

ZQKCCP

IDENTIFICATION

PRODUCT CODE MAINDEC:11-BEQKC-C-D
PRODUCT NAME 11 FAMILY INSTRUCTION EXERCISER
DATE CREATED NOVEMBER 1, 1973
MAINTAINER DIAGNOSTIC GROUP
AUTHOR J. ADAMS

COPYRIGHT © 1971, 1972, 1973 DIGITAL EQUIPMENT CORPORATION

THIS SOFTWARE IS FURNISHED TO PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DEC'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

- 1.0 ABSTRACT
THIS DIAGNOSTIC PROGRAM IS DESIGNED TO BE A COMPREHENSIVE CHECK OF THE PDP11/05 AND PDP11720 PROCESSORS, THE PROGRAM EXECUTES EACH INSTRUCTION IN ALL ADDRESS MODES AND INCLUDES TESTS FOR TRAPS AND THE TELETYPE INTERRUPT SEQUENCE, THE PROGRAM DOES NOT TEST INSTRUCTIONS NOT COMMON TO THE 11/20 OR 11/05, THE PROGRAM RELOCATES THE TEST CODE THROUGHOUT MEMORY 0-20K.
- 2.0 REQUIREMENTS
2.1 EQUIPMENT
PDP11 FAMILY CENTRAL PROCESSOR
OPTIONAL = KW11=L (LINE CLOCK)
OPTIONAL = ALL MEMORY PARITY OPTIONS
- 2.2 STORAGE
THE PROGRAM USES ALL OF THE FIRST 4K OF MEMORY (EXCLUDING THAT AREA OF MEMORY RESERVED FOR THE LOADERS).
- 2.3 PRELIMINARY PROGRAMS
NONE, HOWEVER, THE EMT AND TRAP INSTRUCTION SHOULD BE VERIFIED BEFORE RUNNING.
- 3.0 LOADING AND STARTING PROCEDURE
LOAD PROGRAM USING ABS LOADER
LOAD ADDRESS = 200
PRESS START
SET OPERATING SWITCHES
PASS COUNT IS PRINTED AFTER EACH PASS (SEE SEC 6.5)
"DEGRC DONE" IS PRINTED WHEN DONE (SEE SEC 7.1)
- 4.0 SWITCH SETTINGS
SW15 HALT ON ERROR,,, THIS SWITCH WHEN SET WILL HALT THE PROCESSOR WHEN AN ERROR IS DETECTED, THE PC+2 AND THE CURRENT STATUS AT THE TIME OF THE ERROR IS STORED ON THE STACK (R6); IF THIS SWITCH IS SET BEFORE AN ERROR IS DETECTED THE PROGRAM HALTS AS DESCRIBED ABOVE, THE PROGRAM MAY BE HALTED AFTER THE ERROR TYPEOUT OCCURS BY SETTING SW15 AFTER THE TYPEOUT BEGINS.

SW14 LOOP SUBTEST,,, THIS SWITCH WHEN SET LOOPS THE CURRENT SUBTEST RUNNING REGARDLESS OF ERROR.

SW13 INHIBIT ERROR PRINTOUT = THIS SWITCH WHEN SET INHIBITS THE ERROR PRINTOUT.

SW12 INHIBIT RELOCATION,,, THIS SWITCH WHEN SET CAUSES THE PROGRAM TO BE EXECUTED ONLY IN THE FIRST 4K OF MEMORY, THIS SWITCH CANNOT BE SET WHEN THE PROGRAM IS RUNNING.

SW11 INHIBIT SUBTEST ITERATION,,, THIS SWITCH WHEN SET INHIBITS SUBTEST REITERATION, NORMALLY EACH SUBTEST IS EXECUTED 8 TIMES BEFORE THE NEXT SUBTEST IS RUN.

SETTING SW11 CAUSES EACH TEST TO BE EXECUTED ONCE BEFORE STARTING THE NEXT SUBTEST;

SW10 RING BELL ON ERROR;.; THIS SWITCH WHEN SET WILL RING THE BELL WHEN AN ERROR IS DETECTED;

SW9 INHIBIT RELOCATION - THIS SWITCH WHEN SET INHIBITS RELOCATION OF THE PROGRAM ABOVE 20K;

SW7 INHIBIT END OF PASS TIMEOUT;

SW8 LOAD PDP-11/45 MICRO BREAK REGISTER;.; THIS SWITCH

SW7=0 WHEN SET LOADS THE MICRO BREAK REGISTER WITH THE VALUE SET INTO SW7=0 AT THE BEGINNING OF EACH SUBTEST, WITH THE VALUE SET INTO SW7=0 AT THE BEGINNING OF EACH SUBTEST.

4.5 14/45 DISPLAY REGISTER
THE PASS COUNT IS DISPLAYED IN THE DISPLAY REGISTER;
NOTE) THE PASS COUNT IS ALSO STORED IN LOCATION 1800;

5.0 ERRORS
IF AN ERROR IS DETECTED THE PROGRAM WILL TRAP TO THE ERROR HANDLING ROUTINE (ERROR); IF ENABLED THIS ROUTINE WILL BYTE THE PC AND THE PROCESSOR STATUS AT THE TIME OF THE ERROR; ALSO (IF REQUIRED) THE ORIGINAL PC (WHERE THE PC WAS RELOCATED FROM);

5.0.1 ERROR PRINTOUT FORMAT
ICNT#AAAA PC#BBBBBB PSW#DDDDDD
OR
ICNT#AAAA PC#BBBBBB PSW#DDDDDD PC# RELOCATED FROM CCCCCC

WHERE# AAAA#PASS COUNT
BBBBB#PC AT THE TIME OF THE ERROR
CCCCC#PC OF THE ORIGINAL CODE RELOCATED
DDDD#PSW AT THE TIME OF THE ERROR;

5.1 PARITY ERROR DETECTION
IF A PARITY ERROR IS DETECTED THE PROGRAM WILL TYPE A MESSAGE "PARITY ERROR" AND SCAN MEMORY FOR THE PARITY ERROR; WHEN THE FAILING ADDRESS IS LOCATED THE PROGRAM WILL HALT WITH THE VALUE OF THE ADDRESS+2 IN R0.

5.2 ERROR LOOPING
THE SUBTEST DETECTING THE ERROR MAY BE LOOPED INDEFINITELY BY SETTING SW14; SETTING SW13 WILL INHIBIT THE TYPEOUT AND ALLOW SCOPING THE FAULTY SIGNAL(S).

5.3

UNPREDICTED ERRORS

THE PROGRAM MAY ON OCCASSION DETECT A MEMORY ERROR THE RESULTS OF WHICH WERE NOT PREDICTABLE IN WHICH CASE THE PROGRAM MAY BEHAVE UNPREDICTABLY. WHEN THIS HAPPENS THE USER MUST RETRACE THE PROGRAM STEPS TO RESOLVE WHERE THE ERROR OCCURRED. THE FOLLOWING ITEMS SHOULD BE CONSIDERED AND MAY BE OF USE WHEN RETRACING A FAILURE OF THIS NATURE.

1. HALT THE PROGRAM (IF NECESSARY)
2. EXAMINE RELR1
ADDRESS RELR1 (1006) CONTAINS THE UNRELOCATED VALUE OF THE PC OF THE LAST TEST THAT WAS SUCCESSFULLY EXECUTED.
3. EXAMINE FACTOR
ADDRESS FACTOR (1004) CONTAINS THE RELOCATION FACTOR.
4. EXAMINE ALL LOCATIONS STARTING WITH THE ADDRESS SPECIFIED IN R1/R11 (IF PSW BIT11 00/1) COMPARING THEIR CONTENTS WITH THE CONTENTS OF THE CORRESPONDING UNRELOCATED CODE (SPECIFIED IN 1006) AS SHOWN IN THE LISTING. EXAMINE AND COMPARE UNTIL EITHER A DIFFERENCE IN INSTRUCTION (I.E., THE ERROR) OR THE NEXT 'SCOPE' IS SEEN.

IF THE PROGRAM TRAPS AND HALTS AT A TRAP/INTERRUPT VECTOR+2 (NOTE: THE PDP-11/45 WILL DISPLAY THE ADDRESS OF THE HALT+2 I.E., A FALSE TRAP TO 4 WILL DISPLAY 10).

1A. EXAMINE THE STACK (R6)

THE TOP WORD ON THE STACK CONTAINS THE PC AT THE TIME OF THE TRAP. IF THE PC IS GREATER THAN 20000, THEN

2A. EXAMINE LOCATION 1002 (FACTOR)

THIS LOCATION CONTAINS THE PROGRAM RELOCATION FACTOR WHICH, WHEN SUBTRACTED FROM THE PC GIVES THE PC OF THE ORIGINAL CODE.

6.0

SUBROUTINE ABSTRACTS

6.1

SCOPEA

THE SCOPEA ROUTINE IS ENTERED BY THE SCOPE (EMT) INSTRUCTION AND IS EXECUTED AT THE START OF EACH SUBTEST. THE ROUTINE MONITORS SW14, SW11 AND SW 8 AND TAKES APPROPRIATE ACTION. ALSO, THIS ROUTINE STORES IN R1/R11 THE FIRST ADDRESS OF THE SUBTEST BEING ENTERED.

6.2

ERROR

THE ERROR ROUTINE IS ENTERED BY THE HLT (TRAP) INSTRUCTION.

AND IS EXECUTED WHEN A PREDICTABLE ERROR IS DETECTED; THIS ROUTINE MONITORS SW12, SW13, AND SW18;

6:3

RELOC:
THE RELOC ROUTINE IS ENTERED BY A MOV RELOC,PC INSTRUCTION; THIS ROUTINE RELOCATES THE PROGRAM CODE THROUGHOUT MEMORY, AND 'JUMPS' TO THE RELOCATED CODE AFTER IT HAS BEEN MOVED SUCCESSFULLY; IF THE CODE CANNOT BE RELOCATED (BECAUSE OF INSUFFICIENT MEMORY) THE ROUTINE 'JUMPS' TO THE NEXT SECTION OF UNRELOCATED PROGRAM CODE; THE CODE MOVED IS LESS THAN 1K (2000) BYTES; AT THE START AND END OF EACH SECTION OF CODE TO BE MOVED ARE A SECTION OF CODE WHICH ESTABLISHES THE FIRST ADDRESS OF THE CODE TO BE MOVED, AND SETS A SCOPE POINTER (R1/R15) AND, ALSO A SECTION WHICH ESTABLISHES THE LAST ADDRESS AND 'JUMPS' TO THE RELOCATION (RELOC) ROUTINE; EACH SECTION OF CODE IS IDENTIFIED AS SHOWN BELOW:

1000000000000000FIRST ADDRESS TO BE RELOCATED0000000000

CODE TO BE MOVED AND EXECUTED

1000000000000000LAST ADDRESS OF CODE TO BE RELOCATED 00000000

THE RELOC ROUTINE DOES NOT RELOCATE PROGRAM CODE INTO THE LAST 1000(0) BYTES OF MEMORY, THUS PRESERVING THE LOADERS;

6:4

END
THIS ROUTINE IS ENTERED AT THE COMPLETION OF EACH PASS IT SETS UP (LOADS NEW PROCESSOR STATUS) FOR THE NEXT PASS; AND PRINTS THE PASS COUNT;

ICNT#XXXX

7:0

MISCELLANEOUS

7:1

EXECUTION TIME
THE EXECUTION TIME IS HIGHLY VARIABLE (DEPENDENT ON PROCESSOR, TYPE OF MEMORY, AND AMOUNT OF MEMORY); HOWEVER, WHEN THE PROGRAM IS RUNNING SUCCESSFULLY THERE IS A NOTICEABLE 'FLICKER' DISPLAYED IN THE CONSOLE LIGHT PATTERN THE 'FLICKER' WILL DIM WHEN 'T' BIT TRAP PASSES (EVERY ODD PASS) ARE RUNNING; THE PROGRAM SHOULD BE RUN FOR A MINIMUM OF:

2 PASSES ICNT#2 11/85 OR 11/20

SOME TYPICAL TIMES FOLLOW:

8:0

PROGRAM DESCRIPTION
THE PROGRAM IS DIVIDED INTO FOUR SECTIONS OF POSITION INDEPENDENT RELOCATABLE TEST CODE; EACH SECTION IS APPROXIMATELY 1K WORDS LONG; (EXCEPT SECTION A);

SECTION 0 THIS SECTION TEST THE UNARY INSTRUCTION SET EXECUTING EACH UNARY INSTRUCTION IN EACH ADDRESS MODE (EXCLUDING UNARY INSTRUCTIONS USING ADDRESS MODE 7).

SECTION 1 THIS SECTION TESTS THE UNARY INSTRUCTIONS USING ADDRESS MODE 7 AND BINARYS IN ALL ADDRESS MODES (EXCLUDING BINARY BYTE OPS USING ADDRESS MODE 7).

SECTION 2 THIS SECTION TEST BINARY BYTE OPS USING ADDRESS MODE 7, JMP, JSR AND PROGRAM TRAP (IOT, TRAP AND EMT) INSTRUCTIONS.

SECTION A FOLLOWING SECTION 2 IS A ROUTINE TO ASCERTAIN WHICH OF THE PROGRAM IS RUNNING ON, THE RESULTS ARE USED BY THE FOLLOWING CODE TO CHECK THE ADDITIONAL INSTRUCTIONS/FEATURES OF THE 11/40 AND 11/45.

SECTION 3 THIS SECTION CHECKS THAT EACH BIT IN THE PROCESSOR STATUS WORD (PSW) CAN BE SET CLEARED, RESERVED INSTRUCTION, AND ODD ADDRESS TRAPS.

FOLLOWING SECTION 3 ARE TWO ROUTINES TO CHECK THE TELETYPE PRINTER LOGIC AND A ROUTINE TO START THE KW11=L LINE CLOCK. IF THE KW11=L IS AVAILABLE THE PRIORITY ARBITRATION LOGIC IS TESTED.

AFTER EACH INDIVIDUAL SECTION HAS BEEN EXECUTED THE "RELOC" ROUTINE WILL RELOCATE THE SECTION THROUGHOUT ALL MEMORY UP TO 28K. WHEN THE SECTION HAS BEEN RELOCATED AND EXECUTED IN ALL MEMORY THE "RELOC" ROUTINE WILL RETURN THE PROGRAM TO THE NEXT UNRELOCATED SECTION.

RELOCATION AND EXECUTION OF ALL SECTIONS THROUGHOUT ALL MEMORY CONSTITUTES A SINGLE PASS.

UPON COMPLETION OF A PASS OF THE PROGRAM THE PROGRAM RESTARTS USING A NEW PROCESSOR STATUS DEPENDING ON THE TYPE OF PROCESSOR AND THE PASS COUNT.

8.1

STACK POINTER

THE STACK POINTER IS SET AT 000.

NOTE: IF THE PROGRAM IS RUNNING IN EITHER USER OR SUPERVISOR MODE (NOT APPLICABLE IF 11/20 OR 11/05) THE USER/SUPERVISOR STACK POINTER IS SET TO 500 AND THE KERNEL STACK POINTER IS SET TO 600. THE KERNEL STACK POINTER IS USED ONLY FOR THE SCOPE, HLT, TTY, AND KW11=L (IF AVAILABLE TRAP/INTERUPT ROUTINES).

8.2

POWER FAILURE

A POWER FAIL SERVICE ROUTINE IS INCORPORATED IN THE TEST. WHEN USING THIS PROGRAM THE POWER SHOULD BE TURNED OFF WHEN

RUNNING TO CHECK THE POWER FAIL LOGIC. WHEN THE POWER FAILS
THE PROGRAM WILL TYPE
POWER FAILED
AND RESTART THE PROGRAM AT THE BEGINNING. (START)

9:8

USER DEFINED RELOCATION LIMITS
THE PROGRAM WILL REQUEST A LOWER AND UPPER LIMIT FOR
RELOCATION. THE LIMITS MUST BE BETWEEN 20000 AND 157776.
THE PROGRAM WILL EXECUTE IN THE LOWER 4K (0-17776) AND THE
LIMITS SPECIFIED.
THE STARTING ADDRESS IS 204.
TO RETAIN PREVIOUSLY SPECIFIED LIMITS START AT 210.

INLIST REG.MD,MC
LIST MC
ABS
TITLE FRONT END
CONTAINS DEFINITIONS, REGISTER ASSIGNMENTS AND MACRO CALLS

GENERAL REGISTER ASSIGNMENTS

000000 R0=K0
000001 R1=K1
000002 R2=K2
000003 R3=K3
000004 R4=K4
000005 R5=K5
000006 R6=K6
000007 R7=K7
000008 R8=K8
000009 R9=K9
000010 R10=K10
000011 R11=K11
000012 R12=K12
000013 R13=K13
000014 R14=K14
000015 R15=K15

FLOATING POINT REGISTERS

000000 AC0=K0
000001 AC1=K1
000002 AC2=K2
000003 AC3=K3
000004 AC4=K4
000005 AC5=K5

STACK POINTER REGISTERS

000006 KSP=K6
000006 BSP=K6
000006 USPP=K6

KERNEL STACK POINTER
SUPERVISOR STACK POINTER
USER STACK POINTER

STATUS REGISTER (PSW) BIT ASSIGNMENTS

000001 C=1
000002 V=2
000004 E=4
000010 N=10
000020 T=20
000340 PRTY7=340
000300 PRTY6=300
000200 PRTY4=200
000000 KH=000000
000000 SH=040000
000000 UH=140000
000000 PKH=000000
000000 PSH=010000
000000 PUM=030000
000000 REG=004000

IC BIT
IV BIT
IR BIT
IN BIT
IY BIT
PRIORITY LEVEL 7
PRIORITY LEVEL 6
PRIORITY LEVEL 4
KERNEL MODE
SUPERVISORY MODE
USER MODE
PREVIOUS KERNEL MODE
PREVIOUS SUPERVISORY MODE
PREVIOUS USER MODE
ISELECT R10-R15

VECTOR ADDRESSES

000004 ERRVEC=4

ADDRESS OF ERROR VECTOR

000010 RESVEC=10
000014 TBIVVEC=14
000014 TRTVEC=14
000014 TRTYVEC=14
000020 IOTVEC=20
000024 PFVEC=24
000030 EMTVEC=30
000034 TRAPVEC=34
000044 TPVEC=44
000100 LVVEC=100
000240 PIRVEC=240
000244 PPEVEC=244
000250 MHVEC=250

ADDRESS OF RESERVED INST; TRAP VECTOR
ADDRESS OF IY BIT TRAP VECTOR
ADDRESS OF TRACEI TRAP VECTOR
ADDRESS OF BREAKPOINTI TRAP VECTOR
ADDRESS OF IOT TRAP VECTOR
ADDRESS OF POWER FAIL TRAP VECTOR
ADDRESS OF EMT VECTOR
ADDRESS OF TRAP VECTOR
ADDRESS OF TTY PRINTER INTERRUPT VECTOR
ADDRESS KMI=L LINE CLOCK INT; VECTOR
ADDRESS OF PIRQ VECTOR
ADDRESS OF FLOATING POINT INT; VECTOR
ADDRESS OF MEM MGMT ERROR TRAP VECTOR

REGISTER ADDRESSES

177776 PSW= 177776
177774 SLR= 177774
177772 PIRQ= 177772
177770 UBREAK= 177770
177546 LKS= 177546
177560 TKS= 177560
177562 TKB= 177562
177564 TPS= 177564
177566 TPB= 177566
177572 BRB= 177572
177570 BSR= 177570
177570 DISPLAY=177570
177514 LPB= 177514
177516

ADDRESS OF STATUS REGISTER
ADDRESS OF STACK LIMIT REGISTER
ADDRESS OF PROGRAM INTERRUPT REQUEST
ADDRESS OF MICRO BREAK REGISTER
ADDRESS OF KMI=L STATUS REG;
ADDRESS OF KEYBOARD CSR
ADDRESS OF KEYBOARD BUFFER
ADDRESS OF TELEPRINTER CSR
ADDRESS OF TELEPRINTER BUFFER
ADDRESS OF MEM MGMT REGISTER BRB
ADDRESS OF CONSOL SWITCH REGISTER
ADDRESS OF CONSOL DISPLAY REGISTER
ADDRESS OF LINE PRINTER STATUS REG
ADDRESS OF LINE PRINTER DATA BUFFER

MEMORY MANAGEMENT REGISTER ADDRESSES

172300 KIPDR0= 172300
172302 KIPDR1= 172302
172304 KIPDR2= 172304
172310 KIPDR7= 172310
172340 KIPAR0= 172340
172342 KIPAR1= 172342
172350 KIPAR7= 172350

177600 UIPDR0=177600
177610 UIPDR7=177610
177640 UIPAR0=177640
177650 UIPAR7=177650

172200 BIPDR0=172200
172210 BIPDR7=172210
172240 SIPAR0=172240
172250 SIPAR7=172250

INITIAL STACK POINTER SETTING

000000 BTKPTR= 000
000000 KPTR=000

PROGRAM STACK PTR
KERNEL STACK PTR (USED BY KERNEL WHEN
PROGRAM IS RUNNING IN OTHER THAN KERNEL


```

                                MISCELLANEOUS BIT ASSIGNMENTS
                                C180PT=40000
                                B1719=100000
                                B1714=40000
                                B1713=30000
                                B170=400
                                B176=100
                                P1R4=10000
                                (LEVEL 4 PROGRAM INT; ROOT; (FOR P1R0)

                                INSTRUCTION EQUATES
                                MLY=TRAP
                                SCOPE=ENT
                                (MLY IS A TRAP INST TO THE ERROR ROUTINE
                                (SCOPE IS AN ENT TRAP

                                .LIST ME
                                .=120
                                ROUTINE TO SET PARITY ACTION ENABLE ON MA/MF PARITY MEMORIES
                                PARCSR= 172100
                                PARVEC= 000114
                                (ADDRESS OF FIRST POSSIBLE PARITY REG
                                (ADDRESS OF PARITY INTERRUPT VECTOR

                                R00120 312737 002004 000114 MAMF1 MOV #,PARSRV,00PARVEC (LOAD VECTOR
                                R00126 312737 000340 000116 MOV #340,00PARVEC+0 (AND PRIORITY LEVEL
                                R00134 312737 000000 0000F4 MOV #ERRVEC+2,00ERRVEC (SET TIME OUT TRAP VECTOR
                                R00142 312737 0000F2 0000F6 MOV #RT1,00ERRVEC+0 (DO RT1 ON TIME OUT TRAP
                                R00150 312700 172100 MOV #PARCSR,R0 (GET FIRST POSSIBLE ADDRESS
                                R00154 312702 000001 MOV #1,R2 (SET REGISTER COUNTER

                                R00160 312720 000001 (S) MOV #1,(R0)+ (SET ACTION ENABLE (IF AVAIL)
                                (ABOVE INSTRUCTION WILL SET ACTION ENABLE IF MA/MF PARITY OR SET
                                (DOO PARITY AND HALT ON PARITY ERROR IF MOS PARITY
                                R00164 000302 ASL R2 (CHECK IF 16; REGISTERS HAVE
                                R00166 100374 BCC 15 (BEEN ENABLED
                                R00170 000207 RTS PC (RETURN

                                .=200
                                R00200 312707 002250 MOV #START,PC (GO TO START OF TEST
                                R00204 312707 002404 MOV #START1,PC (GO GET LOWER/UPPER RELOCATION BOUNDARY
                                R00210 312737 002490 MOV #START3,PC (START WITH LAST TYPED BOUNDARY LIMITS

```

```

                                ROUTINE TO SAVE REGISTERS ON THE STACK
                                (CALLED BY SAVE MACRO OR JSR PC,SSAVR
                                SSAVE1 MOV (SP)+,15 (SAVE RETURN PC
                                MOV #0,(SP)
                                MOV #4,(SP)
                                MOV #3,(SP)
                                MOV #2,(SP)
                                MOV #1,(SP)
                                MOV #0,(SP)
                                MOV (PC)+,PC (RETURN
                                (CONTAINS RETURN ADDRESS

                                ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
                                (CALLED BY RESTORE MACRO OR JSR PC,SRESTR
                                SRESTR1 MOV (SP)+,15 (SAVE RETURN PC
                                MOV (SP)+,R0
                                MOV (SP)+,R1
                                MOV (SP)+,R2
                                MOV (SP)+,R3
                                MOV (SP)+,R4
                                MOV (SP)+,R5
                                MOV (PC)+,PC (RETURN
                                (CONTAINS RETURN ADDRESS

                                .=610
                                R00610 312737 000020 000024 (POWER FAIL SUBROUTINE
                                PDWN1 MOV #PUP,00PFVEC
                                R00616 000000 HALT

                                R00620 312737 000010 000024 (POWER UP SUBROUTINE
                                PUP1 MOV #PDWN,00PFVEC (RESTORE POWER FAIL TRAP TO POWER
                                R00626 312700 000000 MOV #KPTR,SP (DOWN ROUTINE ABOVE
                                R00632 000027 CLR (PC)+ (SET STACK PTR
                                R00634 000000 (S) MORD 0 (KILL TIME
                                R00636 000207 177772 (S) INC 15
                                R00642 001375 (S) BNE 23
                                R00644 004767 000302 (S) JSR PC,PRINT (PRINT MESSAGE BEGINNING AT FOLLOWING ADDR
                                R00650 000636 (S) PFAIL
                                R00652 000137 002250 JMP #START (RESTART TEST

                                R00656 000013 047320 #42927 PFAIL1 .ASCII <15><12>'POWER FAILED'<13><12>
                                R00664 020122 040700 #46111
                                R00672 042109 000015 #00
                                R00677 019 000012 001101 PARERR1 .ASCII <13><12>'PARITY ERROR'<13><12>
                                R00704 232111 020131 #1105
                                R00712 047322 000522 #00012

                                .=740
                                (NOTE) THIS CODE USED ONLY BY THE XOR TESTER.
                                (DO USE CODE PLACE 776 (R0,=0) IN SCOPE
                                R00740 312737 000002 000006 FORXOR1 MOV #RT1,00ERRVEC+0 (SET TIME OUT TRAP TO RETURN
                                R00746 000201 SEC (SET C
                                R00750 000737 177000 TST #0177000 (IF A TIME OUT OCCURS THEN WHEN NEXT

