

IDENTIFICATION

-----

PRODUCT CODE:	MAINDEC-11-D50A
PRODUCT NAME:	RF11 DISK DATA
DATE CREATED:	1 NOVEMBER 1970
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	EARL HAIGHT

TABLE OF CONTENTS

<u>SECTION</u>	<u>CONTENTS</u>
1.	ABSTRACT
2.	REQUIREMENTS
2.1	EQUIPMENT
2.2	STORAGE
3.	LOADING PROCEDURE
3.1	METHOD
4.	STARTING PROCEDURE
4.1	WORST CASE OPERATION
4.2	OPERATOR INTERVENTION
4.2.1	CHANGE DISK OPERATING PARAMETERS
4.3	SCOPE LOOP ENTRY POINTS
5.	OPERATING PROCEDURE
5.1	CONTROL SWITCH SETTINGS
5.2	CONVERSATION MODE
5.3	SUBROUTINE ABSTRACT
6.	ERROR REPORTS

1. ABSTRACT

THE RF11 DISK DATA TEST IS A SERIES OF STATIC, ADDRESS AND DATA RELIABILITY ROUTINES WHICH VERIFY TO THE USER THE DISK CONTROL (RF11) AND DISK (RS11) ARE OPERATING CORRECTLY. THIS TEST USED IN CONJUNCTION WITH THE RF11 DISKLESS AND RF11 MULTI DISK ASSURES THE USER OF AN ERROR FREE SYSTEM, WHEN USED IN ITS ENTIRETY.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11/20  
RF11 AND RS11

2.2 STORAGE

STATIC TEST  
PROGRAM OCCUPIES FROM 0 TO 11134

DATA TEST  
MAIN BODY OF PROGRAM OCCUPIES FROM LOC 0 TO 13476  
WRITE OUTPUT BUFFER OCCUPIES FROM LOC 13500 TO 15476  
READ INPUT BUFFER OCCUPIES FROM LOC 15500 TO 17476

3. LOADING PROCEDURE

3.1 METHOD OF LOADING BOTH STATIC AND DATA TEST TAPES

PROGRAM FORMAT ABSOLUTE

- A. VERIFY THE BOOT LOADER IS IN MEMORY.
- B. SET SWITCH REGISTER EQUAL TO \*750

MEMORY SIZE \*

4K	17
8K	37
12K	57
16K	77
20K	117
24K	137
28K	157

C. DEPRESS LOAD ADDRESS.

D. DEPRESS START.

4. STARTING PROCEDURE

4.1 WORST CASE DISK TEST UNIT ZERO FOR BOTH STATIC AND DATA TEST

- A) SET SWITCH REGISTER EQUAL TO 200
- B) DEPRESS LOAD ADDRESS
- C) SET SWITCH REGISTER EQUAL TO ZERO
- D) DEPRESS START

4.2 OPERATOR INTERVENTION FOR DATA TEST ONLY.

- A) SET SWITCH REGISTER EQUAL TO 200
- B) DEPRESS LOAD ADDRESS
- C) SET SWITCH REGISTER EQUAL TO MODE OF OPERATION (REF. SECS.)
- D) DEPRESS START.

## ADDRESS ENTRY POINTS FOR TEST ROUTINES

\*\*\*STATIC ROUTINES\*\*\*

\*\*\*ROUTINES IN STATIC TEST ONLY\*\*\*

300	JMP	STAI1	TEST RESET TO CONTROL REGISTER
304	JMP	STAI2	TEST RESET TO CURRENT ADDRESS REGISTER
310	JMP	STAI3	TEST RESET TO WORD COUNT REGISTER
314	JMP	STAI4	TEST RESET TO DISK ADDRESS REGISTER
320	JMP	STAI5	TEST RESET TO DISK EXT, ADDRESS REGISTER
324	JMP	STAI6	TEST RESET TO DATE BUFFER REGISTER
330	JMP	STAI7	TEST RESET TO MAINTENANCE REGISTER
334	JMP	STAI10	CAN WE SET W/R BITS IN DCS REGISTER
340	JMP	STAI11	CAN WE CLEAR THE DCS REGISTER USING DISK CLEAR,
344	JMP	STAI12	CAN WE SET ALL CMA BITS
350	JMP	STAI13	CAN WE CLEAR ALL CMA BITS USING DISK CLEAR
354	JMP	STAI14	CAN WE SET ALL WC BITS
360	JMP	STAI15	CAN WE CLEAR ALL WC BITS USING DISK CLEAR
364	JMP	STAI16	CAN WE SET ALL THE DAR BITS
370	JMP	STAI17	CAN WE CLEAR ALL THE DAR BITS USING DISK CLEAR
374	JMP	STAI20	CAN WE SET ALL THE DAE BITS
4 0	JMP	STAI21	CAN WE CLEAR ALL THE DAE BITS, USING DISK CLEAR
4 4	JMP	STAI22	EXECUTE A ONE WORD WRITE FOLLOWED BY A ONE WORD WRITE CONTINUE
410	JMP	STAI40	EXECUTE A ONE WORD WRITE CHECK FOLLOWED BY A ONE WORD WRITE CHECK CONTINUE
414	JMP	STAI50	EXECUTE A ONE WORD READ FOLLOWED BY A ONE WORD READ CONTINUE
420	JMP	STAI74	TEST TRACK INCREMENT
424	JMP	STAI77	TEST DISK INCREMENT
430	JMP	STAI09	TEST THAT NED RAISES ERROR FLAG
434	JMP	ST105X	CHECK CMA INHIBIT
440	JMP	NXMTSM	TEST NON-EXISTENT MEMORY ERROR
444	JMP	STAI06	TEST THAT THE DISK WILL NOT TRAP AT PRIORITY 7
450	JMP	STAI10	TEST THAT THE DISK WILL NOT TRAP AT PRIORITY 6
454	JMP	STAI12	TEST THAT THE DISK WILL NOT TRAP AT PRIORITY 5
460	JMP	STAI14	TEST THAT THE DISK WILL TRAP AT PRIORITY 4
464	JMP	ADT1	ADDRESS TEST 1 CHECK TIMING BY EXECUTING A ONE WORD WRITE
470	JMP	ADT2X	ADDRESS TEST 2 CHECK TIMING BY EXECUTING A ONE WORD READ

\*\*\*ROUTINES IN DATA TEST ONLY\*\*\*

600	JMP	ADT2	CHECK THAT WHEN A WORD IS WRITTEN IT DOES NOT ALTER ADJACENT WORDS
604	JMP	ADT3	WRITE EACH WORD ADDR ON ITSELF AND READ BACK TO COMPARE
610	JMP	ADT4	TRACK SELECTION TEST
614	JMP	ADT5	LOOK AHEAD TEST
620	JMP	SPIRAL	SPIRAL TEST
624	JMP	XSPIRAL	SPIRAL TEST EXTENSION
630	JMP	RANEX	RANDOM ADDRESS, DATA AND WORD COUNT TEST
634	JMP	EXTMEN	DISK EXT, MEMORY EXERCISER
640	JMP	PFT1	DISK WRITE POWER FAIL TEST
644	JMP	PFT2	DISK WRITE CHECK POWER FAIL TEST

