERNEL?
 IYES - SKIP OVER
 ISET USER BIT IN SCOPE STATUS
 ISETUP SUPERVISOR STACK
 ISETUP USER STACK
 IINHIBIT TTY OUTPUT?
 IYES, GO CHECK NEXT
 ICHECK FOR TTY PRESENT
 IINITIALIZE BASE DATA
 INO, SETUP INTERRUPT VECTOR
 IRUN KERNEL ONLY
 IYES - SKIP NEXT
 INO - RUN DEVICE IN SUPERVISOR
 ISTART TTY OUTPUT
 IINHIBIT LINE CLOCK?
 IYES, GO CK NEXT
 IINITIALIZE MINUTE COUNTER
 ISERVICE KWI1-L AT LEVEL

001740 032737 000002 001024 BIT #2,0#SREG1
001746 001003 BNE ,+10
001750 052777 040000 176656 BNS #40000,0#KWLST
001756 005067 001770 CLR TIME
001762 052777 000100 176640 BNS #100,0#LKCSR
001770 006300 ST9: R0 ASL
001772 033700 001026 BIT #0#SREG2,R0
001776 001036 BNE ST6
002000 005777 176632 TST #LPCSR
002004 012767 000137 002076 MOV #137,30LPAT
002012 012767 000117 001772 MOV #79,CLINCT
002020 012767 000137 002064 MOV #137,CURPAT
002026 012777 000014 176604 MOV #14,0#LPDR
002034 012777 000370 176600 MOV #LPINTR,0#LPVC
002042 012777 000200 176574 MOV #200,0#LPST
002050 032737 000002 001024 BIT #2,0#SREG1
002056 001003 BNE ,+10
002060 052777 040000 176556 BNS #40000,0#LPST
002066 012777 000100 176542 MOV #100,0#LPCSR
002074 005037 001070 ST6: CLR #0#TRP
002100 012737 000006 000004 MOV #0,0#4
002106 004737 017066 JSR #7,0#USER
002112 004767 001112 JSR #7,DET1
002116 032767 000300 176700 PAGE: BIT #1,SREG1
002124 001402 BEQ ,+6
002126 000167 JMP STRF
002132 004767 011260 JSR #7,NRALL
002136 012777 077406 176612 MOV #77400,0#KIPDR7
002144 012777 077406 176616 MOV #77400,0#KIPDR7
002152 012777 007600 176622 MOV #7600,0#KIPAR7
002160 012777 007600 176626 MOV #7600,0#KIPAR7
002166 032737 000002 001024 BIT #2,0#SREG1
002174 001423 BEQ SEG01
002176 012701 000007 MOV #7,R1
002202 016702 176564 MOV KIPAR0,R2
002206 005003 CLR R3
002210 010312 SETEX: MOV R3,0#2
002212 010362 000020 MOV R3,20(R2)
002216 012762 077406 177740 MOV #77400,-40(R2)
002224 012762 077406 177760 MOV #77400,-20(R2)
002232 005722 TST (R2)+
002234 062703 000200 ADD #200,R3
002240 077115 SOB R1,SETEX
002242 000576 BR SETSEQ
002244 012777 077406 176474 SEG01: MOV #77400,0#KIPDR0
002252 012777 077406 176500 MOV #77400,0#KIPDR0
002260 012777 077406 176442 MOV #77400,0#SIPDR0
002266 012777 077406 176436 MOV #77400,0#SDPDR0
002274 016767 176430 176502 MOV SIPDR0,C0SIPDR
002302 032767 000004 176514 BIT #4,SREG01
002310 001453 BEQ USEALL
002312 012701 000007 MOV #7,R1
002316 016702 176372 MOV UIPAR0,R2

IS TO ALLOW TC11 IN
IRUN KERNEL ONLY?
IYES= BRANCH
INO= RUN DEVICE IN SUPERVISOR
INITIALIZE COUNT
ISTART LINE CLOCK
INHIBIT LINE PRINTER?
IYES, GO CK NEXT
IRESET FOR START OF LINE PATTERN
ILINE COUNT
ILINE FEED TO POSITION BUFFER
IINTERRUPT ENABLE
IPROCESSOR LEVEL 4
IRUN KERNEL ONLY?
IYES= BRANCH
IINTERRUPT ENABLE
INO "T" BIT FIRST PASS
IRESTORE TRAP CATCHER
IFOR I/O PROGRAM INSERTION
ICHECK FOR CORE EXPANSION
INHIBIT KT11-C?
IYES - GO CHECK FOR RUNNING RF11
INO = MAKE ALL PAGES INITIALLY NON-RESIDENT
IMAP KERNEL 7 TO THE EXTERNAL BANK, RW
INHIBIT USER/SUPERVISOR/KERNEL?
INO = BRANCH
IYES - MAP KERNEL PAGES 0-6 TO PA
IMAP KERNEL 0 TO BANK 0, RW
IMAP SUPERVISOR 0
ITO BANK 0, RW
ISET UP POINTER TO TRACK CURRENT SUPERVISOR PAGE
INHIBIT RUNNING 4K AS 32K?
INO, SETUP FOR RUNNING 4K AS 32K
IYES, MAP USER 0-6 TO PA

002322 005003 CLR R3
002324 010312 SETUSE: MOV R3,(R0)
002326 010362 000020 MOV R3,20(R2)
002332 062703 000200 ADD #200,R3
002336 012762 077406 177740 MOV #77400,-40(R2)
002344 012762 077406 177760 MOV #77400,-20(R2)
002352 005722 TST (R2)+
002354 077115 SOB R1,SETUSE
002356 012701 000007 MOV #7,R1
002362 016702 176404 MOV KIPAR0,R2
002366 005003 CLR R3
002370 010312 SETKER: MOV R3,(R0)
002372 010362 000020 MOV R3,20(R2)
002376 062703 000200 ADD #200,R3
002402 012762 077406 177740 MOV #77400,-40(R2)
002410 012762 077406 177760 MOV #77400,-20(R2)
002416 005722 TST (R2)+
002420 077115 SOB R1,SETKER
002422 012777 007600 176276 MOV #7600,0#UDPAR7
002430 012777 077406 176254 MOV #77400,0#UDPAR7
002436 000436 BR SETSUP
002440 012777 077406 176232 USEALL: MOV #77400,0#UIPDR0
002446 012777 077406 176232 MOV #77400,0#UDPDR0
002454 012777 077406 176266 MOV #77400,0#KIPDR1
002462 012777 077406 176272 MOV #77400,0#KIPDR1
002470 012767 000000 176392 MOV #0,CURBNK
002476 016767 176212 176390 MOV UIPAR0,CURIPAR
002504 016767 176212 176344 MOV UDPAR0,CURUPAR
002512 016767 176162 176340 MOV UIPDR0,CURIPDR
002520 016767 176162 176334 MOV UDPDR0,CURUPDR
002526 012767 000670 176332 MOV #0#EGIN,0#BKSTR
002534 012777 007600 176174 SETSUP: MOV #7600,0#SIPAR7
002542 012777 007600 176170 MOV #7600,0#SDPAR7
002550 032767 000002 176246 BIT #2,SREG1
002556 001015 BNE SETSP1
002560 012777 077406 176154 MOV #77400,0#SIPDR7
002566 012777 077406 176150 MOV #77400,0#SDPDR7
002574 012737 015642 002250 MOV #ABSRV,0#250
002602 012737 000340 000252 MOV #340,0#252
002610 000413 BR SETSEQ
002612 012777 077406 176122 SETSP1: MOV #77400,0#SIPDR7
002620 012777 077406 176116 MOV #77400,0#SDPDR7
002626 012737 000252 000250 MOV #252,0#250
002634 005037 000252 CLR #252
002640 012777 000007 176030 SETSEGI: MOV #7,0#SH3
002646 012777 001001 176014 MOV #1001,0#SR0
002654 012737 003210 000004 STRF: MOV #NODEV,0#4
002662 012700 000100 MOV #100,R0
002666 036700 176134 BIT SREG2,R0
002672 001040 BNE STTC
002674 005737 000660 TST #0#RFC0R
002700 016701 176120 MOV SREG1,R1
002704 042701 177770 BIC #177770,R1
002710 005701 TST R1

IMAP KERNEL 0 TO BANK 0, RW
IMAP SUPERVISOR 0
ITO BANK 0, RW
ISET UP POINTER TO TRACK CURRENT SUPERVISOR PAGE
INHIBIT RUNNING 4K AS 32K?
INO, SETUP FOR RUNNING 4K AS 32K
IYES, MAP USER 0-6 TO PA
IMAP KERNEL 0 THRU 6 TO PHYSICAL ADDRESS
IFOR I/O RELOCATION
IMAP USER 7 0-SPACE TO THE EXTERNAL
IBANK TO ALLOW ACCESSING PROCESSOR STATUS
IMAP USER 0 TO BANK 0, RW
IMAP KERNEL 1 TO BANK 0, RW
ICURRENT PAR CONTENTS
ICURRENT PAGE ADDRESS REGISTER ADDRESSES
ICURRENT PAGE DESCRIPTOR REGISTER ADDRESSES
ICURRENT STARTING PC
IMAP SUPERVISOR 7 TO THE
EXTERNAL BANK
ICHECK SWITCH 4 - INHIBIT CYCLING ACCESS KEY?
IYES = BRANCH
IMAP SUPERVISOR PAGE 7 4K, NR
IMAP SUPERVISOR PAGE 7 4K, RW
IENABLE ALL 0-SPACES
ITURN ON MEMORY MANAGEMENT AND ENABLE TRAPPING (

```

002712 001004          BNE STRF1
002714 012777 024132 175742    MOV #IRF+20000,IRFVC
002722 000403          BR .+10
002724 012777 004132 175732    STRF1: MOV #IRF,IRFVC
002732 012777 000240 175726    MOV #240,IRFST
002740 012767 043503 001316    MOV #43503,FUNCTION
002746 109277 175710          INCB #RFGSHH
002752 016777 001312 175674    MOV WORDCT,IRFVC
002760 016777 001302 175670    MOV LLIMIT,IRFCAR
002766 116777 001272 175664    MOV# FUNCTION,IRFCSR
002774 012737 004326 004272    STTC: MOV #TCBUF,IRTCBUF
003002 012737 004326 004274    MOV #TCBUF,IRTCBUFV
003010 012737 000001 004320    MOV #1,IRTCEMA
003016 009037 004322          CLR #IRTCOP
003022 009037 004324          CLR #IRTCRL
003026 006300          ASL R0
003030 036700 175772          BIT SREG2,R0
003034 001041          BNE MODE1
003036 009777 175772          TST #TCST
003042 009067 001250          CLR DATAK
003046 013701 001024          MOV #SREG1,R1
003052 032701 000050          BIT #5,R1
003056 001403          BEQ SETREL
003060 032701 000017          BIT #17,R1
003064 001002          BNE .+6
003066 009237 004324          SETREL: INO #IRTCRL
003072 032701 000007          BIT #7,R1
003076 001007          BNE .+20
003100 012777 025326 175736    MOV #ENDE+20000,IRTCIV
003106 012737 024326 004274    MOV #TCBUF+20000,IRTCBUFV
003114 000403          BR .+10
003116 012777 005326 175720    MOV #ENDE,IRTCIV
003124 012777 000300 175714    MOV #300,IRTCSTA
003132 012777 004503 175672    MOV #R+IRB+DD+UN1,IRTCM
003140 009037 175776          MODE1: CLR #PSR
003144 012737 000006 000004    MOV #6,IR#4
003152 032767 000002 175644    BIT #2,SREG1
003160 001005          BNE HAIN+2
003162 012737 140000 175776    MOV #140000,IRPSR
003170 000401          BR .+4
003172 000001          HAIN: HAIT
003174 036727 175626 002000    BIT #2000
003202 001373          BNE
003204 000167          JMP
  
```

```

ROUTINE EXECUTED IF A DEVICE IS NOT INHIBITED BUT IS FOUND
AND ALTERS THE PC TO SKIP THE STARTUP FOR THE DEVICE
NODEV: B1S R0,IRREG2
SUB #6,(SP)
BIC #17,2(SP)
  
```

```

003226 000002          RTI

IRDP=11 MEMORY DETERMINATION AND SETUP
IRUSE WITH VARIABLE CORE QUANTITY SYSTEMS
003230 012767 104012 007792    DET1: MOV #E0B,DONE
003236 032767 000007 175560    BIT #7,SREG1
003244 001001          BNE .+4
003246 002207          RTS #7
003250 032737 000040 001024    BIT #40,SREG1
003256 001401          BEQ DET4
003260 000207          RTS #7
003262 012737 003332 000004    DET4: MOV #DET2,IR#4
003270 012737 000340 000006    MOV #340,IR#6
003276 000241          CLC
003300 009537 037770          EIGHT: ADC #037770
003304 009537 057770          TWELVE: ADD #057770
003310 009537 077770          SIXTEEN: ADD #077770
003314 009537 117770          TWENTY: ADD #117770
003320 009537 137770          TWOFOR: ADD #137770
003324 009537 157770          TWOEIG: ADD #157770
003330 000437          BR STRT20
003332 012602          DET2: MOV (6)+,IR#2
003334 009726          TST (6)+
003336 062702 000060          ADD #STRT4-EIGHT=4,R2
003342 000112          JMP #R2
003344 009000          MOVE1: CLR #0
003346 018102          MOV #1,X2
003350 062702 013416          ADD #0+2,X2
003354 012021          MOV (0)+,(1)+
003358 020201          CMP #2,X1
003360 001375          BNE .+4
003362 000207          RTS #7
003364 000521          STRT4: BR DET3
003366 000240          NOP
003370 000240          NOP
003372 004767 000110          STRT0: JSR #7,XFER0
003376 000506          MOD4
003400 004767 000072          STRT12: JSR #7,XFER12
003404 000475          MOD8
003406 004767 000054          STRT16: JSR #7,XFER16
003412 000464          MOD12
003414 004767 000036          STRT20: JSR #7,XFER20
003420 000453          MOD16
003422 004767 000020          STRT24: JSR #7,XFER24
003426 000442          MOD20
003430 004767 000002          STRT28: JSR #7,XFER28
003434 000431          MOD24
003436 012701 140000          XFER20: MOV #140000,X1
003442 004767 175676          JSR #7,MOVE
003446 012701 120000          XFER24: MOV #120000,X1
003452 004767 175666          JSR #7,MOVE
  
