

VTV30,VT30-H

VTV30J/H VT30H DSPLY
CVVTBA0

AH F657A MC

COPYRIGHT 1979

FICHE 1 OF 1

NOV 1979

digital

MADE IN USA

1
2
3
4
5

000000

.REPT 0

IDENTIFICATION

PRODUCT CODE: AC-F655A-MC
PRDUCT NAME: CVVTBA0 VTV30J/M-VT30M DSPLY
PRODUCT DATE: OCT 1, 1974
MAINTAINER: COMPUTER SPECIAL SYSTEMS
DIGITAL EQUIPMENT CO. LTD.
READING
BERKS U.K.

7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

COPYRIGHT (C) 1979 BY
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE
USED AND COPIED ONLY IN ACCORDANCE WITH THE TERMS OF
SUCH LICENSE AND WITH THE INCLUSION OF THE ABOVE
COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER COPIES
THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE
TO ANY OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE
SOFTWARE IS HEREBY TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE
WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A
COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR
RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT
SUPPLIED BY DIGITAL.

52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105

VTV30-M/J OR VT30-M VISUAL DIAGNOSTIC

PROGRAM DESCRIPTION

1. ABSTRACT

THIS IS THE SECOND PART OF A TWO PART DIAGNOSTIC FOR THE VTV30-M/J OR VT30-M GRAPHICS DISPLAY CONTROLLER. THE TESTS CONTAINED IN THIS PART OF THE DIAGNOSTIC GENERATE A SERIES OF VISUAL TEST PATTERNS, WHICH ARE INTENDED TO DEMONSTRATE THAT THE OUTPUT SECTION OF THE DISPLAY CONTROLLER IS WORKING CORRECTLY.

THERE ARE NO ERROR REPORTS IN THE FOLLOWING TESTS. THE ONLY WAY THAT ERRORS CAN BE DETECTED IS BY THE OPERATOR WATCHING THE TEST PATTERNS ON THE T.V. SCREEN. THE TESTS ARE DESIGNED TO DEMONSTRATE THAT THE OUTPUTS OF THE PICTURE STORE AND CHARACTER STORE ARE WORKING, AND ALSO, THAT THE TIMING IS WORKING FOR ALL COMBINATIONS OF 6 AND 8 DOT CHARACTER MATRICES. THE PATTERNS PRODUCED ON THE T.V. SCREEN ARE VERY SIMPLE, AND ANY FAULT WILL PRODUCE AN OBVIOUS DISCONTINUITY OR IRREGULARITY IN THE PATTERN.

2. REQUIREMENTS

2.1 EQUIPMENT

- A. PDP-11 COMPUTER
- B. CONSOLE TELETYPE
- C. VTV30-M/J OR VT30-M
- D. DIAGNOSTIC TAPE AND LISTINGS
- E. COLOUR T.V. MONITOR

2.2 STORAGE

THIS PROGRAM REQUIRES A MINIMUM OF 8K WORDS OF MEMORY.

3. LOADING PROCEDURE

THE PROGRAM IS LOADED USING THE ABSOLUTE BINARY LOADER AND IS IN ABSOLUTE BINARY FORMAT. THE PROGRAM CAN ALSO BE LOADED AND RUN IN THE NORMAL XXDP MANNER.

107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156

4. STARTING PROCEDURE

THE PROGRAM HAS A LOAD AND GO FEATURE WHICH AUTOMATICALLY STARTS THE PROGRAM AT ADDRESS 1000 UPON A SUCCESSFUL LOAD.

5. RESTARTING PROCEDURE

THE PROGRAM HAS A RESTART ADDRESS AT 1200 WHICH ALLOWS THE PROGRAM TO BE RESTARTED WITHOUT HAVING TO RE-ENTER THE BUS AND VECTOR ADDRESSES. IF IT IS NECESSARY TO RESTART THE PROGRAM WITH NEW BUS AND VECTOR ADDRESSES, THE ADDRESSES 1000 OR 200 SHOULD BE USED AS THE RESTART ADDRESS.