\*\*\*MAINTENANCE ROUTINES\*\*\*

ROUTINES IN STATIC AND DATA TEST BOTH

650	JMP	SELWC	LOAD WORD COUNT REG, WITH SWITCH REGISTER
654	JMP	SELCMA	LOAD CURRENT MEMORY ADDR REG, WITH SWITCH REGISTER
660	JMP	SELDAR	LOAD DISK ADDR, REGISTER WITH SWITCH REGISTER
664	JMP	SELDAE	LOAD DISK ADDR, EXT, WITH SWITCH REGISTER
670	JMP	SELDBR	LOAD DATA BUFFER REGISTER WITH SWITCH REGISTER
674	JMP	MOVLK	MOVE CONTENTS OF LOOK AHEAD REGISTER INTO DATA LIGHTS
700	JMP	SELDCS	LOAD DISK CONTROL REGISTER WITH SWITCH REGISTER
704	JMP	STAMP	ENABLE READ AMPLIFIERS TO TRACK SELECTED

5. OPERATING PROCEDURE

5.1 CONTROL SWITCH SETTINGS

PROGRAM CONVERSATION

SR15 SET ENTER PROGRAM CONVERSATION MODE  
REF, SEC. 5.2  
RESET OPERATE WORSE CASE DISK ZERO

DELETE TYPEOUT

SR14 SET DELETE TYPEOUTS  
RESET REPORT MESSAGE

HALT ON COMPLETION FLAG

SR13 SET HALT ON FLAG (READY)  
RESET EXECUTE NEXT OPERATION

DELETE COMPARISONS

SR12 SET DELETE DATA COMPARISONS  
RESET COMPARE DATA BUFFERS

LOOP ON TEST

SR11 SET LOOP ON TEST  
RESET CONTINUE TO NEXT TEST

HALT ON ERROR

SR10 SET HALT AFTER ERROR REPORT  
RESET CONTINUE AFTER ERROR REPORT

IDENTIFICATION

PRODUCT CODE: MAINDEC 11-05AA  
PRODUCT NAME: RF11 MULTI DISK  
DATE CREATED: 1 NOVEMBER 1970  
MAINTAINER: DIAGNOSTIC GROUP  
AUTHOR: E. HAIGHT



1. **ABSTRACT**

MULTI DISK IS A HIGH SPEED CONFIDENCE TEST THAT ASSURES THE USER THAT HE CAN TRANSFER DATA CORRECTLY, WITHOUT DESTROYING THE DATA ON THE DISK. MULTI DISK USES ALL EXISTING MEMORY ON THE SYSTEM AS BUFFER AREAS.

2. **REQUIREMENTS**

A. PDP-11 4K TO 28K OF MEMORY

TELETYPE

B. RF11 AND RS11 PLUS UP TO SEVEN ADDITIONAL RS11'S

2.2 **STORAGE**

THE MAIN BODY OF THE PROGRAM OCCUPIES THE FIRST 5K OCTAL (BYTES) OF MEMORY. THREE DATA BUFFERS OCCUPY THE REST OF EXISTING MEMORY.

BUFFER ORDER

OUT BUFFER = RANDOM DATA TO BE WRITTEN ON THE DISK,

IN BUFFER = AREA FOR RANDOM DATA WHEN READ FROM THE DISK,

SAVE BUFFER = AREA TO SAVE DISK WHEN EXERCISING,

3. **LOADING PROCEDURE**

3.1 **METHOD**

PROGRAM FORMAT ABSOLUTE

A. VERIFY THE BOOT LOADER IS IN MEMORY.

B. LOAD TAPE INTO READER.

C. SET SWITCH REGISTER EQUAL TO \*750

MEMORY SIZE \*

4K	17
8K	37
12K	57
16K	77
20K	117
24K	137
28K	157

D. DEPRESS LOAD ADDRESS.

E. DEPRESS START.

4. STARTING PROCEDURE

- A. SET SWITCH REGISTER EQUAT TO 200.
- B. DEPRESS LOAD ADDRESS.
- C. DEPRESS START.
- D. THE PROGRAM AT THIS POINT WILL DETERMINE THE AMOUNT OF EXISTING CORE MEMORY, AND SET UP THE WORD COUNT AND BUFFER AREAS FOR THE PROGRAM.
- E. THE PROGRAM WILL NEXT REPORT THE NUMBER OF EXISTING DISK(S) WHICH ARE ON THE SYSTEM, THE NUMBER WILL RANGE FROM 1 - 10 OCTAL. THE DISK SELECTION LINES MUST BE IN SEQUENTIAL ORDER RANGING FROM 0 TO 7,

06 EXISTENT DISK(S)

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

THE SWITCH REGISTER HAS NO CONTROL OVER THE OPERATION OF THIS PROGRAM.

5.2 STOPPING THE PROGRAM

MULTI DISK MUST ONLY BE STOPPED BY TYPING CONTROL C. IF THE USER STOPS THE PROGRAM BY DEPRESSING CONSOLE HALT, THE DISK SURFACE MAY NOT BE RESTORED.

5.3 PROGRAM ABSTRACT

MULTI DISK WAS DESIGNED TO INSURE THE USER THAT THE DISK SYSTEM IS CAPABLE OF TRANSFERRING DATA CORRECTLY WHILE NOT DESTROYING THE USERS PROGRAMS ON THE DISK SURFACE. THE PROGRAM FIRST READS FROM THE DISK, THE LENGTH OF THE TRANSFER IS DETERMINED BY THE SIZE OF MEMORY. IF AN ERROR OCCURS WHILE READING, THE PROGRAM WILL MAKE UP TO THREE ATTEMPTS AT READING THE DATA. IF THE ERROR STILL EXISTS, THE PROGRAM WILL THEN HALT. IF THE PROGRAM SUCCESSFULLY READS FROM THE DISK WITHIN THE THREE ATTEMPTS, IT WILL THEN GENERATE A RANDOM BUFFER, WRITE IT ON THE DISK, AND READ IT BACK AND VERIFY IT. AFTER COMPARING THE DATA, THE PROGRAM THEN WRITES THE ORIGINAL DATA BACK ON THE DISK, MAKING UP TO THREE ATTEMPTS TO TRANSFER IF AN ERROR IS ENCOUNTERED, BEFORE HALTING. IF THE DATA WAS SUCCESSFULLY TRANSFERRED, THE PROGRAM WILL GO TO THE NEXT DISK BUFFER UNTIL THE COMPLETE DISK SYSTEM IS EXERCISED.

NOTE: EACH WRITE IS FOLLOWED BY A WRITE CHECK.

5. ERR(

.1 ERROR HALTS

THE PROCESSOR HALTED AT LOC. 1316. CAUSE OF HALT, NO BUS ERROR OCCURRED WHEN ADDRESSING NON-EXISTENT MEMORY.

