

RCSL: 44-RT 726
RELEASED: 73082.

MTC 304 TEST

BINARY TAPE: RCSL: 44-RT 727

KEYWORDS: RC 3600, MTC 304, MAG. TAPE.

111

;VERSION: 03.11.71 JEP/PEH PREMINILARY, NOVA 1200

```
;
;      PLEASE READ
;      *****
;      THE DIAGNOSTIC PROGRAM IS PROVIDED
;      TO FIND HARDCORE PROBLEMS THAT ARE
;      RELATED TO BASIC OPERATIONS OF THE
;      TAPE CONTROL. THE RELIABILITY PRO-
;      GRAM IS PROVIDED AS A MORE STRIGENT
;      TEST TO SHOW UP INTERMITTENT AND
;      PATTERN SENSITIVE PROBLEMS
;      FOR THE MOST PART THE DIAGNOSTIC
;      ASSUMES THE MAGNETIC TAPE MEDIA
;      IS PERFECT AND NOT THE CAUSE OF AN
;      ERROR. HOWEVER, IF PARITY ERRORS
;      OCCUR WHILE RUNNING THE DIAGNOSTIC
;      THE TAPE SHOULD ALWAYS BE SUSPECT!
;      *****
;      *****
;
```

```

;MAGNETIC TAPE DIAGNOSTIC
;1.    POWER DOWN ALL TAPE UNITS EXCEPT ONE. FOR A COMPLETE
;      TEST MAKE SUCCESSIVE RUNS OF EACH
;      PE 1600 BPI AND NRZI 800 BPI
;      BY MEANS OF THE 4 PROGRAM
;      SECTIONS DESCRIBED BELOW.
;2.    STARTING LOCATIONS
;      400    BASIC DIAGNOSTIC
;      401    WRITE LOCK TEST
;      402    END TAPE TEST
;      403    DATA LATE TEST
;3.    SWITCH SETTINGS
;      13-15  UNIT # 0-7
;      7-12   DEVICE CODE.
;      4      PE =1  NRZI =0
;      -
;      2      PRINT FAILURE RATE IF "1"
;      1      INHIBIT PRINTING IF "1" ELSE PRINT ALL
;      0      FROM ERROR. GO TO NEXT TEST IF "1" ELSE LOOP
;      SWITCHES 0,1, AND 2 ARE ONLY EFFECTIVE DURING TEST 4.1
;4.    PROCEDURE
;      LOAD TAPE, PUSH FORWARD TO POSITION AT BOT.
;      SHORTCIRCUIT TESTPOINT 40 M TO GROUND, IN ORDER TO
;      AVOID IORST FROM GENERATING OFF-LINE.
;      LOAD PROGRAM WITH BINARY LOADER AND START AT
;      PROPER LOCATION FOR THE TEST TO BE RUN. ALL
;      TESTS SHOULD BE RUN FOR A COMPLETE CHECK. IF
;      A HALT OCCURS CONSULT THE PROGRAM LISTING
;      FOR DIRECTIONS
;      IMMEDIATELY AFTER STARTING A HALT WILL OCCUR
;      TO ALLOW THE SWITCH REGISTER TO BE SET FOR
;      PROPER UNIT SELECT, CONTROL, ETC...
;      AFTER SETTING SWITCHES PUSH CONTINUE.
;4.1   BASIC DIAGNOSTIC
;      EACH TEST MAY CONTAIN SEVERAL CHECK POINTS
;      WHERE AN ERROR HALT CAN OCCUR. GENERALLY
;      SPEAKING, ERRORS MAY BE EXPECTED AT CHECK
;      POINTS FURTHER ON WHEN ONE ERROR IS FOUND.
;      IF A MALFUNCTION IS DETECTED THE PROGRAM WILL
;      HALT AND AC3 WILL CONTAIN THE LOCATION OF
;      THE ERROR+1. CONSULT THE LISTING AND EXAMINE
;      OTHER PERTINENT ACCUMULATORS. IF SWITCHES
;      0&1 ARE DOWN, PRESSING CONTINUE WILL CAUSE A
;      PRINTOUT OF THE ERROR LOCATION AND A SCOPE
;      LOOP WILL BE ENTERED. WHILE IN THE SCOPE LOOP
;      SETTING SWITCH 2 WILL CAUSE THE FAILURE RATE
;      (IN %) TO BE PRINTED. SETTING SWITCH 0 WILL
;      CAUSE THE PROGRAM TO GO ON TO THE NEXT TEST.
;      EACH SUCCESSIVE TEST ASSUMES THAT ALL PREVIOUS
;      TESTS WORK CORRECTLY. BYPASSING A FAILIURE MAY CAUSE
;      IRRELEVANT ERRORMESSAGES IN LATER TESTS

;4.2   WRITE LOCK TEST
;      REMOVE WRITE ENABLE RING. MOUNT, AND POSITION
;      REEL AT BOT. START. AN ATTEMPT IS MADE TO

```

```

; WRITE, WRITE END OF FILE AND ERASE AND STATUS IS CHECK
; AFTER EACH ATTEMPT.
; EXPECTED STATUS IS 013000/017000
; ILLEGAL
; BOT
; WRITE LOCK
; (PE)
; IF THIS STATUS IS FOUND "WRITE LOCK TEST OK"
; IS PRINTED AND THE TEST IS REPEATED. IF THE
; ABOVE STATUS IS NOT FOUND AN ERROR HALT
; OCCURS. AC1=EXPECTED STATUS
; AC0=ACTUAL STATUS
; PUSH CONTINUE TO REPEAT TEST.
;4.3 END TAPE TEST
; FOLLOWING START, 4K RECORDS ARE WRITTEN DOWN
; THE LENGTH OF THE TAPE. WHEN END TAPE STATUS
; IS DETECTED "END TAPE" IS PRINTED AND PROGRAM
; HALTS. PUSHING CONTINUE WILL CAUSE THE
; PROGRAM TO GO ON.
; IF END TAPE IS NOT FOUND STATUS IS CHECKED FOR
; OFF LINE
; REWINDING
; BOT
; WRITE LOCK
; ILLEGAL
; EOF
; DATA LATE
; IF ANY OF THESE IS FOUND THE STATUS IS PRINTED
; AND WRITING CONTINUES.
;4.4 DATA LATE TEST
; FOLLOWING START THE TAPE IS REWOUND. ONE 4K
; RECORD IS THEN WRITTEN AND THE DATA LATE STATUS
; CHECKED. IF THE STATUS IS FOUND, "STATUS DATA LATE"
; IS PRINTED. IF NOT, "STATUS NO DATA LATE" IS
; PRINTED. THE TEST THEN REPEATS
; TO CAUSE A DATA LATE STATUS TO OCCUR, STOP THE
; PROGRAM DURING A WRITE OPERATION. THEN PUSH
; CONTINUE. THE LONG RECORD LENGTH ALLOWS PLENTY
; OF TIME TO INSURE THE PROGRAM STOP OCCURS DURING
; DATA CHANNEL TRANSFER.

```

111

```

000000 000000 .LOC 0
000001 000000 0 ;STORE PC ON INTERRUPT
000002 000002 2 ;GO TO .+1
000003 002000 ISZ 0 ;INTERRUPT RETURN WITH STIP
000004 000000 JMP 00
000020 000000 .LOC 20
000021 000000 IDX0: 0 ;AUTO INDEX INCREMENT REGISTERS
000022 000000 IDX1: 0
000023 000000 IDX2: 0
000024 000000 IDX3: 0
000025 000000 IDX4: 0
000050 000050 .LOC 50

;CONSTANTS
000050 000411 FIRST: A0
000051 003400 CEND: ENOPRG
000052 006400 CMTA: DIB 1,0
000053 004040 UBUFF: LASTPRG+10
000054 004240 IBUFF: LASTPRG+210
000055 000206 PONCX: CX
000056 000200 SAVE0: 0
000057 000200 SAVE1: 0

000012 000012 .RDX 10
000060 001750 .1SEC: 1000
000061 003720 .2SEC: 2000
000062 011610 .5SEC: 5000
000063 000764 .10SEC: 500
000064 047040 .20SEC: 20000
000065 000062 .50MS: 50
000066 000144 .100MS: 100
000067 000310 .200MS: 200
000070 000620 .400MS: 400
000010 000010 .RDX 8
000071 000001 C1: 1
000072 000002 C2: 2
000073 000003 C3: 3
000074 000004 C4: 4
000075 000005 C5: 5
000076 000006 C6: 6
000077 000007 C7: 7
00100 000010 C10: 10
00101 000017 C17: 17
00102 000020 C20: 20
00103 000030 C30: 30
00104 000040 C40: 40
00105 000050 C50: 50
00106 000060 C60: 60
00107 000070 C70: 70
00110 000077 C77: 77
00111 000100 C100: 100
00112 000110 C110: 110
00113 000144 C144: 144
00114 000150 C150: 150
00115 000170 C170: 170
00116 000200 C200: 200
00117 000375 C375: 375
00120 000377 C377: 377
00121 000400 C400: 400
00122 001000 C1000: 1000

```

00123	002000	C2000:	2000
00124	004000	C4K:	4000
00125	010000	C10K:	10000
00126	020000	C20K:	20000
00127	040000	C40K:	40000
00130	007700	C7700:	7700
00131	004104	C4104:	4104

111

00132	010421	C10421:	10421	
00133	021042	C21042:	21042	
00134	042104	C42104:	42104	
00135	004210	C421T:	4210	
00136	000201	C201:	201	
00137	160077	C160077:	160077	
00140	060000	C060000:	060000	
00141	177700	C177700:	177700	
00142	101000	C101000:	101000	
00143	025252	C25252:	25252	
00144	125252	C125252:	125252	
00145	052525	C52525:	52525	
00146	177776	M2:	-2	
00147	177775	M3:	-3	
00150	177774	M4:	-4	
00151	177773	M5:	-5	
00152	177700	M100:	-100	
00153	177000	M1000:	-1000	
	000012		.RDX 10	
00154	000764	D500:	500	
00155	177770	N8:	-8	
00156	177160	N400:	-400	
00157	175436	N1250:	-1250	
00160	000175	KNOVA:	125	
00161	000000	KNON1:	0	
00162	000341	K1200:	225	
00163	000567	KSU:	375	
00164	000764	KSUSC:	500	
00165	000567	K800:	375	
00166	000000	KNON2:	0	
	000010		.RDX 8	
			;PATTERN TABLE, ODD PARITY WRITES	
00167	000167	POINT:	.	
00170	000000		0	
00171	177400		177400	
00172	000377		377	
00173	125252		125252	
00174	052525		52525	
00175	000174	LST:	.-1	
			;PATTERN TABLE, EVEN PARITY WRITES	
00176	000176	PONTR:	.	
00177	125252		125252	
00200	052525		052525	
00201	000401		000401	
00202	020040		020040	
00203	000202	LST1:	.-1	

;VARIABLES

```

00204 000000 CONCX: 0
00205 000000 X170: 0 ;NOOP COMMAND (INCREMENTAL)
00206 000000 CX: 0 ;UNIT # +0 (READ)
00207 000010 C1X: 10 ;UNIT # +10 (WRITE)
00210 000020 C2X: 20 ;UNIT # +20 (SPACE 1 BLOCK FORW)
00211 000030 C3X: 30 ;UNIT # +30 (SPACE 1 BLOCK REV)
00212 000040 C4X: 40 ;UNIT # +40 (WRITE EOF)
00213 000050 C5X: 50 ;UNIT # +50 (ERASE 3.75 INCH)
00214 000060 C6X: 60 ;UNIT # +60 (NRZI CHECK NEXT TRACK)
00215 000070 C7X: 70 ;UNIT # +70 (REWIND)
00216 000100 C10X: 100 ;UNIT # +100 (GO OFFLINE)
00217 000110 C11X: 110 ;UNIT # +110 (NOOP)
00220 000120 C12X: 120 ;UNIT # +120 (NOOP)
00221 000130 C13X: 130 ;UNIT # +130 (NOOP)
00222 000140 C14X: 140 ;UNIT # +140 (NOOP)
00223 000150 C15X: 150 ;UNIT # +150 (NOOP)
00224 000160 C16X: 160 ;UNIT # +160 (NOOP)
00225 000170 C17X: 170 ;UNIT # +170 (USED NOOP)
00226 000000 CTR: 0 ;COUNTER
00227 000000 CPUNO: 0 ;CPU TYPE
00230 000000 WDCNT: 0 ;TEMP STORAGE
00231 000000 INST: 0 ; " "
00232 000000 RECTR: 0 ; " "
00233 000030 DEVICE: MTC ; DEVICE CODE
00234 000000 OLDDDEV: 0 ; PREVIOUS DEVICECODE

```

III

;INDIRECT ADDRESSES

00235	003024	ICKST:	CSTAT
00236	002352	INTIL:	INITIAL
00237	003065	ISTAT:	XSTAT
00240	003063	ISSTA:	XSSTA
00241	002623	IASTA:	XASTA
00242	002621	IBSTA:	XBSTA
00243	002634	IDSTA:	XDSTA
00244	002636	INSTA:	XNSTA
00245	003773	ICRLF:	CRLF
00246	003646	IMESS:	MESS
00247	003667	IPOCT:	POCT
00250	003744	ICHAR:	CHAR
00251	004015	ITYPE:	TYPE
00252	003672	IPDEC:	PDEC
00253	003624	IOUT36:	OUT36
00254	003632	IOPIP:	OPIP
00255	003643	IHAAT:	HAAT
00256	003015	IERAS:	XERAS
00257	002755	IWRT:	XWRT
00260	002663	IGEN:	XGEN
00261	002566	IWEOF:	XWEOF
00262	002563	IBSP:	XBSP
00263	002551	IRWD:	XRWD
00264	002554	IRWS:	XRWS
00265	002752	IWNS:	XWNS
00266	002575	ABTBF:	BTBF
00267	003401	SETP1:	ENTP1
00270	003406	SETUP:	ENTER
00271	003437	LOOP:	CYCLE
00272	003515	ER:	ERR
00273	002712	IRD:	XRD
00274	002707	IRDNS:	XRDNS
00275	002532	ICLR:	XCLR
00276	002505	ICLK:	XCHK
00277	002646	ILD:	XLD
00300	003045	IWT:	XWT
00301	002546	ISPC:	XSPC
00302	001725	IAS0:	AS0
00303	002700	ISEL:	XSEL
00304	003101	ITIM:	XTIM

;DEFINITIONS

000030 MTC=30
000032 FUN=32
000035 OIS=35
000035 .TIO=DIS
006237 STATUS=JSR @ISTAT
006240 SSTATUS=JSR @ISSTA
006241 ASTATUS=JSR @IASTA
006242 BSTATUS=JSR @IBSTA
006244 NSTATUS=JSR @INSTA
006243 DSTATUS=JSR @IDSTA
006256 ERASE=JSR @IERAS
006257 WRITE=JSR @IWRT
006260 GEN=JSR @IGEN
006261 WEOF=JSR @IWEOF
006262 BSPACE=JSR @IRSP
006263 REWIND=JSR @IRWD
006264 RWDSTALL=JSR @IRWS
006265 WRTNS=JSR @IWNS
006300 WAIT=JSR @IWT
006301 SPACE=JSR @ISPC
006272 EHALT=JSR @ER
006273 READ=JSR @IRD
006274 REDNS=JSR @IRDNS
006275 CLEAR=JSR @ICLR
006276 CHECK=JSR @ICHK
006277 LOAD=JSR @ILD
006303 SELECT=JSR @ISEL
006304 TIMER=JSR @ITIM
006255 HAATT=JSR @IHAAT
006253 ATT=JSR @IOUI36
006254 PIP=JSR @IOPIP

111

000400	.LOC 400		
00400	002404	JMP @IDIAG	;MTA DIAGNOSTIC
00401	002404	JMP @IWLT	;WRITE LOCK TEST
00402	002404	JMP @IETT	;END TAPE TEST
00403	002404	JMP @IDLT	;DATA LATE TEST
00404	000410	IDIAG: DIAG	
00405	003270	IWLT: WLT	
00406	003202	IETT: EIT	
00407	003130	IDLT: DLT	

