

IDENTIFICATION

PRODUCT CODE: MAINDEC-8E-D1FB-D
PRODUCT NAME: PDP-8E EXTENDED MEMORY ADDRESS
TEST (EA8E)
DATE: JUNE 14, 1971
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: VERNON FREY

COPYRIGHT © 1971
DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

The PDP-8E Extended Memory Address Test is designed to detect any location that cannot be uniquely addressed. This is performed by a series of four test routines which will test systems equipped with from 8K to 32K words of core memory. Automatic program relocation is provided in order to test all memory fields from each memory field. Teletype print-outs are provided for error identification, and the operator is given a degree of control over the program by various SR settings.

2. REQUIREMENTS

2.1 Equipment

A PDP-8E computer equipped with a minimum of 8K words of core memory.

2.2 Storage

The program occupies core locations 0000 to 3777.

2.3 Preliminary Programs

The Binary Loader must be in memory. Also, all diagnostics for a basic 4K PDP-8E must have been previously run successfully.

3. LOADING PROCEDURE

Load the program with the Binary Loader (BIN). The program may be loaded into any desired core stack by having BIN in that core stack.

4. OPERATING PROCEDURE

4.1 Program and Operator Action

- A. Set the SR to the INSTRUCTION FIELD and DATA FIELD of the stack which contains the program.
- B. Press key EXTD ADDR LOAD.
- C. Set the SR for desired starting address according to the following table.

ADDRESS	TEST EXECUTION
0200	Run all tests
0201	Run only test 1
0202	Run only test 2
0203	Run only test 3
0204	Run only test 4

- D. Press keys ADDR LOAD, CLEAR, and CONT. A setup SR message will be printed.

- E. Set the SR for desired operation according to the following table.

SWITCH	0 (down)	1 (up)
SR00	continue after error	halt after error
SR01	timeout errors	inhibit error timeouts
SR02	normal	TTY bell on error
SR03	relocate program	inhibit program relocation
SR04	normal	change stack limits
SR05	normal	halt after current test
SR06-08	starting stack limit (0-7)	
SR09-11	ending stack limit (0-7)	

Set to 0002

- F. Press key CONT.

4.2 Detailed SR Explanation

- SR00-02 SR02, if set, will ring the TTY bell once for each error. SR00 and SR01 have no effect with SR02 set.
- SR03 SR03 may be set or reset at any time and the program will act accordingly
- SR04 SR04 allows the operator to change the stack limits as defined by SR06-11.
- SR05 SR05 is normal halt for program
- SR06-08 These switches define the starting stack limit (normally 0).
- SR09-11 These switches define the ending stack limit (normally 7)

4.3 Example of Selecting Stacks for Test

Example 1: SR = 0007, 28K system

Stacks selected for testing are 6,5,4,3,2,1,0

Example 2: SR = 0004, 28K System

Stacks selected for testing are 4,3,2,1,0

Example 3: SR = 0022 28K System

Stacks selected for testing are 2
(No relocation will occur)

Example 4: SR = 0041 28K System

Stacks selected for testing are 6,5,4,1,0

NOTE 1: Stacks not in the system are automatically de-selected as is Example 1. Stack 7 is not present therefore not selected.

NOTE 2: A single stack can be selected for testing providing the program is not in that stack as in Example 3.

NOTE 3: Any stack or group of stacks can be by-passed as in Example 4. Stacks 2 and 3 are not selected, stack 7 is not present.

5. ERRORS

The contents of a given memory test location should always be equal to its address or the complement of its address. If it is not, a test error will result. A relocation error will occur if the relocation comparison check fails.

5.1 Test Error Typeouts

For the first error encountered a header will be typed out followed by the pertinent data. For all subsequent errors, only the pertinent data will be typed. The format is as follows:

PR LOC ADDR GOOD BAD TEST

PR LOC = the program address where the error JMS occurred.
(Includes Field)

ADDR = the address of the location in error. (Includes Field)

GOOD = what the data should be.

BAD = what the data is.

TEST = the test (1-4) running when the failure occurred.

5.2 Relocation Error Typeouts

All relocation errors are in the following format:

XXXXX RELOCATION ERROR AT LOCATION_YYYYY

XXXXX = the program address where the error JMS occurred, (Includes Field)

YYYYY = the address of the location in error (Includes Field)

NOTE: After each error print-out the program continues on with the next sequential memory location.

6. RESTRICTIONS

6.1 Starting Restrictions

The program may be restarted at any time from location $\$2000$ of the stack the program is presently in.

6.2 Operating Restrictions

None

7. EXECUTION TIME

The time to run all 4 tests in one core stack is approximately 1/2 second.

During program execution a 5 will be typed on the TTY approximately every 5 minutes of program run time. This allows the operator to determine approximate run time before a failure occurred.

8. SCOPE LOOPS

Two special scope loops have been provided in this program.

8.1 Scope Loop 1

This scope loop writes the value equal to the address specified by the SR into the address specified by the SR. It then loops doing a write-read.

8.1

continued

The address being looped on can be changed simply by changing the switch setting.

- A. Set the SR to the INSTRUCTION FIELD that the program is in and the DATA FIELD wanted to test.
- B. Press key EXTID ADDR LOAD.
- C. Set the SR equal to 3400.
- D. Press key ADDR LOAD.
- E. Set the SR equal to the address to test.
- F. Press keys CLEAR, and CONT.

8.2

Scope Loop 2

This scope loop is the same as Scope Loop 1 except that a group of addresses may be specified. The starting address specified must be less than the ending address specified.

- A. Set the SR to the INSTRUCTION FIELD that the program is in and the DATA FIELD wanted to test.
- B. Press key EXTID ADDR LOAD.
- C. Set the SR equal to 3600.
- D. Press key ADDR LOAD
- E. Set the SR equal to the first address of the group
- F. Press keys CLEAR and CONT. A halt will occur at address 3602.
- G. Set the SR equal to the last address of the group.
- H. Press key CONT.

NOTE:1: The address(s) specified will be looped until stopped by the operator with key HALT. No error checking is done. To resume normal operation, restart program at address 0200-0204 of the current instruction field.

9. PROGRAM DESCRIPTION

9.1 General

The PDP-8E Extended Memory Address Test is intended for use with a PDP-8E equipped with the extended memory option. A total of four tests are executed by the program. (See 9.2 thru 9.5). Each test writes a unique pattern into core memory and the checks for error. The patterns were chosen to aid the operator in the event of addressing errors.

The program automatically relocates itself to each memory field under test to ensure that all fields may be correctly referenced from any field. Fields not present in the system will automatically be de-selected from testing. (See 9.6)

Control of the program is given to the operator by means of the SR. The operator may halt after error, inhibit error printouts, substitute TTY BELL for error indication, halt after test, change field test limits, select all or any one of four tests, inhibit program relocation, and at any time restart the program at location 0200 thru 0204.

9.2 Test 1

Test 1 writes the value of each location into itself in the forward direction. Then each location is read and checked in the forward direction.

9.3 Test 2

Test 2 writes the complement value of each location into itself in the forward direction. Then each location is read and checked in the forward direction.

9.4 Test 3

Test 3 writes the value of each location into itself in the reverse direction. Then each location is read and checked in the reverse direction.

9.5 Test 4

Test 4 writes the complement value of each location into itself in the reverse direction. Then each location is read and checked in the reverse direction.

9.6 Program Relocation

Program relocation is governed by the status of SR bit 3 or by the fact that only one stack is selected for testing. With SR bit 3 down (0 position) program relocation occurs each time the test pattern and its complement have been completely tested in each selected stack. The program first relocates to the highest order 4K stack under test. The program keeps relocating to the next lower stack under test until it reaches the lowest order stack under test. The testing and relocation cycle is then repeated. The contents of the entire stack are relocated which enables any other information (RIM-BIN) to be carried with the program.

The program provides a degree of protection for itself by remembering all stacks where errors occur. When a faulty stack is next in sequence to contain the program, the program will skip the faulty stack and relocate to the first lower order stack which is error free. If all other selected stacks are faulty, program relocation will not take place.

During relocation a comparison check is made to insure no program loss.

For further understanding of how the tests are performed, refer to the listing.

/EXTENDED ADDRESS TEST FOR K8-E EXTENDED MEMORY (VER)
 /COPYRIGHT 1971, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS. 01754
 /PROGRAMMER, VERNON FREY

/SW0=1 HALT AFTER ERROR
 /SW1=1 INHIBIT ERROR TYPEOUT
 /SW2=1 BELL ON ERROR (USEFUL FOR MAINTENANCE)
 /SW3=1 INHIBIT PROGRAM RELOCATION
 /SW4=1 CHANGE STACK LIMITS
 /SW5=1 HALT AFTER CURRENT TEST
 /SW6-SW8 STARTING STACK LIMIT (0-7)
 /SW9-SW11 ENDING STACK LIMIT (0-7)

/PROGRAM STARTING ADDRESS
 /0200 RUN ALL TESTS
 /0201 RUN ONLY TEST 1
 /0202 RUN ONLY TEST 2
 /0203 RUN ONLY TEST 3
 /0204 RUN ONLY TEST 4