```

```

003496 012701 100000 XFER201 MOV #100000,X1
003462 004767 177656 JSR X7,MOVE
003468 012701 000000 XFER161 MOV #000000,X1
003472 004767 177644 JSR X7,MOVE
003476 012701 000000 XFER121 MOV #000000,X1
003502 004767 177636 JSR X7,MOVE
003508 012701 020000 XFER81 MOV #200000,X1
003512 004767 177628 JSR X7,MOVE
003516 000207 RTS #137,DONE+120000 ;RETURN FROM TRANSFERS
003520 012767 000137 H00241 MOV #137,DONE+120000
003526 012767 146456 H00241 MOV #BEGINX+140000,DONE+120000
003534 012767 000137 H00201 MOV #137,DONE+100000
003542 012767 126456 H00241 MOV #BEGINX+120000,DONE+100000
003550 012767 000137 H00161 MOV #137,DONE+40000
003556 012767 106456 H00241 MOV #BEGINX+100000,DONE+60000
003564 012767 000137 H00121 MOV #137,DONE+40000
003572 012767 066456 H00241 MOV #BEGINX+60000,DONE+40000
003600 012767 000137 H00081 MOV #137,DONE+20000
003606 012767 066456 H00241 MOV #BEGINX+60000,DONE+20000
003614 012767 000137 H00041 MOV #137,DONE
003622 012767 026456 H00241 MOV #BEGINX+20000,DONE+2
003630 005037 000006 DET31 CLR #06
003634 012737 000006 MOV #0,#04
003642 000207 RTS X7

;TTY TRANSMITTER PRINT VALUES 0 TO 377
003644 005027 000000 TYOUT: CLR #0 ;INITIALIZE DATA
003646 003646 DATA2#,-2
003650 016777 177772 174742 TYOUT1: MOV DATA2,@TTD0R ;OUTPUT TO DEVICE
003656 104010 RTIK ;RETURN TO MAINLINE
003660 017767 174732 174740 TYOUTR: MOV @TTCSR,TTSAV ;SAVE CONTENTS OF TTY CSR
003666 105767 174734 TSTB TTSAV ;CHECK TO SEE IF READY WAS SET
003672 100401 BMI ,+4 ;BRANCH IF FLAG FOUND
003674 104006 HLT ;FALSE INTERRUPT RETURN
003676 000240 NOP
003700 005267 177742 INC DATA2 ;INCREMENT DATA
003704 022767 000400 177734 CMP #000,DATA2 ;TEST DATA FOR UPPER LIMIT
003712 001754 BEQ TYOUT ;AT UPPER LIMIT START OVER
003714 000755 BR TYOUT1 ;FINISH REST OF DATA

;TEST OF LINE CLOCK -OUTPUT MINUTE COUNT TO UPPER BYTE OF DISPLAY REGISTER,
;ON INTERRUPTS ENTER HERE
;TEST FOR DONE
003716 105777 174706 LK3: TSTB @LKCSR
003722 100401 BMI ,+4
003724 104006 HLT ;FALSE INTERRUPT
003726 042777 000200 174674 BIC #200,@LKCSR
003734 005267 000012 LK4: INC TIME
003740 022767 007020 000004 CMP #3600,@TIME ;ONE MINUTE UP?
003746 003007 BGT LK2 ;NO- EXIT
003750 005027 000000 LK1: CLR #0 ;YES- CLEAR LINE CLOCK TIMER
003752 003752 TIME#,-2
003754 105267 175125 INCB DISPLY+1
003760 016737 175120 177970 MOV DISPLY,@3R ;LOAD DISPLAY REGISTER
;WITH MINUTE COUNT

```

```

003766 104010 LK2: RTIK ;RETURN TO MAINLINE

;LINE PRINTER INTERRUPT SERVICE ROUTINE
003770 105777 174642 LPINTR: TSTB @LPCSR ;TEST FOR FLAG
003774 100401 HLT
003776 104006 BMI ,+4
003780 026727 000006 000117 LP6: CMP CLINCT,#79. ;TEST FOR END OF LINE
003800 001413 BEQ LP4 ;GO GENERATE CR/LF
003810 005227 000000 INC #0 ;INCREMENT LINE POSITION COUNT
003814 026727 000072 000137 CLINCT#,-2 ;POSITION OF LINE
003822 001406 BEQ CURPAT,#137 ;TEST FOR MAXIMUM PATTERN
003824 005267 000062 INC LP3 ;YES - GO TO LP3 AND RESET
003830 000431 BR CURPAT ;NO - INCREMENT TO NEXT PATTERN
003832 012767 000040 000092 LP3: MOV #40,CURPAT ;GO SEND IT TO LINE PRINTER
003840 000425 BR LP2 ;RESET PATTERN AND SEND TO PRINTER
003842 005067 177744 LP4: CLR CLINCT ;RESET LINE COUNT
003846 012777 000012 174964 MOV #12,@LPD0R ;LINE FEED
003854 105777 174956 TSTB @LPCSR
003860 100375 BPL ,+4
003862 026727 000022 000137 SOLPAT,#137 ;START OF LINE PATTERN
003870 001403 BEQ LP5
003872 005267 000012 INC SOLPAT ;INCREMENT START OF LINE
003876 000403 BR LP1
004100 012767 000040 000092 LP5: MOV #40,SOLPAT ;RESET START OF LINE
004106 012727 000000 000090 LP1: MOV #0,#0 ;START OF LINE TO CURRENT
004114 004110 004112 CURPAT#,-2 ;CHARACTER BEING PRINTED
004114 016777 177772 174916 LP2: MOV CURPAT,@LPD0R ;START OF LINE CHARACTER
004122 105777 174910 TSTB @LPCSR ;CURRENT PATTERN TO LINE PRINTER
004126 100724 BMI LP6
004130 104010 RTIK ;RETURN TO MAIN LINE

;IRF11 DISK
004132 105777 174922 IRF1: TSTB @RFCSR ;INTERRUPT VECTOR POINTS HERE
004136 100402 BMI ,+6
004140 104006 HLT ;IRF11 READY NOT UP
004142 000434 BR RFSTANT
004144 005777 174910 TST @RFCSR ;ERROR SET?
004150 100012 BPL ERROK ;BRANCH IF NOT
004152 032777 020000 174900 BIT #20000,@RFCSR ;YES=WRITE CHECK ERROR?
004160 001404 BEQ ERRSET ;NO-BRANCH
004162 104006 HLT ;YES=RF11 WRITE CHECK ERROR
004164 000337 004204 SWAB #FUNCTION ;CHANGE COMMAND TO DO A WRITE
004170 000421 BR RFSTANT
004176 104006 ERRSET: HLT ;IRF11 ERROR SET-NOT WRITE CHECK ERROR
004174 000417 BR RFSTANT
004176 005777 174492 ERROK: TST @RFC ;IRF-11 WORD COUNT NOT ZERO
004202 100002 BPL ,+6
004204 104006 HLT
004206 000412 BR RFSTANT
004210 122777 000003 174430 CMPB #3,@RFDAL ;DISK AT UPPER LIMIT? 7=2, 17=4, 37=8
004216 001010 BNE RF1 I/O
004220 027727 174426 174000 CMP #RFDAL,#174000 ;AS FAR ON DISK AS WE CAN GO?

```

004226	101404			BLOS	RF1	JNO
004230	000337	004264		SWAB	##FUNCTION	ICHANGE COMMAND
004234	109277	174422		RFSTART:	INCB	IINITIALIZE DISK = DAR-DAE
004240	013777	004266	174410	RF1:	##LLIMIT,##FCAR	ICORE BASE
004246	013777	004270	174400		##WRODCT,##FWC	ILENGTH OF TRANSFER
004254	113777	004264	174376		##FUNCTION,##FCAR	IWRITE OR WRITE CHECK TO DISK
004262	104010				RTIK	IRETURN TO MAINLINE CODE
004264	000000			FUNCTION:	0	IDISK COMMAND
004266	006162			LLIMIT:	NXTBLK	IFIRST CORE ADDRESS OF TRANSFER
004270	176000			WRODCT:	-2000	ILENGTH OF TRANSFER

```

;DECTAPE DIAGNOSTIC ROUTINE. THE TAPE ON DRIVE 1 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
;HAS WRITTEN (TCLAST) IS FOUND, IT IS READ INTO THE DECTAPE BUFFER AREA.
;THE PROGRAM INTERRUPT REQUEST FACILITY IS THEN USED TO BOOK A REQUEST
;FOR CHECKING THE DATA AT LEVEL 3, AND NO FURTHER DATA IS READ IN
;UNTIL THAT DATA HAS BEEN CHECKED. AFTER IT IS CHECKED, THE DATA IS
;SCRAMBLED TO GUARANTEE THAT NEW DATA IS REALLY READ IN NEXT TIME, WHILE
;THIS IS GOING ON, BLOCK NUMBERS ARE CHECKED FOR BEING IN ORDER AS THE
;TAPE TRAVELS TOWARD THE FORWARD END ZONE, ONCE THE DATA IS FULLY CHECKED
;THE NEXT BLOCK THAT COMES UP IS READ IN AND THE PROCESS REPEATED, ONCE
;THE BLOCK WHOSE NUMBER IS IN TCFRST HAS BEEN READ, THE TAPE IS DRIVEN
;INTO THE FORWARD END ZONE AND THE WHOLE SEQUENCE IS REPEATED.
  
```

FUNCTION VALUES IN CBR

000004				RD=4		IREAD DATA
000014				WD=14		IWRITE DATA
000002				RB=2		IREAD BLOCK NUMBER
000100				IC=100		IINTERRUPT ENABLE
000001				DO=1		IDO = THE FUNCTION
004000				R=4000		IREVERSE
000400				UN1=400		IUNIT 1
004272	004326			DECTAPE DATA		
004274	004326			TCBUFP:	TCBUF	IVIRTUAL ADDRESS OF BUFFER
004276	000000			TCBUFP:	TCBUF	IVIRTUAL ADDRESS OF BUFFER
004300	001101			TCFRST:	0	IFIRST BLOCK TO BE SEARCHED FOR
004302	000000			TCLAST:	577.	I LAST BLOCK TO BE SEARCHED FOR
004304	000000			TCBLK:	0	I CURRENT BLOCK FOUND
004306	000000			TCEXP:	0	I THE BLOCK THAT IS EXPECTED
				TCSTOR:	0	ISAVES CONTENTS OF TC STATUS
						IREGISTER AT TIME OF INTERRUPT
004310	000000			FBLK:	0	
004312	000000			STOR1:	0	
004314	000000			STOR2:	0	
004316	000000			DATACK:	0	
004320	000001			TCMA:	1	IEXTENDED MEMORY ADDRESS BITS AND DO BIT
004322	000000			TCEOP:	0	IEND OF PASS FLAG
004324	000000			TCREL:	0	ISET IF TC CODE IS TO BE

RELOCATED TO THE SAME BANK AS USER

DECTAPE BUFFER USED FOR BOTH WRITE AND READ FUNCTIONS
 TCBUF: 0
 ,B,+776

```

;DECTAPE ROUTINE TO DRIVE TAPE INTO ENDEZONE
;REGENERATES DECTAPE WRITE BUFFER BEFORE WRITING TAPE
;MUST BE ENTERED WITH FUNCTION SET TO READ BLOCK #
  
```

005326	042777	017777	173510	ENDE:	BIC	#17777,0TCIV	
005334	052777	005326	173502		BIS	#ENDE,0TCIV	
005342	005777	173466			TST	0TCST	IEND ZONE SET?
005346	100403				BNI	ENDE1	IYES=BRANCH
005350	105277	173456			INCB	0TCCH	I NO=GO AGAIN
005354	000002				RTI		
005356	042777	017777	173460	ENDE1:	BIC	#17777,0TCIV	
005364	052777	005360	173452		BIS	#RBN,0TCIV	
005372	005037	004310			CLR	##FBLK	ILOOKING FOR FIRST DESIRED BLOCK
005376	032777	004000	173426		BIT	#4000,0TCCH	I REVERSE SET?
005404	001007				BNE	ENDE2	IYES, NOW AT FORWARD END ZONE, BRANCH
005406	052777	004001	173416		BIS	#4001,0TCCH	I NO, SET IT AND DO
005414	013737	004300	004304		MOV	##TCLAST,##TCEXP	
005422	000002				RTI		
005424	005737	004324		ENDE2:	TST	##TCREL	I TC RELOCATION ALLOWED?
005430	001403				BKREQ		I NO, BRANCH TO BOOK PIRQ
005432	005737	004322			TST	##TCEOP	IYES=DON'T STOP TC IF THIS IS
							I THE INITIAL SEARCH FOR THE
							I FORWARD END ZONE
005436	001003				BNE	STOPTC	
005440	052777	002000	173430	BKREQ:	BIS	#2000,0PIR	IYES, BOOK LEVEL 2 PIRQ TO RESTART
							ITC11 AND REGENERATE WRITE BUFFER
005446	104010			STOPTC:	RTIK		

```

;ROUTINE TO REGENERATE WRITE BUFFER AND START UP DECTAPE WRITING BLOCKS
;FORWARD. MUST RUN AT LOWER LEVEL THAN READ BUFFER DATA CHECK ROUTINE
;BOTH ARE RUN UNDER PIRQ
  