III

```

00410 006236  DIAG:  ;CHECK SELB BUS LINE
00411 006270  A0:   JSR @INTIL
00412 0063500    ;DO SKPBZ, DEVICE 0
00413 006272    SKPBZ 0
00414 006271    EHALT ;CHECK THE SELB BUS LINE
                JSR @LOOP

00415 006270  A1:   ;CHECK SELD BUS LINE
00416 0063700    JSR @SETUP ;DO SKPDZ, DEVICE 0
00417 006272    SKPDZ 0
00420 006271    EHALT ;CHECK SELD BUS LINE
                JSR @LOOP

00421 006270  A2:   ;CHECK MTA BUSY FLOP OFF
00422 0063530    JSR @SETUP ;DO BUSY TEST ON
00423 006272    SKPBZ MTC ;MTA
00424 006271    EHALT ;CANNOT RESET MT BUSY
                JSR @LOOP

00425 006270  A3:   ;CHECK MTA DONE FLOP OFF
00426 0063730    JSR @SETUP ;DONE TEST ON MTA
00427 006272    SKPDZ MTC
00430 006271    EHALT ;CANNOT RESET MT DONE
                JSR @LOOP

00431 006270  A4:   ;CHECK MT SELECT, DOB, DIB, ADDR REG. IN-OUT
00432 0020143    JSR @SETUP ;LOAD CA WITH 125252
00433 0062030    LDA 0,C5252 ;READ BACK AND CHECK
00434 0065430    DOB 0,MTC
00435 106414    DIB 1,MTC
00436 006272    SUB# 0,1,SZR
                EHALT ;AC0=VALUE SENT TO ADDR REG
                ;AC1=VALUE READ BACK
                ;CHECK THE MT SELECT, LOAD ADDRESS
                ;ENABLE ADDR. A&B, AND ADDR,REG.
00437 006271    JSR @LOOP

00440 006270  A5:   ;CHECK CA REGISTER IN&OUT
00441 0020145    JSR @SETUP ;LOAD CA WITH 052525
00442 0062030    LDA 0,C5252 ;READ BACK AND CHECK
00443 0065430    DOB 0,MTC
00444 106414    DIB 1,MTC
00445 006272    SUB# 0,1,SZR
                EHALT ;AC0=VALUE SENT TO ADDR REG
                ;AC1=VALUE READ BACK
                ;CHECK ADDR REG DATA PATHS
00446 006271    JSR @LOOP

```

111

```

00447 020052 A6: ;CHECK FOR ILLEGAL MT SELECT
00450 040404 LDA 0,CMTA ;LOAD CA WITH 052525
00451 006270 A6,1: STA 0,A6,2 ;READ BACK USING EVERY
00452 020145 JSR 0SETUP ;DEVICE CODE EXCEPT MTA
00453 062030 LDA 0,C52525 ;DATA SHOULD NOT BE
00454 065400 DOB 0,MTC ;FOUND
00455 122415 A6,2: DIB 1,0 ;THIS IS CHANGED BY PROG
00456 006272 SUB# 1,0,SNR
EHALT ;AC0=DATA SENT TO ADDR REG
;AC1=DATA READ BACK
;PROBLEM IS IN MT SELECT
;THE DEVICE SELECT GATE,

00457 006271 JSR 0LOOP
00460 020774 LDA 0,A6,2
00461 101400 A6,3: INC 0,0
00462 024110 LDA 1,C77
00463 107405 AND 0,1,SNR
00464 000406 JMP A7 ;DONE ALL GO ON
00465 030233 LDA 2,DEVICE
00466 146415 SUB# 2,1,SNR
00467 000772 JMP A6,3 ;MTA INC AGAIN
00470 040764 STA 0,A6,2
00471 000760 JNP A6,1

;CHECK CA REGISTER, ALL DATA PATTERNS
00472 102400 A7: SUB 0,0 ;LOAD CA, READ BACK
00473 006267 JSR 0SETP1 ;AND CHECK, USE
00474 062030 DOB 0,MTC ;ALL FROM 0-077777
00475 065430 DIB 1,MTC
00476 106414 SUB# 0,1,SZR
00477 006272 EHALT ;AC0=DATA SENT TO ADDR REG
;AC1=DATA READ BACK
;ADDR REG IS PATTERN SENSITIVE

00500 006271 JSR 0LOOP
00501 101400 INC 0,0
00502 126620 SUBZR 1,1
00503 122414 SUB# 1,0,SZR
00504 000767 JMP A7+1

;CHECK IO RESET OF CA REGISTER
00505 006270 A8: JSR 0SETUP ;LOAD CA WITH ALL 1'S
00506 102000 ADC 0,0 ;DO IORST, READ
00507 062030 DOB 0,MTC ;CA AND CHECK FOR 0'S
00510 062677 IORST
00511 065430 DIB 1,MTC
00512 125004 MOV 1,1,SZR
00513 006272 EHALT ;AC0=DATA SENT TO ADDR REG
;AC1=DATA READ BACK
;CHECK IORST, RESET, AND
;ADDRESS REGISTER RESET GATES

00514 006271 JSR 0LOOP

```

111

```

00515 006270 ABA: ;CHECK CA LOAD
00516 102400 JSR #SETUP ;LOAD CA WITH 010421
00517 024132 SUB 0,0 ;LOAD CA WITH ALL ZEROS
00520 066030 LDA 1,C10421 ;READ AND CHECK
00521 062030 DOB 1,MTC
00522 061430 DOB 0,MTC
00523 101004 DIB 0,MTC
00524 006272 MOV 0,0,SZR
EHALT ;CA LOADING ERROR
;AC0=(CA)
;(CA) SHOULD BE ZERO

00525 006271 JSR #LOOP

00526 006270 ABB: JSR #SETUP ;LOAD CA WITH 21042
00527 102400 SUB 0,0 ;LOAD CA WITH ALL ZEROS
00530 024133 LDA 1,C21042 ;READ AND CHECK
00531 066030 DOB 1,MTC
00532 062030 DOB 0,MTC
00533 061430 DIB 0,MTC
00534 101004 MOV 0,0,SZR
00535 006272 EHALT ;CA LOADING ERROR
;AC0=(CA)
;(CA) SHOULD BE ZERO

00536 006271 JSR #LOOP

00537 006270 ABC: JSR #SETUP ;LOAD CA WITH 42104
00540 102400 SUB 0,0 ;LOAD CA WITH ALL ZEROS
00541 024134 LDA 1,C42104
00542 066030 DOB 1,MTC
00543 062030 DOB 0,MTC
00544 061430 DIB 0,MTC
00545 101004 MOV 0,0,SZR
00546 006272 EHALT ;CA LOADING ERROR
;AC0=(CA)
;(CA) SHOULD BE ZERO

00547 006271 JSR #LOOP

00550 006270 ABD: JSR #SETUP ;LOAD CA WITH 4210
00551 102400 SUB 0,0 ;LOAD CA WITH ALL ZEROS
00552 024135 LDA 1,C421T ;READ AND CHECK
00553 066030 DOB 1,MTC
00554 062030 DOB 0,MTC
00555 061430 DIB 0,MTC
00556 101004 MOV 0,0,SZR
00557 006272 EHALT ;CA LOADING ERROR
;AC0=(CA)
;(CA) SHOULD BE ZERO

00560 006271 JSR #LOOP

```

111

00561	006270	A8E:	;CHECK CA LOAD
00562	020132		JSR 0SETUP
00563	024133		LDA 0,C10421
00564	030134		LDA 1,C21042
00565	034135		LDA 2,C42104
00566	062030		LDA 3,C421T
00567	066030		DOB 0,MTC
00570	072030		DOB 1,MTC
00571	076030		DOB 2,MTC
00572	061430		DOB 3,MTC
00573	162414		DIB 0,MTC
00574	006272		SUB# 3,0,SZR
			EHALT ;CA LOADING ERROR
			;AC0=(CA)
			; (CA) SHOULD BE 004210
00575	006271		JSR 0LOOP

111

```

00576 006270 A9: ;CHECK MT START AND BUSY FLOP
00577 006303 JSR @SETUP ;SEND START PULSE
00600 060130 SELECT
00601 063430 NIOS MTC ;SKIP ON BUSY SET
00602 006272 SKPBN MTC
EHALT ;MTA START DID NOT SET BUSY
;CHECK MT START, MT BUSY FLOP
;AND SET LINE, MTC M40 TO GROUND.
;RESTART PROGRAM WITH UNIT ON-LINE.

00603 006271 JSR @LOOP

00604 006270 A10: ;CHECK IORST OF BUSY FLOP (CLEAR COM)
00605 006303 JSR @SETUP ;SEND START PULSE
00606 060130 SELECT
00607 063430 NIOS MTC ;MAKE SURE BUSY IS ON
SKPBN MTC ;DO IORST AND
;MAKE SURE BUSY OFF.
00610 006272 EHALT ;START PULSE DIDN'T SET BUSY
;THIS JUST WORKED IN THE
;PREVIOUS TEST!

00611 062677 IORST
00612 063530 SKPBZ MTC
00613 006272 EHALT ;IORST DIDN'T RESET BUSY
;CHECK CLEAR COM AND
;MT BUSY RESET LINE

00614 006271 JSR @LOOP

00615 006270 A11: ;CHECK CLEAR PULSE
00616 006303 JSR @SETUP ;SEND START PULSE
00617 060130 SELECT
00620 063430 NIOS MTC ;MAKE SURE BUSY IS ON
SKPBN MTC ;SEND CLEAR PULSE
;MAKE SURE BUSY RESETS
00621 006272 EHALT ;MT START DIDN'T SET MT BUSY
;THIS HAS WORKED TWICE BEFORE!

00622 060230 NIOS MTC
00623 063530 SKPBZ MTC
00624 006272 EHALT ;MT CLR DIDN'T RESET BUSY
;CHECK MT CLR THRU CLEAR ON

00625 006271 JSR @LOOP

00626 006270 A12: ;CHECK DOA=MTLC=COMMAND DECODER
00627 006303 JSR @SETUP ;SELECT REWIND
00630 020215 SELECT
00631 061130 LDA @,C7X ;SEND START PULSE
00632 000401 DOAS @,MTC ;MAKE SURE BUSY DOESN'T SET
JMP ,+1
00633 063530 SKPBZ MTC
00634 006272 EHALT ;CHECK E29 GATE DRIVING THE
;MT BUSY SET LINE. REWIND IS ON
;AND MT START IS SETTING BUSY.
;CHECK COMMAND DECODER, E2.

00635 006271 JSR @LOOP
00636 062677 IORST
00637 006300 WAIT
00640 000065 .50MS

```

111

```

00641 006267 A13: ;CHECK DIA AND STATUS REGISTER AT BOT
00642 006303 JSR @SETP1 ;REWIND TAPE
00643 006264 SELECT
00644 006242 RWDSTALL ;WAIT 5 SEC.
00645 010000 BSTATUS ;READ AND CHECK STATUS
00646 006272 @10000
EHALT ;AC1=EXPECTED STATUS, AC0=BAD.
;THIS IS THE FIRST READ STATUS.
;CHECK RD STATUS, REWIND THRU TO
;TAPE AND BOT LINE FROM TAPE.
;ALSO BOT STATUS REG. GATE

00647 006271 JSR @LOOP

;CHECK UNIT SELECT LOGIC, NOOP COMMAND DECODE
00650 020115 A14: LDA @,C170 ;SELECT EACH UNIT
00651 040205 STA @,X170 ;IN SEQUENCE 0-7
00652 006270 A14.1: JSR @SETUP ;CHECK FOR BOT IF
00653 020205 LDA @,X170 ;UNIT IS SELECTED
00654 061130 DOAS @,MTC ;OTHERWISE CHECK FOR
00655 006235 JSR @ICKST ;OFF-LINE.
00656 006272 EHALT ;AC1=EXPECTED STATUS, AC0=BAD
;AC2 13=15=UNIT JUST TESTED
;IF UNIT TESTED=UNIT SELECTED
;PROCEED ON THE STATUS INFO.
;IF NOT CHECK FOR ERRONEOUS OFF-LINE.

00657 006271 JSR @LOOP
00660 010205 ISZ X170
00661 020205 LDA @,X170
00662 024116 LDA 1,C200
00663 122414 SUB# 1,@,SZR
00664 000766 JMP A14.1

;CHECK REWIND=ERASE=REWIND SEQUENCE
00665 006267 A15: JSR @SETP1 ;REWIND WAIT 5 SEC.
00666 006264 RWDSTALL ;ERASE WAIT 1 SEC.
00667 006242 BSTATUS
00670 010000 @10000
00671 006272 EHALT ;BOT NOT FOUND
00672 006256 ERASE ;CHECK STATUS (TUR)
00673 006237 STATUS ;REWIND=CHK STATUS
00674 000000 @
00675 006272 EHALT ;AC1=EXPECTED STATUS, AC0=BAD
00676 006240 SSTATUS ;IF (BOT), ERASE DIDN'T MOVE TAPE.
00677 000000 @ ;CHECK ERASE COMMAND THRU TO TAPE
00700 006272 EHALT
00701 006264 RWDSTALL
00702 006242 BSTATUS
00703 010000 @10000
00704 006272 EHALT ;AC1=EXPECTED STATUS, AC0=BAD
;REWIND FOLLOWING ERASE DIDN'T
;WORK. THE CLUE IS IN THE BAD
;STATUS.

00705 006271 JSR @LOOP

;CHECK REWINDING STATUS BIT
00706 006267 A16: JSR @SETP1 ;DO ERASE-REWIND
00707 006256 ERASE ;CHECK STATUS
00710 006263 REWIND ;DURING REWIND
00711 006241 ASTATUS

```

00712 020000
00713 006272

00714 006300
00715 000062
00716 006271

020000

EHALT ;AC1=EXPECTED STATUS, AC0=BAD
;CHECK REWINDING STATUS BIT GATE

WAIT
.5SEC
JSR @LOOP

111

```

00717 006267 A17: ;CHECK "WAS REWINDING" STATUS
00720 006256 JSR #SETP1
00721 006264 ERASE
00722 006237 RWOSTALL
00723 020000 STATUS
00724 006272 020000
00725 006240 EHALT
00726 010000 SSTATUS ; AC1=EXP AC0=BAD
00727 006272 010000
00730 006271 EHALT
                                JSR #LOOP ;AC1=EXP AC0=BAD

                                ;CHECK SIMPLE 4 CHAR WRITE
00731 006271 A20: JSR #SETUP ;WRITE 2ND
00732 006260 GEN ;CHECK CA, DONE FLAG
00733 000377 A20,1: 377
00734 006257 WRITE
00735 000102 102
00736 006363 SKPDN MTC
00737 006272 EHALT ;DONE FLAG SHOULD BE ON.
                                ;THIS IS THE FIRST COMPLETE WRITE
                                ;AND INVOLVES ALL THE WRITE
                                ;LOGIC. TO COMPLETE A WRITE
                                ;ERF-LOAD DELAY-DLY OVER=STOP=
                                ;TUR=MT DONE, CLEAR MT BUSY.
                                ;CHECK WC OVER, INCREMENT WC.

00740 024053 LDA 1,0BUFF
00741 030072 LDA 2,C2
00742 147000 ADD 2,1
00743 071450 DIB 2,MTC
00744 146414 SUB# 2,1,SZR
00745 006272 EHALT ;ADDR REG. WRONG! AC1=GOOD
                                ;AC2=BAD. TWO OUTPUT XFERS
                                ;SHOULD HAVE OCCURED

00746 006237 STATUS
00747 000000 0
00750 006272 EHALT ;AC1=EXPECTED STATUS. AC0=BAD
                                ;IF CA WRONG YOU MIGHT EXPECT
                                ;BAD STATUS. FIX FIRST PROBLEMS
                                ;FIRST!
                                ;DATA PATTERN IS I=0=1=0
                                ;FOR EACH TRACK. LATERAL
                                ;PARITY IS ALWAYS 1.

00751 006271 JSR #LOOP