6. PROGRAM AND OPERATOR ACTION

THE FOLLOWING OPERATOR REQUESTS ARE MADE BY THE PROGRAM PRIOR TO THE COMMENCEMENT OF THE ACTUAL TESTS.-

TYPE 6 FOR 625-LINES OR 5 FOR 525-LINE DISPLAY
FIRST BUS ADDRESS IS
FIRST VECTOR ADDRESS IS.....
FIRST PRIORITY LEVEL IS

THE OPERATOR SHOULD REPLY TO REQUESTS ABOVE, BY INPUTTING THE CORRECT DATA.

'SELECT DESIRED SWITCH REGISTER SETTINGS.'

'TYPE CNTRL-C TO CONTINUE'

OR

'SWR = 0.'

IN REPLY TO THE REQUEST ABOVE THE OPERATOR SHOULD SELECT DESIRED SWITCH REGISTER OPTIONS AS SET OUT UNDER SWITCH OPTIONS BELOW.

7. SWITCH REGISTER OPTIONS

THIS PROGRAM IS DESIGNED TO RUN EQUALLY EASILY ON PDP-11 PROCESSORS WITH, OR WITHOUT, A HARDWARE SWITCH REGISTER. ON STARTING, A TEST IS DONE TO SEE IF A HARDWARE SWITCH REGISTER IS PRESENT. IF IT IS PRESENT, IT MAY BE USED IN THE NORMAL MANNER.

158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197

THE SWITCH REGISTER SETTINGS ARE:-

SWR15=1 INHIBIT ERROR HALT
SWR14=1 INHIBIT ERROR PRINT-OUT
SWR13=1 FAST ITERATION
SWR12=1 :
SWR11=1 :SCOPE LOOPS, SEE BELOW
SWR10=1 :FOR A DESCRIPTION, AND
SWR09=1 :APPENDIX A FOR EXAMPLES
SWR08=1 :OF THE FIR USE
SWR07=1 :
SWR06=1 SELECTED TEST.
SWR05=1 :
SWR04=1 :
SWR03=1 :TEST NOS.
SWR02=1 :
SWR01=1 :
SWR00=1 :

THE SETTING OF BITS 7, 8, 9, 10, 11, AND 12 IN THE SWITCH REGISTER ARE USED TO SELECT THE TRAP OPTIONS PRESENT IN THE PROGRAM. THE SELECTION IS MADE IN THE FOLLOWING MANNER:

BIT(S) SET	TRAP FUNCTION SELECTED
7	TRAP+2
8	TRAP+4
9	TRAP+10
10	TRAP+20
9 AND 10	TRAP+30
11	TRAP+40
7 AND 11	TRAP+50
10 AND 11	TRAP+60
9, 10 AND 11	TRAP+70
12	USES THE SWITCH REGISTER SETTING THAT WAS IN FORCE WHEN THE LAST TRAP INSTRUCTION WAS EXECUTED.

199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250

IF A HARDWARE SWITCH REGISTER IS NOT PRESENT, THE PROGRAM ASSIGNS A LOCATION IN MEMORY AS A SOFTWARE SWITCH REGISTER, THE OPTIONS REMAINING AS ABOVE. THIS MEANS THAT ALL MODIFICATIONS TO THE SWITCH REGISTER MAY BE MADE USING THE CONSOLE TELETYPE VIA A MONITOR ROUTINE. THIS MONITOR IS CALLED BY TYPING CTRL-G AT THE CONSOLE TELETYPE AND RESPONDS BY PRINTING THE CURRENT CONTENTS OF THE SOFTWARE SWITCH REGISTER, FOLLOWED BY PROMPT CHARACTER (>). THE OPERATOR SHOULD THEN TYPE IN THE NEW SWITCH REGISTER SETTINGS AS AN OCTAL NUMBER, FOLLOWED BY A CARRIAGE RETURN. TYPING CARRIAGE RETURN ALONE WILL CAUSE THE SETTING TO REMAIN UNCHANGED. THE SWITCH REGISTER IS THEN LOADED WITH THE NEW VALUE AND PROGRAM EXECUTION CONTINUES. IF A SETTING IS ENTERED WHICH INCLUDES THE SELECT TEST BIT (SWR06), THE TEST INDICATED BY SWR 00-7 WILL BE SELECTED IMMEDIATELY. THIS DOES NOT APPLY WHEN DEFAULTING ON AN EXISTING SETTING.