```

005450	010137	004312		REGEN:	MOV	R1,##STOR1	ISAVE REGISTERS
005454	010237	004314			MOV	R2,##STOR2	
005460	005037	004310			CLR	##FBLK	IINITIALIZE FIRST BLOCK FOUND FLAG
005464	012701	000001			MOV	#1,R1	I REGENERATE WRITE BUFFER
005470	013702	004274			MOV	##TCBUFP,R2	
005474	010122			REGEN1:	MOV	R1,(R2)+	
005476	210112				MOV	R1,0R2	
005500	005122				COM	(R2)+	
005502	005201				INC	R1	
005504	022701	000201			CMP	#201,R1	
005510	001371				BNE	REGEN1	
005512	013701	004312			MOV	##STOR1,R1	I RESTORE REGISTERS
005516	013702	004314			MOV	##STOR2,R2	
005522	042777	017777	173314		BIC	#17777,0TCIV	ICHANGE INTERRUPT RETURN
005530	052777	005360	173306		BIS	#RBN,0TCIV	ITO READ BLOCK NUMBER ROUTINE
005536	012777	000003	173266		MOV	##R+IE+DO+UN1,0TCCH	I FORWARD READ BLOCK #
005544	013737	004276	004304		MOV	##TCFHST,##TCEXP	
005552	005037	004322			CLR	##TCEOP	

```

005556 000207          RTS      X7
                                IDECTAPE ROUTINE TO PROCESS INTERRUPT FROM READING A BLOCK NUMBER
                                ISETS UP FOR WRITING A BLOCK IF TAPE IS GOING FORWARD
                                ISETS UP FOR READING A BLOCK IF TAPE IS GOING IN REVERSE
                                IDATAK IS NON ZERO WHENEVER A PREVIOUS READ HAS NOT
                                IBEEN FULLY CHECKED-IF NONZERO, NO DATA IS READ IN
005560 017727 173250 004306 RBN1  MOV    @TCST,@TCSTOR    ISAVE STATUS IN CASE OF ERROR
005564 032777 100200 173236 RBN1  BIT    #100200,@TCCH    ICHECK ERROR AND READY
005574 100006          BPL     RBN2          IBRANCH IF ERROR NOT SET
005576 104006          HLT     RBN2          IERROR SET-CHECK CONTENTS OF TCSTOR
005600 012777 004902 173224 RBN11 MOV    #RB+IE+UN1,@TCCH    IREVERSE READ BLOCK NUMBER
005606 000167 177514          JMP     ENDE          IRESTART ON ERROR
005612 001002          BNE     RBN2          IBRANCH IF READY SET
005614 104006          HLT     RBN2          IREADY NOT SET
005616 000770          BR      RBN1          IRESTART
005620 005737 004310 RBN31 TST    @#FBLK          IFIRST EXPECTED BLOCK FOUND YET?
005624 100520          BMI     RBN10         IYES-BRANCH
005626 027737 173204 004304 RBN3  CMP    @TCDT,@#TCEXP    IBLOCK NUMBER EQUAL TO FIRST ONE DESIRED?
005634 001067          BNE     RBN7          INO. BRANCH
005636 005137 004310 RBN3  COB    @#FBLK          IYES, SET FIRST BLOCK FOUND FLAG (NEGATIVE INDIC
005642 042777 017777 173174 RBN41 BIC    #17777,@TCIV    #17777,@TCIV
005650 092777 006162 173106 RBN4  BIS    #NXTBLK,@TCIV    #NXTBLK,@TCIV
005656 012777 177400 173154          MOV    #400,@TCCH    #400,@TCCH
005664 013777 004272 173150          MOV    @TCBUPP,@TCBA    @TCBUPP,@TCBA
005672 032777 004000 173132          BIT    #4000,@TCCH    #4000,@TCCH
005700 001007          BNE     RBN5          IREVERSE SET?
005702 012777 000514 173122          MOV    #WD+IE+UN1,@TCCH #WD+IE+UN1,@TCCH
005710 053777 004320 173114          BIS    @#TCEMA,@TCCH    @#TCEMA,@TCCH
005716 104010          RTIK
005720 005737 004316 RBN51 TST    @#DATAK          IF NOT ZERO INDICATES READ BUFFER
                                INOT YET FULLY CHECKED
                                ISO DON'T READ BLOCK
005724 001011          BNE     RBN6          IAS ZERO-DO SET IT NOW
005726 005137 004316 RBN6  COB    @#DATAK          IAND READ A BLOCK
005732 012777 004904 173072          MOV    #RD+R+IE+UN1,@TCCH #RD+R+IE+UN1,@TCCH
005740 053777 004320 173064          BIS    @#TCEMA,@TCCH    @#TCEMA,@TCCH
005746 104010          RTIK
005750 027737 173062 004276 RBN61 CMP    @TCDT,@#TCPRST    @TCDT,@#TCPRST
005756 003004          BGT     RBN6          .+12
005760 005237 004322          INC    @#TCEOP          @#TCEOP
005764 000167 177336          JMP     ENDE          ENDE
005770 042777 017777 173046 RBN6  BIC    #17777,@TCIV    #17777,@TCIV    IREAD BLOCK NUMBER
005776 092777 005560 173040 RBN6  BIS    #RBN,@TCIV        #RBN,@TCIV
005804 012777 004503 173020          MOV    #RB+R+DO+IE+UN1,@TCCH #RB+R+DO+IE+UN1,@TCCH
006012 104010          RTIK
006014 032777 004000 173010 RBN71 BIT    #4000,@TCCH    #4000,@TCCH
006022 001415          BEQ     RBN9          ICHECK REVERSE BIT
006024 027737 173006 004304 RBN9  BEQ     RBN9          IBRANCH IF FORWARD
006032 003004          CMP    @TCDT,@#TCEXP    @TCDT,@#TCEXP
006034 104006          BGT     RBN8          ICONTINUE IF NOT THERE YET
006036 012777 000502 172766 RBN8  HLT     RBN8          IPAST DESIRED POINT
006044 000167 177256          MOV    #RB+IE+UN1,@TCCH #RB+IE+UN1,@TCCH
006050 105277 172756 RBN81 JMP     ENDE          IREAD BLOCK NUMBER, CHANGE DIRECTION
006054 104010          RTIK
  
```

```

006056 023777 004304 172752 RBN91 CMP    @#TCEXP,@#TODT    @#TCEXP,@#TODT
006064 003371          BGT     RBN8          IBRANCH IF NOT THERE YET
006066 104006          HLT     RBN8          IPAST DESIRED POINT
006070 012777 004902 172734 RBN8  MOV    #RB+R+IE+UN1,@TCCH #RB+R+IE+UN1,@TCCH
006076 000167 177224          JMP     ENDE          IREAD BLOCK NUMBER, CHANGE DIRECTION
006102 032777 004000 172722 RBN101 BIT    #4000,@TCCH    #4000,@TCCH
006110 001011          BNE     RBN11         IREVERSE SET?
006112 023777 004300 172716 RBN101 BNE     RBN11         IYES-BRANCH
006120 002002          CMP    @#TCLAST,@TCDT    @#TCLAST,@TCDT
006122 000167 177200          BGE     RBN11         INO. CHECK FOR PAST LAST DESIRED BLOCK
006126 005237 004304          JMP     ENDE          IF PAST GO TO ENDE ROUTINE
006132 000402          INC    @#TCEXP          IUP BLOCK NUMBER EXPECTED
006134 005337 004304 RBN111 BR      .+6
006140 023777 004304 172678 RBN111 DEC    @#TCEXP          @#TCEXP
006146 001002          CMP    @#TCEXP,@#TODT    @#TCEXP,@#TODT
006150 000167 177466          BNE     RBN4          ISEQUENTIAL BLOCK?
006154 104006          HLT     RBN4          INO
006156 000167 177416          JMP     RBN1          IYES, READ OR WRITE IT
                                IBLOCK NUMBER NOT SEQUENTIAL
                                IRESTART
                                IDECTAPE ROUTINE INTERRUPTED TO AFTER READING OR WRITING A BLOCK
006162 017737 172646 004306 NXTBLK1 MOV    @TCST,@#TCSTOR    ISAVE DTA STATUS IN CASE OF ERROR
006170 032777 100200 172634 NXTBLK1 BIT    #100200,@TCCH    ICHECK ERROR AND READY
006176 100003          BPL     NXTB1        IBRANCH IF ERROR NOT SET
006200 104006          HLT     NXTB1        IERROR SET-CHECK CONTENTS OF TCSTOR
006202 000167 177372          JMP     RBN1          IRESTART
006206 001003          BNE     NXTB1        IBRANCH IF READY SET
006210 104006          HLT     NXTB1        IREADY NOT SET
006212 000167 177362          JMP     RBN1          ICLEARED INDICATES REQUEST ALREADY MADE
006216 105737 004316 NXTB21 TSTB   @#DATAK          IOR NO DATA WAS READ
006222 001405          BEQ     NXTB3        ICLEAR AND BOOK
006224 105037 004316 NXTB31 CLR    @#DATAK          IBOOK A PIRO AT LEVEL 3
006230 092777 004000 172640 NXTB31 BIS    #4000,@PIR        #4000,@PIR
006236 032777 004000 172566 NXTB31 BIT    #R,@TCCH        #R,@TCCH
006244 001022          BNE     NXTB4        IREVERSE SET?
006246 042777 017777 172570 RBN1  BIC    #17777,@TCIV    #17777,@TCIV
006254 112777 000103 172550 NXTB41 MOV    #RB+IE+DO,@TCCH    #RB+IE+DO,@TCCH
006262 023737 004300 004304 RBN1  CMP    @#TCLAST,@#TCEXP    @#TCLAST,@#TCEXP
006270 003004          BGT     RBN1          .+12
006272 092777 005326 172544 RBN1  BIS    #ENDE,@TCIV        #ENDE,@TCIV
006300 104010          RTIK
006302 092777 005560 172534 RBN1  BIS    #RBN,@TCIV        #RBN,@TCIV
006310 104010          RTIK
006312 042777 017777 172524 NXTB41 BIC    #17777,@TCIV    #17777,@TCIV
006320 112777 000103 172504 RBN1  MOV    #RB+IE+DO,@TCCH    #RB+IE+DO,@TCCH
006326 023737 004274 004304 RBN1  CMP    @#TCPRST,@#TCEXP    @#TCPRST,@#TCEXP
006334 002407          BLT     RBN1          .+20
006336 092777 005326 172500 RBN1  BIS    #ENDE,@TCIV        #ENDE,@TCIV
006344 012737 000001 004322 RBN1  MOV    #1,@#TCEOP        #1,@#TCEOP
006352 104010          RTIK
006354 092777 005560 172462 RBN1  BIS    #RBN,@TCIV        #RBN,@TCIV
006362 104010          RTIK
                                IDECTAPE ROUTINE TO CHECK DATA READ IN DURING READ BLOCK
  
```


ITEST MOVE INSTRUCTION FOR INDEXING

```

007132 016067 013362 004244      MOV      A(0),TEMP
007140 026727 004240 129292      CMP      TEMP,#129292
007146 001401                      BEQ      ,+4
007150 104006                      HLT
007152 104400                      SCOPE

007154 012762 092929 013404      MOV      #092929,TEMP(2)
007162 026727 004226 092929      CMP      TEMP+10,#092929
007170 001401                      BEQ      ,+4
007172 104006                      HLT
007174 104400                      SCOPE
  
```

ITEST BIC INSTRUCTION FOR INDEXING

```

007176 012767 177777 004200      MOV      #-1,TEMP
007204 046267 013362 004172      BIC      A(2),TEMP
007212 026727 004166 129292      CMP      TEMP,#129292
007220 001401                      BEQ      ,+4
007222 104006                      HLT
007224 104400                      SCOPE

007226 012767 177777 004140      MOV      #-1,TEMP-10
007234 042767 092929 004132      BIC      #092929,TEMP-10
007242 026727 004126 129292      CMP      TEMP-10,#129292
007250 001401                      BEQ      ,+4
007252 104006                      HLT
007254 104400                      SCOPE
  
```

ITEST SUBTRACT INSTRUCTION FOR INDEXING

```

007256 012767 129292 004120      MOV      #129292,TEMP
007264 166760 004062 013414      SUB      B,TEMP+10(0)
007272 001401                      BEQ      ,+4
007274 104006                      HLT
007276 104400                      SCOPE

007300 012767 092929 004076      MOV      #092929,TEMP
007306 166267 013362 004070      SUB      A(2),TEMP
007314 001401                      BEQ      ,+4
007316 104006                      HLT
007320 104400                      SCOPE
  
```

ITEST UNARYS INDEXED

```

007322 012767 177777 004094      MOV      #-1,TEMP
007330 009060 013414      CLR      D(0)
007334 009767 004044      TST     TEMP
007340 001401                      BEQ      ,+4
007342 104006                      HLT
007344 104400                      SCOPE

007346 012767 177777 004090      MOV      #-1,TEMP
  
```

```

007354 009162 013374      COM      C(2)
007360 009767 004020      TST     TEMP
007364 001401                      BEQ      ,+4
007366 104006                      HLT
007370 104400                      SCOPE

007372 012767 177777 004004      MOV      #-1,TEMP
007400 009262 013374      INC      C(2)
007404 009767 003774      TST     TEMP
007410 001401                      BEQ      ,+4
007412 104006                      HLT
007414 104400                      SCOPE

007416 012767 000001 003700      MOV      #1,TEMP
007424 009360 013414      DEC      D(0)
007430 009767 003790      TST     TEMP
007434 001401                      BEQ      ,+4
007436 104006                      HLT
007440 104400                      SCOPE

007442 012767 000001 003734      MOV      #1,TEMP
007450 009460 013414      NEG      D(0)
007454 022767 177777 003722      CMP      #-1,TEMP
007462 001401                      BEQ      ,+4
007464 104006                      HLT
007466 104400                      SCOPE

007470 012767 177777 003706      MOV      #-1,TEMP
007476 000261                      BEQ
007500 009562 013374      ADC      C(2)
007504 009767 003674      TST     TEMP
007510 001401                      BEQ      ,+4
007512 104006                      HLT
007514 104400                      SCOPE

007516 012767 000001 003060      MOV      #1,TEMP
007524 000261                      BEQ
007526 009660 013414      SBC      D(0)
007532 009767 003646      TST     TEMP
007536 001401                      BEQ      ,+4
007540 104006                      HLT
007542 104400                      SCOPE
  