/NOT COMMANDS FOR THE K8-E EXTENDED MEMORY & INTERRUPT

6004	GTF=6004	/GET INTERRUPT FLAGS	/AC0 LINK
		/AC1 GREATER THAN FLAG	/AC2 INTERRUPT BUS
		/AC3 INTERRUPT INHIBIT FLIP-FLOP	/AC4 INTERRUPT ON
		/AC5 USER FLAG	/AC6-8 INSTRUCTION FIELD
		/AC9-11 DATA FIELD	/RESTORE INTERRUPT FLAGS
6005	RTF=6005	/AC0 LINK	/AC1 GREATER THAN FLAG
		/1 INTERRUPT INHIBIT FLIP-FLOP	/1 INTERRUPT ON
		/AC5 USER FLAG	/AC6-8 INSTRUCTION BUFFER
		/AC9-11 DATA FIELD	/CHANGE TO DATA FIELD 0
		/CHANGE TO DATA FIELD 1	/CHANGE TO DATA FIELD 2
		/CHANGE TO DATA FIELD 3	/CHANGE TO DATA FIELD 4
		/CHANGE TO DATA FIELD 5	/CHANGE TO DATA FIELD 6
6201	CDF0=6201		
6211	CDF1=6211		
6221	CDF2=6221		
6231	CDF3=6231		
6241	CDF4=6241		
6251	CDF5=6251		
6261	CDF6=6261		

```

6271 CDF7=6271 /CHANGE TO DATA FIELD 7
6202 C1F0=6202 /CHANGE TO INSTRUCTION FIELD 0
6212 C1F1=6212 /CHANGE TO INSTRUCTION FIELD 1
6222 C1F2=6222 /CHANGE TO INSTRUCTION FIELD 2
6232 C1F3=6232 /CHANGE TO INSTRUCTION FIELD 3
6242 C1F4=6242 /CHANGE TO INSTRUCTION FIELD 4
6252 C1F5=6252 /CHANGE TO INSTRUCTION FIELD 5
6262 C1F6=6262 /CHANGE TO INSTRUCTION FIELD 6
6272 C1F7=6272 /CHANGE TO INSTRUCTION FIELD 7
6203 CBF0=6203 /CHANGE TO DATA AND INSTRUCTION FIELD 0
6213 CBF1=6213 /CHANGE TO DATA AND INSTRUCTION FIELD 1
6223 CBF2=6223 /CHANGE TO DATA AND INSTRUCTION FIELD 2
6233 CBF3=6233 /CHANGE TO DATA AND INSTRUCTION FIELD 3
6243 CBF4=6243 /CHANGE TO DATA AND INSTRUCTION FIELD 4
6253 CBF5=6253 /CHANGE TO DATA AND INSTRUCTION FIELD 5
6263 CBF6=6263 /CHANGE TO DATA AND INSTRUCTION FIELD 6
6273 CBF7=6273 /CHANGE TO DATA AND INSTRUCTION FIELD 7
6204 CINT=6204 /CLEAR USER INTERRUPT (TIME SHARE)
6214 RIF=6214 /READ DATA FIELD INTO AC BITS 6-8
6224 RIF=6224 /READ INSTRUCTION FIELD INTO AC BITS 6-8
6234 RIB=6234 /READ INTERRUPT BUFFER
/AC6-8 INSTRUCTION FIELD IN USE BEFORE LAST
/ PROGRAM INTERRUPT.
/AC9-11 DATA FIELD IN USE BEFORE LAST
/ PROGRAM INTERRUPT.
/RESTORE MEMORY FIELD
/INSTRUCTION FIELD LOADED FROM SAVE FIELD 0-2
/ DATA FIELD LOADED FROM SAVE FIELD 3-5
/ SKIP ON USER INTERRUPT (TIME SHARE)
/ CLEAR USER FLAG (TIME SHARE)
/ SET USER FLAG (TIME SHARE)

```

*20 /
/CONSTANTS AND POINTERS

```

0020 SW0, 4000 /HALT AFTER ERROR
0021 SW1, 2000 /INHIBIT ERROR TYPEOUT
0022 SW2, 1000 /BELL ON ERROR
0023 SW3, 400 /INHIBIT PROGRAM RELOCATION
0024 SW4, 200 /CHANGE STACK LIMITS
0025 SW5, 100 /HALT AFTER CURRENT TEST
0026 SW6, 70 /STARTING STACK LIMIT (0-7)
0027 SW911, 7 /ENDING STACK LIMIT (0-7)
0030 STACK0, 0 /
0031 STACK1, 0 /
0032 STACK2, 0 /
0033 STACK3, 0 /
0034 STACK4, 0 /
0035 STACK5, 0 /
0036 STACK6, 0 /
0037 STACK7, 0 /
0040 STK0, 0 /

```

/STACKS CONTAIN 0 IF SELECTED FOR TESTING

0041	0000	STK1,	0	/	
0042	0000	STK2,	0	/	
0043	0000	STK3,	0	/0	IF RELOCATE
0044	0000	STK4,	0	/	
0045	0000	STK5,	0	/	
0046	0000	STK6,	0	/	
0047	0000	STK7,	0	/	
0050	0000	NORELO,	0	/	PROG RELOCATION CONTROL (0=INH)
0051	1706	KABOVE,	ABOVE	/	CONTROL UPPER STACKS NOT TESTED
0052	1725	KBELOW,	BELOW	/	CONTROL LOWER STACKS NOT TESTED
0053	0000	HEAD1,	0	/	ERROR HEADING CONTROL
0054	0000	INSAME,	0	/	PROG IN SEL STACK
0055	0000	LEGAL0,	0	/	LEGAL STACK SELECTION
0056	0000	RUNTS1,	0	/	/6003=ALL, 0001=1, 0002=2, 2000=3, 4000=4
0057	0000	TESTAD,	0	/	TEST ADDRESS COUNTER
0060	0000	KBINT,	0	/	HIGHEST ACTUAL STACK IN SYSTEM
0061	0000	SSL,	0	/	STARTING ACTUAL STACK LIMIT 00X0
0062	0000	ESL,	0	/	ENDING STACK LIMIT 000X
0063	0000	STKPIN,	0	/	STACK PROG IS IN 00X0
0064	0000	STKTST,	0	/	STACK SEL FOR TEST 00X0
0065	0000	BDATA,	0	/	BAD DATA
0066	0000	GDATA,	0	/	GOOD DATA
0067	0000	MOVE,	0	/	RELOCATION ADDRESS
0070	1736	KDOWN,	DOWN	/	CONTROL LOWER STACKS TESTED
0071	0000	TEMP,	0	/	INDIRECT ADDRESS TEMP STORAGE - CHEXM
0072	0000	COUNT,	0	/	CHECKERBOARD ERROR COUNTER
0073	0000	ERRLOC,	0	/	CODERR
0074	7777	M1,	-1	/	CODERR - TEST 3 & 4
0075	7776	M2,	-2	/	MESSAGE - LEGAL
0076	7775	M3,	-3	/	MESSAGE - 4 WORDS
0077	7774	M4,	-4	/	MESSAGE - 4 WORDS
0100	7744	M34,	-34	/	MESSAGE - ERRC - STACKS
0101	0007	K7,	7	/	CHEXN
0102	0010	K10,	10	/	CHEXN
0103	0020	K20,	20	/	CHEXN
0104	0030	K30,	30	/	CHEXN
0105	0040	K40,	40	/	CHEXN
0106	0050	K50,	50	/	CHEXN
0107	0060	K60,	60	/	CHEXN
0110	0070	K70,	70	/	CHEXN
0111	0077	K77,	77	/	SIXTY - MESSAGE
0112	0207	K207,	207	/	MESSAGE - CODERR
0113	0212	K212,	212	/	MESSAGE
0114	0215	K215,	215	/	MESSAGE
0115	0240	K240,	240	/	TOSEL
0116	0245	K245,	245	/	MESSAGE
0117	0260	K260,	"0	/	TOSEL - MAKE SEL 0-7
0120	0261	K261,	"1	/	TOSEL
0121	0262	K262,	"2	/	TOSEL
0122	0263	K263,	"3	/	TOSEL
0123	0264	K264,	"4	/	TOSEL
0124	0265	K265,	"5	/	TOSEL
0125	0266	K266,	"6	/	TOSEL
0126	0267	K267,	"7	/	TOSEL
0127	0340	K340,	340	/	MESSAGE

```

0130 0707 K707, 707 /SIXTY
0131 4060 K4060, 4060 /CODERR - ERR
0132 6000 K6000, 6000 /LINK IS A 0 - PROG FIELD TYPEOUT
0133 6003 K6003, 6003 /ALL TESTS
0134 6060 K6060, 6060 /SIXTY
0135 6100 K6100, 6100 /TEST 1
0136 6200 K6200, 6200 /TEST 2
0137 6300 K6300, 6300 /TEST 3
0140 6400 K6400, 6400 /TEST 4
0141 6201 K6201, 6201 /CDF 0
0142 6203 K6203, 6203 /CBF 0
0143 2042 XTYPE, TYPE /TYPEOUT AC ROUTINE POINTER
0144 2050 XMESAG, MESSAGE /TTY ROUTINE POINTER
0145 2000 XSIXTY, SIXTY /SIXTY ROUTINE POINTER
0146 2201 XCODER, CODERR /ERROR ROUTINE POINTER
0147 2200 XRETUR, RETURN /ERROR RETURN POINTER
0150 2242 XSTOP, STOP /STOP ROUTINE POINTER
0151 2241 XADDER, ADDE /ADDRESS OF ERROR TYPEOUT POINTER
0152 0000 FIVE, 0 /FIVE MINUTE TIMER
0153 0000 MIN5, 0 /FIVE MINUTE CONTROL
0154 7100 MIN50, -700 /ALL TESTS FIVE MIN
0155 3500 MIN51, -4300 /TEST 1 FIVE MIN
0156 3500 MIN52, -4300 /TEST 2 FIVE MIN
0157 6000 MIN53, -2000 /TEST 3 FIVE MIN
0160 6000 MIN54, -2000 /TEST 4 FIVE MIN

```

```

*0
0000
0001 0 JMP
0002 2
0003 3
*200
/
/KM8-E EXTENDED MEMORY ADDRESS TEST (EA8E)

```

```

0200 5777' EXTAD, JMP /ALL TESTS
0201 5776' JMP /TEST 1
0202 5775' JMP /TEST 2
0203 5774' JMP /TEST 3
0204 5773' JMP /TEST 4
0205 3056 DCA RUNTST /TEST CONTROL
0206 6002 IOF
0207 6224 RIF
0210 1141 TAD K6201
0211 3212 DCA ,+1
0212 6201 CDF 0
0213 4772' JMS TITLE /MAKE DATA FIELD=INST FIELD
0214 4771' JMS SETSW /TYPEOUT PROGRAM TITLE
0215 7240 STA /TYPEOUT TO SETUP SWITCHES
0216 3050 DCA NORELO /CLEAR INH RELOCATION
0217 3057 DCA TESTAD /CLEAR TEST ADDR COUNTER
0220 7240 STA /RESET ERROR HEADING
0221 3053 DCA HEAD1
0222 1153 TAD MIN5
0223 3152 DCA FIVE /SETUP COUNTER