THE PROCESSOR HALTED AT LOC. 1434. CAUSE OF HALT NO SELECTED DISK ON THE SYSTEM, OR DISKS NOT SELECTED IN SEQUENTIAL ORDER STARTING WITH DISK 0.

THE PROCESSOR HALTED AT LOC. 1750. THE CAUSE OF THE HALT, UNABLE TO READ FROM THE DISK FOR THE SAVE BUFFER, THREE ATTEMPTS WERE MADE.

THE PROCESSOR HALTED AT LOC. 2142. THE CAUSE OF THE HALT, UNABLE TO RESTORE THE DISK SURFACE. THREE ATTEMPTS WERE MADE.

.2 ERROR REPORTS

STATUS ERROR

STATUS ERROR XXDAE, XXXXXDAR, XXXXXDCS  
                  A          B          C

A=THE DISK NUMBER AND EXTENDED DISK ADDRESS BITS.  
B=THE DISK ADDRESS REGISTER  
C=THE DISK CONTROL REGISTER

LAYOUT OF DISK ADDRESS BITS  
          DAE          DAR  
XXX XXX XXX XXX XXX XXX XXX

DISK NO. TRACK ADDRESS WORD ADDRESS

BIT LAYOUT OF DCS REGISTER

BIT15= ERROR  
 BIT14= DISK FREEZE  
 BIT13= WRITE CHECK ERROR  
 BIT12= DATA PARITY ERROR  
 BIT11= NON-EXISTENT DISK  
 BIT10= WRITE LOCKOUT  
 BIT9= MISSED TRANSFER  
 BIT8= DISK CLEAR  
 BIT7= READY  
 BIT6= INTERRUPT ENABLE  
 BIT5= EXTENDED MEMORY 1 (XM1)  
 BIT4= EXTENDED MEMORY 0 (XM0)  
 BIT3= MAINTENANCE  
 BIT2-1= FUNCTION REGISTER

BIT 2	BIT 1	OPERATION
0	0	NOP
1	0	READ
0	1	WRITE
1	1	WRITE CHECK

BIT0= GO (WRITE ONLY BIT)



3.

PROGRAM DESCRIPTION

MULTI DISK IS NOT A DIAGNOSTIC - IT IS MERELY A CONFIDENCE TEST. IF THE USER ENCOUNTERS ANY PROBLEMS WITH THE SYSTEM WHILE OPERATING MULTI DISK, HE SHOULD NOT TRY TO DIAGNOSE THE PROBLEM USING THIS PROGRAM. IF PROBLEMS ARE ENCOUNTERED, IT IS SUGGESTED THAT THE USER REVERTS TO DISK DATA AS A MEANS OF DIAGNOSING HIS PROBLEM.



001000

.EVEN

]  
] I/O ADDRESS POINTERS

001000 177570  
001002 177776  
001004 177566  
001006 177562  
001010 177564  
001012 177560

SRI: 177570  
CSR: 177776  
TPB: 177566  
TKB: 177562  
TPS: 177564  
TKS: 177560

] SWITCH REGISTER  
] PROCESSOR STATUS REGISTER  
] TELETYPE REGISTERS

] DISK I/O REGISTERS

001014 177460  
001016 177462  
001020 177464  
001022 177466  
001024 177470  
001026 177472  
001030 177474  
001032 177476  
001034 000204  
001036 000206  
001040 000208

DCS: 177460  
NCI: 177462  
CMA: 177464  
DAR: 177466  
DAE: 177470  
DBR: 177472  
MAI: 177474  
ADS: 177476  
VECTOR: 204  
STATUS: 206  
PRIORITYBIT7

] DISK CONTROL REGISTER  
] WORD COUNT REGISTER  
] CURRENT ADDRESS REGISTER  
] LOWER 16 BITS OF DISK ADDRESS  
] EXTENSION ADDRESS REGISTER  
] DATA BUFFER REGISTER  
] MAINTENANCE REGISTER  
] LOOK AHEAD REGISTER  
] INTERRUPT VECTOR ADDRESS  
] DISK INTERRUPT STATUS  
] DISK PRIORITY LEVEL

]  
]  
]  
]  
]  
]  
]  
] IRF11 DEDICATE REGISTERS (MEMORY)

001042 000000  
001044 146723  
001046 000000  
001050 000000  
001052 000000  
001054 000000  
001056 000000  
001060 000000  
001062 000000  
001064 000000  
001066 000000  
001070 000000  
001072 000000  
001074 000000  
001076 000000  
001100 000000  
001102 000000  
001104 000000  
001106 000000  
001110 000000  
001112 004446  
001114 000000

FLAG: 0  
RANNU: 146723  
WRDCY: 0  
TRACK: 0  
DMA: 0  
PATNU: 0  
BUF: 0  
TWROCT: 0  
TDMAI: 0  
SHRDCY: 0  
ERCOUNT: 0  
SAVE: 0  
SAV1: 0  
PASS: 0  
DSKNOR: 0  
HRDR: 0  
HIGH: 0  
LIMIT: 0  
TTRACK: 0  
TDSKNOR: 0  
OUTBUF: BUFFER  
INBUF: 0

] INTERNAL PROGRAM FLAG  
] RANDOM NUMBER PRIME  
] WORKING WORD COUNT  
] WORKING DAE  
] WORKING DATA BUFFER (OUT-IN)  
] DATA PATTERN INDEX  
] WORKING DATA BUFFER (OUT-IN)  
] TEMP WORD COUNT  
] TEMP DAR  
] STANDARD WORD COUNT  
] ERROR COUNT FOR MESSAGES,  
  
] POINTER FOR HARD ERROR  
  
] TEMP TRACK NO.  
] TEMP DISK NO.



PALX11 V003 19-NOV-70

15104 PAGE 2-1

001116 000000

SAVBUF: 0

;

IRF11 WORK REGISTERS

;(CAN BE CHANGED IN ANY ROUTINE)

