

IDENTIFICATION

PRODUCT CODE	MAINDEC-08-DHVTA-B-D
PRODUCT NAME	VT8-E VIDEO DISPLAY TEST 1 ✓
DATE CREATED	FEBRUARY 26, 1973
MAINTAINER	DIAGNOSTIC PROGRAMMING
AUTHOR	BRUCE HANSEN

COPYRIGHT © 1972, 1973  
DIGITAL EQUIPMENT CORPORATION



1.

ABSTRACT  
-----

VT8-E VIDEO DISPLAY TEST 1 IS A TEST OF THE VT8-E DISPLAY, CONTROL, AND KEYBOARD. THIS PROGRAM TESTS THE BASIC FUNCTIONS OF THE CONTROL AS WELL AS CHECKING THE ABILITY TO DISPLAY IN THE ALPHA-NUMERIC MODE. WHILE WAITING FOR INTERRUPTS DURING THE VISUAL TESTS A SIMPLE PROCESSOR TEST IS BEING EXECUTED IN THE BACKGROUND.

W A R N I N G - THIS PROGRAM SHOULD ONLY BE HALTED WHEN IN MAINTENANCE AND VISUAL TESTS BY SETTING SR0 TO THE "1" POSITION.

2.

REQUIREMENTS  
-----

2.1

EQUIPMENT  
-----

PDP8-E, -F, OR -M PROCESSOR  
TELETYPE OR HIGH SPEED READER TO READ IN THE PROGRAM  
VT8-E

2.2

STORAGE  
-----

LOCATIONS 0000 THROUGH 7577.

2.3

PRELIMINARY PROGRAMS  
-----

ALL PROCESSOR AND MEMORY TEST PROGRAMS MUST HAVE BEEN RUN SUCCESSFULLY.

3.

LOADING PROCEDURE  
-----

THE BINARY LOADER IS USED TO LOAD THE PROGRAM INTO ANY MEMORY FIELD. REFER TO BINARY LOADER DOCUMENTATION IF UNFAMILIAR WITH ITS USE.

# 4: TEST PROCEDURE

## 4.1 STANDARD TEST PROCEDURE

USE OF THE STANDARD TEST PROCEDURE RESULTS IN ANY ERROR OCCURENCE THAT CAN BE DETECTED BY THE SOFTWARE TO BE INDICATED BY A HALT. CONSULT THE LISTING WHERE THE HALT OCCURRED TO DETERMINE THE NATURE OF THE ERROR. IF AN ERROR OCCURS, REFER TO PARAGRAPHS 5.1 AND 6.0 FOR SWITCH SETTINGS, ERROR EXPLANATIONS, AND ERROR RECOVERIES. THERE ARE SPECIAL STARTING ADDRESSES WHICH ALLOW THE USER TO CHANGE DEVICE CODES, DISPLAY SELECTED CHARACTERS, DO DATA BREAKS USING THE DATA IN THE SR, OR START AT A PARTICULAR SECTION OF THE PROGRAM. REFER TO PARAGRAPH 5.2 FOR THESE SPECIAL STARTING ADDRESSES AND A DESCRIPTION OF EACH.

## 4.2 ALTERNATE VT8-E'S

IN ORDER TO RUN UP TO 8 VT8-E'S SEQUENTIALLY USING THE DIAGNOSTIC, THE IOT CODES FOR THE DEVICES MUST BE DEPOSITED INTO AN IOT TABLE IN MEMORY. THE TABLE IS NORMALLY SETUP TO RUN ONLY ONE VT8-E AND APPEARS IN MEMORY AS FOLLOWS:

LOCATION	CONTENTS
0020	0503
0021	0000
0022	0000
0023	0000
0024	0000
0025	0000
0026	0000
0027	0000
0030	0000

THE CONTENTS OF LOCATION 0020 CONTAINS THE DEVICE CODES OF ONE VT8-E. BITS 0-5 ARE THE DEVICE CODE OF THE DISPLAY (05) AND BITS 6-11 ARE THE DEVICE OF THE KEYBOARD (03). TO TERMINATE THE TABLE, ZEROES MUST BE DEPOSITED INTO THE LOCATION FOLLOWING THE LAST DEVICE CODE DEPOSITED. BELOW IS AN EXAMPLE OF A TABLE SETUP TO RUN SIX CONSECUTIVE VT8-E'S.

LOCATION	CONTENTS
0020	0503
0021	1513
0022	2523
0023	3533
0024	4543
0025	5553
0026	0000
0027	0000
0030	0000

TERMINATOR  
UNUSED  
UNUSED

WARNING! IF THE PROGRAM IS HALTED WHILE RUNNING ALTERNATE VT8-E'S AND THE PROGRAM IS RESTARTED AT ANY OTHER ADDRESS THEN 200,71,72 OR 75 THE PROGRAM WILL RUN THE CURRENT VT8-E SELECTED;

## MAINTENANCE AND ALPHA-NUMERIC TESTS

1. CHECK THE VT8-E CLOCK WITH A SCOPE AND VERIFY THAT FOR 60 HZ A TICK OCCURS EVERY 16 2/3 MS, OR FOR 50 HZ A TICK OCCURS EVERY 20 MS.
2. TURN THE VIDEO TERMINAL ON.
3. LOAD THE PROGRAM INTO ANY DESIRED MEMORY FIELD USING THE BINARY LOADER.
4. SET THE SENSE SWITCH TO A "1".
5. IF IT IS DESIRED TO RUN MORE THAN ONE VT8-E, REFER TO PARAGRAPH 4.2 FOR IOT TABLE SETUP AND THEN GO TO STEP 6 IN THIS PARAGRAPH.
6. LOAD ADDRESS 0200 WITH THE IF AND DF SET TO THE PROGRAM FIELD.
7. CLEAR ALL SWITCHES.
8. IF 50 HZ, SET SR4 = 1.
9. IF 32 CHARACTERS PER LINE, SET SR6 = 1.
10. PLACE THE VALUE OF THE HIGHEST MEMORY FIELD IN SR9-11.
11. DEPRESS CLEAR AND CONTINUE.
12. THE DISPLAY WILL HAVE A BLANK SCREEN FOR THE DURATION OF THE MAINTENANCE TESTS WHICH IS APPROXIMATELY 17 SECONDS.
13. IF RUNNING MORE THAN ONE VT8-E, THE NEXT VT8-E IN THE IOT TABLE WILL BE SELECTED TO RUN THE MAINTENANCE TEST UNTIL ALL VT8-E'S IN THE IOT TABLE ARE TESTED.
14. REFER TO PARAGRAPH 8.2 FOR A DESCRIPTION OF THE PATTERNS DISPLAYED.
15. AT THE CONCLUSION OF THE VISUAL TESTS, "DONE" WILL BE DISPLAYED ON THE SCREEN AND THE TEST WILL THEN LOOP BACK TO THE BEGINNING. IF MORE THAN ONE VT8-E IS BEING TESTED, "DONE" WILL NOT BE DISPLAYED UNTIL THE LAST VT8-E IS TESTED.
16. FOR SYSTEMS WITH MULTIPLE VT8-E'S, REFER TO PARAGRAPH 5.2.6 TO RUN UP TO 4 VT8-E'S SIMULTANEOUSLY.

4.4

KEYBOARD TEST PROCEDURE  
-----

1. THE MAINTENANCE AND VISUAL TESTS SHOULD HAVE BEEN RUN PRIOR TO THIS TEST.
2. MAKE SURE THE VIDEO TERMINAL IS ON.
3. IF IT IS DESIRED TO RUN MORE THEN ONE VT8-E, REFER TO PARAGRAPH 4.2 OR IF THE IOT TABLE HAS ALREADY BEEN SETUP GO TO STEP 4 IN THIS PARAGRAPH.
4. LOAD ADDRESS 0072 WITH THE "IF" AND "OF" SET TO THE PROGRAM FIELD.
5. CLEAR ALL SWITCHES.
6. IF 32 CHARACTERS PER LINE SET SR6 = 1.
7. DEPRESS CLEAR AND CONTINUE.

4.5

KEYBOARD TEST 1 (STANDARD KEYS ONLY)  
-----

1. A "CURSOR 1" WILL BE DISPLAYED IN THE TOP LEFT OF THE SCREEN.
2. CARRIAGE RETURN ENDS THIS TEST AND ENTERS KEYBOARD TEST 2.
3. TO VERIFY THAT ALL KEYS, (EXCEPT SPECIAL FUNCTION KEYS), ARE FUNCTIONING EACH ONE SHOULD BE STRUCK SEVERAL TIMES VERIFYING THAT THE CORRECT CHARACTER APPEARS ON THE SCREEN.
4. IF TOO MANY LINE FEEDS ARE TYPED THE SCREEN WILL BE BLANK.
5. RUBOUT WILL BACK THE CURSOR UP ONE POSITION.

4.6

KEYBOARD TEST 2 (SPECIAL FUNCTION KEYS)  
-----

1. THIS TEST IS ENTERED BY TYPING CARRIAGE RETURN WHILE IN KEYBOARD TEST 1, OR BY STARTING AT ADDRESS 3000.
2. A "CURSOR 2" WILL BE DISPLAYED IN THE TOP LEFT OF THE SCREEN.
3. CARRIAGE RETURN ENDS THIS TEST AND ENTERS KEYBOARD TEST 3.
4. TO VERIFY THAT ALL KEYS FUNCTION CORRECTLY EACH ONE SHOULD BE STRUCK SEVERAL TIMES. THIS ONLY APPLIES TO THE SPECIAL FUNCTION KEYS.

4.7

# KEYBOARD TEST 3 (ALL KEYS) -----

1. THIS TEST IS ENTERED BY TYPING CARRIAGE RETURN IN TEST 2 OR BY STARTING AT ADDRESS 3200.
2. A "CURSOR 3" WILL BE DISPLAYED IN THE TOP LEFT OF THE SCREEN.
3. CARRIAGE RETURN ENDS THIS TEST AND STARTS THE PROGRAM AT ADDRESS 0200 UNLESS SR OPTIONS PREVENT IT. IF MORE THAN ONE VT8-E IS BEING TESTED, THE PROGRAM WILL GO TO KEYBOARD TEST 1 ON THE NEXT DEVICE (PARAGRAPH 4.12) AND THEN CONTINUE THROUGH UNTIL ALL DEVICES ARE TESTED.
4. ALL KEYS SHOULD BE STRUCK SEVERAL TIMES WITH THE SENSE SWITCH IN BOTH POSITIONS. CHECK THAT THE POSITION OF THE SWITCH IS THE SAME AS THE ONE DISPLAYED ON THE SCREEN.

5.1

## OPERATING PROCEDURE -----

5.1.1

### SWITCH REGISTER CONTROL -----

SR0 = 0	NORMAL RUN,
SR0 = 1	HALT PROGRAM.
SR1 = 0	HALT ON ERROR.
SR1 = 1	NO HALT ON ERROR.
SR2 = 0	NO LOOP.
SR2 = 1	REPEAT PATTERN, (SCOPE LOOP)
SR3 = 0	NORMAL RUN.
SR3 = 1	REMAIN IN CURRENT TEST.
SR4 = 0	60 HERTZ
SR4 = 1	50 HERTZ
SR5 = 0	ENABLE BACKGROUND JOB.
SR5 = 1	DISABLE BACKGROUND JOB.
SR6 = 0	64 CHARACTERS PER LINE.
SR6 = 1	32 CHARACTERS PER LINE.
SR7 = 0	NORMAL.
SR7 = 1	ENABLE MANUAL INCREMENT FOR VISUAL TESTS
SR8	WITH SR7 = 1 A CHANGE IN SR8 CHANGES THE PATTERN ON THE SCREEN
	WITH SR7 = 0 AND SR8 = 1 REMAIN IN CURRENT SECTION, (MAINTENANCE, DISPLAY, OR KEYBOARD)
SR9-11	PLACE VALUE OF HIGHEST MEMORY FIELD IN THE SR 9-11, (EG. 0 FOR 4K, 1 FOR 8K, AND UP TO 7 FOR 32K).

# 5.2 STARTING ADDRESSES

## 5.2.1 0070 - DEVICE CODE MODIFICATION

A: SET SR TO 0070, PRESS "LOAD ADDRESS",  
 B: SET SR0-5 TO THE DEVICE CODE OF THE DISPLAY,  
 C: SET SR6-11 TO THE DEVICE CODE OF THE KEYBOARD,  
 D: PRESS "CLEAR" THEN "CONTINUE",  
 E: THE PROGRAM WILL MAKE THE DEVICE CODE CHANGES AND  
 MODIFY THE IOT TABLE TO RUN ONLY ONE VT8-E AND THEN  
 THE PROGRAM WILL HALT.

## 5.2.2 0071 - ENTER AT THE VISUAL SECTION.

## 5.2.3 0072 - ENTER AT THE KEYBOARD SECTION.

## 5.2.4 0073 - STARTING AT THIS ADDRESS ENTERS A TEST WHICH DOES DATA BREAKS USING THE DATA IN THE SR TO PUT IN THE BUFFER AND DISPLAYS THE OUTPUT IN THE MQ.

## 5.2.5 0074 - DISPLAY SELECTED CHARACTER.

A: SET SR TO 0074, PRESS "LOAD ADDRESS",  
 B: SET THE CHARACTER CODE IN SR5-11,  
 C: SET CONTROL BITS FOR CB1,CB2,CB3 AND CB4 IN SR1-4 RESPECTIVELY AS DESCRIBED BELOW

SR1	SR2	SR3	SR4
0	NOP	0	0
0	EBF	0	1
1	BBF	1	0
1	EOS	1	1

NORMAL  
 BLINK  
 BOLD  
 CURSOR

D: PRESS "CLEAR AND CONTINUE",  
 E: PROGRAMS HALTS. CLEAR ALL SWITCHES,  
 F: SET SR2 IF A LOOP ON THIS CHARACTER IS DESIRED  
 OTHERWISE CONSULT THE SR OPTIONS.  
 G: PRESS CONTINUE.

## 5.2.6 0075 - MULTIPLE VT8-E'S. THIS TEST IS SETUP TO RUN 4 VT8-E'S SIMULTANEOUSLY TO INSURE THAT THE VT8-E'S DO NOT COMPLETE FOR DATA BREAKS. A STABLE SWIRL PATTERN SHOULD BE DISPLAYED ON ALL 20 LINES OF ALL VT8-E'S. THE FIRST VT8-E SHOULD HAVE A SWIRL PATTERN STARTING OFF WITH A NUMBER 1 AND THE NEXT WITH A NUMBER 2, ETC. IF 64 CHAR/LINE THERE WILL BE N SPACES AT THE END OF THE LAST LINE, WHERE N IS THE VT8-E NUMBER.

IN ORDER TO RUN 4 VT8-E'S SIMULTANEOUSLY, A TABLE HAS  
 BEEN SETUP IN MEMORY AS SHOWN BELOW

LOCATION	CONTENTS
1517	0503
1520	1513
1521	2523
1522	3533
1523	0000 TERMINATOR



LOCATION 1517 CONTAINS THE DEVICE CODES OF A VT8-E WITH A DISPLAY DEVICE CODE OF 05 AND A KEYBOARD DEVICE CODE OF 03, THE NEXT THREE LOCATIONS CONTAINS DEVICE CODES FOR THREE OTHER VT8-E'S, AND THE LAST LOCATION CONTAINS ZEROES WHICH IS A TERMINATOR TO THE PROGRAM, THESE LOCATIONS MAY BE CHANGED TO ANY OTHER DEVICE CODES BUT A ZERO LOCATION MUST FOLLOW THE LAST DEVICE CODE INSERTED, A ZERO LOCATION TERMINATES THE TABLE.

5:2.7 0200 - STANDARD STARTING ADDRESS

6: ERROR INFORMATION  
-----

ALL ERRORS THAT ARE SOFTWARE DETECTABLE WILL RESULT IN A PROGRAM HALT UNLESS PREVENTED BY THE SR, REFER TO THE PROGRAM LISTING FOR A DESCRIPTION OF THE ERRORS AND UNLESS OTHERWISE STATED IN THE LISTING THE SR MAY BE USED FOR SETTING UP A SCOPE LOOP.

IF THERE IS A FAILURE WHILE RUNNING ALTERNATE VT8-E/S, EXAMINE LOCATION 0031 TO OBTAIN THE POINTER ADDRESS TO THE IOT TABLE FOR THE DEVICE CODES OF THE FAILING VT8-E, IF LOCATION 0031 CONTAINED A 0024 AND LOCATION 0024 CONTAINED A 4543 AS IN THE LAST TABLE IN PARAGRAPH 4.2, THIS WOULD MEAN THAT A VT8-E FAILED WITH A DISPLAY DEVICE CODE OF 45 AND A KEYBOARD DEVICE CODE OF 43.

7: EXECUTION TIME  
-----

THE EXECUTION TIME OF THE MAINTENANCE AND VISUAL TESTS IS APPROXIMATELY 17.5 MINUTES.

8: PROGRAM DESCRIPTION  
-----

8:1 MAINTENANCE TESTS  
-----

1: CHECK THAT KCG WILL CLEAR THE AC. (CKCG)

THE AC IS SET TO 7777 AND THE KCG IS ISSUED AND THE AC IS CHECKED TO BE 0.

2: CHECK THAT KRB WILL CLEAR THE AC. (CKRB)

THE AC IS SET TO 7777 AND THEN KRB IS ISSUED AND ACQ-3 IS CHECKED TO BE CLEAR.

3: CHECK THAT KRS WILL "INCLUSIVE OR", (CKRS)

THE AC IS SET TO 7777 AND THEN KRS IS ISSUED AND THE AC IS CHECKED TO BE 7777.

4. CHECK FOR SENSE SWITCH IN THE "1" POSITION. (SENSE)  
WITH A CLEAR AC, DPMS IS ISSUED TO READ THE STATE OF THE SENSE SWITCH INTO AC0 AND A CHECK IS MADE TO VERIFY IT WAS READ BACK.
5. CHECK THAT DP1A WILL CLEAR THE AC. (CDPLA)  
THE AC IS SET TO 7777 AND THEN DP1A IS ISSUED, THEN THE AC IS THEN CHECKED TO BE 0.
6. CHECK THAT DP5M WILL CLEAR THE AC. (CDPSM)  
THE AC IS SET TO 7777 AND THEN DP5M IS ISSUED, THEN THE AC IS CHECKED TO BE 0.
7. TEST THAT DPCL DOES NOT AFFECT THE AC. (CDPCL)  
DPCL IS ISSUED WITH THE AC = 7777 AND THEN 0000, CHECKING EACH TIME THAT THE AC REMAINS UNCHANGED.
8. CHECK THAT DPG0 WILL CLEAR THE AC. (CDPG0)  
THE AC IS SET TO 7777 AND THEN DPG0 IS ISSUED, THE AC IS THEN CHECKED TO BE 0.
9. CHECK OF THE REAL TIME CLOCK. (CLOCK)  
WAIT FOR THE CLOCK FLAG TO SET AND THEN ONCE IT'S SET CHECK THAT DPCL CLEARED THE FLAG AND THAT DPCL WILL NOT SKIP WITH THE FLAG 0. NOW CHECK THAT A CLOCK FLAG WILL OCCUR WITHIN APPROXIMATELY 40 MSEC.
10. CHECK THAT DPG0 CAN ENABLE THE CLOCK INTERRUPT ENABLE. (INT1)  
DPG0 IS ISSUED WITH THE AC=0001 WHICH SHOULD ENABLE THE CLOCK INTERRUPT ENABLE. THE INTERRUPT IS TURNED ON AND THE PROGRAM ENTERS A 30 MS. TIME-OUT LOOP DURING WHICH TIME AN INTERRUPT FROM THE CLOCK SHOULD TAKE PLACE.
11. CHECK THAT DPG0 CAN DISABLE THE CLOCK INTERRUPT ENABLE. (INT2)  
DPG0 IS ISSUED WITH THE AC=0001 WHICH ENBALES THE CLOCK INTERRUPT. NOW DPG0 IS ISSUED AGAIN ONLY THIS TIME THE AC IS ZERO SO THE CLOCK INTERRUPT SHOULD BE DISABLED. THE INTERRUPT IS TURNED ON AND THE PROGRAM ENTERS A 30 MSEC TIME-OUT LOOP WHICH SHOULD TIME-OUT SINCE THE CLOCK IS DISABLED.
12. CHECK THAT INITIALIZE CAN DISABLE THE CLOCK INTERRUPT ENABLE. (INT3)  
DPG0 IS ISSUED WITH THE AC=0001 WHICH ENBALES THE CLOCK INTERRUPT ENABLE. CAF IS NOW ISSUED WHICH GENERATES INITIALIZE AND SHOULD CLEAR THE CLOCK INTERRUPT ENABLE. THE INTERRUPT IS NOW TURNED ON AND THE PROGRAM ENTERS A 30 MSEC TIME-OUT LOOP WHICH SHOULD TIME-OUT SINCE THE CLOCK IS DISABLED.

13: CHECK THAT THE EXTENDED ADDRESS REGISTER AND COUNTER CAN BE LOADED AND READ BACK, (EXTA)

DPSM IS ISSUED WITH THE AC CONTAINING THE DATA PATTERN TO BE USED TO TEST THE EXTENDED STARTING ADDRESS REGISTERS AND AC11 SET WHICH PLACES AC6-8 INTO THE EXTENDED STARTING ADDRESS REGISTER. DPSM IS ISSUED AGAIN ONLY THIS TIME WITH AC80 WHICH LOADS THE CONTENTS OF THE EXTENDED STARTING ADDRESS REGISTER INTO THE ADDRESS COUNTER. DPMS IS THEN ISSUED TO READ THE CONTENTS OF THE EXTENDED ADDRESS COUNTER INTO THE AC. AN INCREMENTING PATTERN IS USED STARTING WITH 0.

14: CHECK THAT THE EXTENDED ADDRESS COUNTER WILL ONLY INCREMENT WHEN THE ADDRESS COUNTER GOES FROM 7777 TO 0000, (EXTIN)

ISSUE DPLA WITH THE AC80 WHICH LOADS THE STARTING ADDRESS REGISTER WITH 0. NOW ISSUE DPSM WITH THE AC0001 WHICH SETS THE EXTENDED STARTING ADDRESS REGISTER TO 0. DPSM IS NOW ISSUED WHICH LOADS THE STARTING ADDRESS REGISTERS INTO THE ADDRESS COUNTER. ISSUE DPMB WHICH DOES A BREAK AND CAUSES THE ADDRESS COUNTER TO INCREMENT. THE EXTENDED ADDRESS COUNTER IS READ BACK USING DPMS. THE ONLY TIME DPMB SHOULD READ ANYTHING BACK INTO AC6-8 IS WHEN THE ADDRESS COUNTER OVERFLOWED.

15: CHECK THAT THE EXTENDED ADDRESS COUNTER WILL INCREMENT CORRECTLY, (EXT)

USING DPLA SET THE STARTING ADDRESS REGISTER TO 7777. NOW LOAD THE EXTENDED STARTING ADDRESS REGISTER (STARTING WITH 0) USING DPSM WITH AC11=1. NOW LOAD THE ADDRESS COUNTERS BY ISSUING DPSM WITH THE AC80. DO A BREAK BY ISSUING DPMB WHICH WILL OVERFLOW THE ADDRESS COUNTER INTO THE EXTENDED ADDRESS COUNTER. NOW READ BACK THE EXTENDED ADDRESS COUNTER AND CHECK THAT IT WAS PROPERLY INCREMENTED. UPDATE THE TEST PATTERN BY 1 AND REPEAT THE TEST AGAIN UNTIL THE EXTENDED ADDRESS COUNTER INCREMENTS FROM 0 THROUGH 7 TO 0.