```

0224	4770'	JMS	DOWN+2	/CLEAR STACK SELECTION CONTROLS
0225	7604	LAS	SW68	/STARTING STACK LIMIT
0226	0026	AND	SSL	/ENDING STACK LIMIT
0227	3061	DCA	SSL	/OBTAIN -SSL IN AC BITS 9-11
0230	7604	LAS	SW911	/SKIP IF SSL EQUALS ESL
0231	0027	AND	ESL	/CONTINUE CHECK
0232	3062	DCA	MSSL	/READ INSTRUCTION FIELD
0233	4767'	JMS	ESL	
0234	1062	TAD	ESL	
0235	7640	SZA	CLA	
0236	5262	JMP	CLAX	
0237	6224	RIF	CLAX	
0240	7041	CIA	CLAX	
0241	1061	TAD	SSL	
0242	7650	SNA	CLA	
0243	5766'	JMP	PINF	
0244	3050	DCA	NORELO	
0245	1365	TAD	(CHEXB	
0246	3764'	DCA	ABOVE-1	
0247	1062	TAD	ESL	
0250	1051	TAD	KABOVE	
0251	3071	DCA	TEMP	
0252	5471	JMP	TEMP	
0253	1363	TAD	(CHEXE	
0254	3762'	DCA	BELOW+1	
0255	1062	TAD	ESL	
0256	7041	CIA	KBELOW	
0257	1052	TAD	TEMP	
0260	3071	DCA	TEMP	
0261	5471	JMP	TEMP	
0262	4767'	JMS	MSSL	
0263	1062	TAD	ESL	
0264	7710	SPA	CLAX	
0265	5302	JMP	CHEXD	
0266	1361	TAD	(CHEXC1	
0267	3764'	DCA	ABOVE-1	
0270	1062	TAD	ESL	
0271	1051	TAD	KABOVE	
0272	3071	DCA	TEMP	
0273	5471	JMP	TEMP	
0274	1363	TAD	(CHEXE	
0275	3762'	DCA	BELOW+1	
0276	4767'	JMS	MSSL	
0277	1052	TAD	KBELOW	
0300	3071	DCA	TEMP	
0301	5471	JMP	TEMP	
0302	1360	TAD	(CHEXD1	
0303	3762'	DCA	BELOW+1	
0304	4767'	JMS	MSSL	
0305	1052	TAD	KBELOW	
0306	3071	DCA	TEMP	
0307	5471	JMP	TEMP	
0310	1363	TAD	(CHEXE	
0311	3770'	DCA	DOWN+2	
0312	1062	TAD	ESL	

0313	7041	CIA	
0314	1070	TAD	KDOWN
0315	3071	DCA	TEMP
0316	5471	JMP I	TEMP
0317	4757	JMS	HIGHST
0320	1060	TAD	KBINT
0321	1117	TAD	K260
0322	3060	DCA	KBINT
0323	4756	JMS	TSTSYS
0324	1355	TAD	(CHEXE2
0325	3764	DCA	ABOVE-1
0326	1060	TAD	KBINT
0327	0101	AND	K7
0330	1051	TAD	KABOVE
0331	3071	DCA	TEMP
0332	5471	JMP I	TEMP
0355	0400		
0356	3017		
0397	2737		
0360	0310		
0361	0274		
0362	1726		
0363	0317		
0364	1705		
0365	0253		
0366	2712		
0367	1742		
0370	1740		
0371	2621		
0372	2600		
0373	1624		
0374	1617		
0375	1612		
0376	1605		
0377	1600		
0400	0400		
0400	4777	CHEXE2, JMS	TOSEL
0401	4210	JMS	LEGAL
0402	1050	TAD	NORELO
0403	7050	SNA CLA	
0404	5276	JMP	CHEXM
0405	4776	JMS	CHKSW3
0406	5775	JMP	CHEX0
0407	5327	JMP	CHEXM

/RESTORE LOWER FIELDS TESTED
/FIND SYSTEMS HIGHEST STACK

/MAKE HIGHEST STACK 0-7 FOR TYPEOUT
/TYPEOUT # OF STACKS IN SYSTEM

/STORE RETURN ADDRESS

/INCREMENT UPPER STACKS NOT IN SYSTEM

/TYPEOUT STACK TEST SELECTION

/CHECK PROG RELO SW
/RELOCATE PROGRAM
/INHIBIT PROGRAM RELOCATION

/CHECK FOR LEGAL STACK SELECTION

0410	0000	LEGAL, 0	
0411	7300	CLA CLL	
0412	3054	DCA	INSAME
0413	1075	TAD	M2
0414	3055	DCA	LEGAL0
0415	3064	DCA	STKTST
			/CLEAR SAME CONTROL
			/SETUP LEGAL CONTROL

0416	1030	TAD	STACK0
0417	4263	JMS	LEGALA
0420	1102	TAD	K10
0421	3064	DCA	STKTST
0422	1031	TAD	STACK1
0423	4263	JMS	LEGALA
0424	1103	TAD	K20
0425	3064	DCA	STKTST
0426	1032	TAD	STACK2
0427	4263	JMS	LEGALA
0430	1104	TAD	K30
0431	3064	DCA	STKTST
0432	1033	TAD	STACK3
0433	4263	JMS	LEGALA
0434	1105	TAD	K40
0435	3064	DCA	STKTST
0436	1034	TAD	STACK4
0437	4263	JMS	LEGALA
0440	1106	TAD	K50
0441	3064	DCA	STKTST
0442	1035	TAD	STACK5
0443	4263	JMS	LEGALA
0444	1107	TAD	K60
0445	3064	DCA	STKTST
0446	1036	TAD	STACK6
0447	4263	JMS	LEGALA
0450	1110	TAD	K70
0451	3064	DCA	STKTST
0452	1037	TAD	STACK7
0453	4263	JMS	LEGALA
0454	2055	ISZ	LEGAL0
0455	9774	JMP	NOSTK
0456	1054	TAD	INSAME
0457	7640	SEA	CLA
0460	9773	JMP	PINF
0461	3050	DCA	NORELO
0462	5610	JMP	LEGAL

/NO STACK SELECTION

/PROG IN SELECTED FIELD
/ONLY 1 STACK SELECTED

/LEGAL STACK SELECTION SUBROUTINE

0463	0000	LEGALA, 0
0464	7640	SEA
0465	5663	JMP I
0466	2055	ISZ
0467	7410	SKP
0470	5610	JMP I
0471	6224	RIF
0472	3063	DCA
0473	4772	JMS
0474	2054	ISZ
0475	5663	JMP I

/NOT SELECTED

/PROG IN SEL STACK
/YES

/NO PROGRAM RELOCATION AND TEST ONLY 1 STACK

```

0476 6224 CHEXM, RIF
0477 3063 DCA
0500 1371 TAD
0501 3017 DCA
0502 3071 DCA
0503 1417 TAD I
0504 7650 SNA CLA
0505 5310 JMP
0506 2071 ISZ
0507 5303 JMP
0510 1071 TAD
0511 7104 CLL RAL
0512 7006 RTL
0513 3064 DCA
0514 4770 JMS
0515 4767 JMS
0516 7604 LAS
0517 0025 AND
0520 7640 SZA CLA
0521 7402 HLT
0522 7604 LAS
0523 0024 AND
0524 7640 SZA CLA
0525 5766 JMP
0526 5315 JMP

```

```

/STACK PROGRAM IS IN
/FIND STACK SEL FOR TEST
/STACK SEL FOR TEST
/TYPEOUT NO RELOCATION
/TEST THE SELECTED STACK
/NO PROGRAM RELOCATION BUT TEST ALL SELECTED STACKS
/CHANGE STACK LIMITS?
/YES
/NO
/CHANGE STACK LIMITS?
/YES
/NO
/CHANGE STACK LIMITS?
/YES
/NO
/CONTINUE

```

```

0527 4770 CHEXN, JMS
0530 4765 CHEXN0, JMS
0531 7604 LAS
0532 0025 AND
0533 7640 SZA CLA
0534 7402 HLT
0535 7604 LAS
0536 0024 AND
0537 7640 SZA CLA
0540 5766 JMP
0541 4776 JMP
0542 5775 JMP
0543 5330 JMP
0565 0600
0566 0214
0567 1200
0570 2636
0571 0027
0572 1631
0573 2712
0574 2732
0575 1000
0576 1640
0577 3074

```

```

/NO PROGRAM RELOCATION BUT TEST ALL SELECTED STACKS
/TYPEOUT NO RELOCATION
/TEST SEL'D STACKS
/NO PROGRAM RELOCATION BUT TEST ALL SELECTED STACKS
/CHANGE STACK LIMITS?
/YES
/NO
/CHANGE STACK LIMITS?
/YES
/NO
/CONTINUE

```

0600 PAGE

/TEST STACKS CONTROL

```

0600 0000 /CHEXN1, 0
0601 7200 CLA
0602 6224 RIF
0603 3063 DCA
0604 1037 TAD
0605 7640 SZA CLA
0606 5222 JMP
0607 1110 TAD
0610 3064 DCA
0611 3072 DCA
0612 4777 JMS
0613 5222 JMP
0614 4776 JMS
0615 1072 TAD
0616 7640 SZA CLA
0617 2047 ISE
0620 7410 SKP
0621 5217 JMP
0622 1036 CHEXN2, TAD
0623 7640 SZA CLA
0624 5240 JMP
0625 1107 TAD
0626 3064 DCA
0627 3072 DCA
0630 4777 JMS
0631 5240 JMP
0632 4776 JMS
0633 1072 TAD
0634 7640 SZA CLA
0635 2046 ISE
0636 7410 SKP
0637 5235 JMP
0640 1035 CHEXN3, TAD
0641 7640 SZA CLA
0642 5256 JMP
0643 1106 TAD
0644 3064 DCA
0645 3072 DCA
0646 4777 JMS
0647 5256 JMP
0650 4776 JMS
0651 1072 TAD
0652 7640 SZA CLA
0653 2045 ISE
0654 7410 SKP
0655 5253 JMP
0656 1034 CHEXN4, TAD
0657 7640 SZA CLA
0660 5274 JMP
0661 1105 TAD
0662 3064 DCA

```

/STACK PROGRAM IS IN
/STACK SEL FOR TEST
/PROG IN SEL STACK?
/YES
/NO - TEST THE SEL STACK

/STACK SEL FOR TEST
/PROG IN SEL STACK?
/YES
/NO - TEST THE SEL STACK

/STACK SEL FOR TEST
/PROG IN SEL STACK?
/YES
/NO - TEST THE SEL STACK

/STACK SEL FOR TEST
/PROG IN SEL STACK?
/YES
/NO - TEST THE SEL STACK

0663 3072 DCA COUNT
 0664 4777 JMS SAME
 0665 5274 JMP CHEXN5
 0666 4776 JMS TEST
 0667 1072 TAD COUNT
 0670 7640 SZA CLA
 0671 2044 ISZ STK4
 0672 7410 SKP
 0673 5271 JMP
 0674 1033 TAD STACK3
 0675 7640 SZA CLA
 0676 5312 JMP CHEXN6
 0677 1104 TAD K30
 0700 3064 DCA STKTST
 0701 3072 DCA COUNT
 0702 4777 JMS SAME
 0703 5312 JMP CHEXN6
 0704 4776 JMS TEST
 0705 1072 TAD COUNT
 0706 7640 SZA CLA
 0707 2043 ISZ STK3
 0710 7410 SKP
 0711 5307 JMP
 0712 1032 TAD STACK2
 0713 7640 SZA CLA
 0714 5330 JMP CHEXN7
 0715 1103 TAD K20
 0716 3064 DCA STKTST
 0717 3072 DCA COUNT
 0720 4777 JMS SAME
 0721 5330 JMP CHEXN7
 0722 4776 JMS TEST
 0723 1072 TAD COUNT
 0724 7640 SZA CLA
 0725 2042 ISZ STK2
 0726 7410 SKP
 0727 5325 JMP
 0730 1031 TAD STACK1
 0731 7640 SZA CLA
 0732 5346 JMP CHEXN8
 0733 1102 TAD K10
 0734 3064 DCA STKTST
 0735 3072 DCA COUNT
 0736 4777 JMS SAME
 0737 5346 JMP CHEXN8
 0740 4776 JMS TEST
 0741 1072 TAD COUNT
 0742 7640 SZA CLA
 0743 2041 ISZ STK1
 0744 7410 SKP
 0745 5343 JMP
 0746 1030 TAD STACK0
 0747 7640 SZA CLA
 0750 5361 JMP CHEXN9
 0751 3064 DCA STKTST

