

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCKBF-A1-D

PRODUCT NAME: STACK LIMIT TEST  
STANDARD ON 11/45  
KJ11-A OPTION ON 11/40

DATE CREATED: 15 MAR 1972

MAINTAINER: DIAGNOSTIC GROUP

AUTHORS: JOHN ADAMS

COPYRIGHT(c) 1972  
DIGITAL EQUIPMENT CORPORATION  
MAYNARD, MASS

1,0 ABSTRACT  
 THIS PROGRAM INCREMENTLY TESTS THE STACK LIMIT FUNCTION

2,0 REQUIREMENTS

2,1 EQUIPMENT  
 BASIC 11/45 SYSTEM OR  
 11/40 WITH STACK LIMIT OPTION

2,2 STORAGE  
 THIS PROGRAM USES 2 THRU 17500

2,3 PRELIMINARY PROGRAMS  
 DBAA THRU DBMA

3,0 LOADING PROCEDURE  
 LOAD PROGRAM USING ABS LOADER

4,0 STARTING PROCEDURE  
 LOAD ADDRESS 200, PRESS START; THE PROGRAM WILL LOOP  
 AND RING BELL ON PASS COMPLETION,

5,0 OPERATING PROCEDURE

5,1 SWITCH SETTINGS  
 NONE

5,2 SUBROUTINE ABSTRACTS

5,2,1 SCOPE  
 SCOPE IS A MOVE PC,R1 AND STORES THE PC+2 IN R1.

5,2,2 HLT  
 HLT IS A HALT INSTRUCTION,

6,0 ERRORS  
 ALL ERRORS WILL CAUSE A HALT  
 TRAP AND INTERRUPT ERRORS WILL CAUSE A HALT AT VECTOR+2,

6,1 ERROR RECOVERY  
 PRESS CONTINUE TO PROCEED TO NEXT TEST

6,2 ERROR LOOPING  
 TO LOOP ON AN ERROR, PLACE A BRANCH TO THE PREVIOUS SCOPE  
 INSTRUCTION IN PLACE OF THE HALT INSTRUCTION,  
 NOTE THAT IF THE ERROR IS INTERMITTANT THAT THE TEST WILL  
 DROP THRU THE HALT AND PROCEED TO THE NEXT TEST,  
 THEREFORE, TO LOOP THE TEST CONTINUOUSLY REPLACE THE BEQ ,+4  
 INSTRUCTION IMMEDIATELY PRECEDING THE HALT WITH A BRANCH  
 BACK TO THE PREVIOUS SCOPE,  
  
 TO LOOP ON TRAP FAILURES, PATCH IN THE FOLLOWING ROUTINE  
 AT THE ADDRESS OF THE TRAP VECTOR,

```

TRAPVEC;      TRAPVEC+4
TRAPVEC+2;    0
TRAPVEC+4;    012716 ;MOVE SCOPE ADDRESS TO STACK
TRAPVEC+6;    ADDRESS ;ADDRESS OF PREVIOUS SCOPE
TRAPVEC+10;   R00206 ;RETURN TO TEST AT SCOPE
  
```

RESTORE ALL LOCATIONS BEFORE PROCEEDING TO NEXT TEST,

7,0 RESTRICTIONS  
NONE

8,0 MISCELLANEOUS  
ON TRAP ERRORS THE STACK POINTER(R6) WILL CONTAIN  
ADDRESS WHERE THE TRAP OCCURED;

8,1 EXECUTION TIME  
EACH PROGRAM TAKES ABOUT 1 MINUTE;

9,0 PROGRAM DESCRIPTIONS

THIS IS A TEST OF THE STACK LIMIT REGISTER AND INSURES  
CORRECT OPERATION OF THE RED AND YELLOW ZONE BOUNDARIES.  
OVERFLOW TRAPS ARE TESTED FOR ALL VALUES OF THE STACK LIMIT  
REGISTER.