```

III

```

00752 006267 A21: ;CHECK INTERRUPT AND DISABLE
JSR @SETP1 ;WRITE 2 WDS
;CHECK DONE
00753 006257 WRITE ;ALLOW INTERRUPT
00754 000102 102 ;DISABLE INTERRUPT
00755 0063630 SKPDN MTC ;ALLOW INT. AGAIN
00756 006272 EHALT ;NO DONE FLAG! THIS HAS WORKED
;BEFORE! RESTART PROGRAM.

00757 060177 INTEN
00760 000401 JMP ,+1 ;ALLOW 1 INST BEFORE INT.
00761 006272 EHALT ;NO INTERRUPT. CHECK MT INT REG.
;INTR GATE, INTP IN GATE.

00762 061477 INTA 0
00763 024233 LDA 1,DEVICE
00764 106414 SUB# 0,1,SZR
00765 006272 EHALT ;WRONG ADDRESS FROM MTA INT.
;CHECK INTA AND GATES
;AC1=EXPECTED ADDRESS, AC0=BAD

00766 024123 LDA 1,C2000
00767 066077 MSK0 1 ;DISABLE MTA INT
00770 060177 INTEN
00771 000401 JMP ,+1
00772 000402 JMP ,+2
00773 006272 EHALT ;ILLEGAL INTERRUPT! CHECK
;MT INT DISABLE FLOP, AND
;CONTROL GATES

00774 060277 INTDS
00775 006271 JSR @LOOP

```

111

```

;WRITE 16 CHAR RECORD. ODD PARITY
;PATTERN IN A22.2
00776 102000 A22: ADC 0,0
00777 040405 STA 0,A22.2
01000 020167 LDA 0,POINT
01001 040021 STA 0,IDX1
01002 006267 A22.1: JSR 0SETUP1 ;WRITE
;CHECK DONE,CA
;STATUS.
01003 006260 GEN
01004 177777 A22.2: 177777
01005 006257 WRITE
01006 000110 110
01007 063630 SKPDN MTC
01010 006272 EHALLT ;NO DONE FLAG! RECORDS TO NOW
;HAVE BEEN 4 CHAR. THIS ONE IS
;16 CHAR. THIS IS THE ONLY
;DIFFERENCE. READ AFTER WRITE
;DETECTS ERF,AND ERF=LOAD DELAY=
;DLY OVER=STOP=TUR=MT DONE.
;CHECK WC OVER, INCREMENT WC
01011 024053 LDA 1,0BUFF
01012 030100 LDA 2,C10
01013 147000 ADD 2,1
01014 061430 DIB 0,MTC
01015 106414 SUB# 0,1,SZR
01016 006272 EHALLT ;CA WRONG. AC1=CORRECT
;AC0=BAD. CHECK DATA
;CHAN CONTROL AND INCREMENT
;WC. 10 OCTAL WORDS WRITTEN
01017 006237 STATUS
01020 000000 0
01021 006272 EHALLT ;AC1=EXPECTED STATUS, AC0=BAD
;THE CLUE IS IN THE BAD STATUS
;EXPECTED BAD STATUS IF ERRORS
;OCCURRED PREVIOUSLY HERE.
01022 006271 JSR 0LOOP
01023 020021 LDA 0,IDX1
01024 024175 LDA 1,LST
01025 106405 SUB 0,1,SNR
01026 000404 JMP A23 ;GO TO NEXT TEST
01027 022021 LDA 0,0IDX1
01030 040754 STA 0,A22.2
01031 000751 JMP A22.1
;TEST ILLEGAL STATUS
01032 074477 A23: READS 3 ;GIVE START TO UNSELECTED TAPE
01033 030077 LDA 2,C7 ;CONNECTED TO ANOTHER FORMATTER
01034 020075 LDA 0,C5
01035 117000 ADD 0,3
01036 173400 AND 3,2
01037 034115 LDA 3,C170
01040 173000 ADD 3,2
01041 006270 JSR 0SETUP
01042 071030 DOA 2,MTC
01043 060130 NIOS MTC ;START NOOP
01044 006241 ASTATUS
01045 040000 040000
01046 006272 EHALLT ;AC1=EXPECTED STATUS AC0=BAD
;CHECK ILLEGAL FLOP AND GATES

```

01047 006271

JSR #LOOP

111

;TAPE 3

```

01050 074477 A24: ;TEST ILLEGAL STATUS
01051 030077 READS 3 ;GIVE START TO UNSELECTED
01052 175400 LDA 2,C7 ;TAPE
01053 173400 INC 3,3 ;AC2=UNIT
01054 034115 AND 3,2
01055 173000 LDA 3,C170
01056 006270 ADD 3,2
01057 071030 JSR 0SETUP
01060 060130 DOA 2,MTC ;START NOOP
01061 006241 NIOS MTC
01062 040000 ASTATUS
01063 006272 EHALL ;AC1=EXP AC0=BAD
01064 006271 JSR 0LOOP ;CHECK ILLEGAL FLOP AND GATES

```

```

01065 006267 ;TEST ILLEGAL STATUS
01066 006264 A25: JSR 0SETP1 ;REWIND. GIVE
01067 006262 RWOSTALL ;SPACE HACK AT BOT
01070 006241 BSPACE
01071 011000 ASTATUS
01072 006272 011000
01073 006243 EHALL ;AC1=EXPECTED STATUS, AC0=BAD.
01074 001000 OSTATUS ;CHECK SPACE BKD&BOT GATE ON
01075 006272 001000 ;ILLEGAL (E31). CHECK SPACE BKD
01076 006271 EHALL ;COMMAND DECODE
JSR 0LOOP

```

III

```
                                ;CHECK WRITE EOF STATUS
01077 006270 A26: JSR @SETUP      ; WEOF
                                ;AND CHECK STATUS
01100 006261      WEOF
01101 006360      SKPDN MTC
01102 006272      EHALLT ;NO DONE FLAG. CHECK READ
                                ;CIRCUITS, EOF CODE,"1ST CHAR
                                ;IS EOF", STOP.
01103 006237      STATUS
01104 000400      000400
01105 006272      EHALLT ;AC1=EXPECTED STATUS. AC0=BAD
01106 006243      DSTATUS
01107 000400      000400
01110 006272      EHALLT
```

111

```

01111 006267 A28: ;CHECK BACKSPACE
01112 006264 JSR @SETP1 ;REWIND=WRITE 1 REC.
01113 006257 RWDSTALL ;BACKSPACE AND CHECK
01114 000102 WRITE ;DONE AND CORRECT
01115 006262 102 ;STATUS
01116 006300 BSPACE
01117 000067 WAIT
01120 0063630 .200MS
01121 006272 SKPDN MTC
EHALT ;NO DONE FLAG FOLLOWING
;BACKSPACE. CHECK SPACE BKD
;COMMAND DECODE AND DRIVER
;TO TAPE. CHECK CAPSTAN
;SLIPPING ON TAPE.

01122 006237 STATUS
01123 000000 0
01124 006272 EHALT ;AC1=EXPECTED STATUS, AC0=BAD
01125 006271 JSR @LOOP

;CHECK BACKSPACE
01126 006267 A31: JSR @SETP1 ;REWIND=WRITE 1 REC.
01127 006264 RWDSTALL
01130 006257 WRITE ;BACKSPACE 2 RECS.
01131 000102 102 ;CHECK BOT
01132 006262 BSPACE
01133 006300 WAIT
01134 000067 .200MS
01135 006262 BSPACE
01136 006300 WAIT
01137 000060 .1SEC
01140 006237 STATUS
01141 011000 011000
01142 006272 EHALT ;AC1=EXPECTED STATUS, AC0=BAD
01143 006240 SSTATUS ;BOT SHOULD STOP OPERATION
01144 010000 010000
01145 006272 EHALT

01146 006271 JSR @LOOP
;CHECK SPACE FORWARD
01147 006267 A32: JSR @SETP1 ;REWIND=WRITE 3 REC.
01150 006264 RWDSTALL ;REWIND=CHECK STATUS
01151 006257 WRITE ;SPACE FORWARD=CHECK
01152 000302 302 ;FOR NO BOT, & PROPER
01153 006264 RWDSTALL ;ADDR. REG.
01154 006237 STATUS
01155 020000 020000
01156 006272 EHALT ;AC1=EXPECTED STATUS, AC0=BAD
01157 006240 SSTATUS ;THIS HAS WORKED MANY TIMES
01160 010000 010000 ;BEFORE. IF ERROR RESTART PROG
01161 006272 EHALT
01162 006301 SPACE
01163 006300 WAIT
01164 000060 .1SEC
01165 006237 STATUS
01166 000000 0
01167 006272 EHALT ;AC1=EXPECTED STATUS, AC0=BAD.
;THIS IS THE FIRST SPACE FORWARD.
;CHECK COMMAND DECODER=SPACE FWD=
;COUNT WC (E43)

```

01170 006271

JSR 4LOOP

111

```

01171 102000 A33:    ;CHECK READ INSTRUCTION.  READ 16 CHAR
01172 040406      ;RECORDS.  PATTERN IN A33.2 (ODD PARITY)
01173 020167      ADC 0,0      ;REWIND=WRITE 1 REC.
01174 040021      STA 0,A33.2  ;CHECK STATUS=REWIND=
01175 006267      LDA 0,POINT  ;CHECK STATUS=READ=
01176 006264      STA 0,IDX1   ;CHECK DONE,STATUS.
01177 006260      JSR @SETP1   ;ADDR. REG., & DATA
01200 177777      RWDSTALL    ;BUFFER
01201 006257      GEN
01202 000110      A33.2: 177777
01203 006237      WRITE
01204 000000      110
01205 006272      STATUS
                        0
                        EHALLT ;AC1=EXPECTED STATUS, AC0=BAD
                        ;THIS MUCH HAS WORKED BEFORE,
                        ;MANY TIMES!  RESTART THE PROGRAM

01206 006264      RWDSTALL
01207 006237      STATUS
01210 020000      020000
01211 006272      EHALLT ;AC1=EXPECTED STATUS, AC0=BAD
01212 006240      SSTATUS ;LOOKING FOR HOT BEFORE TRYING
01213 010000      010000 ;FIRST READ.  THIS HAS WORKED
01214 006272      EHALLT ;BEFORE!
01215 006275      CLEAR
01216 006273      READ
01217 000110      110
01220 063630      SKPDN MTC
01221 006272      EHALLT ;NO DONE FLAG FOLLOWING READ,
                        ;THIS IS THE FIRST READ INST.
                        ;CHECK READ=DCHI=INCREMENT WC

01222 006237      STATUS
01223 000000      0
01224 006272      EHALLT ;AC1=EXPECTED STATUS, AC0=BAD
                        ;90% OF THE READ LOGIC HAS BEEN
                        ;CHECKED DURING WRITE OPERATIONS.
                        ;PROBLEMS HERE ARE LIKELY IN THE
                        ;PROCESSOR INTERFACE AREA.

01225 020054      LDA 0,IBUFF
01226 024100      LDA 1,C10
01227 177000      ADD 0,1
01230 061430      DIB 0,MTC
01231 106414      SUB# 0,1,SZR
01232 006272      EHALLT ;ADDRESS REGISTER IS WRONG.
                        ;AC1=CORRECT, AC0=ACTUAL.
                        ;10 WORD REC READ.  CHECK
                        ;MT DCH SEL=DCHM0=INCREMENT
                        ;WC

```

III

01233 006276
01234 000010
01235 006272

CHECK
10
EHALT ;INPUT BUFFER CONTAINS WRONG
;DATA. INPUT BUFFER SHOULD EQUAL
;OUTPUT BUFFER FOR THE FIRST 8
;WORDS. AC0=GOOD WORD, AC1=BAD
;AC2=ADDRESS OF INPUT BUFFER

01236 006271
01237 020021
01240 024175
01241 106405
01242 000404
01243 022021
01244 040734
01245 000730

JSR 0LOOP
LDA 0,IDX1
LDA 1,LST
SUB 0,1,SNR
JMP A35
LDA 0,0IDX1
STA 0,A33.2
JMP A33.1

111

```

01246 006267 A35: ;READ 16 CHAR REC. WITH WC=63
01247 006264 JSR @SETP1 ;REWIND= WRITE 1 REC.
01250 006260 RWDSTALL ;CHECK STATUS=REWIND
01251 125252 GEN ;READ=CHECK DONE,
01252 006257 125252 ;STATUS,ADDR REG,
WRITE ;AND DATA
01253 000110 110
01254 006237 STATUS
01255 000000 0
01256 006272 EHALT ;AC1=EXPECTED STATUS, AC0=BAD
;DON'T TRY TO READ IT IF IT
;CANNOT BE WRITTEN PROPERLY.
;THIS HAS WORKED BEFORE,
;RESTART PROGRAM

01257 006264 RWDSTALL
01260 006273 READ
01261 000177 177
01262 063630 SKPDN MTC
01263 006272 EHALT ;NO DONE FLAG. CHECK INCREMENT
;WC=ERF=STOP=1UR

01264 006237 STATUS
01265 000000 0
01266 006272 EHALT ;AC1=EXPECTED STATUS, AC0=BAD
;THIS IS THE FIRST TIME ON A
;READ THAT THE WC DIDN'T MATCH
;THE RECORD LENGTH. CHECK
;ERF GATE ON STOP.

01267 020054 LDA 0,18UFF
01270 024100 LDA 1,C10
01271 107000 ADD 0,1
01272 061430 DIB 0,MTC
01273 106414 SUB# 0,1,SZR
01274 006272 EHALT ;ADDRESS REGISTER WRONG
;AC1=CORRECT ADDRESS, AC0=BAD.
;CHECK INCREMENT WC

01275 006276 CHECK
01276 000010 10
01277 006272 EHALT ;DATA ERROR.
;AC0=GOOD WORD, AC1=BAD WORD
;AC2=ADDRESS OF THE INPUT BUFFER

01300 006271 JSR @LOOP

```

III

```

01301 006267 A36: ;READ 63 CHAR REC. WITH WC=10
01302 006264 JSR @SETP1 ;REWIND=WRITE 1 REC.
01303 006260 RWDSTALL ;CHECK STATUS=REWIND
01304 125252 GEN ;READ=CHECK DONE,
01305 006257 125252 ;STATUS,ADDR, REG.
01306 000177 WRITE ;AND DATA
01307 006237 177
01310 000000 STATUS
01311 006272 0
EHALT ;AC1=EXPECTED STATUS, AC0=BAD
;THIS MUCH HAS WORKED MANY
;TIMES BEFORE. RESTART PROG.

01312 006264 RWDSTALL
01313 006273 READ
01314 000110 110
01315 063630 SKPDN MTC
01316 006272 EHALT ;NO DONE FLAG. CHECK INCREMENT
;WC=WC OVERFLOW=STOP

01317 006237 STATUS
01320 000200 000200
01321 006272 EHALT ;AC1=EXPECTED STATUS, AC0=BAD
;THIS IS THE FIRST TIME A READ IS
;STOPPED BY WC OVERFLOW ALONE.
;LOOK FOR WC OVERFLOW AND CHECK
;THE WC OVERFLOW GATE ON STOP

01322 020054 LDA 0,IBUFF
01323 024100 LDA 1,C10
01324 107000 ADD 0,1
01325 061430 DIB 0,MTC
01326 106414 SUB# 0,1,SZR
01327 006272 EHALT ;ADDRESS REG. WRONG. AC1=CORRECT
;ADDR. AC0=BAD.
;CHECK DCHI LOGIC AND INCREMENT WC

01330 006276 CHECK
01331 000010 10
01332 006272 EHALT ;DATA ERROR. INPUT BUFFER SHOULD
;=OUTPUT BUFFER FOR THE FIRST 8
;WORDS.
;AC0=GOOD WORD, AC0=BAD WORD
;AC2=ADDRESS OF THE INPUT BUFFER

01333 006271 JSR @LOOP