```

000754	103401			BCB	13		INSTRUCTION IS EXECUTED (B) WILL BE SET (AND IF NO TIME OUT (C) WILL BE CLEARED (BRANCH IF (C) SET (TIMED OUT) (ADDRESS OF NEXT SUBTEST TO R1 (RESTORE TIME OUT TRAP (GET RETURN ADDRESS BACK TO SUBTESTS (RETURN EITHER TO LAST OR NEXT SUBTEST
000756	011421			MOV	(SP),R1		
000760	000037	000000	15)	CLR	0	000000	
000764	310116			MOV	R1,(BP)		
000766	000240			NOF			
000770	000002			RTI			
000776	000000			TICKB1	,WORD	0	(CONTAINS CLOCK TICK COUNT
001000	000000			ICNT1	,WORD	0	(CONTAINS PASS COUNT
001002	000000			IFILLS1	,WORD	0	(CONTAINS FILLS COUNT IN ODD BYTE (AND FILLER CHARACTER IN EVEN BYTE
001004	000000			FACTOR1	,WORD	0	(CONTAINS RELOCATION FACTOR
001006	000000			RELOC1	,WORD	0	(GET PC OF ORIGINAL CODE (CONTAINS RELOCATED R1 (THE R1 OF THE (ORIGINAL CODE MOVED)
001010	000000			FRSTAD1	,WORD	0	(CONTAINS FIRST ADDR OF CODE TO BE MOVED
001012	000000			FRSTMEM1	,WORD	0	(CONTAINS LOWER RELOCATION BOUNDARY ADDRESS
001014	000751			OR	FORXOR		(BRANCH TO XOR TESTER CODE
001016	000240			SCOPEA1	NOF		(SCOPE (ENT) SERVICE ROUTINE (THIS ROUTINE ALLOWS THE SUBTEST TO BE CONTINUOUSLY LOOPEO, ITERATED (OR NOT ITERATED) BEFORE BEGINNING NEXT SUBTEST
001020	032766	004000	000012	BIT	#4000,2(SP)		(WAS REGISTER SET BIT SET ON TRAP
001026	001403			BCO	23		(BRANCH IF NOT
001030	032737	004000	177776	BIT	#4000,00PSW		(RETAIN REGISTER SET
001036	032737	040000	177570	BIT	#4000,00PSW		(CHECK BIT 14 (CONTINUOUS LOOP)
001044	001416			SCOPEB1	NOF		(LOAD RETURN ADDRESS
001046	010116			MOV	R1,(SP)		
001050	010137	001000		MOV	R1,00REL1		(REL1 CONTAINS UNRELOCATED R1
001054	163737	001004	001000	SUB	00FACTOR,00REL1		(LOAD R011/45 MICRO BREAK REG
001062	032737	000400	177570	BIT	#400,00PSW		
001070	001403			BCO	13		(LOAD MICRO BREAK REG WITH 50007
001072	113737	177570	177770	NOF	00PSW,00UBREAK		(RETURN TO SUBTEST
001100	000002			RTI			
001102	032737	004000	177570	SCOPEC1	BIT	#4000,00PSW	(SUBTEST ITERATION DESIRED)
001110	001000			BNE	SCOPEE		(BRANCH IF NO ITERATION DESIRED)
001112	005327			DEC	(PC)+		(DECREMENT SUBTEST ITERATION COUNT
001114	000040			SCOPED1	40		(CONTAINS SUBTEST ITERATION COUNT
001116	001393			BNE	SCOPEB		
001120	012707	000040	177766	SCOPEF1	MOV	#40,SCOPED	(RESET ITERATION COUNT
001126	011601			SCOPEE1	MOV	(SP),R1	(GET ADDRESS OF NEXT TEST
001130	000746			RR	SCOPEB		
001132	032737	010000	177570	ROUTINE TO RELOCATE PROGRAM CODE			
001140	001031			RELOC1	BIT	010000,00PSW	(CHECK IF RELOCATION DESIRED (BIT12)
001142	013700	001010		BNE	33		(BRANCH IF NO RELOCATION DESIRED
001146	010005			MOV	00FRSTAD,R0		(GET FIRST ADDRESS OF CODE TO BE MOVED
001150	010204			MOV	R0,R5		(SAVE
001152	100504			MOV	R2,R4		(GET LAST ADDRESS OF CODE TO BE MOVED
				SUB	R5,R4		(R4 CONTAINS # OF WORDS TO RELOCATE

001154	010203			MOV	R2,R3		(SAVE LAST ADDRESS OF CODE TO BE MOVED
001156	005737	001004		TSY	00FACTOR		(FIRST RELOCATION IS TO 20000
001162	001024			BNE	105		
001164	010237	001230		MOV	R2,00RETPC		(SAVE RETURN PC TO NEXT SECTION OF CODE
001170	013702	001012		MOV	00FRSTMEM,R2		(SET FIRST ADDRESS
001174	000204			ADD	R2,R4		(R4 CONTAINS LAST MEMORY ADDRESS (TO BE USED
001176	020437	002304		CHP	R4,00LSTMEM		(CHECK IF SUFFICIENT MEMORY REMAINS
001202	010111			BHI	45		
001204	012022			MOV	(R0)+,(R2)+		(RELOCATE PROGRAM CODE
001206	020003			CHP	R0,R3		(CHECK IF DONE
001210	001375			BNE	15		
001212	024042			CHP	-(R0),-(R2)		(CHECK THAT CODE WAS RELOCATED
001214	001401			BEG	,04		(PROPERLY
001216	104400			WLT			(ERROR) CODE NOT RELOCATED PROPERLY
001220	020000			CHP	R0,R5		(CHECK IF FINISHED CHECKING
001222	001373			BNE	25		
001224	010207			MOV	R2,PC		(GO EXECUTE RELOCATED CODE
001226	011707			MOV	(PC),PC		(RETURN TO NEXT SECTION OF CODE
001230	000000			RETPC1	0		(CONTAINS PC OF NEXT SECTION OF CODE
001232	010046			ROUTINE TO PRINT ASCII MESSAGE			MESSAGE MUST TERMINATE WITH A 0 BYTE
001234	017000	000002		(PRINT) MOV	R0,(SP)		(SAVE R0 ON THE STACK
001240	062766	000002	000012	MOV	02(SP),R0		(GET MESSAGE ADDRESS
				ADD	02,2(SP)		(ADJUST RETURN PC
001246	112046			MOV	(R0)+,(SP)		(PUSH CHAR ON THE STACK
001250	001003			BNE	25		(BRANCH IF NOT TERMINATOR
001252	005720			TSY	(SP)+		(POP TERMINATOR OFF THE STACK
001254	012600			MOV	(SP)+,R0		(RESTORE R0
001256	000207			RTS	PC		(RETURN
001260	004747	000026		JER	PC,55		(TYPE CHARACTER
001264	122726	000012		CHPB	#12,(SP)+		(CHECK IF CHAR WAS A LINE FEED
001270	001306			BNE	15		(BRANCH IF NOT LINE FEED
001272	016746	177504		MOV	IFILLS,=(SP)		(GET # OF FILLERS REQUIRED AFTER (LINE FEED AND FILLER CHARACTER
001276	105306	000001		DECB	1(SP)		(DECREMENT FILLERS COUNT
001302	002770			BLT	35		(BRANCH IF NO MORE FILLERS NEEDED
001304	004747	000002		JSR	PC,55		(TYPE FILLER CHARACTER
001310	000772			RR	45		
001312	105737	177504		TSY	00TPB		(WAIT FOR OUTPUT DEVICE
001316	100375			BPL	,04		(TO BECOME READY
001320	116637	000002	177506	MOV	2(SP),00TPB		(TYPE CHARACTER
001326	000207			RTS	PC		
				NULL=0			
001330				ROUTINE TO PLACE ASCII VALUE OF AN ADDRESS IN TO ADDRESS MESSAGE			
001330	004747	176000		JSR	PC,55AVR		(GO SAVE REGISTERS ON THE STACK
001334	012704	001062		MOV	00G1TB,R4		(ADDRESS WHERE ASCII VALUES ARE STORED

R01342	012701			MOV	R2,R1	SAVE
R01344	006302			ASL	R2	FIRST DIGIT TO R3
R01346	006103			ROL	R3	
R01350	012700	000000		MOV	R0,R0	DIGIT COUNT
R01354	006404			BR	00	PRINT FIRST DIGIT
R01356	006302			ASL	R2	
R01358	006103			ROL	R3	
R01362	009301			DEC	R1	
R01364	001374			SNE	Z0	
R01366	012701	000003		MOV	R3,R1	DIGIT SHIFT COUNT
R01372	116324	001092		MOVW	DIGTAB(3),(4)+	LOAD DIGIT INTO MESSAGE
R01376	009303			CLR	R3	CLEAR INDEX
R01400	009300			DEC	R0	DEC DIGIT COUNT
R01402	001365			SNE	Z0	
R01404	004767	176630		JSR	PC,SRESTR	RESTORE REGISTERS FROM STACK
R01410	000207			RTS	PC	RETURN

R01412	009737	177970		TST	0000R	ERROR SERVICE CALLED BY TRAP INLT INSTRUCTION
R01416	100002			BPL	,04	HALT ON ERROR
R01420	000000			HALT		ERROR PC IS THE TOP WORD
R01422	000002			RTI		ON THE STACK
R01424	032737	020000	177970	BIT	020000,0000R	PRINT OUT DESIRED
R01432	001073			SNE	Z0	BRANCH IF NO PRINTOUT
R01434	011627			MOV	(SP),(PC)+	SAVE PC
R01436	000000			,WORD	0	CONTAINS SAVED PC
R01440	011627	000002		MOV	(SP),(PC)+	GET STATUS ON TRAP
R01444	000000			,WORD	0	CONTAINS STATUS (R0) AT TIME OF TRAP
R01446	004767	176342		JSR	PC,SSAVR	GO SAVE REGISTERS ON THE STACK
R01492	013702	001000		MOV	00ICNT,R2	GET PASS COUNT
R01496	004767	177044		JSR	PC,SFORM0	GO TO FORMAT ROUTINE
R01462	016767	000170	000212	MOV	DIGITS+2,PASSES	LOAD ASCII VALUES
R01470	016767	000172	000206	MOV	DIGITS+4,PASSES	
R01476	004767	177030		JSR	PC,PRINT	PRINT MESSAGE BEGINNING AT FOLLOWING ADDR
R01502	001672			PASCNT		
R01504	016767	177720		MOV	110,R2	GET PC OF ERROR CALL
R01510	009742			TST	=(R2)	DECREMENT PC TO HLT
R01512	004767	177612		JSR	PC,SFORM0	GO TO FORMAT ROUTINE
R01516	004767	177910		JSR	PC,PRINT	PRINT MESSAGE BEGINNING AT FOLLOWING ADDR
R01522	001707			ERRPC		
R01524	004767	177902		JSR	PC,PRINT	PRINT MESSAGE BEGINNING AT FOLLOWING ADDR
R01530	001602			DIGITS		
R01532	004767	177474		JSR	PC,PRINT	PRINT MESSAGE BEGINNING AT FOLLOWING ADDR
R01536	001714			STATUS		
R01540	016767	177700		MOV	120,R2	GET STATUS AT TIME OF ERROR
R01544	004767	177500		JSR	PC,SFORM0	GO TO FORMAT ROUTINE
R01550	004767	177450		JSR	PC,PRINT	PRINT MESSAGE BEGINNING AT FOLLOWING ADDR
R01554	001502			DIGITS		
R01556	016767	177654		MOV	110,R2	GET PC OF ERROR
R01562	009742			TST	=(R2)	
R01564	009737	001004		TST	00FACTOR	
R01570	001412			BCD	100	
R01572	103702	001004		SUB	00FACTOR,R2	FORM PC OF ORIGINAL CODE
R01576	004767	177320		JSR	PC,SFORM0	GO TO FORMAT ROUTINE

R01602	004767	177424		JSR	PC,PRINT	PRINT MESSAGE BEGINNING AT FOLLOWING ADDR
R01606	001721			ERRPCB		
R01610	004767	177410		JSR	PC,PRINT	PRINT MESSAGE BEGINNING AT FOLLOWING ADDR
R01614	001602			DIGITS		
R01616			1001			
R01618	004767	176410		JSR	PC,SRESTR	RESTORE REGISTERS FROM STACK
R01622	032737	002000	177970	BIT	020000,0000R	RING BELL ON ERROR
R01630	001403			BCD	Z0	
R01632	004767	177374		JSR	PC,PRINT	PRINT MESSAGE BEGINNING AT FOLLOWING ADDR
R01636	001747			BELL		
R01640	009737	177970		TST	0000R	HALT AFTER PRINT OUT
R01644	100001			BPL	,04	
R01646	000000			HALT		
R01650	000002			RTI		

R01652	030400			DIGTAB	"01"	
R01654	031402				"23"	
R01656	032404				"49"	
R01660	033406				"67"	
R01662	030000	030000	030000	DIGITS	,ASCIZ '000000'	
R01670	000040			PASCNT		
R01672	009015			,ASCIZ	<15><12>	
R01674	041511	032116	020079	,ASCIZ	'ICNTS'	
R01676	030000	030000	000	PASSES	,ASCIZ '00001'	
R01678	040	041520	000075	ERRPCB	,ASCIZ 'PC'	
R01680	051520	030527	000	STATUS	,ASCIZ 'R0'	
R01682	120	020103	042322	ERRPCB	,ASCIZ 'PC RELOCATED FROM'	
R01684	047514	040503	042324			
R01686	020104	051106	040517			
R01690	000040			SCALPI	,ASCIZ <15><12>	
R01692	009015	000		BELLI	,ASCIZ <7>	
R01694	007	000		,EVEN		

R01752	010044			JSR	R0,RECD	CONVERTED DATA IS PLACED HERE
R01754	009015			,WORD	0	SAVE R0 ON THE STACK
R01756	109737	177900		MOV	R0,=(SP)	CLEAR OLD DATA
R01760	100373			CLR	(R0)	WAIT FOR USER TO TYPE CHARACTER
R01764	113700	177902		TST	00TKB	
R01768	042700	000200		BPL	Z0	GET CHARACTER
R01772	122700	000177		MOVW	00TKB,R0	STRIP MSB
R01774	001010			BIC	0200,R0	CHECK IF RUBOUT
R01776	112700	000177		CHPB	0177,R0	BRANCH IF NOT RUBOUT
R01778	001010			SNE	Z0	TYPE C
R01780	112737	000134	177900	MOVW	01,00TPB	CLEAR CARRY
R01782	000241			CLC		SHIFT LAST TYPED CHARACTER
R01784	000015			ROR	(R0)	OUT OF DATA WORD
R01786	000210			ASR	(R0)	
R01788	000210			ASR	(R0)	
R01790	000790			BR	Z0	GO WAIT FOR NEXT CHARACTER
R01792	110037	177900		MOVW	R0,00TPB	ECHO CHARACTER TYPED

.TITLE DBKCC BASIC II FAMILY INSTRUCTION EXER;

002256	012737	144000	177776	START1	MOV	#144000,00PSW	(ENTER USER MODE WITH REGISTER SET BIT SET
002204	000000				CLR	R0	(CLEAR R0=R16 (IF 11/48 ONLY)
002206	000001				CLR	R1	
002270	000002				CLR	R2	
002272	000003				CLR	R3	
002274	000004				CLR	R4	
002276	000005				CLR	R5	
002300	000006				USP		(CLEAR USER STACK PTR
002302	012737	040000	177776		MOV	#040000,00PSW	(SUPERVISOR MODE (11/48 ONLY)
002310	000006				CLR	R3	(CLEAR SUPERVISOR STACK PTR
002312	000007	177776			CLR	00PSW	(KERNEL MODE
002316	000008				CLR	R0	(CLEAR R0=R9
002320	000001				CLR	R1	
002322	000002				CLR	R2	
002324	000003				CLR	R3	
002326	000004				CLR	R4	
002330	000005				CLR	R5	
002332	012706	000000			MOV	#KPTR,0P	(SET KERNEL STACK PTR
ROUTINE TO DETERMINE LAST MEMORY ADDRESS							
002336	012737	002356	000004		MOV	#13,0ERRVEC	
002344	000007	000000			CLR	00ERRVEC+2	
002350	000008				CLR	R0	
002352	000009				TEST	(R0)+	(WILL TIME OUT WHEN END OF MEMORY
002354	000010				BR	,02	
002356	102700	000002		15)	SUB	#2,R0	(SET VALUE INTO L0YMEM
002362	010027				MOV	R0,(PC)+	(CONTAINS VALUE OF LAST MEMORY ADDRESS
002364	000000			L0YMEM)	WORD	0	(SET PROTECTION FOR LOADERS
002366	102737	004000	002364		SUB	#4000,00L0YMEM	(SET LOWER BOUNDARY AT 20000
002374	012737	020000	001012		MOV	020000,00FR0YMEM	(GO TO START 3
002402	000422				BR	START3	
002404				START1)	JSR	PG,PRINT	(PRINT MESSAGE BEGINNING AT FOLLOWING ADDR
002406	004707	176022			M001		
002410	017040				JSR	R5,RECD	(GET LOWER LIMIT
002412	004567	177334			0		(CONTAINS TYPED LOWER LIMIT
002416	000000			15)	WORD	0	(SET IN LOWER LIMIT
002420	016737	177772	001012		MOV	15,00FR0YMEM	(PRINT MESSAGE BEGINNING AT FOLLOWING ADDR
002426	004707	176000			JSR	PG,PRINT	
002432	017055				M002		
002434	004567	177312			JSR	R5,RECD	(GET UPPER LIMIT
002440	000000				0		(CONTAINS UPPER LIMIT
002442	016737	177772	002364	25)	WORD	0	
002490	000007	001000		START3)	CLR	00ICNT	(CLEAR PASS COUNT
002494	004707	173440			JSR	PG,MANF	(GO ENABLE PARITY IF AVAILABLE
002496	012737	000000	000004	START2)	MOV	#ERRVEC+2,0ERRVEC	(SET ERROR TRAP TO M1LT AT 6
002406	012700	000000			MOV	00KPTR,0P	(SET STACK PTR
002472	012737	001000	177370		MOV	00ICNT,00DISM1T	(DISPLAY PASS COUNT
002300	012737	001010	000030		MOV	#SCOPEA,00EMTYVEC	(SET (SCOPE) TRAP VECTOR
002306	012737	001412	000034		MOV	#ERR0R,00TRAPVEC	(SET TRAP (M1LT) VECTOR
002314	012737	000200	000036		MOV	0200,00TRAPVEC+2	(PRIORITY LEVEL 6 ON TRAP

002522	010770				RELOC1	MOV	PG,R0	(GET PC
002524	000740					TEST	-(R0)	(R0 CONTAINS THE ADDRESS OF RELO
002526	010037	001010			MOV	R0,00FR0YMEM	(SAVE	
002532	010700				MOV	PG,R0	(GET CURRENT PC	
002534	102700	002534			SUB	0, R0	(SUBTRACT RELOCATION FACTOR	
002540	010037	001004			MOV	R0,00FACTOR	(SAVE RELOCATION FACTOR	
002544	010701				MOV	PG,R1	(SET NEW SCOPE PTR	
ICHECK BRANCH INSTRUCTIONS								
002546	000237				CCC		(CC=00000	
002550	103407				CC0	CC0	(SAME IS 010	
002552	102400				CC0	CC0		
002554	001405				CC0	CC0		
002556	100404				CC0	CC0		
002560	002403				CC0	CC0		
002562	003402				CC0	CC0		
002564	101401				CC0	CC0		
002566	101001				CC0	CC0		
002570	104400				CC0	CC0		
CONTINUE								
002572	000270				CC1	CC1	(NONE OF THE ABOVE BRANCHES FAILED	
002574	100003				CC1	CC1		
002576	002002				CC1	CC1		
002600	003001				CC1	CC1		
002602	002401				CC1	CC1		
002604	104400				CC1	CC1		
CONTINUE								
002606	000202				CC2	CC2	(NONE OF THE ABOVE BRANCHES FAILED	
002610	102003				CC2	CC2		
002612	002402				CC2	CC2		
002614	003401				CC2	CC2		
002616	002001				CC2	CC2		
002620	104400				CC2	CC2		
CONTINUE								
002622	000201				CC3	CC3	(NONE OF THE ABOVE BRANCHES FAILED	
002624	103002				CC3	CC3		
002626	101001				CC3	CC3		
002630	003001				CC3	CC3		
002632	104400				CC3	CC3		
CONTINUE								
002634	000204				CC4	CC4	(NONE OF THE ABOVE BRANCHES FAILED	
002636	001003				CC4	CC4		
002640	003002				CC4	CC4		
002642	101001				CC4	CC4		
002644	003401				CC4	CC4		
002646	104400				CC4	CC4		
002650	100000				CC4	CC4		

TEST UNARY CONDITION CODES
ICLR R0

002092	000277		SCC		
002094	000244		CLC		
002096	003000		CLR	RB	IR0=0,CC'S=0100
002098	103404		SCS	CLRB	
002100	102403		BVS	CLRB	
002102	001002		BNE	CLRB	
002104	100401		BMI	CLRB	
002106	003401		BLE	,+4	
002108	104400	CLRB	HLT		ERROR: INCORRECT CC'S AFTER CLR
002110	000277		BCC		
002112	000244		CLB		
002114	003700		TST	RB	IR0=0,CC'S=0100
002116	103404		SCS	TSRB	
002118	102403		BVS	TSRB	
002120	001002		BNE	TSRB	
002122	100401		BMI	TSRB	
002124	101401		BDB	,+4	
002126	104400	TSRB	HLT		ERROR: INCORRECT CC'S AFTER TST
002128	000237		CCC		
002130	000200		ISEV		
002132	003000		COM	RB	IR0=1,CC'S=0101
002134	103004		BCC	COMB	
002136	100403		BVS	COMB	
002138	001402		BEO	COMB	
002140	100001		BPL	COMB	
002142	002401		BGT	,+4	
002144	104400	COMB	HLT		ERROR: INCORRECT CC'S AFTER COM
002146	000201		BEC		
002148	003500		ADC	RB	IR0=00000,CC'S=0101
002150	103003		BCC	ADCB	
002152	102402		BVS	ADCB	
002154	001001		BNE	ADCB	
002156	002001		BGE	,+4	
002158	104400	ADCB	HLT		ERROR: INCORRECT CC'S AFTER ADC
002160	000201		BEC		
002162	000000		ROR	RB	IR0=00000,CC'S=0100
002164	103404		BCS	RORB	
002166	102003		BVC	RORB	
002168	001402		BEO	RORB	
002170	100001		BPL	RORB	
002172	003001		BGT	,+4	
002174	104400	RORB	HLT		ERROR: INCORRECT CC'S AFTER ROR
002176	000277		SCC		
002178	000242		CLV		
002180	003300		DEC	RB	IR0=07777,CC'S=0011
002182	103004		BCC	DECB	
002184	102003		BVC	DECB	
002186	001402		BEO	DECB	
002188	100401		BMI	DECB	
002190	003401		BLE	,+4	

003016	104400	DECB	HLT		ERROR: INCORRECT CC'S AFTER DEC
003018	000237		CCC		
003020	003200		INC	RB	IR0=00000,CC'S=0100
003022	103404		BCS	INCB	
003024	102003		BVC	INCB	
003026	001402		BEO	INCB	
003028	100001		BPL	INCB	
003030	003001		BGT	,+4	
003032	104400	INCB	HLT		ERROR: INCORRECT CC'S AFTER INC
003034	000277		SCC		
003036	000242		CLV		
003038	003400		NEC	RB	IR0=00000,CC'S=0101
003040	103003		BCC	NECB	
003042	102002		BVC	NECB	
003044	001401		BEO	NLCB	
003046	002001		BGE	,+4	
003048	104400	NECB	HLT		ERROR: INCORRECT CC'S AFTER NEC
003050	000201		BEC		
003052	000300		ASL	RB	IR0=00000,CC'S=0111
003054	103004		BCC	ASLB	
003056	102003		BVC	ASLB	
003058	001002		BNE	ASLB	
003060	100401		BMI	ASLB	
003062	101401		BDB	,+4	
003064	104400	ASLB	HLT		ERROR: INCORRECT CC'S AFTER ASL
003066	000100		ROL	RB	IR0=00001,CC'S=0000
003068	103402		BCS	ROLB	
003070	003401		BLE	ROLB	
003072	002001		BGE	,+4	
003074	104400	ROLB	HLT		ERROR: INCORRECT CC'S AFTER ROL
003076	000200		ASR	RB	IR0=00000,CC'S=0111
003078	103003		BCC	ASRB	
003080	102002		BVC	ASRB	
003082	001001		BNE	ASRB	
003084	002401		BGT	,+4	
003086	104400	ASRB	HLT		ERROR: INCORRECT CC'S AFTER ASR
003088	000277		SCC		
003090	003000		BBC	RB	IR0=1,CC'S=0100
003092	103002		BCC	BBCB	
003094	102401		BVS	BBCB	
003096	003401		BLE	,+4	
003098	104400	BBCB	HLT		ERROR: INCORRECT CC'S AFTER BBC
003100	003400		NCB	RB	IR0=00001,CC'S=0000
003102	000300		SWAB	RB	IR0=00000,CC'S=0100
003104	103403		BVS	SWABB	
003106	102402		BVC	SWABB	
003108	001001		BNE	SWABB	