TITLE MAINDEC-11-DCKRF-A POP11/25-45 STK LIM TEST  
NLIST MC,MO,SEC  
LIST ME  
AHS

TEST DCKBFA- STACK LIMIT REGISTER TEST  
THE STACK LIMIT REGISTER ALLOWS THE OVERFLOW BOUNDARIES TO BE CHANGED,  
FOR EXAMPLE IF THE STACK LIMIT REGISTER IS CLEAR THE BOUNDARY IS AT  
1423 (YELLOW ZONE) AND 348 (RED ZONE), IN ALL CASES THE YELLOW ZONE BOUNDARY  
IS AT 400(8) PLUS THE VALUE IN THE STACK LIMIT REGISTER, AND THE RED  
ZONE BEGINS 2\*(8) WORDS BELOW THE YELLOW ZONE.  
THIS TEST CHECKS THAT THE STACK LIMIT IS 400 GREATER THAN THE CONTENTS  
OF THE STACK LIMIT REGISTER (COPE PERMITTING), AND CHECKS THE LENGTH  
OF THE YELLOW ZONE AND THE BEGINNING OF THE RED ZONE.

STARTING PROCEDURE  
LOAD ADDRESS=200  
PRESS START  
BELL WILL RING WHEN TEST IS COMPLETE

EQUATE STATEMENTS

000000  
000001  
000002  
000003  
000004  
000005  
000006  
000006  
000007

R0=X0  
R1=X1  
R2=X2  
R3=X3  
R4=X4  
R5=X5  
R6=X6  
SP=X6  
PC=X7

REGISTER ADDRESSES

177776  
177770  
177570  
177564  
177566  
177570  
000000  
022626  
010701  
000340  
000004

PSW=177776  
UBREAK=177770  
SWR=177570  
TPS=177564  
TPB=177566  
DISPLAY=177570

ADDRESS OF PROCESSOR STATUS WORD  
ADDRESS OF POP11/45 MICRO BREAK REGISTER  
ADDRESS OF CONSOLE SWITCH REGISTER

ADDRESS OF CONSOLE DISPLAY REGISTER

HLT=HALT  
POP2=22626  
SCOPE=010701  
PRY7=340  
ERRVEC=4

MOVE PC TO R1  
PRIORITY LEVEL 7  
ADDRESS OF ERROR VECTOR

000000 000002 ,+2  
000002 000000 HALT  
000004 000006 ,+2  
000006 000008 HALT  
000010 000012 ,+2  
000012 000014 HALT  
000014 000016 ,+2  
000016 000018 HALT  
000020 000022 ,+2  
000022 000000 HALT

000024 000026 ,+2  
000026 000000 HALT  
000030 000032 ,+2  
000032 000000 HALT  
000034 000036 ,+2  
000036 000000 HALT  
000040 000042 ,+2  
000042 000000 HALT  
000044 000046 ,+2  
000046 000000 HALT  
000050 000052 ,+2  
000052 000000 HALT  
000054 000056 ,+2  
000056 000000 HALT  
000060 000062 ,+2  
000062 000000 HALT  
000064 000066 ,+2  
000066 000000 HALT  
000070 000072 ,+2  
000072 000000 HALT  
000074 000076 ,+2  
000076 000000 HALT  
000100 000102 ,+2  
000102 000000 HALT  
000104 000106 ,+2  
000106 000000 HALT  
000110 000112 ,+2  
000112 000000 HALT  
000114 000116 ,+2  
000116 000000 HALT  
000120 000122 ,+2  
000122 000000 HALT  
000124 000126 ,+2  
000126 000000 HALT  
000130 000132 ,+2  
000132 000000 HALT  
000134 000136 ,+2  
000136 000000 HALT  
000140 000142 ,+2  
000142 000000 HALT  
000144 000146 ,+2  
000146 000000 HALT  
000150 000152 ,+2  
000152 000000 HALT  
000154 000156 ,+2  
000156 000000 HALT  
000160 000162 ,+2  
000162 000000 HALT  
000164 000166 ,+2  
000166 000000 HALT  
000170 000172 ,+2  
000172 000000 HALT  
000174 000176 ,+2  
000176 000000 HALT