```

111

```

01334 006267 A39: ;CHECK READ EOF
01335 006264 JSR 0SETP1 ;REWIND=WRITE A 16
01336 006257 RWDSTALL;CHAR RFC., WRITE EOF
01337 000110 WRITE;CHECK STATUS
01340 006261 110;REWIND=READ 2 REC.
01341 006237 WEOF;CHECK STATUS,CA
01342 000400 STATUS
01343 006272 000400
01344 006240 EHALT ;AC1=EXPECTED STATUS, AC0=BAD
01345 000000 SSTATUS ;THIS HAS WORKED BEFORE. RESTART
01346 006272 0
01347 006264 EHALT
01350 006273 RWDSTALL
01351 000210 READ
01352 024054 210
01353 061430 LDA 1,IBUFF
01354 070477 DIB 0,MTC
01355 034124 READS 2
01356 157404 LDA 3,C4K
01357 000414 AND 2,3,SZR ;NRZI
01360 125400 JMP A39.2 ;PE
01361 106414 INC 1,1
01362 006272 SUB# 0,1,SZR
;ADDRESS REG WRONG. AC1=CORRECT
;AC0=BAD. EOF SHOULD INCR
;WC AND ADDR. BY 1 IF NRZI
; IF PE MUST WC>=16

01363 006237 STATUS
01364 000410 000410
01365 006272 EHALT ;AC1=EXP, AC0=BAD
01366 006240 A39.1: SSTATUS ;LOOKING FOR READ EOF
01367 000000 0
01370 006272 EHALT

01371 006271 JSR 0LOOP
01372 000407 JMP A40

01373 106414 A39.2: SUB# 0,1,SZR
01374 006272 EHALT ;ADDRESS REG WRONG. AC1=CORRECT
;AC0=BAD. EOF SHOULD NOT INCR
;WC AND ADDR. IF PE.

01375 006237 STATUS
01376 000400 000400
01377 006272 EHALT ;AC1=EXP, AC0=BAD
;LOOKING FOR READ EOF.

01400 000766 JMP A39.1

```

III

```

01401 006267 A40: ;CHECK READ EOF ON SPACE BACK
01402 006264 JSR @SETP1 ;REWIND=WRITE 2 REC
01403 006257 RWDSTALL ;WRITE EOF THEN 3 REC
01404 000210 WRITE ;BACKSPACE 10
01405 006201 210 ;CHECK STATUS AND
01406 006237 WEOF ;ADDR REG.
01407 000400 STATUS
01410 006272 EHALT ;AC1=EXPECTED STATUS, AC0=BAD
;THIS HAS WORKED BEFORE.
;RESTART PROGRAM

01411 006257 WRITE
01412 000310 310
01413 006262 BSPACE
01414 006300 WAIT
01415 000065 .50MS
01416 006262 BSPACE
01417 006300 WAIT
01420 000065 .50MS
01421 006262 BSPACE
01422 006300 WAIT
01423 000065 .50MS
01424 006262 BSPACE
01425 006300 WAIT
01426 000065 .50MS
01427 006237 STATUS
01430 000400 000400
01431 006272 EHALT ;AC1=EXPECTED STATUS, AC0=BAD
;LOOKING FOR EOF. CHECK
;EOF GATE ON STOP

01432 006262 BSPACE
01433 006300 WAIT
01434 000070 .400MS
01435 006244 NSTATUS
01436 000400 000400
01437 006272 EHALT ;AC1=EXP AC0=BAD
;EOF BIT CANNOT BE RESET

01440 006271 JSR @LOOP

```

III

;TAPE 4

```

01441 006267 A41: ;CHECK READ EOF ON SPACE FORWARD
01442 006264 JSR @SETP1 ;REWIND-WRITE 2 REC
01443 006257 RWDSTALL ;WEOF-WRITE 3 REC
01444 000210 WRITE ;REWIND-SPACE 5
01445 006261 210 ;CHECK STATUS, ADDR
01446 006237 WEOF ;REG.
01447 000400 STATUS
01450 006272 EHALL ;AC1=EXPECTED STATUS, AC0=BAD
;THIS HAS WORKED BEFORE.
;RESTART PROGRAM

01451 006257 WRITE
01452 000310 310
01453 006264 RWDSTALL
01454 006301 SPACE
01455 006300 WAIT
01456 000060 .1SEC
01457 006301 SPACE
01460 006300 WAIT
01461 000060 .1SEC
01462 006301 SPACE
01463 006300 WAIT
01464 000060 .1SEC
01465 006237 STATUS
01466 000400 000400
01467 006272 EHALL ;AC1=EXPECTED STATUS, AC0=BAD
;LOOKING FOR EOF STATUS. THIS IS
;THE SAME AS THE LAST EXCEPT THAT
;SPACING IS FORWARD

01470 006301 SPACE
01471 006300 WAIT
01472 000060 .1SEC
01473 006244 NSTATUS
01474 000400 000400
01475 006272 EHALL ;AC1=EXP AC0=BAD
;EOF BIT CANNOT BE RESET
01476 006271 JSR @LOOP

```

III

```

;CRC-LPC CHECK
;9 TRACK ONLY NRZI
01477 074477 A42: READS 3
01500 030111 LDA 2,C100
01501 000401 JMP ,+1
01502 002302 JMP #IA50 ;SKIP OVER 9 TRK TESTS
01503 006267 JSR #SETP1 ;REWIND-WRITE 1 REC
01504 006264 RWDSTALL ;CHECK STATUS-REWIND
01505 006277 LOAD ;READ( TEST MODE)
01506 002605 PAT1 ;CHECK STATUS,CA,DATA
01507 006257 WRITE
01510 000102 102
01511 006237 STATUS
01512 000001 1
01513 006272 EHALT ;AC1=EXPECTED STATUS, AC0=BAD
;THIS HAS WORKED BEFORE.
;RESTART PROGRAM

01514 006264 RWDSTALL
01515 006275 CLEAR
01516 024146 LDA 1,M2
01517 067030 DDC 1,MTC
01520 024054 LDA 1,IBUFF
01521 127240 ADDOR 1,1 ;SET BIT 0
01522 066030 DOB 1,MTC
01523 024206 LDA 1,CX
01524 065130 DOAS 1,MTC
01525 006300 WAIT
01526 000060 .1SEC
01527 006237 STATUS
01530 000001 1
01531 006272 EHALT ;AC1=EXPECTED STATUS, AC0=BAD
;JUST READ 4 CHAR REC IN TEST MODE.
;TEST MODE IS THE ONLY NEW
;VARIABLE

```

III

01532 061430
 01533 024054
 01534 030072
 01535 147000
 01536 106414
 01537 006272

01540 131000
 01541 066430
 01542 045000
 01543 006276
 01544 000003
 01545 006272

DIB 0,MTC
 LDA 1,IRUFF
 LDA 2,C2
 ADD 2,1
 SUB# 0,1,SZR
 EHALL ;ADDR REG WRONG. AC1=CORRECT
 ;AC0=ACTUAL

MOV 1,2
 DIC 1,MTC
 STA 1,0,2
 CHECK
 3

EHALL ;DATA ERROR. 2 WORDS WERE WRITTEN
 ;3 WORDS WERE READ. THE 3RD WORD IS
 ;CRC(LEFT) LPC(RIGHT)
 ;THIS TEST LOOKS FOR BIT PICKUPS
 ;IN THE CRC GENERATOR. DATA IS CHOSEN
 ;TO PRODUCE A MINIMUM 1'S IN CRC
 ;GENERATION. THE SEQUENCE XOR-SHIFT-
 ;COMP. IS DONE ONCE FOR EACH OF THE
 ;4 CHAR. XOR 1&3 PRODUCE A 1 IN CRC1
 ;WHICH IS THEN SHIFTED INTO CRC2.
 ;ALL OTHER CASES PRODUCE 0'S.
 ;1'S ARE WRITTEN ON TAPE BECAUSE
 ;OF CRC COMPLEMENT AT THE END.
 ;AC2= ADDR. OF IRUFF
 ;AC1= BAD WORD
 ;AC0= GOOD WORD
 ;EXPECTED DATA IS 000200
 ; 000200
 ; 153600

01546 006271

JSR @LOOP

III

```

;CRC=LPC CHECK
;9 TRACK ONLY
01547 006267 A43: JSR #SETP1 ;REWIND-WRITE 1 REC
01550 006264 RWDSTALL ;CHECK STATUS-REWIND
01551 006277 LOAD ;READ( TEST MODE)
01552 002610 PAT2 ;CHECK STATUS,CA,DATA
01553 006257 WRITE
01554 000102 102
01555 006237 STATUS
01556 000001 1
01557 006272 EHALL ;AC1=EXPECTED STATUS, AC0=BAD
;THIS HAS WORKED BEFORE.
;RESTART PROGRAM

01560 006264 RWDSTALL
01561 006275 CLEAR
01562 024146 LDA 1,M2
01563 067030 DOC 1,MTC
01564 024054 LDA 1,IRUFF
01565 127240 ADDR 1,1 ;SET BIT 0
01566 066030 DOB 1,MTC
01567 024206 LDA 1,CX
01570 065130 DOAS 1,MTC
01571 006300 WAIT
01572 000060 .1SEC
01573 006237 STATUS
01574 000001 1
01575 006272 EHALL ;AC1=EXPECTED STATUS, AC0=BAD
;JUST READ 4 CHAR REC IN TEST MODE.
;TEST MODE IS THE ONLY NEW
;VARIABLE

```

III

01576 061430
 01577 024054
 01600 030072
 01601 147000
 01602 106414
 01603 006272

 01604 131000
 01605 066430
 01606 045000
 01607 006276
 01610 000003
 01611 006272

DIB 0,MTC
 LDA 1,IBUFF
 LDA 2,C2
 ADD 2,1
 SUB# 0,1,SZR
 EHALLT ;ADDR REG WRONG. AC1=CORRECT
 ;AC0=ACTUAL.

MOV 1,2
 DIC 1,MTC
 STA 1,0,2
 CHECK
 3

EHALLT ;DATA ERROR. 2 WORDS WERE WRITTEN
 ;3 WORDS WERE READ. THE 3RD WORD IS
 ;CRC(LEFT) LPC(RIGHT)
 ;THIS TEST LOOKS FOR BIT DROPS IN THE
 ;CRC GENERATOR. DATA IS CHOSEN
 ;TO PRODUCE A MINIMUM OF 0'S. THE
 ;SEQUENCE XOR-SHIFT-COMP OCCURS
 ;ONCE FOR EACH OF THE 4 CHAR. FOLLOWING
 ;EACH COMP CRC 4,5,6,7 ARE ZEROS.
 ;ALL OTHER BITS ARE ALWAYS 1'S.
 ;ZEROS ARE WRITTEN ON TAPE BECAUSE
 ;OF CRC COMP AT END
 ;AC2=ADDR OF IBUFF
 ;AC1= BAD WORD
 ;AC0= GOOD WORD
 ;EXPECTED DATA IS 177474
 ; 136074
 ; 112074

01612 006271

JSR 0LOOP

III

```

01013 006267 A44: ;CRC=LPC CHECK
01014 006264 ;9 TRACK ONLY
01015 006277 JSR 0SETP1 ;REWIND-WRITE 1 REC
01016 006213 RWDSTALL ;CHECK STATUS-REWIND
01017 006257 LOAD ;READ( TEST MODE)
01020 000102 PAT3 ;CHECK STATUS,CA,DATA
01021 006237 WRITE
01022 000001 102
01023 006272 STATUS
1
EHALT ;AC1=EXPECTED STATUS, AC0=BAD
;THIS HAS WORKED BEFORE
;RESTART PROGRAM

01024 006264 RWDSTALL
01025 006275 CLEAR
01026 024146 LDA 1,M2
01027 067030 DDC 1,MTC
01030 024054 LDA 1,IBUFF
01031 127240 ADDR 1,1 ;SET BIT 0
01032 066030 DDB 1,MTC
01033 024206 LDA 1,CX
01034 065130 DOAS 1,MTC
01035 006300 WAIT
01036 000060 .1SEC
01037 006237 STATUS
01040 000001 1
01041 006272 EHALT ;AC1=EXPECTED STATUS, AC0=BAD
;JUST READ 4 CHAR REC IN TEST MODE.
;TEST MODE IS THE ONLY NEW
;VARIABLE

```

111

01642 061430
01643 024054
01644 030072
01645 147000
01646 106414
01647 006272

01650 131000
01651 066430
01652 045000
01653 006276
01654 000003
01655 006272

DIB 0,MTC
LDA 1,IBUFF
LDA 2,C2
ADD 2,1
SUB# 0,1,SZR
EHALT ;ADDR REG WRONG. AC1=CORRECT
;AC0=ACTUAL.

MOV 1,2
DIC 1,MTC
STA 1,0,2
CHECK
3

EHALT ;DATA ERROR. 2 WORDS WERE WRITTEN
;3 WORDS WERE READ. THE 3RD WORD IS
;CRC(LEFT) LPC(RIGHT)
;THIS TEST IS A GATE CHECK FOR THE
;CRC GENERATOR. DATA IS CHOSEN
;TO PRODUCE ALL COMBINATIONS OF XOR
;ON EACH CRC BIT. THIS 4 CHAR
;RECORD CATCHES 80% OF THE CASES.
;AC2=ADDR OF IBUFF
;AC1= BAD WORD
;AC0= GOOD WORD
;EXPECTED DATA IS 052652
; 076232
; 037232

01656 006271

JSR 0LOOP

III

```

;CRC-LPC CHECK
;9 TRACK ONLY
01657 006267 A45: JSR #SETP1 ;REWIND-WRITE 1 REC
01660 006264 RWDSTALL ;CHECK STATUS-REWIND
01661 006277 LOAD ;READ( TEST MODE)
01662 006216 PAT4 ;CHECK STATUS,CA,DATA
01663 006257 WRITE
01664 000102 102
01665 006237 STATUS
01666 000001 1
01667 006272 EHALL ;AC1=EXPECTED STATUS, AC0=BAD
;THIS HAS WORKED BEFORE.
;RESTART PROGRAM

01670 006264 RWDSTALL
01671 006275 CLEAR
01672 024146 LDA 1,M2
01673 067030 DDC 1,MTC
01674 024054 LDA 1,IBUFF
01675 127240 ADDR 1,1 ;SET BIT 0
01676 066030 DOB 1,MTC
01677 024206 LDA 1,CX
01700 065130 DGAS 1,MTC
01701 006300 WAIT
01702 000060 .1SEC
01703 006237 STATUS
01704 000001 1
01705 006272 EHALL ;AC1=EXPECTED STATUS, AC0=BAD
;JUST READ 4 CHAR REC IN TEST MODE.
;TEST MODE IS THE ONLY NEW
;VARIABLE

```

III

01706 061430
01707 024054
01710 030072
01711 147000
01712 106414
01713 006272

01714 131000
01715 066430
01716 045000
01717 006276
01720 000003
01721 006272

01722 006271
01723 002401
01724 001725

DIB 0,MTC
LDA 1,IBUFF
LDA 2,C2
ADD 2,1
SUB# 0,1,SZR
EHALT ;ADDR REG WRONG. AC1=CORRECT
;AC0=ACTUAL.
MOV 1,2
DIC 1,MTC
STA 1,0,2
CHECK
3
EHALT ;DATA ERROR. 2 WORDS WERE WRITTEN
;3 WORDS WERE READ. THE 3RD WORD IS
;CRC(LEFT) LPC(RIGHT)
;THIS TEST IS A GATE CHECK FOR THE
;CRC GENERATOR. DATA IS CHOSEN TO
;PRODUCE ALL COMBINATIONS OF XOR
;ON EACH CRC BIT. THIS TEST CATCHES
;THE REMAINING 20% OF THE CASES
;NOT COVERED IN THE PREVIOUS TEST.
;AC2=ADDR OF IBUFF
;AC1= BAD WORD
;AC0= GOOD WORD
;EXPECTED DATA IS 167141
; 000200
; 063600