16: CHECK THAT DP60 CAN LOAD THE EXTENDED STARTING ADDRESS REGISTER, (EDPGQ)

THE AC IS LOADED WITH 0001 AND DPSM IS ISSUED WHICH CLEARS THE EXTENDED STARTING ADDRESS REGISTER AND DPSM IS ISSUED AGAIN WITH THE AC ZERO WHICH NOW LOADS 0 INTO THE EXTENDED STARTING ADDRESS COUNTER. DP60 IS ISSUED WITH THE AC80 WHICH SHOULD SET THE EXTENDED STARTING ADDRESS REGISTER TO 7. IN ORDER TO VERIFY THAT IT HAPPENED, DPMS IS ISSUED WHICH LOADS THE EXTENDED STARTING ADDRESS REGISTER INTO THE EXTENDED ADDRESS COUNTER WHICH IS READ INTO THE AC WITH DPMS. THE AC SHOULD BE 0070.

17. CHECK THE VT8-E BUFFER USING CONSTANT DATA, (DATA1)

OBTAIN THE DATA PATTERN AND DO 31 OR 63 BREAKS TO PRIME THE BUFFER. NOW DO 1 MORE BREAK AND VERIFY THAT THE DATA PATTERN RECEIVED IS THE SAME ONE THAT WAS SENT. EACH PATTERN IS DONE 4096 TIMES.  
DATA PATTERNS USED: 7777 0000 7777 5252 2525 7700 0077 7007 0770.

18. CHECK THE VT8-E BUFFER USING A SPECIAL BINARY COUNT PATTERN, (DATA2)

A SPECIAL BINARY COUNT PATTERN IS THE BINARY COUNT AND THEN THE 1'S COMPLEMENT. EGI (0000 7777 0001 7776 0002 7775 ETC.)

A: GET THE PATTERN,  
B: DO A SINGLE BREAK,  
C: 32ND OR 64TH BREAK?  
D: NO - GO TO G; YES - GO TO E;  
E: SIMULATE THE CORRECT OUTPUT AND CHECK THE ACTUAL AGAINST THE SIMULATED;  
F: IF ERROR - HALT (SEE THE LISTING).  
G: UPDATA DATA PATTERN (EGI IF 0000 THEN 7777; IF 7777 THEN 0001; IF 0001 THEN 7776) ETC;  
H: ALL PATTERNS BEEN OUTPUT?  
I: NO - GO TO A; YES - GO TO NEXT TEST.

19. ADDRESS TEST (ADDR1)

THIS TEST STARTS AT ADDRESS 0 OF FIELD 0 AND CHECKS EVERY ADDRESS IN EVERY FIELD FOR AS MANY FIELDS AS SELECTED IN SR9-11. THE ADDRESS AND ITS COMPLEMENT ARE THE DATA PATTERNS USED.

A: GET THE ADDRESS AND SAVE THE CONTENTS IN THE MG.  
B: GET THE DATA PATTERN AND PLACE IT IN THE ADDRESS; (ADDRESS OR ITS COMPLEMENT)  
C: DO 32 OR 64 BREAKS ON THE SAME ADDRESS;  
D: SAVE THE OUTPUT OF THE BUFFER, CONTENTS OF THE MG.  
E: RESTORE THE ADDRESS WITH THE CONTENTS OF THE ACTUAL;  
F: CHECK THE EXPECTED AGAINST THE ACTUAL;  
G: IF ERROR - HALT; (SEE LISTING).  
H: UPDATA PATTERN AND ADDRESS.  
I: ALL ADDRESSES CHECKED?  
J: NO - GO TO A; YES - GO TO NEXT TEST.

# 20. CURRENT ADDRESS INCREMENT TEST. (CAINC)

THIS TEST CHECKS THAT THE ADDRESS COUNTER INCREMENTS CORRECTLY.

A: LOAD ADDRESS 0000 INTO STARTING ADDRESS REGISTER;  
 B: SAVE THE CONTENTS OF THE ADDRESS TO BE TESTED IN THE NO;  
 C: PLACE A VALUE EQUIVALENT TO THE ADDRESS IN THE ADDRESS;  
 D: DO 1 BREAK;  
 E: RESTORE THE ADDRESS AND SAVE THE OUTPUT OF THE BUFFER;  
 F: HAVE 32 OR 64 BREAKS BEEN PERFORMED?  
 G: NO - GO TO J1 YES - NEXT;  
 H: IS THE OUTPUT THE SAME AS WHAT WAS EXPECTED?  
 I: NO - HALT. (SEE LISTING)  
 J: +1 TO THE DATA;  
 K: HAS ADDRESS 7777 BEEN OUTPUT YET?  
 L: NO - GO TO C1 YES - GO TO NEXT TEST.

## 8.2 VISUAL DISPLAY TESTS

### 1. DISPLAY A FULL SCREEN OF A SINGLE CHARACTER. (DSCHAR)

THIS TEST DISPLAYS A FULL SCREEN OF A SINGLE CHARACTER FOR 2 SECONDS STARTING WITH CODE 0040 (SPACE) AND INCREMENTING UP TO CODE 0137 (\*). EACH COMPLETE CHARACTER SET IS DISPLAYED IN NORMAL, BLINK, BRIGHT, AND THEN FINALLY CURSOR MODE.

A: SET UP SO SPACE (0040) IS FIRST CHARACTER DISPLAYED.  
 B: LOAD BUFFER WITH CHARACTER AND CB2+CB3 CONTROL BITS.  
 C: SET TIMER FOR 2 SECONDS.  
 D: DISPLAY CHARACTER FOR DURATION OF TIMER.  
 E: MANUAL INCREMENT?  
 F: NO - NEXT1 YES - GO TO D,  
 G: +1 TO CHARACTER;  
 H: CODE 137 "\*" BEEN DISPLAYED YET?  
 I: NO - GO TO B1 YES - NEXT.  
 J: INCREMENT CB2+CB3 CONTROL WORD  
 K: CURSOR CONTROL BEEN SEEN YET?  
 L: NO - GO TO B1 YES - GO TO NEXT TEST.

## 2. DISPLAY A RIPPLE PATTERN. (DISRIP)

THIS TEST DISPLAYS A RIPPLE OR INCREMENTING PATTERN OF THE CHARACTER SET ON EACH LINE STARTING WITH CODE 0040 (SPACE) AND ENDING WITH CODE 0137 (\*) IF JUMPED FOR 64 CHARACTERS PER LINE, HOWEVER IF SET FOR 32 CHARACTERS PER LINE IT WILL REQUIRE 2 LINES TO DISPLAY THE CHARACTER SET.

```

EG1  !"#%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNQRSTUUVWXYZ/3**
641  !"#%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNQRSTUUVWXYZ/3**
321  !"#%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNQRSTUUVWXYZ/3**

```

THE PATTERN IS DISPLAYED IN NORMAL, BLINK, BRIGHT, AND FINALLY CURSOR MODE.

- A. SET TIMER FOR 5 SECONDS.
- B. LOAD RIPPLE PATTERN.
- C. DISPLAY RIPPLE PATTERN FOR DURATION OF TIMER.
- D. STOP DISPLAY.
- E. INCREMENT THE CB2CB3 CONTROL WORD
- F. CURSOR CONTROL DISPLAYED YET?
- G. NO - GO TO B1 YES - GO TO NEXT TEST.

## 3. DISPLAY A SWIRL PATTERN (DISSWL)

THIS TEST DISPLAYS A SWIRL PATTERN.

```

EG1  !"#%
      !"#%&
      !"#%&'

```

IN THE CASE OF 64 CHARACTERS PER LINE EACH LINE WILL CONTAIN THE ENTIRE CHARACTER SET AND IF 32 CHARACTERS PER LINE IT WILL REQUIRE 2 LINES TO DISPLAY THE SET. THIS TEST IS RUN WITH CB2CB3 IN THE NORMAL MODE, THE SWIRL IS MOVED OR UPDATED EVERY SECOND.

## 4. DISPLAY A RIPPLE PATTERN WITH ALL CONTROL BITS SEEN. (ALL)

THIS TEST IS SIMILAR TO 8.2(2.) EXCEPT EACH COMPLETE CHARACTER SET IS SHOWN WITH A DIFFERENT CB2CB3 CONTROL BIT SET.

```

EG1  ABC (NORMAL)
      ABC (BLINK)
      ABC (BRIGHT)
      ABC (CURSOR)
      ABC (NORMAL) - REPEAT

```

THE PATTERN IS DISPLAYED FOR 10 SECONDS.

# 5. LINE FEED TEST: (LFTST)

THE ENTIRE BUFFER IS FILLED WITH "CURSOR ?". A VERTICAL COLUMN OF "BRIGHT #" IS FORMED WITH "I" ON THE LEFT AND LINE FEEDS ON THE RIGHT WHICH SHOULD PRODUCE A HORIZONTAL MOVEMENT OF THE VERTICAL COLUMN OF "BRIGHT #" WITH NOTHING VISUAL TO THE RIGHT OF IT. A LINE FEED FAILURE WILL MOST LIKELY DISRUPT THE VERTICAL COLUMN AND "CURSOR ?" WILL BE SEEN. IF AT ANY TIME "CURSOR ?" IS SEEN THERE HAS BEEN A FAILURE.

```

EGI (L=LINE FEED)      .....*L???  ==>
                        .....*L???  ==>
                        .....*L???  ==>

```

THE COLUMN IS MOVED TO THE RIGHT EVERY .5 SECONDS.

# 6. TEST OF "BEGIN BLANK FIELD" AND "END BLANK FIELD". (FLDST)

THE ENTIRE BUFFER IS FILLED WITH "K/S". SET BBF IN THE FIRST CHARACTER WHICH WILL BE A "CURSOR ?" AND MOVE EBF WHICH WILL BE A "BRIGHT E". THE "BRIGHT ?/S" WHICH ARE LOADED BETWEEN BBF AND EBF SHOULD NEVER BE SEEN. IF "CURSOR ?" AND/OR "BRIGHT ?" ARE SEEN THIS CONSTITUTES AN ERROR. EBF IS MOVED RIGHT AND DOWN, AND WHEN IT REACHES THE LOWER RIGHT THE SCREEN SHOULD BE EMPTY. NOW MOVE BBF RIGHT AND DOWN EACH TIME PLACING A "BRIGHT 8" ON THE LEFT OF IT AND ">/S" BEHIND IT. THE TEST IS COMPLETE WHEN THE SCREEN IS FULL OF ">/S". THE BBF OR EBF MOVEMENT TAKES PLACE EVERY .1 SECONDS.

# 7. END OF SCREEN TEST: (EOSTST)

THIS TEST CHECKS THAT NOTHING PAST THE "END OF SCREEN" CONTROL BIT SHOULD BE SEEN. LOAD THE BUFFER WITH "BRIGHT ?". START AT THE UPPER LEFT WITH EOS WHICH WILL BE A "CURSOR ?". ON THE LEFT PLACE A "BRIGHT 0" AND BEHIND THAT "X/S". STARTING AT THE UPPER LEFT THE USER SHOULD SEE THE SCREEN BEING FILLED WITH "X/S" WITH A "BRIGHT 0" PRECEDING THEM. TEST IS COMPLETE WHEN "EOS" IS AT THE BOTTOM RIGHT OF THE SCREEN AND THE SCREEN IS FULL OF "X/S". IF AT ANY TIME A "?" IS SEEN THIS CONSTITUTES A FAILURE. EOS IS MOVED EVERY .1 SECONDS.

# 8. SHRINKAGE TEST: (SHRINK)

PERIODS ARE PLACED IN EACH CORNER OF THE SCREEN WITH "CURSOR SPACE" FILLING THE REST OF THE SCREEN. AT ALL TIMES THE "PERIODS" SHOULD REMAIN ALMOST MOTIONLESS. THE DURATION OF THE TEST IS 10 SECONDS.

KEYBOARD TESTS  
\*\*\*\*\*

## 1. KEYBOARD TEST 1 (KEYIST)

THIS TEST ECHOES THE CHARACTER TYPED ON THE KEYBOARD ON THE SCREEN AND CHECKS THAT KIE CAN DISABLE AND ENABLE THE KEYBOARD INTERRUPT. KRB IS TESTED FOR READING THE KEYBOARD BUFFER AND THAT IT CAN CLEAR THE KEYBOARD FLAG. THIS TEST IS ONLY FOR ECHOING CHARACTERS FROM THE STANDARD KEYBOARD.

```

A: CHECK KEYBOARD FLAG TO BE CLEAR AFTER INITIALIZATION.
B: KEYBOARD FLAG CLEAR?
C: NO - HALT (SEE LISTING); YES - NEXT.
D: FILL BUFFER WITH "CURSOR ?".
E: DISPLAY "CURSOR 1".
F: WAIT FOR INPUT FROM THE KEYBOARD.
G: KEYBOARD FLAG SET YET? NO - GO TO F; YES - NEXT.
H: ENABLE THE KEYBOARD INTERRUPT AND TURN THE INTERRUPT ON.
I: DID AN INTERRUPT OCCUR?
J: NO - HALT (SEE LISTING); YES - NEXT.
K: DISABLE THE KEYBOARD INTERRUPT AND TURN THE INTERRUPT ON.
L: DID AN INTERRUPT OCCUR?
M: NO - NEXT; YES - HALT (SEE LISTING)
N: ENABLE THE KEYBOARD INTERRUPT AND TURN THE INTERRUPT ON.
O: DID AN INTERRUPT OCCUR?
P: NO - HALT (SEE LISTING); YES - NEXT.
Q: SET ACB777.
R: READ THE KEYBOARD BUFFER AND CLEAR THE FLAG (KRB).
S: SAVE THE 7-BIT CODE.
T: IS THE KEYBOARD FLAG SET?
U: NO - NEXT; YES - HALT (SEE LISTING).
V: IF A "REBOUT" DELETE A CHARACTER ON THE SCREEN.
W: IF A "CARRIAGE RETURN" - GO TO THE NEXT TEST.
X: MOVE CURSOR RIGHT ONE AND INSERT THE NEW CHARACTER.
Y: IF THE SCREEN IS FILLED WITH CHARACTERS GO TO A.
Z: GO TO B.

```

## 2. KEYBOARD TEST 2: (CURIST)

THIS TEST WILL DISPLAY IN "ENGLISH" THE DEFINITION OF THE SPECIAL FUNCTION KEY TYPED.  
 EGI UP, DOWN, HOME, EOS, EOL, LEFT, RIGHT.  
 AN ILLEGAL CODE WILL DISPLAY "WHAT ?".  
 KCF IS CHECKED THAT IT CAN CLEAR THE KEYBOARD FLAG;  
 CARRIAGE RETURN ENDS THIS TEST.

```

A: INITIALIZE AND DISPLAY "CURSOR 2".
B: WAIT FOR KEYBOARD INPUT.
C: IS THE KEYBOARD FLAG SET?
D: NO - GO TO C; YES - NEXT.
E: CLEAR THE KEYBOARD FLAG WITH KCF.
F: IS THE KEYBOARD FLAG SET?
G: NO - NEXT; YES - HALT (SEE LISTING).
H: SAVE THE 7 BIT CODE.
I: IF A CARRIAGE RETURN - GO TO THE NEXT TEST.
J: IF "CURSOR LEFT" DISPLAY "LEFT".
K: IF "CURSOR RIGHT" DISPLAY "RIGHT".

```



```

L: IF "CURSOR UP" DISPLAY "UP".
M: IF "CURSOR DOWN" DISPLAY "DOWN".
N: IF "EOL" DISPLAY "EOL".
O: IF "EOS" DISPLAY "EOS".
P: IF "HOME" DISPLAY "HOME".
Q: IF NOT I = PI DISPLAY "WHAT ?".
R: GO TO B.

```

### 3. KEYBOARD TEST 3. (OCT)

THIS TEST DISPLAYS THE CHARACTER, ASCII CODE, AND SENSE SWITCH POSITION. CAP IS CHECKED THAT IT CAN CLEAR THE KEYBOARD KEYBOARD FLAG.

```

EGI A = 0301 SW = 1
A A = 0301 SW = 0

```

```

A: INITIALIZE
B: DISABLE THE KEYBOARD INTERRUPT WITH KIE.
C: ISSUE CAF WHICH SHOULD ENABLE THE KEYBOARD INTERRUPT.
D: DISPLAY "CURSOR 3".
E: WAIT FOR INPUT FROM THE KEYBOARD.
F: KEYBOARD FLAG SET? NO - GO TO E; YES - NEXT.
G: READ CHARACTER FROM THE KEYBOARD BUFFER WITH KRS.
H: TURN THE INTERRUPT ON.
I: DID AN INTERRUPT OCCUR?
J: NO - ERROR (SEE LISTING); YES - NEXT.
K: IS THE KEYBOARD FLAG SET?
L: NO - NEXT; YES - HALT (SEE LISTING).
M: RESTART DISPLAY.
N: IF CARRIAGE RETURN WAS TYPED - GO TO NEXT TEST.
O: READ THE SWITCH STATUS FROM THE TERMINAL.
P: IF CODE "XX40" TO CODE "X137" DISPLAY
  "CHARACTER" = AND ASCII CODE, AND
  SWITCH POSITION. EGI A = 0301 SW = 1
Q: IF CODE <0040 OR >0137 DISPLAY THE ASCII
  CODE AND SWITCH POSITION. EGI (RUBOUT) 0377 SW = 1
R: GO TO E.

```

U

U

U

U

U

```

/VT8-E VIDEO DISPLAY TEST 1 MAINDEC=08-DHVTA=B-L
/
/ALPHA=NUMERIC
/COPYRIGHT 1972, 1973, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS, 01754,
/PROGRAMMER ED FORTMILLER
/
/VT8-E EQUATE STATEMENTS:
6050 DPLA=6050 /LOAD STARTING ADDRESS OF DATA BUFFER,
/AC0=11 --> STARTING ADDRESS REGISTER, 0-->AC,
6051 DP00=6051 /LOAD STARTING EXTENDED ADDRESS OF DATA BUFFER,
/AC6=8 --> EXTENDED ADDRESS REGISTER,
/GO = START DISPLAY AFTER NEXT VERTICAL RETRACE IN
/ONE OF FOUR MODES SPECIFIED BY AC10,11,
/AC10 AC11
/ 0 0 ALPHA=NUMERIC; INTERRUPT DISABLED
/ 0 1 ALPHA=NUMERIC; INTERRUPT ENABLED
/ 1 0 GRAPHIC; INTERRUPT DISABLED
/ 1 1 GRAPHIC; INTERRUPT ENABLED,
/
/0 --> AC,
6052 DPSM=6052 /STOP DISPLAY, INHIBIT VIDEO AND FURTHER
/DEVICE INITIATED BREAKS,
/ENTER MAINTENANCE MODE AND IF AC11=1,
/AC6=8 --> EXT, STARTING ADDRESS REGISTER,
/IF AC11=0, PREPARE FOR SINGLE DATA BREAK,
/(STARTING ADDR REG)--> ADDR COUNTER(CA),
/0 --> AC,
6053 DPHB=6053 /MAINTENANCE INSTRUCTION,
/PERFORM A SINGLE ONE-CYCLE DATA BREAK,
/(BREAK ADDRESS)--> INTERFACE DATA BUFFER,
/BREAK ADDR=1 --> BREAK ADDR REGISTER,
6054 DPHD=6054 /MAINTENANCE INSTRUCTION,
/READ DATA BUFFER (DATA BUFFER)--> AC0=11,
6055 DPMS=6055 /MAINTENANCE INSTRUCTION,
/READ EXTENDED BREAK ADDRESS REGISTER
/AND SENSE SWITCH (EXT ADDR REG)--> AC6=8,
6056 DPCL=6056 /SKIP ON REAL TIME CLOCK FLAG AND CLEAR IF SET,
6057 DPBELL=6057 /GENERATE A .5 SECOND BURST OF 1.56 KHZ TONE,

/EQUATE STATEMENTS FOR CREF LISTING:
7402 HLT=7402 /HALT,
7604 LAS=7604 /READ SWITCHES,

```

```

7404 OSR=7404 /INCLUSIVE "OR" SWITCHES TO AC,
7002 BSW=7002 /SWAP BYTES IN AC,

/EQUATE STATEMENTS FOR THE MQ,
7421 MQL=7421 /LOAD MQ FROM AC THEN CLEAR AC,
7501 MQA=7501 /INCLUSIVE "OR" THE MQ WITH THE AC,
7621 CAH=7621 /CLEAR AC AND MQ,
7521 SHP=7521 /SWAP AC AND MQ,
7701 ACL=7701 /LOAD MQ IN TO AC,
7721 CLASHP=7721 /LOAD AC FROM MQ THEN CLEAR MQ,

/MISCELLANEOUS EQUATE STATEMENTS:
0000 OPEN=0000 /PROGRAM MODIFIABLE,
7000 FUTURE=NOP /PATCH SPACE,

/EQUATE STATEMENTS FOR PROCESSOR IOT'S:
6000 SKON=6000 /SKIP IF INTERRUPT ON, AND TURN OFF,
6001 ION=6001 /TURN INTERRUPT ON,
6002 IOF=6002 /TURN INTERRUPT OFF,
6003 SRO=6003 /SKIP ON INTERRUPT REQUEST,
6004 GTF=6004 /GET INTERRUPT FLAGS,
6005 RTF=6005 /RESTORE INTERRUPT FLAGS,
6007 CAF=6007 /CLEAR ALL FLAGS, AC AND LINK,

/EQUATE STATEMENTS FOR MEMORY EXTENSION:
6214 RDF=6214 /READ DATA FIELD,
6224 RIF=6224 /READ INSTRUCTION FIELD,
6234 RIB=6234 /READ INTERRUPT BUFFER,
6244 RMF=6244 /RESTORE MEMORY FIELD,
6201 CDF=6201 /CHANGE DATA FIELD,

/EQUATE STATEMENTS FOR KEYBOARD:
6030 KCF=6030 /CLEAR KEYBOARD FLAG,
6031 KSF=6031 /SKIP ON KEYBOARD FLAG,
6032 KCC=6032 /CLEAR KEYBOARD FLAG AND AC,
6034 KRS=6034 /"OR" KEYBOARD BUFFER WITH AC 5=11
/1-->AC4; 0-->AC0=3,
/ENABLE INTERRUPT IF AC 11 = 1,
/DISABLE INTERRUPT IF AC 11 = 0,
/SEE "KCC" AND "KRS",
6036 KRB=KCC KRS

/DEFINES:
4577 UDPLA=JMS I [EDPLA
4576 UDPCO=JMS I [EDPCO
4575 UDPSH=JMS I [EDPSH
4574 UDPMB=JMS I [EDPMB
4573 UDPMO=JMS I [EDPMO
4572 UDPHS=JMS I [EDPHS

```

```

4571 UOPCL=JMS I [ZOPCL
4570 UOPBEL=JMS I [ZOPBEL
4567 XOPGO=JMS I [XOPGOX
4566 SETUP=JMS I [SETUPX
4565 SHOW=JMS I [SHOWX
4564 LOOP=JMS I [XLOOP
4560 HONEDF=JMS HOME
4563 SR1=JMS I [SW1
4562 SR1A=JMS I [SW1A
4561 SR2=JMS I [SW2
4560 SR3=JMS I [SW3
4557 SR6=JMS I [SW6
4556 SR7=JMS I [SW7

```

## /CONTROL WORD DEFINES:

```

1000 EBF=1000
2000 BBF=2000
3000 EOS=3000
0000 NORMAL=0000
0200 BLINK=0200
0400 BRIGHT=0400
0600 CURSOR=0600

```

## /OTHER DEFINES:

```

0036 LFADD=TEMP2
0037 LFEND=TEMP3
0040 LENGTH=TEMP4
0041 ISTORE=TEMP5
0177 BIT7=177
0200 BEGIN=0200
4700 BUFFER=4700
7400 BWA=BUFFER+100+2400
0375 ALTHOD=375
0377 RUBOUT=377
0212 LINEFD=212
0215 CARRET=215
0010 CURLFT=10
0030 CURRHT=30
0032 CURUP=32
0013 CURDOWN=13
0035 CURHME=35
0036 EOL=36
0037 EOS=37