CHEXN5,

CHEXN6,

CHEXN7,

CHEXN8,

/STACK SEL FOR TEST
 /PROG IN SEL STACK?
 /YES
 /NO - TEST THE SEL STACK

/STACK SEL FOR TEST
 /PROG IN SEL STACK?
 /YES
 /NO - TEST THE SEL STACK

/STACK SEL FOR TEST
 /PROG IN SEL STACK?
 /YES
 /NO - TEST THE SEL STACK

/STACK SEL FOR TEST

```

0752 3072 DCA COUNT
0753 4777 JMS SAME
0754 5361 JMP CHEXN9
0755 4776 JMS TEST
0756 1072 TAD COUNT
0757 7640 SZA CLA
0760 2040 ISZ STK0
0761 5600 JMP I CHEXN1
0762 5360 JMP .-2

```

```

0776 1200
0777 1631
0778 1000

```

PAGE

```

/CHECK ALL SELECTED STACKS FROM EACH SELECTED STACK
/
CHEX0, 1000 4777 JMS PREL /TYPEOUT RELOCATION
1001 4776 JMS RESTK /RESTORE STK(S)
1002 4775 JMS CHEXN1 /TEST FROM PRESENT STACK
1003 4774 JMS CHKSW3
1004 7410 SKP CHEXN
1005 5773 JMP CHEXN /STACK PROGRAM IS IN
1006 6224 RIF
1007 3063 DCA STKPIN
1010 1047 TAD STK7
1011 7640 SZA CLA CHEX00
1012 5224 JMP K70
1013 1110 TAD STKTST
1014 3064 DCA STKTST
1015 4772 JMS SAME
1016 7410 SKP RELO
1017 4771 JMS CHEXN1
1020 4775 JMS CHEXN1
1021 4774 JMS CHKSW3
1022 7410 SKP
1023 5773 JMP CHEXN
1024 1046 TAD STK6
1025 7640 SZA CLA CHEX01
1026 5240 JMP K60
1027 1107 TAD STKTST
1030 3064 DCA STKTST
1031 4772 JMS SAME
1032 7410 SKP
1033 4771 JMS RELO
1034 4775 JMS CHEXN1
1035 4774 JMS CHKSW3
1036 7410 SKP
1037 5773 JMP CHEXN
1040 1045 TAD STK5
1041 7640 SZA CLA CHEX02
1042 5254 JMP K50
1043 1106 TAD STKTST
1044 3064 DCA

```

```

/STACK SEL FOR MOVE TO
/PROG IN MOVE STACK?
/YES
/NO - RELOCATE PROGRAM
/TEST ALL SEL STACKS

```

1045	4772'	JMS	SAME
1046	7410	SKP	
1047	4771'	JMS	RELO
1050	4775'	JMS	CHEXN1
1051	4774'	JMS	CHKSW3
1052	7410	SKP	
1053	5773'	JMP	CHEXN
1054	1044	TAD	STK4
1055	7640	JMP	CHEX03
1056	5270	JMP	SEA CLA
1057	1105	TAD	K40
1060	3064	DCA	STKTST
1061	4772'	JMS	SAME
1062	7410	SKP	
1063	4771'	JMS	RELO
1064	4775'	JMS	CHEXN1
1065	4774'	JMS	CHKSW3
1066	7410	SKP	
1067	5773'	JMP	CHEXN
1070	1043	TAD	STK3
1071	7640	JMP	SEA CLA
1072	5304	JMP	CHEX04
1073	1104	TAD	K30
1074	3064	DCA	STKTST
1075	4772'	JMS	SAME
1076	7410	SKP	
1077	4771'	JMS	RELO
1100	4775'	JMS	CHEXN1
1101	4774'	JMS	CHKSW3
1102	7410	SKP	
1103	5773'	JMP	CHEXN
1104	1042	TAD	STK2
1105	7640	JMP	SEA CLA
1106	5320	JMP	CHEX05
1107	1103	TAD	K20
1110	3064	DCA	STKTST
1111	4772'	JMS	SAME
1112	7410	SKP	
1113	4771'	JMS	RELO
1114	4775'	JMS	CHEXN1
1115	4774'	JMS	CHKSW3
1116	7410	SKP	
1117	5773'	JMP	CHEXN
1120	1041	TAD	STK1
1121	7640	JMP	SEA CLA
1122	5334	JMP	CHEX06
1123	1102	TAD	K10
1124	3064	DCA	STKTST
1125	4772'	JMS	SAME
1126	7410	SKP	
1127	4771'	JMS	RELO
1130	4775'	JMS	CHEXN1
1131	4774'	JMS	CHKSW3
1132	7410	SKP	
1133	5773'	JMP	CHEXN

1134 1040
 1135 7640
 1136 5344
 1137 3064
 1140 4772
 1141 7410
 1142 4771
 1143 4775
 1144 7604
 1145 0025
 1146 7640
 1147 7402
 1150 7604
 1151 0024
 1152 7640
 1153 9770
 1154 4774
 1155 5206
 1156 9773
 1170 0214
 1171 1646
 1172 1631
 1173 0527
 1174 1640
 1175 0600
 1176 3051
 1177 2672
 1200 1200

CHEX06, TAD STK0
 SEA CLA CHEX07
 JMP STKTST
 DCA SAME
 JMS RELO
 JMS CHEXN1
 JMS CHEXN1
 LAS AND SWS
 CHEX07, AND SWS
 SEA CLA
 HLT
 LAS
 AND SW4
 SEA CLA CHEXA
 JMP CHKSW3
 JMS CHEX0A
 JMP CHEXN
 JMP

/HALT AFTER TEST

/CHANGE STACK LIMITS?

/YES

/NO

/RELOCATE THE PROGRAM

/INHIBIT PROGRAM RELOCATION

PAGE

/RUN THE SELECTED TEST(S) ON THE SELECTED FIELD (STKTST)

TEST, 0
 CLA CLL STKTST /UPDATE CDF TEST DATA FIELDS
 TAD K6201
 DCA TDF1
 TAD TDF1
 DCA TDF2
 TAD TDF2
 DCA TDF3
 TAD TDF3
 DCA TDF4
 TAD TDF4
 DCA COUNT /CLEAR ERROR COUNT
 TAD RUNTST
 RAR SEL CLA
 SEL CLA TEST1 /EXECUTE TEST 1
 JMS RUNTST
 TAD
 RTR
 SEL CLA TEST2 /EXECUTE TEST 2
 JMS RUNTST
 TAD

1225	7006	RTL	
1226	7630	SZL	CLA
1227	4773	JMS	TEST3
1230	1056	TAD	RUNTST
1231	7004	RAL	
1232	7630	SZL	CLA
1233	4772	JMS	TEST4
1234	7604	LAS	
1235	0024	AND	SW4
1236	7640	SZA	CLA
1237	5771	JMP	CHEXA
1240	2192	ISZ	FIVE
1241	5600	JMP	I
1242	1153	TAD	MINS
1243	3152	DCA	FIVE
1244	4544	JMS	I
1245	4543		XMESAG
1246	6500		
1247	5600	JMP	I
			TEST

/EXECUTE TEST 3
/EXECUTE TEST 4
/CHANGE STACK LIMITS?
/YES
/NOT 5 MINUTES YET
/RESTORE TIMER

/TEST 1
/WRITE THE VALUE OF EACH LOCATION INTO ITSELF AND CHECK

1250	0000	TEST1,	0	TESTAD	/CLEAR TEST ADDRESS COUNTER
1251	3057	DCA			/CHANGE TO TEST DATA FIELD
1252	6201	CDF0			/WRITE MEMORY
1253	1057	TAD			/4096 TIMES
1254	3457	DCA	I		/READ AND CHECK
1255	2057	ISE			/ADDRESS ERROR
1256	5253	JMP			/CONTINUE READ AND CHECK
1257	1057	TAD			/CHANGE TO PROGRAM DATA FIELD
1260	7041	CIA			/DONE
1261	1457	TAD	I		/GOOD
1262	7640	SZA	CLA		/BAD
1263	5273	JMP			/CHANGE TO PROGRAM DATA FIELD
1264	2057	ISZ			/DONE
1265	9257	JMP			/GOOD
1266	1063	TAD			/BAD
1267	1141	TAD			/CHANGE TO PROGRAM DATA FIELD
1270	3271	DCA			/DONE
1271	6201	CDF0			/GOOD
1272	5650	JMP	I		/BAD
					/CHANGE TO PROGRAM DATA FIELD
					/DONE
1273	1057	ADDER1,	TAD	TESTAD	/GOOD
1274	3066	DCA		GDATA	/BAD
1275	1457	TAD	I	TESTAD	/CHANGE TO PROGRAM DATA FIELD
1276	3065	DCA		BDATA	/ADDRESS ERROR TEST1
1277	1063	TAD		STKPIN	
1300	1141	TAD		K6201	
1301	3302	DCA		.*+1	
1302	6201	CDF0			
1303	4770	JMS		ERR1	
1304	1064	TAD		STKTST	

1305	1141	TAD	K6201	
1306	3307	DCA	.+1	
1307	6201	COF0		/CHANGE TO TEST DATA FIELD
1310	5264	JMP	ADDR1	

1370	2256			
1371	0214			
1372	1514			
1373	1443			
1374	1400			
1375	1516			
1376	1445			
1377	1402			
	1400			