THE SWR MONITOR IS ALSO CALLED AUTOMATICALLY IF AN ERROR IS DETECTED AND SWR BIT 15 IS NOT SET. OCTAL EQUIVALENTS FOR THE SWITCHES ARE AS FOLLOWS:

SWR15 =	100000
SWR14 =	40000
SWR13 =	20000
SWR12 =	10000
SWR11 =	4000
SWR10 =	2000
SWR09 =	1000
SWR08 =	400
SWR07 =	200
SWR06 =	100
SWR05 =	.0
SWR04 =	20
SWR03 =	10
SWR02 =	4
SWR01 =	2
SWR00 =	1

TO SET A COMBINATION OF THESE SWITCHES, SIMPLY ADD TOGETHER THE CORRESPONDING OCTAL NUMBERS AND ENTER THE TOTAL IN RESPONSE TO "SWR= X>". (LEADING ZEROS MAY BE IGNORED).

FOR WORST CASE TESTING, ALL SWITCHES SHOULD BE ZERO. IT IS POSSIBLE, WITH THESE SWITCH REGISTER OPTIONS, TO EXECUTE ONLY A PRE-SELECTED TEST WITH THE FACILITY TO LOOP ON THAT TEST OR TO START THE PROGRAM PASS OR FINISH THE PROGRAM PASS AT ANY PARTICULAR TEST.

252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301

8. ERROR REPORTS

THE FORMAT OF THE ERROR REPORT IS AS FOLLOWS:-

E# ABBB	AT PC:CCCC
GOOD: DDDD	BAD: EEEE
STATUS: FFFF	ADDRESS: GGGG
DATA: KKKK	CALLED FROM: H'MH
	ERROR COUNT = JJJJ

WHERE:

AA IS THE TEST NUMBER
BB IS THE ERROR NUMBER
CCCC IS THE ADDRESS WHERE THE ERROR REPORT OCCURS.
DDDD IS THE DATA EXPECTED
EEEE IS THE DATA RECEIVED
FFFF GGGG AND KKKK ARE CONTENTS OF REGISTERS.
HHH IS THE ADDRESS IN THE MAINLINE CODE WHERE THE
SUBROUTINE, WHERE THE ERROR REPORT IS
GENERATED, IS CALLED FROM.
JJJJ IS THE NUMBER OF ERRORS REPORTED TO DATE IN
THIS SECTION.

9. ONLINE MODIFICATIONS

AVAILABLE TO THE USER IS A ROUTINE TO MODIFY PROGRAM
LOCATIONS. IT IS ENTERED BY TYPING CNTRL-O DURING THE
RUNNING OF THE TESTS. ON ENTRY, A PROMPT 'S' IS MADE
FOR THE ADDRESS TO BE MODIFIED. IF NO ADDRESS IS GIVEN,
IT IS ASSUMED THAT NO MODIFICATIONS ARE REQUIRED AND THE
ROUTINE WILL COMPLETE. IF AN ADDRESS IS SPECIFIED, IT
WILL BE CHECKED TO SEE IF IT IS EVEN AND IN EXISTANCE.
ONCE IT HAS BEEN CHECKED, ITS CONTENTS ARE DISPLAYED AND
A PROMPT '/' IS MADE FOR THE NEW CONTENTS. IF NO NEW
VALUE IS GIVEN, THE EXISTING VALUE WILL BE LOADED.
HAVING DEALT WITH THAT ADDRESS, THE ROUTINE WILL THEN
EXAMINE THE TERMINATING CHARACTER TO DETERMINE THE NEXT
OPERATION TO PERFORM.

TYPING <ESC> WILL COMPLETE THE MODIFICATIONS BEING
DONE.
<CR> WILL CAUSE A PROMPT FOR THE NEXT ADDRESS
TO BE MODIFIED.
<LF> WILL TAKE THE NEXT ADDRESS TO BE MODIFIED
AS THE CURRENT ADDRESS+2.

303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347

10. PROGRAM DESCRIPTION

NOTE: ALL FIGURES REFERENCED CAN BE FOUND IN THE OPTION DESCRIPTION SECTION DEALING WITH THE DIAGNOSTIC.