```

## /MACRO:

```

DEFINE NPAGE
< JMP I (+200&7600 /GO TO NEXT PAGE>

```

## /STARTING ADDRESS:

```

/ 0200

```

## /SWITCH REGISTER SETTINGS:

```

/ SR0 = 0 NORMAL RUN;
/ SR0 = 1 HALT PROGRAM;
/ SR1 = 0 HALT ON ERROR;
/ SR1 = 1 NO HALT ON ERROR;
/ SR2 = 0 NO LOOP;
/ SR2 = 1 REPEAT PATTERN;
/ SR3 = 0 NORMAL RUN;
/ SR3 = 1 REMAIN IN TEST;
/ SR4 = 0 60 CYCLE;
/ SR4 = 1 50 CYCLE;
/ SR5 = 0 ENABLE BACKGROUND JOB;
/ SR5 = 1 DISABLE BACKGROUND JOB;
/ SR6 = 0 64 CHARACTER;
/ SR6 = 1 32 CHARACTER;
/ SR7 = 0 NORMAL;
/ SR7 = 1 ENABLE MANUAL INCREMENT;
/ SR8 WITH SR7 = 1 A CHANGE IN SR8 INCREMENTS
/ THE PATTERN. (VISUAL TESTS ONLY)
/ WITH SR7 = 0 AND SR8 = 1 REMAIN IN
/ SECTION.
/ SR9=11 PLACE VALUE OF HIGHEST MEMORY FIELD.
/ (E.G. 0 FOR 4K, 1 FOR 8K, AND UP TO
/ 7 FOR 32K);
/ REQUIRED BUFFER SIZES:
/ 32 CHAR ALPHA=NUMERIC 20 LINES * 32 CHAR PER LINE = 0640 (DECIMAL)
/ 64 CHAR ALPHA=NUMERIC 20 LINES * 64 CHAR PER LINE = 1280 (DECIMAL)
/ GRAPHICS 192 DOTS (=3) PERLINE * 200 LINES = 3200 (DECIMAL)

```

```

0000 *0
0000 0000 ZERO, OPEN
0001 0000 ONE, OPEN
0002 0000 TWO, OPEN
0003 0000 THREE, OPEN
0004 0000 FOUR, OPEN
0005 0000 FIVE, OPEN
0006 0000 SIX, OPEN
0007 0000 SEVEN, OPEN
0010 *10

```

```

0010 0000 A110, OPEN
0011 0000 A111, OPEN
0012 0000 A112, OPEN
0013 0000 A113, OPEN
0014 0000 A114, OPEN
0015 0000 A115, OPEN
0016 0000 A116, OPEN
0017 0000 A117, OPEN

0020 *20

0020 0503 IQTTAB, 0503
0021 0000 OPEN
0022 0000 OPEN
0023 0000 OPEN
0024 0000 OPEN
0025 0000 OPEN
0026 0000 OPEN
0027 0000 OPEN
0030 0000 0

0031 0000 WORKVT, OPEN

/THESE LOCATIONS ARE CLEARED AT THE START OF EACH TEST,

0032 0000 GDATA, OPEN
0033 0000 BDATA, OPEN
0034 0000 ADATA, OPEN
0035 0000 TEMP1, OPEN
0036 0000 TEMP2, OPEN
0037 0000 TEMP3, OPEN
0040 0000 TEMP4, OPEN
0041 0000 TEMP5, OPEN
0042 0000 COUNT1, OPEN
0043 0000 COUNT2, OPEN
0044 0000 COUNT3, OPEN
0045 0000 TIME, OPEN
0046 0000 TIMEEX, OPEN
0047 0000 CB2CB3, OPEN
0050 0000 CHAR, OPEN
0051 0000 PSET, OPEN
0052 0000 FLO, OPEN
0053 0000 AC, OPEN
0054 0000 HQ, OPEN
0055 0000 LINK, OPEN
0056 0000 RETUJM, OPEN

0057 0000 IDPSM, OPEN /CONTAINS A "DPSM" OR A "NOP",

/THIS ROUTINE SETS THE DF=IF,

0060 0000 HOME, OPEN
0061 6224 RIF /
0062 1155 TAO (CDF 00 /

```

```

0063 3064 DCA ,+1 /
0064 0000 OPEN /CHANGE DF,
0065 5460 JMP I HOME /EXIT OF NOW =/S IF,

0070 *70

/SPECIAL STARTING ADDRESSES:

0070 5754' JMP DVCOD /THIS SPECIAL STARTING ADDRESS IS USED TO.
/CHANGE DEVICE CODES,
/SR0 = 5: DISPLAY
/SR6 =11: KEYBOARD
0071 5753' JMP DSCHAR=1 /THIS STARTS THE PROGRAM AT THE VISUAL SECTION,
0072 5752' JMP KEYST=1 /THIS STARTS THE PROGRAM AT THE KEYBOARD SECTION,
0073 5751' JMP DP /THIS TEST DOES CONTINUOUS MAINTENANCE BREAKS
/USING THE DATA PATTERN IN THE SR AND
/DISPLAYS THE OUTPUT IN THE HQ,

0074 5750' JMP SETCHR /THIS TEST WILL DISPLAY A FULL SCREEN OF
/THE CHARACTER PLACED IN THE SR, AFTER
/PRESSING CONTINUE, SELECT DESIRED SR OPTIONS,

0075 5747' JMP MULVT8 /THIS SPECIAL STARTING ADDRESS IS USED
/TO RUN A SWIRL PATTERN ON 4 VT8E'S
/SIMULTANEOUSLY,

0200 *200

0200 4355 JMS INITOC /INITIALIZE TO THE FIRST DEVICE CODE

/TEST THAT KCC WILL CLEAR THE AC,

0201 4566 CKCC, SETUP /INIT
0202 7240 CLA CMA /7777
0203 6032 KEY17, KCC /CLEAR KBRD FLAG AND AC
0204 7450 SNA /IS AC CLEAR?
0205 5210 JMP ,+3 /YES, KCC OK
0206 4562 SR1A /HALT?
0207 7402 HLT /KCC FAILED TO CLEAR THE AC,
/AC SHOULD BE 0,

0210 4564 LOOP /
0211 0202 KEY17=1 /RETURN POINTER
0212 5201 JMP CKCC /YES,

/CHECK THAT KRB WILL CLEAR THE AC,

0213 4566 CKRB, SETUP /INIT
0214 7240 CLA CMA /7777
0215 6036 KEY18, KRB /TRY TO CLEAR AC
0216 0146 AND /JUST CHECK 0=3
0217 7450 SNA /CLEART
0220 5223 JMP ,+3 /OK
0221 4562 SR1A /HALT?

```

```

0222 7402      HLT              /KRB FAILED TO CLEAR THE AC=3
                                /AC CONTAINS THE RESULT OF KRB;
                                /AC SHOULD BE 0 AT THIS TIME,
                                /
0223 4564      LOOP            /RETURN POINTER
0224 0214      KEY18=1          /YES,
0225 5213      JMP      CKRB

/CHECK THAT KRS WILL NOT CLEAR THE AC,

0226 4566      CKRS,  SETUP      /INIT
0227 7240      CLA CMA          /7777
0228 6034      KEY19,  KRS       /INCLUSIVE OR TO AC,
0229 7421      MQL             /SAVE AC
0230 7501      MQA             /RESTORE AC
0231 7101      CLL IAC         /+1 TO MAKE AC 0 IF OK
0232 7650      SNA CLA        /AC NOW 0?
0233 5241      JMP      ,+4      /YES, KRS OK
0234 4563      SR1            /HALT?
0235 7501      MQA            /RESTORE BAD AC
0236 7402      HLT            /KRS DID NO INCLUSIVE OR WITH AC
                                /AC SHOULD BE 7777,
                                /
0241 4564      LOOP            /RETURN POINTER
0242 0227      CKRS+1          /YES
0243 5226      JMP      CKRS

```

```

/CHECK FOR THE SENSE SWITCH ON DISPLAY TO BE SET,

0244 4566      SENSE,  SETUP      /INIT
0245 4572      UDPHS          /READ SWITCH
0246 7104      CLL RAL        /PUT IN LINK
0247 7210      CLA RAR        /PUT IT BACK IN AC WITH REST OF AC CLEAR,
0248 7440      SEA           /SENSE SWITCH IN THE "1" POSITION?
0249 5234      JMP      ,+3      /YES,
0250 4562      SR1A          /HALT?
0251 7402      HLT            /SENSE SWITCH IN THE "0" POSITION
                                /OR DPHS CANNOT READ IT BACK,
                                /AC CONTAINS WHAT WAS READ FROM
                                /THE STATUS REGISTER, AC SHOULD
                                /BE 4000,
                                /
0254 4564      LOOP            /RETURN POINTER
0255 0245      SENSE+1        /YES,
0256 5244      JMP      SENSE

```

```

/TEST THAT DPLA WILL CLEAR THE AC,

0257 4566      CDPLA,  SETUP      /INITIALIZE
0258 7340      CLA CLL CMA      /7777
0259 4577      UDPLA          /ISSUE DPLA
0260 7450      SNA           /DID THE AC GET CLEARED?
0261 5266      JMP      ,+3      /YES, A-OKAY
0262 4562      SR1A          /SHALL WE HALT?
0263 7402      HLT            /DPLA FAILED TO CLEAR THE AC,

```

```

0266 4564      LOOP            /AC SHOULD BE 0,
0267 0260      CDPLA+1        /RETURN POINTER
0270 5257      JMP      CDPLA    /YES,

/TEST THAT DPSM WILL CLEAR THE AC

0271 4566      CDPSM,  SETUP      /INITIALIZE,
0272 7340      CLA CLL CMA      /7777
0273 4575      UDPSM          /ISSUE DPSM
0274 7450      SNA           /DID THE AC GET CLEARED?
0275 5300      JMP      ,+3      /YES, A-OKAY
0276 4562      SR1A          /HALT?
0277 7402      HLT            /DPSM FAILED TO CLEAR THE AC,
                                /AC SHOULD BE 0,
                                /
0300 4564      LOOP            /RETURN POINTER
0301 0272      CDPSM+1        /YES,
0302 5271      JMP      CDPSM

```

```

/TEST THAT DPCL WILL NOT AFFECT THE AC,

0303 4566      DPCL,  SETUP      /INITIALIZE
0304 7340      CLA CLL CMA      /7777
0305 4571      UDPCL          /ISSUE DPCL
0306 7000      NOP           /COVERS NO SKIP CONDITION
0307 7421      MQL             /SAVE AC
0308 7501      MQA             /RESTORE AC
0309 7001      IAC           /+1 TO MAKE AC 0 FOR CHECK,
0310 7450      SNA           /WAS AC 7777 AFTER DPCL?
0311 5317      JMP      ,+4      /YES,
0312 7501      MQA            /GET AC BACK
0313 4562      SR1A          /HALT?
0314 7402      HLT            /DPCL DID SOMETHING TO THE AC,
                                /AC SHOULD BE 7777,
                                /
0317 4561      SR2            /LOOP?
0318 5304      JMP      CDPL+1   /YES,
0319 7300      DPCL1,  CLA CLL  /0
0320 4571      UDPCL          /ISSUE DPCL
0321 7000      NOP           /IN CASE OF SKIP
0322 7450      SNA           /WAS AC STILL 0?
0323 5330      JMP      ,+3      /YES,
0324 4562      SR1A          /HALT?
0325 7402      HLT            /DPCL SET SOMETHING IN THE AC,
                                /AC SHOULD BE 0,
                                /
0330 4561      SR2            /LOOP?
0331 5321      JMP      DPCL1    /YES,
0332 0205      IS2            /DONE 4096 TIMES?
0333 5304      JMP      DPCL+1   /NO,
0334 4560      SR3            /REMAIN IN THIS TEST?
0335 5303      JMP      DPCL     /YES,

```

/TEST THAT DPGO WILL CLEAR THE AC,

```

0336 4566 CDPGO, SETUP /INITIALIZE;
0337 7340 CLA CLL CHA /SET AC TO 7777
0340 4576 UDPGO /GO, THIS SHOULD ALSO 0 THE AC;
0341 7421 MQL /SAVE THE AC IN THE HQ;
0342 4575 UDP5M /STOP
0343 7501 MQA /HQ TO AC;
0344 7650 SNA CLA /WAS AC 0?
0345 5351 JMP ,+4 /YES, A-OKAY;
0346 4563 SR1 /HALT?
0347 7501 MQA /BAD AC TO THE AC;
0350 7402 HLT /DPGO FAILED TO CLEAR THE AC;
/CONTENTS OF AC AFTER DPGO ARE
/IN THE AC AT THIS TIME;
/AC SHOULD BE 0;
/
0351 4564 LOOP /RETURN POINTER
0352 0337 CDPGO+1 /YES;
0353 5336 JMP CDPGO
NPAGE
0354 5777 JMP I (+20087600 /GO TO NEXT PAGE
/
0355 0000 INIDC, 0
0356 1376 TAD (10TTAB
0357 3031 DCA WORKVT
0360 4775 JMS DEVCOD
0361 5795 JMP I INIDC

```

```

0375 4123
0376 0020
0377 0400
0400

```

PAGE

/TEST OF THE REAL TIME CLOCK;

```

0400 4566 CLOCK, SETUP /INITIALIZE
0401 3035 DCA TEMP1 /0 --> TEMP1 FOR TIMER;
0402 6056 CLOCK1, DPCL /WAIT FOR THE FLAG TO SET
0403 5202 JMP ,+1 /GO WAIT SOME MORE;
0404 6056 CLOCK2, DPCL /IS IT SET BY ANY CHANCE?
0405 5210 JMP ,+3 /NO, FLAG OK;
0406 4562 SR1A /HALT?
0407 7402 HLT /THE CLOCK FLAG SET IMMEDIATELY
/AFTR BEING SET OR DPCL DID NOT
/CLEAR THE CLOCK FLAG OR SKIPPED ON NO FLAG;
/LOOP?
0410 4561 SR2 /YES;
0411 5202 JMP CLOCK1 /MAKE SURE IT SETS BY ABOUT 40 MSEC;
0412 4571 CLOCK3, UOPCL /NOT SET YET;
0413 7410 SKP /SET, OK;
0414 5221 JMP ,+5 /"GROSS" TIMER;
0415 2035 ISZ TEMP1 /GO TIME SOME MORE;
0416 5212 JMP CLOCK3 /FLAG NOT SET, SHALL WE HALT?
0417 4562 SR1A

```

```

0420 7402 HLT /FLAG NOT SET IN ABOUT 40 MSEC;
/OR DPCL FAILED;
/FLAG SHOULD SET IN 16.67 MSEC FOR 60 HZ;
/FLAG SHOULD SET IN 20.0 MSEC FOR 50 HZ;
/LOOP?
0421 4561 SR2 /YES;
0422 5201 JMP CLOCK1-1 /REMAIN IN THIS TEST?
0423 4560 SR3 /YES;
0424 5200 JMP CLOCK

```

```

/TEST THAT THE INTERRUPT ENABLE CAN BE SET
/USING DPGO WITH AC11 = 1;

```

```

0425 4566 INT1, SETUP /GENERAL INITIALIZE;
0426 1145 DECIMAL /
TAD C-10 /NUMBER OF PASSES AS MINUS
0427 3042 OCTAL /
DCA COUNT1 /SAVE
0430 4777 JMS INTLD /SET UP FOR INTERRUPT;
0431 0440 INT1B /INT RETURN ADDRESS
0432 7301 INT1A, CLA CLL IAC /ENABLE INT, BIT
0433 4576 UDPGO /START DISPLAY
0434 4575 UDP5M /STOP
0435 6001 ION /TURN INTERRUPT SYSTEM ON;
0436 4776 JMS I (WAIT /WAIT ABOUT 30 MSEC;
0437 5243 JMP INT1C /NO INTERRUPT;
0440 4571 INT1B, UOPCL /HAS IT THE CLOCK FLAG?
0441 7402 HLT /NO, ILLEGAL INTERRUPT
/NO SCOPE LOOPING PROVIDED;
/NO ATTEMPT WILL BE MADE TO
/CLEAR THE ILLEGAL INTERRUPT;
/RESTART AT BEGINNING OF TEST OR PROGRAM;
/
0442 5246 INT1C, JMP INT1D /CORRECT INTERRUPT
0443 4563 SR1 /HALT?
0444 7402 HLT /WITH CLOCK INTERRUPT ENABLED
/THE CLOCK FLAG FAILED TO INTERRUPT;
/
0445 6002 INT1D, IOF /
0446 4564 LOOP /RETURN POINTER
0447 0432 INT1A /YES;
0450 5225 JMP INT1

```

```

/TEST THAT THE INTERRUPT ENABLE CAN BE DISABLED
/BY ISSUING DPGO WITH A 0 AC;

```

```

0451 4566 INT2, SETUP /GENERAL INITIALIZE;
0452 1145 DECIMAL /
TAD C-10 /NUMBER OF PASSES AS MINUS;
0453 3042 OCTAL /
DCA COUNT1 /SAVE;
0454 4777 JMS INTLD /SET UP FOR INTERRUPTS;
0455 0466 INT2B /RETURN ADDRESS
0456 7301 INT2A, CLA CLL IAC /UDPGO
0457 4576 UDPGO /GO
0460 4576 UDPGO /DISABLE INT;

```

```

0461 4575 UDPSM /STOP
0462 6001 ION /ENABLE INTERRUPT SYSTEM,
0463 4776 JMS I (WAIT /WAIT ABOUT 30 MSEC,
0464 6002 IOF /DISABLE INT, SYSTEM,
0465 5272 JMP INT2C /DISABLE FUNCTIONED OK,
0466 4571 INT2B, UDPCL /CLOCK FLAG CAUSE INT,
0467 7402 HLT /NO, ILLEGAL INTERRUPT,
/NO, SCOPE LOOP PROVIDED OR NO ATTEMPT
/Will BE MADE BY THIS PROGRAM TO LOCATE IT,
/RESTART PROGRAM OR TEST FOR RECOVERY,
0470 4562 SR1A /HALT?
0471 7402 HLT /DPOG FAILED TO DISABLE THE
/CLOCK INTERRUPT FACILITY,
0472 4564 INT2C, LOOP /
0473 0456 INT2A /RETURN POINTER
0474 5251 JMP INT2 /YES,

```

/TEST THAT INITIALIZE CAN DISABLE THE INTERRUPT ENABLE;

```

0475 4566 INT3, SETUP /GENERAL INITIALIZE,
0476 1375 TAD (" &B17 EOS /LOAD THE BUFFER WITH
0477 4544 JMS I CLOBUFF /INVISIBLE CHARACTERS,
0500 4777 JMS INTLO /SETUP FOR INTERRUPTS,
0501 0511 INT3B /RETURN ADDRESS
0502 7301 INT3A, CLA CLL IAC /ENABLE BIT
0503 4576 UDPOG /GO,
0504 6007 CAF /DISABLE THE INT ENABLE
0505 6001 ION /ENABLE INTERRUPTS
0506 4776 JMS I (WAIT /WAIT ABOUT 30 MSEC,
0507 6002 IOF /TURN INT OFF
0510 5313 JMP INT3C /CAF FUNCTIONED OK,
0511 4562 INT3B, SR1A /HALT?
0512 7402 HLT /INITIALIZE FAILED TO DISABLE
/THE CLOCK INTERRUPT ENABLE F,F, OR AN
/ILLEGAL INTERRUPT OCCURRED,
0513 4561 INT3C, SR2 /LOOP?
0514 5302 JMP INT3A /YES,
0515 4560 SR3 /REMAIN IN TEST?
0516 5275 JMP INT3 /YES,

```

/THIS TEST CHECKS THAT THE "EXTENDED STARTING ADDRESS REGISTER"  
/CAN BE LOADED AND READ BACK, AN INCREMENTING PATTERN IS USED  
/DOING EACH PATTERN 4096 TIMES;

```

0517 4566 EXTA, SETUP /INITIALIZE,
0520 1143 DECIMAL /
TAD C=8 /
0521 3036 DCA TEMP2 /ALL 8 PATTERNS OF THE EXT REG,
0522 1032 EXTA2, TAD GDATA /GET PATTERN
0523 7421 MQL /SAVE IT IN THE "MQ"
0524 7501 EXTA3, MQA /MQ ==> AG
0525 7101 CLL IAC /SET AC11 SO THE -

```

```

0526 4575 UDPSM /EXT STARTING REG WILL BE LOADED,
0527 4575 UDPSM /EXT STARTING REG ==> ADDR COUNTER
0530 4572 UDPSM /READ BACK THE EXT STARTING REG,
0531 0142 AND C1777 /ONLY THE EXT BITS,
0532 3033 DCA BDATA /SAVE THE PATTERN,
0533 7501 MQA /"SHOULD BE" TO THE AC,
0534 7041 CMA IAC /NEGATE IT
0535 1033 TAD BDATA /AND ADD BDATA TO AC
0536 7650 SNA CLA /DID WE RECEIVE WHAT WE SENT?
0537 5343 JMP ,+4 /YES, A-OKAY,
0540 4563 SR1 /HALT?
0541 1033 TAD BDATA /"RECEIVED" FROM THE EXT REG TO THE AC
0542 7402 HLT /EXTENDED STARTING ADDRESS REGISTER
/NOT BEING LOADED OR IT CANNOT BE
/READ BACK,
/NO=SENT! AC=RECEIVED,
/LOOP ON SAME DATA?
0543 4561 SR2 /YES,
0544 5324 JMP EXTA3 /DONE THIS PATTERN 4096 TIMES?
0545 2035 IS2 TEMP1 /NO, REPEAT
0546 5324 JMP EXTA3 /UPDATE THE -
0547 7501 MQA /PATTERN,
0550 1141 TAD C10 /ALL PATTERNS DONE?
0551 2036 IS2 TEMP2 /NO,
0552 5322 JMP EXTA2 /YES, REMAIN IN THIS TEST?
0553 4560 SR3 /YES,
0554 5317 JMP EXTA /YES,
0555 5774 NPAGE /GO TO NEXT PAGE
JMP I C1+20067600

```

```

0574 0600
0575 3040
0576 4400
0577 4163
0600

```

PAGE

/TEST THAT +1 TO THE EXTENDED ADDRESS COUNTER  
/WILL ONLY OCCUR WHEN THE ADDRESS COUNTER  
/INCREMENTS FROM 7777 ==> 0000;

```

0034 ADDR=ADATA
0032 EXTADR=GDATA

0600 4566 EXTIN, SETUP /INITIALIZE,
0601 1034 EXTIN1, TAD ADDR /LOAD THE "ADDRESS REGISTER" -
0602 4577 UDPLA /IN THE VT8-E,
0603 7101 CLL IAC /SET "EXT ADDRESS REGISTER" -
0604 4575 UDPSM /TO 0,
0605 4575 UDPSM /ADDRESS REGISTERS ==> ADDRESS COUNTERS,
0606 1034 EXTIN2, TAD ADDR /SIMULATE +1 TO -
0607 7101 CLL IAC /THE ADDRESS COUNTER AND -
0610 7421 MQL /KEEP IT IN THE MQ,
0611 7430 SEL /IS IT GOING TO BE 7777 ==> 0000,
0612 7103 CLL IAC BSW /YES, 100 TO "EXTADR" FOR -

```



```

0613 3032 DCA EXTADR /SIMULATED ANSWER;
0614 4574 UDPMB /BREAK;
0615 4572 UDPMB /READ THE EXTENDED ADDRESS COUNTER;
0616 0377 AND (70 /KEEP ONLY THE EXTENDED BITS;
0617 3033 DCA BDATA /SAVE THEM;
0620 7501 MQA /WAS A +1 TO THE
0621 7650 SNA CLA /EXTENDED ADDRESS COUNTER GENERATED?
0622 1143 TAD C=10 /YES, IT WAS SUPPOSED TO,
0623 1033 TAD BDATA /NO,
0624 7650 SNA CLA /IS THE EXTENDED ADDRESS COUNTER CORRECT?
0625 5235 JMP EXTIN3 /YES, OK;
0626 7604 LAS /HALT;
0627 7104 CLL RAL /
0630 7710 SPA CLA /
0631 5235 JMP EXTIN3 /NO HALT;
0632 1033 TAD BDATA /COMBINE "EXPECTED" AND "RECEIVED" -
0633 1032 TAD EXTADR /INTO ONE WORD;
0634 7402 HLT /EXTENDED ADDRESS COUNTER ERROR,
/ THE EXTENDED ADDRESS COUNTER WAS
/ INCREMENTED AT SOME OTHER TIME THEN
/ WHEN THE ADDRESS COUNTER INCREMENTED
/ FROM 7777 TO 0000;
/ AC 3=5 CONTAINS "SHOULD BE";
/ AC 6=8 CONTAINS "RECEIVED";
/ HQ CONTAINS WHAT THE ADDRESS COUNTER
/ SHOULD CONTAIN;
/ LOOP?
0635 4561 EXTIN3, SR2 /YES,
0636 5201 JMP EXTIN1 /SETUP FOR NEXT;
0637 2034 ISZ ADDR /DO ANOTHER;
0640 5206 JMP EXTIN2 /
0641 7501 MQA /
0642 7640 SZA CLA /
0643 7402 HLT /
0644 4560 SR3 /REMAIN IN THIS TEST?
0645 5200 JMP EXTIN /YES,