803134	000001		DOE	,04	
803136	104400		HLT		ERROR! INCORRECT OPCODE AFTER SWAB
803100	104000		SCOPE		
ICHECK REGISTER SELECTION					
803102	000000		CLR	R0	
803104	000277		BCC		
803106	004100		ROL	R0	IR0=1
803170	010002		MOV	R0,R2	
803172	006302		ABL	R2	IR0=2
803174	010203		MOV	R2,R3	
803176	006303		ABL	R3	IR3=4
803200	010304		MOV	R3,R4	
803202	006304		ABL	R4	IR4=10
803204	010405		MOV	R4,R5	
803206	006305		ABL	R5	IR5=20
803210	010500		MOV	R5,(SP)	IR5=20
803212	000410		BIS	R4,(SP)	IR5=20
803214	000310		BIS	R3,(SP)	IR5=20
803216	000210		BIS	R2,(SP)	IR5=20
803220	000010		BIS	R0,(SP)	IR5=20
803222	022720	000037	CMP	037,(SP)+	IR5=20
803226	001401		BEO		IR5=20
803230	104400		HLT	,04	IR5=20

ICHECK THAT ALL BITS CAN BE SET & CLEARED IN ALL REGISTERS

803232	000237		CCC		
803234	112700	000377	MOV	0377,R0	IR5=20
803240	006100		ROL	R0	IR5=20
803242	103770		DCB	15	IR5=20
803244	009200		INC	R0	IR5=20
803246	001401		BEO	,04	IR5=20
803250	104400		HLT		IR5=20
803252	012700	000020	MOV	016,,R0	IR5=20
803256	009002		CLR	R2	IR5=20
803260	000201		SEC		IR5=20
803262	004002		ROR	R2	IR5=20
803264	009300		DEC	R0	IR5=20
803266	001374		BNE	25	IR5=20
803270	009102		COM	R2	IR5=20
803272	001401		BEO	,04	IR5=20
803274	104400		HLT		IR5=20
803276	012703	100000	MOV	010000,R3	IR5=20
803282	006203		ASR	R3	IR5=20
803284	103370		BCC	35	IR5=20
803286	009203		INC	R3	IR5=20
803310	001401		BEO	,04	IR5=20
803312	104400		HLT		IR5=20
803314	112704	177401	MOV	0177401,R4	IR5=20
803320	000404		ADD	R4,R4	IR5=20

803322	103370		BCC	45	IR5=20
803324	009704		TST	R4	IR5=20
803326	001401		BEO	,04	IR5=20
803330	104400		HLT		IR5=20
803332	012705	000001	MOV	01,R5	IR5=20
803336	006305		ABL	R5	IR5=20
803340	102370		BYC	55	IR5=20
803342	006305		ABL	R5	IR5=20
803344	103002		BCC	45	IR5=20
803346	009705		TST	R5	IR5=20
803350	001401		BEO	,04	IR5=20
803352	104400		HLT		IR5=20
ICHECK REGISTER VOLATILITY					
803354	009002		CLR	R2	IR5=20
803356	009102		COM	R2	IR5=20
803360	010203		MOV	R2,R3	IR5=20
803362	000237		CCC		IR5=20
803364	006002		COM	R2	IR5=20
803366	006272		ASR	R2	IR5=20
803370	010324		MOV	R3,R4	IR5=20
803372	009302		DEC	R2	IR5=20
803374	001375		BNE	75	IR5=20
803376	009203		INC	R3	IR5=20
803400	001002		BNE	85	IR5=20
803402	009204		INC	R4	IR5=20
803404	001401		BEO	,04	IR5=20
803406	104400		HLT		IR5=20
ICHECK TRANSFER OF REGISTER DATA BETWEEN THE 00 AND 0D REGISTERS (11/45)					
803410	032737	000020	032737	BIT	020,00PSW
803416	001052		BNE	75	IR5=20
803420	010140		MOV	R1,(SP)	IR5=20
803422	010627		MOV	SP,(PC)+	IR5=20
803424	000000		,WORD	0	IR5=20
803426	010727		MOV	PC,(PC)+	IR5=20
803430	000000		,WORD	0	IR5=20
803432	009207	177772	INC	25	IR5=20
803436	016700	177700	MOV	25,R0	IR5=20
803442	010001		MOV	R0,R1	IR5=20
803444	010102		MOV	R1,R2	IR5=20
803446	010203		MOV	R2,R3	IR5=20
803450	010304		MOV	R3,R4	IR5=20
803452	010405		MOV	R4,R0	IR5=20
803454	102737	000340	0340,00PSW	BIS	0340,00PSW
803462	010900		MOV	R0,SP	IR5=20
803464	010627		MOV	SP,(PC)+	IR5=20
803466	000000		,WORD	0	IR5=20
803470	016700	177730	MOV	15,SP	IR5=20
803474	142737	000340	0340,00PSW	BIC	0340,00PSW
803502	026700	177700	CMP	48,R0	IR5=20
803506	001004		BNE	90	IR5=20

803510	886307	177714	ASL	25		ISRIPT TEST DATA UNTIL 0 00000
803514	881359		BNE	35		
803516	880411		BR	05		
803520	810046		MOV	R0,(SP)		IGET OS REG 0
803522	810146		MOV	R1,(SP)		ITC:??
803524	810246		MOV	R2,(SP)		
803526	810346		MOV	R3,(SP)		
803530	810446		MOV	R4,(SP)		
803532	810546		MOV	R5,(SP)		
803534	104400		HLT			!ERROR! DATA IN OS STK PTR NOT 0 OS REG 0 (OS REG 005 REG 5 ARE ON THE STACK !RESTORE STACK PTR !RESTORE SCOPE PTR
803536	816706	177602	MOV	10,SP		
803542	812601		MOV	(SP),R1		
803544	104000		781	SCOPE		
!TEST UNARY WORD INSTRUCTIONS USING ADDRESS MODE 1						
803546	880401		BR	,04		!RESERVE ADDRESS FOR TESTS
803550	880000		,WORD	0		
803552	810702		MOV	PC,R2		!R2 POINTS TO RESERVED WORD
803554	102702	880000	SUB	#4,R2		!PRESET (R2)
803560	895012		CLR	(R2)		
803562	880201		SEC			
803564	880012		ROR	(R2)		!(R2)=[0000,CC=10]0
803566	101402		BLOS	ROR1		
803570	100001		BPL	ROR1		
803572	882001		BGE	,04		
803574	104400		80811	HLT		!ERROR! INCORRECT CC'S AS SHOWN ABOVE
803576	880257		CCC			
803600	880201		SEC			
803602	885312		DEC	(R2)		!(R2)=[077777,CC=00]1
803604	103001		BCC	DEC1		
803606	883401		BLE	,04		
803610	104400		DEC11	HLT		!ERROR! INCORRECT CC'S AS SHOWN ABOVE
803612	880257		CCC			
803614	880201		SEC			
803616	885512		ADC	(R2)		!(R2)=[0000,CC=10]0
803620	103403		BCS	ADC1		
803622	102002		BVC	ADC1		
803624	100001		BPL	ADC1		
803626	881001		RNE	,04		
803630	104400		80211	HLT		!ERROR! INCORRECT CC'S AS SHOWN ABOVE
803632	880112		ROL	(R2)		!(R2)=[000000,CC=00]1
803634	103003		BCC	ROL1		
803636	102002		BVC	ROL1		
803640	881001		BNE	ROL1		
803642	100001		BPL	,04		
803644	104400		80111	HLT		!ERROR! INCORRECT CC'S AS SHOWN ABOVE
803646	880112		ROL	(R2)		!(R2)=[000001,CC=00]0
803650	101402		BLOS	ROL1A		!BRANCH IF C OR Z IS SET

803652	102401		BVS	ROL1A		
803654	100001		BPL	,04		
803656	104400		8011A	HLT		
803660	886212		ASR	(R2)		!(R2)=[000000,CC=01]1
803662	103003		BCC	ASR1		
803664	102002		BVC	ASR1		
803666	881001		RNE	ASR1		
803670	100001		BPL	,04		
803672	104400		ASR11	HLT		!ERROR! INCORRECT CC'S AS SHOWN ABOVE
803674	880012		ROR	(R2)		!(R2)=[00000,CC=10]0
803676	103403		BCS	ROR1A		
803700	102002		BVC	ROR1A		
803702	881401		BEQ	ROR1A		
803704	100401		BHI	,04		
803706	104400		8011A	HLT		
803710	880201		SEC			
803712	885212		INC	(R2)		!(R2)=[00001,CC=10]0
803714	103003		BCC	INC1		
803716	102402		BVS	INC1		
803720	881401		BEQ	INC1		
803722	100401		BHI	,04		
803724	104400		INC11	HLT		!ERROR! INCORRECT CC'S AS SHOWN ABOVE
803726	885612		SBC	(R2)		!(R2)=[00000,CC=10]0
803730	103403		BCS	SBC1		
803732	102402		BVS	SBC1		
803734	881401		BEQ	SBC1		
803736	100401		BHI	,04		
803740	104400		80211	HLT		!ERROR! INCORRECT CC'S AS SHOWN ABOVE
803742	880201		SEC			
803744	885612		SBC	(R2)		!(R2)=[077777,CC=00]0
803746	103403		BCS	SBC1A		
803750	102002		BVC	SBC1A		
803752	881401		BEQ	SBC1A		
803754	100001		BPL	,04		
803756	104400		8021A	HLT		!ERROR! INCORRECT CC'S AS SHOWN ABOVE
803760	880201		SEC			
803762	885512		ADC	(R2)		!(R2)=[00000,CC=10]0
803764	100401		BHI	,04		
803766	104400		HLT			
803770	880201		BCC			
803772	886312		ASL	(R2)		!(R2)=[000000,CC=00]1
803774	103003		BCC	ASL1		
803776	102002		BVC	ASL1		
803780	881001		BNE	ASL1		
803782	100001		BPL	,04		
803784	104400		ASL11	HLT		!ERROR! INCORRECT CC'S AS SHOWN ABOVE

804006	009112	COM	(R2)	;(R2)0E7777,CC=1001
804010	100002	BCC	COM1	
804012	102401	BVB	COM1	
804014	100421	BMI	,+4	
804016	104400	COM1	HLT	;ERROR: INCORRECT CC'S AS SHOWN ABOVE
804020	000200	CLN		
804022	000712	TST	(R2)	;(R2)0E7777,CC=1000
804024	103403	DCB	TST1	
804026	100402	BVB	TST1	
804030	100001	BPL	TST1	
804032	001001	BNE	,+4	
804034	104400	TST1	HLT	;ERROR: INCORRECT CC'S AS SHOWN ABOVE
804036	000202	SEV		
804040	003412	NEG	(R2)	;(R2)000001,CC=0000
804042	100002	BCC	NEG1	
804044	102401	BVB	NEG1	
804046	001001	BNE	,+4	
804050	104400	NEG1	HLT	;ERROR: INCORRECT CC'S AS SHOWN ABOVE
804052	009312	DEC	(R2)	;(R2)000000,CC=0101
804054	103001	BCC	DEC1A	
804056	001401	BEQ	,+4	
804060	104400	DEC1A	HLT	;ERROR: INCORRECT CC'S AS SHOWN ABOVE
804062	104000	SCOPE		
;CHECK UNARY BYTE INSTRUCTIONS USING ADDRESS MODE 1				
804064	000401	BR	,+4	;RESERVE A WORD
804066	000000	,WORD	0	;ADDRESS RESERVED FOR TESTS
804070	010703	MOV	PC,R3	
804072	102703	SUB	R4,R3	;R3 POINTS TO EVEN BYTE OF WORD
804076	010304	MOV	R3,R4	;R4 POINTS TO ODD BYTE OF WORD
804100	000204	INC	R4	
804102	000013	CLR	(R3)	;PRESH DATA
804104	000201	1S1	SEC	
804106	100513	ADCB	(R3)	;ADD CARRY TO EVEN BYTE
804110	100402	BMI	25	;UNTIL EVEN BYTE BECOMES NEGATIVE
804112	100214	INCB	(R4)	;INCREMENT ODD BYTE
804114	000773	BR	13	
804116	102401	2S1	BVB	;(R3)0077000E17743(2003),CC=1010
804120	104400	HLT	,+4	
804122	000242	CLV		
804124	100214	INCB	(R4)	;(R3)0100200E10003(2003),CC=1010
804126	103402	SCB	INCB1	
804130	102001	BVC	INCB1	
804132	100401	BMI	,+4	
804134	104400	INCB1	HLT	;ERROR: INCORRECT CC'S AS SHOWN ABOVE
804136	100114	ROLB	(R4)	;(R3)0000200E10003(2003),CC=0111
804140	100002	BCC	ROLB1	
804142	102001	BVC	ROLB1	
804144	001401	BEQ	,+4	

804146	104400	ROLB1	HLT	;ERROR: INCORRECT CC'S AS SHOWN ABOVE	
804150	100614	SBDB	(R4)	;(R3)0177000E17743(2003),CC=1001	
804152	103002	BCC	SBDB1		
804154	102401	BVB	SBDB1		
804156	100401	BMI	,+4		
804160	104400	SBDB1	HLT	;ERROR: INCORRECT CC'S AS SHOWN ABOVE	
804162	100313	ASLB	(R3)	;(R3)0E7700,CC=0111	
804164	100002	BCC	ASLB1		
804166	102001	BVC	ASLB1		
804170	001401	REQ	,+4		
804172	104400	ASLB1	HLT	;ERROR: INCORRECT CC'S AS SHOWN ABOVE	
804174	100413	NEGB	(R3)	;(R3)0E7700,CC=0100	
804176	103402	BCS	NEGB1		
804200	102401	BVB	NEGB1		
804202	001401	BEQ	,+4		
804204	104400	NEGB1	HLT	;ERROR: INCORRECT CC'S AS SHOWN ABOVE	
804206	000277	BCC			
804210	100313	DECB	(R3)	;(R3)0E7777,CC=1001	
804212	100002	BCC	DECB1		
804214	102401	BVB	DECB1		
804216	001001	BNE	,+4		
804220	104400	DECB1	HLT	;ERROR: INCORRECT CC'S AS SHOWN ABOVE	
804222	000241	CLC			
804224	100013	RORB	(R3)	;(R3)0E7757,CC=0011	
804226	100002	BCC	RORB1		
804230	102001	BVC	RORB1		
804232	100001	BPL	,+4		
804234	104400	RORB1	HLT	;ERROR: INCORRECT CC'S AS SHOWN ABOVE	
804236	000241	CLC			
804240	100114	COMB	(R4)	;(R3)0000177,CC=0101	
804242	100002	BCC	COMB1		
804244	102401	BVB	COMB1		
804246	001401	BEQ	,+4		
804250	104400	COMB1	HLT	;ERROR: INCORRECT CC'S AS SHOWN ABOVE	
804252	100213	1S1	ASRB	(R3)	;SHIFT EVEN BYTE UNTIL V CLEARS
804254	102002	BVC	25		
804256	100514	ADCB	(R4)	;AND ADD CARRY TO ODD BYTE	
804260	000774	BR	13		
804262	103401	2S1	BVB	;(R3)000000,CC=0001	
804264	001401	BEQ	,+4		
804266	104400	ASRB1	HLT	;ERROR: INCORRECT CC'S AS SHOWN ABOVE	
804270	100214	ASRB	(R4)		
804272	100214	ASRB	(R4)	;(R3)000000,CC=0001	
804274	100002	BCC	ASRB1A		
804276	102001	BVC	ASRB1A		
804280	001001	BNE	,+4		

804302	104400	ARB01A	MLT			(ERROR) INCORRECT CC'S AS SHOWN ABOVE
804304	105314		DECB	(R4)		(R3)=000000,CC=0100
804306	001401		BEO	,04		
804310	102400		MLT			(ERROR) INCORRECT CC'S AS SHOWN ABOVE
804312	000201		BCC			
804314	100014		RORB	(R4)		(R3)=00000,CC=1010
804316	103402		BOR	ROR01A		
804320	102001		BVC	ROR01A		
804322	100401		BHI	,04		
804324	104400	RCR01A	MLT			(ERROR) INCORRECT CC'S AS SHOWN ABOVE
804326	000242		CLV			
804330	105314		DECB	(R4)		(R3)=077400,CC=0100
804332	102401		BVS	,04		
804334	104400		MLT			
804336	000201		BCC			
804340	105313		DECB	(R3)		(R3)=077777,CC=1001
804342	103002		BCC	DEC01A		
804344	102401		BVS	DEC01A		
804346	100401		BHI	,04		
804350	104400	DEC01A	MLT			(ERROR) INCORRECT CC'S AS SHOWN ABOVE
804352	000277		BCC			
804354	000313		SWAB	(R3)		(R3)=175777=[17743][177],CC=0000
804356	103402		BOR	SHAB1		
804360	102401		BVS	SHAB1		
804362	100001		BPL	,04		
804364	104400	SHAB1	MLT			(ERROR) INCORRECT CC'S AS SHOWN ABOVE
804366	105714		TSYB	(R4)		(R3)=175777=[17743][177],CC=1000
804370	103402		BOR	TSYB1		
804372	102401		BVS	TSYB1		
804374	100401		BHI	,04		
804376	104400	TSYB1	MLT			(ERROR) INCORRECT CC'S AS SHOWN ABOVE
804400	105014		CLRB	(R4)		(R3)=000177=[00003][177],CC=0100
804402	001401		BEO	,04		
804404	104400		MLT			
804406	106313		ASLB	(R3)		(R3)=000376,CC=1010
804410	103402		BOR	ASLB1A		
804412	102001		BVC	ASLB1A		
804414	100401		BHI	,04		
804416	104400	ASLB1A	MLT			(ERROR) INCORRECT CC'S AS SHOWN ABOVE
804420	105113		COMB	(R3)		(R3)=000001,CC=0001
804422	103002		BCC	COMB1A		
804424	102401		BVS	COMB1A		
804426	100001		BPL	,04		
804430	104400	COMB1A	MLT			(ERROR) INCORRECT CC'S AS SHOWN ABOVE
804432	000313		SWAB	(R3)		(R3)=000400,CC=0100

804434	001401		BEO	,04		
804436	104400		MLT			
804440	105213		INCB	(R3)		
804442	000201		BCC			
804444	100613		SBOR	(R3)		(R3)=000400,CC=0100
804446	001401		BEO	,04		
804450	104400		MLT			
804452	022713	000400	CMF	0400,(R3)		(CHECK REMAINING RESULT)
804456	001401		BEO	,04		
804460	104400		MLT			
804462	104000		SCOPE			
(CHECK UNARY WORD OPS USING ADDRESS MODES 2 AND 4 (AUTO INC/DEC))						
804464	200401		BR	,04		
804466	000000		WORD	0		(ADDRESS RESERVED FOR TESTS)
804470	010704		MOV	PC,R4		
804472	102704	000000	SUB	R4,R4		(R4 AND R5 POINT TO
804476	010400		MOV	R4,R5		RESERVED WORD
804480	000013		CLR	(R5)		(RESET DATA0)
804502	200277		BCC			
804504	200244		CL2			
804506	005725		TST	(R5),		(R5)=000000,CC=0100
804510	103402		BOS	TSY2		
804512	102401		BVS	TSY2		
804514	001401		BEO	,04		
804516	104400	TSY2	MLT			(ERROR) INCORRECT CC'S AS SHOWN ABOVE
804520	005145		COM	=(R5)		(R5)=177777,CC=1001
804522	103001		BCC	COM4		
804524	100401		BHI	,04		
804526	104400	COM4	MLT			(ERROR) INCORRECT CC'S AS SHOWN ABOVE
804530	000241		CLC			
804532	000024		ROR	(R4),		(R4)=077777,CC=0011
804534	100002		BCC	ROR2		
804536	102001		BVC	ROR2		
804540	100001		BPL	,04		
804542	104400	ROR2	MLT			(ERROR) INCORRECT CC'S AS SHOWN ABOVE
804544	000257		CCC			
804546	000244		INC	=(R4)		(R4)=00000,CC=1010
804550	102002		BVC	INC4		
804552	001401		BEO	INC4		
804554	100401		BHI	,04		
804556	104400	INC4	MLT			(ERROR) INCORRECT CC'S AS SHOWN ABOVE
804560	000201		BCC			
804562	000324		SWAB	(R4),		(R4)=000200,CC=1000
804564	103401		BOR	SWAB2		
804566	100401		BHI	,04		
804570	104400	SWAB2	MLT			(ERROR) INCORRECT CC'S AS SHOWN ABOVE

804372	000435		NEG	(R0)←	(R0)←177600,CC=1001
804374	103001		BCC	NEG2	
804376	100401		SHI	,04	
804378	104400	NEG21	HLT		ERROR: INCORRECT CC'S AS SHOWN ABOVE
804382	000444		CLR	←(R4)	
804384	001401		BEG	,04	(R4)←000000,CC=1000
804386	104400		HLT		
804390	000201		BCC		
804392	000445		ROR	←(R5)	(R5)←100000,CC=1010
804394	000201		BCC		
804396	000525		ADC	(R0)←	(R0)←100001,CC=1000
804398	102401		BVS	ADC2	
804400	100401		SHI	,04	
804402	104400	ADC21	HLT		ERROR: INCORRECT CC'S AS SHOWN ABOVE
804406	000202		SEV		
804408	000224		ASR	(R4)←	(R4)←140000,CC=1001
804410	103002		BCC	ASR2	
804412	102401		BVS	ASR2	
804414	100401		SHI	,04	
804416	104400	ASR21	HLT		ERROR: INCORRECT CC'S AS SHOWN ABOVE
804420	000202		SEV		
804422	000144		ROL	←(R4)	(R4)←100001,CC=1001
804424	103002		BCC	ROL4	
804426	102401		BVS	ROL4	
804428	100401		SHI	,04	
804430	104400	ROL41	HLT		ERROR: INCORRECT CC'S AS SHOWN ABOVE
804434	000445		BCC	←(R5)	(R5)←100000,CC=1000
804436	103001		BCC	,04	
804438	104400		HLT		ERROR: 'C' BIT FAILED TO CLEAR
804442	000325		DEC	(R5)←	(R5)←077777,CC=0010
804444	103402		BVS	DEC2	
804446	102001		BVC	DEC2	
804448	100001		SPL	,04	
804450	104400	DEC21	HLT		ERROR: INCORRECT CC'S AS SHOWN ABOVE
804454	000324		ASL	(R4)←	(R4)←177776,CC=1010
804456	102401		BVS	,04	
804458	104400		HLT		
804462	000344		ASL	←(R4)	(R4)←177774,CC=1001
804464	103003		BCC	ASL4	
804466	102402		BVS	ASL4	
804468	001401		BEG	ASL4	
804470	100401		SHI	,04	
804472	104400	ASL41	HLT		ERROR: INCORRECT CC'S AS SHOWN ABOVE
804720	022724	177774	CHP	←177774,(R4)←	
804722	001401		BEG	,04	
804724	104400		HLT		

804730	020425		CHP	R4,R5	
804732	001401		BEG	,04	
804734	104400		HLT		
804736	104000		SCOPE		
804740	000401		BR	,04	RESERVE A WORD
804742	000000		WORD	0	RESERVED WORD
804744	010700		MOV	PC,R5	
804746	102700	000004	SUB	R4,R5	(R5) POINTS TO EVEN BYTE OF RESERVED WORD
804748	010500		MOV	R5,R0	
804750	010002		MOV	R0,R2	
804752	000202		INC	R2	(R5) POINTS TO ODD BYTE OF RESERVED WORD
804754	000010		CLR	(R0)	PRESET
804762	000277		BCC		
804764	000241		CLC		
804766	100125		COMB	(R5)←	(R5)←000377,CC=1001
804768	103002		BCC	COMB2	
804770	102401		BVS	COMB2	
804772	100401		SHI	,04	
804774	104400	COMB21	HLT		ERROR: INCORRECT CC'S AS SHOWN ABOVE
805000	100542		ADCB	←(R2)	(R2)←000000,CC=0101
805002	001401		BEG	,04	
805004	104400		HLT		ERROR: INCORRECT RESULT AS SHOWN ABOVE
805006	100525		ADCB	(R5)←	(R5)←000400,CC=0000
805008	103401		BVS	ADCB2	
805010	001001		BNE	,04	
805012	001001		BNE	,04	
805014	104400	ADCB21	HLT		ERROR: INCORRECT CC'S AS SHOWN ABOVE
805016	000203		+SEC1SEV		
805018	100045		RORB	←(R5)	(R5)←100000,CC=1001
805020	100003		BCC	RORB4	
805022	102402		BVS	RORB4	
805024	001401		BEG	RORB4	
805026	100401		SHI	,04	
805028	104400	RORB41	HLT		ERROR: INCORRECT CC'S AS SHOWN ABOVE
805034	000277		BCC		
805036	104122		ROLB	(R2)←	(R2)←100001,CC=0000
805038	103403		BVS	ROLB2	
805040	102402		BVS	ROLB2	
805042	001401		BEG	ROLB2	
805044	100001		SPL	,04	
805046	104400	ROLB21	HLT		ERROR: INCORRECT CC'S AS SHOWN ABOVE
805052	000277		CCC		
805054	100225		ASRB	(R5)←	(R5)←140001,CC=1010
805056	103402		BVS	ASRB2	
805058	100001		BVC	ASRB2	
805060	100401		SHI	,04	
805062	104400	ASRB21	HLT		ERROR: INCORRECT CC'S AS SHOWN ABOVE