```

ITEST JMP INDIRECT

```

007544 010704      MOV      X7,X4
007546 062704      ADD      #10,X4
007552 000114      JRF      #4
007554 104006      HLT
007556 000240      NOP
007560 104400      SCOPE
  
```

ITEST INDIRECT ADDRESSING WITH INDEXING

007562	027727	003566	125292	ITEST COMPARE INSTRUCTION	
007570	001401			CMP	#0+2,#125292
007572	104006			BEQ	,+4
007574	104400			HLT	
				SCOPE	
007576	012777	125292	003602	ITEST MOVE INSTRUCTIONS	
007604	026767	003542	003572	MOV	#125292,@TEMP+2
007612	001401			CMP	@TEMP
007614	104006			BEQ	,+4
007616	104400			HLT	
				SCOPE	
007620	012767	177777	003556	ITEST BIC INSTRUCTION INDIRECT WITH INDEXING	
007626	042777	125292	003552	MOV	#-1,TEMP
007634	022767	052525	003542	BIC	#125292,@TEMP+2
007642	001401			CMP	#52525,TEMP
007644	104006			BEQ	,+4
007646	104400			HLT	
				SCOPE	
007650	012767	125292	003526	ITEST SUBTRACT INSTRUCTION INDIRECT WITH INDEXING	
007656	167777	003472	003522	MOV	#125292,TEMP
007664	005767	003514		SUB	@0+2,@TEMP+2
007670	001401			TST	TEMP
007672	104006			BEQ	,+4
007674	104400			HLT	
				SCOPE	
007676	012767	125292	003508	ITEST ADD INDIRECT WITH INDEXING	
007704	067777	003468	003474	MOV	#125292,TEMP
007712	026727	003466	177777	ADD	@A+0,@TEMP+2
007720	001401			CMP	TEMP,#-1
007722	104006			BEQ	,+4
007724	104400			HLT	
				SCOPE	
007726	012767	177777	003450	ITEST UNARYS INDIRECT WITH INDEXING	
007734	005077	003446		MOV	#-1,TEMP
007740	005767	003440		CLR	@TEMP+2
007744	001401			TST	TEMP
007746	104006			BEQ	,+4
007750	104400			HLT	
				SCOPE	
007752	012767	125292	003424	MOV	#125292,TEMP
007760	005177	003422		COM	@TEMP+2
007764	022767	052525	003412	CMP	#52525,TEMP
007772	001401			BEQ	,+4
007774	104006			HLT	
007776	104400			SCOPE	
010000	005067	003400		CLR	TEMP
010004	005377	003376		DEC	@TEMP+2

010010	026727	003370	177777	CMP	TEMP,#-1
010016	001401			BEQ	,+4
010020	104006			HLT	
010022	104400			SCOPE	
010024	012767	000001	003392	MOV	#1,TEMP
010032	005477	003350		NEG	@TEMP+2
010036	022767	177777	003340	CMP	#-1,TEMP
010044	001401			BEQ	,+4
010046	104006			HLT	
010050	104400			SCOPE	
010052	012767	177777	003324	MOV	#-1,TEMP
010060	000261			SEC	
010062	005377	003320		ADC	@TEMP+2
010066	005767	003312		TST	TEMP
010072	001401			BEQ	,+4
010074	104006			HLT	
010076	104400			SCOPE	
010100	012767	000001	003276	MOV	#1,TEMP
010106	000261			SEC	
010110	005677	003272		SRC	@TEMP+2
010114	005767	003264		TST	TEMP
010120	001401			BEQ	,+4
010122	104006			HLT	
010124	104400			SCOPE	
010126	012767	177777	003248	ITEST BIC INSTRUCTION	
010134	012703	177772		MOV	#-1,C
010140	000503			MOV	#-6,X3
010142	010304			ADD	R0,R3
010144	047374	013362	015404	MOV	X3,X4
010152	022767	052525	003214	BIC	@A(3),@TEMP(4)
010160	001401			CMP	#52525,C
010162	104006			BEQ	,+4
010164	104400			HLT	
				SCOPE	
010166	126162	013362	013362	ITEST COMPARE INSTRUCTION INDEXED	
010174	001401			CMPB	A(1),A(2)
010176	104006			BEQ	,+4
010178	104006			HLT	
010200	126261	013362	013362	CMPB	A(2),A(1)
010206	001401			BEQ	,+4
010210	104006			HLT	
010212	104400			SCOPE	
010214	116267	013362	003162	ITEST MOVE BYTE INSTRUCTION INDEXED	
010222	126727	003196	003125	MOVB	A(2),TEMP
010230	001401			CMPB	TEMP,#000125
010232	104006			BEQ	,+4
010234	104400			HLT	
				SCOPE	

TEST BICB INSTRUCTION FOR INDEXING

010236	012767	177777	003150	MOV	#=1,TEMP+10
010244	142762	125252	013404	BICB	#125252,TEMP(2)
010252	126727	003136	002525	CHPB	TEMP+10,#2525
010260	001401			BEQ	,+4
010262	104006			HLT	
010264	104400			SCOPE	

TEST UNARYS INDEXED (BYTE INSTRUCTIONS)

010266	012767	177777	003110	MOV	#=1,TEMP
010274	012703	177771		MOV	#=7,X3
010300	000503			ADD	R9,R3
010302	105063	013414		CLRB	D(3)
010306	026727	003072	003377	CMP	TEMP,#003377
010314	001401			BEQ	,+4
010316	104006			HLT	
010320	104400			SCOPE	
010322	012767	177777	003054	MOV	#=1,TEMP
010330	105160	013414		COMB	D(0)
010334	105767	003044		TSTB	TEMP
010340	001401			BEQ	,+4
010342	104006			HLT	
010344	104400			SCOPE	
010346	012767	177777	003030	MOV	#=1,TEMP
010354	105260	013414		INCB	D(0)
010360	026727	003020	177400	CMP	TEMP,#177400
010366	001401			BEQ	,+4
010370	104006			HLT	
010372	104400			SCOPE	
010374	012767	000001	003002	MOV	#1,TEMP
010402	105362	013374		DECB	C(2)
010406	105767	002772		TSTB	TEMP
010412	001401			BEQ	,+4
010414	104006			HLT	
010416	104400			SCOPE	
010420	012767	000001	002736	MOV	#1,TEMP
010426	105460	013414		NEGB	D(0)
010432	026727	002746	000377	CMP	TEMP,#377
010440	001401			BEQ	,+4
010442	104006			HLT	
010444	104400			SCOPE	
010446	012767	177777	002730	MOV	#=1,TEMP
010454	000261			SEC	
010456	105362	013374		ADDB	C(2)
010462	026727	002716	177400	CMP	TEMP,#177400
010470	001401			BEQ	,+4

010472	104006			HLT	
010474	104400			SCOPE	
010476	012767	000401	002700	MOV	#401,TEMP
010504	012704	177771		MOV	#=7,X4
010510	000504			ADD	R9,R4
010512	000261			SEC	
010514	105664	013414		SCCB	D(4)
010520	022767	000001	002656	CMP	#1,TEMP
010526	001401			BEQ	,+4
010530	104006			HLT	
010532	104400			SCOPE	

TEST INDIRECT ADDRESSING WITH INDEXING
 TEST COMPARE INSTRUCTION

010534	127727	002014	125252	CHPB	#0+2,#125252
010542	001401			BEQ	,+4
010544	104006			HLT	
010546	104400			SCOPE	
010550	122777	125252	002576	CHPB	#125252,#0+2
010556	001401			BEQ	,+4
010560	104006			HLT	
010562	104400			SCOPE	

TEST MOVE INSTRUCTIONS

010564	117703	002564		MOVB	#0+2,X3
010570	122703	125252		CHPB	#125252,X3
010574	001401			BEQ	,+4
010576	104006			HLT	
010600	104400			SCOPE	
010602	112777	125252	002576	MOVB	#125252,@TEMP+2
010610	126767	002536	002566	CHPB	@TEMP
010616	001401			BEQ	,+4
010620	104006			HLT	
010622	104400			SCOPE	

TEST BIC INSTRUCTION INDIRECT WITH INDEXING

010624	012704	177777		MOV	#=1,X4
010630	147704	002520		BICB	#0+2,X4
010634	120427	052525		CHPB	X4,#52525
010640	001401			BEQ	,+4
010642	104006			HLT	
010644	104400			SCOPE	
010646	012767	177777	002530	MOV	#=1,TEMP
010654	142777	125252	002524	BICB	#125252,@TEMP+2
010662	122767	052525	002514	CHPB	#52525,TEMP
010670	001401			BEQ	,+4
010672	104006			HLT	
010674	104400			SCOPE	

TEST UNARYS INDIRECT WITH INDEXING

010676	012767	177777	002500	MOV	#=1,TEMP
010704	105077	002476		CLRB	@TEMP+2
010710	105767	002470		YSTB	TEMP
010714	001401			BEQ	,+4
010716	104006			HLT	
010720	104400			SCOPE	
010722	012767	125252	002454	MOV	#125252,TEMP
010730	105177	002452		COMB	@TEMP+2
010734	122767	052525	002442	CHPB	#052525,TEMP
010742	001401			BEQ	,+4
010744	104006			HLT	
010746	104400			SCOPE	
010750	005067	002430		CLR	TEMP
010754	105277	002426		INCB	@TEMP+2
010760	122767	000001	002416	CHPB	#1,TEMP
010766	001401			BEQ	,+4
010770	104006			HLT	
010772	104400			SCOPE	
010774	005067	002404		CLR	TEMP
011000	105377	002402		DECB	@TEMP+2
011004	126727	002374	177777	CHPB	TEMP,#=1
011012	001401			BEQ	,+4
011014	104006			HLT	
011016	104400			SCOPE	
011020	012767	000001	002396	MOV	#1,TEMP
011026	105477	002354		NEGB	@TEMP+2
011032	122767	177777	002344	CHPB	#=1,TEMP
011040	001401			BEQ	,+4
011042	104006			HLT	
011044	104400			SCOPE	
011046	012767	177777	002330	MOV	#=1,TEMP
011054	000261			SEC	
011056	105577	002324		ADCB	@TEMP+2
011062	022767	177400	002314	CMP	#177400,TEMP
011070	001401			BEQ	,+4
011072	104006			HLT	
011074	105767	002304		YSTB	TEMP
011100	001401			BEQ	,+4
011102	104006			HLT	
011104	104400			SCOPE	
011106	012767	000001	002270	MOV	#1,TEMP
011114	000261			SEC	
011116	105377	002264		DECB	@TEMP+2
011122	005767	002256		TST	TEMP
011126	001401			BEQ	,+4
011130	104006			HLT	
011132	104400			SCOPE	

TEST OF COMBINED INDEXING AND INDIRECT

011134	012703	177772		MOV	#=6,X3
011140	012704	000002		MOV	#=2,X4
011144	060503			ADD	R3,R3
011146	060504			ADD	R3,R4
011150	127374	013302	013302	CHPB	#A(3),#A(4)
011156	001401			BEQ	,+4
011160	104006			HLT	
011162	104400			SCOPE	

TEST BIC INSTRUCTION

011164	012703	000006		MOV	#=6,X3
011170	012767	177777	002206	MOV	#=1,TEMP
011176	060503			ADD	R3,R3
011200	147367	013302	002176	BICB	#A(3),TEMP
011206	122767	125252	002170	CHPB	#125252,TEMP
011214	001401			BEQ	,+4
011216	104006			HLT	
011220	104400			SCOPE	

011222	012704	177772		MOV	#=6,X4
011226	012767	177777	002140	MOV	#=1,C
011234	060504			ADD	R3,R4
011236	142774	125252	013404	BICB	#125252,@TEMP(4)
011244	126727	002124	000125	CHPB	C,#000125
011252	001401			BEQ	,+4
011254	104006			HLT	
011256	104400			SCOPE	

011260	012703	013354		MOV	#B+2,X3
011264	060503			ADD	R3,R3
011266	023367	002060		CMP	#(3)+,B
011272	001401			BEQ	,+4
011274	104006			HLT	
011276	104400			SCOPE	

IADDRESS OF ADDRESS OF B

011300	012704	013356		MOV	#B+4,X4
011304	060504			ADD	R3,R4
011306	025467	002040		CMP	#=(4),B
011312	001401			BEQ	,+4
011314	104006			HLT	
011316	104400			SCOPE	

011320	012703	013400		MOV	#Q+4,X3
011324	060503			ADD	R3,R3
011326	012767	177777	002040	MOV	#=1,C
011334	105053			CLRB	#=(3)
011336	026727	002032	177400	CMP	C,#177400
011344	001401			BEQ	,+4
011346	104006			HLT	
011350	104400			SCOPE	