1

000200	000202	,+2
000202	000200	HALT
000204	000206	,+2
000206	000200	HALT
000210	000212	,+2
000212	000200	HALT
000214	000216	,+2
000216	000200	HALT
000220	000222	,+2
000222	000200	HALT
000224	000226	,+2
000226	000200	HALT
000230	000232	,+2
000232	000200	HALT
000234	000236	,+2
000236	000200	HALT
000240	000242	,+2
000242	000200	HALT
000244	000246	,+2
000246	000200	HALT
000250	000252	,+2
000252	000200	HALT
000254	000256	,+2
000256	000200	HALT
000260	000262	,+2
000262	000200	HALT
000264	000266	,+2
000266	000200	HALT
000270	000272	,+2
000272	000200	HALT
000274	000276	,+2
000276	000200	HALT
000300	000302	,+2
000302	000200	HALT
000304	000306	,+2
000306	000200	HALT
000310	000312	,+2
000312	000200	HALT
000314	000316	,+2
000316	000200	HALT
000320	000322	,+2
000322	000200	HALT
000324	000326	,+2
000326	000200	HALT
000330	000332	,+2
000332	000200	HALT
000334	000336	,+2
000336	000200	HALT
000340	000342	,+2
000342	000200	HALT
000344	000346	,+2
000346	000200	HALT
000350	000352	,+2
000352	000200	HALT

000354	000356	,+2
000356	000200	HALT
000360	000362	,+2
000362	000200	HALT
000364	000366	,+2
000366	000200	HALT
000370	000372	,+2
000372	000200	HALT
000374	000376	,+2
000376	000200	HALT
000400	000402	,+2
000402	000200	HALT
000404	000406	,+2
000406	000200	HALT
000410	000412	,+2
000412	000200	HALT
000414	000416	,+2
000416	000200	HALT
000420	000422	,+2
000422	000200	HALT
000424	000426	,+2
000426	000200	HALT
000430	000432	,+2
000432	000200	HALT
000434	000436	,+2
000436	000200	HALT
000440	000442	,+2
000442	000200	HALT
000444	000446	,+2
000446	000200	HALT
000450	000452	,+2
000452	000200	HALT
000454	000456	,+2
000456	000200	HALT
000460	000462	,+2
000462	000200	HALT
000464	000466	,+2
000466	000200	HALT
000470	000472	,+2
000472	000200	HALT
000474	000476	,+2
000476	000200	HALT
000500	000502	,+2
000502	000200	HALT
000504	000506	,+2
000506	000200	HALT
000510	000512	,+2
000512	000200	HALT
000514	000516	,+2
000516	000200	HALT
000520	000522	,+2
000522	000200	HALT
000524	000526	,+2
000526	000200	HALT

2

2

```

000530 000532 ,+2
000532 000530 HALT
000534 000536 ,+2
000536 000538 HALT
000540 000542 ,+2
000542 000544 HALT
000544 000546 ,+2
000546 000548 HALT
000550 000552 ,+2
000552 000554 HALT
000554 000556 ,+2
000556 000558 HALT
000560 000562 ,+2
000562 000564 HALT
000564 000566 ,+2
000566 000568 HALT
000570 000572 ,+2
000572 000574 HALT
000574 000576 ,+2
000576 000578 HALT
000580 000582 ,+2
000582 000584 HALT
000584 000586 ,+2
000586 000588 HALT
000590 000592 ,+2
000592 000594 HALT
000594 000596 ,+2
000596 000598 HALT
000600 000602 ,+2
000602 000604 HALT
000604 000606 ,+2
000606 000608 HALT
000610 000612 ,+2
000612 000614 HALT
000614 000616 ,+2
000616 000618 HALT
000620 000622 ,+2
000622 000624 HALT
000624 000626 ,+2
000626 000628 HALT
000630 000632 ,+2
000632 000634 HALT
000634 000636 ,+2
000636 000638 HALT
000640 000642 ,+2
000642 000644 HALT
000644 000646 ,+2
000646 000648 HALT
000650 000652 ,+2
000652 000654 HALT
000654 000656 ,+2
000656 000658 HALT
000660 000662 ,+2
000662 000664 HALT
000664 000666 ,+2
000666 000668 HALT
000670 000672 ,+2
000672 000674 HALT
000674 000676 ,+2
000676 000678 HALT
000700 000702 ,+2
000702 000704 HALT