001120 000000

WORK1: 0

001122 000000

WORK1: 0

001124 000000

WORK2: 0

```

001126 000005          START: RESET          ICLEAR THE WORLD
001130 012706 001000      MOV          #1000,%6      ISET UP STACK
001134 012767 000006 176642  MOV          #6,4
001142 005067 176640      CLR          6
001146 012777 002270 177660  MOV          #DKINT,@VECTOR  ISET UP DISK POINTER
001154 012777 000340 177654  MOV          #340,@STATUS    ILOCK UP INTERRUPTS
001162 012767 003610 176640  MOV          #EMTRP,30      ISET UP TTY POINTER
001170 012767 000340 176634  MOV          #340,32      ILOCK UP INTERRUPTS
001176 012767 002216 176630  MOV          #DISK,34      ISET UP DISK HANDLER POINTER
001204 012767 000340 176624  MOV          #340,36      ILOCK UP INTERRUPTS
001212 012777 000340 177562  MOV          #340,@CSR      ILOCK UP INTERRUPT LEVELS
001220 005067 177616      CLR          FLAG          ICLEAR PROGRAM FLAG
001224 005067 177620      CLR          TRACK        ICLEAR TRACK REGISTERS
001230 005067 177616      CLR          DMA          ICLEAR DAR REGISTERS
001234 005067 177614      CLR          PATNU        ICLEAR PATTYR COUNT
001240 012767 003464 176552  MOV          #XWAIT,20
001246 005067 176550      CLR          22

;THIS ROUTINE DETERMINES HOW MUCH MEMORY IS ON THE SYSTEM
;
001252 012777 000340 177522  EXTMEM: MOV          #340,@CSR      ILOCK UP PRIORITY LEVELS
001260 012767 001332 176516      MOV          @MAXREF,4     ISET UP PROCESSOR TRAP
001266 012767 000340 176512      MOV          #340,6
001274 012767 017446 177566      MOV          #17446,SAVE   ISET UP REFERENCE TO 4K
001302 000777 177562      EXREF: TST          @SAVE   IREFERENCE MEMORY
001306 022767 177446 177554      CMP          #177446,SAVE  ITEST FOR GREATER THAN 20K
001314 001002          BNE          ,+6
001316 000000          HALT          IPROCESSOR ERROR WITH BUSS TRAP
001320 000777          BR
001322 062767 000000 177540      ADD          #20000,SAVE   ISET UP FOR NEXT REFERENCE
001330 000764          BR          EXREF       IREFERENCE NEXT BANK
001332 012706 001000      MAXREF: MOV          #1000,%6  IRESET STACK
001336 162767 020000 177524      SUB          #20000,SAVE
001344 032767 002000 177470      BIT          #BIT10,FLAG

;ROUTINE TO REPORT THE NUMBER OF EXISTENT DISK(S)
;ON THE SYSTEM.
001352 005067 177542      REPN: CLR          WORK
001356 016777 177536 177440      INDSK: MOV          WORK,@DAE  ILOAD DISK EXT. ADDR,BITS
001364 032777 004000 177422      BIT          #BIT11,@DCS   ITEST FOR NON-EXISTENT DISK
001372 001010          BNE          NEDST       IBRANCH IF NON EXISTENT DISK SET
001374 062767 000004 177516      ADD          #4,WORK      ISET UP NEXT DISK NO.
001402 022767 000040 177510      CMP          #40,WORK     ICHECK FOR 8 DISKS
001410 001401          BEQ          ,+4        IBRANCH IF 8 DISKS
001412 000761          BR          INDSK

```

001414	006067	177500		NEDST:	ROR	WORK	
001420	006067	177474			ROR	WORK	
001424	032767	000017	177466		BIT	#17,WORK	
001432	001002				BNE	,+6	
001434	000000				HALT		
001436	000777				BR		
001440	004567	002332			JSR	%5,CONV	
001444	001120				WORK		
001446	004351				MES11		
001450	000002				2		
001452	104000				EMT	>0	
001454	004167				HEU5A		
001456	004351				MES11		
001460	177777				-1		
001462	005367	177432			DEC	WORK	
001466	006167	177426			ROL	WORK	
001472	006167	177422			ROL	WORK	
001476	016767	177416	177372		MOV	WORK,DSKNOR	
					;ROUTINE TO SET UP WRITE/READ BUFFERS		
					;		
					;		
001504	106767	177402	177356		SUB	OUTBUF,SAVE	
001512	005067	177402			CLR	WORK	
001516	102767	000003	177344	SETBUF:	SUB	#3,SAVE	;DIVIDE MEMORY BY 3
001524	022767	000003	177336		CMP	#3,SAVE	
001532	101003				BHI	OVERFLO	;BRANCH IF NO REMAINDER
001534	005267	177360			INC	WORK	;+1 COUNT
001540	008766				BR	SETBUF	
001542	042767	000001	177330	OVERFLO:	BIC	#BIT0,WORK	
001550	016767	177336	177336		MOV	OUTBUF,INBUF	;SET UP IN-BUFFER
001556	066767	177336	177330		ADD	WORK,INBUF	;
001564	016767	177324	177324		MOV	INBUF,SAVBUF	;SET UP SAVE BUFFER
001572	066767	177322	177316		ADD	WORK,SAVBUF	
001600	016767	177314	177256		MOV	WORK,SWRDCT	;SET UP WORD COUNT
001606	000241				CLC		
001610	006067	177250			ROR	SWRDCT	;DIVIDE BYTESX2
001614	012767	003554	176236		MOV	#CNTLC,50	;SET UP TTY VECTOR
001622	012767	000340	176232		MOV	#340,62	
001630	052777	000100	177154		BIS	#BIT6,#TKS	;ENABLE TTY INTERRUPT
					;		
001636	016767	177222	177202	DATAT:	MOV	SWRDCT,WRDCT	;SET UP WORD COUNT
001644	012767	001636	177226		MOV	#DATAT,HRDR	;SET UP FOR HARD ERROR
001652	005067	177174			CLR	DMA	
001656	005067	177166			CLR	TRACK	
001662	012777	000340	177112	RDSAV:	MOV	#340,#DCSR	;LOCK UP PROCESSOR PRIORITY
001670	016767	177222	177160		MOV	SAVBUF,BUF	;SET UP CURRENT ADDR
001676	042767	000003	177136		BIC	#3,FLAG	

001704	042767	001000	177130	LDAT:	BIC	#BIT9,FLAG	
001712	005267	177124			INC	FLAG	
001716	104505				READ	+100	
001720	000004				IOT		IWAIT FOR FLAG IN BACKGROUND TEST
001722	032767	001000	177112		BIT	#BIT9,FLAG	ICHECK FOR ERROR
001730	001414				BEQ	WRDAY	IWRITE RANDOM DATE
001732	016767	177104	177160		MOV	FLAG,WORK	ITRY TO READ 3 TIMES
001740	042767	177774	177152		BIC	#177774,WORK	
001746	022767	000003	177144		CMP	#3,WORK	
001754	001353				BNE	LDAT	
001756	000000				HALT		IUNABLE TO READ FROM
001760	000777				BR	,	IDISK
001762	012767	002102	177110	WRDAT:	MOV	#WRSAB,WRDR	ISET UP FOR HARD ERROR
001770	016767	177116	177060		MOV	OUTBUF,BUF	ISET UP RANDOM BUFFER
001776	104503				WRITE	+100	
002000	000004				IOT		IWAIT FOR FLAG IN BACK GROUND
002002	042767	000003	177032		BIC	#3,FLAG	
002010	016767	177100	177040		MOV	INBUF,BUF	ISET UP CORRENT ADDR
002016	042767	001000	177016	RANRD:	BIC	#BIT9,FLAG	
002024	005267	177012			INC	FLAG	
002030	104505				READ	+100	
002032	000004				IOT		IWAIT FOR FLAG IN BACKGROUND
002034	004567	001176			JSR	#5,COMPARE	ICOMPARE DATA
002040	032767	001000	170774		BIT	#BIT9,FLAG	ICHECK FOR ERROR
002046	001415				BEQ	WRSAB	IRESTORE DISK BUFFER
002050	016767	170766	177042		MOV	FLAG,WORK	ICHECK FOR 3 RE-READS
002056	042767	177774	177034		BIC	#177774,WORK	
002064	022767	000003	177026		CMP	#3,WORK	
002072	001351				BNE	RANRD	IGG RE-READ DATE
002074	042767	000003	170740		BIC	#3,FLAG	
002102	042767	001000	170732	WRSAB:	BIC	#BIT9,FLAG	ICLEAR PROGRAM ERROR FLAG
002110	016767	177002	170740		MOV	SABUF,BUF	ISET UP CURRENT ADDRESS
002116	104503				WRITE	+100	
002120	000004				IOT		IWAIT IN BACKGROUND TEST
002122	104507				WRCHECK	+100	
002124	000004				IOT		IWAIT FOR FLAG
002126	032767	001000	170706		BIT	#BIT9,FLAG	ICHECK FOR ERROR
002134	001413				BEQ	STORBU	
002136	016767	170700	170754		MOV	FLAG,WORK	
002144	042767	177774	170746		BIC	#177774,WORK	
002152	022767	000003	170740		CMP	#3,WORK	
002160	001350				BNE	WRSAB	
002162	000000				HALT		
002164	005767	170652		STORBU:	TST	FLAG	
002170	100410				BMI	RESTORE	IRESTORE DISK AND EXIT
002172	004767	000342			JSR	X7,DISBUF	ISET UP NEXT DISK BUFFER
002176	000167	177460			JMP	RDSAB	IREAD NEXT BUFFER
002202	104001				EMT	+1	IREPORT END.
002204	004441				END		
002206	000167	177424			JMP	DATAT	IRECYCLE TEST
002212	000000			RESTORE:	HALT		
002214	000777				BR	,	IGG NOT RESTORE MEM