011352	012767	177777	002014	MOV	#=1,C
--------	--------	--------	--------	-----	-------

011360	012703	177772		MOV	#=6,X3	
011364	060503			ADD	R3,R3	
011366	010304			MOV	R3,R4	
011370	147374	013362	013404	BICB	#A(3),@TEMP(4)	
011376	022767	177525	001770	CHP	#177525,C	
011404	001401			BEQ	,+4	
011406	104006			HLT		
011410	104400			SCOPE		
:TEST JSR INSTRUCTION						
011412	012703	011424		MOV	#TJSR1,R3	
011416	060503			ADD	R3,R3	
011420	004767	000002		JSR	X7, TJSR2	:PLACE PC ON STACK
011424	000404			TJSR1: BR	TJSR3	:RETURN HERE ON RTS X7
011426	021003			TJSR2: CHP	#X6,R3	:CHECK FOR CORRECT PC ON STACK
011430	001401			BEQ	,+4	
011432	104006			HLT		:INCORRECT PC ON STACK
011434	000207			RTS	X7	:RETURN TO INST AFTER JSR
011436	104400			TJSR3: SCOPE		
011440	000257			CCC		
011442	012704	011452		MOV	#TJSR4,R4	
011446	060504			ADD	R3,R4	
011450	004717			JSR	X7,@X7	:INSTRUCTION UNDER TEST
011452	021004			TJSR4: CHP	#X6,R4	:TEST THE STACK
011454	001401			BEQ	,+4	
011456	104006			HLT		:PC OF JSR DID NOT GO TO STACK
011460	005726			TST	(6)+	:REPOSITION THE STACK
011462	104400			SCOPE		
:TEST NESTED SUBROUTINES						
011464	000257			CCC		:CLEAR CONDITION CODES
011466	004767	001552		JSR	X7,SUBR6	
011472	100401			BMI	,+4	
011474	104006			HLT		:JSR OR RTS FAILED
011476	001401			BEQ	,+4	
011500	104006			HLT		:JSR OR RTS FAILED
011502	102401			BVS	,+4	
011504	104006			HLT		:JSR OR RTS FAILED
011506	103401			BCS	,+4	
011510	104006			HLT		:JSR OR RTS FAILED
011512	104400			SCOPE		
:TEST ROTATE ODD BYTE						
011514	104400			SCOPE		
011516	000257			CCC		:CLEAR "C"
011520	012767	123456	001656	MOV	#123456,TEMP	
011526	106067	001653		RORB	TEMP+1	:ROTATE ODD BYTE
011532	103401			BCS	,+4	
011534	104006			HLT		:C NOT SET
011536	102401			BVS	,+4	
011540	104006			HLT		:V NOT SET
011542	022767	051456	001034	CHP	#051456,TEMP	
011550	001401			BEQ	,+4	

011552	104006			HLT		:ROTATE FAILED
011554	104400			SCOPE		
011556	000277			SCC		:SET C
011560	012767	123456	001616	MOV	#123456,TEMP	
011566	106067	001613		RORB	TEMP+1	
011572	103401			BCS	,+4	
011574	104006			HLT		:C NOT SET
011576	102001			BVC	,+4	
011600	104006			HLT		:V NOT CLEARED
011602	022767	191456	001574	CHP	#191456,TEMP	
011610	001401			BEQ	,+4	:ROTATE FAILED
011612	104006			HLT		
011614	104400			SCOPE		
011616	000257			CCC		
011620	012767	123456	001556	MOV	#123456,TEMP	
011626	106167	001553		ROLB	TEMP+1	
011632	103401			BCS	,+4	
011634	104006			HLT		:C NOT SET
011636	102401			BVS	,+4	
011640	104006			HLT		:V NOT SET
011642	022767	047056	001534	CHP	#047056,TEMP	
011650	001401			BEQ	,+4	:ROTATE BYTE FAILED
011652	104006			HLT		
011654	104400			SCOPE		
011656	000277			SCC		:SET C
011660	012767	123456	001516	MOV	#123456,TEMP	
011666	106167	001513		ROLB	TEMP+1	
011672	103401			BCS	,+4	
011674	104006			HLT		:C NOT SET
011676	102401			BVS	,+4	
011700	104006			HLT		:V NOT SET
011702	022767	047456	001474	CHP	#047456,TEMP	
011710	001401			BEQ	,+4	
011712	104006			HLT		:ROTATE ODD BYTE FAILED
011714	104400			SCOPE		
011716	000257			CCC		:CLEAR C
011720	012767	177777	001456	MOV	#=1,TEMP	
011726	106267	001453		ASRB	TEMP+1	
011732	103401			BCS	,+4	
011734	104006			HLT		:C NOT SET
011736	102001			BVC	,+4	
011740	104006			HLT		:V NOT CLEARED
011742	026727	001434	177777	CHP	TEMP,#-1	
011750	001401			BEQ	,+4	
011752	104006			HLT		:SHIFT FAILED
011754	104400			SCOPE		
011756	000277			SCC		
011760	012767	177777	001416	MOV	#=1,TEMP	
011766	106367	001413		ASLB	TEMP+1	

011772	103401			BCS	.+4	
011774	104006			HLT		IC NOT SET
011776	102001			BVC	.+4	
012000	104006			HLT		IV NOT CLEARED
012002	026727	001376	177377	CHP	TEMP,#177377	
012010	001401			SEQ	.+4	
012012	104006			HLT		ISHIPT BYTE FAILED
012014	104400			SCOPE		
IEND OF RESTRICTION ON R0-R2						
012016	005067	001322		CLR	ICOUNT	IND ITERATION
012022	032767	004000	166776	BIT	#4000,SREG2	
012030	001076			BNE	TSRT2A	
ITEST ALL COMBINATIONS OF NUMBERS WITH COMPARE INSTRUCTION						
012032	005000			COMPAR	CLR X0	IINIT X0
012034	005001				CLR X1	IINIT X1
012036	020001			CHP1	CHP X0,X1	IARE THEY EQUAL?
012040	001401			SEQ	.+4	
012042	104006			HLT		IR0 AND R1 DID NOT COMPARE
012044	020027	177777		CHP	X0,#=1	IAT UPPER LIMIT
012050	001403			SEQ	CHP2	IYES EXIT
012052	005200			INC	X0	IINCREMENT TO NEXT NUMBER
012054	005201			INC	X1	
012056	000767			BR	CHP1	
012060	104400			CHP2	SCOPE	
ITEST ROTATING ALL NUMBERS						
012062	012767	177777	000132	MOV	#=1,REFF	IINITIALIZE BASE NUMBER
012070	005267	000126		TSROT	REFF	IINCREMENT NUMBER
012074	004767	000014		JSR	X7,ROTALL	IGO TO COMPARE ROUTINE
012100	026727	000116	177777	CHP	REFF,#=1	ITEST ALL VALUES
012106	001370			BNE	TSROT	INDO TEST THEM ALL
012110	104400			SCOPE		
012112	000445			BR	TSRT2A	IWE ARE DONE
012114	016767	000102	000102	ROTALL	MOV REFF,TEST	
012122	006067	000076		ROR	TEST	
012126	006067	000072		ROR	TEST	
012132	006067	000066		ROR	TEST	
012136	006167	000062		ROL	TEST	
012142	006167	000056		ROL	TEST	
012146	006167	000052		ROL	TEST	
ITEST COMBINATION OF N, C AND V						
012152	100004			BPL	.+12	
012154	103007			BCC	.+20	I2=1
012156	102013			BVC	.+30	I2=1, C=1
012160	104006			HLT		I2=C, BUT V=1
012162	000411			BR	.+24	
012164	103006			BCC	.+16	I2=0
012166	102407			BVS	.+20	I2=0, C=1
012170	104006			HLT		I2 NOT EQUAL C, V=1
012172	000405			BR	.+14	

012174	102404			BVS	.+12	I2=1, C=0
012176	104006			HLT		I2 NOT EQUAL C, V=1
012200	000402			BR	.+6	
012202	102001			BVC	.+4	I2=0, C=0
012204	104006			HLT		I2=C, BUT V=1
012206	026767	000012	000006	CHP	TEST,REFF	
012214	001401			SEQ	.+4	
012216	104006			HLT		IINITIAL NOT EQUAL TO FINAL
012220	000207			RTS	X7	
012222	000000			REFF	0	
012224	000000			TEST	0	
	012222				REF=REFF	
ITEST ROTATING BYTE EVEN/ODD, ALL NUMBERS						
012226	012767	177777	177766	TSRT2A	MOV #=1,REFF	
012234	005267	177762		TSROT2	INC REFF	
012240	004767	000016		JSR	X7,ROTBE	
012244	004767	000120		JSR	X7,ROTBO	
012250	022767	177777	177744	CHP	#=1,REFF	
012256	001366			BNE	TSROT2	
012260	000503			BR	ROTEN	
012262	016767	177734	177734	ROTBE	MOV REFF,TEST	
012270	106067	177730		RORB	TEST	IROTATE BYTE EVEN
012274	106067	177724		RORB	TEST	
012300	106067	177720		RORB	TEST	
012304	106167	177714		ROLB	TEST	
012310	106167	177710		ROLB	TEST	
012314	106167	177704		ROLB	TEST	
ITEST COMBINATION OF N, C AND V						
012320	100004			BPL	.+12	
012322	103007			BCC	.+20	I2=1
012324	102013			BVC	.+30	I2=1, C=1
012326	104006			HLT		I2=C, BUT V=1
012330	000411			BR	.+24	
012332	103006			BCC	.+16	I2=0
012334	102407			BVS	.+20	I2=0, C=1
012336	104006			HLT		I2 NOT EQUAL C, V=1
012340	000405			BR	.+14	
012342	102404			BVS	.+12	I2=1, C=0
012344	104006			HLT		I2 NOT EQUAL C, V=1
012346	000402			BR	.+6	
012350	102001			BVC	.+4	I2=0, C=0
012352	104006			HLT		I2=C, BUT V=1
012354	026767	177644	177640	CHP	TEST,REFF	
012362	001401			SEQ	.+4	
012364	104006			HLT		
012366	000207			RTS	X7	
012370	106067	177631		ROTBO	RORB TEST+1	IROTATE BYTE ODD
012374	106067	177625		RORB	TEST+1	
012400	106067	177621		RORB	TEST+1	
012404	106167	177615		ROLB	TEST+1	
012410	106167	177611		ROLB	TEST+1	
012414	106167	177605		ROLB	TEST+1	
ITEST COMBINATION OF N, C AND V						

012420	100004			BPL	.*12	
012422	103007			BCC	.*20	IZ=1
012424	102013			BVC	.*30	IZ=1, C=1
012426	104006			HLT		IZ=C, BUT V=1
012430	000411			BR	.*24	
012432	103006			BCC	.*10	IZ=0
012434	102407			BVS	.*20	IZ=0, C=1
012436	104006			HLT		IZ NOT EQUAL C, V=1
012440	000405			BR	.*14	
012442	102404			BVS	.*12	IZ=1, C=0
012444	104006			HLT		IZ NOT EQUAL C, V=1
012446	000402			BR	.*06	
012450	102001			BVC	.*04	IZ=0, C=0
012452	104006			HLT		IZ=C, BUT V=1
012454	026767	177544	177540	CMP	TEST, REFF	
012462	001401			BEQ	.*04	
012464	104006			HLT		
012466	000207			RTB	X7	
012470	104400					

ROTEIN1 SCOPE

INHIBIT SOME TESTS WHEN SW11 IS SET

012472	032767	004000	106326	BIT	#4000, SREG2	
012500	001402			BEQ	TSTARI	
012502	000167	000502		JMP	DONE	

ADD AND SUBTRACT ALL NUMBERS AGAINST FIXED NUMBERS
 IA+0=C, C-A=B, BF SHOULD EQUAL BI

012506	011667	000066		TSTARI	MOV	#06, NUMA
012512	012767	000001	177502		MOV	#1, REFF
012520	005267	177476		ARITST1	INO	REF
012524	004767	000012			JSR	X7, ADSUB
012530	022767	177777	177464		CMP	#-1, REFF
012536	001370				BNE	ARITST
012540	000420				BR	ARIEND
012542	016767	177454	177454	ADSUB1	MOV	REF, TEST
012550	066767	000024	177446		ADD	NUMA, TEST
012556	106767	000016	177440		SUB	NUMA, TEST
012564	026767	177432	177432		CMP	REF, TEST
012572	001401				BEQ	.*04
012574	104006				HLT	
012576	000207				RTS	X7
012600	000000					
012602	104400					

NUMA1
 ARIEND1 SCOPE

TEST COMPLEMENTING ALL NUMBERS

012604	005067	000574			CLR	TEMP	I0BASE DATA
012610	005067	000574			CLR	TEMP+4	I0BASE REFERENCE
012614	005167	000564		TCOM1	COM	TEMP	I0COMPLEMENT DATA
012620	005367	000564			DEC	TEMP+4	I0DECREMENT REFERENCE
012624	026767	000554	000556		CMP	TEMP, TEMP+4	I0COMPARE
012632	001401				BEQ	.*04	I0TEST
012634	104006				HLT		I0COMPLEMENT OR DECREMENT FAILED
012636	005167	000542			COM	TEMP	
012642	005267	000536			INC	TEMP	I0INCREMENT AND TEST FOR DONE