JSR 0LOOP
JMP 0,+1
A50

111

```

;SPACING CHECK.
;DURING SPACING MANY IO COMMANDS ARE
;ISSUED TO GENERATE NOISE.
01725 006267 A50: JSR @SETP1 ;WRITE 500 2ND REC.
01726 006263 REWIND ;SPACEBACK 500
01727 006304 TIMER ;CHECK CA
01733 000064 .20SEC
01731 006242 BSTATUS
01732 010000 010000
01733 006272 EHALL ;NO BOT 20 SEC.
;FOLLOWING REWIND
01734 020102 LDA 0,C20
01735 040232 STA 0,RECTR
01736 006265 A50.1: WRINS
01737 000102 102
01740 006304 TIMER
01741 000060 .1SEC
01742 000401 JMP .+1
01743 063630 SKPDN MTC
01744 006272 EHALL ;NO DONE 1 SEC FOLLOWING
;WRITE COMMAND
01745 014232 DSZ RECTR
01746 000770 JMP A50.1
01747 006271 JSR @LOOP
01750 002401 JMP .+1
01751 002206 ETST
01752 020154 LDA 0,D500
01753 100400 NEG 0,0
01754 063030 DOC 0,MTC ;WC=500
01755 102400 SUB 0,0
01756 062030 DOB 0,MTC ;ADDR. REG= 0
01757 020211 LDA 0,C3X
01760 061130 DOAS 0,MTC ;START BACKSPACE
01761 102400 SUB 0,0
01762 064430 A50.2: DIA 1,MTC
01763 064430 DIA 1,MTC
01764 101404 INC 0,0,SZR
01765 000775 JMP .-3
01766 064430 DIA 1,MTC
01767 125203 MOVR 1,1,SNC
01770 000772 JMP A50.2
01771 061430 DIA 0,MTC
01772 024154 LDA 1,D500
01773 122414 SUB# 1,0,SZR
01774 006272 EHALL ;ADDRESS REG. WRONG.
;SHOULD HAVE COUNTED 500
;REC ON BACKSPACE.
;AC1=EXPECTED COUNT, AC0=ACTUAL
01775 006262 BSPACE
01776 064430 DIA 1,MTC
01777 063530 SKPRZ,MTC
02000 000776 JMP .-2
02001 020125 LDA 0,C10K
02002 107405 AND 0,1,SNR
02003 006272 EHALL ;BACKSPACE 1 DID NOT PRODUCE
;BOT. ADDR REG COUNTED 500.
;WC REG WAS COUNTING FASTER
;BAD WC REG.
02004 006271 JSR @LOOP

```

```

;SPACING CHECK.
;DURING SPACING MANY IO COMMANDS ARE
;ISSUED TO GENERATE NOISE.
02005 006267 A51: JSR @SETP1 ;WRITE 500 2ND REC.
02006 006263 REWIND ;SPACEBACK 500
02007 006304 TIMER ;CHECK CA
000012 .RDX 10
02010 047140 ;SPACEBACK 1 AND
000010 .RDX 8
02011 000201 201 ;CHECK FOR BOT
02012 006272 EHALT ;NO BOT-TUR 20 SEC.
;FOLLOWING REWIND
02013 020154 LDA 0,0500
02014 040232 STA 0,RECTR
02015 006265 A51.1: WRTNS
02016 000102 102
02017 006304 TIMER
000012 .RDX 10
02020 001750 1000
000010 .RDX 8
02021 000001 1
02022 006272 EHALT ;NO TUR 1 SEC FOLLOWING
;WRITE COMMAND
02023 014232 DSZ RECTR
02024 000771 JMP A51.1
02025 020154 LDA 0,0500
02026 100400 NEG 0,0
02027 063030 DDC 0,MTG ;WC=500
02030 102400 SUB 0,0
02031 062030 DOB 0,MTG ;ADDR. REG= 0
02032 020211 LDA 0,C3X
02033 061130 DOAS 0,MTG ;START BACKSPACE
02034 102400 SUB 0,0
02035 065430 A51.2: DIB 1,MTG
02036 065430 DIB 1,MTG
02037 101404 INC 0,0,SZR
02040 000775 JMP .-3
02041 064430 DIA 1,MTG
02042 125203 MOVR 1,1,SNC
02043 000772 JMP A51.2
02044 061430 DIB 0,MTG
02045 024154 LDA 1,0500
02046 122414 SUB# 1,0,SZR
02047 006272 EHALT ;ADDRESS REG. WRONG.
;SHOULD HAVE COUNTED 500
;REC ON BACKSPACE.
;AC1=EXPECTED COUNT, AC0=ACTUAL
02050 006262 BSPACE
02051 064430 DIA 1,MTG
02052 125203 MOVR 1,1,SNC
02053 000776 JMP .-2
02054 020111 LDA 0,C100
02055 107405 AND 0,1,SNR
02056 006272 EHALT ;BACKSPACE 1 DID NOT PRODUCE
;BOT. ADDR REG COUNTED 500.
;WC REG WAS COUNTING FASTER
;BAD WC REG.
02057 006271 JSR @LOOP

```

```

;SPACING CHECK.
;DURING SPACING MANY IO COMMANDS ARE
;ISSUED TO GENERATE NOISE.
02060 006267 A52: JSR @SETP1 ;WRITE 500 2ND REC.
02061 006263 REWIND ;SPACEBACK 500
02062 006304 TIMER ;CHECK CA
000012 .RDX 10
02063 0047040 20000 ;SPACEBACK 1 AND
000010 .RDX 8
02064 000201 201 ;CHECK FOR BOT
02065 006272 EHALLT ;NO BOT-TUR 20 SEC.
;FOLLOWING REWIND
02066 020154 LDA 0,DS00
02067 040232 STA 0,RECTR
02070 006265 A52.1: WRINS
02071 000102 102
02072 006304 TIMER
000012 .RDX 10
02073 001750 1000
000010 .RDX 8
02074 000001 1
02075 006272 EHALLT ;NO TUR 1 SEC FOLLOWING
;WRITE COMMAND
02076 014232 DSZ RECTR
02077 000771 JMP A52.1
02100 020154 LDA 0,DS00
02101 100400 NEG 0,0
02102 063030 DOC 0,MTC ;WC=500
02103 102400 SUB 0,0
02104 062030 DOB 0,MTC ;ADDR. REG= 0
02105 020211 LDA 0,C3X
02106 061130 DOAS 0,MTC ;START BACKSPACE
02107 102400 SUB 0,0
02110 066430 A52.2: DIC 1,MTC
02111 066430 DIC 1,MTC
02112 101404 INC 0,0,SZR
02113 000775 JMP .-3
02114 064430 DIA 1,MTC
02115 125203 MOVR 1,1,SNC
02116 000772 JMP A52.2
02117 061430 DIA 0,MTC
02120 024154 LDA 1,DS00
02121 122414 SUB# 1,0,SZR
02122 006272 EHALLT ;ADDRESS REG. WRONG.
;SHOULD HAVE COUNTED 500
;REC ON BACKSPACE.
;AC1=EXPECTED COUNT, AC0=ACTUAL
02123 006262 BSPACE
02124 064430 DIA 1,MTC
02125 125203 MOVR 1,1,SNC
02126 000776 JMP .-2
02127 020111 LDA 0,C100
02130 107405 AND 0,1,SNR
02131 006272 EHALLT ;BACKSPACE 1 DID NOT PRODUCE
;BOT. ADDR REG COUNTED 500.
;WC REG WAS COUNTING FASTER
;BAD WC REG.
02132 006271 JSR @LOOP ;SPACING CHECK.

```

```

;DURING SPACING MANY IO COMMANDS ARE
;ISSUED TO GENERATE NOISE.
02133 006267 A53: JSR @SETP1 ;WRITE 500 2ND REC.
02134 006263 REWIND ;SPACEBACK 500
02135 006304 TIMER ;CHECK CA
000012 .RDX 10
02136 047040 20000 ;SPACEBACK 1 AND
000010 .RDX 8
02137 000201 201 ;CHECK FOR BOT
02140 006272 EHALLT ;NO BOT-TUR 20 SEC.
;FOLLOWING REWIND
02141 020154 LDA 0,D500
02142 040232 STA 0,RECTR
02143 006265 A53.1: WRINS
02144 000102 102
02145 006304 TIMER
000012 .RDX 10
02146 001750 1000
000010 .RDX 8
02147 000001 1
02150 006272 EHALLT ;NO TUR 1 SEC FOLLOWING
;WRITE COMMAND
02151 014232 DSZ RECTR
02152 000771 JMP A53.1
02153 020154 LDA 0,D500
02154 100400 NEG 0,0
02155 063030 DDC 0,MTC ;WC=500
02156 102400 SUB 0,0
02157 062030 DOB 0,MTC ;ADDR. REG= 0
02160 020211 LDA 0,C3X
02161 061130 DOAS 0,MTC ;START BACKSPACE
02162 102400 SUB 0,0
02163 060330 A53.2: NIOP MTC
02164 060330 NIOP MTC
02165 101404 INC 0,0,SZR
02166 000775 JMP .-3
02167 064430 DIA 1,MTC
02170 125203 MOVR 1,1,SNC
02171 000772 JMP A53.2
02172 061430 DIB 0,MTC
02173 024154 LDA 1,D500
02174 122414 SUB# 1,0,SZR
02175 006272 EHALLT ;ADDRESS REG. WRONG.
;SHOULD HAVE COUNTED 500
;REC ON BACKSPACE.
;AC1=EXPECTED COUNT, AC0=ACTUAL
02176 006262 BSPACE
02177 064430 DIA 1,MTC
02200 125203 MOVR 1,1,SNC
02201 000776 JMP .-2
02202 020111 LDA 0,C100
02203 107405 AND 0,1,SNR
02204 006272 EHALLT ;BACKSPACE 1 DID NOT PRODUCE
;BOT. ADDR REG COUNTED 500.
;WC REG WAS COUNTING FASTER
;BAD WC REG.
02205 006271 JSR @LOOP

```

III

02206	006263	ETST:	REWIND
02207	006245		JSR @ICRLF
02210	006246		JSR @IMESS
02211	002220		RUNCO
02212	006254		PIP
02213	006300		WAIT
02214	000064		.20SEC
02215	006245		JSR @ICRLF
02216	002401		JMP @.+1
02217	000411		AN
02220	052522	RUNCO:	.TXT!PU
02221	020116	N	
02222	047503	CO	
02223	050115	MP	
02224	042514	LE	
02225	042524	TE	
02226	000104	DI	

III

```

02227 020511 ;LONGITUDINAL PARITY CHECK
02230 040447 LDA 0,LSTRT ;CAUSE LONG PARITY ERROR
02231 102400 STA 0,LPOIT ;IN 2 TRK PAIRS
02232 040444 SUB 0,0 ;CHECK PARITY STATUS
02233 006256 STA 0,RBCT
02234 006260 ERASE
02235 037477 A54: GEN
02236 020053 37477
02237 101400 LDA 0,ORUFF
02240 040020 INC 0,0
02241 022436 STA 0,IDX0
02242 101005 LDA 0,LPOIT
02243 000435 MOV 0,0,SNR
02244 042020 JMP CKED
02245 102400 A54.1: STA 0,IDX0
02246 024156 SUB 0,0
02247 042020 LDA 1,N400
02250 125404 STA 0,IDX0
02251 000776 INC 1,1,SZR
02252 006267 JMP .-2
02253 020053 JSR #SETP1
02254 062030 LDA 0,ORUFF
02255 020156 DOB 0,MTC
02256 063230 LDA 0,N400
02257 020114 DOCC 0,MTC
02260 024206 LDA 0,C150
02261 123000 LDA 1,CX
02262 061130 ADD 1,0
02263 006300 DOAS 0,MTC
02264 000070 WAIT
02265 030411 .400MS
02266 060430 LDA 2,RBCT
02267 024123 DIA 0,MTC
02270 123415 LDA 1,C2000
02271 006272 AND# 1,0,SNR
;NO PARITY ERROR, # IN AC2
;INDICATES WHICH TRACKS SHOULD
;CAUSE ERROR, (SEE BELOW)
;0=RB2-P 6=RB2-3 14=RB0-P
;1=RB3-P 7=RB3-4 15=RB1-P
;2=RB4-P 10=RB4-5 16=RB0-1
;3=RB5-P 11=RB5-6 17=RB0-2
;4=RB6-P 12=RB6-7 20=RB1-2
;5=RB7-P 13=RB7-2
02272 006271 JSR @LOOP

```

111

```

02273 010404      ISZ LPOIT
02274 010402      ISZ RRCT
02275 000737      JMP AS4
02276 000000      RRCT: 0
02277 000000      LPOIT: 0
02300 060430      CKED: DIA 0, MTC
02301 024111      LDA 1, C103
02302 010775      ISZ LPOIT
02303 123405      AND 1, 0, SNR
02304 000404      JMP .+4
02305 022772      LDA 0, 0, LPOIT
02306 101004      MOV 0, 0, SZR
02307 000735      JMP AS4.1
02310 006263      REWIND
02311 006237      STATUS
02312 020000      020000
02313 000776      JMP .-2
; LONGITUDINAL PARITY DATA LIST
; 7-9 TRACK
02314 024000      LST79: 4000 ; R02-P
02315 010000      10000 ; 3-P
02316 020000      20000 ; 4-P
02317 000400      400 ; 5-P
02320 001000      1000 ; 6-P
02321 002000      2000 ; 7-P
02322 030000      30000 ; 2-3
02323 014000      14000 ; 3-4
02324 006000      6000 ; 4-5
02325 003000      3000 ; 5-6
02326 001400      1400 ; 6-7
02327 000600      600 ; 7-2
02330 000000      0
; 9 TRACK ONLY
02331 100000      LST9: 100000 ; 0-P
02332 040000      40000 ; 1-P
02333 140000      140000 ; 0-1
02334 120000      120000 ; 0-2
02335 060000      60000 ; 1-2
02336 000000      0
02337 000000      0
02340 002314      LSTRT: LST79

```

111

```

;TAPE 5
;MTA DIAGNOSTIC INITIALIZATION
02341 042523 SWPAR: .TXT!SE
02342 020124 I
02343 053523 SW
02344 041524 IC
02345 020123 S
02346 020046 &
02347 047503 CO
02350 052116 NT
02351 020000 I
02352 054510 INITIAL:STA 3,IRET
02353 006245 JSR @ICRLF
02354 006246 JSR @IMESS
02355 002341 SWPAR
02356 006255 HAATT
02357 063077 HALT ;SET SWITCHES & CONTINUE
02360 006245 JSR @ICRLF
02361 020233 LDA 0,DEVICE
02362 040234 STA 0,OLDDDEV
02363 074477 READS 3
02364 161000 MOV 3,0 ;SAVE SWR IN AC0 AND AC3
02365 024077 LDA 1,C7
02366 030100 LDA 2,C10
02367 137400 AND 1,3 ;AC3=UNIT #
02370 024055 LDA 1,PONCX
02371 044204 STA 1,CONCX
02372 024116 LDA 1,C200
02373 167000 ADD 3,1
02374 056204 NEXCX: STA 3,@CONCX
02375 010204 ISZ CONCX
02376 157000 ADD 2,3
02377 166414 SUB# 3,1 SZR
02400 000774 JMP NEXCX
02401 101200 MOVR 0,0
02402 101200 MOVR 0,0
02403 101200 MOVR 0,0 ;DEVICE:=SWR SHIFT (-3)
02404 030110 LDA 2,C77 ;EXTRACT B;
02405 143400 AND 2,0
02406 040233 STA 0,DEVICE

;MODIFY ALL MTC INSTRUCTIONS
02407 030050 LDA 2,FIRST
02410 021000 GMOR: LDA 0,0,2
02411 024137 LDA 1,C160077
02412 123400 AND 1,0
02413 024140 LDA 1,C060000
02414 034234 LDA 3,OLDDDEV
02415 167000 ADD 3,1
02416 106414 SUB# 0,1,SZR
02417 000407 JMP DEV1 ;NOT MTA
02420 021000 LDA 0,0,2
02421 024141 LDA 1,C177700
02422 123400 AND 1,0
02423 024233 LDA 1,DEVICE
02424 123000 ADD 1,0
02425 041000 STA 0,0,2
02426 151400 DEV1: INC 2,2
02427 020051 LDA 0,CEND