```

/TEST THAT THE EXTENDED ADDRESS COUNTER  
/WILL INCREMENT CORRECTLY,  
/LOAD THE STARTING ADDRESS REGISTER  
/WITH 7777 AND DO A BREAK, THE EXTENDED ADDRESS  
/REGISTER SHOULD BE +1 AFTER THE BREAK,

```

0646 4566 EXT, SETUP /INITIALIZE;
0647 1143 EXT1, TAD C=8 /COUNTER FOR INCREMENTING
OCTAL /
DCA COUNT1 /SAVE THE END COUNT;
0650 3042 CLA CLL CMA /7777
0651 7340 UDPLA /LOAD THE ADDRESS REGISTER WITH 7777;
0652 4577 TAD EXTADR /GET THE EXTENDED ADDRESS;
0653 1032 TAD C10 /ADD 10 FOR THE SIMULATED;
0654 1141 AND (70 /MASK FOR OVERFLOW
0655 0377

```

```

0656 7421 MQL /SAVE AS THE SIMULATED;
0657 1032 TAD EXTADR /EXTENDED ADDRESS
0660 7101 CLL IAC /+1 SO MAINTENANCE WILL LOAD THE CORRECT
/REGISTER;
0661 4575 UDPSH /AC 6=8 ==> EXT STARTING ADD, REG;
0662 4575 UDPSH /EXT STARTING ADD, REGS, ==> ADD COUNTERS;
0663 4574 UDPMB /BREAK, WHICH WILL INCREMENT ADD, COUNTER;
0664 4572 UDPMB /READ THE EXT ADD, COUNTER BACK;
0665 0377 AND (70 /ONLY EXT BITS;
0666 3033 DCA BDATA /SAVE IT;
0667 7501 MQA /SIMULATED TO AC;
0670 7041 CMA IAC /NEGATE FOR CHECKING;
0671 1033 TAD BDATA /ADD RECEIVED;
0672 7650 SNA CLA /SIMULATED SAME AS RECEIVED?
0673 5277 JMP EXT4 /YES, OK;
0674 4563 SR1 /HALT;
0675 1033 TAD BDATA /RECEIVED TO THE AC;
0676 7402 HLT /EXTENDED ADDRESS COUNTER ERROR,
/ THE EXTENDED ADDRESS COUNTER FAILED
/ TO INCREMENT CORRECTLY WHEN
/ THE ADDRESS COUNTER INCREMENTED FROM
/ 7777 TO 0000;
/ AC CONTAINS THE EXTENDED ADDRESS COUNTER;
/ HQ CONTAINS WHAT THE EXT, ADD, COUNTER SHOULD BE;
/ LOOP?
0677 4561 EXT4, SR2 /YES,
0678 5257 JMP EXT3 /UPDATE EXTADR FOR NEXT PATTERN
0679 1032 TAD EXTADR /
0680 1141 TAD C10 /
0681 3032 DCA EXTADR /UPDATE DONE;
0682 2042 ISZ COUNT1 /ALL PATTERNS DONE?
0683 5253 JMP EXT2 /NO;
0684 2043 ISZ COUNT2 /HAS THIS TEST MADE 4096 PASSES?
0685 5247 JMP EXT1 /NO;
0686 4560 SR3 /REMAIN IN THIS TEST?
0687 5246 JMP EXT /YES,

```

/TEST THAT DPGO CAN LOAD THE  
/EXTENDED STARTING ADDRESS REGISTER,

```

0712 4566 EDPGO, SETUP /INITIALIZE;
0713 1376 TAD C=100 /
/DO TEST THIS NUMBER OF TIMES,
/
0714 3044 DCA COUNT3 /SAVE COUNT;
0715 4577 UDPLA /STARTING ADDRESS OF 0
0716 7301 EDPGO1, CLA CLL IAC /SET MAINT AND, 0 ==> EXT ADDR REG;
0717 4575 UDPSH /DO ABOVE;
0720 4575 UDPSH /ADDR REG ==> COUNTERS
0721 1377 TAD (70 /LOAD FOR FIELD7
0722 7421 MQL /SAVE IN HQ;
0723 7501 MQA /70
0724 4576 UDPGO /TEST
0725 4575 UDPSH /ADDR REG ==> COUNTERS

```

```

0726 4572 UDPMH /READ BACK EXT ADDR COUNTER;
0727 0377 AND (78 /SAVE ONLY FIELD BITS;
0730 3033 DCA BDATA /SAVE ALWAYS
0731 7501 MQA /GOOD TO AC FOR CHECK
0732 7041 CMA IAC /NEGATE GOOD
0733 1033 TAD BDATA /ADD RECEIVED TO CHECK IF SAME
0734 7650 SNA CLA /EXPECTED OK
0735 5341 JMP EDPG02 /OK
0736 4563 SR1 /HALT?
0737 1033 TAD BDATA /BAD TO AC
0740 7402 HLT /DPG0 FAILED TO LOAD THE EXT
/STARTING REG; AC SHOULD = 70;
/AC CONTAINS THE VALUE OF THE EXT, ADDRESS COUNTER;
/LOOP?
0741 4561 EDPG02, SR2 /YES,
0742 5316 JMP EDPG01 /REMAIN IN THIS TEST?
0743 4560 SR3 /YES,
0744 5312 JMP EDPG0
NPAGE
0745 5775 JMP I (,420087600 /GO TO NEXT PAGE

0775 1000
0776 7634
0777 0070
1000

```

PAGE

/TEST OF THE VT8-E BUFFER USING CONSTANT DATA  
/PATTERNS: 7777 0000 7777 5252 2929 7700 0077 7007 0770.  
/GET THE DATA AND DO 31 OR 63 BREAKS TO PRIME THE BUFFER  
/NOW DO 4096 BREAKS ON THE DATA CHECKING AFTER EACH BREAK;

```

1000 4566 DATA1, SETUP /INITIALIZE
1001 1140 TAD /ADDRESS OF DATA
1002 3034 DCA ADATA /STORE FOR ERROR DISPLAY,
1003 7201 CLA IAC /0001
1004 6214 ROP /GET FIELD
1005 4575 UDPMH /SET MAINTENANCE MODE
1006 6214 ROP /FIELD TO AC
1007 3032 DCA FLD /FOR ERROR REPORTING
1010 1377 TAD (FDATAB=1 /SET AUTO-INDEX FOR -
1011 3040 DCA A110 /OBTAINING COUNT AND DATA;
1012 1410 TAD I A110 /GET NUMBER OF PATTERNS;
1013 3035 DCA TEMP1 /SAVE,
1014 1410 DATA1A, TAD I A110 /GET DATA,
1015 3032 DCA GDATA /SAVE,
1016 3042 DCA COUNT1 /SET FOR 4096 COUNT,
1017 7201 CLA IAC /1
1020 4537 JMS I CPRIME /PRIME THE BUFFER -
1021 0032 GDATA /
1022 1140 TAD /SET BUFFER -
1023 4577 UDPLA /ADDRESS;
1024 4575 DATA1B, UDPMH /START ADDR REGS ==> ADDR COUNTER,
1025 4574 UDPMH /SINGLE BREAK;

```

```

1026 4573 UDPMH /LOAD AC FROM DATA BUFFER;
1027 3033 DCA BDATA /SAVE RECEIVED DATA;
1030 1032 TAD GDATA /TAKE THE EXPECTED DATA AND -
1031 7041 CMA IAC /NEGATE IT; NOW ADD -
1032 1033 TAD BDATA /THE RECEIVED DATA AND SEE -
1033 7650 SNA CLA /IF THEY ARE THE SAME,
1034 5240 JMP ,+4 /DATA A-OKAY;
1035 4563 SR1 /SHALL WE HALT?
1036 7402 HLT /RECEIVED NOT SAME AS EXPECTED; PRESS CONTINUE
1037 4565 SHON /TO DISPLAY DATA, AC=RECEIVED; HQ=EXPECTED,
1040 4561 SR2 /SHALL WE LOOP?
1041 5224 JMP DATA1B /YES,
1042 2042 ISZ COUNT1 /DONE 4096 TIMES?
1043 5224 JMP DATA1B /NO,
1044 2035 ISZ TEMP1 /ALL PATTERNS DONE?
1045 5214 JMP DATA1A /NO,
1046 4560 SR3 /REMAIN IN THIS TEST?
1047 5200 JMP DATA1 /YES,
NPAGE
1050 5776 JMP I (,420087600 /GO TO NEXT PAGE

```

/MANUAL INCREMENT ROUTINE  
/SR7 SET ENABLES THE ROUTINE.  
/A CHANGE IN SR8 MANUALLY STEPS  
/DISPLAY SINGLE STEP.

```

1051 0000 BUMP, OPEN /READ SWITCHES;
1052 7604 LAS /MASK FOR SR7;
1053 0136 AND C20 /WAS SR7 SET? ALSO READ SW,
1054 7494 SNA OSR /SR7 NOT SET;
1055 5270 JMP BUMP1 /MASK FOR SR8;
1056 0141 AND C10 /IF SET; AC=7777; IF NOT; AC=0
1057 7640 SZA CLA /SR8 SET;
1060 7040 CMA /ADD THIS
1061 1051 TAD PSET /1
1062 7001 IAC /
1063 7640 SZA CLA /
1064 5651 JMP I BUMP /COMPLEMENT -
1065 1051 TAD PSET /THE CHANGE -
1066 7140 CLL CMA /INDICATOR;
1067 3051 DCA PSET /1 FOR EXIT;
1070 2251 BUMP1, ISZ BUMP /AC=L=0,
1071 7300 CLA CLL /EXIT,
1072 5651 JMP I BUMP

```

/INITIALIZATION ROUTINE.

```

1073 0000 SETUPX, OPEN /CLEAR ALL;
1074 6007 CAF /
1075 4560 SR3 /NO TUNE IF SR3 IS SET;
1076 7610 SKP CLA /SOUND A TUNE AT THE BEGINNING OF
1077 4570 UDPMH /EACH TEST;
/0 ==> AC+HQ;
1100 7621 CAM /RMF INSTRUCTION
1101 1375 TAD (RMF

```

```

1102 6201 CDF 00 /DF 0
1103 3774 DCA I (ONE /PUT IN 1 OF FIELD 0
1104 1373 TAD (JMP ONE /JMP 1
1105 3772 DCA I (TWO /PUT IN 2 OF FIELD 0
1106 4060 HOMEDF /DF = IF
1107 1135 TAD (JMP I TWO /PUT IN PROGRAM FIELD;
1110 3001 DCA ONE /LOCATION 1,
1111 1371 TAD (=LINK+GDATA=1 /SETUP TO ZERO PAGE ZERO STORAGE;
1112 3007 DCA SEVEN /SAVE COUNT OF LOCATIONS
1113 1370 TAD (GDATA=1 /SET AUTO-INDEX
1114 3010 DCA A110 /SAVE
1115 3410 DCA I A110 /0 ==> STORAGE LOCATION
1116 2007 ISZ SEVEN /DONE?
1117 5315 JMP ,=2 /NO, GO ZERO ANOTHER;
1120 1134 TAD (ISZTST /SET UP BACKGROUND JOB
1121 3533 DCA I (BACKST /
1122 3532 DCA I (BUFFER /0 ==> FIRST LOCATION IN THE BUFFER.
1123 5673 JMP I SETUPX /EXIT, AC=L+FLAGS=0

```

```

1170 0031
1171 7754
1172 0002
1173 5001
1174 0001
1175 6244
1176 1200
1177 4547
1200

```

PAGE

/THIS IS A TEST OF THE VT8-E DATA BUFFER  
/USING A SPECIAL BINARY COUNT PATTERN,  
/(THE BINARY COUNT AND ITS COMPLEMENT)  
/THE DATA APPLIED TO THE INPUT IS DELAYED 32 OR  
/64 BREAKS BEFORE SEEN ON THE OUTPUT, IF  
/SR2 IS SET TO LOOP THEN AFTER 32 OR 64 BREAKS THE SAME  
/DATA AS APPLIED WILL CONTINUOUSLY BE SEEN, WHEN SR2 IS  
/LOWERED THE PROGRAM WILL PICK UP WHERE IT LEFT OFF +1,  
/[THE BUFFER HOLDS THE SPECIAL BINARY COUNT INSTEAD OF BEING  
/FILLED WITH THE SAME DATA AS IN "DATA1",]

/DEFINES:

```

0035 NOCNT=TEMP1
0036 LOOPFL=TEMP2
0037 INDATA=TEMP3
0040 FILBUF=TEMP4
0043 UPDOWN=COUNT2

```

```

1200 4566 DATA2, SETUP /INITIALIZE.
1201 1132 TAD (BUFFER /ADDRESS OF DATA
1202 3034 DCA ADATA /STORE FOR ERROR DISPLAY
1203 7201 CLA IAC /0001
1204 6214 RDF /GET FIELD
1205 4575 UDPSM /SET MAINT, AND FIELD;

```

```

1206 6214 RDF /FIELD TO AC
1207 3052 DCA FLD /FOR ERROR REPORTING;
1210 1132 TAD (BUFFER /
1211 4577 UDPLA /AC ==> STARTING ADDR REG;
1212 7344 CLA CLL CMA RAL /2,
1213 3043 DCA UPDOWN /2 ==> UPDOWN;
1214 4557 SR6 /32 FOR 64?
1215 1131 TAD (32 /SET A COUNTER SO A FULL =
1216 1130 TAD (=64 /BUFFER CAN BE DETERMINED;
1217 1036 TAD /
1220 3040 DCA LOOPFL /IF LOOP WAS SET, THEN 1 MORE BREAK,
1221 1036 TAD FILBUF /SAVE THE COUNT,
1222 7650 SNA CLA /
1223 5237 JMP DATA2B /IF LOOP FLAG SET THEN ZERO IT;
1224 1043 TAD UPDOWN /ITS ZERO,
1225 7130 CLL CML RAR /LOOK AT THE WAY WE WERE COUNTING
1226 7430 SEL /IF 7776 THEN MAKE IT 7777;
1227 7344 CLA CLL CMA RAL /IF 7777 THEN MAKE IT 7776;
1230 3043 DCA UPDOWN /7776
1231 7630 SEL CLA /DIRECTION OF COUNT NOW RESET,
1232 5236 JMP DATA2B-1 /CHECK IF DOWN;
1233 2042 ISZ COUNT1 /UP.
1234 7610 SKP CLA /UPDATE COUNT1 SINCE LOOP BYPASSED IT;
1235 5352 JMP DATA2K /COUNT DID NOT GO TO ZERO;
1236 3036 DCA LOOPFL /COUNT1 OVERFLOWED SO LOOK AT SR3 TO DECIDED WHAT TO DO,
1237 1037 TAD INDATA /0 ==> LOOPFL;
1240 3532 DCA I (BUFFER /GET THE DATA TO BE PUT IN THE =
1241 4575 UDPSM /BUFFER NEXT AND SAVE IT,
1242 4574 UDPSM /START ADDR REG ==> ADDR COUNTER;
1243 4573 UDPSM /SINGLE BREAK;
1244 3033 DCA BDATA /READ VT8-E BUFFER;
1245 1040 TAD FILBUF /AND SAVE IT;
1246 7650 SNA CLA /IF VT8-E BUFFER IS CLEAR OF GARBAGE
1247 5252 JMP ,+3 /THEN CHECKS CAN BE MADE;
1250 2040 ISZ FILBUF /OK TO CHECK;
1251 5331 JMP DATA2H /CLEAR OF JUNK YET?
1252 1036 TAD LOOPFL /NO.
1253 7640 SZA CLA /IF LOOPFL IS 0 THEN WE MUST =
1254 5301 JMP DATA2E /CALCULATE THE OUTPUT OF THE BUFFER,
1255 1043 TAD UPDOWN /LOOPFL=1,
1256 7110 CLL RAR /COUNTING UP OR DOWN?
1257 7630 SEL CLA /CHECK AC11
1260 5270 JMP DATA2C /SKIP IF "UP"
1261 1037 TAD INDATA /"DOWN"
1262 7040 CMA /FROM AN "UP" =
1263 3041 DCA TEMP5 /CALCULATE THE "DOWN" =
1264 4557 SR6 /DATA,
1265 1377 TAD (=16 /32 OR 64?
1266 1131 TAD (32 /
1267 5276 JMP DATA2D /
1270 1037 DATA2C, TAD INDATA /FROM A "DOWN" =

```

1271	7041	CHA	IAC	/CALCULATE THE -
1272	3041	DCA	TEMP5	/"UP" DATA,
1273	4537	SR6		/32 OR 64?
		DECIMAL		/
1274	1376	TAD	(16	/32
1275	1375	TAD	(-32	/64
		OCTAL		/
1276	1041	DATA2D,	TAD	TEMP5
1277	3032	DCA	GDATA	/ADD TO TEMP5
1300	3035	DCA	NOCNT	/SAVE AS THE EXPECTED DATA
1301	1032	DATA2E,	TAD	GDATA
1302	7041	CHA	IAC	/ALWAYS 0 NOCNT
1303	1033	TAD	BDATA	/TAKE THE "EXPECTED" DATA -
1304	7650	SNA	CLA	/AND NEGATE IT, NOW ADD -
1305	5311	JMP	DATA2F	/THE RECEIVED DATA AND SEE IF -
1306	4563	SR1		/ITS THE SAME?
1307	7402	HLT		/DATA A-OKAY?
				/HALT ON ERROR?
				/YES, DATA RECEIVED IS INCORRECT,
				/PRESS CONTINUE TO DISPLAY THE -
				/"BAD" DATA IN THE "AC" AND THE
				/"GOOD" IN THE "MQ",
				/SHOW THE DATA,
				/LOOP ON DATA ON OUTPUT?
				/YES,
				/WERE WE LOOPING?
				/CHECK?
				/YES, MUST RESET,
				/NO,
				/ARE WE ALL READY LOOPING?
				/CHECK?
				/YES,
				/SET THE DATA SO WE'RE -
				/CONSTANT INSTEAD OF COUNTING,
				/PRIME THE BUFFER,
				/WITH CONSTANT DATA,
				/SET LOOP FLAG,
				/7777 ==> LOOPFL,
				/TEST,
				/SET NOCNT SO A COUNT -
				/ISN'T STARTED UNTIL JUNK IS CLEARED,
				/GET LAST DATA TO ENTER BUFFER,
				/COMPLEMENT IT,
				/NEXT DATA TO BE UP OR DOWN?
				/"DOWN"
				/+1 TO MAKE NEW "UP" DATA
				/STORE,
				/RESET COUNT,
				/SAVE,
				/GO LOAD BUFFER WITH NEW,
				/SAVE NEW "DOWN",
				/SHALL THE COUNT BE UPDATED?
				/CHECK?
				/NO,
				/YES, DONE TEST?
				/NO,
				/REMAIN IN THIS TEST?

1353	5200	JMP	DATA2	/YES,
		NPAGE		/
1354	5774	JMP I	(,=200&7600	/GO TO NEXT PAGE

1374	1400
1375	7740
1376	0020
1377	7760
	1400

PAGE

/ADDRESS TEST  
/BREAK FROM EVERY ADDRESS USING THE ADDRESS  
/AND ITS COMPLEMENT FOR DATA,  
/THE ADDRESS SELECTED TO BE TESTED HAS ITS CONTENTS  
/SAVED IN THE "MQ", THE VALUE OF THE ADDRESS IS PLACED IN  
/THE ADDRESS AND AFTER 32 OR 64 BREAKS ITS READ BACK AND  
/CHECKED, ADDRESS IS THEN RESTORED AND THE COMPLEMENT OF THE  
/ADDRESS IS CALCULATED AND AGAIN THE ADDRESS IS SAVED IN THE  
/"MQ", THE COMPLEMENT PATTERN IS PLACED IN THE ADDRESS AND  
/32 OR 64 BREAKS ARE DONE AND THE DATA CHECKED,  
/THIS TEST STARTS AT FIELD 0 ADDRESS 0000 AND CHECKS EVERY ADDRESS  
/IN EVERY FIELD FOR AS MANY FIELDS AS SELECTED IN SR 9-11,

1400	4566	ADDR1,	SETUP	/INITIALIZE,
1401	1777	TAD	DDPSM	/GET A "DDPSM" INSTRUCTION
1402	3057	DCA	IDPSM	/AND STORE IT SO BREAK ROUTINE
				/CAN GET THE INSTRUCTION,
				/32 OR 64?
				/
1403	4557	SR6		/32
		DECIMAL		/64
1404	1131	TAD	(32	/
1405	1130	TAD	(-64	/
		OCTAL		/
1406	3042	DCA	COUNT1	/SAVE
1407	7604	LAS		/READ FIELD FROM SW,
1410	0376	AND	(7	/KEEP ONLY FIELD BITS,
1411	7040	CHA		/CHANGE FOR A COUNT
1412	3044	DCA	COUNT3	/SAVE COUNT,
1413	7344	ADDR1A,	CLA CLL	/2
1414	3043	DCA	COUNT2	/THIS COUNT SERVES TO DETERMINE
				/WHETHER ITS THE ADDRESS OR ITS
				/COMPLEMENT,
				/GET ADDRESS TO BE TESTED
				/ADDRESS ==> STARTING ADDR REG
				/0001,
				/GET FIELD
				/SET MAINTENANCE
				/
1415	1034	TAD	ADATA	/GET ADDRESS TO BE TESTED,
1416	4577	UDPLA		/ADD THE ADDRESS TO THE ROUTINE
1417	7301	CLA CLL	IAC	/TO STORE DATA IN THE ADDRESS AND
1420	1052	TAD	FLD	/DO THE BREAKS, IF THE ADDRESS EQUALS
1421	4575	UDPSM		/THE ROUTINE ADDRESS, ROUTINE "LOADDA"
1422	7100	CLL		/IS THEN USED TO PERFORM THE SAME FUNCTION,
1423	1034	ADDR1B,	TAD	
1424	1375	TAD	(-LOADDB	

```

1425 7630 SEL CLA /ADDRESS EQUAL TO LDADD0?
1426 4774 JMS LDADD0 /YES,
1427 4773 JMS LDADD0 /
1430 6094 I1, DPMO /READ BACK BUFFER,
1431 3033 DCA BDATA /SAVE THE DATA
1432 1037 TAD INDATA /GET WHAT SHOULD HAVE COME BACK
1433 7041 CMA IAC /NEGATE IT,
1434 1033 TAD BDATA /ADD RECEIVED AND SEE IF ITS OK
1435 7650 SNA CLA /WAS IT OK?
1436 5244 JMP ADDR10 /YES, OK,
1437 1037 TAD INDATA /SET THE EXPECTED FOR ERROR REPORTING,
1440 3032 DCA GDATA /SAVE EXPECTED,
1441 4563 SR1 /HALT?
1442 7402 HLT /YES, RECEIVED THE INCORRECT DATA FROM
/ THE VT8-E DATA BUFFER, PRESS CONTINUE
/ TO DISPLAY THE GOOD AND BAD DATA
/ AND THE ADDRESS BEING TESTED,
/ DISPLAY INFORMATION,
/ SEE THE LISTING AT THE HALT ADDRESS,

1443 4565 SHOW /LOOP?
1444 4561 ADDR10, SR2 /YES,
1445 5222 JMP ADDR1B-1 /GET THE DATA
1446 1037 TAD INDATA /COMPLEMENT IT,
1447 7140 CLL CMA /COMPLEMENT OR ADDRESS?
1450 2043 ISZ COUNT2 /COMPLEMENT OF ADDRESS,
1451 5266 JMP ADDR1C /+1 TO MAKE DATA FOR NEXT ADDRESS,
1452 7001 IAC /SAVE NEW DATA
1453 3037 DCA INDATA /MAKE NEW ADDRESS, DONE ALL?
1454 2034 ISZ ADATA /NOT ALL DONE,
1455 5213 JMP ADDR1A /GET PRESENT FIELD
1456 1052 TAD FLD /ADD 10 FOR UPDATE
1457 1141 TAD C10 /SAVE NEW FIELD TO BE TESTED,
1460 3052 DCA FLD /ALL FIELDS DONE?
1461 2044 ISZ COUNT3 /NO, DO NEXT,
1462 5213 JMP ADDR1A /REMAIN IN THIS TEST?
1463 4560 SR3 /YES,
1464 5200 JMP ADDR1 /GO TO NEXT TEST
NPAGE /GO TO NEXT PAGE
1465 5772 JMP I ('+20047600 /GO TO NEXT PAGE
1466 3037 ADDR1C, DCA INDATA /SAVE COMPLEMENT DATA
1467 5222 JMP ADDR1B-1 /DO IT,

/THIS ROUTINE RUNS A SWIRL PATTERN SIMULTANEOUSLY ON 4 VT8E'S

1470 4566 MULVT8, SETUP
1471 1127 TAD C'148117 NORMAL
1472 3050 DCA CHAR
1473 4771 JMS L0SWL
1474 1370 TAD (3000
1475 3767 DCA I (23+100+BUFFER+100-1
1476 1132 TAD CBUFFER
1477 3316 DCA BUFCA
1500 1366 TAD (MULTAB
1501 3031 DCA WORKVT
1502 4765 MULV1, JMS DEVCOD
1503 1316 TAD BUFCA