PALX11 V003 19-NOV-70 15104 PAGE 6-1

002420 104001

EMT+1

002422 004215

MES2

002424 032777 040000 176362

BIT #BIT14,#DCS

```

002432 001430      BEQ      SOFTER      ;GO AND CONTINUE SOFT ERROR
002434 017767 176364 176456      MOV      @DAE,WORK    ;FETCH ERROR EXT, BITS
002442 000367 176452      SWAB     WORK
002446 042767 177400 176444      BIC      #177400,WORK
002454 004567 001316      JSR      X5,CONV     ;CONVERT TO ASCII
002460 001120      WORK
002462 004231      MES2A
002464 000003      3
002466 104000      EMT+0
002470 004167      HED5A
002472 004231      MES2A
002474 177777      -1
002476 052777 000400 176310      BIS      #BITS,@DCS    ;CLEAR THE DISK
002504 012706 001000      MOV      #1000,%6
002510 000177 176364      JMP      @HDRR
002514 005777 176276      SOFTER: TST     @WC    ;EXIT HARD ERROR
002520 001404      BEQ      INTEXT     ;CHECK FOR X-FER DONE
002522 052777 000001 176264      DIS      #BITS,@DCS  ;EXIT FROM ROUTINE
002530 000002      RTI
002532 012706 000774      INTEXT: MOV     #774,%6 ;SET GO AND CONTINUE
002536 000002      RTI                ;RESET STACK
                                ;EXIT
;
;ROUTINE TO SETUP DISK BUFFERS
;ADD WORD COUNT TO STARTING DISK ADDRESSES
;COMPARE CALCULATED ADDRESS TO TERMINATING ADDRESS
;
002540 066767 176302 176304      DISBUF: ADD     WRDCT,DMA ;ADD WORD COUNT TO LOWER 16 BITS
002546 103002      BCC     COMDAR
002550 000267 176274      INC     TRACK
002554 026777 176272 176240      COMDAR: CMP     DMA,@DAR ;OVERFLOW ADD ONE TO TRACK
002562 001403      BEQ     CMDAE       ;COMPARE LOWER 16 BITS
002564 052767 000100 176250      BIS      #BITS,FLAG
;
CMDAE: MOV     @DAE,WORK1 ;FETCH EXT, ADDR BITS
002600 042767 177740 176314      BIC      #177740,WORK1 ;HASH TRACK AND DISK ADDR
002606 042767 177740 176234      BIC      #177740,TRACK
002614 026767 176230 176300      CMP     TRACK,WORK1  ;ARE THEY EQUAL
002622 001100      BNE     ERADR       ;ERROR IN DAE REG
002624 105767 176212      TSTB   FLAG         ;CHECK FOR LAST DISK BUFFER
002630 100015      BPL     EXTCME
002632 005067 176214      CLR     DMA         ;CLEAR LOWER 16 BITS
002636 005067 176206      CLR     TRACK      ;CLEAR EXT, ADDR, BITS.
002642 042767 000200 176172      BIC      #200,FLAG
002650 062716 000004      ADD     #4,(6)     ;INC STOCK POINTER
002654 016767 176204 176164      MOV     SWRDCT,WRDCT
002662 000457      BR      EXTDR      ;EXIT