009044	100242	INCB	=(R2)	(R0)0140002,CC=0000
009070	000277	BCC		
009072	100272	ASRB	(R2)+	(R0)0140001,CC=0000
009074	100402	BCC	ASRB2A	
009076	100401	BVC	ASRB2A	
009100	100001	BPL	,04	
009102	104400	ASRB2A1	HLT	ERROR! INCORRECT CC'S AS SHOWN ABOVE
009104	000266	+DEZISEV		ERT 3,Y
009106	100345	ASLB	=(R3)	(R0)0100001,CC=1001
009110	100003	BCC	ASLB4	
009112	102402	BVC	ASLB4	
009114	001401	BEO	ASLB4	
009116	100401	BMI	,04	
009120	104400	ASLB41	HLT	ERROR! INCORRECT CC'S AS SHOWN ABOVE
009122	100322	DECB	(R2)+	(R0)00740010(07743(001),CC=0010
009124	100002	BCC	DECB2	
009126	102001	BVC	DECB2	
009130	100001	BPL	,04	
009132	104400	DECB21	HLT	ERROR! INCORRECT CC'S AS SHOWN ABOVE
009134	100645	SBCB	=(R5)	(R0)007400,CC=0100
009136	100402	BCC	SBCB4	
009140	102401	BVC	SBCB4	
009142	001401	BEO	,04	
009144	104400	SBCB41	HLT	ERROR! INCORRECT CC'S AS SHOWN ABOVE
009146	100442	NECB	=(R2)	(R0)010400,CC=1001
009150	100002	BCC	NECB4	
009152	102401	BVC	NECB4	
009154	100401	BMI	,04	
009156	104400	NECB41	HLT	ERROR! INCORRECT CC'S AS SHOWN ABOVE
009160	100725	TSYB	(R5)+	(R0)0100400,CC=0100
009162	100401	BCC	TSYB2	
009164	001401	BEO	,04	
009166	104400	TSYB21	HLT	
009170	100722	TSYB	(R2)+	(R0)0100400,CC=1000
009172	001401	BEO	TSYB2A	
009174	100401	BMI	,04	
009176	104400	TSYB2A1	HLT	
009200	000261	SEC		
009202	000342	SWAB	=(R2)	(R0)000201,CC=1000
009204	100401	BCC	SWAB4	
009206	100401	BMI	,04	
009210	104400	SWAB41	HLT	
009212	000277	BCC		
009214	100225	INCB	(R5)+	(R0)0004010(0004)(201),CC=0000
009216	100003	BCC	INCB2	
009220	102402	BVC	INCB2	

009222	001401	BEO	INCB2	
009224	100001	BPL	,04	
009226	104400	INCB21	HLT	
009230	002227	000001	CHP	(R2)+,000001
009234	001401	BEO	,04	CHECK END RESULT
009236	104400	HLT		
009240	020205	CHP	R2,R5	CHECK REGISTERS
009242	001401	BEO	,04	
009244	104400	HLT		
009246	104000	SCOPE		
009250	000402	YCHECK	UNARY WORD OPS USING ADDRESS MODES 3 AND 5	
009252	200000	RR	,04	(RESERVE 2 WORDS
009254	000000	WORD	0	11 FOR THE ADDRESS
009256	010703	WORD	0	AND 1 FOR DATA
009260	102703	MOV	PC,R3	
009264	000013	SUB	#4,R3	
009266	010300	CLR	(R3)	(PRESET DATA
009270	000743	MOV	R3,R0	IRB POINTS TO DATA WORD
009272	010013	TSY	=(R3)	
009274	010304	MOV	R0,(R3)	
		MOV	R3,R4	
009276	000257	CCC		
009300	000333	TSY	0(R3)+	(R0)000000,CC=0100
009302	001401	BEO	,04	
009304	104400	HLT		
009306	000261	SEC		
009310	000003	ROR	0=(R3)	(R0)010000,CC=1010
009312	100402	RCS	ROR5	
009314	102001	BVC	ROR5	
009316	100401	BMI	,04	
009320	104400	ROR51	HLT	
009322	000257	CCC		
009324	004234	ASR	=(R4)+	(R0)014000,CC=1010
009326	102001	BVC	ASR3	
009330	100401	BMI	,04	
009332	104400	ASR31	HLT	
009334	000250	CLN		
009336	004333	ASL	0(R3)+	(R0)010000,CC=1001
009340	100002	BCC	ASL3	
009342	102401	BVC	ASL3	
009344	100401	BMI	,04	
009346	104400	ASL31	HLT	
009350	000277	BCC		
009352	000334	DEC	0=(R4)	(R0)0007777,CC=0010
009354	100003	BCC	DEC9	
009356	102002	BVC	DEC9	
009360	001401	BEO	DEC9	

PC	OP	COND	OPCODE	COND	COND	COND
003362	000001		BPL	,04		
003364	104400	DECS1	HLT			
003366	009493		NEC	0=(R3)		(R0)000001, CC=1001
003370	103002		BCC	NEC5		
003372	102401		BVS	NEC5		
003374	100401		BMI	,04		
003376	104400	NECS1	HLT			
003400	000202		BEV			
003402	009134		COM	0=(R4)		(R0)007776, CC=0001
003404	103001		BCC	COM3		
003406	102001		BVC	,04		
003410	104400	COM31	HLT			
003412	009233		INC	0=(R3)		(R0)007777, CC=0001
003414	100001		BCC	INC3		
003416	100001		BPL	,04		
003420	104400	INCS1	HLT			
003422	009554		ADC	0=(R4)		(R0)000000, CC=1010
003424	103402		BCC	ADC5		
003426	102001		BVC	ADC5		
003430	100401		BMI	,04		
003432	104400	ADCS1	HLT			
003434	000297		CCC			
003436	006134		ROL	0=(R4)		(R0)000000, CC=0111
003440	103002		BCC	ROL3		
003442	102001		BVC	ROL3		
003444	001401		BEQ	,04		
003446	104400	ROL31	HLT			
003490	009233		INC	0=(R3)		(R0)000001, CC=0001
003492	009694		SBC	0=(R4)		(R0)000000, CC=0100
003494	103401		BCC	SBC5		
003496	001401		REQ	,04		
003498	104400	SCCS1	HLT			
003492	100000		SCOPE			
003464	000423		ICHECK	UNARY BYTE OPS USING ADDRESS WORDS 3 AND 5		
003466	000070		OR	,+10		RESERVE 3 WORDS
003470	000000		,WORD	0		IF FOR EVEN BYTE ADDRESS
003472	000000		,WORD	0		IF FOR ODD BYTE ADDRESS
003474	010772		,WORD	0		AND 1 FOR DATA
003476	009742		MOV	PC, R2		
003478	009742		TST	0=(R2)		BACK R2 UP TO
003480	009742		TST	0=(R2)		DATA WORD
003482	010200		MOV	R2, R0		R0 POINTS TO THE DATA WORD
003484	000010		CLR	(R0)		PRESET DATA
003486	009742		TST	0=(R2)		BACK R2 UP TO
003488	009742		TST	0=(R2)		EVEN BYTE ADDRESS WORD
003490	010072		MOV	R0, (R2)		LOAD ADDRESS
003492	009200		INC	R0		ODD BYTE ADDRESS

003516	010072		MOV	R0, (R2)		LOAD ODD BYTE ADDRESS
003520	010200		MOV	R2, R0		RESET R0
003522	010205		MOV	R2, R5		
003524	109192		COMB	0=(R2)		(R0)017400, CC=1001
003526	103001		BCC	COMB5		
003530	100401		BMI	,04		
003532	104400	COMB51	HLT			
003534	109792		TSTB	0=(R2)		(R0)017400, CC=0100
003536	001401		BEQ	,04		
003540	104400		HLT			
003542	000202		SEV			
003544	100255		ASRB	0=(R5)		(R0)017400, CC=1001
003546	103002		BCC	ASRB5		
003550	102401		BVS	ASRB5		
003552	100401		BMI	,04		
003554	104400	ASRB51	HLT			
003556	109232		INCB	0=(R2)		(R0)017401, CC=000
003560	103001		BCC	INCB3		
003562	100001		BPL	,04		
003564	104400	INCB31	HLT			
003566	000241		CLC			
003570	100095		RORB	0=(R5)		(R0)017400, CC=0111
003572	103003		BCC	RORB5		
003574	102002		BVC	RORB5		
003576	001001		BNE	RORB5		
003580	100001		BPL	,04		
003582	104400	RORB51	HLT			
003604	100332		ASLB	0=(R2)		(R0)017400, CC=1001
003606	103002		BCC	ASLB3		
003610	102401		BVS	ASLB3		
003612	100401		BMI	,04		
003614	104400	ASLB31	HLT			
003616	109592		ADCB	0=(R2)		(R0)017400, CC=1000
003620	103401		BCC	ADCB5		
003622	100401		BMI	,04		
003624	104400	ADCB51	HLT			
003626	000277		SCC			
003630	100135		ROLB	0=(R5)		(R0)017401, CC=0000
003632	101402		BLOS	ROLB5		BRANCH IF C OR E IS SET
003634	102401		BVS	ROLB5		
003636	100001		BPL	,04		
003640	104400	ROLB51	HLT			
003642	000392		SWAB	0=(R2)		(R0)000077, CC=1000
003644	100401		BMI	,04		
003646	104400		HLT			

009650	000201		BEC			
009650	109639		BECB	0(R0)+		((R0):000377, CC=0100
009654	103401		BCS	00CB3		
009656	001401		BEO	,+4		
009660	104400		00CB3:	HLT		
009662	109632		NECB	0(R2)+		((R0):000001
009664	109392		DECB	0+(R2)		((R0):000000, CC=0101
009666	103001		BCC	DECB9		
009670	001401		BEO	,+4		
009672	104400		DECB9:	HLT		
009674	104400		SCOPE			
009676	009027		ICHECK	UNARY	WORD OPS USING ADDRESS MODE 6 (PC)	
009700	000000		UNM0:	CLR	(PC)+	(PRESET DATA 0 0
009702	010700			,WORD	0	(RESERVED FOR DATA
009704	024040			MOV	PC,RR	
009706	000277			CMF	=(R0),=(R0)	(RR POINTS TO DATA WORD
009710	006107	177704		SCC		
009714	103403			ROL	UNM6	((R0):000001, CC=0000
009716	102402			BCS	ROL6	
009720	001401			BVS	ROL6	
009722	100001			BEO	ROL6	
009724	104400		ROL6:	BPL	,+4	
009726	005107	177706		HLT		
009732	103002			COM	UNM6	((R0):017770, CC=1001
009734	102401			BCC	COM6	
009736	100401			BVS	COM6	
009740	104400		COM6:	BMI	,+4	
				HLT		

003742	004267	177732	ASR	UHM6	(R0)017777, CC=0010
003746	103402		BCS	ASR0	
003750	102001		BVC	ASR0	
003752	104400		BMI	,04	
003754	104400		HLT		
			ASR61		
003756	000277		SCC		
003760	005447	177714	NEB	UHM6	(R0)000001, CC=0001
003764	103003		BCC	NEB6	
003766	102402		BVS	NEB0	
003770	001401		BEQ	NEB0	
003772	100001		BPL	,04	
003774	104400		HLT		
			NEB61		
003776	000277		SCC		
004000	006647	177674	ROR	UHM6	(R0)010000, CC=1001
004004	103003		BCC	ROR6	
004006	102402		BVS	ROR6	
004010	001401		BEQ	ROR6	
004012	100401		BMI	,04	
004014	104400		HLT		
			ROR61		
004016	005647	177690	SBC	UHM6	(R0)007777, CC=0010
004022	103402		BCS	SBC0	
004024	102001		BVC	SBC0	
004026	100001		BPL	,04	
004030	104400		HLT		
			SBC61		
004032	000242		OLV		
004034	009247	177640	INC	UHM6	(R0)010000, CC=1011
004040	103403		BCS	INC6	
004042	102002		BVC	INC6	
004044	001401		BEQ	INC6	
004046	100401		BMI	,04	
004050	104400		HLT		
			INC61		
004052	004207	177422	ASR	UHM6	(R0)014000, CC=1010
004056	000201		SEC		
004060	006307	177614	ASL	UHM6	(R0)010000, CC=1001
004064	103003		BCC	ASL6	
004066	102401		BVS	ASL6	
004070	100401		BMI	,04	
004072	104400		HLT		
			ASL61		
004074	009307	177600	DEC	UHM6	(R0)007777, CC=0011
004100	103002		BCC	DEC0	
004102	102001		BVC	DEC0	
004104	100001		BPL	,04	
004106	104400		HLT		
			DEC61		
004110	005647	177504	A0C	UHM6	(R0)010000, CC=1010
004114	103402		BCS	A0C0	

004116	102001		BVC	A0C6	
004120	100401		BMI	,04	
004122	104400		HLT		
			A0C61		
004124	000242		CLV		
004126	000307	177546	SWAB	UHM6	
004132	102401		BMI	,04	
004134	104400		HLT		
004136	022710	000200	CMF	#200,(R0)	
004142	001401		BEQ	,04	
004144	104400		HLT		
004146	104000		SCOPE		
			ICHECK		
			UNARY		
			BYTE OPS (EVEN/ODD) USING ADDRESS MODE 6 (PC)		
004150	012770	000312	MOV	#UBM6,R0	
004154	003770	001004	ADD	#FACTOR,R0	(R0 POINTS TO ADDRESS OF DATA
004160	005007	000320	CLR	UBM6	ICLEAR DATA
004164	000277		BCC		
004166	000244		CLE		
004170	107707	000310	TSTB	UBM6	
004174	103403		BCS	TSTB6	
004176	102402		BVS	TSTB6	
004200	001001		BNE	TSTB6	
004202	100001		BPL	,04	
004204	104400		HLT		
			TSTB61		
004206	000297		CCC		
004210	107707	000277	TSTB	UBM6+1	(TEST ODD BYTE
004214	001401		BEQ	,04	
004216	104400		HLT		
			CCC61		
004220	105647	000200	S000	UBM6	(R0)000000, CC=0100
004224	103402		BCS	S0006	
004226	102401		BVS	S0006	
004230	001401		BEQ	,04	
004232	104400		HLT		
			S00061		
004234	000201		IS1	SEC	
004236	105207	000290	INCB	UBM6	(LOOP UNTIL (R0)077000, CC=0111
004242	100403		BMI	25	
004244	105507	000243	A0CB	UBM6+1	(INCB INBY INCREMENTS EVEN BYTE
004250	000771		BR	15	(A0CB INCREMENTS ODD BYTE
004252	103001		BCC	INCB6	
004254	102401		BVS	,04	
004256	104400		HLT		
			INCB61		
004260	106307	000220	ASLB	UBM6	(R0)0077000, CC=0111
004264	103003		BCC	ASLB6	
004266	102002		BVC	ASLB6	
004270	001001		BNE	ASLB6	
004272	100001		BPL	,04	
004274	104400		HLT		
			ASLB61		
004276	000242		CLV		
004300	105507	000207	A0CB	UBM6+1	(R0)010000, CC=1010

704304	103402		BCB	ADCB	
704306	103401		BVC	ADCB6	
704310	104401		BMI	,04	
704312	104400		ADCB61	HLT	
704314	000201		BEC		
704316	106001	000171	RDR6	UBM6+1	(R0)0[00000, CC=1016
704322	103402		BCC	RDR66	
704324	103001		BVC	RDR66	
704326	104401		BMI	,04	
704328	104400		RDR661	HLT	
704332	105107	000134	COMB	UBM6	(R0)0[00377 CC=1001
704336	103002		BCC	COMB6	
704340	102401		BVS	COMB6	
704342	100401		BMI	,04	
704344	104400		COMB61	HLT	
704346	000202		BEV		
704350	105407	000137	NEOB	UBM6+1	(R0)0[00377, CC=0001
704354	103002		BCC	NEOB6	
704356	102401		BVS	NEOB6	
704360	100001		BPL	,04	
704362	104400		NEOB61	HLT	
704364	106107	000120	ROLB	UBM6+1	(R0)0[00777, CC=1010
704370	103402		BCC	ROLB6	
704372	102001		BVC	ROLB6	
704374	100401		BMI	,04	
704376	104400		ROLB61	HLT	
704400	106207	000100	ASRB	UBM6	(R0)0[00777, CC=1001
704404	103002		BCC	ASRB6	
704406	102401		BVS	ASRB6	
704410	100401		BMI	,04	
704412	104400		ASRB61	HLT	
704414	105207	000072	INCB	UBM6	(R0)0[00400, CC=0[01
704420	103002		BCC	INCB6A	
704422	102401		BVS	INCB6A	
704424	001401		BCQ	,04	
704426	104400		INCB6A1	HLT	
704430	105307	000057	DECB	UBM6+1	(R0)0[00000, CC=1001
704434	103002		BCC	DECB6A	
704436	102402		BVS	DECB6A	
704440	001401		BEQ	DECB6A	
704442	100401		BMI	,04	
704444	104400		DECB6A1	HLT	
704446	000307	000040	SWAB	UBM6	(R0)000200, CC=1000
704452	103401		BCC	SWAB6	
704454	100401		BMI	,04	
704456	104400		SWAB61	HLT	

704460	104107	000020	ROLB	UBM6	(R0)0000000, CC=0[11
704464	103002		BCC	ROLB6A	
704466	102001		BVC	ROLB6A	
704470	001401		BEQ	,04	
704472	104400		ROLB6A1	HLT	
704474	005707	000012	TST	UBM6	(R0)0000000, CC=0[00
704480	103402		BCC	TST6	
704482	102401		BVS	TST6	
704484	001401		BEQ	,04	
704486	104400		TST61	HLT	
704510	000401		BR	,04	(RESERVE A WORD
704512	000000		UBM61	WORD	WORD RESERVED FOR DATA
704514	104000		SCOPE		
704516	010702		MOV	PC,R2	
704520	002702	000012	ADD	#12,R2	
704524	012707	001132	MOV	@RELOC,PC	(OO RELOCATE PROGRAM CODE
704530	000200		NOP		(PROGRAM RETURNS HERE+2

1000000000000000 LAST ADDRESS OF CODE TO BE RELOCATED 000000000000

Address	Instruction	Comments	Operation	Mode	Notes
80532	MOV PC, R2		MOV	PC, R2	SET PC
80534	TEST = (R0)		TEST	= (R0)	R0 CONTAINS THE ADDRESS OF REL1
80536	MOV R0, RFRSTAD	001010	MOV	R0, RFRSTAD	SAVE
80542	MOV PC, R0		MOV	PC, R0	SET CURRENT PC
80544	MOV R1, R0	000344	MOV	R1, R0	SUBTRACT RELOCATION FACTOR
80550	MOV R0, RFACTOR	001004	MOV	R0, RFACTOR	SAVE RELOCATION FACTOR
80554	MOV PC, R1		MOV	PC, R1	SET NEW SCOPE PTR
80556	UNARY BR UMY	000403	UNARY	BR UMY	RESERVE 3 WORDS FOR ADDRESSES & DATA
80560	UNARY BR UMY	000000	UNARY	BR UMY	CONTAINS ADDRESS OF UMY?
80562	UNARY BR UMY	000000	UNARY	BR UMY	CONTAINS DATA
80564	UNARY BR UMY	000000	UNARY	BR UMY	CONTAINS ADDRESS OF UMY?
80566	MOV PC, R0		MOV	PC, R0	
80570	TEST = (R0)		TEST	= (R0)	
80572	TEST = (R0)		TEST	= (R0)	
80574	CLR R0, R2		CLR	R0, R2	CLEAR TEST DATA
80576	MOV R2, = (R0)		MOV	R2, = (R0)	SET UP ADDRESS
80580	TEST (R0) = 0		TEST	(R0) = 0	MOVE R0 TO NEXT ADDRESS
80582	TEST (R0) = 0		TEST	(R0) = 0	
80584	MOV R2, R0		MOV	R2, R0	SET NEXT ADDRESS
80586	MOV R2, R0		MOV	R2, R0	SET R0 POINTING TO DATA
80588	CLC		CLC		
80590	CLC		CLC		
80592	TEST R2(2)	000002	TEST	R2(2)	(R0) = 000000, CC = 0100
80594	DEC .04		DEC	.04	
80596	HLT		HLT		
80602	BCC R0=17777	177770	BCC	R0=17777	(R0) = 17777, CC = 1001
80604	BCC R0=17777		BCC	R0=17777	
80606	BCS R0=17777		BCS	R0=17777	
80608	BVS R0=17777		BVS	R0=17777	
80610	BHI R0=17777		BHI	R0=17777	
80612	HLT		HLT		
80614	ASL R0=17777	000002	ASL	R0=17777	(R0) = 17777, CC = 1001
80616	ASL R0=17777		ASL	R0=17777	
80618	ASL R0=17777		ASL	R0=17777	
80620	BHI R0=17777		BHI	R0=17777	
80622	HLT		HLT		
80624	CCC R0=17777	000002	CCC	R0=17777	(R0) = 17777, CC = 1000
80626	DEC R0=17777		DEC	R0=17777	
80628	BCS R0=17777		BCS	R0=17777	
80630	BVS R0=17777		BVS	R0=17777	
80632	BHI R0=17777		BHI	R0=17777	
80634	HLT		HLT		

80672	SEV		SEV		
80674	ASR R0=17777	177770	ASR	R0=17777	(R0) = 17777, CC = 1001
80676	BCC ASR7		BCC	ASR7	
80678	BVS ASR7		BVS	ASR7	
80680	BHI ASR7		BHI	ASR7	
80682	HLT		HLT		
80684	CLC		CLC		
80686	SEV		SEV		
80688	ROR R0=17777	177770	ROR	R0=17777	(R0) = 17777, CC = 0000
80690	BLOS ROR7		BLOS	ROR7	BRANCH IF C OR N IS SET
80692	BVS ROR7		BVS	ROR7	
80694	BPL ROR7		BPL	ROR7	
80696	HLT		HLT		
80698	SEV		SEV		
80700	NEG R0=10001	000002	NEG	R0=10001	(R0) = 10001, CC = 1001
80702	BCC NEG7		BCC	NEG7	
80704	BVS NEG7		BVS	NEG7	
80706	BHI NEG7		BHI	NEG7	
80708	HLT		HLT		
80710	CLN SWAB R0=10000	177770	CLN	SWAB R0=10000	(R0) = 10000, CC = 1000
80712	BWAB SWAB7		BWAB	SWAB7	
80714	BSC SWAB7		BSC	SWAB7	
80716	BHI SWAB7		BHI	SWAB7	
80718	HLT		HLT		
80720	SEV		SEV		
80722	COM R0=17717	000002	COM	R0=17717	(R0) = 17717, CC = 1001
80724	BCC COM7		BCC	COM7	
80726	BVS COM7		BVS	COM7	
80728	BHI COM7		BHI	COM7	
80730	HLT		HLT		
80732	SWAB R0=17777	000002	SWAB	R0=17777	(R0) = 17777, CC = 1000
80734	BHI R0=17777		BHI	R0=17777	
80736	HLT		HLT		
80738	BCC ADC R0=17777	177770	BCC	ADC R0=17777	(R0) = 17777, CC = 0000
80740	ADC R0=17777		ADC	R0=17777	
80742	BCS R0=17777		BCS	R0=17777	
80744	BVS R0=17777		BVS	R0=17777	
80746	BPL R0=17777		BPL	R0=17777	
80748	HLT		HLT		
80750	INC R0=10000	000002	INC	R0=10000	(R0) = 10000, CC = 1010
80752	INC R0=10000		INC	R0=10000	
80754	BHI R0=10000		BHI	R0=10000	
80756	HLT		HLT		
80758	CCC R0=10000	000207	CCC	R0=10000	(R0) = 10000, CC = 0111
80760	ROL R0=17777	000172	ROL	R0=17777	(R0) = 10000, CC = 0111

807056	103002		BCC	R0L7	
807060	102001		RVC	R0L7	
807062	001401		BEO	,+4	
807064	104400		R0L7:	HLT	
807066	104000			SCOPE	
807070	003720		ICHECK	UNARY BYTE OPS USING ADDRESS MODE 7	
807072	003210		TST	(R0)+	
807074	003740		INC	(R0)	INCRD FOLLOWING UNWY CONTAINS ADDRESS
807076	003010		TST	-(R0)	(OF ODD BYTE, R0 POINTS TO DATA WORD
807100	010701		CLR	(R0)	IPRESET DATA
			MOV	PC,R1	ISRT SCOPE PTR
INDTE: 02(2) REFERENCES THE ODD BYTE, AND 002(2) REFERENCES THE EVEN BYTE.					
807102	000203		+SECISEV		ISRT C AND V
807104	103672	000002	SBCB	02(2)	;(R0)017400, CC=1001
807110	103003		BCC	R0CB7	
807112	102402		BVS	SBCB7	
807114	001401		BEO	R0CB7	
807116	102401		BM1	,+4	
807120	104400		SBCB7:	HLT	
807122	000277		SCC		ISRT CONDITION CODES
807124	103572	177770	ADCB	002(2)	;(R0)017401, CC=0000
807130	103403		BOS	ADCB7	
807132	102402		BVS	ADCB7	
807134	001401		BEO	ADCB7	
807136	100001		BPL	,+4	
807140	104400		ADCB7:	HLT	
807142	103172	177770	COMB	002(2)	;(R0)017770, CC=1001
807144	103002		BCC	COMB7	
807150	103401		BVS	COMB7	
807152	100401		BM1	,+4	
807154	104400		COMB7:	HLT	
807156	000241		GLC		ICLEAR CARRY
807160	104072	000002	RORB	02(2)	;(R0)007776, CC=0011
807164	103002		RCC	RORB7	
807166	102001		RVC	RORB7	
807170	100001		BPL	,+4	
807172	104400		RORB7:	HLT	
807174	103272	000002	INCB	02(2)	;(R0)000376, CC=1011
807200	103002		BCC	INCB7	
807202	102001		BVC	INCB7	
807204	100401		BM1	,+4	
807206	104400		INCB7:	HLT	
807210	103372	177770	DECB	002(2)	;(R0)000375, CC=1001
807214	103002		BCC	DECB7	
807216	102401		BVS	DECB7	
807220	100401		BM1	,+4	
807222	104400		DECB7:	HLT	