```

```

1504 4577 UPPLA
1505 6214 RDP
1506 4576 UDFGO
1507 2316 ISZ BUFCA
1510 2031 ISZ WORKVT
1511 1431 TAD I WORKVT
1512 7640 SEA CLA
1513 5302 JMP MULV1
1514 7001 IAC
1515 5314 JMP ,=1

1516 0000 BUFCA, OPEN

1517 0503 MULTAB, 0503
1520 1513 1513
1521 2523 2523
1522 3533 3533
1523 0000 0

```

```

1565 4123
1566 1517
1567 7277
1570 3000
1571 3054
1572 1600
1573 4453
1574 4424
1575 3325
1576 0007
1577 4011
1600 1600

```

PAGE

/CURRENT ADDRESS INCREMENT TEST!  
/THIS TEST PLACES THE VALUE OF THE ADDRESS IN THE ADDRESS  
/AND DOES A BREAK STARTING AT ADDRESS 0 CURRENT FIELD AND PERFORMES  
/THIS PROCEDURE FOR EVERY ADDRESS IN THE CURRENT FIELD, THE DATA  
/COMING OUT OF THE BUFFER IS LAGGING THE INPUT BY 32 OR 64 BREAKS,  
/IF LOOP IS SET, BREAKS USING THE SAME ADDRESS AND DATA WILL BE  
/DONE, RESETING LOOP RESUMES THE PROGRAM FROM WHERE  
/IT LEFT OFF +1.

```

1600 4566 CAINC, SETUP /INITIALIZE
1601 7201 CLA IAC /SET 11 SO THE EXTENDED
1602 6214 RDP /ADDRESS WILL BE LOADED,
1603 4575 UDFSM /LOAD EXT ADDRESS REGISTER,
1604 6214 RDP /FIELD TO AC
1605 3052 DCA FLD /SAVE FOR ERROR REPORTING
1606 4557 SR6 /32 OR 64?
/
1607 1377 TAD (=32 /32
1610 1376 TAD (=64 /64
/
1611 7041 CMA IAC /NEGATE FOR COUNT

```



```

1747 5767' JMP CKCC /
1750 4766' JMS SR7SR0 /
1751 5526 JMP I CBEGIN /
NPAGE /
1752 5765 JMP I (+20087600 /GO TO NEXT PAGE

```

```

1765 2000
1766 3747
1767 0201
1770 4123
1771 4011
1772 4453
1773 4424
1774 3325
1775 7000
1776 0100
1777 7740
2000

```

PAGE

/DISPLAY FULL SCREEN OF A SINGLE CHARACTER  
/FOR A PERIOD OF 2 SECONDS, STARTING WITH  
/7 BIT CODE 40, WHILE DISPLAYING A  
/BACKGROUND JOB IS BEING RUN UNLESS THE  
/SR PREVENTS IT,

```

2000 4777' JMS INITDC /INITIALIZE THE DEVICE CODE
2001 4566 DSCHAR, SETUP /INITIALIZE,
2002 1131 TAD C" &BIT7 NORMAL /7-BIT CODE FOR SPACE,
2003 3050 DCA CHAR /START WITH SPACE,
2004 1050 DSCHA1, TAD CHAR /COMBINE CHARACTER,
2005 1047 TAD CB2CB3 /AND CONTROL BITS,
2006 4544 JMS I CLDBUFF /LOAD THE BUFFER,
2007 4525 DSCHA2, JMS I CSETTIM /SET TIME TO DISPLAY
DECIMAL
2010 0024 2*10 /2 SECONDS,
OCTAL
2011 4556 SR7 /MANUAL INCREMENT,
2012 5215 JMP ,+3 /NO, SELF TIMED
2013 7344 DSCHA3, CLA CLL CMA RAL /-2
2014 3046 DCA TIMEX /
2015 1132 TAD CBUFFER /BEGINNING OF BUFFER
2016 4547 XDPGO /GO DISPLAY
2017 4524 JMS I CBUMP /CHECK FOR MANUAL INCREMENT,
2020 5213 JMP DSCHA3 /DISPLAY SAME,
2021 4523 JMS I CINCBUF /INCREMENT ENTIRE BUFFER,
2022 5207 JMP DSCHA2 /DISPLAY
2023 4522 JMS I CB3CB2 /INCREMENT CONTROL BITS,
2024 5204 JMP DSCHA1 /GO DISPLAY
2025 4575 UDPSH /STOP DISPLAY,
2026 4540 SR3 /REMAIN IN THIS TEST?
2027 5201 JMP DSCHAR /YES,

```

/THIS TEST DISPLAYS A RIPPLE PATTERN CHANGING

/THE CONTROL BIT EVERY 5 SECONDS,

```

2030 4566 DISRIP, SETUP /INITIALIZE,
2031 4525 DISR11, JMS I CSETTIM /SET TIMER FOR-
DECIMAL
2032 0062 5*10 /TIME
OCTAL
2033 4776' JMS LOINC /LOAD BUFFER WITH A RIPPLE PATTERN
2034 1132 TAD CBUFFER /STARTING ADDRESS OF BUFFER
2035 4567 XDPGO /GO DISPLAY
2036 4575 UDPSH /STOP WHILE CHANGING DATA,
2037 4522 JMS I CB3CB2 /INCREMENT CONTROL BIT,
2040 5231 JMP DISR11 /REPEAT TEST WITH NEW CONTROL BIT
2041 4540 SR3 /REMAIN IN TEST?
2042 5230 JMP DISRIP /YES,

```

/TEST TO DISPLAY A SWIRL PATTERN

```

2043 4566 DISSWL, SETUP /INITIALIZE,
2044 1131 TAD C" &BIT7 NORMAL /INITIALIZE "CHAR" -
2045 3050 DCA CHAR /WITH 7 BIT CODE FOR SPACE,
2046 1130 TAD C-100 /COUNTER TO DISPLAY -
2047 3036 DCA TEMP2 /1 PASS OF A SWIRL,
2050 4775' DISSW1, JMS LOSWL /LOAD BUFFER WITH SWIRL PATTERN,
2051 4525 JMS I CSETTIM /SET TIMER,
DECIMAL
2052 0012 10*1 /TIME,
OCTAL
2053 4556 SR7 /MANUAL INCREMENT,
2054 5257 JMP ,+3 /NO, SELF TIMED,
2055 7344 DISSW2, CLA CLL CMA RAL /-2
2056 3046 DCA TIMEX /
2057 1132 TAD CBUFFER /STARTING ADDRESS OF BUFFER,
2060 4567 XDPGO /GO DISPLAY,
2061 4524 JMS I CBUMP /CHECK FOR MANUAL INCREMENT,
2062 5255 JMP DISSW2 /DISPLAY SAME,
2063 2050 ISZ CHAR /UPDATE CHARACTER,
2064 2036 ISZ TEMP2 /DONE YET?
2065 5250 JMP DISSW1 /NO,
2066 5271 JMP ,+3 /FORGET OTHER PATTERNS,
2067 4522 JMS I CB3CB2 /INCREMENT CONTROL BITS,
2070 5244 JMP DISSWL+1 /RELOAD,
2071 4575 UDPSH /STOP DISPLAY,
2072 4540 SR3 /REMAIN IN THIS TEST?
2073 5243 JMP DISSWL /YES,

```

/THIS TEST DISPLAYS A RIPPLE PATTERN WITH  
/EACH ROW OF 64 CHARACTERS HAVING THE CB2=CB3 CONTROL  
/BIT INCREMENTED,

```

2074 4566 ALL, SETUP /INIT
2075 4525 JMS I CSETTIM /SET TIME FOR DURATION OF TEST
2076 0144 10*10
2077 1121 TAD CBUFFER-1 /SET AUTO INDEX

```

```

2100 3010 DCA A110 /
2101 4557 SR6 /32 OR 647
2102 1374 TAD (10 /32
2103 1373 TAD (=20 /64
2104 3043 DCA COUNT2 /COUNT TO DETERMINE END OF SCREEN,
2105 1130 ALL1, TAD (=64 /NUMBER OF PATTERNS
/
2106 3044 DCA COUNT3 /SAVE
2107 1372 TAD (" 8BIT7 /START WITH 40
2110 3035 DCA TEMP1 /
2111 1035 ALL2, TAD TEMP1 /GET CODE FOR CHARACTER
2112 1036 TAD TEMP2 /ADD CONTROL CODE
2113 3410 DCA I A110 /STORE
2114 2035 ISZ TEMP1 /UPDATE FOR NEXT CHAR
2115 2044 ISZ COUNT3 /DONE 64 YET?
2116 5311 JMP ALL2 /NO
2117 2043 ISZ COUNT2 /DONE SCREEN?
2120 7410 SKP /NO
2121 5327 JMP ALL3 /YES,
2122 1036 TAD TEMP2 /GET CONTROL CODE
2123 1126 TAD (=20 /UPDATE FOR NEXT 64
2124 0371 AND (=700 /KEEP ONLY CONTROL BITS
2125 3036 DCA TEMP2 /SAVE
2126 5305 JMP ALL1 /DO ANOTHER 64
2127 1132 ALL3, TAD (=BUFFER /ADDRESS OF DATA
2130 4567 XDPGO /GO DISPLAY
2131 4575 UDPSM /STOP FOR EXIT
2132 4560 SR3 /REMAIN IN THIS TEST?
2133 5274 JMP ALL /YES,
/
2134 5770 NPAGE JMP I (=20087600 /GO TO NEXT PAGE

```

/ROUTINE TO INCREMENT 7 BIT CODE  
/IN THE ENTIRE BUFFER BY ONE,  
/EXIT +1 IF CODE 137,

```

2135 0000 INCBUF, OPEN /
2136 7300 CLA CLL /
2137 1767 TAD BUFFER /IS CODE 137 =
2140 0120 AND (=BIT7 /
2141 1366 TAD (=137 /BEEN DISPLAYED -
2142 7640 SEA CLA /YET?
2143 5350 JMP I +5 /NO,
2144 1131 TAD (" 8BIT7 NORMAL /
2145 3050 DCA CHAR /YES, 0 CHAR,
2146 2335 ISZ INCBUF /+1 FOR EXIT,
2147 5735 JMP I INCBUF /EXIT, AC=0,
2150 4557 SR6 /32 OR 64 CHARACTER?
/
2151 1117 TAD (=32+20 /32
2152 1116 TAD (=64+20 /64
/
2153 3035 DCA TEMP1 /
2154 1121 TAD (=BUFFER-1 /
2155 3010 DCA A110 /BUFFER-1=>A110

```

```

2156 2410 ISZ I A110 /INCREMENT CHARACTER
2157 2035 ISZ TEMP1 /DONE ENTIRE BUFFER
2160 5356 JMP I +2 /NO, REPEAT,
2161 5735 JMP I INCBUF /YES, EXIT, AC=0,

```

```

2166 7641
2167 4700
2170 2200
2171 0700
2172 0040
2173 7754
2174 0012
2175 3054
2176 3507
2177 0355
2200

```

PAGE

/TEST OF LINE FEED;  
/THE ENTIRE BUFFER IS FILLED WITH "CURSOR ?"  
/A VERTICAL COLUMN OF "BRIGHT \*" IS FORMED WITH "I"  
/ON THE LEFT AND LINE FEEDS ON THE RIGHT, WHICH SHOULD  
/PRODUCE A HORIZONTAL MOVEMENT OF THE VERTICAL COLUMN  
/OF "I" WITH NOTHING VISUAL TO THE RIGHT OF IT, A LINE  
/FEED FAILURE WILL LIKELY DISRUPT THE VERTICAL COLUMN  
/AND "CURSOR ?" WILL BE SEEN,

```

2200 4566 LFTST, SETUP /INITIALIZE,
2201 4557 SR6 /32 OR 64
/
2202 1131 TAD (=32 /32
2203 1130 TAD (=64 /64
/
2204 3040 DCA LENGTH /
2205 1377 TAD (" 8BIT7 CURSOR /FILL ENTIRE DISPLAY BUFFER
2206 4544 JMS I (=LDBUF /WITH "CURSOR ?",
2207 7201 CLA IAC /1
2210 3036 DCA I LFA00 /NUMBER TO DETERMINE POSITION,
2211 4525 LFTST1, JMS I (=SETTIM /SET TIMER,
/
2212 0005 S /TIME,
2213 1376 TAD (=20 /NUMBER OF LINES AS NEGATIVE
/
2214 1037 TAD LFA00 /ADD ACTIVE BUFFER
2215 3037 DCA LFA00 /AND SAVE FOR COUNTER,
2216 7346 CLA CLL OMA RTL /3
2217 1036 TAD LFA00 /
2220 7710 SPA CLA /READY FOR PERIODS YET?
2221 5236 JMP LFTST2 /NO, CHECK "I"
2222 1037 TAD LFA00 /LF BUFFER LENGTH=>TEMP1
2223 3035 DCA TEMP1 /COUNTER,
2224 1121 TAD (=BUFFER-1 /SET AUTO INDEX
2225 3010 DCA A110 /FOR LOADING "I"
2226 1375 TAD (" 8BIT7 NORMAL /ALWAYS LOAD ENTIRE LF BUFFER WITH

```



```

2227 3410 DCA I A110 /" "
2230 2035 ISZ TEMP1 /DONE YET?
2231 5226 JMP ,=3 /NO,
2232 7344 CLA CLL CHA RAL /#2
2233 1036 TAD LFAADD /
2234 7710 SPA CLA /READY FOR " " YET?
2235 5231 JMP LFTST3 /NO, GO LOAD THE LF'S,
2236 1115 LFTST2, TAD CBUFFER=2 /SET "ISTORE"
2237 3041 DCA ISTORE /FOR LOADING " "
          DECIMAL /
2240 1376 TAD (=20 /NUMBER OF LF TO LOAD,
          OCTAL /
2241 3035 DCA TEMP1 /COUNTER,
2242 1036 TAD LFAADD /ADD POSITION INDICATOR -
2243 1041 TAD ISTORE /AND MAKE AN ADDRESS,
2244 3041 DCA ISTORE /AND SAVE,
2245 1374 TAD ("&BIT7 BRIGHT /LOAD "BRIGHT "
2246 3441 DCA I ISTORE /LOAD,
2247 2035 ISZ TEMP1 /DONE?
2250 5242 JMP ,=6 /NO,
2251 1121 LFTST3, TAD CBUFFER=1 /SET "ISTORE"
2252 3041 DCA ISTORE /LOADING "LF",
          DECIMAL /
2253 1376 TAD (=20 /
          OCTAL /
2254 3035 DCA TEMP1 /COUNTER,
2255 1036 TAD LFAADD /ADD POSITION INDICATOR -
2256 1041 TAD ISTORE /AND MAKE AN ADDRESS,
2257 3041 DCA ISTORE /SAVE FOR INDIRECT,
2260 1373 TAD (LINEFD&BIT7 /7 BIT FOR LF,
2261 3441 DCA I ISTORE /PUT IN BUFFER,
2262 2035 ISZ TEMP1 /DONE?
2263 5255 JMP ,=6 /NO,
2264 4556 LFTST4, SR7 /MANUAL INCREMENT?
2265 5270 JMP ,+3 /NO, AUTO,
2266 7344 CLA CLL CHA RAL /#2 FOR TIME,
2267 3046 DCA TIMEX /SET TIMEX FOR MANUAL INCREMENT,
2270 1132 LFTST5, TAD CBUFFER /STARTING ADDRESS OF BUFFER,
2271 4567 XDPGO /GO DISPLAY,
2272 4524 JMS I CBUMP /CHECK MANUAL INCREMENT,
2273 5270 JMP LFTST5 /YES, REPEAT,
2274 4575 UDPSM /STOP DISPLAY,
2275 2036 ISZ LFAADD /UPDATE POSITION INDICATOR,
2276 2040 ISZ LENGTH /DONE YET?
2277 5211 JMP LFTST1 /NO,
2300 4560 SR3 /REMAIN IN THIS TEST?
2301 5200 JMP LFTST /YES,

```

/TEST TO CHECK "BBF" AND "EBF",  
 /FILL ENTIRE BUFFER WITH "<",  
 /SET "BBF" IN FIRST CHARACTER ("CURSOR ?") AND MOVE  
 /"EBF" WHICH WILL BE A BRIGHT E, BETWEEN THE "BBF" AND  
 /"EBF" THE CHARACTERS ARE "BRIGHT ?", MOVE "EBF" UNTIL ITS AT  
 /THE LOWER RIGHT AT WHICH TIME THE SCREEN SHOULD BE EMPTY,

/NOW MOVE "BBF" ONE POSITION PLACING A "B" ON THE LEFT  
 /OF IT AND ">'S" BEHIND IT, TEST IS COMPLETE WHEN THE SCREEN  
 /IS FULL OF ">'S",

```

2302 4566 FLDST, SETUP /INITIALIZE,
2303 4525 JMS I CSETTIM /SET TIMER FOR -
          DECIMAL /
2304 0001 1 /TIME,
          OCTAL /
2305 1372 TAD ("&BIT7 NORMAL /LOAD ENTIRE BUFFER WITH "<",
2306 4544 JMS I CLDBUFF /
2307 4557 SR6 /32 OR 64?
          DECIMAL /
2310 1117 TAD C32+20 /32
2311 1116 TAD C=64+20 /64
          OCTAL /
2312 3035 DCA TEMP1 /SAVE AS A COUNTER
2313 7001 IAC /
2314 1035 TAD TEMP1 /
2315 3036 DCA TEMP2 /TEMP2 = TEMP1 + 1
2316 1121 TAD CBUFFER=1 /
2317 3010 DCA A110 /BBF
2320 1132 TAD CBUFFER /
2321 3011 DCA A111 /INBETWEEN BBF+EBF
2322 1132 TAD CBUFFER /
2323 3012 DCA A112 /EBF
2324 1371 TAD CBUFFER=2 /
2325 3013 DCA A113 /2 CHARACTERS BEFORE BBF
2326 1121 TAD CBUFFER=1 /JUST BEFORE BBF
2327 3014 DCA A114 /
2330 1370 TAD ("&BIT7 CURSOR BBF /
2331 3410 DCA I A110 /LOAD FIRST BBF,
2332 5335 JMP ,+3 /DO NOT DO IN BETWEEN THE FIRST TIME,
2333 1367 FLDTS1, TAD ("&BIT7 BRIGHT /
2334 3411 DCA I A111 /LOAD IN BETWEEN
2335 1366 TAD ("&BIT7 BRIGHT EBF /
2336 3412 DCA I A112 /LOAD EBF
2337 1132 FLDTS4, TAD CBUFFER /
2340 4567 XDPGO /GO DISPLAY
2341 4524 JMS I CBUMP /CHECK MANUAL INCREMENT,
2342 5337 JMP FLDTS4 /AUTOMATIC
2343 2035 ISZ TEMP1 /EBF AT LOWER RIGHT YET?
2344 5333 JMP FLDTS1 /NO,
2345 4570 UDPSM /SOUND A TUNE DO TO END OF THIS TEST
2346 1370 FLDTS2, TAD ("&BIT7 CURSOR BBF /
2347 3410 DCA I A110 /LOAD BBF
2350 1365 TAD ("&BIT7 NORMAL /
2351 3413 DCA I A113 /POSITIONS 2 CHARACTER BEFORE BBF
2352 1364 TAD ("&BIT7 BRIGHT /
2353 3414 DCA I A114 /THE CHARACTER BEFORE BBF,
          NPAGE /
2354 5763 JMP I (=200&7600 /GO TO NEXT PAGE
2363 2400
2364 0502

```

2365 0076  
2366 1505  
2367 0477  
2370 2677  
2371 4676  
2372 0074  
2373 0012  
2374 0492  
2375 0056  
2376 7754  
2377 0677  
2400

PAGE

2400 1132 FLDT3: TAD CBUFFER /  
2401 4567 XDPGO /GO DISPLAY;  
2402 4524 JMS I CBUMP /CHECK MANUAL INCREMENT;  
2403 5200 JMP FLDT3 /AUTOMATIC  
2404 2036 ISZ TEMP2 /BFF AT LOWER RIGHT-1?  
2405 5777 JMP FLDT2 /NO;  
2406 4575 UDPSM /STOP THE DISPLAY;  
2407 4560 SR3 /REMAIN IN ENTIRE TEST?  
2410 5776 JMP FLDT1 /YES;

/TEST TO CHECK "EOS" (END OF SCREEN)  
/LOAD ENTIRE BUFFER WITH "BRIGHT ?"  
/START AT THE UPPER LEFT WITH EOS  
/WHICH WILL BE A "CURSOR ?" ON THE LEFT PUT  
/A "BRIGHT ?" AND BEHIND THE "X'S" STARTING AT THE  
/UPPER LEFT THE USER SHOULD SEE THE SCREEN BEING  
/FILLED WITH X'S WITH A "0" PRECEDING THEM, TEST  
/IS COMPLETE WHEN "EOS" IS AT THE BOTTOM RIGHT AND  
/THE SCREEN IS FULL OF X'S;

2411 4566 EOST1: SETUP /INITIALIZE;  
2412 4557 SR4 /32 OR 64?  
2413 1117 TAD C32:20 /32  
2414 1116 TAD C64:20 /64  
2415 3037 DCA TEMP3 /SAVE TO DETERMINE END OF SCREEN  
2416 4525 JMS I CSETTIM /SET THE TIMER FOR "  
2417 0001 DECIMAL /TIME  
2418 1 OCTAL  
2420 1375 TAD ("1&BIT7 BRIGHT /LOAD ENTIRE BUFFER -  
2421 4544 JMS I CLDBUFF /WITH "BRIGHT ?";  
2422 1121 TAD CBUFFER-1 /INIT, A110;  
2423 3010 DCA A110 /  
2424 1374 TAD CBUFFER-2 /INIT A111;  
2425 3011 DCA A111 /  
2426 1373 TAD CBUFFER-3 /INIT, A112;  
2427 3012 DCA A112 /  
2430 1372 EOSTS1: TAD ("1&BIT7 CURSOR EOS /LOAD EOS;  
2431 3410 DCA I A110 /  
2432 1371 TAD ("0&BIT7 BRIGHT /LOAD LAST VISIBLE CHARACTER;