TEST 6

THIS TEST PATTERN STARTS WITH AN ALL BLUE BACKGROUND ON THE SCREEN. THE PICTURE THEN SLOWLY CHANGES TO ALL RED FOREGROUND, STARTING IN THE TOP LEFT-HAND CORNER OF THE SCREEN AND ADVANCING ALONG A DIAGONAL LINE (LIKE AN EXPANDING WEDGE). THE BOUNDARY BETWEEN THE TWO COLOURS SHOULD BE A SMOOTH STRAIGHT LINE, ADVANCING AT A STEADY RATE. AFTER A SHORT PAUSE, THE SCREEN WILL CHANGE (IN THE SAME MANNER) TO ALL GREEN BACKGROUND, BLUE FOREGROUND, RED BACKGROUND, GREEN FOREGROUND AND FINALLY, BLUE BACKGROUND.

THIS SEQUENCE WILL OCCUR TWICE DURING THE TEST.

THIS PATTERN, SHOWN IN FIGURE 6.1, IS FORMED USING THE CHARACTERS SHOWN IN FIGURE 6.2. FOR CHARACTERS 0 TO 42, THE SHADED PARTS OF THE MATRIX INDICATES FOREGROUND COLOUR, AND FOR CHARACTERS 177 TO 165, THE SHADED PARTS INDICATE BACKGROUND COLOUR.

THE PICTURE STARTS WITH ALL BACKGROUND (ALL CHARACTER ZERO), THE STORE IS THEN LOADED AS SHOWN IN FIGURE 6.3. THE ADVANCING LINE IS PRODUCED BY LOADING THE CHARACTERS IN SEQUENCE ALONG THE SHADED LINE. THE LINE IS THEN MOVED FORWARD ONE CHARACTER AND THE CHARACTERS ARE AGAIN LOADED IN SEQUENCE, I.E. WHEN CHANGING FROM ALL BACKGROUND TO ALL FOREGROUND, THE SEQUENCE IS AS FOLLOWS:-

CHARACTER 0 IS LOADED IN THE LEADING EDGE AND CHARACTER 40 IS LOADED IN THE TRAILING EDGE OF THE SHADED LINE IN FIGURE 6.3. THEN 1 AND 100 ARE LOADED, 2 AND 52, 4 AND 101 AND 42, AND FINALLY, 20 AND 177. THE SHADED ROW IS THEN MOVED FORWARD ONE CHARACTER AND THE SEQUENCE REPEATED. THE SCREEN FINISHES UP AS ALL CHARACTER 177, WHICH IS ALL FOREGROUND COLOUR.

349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402

TEST 7

THIS TEST PATTERN STARTS WITH THE TOP HALF OF THE SCREEN RED AND THE BOTTOM HALF OF THE SCREEN BLUE. THE PATTERN NOW CONSISTS OF A GREEN DIAGONAL LINE (SLOPING NORTHEAST TO SOUTHWEST) MOVING FROM LEFT TO RIGHT ACROSS THE TOP HALF OF THE SCREEN (I.E. ON THE RED BACKGROUND) AND THEN ACROSS THE BOTTOM HALF OF THE SCREEN (I.E. ON THE BLUE BACKGROUND).

THE LINE SHOULD BE STRAIGHT AND MOVE AT A STEADY RATE. NOTE: ON SOME COLOUR MONITORS, THE LINE MAY APPEAR AS YELLOW ON THE RED BACKGROUND AND AS CYAN ON THE BLUE BACKGROUND.

TO GENERATE THIS PATTERN, THE PICTURE STORE IS LOADED, AS SHOWN IN FIGURE 7.2. THE COLOUR FOR THE TOP OF THE SCREEN IS GREEN FOREGROUND AND RED BACKGROUND AND THE BOTTOM OF THE SCREEN IS GREEN FOREGROUND ON BLUE BACKGROUND.

THE PATTERN IS FORMED BY MODIFYING THE CONTENTS OF THE CHARACTER STORE. AT THE START OF THE TEST, THE CHARACTER STORE CONTAINS ALL ZEROES. THE CHARACTER MATRICES USED ARE SHOWN IN FIGURE 7.3. CHARACTER 1(A) IS FIRST LOADED INTO CHARACTER STORE ADDRESS ZERO AND CHARACTER 1(B) INTO CHARACTER STORE ADDRESS ONE, THEN CHARACTERS 2(A) AND 2(B) ARE LOADED INTO CHARACTER STORE ADDRESSES ZERO AND ONE, ETC., UNTIL CHARACTERS 8(A) AND 8(B) ARE LOADED, THEN THIS IS REPEATED, USING CHARACTER STORE ADDRESSES 1 AND 2, AND SO ON, THROUGH THE COMPLETE CHARACTER STORE. THIS CYCLES THROUGH THE STORE TWICE.