807224	104372	000002	ASLB	02(2)	;(R0)000375, CC=0111
807230	103002		BCC	ASLB7	
807232	102001		RVC	ASLB7	
807234	001401		BEO	,+4	
807236	104400		ASLB7:	HLT	
807240	000241		CLC		ICLEAR CARRY
807242	106272	177770	ASRB	002(2)	;(R0)000376, CC=1001
807246	103002		BCC	ASRB7	
807250	102401		BVS	ASRB7	
807252	100401		BM1	,+4	
807254	104400		ASRB7:	HLT	
807256	103472	000002	NECB	02(2)	;(R0)000376, CC=0100
807262	103402		BOS	NECB7	
807264	102401		BVS	NECB7	
807266	001401		BEO	,+4	
807270	104400		NECB7:	HLT	
807272	000202		SEV		;(R0)000374, CC=1001
807274	106172	177770	R0LB	002(2)	
807300	103002		BCC	R0LB7	
807302	102401		BVS	R0LB7	
807304	100401		BM1	,+4	
807306	104400		R0LB7:	HLT	
807310	103272	177770	INCB	002(2)	;(R0)000375, CC=1001
807314	103272	177770	INCB	002(2)	;(R0)000376, CC=1001
807320	103572	177770	ADCB	002(2)	;(R0)000377, CC=1000
807324	103172	177770	COMB	002(2)	;(R0)000000, CC=0100
807330	001401		BEO	,+4	
807332	104400		HLT		
807334	104000		SCOPE		
807336	000277		ICHECK	BINARY OPS USING ADDRESS MODE 8	ISRT CONDITION CODES
807340	010700		SCC		INR=PC, CC=0001
807342	103002		MOV	PC,R0	
807344	102401		BCC	MOV0	
807346	001001		BVS	MOV0	
807350	104400		BNE	,+4	
807350	104400		MOV0:	HLT	
807352	010002		MOV	R0,R2	IR2=R0
807354	000242		SEV		ISRT V
807356	100002		SUB	R0,R2	IR2=000000, CC=0100
807360	103402		BOS	SUB0	
807362	102401		BVS	SUB0	
807364	001401		BEO	,+4	
807366	104400		SUB0:	HLT	
807370	000244		CLZ		
807372	010203		MOV	R2,R3	IR2=R3000000, CC=0100
807374	103401		BOS	MOV0A	

007376	001401		BEG	,04	
007400	104400	MOV0A1	HLT		
007402	000297		CCC		
007404	000272		*BEVISEN		{SET V & N
007406	020203		CMF	R2,R3	{RRR3000000, CC00100
007410	103403		CCB	CHFB	
007412	102402		BVB	CHFB	
007414	001001		BNE	CHFB	
007416	100001		BPL	,04	
007420	104400	CHP01	HLT		
007422	010002		MOV	R0,R2	{RRR02
007424	010203		MOV	R2,R3	{RRR20R3
007426	000203		ADD	R2,R3	{RRR20R0
007430	004302		ABL	R2	{RRR00R0
007432	020203		CMF	R2,R3	{RRR0320R0
007434	001401		BEG	,04	
007436	104400		HLT		{ERROR! CHECK ADD INSTRUCTION

ITOC FOLLOWING SUBTEST SHIFTS 1 BIT THROUGH R2 AND R5 AND DOES A BIT TEST (BIT) USING R2 AND R5.

007440	005002		CLR	R2	
007442	005202		INC	R2	
007444	000402		BR	Z0	
007446	006302	1S1	ABL	R2	
007450	100407		BMI	40	
007452	010205	2S1	MOV	R2,R5	
007454	000277		SCC		
007456	030205		BIT	R2,R5	{RRR05
007460	103002		BCC	35	
007462	102401		BVB	35	
007464	001370		BNE	15	
007466	104400		HLT		
007470	010205	3S1	MOV	R2,R5	
007472	000297	0S1	CCC		
007474	030205		BIT	R2,R5	
007476	100401		BMI	,04	
007500	104400		HLT		
007502	000002		CLR	R2	
007504	000277		BCC		
007506	030002		BIS	R0,R2	
007510	103002		BCC	0100	
007512	102401		BVS	0100	
007514	001001		BNE	,04	
007516	104400	R1001	HLT		
007520	010003		MOV	R0,R3	
007522	000277		SCC		
007524	000244		CLZ		
007526	040003		BIC	R0,R3	
007530	103003		BCC	BIC0	
007532	102402		BVB	BIC0	

007534	001001		BNE	BIC0	
007536	100001		BPL	,04	
007540	104400	B1001	HLT		
007542	010004		MOV	R0,R4	
007544	003104		COM	R4	
007546	140004		BIC	R0,R4	
007550	003104		COM	R4	
007552	020004		CMF	R0,R4	
007554	001401		BEG	,04	
007556	104400		HLT		
007560	010004		MOV	R0,R4	
007562	003104		COM	R4	
007564	212403		MOV	R4,R3	
007566	030003		BIS	R0,R3	
007570	103001		BCC	0100A	
007572	100401		BMI	,04	
007574	104400	B100A1	HLT		
007576	005203		INC	R3	
007600	001401		BEG	,04	
007602	104400		HLT		
007604	010304		MOV	R3,R4	{RRR0400
007606	003103		COM	R3	{RRR17777
007610	000241		SEC		{SET C
007612	000004		ROR	R4	{R4010000
007614	000304		ADD	R3,R4	{RRR17777,R40077777, CC00011
007616	103003		BCC	ADD0	
007620	102012		BVC	ADD0	
007622	001401		BEG	ADD0	
007624	100001		BPL	,04	
007626	104400	ADD01	HLT		
007630	010700		MOV	PC,RP	
007632	020207		CMF	(R0), (R0)*	
007634	020007		CMF	R0,PC	
007636	001401		BEG	,04	
007640	104400		HLT		
007642	010700		MOV	PC,RP	
007644	002700	000010	ADD	#10, R0	
007650	010002		MOV	R0,R2	
007652	020700		CMF	PC,R0	
007654	001002		BNE	CHP0A	
007656	020200		CMF	R2,R0	
007660	001401		BEG	,04	
007662	104400	CHP0A1	HLT		
007664	104000		SCOPE		

ICHECK BINARY OPS USING ADDRESS MODE 1
 BR ,06 {RESERVE TWO WORDS
 ,WORD 0 {RESERVED FOR SOURCE DATA
 ,WORD 0 {RESERVED FOR DESTINATION DATA
 MOV PC,R4

007676	009744	TSY	=(R4)	
007700	009044	CLR	=(R4)	(R4) POINTS TO DESTINATION DATA
007702	010403	MOV	R4,R3	
007704	009043	CLR	=(R3)	(R3) POINTS TO SOURCE DATA
007706	000113	COM	(R3)	((R3)017777
007710	009214	INC	(R4)	((R4)000001
007712	000202	BEV		(SET V
007714	001314	ADD	(R3),(R4)	((R3)017777,(R4)000000, CC=0101
007716	103002	BCC		
007720	102401	BVS		
007722	001401	BEQ	,04	
007724	104400	ADD1	HLT	
007726	000277	SCC		
007730	000230	CLN		
007732	021314	CHP	(R3),(R4)	((R3)017777,(R4)000000, CC=1000
007734	103403	BGE	CMP1	
007736	102402	BVS	CMP1	
007740	001401	BEQ	CMP1	
007742	100401	BHI	,04	
007744	104400	CH01	HLT	
007746	000277	SCC		
007750	000244	CLZ		
007752	031314	BIT	(R3),(R4)	((R3)017777,(R4)000000, CC=0101
007754	103002	BCC		
007756	102401	BVS	BIT1	
007760	001401	BEQ	BIT1	
007762	104400	BIT1	HLT	
007764	000277	SCC		
007766	000245	+CLCICLZ		
007770	009114	COM	(R4)	((R4)017777
007772	101314	SUB	(R3),(R4)	((R3)017777,(R4)000000, CC=0100
007774	103402	BGS	SUB1	
007776	102401	BVS	SUB1	
010000	001401	BEQ	,04	
010002	104400	SUB1	HLT	
010004	109013	CLR0	(R3)	((R3)017700
010006	000313	SWAB	(R3)	((R3)000377
010010	000270	SEN		
010012	011314	MCV	(R3),(R4)	((R3)0(R4)0000377
010014	100001	BPL	,04	
010016	104400	HLT		
010020	000314	SWAB	(R4)	((R3)000377,(R4)017700
010022	000203	+SECISEV		(SET C & V
010024	051314	BIS	(R3),(R4)	((R3)000377,(R4)017777, CC=1001
010026	103002	BCC	BIS1	
010030	102401	BVS	BIS1	
010032	100401	BHI	,04	
010034	104400	BIS1	HLT	

010036	041314	BIC	(R3),(R4)	((R3)000377,(R4)017700, CC=1001
010040	103002	BCC	BIC1	
010042	102401	BVS	BIC1	
010044	100401	BHI	,04	
010046	104400	BIC1	HLT	
010050	000202	SEV		(SET V
010052	021314	CHP	(R3),(R4)	((R3)000377,(R4)017700, CC=0001
010054	103003	BCC	CMP1A	
010056	102402	BVS	CMP1A	
010060	001401	BEQ	CMP1A	
010062	100001	BPL	,04	
010064	104400	CMP1A	HLT	
010066	009013	CLR	(R3)	((R3)000000
010070	000241	SEC		
010072	000013	NOR	(R3)	((R3)010000
010074	011314	MOV	(R3),(R4)	((R3)0(R4)010000
010076	009114	COM	(R4)	((R4)017777
010100	101314	SUB	(R3),(R4)	((R3)010000,(R4)017777, CC=1011
010102	103002	BCC	SUB1A	
010104	102001	BVC	SUB1A	
010106	100401	BHI	,04	
010110	104400	SUB1A	HLT	
010112	000277	SCC		
010114	101314	SUB	(R3),(R4)	((R3)010000,(R4)017777, CC=0000
010116	101402	BLOS	SUB10	(BRANCH IF C OR Z IS SET
010120	102401	BVS	SUB10	
010122	100001	BPL	,04	
010124	104400	SUB10	HLT	
010126	011314	MOV	(R3),(R4)	((R3)010000,(R4)010000, CC=1000
010130	001401	BEQ	MOV1	
010132	100401	BHI	,04	
010134	104400	MOV1	HLT	
010136	001314	ADD	(R3),(R4)	((R3)010000,(R4)000000, CC=0111
010140	103003	BCC	ADD1A	
010142	102002	BVC	ADD1A	
010144	001001	BNE	ADD1A	
010146	100001	BPL	,04	
010150	104400	ADD1A	HLT	
010152	009113	COM	(R3)	((R3)017777
010154	011314	MOV	(R3),(R4)	((R3)017777
010156	001314	ADD	(R3),(R4)	((R3)017777,(R4)017776, CC=1010
010160	103402	BGS	ADD10	
010162	102001	BVC	ADD10	
010164	100401	BHI	,04	
010166	104400	ADD10	HLT	
010170	002714	ADD	02,(R4)	
010174	009714	TSY	(R4)	(CHECK FINAL RESULT

110176 001401
110200 104400
110202 104400

BCC ,+4
HLT
SCOPE

ICHECK BINARY BYTE OPS USING ADDRESS MODE 1

110204 000402
110206 000000
110210 000000
110212 010703
110214 000745
110216 000045
110220 010502
110222 000042
110224 000202
110226 100112

BR ,+6
,WORD R
,WORD B
MOV PC,R3
TST =(R3)
CLR =(R3) ;(R3)=000000
MOV R3,R2
CLR =(R2) ;(R2)=000000
INC R2 ;R2 POINTS TO ODD BYTE
COMB (R2) ;(R2)=177400

110230 000277
110232 111215
110234 100005
110236 102404
110240 001403
110242 100002
110244 100215
110246 001401
110250 104400

BCC
MOVB (R2),(R3) ;(R2)=177400,(R3)=000377,CC=1001
BCC MOVB1
BVS MOVB1
BEO MOVB1
BPL MOVB1
INCB (R3) ;CHECK RESULT
BEO
MOVB1: HLT ,+4

110252 104312
110254 102376
110256 104012
110260 100315
110262 104015
110264 000277
110266 121512
110270 102001
110272 100401
110274 104400

ASLB (R2) ;SHIFT (R2) UNTIL
BYC ,+2 ;(R2)=000000
RORB (R2) ;(R2)=100000
DECB (R3) ;(R3)=00377
RORB (R3) ;(R3)=000177
CCC
CHPB (R3),(R2) ;(R3)=000177,(R2)=100000,CC=1010
BVC CHPB1
BMI ,+4
CHPB1: HLT

110276 000003
110300 000201
110302 000003
110304 000315
110306 000273
110310 131215
110312 100002
110314 102401
110316 001401
110320 104400

CLR R3
SEC
ROR R3 ;R3=100000
RIS R3,(R3) ;(R3)=100177
+SEC13V1SEN ;SET C,V, & M
RIB (R2),(R3) ;(R2)=100000,(R3)=100177,CC=0101
ACC B1701
RVS B1701
BEO ,+4
B1701: HLT

110322 131215
110324 100001
110326 100401
110330 104400

BISB (R2),(R3) ;(R2)=100000,(R3)=100377,CC=1001
BCC B1801
BMI ,+4
B1801: HLT

110332 141215

BICB (R2),(R3) ;(R2)=100000,(R3)=100177,CC=0001

110334 100002
110336 001401
110340 100001
110342 104400

RCC BICB1
BEO BICB1
BPL ,+4
B1C01: HLT

110344 100112
110346 121215
110350 001401
110352 104400

COMB (R2) ;(R2)=077400,(R3)=100177
CHPB (R2),(R3)
BEO ,+4
HLT

110354 141512
110356 001072
110360 100712
110362 001401
110364 104400

BICB (R3),(R2) ;(R3)=100177,(R2)=000000,CC=0100
BNE B1C01A
TSYB (R2)
BEO ,+4
B1C01A: HLT

110366 000402
110370 000000
110372 000000
110374 100725
110376 000745
110400 100045
110402 010504
110404 100044
110406 010473
110410 100043
110412 010302
110414 100042

BR ,+6 ;RESERVE TWO WORDS FOR DATA
,WORD E ;SOURCE DATA
,WORD B ;DEST DATA
MOV PC,R3
TST =(R3)
CLRB =(R3) ;R3 POINTS TO DEST ODD BYTE
MOV R3,R4 ;R4 POINTS TO DEST EVEN BYTE
CLRB =(R4)
MOV R4,R3 ;R3 POINTS TO SOURCE ODD BYTE
CLRB =(R3)
MOV R3,R2 ;R2 POINTS TO SOURCE EVEN BYTE
CLRB =(R2)

110416 000201

COMMENTS ARE LEAST SIGNIFICANT 4 BITS OF BYTES POINTED TO BY R2,R3
R4, AND R5 RESPECTIVELY AND THE REMAINING BITS ARE 0'S,
SEC ;SET CARRY

110420 104112
110422 111214
110424 104112
110426 111213
110430 104112
110432 111315
110434 104112
110436 104113
110440 131512
110442 131512
110444 001402
110446 131314
110450 131413
110452 001403
110454 100213
110456 121314
110460 001000
110462 100113
110464 121315
110466 001019
110470 100212

ROLB (R2) ;(R2),(R3),(R4),(R5)
MOVB (R2),(R4) ;0001,0000,0000,0000
ROLB (R2) ;0001,0000,0001,0000
MOVB (R2),(R3) ;0010,0010,0001,0000
ROLB (R2) ;0100,0010,0001,0000
MOVB (R3),(R5) ;1000,0100,0001,0010
ROLB (R2) ;1000,0100,0001,0010
ROLB (R3) ;1000,0100,0001,0010
RIB (R2),(R3) ;1000,0100,0001,1010
RIB (R3),(R2) ;1000,0100,0001,1010
BEO B1N1
RIB (R3),(R4) ;1000,0100,0101,1010
RIB (R4),(R3) ;1000,0100,0101,1010
BEO B1N1
INCB (R3) ;1000,0101,0101,1010
CHPB (R3),(R4) ;1000,0101,0101,1010
BNE B1N1
ROLB (R3) ;1000,1010,0101,1010
CHPB (R3),(R5) ;1000,1010,0101,1010
BNE B1N1
ASRB (R2) ;1000,1010,0101,1010

10472	131214		B17B	(R2),(R4)	{0100,1010,0101,1010
10474	081412		BEO	B1N1	
10476	106015		RORB	(R5)	{0100,1010,0101,0101
10500	121415		CMPB	(R4),(R5)	{0100,1010,0101,0101
10502	081007		BNE	B1N1	
10504	105314		DECB	(R4)	{0100,1010,0100,0101
10506	141214		BICB	(R2),(R4)	{0100,1010,0000,0101
10510	081004		BNE	B1N1	
10512	111314		MOVB	(R3),(R4)	{0100,1010,1010,0101
10514	106213		ASRB	(R3)	{0100,0101,1010,0101
10516	141315		BICB	(R3),(R5)	{0100,0101,1010,0101
10520	081401		REG	,+4	
10522	104400		HLT		
10524	104000		B1411	SCOPE	
ICHECK BINARY WORD OPS USING ADDRESS MODE 2 & 4 SET DESTINATION REGISTER					
10526	010409		MOV	R4,R5	
10530	012715	000001	MOV	R1,(R5)	
10534	012712	177777	MOV	R1,(R2)	
10540	000237		CCC		
10542	000202		SEV		
10544	062225		ADD	(R2)+,(R5)+	{(R2)=177777,(R5)=000000,CC=0101
10546	103002		ADD	ADD	
10550	102401		BVS	ADD	
10552	081401		BEO	,+4	
10554	104400		A0021	HLT	
SET V					
10556	002202		SEV		{(R5)=000000,CC=1001
10560	024527	000001	CMF	=(R5),01	
10564	103002		RCC	CMF2	
10566	102401		BVS	CMF2	
10570	100401		BMI	,+4	
10572	104400		CMF21	HLT	
{(R2)=177777,(R5)=177777,CC=1001					
10574	054225		B1B	=(R2),(R5)+	
10576	103001		BCC	B1B2	
10600	100401		BMI	,+4	
10602	104400		W1921	HLT	
10604	000277		SCC		
10606	000244		CLF		
10610	102245		SUB	(R2)+,(R5)	{(R2)=177777,(R5)=000000,CC=0100
10612	103402		BCS	SUB2	
10614	102401		BVS	SUB2	
10616	081401		REG	,+4	
10620	104400		SUB21	HLT	
{(R2)=000001					
10622	005442		REG	=(R2)	
10624	009115		COM	(R5)	{(R5)=177777
10626	000277		SCC		
10630	000290		CLN		
10632	042225		RIC	(R2)+,(R5)+	{(R2)=000001,(R5)=177776,CC=1001
10634	103003		RCC	BIC2	
10636	102402		BVS	BIC2	
10640	081401		REG	BIC2	

10642	102401		BMI	,+4	
10644	104400		BIC21	HLT	
125252					
10646	012742		MOV	0125252,(R2)	
10652	012245		MOV	(R2)+,(R5)	
10654	005125		COM	(R5)+	{(R5)=052525
10656	000202		SEV		
10660	034245		BIT	=(R2),(R5)	{(R2)=125252,(R5)=052525,CC=0101
10662	103002		RCC	BIT2	
10664	102401		BVS	BIT2	
10666	081401		BEO	,+4	
10670	104400		B1921	HLT	
{(R2)=125252,(R5)=177777,CC=1001					
10672	000202		SEV		
10674	052225		B1B	(R2)+,(R5)+	
10676	103002		RCC	B,52A	
10680	102401		RVS	B152A	
10682	100401		BMI	,+4	
10684	104400		B192A1	HLT	
125252					
10686	042745		BIC	0125252,(R5)	{(R5)=052525
10692	009125		COM	(R5)+	{(R5)=125252
10694	042445		CMF	=(R2),(R5)	
10696	081401		BEO	,+4	
10698	104400		HLT		
{(R2)=177777					
10722	005012		CLR	(R2)	
10724	009122		COM	(R2)+	{(R2)=177777
10726	102742	000001	SUB	R1,(R2)	{(R2)=177776,CC=1000
10732	103402		BCS	SUB2A	
10734	102401		BVS	SUB2A	
10736	100401		BMI	,+4	
10740	104400		SUB2A1	HLT	
10742	104000		SCOPE		
GET CURRENT PC					
10744	010702		MOV	PC,R2	
10746	010205		MOV	R2,R5	{MOVE TO R5
10750	124245		CMFB	=(R2),(R5)	{COMPARE ALL PREVIOUS MEMORY ADDRESSES
10752	081401		BEO	,+4	
10754	104400		HLT		{ERROR!
10756	020237	001010	CMF	R2,00PRSTAD	{CHECK FOR LOW LIMIT
10762	081372		BNE	IS	
10764	104000		SCOPE		
ICHECK BINARY BYTE OPS USING ADDRESS MODES 2 & 4; RESERVE TWO WORDS SOURCE DATA DESTINATION DATA					
10766	000402		BR	,+6	{RESERVE TWO WORDS
10770	000000		,WORD	0	{SOURCE DATA
10772	000000		,WORD	0	{DESTINATION DATA
10774	010703		MOV	PC,R5	
10776	009743		YBT	=(R3)	
11000	112743	000200	MOVB	0200,(R3)	
11004	112743	000377	MOVB	0377,(R3)	{(R3)=100377

DBKCC

R11016	112744	000000	MOVE	R0,(R4)	(R4)0077400
R11022	001401		BEG	,04	
R11024	104400		HLT		
R11026	192324		B130	(R3),,(R4)	(R3)0100377,(R4)0077777
R11030	100401		BMI	,04	
R11032	104400		HLT		
R11034	122324		CMPS	(R3),,(R4)	
R11036	103402		DCB	CMPS2	
R11040	102001		BVC	CMPS2	
R11042	100001		BPL	,04	
R11044	104400	CHP021	HLT		
R11046	000201		BCC		
R11050	134344		B170	=(R3),=(R4)	
R11052	103002		BCC	B1702	
R11054	102401		BVS	B1702	
R11056	001401		BEO	,04	
R11058	104400	B19021	HLT		
R11062	000244		CL2		
R11064	144344		B1C0	=(R3),=(R4)	(R3)0100377,(R4)0077400
R11066	001401		BEO	,04	
R11070	104400		HLT		
R11072	104000		SCOPE		
ICHECK BINARY WORD OPS USING ADDRESS MODES 3 & 5.					
R11074	000404		BR	25	(RESERVE SPACE FOR DATA AND ADDRESSES
R11076	000000		,WORD	0	(CONTAINS ADDRESS OF SOURCE DATA
R11100	000070		,WORD	0	(CONTAINS ADDRESS OF DEST DATA
R11102	000070		,WORD	0	(CONTAINS SOURCE DATA
R11104	000070		,WORD	0	(CONTAINS DEST DATA
R11106	010701	251	MOV	PC,R1	
R11110	010130		MOV	R1,R0	(SET SCOPE PTR
R11112	024040		CHP	=(R0),=(R0)	ADJUST R0
R11114	010005		MOV	R0,R0	RS POINTS TO DEST DATA
R11116	024545		CHP	=(R0),=(R0)	ISUB 4 FROM R0
R11120	010015		MOV	R0,(R0)	RS POINTS TO ADDRESS OF DEST DATA
R11122	010072		MOV	R0,R2	
R11124	010024		MOV	R0,R4	IR4 POINTS TO DEST DATA
R11126	005740		TST	=(R0)	
R11130	010003		MOV	R0,R0	RS POINTS TO SOURCE DATA
R11132	010042		MOV	R0,(R2)	IR2 POINTS TO ADDRESS OF SOURCE DATA
R11134	005013		CLR	(R0)	(PRESET SOURCE DATA
R11136	005014		CLR	(R4)	(PRESET DEST DATA
R11140	000277		SCC		
R11142	000244		CL2		
R11144	103235		SUR	=(R2),=(R0)	(R3)000000,(R4)000000,CC=100
R11146	103472		RCS		
R11150	102401		BVS	SUB3	
R11152	001401		BEO	,04	
R11154	104400	SUG31	HLT		