2433 3411 DCA I A111 /  
2434 1370 TAD ("X&BIT7 NORMAL /LOAD A "NORMAL" X";  
2435 3412 DCA I A112 /  
2436 1132 EOSTS2: TAD CBUFFER /STARTING ADDRESS OF BUFFER  
2437 4567 XDPGO /GO, DISPLAY  
2440 4524 JMS I CBUMP /CHECK MANUAL INCREMENT  
2441 5236 JMP EOSTS2 /AUTOMATIC  
2442 2037 ISZ TEMP3 /FULL SCREEN COMPLETE?  
2443 5230 JMP EOSTS1 /NO;  
2444 4575 UDPSM /STOP THE DISPLAY;  
2445 4560 SR3 /REMAIN IN TEST?  
2446 5211 JMP EOST1 /YES;

/TEST FOR SHRINKAGE  
/PERIODS ARE PLACED IN EACH CORNER  
/OF THE SCREEN WITH "CURSOR SPACE" FILLING  
/THE REST OF THE SCREEN, THE PERIODS  
/SHOULD REMAIN RATHER MOTIONLESS AT ALL  
/TIMES;

2447 4566 SHRINK: SETUP /INITIALIZE;  
2448 1367 TAD (" &BIT7 CURSOR /SPACE WITH CURSOR SET;  
2449 4544 JMS I CLDBUFF /LOAD BUFFER  
2452 1366 TAD (" &BIT7 BRIGHT /PLACE A BRIGHT PERIOD  
2453 3532 DCA I CBUFFER /IN UPPER LEFT;  
2454 4557 SR4 /32 OR 64 CHART  
2455 5265 JMP SHR32 /32  
2456 1366 TAD (" &BIT7 BRIGHT /64  
2457 3765 DCA I CBUFFER+100-1 /PLACE "PERIOD" IN UPPER RIGHT  
2460 1366 TAD (" &BIT7 BRIGHT /  
2461 3764 DCA I C23+100+BUFFER /PLACE "PERIOD" IN LOWER LEFT  
2462 1366 TAD (" &BIT7 BRIGHT /  
2463 3763 DCA I C23+100+BUFFER+100-1 /PLACE "PERIOD" IN LOWER RIGHT;  
2464 5273 JMP SHRIN1 /  
2465 1366 SHR32: TAD (" &BIT7 BRIGHT /32 CHAR SETUP;  
2466 3762 DCA I CBUFFER+40-1 /PLACE "PERIOD" IN UPPER RIGHT  
2467 1366 TAD (" &BIT7 BRIGHT /  
2470 3761 DCA I C23+40+BUFFER /PLACE "PERIOD" IN LOWER LEFT;  
2471 1366 TAD (" &BIT7 BRIGHT /  
2472 3760 DCA I C23+40+BUFFER+40-1 /PLACE "PERIOD" IN LOWER RIGHT  
2473 4525 SHRIN1: JMS I CSETTIM /SET TEST TIMER FOR-  
2474 0144 DECIMAL /TIME  
2475 1132 OCTAL  
2476 4567 TAD CBUFFER /OBTAIN BUFFER STARTING ADDRESS  
2477 4575 XDPGO /GO DISPLAY  
2500 4560 UDPSM /STOP DISPLAY  
2501 5247 SR3 /REMAIN IN THIS TEST?  
2502 2031 JMP SHRINK /YES;  
2503 1431 TAD I WORKVT  
2504 7690 SNA CLA  
2505 5310 JMS \*3  
2506 4797 JMS DEVCOD  
2507 5796 JMP OSCHAR

```

2510 4755'      JMS      SRTSR8      /LOOK AT SR7 AND SR8
2511 5754      JMP I    (DSCHAR=1.  /
                                     /
                                     /REPORT END OF TEST,
                                     /
2512 4566      DONE1,  SETUP      /INITIALIZE,
2513 7604      LAR          /READ SR
2514 0353      AND          (1430  /LOOK FOR CERTAIN SW SET,
2515 7640      SZA CLA      /ANY SET?
2516 5526      JMP I    [BEGIN  /YES, DON'T REPORT END,
2517 4557      SR6         /
                                     /DECIMAL
2520 1117      TAO          [32+20 /32
2521 1116      TAO          [-64+20 /64
2522 3042      DCA          COUNT1 /
2523 4525      JMS I    [SETTIM  /
2524 0062      5+10        /
                                     /OCTAL
2525 1121      TAO          [BUFFER=1 /
2526 3011      DCA          AII1 /
2527 1352      DONE1,  TAO      [DONE=1 /
2530 3010      DCA          AII0 /
2531 1410      DONE2,  TAO I    AII0 /
2532 7450      SNA         /
2533 5327      JMP          DONE1 /
2534 3411      DCA I        AII1 /
2535 2042      ISZ          COUNT1 /BUFFER FILLED WITH "DONE"?
2536 5331      JMP          DONE2 /NO
2537 1132      TAO          [BUFFER /
2540 4567      XDPGO        /GO
2541 4575      UDPSM        /STOP
2542 5526      JMP I    [BEGIN  /

```

2552	4647
2553	1430
2554	2000
2555	3747
2556	2001
2557	4123
2560	6077
2561	6040
2562	4737
2563	7277
2564	7200
2565	4777
2566	0456
2567	0640
2570	05130
2571	0517
2572	3677
2573	4675
2574	4676
2575	0477

2576	2302
2577	2346
	2600

PAGE

```

/KEYBOARD TEST 1
/CARRIAGE RETURN ENDS TEST
/RBOUT IS ACTIVE
/TEST THAT KIE WILL DISABLE AND ENABLE KBRD INTERRUPTS;
/TEST THAT KRB CAN READ THE BUFFER AND THAT IT CAN CLEAR THE FLAG,

```

```

2600 4777' JMS INITDC
2601 4566 KEYTST, SETUP /INIT
2602 6030 KEY11, KCF /CLEAR FLAG
2603 6031 KEY11A, KSF /FLAG SET?
2604 5207 JMP ,*3 /NO, OK,
2605 4562 SR1A /HALT?
2606 7402 HLT /KBRD FLAG NO CLEAR OR KSF SKIPPED
/ON NO KBRD FLAG,
/LOOP?
2607 4561 SR2 /YES,
2610 5202 JMP KEY11 /FILL ENTIRE BUFFER WITH
2611 1376 TAD ("1&B17 CURSOR /"CURSOR ?" INCASE "EOS" FAILS;
2612 4544 CLOBUFF JMS I /STORING CHARACTERS,
2613 1121 TAD CBUFFER-1 /SET A110 FOR =
2614 3010 DCA A110 /STORING CHARACTERS,
2615 1132 TAD CBUFFER /SET A111 FOR =
2616 3011 DCA A111 /SETTING "EOS";
2617 1121 TAD CBUFFER-1 /SET A112 FOR =
2620 3012 DCA A112 /SETTING "CURSOR";
2621 1375 TAD ("X&B17 CURSOR EOS /
2622 3411 DCA I A111 /SET "EOS";
2623 1374 TAD ("1&B17 CURSOR /SET "CURSOR";
2624 3412 DCA I A112 /PUT IN THE BUFFER;
2625 4557 SR6 /32 OR 64?
/
2626 1117 TAD C32+20 /32
2627 1116 TAD C=64+20 /64
/
2630 3035 DCA TEMP1 /SAVE SO END OF SCREEN CAN BE DETERMINED;
2631 4357 JMS GO /GO DISPLAY
2632 6031 KEYV1, KSF /KBRD FLAG SET?
2633 5232 JMP ,*4 /NO, THEN WAIT?
2634 4773' KEYV3, JMS INTLD /SET UP FOR INTERRUPT RETURN
2635 2644 KEY4 /
2636 7201 CLA IAC /0001
2637 6035 KEYV10, KIE /ENABLE TTY INTERRUPT
2640 6001 ION /TURN INTERRUPT ON
2641 4563 SR1 /INTERRUPT SHOULD OCCUR DURING THIS JMS
2642 6002 IOF /NO INTERRUPT; TURN INT OFF,
2643 7402 HLT /WITH KBRD FLAG SET AND TTY INT ENABLED
/NO INTERRUPT OCCURRED;
/
2644 6002 KEY4, IOF /
2645 4561 SR2 /LOOP?
2646 5232 JMP KEY1 /YES,
2647 4773' KEYV5, JMS INTLD /SET UP RETURN FROM INTERRUPT,

```

```

2630 2656 KEY6 /
2651 6035 KEY11, KIE /DISABLE TTY INTERRUPT,
2652 6001 ION /INTERRUPT ON
2653 7000 NOP /GIVE INTERRUPT A CHANCE TO INT
2654 6002 IOF /TURN IT OFF
2655 5241 JMP ,+4 /INT ENABLE OK
2656 6002 KEY6, IOF /
2657 4542 SR1A /HALT?
2660 7402 HLT /KIE WITH AC=0 FAILED TO DISABLE TTY INT,
/OR AN EXTRA KEY WAS HIT ON ANOTHER DEVICE
/LOOP?

2661 4561 SR2 /
2662 5247 JMP KEYS /
2663 4773 KEY7, JMS INTLD /SET UP FOR INT
2664 2673 KEYS /
2665 7201 CLA IAC /0001
2666 6035 KEY112, KIE /ENABLE TTY INT
2667 6001 ION /TURN INT ON
2670 4563 SR1 /HALT
2671 6002 IOF /
2672 7402 HLT /KIE WITH AC11=1 FAILED TO ENABLE TTY INT,
2673 4561 SR2 /LOOP?
2674 5243 JMP KEY7 /YES,
2675 7240 KEY9, CLA CMA /7777
2676 6036 KEY116, KRB /CLEAR FLAG AND READ BUFFER
2677 0120 AND EB17 /7 BIT CODE
2700 7421 MQL /SAVE IN HQ
2701 6031 KEY113, KSF /FLAG SET?
2702 5305 JMP ,+3 /NO OK,
2703 4562 SR1A /HALT?
2704 7402 HLT /KRB FAILED TO CLEAR THE KRB FLAG,
2705 4561 SR2 /LOOP?
2706 5275 JMP KEY9 /
2707 7501 MQA /MQ ==> AQ,
2710 1372 TAD (=RUBOUT+200 /
2711 7650 SNA CLA /WAS IT A RUBOUT?
2712 5333 JMP RUB /YES,
2713 7501 MQA /
2714 1371 TAD (=CARRET+200 /
2715 7650 SNA CLA /WAS IT A CR?
2716 5327 JMP KEY2 /YES, END OF TEST,
2717 1375 TAD ("X&BIT7 CURSOR EOS /MOVE EOS
2720 3411 DCA I A111 /IN BUFFER,
2721 1374 TAD ("X&BIT7 CURSOR /SET "CURSOR",
2722 3412 DCA I A112 /
2723 7501 MQA /CHARACTER TO AC,
2724 3410 DCA I A110 /STORE TO BE DISPLAYED,
2725 2035 ISE TEMP1 /END OF BUFFER?
2726 5232 JMP KEY1 /NO,
2727 4560 KEY2, SR3 /YES, REMAIN IN THIS TEST?
2730 5201 JMP KEYST /YES,
2731 4575 UDPSM /STOP FOR EXIT,
2732 5770 RUB, JMP CURTST /GO TO NEXT TEST,
2733 1367 TAD (=BUFFER+1 /WILL RUBOUT MOVE CHARACTER -
2734 1010 TAD A110 /BEHIND BUFFER?
2735 7650 SNA CLA /

```

```

2736 5232 JMP KEY1 /YES,
2737 7240 CLA CMA /
2740 1010 TAD A110 /
2741 3010 DCA A110 /DECREMENT A110 FOR RUBOUT
2742 1010 TAD A110 /
2743 3012 DCA A112 /
2744 1374 TAD ("X&BIT7 CURSOR /
2745 3412 DCA I A112 /
2746 1012 TAD A112 /
2747 3011 DCA A111 /RESET A111
2750 1375 TAD ("X&BIT7 CURSOR EOS /
2751 3411 DCA I A111 /RESET EOS
2752 7240 CLA CMA /
2753 1035 TAD TEMP1 /RESET TEMP1
2754 3035 DCA TEMP1 /
2755 5232 JMP KEY1 /
2756 5770 NPAGE /GO TO NEXT PAGE
JMP I (=200&7600

```

/ROUTINE TO GO DISPLAY WITH OUT BACKGROUND  
/USED ONLY IN KRB TESTS,

```

2757 0000 GO, OPEN /
2760 1132 TAD CBUFFER /ADDRESS OF DATA
2761 4577 UDPLA /AC ==> STARTING ADDR REG
2762 6214 RDF /GET FIELD
2763 4576 UDPGO /GO DISPLAY
2764 5757 JMP I GO /EXIT,

```

```

2767 3101
2770 3000
2771 7663
2772 7601
2773 4163
2774 0641
2775 3780
2776 0677
2777 0355
3000

```

PAGE

/KEYBOARD TEST 2  
/TEST OF SPECIAL KEYS  
/TEST THAT KCF WILL CLEAR THE KRB FLAG,

```

3000 4566 CURTST, SETUP /INITIALIZE,
3001 1377 TAD ("X&BIT7 CURSOR /
3002 4544 JMS I CLOBUFF /LOAD BUFFER WITH CONTENTS OF AC,
3003 1376 TAD ("X&BIT7 CURSOR /
3004 3532 DCA I CBUFFER /
3005 1375 TAD ("X&BIT7 CURSOR EOS /
3006 3514 DCA I CBUFFER+1 /
3007 1132 TAD CBUFFER /

```

```

3010 4577 CUR1, UDPLA /AC ==> STARTING ADDR REG;
3011 6214 RDF /GET FIELD;
3012 4576 UDPSGO /GO DISPLAY;
3013 6031 KEY14, KSF /WAIT FOR KBRD FLAG;
3014 5213 JMP /WAIT
3015 6030 KEY15, KGF /CLEAR FLAG
3016 6031 KEY14, KSF /FLAG SET?
3017 5222 JMP /NO OK
3020 4562 SR1A /HALT?
3021 7402 HLT /KGF FAILED TO CLEAR THE KBRD FLAG
3022 4561 SR2 /LOOP?
3023 5215 JMP KEY15 /YES;
3024 6034 KEY115, KRS /READ BUFFER
3025 0120 AND /7 BIT
3026 3035 DCA TEMP1 /SAVE CHARACTER;
3027 1035 TAD TEMP1 /GET CHAR BACK
3030 1374 TAD (=CARRET+200) /CHECK FOR CARRIAGE RET;
3031 7650 SNA CLA /WAS IT ONE?
3032 5250 JMP CUR2 /YES, END OF TEST;
3033 1373 TAD (CURPOS-1) /ADDRESS-1 OF TABLE FOR CHECKING CODE;
3034 3010 DCA A110 /TO AUTO INDEX;
3035 1372 TAD (CURMSG-1) /FOR DETERMINING ADDRESS OF MSG;
3036 3042 DCA COUNT1 /
3037 1410 TAD I A110 /
3040 2042 ISZ COUNT1 /+1 TO ADDRESS OF MSG;
3041 7450 SNA /WAS THE CODE FOUND?
3042 5246 JMP /NO, DISPLAY "WHAT?";
3043 1035 TAD TEMP1 /GET CHARACTER CODE;
3044 7640 SNA CLA /DOES TABLE MATCH CHARACTER;
3045 5237 JMP /NO, TRY NEXT CODE IN TABLE;
3046 1442 TAD I COUNT1 /
3047 5210 JMP CUR1 /ADDRESS OF MSG IN AC;
3050 4575 CUR2, UDPSH /STOP
3051 4560 SR3 /REMAIN IN THIS TEST;
3052 5200 JMP CURTST /YES;
NPAGE /GO TO NEXT TEST;
3053 5771 JMP I (=20247600) /GO TO NEXT PAGE

```

/ROUTINE TO LOAD BUFFER WITH A SWIRL PATTERN;

```

3054 0000 LDSWL, OPEN /
3055 7300 CLA CLL /
3056 1050 TAD CHAR /GET CHARACTER
3057 3037 DCA TEMP3 /SAVE IT;
3060 1121 TAD (=BUFFER-1) /BUFFER-1 FOR USE IN-
3061 3010 DCA A110 /AUTO INDEX;
3062 1130 TAD (=64) /
3063 3370 DCA ("L /COUNTER FOR LINE LENGTH;
3064 4557 SR6 /32 OR 64 CHARACTER?
3065 5324 JMP SWL32 /32 CHAR
3066 1116 TAD (=64*20) /

```

```

3067 3367 OCTAL /
3070 1370 LDSWL1, DCA ("C /BUFFER SIZE COUNTER;
3071 3040 DCA ("L /
3072 1037 TAD TEMP4 /
3073 0120 AND /GET A NUMBER AND
3074 3035 DCA TEMP1 /MAKE A 7 BIT CHARACTER OUT OF IT;
3075 1047 LDSWL2, TAD CB2CB3 /SAVE CHARACTER
3076 1035 TAD TEMP1 /ADD CONTROL BITS TO -
3077 3410 DCA I A110 /CHARACTER AND -
3078 2367 ISZ ("C /STORE IN THE BUFFER;
3101 7610 SKP CLA /BUFFER FILLED?
3102 5654 JMP I LDSWL /NO;
3103 2035 ISZ TEMP1 /YES, EXIT;
3104 1035 TAD TEMP1 /INCREMENT FUTURE CHARACTER;
3105 1366 TAD (=140) /
3106 7640 SNA CLA /IF CODE 140 SET TO 40
3107 5312 JMP /
3110 1131 TAD (" 8BIT7 NORMAL /
3111 3035 DCA TEMP1 /
3112 2040 ISZ TEMP4 /LINE COMPLETE
3113 5275 JMP LDSWL2 /NO;
3114 2037 ISZ TEMP3 /YES, INCREMENT FUTURE CHARACTER;
3115 1037 TAD TEMP3 /
3116 1366 TAD (=140) /IF CODE 140 SET TO 40;
3117 7640 SNA CLA /IS IT 140?
3120 5270 JMP LDSWL1 /NO, OK;
3121 1131 TAD (" 8BIT7 NORMAL /RESET TEMP3 TO 40
3122 3037 DCA TEMP3 /STORE IT;
3123 5270 JMP LDSWL1 /
3124 1365 SWL32, TAD (=32 /SET LINE LENGTH FOR
3125 3370 DCA ("L /32 CHARACTERS;
3126 1117 TAD (=32*20 /
3127 5266 JMP LDSWL1-2 /
3165 7740 /
3166 7640 /
3167 0303 /
3170 0314 /
3171 3200 /
3172 4571 /
3173 4561 /
3174 7763 /
3175 3677 /
3176 0662 /
3177 0677 /
3200 /

```

PAGE

/KEYBOARD TEST 3.  
/TEST TO PRINT OUT THE LETTER AND OCTAL VALUE  
/AND POSITION OF SWITCH.

/TEST THAT CAF WILL ENABLE THE KBRD INTERRUPT  
/AND THAT IT CAN CLEAR THE FLAG,

```

3200 4566 OCT, SETUP /GENERAL INITIALIZE,
3201 4777 JMS INTLO /SET UP FOR INTERRUPT
3202 3222 /
3203 6035 KEY117, KIE /DISABLE INT
3204 6007 CAF /ENABLE INTERRUPT FOR KBRD
3205 1376 TAO ("3&BIT7 CURSOR /CURSOR CHARACTER,
3206 3532 DCA I [BUFFER /STORE
3207 1375 TAO ("1&BIT7 CURSOR EOS /
3208 3774 DCA I [BUFFER+1 /
3209 4773 JMS GO /DISPLAY
3210 6031 KSF OCT1, /FLAG FROM KBRD SET?
3211 5212 JMP ,+1 /NO, THEN WAIT FOR A CHARACTER,
3212 6034 KEY16, KRS /READ CHARACTER,
3213 7421 MQL /SAVE IN THE HQ,
3214 6001 ION /
3215 4563 SR1 /INTERRUPT SHOULD OCCUR DURING THIS JMS
3216 6002 IOF /
3217 7402 HLI /INITIALIZE FAILED TO SET KBRD INTERRUPT
3218 6002 /ENABLE,
3219 4561 OCT4, IOF /LOOP?
3220 5201 SR2 /YES,
3221 6007 JMP OCT+1 /INIT -CLEAR FLAG
3222 6031 KEY118, KSF /FLAG SET?
3223 5232 JMP ,+3 /NO, OK
3224 4563 SR1 /HALT?
3225 7402 HLI /INITIALIZE FAILED TO CLEAR KBRD FLAG,
3226 7000 NOP /MUST BE HERE ON ACCOUNT OF SCOPE LOOP,
3227 4561 SR2 /LOOP?
3228 5225 JMP OCT5 /YES,
3229 4773 JMS GO /RESTART DISPLAY
3230 7501 MQA /GET IT BACK
3231 0120 AND [BIT7 /MAKE IT A 7 BIT,
3232 3035 DCA TEMP1 /SAVE THE 7 BIT CODE,
3233 1035 TAO TEMP1 /GET THE CHARACTER JUST TYPED
3234 1372 TAO (=CARRET+200 /
3235 7650 SNA CLA /IS IT A CARRIAGE RETURN?
3236 5314 JMP OCT3 /YES IT WAS,
3237 4572 UDPMS /READ STATUS FOR SENSE SWITCH
3238 7104 CLL RAL /PUT IT IN THE LINK,
3239 7204 CLA RAL /BACK TO AC11
3240 1371 TAO ("0&BIT7 /EITHER 0 OR 1,
3241 3341 DCA MOCT2 /
3242 1035 TAO TEMP1 /GET CODE,
3243 7041 CMA IAC /NEGATE
3244 1131 TAO [40 /
3245 7540 SMA SZA /CODE LESS THAN 40?
3246 5262 JMP ,+4 /NO,
3247 1370 TAO ("&BIT7-40 /
3248 7700 SMA CLA /IS IT MORE THAN 137?
3249 5266 JMP ,+5 /NO, ITS IN THE RANGE OF 40-137
3250 7300 CLA CLL /0

```

```

3263 1367 TAO ("7&BIT7 BBF /BLANK FIRST PART BECAUSE OF NON-PRINT TYPE,
3264 3322 DCA MOCT /SET BBF
3265 5270 JMP ,+3 /GO SET OCTAL
3266 1035 TAO TEMP1 /GET 7 BIT CODE
3267 3322 DCA MOCT /STORE PRINTABLE CHARACTER,
3268 1366 TAO (=4 /
3269 3042 DCA COUNT1 /
3270 1365 TAO (MOCT1-1 /SET AUTO INDEX
3271 3010 DCA A110 /FOR STORING OCTAL
3272 7501 MQA /GET 7-8 BIT CODE,
3273 7104 CLL RAL /LEFT ONE ON ACCOUNT OF LINK
3274 7004 RAL /LEFT 1
3275 7006 RTL /LEFT 2; SHOULD POSITION FIRST TO STORE,
3276 7421 MQL /SAVE IN HQ,
3277 7501 MQA /MQ -> AC,
3278 0364 AND (7 /SAVE 9-11
3279 1371 TAO ("8-200 /ADD TO MAKE 7 BIT NUMBER,
3280 3410 DCA I A110 /PUT IN MESSAGE BUFFER,
3281 7501 MQA /
3282 2042 ISE COUNT1 /DONE ALL 4
3283 5276 JMP OCT2 /NO
3284 7300 CLA CLL /0
3285 1363 TAO (MOCT /ADDRESS OF MESSAGE
3286 4577 UDPLA /LOAD ADDRESS
3287 5212 JMP OCT1 /GO WAIT FOR ANOTHER CHARACTER,
3288 4560 SR3 /REMAIN IN THIS TEST?
3289 5200 JMP OCT /YES,
3290 5762 JMP MOREVT /MORE VT'S TO DO
3291 4761 JMS SR7SR8 /LOOK AT SR7 AND SR8 TO SEE
3292 5760 JMP KEYST=1 /IF WE REMAIN IN THIS SECTION,
3293 5526 JMP I [BEGIN /REMAIN IN KBRD SECTION
3294 5760 /DO TEST FROM BEGINNING,
3295 0077 MOCT, "7&BIT7
3296 0040 " &BIT7
3297 0275 "4&BIT7 BLINK
3298 1040 " &BIT7 EBF
3299 0077 MOCT1, "7&BIT7
3300 0077 "7&BIT7
3301 0077 "7&BIT7
3302 0077 "7&BIT7
3303 0040 " &BIT7
3304 0123 "5&BIT7
3305 0127 "4&BIT7
3306 0040 " &BIT7
3307 0275 "8&BIT7 BLINK
3308 0040 " &BIT7
3309 0077 MOCT2, "7&BIT7
3310 3330 "X&BIT7 BLINK EOS
3311 0000 /ROUTINE FOR LOOPING,
3312 0000 XLOOP, OPEN
3313 7200 CLA /MAKE SURE THE AC IS 0,