PAGE

/TEST 2

/WRITE THE COMPLEMENT VALUE OF EACH LOCATION INTO ITSELF AND CHECK

1400	0000	TEST2,	0	TESTAD	/CLEAR TEST ADDRESS COUNTER
1401	3057	DCA			/CHANGE TO TEST DATA FIELD
1402	6201	COF0			
1403	1057	TEST2A,	TAD	TESTAD	
1404	7040	GMA			
1405	3437	DCA I		TESTAD	/WRITE MEMORY
1406	2057	ISE		TESTAD	/4096 TIMES
1407	5203	JMP	TEST2A		/READ AND CHECK
1410	1057	TEST2B,	TESTAD		
1411	7001	IAC			
1412	1457	TAD I	TESTAD		
1413	7640	SEA	CLA		
1414	5224	JMP	ADDR2		/ADDRESS ERROR
1415	2057	ISE	TESTAD		/CONTINUE READ AND CHECK
1416	5210	JMP	TEST2B		
1417	1063	TAD	STKPIN		
1420	1141	TAD	K6201		
1421	3222	DCA	.+1		
1422	6201	COF0			/CHANGE TO PROGRAM DATA FIELD
1423	5600	JMP I	TEST2		/DONE

1424	1057	ADDR2,	TAD	TESTAD	
1425	7040	GMA			
1426	3066	DCA	GDATA		/GOOD
1427	1457	TAD I	TESTAD		
1430	3065	DCA	BOATA		/BAD
1431	1063	TAD	STKPIN		
1432	1141	TAD	K6201		
1433	3234	DCA	.+1		
1434	6201	COF0			/CHANGE TO PROGRAM DATA FIELD
1435	4777	JMS	ERR2		/ADDRESS ERROR TEST 2
1436	1064	TAD	STKST		
1437	1141	TAD	K6201		
1440	3241	DCA	.+1		
1441	6201	COF0			/CHANGE TO TEST DATA FIELD

1442 5215

JMP ADDR2

/TEST 3

/WRITE THE VALUE OF EACH LOCATION INTO ITSELF AND CHECK BACKWARDS

1443 0000
 1444 3057
 1445 6201
 1446 1057
 1447 1074
 1450 3057
 1451 1057
 1452 3457
 1453 1057
 1454 7640
 1455 5246
 1456 1057
 1457 1074
 1460 3057
 1461 1057
 1462 7041
 1463 1457
 1464 7640
 1465 5276
 1466 1057
 1467 7640
 1470 5256
 1471 1063
 1472 1141
 1473 3274
 1474 6201
 1475 5643

TEST3, 0
 DCA TESTAD
 CDF0
 TDF3, TAD TESTAD
 TEST3A, TAD M1
 TAD TESTAD
 DCA TESTAD
 TAD TESTAD
 DCA I TESTAD
 TAD TESTAD
 SZA CLA
 JMP TEST3A
 TAD TESTAD
 M1
 DCA TESTAD
 TAD TESTAD
 CIA
 TAD I TESTAD
 SZA CLA
 JMP ADDR3
 TAD TESTAD
 DCA
 TAD TEST3B
 STKPIN
 K6201
 .+1
 DCA
 CDF0
 JMP I TEST3

/CLEAR TEST ADDRESS COUNTER
/CHANGE TO TEST DATA FIELD

/WRITE MEMORY

/4096 TIMES

/READ AND CHECK

/ADDRESS ERROR

/CONTINUE READ AND CHECK

/CHANGE TO PROGRAM DATA FIELD
/DONE

/GOOD

/BAD

/CHANGE TO PROGRAM DATA FIELD
/ADDRESS ERROR TEST 3

/CHANGE TO TEST DATA FIELD

/TEST 4

/WRITE THE COMPLEMENT VALUE OF EACH LOCATION INTO ITSELF
/AND CHECK BACKWARDS

1514 0000
 1515 3057

TEST4, 0
 DCA TESTAD

/CLEAR TEST ADDRESS COUNTER

/CHANGE TO TEST DATA FIELD

TOF4, CDF0
TEST4A, TAD

1516 6201
1517 1057

TESTAD
M1

1520 1074
1521 3057

DCA
TESTAD

1522 1057
1523 7040

TAD
CMA

1524 3457
1525 1057

DCA I
TESTAD

1526 7640
1527 5317

TAD
SZA CLA

1530 1057
1531 1074

JMP
TESTAD

1532 3057
1533 1057

TAD
DCA

1534 7001
1535 1457

TAD
IAC

1536 7640
1537 5350

TAD I
SZA CLA

1540 1057
1541 7640

JMP
TESTAD

1542 5330
1543 1063

ADDR4,
SZA CLA

1544 1141
1545 3346

TAD
DCA

1546 6201
1547 5714

TAD
CDF0

1550 1057
1551 7040

JMP I
TESTAD

1552 3066
1553 1457

ADDER4,
CMA

1554 3065
1555 1063

TAD
DCA

1556 1141
1557 3360

TAD
DCA

1560 6201
1561 4775

CDF0
JMS

1562 1064
1563 1141

ERR4
STKTST

1564 3365
1565 6201

TAD
DCA

1566 5340

CDF0
JMP

1575 2416
1576 2400

ADDER4,
CMA

1577 1600

TAD
DCA

1600 7200
1601 1154

DCA
TESTAD

1602 3153

BDATA
STKPIN

1602 3153

K6201
.*1

1602 3153

TESTAD
K6201

1602 3153

.*1
ERR4

1602 3153

STKTST
K6201

1602 3153

.*1
ADDR4

1602 3153

ADDER4,
CMA

1602 3153

TAD
DCA

1602 3153

CDF0
JMS

1602 3153

ERR4
STKTST

1602 3153

K6201
.*1

1602 3153

ADDER4,
CMA

1602 3153

TAD
DCA

1602 3153

CDF0
JMP

1602 3153

ADDER4,
CMA

1602 3153

TAD
DCA

1602 3153

CDF0
JMS

1602 3153

ERR4
STKTST

1602 3153

K6201
.*1

1602 3153

ADDER4,
CMA

1602 3153

TAD
DCA

1602 3153

/SETUP 5 MINUTE TIMER & TEST SELECTED TO RUN
/RUN0, CLA
TAD
DCA

/RUN ALL TESTS
MIN50
MIN5

PAGE

/CHANGE TO PROGRAM DATA FIELD
/ADDRESS ERROR TEST 4

/CHANGE TO PROGRAM DATA FIELD
/DONE

/CONTINUE READ AND CHECK

/ADDRESS ERROR

/READ AND CHECK

/4096 TIMES

/WRITE MEMORY

```

1603 1133 TAD K6003
1604 5777 JMP EXTAD0
1605 7200 CLA
1606 1155 TAD MIN51
1607 3153 DCA MIN5
1610 7001 IAC
1611 5777 JMP EXTAD0
1612 7300 CLA CLL
1613 1156 TAD MIN52
1614 3153 DCA MIN5
1615 7005 IAC RAL
1616 5777 JMP EXTAD0
1617 7200 CLA
1620 1157 TAD MIN53
1621 3153 DCA MIN5
1622 7132 STL RTR
1623 5777 JMP EXTAD0
1624 7200 CLA
1625 1160 TAD MIN54
1626 3153 DCA MIN5
1627 7130 STL RAR
1630 5777 JMP EXTAD0
    
```

/RUN ONLY TEST 1

/RUN ONLY TEST 2

/RUN ONLY TEST 3

/RUN ONLY TEST 4

/RETURN IF PROGRAM IS IN SELECTED STACK
 /RETURN+1 IF PROGRAM IS NOT IN SELECTED STACK

```

1631 0000 SAME, 0
1632 1063 TAD STKPIN
1633 7041 CIA
1634 1064 TAD STKTST
1635 7640 SZA CLA
1636 2231 ISZ SAME
1637 5631 JMP I SAME
    
```

/PROG NOT IN SEL STACK

/CHECK PROGRAM RELOCATION SWITCH
 /RETURN IF RELOCATE, RETURN+1 IF INHIBIT RELOCATION

```

1640 0000 CHKSW3, 0
1641 7604 LAS
1642 0023 AND SW3
1643 7640 SZA CLA
1644 2240 ISZ CHKSW3
1645 5640 JMP I CHKSW3
    
```

/INHIBIT RELOCATION

/RELOCATE THE PROGRAM

```

1646 0000 RELO, 0
1647 7200 CLA
1650 3072 DCA COUNT
1651 3067 DCA MOVE
1652 1141 TAD K6201
1653 1063 TAD STKPIN
1654 3265 DCA RELO2
1655 1141 TAD K6201
    
```

/CLEAR ERROR COUNTER

```

1656 1064 STKTST
1657 3267 DCA RELO3
1660 1265 TAD RELO2
1661 3272 DCA RELO4
1662 1142 TAD K6203
1663 1064 TAD STKTST
1664 3303 DCA RELO5
1665 6201 RELO2, CDF0 /MOVE FROM DATA FIELD
1666 1467 TAD I MOVE
1667 6201 RELO3, CDF0 /MOVE TO DATA FIELD
1670 3467 DCA I MOVE
1671 1467 TAD I MOVE
1672 6201 RELO4, CDF0 /MOVE FROM DATA FIELD
1673 7041 CIA
1674 1467 TAD I MOVE
1675 7640 SZA CLA
1676 4776 JMS ERRM /MOVE ERROR
1677 2067 ISZ MOVE
1700 5265 JMP RELO2
1701 1072 TAD COUNT
1702 7650 SNA CLA
1703 6203 CBFO /SKIP IF MOVE ERROR
1704 5646 JMP I RELO /CHANGE TO NEW PROG FIELD

```

```

/INCREMENT CONTROL OF UPPER STACKS NOT TESTED AND/OR
/STACKS NOT IN THE SYSTEM

```

```

1705 0000
1706 2031 ABOVE, ISZ STACK1 /RETURN ADDRESS
1707 2032 ISZ STACK2
1710 2033 ISZ STACK3
1711 2034 ISZ STACK4
1712 2035 ISZ STACK5
1713 2036 ISZ STACK6
1714 2037 ISZ STACK7
1715 5705 JMP I ABOVE-1

```

```

/INCREMENT CONTROL OF LOWER STACKS NOT TESTED

```

```

1716 2036 ISZ STACK6
1717 2035 ISZ STACK5
1720 2034 ISZ STACK4
1721 2033 ISZ STACK3
1722 2032 ISZ STACK2
1723 2031 ISZ STACK1
1724 2030 ISZ STACK0
1725 5726 JMP I .+1
1726 0000