012646	001362			BNE	TCOM		I0NOT FINISHED GO LOOP
012650	104400				SCOPE		
I0TEST COMB (EVEN BYTE)							
012652	005067	000526			CLR	TEMP	I0BASE DATA
012656	005067	000526			CLR	TEMP+4	I0REFERENCE DATA
012662	105167	000516		TCOM21	COMB	TEMP	
012666	005367	000516			DEC	TEMP+4	
012672	126767	000506	000510		CMPB	TEMP, TEMP+4	I0COMPARE
012700	001401				BEQ	.*04	
012702	104006				HLT		I0COMPLEMENT OR INCREMENT BYTE FAILED
012704	105167	000474			COMB	TEMP	
012710	105267	000470			INCB	TEMP	
012714	001362				BNE	TCOM2	
012716	104400				SCOPE		
I0TEST COMB (ODD BYTE)							
012720	005067	000460			CLR	TEMP	I0BASE DATA
012724	005067	000460			CLR	TEMP+4	I0REFERENCE DATA
012730	105167	000451		TCOM31	COMB	TEMP+1	I0ODD BYTE
012734	005367	000450			DEC	TEMP+4	
012740	126767	000441	000442		CMPB	TEMP+1, TEMP+4	
012746	001401				BEQ	.*04	
012750	104006				HLT		I0COMPLEMENT BYTE FAILED
012752	105167	000427			COMB	TEMP+1	
012756	105267	000423			INCB	TEMP+1	
012762	001362				BNE	TCOM3	
012764	104400				SCOPE		
I0TEST COMPARE ALL VALUE EVEN BYTE WITH ODD							
012766	005067	000412			CLR	TEMP	I0BASE VALUE
012772	126767	000406	000405	TSCOMB1	CMPB	TEMP, TEMP+1	I0COMPARE
013000	001401				BEQ	.*04	
013002	104006				HLT		I0COMPARE FAILED
013004	002001				BGE	.*04	
013006	104006				HLT		I0V IS NOT = TO N
013010	003401				BLE	.*04	
013012	104006				HLT		I0V IS SET
013014	062767	000401	000362		ADD	#401, TEMP	
013022	022767	177777	000354		CMP	#-1, TEMP	
013030	001360				BNE	TSCOMB	
013032	104400				SCOPE		
I0TEST TO SEE IF I/O DEVICES WERE SELECTED							
013034	016767	105766	000026		MOV	SREG2, CKWAIT	I0GET DEVICE INHIBIT FLAGS
013042	005167	000022			COM	CKWAIT	I0COMPLEMENT SO DEVICES SELECTED
013046	032767	000101	000014		BIT	#161, CKWAIT	I0CORRESPOND TO ONES
013054	001400				BEQ	WAIT4	I0ANY DEVICES BESIDES TC11 SELECTED?
013056	000001				WAIT		I0BRANCH IF NO
013060	000001				WAIT		I0INTERRUPTS WILL OCCUR IF DEVICES
013062	000001				WAIT		I0BESIDES TC11 ARE SELECTED
013064	000001				WAIT		
013066	000401				BR	.*04	

```

013070 000000 CKWAIT: 0
013072 104400 WAIT4: SCOPE

ITEST SWAB
013074 012767 000200 177122 MOV #0200,TEST
013102 000367 177110 SWAB TEST
013106 100001 BPL ,+4
013110 104006 HLT
013112 001401 BEQ ,+4
013114 104006 HLT
013116 000367 177102 SWAB TEST
013122 100401 BHI ,+4
013124 104006 HLT
013126 001001 BNE ,+4
013130 104006 HLT
013132 104400 SCOPE

ITEST ALL COMBINATIONS OF SWAB
013134 000067 177064 CLR TEST INUMBER UNDER TEST
013140 000067 177056 CLR REF IREFERENCE NUMBER
013144 000367 177054 SWABA: SWAB TEST IOPERATION UNDER TEST
013150 026767 177050 177044 CMP TEST,REF ITEST SWAB INSTRUCTION
013156 001401 BEQ ,+4
013160 104006 HLT ISWAB FAILED
013162 000367 177036 SWAB TEST
013166 000267 177030 INC REF IINCREMENT REFERENCE NUMBER
013172 100267 177027 INCB TEST+1 IINC TEST NUMBER
013176 001362 BNE SWABA ILOOP TILL DONE
013200 104400 SCOPE
013202 012767 000400 000134 MOV #400,ICOUNT
  
```

```

IEND OF USER CODE IN BANK
ICALL KERNEL
IALTERED IN CORE EXPANSION
013210 104012 DONE: EOB
013212 000240 NOP ITO ALLOW CORE EXPANSION TO PATCH IN JMP

IGROUP OF NESTED SUBROUTINES
013214 000207 SUBR1: RTS X7 IONE INSTRUCTION
013216 000277 SUBR2: SCC X7 IONE DEEP
013220 000207 RTS X7
013222 004767 177770 SUBR3: JSR X7,SUBR2 ITWO DEEP
013226 000207 RTS X7
013230 004767 177766 SUBR4: JSR X7,SUBR3 ITHREE DEEP
013234 000207 RTS X7
013236 004767 177766 SUBR5: JSR X7,SUBR4 IFOUR DEEP
013242 000207 RTS X7
013244 004767 177766 SUBR6: JSR X7,SUBR5 IFIVE DEEP
013250 000207 RTS X7
  
```

SBTTL UTILITY ROUTINES

```

ISCOPE AND/OR ITERATION LOOP FOR EACH TEST
013252 032767 002000 100546 SCOPE: BIT #2000,SREG2 IINHIBIT PROCESSOR TESTS?
013260 001403 BEQ ,+10 INO
013262 022626 CMP (SP)+,(SP)+ IYES
013264 000167 167702 JMP MAIN IYES,SCOPE
013270 032767 040000 100530 BIT #4000,SREG2 ITEST SR FOR SCOPE
013276 001012 BNE SCOPEB IYES,SCOPE
013300 032767 004000 100520 BIT #4000,SREG2 INO-TEST FOR ITERATION
013306 001011 BNE SCOPEF IINHIBIT ITERATION
013310 026767 000032 000026 CMP SCOPEF,ICOUNT ICOMPARE CURRENT COUNT TO MAX NUMBER
013316 100005 BPL SCOPEF IEXIT-DONE
013320 000267 000022 ING SCOPEF IINCREMENT COUNT
013324 016716 000020 SCOPEB: MOV RETURN,#SP
013330 000002 RTI
013332 000067 000010 SCOPEF: CLR SCOPEF ICLEAR COUNT
013336 011667 000006 MOV #06,RETURN ISAVE SCOPE RETURN POINTER
013342 000002 RTI IRETURN INLINE=NEXT TEST
013344 000400 ICOUNT: 400 ITERATION COUNT
013346 000000 SCOPEF: 0 ICOUNT LOCATION FOR ITERATION LOOP
013350 000000 RETURN: 0 IADDRESS OF LAST TEST
  
```

```

IFIXED VALUES FOR USE IN TEST
013352 120252 BI 120252
013354 013352 0
013356 002525 02525 IADDRESS OF B
013362 177777 AI 0
013364 013366 ,#B+10
013366 013366 ,#A+4
013366 120252 ,#A+4 120252
013370 013372 A+10 IADDRESS OF A+10
013372 002525 002525
  
```

```

IFOR STORAGE
013374 000000 CI 0
013376 013374 C IADDRESS OF C
013404 013404 ,#C+10
013404 000000 TEMPI: 0
013406 013404 TEMP IADDRESS OF TEMP
013412 013414 ,#TEMP+6
013414 000000 DI 0 IADDRESS OF TEMP+10 OR "0"
  
```

```

ISUBROUTINE TO INITIALIZE ALL PAGES TO NR, BANK 0, 1 PAGE. UP
013416 010146 NRALL: MOV R1,=(R0) ISAVE REGISTERS
013420 010246 MOV R2,=(R0)
013422 010346 MOV R3,=(R0)
013424 012701 001010 MOV #IPORTAB,R1 IR1 HOLDS ADDRESS OF CURRENT POSITION
013430 012703 000040 NRLOOP: MOV #32,,R3 IIN TABLE OF ADDRESSES
013434 012102 MOV (R1)+,R2 IR3 USED AS COUNTER
013436 000022 CLR (R2)+ IR2 CONTAINS ADDRESS OF PDR OR
013440 077302 SOB R3,,-2 IPAR TO BE CLEARED
ICLEAR ALL ASR'S FOR THIS MODE
  
```



```

013442 020127 001022      CMP      R1,#IPDREND      ;CHECK FOR DONE
013446 003770      BLE      NRLOOP          ;CLEAR ALL IN NEXT MODE IF NOT DONE
013450 012603      MOV      (R6)+,R3
013452 012602      MOV      (R6)+,R2
013454 012601      MOV      (R6)+,R1
013456 000207      RTS      X7

;PROGRAMMED INTERRUPT REQUEST SERVICE ROUTINE
013460 117737 165412 177776  PIRSRV: MOVB   #PIR,#SPSR      ;SET PRIORITY
013466 032777 000002 165402  BIT     #2,#PIR          ;CHECK FOR LEVEL 2 VS LEVEL 3
013474 001006      BNE     PIR3           ;IF SET, IS LEVEL 3
013476 042777 002000 165372  BIC     #2000,#PIR
013504 004767 171740      JSR     X7,REGEN       ;LEVEL 2-REGENERATE WRITE BUFFER
013510 000002      RTI
013512 042777 004000 169356  PIR3:  BIC     #4000,#PIR
013520 004767 172640      JSR     X7,CCK        ;LEVEL 3-CHECK DATA READ FROM DTA
013524 000002      RTI

;EMT HANDLER
;FIRST 3 CALLS LEFT OPEN IN TABLE FOR EASY PATCHES
013526 162716 000002      EMTSRV: SUB     #2,#SP      ;GET CALL
013532 006576 000000      MFPD   #1,#SP
013536 012667 000022      MOV     (SP)+,EPC
013542 062716 000002      ADD     #2,#SP
013546 105067 000113      CLRB   EPC+1          ;SAVE OFFSET ONLY
013552 062767 013566 000004  ADD     #EMTAB,EPC     ;POINT TO TABLE OF ADDRESSES
013560 017707 000000      MOV     #EPC,PC       ;JUMP TO DESIRED ROUTINE
013564 000000      EPC:   0
          PATCH100
          PATCH200
          PATCH300
013566 000000      EMTAB: PATCH1         ;PATCH IN ADDRESS OF ROUTINE
013570 000000      PATCH2
013572 000000      PATCH3
013574 016250      PRINT          ;ERROR PRINTOUT
013576 013602      RTIEX        ;RTI VIA KERNEL
013600 013670      EOBSSRV      ;END OF BANK

013602 022626      RTIEX: CMP     (SP)+,(SP)+  ;FIX KERNEL STACK POINTER
013604 032737 030000 177776  BIT     #30000,#SPSR   ;PREVIOUS KERNEL?
013612 001001      BNE     .+4          ;BRANCH IF NOT
013614 000002      RTI           ;RETURN IF FROM KERNEL
013616 106506      MFPD   SP           ;GET PREVIOUS STACK POINTER
013620 012667 000042      MOV     (SP)+,SPSAV   ;RESTORE KERNEL STACK
013624 062767 000000 000034  ADD     #2,SPSAV
013632 106577 000030      MFPD   #SPSAV
          ;PUSH PS ON KERNEL STACK
013636 162767 000002 000022  SUB     #2,SPSAV
013644 106577 000016      MFPD   #SPSAV
          ;PUSH PC ON KERNEL STACK
013650 042767 000004 000010  ADD     #4,SPSAV
013656 016746 000004      MOV     SPSAV,(SP)
013662 106606      MFPD   SP
          ;RESTORE CORRECTED PREVIOUS STACK
          ;POINTER
          ;RETURN
013664 000002      RTI
013666 000000      SPSAV: 0
  
```

```

;END OF BANK SERVICE- DOES ALL MAPPING FUNCTIONS AFTER STARTUP
;CALLED AT THE END OF EACH BACKGROUND CPU PASS THRU 4K (VIRTUAL) UNLESS
;CORE EXPANDED (IN WHICH CASE IT IS CALLED AFTER EACH PASS THRU ALL MEMORY
;TO 20K)
013670 113737 177571 001027  EOBSSRV: MOVB   #MSR+1,#MSREG2+1  ;READ SWITCHES AGAIN
013676 032737 000001 001024  BIT     #1,#MSREG1    ;K11-C INHIBITED?
013704 001503      BEQ     EOB2         ;NO - CONTINUE
013706 004767 003024      JSR     X7,BELL      ;YES-SIGNAL END OF PASS
013712 042766 000020 000002  BIC     #20,2(SP)    ;CLEAR TRACE BIT OF STATUS ON STACK
013720 012737 000016 000014  MOV     #16,#M14     ;SETUP TRACE RETURN TO CAUSE HALT
013726 005037 000016      CLR     #M16
          ;IF A TRACE TRAP OCCURS
013732 032737 010000 001026  BIT     #10000,#MSREG2 ;INHIBIT TRACE TRAPPING?
013740 001011      BNE     EOB1A       ;YES - BRANCH
013742 005167 165122      COM     TRPB        ;SWITCH TRACE FLAG
013746 100006      BPL     EOB1A       ;IF NOT SET, LEAVE TRACE OFF
013750 052766 000020 000002  BIS     #20,2(SP)    ;IF SET, SET TRACE BIT OF STATUS ON STACK
013756 012737 016246 000014  MOV     #TRTRP,#M14
013764 105267 165114      INCB   DISPLAY
          ;E0B1A:
013770 016737 165110 177570  MOV     DISPLAY,#MSR
013776 032737 000040 001024  BIT     #40,#MSREG1
          ;E0B1C:
014004 001040      BNE     EOB1C       ;CORE EXPANSION INHIBITED?
014006 013746 000006      MOV     #6,-(SP)    ;YES, BRANCH OVER
014012 013746 000004      MOV     #4,-(SP)    ;NO, COPY NEW DYNAMIC SWITCHES TO
          ;ALL BANKS UP TO 20K
014016 012737 014066 000004  MOV     #EOB1B,#M4
014024 005037 000006      CLR     #M6
014030 013701 001026      MOV     #MSREG2,R1
014034 010137 021026      MOV     R1,#MSREG2+20000
014040 010137 041026      MOV     R1,#MSREG2+40000
014044 010137 061026      MOV     R1,#MSREG2+60000
014050 010137 081026      MOV     R1,#MSREG2+80000
014054 010137 101026      MOV     R1,#MSREG2+100000
014060 010137 121026      MOV     R1,#MSREG2+120000
014064 010137 141026      MOV     R1,#MSREG2+140000
014066 000401      BR     .+4
014070 022626      EOB1B: CMP     (SP)+,(SP)+
014074 012637 000004      MOV     (SP)+,#4
014080 012637 000006      MOV     (SP)+,#6
          ;E0B1C:
014100 012716 006456      MOV     #BEGINX,(SP) ;SETUP PC TO RETURN TO BEGINX
014104 000002      RTI
014106 012716      EOB1C: MOV     #BEGIN,(SP) ;IF NO CORE EXPANSION, RETURN TO
          ;BEGIN
014112 000002      RTI
014114 032737 000002 001024  EOB2:  BIT     #2,#MSREG1
          ;E0B1:
014122 001271      BNE     EOB1        ;USER/SUPERVISOR/KERNEL INHIBITED?
014124 032737 001000 001026  BIT     #1000,#MSREG2 ;YES - SET PC AND RETURN
014132 001132      BNE     SUPD1       ;STOP CYCLING SUPERVISOR PAGES?
014134 032737 001064 000734  CMP     #MSIPDR,#MSIPDR6 ;YES- BRANCH
          ;NO-LAST SUPERVISOR PAGE DONE (PAGE 6)?
014142 001455      BEQ     SUPCY       ;YES, BRANCH TO RECYCLE
014144 016701 164714      MOV     CSIPDR,R1
014150 005721      TST   (R1)+
          ;ADDRESS NEXT_PDR
014152 012711 077406      MOV     #77400,#R1
014156 012711 077406 000020  MOV     #77400,20(R1)
014164 062777 020000 164430  ADD     #20000,#TRPVC ;UPDATE SUPERVISOR I/O VECTORS
          ;TO REFERENCE NEXT PAGE
014172 062777 020000 164412  ADD     #20000,#TRVC
  