```

02430 112414
02431 020757
02432 020225
02433 061130
02434 006263

SUB# 0,2,SZR
JMP GMOR
LDA W,C17X
DOAS W,MTC
REWIND

```

02435 020111          LDA 0,C100          ;FIND CPUNO
02436 126400          SUB 1,1             ;WRITE(DIS,0);
02437 061135          DOAS 0,DIS          ;NOVA=12.6 US,0
02440 125400          INC 1,1 ;SUPER=4.5 US, 3
02441 063535          SKPBZ DIS           ;SUSC=3.7 US, 4
02442 000776          JMP ,-2 ;800=3.2 US, 5
02443 030157          LDA 2,N1250         ;STEP -1250
02444 034155          LDA 3,N8            ;7 TIMES
02445 141000          MOV 2,0 ;STARTING AT -1250=SPEED
02446 175405          NXTYP: INC 3,3 SNR   ;NEXT STEP OR
02447 004414          JSR NOTYP           ;TYPE OF NOVA NOT FOUND
02450 143020          ADDZ 2,0            ;SPEED :=SPEED-1250
02451 107013          ADD# 0,1 SNC        ;NO. OF SKPBZ OF DIS
02452 000774          JMP NXTYP           ;< SPEED THEN GOTO NXTYP ELSE
02453 030077          LDA 2,C7            ;CONVERT TIMES TO
02454 173000          ADD 3,2 ;CPU NO. AS MENTIONED
02455 050227          STA 2,CPUNO         ;ABOVE
02456 006245          JSR @ICRLF
02457 030072          LDA 2,C2
02460 050227          STA 2,CPUNO
02461 002401          JMP @IRET           ;RETURN
02462 000000          IRET: 0 ;RETURN ADDRESS
02463 020075          NOTYP: LDA 0,C5
02464 117000          ADD 0,3
02465 054407          STA 3,NOTRE
02466 006245          JSR @ICRLF
02467 006246          JSR @IMESS
02470 002475          NOTYM
02471 006255          HAATT
02472 063077          HALT ;PUT CPUNO INTO AC2
                                ;SEE ABOVE
02473 002401          JMP @NOTRE
02474 000000          NOTRE: 0
02475 042523          NOTYM: ,TXTISE
02476 020124          T
02477 050103          CP
02500 047125          UN
02501 020117          0
02502 020076          >
02503 041501          AC
02504 000062          2!

```

```

;COMPARE IBUFF WITH OBUFF
;
;      CHECK
;      ARG
;
;# WORDS IS IN ARG FOLLOWING CALL
;COMPARE WORD FOR WORD
;WORD N+1 IN IBUFF MUST BE ZERO
;IF ERROR RETURN TO CALL +2
;OTHERWISE CALL+3
;EXIT WITH AC1=BAD WORD
;AC2=GOOD WORD
;AC2=ADDR. OF IBUFF
02505 054440 XCHK: STA 3,RCHK
02506 024053 LDA 1,OBUFF
02507 044022 STA 1,IDX2
02510 014022 DSZ IDX2
02511 024054 LDA 1,IBUFF
02512 044023 STA 1,IDX3
02513 014023 DSZ IDX3
02514 031400 LDA 2,0,3 ;GET ARG
02515 050427 STA 2,CNTR
02516 022022 XCHK,: LDA 0,0,IDX2
02517 026023 LDA 1,0,IDX3
02520 106414 SUB# 0,1,SNR
02521 000406 JMP LEAV
02522 014422 DSZ CNTR
02523 000773 JMP XCHK,
02524 026023 LDA 1,0,IDX3
02525 125005 MOV 1,1,SNR ;WORD FOLLOWING LAST
;MUST BE ZERO
02526 010417 ISZ RCHK
02527 010416 LEAV: ISZ RCHK
02530 030054 LDA 2,IBUFF
02531 002414 JMP 0RCHK

;CLEAR INPUT BUFFER TO ZEROS
02532 020054 XCLR: LDA 0,IBUFF
02533 040022 STA 0,IDX2
02534 014022 DSZ IDX2
02535 020111 LDA 0,C100
02536 040406 STA 0,CNTR
02537 102400 SUB 0,0
02540 042022 STA 0,0,IDX2
02541 014403 DSZ CNTR
02542 000776 JMP .-2
02543 001400 JMP 0,3
02544 000000 CNTR: 0
02545 000000 RCHK: 0

```

III

```

;DO A SPACE FORWARD
02546 024210 XSPC: LDA 1,C2X
02547 065130 DOAS 1,MTC ;DO IT
02550 001400 JMP 0,3

;REWIND SUBROUTINE
02551 024215 XRWD: LDA 1,C7X
02552 065130 DOAS 1,MTC
02553 001400 JMP 0,3

;REWIND FOLLOWED BY A 5 SEC STALL
02554 054406 XRWS: STA 3,RRWS
02555 024215 LDA 1,C7X
02556 065130 DOAS 1,MTC
02557 006300 WAIT
02560 000062 .5SEC
02561 002401 JMP 0RRWS
02562 000000 RRWS: 0

;BACKSPACE ROUTINE
02563 024211 XBSP: LDA 1,C3X
02564 065130 DOAS 1,MTC
02565 001400 JMP 0,3

;WRITE EOF
02566 054406 XWEOF: STA 3,RWEOF
02567 024212 LDA 1,C4X
02570 065130 DOAS 1,MTC
02571 006300 WAIT
02572 000070 .400MS
02573 002401 JMP 0RWEOF
02574 000000 RWEOF: 0

;BAD TAPE STATUS WRITE BUFFER
02575 020001 BTBF: 020001 ;1
02576 010002 010002 ;2
02577 004004 004004 ;3
02600 002010 002010 ;4
02601 000000 000000 ;5
02602 000002 000002 ;6
02603 000420 000420 ;7
02604 020040 020040 ;8

;CRC CHECK WRITE BUFFERS
02605 000200 PAT1: 000200
02606 000200 000200
02607 153600 153600
02610 177474 PAT2: 177474
02611 136074 136074
02612 112074 112074
02613 052652 PAT3: 052652
02614 076232 076232
02615 037232 037232
02616 167141 PAT4: 167141
02617 000200 000200
02620 063600 063600

```

III

```

02621 030225 XBSTA: ;GIVE MT-START BEFORE GET STATUS
02622 071130 LDA 2,C17X
DOAS 2, MTC

;GET ARGUMENT STATUS
;AND CHECK THAT ALL OF
;THE BITS ARE PRESENT IN THE ACTUAL STATUS
;RETURN+2 IF BITS ARE PRESENT, OTHERWISE +1
;EXIT WITH AC1=EXPECTED, AC0=ACTUAL BITS
;
; ASTATUS
; ARG

02623 054410 XASTA: STA 3,RXAS
02624 010407 ISZ RXAS
02625 025400 LDA 1,0,3 ;AC1=EXPECTED BITS
02626 060430 DIA 0,MTC
02627 123400 AND 1,0
02630 122415 SUB# 1,0,SNR
02631 010402 ISZ RXAS ;OK
02632 002401 JMP 0RXAS
02633 000000 RXAS: 0

;GIVE MT-START BEFORE FETCH STATUS
02634 030225 XDSTA: LDA 2,C17X
02635 071130 DOAS 2, MTC

;FETCH ARGUMENT STATUS
;AND CHECK THAT NONE OF
;THE BITS ARE PRESENT IN ACTUAL STATUS
;RETURN +1 IF BITS PRESENT, OTHERWISE +2
;EXIT WITH AC1=NON-EXPECTED, AC0=ACTUAL STATUS
;
; NSTATUS
; ARG

02636 054407 XNSTA: STA 3,RXNS
02637 010406 ISZ RXNS
02640 025400 LDA 1,0,3 ;AC1=NONEXPECTED STATUS
02641 060430 DIA 0,MTC ;AC0=ACTUAL STATUS
02642 107415 AND# 0,1,SNR
02643 010402 ISZ RXNS ;OK
02644 002401 JMP 0RXNS
02645 000000 RXNS: 0

```

111

```

;LOAD OBUFF WITH 3 WORDS.
;
;      LOAD
;      ARG
;
;ARGUMENT=ADDR OF 3 WORD FIELD
02646 031400 XLD: LDA 2,0,3
02647 050023 STA 2,IDX3
02650 030053 LDA 2,OBUFF
02651 050024 STA 2,IDX4
02652 014023 DSZ IDX3
02653 014024 DSZ IDX4
02654 024147 LDA 1,M3
02655 032023 LDA 2,0,IDX3
02656 052024 STA 2,0,IDX4
02657 125404 INC 1,1,SRZ
02660 000775 JMP .-3
02661 175400 INC 3,3
02662 001400 JMP 0,3

;GENERATE ARGUMENT DATA INTO OBUFF
;ALWAYS GENERATE 64 WORDS ALL SAME
;
;      GEN
;      ARG
;
02663 054414 XGEN: STA 3,RGEN
02664 010413 ISZ RGEN
02665 031400 LDA 2,0,3 ;GET DATA WORD
02666 024053 LDA 1,OBUFF
02667 044020 STA 1,IDX0 ;SET IR0
02670 014020 DSZ IDX0
02671 024152 LDA 1,M100 ;LOAD 64 COUNTER
02672 044226 STA 1,CTR
02673 052020 STA 2,0,IDX0
02674 010226 ISZ CTR
02675 000776 JMP .-2
02676 002401 JMP 0,RGEN
02677 000000 RGEN: 0

;WAIT 50MS THEN SELECT UNIT
02700 054406 XSEL: STA 3,RSEL
02701 006300 WAIT
02702 000065 .50MS
02703 020213 LDA 0,CSX
02704 061030 DOA 0,MTX
02705 002401 JMP 0,RSEL
02706 000000 RSEL: 0

```

```

;READ NO STALL (RFDNS)
;JUST LIKE (READ) EXCEPT THAT ONLY
;ONE RECORD IS READ AND A RETURN
;IS EXECUTED IMMEDIATELY.
02707 102400 XRDNS: ADC 0,0
02710 040441 STA 0,RDSW ;SET SWITCH
02711 000403 JMP XRD+2

;READ SUBROUTINE
;ARGUMENT FOLLOWING CALL SPECIFIES
;#WORDS, AND PARITY TYPE
;BITS 10-15=WORDS BIT 0= 1 FOR EVEN
;BITS 4-9=RECORDS 0 FOR ODD
;1 SEC DELAY BETWEEN EACH READ.
;NO STATUS,BUSY,OR DONE CHECKING!
02712 102400 XRD: SUB 0,0
02713 040436 STA 0,RDSW
02714 054434 STA 3,RRD
02715 010433 ISZ RPD
02716 031400 LDA 2,0,3
02717 020110 LDA 0,C77
02720 113400 AND 0,2
02721 150400 NEG 2,2
02722 050230 STA 2,WDCNT ;SAVE WORD COUNT
02723 031400 LDA 2,0,3
02724 020206 LDA 0,CX
02725 040231 STA 0,INST
02726 020130 LDA 0,C7700
02727 143400 AND 2,0 ;AC0=#RRCX100
02730 024111 LDA 1,C100 ;AC1=100
02731 060230 XRD.1: NIQC MTC
02732 030230 LDA 2,WDCNT
02733 073030 DOC 2,MTC ;SET WC
02734 030054 LDA 2,IBUFF
02735 072030 DOB 2,MTC ;SET CA
02736 030231 LDA 2,INST
02737 071130 DOAS 2,MTC ;START READ
02740 030411 LDA 2,RDSW ;SHECK SWITCH
02741 151004 MOV 2,2,SZR
02742 002406 JMP 0RRD
02743 006300 WAIT
02744 000060 .1SEC
02745 122404 SUB 1,0,SZR
02746 000763 JMP XRD.1
02747 002401 JMP 0RRD
02750 000000 RRD: 0
02751 000000 RDSW: 0

```

111

```

;WRITE NO STALL (WRTNS)
;JUST LIKE (WRITE) EXCEPT THAT ONLY
;ONE RECORD IS WRITTEN AND RETURN
;IS EXECUTED IMMEDIATELY
02752 102000 XWNS: ADC 0,0
02753 040441 STA 0,SWTCH
02754 000403 JMP XWRT+2

;WRITE SUBROUTINE
;ARGUMENT FOLLOWING CALL SPECIFIES
;#RECORDS AND WORDS, AND PARITY.
;BITS10-15=#WORDS BIT 0 =1FOR EVEN
;BITS 4- 9=#RECORDS 0 FOR ODD
;1 SEC DELAY BETWEEN WRITES
;NO STATUS,BUSY,DONE,CHECKING!
02755 102400 XWRT: SUB 0,0
02756 040436 STA 0,SWTCH
02757 054434 STA 3,RWRT
02760 010433 ISZ RWRT
02761 031400 LDA 2,0,3
02762 020110 LDA 0,C77
02763 113400 AND 0,2
02764 150400 NEG 2,2
02765 050230 STA 2,WDCNT ;SAVE WORD COUNT
02766 031400 LDA 2,0,3
02767 020207 LDA 0,C1X
02770 040231 STA 0,INST ;SAVE INST
02771 020130 LDA 0,C7700
02772 143400 AND 2,0 ;AC0=#RECX100
02773 024111 LDA 1,C100
02774 060230 XWRT,1: NIOC MTC
02775 030230 LDA 2,WDCNT
02776 073030 DDC 2,MTC ;SET WC
02777 030053 LDA 2,0BUFF
03000 072030 DOB 2,MTC ;SET CA
03001 030231 LDA 2,INST
03002 071130 DOAS 2,MTC ;START WRITE
03003 030411 LDA 2,SWTCH ;CHECK SWITCH
03004 151004 MOV 2,2,SZR
03005 002406 JMP 0RWRT
03006 006300 WAIT
03007 000060 .1SEC
03010 122404 SUB 1,0,SZR
03011 000763 JMP XWRT,1
03012 002401 JMP 0RWRT
03013 000000 RWRT: 0
03014 000000 SWITCH: 0

```

III

```

;ERASE ROUTINE
03015 054406 XERAS: STA 3,XERAR
03016 020213 LDA 0,C5X
03017 061130 DOAS 0,MTC
03020 006300 WAIT
03021 000070 .400NS
03022 002401 JMP @XERAR
03023 000000 XERAR: 0

;CHECK STATUS FOR BOT OR OFF-LINE.
;AC0=UNIT BEING SELECTED.
;RETURN+1 IF ERROR
03024 054420 CSTAT: STA 3,CSTAR
03025 074477 READS 3
03026 024077 LDA 1,C7
03027 123400 AND 1,0 ;AC0=ACTUAL UNIT BEING SEL.
03030 137400 AND 1,3 ;AC3=UNIT IN SWR
03031 162415 SUB# 3,0,SNR
03032 000406 JMP ON ;SHOULD BE AT BOT
03033 006244 NSTATUS
03034 040000 040000
03035 010407 OKRET: ISZ CSTAR ;OK, RETURN+2
03036 030205 LDA 2,X170 ;X170=UNIT#+170
03037 002405 JMP @CSTAR
03040 006241 ON: ASTATUS
03041 010000 010000
03042 000774 JMP OKRET+1 ;ERROR,AC1=GOOD STAT.
03043 000772 JMP OKRET ; AC0=BAD
03044 000000 CSTAT: 0

;DELAY SUBROUTINE
;ARGUMENT FOLLOWING CALL IS ADDRESS
;OF DELAY CONSTANT.
;DELAY IN INCREMENTS OF 1MS
;AC2&3 ARE USED
03045 054415 XWT: STA 3,RDLAY
03046 010414 ISZ RDLAY
03047 033400 LDA 2,00,3
03050 050411 STA 2,COUNT
03051 034227 LDA 3,CPUNO
03052 031560 LDA 2,KNOVA,3
03053 050226 XWTLO: STA 2,CTR
03054 014226 DSZ CTR
03055 000777 JMP ,-1
03056 014403 DSZ COUNT
03057 000774 JMP XWTLO
03060 002402 JMP @RDLAY
03061 000000 COUNT: 0
03062 000000 RDLAY: 0