```

```

/CLEAR ALL STACKS OR STACKS TO BE TESTED

```

```

1727 3037 DCA STACK7
1730 3036 DCA STACK6
1731 3035 DCA STACK5

```

1732	5034	DCA	STACK4	
1733	3033	DCA	STACK3	
1734	3032	DCA	STACK2	
1735	3031	DCA	STACK1	
1736	3030	DCA	STACK0	
1737	5740	JMP I	.+1	/RETURN ADDRESS
1740	0000	Ø		/CLEAR ALL STACK SELECTION CONTROLS
1741	5527	JMP	.-12	
/OBTAIN -SSL (MINUS STARTING STACK LIMIT)				
/MSSL,				
1742	0000	Ø		
1743	1061	TAD	SSL	
1744	7112	CLL	RTR	
1745	7010	RAR		
1746	7041	CIA		
1747	5742	JMP I	MSSL	
1776	2434			
1777	0205			
	2000			PAGE

/CONVERT OCTAL NUMBERS FOR TYPEOUT				
/SIXTY, Ø				
2000	0000	CLA	CLL	
2001	7300	TAD I	SIXTY	/GET ADDRESS OF OPERAND
2002	1600	DCA	SIXTYØ	
2003	3237	ISE	SIXTY	/GET STORAGE ADDRESS
2004	2200	TAD I	SIXTY	/CORRECT RETURN ADDRESS
2005	1600	DCA	SIXTY1	
2006	3240	ISE	SIXTY	/AC=7700
2007	2200	TAD	K77	/AND OPERAND FIRST 2 DIGITS
2010	1111	CMA		
2011	7040	AND I	SIXTYØ	
2012	0637	CLL	RTR	
2013	7112	RTR		
2014	7012	RTR		
2015	7012	RTR		
2016	4224	JMS	CNV	/POSITION FIRST 2 DIGITS
2017	2240	ISE	SIXTY1	/CONVERT DIGITS FOR TYPEOUT
2020	1111	TAD	K77	/INCREMENT STORAGE ADDRESS
2021	0637	AND I	SIXTYØ	/AND OPERAND SECOND 2 DIGITS
2022	4224	JMS	CNV	/CONVERT DIGITS FOR TYPEOUT
2023	5600	JMP I	SIXTY	
2024	0000	Ø		
2025	3241	DCA	SIXTY2	/SAVE DIGITS
2026	1241	TAD	SIXTY2	
2027	7106	CLL	RTL	
2030	7004	RAL		
2031	0130	AND	K707	/AND LEFT DIGIT
2032	1241	TAD	SIXTY2	
2033	0130	AND	K707	/AND RIGHT DIGIT
2034	1134	TAD	K6060	
2035	3640	DCA I	SIXTY1	/STORE CONVERTED DIGITS

2036 5624 JMP I CNV

2037 0000 SIXTY0, 0 /ADDRESS OF OPERAND
2040 0000 SIXTY1, 0 /STORAGE ADDRESS
2041 0000 SIXTY2, 0 /TEMPORARY STORAGE

/TYPEOUT CHARACTER IN AC AND RETURN

2042 0000 TYPE, 0 /TYPEOUT CHARACTER IN AC AND RETURN
2043 6046 TLS /TRANSMIT CHARACTER
2044 6041 TSF /TRANSMIT CHARACTER
2045 5244 JMP .-1 /WAIT FOR FLAG
2046 7300 CLA CLL /WAIT FOR FLAG
2047 5642 JMP I TYPE

/TELETYPE OUTPUT ROUTINE WITH BELL

2050 0000 MESSAGE, 0 /TELETYPE OUTPUT ROUTINE WITH BELL
2051 7240 STA /FIRST WORD -1
2052 1250 TAD MESSAGE /FIRST WORD -1
2053 3010 DCA 10 /FIRST WORD -1
2054 1410 TAD I /FIRST WORD -1
2055 3266 DCA MSRGT /FIRST WORD -1
2056 1266 TAD MSRGT /FIRST WORD -1
2057 7112 CLL RTR /FIRST WORD -1
2060 7012 RTR /FIRST WORD -1
2061 7012 RTR /FIRST WORD -1
2062 4267 JMS /FIRST WORD -1
2063 1266 TAD /FIRST WORD -1
2064 4267 JMS /FIRST WORD -1
2065 5254 JMP MESSAGE+4 /FIRST WORD -1
2066 0000 MSRGT, 0 /FIRST WORD -1
2067 0000 TYPECH, 0 /FIRST WORD -1
2070 0111 AND /FIRST WORD -1
2071 7490 SNA /FIRST WORD -1
2072 5410 JMP I /FIRST WORD -1
2073 1100 TAD /FIRST WORD -1
2074 7440 SZA /FIRST WORD -1
2075 5300 JMP .+3 /FIRST WORD -1
2076 1112 TAD /FIRST WORD -1
2077 5320 JMP MTP /FIRST WORD -1
2100 1077 TAD M4 /FIRST WORD -1
2101 7500 SMA /FIRST WORD -1
2102 5305 JMP .+3 /FIRST WORD -1
2103 1127 TAD K340 /FIRST WORD -1
2104 5320 JMP MTP /FIRST WORD -1
2105 1076 TAD M3 /FIRST WORD -1
2106 7440 SZA /FIRST WORD -1
2107 5312 JMP .+3 /FIRST WORD -1
2110 1113 TAD K212 /FIRST WORD -1
2111 5320 JMP MTP /FIRST WORD -1
2112 1075 TAD M2 /FIRST WORD -1
2113 7440 SZA /FIRST WORD -1

/IS IT END OF MESSAGE?
/RETURN TO PROGRAM
/SUBTRACT 34

/CODE IS BELL

/SUBTRACT 4
/CODE LESS THAN 40?
/NO

/YES, ADD 300, CODE IS ALPHA
/SUBTRACT 3

/CODE IS LINE FEED
/SUBTRACT 2

/POSITION FIRST CHARACTER
/TYPEOUT FIRST CHARACTER
/TYPEOUT SECOND CHARACTER
/CONTINUE TYPING

```

2114 5317 JMP .+3
2115 1114 TAD K215
2116 7410 SKP
2117 1116 TAD K245
2120 4242 JMS TYPE
2121 5667 JMP I TYPECH
2200 PAGE
    
```

```

/ERROR ROUTINE (BELL ON ERROR HAS PRIORITY)
/
    
```

```

2200 0000 RETURN, 0
2201 6002 CODERR, IOF
2202 7604 LAS
2203 0022 AND SW2
2204 7650 SNA CLA
2205 5211 JMP .+4
    
```

```

2206 1112 RBELL, TAD K207
2207 4543 JMS I XTYPE
2210 5600 JMP I RETURN
2211 7604 LAS
2212 0021 AND SW1
2213 7640 SEA CLA
2214 5242 JMP STOP
2215 6224 RIF
2216 7012 RTR
2217 7010 RAR
2220 0101 AND
2221 1131 TAD K4060
2222 3233 DCA ERROR0
2223 1200 TAD RETURN
2224 1074 TAD M1
2225 3073 DCA ERRLOC
2226 4545 JMS I XSIXTY
2227 0073 ERRLOC
2230 2234 ERROR1
2231 4544 JMS I XMESAG
2232 4543 4543
    
```

```

2233 0000 ERROR0, 0
2234 0000 ERROR1, 0
2235 0000 0
2236 4040 4040
2237 0000 0000
2240 5641 JMP I .+1
2241 0000 ADDER, 0
2242 7604 STOP, LAS
2243 0020 AND SW0
2244 7650 SNA CLA
2245 5251 JMP LIMIT
2246 1200 TAD RETURN
2247 1074 TAD M1
2250 7402 HLT
2251 7604 LAS
2252 0024 AND SW4
    
```

```

/PROGRAM RETURN ADDRESS
/CHECK FOR BELL ON ERROR
/BELL CODE
/RING BELL
/CHECK FOR INHIBIT TYPEOUT
/INHIBIT TYPEOUT
/READ INST FIELD
/TYPEOUT ERROR LOCATION
/FIELD
/PROGRAM LOCATION OF ERROR JMS
/TYPEOUT ERROR
/ADDRESS OF ERROR TYPEOUT
/HALT AFTER ERROR
/INHIBIT ERROR HALT
/HALT WITH AC=ERROR LOC
/CHANGE STACK LIMITS?
    
```


2253 7640 SZA CLA
 2254 5777 JMP CHEXA
 2255 5600 JMP I RETURN

/YES
 /NO

/ADDRESS ERROR TEST 1
 /ADDRESS ERROR TEST 1
 ERR1,
 0
 ISZ COUNT /ADDRESS ERROR OCCURRED
 SKP
 JMP -2
 CLA
 TAD ERR1
 DCA I XRETUR /STORE RETURN ADDRESS
 TAD +3
 DCA I XADDER /STORE ERROR TYPEOUT ADDRESS
 SKP
 PERR1
 TAD K6100
 DCA Z24
 LAS SW2 /BELL ON ERROR?
 AND RBELL /YES
 SZA CLA
 JMP I SN1 /INHIBIT ERROR TYPEOUT?
 ISZ XSTOP /YES
 SKP HEAD1
 JMS HEAD12
 JMP I XCODER /TYPEOUT ERROR HEADING
 /GO TO ERROR ROUTINE

/TEST 1

/BELL ON ERROR?

/YES

/INHIBIT ERROR TYPEOUT?