```

```

000704 000706 ,+2
000706 000708 HALT
000710 000712 ,+2
000712 000714 HALT
000714 000716 ,+2
000716 000718 HALT
000720 000722 ,+2
000722 000724 HALT
000724 000726 ,+2
000726 000728 HALT
000730 000732 ,+2
000732 000734 HALT
000734 000736 ,+2
000736 000738 HALT
000740 000742 ,+2
000742 000744 HALT
000744 000746 ,+2
000746 000748 HALT
000750 000752 ,+2
000752 000754 HALT
000754 000756 ,+2
000756 000758 HALT
000760 000762 ,+2
000762 000764 HALT
000764 000766 ,+2
000766 000768 HALT
000770 000772 ,+2
000772 000774 HALT
000774 000776 ,+2
000776 000778 HALT
000200 000200 ,=200
000200 000167 000684 JMP START
000500 ,=500
000500 000000 ITAGS
000502 000740 ICNTI 5
000504 17774 SPBOT 768
000506 17775 SLRI 17774
000510 000000 SLHI 17775
TENPI 5

```

```

(CONTAINS PASS COUNT
(ADDRESS OF STACK LIMIT REGISTER
(HIGH (ODD BYTE)

```

```

001010      001010
001010 016706 177466 STAHI MOV #SPH0T,#6 INITIALIZE STACK POINTER
001014 012737 001340 177776 MOV #PHI17,#PHSW ILOCK OUT INTERRUPTS
001022 009067 177452 CLR ICNT ICLR PASS COUNT

001026 016706 177466 HEGINI MOV #SPH0T,#6 INITIALIZE STACK POINTER
001032 016737 177442 177570 MOV ICNT,#DISPLAY IDISPLAY PASS COUNT
001040 032737 000400 177570 BIT #400,#SWH ILOAD POP11/45 MICRO BREAK REGISTER?
001046 001403 BEQ ,+10
001050 113737 177572 177770 MOV# #SWH,#SUBREAK ILOAD MICRO BREAK REG WITH SR0=7

I CHECK THAT CP CAN TIME OUT TRAP
001056 012737 001074 000004 MOV #TOHET,#ERRVEC ILOAD TIMEOUT TRAP VECTOR
001064 009037 173000 CLR #4173000 IADDRESS 173000 ALWAYS TIMES OUT ON
I DATIP/DATO BUS CYCLE
001070 000000 HLT BR IERROR! FAILED TO TIME OUT TRAP
001072 000759 TOHETI CMP BEGIN ILOOP TEST
001074 022626 (6)+,(6)+ IRESTORE THE STACK

I TEST THAT THE STACK LIMIT REGISTER CAN BE REFERENCED
I USING DATI, DATIP/DATO
001076 010701 T0I SCOPE
001100 012767 001122 176670 MOV #TRA,ERRVEC ILOAD ERROR VECTOR
001106 009067 176674 CLR ENRVEC+2
001112 017737 177366 177774 MOV #SLH,#177774 IREFERENCE STACK LIMIT REGISTER
001120 000403 BR T0B IGO TO NEXT TEST
001122 022626 T0AI POP2
001124 000000 HLT BR IERROR! CANNOT REFERENCE STACK LIMIT REG,
001126 000763 T0I T0B ILOOP TEST IF ERROR

I USING DATI, DATIP/DATOB
001130 012767 001146 176640 T0B1 MOV #T0C,ERRVEC ILOAD ERROR VECTOR