```

```

3345 1743 TAD I XLOOP /GET THE RETURN POINTER,
3346 3006 DCA SIX /SAVE THE POINTER,
3347 4561 SR2 /LOOP?
3350 5406 JMP I SIX /YES, LOOP ON TEST,
3351 2042 ISZ COUNT1 /DONE YET?
3352 5406 JMP I SIX /NO, DO TEST AGAIN,
3353 2343 ISZ XLOOP /+1 FOR RETURN
3354 4560 SR3 /REMAIN IN CURRENT TEST?
3355 5743 JMP I XLOOP /YES,
3356 2343 ISZ XLOOP /+1 TO EXIT TEST
3357 5743 JMP I XLOOP /EXIT CURRENT TEST,

```

```

3360 2600
3361 3747
3362 3547
3363 3322
3364 0007
3365 3325
3366 7774
3367 2077
3370 0077
3371 0060
3372 7763
3373 2757
3374 4701
3375 3677
3376 0663
3377 4163
3400

```

PAGE

/STARTS DISPLAY AND BACKGROUND JOB,

```

3400 0000 DPGOX, OPEN /LOAD STARTING ADDRESS OF DATA BUFFER,
3401 4577 UDPLA /IF "BLINK" IS SET THEN THE TIME
3402 4777 JMS I (CALTIM /MUST BE CHANGED,
/ALWAYS DISABLE CONSOLE TTY INTERRUPT,
/LOOK AT BACKGROUND SW,
/AND ONLY KEEP THAT ONE,
/IS BACKGROUND ALLOWED?
3403 6035 KEYI19, KIE /NO,
3404 7604 LAS /SET UP FOR INTERRUPT
3405 0113 AND C100 /RETURN ADDRESS,
3406 7640 SZA CLA /
3407 5265 JMP NOBACK /SET UP TO COUNT BACKGROUND PASSES
3410 4776 JMS INTLD /
3411 3426 XDPG01, DECIMAL /FOR NO INTERRUPT DETECTION,
/0001,
/GET FIELD,
/ALPHA-NUMERIC INT, ENABLED GO,
/RESTORE LINK,
/
3412 1145 TAD C-10 /
3413 3775 DCA BACKPA /
3414 7301 XDPG04, CLA CLL IAC /
3415 6214 RDF /
3416 4576 UDPGO /
3417 1055 FLSET, TAD LINK /
3420 7104 CLL RAL /
3421 1054 TAD HQ /

```

```

3422 7421 MQL /RESTORE HQ,
3423 1053 TAD AC /RESTORE AC,
3424 6001 ION /INTERRUPT ENABLED,
3425 5706 JMP I BACKST /START BACKGROUND
3426 4571 XDPG01, UDPLC /CLOCK FLAG SET?
3427 7402 HLT /NO, ILLEGAL INTERRUPT,
/NO SCOPE LOOPING PROVIDED,
/PROGRAM MUST BE RESTARTED,
/SAVE THE AC,
/
3430 3053 DCA AC /
3431 1145 TAD C-10 /SET UP TO COUNT BACKGROUND PASSES
3432 3775 DCA BACKPA /FOR NO INTERRUPT DETECTION,
3433 2045 ISZ TIME /DONE DISPLAYING REQUIRED TIME?
3434 5245 JMP XDPG02 /
3435 7604 LAS /LOOK AT LOOP SWITCH,
3436 7510 SPA /CHECK HALT SWITCH
3437 7402 HLT /SR0 WAS SET TO HALT PROGRAM,
3440 0112 AND C1000 /
3441 7650 SNA CLA /LOOP SWITCH SET?
3442 5254 JMP XDPG03 /NO
3443 1145 TAD C-10+1 /RESET TIME -
3444 3045 DCA TIME /COUNTER FOR 1 SECOND
3445 6201 XDPG02, CDF 00 /CHANGE TO DF 0
3446 1774 TAD I (ZERO /GET CONTENTS OF 0 OF FIELD 0 -
3447 3000 DCA ZERO /AND PLACE IT IN PROGRAM FIELD,
3450 4060 HOMEDF /CHANGE BACK TO PROGRAM DF,
3451 1053 TAD AC /RESTORE AC,
3452 6001 ION /INTERRUPT ON,
3453 5400 JMP I ZERO /CONTINUE BACKGROUND,
3454 7010 XDPG03, RAR /L --> AC0,
3455 3055 DCA LINK /SAVE THE LINK,
3456 6201 CDF 00 /GO TO DF 0
3457 1774 TAD I (ZERO /GET CONTENTS OF FIELD 0 -
3460 3306 DCA BACKST /AND SAVE IN PROGRAM FIELD,
3461 4060 HOMEDF /RESTORE DF,
3462 7701 ACL /LOAD HQ INTO AC,
3463 3054 DCA HQ /SAVE HQ,
3464 5600 JMP I DPGOX /EXIT, AC=L=0
3465 6214 NOBACK, RDF /GET FIELD,
3466 4576 UDPGO /ALPHA-NUMERIC INTERRUPT DISABLED,
3467 1373 NBACK2, TAD ("2 /OBTAIN LAST "AC" FROM THIS NON-BACK JOB,
3470 4571 UDPLC /CLOCK FLAG?
3471 5275 JMP /NO,
3472 2045 ISZ TIME /YES, DONE TIME?
3473 5270 JMP /NO, WAIT,
3474 5277 JMP /
3475 7001 IAC /INCREMENT THE AC SO THE PROGRAM LOOKS -
3476 5270 JMP /LIKE SOMETHING IS RUNNING,
3477 3373 DCA ("2 /SAVE THE "AC",
3500 4561 SR2 /LOOP?
3501 7610 SKP CLA /YES,
3502 5600 JMP I DPGOX /NO, EXIT, AC=0,

```

```

3503 1145      DECIMAL
              TAD C=10+1 /SET TIMER FOR
              OCTAL /
3504 3045      DCA TIME /1 SECOND;
3505 5267      JMP NBACK2 /
3506 4205      BACKST, ISETST /THIS LOCATION GETS MODIFIED;

/ROUTINE TO LOAD A BUFFER WITH
/AN INCREMENTING 7 BIT PATTERN;
/PLUS CONTROL BITS;

3507 0000      LDINC, OPEN
3510 7300      CLA CLL /
3511 4557      SR6 /32 OR 64 CHARACTER,
              DECIMAL /
3512 1117      TAD C32+20 /32;
3513 1116      TAD C=64+20 /64;
              OCTAL /
3514 3035      DCA TEMP1 /SAVE COUNT,
3515 1121      TAD CBUFFER=1 /BUFFER=1 FOR
3516 3010      DCA AI10 /AUTO-INDEX USE;
3517 1131      LDINC1, TAD C" 8BIT7 NORMAL /7-BIT CODE FOR SPACE;
3520 3050      DCA CHAR /0==> CHAR;
              DECIMAL /
3521 1130      TAD C=64 /
              OCTAL /
3522 3372      DCA ("C /COUNT;
3523 1050      LDINC2, TAD CHAR /CONTENTS OF CHAR AND
3524 0120      AND CB17 /MASK A 177 FOR A 7-BIT CODE;
3525 1047      TAD CB2CB3 /NOW ADD THE CONTROL BITS TO IT;
3526 3410      DCA I AI10 /STORE IN THE BUFFER;
3527 2050      ISZ CHAR /UPDATE FUTURE CHARACTER
3530 7000      NOP /
3531 2035      ISZ TEMP1 /COMPLETED BUFFER YET?
3532 7610      SKP CLA /NO
3533 5707      JMP I LDINC /YES, EXIT; AC=0;
3534 2372      ISZ ("C /
3535 5323      JMP LDINC2 /
3536 5317      JMP LDINC1 /

3537 7604      DVCOD, LAS
3540 3020      DCA IOTTAB
3541 3021      DCA IOTTAB+1
3542 1371      TAD (IOTTAB
3543 3031      DCA WORKVT
3544 4770      JMS DEVCOD
3545 7402      HLT
3546 5526      JMP I CBEGIN

3547 2031      MOREVT, ISZ WORKVT
3550 1431      TAD I WORKVT
3551 7650      SNA CLA
3552 5767      JMP OCT3+3

```

```

3553 4770      JMS DEVCOD
3554 5766      JMP KEYTST

```

```

3566 2601
3567 3317
3570 4123
3571 0020
3572 0303
3573 0332
3574 0000
3575 4364
3576 4163
3577 3720
3600

```

PAGE

/ROUTINE TO INCREMENT CB2+CB3

```

3600 0000      CB3CB2, OPEN
3601 7300      CLA CLL /
3602 1047      TAD CB2CB3 /CONTROL BITS =
3603 1377      TAD C=000 /2 AND 3 BEEN
3604 7640      SZA CLA /DISPLAYED YET?
3605 5211      JMP ,+4 /NO;
3606 2200      ISZ CB3CB2 /YES, +1 FOR EXIT;
3607 3047      DCA CB2CB3 /0==>CB2CB3;
3610 5600      JMP I CB3CB2 /EXIT, AC=0;
3611 1047      TAD CB2CB3 /ADD 200 =
3612 1126      TAD C200 /TO THE CONTROL -
3613 3047      DCA CB2CB3 /WORD;
3614 5600      JMP I CB3CB2 /EXIT, AC=0;

```

/ROUTINE TO SET THE AMOUNT OF TIME;

```

3615 0000      SETTIM, OPEN
3616 7300      CLA CLL /
3617 3046      DCA TIMEX /0==>TIMEX;
3620 1615      TAD I SETTIM /GET AMOUNT OF TIME (SECONDS)
3621 7041      CMA IAC /NEGATE SECONDS;
3622 3376      DCA ("A /SAVE IT;
3623 7604      LAS /LOOK FOR 90-00 CYCLE SW;
3624 0126      AND C200 /
3625 7640      SZA CLA /SKIP IF 00 CYCLE
3626 7001      IAC /1 (50 CYCLE);
3627 1375      TAD ("B /00 CYCLE
3630 3374      DCA ("B /SAVE AS COUNT REQUIRED FOR .1 SECOND;
3631 1046      TAD TIMEX /ADD ACCUMULATED TIME =
3632 1374      TAD ("B /PLUS .1 SECOND AND =
3633 3076      DCA TIMEX /SAVE;
3634 2376      ISZ ("A /TIMEX SET FOR AMOUNT OF TIME?
3635 5231      JMP ,+4 /NO;
3636 2215      ISZ SETTIM /+1 FOR EXIT;
3637 5615      JMP I SETTIM /EXIT, AC=0;

```



/SPECIAL ENTRANCE TO DISPLAY DESIRED CHARACTER;

```

3640 4566  SETCHR, SETUP
3641 7604      LAS
3642 0120      AND [BIT7
3643 3050      DCA CHAR
3644 7604      LAS
3645 0373      AND (3600
3646 3047      DCA CB2CB3
3647 7402      HLT
3650 5772      JMP DSCH11

```

/READ SWITCHES FOR CHARACTER  
/KEEP 7-BIT CODE  
/SAVE FOR THE CHARACTER;  
/READ SWITCHES FOR CONTROL BIT;  
/SAVE CONTROL BITS;  
/SET SR OPTIONS  
/GO DISPLAY

/ROUTINE TO PRIME THE VT8-E DATA BUFFER WITH  
/THE DATA THAT IS IN THE LOCATION DEFINED  
/AFTER THE CALL.

```

3651 0000  PRIME, OPEN
3652 4597      SR6
3653 1131      DECIMAL
3654 1371      TAD [32
3655 3044      TAD (-64+1
3656 1651      OCTAL
3657 4577      DCA COUNT3
3658 4575      TAD I PRIME
3659 4575      UDPLA
3660 4575      UDPSM
3661 4574      UDPMB
3662 2044      ISZ COUNT3
3663 5240      JMP ,=3
3664 2251      ISZ PRIME
3665 5651      JMP I PRIME

```

/32 OR 64?  
/  
/32  
/64  
/  
/  
/ADDRESS OF DATA  
/  
/MAINTENANCE MODE  
/SINGLE BREAK;  
/BUFFER FILLED =1?  
/NO,  
/+1 FOR EXIT;  
/EXIT, AC=0;

/ROUTINE TO DISPLAY THE RECEIVED AND  
/EXPECTED DATA IN THE AC AND MQ;

```

3666 0000  SHOWX, OPEN
3667 7300      CLA CLL
3668 1032      TAD GDATA
3669 7421      MQL
3670 1033      TAD BDATA
3671 7402      HLT
3672 7621      CAM
3673 1052      TAD FLD
3674 7421      MQL
3675 1034      TAD ADATA
3676 7402      HLT
3677 7300      CLA CLL
3678 5666      JMP I SHOWX

```

/0  
/EXPECTED DATA TO MQ  
/  
/RECEIVED DATA TO AC  
/MQ CONTAINS EXPECTED;  
/AC CONTAINS RECEIVED;  
/0 --> AC+MQ;  
/GET FIELD TESTED  
/SAVE IN THE MQ FOR VIEWING  
/  
/AC = ADDRESS OF DATA;  
/MQ = MEMORY FIELD;  
/0  
/EXIT, AC+L+MQ=0;

/ROUTINE TO LOAD THE DATA BUFFER  
/WITH THE DATA IN THE SR, THE

/DATA READ FROM THE DATA BUFFER  
/IS LOADED INTO THE MQ;

```

3703 4566  DP, SETUP
3704 7604      LAS
3705 3932      DCA I [BUFFER
3706 1132      TAD [BUFFER
3707 4577      UDPLA
3708 7301      CLA CLL IAC
3709 6214      ROF
3710 4575      UDPSM
3711 4575      UDPSM
3712 4574      UDPMB
3713 4573      UDPMO
3714 7421      MQL
3715 5304      JMP DP+1

```

/INITIALIZE;  
/READ DATA PATTERN  
/STORE;  
/BUFFER ADDRESS -->  
/STARTING ADDRESS REGISTER;  
/SET EXT, STARTING ADDRESS REGISTER;  
/FIELD;  
/  
/STARTING ADDR REGS --> ADDRESS COUNTER;  
/BREAK  
/READ DATA  
/STORE OUTPUT FROM BUFFER IN THE "MQ"  
/REPEAT UNTIL STOPPED;

/ROUTINE TO ADJUST "TIME" FOR SYNCHROIZATION WITH  
/THE BLINK FREQUENCY;  
/FOR EVERY 15 TICKS AN ADDITIONAL 1 IS ADDED;

```

3720 0000  CALTIM, OPEN
3721 1770      TAD BUFFER
3722 0367      AND (CURSOR
3723 1366      TAD (=BLINK
3724 7690      SNA CLA
3725 5330      JMP ,+3
3726 1046      TAD TIMEX
3727 5345      JMP CAL3
3728 1046      TAD CAL3
3729 7041      CMA IAC
3730 7421      MQL
3731 7501      HQA
3732 1365      DECIMAL
3733 1365      TAD (-15
3734 7510      OCTAL
3735 5343      SPA
3736 7521      JMP CAL2
3737 7001      SWP
3738 7521      IAC
3739 7521      SWP
3740 5334      JMP CAL1
3741 7721      CLASWP
3742 7041      CMA IAC
3743 3045      DCA TIME
3744 5720      JMP I CALTIM

```

/LOOK AT A CHARACTER IN THE BUFFER  
/KEEP ONLY THESE CONTROL BITS,  
/ADD =BLINK TO SEE IF  
/IF BLINK IS SET;  
/BLINK IS SET;  
/GET CALCULATED TIME COUNT  
/AND PUT IT IN TIME;  
/GET ORIGINAL TIME COUNT;  
/NEGATE IT  
/STORE IN MQ  
/MQ --> AC  
/  
/SUBTRACT 15;  
/  
/CAN WE SUBTRACT ANOTHER 15?  
/NO  
/AC --> MQ; MQ --> AC;  
/+1 ON TIME;  
/AC --> MQ; MQ --> AC;  
/GO DO ANOTHER SUBTRACTION;  
/MQ --> AC; 0 --> MQ;  
/NEGATE NEW TIME  
/SAVE;  
/EXIT;

/ROUTINE TO EXIT +1 IF SR7 AND SR8 ARE  
/NOT 0 AND 1 RESPECTIVELY;

```

3747 0000  SR7SR8, OPEN
3748 7604      LAS
3749 7102      CLL BSW
3750 7006      RTL
3751 7040      CMA

```

/LOOK AT SR  
/POSITION 7+8 FOR TESTING  
/GET INTO LINK AND AC0  
/CHANGE AC0

```

3754 7720 SNL SMA CLA /SR7=0 AND SR8=1;
3755 5747 JMP I SR7SR8 /YES;
3756 2347 ISR SR7SR8 /NO, THEN +1;
3757 5747 JMP I SR7SR8 /EXIT +1;

3765 7761
3766 7600
3767 0600
3770 4700
3771 7701
3772 2004
3773 3600
3774 0302
3775 7772
3776 0301
3777 7200
4000

```

PAGE

/IOT SUBROUTINES:

```

4000 0000 ZDPLA, OPEN
4001 6050 DDPLA, DPLA
4002 5600 JMP I ZDPLA
4003 7402 HLT /DPLA SKIPPED;
/NO SCOPE LOOP PROVIDED;
/SR SETTINGS DO NOT APPLY;

4004 0000 ZDPGO, OPEN
4005 6051 DDGO, DPGO
4006 5604 JMP I ZDPGO
4007 7402 HLT /DPGO SKIPPED;
/NO SCOPE LOOP PROVIDED;
/SR SETTINGS DO NOT APPLY;

4010 0000 ZDPSM, OPEN
4011 6052 DDPSM, DPSM
4012 5610 JMP I ZDPSM
4013 7402 HLT /DPSM SKIPPED;
/NO SCOPE LOOP PROVIDED;
/SR SETTINGS DO NOT APPLY;

4014 0000 ZDPMB, OPEN
4015 6053 DDPMB, DPMB
4016 5614 JMP I ZDPMB
4017 7402 HLT /DPMB SKIPPED;
/NO SCOPE LOOP PROVIDED;
/SR SETTINGS DO NOT APPLY;

4020 0000 ZDPMD, OPEN
4021 6054 DDPMO, DPMO
4022 5620 JMP I ZDPMD
4023 7402 HLT /DPMO SKIPPED;
/NO SCOPE LOOP PROVIDED;
/SR SETTINGS DO NOT APPLY;

```

```

4024 0000 ZDPMS, OPEN
4025 6055 DDPMO, DPMO
4026 5624 JMP I ZDPMS
4027 7402 HLT /DPMO SKIPPED;
/NO SCOPE LOOP PROVIDED;
/SR SETTINGS DO NOT APPLY;

4030 0000 ZDPCL, OPEN
4031 6056 DDPCG, DPCL
4032 5630 JMP I ZDPCL
4033 2230 ISR ZDPCL
4034 5630 JMP I ZDPCL /DPCL NO SKIP;
/DPCG SKIPPED;

4035 0000 ZDPBEL, OPEN
4036 6057 DDPBEL, DPBELL
4037 5635 JMP I ZDPBEL
4040 7402 HLT /DPBELL SKIPPED;
/NO SCOPE LOOP PROVIDED;
/SR SETTINGS DO NOT APPLY;

```

/ROUTINE TO EXIT +2 IF SR1=1.

```

4041 0000 SW1, OPEN
4042 7604 LAS
4043 7104 RAL CLL
4044 7700 SMA CLA
4045 5641 JMP I SW1
4046 2241 ISR SW1
4047 2241 ISR SW1
4050 5641 JMP I SW1

4051 0000 SW1A, OPEN
4052 7421 MQL
4053 7604 LAS
4054 7106 CLL RTL
4055 7200 CLA
4056 7501 MQA
4057 7420 SNL
4060 5651 JMP I SW1A
4061 2251 ISR SW1A
4062 5651 JMP I SW1A

```

/ROUTINE TO EXIT +1 IF SR2=0.

```

4063 0000 SW2, OPEN
4064 7604 LAS
4065 7510 SPA
4066 7402 HLT /SR0 WAS SET TO HALT THE PROGRAM;
4067 7106 RTL CLL

```

```

4070 7710 SPA CLA
4071 5663 JMP I SW2
4072 2263 ISZ SW2
4073 5663 JMP I SW2

/ROUTINE TO EXIT +1 IF SR3=0,

4074 0000 SW3, OPEN
4075 7604 LAS
4076 7106 RTL CLL
4077 7104 RAL CLL
4100 7710 SPA CLA
4101 5674 JMP I SW3
4102 2274 ISZ SW3
4103 5674 JMP I SW3

/ROUTINE TO EXIT+1 IF SR6=0,

4104 0000 SW6, OPEN
4105 7604 LAS
4106 7102 BSW CLL
4107 7710 SPA CLA
4110 5704 JMP I SW6
4111 2304 ISZ SW6
4112 5704 JMP I SW6

/ROUTINE TO EXIT +1 IF SR7=1,

4113 0000 SW7, OPEN
4114 7604 LAS
4115 7102 CLL BSW
4116 7104 CLL RAL
4117 7700 SMA CLA
4120 5713 JMP I SW7
4121 2313 ISZ SW7
4122 5713 JMP I SW7

/ROUTINE TO CHANGE IOT DEVICE CODE:
/AC 0=5 DISPLAY IOT
/AC 6=11 KEYBOARD IOT,

4123 0000 DEVCOD, 0
4124 4060 HOMEDF
4125 1431 TAD I WORKVT
4126 3035 DCA TEMP1
4127 1111 TAD DISIOT
4130 3036 DCA TEMP2
4131 1035 TAD TEMP1
4132 7012 RTR
4133 7010 RAR
4134 0110 AND C0770
4135 3037 DCA TEMP3
4136 4350 JMS DVCOM
4137 1107 TAD CKEYIOT
4140 3036 DCA TEMP2

/GET NEW DEVICE CODE FROM TABLE
/IOT CODE'S,
/SET STARTING ADDRESS OF DISPLAY IOT TABLE,
/SAVE AT TEMP2,
/OBTAIN NEW IOT,
/AND STORE AT TEMP3,
/
/
/PERFORM IOT SELECTION
/GET STARTING ADDRESS OF KEYBOARD IOT TABLE,
/AND SAVE

```

```

4141 1035 TAD TEMP1
4142 7006 RTL
4143 7004 RAL
4144 0110 AND C0770
4145 3037 DCA TEMP3
4146 4350 JMS DVCOM
4147 5723 JMP I DEVCOD
4150 0000 DVCOM, OPEN
4151 1436 TAD I TEMP2
4152 7450 SNA
4153 5750 JMP I DVCOM
4154 3040 DCA TEMP4
4155 1440 TAD I TEMP4
4156 0377 AND C7007
4157 1037 TAD TEMP3
4160 3440 DCA I TEMP4
4161 2036 ISZ TEMP2
4162 5351 JMP DVCOM+1

/REMOVE OLD CODE,
/INSERT NEW CODE,
/PUT BACK NEW ITO,
/SET UP FOR NEXT IOT CODE,

/ROUTINE TO LOAD ADDRESS "ONE + TWO" FOR INTERRUPTS

4163 0000 INTLD, OPEN
4164 1135 TAD C JMP I TWO
4165 3001 DCA ONE
4166 1763 TAD I INTLD
4167 3002 DCA TWO
4170 2363 ISZ INTLD
4171 6035 KEYI20, KIE
4172 5763 JMP I INTLD

```

```

4177 7007 PAGE
4200 4200
4205 0,5

/BACKGROUND PROCESSOR TEST!
/TESTS ARE RUN WHILE WAITING FOR INTERRUPT

/ISZ TEST!

4205 7040 ISZTST, CMA
4206 3355 DCA XTEMP5
4207 3392 DCA XTEMP2
4210 3351 DCA XTEMP1
4211 2351 ISZ XTEMP1
4212 2392 ISZ XTEMP2
4213 5211 JMP ,2
4214 1392 TAD XTEMP2
4215 7440 SZA
4216 7402 HLT

```

/ISZ FAILURE,

4217 7240 CLA CHA  
 4220 1351 TAD XTEMP1  
 4221 7440 SEA  
 4222 7402 HLT  
 4223 2355 ISZ XTEMP5  
 4224 7410 SKP  
 4225 5210 JMP ISZTST+3

/NO SCOPE LOOP PROVIDED  
 /SR SETTINGS DO NOT APPLY;

/ISZ FAILURE;  
 /NO SCOPE LOOP PROVIDED  
 /SR SETTINGS DO NOT APPLY;

/ROTATE 1 TEST1

4226 1352 ROT1TS, TAD XTEMP2  
 4227 7130 STL RAR  
 4230 7004 RAL  
 4231 7420 SNL  
 4232 7402 HLT

/ROTATE FAILURE;  
 /NO SCOPE LOOP PROVIDED  
 /SR SETTINGS DO NOT APPLY;

4233 7041 CMA IAC  
 4234 1352 TAD XTEMP2  
 4235 7440 SEA  
 4236 7402 HLT

/ROTATE FAILURE;  
 /NO SCOPE LOOP PROVIDED  
 /SR SETTINGS DO NOT APPLY;

4237 2352 ISZ XTEMP2  
 4240 5226 JMP ROT1TS

/ROTATE 2 TEST1

4241 1352 ROT2TS, TAD XTEMP2  
 4242 7100 CLL RTL  
 4243 7012 RTR  
 4244 7430 SEL  
 4245 7402 HLT

/ROTATE FAILURE;  
 /NO SCOPE LOOP PROVIDED  
 /SR SETTINGS DO NOT APPLY;

4246 7041 CMA IAC  
 4247 1352 TAD XTEMP2  
 4250 7440 SEA  
 4251 7402 HLT

/ROTATE FAILURE;  
 /NO SCOPE LOOP PROVIDED  
 /SR SETTINGS DO NOT APPLY;

4252 2352 ISZ XTEMP2  
 4253 5241 JMP ROT2TS

/TAD TEST ADD EVERY COM TO RAN NO.