/YES

/TYPEOUT ERROR HEADING
 /GO TO ERROR ROUTINE

/FIELD OF ERROR

/FAILING ADDRESS

/GOOD

/BAD

/FAILING ADDRESS

/GOOD

2256 0000
 2257 2072
 2260 7410
 2261 5257
 2262 7200
 2263 1256
 2264 3547
 2265 1270
 2266 3551
 2267 7410
 2270 2307
 2271 1135
 2272 3340
 2273 7604
 2274 0022
 2275 7640
 2276 5206
 2277 7604
 2300 0021
 2301 7640
 2302 5550
 2303 2053
 2304 7410
 2305 4776
 2306 5546

/ADDRESS ERROR TEST 1
 ERR1,
 0
 ISZ COUNT /ADDRESS ERROR OCCURRED
 SKP
 JMP -2
 CLA
 TAD ERR1
 DCA I XRETUR /STORE RETURN ADDRESS
 TAD +3
 DCA I XADDER /STORE ERROR TYPEOUT ADDRESS
 SKP
 PERR1
 TAD K6100
 DCA Z24
 LAS SW2 /BELL ON ERROR?
 AND RBELL /YES
 SZA CLA
 JMP I SN1 /INHIBIT ERROR TYPEOUT?
 ISZ XSTOP /YES
 SKP HEAD1
 JMS HEAD12
 JMP I XCODER /TYPEOUT ERROR HEADING
 /GO TO ERROR ROUTINE

ERR1A,

PERR1,
 TAD STKTST
 CLL RTR
 RAR
 TAD K4060
 DCA Z20
 JMS I XSIXTY
 TESTAD
 Z21 XSIXTY
 JMS I XSIXTY
 GDATA
 Z22 XSIXTY
 JMS I XSIXTY
 BDATA
 Z23 XMESAG
 JMS I XMESAG
 0
 0
 0
 0
 4040
 0
 0
 Z20,
 Z21,
 Z22,
 0000
 0000
 0000
 4040
 0000
 0000

PERR1,
 TAD STKTST
 CLL RTR
 RAR
 TAD K4060
 DCA Z20
 JMS I XSIXTY
 TESTAD
 Z21 XSIXTY
 JMS I XSIXTY
 GDATA
 Z22 XSIXTY
 JMS I XSIXTY
 BDATA
 Z23 XMESAG
 JMS I XMESAG
 0
 0
 0
 0
 4040
 0
 0
 Z20,
 Z21,
 Z22,
 0000
 0000
 0000
 4040
 0000
 0000

2307 1064
 2310 7112
 2311 7010
 2312 1131
 2313 3326
 2314 4545
 2315 0057
 2316 2327
 2317 4545
 2320 0066
 2321 2332
 2322 4545
 2323 0065
 2324 2335
 2325 4544
 2326 0000
 2327 0000
 2330 0000
 2331 4040
 2332 0000
 2333 0000

2334	4040	4040	
2335	0000	0	
2336	0000	0	/BAD
2337	4040	4040	
2340	0000	0	/TEST
2341	5550	JMP I	XSTOP

/ADDRESS ERROR TEST 2

```

ERR2,
2342 0000
2343 2072
2344 7410
2345 5343
2346 7200
2347 1342
2350 3547
2351 1354
2352 3551
2353 7410
2354 2307
2355 1136
2356 3340
2357 5273
2376 2477
2377 0214
2400

COUNT
.-2
ERR2
XRETUR
.+3
XADDER
K6200
Z24
ERR1A

/ADDRESS ERROR OCCURRED
/STORE RETURN ADDRESS
/STORE ERROR TYPEOUT ADDRESS
/TEST 2
PAGE

```

```

/ADDRESS ERROR TEST 3
ERR3,
2400 0000
2401 2072
2402 7410
2403 5201
2404 7200
2405 1200
2406 3547
2407 1212
2410 3551
2411 7410
2412 2307
2413 1137
2414 3777
2415 5776

COUNT
.-2
ERR3
XRETUR
.+3
XADDER
K6300
Z24
ERR1A

/ADDRESS ERROR TEST 3
/ADDRESS ERROR OCCURRED
/STORE RETURN ADDRESS
/STORE ERROR TYPEOUT ADDRESS
/TEST 3

```

```

/ADDRESS ERROR TEST 4
ERR4,
2416 0000
2417 2072
2420 7410
2421 5217
2422 7200
2423 1216
2424 3547
2425 1250
2426 3551
2427 7410
2430 2307
2431 1140
2432 3777
2433 5776

COUNT
.-2
ERR4
XRETUR
.+3
XADDER
K6400
Z24
ERR1A

/ADDRESS ERROR TEST 4
/ADDRESS ERROR OCCURRED
/STORE RETURN ADDRESS
/STORE ERROR TYPEOUT ADDRESS
/TEST 4

```

```

/RELOCATION MOVE ERROR OCCURRED
/ERRM, 0 COUNT /RELO ERROR OCCURRED
2434 0000
2435 2072
2436 7410
2437 5235
2440 7200
2441 1234
2442 3547
2443 1246
2444 3551
2445 5246
2446 2447
ERRM
ISZ
SKP
JMP
CLA
TAD
DCA I
TAD
DCA I
JMP I
PERRM
XRETUR
XADDER
XCODER

```

```

/STORE RETURN ADDRESS
/STORE ERROR TYPEOUT ADDRESS
PERRM, TAD STKTST
2447 1064
2450 7112
2451 7010
2452 1131
2453 3270
2454 4545
2455 0067
2456 2471
2457 4544
2460 2205
2461 1417
2462 4005
2463 2222
2464 4001
2465 2440
2466 0000
2467 4544
2470 0000
2471 0000
2472 0000
2473 0000
2474 7240
2475 3053
2476 5550

```

"RELO ERR AT "

```

JMS I XMESAG
0
0
0
0
STA
DCA
JMP I
HEAD1
XSTOP

```

```

/TYPEOUT TEST 1 OR 2 ERROR HEADING
/HEAD12, 0
2477 0000
2500 4544
2501 4543
2502 2022
2503 4014
2504 1703
2505 4040
2506 4001
2507 0404
2510 2240
2511 4040
JMS I XMESAG
TEXT "%#PR LOC ADDR GOOD BAD TEST"

```

2512 0717
 2513 1704
 2514 4040
 2515 0201
 2516 0440
 2517 4024
 2520 0523
 2521 2400
 2522 5677

JMP I HEAD12

2576 2273
 2577 2340
 2600

PAGE

/TYPEOUT PROGRAM TITLE
 /

2600 0000
 2601 4544
 2602 4543
 2603 4305
 2604 0170
 2605 5505
 2606 4005
 2607 3024
 2610 4015
 2611 0515
 2612 4001
 2613 0404
 2614 2240
 2615 2405
 2616 2324
 2617 4300
 2620 5600

TITLE, 0
 JMS I XMESAG
 TEXT "X#EA8-E EXT MEM ADDR TEST#"

JMP I TITLE

/TYPEOUT TO SET SWITCHES
 /

2621 0000
 2622 4544
 2623 4543
 2624 2305
 2625 2425
 2626 2040
 2627 2322
 2630 4046
 2631 4003
 2632 1716
 2633 2400
 2634 7402
 2635 5621

SETSW, 0
 JMS I XMESAG
 TEXT "%SETUP SR & CONT"

HLT
JMP I SETSW

/WAIT FOR SWITCH SETTING

/TYPEOUT 'NO PROGRAM RELOCATION WILL OCCUR'
 /

PNOREL, 0

2637 4544 JMS I XMESAG
2640 4543 TEXT
2641 1617
2642 4022 "%#NO RELOCATION, PROG IN STACK "
2643 0514
2644 1703
2645 0124
2646 1117
2647 1654
2650 4020
2651 2217
2652 0740
2653 1116
2654 4023
2655 2401
2656 0313
2657 4000
2660 6224
2661 7106
2662 7004
2663 1132
2664 3266
2665 4544
2666 0000
2667 7240
2670 3053
2671 5636

RIF
CLL RTL
RAL
TAD K6000
DCA Z8
JMS I XMESAG
0
STA
DCA HEAD1
JMP I PNOREL
Z8, /RESET ERROR HEADING

/PROGRAM RELOCATION WILL OCCUR

2672 0000
2673 4544 JMS I XMESAG
2674 4543 TEXT
2675 2022 "%#PROG WILL RELOCATE"
2676 1707
2677 4027
2700 1114
2701 1440
2702 2205
2703 1417
2704 0301
2705 2405
2706 0000
2707 7240
2710 3053
2711 5672

STA
DCA HEAD1
JMP I PNOREL
/RESET ERROR HEADING

/TYPEOUT 'PROGRAM IS IN SELECTED FIELD'

2712 4544
2713 4543 JMS I XMESAG
2714 2022 TEXT
2715 1707 "%#PROGRAM IN SELECTED FIELD"
2716 2201
2717 1540

2720 1116
 2721 4023
 2722 0514
 2723 0503
 2724 2405
 2725 0440
 2726 0611
 2727 0514
 2730 0400
 2731 5777'

JMP CHEXA /SETUP SWITCHES AGAIN

/TYPEOUT 'NONE' FOR NO LEGAL STACK SELECTION

2732 4544
 2733 1617
 2734 1605
 2735 0000
 2736 5777'

NOSTK, JMS I XNESAG
 TEXT "NONE"

JMP CHEXA

/FIND HIGHEST STACK NUMBER IN THIS SYSTEM

2737 0000
 2740 7300
 2741 3060
 2742 6211
 2743 4776'
 2744 6221
 2745 4776'
 2746 6231
 2747 4776'
 2750 6241
 2751 4776'
 2752 6251
 2753 4776'
 2754 6261
 2755 4776'
 2756 6271
 2757 4776'
 2760 5737
 2776 3000
 2777 3000

HIGHST, 0
 CLA CLL
 DCA KBINT /CLEAR HIGH STACK COUNTER
 CDF1
 JMS CSS /CHECK FOR FIELD 1
 CDF2 CSS /CHECK FOR FIELD 2
 JMS CSS /CHECK FOR FIELD 3
 CDF3 CSS /CHECK FOR FIELD 4
 JMS CSS /CHECK FOR FIELD 5
 CDF4 CSS /CHECK FOR FIELD 6
 JMS CSS /CHECK FOR FIELD 7
 CDF5
 JMS
 CDF6
 JMS
 CDF7
 JMS
 JMP I HIGHST

KHIGH, JMP I

PAGE

/CHECK IF SELECTED STACK IS IN SYSTEM

3000 0000
 3001 7300
 3002 6224
 3003 1141
 3004 3210
 3005 1074
 3006 3615

CSS,
 0
 CLA CLL
 RIF
 TAD K6201
 DCA CSSB
 TAD M1
 DCA I CHECK

```

3007 1615 TAD I CHECK
3010 6201 CSSB, CDF 00 /PROGRAM DATA FIELD
3011 7650 SNA CLA KHIGH /SKIP IF STACK IS IN SYSTEM
3012 5777 JMP ISZ KBINT /INCREMENT STACK COUNTER
3013 2060 ISZ KBINT
3014 5600 JMP I CSS

```

```

3015 3016 CHECK, CHECK0
3016 0000 CHECK0, 0

```

/TYPEOUT NUMBER OF STACKS IN SYSTEM

```

/ TSTSYS, 0
3017 0000 JMS I XMESAG
3020 4544 4543
3021 4543 0000
3022 0000 TAD
3023 1060 IAC
3024 7001 JMS I XTYPE /TYPEOUT NUMBER
3025 4543 JMS I XMESAG " STACKS IN THIS SYSTEM"
3026 4544 TEXT
3027 4023
3030 2401
3031 0315
3032 2340
3033 1116
3034 4024
3035 1011
3036 2340
3037 2351
3040 2524
3041 0515
3042 0000
3043 5617