```

111

```

03063 030225 XSSTA: ;GIVE MT-START BEFORE COMBINE STATUS
03064 071130 LDA 2,C17X
DOAS 2,MTX

;COMBINE ARGUMENT STATUS WITH SWR
;STATUS AND CHECK AGAINST ACTUAL.
;RETURN+2 IF STATUS MATCH, +1 OTHERWISE
;EXIT WITH AC1=EXPECTED STATUS, AC0=ACTUAL
;
; STATUS
; ARG
;
03065 054413 XSTAT: STA 3,STATR
03066 010412 ISZ STATR
03067 025400 LDA 1,0,3
03070 074477 READS 3
03071 030124 LDA 2,C4K ;PE=1
03072 157400 AND 2,3
03073 167000 ADD 3,1 ;AC1=EXP STATUS
03074 060430 DIA 0,MTX ;AC0=ACTUAL STATUS
03075 106415 SUB# 0,1,SNR
03076 010402 ISZ STATR ;OK
03077 002401 JMP 0,STATR
03100 000000 STATR: 0

;TIMER SUBROUTINE
;
; TIMER
; TIME TO WAIT (MAX)
; TYPE OF STATUS SUBROUTINE
; EXIT STATUS
; TIMEOUT RETURN
; GOOD RETURN
;
03101 054425 XTIM: STA 3,RTIM
03102 033400 LDA 2,00,3 ;GET TIME
03103 050424 STA 2,TIMCTR
03104 010422 ISZ RTIM ;GET STATUS TYPE
03105 032421 LDA 2,0RTIM
03106 050405 STA 2,XTIM.
03107 010417 ISZ RTIM ;GET STATUS
03110 032416 LDA 2,0RTIM
03111 050403 STA 2,TSTAT
03112 010414 ISZ RTIM
03113 006237 XTIM.: STATUS
03114 000000 TSTAT: 0
03115 000402 JMP .+2 ;WRONG STATUS
03116 000406 JMP XTIM1 ;CORRECT STATUS
03117 006300 WAIT ;STALL 1 MS
03120 000071 C1
03121 014406 DSZ TIMCTR
03122 000771 JMP XTIM.
03123 002403 JMP 0RTIM
03124 010402 XTIM1: ISZ RTIM
03125 002401 JMP 0RTIM
03126 000000 RTIM: 0 ;RETURN
03127 000000 TIMCTR: 0 ;TIME TO COUNT

```

```

;DATA LATE TEST
;CONTINUALLY WRITE 4K RECORDS.
;CHECK STATUS AFTER EACH WRITE
;AND PRINT IT OUT. DO CONSOLE
;STOP-CONTINUE TO CAUSE DATA LATE.

03130 006236 DLT: JSR @INTIL
03131 006263 REWIND
03132 006242 BSTATUS
03133 010000 010000
03134 000776 JMP .-2 ;LOOK FOR BOT
03135 062677 DLT.1: IORST ;SET CA & WC TO 0
03136 020207 LDA 0,CIX
03137 061130 OAS 0,MTC ;START WRITE
03138 063630 SKPDN MTC
03141 000777 JMP .-1
03142 006246 JSR @IMESS
03143 003165 MDL
03144 064430 DIA 1,MTC ;GET STATUS
03145 006247 JSR @IPOCT ;PRINT IT
03146 006253 ATT
03147 006245 JSR @ICRLF
03150 006241 ASTATUS
03151 000100 000100 ;DATA LATE
03152 000410 JMP NODL ;NO
03153 006246 YDL: JSR @IMESS ;YES
03154 003175 MDL2
03155 006253 ATT
03156 006241 ASTATUS
03157 000020 000020 ;EOT
03160 000755 JMP DLT.1 ;NO
03161 000750 JMP DLT+1 ;YES
03162 006246 NODL: JSR @IMESS
03163 003173 MDL1
03164 000767 JMP YDL
03165 005215 MDL: .TXTE 1<15><12>
03166 152123 ST
03167 152101 AT
03170 051525 US
03171 120240
03172 000000 !
03173 147516 MDL1: .TXTE !NO
03174 000240 !
03175 040504 MDL2: .TXTE !DA
03176 040724 TA
03177 146240 L
03200 152101 AT
03201 000305 EI

```

03362 042101 AD
03363 000000 I
03364 005215 MWLS: .TXTEI<15><12>
03365 142722 RE
03366 027115 M.
03367 151240 R
03370 047311 IN
03371 120107 G
03372 120246 &
03373 147703 CO
03374 152116 NT
03375 000000 I
03376 000000 SAC0: 0
03377 000000 SAC1: 0

03400 000000 ENDPRG: 0

111

```

;TAPE 6
03401 054435 ENIP1: STA 3,LOOPR
03402 034420 LDA 3,ITRST
03403 054420 STA 3,ITRAG
03404 176520 SUBZL 3,3
03405 000405 JMP ENTCON
03406 054430 ENTER: STA 3,LOOPR ;LOOP ITERATE RETURN
03407 034412 LDA 3,ITRRE
03408 054413 STA 3,ITRAG
03409 034411 LDA 3,ITRST ;THIS ROUTINE INITIALIZES
03410 054412 ENTCON: STA 3,ITR
03411 054412 STA 3,ITRCT ;EACH TEST
03412 176400 SUB 3,3
03413 054411 STA 3,ESWIT
03414 054411 STA 3,ERRCT
03415 062677 IORST ;I/O RESET
03416 002416 JMP 0LOOPR

03421 000144 ITRRESET: 144
03422 000012 ITRSTART: 12
03423 000000 ITRAG: 0
03424 000000 ITR: 0
03425 000000 ITRCT: 0
03426 000000 ESWIT: 0 ;FIRSTERROR
03427 000000 ERRCT: 0
03428 000000 RETURN: 0
03429 000000 SAV2: 0
03430 000000 SAV1: 0
03431 000000 SAV0: 0
03432 000040 CH40: 40
03433 000245 PCENT: 245
03434 000000 LOOPR: 0
03435 054771 CYCLE: STA 3,RETURN ;END OF TEST ITERATION
03436 050771 STA 2,SAV2 ;ROUTINE
03437 044771 STA 1,SAV1 ;SAVE THE ACS'
03438 040771 STA 0,SAV0
03439 014762 DSZ ITRCT
03440 000433 JMP CYCTS ;NOT N TIMES ITERATED
03441 030761 LDA 2,ESWIT ;IF SWITCH 2=(1)
03442 151005 MOV 2,2,SNR ;AND A ERROR HAS OCCURRED
03443 000442 JMP NOEX ;THE ERROR RATE WILL
03444 074477 HEADS 3 ;BE PRINTED
03445 175120 MOVZL 3,3
03446 175100 MOVL 3,3
03447 175103 MOVL 3,3,SNC
03448 000416 JMP ZERRC

03449 020757 LDA 0,CH40
03450 006250 JSR 01CHAR
03451 024750 LDA 1,ERRCT
03452 030113 LDA 2,C144
03453 004527 JSR MULT ;AC1XAC2
03454 030742 LDA 2,ITR
03455 004513 JSR DIVID ;AC0-1?AC2
03456 006252 JSR 0IPDEC ;PRINT VALUE
03457 020750 LDA 0,PCENT ;EXAMPLE: 89%
03458 006250 JSR 01CHAR
03459 063635 SKPDN .TTO ;WAIT DONE FLAG
03460 000777 JMP ,-1

```

111

```

;END TAPE TEST. WRITE TO EOT AND
;HALT AFTER MESSAGE. NOTE ERRORS
;ALONG THE WAY
03202 006236 ETT: JSR @INTIL
03203 062677 IORST ;SET CA & WC TO 0
03204 020207 LDA 0,C1X
03205 061130 DOAS 0,MTC ;WRITE
03206 063630 SKPDN MTC
03207 000777 JMP , -1
03210 006244 NSTATUS
03211 000020 000020
03212 000413 JMP ENDT ;END TAPE
03213 006244 NSTATUS
03214 173500 173500
03215 000402 JMP , +2
03216 000765 JMP ETT+1 ;NO EOT, NO ERRORS
03217 006246 JSR @IMESS
03220 003253 MSTAR
03221 064430 DIA 1,MTC
03222 006247 JSR @IPOCT
03223 006253 ATT
03224 000757 JMP ETT+1
03225 006246 ENDT: JSR @IMESS
03226 003261 MENDT
03227 006244 NSTATUS
03230 173500 173500
03231 000402 JMP , +2
03232 000405 JMP ENDT2
03233 006246 JSR @IMESS
03234 003264 MESTA
03235 064430 DIA 1,MTC
03236 006247 JSR @IPOCT
03237 006255 ENDT2: HAATT
03240 063077 HALT
03241 006262 BSPACE
03242 006300 WAIT
03243 000062 .5SEC
03244 006262 BSPACE
03245 006300 WAIT
03246 000062 .5SEC
03247 006262 BSPACE
03250 006300 WAIT
03251 000062 .5SEC
03252 000731 JMP ETT+1
03253 005215 MSTAR: .TXTE !<15><12>
03254 151327 WR
03255 051640 S
03256 040724 TA
03257 120324 T
03260 000240 I
03261 005215 MENDT: .TXTE !<15><12>
03262 147705 EQ
03263 000324 TI
03264 020054 MESTA: .TXT!,
03265 052123 ST
03266 052101 AT
03267 000040 !

```

III

```

03270 006246 WLT: ;WRITE LOCK TEST
03271 003364 JSR @IMESS
03272 006255 MWLS
03273 063077 HAATT
03274 006245 HALT
03275 006236 JSR @ICRLF
03276 062677 REP: JSR @INTIL
03277 006263 IORST
03300 006242 REWIND
03301 010000 BSTATUS
03302 000776 010000
03303 006257 SOK: JMP .-2 ;LOOK FOR BOT
03304 000102 WRITE
03305 063630 102
03306 000777 SKPDN MTC
03307 006241 JMP .-1
03310 013000 ASTATUS
03311 000421 013000
03312 006261 JMP ERWL ;BAD STATUS
03313 063630 WEOF
03314 000777 SKPDN MTC
03315 006241 JMP .-1
03316 013000 ASTATUS
03317 000413 013000
03320 006256 JMP ERWL
03321 063630 ERASE
03322 000777 SKPDN MTC
03323 006241 JMP .-1
03324 013000 ASTATUS
03325 000405 013000
03326 006246 JMP ERWL
03327 003343 JSR @IMESS ;GOOD STATUS
03330 006253 MWLO
03331 000745 ATT
03332 040444 ERWL: JMP REP
03333 044444 STA 0,SAC0
03334 006246 STA 1,SAC1
03335 003353 JSR @IMESS ;AC1=EXPECTED STATUS
03336 006255 MWLB ;AC0=ACTUAL STATUS
03337 020437 HAATT
03340 024437 LDA 0,SAC0
03341 063077 LDA 1,SAC1
03342 000734 HALT
03343 005215 MWLO: JMP REP
03344 151327 .TXTE 1<15><12>
03345 152311 WR
03346 120305 IT
03347 147714 E
03350 045703 LO
03351 147640 CK
03352 000113 O
03353 005215 MWLB: .TXTE 1<15><12>
03354 151327 WR
03355 152311 IT
03356 120305 E
03357 147714 LO
03360 045703 CK
03361 041240 B

```

03471 006253
03472 102400
03473 040734
03474 034727
03475 054727
03476 054727

ZERRC:

ATT
SUB 0,0
STA 0,ERRCT
LDA 3,ITRAG
STA 3,ITR
STA 3,ITRCT

;RESET ERROR COUNT

111

03477	020734	CYCTS:	LDA 0,SAV0	;RESTORE ACS
03500	024732		LDA 1,SAV1	
03501	030730		LDA 2,SAV2	
03502	034724		LDA 3,ESWIT	
03503	175004		MOV 3,3,SNR	
03504	074477		READS 3	
03505	062677		IORST	
03506	175113		MOVL# 3,3,SNR	;SWITCH 0
03507	002727		JMP 0,LOOPR	; (0)=LOOP ROUTINE
03510	002720		JMP 0,RETURN	; (1)=PROCEED TO NEXT TEST
03511	020722	NOEX:	LDA 0,SAV0	;NORMAL EXIT, NO ERR.
03512	024720		LDA 1,SAV1	
03513	030716		LDA 2,SAV2	
03514	002714		JMP 0,RETURN	
03515	054713	ERR:	STA 3,RETURN	;ERROR SUBROUTINE
03516	034711		LDA 3,ERRCT	
03517	175005		MOV 3,3,SNR	
03520	000404		JMP ERR1	
03521	010706	ERET:	ISZ ERRCT	;COUNT
03522	101010		MOV# 0,0	;ERRORS
03523	002705		JMP 0,RETURN	;EXIT
03524	040707	ERR1:	STA 0,SAV0	
03525	044705		STA 1,SAV1	
03526	050703		STA 2,SAV2	
03527	074477		READS 3	
03530	175120		MOVZL 3,3	
03531	175102		MOVL 3,3 SZC	
03532	000433		JMP ERR3	
03533	006245		JSR 0,ICRLF	;PRINT CARRIAGE
03534	006246		JSR 0,IMESS	;AND HEADER
03535	003574		HEADER	
03536	020672		LDA 0,RETURN	
03537	126000		ADC 1,1	
03540	107000		ADD 0,1	
03541	004526		JSR POCT	;PC OF ERROR
03542	020664		LDA 0,ESWIT	
03543	101005		MOV 0,0,SNR	
03544	000406		JMP ERR2	
03545	020606		LDA 0,SAV0	
03546	024664		LDA 1,SAV1	
03547	030662		LDA 2,SAV2	;RESTORE ACS
03550	006253		ATT	
03551	000750		JMP ERET	
03552	006255	ERR2:	HAATT	
03553	020660		LDA 0,SAV0	
03554	024656		LDA 1,SAV1	
03555	030654		LDA 2,SAV2	
03556	034652		LDA 3,RETURN	;ERROR. C(3)=PC
03557	063077		HALT	;OPERATOR-SET SWITCHES
03560	054646		STA 3,ESWIT	
03561	020652		LDA 0,SAV0	
03562	024650		LDA 1,SAV1	
03563	030646		LDA 2,SAV2	
03564	000735		JMP ERET	
03565	020641	ERR3:	LDA 0,ESWIT	
03566	101005		MOV 0,0,SNR	
03567	000763		JMP ERR2	
03570	020643		LDA 0,SAV0	

03571 024641 LDA 1,SAV1
03572 030637 LDA 2,SAV2
03573 000726 JMP ERET
03574 041520 HEADER: .TXT IPC
03575 000040 !