```

002664	042767	177774	176230	EXTCMEI	BIC	#177774,WORK1	I MASK EXT. TRACK BITS
002672	022767	000003	176222		CMP	#3,WORK1	I COMPARE FOR LAST TRACK
002700	001033				BNE	AKH	I NOT LAST TRACK EXIT
002702	017767	176116	176210		MOV	@DAE,WORK	
002710	042767	177743	176202		BIC	#177743,WORK	
002716	026767	176176	176152		CMP	WORK,DSKNOR	
002724	001021				BNE	AKH	
002726	017767	176070	176164		MOV	@DAR,WORK	I FETCH LOWER 16 BITS OF ADDRESS
002734	066767	176106	176156		ADD	WRDCT,WORK	I WILL DISK OVERFLOW
002742	103012				BCC	AKH	
002744	052767	000200	176070		BIS	#200,FLAG	
002752	017767	176044	176066		MOV	@DAR,WRDCT	I DISK WILL OVERFLOW
002760	005167	176062			COM	WRDCT	I SET UP NEW WORD COUNT
002764	005267	176056			INC	WRDCT	I
002770	017767	176030	176052	AKNI	MOV	@DAE,TRACK	
002776	042767	177740	176044		BIC	#177740,TRACK	I MASK TRACK BITS
003004	017767	176012	176040		MOV	@DAR,DMA	I LOWER 16 BITS OF ADDRESS
003012	032767	000100	176022		BIT	%BIT6,FLAG	
003020	001001				BNE	,04	
003022	000007			EXTDR:	RTS	%7	I EXIT
003024	004067	000746		ERADR:	JSR	%5,CONV	I CONVERT DMA REG COUNT TO ASCII
003030	001052				DMA		
003032	004171				MES1		
003034	000006				6		
003036	004067	000734			JSR	%5,CONV	I CONVERT TRACK REG COUNT TO ASCII
003042	001050				TRACK		
003044	004205				MES1A		
003046	000002				2		
003050	104000				EMT	00	I REPORT ERROR
003052	004142				HED4		
003054	004205				MES1A		
003056	004171				MES1		
003060	177777				-1		
003062	000067	175764			CLR	DMA	I DISK ADDRESS ERROR RE-START PROG.
003066	000067	175756			CLR	TRACK	
003072	000000				HALT		



```

)
)
)
)ROUTINE TO FILL BUFFERS WITH RANDOM DATA
)
)
)ENTER FROM JSR X7,RANEX
)
)RANEX:
003074 016767 175746 176016      MOV      WRDCT,WORK      ;SET UP WORK
003102 012705 001112              MOV      #OUTBUF,%5     ;LOC. OF OUTBUFFER
)
)
)RANDOM DATA GENERATOR SUBROUTINE
DAYGEN: 003106 016700 000120      MOV      LONUM,%0       ;SET UP R0 WITH 5 DIGITS LOW
003112 016701 000116      MOV      HINUM,%1       ;SET UP R1 WITH 5 DIGITS HIGH
003116 012703 000007      MOV      #7,%3          ;SET UP SHIFT COUNT
003122 005002              CLR      %2              ;CLEAR R2
003124 006300      SHIFT:  ASI      %0        ;SHIFT R0 LEFT AND
003126 006101              ROL      %1              ;ROTATE CARRY INTO LSB OF R1 INTO
003130 006102              ROL      %2              ;ROTATE CARRY OUT OF R1 INTO R2
003132 005303              DEC      %3              ;DECREMENT R3
003134 001373              BNE     SHIFT           ;CONTINUE SHIFT LOOP
003136 006700 000070      ADD      LONUM,%0       ;ADDN IN NUMBER TO MAKE X 129
003142 005001              ADC      %1              ;PROPOGATE CARRY
003144 006701 000064      ADD      HINUM,%1       ;ADDN IN NUMBER TO MAKE X 129
003150 005002              ADC      %2              ;PROPOGATE CARRY
003152 002700 001057      ADD      #1057,%0       ;ADDN LOW CONSTANT
003156 005001              ADC      %1              ;PROPOGATE CARRIES
003160 005002              ADC      %2              ;PROPOGATE AGAIN
003162 002701 047401      ADD      #47401,%1      ;ADDN HIGH CONSTANT
003166 005002              ADC      %2              ;PROPOGATE CARRY
003170 002702 000006      ADD      #6,%2          ;ADDN HIGHEST CONSTANT
003174 000200              ADD      %2,%0          ;REPRIME R0 WITH HIGH DIGIT
003176 005001              ADC      %1              ;PROPOGATE CARRY
003200 010067 000026      MOV      %0,LONUM      ;PUT R0 BACK IN LONUM

```

```

003204 010025          MOV      X0,(5)+      ;HOLD LONUM FOR PROGRAM
003206 005367 175706    DEC      WORK
003212 001406          BEQ      EXGEN
003214 010167 000014    MOV      X1,HINUM    ;PUT R1 BACK IN HINUM
003220 010125          MOV      X1,(5)+      ;HOLD HINUM FOR PROGRAM
003222 005367 175672    DEC      WORK
003226 001327          BNE     DAYGEN
003230 000207          EXGEN: RTS      X7      ;RETURN TO PROGRAM
003232 000000          LONUM: 0
003234 000000          HINUM: 0
;
;
;DATA COMPARISON ROUTINE
;IF AN ERROR OCCURS BETWEEN THE OUT-BUFFER AND
;THE IN-BUFFER AN ERROR WILL BE REPORTED IN THE
;FOLLOWING MANNER
;DATA ERROR XX DAE XXXXXX DAR XXXXXX GOOD DATA XXXXXX BAD DATA
;
;
003236 005067 175652    COMPARE: CLR      WORK2      ;WORD COUNT
003242 016767 175644 175620  MOV      OUTBUF,SAVE    ;SET UP OUTBUFFER POINTER
003250 016767 175640 175614  MOV      INBUF,SAVE1   ;SET UP IN BUFFER POINTER
003256 027777 175606 175606  WRDCMP: CMP      @SAVE,@SAVE1 ;COMPARE BUFFERS
003264 001016          WDCMP: BNE     WDERR      ;WORD IN ERROR
003266 005267 175632    WRDINC: INC      WORK2      ;+1 WORD COUNT
003272 026767 175550 175624  CMP      WRDCY,WORK2    ;IS COMPLETE BUFFER CHECKED
003300 001407          BEQ      ADAM
003302 062767 000002 175560  ADD      #2,SAVE
003310 062767 000002 175554  ADD      #2,SAVE1
003316 000757          BR      WRDCMP
003320 000205          ADAM: RTS      X5      ;COMPARE NEXT WORD
;EXIT THIS ROUTINE
;
;
003322 052767 001000 175512  WDERR: BIS      #BIT9,FLAG ;SET ERROR BIT
003330 016767 175516 175562  MOV      DMA,WORK      ;FETCH STARTING DISK ADDR
003336 016767 175506 175556  MOV      TRACK,WORK1   ;
003344 066767 175534 175546  ADD      WORK2,WORK    ;CALCULATE FAILING ADDR
003352 103002          BCC     ,+6           ;SHOULD DAE BE INCREMENTED
003354 005267 175542    INC      WORK1
003360 004567 000412    JSR     X5,CONV      ;CONVERT WORD ADDR TO ASCII
003364 001120          WORK
003366 004171          MES1
003370 000006          6
003372 004567 000400    JSR     X5,CONV      ;CONVERT TRACK ADDR TO ASCII
003376 001122          WORK1
003400 004205          MES1A
003402 000002          2
003404 017767 175460 175506  MOV      @SAVE,WORK    ;FETCH GOOD DATA
003412 004567 000360    JSR     X5,CONV      ;CONVERT GOOD DATA TO ASCII
003416 001120          WORK
003420 004314          MES5
003422 000006          6
003424 017767 175442 175466  MOV      @SAVE1,WORK   ;FETCH BAD DATA
003432 004567 000340    JSR     X5,CONV      ;CONVERT TO ASCII
003436 001120          WORK

```

PALX11 V003 19-NOV-70 15104 PAGE 10-1

003440 004330  
003442 000006

MES6  
6

```

003444 104000          EMT      +0          IPRINT MESSAGE
003446 004060          HED1
003450 004205          MES1A
003452 004171          MES1
003454 004314          MES5
003456 004330          MES6
003460 177777          -1
003462 000701          BR      WRDINC          IGO COMPARE NEXT WORD

```

);  
);  
);  
);  
);  
);

BACKGROUND TEST FOR INTERRUPTS

```

003464 012767 003552 174322 XWAIT: MOV      #RTIX,14      ;SET UP TRACE TRAP
003472 005067 174320          CLR      16
003476 005067 175372          CLR      PASS          ;SET UP TIME BASE
003502 052777 000020 175272          BIS      #BIT4,%CSR    ;SET TRACE BIT
003510 003027 000000          CLR      #0
003514 005267 177772          XINCW: INC     XINCW-2
003520 105767 177766          YSTB   XINCW-2
003524 100373          BPL     XINCW
003526 005267 175342          INC     PASS
003532 001401          BEQ     ,+4
003534 000765          BR      XINCW-4
;REPORT BACKGROUND TEST TIMED OUT
003536 042777 000020 175236          BIS      #BIT4,%CSR    ;CLEAR TRACE BIT
003544 104001          EMT+1
003546 004401          TIMO
003550 000000          HALT
003552 000002          RTIX: RTI