001136 117777 177342 177342 MOV# #SLH,#SLH IREFERENCE ODD BYTE
001144 000403 BR T0D
001146 022626 T0C1 POP2
001150 000000 HLT BR IERROR! CANNOT REFERENCE STACK LIMIT
001152 000766 T0B T0B IUSING BYTE INSTRUCTION,

001154 012767 000006 176622 T0D1 MOV #6,ERRVEC IRESTORE ERROR TRAP VECTOR
001162 000400 BR T1 IGO TO NEXT TEST

I TEST THAT EACH BIT OF THE STACK LIMIT REGISTER CAN BE SET
I AND CLEARED, THIS TEST ROTATES A BIT THROUGH THE STACK LIMIT REGISTER,
001164 010701 T1I SCOPE
001166 012702 000400 MOV #400,#R2 ILOAD TEST VALUE
001172 010277 177306 T1A1 MOV #R2,#SLH ILOAD TEST VALUE INTO STACK LIM, REG,
001176 017700 177302 MOV #SLH,#R0 IGET RESULT
001202 020002 CMP #R0,#R2 ICHECK RESULT
001204 001401 BEQ ,+4 IBRANCH IF RESULT IS CORRECT
001206 000000 HLT IERROR! INCORRECT RESULT, R2 HAS CORRECT
I RESULT AND R0 HAS INCORRECT RESULT
    
```

```

001210 009077 177270 CLR #SLR ICLR STACK LIM, REG;
001214 117700 177264 MOV #SLR,#R0 IGET AND CHECK RESULT
001220 001401 BEQ ,+4
001222 000000 HLT IERROR! INCORRECT RESULT
001224 006302 ASL R2 ISHIFT TEST VALUE
001226 103361 BCC T1A IBRANCH IF NOT DONE

I THIS TEST INCREMENTS THE STACK LIMIT REGISTER
001210 010701 T2I SCOPE
001212 009067 177252 CLR TEMP
001216 009077 177242 CLR #SLH ICLR STACK LIMIT REGISTER
001218 117700 177236 T2A1 MOV #SLR,#R0 IGET RESULT
001220 020067 177236 CMP #R0,TEMP ICHECK RESULT
001222 001401 BEQ ,+4
001224 000000 HLT IERROR! STACK LIM, REG, WAS INCORRECT DATA
I TEMP = CORRECT RESULT

001256 105277 177224 INCB #SLH IINCREMENT VALUE IN STACK LIM, REG;
001262 105267 177223 INCB TEMP+1 IINCREMENT TEST VALUE
001266 001365 BNE T2A IBRANCH IF ALL VALUES NOT TESTED
    
```

4

TEST THAT RESET CLEARS THE STACK LIMIT REGISTER

001270	010781			T3I	SCOPE		
001272	012777	177777	177234		MOV	#-1,SLR	JPRESET SLR
001320	017700	177200			MOV	@SLR,R0	ISAVE SLR
001334	022700	177400			CMP	#177400,R0	ICHECK THAT SLR WAS LOADED
001330	001401				BEQ	,+4	
001332	000000				HLT		JERROR! SLR FAILED TO LOAD
001334	000000				RESET		JRESET CLEARS SLR
001336	017700	177162			MOV	@SLR,R0	JGT RESULT OF RESET AND CHECK RESULT
001322	001402				BEQ	T4	GO TO NEXT TEST IF RESET CLEARED SLR
001324	000000				HLT		JERROR! RESET FAILED TO CLEAR SLR
001326	000760				BR	T3	LOOP TEST IF ERROR