111

```

03576 054424 DIVID: STA 3,MSAV      ;DIVIDE DOUBLE LENGTH AC
03577 034424      LDA 3,M20      ;AC0-1?AC2
03578 125120      MOVZL 1,1      ;RESULT IN AC1
03579 101100 DLOOP: MOVL 0,0      ;REMAINDER IN AC 0
03580 142412      SUB# 2,0,SZC
03581 142400      SUB 2,0
03582 125100      MOVL 1,1
03583 175404      INC 3,3,SZR
03584 000773      JMP DLOOP
03585 002413      JMP #MSAV
03586 102460 MULT: SUBC 0,0      ;MULTIPLY
03587 054411      STA 3,MSAV      ;AC1XAC2
03588 125411      LDA 3,M20      ;RESULT IN DOUBLE
03589 125203 MLOOP: MOVR 1,1,SNC ;LENGTH. AC: AC0-1
03590 101201      MOVR 0,0,SKP
03591 143220      ADDZR 2,0
03592 175404      INC 3,3,SZR
03593 000774      JMP MLOOP
03594 125260      MOVR 1,1
03595 002401      JMP #MSAV
03596 000000 MSAV: 0
03597 177760 M20: -20

```

;RC3600, ATTENTION DISPLAY OUTPUT
;BY MEANS OF ACOUSTIC ALARM FOR 50 MS
;AND WAIT FOR 10 SECONDS TO ALLOW
;OPERATOR TO READ THE MESSAGE

```

03624 054405 OUT36: STA 3,OUTRE
03625 006254      PIP
03626 006300      WAIT
03627 000063      .10SEC
03628 002401      JMP #OUTRE
03629 000000 OUTRE: 0

```

```

03632 054410 OPIP: STA 3,OPIPR
03633 020124      LDA 0,C4K
03634 061032      DOA 0,FUN
03635 006300      WAIT
03636 000065      .50MS
03637 102400      SUB 0,0
03638 061032      DOA 0,FUN
03639 002401      JMP #OPIPR
03640 000000 OPIPR: 0

```

;RC3600, ATTENTION HALT
;BY MEANS OF ACOUSTIC ALARM FOR 50 MS

```

03643 054766 HAAT: STA 3,OUTRE
03644 006254      PIP
03645 002764      JMP #OUTRE

```

```

;TTO NON INTERRUPT PACKAGE
;"MESS" PRINTS ASCII MESSAGES AS SPECIFIED BY ASSEMBLER
;"CHAR" PRINTS ASCII CHARACTER,C(0)R,C(0)L MUST BE 0
;WILL RETURN +2 IF C(0)R=0,CORRECTS THE PARITY,11 SIMULATE TAB
;"TYPE" PRINTS C(0)R, MUST HAVE PROPER PARITY. RETURN IS
;TO CALL+1,REPLACE THIS ROUTINE WITH INTERRUPT TYPE IF DESIRED.
;"CRLF" PRINTS A CARRIAGE RETURN
;"POCT" PRINTS C(1) IN OCTAL FOLLOWED BY A TAB
;"PDEC" PRINTS C(1) IN DECIMAL,LEADING ZEROS SUPPRESSED,
;FOLLOWED BY A TAB.
;"ZOCT" PRINTS C(1) IN OCTAL FOLLOWED BY A TAB,
;LEADING ZEROES ARE SUPPRESSED.
MESS:   STA 3,MESSR      ;PRINT A TEXT MESSAGE
        ISZ MESSR
03646 054546          LDA 2,0,3      ;C(2) POINTS TO MESSAGE
03647 010545          LDA 1,C377    ;A 8 BIT MASK
03650 031400          LDA 0,0,2      ;C(0)=DATA WORD
03651 024120          MOVL# 1,1,SZC
03652 021000          ANDS 1,0,SKP
03653 125112          AND 1,0,SKP    ;C(0)=DATA CHARACTER RIGHT
03654 123701          INC 2,2        ;INC TO NEXT WORD
03655 123401          COM 1,1        ;FLIP MASK
03656 151400          JSR CHAR       ;PRINT
03657 124000          JMP MESS+4     ;ANOTHER
03660 004464          SKPBZ ,TTO
03661 000771          JMP .-1
03662 063535          JMP 0,MESSR    ;LAST
03663 000777          ZOCT: LDA 0,CH240
03664 002530          MOV 0,0,SKP
03665 020525          POCT: LDA 0,C60
03666 101001          LDA 2,OCTAB    ;PRINT C(1) IN OCTAL
03667 020106          JMP .+3
03670 030433          PDEC: LDA 2,DECTB ;PRINT C(1) IN DECIMAL
03671 000403          LDA 0,CH240    ;SUPPRESS LEADING ZEROS
03672 030441          STA 3,RADRET   ;BOTH ENTRIES PRINT NUMBER
03673 020517          STA 0,ZSUPP    ;THEN TAB TO NEXT POSITION
03674 054447          STA 2,+.1
03675 040445          DECOCT: 0      ;A"LDA 2,TABLE" INSTRUCTION
03676 050401          ISZ .-1
03677 000000          LDA 3,RADRET   ;SETUP "TAB" AT END
03700 010777          LDA 0,C7
03701 034442          MOV 2,2,SNR    ;IF TABLE ENTRY=0
03702 020077          JMP CHAR      ;EXIT WITH TAB
03703 151005          LDA 3,ZSUPP    ;ZEROS SUPPRESS STUF
03704 000440          SUB 0,0
03705 034435          DECO:  SUBL# 2,1,SZC
03706 102400          JMP DECP
03707 146512          SUB 2,1        ;FORM THE DIGIT
03710 000405          LDA 3,C60     ;DO NOT SUPPRESS FOLLOWING ZEROES
03711 146400          INC 0,0
03712 034106          JMP DECO
03713 101400          DECP:  MOVZR# 2,2,SNR ;IF LAST DIGIT THEN
03714 000773          LDA 3,C60     ;C(3)=ZERO, NOT SUPPRESS CHAR
03715 151235          STA 3,ZSUPP   ;C(0)=DIGIT
03716 034106          ADD 3,0       ;MAKE ASCII
03717 054423          JSR CHAR      ;PRINT
03720 163000          JMP DECOCT    ;GET NEXT DIGIT
03721 004423
03722 000755

```

111

```

03723 030425 UCTAB: LDA 2,.,+1+,-DECOCT
03724 100000 100000
03725 010000 10000
03726 001000 1000
03727 000100 100
03730 000010 10
03731 000001 1
03732 000000 0
03733 030435 DECTB: LDA 2,.,+1+,-DECOCT
03734 023420 .RDX 10 10000
03735 001750 1000
03736 000144 100
03737 000012 10
03740 000001 1
03741 000000 0
03742 000000 .RDX 8
03743 000000 ZSUPP: 0
03744 054442 RADRET: 0
03745 101325 CHAR: STA 3,CHRET ;PRINT C(0) RIGHT
03746 001401 MOVZS 0,0,SNR ;RETURN +2 IF NULL
03747 040440 JMP 1,3
03750 176000 STA 0,CHSAV
03751 117000 ADC 3,3 ;COMPUTE THE PARITY
03752 163404 ADD 0,3
03753 000775 AND 3,0,SZR
03754 176660 JMP .-3
03755 020432 SUBCR 3,3 ;COMBINE PARITY WITH CHAR
03756 163300 LDA 0,CHSAV
03757 034426 ADDS 3,0
03760 116405 CHAR1: LDA 3,CHTAB ;IS THIS A TAB
03761 000403 SUB 0,3,SNR
03762 004433 JMP .+3 ;YES
03763 002423 JSR TYPE ;NO PRINT IT
03764 020424 JMP 0,CHRET ;EXIT
03765 034424 LDA 0,CHORZ ;SIMULATE A TAB
03766 117405 LDA 3,CHAR7 ;VIA 1 TO 8 SPACES
03767 002417 AND 0,3,SNR
03770 020422 JMP 0,CHRET
03771 004424 LDA 0,CH240
03772 000772 JSR TYPE
JMP .-6

```

111

```

03773 054420 CRLF: STA 3,CRLF      ;SAVE RETURN
03774 020410 LDA 0,C215
03775 004747 JSR CHAR          ;PRINT CARRIAGE AND LF
03776 020405 LDA 0,C12
03777 004745 JSR CHAR
04000 102400 SUB 0,0
04001 040407 STA 0,CHORZ      ;CLEAR HORZ POSITION
04002 002411 JMP #CRLF      ;EXIT
04003 000012 C12: 12
04004 000215 C215: 215
04005 000011 CHTAB: 11
04006 000000 CHRET: 0
04007 000000 CHSAV: 0
04010 000000 CHORZ: 0
04011 000007 CHAR7: 7
04012 000240 CH240: 240
04013 000000 CRLF: 0
04014 000000 MESSR: 0
04015 054412 TYPE: STA 3,TYPRET ;TYPE THE C(0)R IF
04016 010772 ISZ CHORZ
04017 074477 READS 3 ;SWITCH 1(0).
04020 175100 MOVL 3,3
04021 175102 MOVL 3,3,SZC
04022 002405 JMP #TYPRET ;INHIBIT TYPE EXIT.
04023 063535 SKPBZ ,TTO
04024 000777 JMP , -1
04025 061135 DOAS 0, ,TTO
04026 002401 JMP #TYPRET
04027 000000 TYPRET: 0
04030 000000 LASTPR: 0

```

.END

A0	000411
A1	000415
A10	000604
A11	000615
A12	000626
A13	000641
A14	000650
A14.1	000652
A15	000665
A16	000706
A17	000717
A2	000421
A20	000731
A20.1	000733
A21	000752
A22	000776
A22.1	001002
A22.2	001004
A23	001032
A24	001050
A25	001065
A26	001077
A28	001111
A3	000425
A31	001126
A32	001147
A33	001171
A33.1	001175
A33.2	001200
A35	001246
A36	001301
A39	001334
A39.1	001366
A39.2	001373
A4	000431
A40	001401
A41	001441
A42	001477
A43	001547
A44	001613
A45	001657
A5	000440
A50	001725
A50.1	001736
A50.2	001762
A51	002005
A51.1	002015
A51.2	002035
A52	002060
A52.1	002070
A52.2	002110
A53	002133
A53.1	002143
A53.2	002163
A54	002234
A54.1	002244
A6	000447
A6.1	000451
A6.2	000454
A6.3	000461

A7	000472
A8	000505
A8A	000515
A8B	000526
A8C	000537
A8D	000550
A8E	000561
A9	000576
ABTBF	000266
ASTAT	006241
ATT	006253
BSPAC	006262
BSTAT	006242
BTBF	002575
C0600	000140
C1	000071
C10	000100
C100	000111
C1000	000122
C1010	000142
C1042	000132
C10K	000125
C10X	000216
C110	000112
C11X	000217
C12	004003
C1252	000144
C12X	000220
C13X	000221
C144	000113
C14X	000222
C150	000114
C15X	000223
C1600	000137
C16X	000224
C17	000101
C170	000115
C1777	000141
C17X	000225
C1X	000207
C2	000072
C20	000102
C200	000116
C2000	000123
C201	000136
C20K	000126
C2104	000133
C215	004004
C2525	000143
C2X	000210
C3	000073
C30	000103
C375	000117
C377	000120
C3X	000211
C4	000074
C40	000104
C400	000121
C40K	000127
C4104	000131

C4210	000134
C421T	000135
C4K	000124
C4X	000212
C5	000075
C50	000105
C5252	000145
C5X	000213
C6	000076
C60	000106
C6X	000214
C7	000077
C70	000107
C77	000110
C7700	000130
C7X	000215
CEND	000051
CH240	004012
CH40	003434
CHAR	003744
CHAR1	003757
CHAR7	004011
CHECK	006276
CHORZ	004010
CHREF	004006
CHSAV	004007
CHTAB	004005
CKED	002300
CLEAR	006275
CMTA	000052
CNTR	002544
CONCX	000204
COUNT	003061
CPUND	000227
CRLF	003773
CRLFR	004013
CSTAR	003044
CSTAT	003024
CTR	000226
CX	000206
CYCLE	003437
CYCTS	003477
D500	000154
DECOC	003677
DECOI	003707
DECP	003715
DECTB	003733
DEV1	002426
DEVIC	000233
DIAG	000410
DIS	000035
DIVID	003576
DLOOP	003601
DLT	003130
DLT.1	003135
DSTAT	006243
EHALT	006272
ENDPR	003400
ENDT	003225
ENDT2	003237

ENTCO	003412
ENTER	003406
ENTP1	003401
ER	000272
ERASE	006256
ERET	003521
ERR	003515
ERR1	003524
ERR2	003552
ERR3	003565
ERRCT	003427
ERWL	003332
ESWIT	003426
ETST	002206
ETT	003202
FIRST	000050
FUN	000032
GEN	006260
GMOR	002410
HAAT	003643
HAATT	006255
HEAD	003574
IASO	000302
IATA	000241
IBSP	000262
IBSTA	000242
IBUFF	000054
ICAR	000250
ICAK	000276
ICKSI	000235
ICLR	000275
ICRLF	000245
IDIAG	000404
IDLT	000407
IDSTA	000243
IDX0	000020
IDX1	000021
IDX2	000022
IDX3	000023
IDX4	000024
IERAS	000256
IETT	000406
IGEN	000260
IHAAT	000255
ILD	000277
IMESS	000246
INITI	002352
INST	000231
INSTA	000244
INTIL	000236
IOPIP	000254
IOUTS	000253
IPDEC	000252
IPOCI	000247
IRD	000273
IRONS	000274
IRET	002462
IRWD	000263
IRWS	000264
ISEL	000303

ISPC	000301
ISSTA	000240
ISTAT	000237
ITIM	000304
ITR	003424
ITRAG	003423
ITRCI	003425
ITRRE	003421
ITRST	003422
ITYPE	000251
IWEOF	000261
IWLT	000405
IWNS	000265
IWRT	000257
IWT	000300
K1200	000162
K800	000165
KNON1	000161
KNON2	000166
KNOVA	000160
KSU	000163
KSUSC	000164
LASTP	004030
LEAV	002527
LOAD	006277
LOOP	000271
LOOPR	003436
LPOII	002277
LST	000175
LST1	000203
LST79	002314
LST9	002331
LSTRT	002340
M100	000152
M1000	000153
M2	000146
M20	003623
M3	000147
M4	000150
M5	000151
MDL	003165
MDL1	003173
MDL2	003175
MENDI	003261
MESS	003646
MESSR	004014
MESTA	003264
MLOOP	003613
MSAV	003622
MSTAR	003253
MTC	000030
MULT	003610
MWLB	003353
MWLO	003343
MWLS	003364
N1250	000157
N400	000156
N8	000155
NEXCX	002374
NODL	003162

NOEX	003511
NOTRE	002474
NOTYM	002475
NOTYP	002463
NSTAT	006244
NXTYP	002446
ORUFF	000053
OCTAB	003723
OKREI	003035
OLDDE	000234
ON	003040
OPIP	003632
OPIPR	003642
OUT36	003624
OUTRE	003631
PAT1	002605
PAT2	002610
PAT3	002613
PAT4	002616
PCENT	003435
PDEC	003672
PIP	006254
POCT	003667
POINT	000167
PONCX	000055
PONTR	000176
RADRE	003743
RBCT	002276
RCHK	002545
RDLAY	003062
RDSW	002751
READ	006273
RECTR	000232
REDNS	006274
REP	003276
RETUR	003430
REWIN	006263
RGEN	002677
RRD	002750
RRWS	002562
RSEL	002706
RTIM	003126
RUNCU	002220
RWDST	006264
RWEOF	002574
RWRT	003013
RXAS	002633
RXNS	002645
SAC0	003376
SAC1	003377
SAV0	003433
SAV1	003432
SAV2	003431
SAVE0	000056
SAVE1	000057
SELEC	006303
SETP1	000267
SETUP	000270
SOK	003303
SPACE	006301

SSTAT	006240
STATR	003100
STATU	006237
SWPAR	002341
SWTCH	003014
TIMCT	003127
TIMER	006304
TSTAT	003114
TYPE	004015
TYPRE	004027
WAIT	006300
WDCNT	000230
WEOF	006261
WLT	003270
WRITE	006257
WRTNS	006265
X170	000205
XASTA	002623
XBSP	002563
XBSTA	002621
XCHK	002505
XCHK,	002516
XCLR	002532
XDSTA	002634
XERAR	003023
XERAS	003015
XGEN	002663
XLD	002646
XNSTA	002636
XRD	002712
XRDNS	002707
XRD.1	002731
XRWD	002551
XRWS	002554
XSEL	002700
XSPC	002546
XSSTA	003063
XSTAT	003065
XTIM	003101
XTIM1	003124
XTIM,	003113
XWEOF	002566
XWNS	002752
XWRT	002755
XWRT,	002774
XWT	003045
XWTLO	003053
YDL	003153
ZERRC	003472
ZOCT	003665
ZSUPP	003742
.100M	000066
.10SE	000063
.1SEC	000060
.200M	000067
.20SE	000064
.2SEC	000061
.400M	000070
.50MS	000065
.5SEC	000062

.ITD 000035

SNIS	001262
TIME	000047
TIMED	000526
TIMFY	000051
TYPE	001224
XDIV	000512
XICPL	000056
XIMES	000060
XIPDF	000057
XX14	000651
ZOOT	001116
ZSHIP	001173