4254 3353 TADTST, DCA XTEMP3  
 4255 1356 TAD PRAN1  
 4256 7104 CLL RAL  
 4257 7430 SEL  
 4260 7001 IAC

4261 3356 DCA PRAN1  
 4262 1357 TAD PRAN2  
 4263 1356 TAD PRAN1  
 4264 3357 DCA PRAN2  
 4265 1357 TAD PRAN2  
 4266 3354 DCA XTEMP4  
 4267 1357 TAD PRAN2  
 4270 1353 TAD XTEMP3  
 4271 7041 CMA IAC  
 4272 1354 TAD XTEMP4  
 4273 7440 SEA  
 4274 7402 HLT

/TAD FAILURE;  
 /NO SCOPE LOOP PROVIDED  
 /SR SETTINGS DO NOT APPLY;

4275 2354 ISZ XTEMP4  
 4276 7000 NOP  
 4277 2353 ISZ XTEMP3  
 4300 5267 JMP I=11

/JMS TEST MAKE 13 PASSES OF 128 CONSECUTIVE JMS,  
 /AND COMPARE RESULTS.

4301 1363 JMSTST, TAD K7763X  
 4302 3351 DCA XTEMP1  
 4303 1362 TAD K7600X  
 4304 3352 DCA XTEMP2  
 4305 1360 TAD JMSLOC  
 4306 3353 DCA XTEMP3  
 4307 1361 TAD JMSKON  
 4310 3354 DCA XTEMP4  
 4311 1377 TAD (JHRETU  
 4312 3056 DCA RETUJM  
 4313 1354 TAD XTEMP4  
 4314 3753 DCA I XTEMP3  
 4315 2354 ISZ XTEMP4  
 4316 2353 ISZ XTEMP3  
 4317 2352 ISZ XTEMP2  
 4320 5313 JMP I=5  
 4321 1106 TAD (JMP I RETUJM  
 4322 3753 DCA I XTEMP3  
 4323 5760 JMP I JMSLOC

4324 1362 JMRETU, TAD K7600X  
 4325 3352 DCA XTEMP2  
 4326 1360 TAD JMSLOC  
 4327 3353 DCA XTEMP3  
 4330 1353 TAD XTEMP3  
 4331 7040 CMA  
 4332 1753 TAD I XTEMP3  
 4333 7440 SEA  
 4334 7402 HLT

/JMS FAILURE;  
 /NO SCOPE LOOP PROVIDED  
 /SR SETTINGS DO NOT APPLY;

4335 2353 ISZ XTEMP3  
 4336 2352 ISZ XTEMP2

```

4337 5330 JMP JMRETU+4
4340 2351 ISZ XTEMP1
4341 5303 JMP JMSTST+2
4342 7501 MQA
4343 7001 IAC
4344 7421 MQL
4345 2364 ISZ BACKPA
4346 5205 JMP ISETST
4347 7402 HLT

/NO INTERRUPT OCCURRED WITHIN 10
/PASSES OF THE BACKGROUND JOB,
/CLOCK RUNNING SLOW OR INTERRUPT
/FAILING,
/NO SCOPE LOOP PROVIDED
/SR SETTINGS DO NOT APPLY,
/START TEST FROM BEGINNING,

4350 5776 JMP BEGIN

4351 0000 XTEMP1, OPEN
4352 0000 XTEMP2, OPEN
4353 0000 XTEMP3, OPEN
4354 0000 XTEMP4, OPEN
4355 0000 XTEMP5, OPEN
4356 4263 PRAN1, 4263
4357 2634 PRAN2, 2634

4360 7400 JMSLOC, BWA
4361 4200 JMSKON, 4200
4362 7600 K7600X, 7600
4363 7763 K7763X, 7763
4364 0000 BACKPA, OPEN

/BACKGROUND PASS COUNT

```

4376 0200  
4377 4324  
4400 4400

PAGE

/ROUTINE TO WAIT THE DURATION OF 2 "ISZ" LOOPS;

```

4400 0000 WAIT, OPEN
4401 3252 DCA LDA /
4402 2252 ISZ LDA /
4403 5202 JMP ,=1 /
4404 2252 ISZ LDA /
4405 5204 JMP ,=1 /
4406 5600 JMP I WAIT /

```

/ROUTINE TO LOAD BUFFER WITH THE  
/CONTENTS OF THE AC.

```

4407 0000 LOBUFF, OPEN
4410 3252 DCA LDA /SAVE THE CHARACTER,
4411 1121 TAD CBUFFER-1 /BUFFER-1=>A110,
4412 3010 DCA A110 /
4413 4557 SR6 /32 OR 64 CHARACTER?

```

```

4414 1117 DECIMAL /
4415 1116 TAD [32+20 /32,
TAD [64+20 /64,
OCIAL /
4416 3302 DCA LDB /
4417 1292 TAD LDA /CHARACTER BUFFER,
4420 3410 DCA I A110 /
4421 2302 ISZ LDB /
4422 5217 JMP ,=3 /NO,
4423 5607 JMP I LOBUFF /YES, EXIT;

```

/THIS ROUTINE SAVES THE CONTENTS OF THE ADDRESS  
/SPECIFIED BY LOCATION "ADATA" IN THE HQ, THE ADDRESS  
/IS THEN LOADED WITH THE DATA IN LOCATION "GDATA"  
/EITHER 32 OR 64 BREAKS ARE THEN PERFORMED DEPENDING ON THE  
/SETTING OF THE SR, THE ORIGINAL CONTENTS OF THE ADDRESS ARE  
/RESTORED BEFORE EXIT,

```

4424 0000 LOADDA, OPEN
4425 2224 ISZ LOADDA /*1 FOR EXIT,
4426 1042 TAD COUNT1 /GET THE NUMBER OF BREAKS
4427 3242 DCA COUNT1 /SAVE THIS NUMBER,
4430 1057 TAD IDPSH /GET EITHER A "NOP" OR "DPSH"
4431 3242 DCA LDAI /STORE INSTRUCTION
4432 1052 TAD FLD /GET FIELD
4433 1155 TAD CCDF 00 /CDF 0 INST,
4434 3235 DCA /STORE CDF INST,
4435 0000 OPEN ,*1 /EXECUTE CDF XX
4436 1434 TAD I ADATA /GET THE INFORMATION IN THE ADDRESS,
4437 7421 MQL /SAVE IT,
4440 1037 TAD INDATA /GET TEST DATA
4441 3434 DCA I ADATA /PUT TEST DATA IN THE ADDRESS,
4442 0000 LOAD1, OPEN /WILL CONTAIN A "NOP" OR "DPSH"
4443 6053 IS, DPMB /DO A SINGLE BREAK,
4444 2252 ISZ LDA /ENOUGH BREAKS TO FILL BUFFER?
4445 5242 JMP ,=3 /NO, DO SOME MORE,
4446 7501 MQA /RESTORE THE ADDRESS WITH -
4447 3434 DCA I ADATA /ITS ORIGINAL CONTENTS,
4450 4060 HOMEDF /BACK TO PROGRAM DF,
4451 5624 JMP I LOADDA /EXIT,
4452 0000 LDA, OPEN /COUNT LOCATION,

4453 0000 LOADDB, OPEN
4454 1042 TAD COUNT1 /GET THE NUMBER OF BREAKS
4455 3302 DCA LDB /SAVE THIS NUMBER,
4456 1057 TAD IDPSH /GET INSTRUCTION
4457 3272 DCA LDB1 /SAVE
4460 1034 TAD ADATA /
4461 3303 DCA LDB1 /
4462 1052 TAD FLD /FIELD
4463 1155 TAD CCDF 00 /CDF 0 INST,
4464 3245 DCA ,*1 /STORE CDF INSTRUCTION,
4465 0000 OPEN /EXECUTE CDF XX INSTRUCTION,
4466 1703 TAD I LDB1 /GET THE INFORMATION IN THE ADDRESS,
4467 7421 MQL /SAVE IT,

```

```

4470 1037      TAD      INDATA      /GET TEST DATA
4471 3703      DCA I   LDB1      /PUT TEST DATA IN THE ADDRESS,
4472 0000      LDB1,  OPEN      /WILL CONTAIN A "NOP" OR "DPSM"
4473 0053      I3,    DPHB      /DO A SINGLE BREAK,
4474 2302      ISZ     LDB      /ENOUGH BREAKS TO FILL BUFFER?
4475 5272      JMP     ,=3      /NO, DO SOME MORE,
4476 7501      MQA     DCA I   LDB1 /RESTORE THE ADDRESS WITH =
4477 3703      DCA I   LDB1      /ITS ORIGINAL CONTENTS,
4480 4060      HOMEDF  JMP I   LDADDB /RESTORE PROGRAM DF,
4481 5653      LDB,   OPEN      /EXIT,
4482 0000      LDB1,  OPEN      /COUNT LOCATION,
4483 0000

4504 4001      DISIOY, DDPLA
4505 4005      DDPGO
4506 4011      DDPDM
4507 4015      DDPMB
4510 4021      DDPMD
4511 4025      DDPMS
4512 4031      DDPCL
4513 4036      DDPBEL
4514 0402      CLOCK1
4515 0404      CLOCK2
4516 1430      I1
4517 4473      I3
4520 4443      I5
4521 0000      0
4522 2602      KEY10Y, KEY11
4523 2603      KEY11A
4524 2632      KEY1
4525 3013      KEY14
4526 3015      KEY15
4527 3214      KEY16
4530 0003      KEY17
4531 0015      KEY18
4532 0030      KEY19
4533 2637      KEY110
4534 2651      KEY111
4535 2666      KEY112
4536 2701      KEY113
4537 3016      KEY114
4540 3024      KEY115
4541 2676      KEY116
4542 3203      KEY117
4543 3226      KEY118
4544 3212      OCT1
4545 3403      KEY119
4546 4171      KEY120
4547 0000      0

/ DATA FOR CONSTANT DATA TEST:

4550 7767      FDATA8, -FDATA8+FDATA8
4551 7777      7777
4552 0000      0000

```

```

4553 7777      7777
4554 5252      5252
4555 2525      2525
4556 7700      7700
4557 0077      0077
4560 7007      7007
4561 0770      FDATA8, 0770

4562 7770      CURPOS, -CURLFT
4563 7750      -CURRHT
4564 7746      -CURUP
4565 7765      -CURDOWN
4566 7743      -CURHME
4567 7742      -EOL
4570 7741      -EOS
4571 0000      0

4572 4602      CURMSG, MLEFT
4573 4607      MRIGHT
4574 4615      MUP
4575 4620      MDOWN
4576 4625      MHOME
4577 4636      MEDL
4600 4632      MEDS
4601 4642      MWHAT

4602 0114      MLEFT, "L&BIT7/"E&BIT7/"F&BIT7/"T&BIT7/"?&BIT7 BLINK EOS
4603 0105
4604 0106
4605 0124
4606 3277
4607 0122      MRIGHT, "R&BIT7/"I&BIT7/"G&BIT7/"H&BIT7/"T&BIT7/"?&BIT7 BLINK EOS
4610 0111
4611 0107
4612 0110
4613 0124
4614 3277
4615 0125      MUP, "U&BIT7/"P&BIT7/"?&BIT7 BLINK EOS
4616 0120
4617 3277
4620 0104      MDOWN, "D&BIT7/"Q&BIT7/"W&BIT7/"N&BIT7/"?&BIT7 BLINK EOS
4621 0117
4622 0127
4623 0116
4624 3277
4625 0110      MHOME, "H&BIT7/"O&BIT7/"M&BIT7/"E&BIT7/"?&BIT7 BLINK EOS
4626 0117
4627 0115
4630 0105
4631 3277
4632 0105      MEOS, "E&BIT7/"Q&BIT7/"S&BIT7/"?&BIT7 BLINK EOS
4633 0117
4634 0123
4635 3277
4636 0105      MEOL, "E&BIT7/"O&BIT7/"L&BIT7/"?&BIT7 BLINK EOS

```

```

4637 0117
4640 0114
4641 3277
4642 0127 MHWAT, "H&BIT7;"H&BIT7;"A&BIT7;"T&BIT7;"7&BIT7 BLINK;"7&BIT7 BLINK EQS
4643 0110
4644 0101
4645 0124
4646 0277
4647 3277
4650 0304 MDONE, "D&BIT7 BLINK;"0&BIT7 BLINK;"N&BIT7 BLINK;"E&BIT7 BLINK;"0
4651 0317
4652 0316
4653 0305
4654 0000

```

/DISPLAY BUFFER USES LOCATIONS 4670-7377;  
 /BACKGROUND WORKING AREA USES LOCATIONS 7400-7577;

4670 \*4670  
 XLIST

\$

```

0106 5456
0107 4522
0110 0770
0111 4504
0112 1000
0113 0100
0114 4701
0115 4676
0116 5400
0117 1200
0120 0177
0121 4677
0122 3600
0123 2135
0124 1051
0125 3615
0126 0200
0127 0061
0130 7700
0131 0040
0132 4700
0133 3506
0134 4205
0135 5402
0136 0020
0137 3651
0140 0032
0141 0010
0142 1777

```

```

0143 7770
0144 4407
0145 7766
0146 7400
0147 1470
0150 3640
0151 3703
0152 2600
0153 2000
0154 3537
0155 6201
0156 4113
0157 4104
0160 4074
0161 4063
0162 4051
0163 4041
0164 3343
0165 3666
0166 1073
0167 3400
0170 4035
0171 4030
0172 4024
0173 4020
0174 4014
0175 4010
0176 4004
0177 4000

```

[illegible][illegible]



AC	0053	CAM	7621	DDPMB	4015	EXTIN3	0635
AQL	7701	CARRET	0215	DDPMD	4021	F0ATAB	4550
ADATA	0034	CB2CB3	0047	DDPMS	4025	F0ATAE	4561
ADDR	0034	CB3CB2	3600	DDPSH	4031	FILBUF	0040
ADDR1	1400	CDP	0201	DEVCOD	4123	FIVE	0005
ADDR1A	1413	CDPCL	0303	DISIOT	4504	FLO	0052
ADDR1B	1423	CDPCL1	0321	DISRI1	2031	FLOTS1	2333
ADDR1C	1466	CDPG0	0336	DISRI2	2030	FLOTS2	2346
ADDR1D	1444	CDPLA	0257	DISSH1	2050	FLOTS3	2400
AI10	0010	CDPSM	0271	DISSH2	2055	FLOTS4	2337
AI11	0011	CHAR	0050	DISSHL	2043	FLOTST	2302
AI12	0012	CKCC	0201	DONE	2542	FLSET	3417
AI13	0013	CKRB	0213	DONE1	2527	FOUR	0004
AI14	0014	CKRS	0226	DONE2	2531	FUTURE	7000
AI15	0015	CLASWP	7721	DP	3703	GDATA	0032
AI16	0016	CLOCK	0400	DPBELL	6057	GO	2757
AI17	0017	CLOCK1	0402	DPCL	6056	GTF	6004
ALL	2074	CLOCK2	0404	DPG0	6051	HLT	7402
ALL1	2105	CLOCK3	0412	DPQOX	3400	HOME	0060
ALL2	2111	COUNT1	0042	DPLA	6050	HOMEDF	4060
ALL3	2127	COUNT2	0043	DPH8	6053	I1	1430
ALTM0D	0375	COUNT3	0044	DPH0	6054	I3	4473
BACKPA	4364	CUR1	3010	DPHS	6055	I5	4443
BACKST	3506	CUR2	3050	DPSH	6052	IDPSH	0057
BBF	2000	CURDWN	0013	DSCHA1	2004	INCBUF	2135
B0DATA	0033	CURHME	0035	DSCHA2	2007	INDATA	0037
BEGIN	0200	CURLFT	0010	DSCHA3	2043	INITOC	0355
BIT7	0177	CURMSG	4572	DSCHAR	2001	INT1	0425
BLINK	0200	CURPOS	4562	DVGOO	3537	INT1A	0432
BRIGHT	0400	CURRHT	0030	DVGOM	4150	INT1B	0440
BSW	7002	CURSOR	0600	E0S	0037	INT1C	0443
BUFCA	1516	CURTST	3000	EBF	1000	INT1D	0446
BUFFER	4700	CURUP	0032	EDPG0	0712	INT2	0451
BUMP	1091	DATA1	1000	EDPG01	0716	INT2A	0456
BUMP1	1070	DATA1A	1014	EDPG02	0741	INT2B	0466
BWA	7400	DATA1B	1024	EOL	0036	INT2C	0472
CAF	6007	DATA2	1200	EOS	3000	INT3	0475
CAINC	1600	DATA2A	1214	EOSTS1	2430	INT3A	0502
CAINC1	1613	DATA2B	1237	EOSTS2	2436	INT3B	0511
CAINC2	1625	DATA2C	1270	EOSTS3	2411	INT3C	0513
CAINC3	1632	DATA2D	1276	EXT	0646	INTLO	4163
CAINC4	1634	DATA2E	1301	EXT1	0647	IOF	0002
CAINC5	1661	DATA2F	1311	EXT2	0653	ION	6001
CAINC6	1671	DATA2G	1317	EXT3	0657	IOTTAB	0020
CAINC7	1677	DATA2H	1331	EXT4	0677	ISTORE	0041
CAINC8	1725	DATA2I	1333	EXTA	0517	ISZTST	4205
CAINC9	1727	DATA2J	1344	EXTA2	0522	JMRETU	4324
CAINCA	1740	DATA2K	1352	EXTA3	0524	JMSKON	4361
CALL1	3734	DDPBEL	4036	EXTADR	0032	JMSLOC	4360
CALL2	3743	DDPCL	4031	EXTIN	0600	JMSTST	4301
CALL3	3745	DDPG0	4005	EXTIN1	0601	K7600X	4362
CALTIM	3720	DDPLA	4001	EXTIN2	0606	K7763X	4363

KCC	6032	LFADD	0036	RMF	6244	UDPHS	4572
KCF	6030	LFEND	0037	ROT1TS	4226	UDPSH	4575
KEY1	2632	LFTST	2200	ROT2TS	4241	UPDOWN	0043
KEY2	2727	LFTST1	2211	RTF	6005	WAIT	4400
KEY3	2634	LFTST2	2236	RUB	2733	WORKVT	0031
KEY4	2644	LFTST3	2251	RUBOUT	0377	XDPG0	4567
KEY5	2647	LFTST4	2264	SENSE	0244	XDPG01	3426
KEY6	2656	LFTST5	2270	SETCHR	3640	XDPG02	3445
KEY7	2663	LINEFD	0212	SETTI	3630	XDPG03	3454
KEY8	2673	LINK	0055	SETTIM	3615	XDPG04	3414
KEY9	2675	LOOP	4564	SETUP	4566	XLOOP	3343
KEY11	2602	LOOPFL	0036	SETUPX	1073	XTEMP1	4351
KEY110	2637	MDONE	4650	SLVEN	0007	XTEMP2	4352
KEY111	2651	MDOWN	4620	SHOW	4565	XTEMP3	4353
KEY112	2666	MEOL	4636	SHOWX	3666	XTEMP4	4354
KEY113	2701	MEOS	4632	SHR32	2465	XTEMP5	4355
KEY114	3016	MHOME	4625	SHRIN1	2473	ZDPBEL	4035
KEY115	3024	MLEFT	4602	SHRINK	2447	ZDPCL	4030
KEY116	2676	MOCT	3322	SIX	0006	ZDPG0	4004
KEY117	3203	MOCT1	3326	SKON	6000	ZDPLA	4000
KEY118	3226	MOCT2	3341	SR1	4563	ZDPMB	4014
KEY119	3403	MOREVT	3547	SR1A	4562	ZDPHD	4020
KEY11A	2603	MQ	0054	SR2	4561	ZDPMS	4024
KEY120	4171	MQA	7501	SR3	4560	ZDPSH	4010
KEY14	3013	MQL	7421	SR6	4557	ZERO	0000
KEY15	3015	MRIGHT	4607	SR7	4556		
KEY16	3214	MULTAB	1517	SR7SR8	3747		
KEY17	0203	MULV1	1502	SRQ	6003		
KEY18	0215	MULVT8	1470	SW1	4041		
KEY19	0230	MUP	4615	SW1A	4051		
KEY1OT	4522	MHHAT	4642	SW2	4063		
KEYTST	2601	NBACK2	3467	SW3	4074		
KIE	6035	NBACK	3465	SW6	4104		
KRB	6036	NOCNT	0035	SW7	4113		
KRS	6034	NORMAL	0000	SWL32	3124		
KSF	6031	OCT	3200	SWP	7521		
LAS	7604	OCT1	3212	TADTST	4254		
LOA	4452	OCT2	3276	TEMP1	0035		
LOADADA	4424	OCT3	3314	TEMP2	0036		
LOADDB	4453	OCT4	3222	TEMP3	0037		
LOAI	4442	OCT5	3225	TEMP4	0040		
LDB	4502	ONE	0001	TEMP5	0041		
LDB1	4503	OPEN	0000	THREE	0003		
LDB1	4472	OSR	7404	TIME	0045		
LOBUFF	4407	PRAN1	4356	TINEX	0046		
LOINC	3507	PRAN2	4357	TWO	0002		
LOINC1	3517	PRIME	3651	UDPBEL	4570		
LOINC2	3523	PSET	0051	UDPCL	4571		
LDSWL	3054	RDF	6214	UDPG0	4576		
LDSWL1	3070	RETUJH	0056	UDPLA	4577		
LDSWL2	3075	RIB	6234	UDPHB	4574		
LENGTH	0040	RIF	6224	UDPHD	4573		

ERRORS DETECTED: 0

LINKS GENERATED: 53

RUN-TIME: 24 SECONDS

3K CORE USED