TEST THAT THE CLEAR INSTRUCTION CLEARS THE STACK LIMIT REGISTER

001330	010781			T4I	SCOPE		
001332	112777	177777	177140		MOV8	#-1,SLH	JPRESET ODD BYTE
001340	017700	177140			MOV	@SLH,R0	JGT RESULT
001344	022700	177400			CMP	#177400,R0	ICHECK RESULT
001350	001401				BEQ	,+4	
001352	000000				HLT		JERROR! SLR DID NOT PRESET
001354	000000	177124			CLR	@SLH	
001360	017700	177120			MOV	@SLR,R0	JGT RESULT OF CLEAR & BRANCH IF CLEAR
001364	001411				BEQ	T5	GO TO NEXT TEST
001366	000000				HLT		JERROR! CLR INST FAILED TO CLEAR SLR
001370	000757				BR	T4	LOOP TEST IF ERROR

#1410

TEST THAT AN OVERFLOW ERROR OCCURS FOR ALL STACK LIMIT REGISTER VALUES, (PROVIDED CORE IS AVAILABLE).

001410	010701			T5I	SCOPE		
001412	012702	000010			MOV	#10,R2	JINITIALIZE STACK VALUE AND
001416	012703	177400			MOV	#-400,R3	STACK LIMIT REGISTER VALUE
001422	062702	000400		T5A1	ADD	#400,R2	JLOAD NEW STACK VALUE
001426	062703	000400			ADD	#400,R3	JAND NEW STACK LIM, REG. VALUE
001432	005037	000000			CLR	#0	JCLEAR ADDRESS 0
001436	016706	177040			MOV	SPBOT,SP	JINITIALIZE THE STACK POINTER
001442	012767	001704	176334	T5B1	MOV	#LIMX,ERRVEC	JLOAD TIME OUT TRAP
001450	016204	177776			MOV	-2(2),R4	ISAVE STACK LOCATIONS
001454	016205	177774			MOV	-4(2),R5	JEXIT TEST IF EITHER INST TIMES OUT
001460	012767	001504	176316		MOV	#LIMH,ERRVEC	JLOAD OVERFLOW VECTOR
001466	010206				MOV	R2,SP	JLOAD STACK POINTER
001470	010377	177010			MOV	R3,@SLH	JLOAD STACK LIM, REG.
001474	016666	177770	177770		MOV	-10(6),-10(6)	JREFERENCE LIMIT ADDRESS
001502	000001				BR	,+4	JSHOULD NOT HAVE TRAPPED
001504	000000			LIMH1	HLT		JERROR! REFERENCE TO LIMIT ADDRESS
001506	010206				MOV	R2,SP	JCAUSED AN OVERFLOW
001510	012767	001520	176260		MOV	#LIMH,ERRVEC	JREPOINT STACK POINTER
							JREPOINT OVERFLOW VECTOR