```

```

JMP I TSTSYS

```

/TYPEOUT CHARACTER IN THE AC AND A SPACE

```

/ TYPEPSP, 0
3044 0000 JMS I XTYPE /TYPEOUT CHAR IN AC
3045 4543 TAD K240
3046 1115 JMS I XTYPE /TYPE A SPACE
3047 4543 JMP I TYPEPSP
3050 5644

```

/RESTORE STACKS FOR RELOCATION

```

/ RESTK, 0
3051 0000 CLA
3052 7200 TAD
3053 1030 DCA
3054 3040 TAD
3055 1031 DCA
3056 3041 TAD
3057 1032

```


3060 3042
 3061 1033
 3062 3043
 3063 1034
 3064 3044
 3065 1035
 3066 3045
 3067 1036
 3070 3046
 3071 1037
 3072 3047
 3073 5651

DCA STK2
 TAD STACK3
 DCA STK3
 TAD STACK4
 DCA STK4
 TAD STACK5
 DCA STK5
 TAD STACK6
 DCA STK6
 TAD STACK7
 DCA STK7
 JMP I RESTK

3074 0000
 3075 4544
 3076 4543
 3077 2324
 3100 0103
 3101 1323
 3102 4023
 3103 0514
 3104 4704
 3105 4001
 3106 2205
 3107 4000
 3110 1037
 3111 7640
 3112 5315
 3113 1126
 3114 4244
 3115 1036
 3116 7640
 3117 5322
 3120 1125
 3121 4244
 3122 1035
 3123 7640
 3124 5327
 3125 1124
 3126 4244
 3127 1034
 3130 7640
 3131 5334
 3132 1123
 3133 4244
 3134 1033
 3135 7640
 3136 5341
 3137 1122
 3140 4244
 3141 1032

/TYPEOUT STACKS SELECTED FOR TESTING

TOSEL, 0
 JMS I XMESAG
 TEXT
 "%#STACKS SEL'D ARE "

TAD STACK7
 SEA CLA
 JMP .+3
 TAD K267
 JMS TYPESP
 TAD STACK6
 SEA CLA
 JMP .+3
 TAD K266
 JMS TYPESP
 TAD STACK5
 SEA CLA
 JMP .+3
 TAD K265
 JMS TYPESP
 TAD STACK4
 SEA CLA
 JMP .+3
 TAD K264
 JMS TYPESP
 TAD STACK3
 SEA CLA
 JMP .+3
 TAD K263
 JMS TYPESP
 TAD STACK2

/STACK 7 IS SELECTED
 /STACK 6 IS SELECTED
 /STACK 5 IS SELECTED
 /STACK 4 IS SELECTED
 /STACK 3 IS SELECTED

```

3142 7640 SZA CLA
3143 5346 JMP .+3
3144 1121 TAD K262
3145 4244 JMS TYPESP
3146 1031 TAD STACK1
3147 7640 SZA CLA
3150 5353 JMP .+3
3151 1120 TAD K261
3152 4244 JMS TYPESP
3153 1030 TAD STACK0
3154 7640 SZA CLA
3155 5360 JMP .+3
3156 1117 TAD K260
3157 4244 JMS TYPESP
3160 5674 JMP I TOSEL
    
```

/STACK 2 IS SELECTED

/STACK 1 IS SELECTED

/STACK 0 IS SELECTED

/TWO SPECIAL SCOPE LOOPS

```

3177 2760
3400 3400 LAS
3400 7604 LOOP1,
3401 3206 DCA SWAD
3402 1206 TAD SWAD
3403 3606 DCA I SWAD
3404 1606 TAD I SWAD
3405 5200 JMP LOOP1

3406 0000 SWAD, 0

3600 3600 LAS
3600 7604 LOOP2,
3601 3224 DCA FIRST
3602 7402 HLT
3603 7604 LAS
3604 3225 DCA LAST
3605 1224 TAD FIRST
3606 3226 DCA SWAD0
3607 1226 TAD SWAD0
3610 3626 DCA I SWAD0
3611 1626 TAD I SWAD0
3612 7200 CLA
3613 1226 TAD SWAD0
3614 7041 CIA
3615 1225 CIA LAST
3616 7650 SNA CLA
3617 5205 JMP LOOP2A
3620 2226 ISE SWAD0
3621 5207 JMP LOOP2B
3622 7402 HLT
3623 5200 JMP LOOP2

3624 0000 FIRST, 0
    
```

/SWITCH ADDRESS

/READ LOWER LIMIT

/READ UPPER LIMIT

/HALT RESULTED IN ILLEGAL LIMITS

/EXTENDED ADDRESS TEST FOR K8-E EXTENDED MEMORY (VER 1)

3625 0000 LAST, 0
3626 0000 SWADD, 0

PAL10 V141

2-JUN-71

21:26

PAGE 2-8

8

4000
4100
4200
4300
4400
4500
4600
4700

5000
5100
5200
5300
5400
5500
5600
5700

6000
6100
6200
6300
6400
6500
6600
6700

7000
7100
7200
7300
7400
7500
7600
7700

ABOVE	1706	CHEXN9	0761	K215	0114	MIN51	0155
ADDER	2241	CHEX0	1000	K240	0115	MIN52	0156
ADDER1	1273	CHEX00	1024	K245	0116	MIN53	0157
ADDER2	1424	CHEX01	1040	K260	0117	MIN54	0160
ADDER3	1476	CHEX02	1054	K261	0120	MOVE	0067
ADDER4	1550	CHEX03	1070	K262	0121	MSRGT	2066
ADDR1	1264	CHEX04	1104	K263	0122	MSSL	1742
ADDR2	1415	CHEX05	1120	K264	0123	MTP	2120
ADDR3	1466	CHEX06	1134	K265	0124	NORELO	0050
ADDR4	1540	CHEX07	1144	K266	0125	NOSTK	2732
BDATA	0065	CHEX0A	1006	K267	0126	PERR1	2307
BELOW	1725	CHKSWS	1640	K30	0104	PERRM	2447
CBF0	6203	CIF0	6202	K340	0127	PINF	2712
CBF1	6213	CIF1	6212	K40	0105	PNOREL	2636
CBF2	6223	CIF2	6222	K4060	0131	PREL	2672
CBF3	6233	CIF3	6232	K50	0106	RBELL	2206
CBF4	6243	CIF4	6242	K60	0107	RDF	6214
CBF5	6253	CIF5	6252	K6000	0132	RELO	1646
CBF6	6263	CIF6	6262	K6003	0133	RELO2	1665
CBF7	6273	CIF7	6272	K6060	0134	RELO3	1667
CF0	6201	CINT	6204	K6100	0135	RELO4	1672
CF1	6211	CNV	2024	K6200	0136	RELO5	1703
CF2	6221	CODERR	2201	K6201	0141	RELO5	3051
CF3	6231	COUNT	0072	K6203	0142	RESTK	3051
CF4	6241	CSS	3000	K6300	0137	RETURN	2200
CF5	6251	CSSB	3010	K6400	0140	RIB	6234
CF6	6261	CUF	6264	K7	0101	RIF	6224
CF7	6271	DOWN	1736	K70	0110	RMF	6244
CHECK	3015	ERR1	2256	K707	0130	RTF	6005
CHECK0	3016	ERR1A	2273	K77	0111	RUN0	1600
CHEXA	0214	ERR2	2342	KABOVE	0051	RUN1	1605
CHEXB	0253	ERR3	2400	KBELOW	0052	RUN2	1612
CHEXC	0262	ERR4	2416	KBINT	0060	RUN3	1617
CHEXC1	0274	ERRLOC	0073	KDOWN	0070	RUN4	1624
CHEXD	0302	ERRM	2434	KHIGH	2760	RUNTST	0056
CHEXD1	0310	ERROR0	2233	LAST	3625	SAME	1631
CHEXE	0317	ERROR1	2234	LEGAL	0410	SETSW	2621
CHEXE2	0400	ESL	0062	LEGAL0	0055	SINT	6254
CHEXM	0476	EXTAD	0200	LEGALA	0463	SIXTY	2000
CHEXM1	0503	EXTAD0	0205	LEGALA	0463	SIXTY0	2037
CHEXM2	0510	FIRST	3624	LIMIT	2251	SIXTY1	2040
CHEXM3	0515	FIVE	0152	LOOP1	3400	SIXTY2	2041
CHEXN	0527	GDATA	0066	LOOP2	3600	SSL	0061
CHEXN0	0530	GTF	6004	LOOP2A	3605	STACK0	0030
CHEXN1	0600	HEAD1	0053	LOOP2B	3607	STACK1	0031
CHEXN2	0622	HEAD12	2477	M1	0074	STACK2	0032
CHEXN3	0640	HIGHST	2737	M2	0075	STACK3	0033
CHEXN4	0656	INSAME	0054	M3	0076	STACK4	0034
CHEXN5	0674	K10	0102	M4	0100	STACK5	0035
CHEXN6	0712	K20	0103	MESSAGE	2050	STACK6	0036
CHEXN7	0730	K207	0112	MIN5	0153	STACK7	0037
CHEXN8	0746	K212	0113	MIN50	0154	STK0	0040
						STK1	0041

STK2	0042	0042
STK3	0043	2471
STK4	0044	2326
STK5	0045	2327
STK6	0046	2332
STK7	0047	2335
STKPIN	0063	2340
STKTST	0064	2666
STOP	2242	
SUF	6274	
SW0	0020	
SW1	0021	
SW2	0022	
SW3	0023	
SW4	0024	
SW5	0025	
SW68	0026	
SW911	0027	
SWAD	3406	
SWAD0	3626	
TDF1	1252	
TDF2	1402	
TDF3	1445	
TDF4	1516	
TEMP	0071	
TEST	1200	
TEST1	1250	
TEST1A	1253	
TEST1B	1257	
TEST2	1400	
TEST2A	1403	
TEST2B	1410	
TEST3	1443	
TEST3A	1446	
TEST3B	1456	
TEST4	1514	
TEST4A	1517	
TEST4B	1530	
TESTAD	0057	
TITLE	2600	
TOSEL	3074	
TSTSYS	3017	
TYPE	2042	
TYPECH	2067	
TYPESP	3044	
XADDER	0151	
XCODER	0146	
XMESAG	0144	
XRETUR	0147	
XSIXTY	0145	
XSTOP	0150	
XTYPE	0143	

Z10	
Z11	
Z20	
Z21	
Z22	
Z23	
Z24	
Z8	

2470	
2471	
2326	
2327	
2332	
2335	
2340	
2666	

ERRORS DETECTED: 0

LINKS GENERATED: 133

RUN-TIME: 13 SECONDS

3K CORE USED