DBKCC

R11156	052792	100000	B13	#100000,0=(R2)	(R3)0100000
R11162	002795	000001	ADD	R1,0=(R0)	(R4)000001
R11166	103235		SUB	=(R2),0=(R0)	(R3)0100000,(R4)0100001,CC=1011
R11170	103002		BCC	SUB3A	
R11172	102001		BVC	SUB3A	
R11174	100401		BMI	,04	
R11176	104400	SUB3A1	HLT		
R11200	005414		NEG	(R4)	(R4)0077777
R11202	035255		B17	=(R2),0=(R0)	(R3)0100000,(R4)0077777
R11204	001401		BEO	,04	
R11206	104400		HLT		
R11210	023235		CHP	=(R2),0=(R0)	
R11212	102401		BVS	,04	
R11214	104400		HLT		
R11216	005192		COM	0=(R2)	
R11220	000297		CCC		
R11222	063235		ADD	=(R2),0=(R0)	
R11224	102001		BVC	ADD3	
R11226	100401		BMI	,04	
R11230	104400	ADD31	HLT		
R11232	000201		SEC		
R11234	045235		B1C	=(R2),0=(R0)	(R3)0077777,(R4)0100000
R11236	103001		BCC	B1C3	
R11240	100401		BMI	,04	
R11242	104400	B1C31	HLT		
R11244	005135		COM	0=(R0)	(R4)0077777
R11246	023235		CHP	=(R2),0=(R0)	(R3)0077777,(R4)0077777
R11250	001401		BEO	,04	
R11252	104400		HLT		
R11254	104000		SCOPE		
ICHECK BINARY BYTE OPS USING ADDRESS MODES 3 & 5.					
R11256	000406		BR	15	(RESERVE SPACE FOR ADDRESSES & DATA
R11260	000070		,WORD	0	(CONTAINS ADDRESS OF SOURCE DATA (EVEN BYTE)
R11262	000070		,WORD	0	(CONTAINS ADDRESS OF SOURCE DATA (ODD BYTE)
R11264	000070		,WORD	0	(CONTAINS ADDRESS OF DEST DATA (EVEN BYTE)
R11266	000070		,WORD	0	(CONTAINS ADDRESS OF DEST DATA (ODD BYTE)
R11270	000070		,WORD	0	(CONTAINS SOURCE DATA
R11272	000070		,WORD	0	(CONTAINS DEST DATA
R11274	010700	151	MOV	PC,R0	
R11276	024040		CHP	=(R0),=(R0)	IR0=ADDRESS OF DEST DATA
R11300	010003		MOV	R0,R0	RS " "
R11302	010305		MOV	R0,R0	RS " "
R11304	005743		TST	=(R0)	ISUB 2 FROM R0
R11306	010043		MOV	R0,(R0)	RS POINTS TO ADDRESS OF DEST DATA
R11310	005213		INC	(R0)	(ODD BYTE
R11312	010043		MOV	R0,(R0)	EVEN BYTE
R11314	010304		MOV	R0,R4	
R11316	005740		TST	=(R0)	(R0=ADDRESS OF SOURCE DATA

811322	005214			INC	(R4)			ODD BYTE
811324	010044			MOV	R0,(R4)			EVEN BYTE
811326	000201			SEC				ISBT CARRY
811330	012734	177001		MOV	#177001,(R4)+			
811334	112734	000200		MOV	#200,(R4)+			SOURCE DATA=100001
811340	115433			MOV	#=(R4),(R3)+			
811342	115433			MOV	#=(R4),(R3)+			DEST DATA=000000
811344	103401			BC	,+4			
811346	104400			HLT				ERROR! MOV DOES AFFECT C BIT IN PSW
811350	022715	000000		CMP	000,(R3)			CHECK DEST DATA
811354	001401			BEQ	,+4			
811356	104400			HLT				ERROR! INCORRECT RESULT
811360	024343			CMP	=(R3),(R3)			POINT R4 BACK TO EVEN BYTE
811362	133433			BIBB	R(R4),(R3)+			
811364	133433			BIBB	R(R4),(R3)+			DEST DATA=100001
811366	022715	100001		CMP	#100001,(R3)			CHECK RESULT
811372	001401			BEQ	,+4			
811374	104400			HLT				ERROR! INCORRECT DEST DATA AFTER BIBB
811376	149433			BICB	#=(R4),(R3)			
811400	149433			BICB	#=(R4),(R3)			
811402	133433			BIBB	R(R4),(R3)+			
811404	001000			ONE	BIBB			
811406	133433			RIBB	#=(R4),(R3)+			
811410	001001			ONE	,+4			
811412	104400			HLT				
811414	123433			CMPB	#(R4),(R3)			
811416	001000			BNE	CMPB			
811420	123433			CMPB	#(R4),(R3)			
811422	001401			BEQ	,+4			
811424	104400			HLT				
811426	104000			SCOPE				
811430	000402			ICHECK	BINARY OPS USING ADDRESS	MODE 4		
811432	000000			OR	,+4			RESERVE TWO LOCATIONS
811434	000000			SDATA	WORD	#		RESERVED FOR SOURCE DATA
811436	000000			DDATA	WORD	#		RESERVED FOR DESTINATION DATA
811438	013702	001004		MOV	#FACTOR,R2			GET RELOCATION FACTOR AND USE AS AN
811442	010205			MOV	R2,R3			INDEX VALUE TO POINT TO DATA
811444	000000	011434		CLR	DDATA(5)			PRESET DESTINATION DATA
811450	012762	000001	011432	MOV	#1,SDATA(2)			THIS ROUTINE PUT A 1 BIT INTO EVERY
811456	036265	011432	011434	RIS	SDATA(2),DDATA(5)			OTHER BIT POSITION IN THE DEST-
811464	006362	011432		ASL	SDATA(2)			INATION ADDRESS (92929)
811470	006362	011432		ASL	SDATA(2)			
811474	103370			BCC	IS			
811476	022765	052925	011434	CMP	#52925,DDATA(5)			CHECK RESULT
811504	001401			BEQ	,+4			
811506	104400			HLT				ERROR! INCORRECT RESULT
811510	012762	177777	011432	MOV	#=1,SDATA(2)			
811516	046502	011434	011432	BIC	DDATA(5),SDATA(2)			SOURCE DATA=129292
811524	036265	011432	011434	RIS	SDATA(2),DDATA(5)			
811532	001401			BEQ	,+4			

811534	104400			HLT				ERROR! BIT INST FAILED
811536	006365	011434		ASL	DDATA(5)			DDATA=129292
811542	026265	011432	011434	CMP	SDATA(2),DDATA(5)			
811550	001401			BEQ	,+4			
811552	104400			HLT				ERROR! CMP INST FAILED
811554	000237			CCC				
811556	066265	011432	011434	ADD	SDATA(2),DDATA(5)			
811564	103002			BCC	ADD6			
811566	102001			BVC	ADD6			
811570	100001			BPL	,+4			
811572	104400			HLT				
811574	006362	011432		ASL	SDATA(2)			SDATA=92924
811600	164265	011432	011434	SUB	SDATA(2),DDATA(5)			
811606	103401			BCC	SUB6			
811610	001401			BEQ	,+4			
811612	104400			HLT				
811614	112700	000377		MOV	#377,R0			IR0=177777 (MOV R0 EXTENDS SIGN)
811620	010002	011432		MOV	R0,SDATA(2)			
811624	012765	177777	011434	PGV	#=1,DDATA(5)			
811632	166500	011434		SUB	DDATA(5),R0			
811636	001401			BEQ	,+4			
811640	104400			HLT				
811642	064265	011432	011434	ADD	SDATA(2),DDATA(5)			
811650	006362	011432		ASL	SDATA(2)			
811654	003102	011432		COM	SDATA(2)			
811660	036265	011432	011434	BIT	SDATA(2),DDATA(5)			
811666	001401			BEQ	,+4			
811670	104400			HLT				
811672	003102	011432		COM	SDATA(2)			
811676	026265	011432	011434	CMP	SDATA(2),DDATA(5)			
811704	001401			BEQ	,+4			
811706	104400			HLT				
811710	026260	011432		CMP	SDATA(2),R0			
811714	001372			BNE	IS			
811716	104000			SCOPE				
811720	013702	001004		MOV	#FACTOR,R2			GET INDEX VALUE
811724	010205			MOV	R2,R4			R2 FOR SOURCE EVEN BYTE INDEX; R4 FOR
811726	010403			MOV	R4,R3			DEST ODD BYTE; R3 FOR SOURCE EVEN
811730	000203			INC	R3			AND R3 FOR DEST ODD BYTE
811732	010309			MOV	R3,R9			
811734	000201			SEC				ISBT CARRY
811736	012762	129292	012002	MOV	#129292,SDATA(2)			
811744	112763	177129	012002	MOV	#177129,SDATA(3)			SOURCE DATA = 002000
811752	016264	012002	012004	MOV	SDATA(2),DDATA(4)			
811760	037764	129129	012004	BIS	#129129,DDATA(4)			DEST DATA = 177777
811766	136263	012002	012002	BIBB	SDATA(2),DDATA(3)			

ICHECK BINARY BYTE OPS USING ADDRESS MODE 6
(NOTE: SDATAB(2), AND DDATAB(4) REFERENCE EVEN BYTE OF SOURCE & DEST DATA
(AND SDATAB(3), AND DDATAB(5) REFERENCE ODD BYTE OF SOURCE & DEST DATA

011774	001401			BEG	,+4	
011776	104400		017001	HLT		
012000	146204	012002	012004	BICB	00ATB(2),00ATB(4)	
012006	103401			BCC	,+4	[ERROR MOV,BIS,BIT/BIC DO NOT AFFECT PC]
012010	104400			HLT		
012012	126304	012002	012004	CHPB	00ATB(3),00ATB(4)	
012020	001401			BEG	,+4	
012022	104400			HLT		
012024	146305	012002	012004	BICB	00ATB(3),00ATB(5)	
012032	126205	012002	012004	CHPB	00ATB(2),00ATB(5)	
012040	001401			BEG	,+4	
012042	104400			HLT		
012044	136504	012004	012004	BIB	00ATB(5),00ATB(4)	
012052	001401			BEG	,+4	
012054	104400			HLT		
012056	104000			SCOPE		
012060	000400			BR	UB7	[RESERVE TWO WORDS
012062	000000		00ATB(1),WORD	B		[RESERVED FOR SOURCE DATA
012064	000000		00ATB(1),WORD	B		[RESERVED FOR DEST DATA
[CHECK BINARY WORD OPS USING ADDRESS MODE 7						
[R0=ADDRESS OF SOURCE DATA, AND R3= ADDRESS OF DEST DATA						
012066	000000			SB(N7)	,WORD	[CONTAINS ADDRESS OF SOURCE DATA
012070	000000			DB(N7)	,WORD	[CONTAINS ADDRESS OF DEST DATA
012072	000000			,WORD	B	[CONTAINS SOURCE DATA
012074	000000			,WORD	B	[CONTAINS DEST DATA
012076	010700			UB7	MOV	PC,R0
012100	224000			CHP	=(R0),=(R0)	
012102	010002			MOV	R0,R2	
012104	224242			CHP	=(R2),=(R2)	
012106	010012			MOV	R0,(R2)	
012110	010203			MOV	R2,R3	
012112	024043			CHP	=(R0),=(R3)	
012114	010013			MOV	R0,(R3)	
012116	000201			SEC		
012120	012777	100000	177700	MOV	#100000,00B(N7)	[SOURCE DATA = 100000
012126	017777	177734	177734	MOV	00B(N7),00B(N7)	[DEST DATA = 100000
012134	103001			RCC		
012136	100401			BMI	,+4	
012140	104400			HLT		
012142	000377	177722		ASL	00B(N7)	[DEST DATA = 000000
012146	102001			BVC	,+4	
012150	001401			BEG	,+4	
012152	104400			HLT		
012154	027777	177700	177700	CHP	00B(N7),00B(N7)	[(R2)=100000,(R3)=000000
012162	103402			BCC	CHP7	
012164	102401			BVS	CHP7	

012166	100401			BMI	,+4	
012170	104400			HLT		
012172	167777	177670	177670	SUB	00B(N7),00B(N7)	[(R2)=100000,(R3)=100000
012200	103003			BCC	SUB7	
012202	102002			BVC	SUB7	
012204	001401			BEG	SUB7	
012206	100401			BMI	,+4	
012210	104400			HLT		
012212	006277	177650		ASR	00B(N7)	[(R2)=140000
012216	007777	177644	177644	ADD	00B(N7),00B(N7)	[(R2)=140000,(R3)=040000
012224	103003			RCC	ADD7	
012226	102002			BVC	ADD7	
012230	001401			BEG	ADD7	
012232	100001			BPL	,+4	
012234	104400			HLT		
012236	047777	177624	177624	BIC	00B(N7),00B(N7)	[(R2)=140000,(R3)=000000
012244	001401			BEG	,+4	
012246	104400			HLT		
012250	057777	177612	177612	BIS	00B(N7),00B(N7)	[(R2)=140000,(R3)=140000
012256	100401			BMI	,+4	
012260	104400			HLT		
012262	027777	177600	177600	CHP	00B(N7),00B(N7)	
012270	001401			BEG	,+4	
012272	104400			HLT		
012274	104000			SCOPE		
[SOME MISCELLANEOUS OPERATION INVOLVING THE PC						
[NOTE: NONE OF THESE OPERATIONS SHOULD AFFECT THE PC						
012276	005000			CLR	R0	
012300	005067	000072		CLR	15	
012304	010707			MOV	PC,PC	
012306	120707			CHPB	PC,PC	
012310	030707			BIF	PC,PC	
012312	000007			ADD	R0,PC	
012314	100707			TSTB	PC	
012316	005007			ADC	PC	
012320	021007			CHP	(R0),PC	
012322	131007			BIB	(R0),PC	
012324	062707	000000		ADD	#0,PC	
012330	023707	001004		CHP	00FACTOR,PC	
012334	133707	001004		BIB	00FACTOR,PC	
012340	000240			NOP		
[THE NEXT TWO INSTRUCTIONS CAUSE THE PROGRAM TO JUMP TO THE UNRELOCATED						
[CODE AND TO RETURN ON THE FOLLOWING INST (IF THE CODE IS RELOCATED)						
012342	163707	001004		SUB	00FACTOR,PC	[JUMPS TO UNRELOCATED CODE
012344	063707	001004		ADD	00FACTOR,PC	[RETURNS
012352	000240			NOP		
012354	024007			CHP	=(SP),PC	
012356	100000			BIB	1SP,PC	

812368	826787	888812	CHP	13,PC	
812384	166787	888808	BUS	13,PC	
812378	848787	888802	BIC	13,PC	
812374	888481		BR	,+4	IBRANCH OVER 13
812376	888888		B		
812488	184888		SCOPE		
812482	818782		MOV	PC,R2	
812484	862782	888812	ADD	R12,R2	
812418	812787	881132	MOV	#RELOC,PC	IC0 RELOCATE PROGRAM CODE
812414	888248		MOV		IPROGRAM RETURNS HERE+2
					111111111111 LAST ADDRESS OF CODE TO BE RELOCATED 1111111111
					1222222222222222 FIRST ADDRESS TO BE RELOCATED 2222222222
812416	818788		REL2) MOV	PC,RR	IGET PC
812428	888748		TST	-(RR)	IRR CONTAINS THE ADDRESS OF REL2
812422	818837	881818	MOV	RR,#FRSTAD	ISAVE
812426	818788		MOV	PC,RR	IGET CURRENT PC
812438	162788	812438	SUB	R1,RR	(SUBTRACT RELOCATION FACTOR
812434	818837	881884	MOV	RR,#FACTOR	ISAVE RELOCATION FACTOR
812448	818781		MOV	PC,R1	ISSET NEW SCOPE PTR
			ICHECK BINARY BYTE OPS USING ADDRESS MODE 7		
812442	888486		BR	BING7	IPRESERVE SPACE FOR ADDRESSES & DATA
812444	888888		SBING7) ,WORD	B	ICONTAINS ADDRESS OF SOURCE EVEN BYTE
812446	888888		,WORD	B	ICONTAINS ADDRESS OF SOURCE ODD BYTE
812458	888888		,WORD	B	ICONTAINS ADDRESS OF DEST EVEN BYTE
812452	888888		,WORD	B	ICONTAINS ADDRESS OF DEST ODD BYTE
812454	888888		DBING7) ,WORD	B	ICONTAINS SOURCE DATA
812456	888888		,WORD	B	ICONTAINS DEST DATA
812468	818788		BING7) MOV	PC,RR	
812462	828848		CHP	-(RR),-(RR)	(RR = ADDRESS OF DEST DATA
812464	818888	177772	MOV	RR,#6(RR)	ILOAD ADDRESS OF DEST EVEN BYTE DATA
812478	818888	177774	MOV	RR,#4(RR)	
812474	888288	177774	INC	-(RR)	ILOAD ADDRESS OF DEST ODD BYTE DATA

812508	888748		TST	-(RR)	IRR=ADDRESS OF SOURCE DATA
812512	818888	177778	MOV	RR,#18(RR)	ILOAD ADDRESS OF SOURCE EVEN BYTE DATA
812506	818888	177772	MOV	RR,#6(RR)	
812512	888288	177772	INC	-(RR)	ILOAD ADDRESS OF SOURCE ODD BYTE DATA
812516	888882		CLR	R2	ISSET INDEX REGISTERS
812528	812783	888882	MOV	R2,R3	(#SBING7(2)I#SBING7(3) REFERENCE EVEN &
812524	812784	177774	MOV	#4,R4	IODD BYTE SOURCE DATA) #DBI#7(4)I#DBING7(5)
812538	812785	177778	MOV	#2,R5	IPREFERENCE DEST EVEN & ODD BYTE DATA
812534	888828		CLR	(RR)#	IPRESET SOURCE DATA
812536	888818		CLR	(RR)	IPRESET DEST DATA
812548	813746	881884	MOV	#FACTOR,#(SP)	IGET RELOCATION FACTOR
812544	861682		ADD	(SP),R2	IAND ADD TO INDEX VALUES
812546	861683		ADD	(SP),R3	
812558	861684		ADD	(SP),R4	
812552	862685		ADD	(SP),R5	
812554	112773	177777	MOV8	#=1,#SBING7(3)	ISRC DATA = 177488
812582	132772	888377	BITB	#377,#SBING7(2)	ICHECK THAT EVEN BYTE WAS NOT AFFECTED
812578	881481		BEQ	,+4	IBY MOV8 INSTRUCTION
812572	184488		HLT		
812574	157374	812444	BISB	#SBING7(3),#DBING7(4)	
812682	185274	812454	#DBING7(4)		ICHECK THAT BIS SET ALL BITS
812686	881481		BEQ	,+4	
812618	184488		HLT		
812612	185375	812454	DECB	#DBING7(5)	IDEST DATA = 177488
812616	885274	812454	INC	#DBING7(4)	IDEST DATA = 177488
812622	125375	812444	CHPB	#SBING7(3),#DBING7(5)	
812638	881481		BEQ	,+4	
812632	184488		HLT		
812634	149375	812444	BICB	#SBING7(3),#DBING7(5)	
812642	881481		BEQ	,+4	
812644	184488		HLT		
812646	185873	812444	CLRB	#SBING7(3)	ISRC DATA = 888888
			ITW8 ROUTINE SETS ALL BITS IN THE SOURCE ODD BYTE BY BISING A BIT FROM		
			ITHE DEST EVEN BYTE INTO THE SOURCE ODD BYTE		
812652	157473	812454	BISB	#DBING7(4),#DBING7(3)	
812688	186174	812454	ROLB	#DBING7(4)	
812684	185372		BCC	BIS7	
812686	822772	177488	CHP	#177488,#DBING7(2)	ICHECK RESULT
812674	881481		BEQ	,+4	
812676	184488		HLT		
812788	888372	812444	SWAB	#SBING7(2)	ISRC DATA = 888377
812784	112775	888288	MOV8	#288,#DBING7(5)	IDEST DATA = 188888
812712	149572	812454	BICB	#DBING7(5),#DBING7(2)	
			BEQ		

012724	103372			BCC	BIC7	
012726	009772	012444		TST	030(IND7(2)	
012732	001401			BES	,+4	
012734	104400			HLT		
012736	104000			SCOPE		
012740	012702	000001	0AERR1	MOV	R1,R2	(LOAD R2 WITH ODD #
012744	010703			MOV	PC,R3	
012746	000401			BR	,+4	(RESERVE SPACE FOR A WORD
012750	000000			,WORD	B	(WILL CONTAIN AN ODD ADDRESS
012752	009773			TST	(R3)+	(STEP R3 TO POINT TO WORD ABOVE
012754	010313			MOV	R3,(R3)	
012756	009213			INC	(R3)	(AND MAKE ODD
012760	012737	013100	000004	MOV	013,0ERRVEC	(SET ODD ADDRESS & RESERVED INSTRUCTION
012766	003737	001004	000004	ADD	00FACTOR,0ERRVEC	
012774	013737	000004	000010	MOV	00ERRVEC,00RESVEC	(TO TRAP TO IS BELOW
013002	000277			BCC		(SET ALL CC'S
013004	100212			SUB	R2,(R2)	
013006	104400			HLT		
013010	000222			ADD	R2,(R2)+	
013012	104400			HLT		
013014	000342			ASL	=(R2)	
013016	104400			HLT		
013020	100912			MFPD	(R2)	(NOTE) MAY BE RESERVED
013022	104400			HLT		
013024	170412			CLAP	(R2)	
013026	104400			HLT		
013030	000222			BIC	(R2)+,R2	
013032	104400			HLT		
013034	104202			SUB	=(R2),R2	
013036	104400			HLT		
013040	159272			BISB	0=(R2),R2	
013042	104400			HLT		
013044	100932			ADCB	0(R2)+	
013046	104400			HLT		
013050	103302			SUB	0(R3)+,R2	
013052	104400			HLT		
013054	009733			TST	0(R3)+	
013056	104400			HLT		
013060	100533			MFPD	0(R3)+	
013062	104400			HLT		
013064	170493			CLRD	0=(R3)	
013066	104400			HLT		
013070	137702	177775		BITB	0,+1,R2	
013074	104400			HLT		
013076	109477	177775		NEGB	0,+1	
013102	104400			HLT		
013104	000470			BR	ZS	
013106	702710	000002	131	ADD	02,(SP)	(ADJUST RETURN PC
013112	032706	000017	000002	BIS	017,2(SP)	(SET CONDITION CODES ON RETURN
013120	000002			PTI		

013122	012726	000900	000004	251	MOV	0STKPTR,SP	(RESET STACK PTR
013126	012737	000906	000004		MOV	0ERRVEC+2,0ERRVEC	
013134	012737	000012	000010		MOV	0RESVEC+2,0RESVEC	
013142	104000				SCOPE		
ICHECK JMP INSTRUCTIONS							
013144	010700				MOV	PC,RP	(SET ADDRESS FOR JMP INST)
013146	002700	000012			ADD	012,R0	(SET CC'S
013152	000277				BCC		
013154	702110				JMP	(R0)	
013156	000402				BR	,+6	(JMP INST JUMPS HERE
013160	000290				CLN		
013162	000775				BR	,+4-	
013164	103003				BCC	JMP1	
013166	102002				RVC	JMP1	
013170	001001				BNE	JMP1	
013172	100001				BPL	,+4	
013174	104400			JMP11	HLT		(ERROR! INCORRECT CC'S AFTER JMP
013176	009002				CLR	R2	(SET INDICATOR
013200	010703				MOV	PC,R3	
013202	000401				BR	,+4	(RESERVE WORD FOR JMP ADDRESS
013204	000000				,WORD	B	(CONTAINS ADDRESS FOR JMP INST
013206	009773				TST	(R3)+	
013210	010313				MOV	R3,(R3)	
013212	010300				MOV	R3,R0	
013214	002713	000022			ADD	022,(R3)	(R3) IS JMP ADDRESS
013220	010300				MOV	R3,R0	
013222	000133				JMP	0(R3)+	(JUMP TO ADDRESS CONTAINED IN R3
013224	000402				BR	,+6	
013226	009102				COM	R2	(COMPLEMENT INDICATOR
013230	000775				BR	,+4	
013232	009202				INC	R2	(CHECK INDICATOR
013234	001003				BNE	JMP3	
013236	009700				TST	(R0)+	
013240	020003				CHP	R0,R3	(CHECK AUTO-INC R3
013242	001401				BES	,+4	
013244	104400			JMP31	HLT		
013246	009002				CLR	R2	(SET INDICATOR
013250	010704				MOV	PC,R4	(SET UP JMP REGISTER
013252	010400				MOV	R4,R0	(SET UP CHECK REGISTER
013254	000402				BR	15	
013256	009102				COM	R2	(COMPLEMENT INDICATOR
013260	000403				BR	ZS	
013262	022474	131			CHP	(R4)+,(R4)+	
013264	009774				TST	(R4)	(R4) JMP ADDRESS
013266	000144				JMP	=(R4)	(USE R4 AS ADDRESS
013270	705202	251			INC	R2	(CHECK INDICATOR
013272	001003				BNE	JMP4	
013274	022000				CHP	(R0)+,(R0)+	(CHECK AUTO-INC R0