```

);

```

)
)TEST FOR CONTROL "C"
)UPON RECEIPT OF CNTLC SET FLAG BIT
)

```

```

003554 017767 175226 175336 CNTLC: MOV @TKB,WORK IFETCH KEYBOARD BUFFER
003562 042767 000200 175330 BIC #BIT7,WORK IMASK FOR 33
003570 022767 000003 175322 CMP #3,WORK ITEST FOR CNTLC
003576 001003 BNE ,+10 INOT CNTLC EXIT
003600 052767 100000 175234 BIS #BIT15,FLAG ISET FLAG BIT
003606 000002 RTI IEXIT ROUTINE

```

```

)
)ROUTINE TO DECODE EMT CALLS
)EMT+1=TYPE ONE LINE OF TEXT
)EMT+0=TYPE A SERIED OF LINES
EMTRP: MOV (6),%0

```

```

003610 011600
003612 022740 104001 CMP #EMT+1,-(0) IWHAS THE CALL EMT+1
003616 001052 BNE TYP5 INOI TYPE A SERIES
003620 000400 BR TYP ITYPE ONE LINE

```

```

)SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER,
TYP1: MOV @%6,%0 IGET ADDRESS THAT CONTAINS MESSAGE ADDRESS
ADD #2,@%6 ISET UP EXIT.
MOV @%0,%0 IADDRESS OF MESSAGE TO R0.

```

```

003622 011600
003624 062716 000002 TYPFA: MOVB (0)+,TYPDAT IGET CHARACTER
003632 112067 000104 CMPB #100,TYPDAT ICHECK FOR "@" CHARACTER
003636 122767 000100 000076 BNE TYP5 ICRANCH IF NOT "@",
003644 001001 RTI ITERMINATOR CHAR. DONE. EXIT.
003646 000002 TYPG: CMPB #45,TYPDAT ICHECK FOR "%".
003650 122767 000045 000064 BEQ TYP5 IBRANCH IF "%".
003656 001416 CMPB #42,TYPDAT INOT "%", CHECK FOR "#".
003660 122767 000042 000054 BEQ TYP5 IBRANCH IF "#"
003666 001417 JSR %7,TYPD ITYPE CHAR IN TYPDAT
003670 004767 BR TYP5
003674 000756 TYPD: MOVB TYPDAT,@TPB IOUTPUT CHARACTER TO PRINTER
003676 116777 000040 175100 TSTB @TPS IWAIT FOR DONE FLAG.
003704 105777 175100 BPL ,+4
003710 100375 TYEXIT: RTS %7 IEXIT
003712 000207 TYPFI: MOVB #15,TYPDAT IMOVE CARRIAGE RETURN CODE TO TYPDAT
003714 112767 000015 000020 JSR %7,TYPD IGO TYPE CHAR.
003722 004767 177750 TYPGI: MOVB #12,TYPDAT IMOVE LF CODE TO TYPDAT.
003726 112767 000012 000006 JSR %7,TYPD IGO TYPE CHAR.
003734 004767 177736 BR TYP5
003740 000734 TYPDAT: 0
003742 000000

```

```

)SUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER
TYP5: MOV @%6,%0 IGET ADDRESS THAT CONTAINS MESSAGE ADDRESS
ADD #2,@%6 IUPDATE TO NEXT MESSAGE ADDRESS
MOV @%0,TYPSB IADDRESS OF MESSAGE TO TYPSB
CMP #-1,TYPSB ICHECK FOR TERMINATOR
BNE TYP5A IBRANCH IF NOT TERMINATOR.
RTI ITERMINATOR. EXIT

```

```

003744 011600
003746 062716 000002
003752 011067 000014
003756 022767 177777 000006
003764 001001
003766 000002

```

003770 104001  
 003772 000000  
 003774 000763

TYPSA: EMT \*1 ;CALL ON TYP SUB TO TYPE MESSAGE  
 TYPSB: 0 ;ADDRESS OF MESSAGE GOES HERE  
 BR TYP5 ;GO PROCESS NEXT MESSAGE

;  
 ;  
 ;  
 ;  
 ;OCTAL TO ASCII CONVERT ROUTINE  
 ;

;ENTER ROUTINE AS FOLLOWS  
 ;JSR%5,CONV  
 ;ADDR#=ADDRESS OF NUMBER TO BE CONVERTED  
 ;ADDR BYTE=LSB OF WHERE ASCII IS GOING  
 ;ASCII#=THE NUMBER OF ASCII CHAR, TO BE CONVERTED  
 ;  
 ;

003776 013567 000054  
 004002 012501  
 004004 012502  
 004006 060201  
 004010 016703 000042  
 004014 042703 177770  
 004020 052703 000060  
 004024 110341  
 004026 042767 000007 000022  
 004034 006067 000016  
 004040 006067 000012  
 004044 006067 000006  
 004050 005302  
 004052 001356  
 004054 000205

CONVI MOV 0(5)+,ACNVX ;VALUE OF # TO BE CONVERTED  
 MOV (5)+,%1 ;ASCII ADDR  
 MOV (5)+,%2 ;# OF ASCII CHAR  
 ADD %2,%1  
 ACVNI MOV ACNVX,%3  
 BIC #177770,%3 ;ISOLATE LEAST SIGNIFICANT OCTAL#  
 ADD #60,%3 ;SET UP ASCII#  
 MOVB %3,-(1) ;STORE ASCII CHAR  
 BIC #7,ACNVX  
 ROR ACNVX ;ROTATE OCTAL#  
 ROR ACNVX  
 ROR ACNVX  
 DEC %2 ;I-1 FROM ASCII CHAR COUNT  
 BNE ACVN  
 RTS %5 ;EXIT # CONVERTED

004056 000000

ACNVX: 0 IWORK REGISTER

;  
;  
;  
;  
;  
;  
;  
;

ERROR MESSAGE HEADERS

;

004060 045  
004061 104  
004062 101  
004063 124  
004064 101  
004065 040  
004066 105  
004067 122  
004070 122  
004071 117  
004072 122  
004073 040  
004074 100

HE01: ,EVEN  
,ASCII /XDATA ERROR #/

;

;

004075 045  
004076 123  
004077 124  
004100 101  
004101 124  
004102 125  
004103 123  
004104 040  
004105 105  
004106 122  
004107 122  
004110 117  
004111 122  
004112 040  
004113 100

HE02: ,ASCII /XSTATUS ERROR #/

;

;

004114 045  
004115 117  
004116 126  
004117 105  
004120 122  
004121 106  
004122 114