```

014200	062777	020000	164424	ADD	#20000,#KWLVC	
014206	032737	000040	001026	BIT	#40,#SREG2	IUPDATE LP#1 VECTOR ONLY IF USED
014214	001003			BNE	,#10	
014216	062777	020000	164416	ADD	#20000,#LPLVC	
014224	013737	177776	001072	MOV	#PPSAV,#PPSAV	IUPDATE SUPERVISOR STACK POINTER
014232	042737	030000	177776	BIC	#30000,#PPSR	
014240	092737	010000	177776	BIS	#10000,#PPSR	
014246	106506			SP		
014250	062716	020000		ADD	#20000,(SP)	
014254	106606			HTPD	SP	
014256	013737	001072	177776	MOV	#PPSAV,#PPSR	
014264	005061	177776		CLR	-2(R1)	IMAKE PREVIOUS PAGE NON-RESIDENT
014270	005061	000016		CLR	16(R1)	
014274	000447			BR	SUPDON	
014276	016701	164426		SUPCYC: MOV	SIPDR0,R1	IMAP TO FIRST SUPERVISOR PAGE
014302	012711	077406		MOV	#77406,#R1	
014306	012761	077406	000020	MOV	#77406,20(R1)	
014314	042777	160000	164300	BIC	#160000,#ITPVC	
014322	042777	160000	164262	BIC	#160000,#ITTRVC	
014330	042777	160000	164274	BIC	#160000,#KWLVC	
014336	042777	160000	164276	BIC	#160000,#LPLVC	
014344	013737	177776	001072	MOV	#PPSR,#PPSAV	IMAP SUPERVISOR STACK TO PAGE 0
014352	042737	030000	177776	BIC	#30000,#PPSR	
014360	092737	010000	177776	BIS	#10000,#PPSR	
014366	106506			SP		
014370	042716	160000		BIC	#160000,(SP)	
014374	106606			HTPD	SP	
014376	013737	001072	177776	MOV	#PPSAV,#PPSR	
014404	005061	000014		CLR	14(R1)	IMAP SUPERVISOR PAGE 6 NON-RESIDENT
014410	005061	000034		CLR	34(R1)	
014414	010137	001064		SUPDON: MOV	R1,#SIPDR	
014420	042737	000340	177776	SUPD1: BIC	#340,#PPSR	
014426	032737	000420	001024	BIT	#420,#SREG01	IINHIBIT CYCLING ACCESS KEY OF S77
014434	001010			BNE	SUPD2	IYES, SKIP
014436	005277	164302		INC	#9DPDR7	INO, SETUP NEXT KEY IN SUPERVISOR
014442	042777	000010	164274	BIC	#10,#9DPDR7	IPAGE 7
014450	017777	164270	164264	MOV	#9DPDR7,#SIPDR7	
014456	032737	000004	001024	SUPD2: BIT	#4,#SREG1	IINHIBIT 4K AS 32K?
014464	001402			BEQ	,#6	
014466	000167	177214		JMP	EOB1	IYES - SET PC AND RETURN
014472	023737	001054	000720	CHP	#CURIPAR,#UIPAR7	ILAST USER ASR DONE?
014500	001460			BEQ	NXTBANK	IYES - GO FIND NEXT BANK
014502	062737	020000	000034	ADD	#20000,#J34	IUPDATE SCOPE VECTOR ADDRESS IN BANK 0
014510	062737	020000	001066	ADD	#20000,#BANKSTR	IUPDATE BANK START TO REFERENCE CURRENT ASR
014516	013716	001066		MOV	#BANKSTR,(SP)	
014522	005077	164326		CLR	#CURIPAR	ISET PREVIOUS PAGE NR, BANK 0
014526	005077	164324		CLR	#CURIPAR	
014532	005077	164322		CLR	#CURIPDR	
014536	005077	164320		CLR	#CURIPDR	
014542	062767	000002	164304	NXTSEGI: ADD	#2,CURIPAR	IUPDATE POINTERS TO NEXT PAGE
014550	062767	000002	164300	ADD	#2,CURIPAR	
014556	062767	000002	164274	ADD	#2,CURIPDR	
014564	062767	000002	164270	ADD	#2,CURIPDR	
014572	012777	077406	164260	MOV	#77406,#CURIPDR	ISET NEXT PAGE RW, 4K

014600	012777	077406	164234	MOV	#77406,#CURIPDR	IMAP NEXT PAGE TO CURRENT BANK
014606	016777	164236	164240	MOV	CURBANK,#CURIPAR	
014614	016777	164230	164234	MOV	CURBANK,#CURIPAR	
014622	092737	030000	177776	BIS	#30000,#PPSR	ISET PREVIOUS MODE TO USER
014630	106506			R6		IPICK UP USER STACK POINTER
014632	062716	020000		ADD	#20000,#R6	IMAP IT TO NEXT ASR
014636	106606			HTPD	R6	IPIUT IT BACK
014640	000002			RTI		IGO BACK TO MAINLINE
014642	005327	000000		NXTBANK: DEC	#0	INSTALL SO DOUBLE BELL WILL BE
014646	001375			BNE	,#4	INOTED
014650	004767	002062		JSR	X7,BELL	
014654	012746	000400		NXBANK1: MOV	#UBUFF,-(SP)	IMAP USER STACK TO PAGE 0
014660	092737	030000	177776	BIS	#30000,#PPSR	
014666	106606			HTPD	R6	
014670	042737	160000	000034	BIC	#160000,#J34	IINITIALIZE SCOPE VECTOR ADDRESS
014676	013746	000004		MOV	#04,-(SP)	ISTORE ADDRESS ERROR RETURN
014702	013746	000006		MOV	#06,-(SP)	
014706	012737	015634	000004	MOV	#CORCK,#04	ISETUP TRAP RETURN
014714	005037	000006		CLR	#06	
014720	013737	001050	001052	MOV	#CURBANK,#OLDBNK	ISAVE ADDRESS OF PREVIOUS BANK
014726	062737	000200	001050	BNKTST: ADD	#200,#CURBANK	
014734	022737	007000	001050	CHP	#7000,#CURBANK	ICHECK FOR EXTERNAL BANK
014742	001066			BNE	EOB3	IIF NOT, TEST FOR ITS PRESENCE
014744	012737	000000	001050	MOV	#0,#CURBANK	ISTART OVER, TESTING BANK 0
014752	105237	001104		INCB	#DISPLY	
014756	013737	001104	177570	MOV	#DISPLY,#PSR	
014764	013701	000042		LOGIC: MOV	#042,R1	
014770	001412			BEQ	BNKT	
014772	000005			RESET		
014774	005046			CLR	-(SP)	ICLEAR T-BIT VIA RTI
014776	012746	015004		MOV	#LOGICAL,-(SP)	
015002	000002			RTI		
015004	004711			LOGICAL: JSR	X7,#R1	
015006	000240			NOP		
015010	000240			NOP		
015012	000240			NOP		
015014	000000			HALT		
015016	032737	000001	001026	BNKT1: BIT	#1,#SREG2	IPTY OUTPUT SELECTED?
015024	001410			BEQ	BNKT1	IYES, SKIP OUTPUTTING ASTERISK
015026	004767	001734		JSR	X7,ORLF	
015032	105777	163960		TSTB	#TCR	IWAIT FOR TELETYPE
015036	100375			SPL	,#4	
015040	012777	000252	163552	MOV	#252,#TDBR	IOUTPUT ASTERISK TO SIGNAL END OF PASS
015046	042766	000020	000006	BNKT1: BIC	#20,6(SP)	ICLEAR TRACE BIT OF STATUS ON STACK
015054	012737	000016	000014	MOV	#16,#014	
015062	005037	000016		CLR	#016	
015066	032737	010000	177570	BIT	#10000,#PSR	
015074	001011			BNE	EOB3	
015076	005167	163766		COH	TRP0	
015102	100006			SPL	EOB3	
015104	052766	000020	000006	BIS	#20,6(SP)	
015112	012737	016246	000014	MOV	#TRTRP,#014	
015120	016777	163724	163650	EOB3: MOV	CURBANK,#KIPAR2	IMAP KERNEL PAGE 2 TO BANK BEING LOOKED FOR
015126	016777	163716	163654	MOV	CURBANK,#KIPAR2	


```
019774 000240 GENR1 NOP  
019776 142767 000007 000016 BICB #7,RSNEG  
010004 051667 000012 BIS (SP),RSNEG  
010010 004767 000170 JSR X7,SHIPT  
010014 016667 177774 000002 MOV -4(SP),RSREG+2  
010022 162700 RSREG1 SUB (7)+,X0  
010024 000000 .WORD 0  
010026 017716 162640 NXTSET1 MOV #SR1,(SP)  
010032 000316 SWAB (SP)  
010034 003367 000142 DEC ALLREG  
010040 001327 BNC LSTSET  
010042 017716 162626 REEX1 MOV #SR2,(SP)  
010046 013746 000014 MOV #14,-(SP)  
010052 013746 000016 MOV #16,-(SP)  
010056 016646 000006 MOV 6(SP),-(SP)  
010062 016646 000006 MOV 6(SP),-(SP)  
010066 042716 160000 BIC #160000,(SP)  
010072 042766 140000 000002 BIC #140000,2(SP)  
010100 052766 000020 000002 BIS #20,2(SP)  
010106 012737 016132 000014 MOV #TSEG,#14  
010114 012737 000340 000016 MOV #340,#16  
010122 042777 176000 162540 BIC #176000,#SR0  
010130 000006 RTT  
010132 042766 007737 000012 TSEG1 BIC #7737,12(SP)  
010140 042766 170020 000002 BIC #170020,2(SP)  
010146 056606 000002 000012 BIS 2(SP),12(SP)  
010154 042766 017777 000010 BIC #17777,10(SP)  
010162 016666 000010 BIS (SP),10(SP)  
010166 022626 000016 CMP (SP)+,(SP)+  
010170 012637 000016 MOV (SP)+,#16  
010174 012637 000014 MOV (SP)+,#14  
010200 000002 RTI  
010202 000000 ALLREG1 NOP  
010204 000240 SHIPT1 MOV #SR1,-(SP)  
010206 017746 162460 000001 177762 CMP #1,ALLREG  
010212 022767 000001 .+4  
010220 001001 BNC (SP)  
010222 000316 SWAB (SP)  
010224 105066 000001 CLRB 1(SP)  
010230 006716 SXT (SP)  
010232 042716 000007 BIC #7,(SP)  
010236 006216 ASR (SP)  
010240 006216 ASR (SP)  
010242 006226 ASR (SP)+  
010244 000207 RTS X7  
  
:RRT EXECUTED WHEN TRACE IS ON  
016246 000006 TRTRP1 RTT  
  
:ENTERED WITH SYSTEM TRAP CALL (HLT)  
:PRINT OUT THE ERROR PC+2, STATUS REGISTER,  
:LOCATION IN BACKGROUND (TOP 12 BITS OF 10 BIT ADDRESS), AND  
:CONTENTS OF RETURN IN CURRENT BANK  
016250 005767 000200 PRINT1 TST PRTON :CHECK PRINT ON FLAG
```

```
016254 001401 BEQ .+4  
016256 000002 RTI  
016260 005267 000250 INC PRTON  
016264 012767 000340 161504 MOV #340,PSR :SET PRIORITY TO 7  
016272 036727 161272 020000 BIT SR,#20000 :TEST FOR INHIBIT PRINT OUT  
016300 001401 BEQ .+4 :BRANCH TO PRINT  
016302 000500 BR CK :INHIBIT, CHECK FOR HALT  
016304 012667 000220 MOV (6)+,SAVPC :PC OF FAILING ROUTINE  
016310 012667 000216 MOV (6)+,SAVPSR :PSR OF ERROR CONDITION  
016314 024646 CMP -(6),-(6) :RESTORE STACK  
016316 012767 000200 161492 MOV #200,PSR  
016324 004767 000436 JSR X7,CRLF :OUTPUT CARRIAGE RETURN AND LINE FEED  
016330 016767 000174 000356 MOV SAVPC,PTEMP1 :LOAD WITH FAILING PC+2  
016336 004767 000174 JSR X7,PROCT :PRINT FAILING PC+2  
016342 004767 000352 JSR X7,SPACE  
016346 016767 000140 000340 MOV SAVPSR,PTEMP1 :LOAD PROCESSOR STATUS  
016354 004767 000156 JSR X7,PROCT :PRINT PROCESSOR STATUS  
016360 004767 000334 JSR X7,SPACE  
016364 016767 162460 000322 MOV CURBNK,PTEMP1  
016372 004767 000140 JSR X7,PROCT  
016376 004767 000316 JSR X7,SPACE  
016402 012737 000340 177776 MOV #340,#PSR  
016410 017746 162372 MOV #KDPAR1,-(SP)  
016414 017746 162342 MOV #KDPDR1,-(SP)  
016420 016777 162424 162360 MOV CURBNK,#KDPAR1  
016426 012777 077406 162326 MOV #77406,#KDPDR1  
016434 013767 033350 000252 MOV #RETURN+20000,PTEMP1 :GET CONTENTS OF CURRENT RETURN  
016442 012677 162314 MOV (SP)+,#KDPDR1  
016446 012677 162334 MOV (SP)+,#KDPAR1  
016452 012737 000200 177776 MOV #200,#PSR  
016460 004767 000052 JSR X7,PROCT  
016464 004767 000230 JSR X7,SPACE  
016470 032737 000001 001026 BIT #1,#RSREG2 :TTY OUTPUT ALLOWED?  
016476 001002 BNE .+6 :NO, SKIP  
016500 004767 000262 JSR X7,CRLF :YES= OUTPUT CR,LF  
016504 005767 161060 CK1 TST SR :CHECK SR FOR WALT SWITCH  
016510 100001 BPL .+4 :BRANCH IF NOT SET  
016512 000000 HALT :HALT ON ERROR UP  
016514 005067 000014 CLR PRTON :ROUTINE DONE = CLEAR FLAG  
016520 000002 RTI :RETURN TO MAIN LINE  
016522 000000 SAVR21 0  
016524 000000 SAVR31 0  
016526 000000 SAVR41 0  
016530 000000 SAVPC1 0  
016532 000000 SAVPSR1 0  
016534 000000 PRTON1 0  
  
016536 012727 000006 010542 PROCT1 MOV #6,PTEMP3 :CLEAR R4 FOR COUNTING CHARACTERS OUTPUT  
016542 016542 PTEMP3#,-2  
016544 005067 000142 CLR PRFLG :INITIALIZE CARRY FLAG FOR ROTATES  
016550 012767 000200 000140 MOV #200,PTEMP2 :SETUP R3  
016556 005767 000132 TST PTEMP1 :CHECK BIT 15 OF NUMBER  
016562 100002 BPL .+6 :BRANCH IF ZERO  
016564 005267 000126 INC PTEMP2 :INCREMENT R3 IF ONE
```

```

016570 006167 000120          ROL      PTEMP1      ;ROTATE LEFT MOST OCTAL TO RIGHT END
016574 006167 000114          ROL      PTEMP1
016600 005567 000106          ADC      PRFLG
016604 105777 162000          P,WAIT: TSTB      @TCSR      ;WAIT FOR TTY READY
016610 100375                    BPL      P,WAIT
016612 016777 000100 162000          MOV      PTEMP2,@TOBR      ;OUTPUT NEXT CHARACTER
016620 005367 177716          P,CONT: DEC      PTEMP3
016624 001001                    BNE      P,CNT1
016626 000207                    RTS
016630 000241          P,CNT1: CLC
016632 005767 000054          TST
016636 001403                    BEQ      PRFLG
016640 005067 000046          CLR      PRFLG
016644 000261                    SEC
016646 006167 000042          ROL      PTEMP1
016652 006167 000036          ROL      PTEMP1
016656 006167 000032          ROL      PTEMP1
016662 005567 000022          ADC      PRFLG
016666 016767 000022 000022          MOV      PTEMP1,PTEMP2      ;STORE CARRY
016674 042767 177770 000014          BIC      #177770,PTEMP2      ;LOAD DATA INTO R3
016702 052767 000260 000006          BIS      #260,PTEMP2      ;CLEAR ALL BUT LOWEST OCTAL DIGIT
016710 000735                    BR       P,WAIT
016712 000000          PRFLG: 0
016714 000000          PTEMP1: 0
016716 000000          PTEMP2: 0      ;SET TO ASCII EQUIVALENT
;CONTAINS VALUE TO BE OUTPUT
;SCRATCH
;LOOP

016720 105777 161672          ;ISSUE SPACE
016724 100375          SPACE: TSTB      @TCSR      ;WAIT FOR TTY READY
016726 012777 000240 161664          BPL      ,=4
016734 000207          MOV      #240,@TOBR      ;OUTPUT A SPACE
016734 000207          RTS      X7

016736 032737 000001 001026          ;BELL ON PASS COMPLETE
016744 001001          BELL: BIT      #1,@SREG2      ;TTY OUTPUT SELECTED?
016746 000207          BNE      ,=4      ;NO, RING BELL
016750 105777 161642          RTS      X7      ;YES, SKIP
016754 100375          TSTB      @TCSR
016756 012777 000244 161634          BPL      ,=4
016764 000207          MOV      #244,@TOBR
016764 000207          RTS      X7

016766 105777 161624          ;SUBROUTINE TO OUTPUT CARRIAGE RETURN AND LINEFEED
016772 100375          CRLF: TSTB      @TCSR      ;WAIT FOR TTY READY
016774 012777 000215 161616          BPL      ,=4
017002 105777 161610          MOV      #215,@TOBR      ;OUTPUT CARRIAGE RETURN
017006 100375          TSTB      @TCSR      ;WAIT FOR TTY READY
017010 012777 000212 161602          BPL      ,=4
017016 000207          MOV      #212,@TOBR      ;OUTPUT LINEFEED
017016 000207          RTS      X7      ;RETURN

017020 013746 000024          ;ENTER HERE ON POWER FAIL
017024 010667 000030          PFAIL: MOV      #24,@(6)
017030 012737 017040 000024          MOV      %0,SAVR6      ;SAVE CONTENTS OF POWER FAIL VECTOR
017030 012737 017040 000024          MOV      #RESTMT,@#24      ;STORE STACK POSITION
017030 012737 017040 000024          MOV      #RESTMT,@#24      ;LOAD VECTOR WITH POWER UP RETURN ADDRESS