013300	001401		BEG	,04		
013302	104400	JMP41	HLT			
013304	210703		MOV	PC,R3		
013306	000401		BR	,04	(RESERVE WORD FOR JMP ADDRESS	
013310	000000	15	,WORD	0	(CONTAINS JUMP ADDRESS	
013312	005703		TST	(R3),		
013314	010313		MOV	R3,(R3)		
013316	002703	000010	ADD	#10,(R3),		
013322	210300		MOV	R3,R0	(LOAD CHECK REGISTER	
013324	000402		BR	35		
013326	005102	25	COM	R2		
013330	000401		BR	45		
013332	000153	35	JMP	00,(R3)	(JUMP TO 25 VIA 15 ABOVE	
013334	005202	45	INC	R2	(CHECK INDICATOR	
013336	001003		BNE	JMP5		
013340	005740		TST	=(R0)		
013342	020003		CMP	R0,R3	(CHECK AUTO-DEC R3	
013344	001401		BEG	,04		
013346	104400	JMP51	HLT			
013350	000402		BR	25		
013352	005102	15	COM	R2	(COMPLEMENT INDICATOR	
013354	000402		BR	35		
013356	000107	177770	JMP	15		
013362	005202	35	INC	R2		
013364	001401		BEG	,04		
013366	104400	JMP61	HLT			
013370	212707	013400 000020	MOV	#15,R5	(SET UP JMP ADDRESS	
013376	005707	001004 000012	ADD	00FACTOR,R5	(ADD RELOCATION FACTOR	
013404	000402		BR	25	(GO TO JMP 075 INST	
013406	005102	15	COM	R2	(COMPLEMENT INDICATOR	
013410	000403		BR	35	(GO TO CHECK ROUTINE	
013412	000177	000000	JMP	075	(JUMP TO 15 ABOVE VIA 75	
013416	000000		,WORD	0	(CONTAINS JMP ADDRESS	
013420	005202	35	INC	R2	(CHECK INDICATOR	
013422	001401		BEG	,04		
013424	104400	JMP71	HLT			
013426	104400		SCOPE			
013430	013705	001004	(CHECK JSR INSTRUCTIONS			
013434	012702	013400	JSR(TST)	MOV	00FACTOR,R5	(GET RELOCATION FACTOR
013440	000502		MOV	#35,R2	(FORM REST ADDR	
013442	000277		ADD	R5,R2	(ADD RELOCATION FACTOR	
013444	000242		SCC		(PRESET CC'S	
013446	000242		CLV			
013448	004512		JSR	R5,(R2)	(GO TO 35 VIA R2	
013450	005702	15	TST	R2	(CHECK INDICATOR	
013452	001017		RNE	JSR1	(R2 SHOULD=0	
013454	023705	001004	CMP	00FACTOR,R5	(CHECK THAT R5 R5 RESTORED R5	
013460	001014		RNE	JSR1		
013462	000414		BR	JSR1A	(EXIT TO SCOPE	
013464	000205	25	RTS	R5	(RETURN FROM SUBROUTINE	

013466	103011	35	BCC	JSR1	(CHECK THAT JSR DID NOT	
013470	102410		RYS	JSR1		
013472	001007		BNE	JSR1	(AFFECT CC'S	
013474	100000		RPL	JSR1		
013476	005000		CLF	R2	(CLEAR INDICATOR	
013500	012704	013490	MOV	#15,R4	(GET UNRELOCATED RETURN ADDRESS	
013504	001604		ADD	(SP),R4	(ADD RELOCATION FACTOR (OLD R5)	
013508	020405		CMP	R4,R5	(CHECK THAT OLD R5 WAS PLACED ON THE	
013510	001705		REQ	25	(STACK; & THAT NEW R5 CONTAINS RETURN PC	
013512	104400	JSR11	HLT		(ERROR) ABOVE	
013514	013704	001004	JSR1A1	MOV	00FACTOR,R4	(GET RELOCATION FACTOR
013520	005000		CLR	R0	(SET INDICATOR	
013522	012705	013942	MOV	#15,R5		
013526	000405		ADD	R4,R5	(SET UP JSR DEFERRED ADDR	
013530	010502	013590	MOV	R5,R2		
013532	212715		MOV	#5,(R5)		
013536	000415		ADD	R4,(R5)	(R5) DEST ADDR	
013540	000401		BR	25	(RESERVE WORD FOR ADDRESS	
013542	000000	15	,WORD	0	(CONTAINS DEST ADDR FOR JSR	
013544	004435	25	JSR	R4,0(R5),	(JSR TO 35 VIA 15 ABOVE	
013546	005200	35	INC	R0	(CHECK INDICATOR	
013550	001013		BNE	JSR3		
013552	000413		BR	JSR3A		
013554	005100	45	COM	R0	(COMPLEMENT INDICATOR	
013556	000204		RTS	4	(RETURN FROM SUBROUTINE	
013560	012703	013940	MOV	#35,R3	(GET UNRELOCATED RETURN ADDRESS	
013564	001603		ADD	(SP),R3	(ADD RELOCATION FACTOR (OLD R4)	
013566	020403		CMP	R4,R3		
013570	001003		BNE	JSR3		
013572	005722		TST	(R2),		
013574	020205		CMP	R2,R5	(CHECK AUTO-INC R5	
013576	001704		BEG	45	(GO TO RTS	
013600	104400	JSR31	HLT		(ERROR ABOVE	
013602	013704	001004	JSR3A1	MOV	00FACTOR,R4	(GET UNRELOCATED RETURN ADDRESS
013606	010400		MOV	R4,R5	(ADD RELOCATION FACTOR (OLD R4)	
013610	013703		MOV	PC,R3		
013612	000401		BR	25		
013614	000405	15	BR	45		
013616	022323	25	CMP	(R3),,(R3),		
013620	000277		SCC			
013622	004443		JSR	R4,0(R3)	(GO TO 25	
013624	104400	35	HLT			
013626	000414		BR	JSR4A		
013630	103012	45	BCC	JSR4		
013632	102011		BVC	JSR4		
013634	001010		RNE	JSR4		
013636	100007		BPL	JSR4		
013640	012702	013620	MOV	#35,R2	(GET UNRELOCATED RETURN ADDRESS	
013644	001602		ADD	(SP),R2	(ADD RELOCATION FACTOR (OLD R4)	
013646	020204		CMP	R2,R4	(CHECK THAT CALCULATED RETURN	
013650	001003		BNE	JSR4		
013652	000414		BR	JSR4A	(PC = NEW R4	

013694	000204			RTB	R4		
013696	104400			HLT			
			JBR41				
013600	000401			JBR441	DR	23	
013602	000400			LSI	DR	33	
013604	010700			DSI	MOV	PC,R0	
013606	004707	177770		JBR	PC,15		
013672	100407			BMI	JBR6A		
013674	104400			HLT			
013676	022070			3SI	CHP	(R0)+,(R0)+	
013700	000110			CHP	R0,(R0)		[CHECK THAT RETURN ADDRESS IS ON THE
013702	001401			BEQ	,+4		STACK
013704	104400			HLT			
013706	000270			SEN			[SET N
013710	000207			RTS	PC		
013712	104000			JBR6A1	SCOPE		
				ICHECK	10Y TRAP	(AND R0L0/ASL0)	
013714	012737	013740	000020	MOV	#10T1,#010TVCC		
013722	003737	001004	000020	ADD	#FACTOR,#010TVCC		[ADD RELOCATION FACTOR
013730	000201			SEC			[SET CARRY
013732	013737	177770	000022	MOV	#PSW,#010TVCC02		[RETAIN CURRENT PSW ON TRAP
013740	000000			CLR	R0		[PRESET R0
013742	000004			10Y			
013744	000403			BR	10Y1A		
013746	104100			10Y1I	R0L0		[ROTATE R0
013750	100376			BVC	,+2		[UNTIL V SETS (R0+200)
013752	000002			RTI			
013754	106300			10Y1A1	ASL0	R0	[SWIPT SHOULD SET CARRY
013756	103004			BCC	10Y10		
013760	102003			BVC	10Y10		
013762	001002			BNE	10Y10		
013764	009700			TSY	R0		[R0 SHOULD BE
013766	001401			BEQ	,+4		
013770	104400			10Y10I	HLT		[ERROR! R0L/ASL FAILED TO SET CC'S PROPERLY
013772	012737	000022	000020	MOV	#10TVCC+2,#010TVCC		[RESTORE 10Y TRAP
014000	005037	000022		CLR	#10TVCC+2		VECTOR
014004	104000			SCOPE			
				ICHECK	EMT TRAP	SEQUENCE	
014006	013746	000030		MOV	#EMTVCC,#(R0)		[SAVE SCOPE PTR
014012	012737	014040	000030	MOV	#EMT1,#EMTVCC		[SET EMT TRAP VECTOR
014020	003737	001004	000030	ADD	#FACTOR,#EMTVCC		[ADD RELOCATION FACTOR
014026	000202			SEC			[SET V
014030	013737	177770	000032	MOV	#PSW,#EMTVCC02		[RETAIN CURRENT PSW ON TRAP
014036	000205			=SEC:SEC			
014040	104000			EMT			[TRAP TO EMT1
014042	001433			BEQ	EMT10		[GO TO EMT1C
014044	104400			HLT			[ERROR! INCORRECT CC'S WERE SET ON RETURN
014046	100227			1EMT1I	BVC	EMT10	[V/ SHOULD'VE SET ON EMT TRAP
014050	100100			R0	COMB	R0	[R0+000377,CC10=1001
014052	100500			ADCB	R0		[R0+000000,CC10=0101
014054	104000			R0RB	R0		[R0+000200,CC10=1010

014056	100023			BVC	EMT10		
014060	100022			BPL	EMT10		
014062	000297			CCC			
014064	100400			NEGB	R0		[R0+000200,CC10=0101
014066	102017			BVC	EMT10		
014070	100016			BPL	EMT10		
014072	000242			CLV			[CLEAR /V/
014074	000201			SEC			[AND SET /C/
014076	100300			DECB	R0		[R0+000177,CC10=0011
014100	102012			BVC	EMT10		
014102	100411			BMI	EMT10		
014104	000242			CLV			[CLEAR /V/
014106	100200			INCB	R0		[R0+000200,CC10=0101
014110	100006			BCC	EMT10		
014112	102005			BVC	EMT10		
014114	100004			BPL	EMT10		
014116	000242			CLV			[CLEAR /V/
014120	106200			ASRB	R0		[SWIPT R0 UNTIL /V/ CLEARS
014122	102776			BYS	,+2		
014124	000401			BR	,+4		
014126	104400			EMT10I	HLT		[ERROR!
014130	000002			RTI			[EXIT WITH R0+000377
014132	100500			EMT10I	ADCB	R0	[R0+000000
014134	100003			BCC	EMT10		
014136	001002			BNE	EMT10		
014140	009700			TSY	R0		
014142	001401			BEQ	,+4		
014144	104400			EMT10I	HLT		
014146	012637	000030		MOV	(R0)+,#EMTVCC		[RESTORE SCOPE PTR
014150	005037	000032		CLR	#EMTVCC+2		
014156	104000			SCOPE			
				ICHECK	TRAP INSTRUCTION TRAP SEQUENCE		
				HLT=10Y			[RECEP'INE HLT
014160	000004			MOV	#TRAPVEC,#010TVCC		[SET 10Y (HLT) TRAP VECTOR
014166	012737	000034	000020	MOV	#TRAP1,#0TRAPVEC		[SET TRAP VECTOR
014174	003737	001004	000034	ADD	#FACTOR,#0TRAPVEC		[ADD RELOCATION FACTOR
014202	000270			SEN			[SET N
014204	013737	177770	000036	MOV	#PSW,#0TRAPVEC02		[RETAIN CURRENT PSW ON TRAP
014212	000201			SEC			[SET CARRY
014214	110700			MOV0	PC,R0		
014216	000204			SEN			[SET 0 BIT
014220	104400			TRAP			[TRAP TO TRAP1
014222	103401			3SI	,+4		
014224	000004			BCC			
014226	001401			HLT			
014230	000004			BEQ	,+4		
014232	000412			TRAP1I	BR	TRAP10	
014234	100401			BMI	,+4		[N BIT GOT SET ON TRAP
014236	000004			HLT			
014240	002700	000004		ADD	R0,R0		
014244	120016			CHP0	R0,(R0)		[CHECK LOW BYTE OF RETURN PC ON
014246	001401			BEQ	,+4		STACK
014250	000004			HLT			

```

814292 124646      CHPB  =18P),=(SP)
814294 032626      BIT   (SP)+,(SP)+
814296 088872      RTI   ;RETURN TO INST FOLLOWING TRAP (13)

814288 013737 000020 000034 TRAP(1) MOV  @10TVEC,@TRAPVEC ;RESTORE TRAP (HLT) TRAP VECTOR
814286 012737 000020 000036 MOV  @PRTY4,@TRAPVEC2
814274 012737 000022 000038 MOV  @10TVEC+2,@10TVEC
814302 005937 000022      CLR  @10TVEC+2
814306 104008      SCOPE
104408      HLT=TRAP ;RESTORE HLT TO A TRAP INST

814310 010702      MOV  PC,R2
814312 002722 000012 ADD  #12,R2
814316 012727 001132 MOV  @RELOC,PC ;GO RELOCATE PROGRAM CODE
814322 000240      NOP  ;PROGRAM RETURNS HERE+2
1222222222222222 LAST ADDRESS OF CODE TO BE RELOCATED 222222222222

814324 010701      MOV  PC,R1 ;SET SCOPE POINTER

;THE BELOW ROUTINE ASCERTAINS WHICH CP & CP OPTIONS THE PROGRAM IS RUN-
;NING ON AND SETS AN INDICATOR IN OPT,CP ACCORDINGLY.
CPCHK1 TST  ICNT ;CHECK IF PASS 0
      BNE  REL3 ;DO NOT EXECUTE ROUTINE IF NOT PASS 0
      MOV  @RT1,@ERRVEC+2 ;SET UP ERROR TRAP TO RETURN
      CLR  #3,RP

      SEC
      TST  @PPIRQ ;RR=3 IF 11/45
      R0   ;RR=2 IF 11/40

      SEC
      TST  @PSS+1 ;RR=1 IF 11/20

      SBC  R0
      CLR  @0177700 ;RR=0 IF 11/05
      ASL  R0 ;SHIFT INDICATOR
      MOV  @R0,(PC)+ ;SET CP INDICATOR

OPT,CP1 WORD 0 ;CONTAINS OPTION & CP INDICATORS
      EVEN BYTE1 @11/05,@11/20,@11/40,@11/45
      ODD  BYTE1 @00=MEM MGMT,100=EIS, 40=11/45 FLOATING POINT

      SEC
      TST  @0SRP ;CHECK IF MEM MGMT IS AVAILARLE
      BCS  15

      R15B @200,OPT,CP+1 ;SET MEM MGMT AVAIL INDICATOR
      MOV  @RT1,@RESVEC+2 ;SET RESERVED INSTRUCTION VECTOR
      CLR  R2

      SEC
      ASH  R2,R2 ;WILL TRAP IF 11/40 WITHOUT EIS
      BCS  25 ;BRANCH IF NO EIS AVAILARLE
      R18 @EISOPT,OPT,CP ;SET EIS AVAIL INDICATOR
      SEC ;SET CARRY
      TST  R0 ;WILL CLEAR CARRY IF 11/45 FLOATING POINT
      CFCC ;EIS AVAIL; COPY FLOATING CP'S INTO PSS
      RCS  35 ;BRANCH IF FLOATING POINT IS NOT AVAIL;
      S18B @40,OPT,CP+1 ;SET FLOATING POINT OPTION AVAIL INDICATOR
      CLR  @ERRVEC+2 ;RESTORE ERROR TRAP TO HALT ON TRAP
      CLR  @RESVEC+2

```

```

814470 126727 1777F2 000004      CHPB  OPT,CP,#4 ;CHECK IF CP IS 11/40, OR 11/45
814476 002403      RL7  REL3 ;BRANCH IF 11/05 OR 11/20
814500 012707 0000F6 002176      MOV  @RT1,RTI1 ;SET /RT/ BIT RETURN TO RTI

;3333333333333333 FIRST ADDRESS TO BE RELOCATED 3333333333
814506 010700      REL31 MOV  PC,R0 ;GET PC
814510 005740      TST  @1R0 ;IF CONTAINS THE ADDRESS OF REL3
814512 010037 001010      MOV  @0,@FRSTAD ;SAVE
814516 010700      MOV  PC,RP ;GET CURRENT PC
814520 102700 014920      SUB  #1,R0 ;SUBTRACT RELOCATION FACTOR
814524 010037 001004      MOV  @R0,@FFACTOR ;SAVE RELOCATION FACTOR
814530 010701      MOV  PC,R1 ;SET NEW SCOPE PTR

;CHECK STACK OVERFLOW
814532 013767 1777F6 000350 OVPLM1 MOV  @PSS+75 ;SAVE STATUS IN 75 BELOW
814540 000037 1777F6      CLR  @PSS ;SET KERNEL MODE
814544 010746      MOV  PC,(SP) ;PUSH CURRENT PC ONTO STACK
814546 002716 000200      ADD  @25,(SP) ;FORM ADDRESS OF 28 BELOW
814552 011637 200004      MOV  (SP),@ERRVEC ;SET ERROR VECTOR
814556 012737 000340 000006      MOV  @340,@ERRVEC+2 ;SET PRIORITY LEVEL 7 ON TRAP
814564 002716 000074      ADD  @415+25,(SP) ;FORM ADDRESS OF 415 BELOW
814570 012637 000020      MOV  (SP)+,@10TVEC ;SET 10T TRAP VECTOR TO 415
814574 012746 000340      MOV  @340,(SP) ;340,@1SP
814600 011637 000022      MOV  (SP),@10TVEC+2 ;SET PRIORITY LEVEL 7 ON 10T TRAP
814604 010746      MOV  PC,(SP) ;PUSH CURRENT PC ONTO THE STACK
814606 002716 000006      ADD  @0,(SP) ;ADD OFFSET TO INST FOLLOWING RTI
814612 000002      RTI  ;SET PRIORITY LEVEL 7,CLEAR /RT/ BIT
;AND EXECUTE FOLLOWING INST NEXT

814614 012703 000370      MOV  #376,R3 ;LOAD 376 INTO ADDRESS 376
814620 010313      MOV  R3,(R3) ;SET STACK PTR AT BOUNDARY
814622 010376      MOV  #3,SP
814624 032767 140000 000256      BIT  #UM,75 ;CHECK IF ENTERED TEST IN KERNEL
814632 001030      BNE  15 ;MODE; BRANCH IF NOT IN KERNEL

;THE BELOW INSTRUCTIONS SHOULD NOT CAUSE AN OVERFLOW TRAP
814634 005716      TST  (SP) ;BECAUSE TST IS A NON MODIFYING INST
814636 021666 1777F6 014376      CMP  (SP),@2(SP) ;SD IS COMPARE
814642 122737 000002      CHPB  #2,@OPT,CP ;CHECK IF 11/20 OR 11/35
814650 002411      BLT  125 ;BRANCH IF 11/40 OR 11/45
814652 001474      BEQ  115 ;BRANCH IF 11/20
814654 012767 000014 000176      MOV  @14,915 ;CHANGE CHECK WORD IN 915 IF 11/05
814662 000407      BR  105
814664 012767 000034 000106 1151      MOV  @34,915 ;CHANGE CHECK WORD IN 915 IF 11/20
814672 000403      BR  105
814674 012656      MOV  (SP)+,@(SP) ;BECAUSE OF ADDRESS MODE 9
814676 005476 000000      B10  -(SP),0(SP) ;BECAUSE OF ADDRESS MODE 7
814702 005006 000004 1051      CLR  4(SP) ;BECAUSE DEST ADDRESS IS > 376
814706 007636 000000      B10  0(SP),0(SP)+ ;BECAUSE OF ADDRESS MODE 3
814712 000423      BR  35 ;BRANCH OVER NON KERNEL MODE TESTS

```

```

R14714 154737 000171 177777 151 B100 75=1,00PSW=1 ;RESTORE MODE BITS IN PSW
R14722 012706 000376 MOV 0376,SP ;SET STACK PTR
R14726 010646 177776 MOV =2(0P),=(0P) ;SHOULD NOT TRAP
R14732 051610 B10 (0P),(0P)
R14734 061680 177776 ADD (0P),=2(0P)
R14740 105037 177777 CLRB 00PSW=1 ;SET KERNEL MODE
R14744 000406 BR 05 ;EXIT TEST

;ERROR SERVICE ROUTINE
R14746 012600 251 MOV (0P),R0 ;SAVE PC OF INSTRUCTION THAT TRAPPED
R14750 012602 MOV (0P),R2 ;SAVE PSW
R14752 012706 000300 MOV 05TKPTR,SP ;SET STACK PTR
R14756 104400 HLT ;[ERROR] AN INSTRUCTION THAT WAS NOT
;SUSPECTED TO TRAP TRAPPED
;R0 CONTAINS PC, R2 CONTAINS PSW
;EXIT TEST
R14760 000450 BR 05

;THE BELOW INSTRUCTIONS WILL CAUSE A STACK OVERFLOW
;STACK PTR IS AT 376
R14762 062737 000000 000004 351 ADD #45=20,00ERRVEC ;SET ERROR VECTOR TO 45
R14770 010306 MOV R3,SP ;SET STACK PTR AT 376
R14772 012702 000001 MOV #1,R2
R14776 000000 CLR R0
R15000 000010 CLR (0P) ;SETS BIT 0 IN R0
R15002 004302 ASL R2 ;SHIFT INDICATOR BIT
R15004 100206 INCB (0P)+ ;SETS BIT 1 IN R0
R15006 004302 ASL R2
R15010 000746 ADD PC,=(0P) ;SETS BIT 2 IN R0
R15012 004302 ASL R2
R15014 000000 TOT ;SETS BIT 3 IN R0
R15016 004302 ASL R2
R15020 004767 000014 JSR PC,475 ;SETS BIT 4 IN R0
R15024 004302 ASL R2 ;[NOTE] 11/05 WITHOUT ECO # K011A-00005
;DOES NOT SET BIT 4.
R15026 052600 177776 B10 SP,=2(0P) ;SETS BIT 5 IN R0
R15032 000407 BR 05

;PROGRAM WILL TRAP HERE ON OVERFLOW TRAP
R15034 050200 451 B10 R2,R0 ;SET APPROPRIATE BIT IN R0
R15036 000002 RTI ;RETURN FROM TRAP

R15040 000207 4001 RTS PC

R15042 012737 000022 000020 4151 MOV #107VEC+2,0=107VEC
R15050 000002 RTI

;CHECK THAT ABOVE INSTRUCTIONS DID TRAP
R15052 012726 000000 561 MOV 05TKPTR,SP ;SET STACK PTR
R15056 022700 5051 CMP (PC)+,R0 ;EACH INSTRUCTION SET A BIT IN R0
R15060 000000 5151 ,WORD R ;CONTAINS CHECK WORD
R15062 001407 REC 05 ;R0=77 IF 40 OR 49,14 IF 05,34 IF 20
R15064 100737 014376 TSTB 00OPT,CP ;CHECK IF 11/05
R15070 001003 BNE 525 ;BRANCH IF NOT AN 11/05
R15072 022700 000034 CMP #34,R0 ;USE ECO K011A-00005 CHECK WORD
R15076 001401 REC 05

```

```

R15100 104400 5251 HLT

;EXIT ROUTINE
R15102 012706 000000 651 MOV 0KPTR,SP ;SET KERNEL STACK PTR
R15106 012746 MOV (PC)+,=(0P) ;PUSH OLD PSW ONTO STACK
R15110 000000 751 ,WORD R ;CONTAINS SAVED PSW
R15112 010746 MOV PC,=(0P) ;PUSH CURRENT PC ONTO STACK
R15114 062716 000000 ADD 06,(0P) ;ADD OFFSET
R15120 000002 RTI
R15122 012706 000500 MOV 05TKPTR,SP ;SET STACK PTR
R15126 012737 000000 000004 MOV 0ERRVEC+2,0=ERRVEC
R15134 104400 SCOPE

;CHECK THAT ALL RESERVED INSTRUCTIONS TRAP (TO LOCATION 01)
R15136 012737 000002 001114 RESTRAP MOV #2,0=SCOPED ;LIMIT TO TWO ITERATIONS
R15144 010701 MOV PC,R1 ;SET SCOPE POINTER
R15146 012732 015206 MOV 055,R2 ;GET ADDRESS OR RESERVED INSTRUCTION TABLE
R15152 063702 021004 ADD 055,R2
R15156 122737 000004 014376 CHMB #4,00OPT,CP ;ADJUST TABLE ADDRESS IF 11/05; 11/05
R15164 003402 BLE 115 ;35=11749, 11/49 TABLE, 65=11/05
R15166 062702 000036 ADD 005=05,R2 ;11/20 TABLE
R15172 132737 000000 014377 1151 BITB #40,00OPT,CP=1 ;CHECK IF 11/45 FLOATING POINT IS AVAIL.
R15200 001402 BEQ +6 ;BRANCH IF NOT AVAILABLE
R15202 005067 000110 CLR 505 ;SET TABLE TERMINATOR AT GROUP 7
R15206 012737 019244 000010 MOV 045,0=RESVEC ;SET RESERVED INSTRUCTION TRAP
R15214 063737 001004 000010 ADD 00FACTOR,0=RESVEC
R15222 012203 151 MOV (R2)+,R3 ;GET FIRST RESERVED INSTRUCTION
R15224 001454 BEQ 75 ;0 TERMINATES THE TABLE
R15226 012204 MOV (R2)+,R4 ;GET LAST RESERVED INSTRUCTION IN GROUP
R15230 010317 251 MOV R3,(PC) ;EXECUTE RESERVED INSTRUCTION
R15232 000000 ,WORD R ;CONTAINS RESERVED INSTRUCTION
R15234 104400 HLT ;[ERROR] INSTRUCTION IN R3
R15236 104400 HLT ;[05] ABOVE FAILED TO CAUSE A
R15240 104400 HLT ;RESERVED INSTRUCTION TRAP
R15242 000405 BR 415
R15244 012716 019250 451 MOV 0415,(0P) ;ADJUST RETURN PC
R15250 063716 001004 ADD 00FACTOR,(0P) ;TO RETURN TO 415
R15254 000002 RTI ;RETURN TO 415
R15256 020304 4151 CMP R3,R4 ;HAS GROUP OF RESERVED INSTRUCTIONS
R15260 001760 BEQ 15 ;BEEN EXECUTED
R15262 000203 INC R3 ;INCREMENT THIS RESERVED INSTRUCTION
R15264 000761 BR 25 ;TO NEXT ONE AND EXECUTE

;TABLE OF 11/40,11/45 RESERVED INSTRUCTIONS (0 TERMINATED THE TABLE)
R15266 000007 501 7 ;GROUP 1
R15270 000077 77 " "
R15272 000210 210 " ;GROUP 2
R15274 000227 227 " "
R15276 007000 7000 " ;GROUP 3
R15300 007777 7777 " "
R15302 070000 70000 " ;GROUP 4
R15304 076777 76777 " "
R15306 104400 104400 " ;GROUP 5
R15310 104477 104477 " "
R15312 104700 104700 " ;GROUP 6