001516	016666	177760	177760		MOV	-12(4),-12(6)	JREFERENCE FIRST 'YELLOW' ADDRESS
001524	020000				HLT		JERROR! SHOULD HAVE TRAPPED
001526	020337	000000		LIMB1	TST	#0	JWAS IT A 'RED' OVERFLOW?
001532	001401				BEQ	,+4	
001534	020000				HLT		JERROR! A 'RED' OVERFLOW OCCURRED WHEN
001536	012706				MOV	R2,SP	IA 'YELLOW' ADDRESS WAS REFERENCED,
001542	020337	000000			CLR	#0	JREPOINT STACK POINTER
001544	012767	001562	176232		MOV	#LIMC,ERRVEC	JREPOINT OVERFLOW VECTOR
001552	016666	177730	177730		MOV	-50(6),-50(6)	JREFERENCE LAST 'YELLOW' ADDRESS
001560	020000				HLT		JERROR! SHOULD HAVE TRAPPED
001562	020337	000000		LIMC1	TST	#0	JWAS IT A 'RED' OVERFLOW
001566	021401				BEQ	,+4	
001570	000000				HLT		JERROR! A 'RED' OVERFLOW OCCURRED WHEN
001572	020337	000000			CLR	#0	JTHE LAST 'YELLOW' ADDRESS WAS REFERENCED
001576	012706				MOV	R2,SP	JCLEAR ADDRESS 0
001602	012767	001622	176170		MOV	#LIMD,ERRVEC	JREPOINT THE STACK POINTER
001606	016267	177720	176674		MOV	-52(2),TEMP	JREPOINT THE OVERFLOW TRAP
001614	020166	177720			COM	-52(6)	JGT 'RED' LOCATION
001622	000000			LIMCC1	HLT		JREFERENCE 'RED' ADDRESS
001622	020267	177720	176660	LIMD1	CMP	-52(2),TEMP	JERROR! NO OVERFLOW TRAP WHEN 'RED' ADDRESS
001630	001401				BEQ	,+4	JWAS REFERENCED
001632	000000				HLT		JWAS INSTRUCTION ABORTED?
001634	016762	176650	177720		MOV	TEMP,-52(2)	JERROR! A 'RED' OVERFLOW OCCURRED WHEN
001642	020706				TST	X6	JTHE LAST 'YELLOW' ADDRESS WAS REFERENCED
001644	001401				BEQ	,+4	JRESTORE 'RED' LOCATION IN ANY EVENT
001646	020000				HLT		JWAS STACK POINTER CHANGED TO 0?
001650	022737	001620	000000		CMP	#LIMCC,#0	J'RED' OVERFLOW DID NOT ASSUME NEW STACK
001656	001401				BEQ	,+4	IS RETURN ADDRESS ON NEW STACK?
001660	000000				HLT		JERROR! RETURN ADDRESS NOT SAVED ON NEW
001662	000000				CLR	@SLR	JSTACK
001666	016706	176610			MOV	SPBOT,SP	JGET READY TO GET NEW VALUES FOR TEST
001672	013462	177770			MOV	R4,-2(2)	JINITIALIZE THE STACK POINTER
001676	013962	177774			MOV	R5,-4(2)	JRESTORE STACK ADDRESS DATA
001702	000000				BR	T5A	
001704	012767	000000	176872	LIMX1	MOV	#6,ERRVEC	JGET NEW VALUES
							JRESTORE TIME OUT TRAP VECTOR
001712	000267	176562		END1	INC	ICNT	
001716	026727	176550	002000		CMP	ICNT,#2000	JHAVE 2000 PASSES BEEN COMPLETED
001724	001402				BEQ	DONE	
001726	000167	177074			JMP	BEGIN	
001732	012767	000000	175620	DONE1	MOV	#7,TPB	JRING BELL
001740	000767	175620			TSTB	TPS	
001744	000375				BPL	,+4	
001746	013702	000042			MOV	#42,X2	JGET DECTAPE MONITOR RETURN ADDRESS
001752	001404				BEQ	DONE1	JDO NOT RETURN IF (42)=0
001754	004712				JSR	7,(2)	JRETURN TO DECTAPE MONITOR
001756	000240				NOP		JACT11



BEGIN	001020	DISPLA	= 177570	DONE	001732	DONE1	001764
END	001712	EMRVEC	= 000004	HLT	= 000000	ICNT	000500
LIMA	001504	LIMB	001520	LIMC	001562	LIMCC	001620
LIMD	001622	LIMX	001704	PC	=%000007	POP2	= 022626
PRTY7	= 000340	PSW	= 177776	R0	=%000000	R1	=%000001
R2	=%000002	R3	=%000003	R4	=%000004	R5	=%000005
R6	=%000006	SCOPE	= 010701	SLM	000506	SLR	000504
SP	=%000000	SPOOT	000402	START	001010	SWR	= 177570
TEMP	000010	TORAY	001074	TPB	= 177566	TPS	= 177564
T3	001076	T8A	001122	T8B	001130	T8C	001146
T8D	001154	T1	001104	T1A	001172	T2	001230
T2A	001242	T3	001270	T4	001330	T5	001410
T5A	001422	T5B	001442	UBREAK	= 177770	.	= 001770

ERRORS DETECTED: 0

6

001760	000240			NOP		IOVERLAY
001762	000240			NOP		IAREA
001764	000167	177020	DONE11	JMP	START	IRESTART
	000001			END		

6