```

```

017036 000000          ;HALT ON POWER DOWN
017040 016706 000014          RESTR: MOV      SAVR6,%6      ;RESTORE STACK WHEN POWERING UP
017044 012637 000024          MOV      (0)+,@#24      ;RESTORE ORIGINAL CONTENTS OF POWER FAIL VECTOR
017050 022626          CMP      (3P)+,(3P)+
017052 104006          HLT
017054 000167 162170          JHP      RSTRY      ;RESTORE THE STACK
017060 000000          SAVR6: 0      ;POWER FAIL OCCURRED

017062 000207          USER: RTS      X7
017760 017760          ,=17760
017760 000000          KSTACK: 0
017760 000001          ,END
;OVERLAY USER ROUTINE HERE

```

A	013362	ABSRV	015642	ADSUB	012942	ALLREG	016292
ARIEND	012602	ARITST	012920	B	013352	BEGIN	004670
BEGINX	004456	BEGIN1	007012	BEGIN2	007034	BELL	014786
BKREQ	005440	BNKLP	015616	BNKSTR	001066	BNKT	015040
BNKTSY	014726	BNKT1	015046	C	013374	CC	177776
CK	016504	CKWAIT	013070	CLINCT	000012	CHP1	012086
CHP2	012060	COMPAT	012032	CORCK	015034	CORHOV	015224
CRLF	016766	CSIPDR	001064	CURBNK	001050	CURDPA	001096
CURDPD	001062	CURIPA	001054	CURIPB	001040	CURPAT	000112
D	013414	DATAK	004316	DATA2	000044	DET1	003230
DET2	003332	DET3	003630	DET4	003262	DISPLY	001104
DO	000001	DONE	013210	EIGHT	003300	EMTAB	013566
EMTSRV	013526	ENDZ	009306	ENDZ1	000350	ENDZ2	005424
EOB	104012	EOBSRV	013670	EOB1	013700	EOB1A	013764
EOB1B	014066	EOB1C	014106	EOB2	014114	EOB3	015120
EOB4	015230	EOB4A	015376	EOB5	015062	EOB6	015022
EPC	013564	ERROK	004176	ERMSET	004172	FBLK	004310
FUNCT1	004264	GENR	015774	HLT	104000	ICOUNT	013344
IE	000100	IPDRN	001022	IPDRTA	001010	IRF	000132
KDPAR0	001004	KOPAR1	001006	KOPAR2	001010	KOPAR3	001012
KDPAR7	001014	KOPDR0	000760	KOPDR1	000762	KOPDR2	000764
KOPDR3	000766	KOPDR7	000770	KIPAR0	000772	KIPAR1	000774
KIPAR2	000776	KIPAR3	001000	KIPAR7	001002	KIPDR0	000746
KIPDR1	000750	KIPDR2	000752	KIPDR3	000754	KIPDR7	000756
KSTACK	017760	KWLST	000634	KWLVC	000632	LKCSR	000630
LK1	003750	LK2	003766	LK3	003716	LK4	003734
LLIMIT	004266	LOGIC	014764	LOGICA	015004	LPCSR	000636
LPDR	000640	LPINTR	003770	LPST	000644	LPVC	000642
LP1	004106	LP2	004114	LP3	004032	LP4	004022
LP5	004100	LP6	004080	LSTSET	015720	MAIN	003172
MODE1	003140	MOD12	003564	MOD16	003550	MOD20	003534
MOD24	003520	MOD4	003614	MOD8	003600	MOVE	003344
NODEV	003210	NOP	000240	NRALL	013410	NRL00P	013430
NUMA	012600	NXBK1	014654	NXTBLK	006142	NKTBK	014642
NXTB1	004206	NXTB2	006216	NXTB3	006230	NXTB4	006312
NXTSEG	014542	NXTSET	014026	OLDBNK	001052	PAGE	002116
PARDN	001504	PARFD	001234	PARSET	001436	PARSRV	001106
PARTO	001230	PAR1	001152	PAR2	001162	PAR3	001170
PATCH1	000000	PATCH2	000000	PATCH3	000000	PC	1300007
PFALL	017020	PIR	001076	PIRQ1	001100	PIRQST	001102
PIRSRV	013460	PIR3	013512	PIRFLG	016712	PRINT	016200
PROCT	014536	PRTON	014534	PSR	177776	PSSAV	001072
PTEMP1	014714	PTEMP2	014716	PTEMP3	016542	P.CNT1	016630
P.CONT	014620	P.WAIT	014604	R	004000	RG	000002
RBN	005560	RBN1	005600	RBN10	006102	RBN11	006134
RBN2	005612	RBN3	005620	RBN4	005642	RBN5	005720
RBN6	005750	RBN7	006014	RBN8	006050	RBN9	006096
RD	000004	REEX	014042	REF	012222	REFF	012222
REGEN	005450	REGENL	005474	REG1	001026	RESTRT	017040
RETURN	013350	RFCAR	000056	RFCSR	000040	RFCSRH	000042
RFDAC	000646	RFDAR	000682	RFDOR	000050	RFBT	000066
RFBSTAR	004234	RFYC	000064	RFMC	000064	RF1	004240
ROTALL	012114	RTBE	012262	ROTBO	012370	ROTEN1	012470
RSREG	016022	RSTRT	001250	RTIK	104010	RTIKEX	013602

R0	1000000	R1	1000001	R2	1000002	R3	1000003
R4	1000004	R5	1000005	R6	1000006	SAVPC	016530
SAVPSR	016532	SAVR2	016522	SAVR3	016024	SAVR4	016526
SAVR6	017060	SBUFF	000000	SCOPE	104400	SCOPEB	013384
SCOPEC	013292	SCOPEF	013346	SCOPEG	013332	SOPAR7	000740
SDPDR0	000732	SDPDR7	000744	SEGCT	001030	SEG1	002244
SEGREC	001024	SETEX	002210	SETKER	002370	SETREL	003066
SETSEG	002640	SETSP1	002612	SETSUP	002534	SETUSE	002324
SHIFT	016204	SIPAR7	000736	SIPDR0	000730	SIPDR6	000734
SIPDR7	000742	SOLPAT	004110	SP	1000000	SPACE	014720
SPSAV	013666	SR	177570	SREG1	001024	SREG2	001026
SR0	000670	SR1	000672	SR2	000674	SR3	000676
START	001264	STARTX	001334	START1	001352	STATUS	177776
STOPTC	005446	STOR1	004312	STOR2	004314	STRF	002694
STRF1	002724	STR12	003400	STNT16	003406	STR120	003414
STR124	003422	STR20	003430	STR4	003364	STR8	003372
STC	002774	ST0	001376	ST1	001070	ST2	001700
ST3	001702	ST4	001704	ST5	001770	ST6	002074
SUBR1	013214	SUBR2	013216	SUBR3	013222	SUBR4	013230
SUBR5	013236	SUBR6	013244	SUPCYC	014270	SUPDON	014414
SUPD1	014420	SUPD2	014456	SWABA	013144	SXTEEN	003310
TCBA	001042	TCLBK	004302	TCLUF	004326	TCLUPF	004272
TCLUFV	004274	TCKK	006364	TCKK1	006412	TCKK2	006448
TCCM	001032	TCDT	001036	TCEMA	004320	TCEQP	004322
TCEXP	004304	TCFRST	004276	TCIV	001044	TCLAST	004308
TCOM	012614	TCDH2	012642	TCOM3	012730	TCREL	004324
TCRS	000616	TCST	001034	TCSTA	001046	TCSTOR	004306
TCWC	001040	TDBR	000620	TEMP	013404	TEST	012224
TIME	003782	TJSR1	011424	TJSR2	011426	TJSR3	011436
TJSR4	011452	TRCSR	000606	TRDR	000010	TRPB	000070
TRTRP	016246	TSCOMB	012772	TSEC	016132	TSROT	012070
TSROT2	012234	TSRT2A	012226	TSTARI	012986	TTCSR	000616
TTDBR	000620	TTPST	000624	TTPVC	000622	TTST	000614
TTVC	000612	TTSAV	000626	THELVE	003304	TWENTY	003314
TWOIG	003324	THOFOR	003320	TYOUT	003644	TYOUTR	003640
TYOUT1	003650	UBRK	001074	UBUFF	000400	UDPAR0	000722
UDPAR1	000724	UDPAR7	000726	UDPDR0	000706	UDPAR1	000710
UDPDR7	000712	UIPAR0	000714	UIPAR1	000716	UIPAR7	000720
UIPDR0	000700	UIPDR1	000702	UIPDR7	000704	UN1	000400
USEALL	002440	UBEN	017062	WAIT4	013072	WD	000014
WORDCT	004270	XFER12	003476	XFER16	003460	XFER20	003486
XFER24	003446	XFER28	003436	XFER8	003380		017742

ERRORS DETECTED: 0

#DCKTGD,DCKTGD=DCKTGD,P11
 RUN-TIME: 10 20 0 SECONDS
 CORE USED: 5K