```

19314	187777		187777		"	
19316	178888	0001	178888		GROUP 7	FLOATING POINT
19320	177777		177777		"	INSTRUCTIONS
19322	000000		0		IS TERMINATED THE TABLE	
19324	000000				TABLE OF 11/09, 11/20 RESERVED	INSTRUCTIONS (0 TERMINATES THE TABLE)
19326	000077		0		GROUP 1	
19330	000210		77		"	
19332	000237		210		GROUP 2	
19334	006400		837		"	
19336	007777		6400		GROUP 3	
19340	070000		7777		" 3	
19342	077777		70000		GROUP 4	
19344	106400		77777		"	
19346	107777		106400		GROUP 5	
19350	170000		107777		"	
19352	177777		170000		GROUP 6	
19354	000000		177777		"	
19356	012737	000012 000010 751	F		IS TERMINATES THE TABLE	
19360	104000		MOV	#RESVEC+2,0#RESVEC	RESTORE RESERVED TRAP TO HALT AT 12	
			SCOPE			
					ICHECK THAT ALL BITS IN THE PROCESSOR STATUS WORD (PSW) CAN BE SET AND	
					ICLEARED,	
19366	213767	177776 000192	MOV	#PSW,35	ISAVE STATUS	
19374	205037	177776	CLR	#PSW	ICLEAR MODE BITS IN PSW	
19400	005046		CLR	=(SP)	ROUTINE TO CLEAR	
19402	210746		MOV	PC,=(SP)	STATUS WORD (PSW)	
19404	262716	000000	ADD	#0,(SP)		
19410	000072		RTI		ICLEAR PSW & EXECUTE FOLLOWING INST	
19412	213746	000010	MOV	#RTBITVEC+2,=(SP)		
19416	212704	177776	MOV	#PSW,R4	LOAD ADDRESS OF PSW INTO R4	
19422	000290		CLN			
19424	205714		TST	(R4)	ICHECK THAT PSW WAS CLEARED	
19426	201401		REQ	,#4		
19430	104400		HLT		(ERROR) PSW FAILED TO CLEAR	
19432	113700	014376	MOV	#PCPT,CR,R0	GET CP TYPE	
19436	016000	017020	MOV	PSWBIT(0),R0	GET BIT MASK FOR TEST #WHOSE BITS IN	
					(THE PSW WHICH CAN BE SET/CLEARED,	
					ICHECK IF MEM MGMT IS AVAILABLE	
19442	005737	014376	TST	#PCPT,CP	BRANCH IF NOT AVAILABLE	
19446	100002		BPL	105	ICHECK IF MEM MGMT IS AVAILABLE	
19450	052700	170000	RIS	#170000,R0	SET BITS 15-12 IF MEM MGMT	
19454	012702	000001	MOV	#1,R2	R2 = TEST BIT	
19460	030200		BIT	R2,R0	ICHECK IF BIT CAN BE SET/CLEARED	
19462	201423		BEO	25		
19464	005037	000010	CLR	#RTBITVEC+2		
19470	030227	000020	BIT	R2,#20	ICHECK IF TEST WILL SET IT BIT	
19474	201403		BEO	205		
19476	212737	000002 000010	MOV	#RT1,#RTBITVEC+2	SET RT1 INTO RETURN	
19504	205014		CLR	(R4)	ICLEAR PSW	
19506	250214		RIS	R2,(R4)	SET R2 INTO PSW	
19510	011403		MOV	(R4),R3	GET BIT	
19512	020203		CHP	R2,R3	ICHECK THAT BIT WAS SET IN PSW	

19514	201401		BEO	,#4	(ERROR) BIT IN R2 FAILED TO SET IN PSW	
19516	104400		HLT		ICLEAR 2 BIT	
19520	200244		CLC		ICLEAR BIT IN PSW	
19522	240214		BIC	R2,(R4)	GET PSW RESULT	
19524	211403		MOV	(R4),R3	BRANCH IF BIC ABOVE CLEARED BIT IN PSW	
19526	001421		BEO	25	(ERROR) BIT IN R2 FAILED TO CLEAR IN PSW	
19530	104400		HLT			
19532	206302		ASL	R2	SHIFT TEST BIT	
19534	105391		BCC	15	BRANCH IF ALL BITS NOT TESTED	
19536	005014		CLR	(R4)	ICLEAR STATUS	
19540	012637	000010	MOV	(SP)+,#RTBITVEC+2	RESTORE T BIT RETURN	
19544	212746		MOV	(PC)+,=(SP)	PUSH ORIGINAL STATUS ON STACK	
19546	000000		MOV	0	CONTAINS ORIGINAL PSW	
19550	010746		MOV	PC,=(SP)	SET RETURN PC	
19552	262716	000000	ADD	#0,(SP)		
19556	000002		RTI		RETURN	
19560	104000		SCOPE			
19562	213704	177776	MOV	#PSW,R4	ISAVE PSW IN R4	
19566	010440		MOV	R4,=(SP)	PUSH R4 ONTO STACK	
19570	112716	000000	MOV	#300,(SP)	SET PRIORITY LEVEL 6 AND	
19574	210746		MOV	PC,=(SP)	ICLEAR IT BIT AND EXECUTE	
19576	262716	000000	ADD	#0,(SP)	INSTRUCTION FOLLOWING RTI	
19602	000002		RTI			
					ICHECK THAT ALL BITS IN THE CURRENT STACK PTR CAN BE SET/CLEARED	
					CHRSPT:	
19604	010603		MOV	SP,R3	ISAVE STACK PTR	
19606	000297	000377	MOV	CCC		
19610	112700		MOV	#377,SP	SET STACK PTR = #1	
19614	206000		FOR	SP	ROTATE 0 BIT THROUGH ALL BIT	
19616	103776		BCS	15	BIT POSITIONS	
19620	005200		INC	SP	SHOULD INCREMENT SP TO 0	
19622	201403		BEO	25		
19624	210602		MOV	SP,R2	ISAVE ERROR STACK PTR	
19626	010300		MOV	R3,SP	SET STACK PTR FOR TRAP	
19630	104400		HLT		(ERROR)	
19632	010300		MOV	R3,SP	RESTORE ORIGINAL STACK PTR	
					ICHECK BYTE OPERATIONS USING THE STACK	
					SPCHK1:	
19634	010600		MOV	SP,R0	ISAVE STACK PTR	
19636	010003		MOV	R0,R3		
19640	005043		CLR	=(R3)		
19642	112746	177777	MOV	#1,=(SP)	((SP) = 377	
19646	222713	000377	CHP	#377,(R3)	ICHECK THAT ONLY EVEN BYTE WAS AFFECTED	
19650	001002		BNE	15		
19654	220300		CHP	R3,SP	ICHECK AUTO-DEC	
19656	001401		BEO	,#4		
19660	104400		HLT			
19662	109270		INCB	(SP)+		
19664	005723		TST	(R3)+	ICHECK RESULT	
19666	201002		BNE	25		
19670	020070		CHP	R0,SP	ICHECK AUTO-INC	


```

15972 001401     BEQ    ,04
15974 104400     HLT

15976 001443     COM    =(R3)           ;(R3)=I7777
15978 144613     BICB  =(SP),(R3)
15980 022713 177400     CMP    #177400,(R3) ;CHECK RESULT
15982 001802     BNE    35
15984 020603     CMP    SP,R3
15986 001401     BEQ    ,04
15988 104400     HLT

15992 132027 000377     BITB  (SP),#377
15994 001802     BNE    45
15996 020600     CMP    SP,R0
15998 001401     BEQ    ,04
16000 104400     HLT

16004 012746 000001     MOV    #1,(SP)
16006 060706 000002     ADD    #2,SP
16008 012702 177401     MOV    #177401,R2
16010 120244     CMPB  R2,#(SP)
16012 001004     BNE    55
16014 122602     CMPB  (SP),R2
16016 001802     BNE    55
16018 020000     CMP    R0,SP
16020 001401     BEQ    ,04
16022 104400     HLT
16024 010444     MOV    #4,(SP) ;RESTORE ORIGINAL PSW TO STACK
16026 010744     MOV    PC,#(SP)
16028 062716 000006     ADD    #6,(SP)
16030 200002     RTI
16032 104000     SCOPE

;CHECK THAT 'C' BIT SETS/CLEAR PROPERLY
16034 212727 177770     CBITI MOV    #177770,(PC)+ ;LOAD CONSTANT
16036 000000     15) ;WORD #
16038 010700     MOV    PC,R0 ;GET CURRENT PC
16040 162700 000004     SUB    #4,R0 ;POINT R0 TO 15 ABOVE
16042 003520     ADC    (R0)+ ;ADD 'C' BIT TO 15 ABOVE
16044 004340     ASL   -(R0) ;SHIFT 15
16046 102375     BVC   25 ;UNTIL 'C' BIT SETS
16048 022767 077770 177794     CMP    #077770,15 ;CHECK RESULT
16050 001401     BEQ    ,04
16052 104000     HLT ;ERROR: INCORRECT RESULT IN 15 ABOVE
;R0=ADDRESS OF DATA

;CHECK THAT CONDITION CODES ARE SET PROPERLY WHEN A NUMBER (CURRENT PC)
;AND THAT NUMBER #1 ARE COMPARED, AND VICE VERSA,
16054 210700     CMPI  MOV    PC,R0 ;GET CURRENT PC
16056 210002     MOV    R0,R2 ;SAVE IN R2
16058 209202     INC   R2 ;MAKE R2 = R0+1
16060 000277     BCC
16062 000291     +CLC:CLN ;CLEAR C & N BITS
16064 220002     CMP   R0,R2 ;COMPARE # WITH #+1

```

```

16066 103003     BCC   15 ;CARRY BIT SHOULD SET
16068 102402     BVS  15 ;V BIT SHOULD CLEAR
16070 001401     BEQ   15 ;Z BIT SHOULD CLEAR
16072 100401     BMI  ,04 ;N BIT SHOULD SET
16074 104400     HLT  ;ERROR: COMPARE # WITH #+1 FAILED TO
;SET CONDITION CODES IN PSW CORRECTLY

16078 000277     BCC
16080 120200     CMPB R2,R0 ;SET CONDITION CODES IN PSW
16082 103403     BCS  25 ;COMPARE #+1 WITH #
16084 102402     BVS  25 ;C BIT SHOULD CLEAR
16086 001401     BEQ   25 ;V BIT SHOULD CLEAR
16088 100001     BPL  ,04 ;Z BIT SHOULD CLEAR
16090 104400     HLT  ;N BIT SHOULD CLEAR
;ERROR: COMPARE #+1 WITH # FAILED TO SET
;CONDITION CODES IN PSW CORRECTLY

;24 NOP (240) INSTRUCTIONS FOLLOW; THESE NOPS MAY
;BE CHANGED TO TEST CODE IF THE NEED ARISES, THE TEST CODE SHOULD
;BE POSITION INDEPENDENT AND SHOULD RUN WHEN RELOCATED BY THE PROGRAM;

16100 000240     NOP
16102 000240     NOP
16104 000240     NOP
16106 000240     NOP
16108 000240     NOP
16110 000240     NOP
16112 000240     NOP
16114 000240     NOP
16116 000240     NOP
16118 000240     NOP
16120 000240     NOP
16122 000240     NOP
16124 000240     NOP
16126 000240     NOP
16128 000240     NOP
16130 000240     NOP
16132 000240     NOP
16134 000240     NOP
16136 000240     NOP
16138 000240     NOP
16140 000240     NOP
16142 000240     NOP
16144 000240     NOP
16146 000240     NOP
16148 000240     NOP
16150 000240     NOP
16152 000240     NOP
16154 000240     NOP
16156 000240     NOP
16158 104000     SCOPE

16160 010702     MOV    PC,R2
16162 062702 000012     ADD    #12,R2
16164 012707 001132     MOV    #RELOC,PC ;OO RELOCATE PROGRAM CODE
16166 000240     NOP ;PROGRAM RETURNS HERE+

1383333333333333 LAST ADDRESS OF CODE TO BE RELOCATED 3333333333

```

16202	16781	000100	177564	MOV	PC,R1		
16204	032737			BIT	#100,#0TPS	ICHECK IF TTY IS READY	
16212	001374			BNE	,#0		
16214	012737	01627C	000004	MOV	#31,#0TPVEC	ISSET TTY INTERRUPT VECTOR	
16222	012737	000200	000000	MOV	#200,#0TPVEC*2	IPRIORITY LEVEL 4 ON INTERRUPT	
16230	012767	014300	000004	MOV	#NULLS,#00	IADDRESS OF MESSAGE TO BE TYPED	
16236	117737	000000	177566	MOV	#MSG,#0TPS	ITYPE FIRST CHARACTER OF MESSAGE	
16244	109737	177564		TSYB	#0TPS		
16250	100379			BPL	,#4		
16252	006237	177564		LSR	#0TPS	ISSET IE BIT IN TTY CSR REG	
16256	000001			WAIT		IWAIT FOR FIRST INTERRUPT	
16260	000414			RR	#W11		
16262	006337	177564	25I	ASL	#0TPS	ICLEAR IE BIT	
16266	000002			RTI			
16270	117737	000000	177566	35I	MOV	#MSG,#0TPS	ITYPE CHARACTER
16276	001771			BEQ	25	IBRANCH IF TERMINATOR	
16300	009227			INC	(PC)+	ISSET MSG TO NEXT CHAR ADDRESS	
16302	000000			,WORD	0	ICONTAINS ADDRESS OF CHAR TO BE TYPED	
16304	000002			RTI		IRETURN	
16306	020019	000019		NULLSI	,ASCII	<13><40><19>	
				,EVEN			
						ROUTINE TO TURN ON KW11-L LINE CLOCK IF AVAILABLE	
16312	012737	000002	000000	KW11	MOV	#RT1,#0ERRVEC*2	ISSET UP DIRECT RT1 ON TRAP
16320	012737	014494	000100	MOV	#45,#0LKVEC	ILOAD INTERRUPT VECTOR	
16326	012737	000300	000102	MOV	#300,#0LKVEC*2	ISSET PRIORITY LEVEL 6 ON INT.	
16334	000202			SEV		ISSET TIME OUT INDICATOR	
16336	052737	000100	177566	010	#100,#0PKS	ISSET INTERRUPT ENABLE	
16344	102446			015	05	ISKIP PRIORITY ARBITRATION TEST	
						IBELOW IF NO KW11-L	
						ROUTINE TO CHECK PRIORITY ARBITRATION LOGIC	
						IBELOW TEST WILL INHIBIT INTERRUPTS ON LEVEL 6 AND ABOVE (LOCKING	
						OUT THE LINE CLOCK) AND THEN SET UP THE TTY TO INTERRUPT; NEXT THE	
						IPRIORITY LEVEL WILL BE SET TO 0 ALLOWING INTERRUPTS IN WHICH CASE	
						ITHE LINE CLOCK (AT LEVEL 6) SHOULD INTERRUPT BEFORE THE TTY (AT LEVEL 4);	
16346	132737	000020	177776	BIT	#20,#0PSW	ICHECK IF IE BIT IS SET	
16354	001042			BNE	05	IDO NOT DO TEST IF SET	
16356	112737	000300	177776	MOV	#300,#0PSW	ISSET PRIORITY LEVEL 6	
16364	013727	000004		MOV	#0TPVEC,(PC)+	ISAVE TTY INTERRUPT VECTOR	
16370	000000			,WORD	?	ICONTAINS CURRENT TTY VECTOR	
16372	109737	177564	15I	TSYB	#0TPS	ICHECK IF READY	
16376	102379			BPL	,#4	IWAIT FOR TTY TO BECOME READY	
16400	012737	016424	000004	MOV	#25,#0TPVEC	ISSET NEW VECTOR	
16406	009227			INC	(PC)+	ISTALL WAITING FOR LINE CLOCK	
16410	000000			,WORD	0	ITO BE READY	
16412	012737	016430	000100	MOV	#35,#0LKVEC	ISSET LINE CLOCK VECTOR	
16420	109237	177776		CLRB	#0PSW	ISSET PRIORITY LEVEL 0	
16424	104400			WLY		IERROR! EITHER TTY INTERRUPTED	
16426	000419			OR	05	IEKIT YES	
16430	016737	177734	000004	35I	MOV	15,#0TPVEC	IRESTORE TTY VECTOR

16436	012737	016434	000100	MOV	#45,#0LKVEC	ISSET LINE CLOCK VECTOR
16444	109037	177776		CLRB	#0PSW	IRESTORE PRIORITY LEVEL 0
16446	012710	016462		MOV	#51,(SP)	ISSET RETURN ADDRESS TO 51 BELOW
16454	009207	162310	45I	INC	TICKS	INCREMENT TICK COUNT
16460	000002			RTI		IRETURN
16462	009037	000000	55I	CLR	#0ERRVEC*2	IRESTORE ERROR TRAP TO HALT AT 6
16466	000240			ENDI	NOP	
16470	009037	177776	END1	CLR	#0PSW	ICLEAR MODE BITS IN PSW
16474	009046			CLR	=(SP)	ICLEAR PSW
16476	012740	016504		MOV	#1,#=(SP)	
16482	000002			RTI		IGO TO NEXT INBT WITH #SW#0
16484	012706	000000		MOV	#KPTR,SP	ISSET KERNEL STACK PTR (NOT APPLICABLE
16490	032737	000100	177564	BIT	#120,#0TPS	IFOR 11/20,11/05 CH'S)
16496	001374			BNE	,#0	ICHECK IF OUTPUT DEVICE IS BUSY
16498	109737	177570		TSYB	#0SWR	IFIS AVAILABLE
16504	100020			BPL	15	IDELTE END OF PASS TYPE OUT IF SW#0
16506	016702	162246		MOV	ICNT,R2	IBRANCH IF SW7 IS DOWN
16512	004767	162572		JSR	PC,SPORMR	IGET PASS COUNT
16516	012702	001664		MOV	#0IGITS*2,R2	IGM TO FORMAT ROUTINE
16522	012703	001702		MOV	#PASSES,R3	IGET ASCII VALUES
16526	012223			MOV	(R2)+(R3)+	IAND MOVE THEM INTO MESSAGE
16530	012223			MOV	(R2)+(R3)+	
16532	012737	001072	014302	MOV	#PASCNT,#MSG	IPASS MESSAGE ADRS TO TELETYPE SERVICE
16536	052737	000100	177564	015	#100,#0TPS	ISSET IE BIT
16540	012737	000010	000024	MOV	#PDMV,#0PVEC	IENABLE POWER FAIL TRAP
16544	012737	000340	000026	MOV	#340,#0PVEC*2	IPRIORITY 7 ON POWER FAIL
16548	009207	162172		INC	ICNT	
16554	114700	175504		MOV	OPT,CP,00	IGET CP TYPE
16560	026067	017030	162100	CMP	PASTAB(00),ICNT	ICHECK IF END OF TEST
16566	001002			BNE	25	IBRANCH IF NOT AT END
16572	000107	000000		JMP	DONE	
16576	016702	162146	25I	MOV	ICNT,R2	IGET PASS COUNT
16582	006372			ASL	R2	
16588	040024	010710		BIC	CPASS(0),R2	ILIMIT PASS COUNT TO 0+6
16594	009037	000016		CLF	#016	ICLEAR ? BIT TRAP ADDRESS
16598	012737	000040	001122	MOV	#00,#SCOPE*2	ISSET ITERATION COUNT = 40
16604	016216	016770		MOV	PSWAB(2),(SP)	IPUSH NEXT PASS PSW ON STACK
16610	032716	000020		BIT	#2R,(SP)	IWILL IE BIT BE SET ON NEXT PASS?
16616	001400			BEQ	35	IBRANCH IF NOT
16622	012737	000002	001122	MOV	#2,#SCOPE*2	ISSET ITERATION COUNT = 2 FOR IE BIT
16628	016737	000000	000016	MOV	RT1,#016	ISSET IE BIT TRAP TO RETURN VIA 16
16634	012746	002400	35I	MOV	START2,=(SP)	IRESART PROGRAM AT START2
16640	000002			RTI	RTI	IRESART PROGRAM AT START2 WITH NEW PSW
						IFROM TABLE BELOW) NOTE THE RTI IS
						IOORGANIZED TO AN RTI IF NOT IN 11/09,11/20

R16716	189737	177904	TEST	DDTDB	WAIT FOR LAST CHARACTER TO BE PRINTED
R16722	188375		BPL	,04	
R16724	089827		CLR	(PC)0	
R16726	088888		WORD	0	
R16730	089267	177772	INC	18	DELAY WAITING FOR TELETYPE TO FINISH
R16734	081375		BNE	25	TYPING CHARACTER BEFORE ISSUING RESET
R16736	088888		RESET		
R16740	084747	162200	JSR	PG,,PRINT	PRINT MESSAGE BEGINING AT FOLLOWING ADDR
R16744	017871		ENOMSG		
R16746	013782	088842	MOV	0042,RE	CHECK ODP/ACT11 MONITOR ADDR
R16752	081484		BEQ	DONE1	
R16754	084712		LOGICAL JSR	PG,(R2)	GO TO ODP/ACT11 MONITOR VIA 48
R16756	088248		NOP		
R16758	088248		NOP		
R16762	088248		NOP		
R16764	080137	088498	DONE1 JMP	00START3	RESTART PROGRAM

THE BELOW TABLE REPRESENTS THE 'NEW' PSW SET BY THE PROGRAM ON SUCCESSIVE PASSES;
NOTE THE BELOW TABLE MAY BE MODIFIED TO CAUSE THE PROGRAM TO RUN UNDER USER DEFINED PARAMETERS BY PATCHING IN THE DESIRED PASS PARAMETER FOR EXAMPLE TO CAUSE THE PROGRAM TO RUN WITHOUT SETTING THE 'P' BIT IN ALL PASSES PATCH OUT THE 'P' BIT IN THE TABLE

R16770	088888		PSW(0)	088888	CALL II FAMILY CP'S
R16772	088888			088888	
R16774	140888			140888	11/48, 11/49 ONLY
R16776	140888			140888	
R17000	144888			144888	11/48 ONLY
R17002	144888			144888	
R17004	144888			144888	
R17006	144888			144888	

THE BELOW TABLE IS THE 'BIT HIGH' USED TO DETERMINE THE INDEX VALUE INDEXED TO SET THE 'NEW' PSW;

R17010	177774		CP(PSW)	177774	11/88
R17012	177774			177774	11/98
R17014	177778			177778	11/48
R17016	177768			177768	11/49

THE BELOW TABLE REPRESENTS THOSE BITS IN THE CP WHICH CAN BE SET/CLEARED EXCLUDING THE REGISTER SET BIT IN THE 11/49.

R17020	088377		PSW(1)	088377	11/99
R17022	088377			088377	11/98
R17024	088357			088357	11/48
R17026	178357			178357	11/49 (RESET BIT IS CHECKED ELSEWHERE)

THE BELOW TABLE CONTAINS THE # OF PASSES REQUIRED TO COMPLETE TEST

R17030	088882		PAS(0)	WORD 2	11/98
R17032	088882			WORD 2	11/98
R17034	088874			WORD 4	11/48
R17036	088818			WORD 18	11/49

R17040	085015	047914	020127	MS(1)	,ASCIZ <19><12>'LOW LIMITY'
R17046	044514	044515	037524		

R17054	088				
R17055	110	043911	020118	MS(2)	,ASCIZ 'HIGH LIMITY'
R17062	044514	044515	037524		
R17070	088				
R17071	015	042012	050532	ENOMSG)	,ASCIZ <19><12>'D20K0C DONE'
R17076	041313	042048	047117		
R17104	088105				
	088821				,END

Table listing symbols and their addresses for page 73, including entries like AC0, AC1, ADCB0, ADCB1, etc.

Table listing symbols and their addresses for page 73, including entries like ACR, ADCB2, ADCB3, etc.

Table listing symbols and their addresses for page 73, including entries like ICS, I0CB0, I0CB1, etc.

Table listing symbols and their addresses for page 74, including entries like NEG1, NEG2, NULL, OVFLW, etc.

Table listing symbols and their addresses for page 74, including entries like NEG3, NULLS, PARCSR, PASSES, etc.

Table listing symbols and their addresses for page 74, including entries like NEG6, PARRR, PAREPR, etc.

DRKCC, DRKCC/ROL-DRKCC
NON-TIME: 13 23 0 SECONDS
CORE DRKCC: 74