HE03: ,ASCII /XOVERFLO X-FER ERROR #/

004123	117
004124	040
004125	130
004126	055
004127	106
004130	105
004131	122
004132	040
004133	105
004134	122
004135	122
004136	117
004137	122
004140	040
004141	100

004142	045
004143	104
004144	111
004145	123
004146	113
004147	040
004150	101
004151	104
004152	104
004153	122
004154	105
004155	123
004156	123
004157	040
004160	105
004161	122
004162	122
004163	117
004164	122
004165	040
004166	100

```

|
| HED41 ,ASCII /XDISK ADDRESS ERROR 0/

```

004167	045
004170	100

```

|
| HED5A1 ,ASCII /X0/

```

```

|
|
|
|
| MESSAGE TRAILERS
|
|

```

004171	040
004172	040
004173	040

```

|
| MES11 ,ASCII / DAR 0/

```



004174 040  
004175 040  
004176 040  
004177 040  
004200 104  
004201 101  
004202 122  
004203 040  
004204 100

;  
;  
;  
MES1A1 ,ASCII / DAE #/

004205 040  
004206 040  
004207 040  
004210 104  
004211 101  
004212 105  
004213 040  
004214 100

;  
;  
;  
MES21 ,ASCII / DCS #/

004215 040  
004216 040  
004217 040  
004220 040  
004221 040  
004222 040  
004223 040  
004224 104  
004225 103  
004226 123  
004227 040  
004230 100

;  
MES2A1 ,ASCII / HARD ERROR#/

004231 040  
004232 040  
004233 040  
004234 040  
004235 040  
004236 040  
004237 040  
004240 110  
004241 101  
004242 122  
004243 104  
004244 040  
004245 105  
004246 122  
004247 122  
004250 117

004251 122  
004252 100

004253 040  
004254 040  
004255 040  
004256 040  
004257 040  
004260 127  
004261 117  
004262 122  
004263 104  
004264 040  
004265 103  
004266 117  
004267 125  
004270 116  
004271 124  
004272 100

;  
;  
MES31 ,ASCII / WORD COUNT#/  
;

004273 040  
004274 040  
004275 040  
004276 040  
004277 040  
004300 040  
004301 040  
004302 127  
004303 117  
004304 122  
004305 104  
004306 040  
004307 101  
004310 104  
004311 122  
004312 056  
004313 100

;  
;  
;  
MES41 ,ASCII / WORD ADR.#/  
;

004314 011  
004315 107  
004316 117  
004317 117  
004320 104  
004321 040  
004322 104  
004323 101  
004324 124

;  
;  
;  
MES51 ,ASCII / GOOD DATA #/  
;

004325 101  
004326 040  
004327 100

;  
;  
;  
MES61 ,ASCII / BAD DATA #/

004330 040  
004331 040  
004332 040  
004333 040  
004334 040  
004335 040  
004336 040  
004337 102  
004340 101  
004341 104  
004342 040  
004343 104  
004344 101  
004345 124  
004346 101  
004347 040  
004350 100

```

      |
      |
004351      040      MES111 ,ASCII / EXISTENT DISK(S),0/
004352      040
004353      040
004354      105
004355      130
004356      111
004357      123
004360      124
004361      105
004362      116
004363      124
004364      040
004365      104
004366      111
004367      123
004370      113
004371      050
004372      123
004373      051
004374      036
004375      100

```

```

      |
      |
004376      040      MES121 ,ASCII / 0/
004377      040
004400      100

```

```

      |
      |
004401      045      TIM01 ,ASCII /%PROCESSOR BACKGROUND TIMED OUT0/
004402      120
004403      122
004404      117
004405      103
004406      105
004407      123
004410      123
004411      117
004412      122
004413      040
004414      102
004415      101
004416      103
004417      113
004420      107
004421      122
004422      117
004423      125
004424      116
004425      104
004426      040

```

004427 124  
004430 111  
004431 115  
004432 105  
004433 104  
004434 040  
004435 117  
004436 125  
004437 124  
004440 100

004441 045  
004442 105  
004443 116  
004444 104  
004445 100

004446 004446  
004446 000000  
004446 000001

;  
;  
END: ,ASCII /XEND0/

BUFFER: 0 ,EVEN  
0 ,END

ACNVX	004056	HED1	004060	TDSKNO	001110
ACVN	004010	HED2	004075	TIMO	004481
ADAM	003320	HED3	004114	TKB	001006
ADS	001032	HED4	004142	TKS	001012
AKH	002770	HED5A	004167	TPB	001004
BIT0	000001	HIGH	001102	TPS	001010
BIT1	000002	HINUM	003234	TRACK	001030
BIT10	002000	HRDR	001100	TTRACK	001106
BIT11	004000	INBUF	001114	TWRDCT	001060
BIT12	010000	INOSK	001356	TYEXIT	003712
BIT13	020000	INTEXT	002532	TYP	003622
BIT14	040000	LDAT	001704	TYPA	003632
BIT15	100000	LIMIT	001104	TYPC	003650
BIT2	000004	LONUM	003232	TYPD	003676
BIT3	000010	MA	001030	TYPDAT	003742
BIT4	000020	MAXREF	001332	TYPF	003714
BIT5	000040	MES1	004171	TYPG	003726
BIT6	000100	MES11	004351	TYPS	003744
BIT7	000200	MES12	004376	TYPSA	003770
BIT8	000400	MES1A	004205	TYPSB	003772
BIT9	001000	MES2	004215	VECTOR	001034
BUF	001056	MES2A	004231	WC	001016
BUFFER	004446	MES3	004253	WDERR	003322
CMA	001020	MES4	004273	WORK	001120
CHDAE	002572	MES5	004314	WORK1	001122
CNTLC	003954	MES6	004330	WORK2	001124
COMDAR	002554	NEOST	001414	WRCHEC	104407
COMPAR	003236	OUTBUF	001112	WRDAY	001762
CONV	003776	OVERFL	001542	WRDCMP	003296
CSR	001002	PASS	001074	WRDCT	001046
DAE	001024	PATNU	001054	WRDINC	003266
DAR	001022	PRIORI	001040	WRITE	104403
DATAT	001636	RANEX	003074	WRSAY	002102
DAYGEN	003106	RANNU	001044	XINCW	003514
DBR	001026	RANRD	002016	XWAIT	003464
DCS	001014	RDSAV	001662		
DELMES	002344	READ	104405		
DISBUF	002540	REPON	001352		
DISK	002216	RESTOR	002212		
DKINT	002270	RTIX	003552		
DMA	001052	SAV1	001072		
DSKNOR	001076	SAVBUF	001116		
EMTRP	003610	SAVE	001070		
END	004441	SETBUF	001516		
ERADR	003024	SHIFT	003124		
ERCOUN	001066	SOFTER	002514		
EXGEN	003230	SR	001000		
EXREF	001302	START	001126		
EXTCME	002664	STATUS	001036		
EXTDR	003022	STORBU	002104		
EXTMEM	001252	SHRDCT	001064		
FLAG	001042	TDMA	001062		

RRORS DETECTED: 0

UN-TIME: 8 SECONDS

K CORE USED