TEST 10

THIS TEST PATTERN DEMONSTRATES THE OPERATION OF BLINK AT EVERY CHARACTER POSITION.

THE PATTERN STARTS AS A SCREEN FULL OF THE SAME CHARACTERS, THE FIRST ROW BEING RED CHARACTERS ON A BLACK BACKGROUND, THE NEXT ROW IS GREEN ON RED, THEN YELLOW ON GREEN, BLUE ON YELLOW, MAGENTA ON BLUE, CYAN ON MAGENTA, WHITE ON CYAN AND THEN BLACK ON WHITE. THE COLOURS ARE THEN REPEATED, IN THIS ORDER, TO THE END OF THE SCREEN.

THE PATTERN NOW CONSISTS OF AN ADVANCING WEDGE, AS IN TEST 7, THE CHARACTERS AND COLOURS REMAIN UNCHANGED BUT ALL CHARACTERS WITHIN THE ADVANCING WEDGE ARE MADE TO BLINK. AFTER ALL THE CHARACTERS ON THE SCREEN ARE BLINKING, THEY WILL BE STOPPED BLINKING IN THE SAME MANNER.

404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449

TEST 11

THIS TEST PATTERN USES PRESET TO DEMONSTRATE THAT THE SCREEN WILL PRESET TO ANY FOREGROUND OR BACKGROUND COLOUR.

THE TESTS STARTS WITH THE SCREEN ALL BLACK. AFTER A SHORT PAUSE, THE SCREEN WILL CHANGE TO ALL BLUE. THERE WILL BE A SHORT WAIT AND THE SCREEN WILL THEN FLASH BLACK BUT WILL STAY ALL BLUE. AFTER ANOTHER SHORT PAUSE, THE SCREEN WILL CHANGE TO ALL CYAN (BLUE AND GREEN), THEN, AFTER A SHORT PAUSE, WILL FLASH BLACK BUT SHOULD STAY ALL CYAN, AND AFTER A FURTHER WAIT, WILL CHANGE TO ALL GREEN. IN THE SAME MANNER, THE SCREEN WILL CYCLE THROUGH ALL THE COLOURS IN THE FOLLOWING ORDRFR:-

STARTING WITH BLACK, IT WILL GO TO BLUE, THEN CYAN (BLUE AND GREEN), GREEN, WHITE (BLUE AND GREEN AND RED), MAGENTA (BLUE AND RED), YELLOW (GREEN AND RED), RED, AND BACK TO BLACK. THIS SEQUENCE WILL OCCUR TWICE.

TEST 12

THE TEST PATTERNS CONTAINED IN THIS TEST DEMONSTRATE THAT THE DISPLAY IS WORKING FOR ALL COMBINATIONS OF 6 AND 8, V AND H OF THE CHARACTER MATRIX.

THE TEST PATTERN STARTS WITH A BLUE BACKGROUND, AND OUTLINES THE PICTURE WITH A SINGLE DOT WIDTH WHITE LINE. A DIAGONAL LINE IS DRAWN FROM EACH CORNER OF THE SCREEN TO MEET ON THE CENTRE LINE OF THE PICTURE. A LINE TWO DOTS WIDE IS DRAWN ALONG THE CENTRE LINE, ABOVE THIS CENTRE LINE IS WRITTEN THE MATRIX SIZE, I.E. 6H, 8V FOR A MATRIX OF 6 DOTS HORIZONTALLY AND 8 DOTS VERTICALLY.

THE PATTERN IS FIRST DISPLAYED FOR 6H AND 8V, THEN 6V AND 8H, THEN 8 X 8, AND FINALLY, 6 X 6.

TEST 13

THIS TEST DEMONSTRATES THAT THE CURSOR CAN BE DISPLAYED AT ALL X AND Y ADDRESSES.

451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475

THE TEST PATTERN CONSISTS OF THE PATTERN USED FOR THE 6 X 6 CHARACTER MATRIX IN TEST 2. THE CURSOR STARTS IN THE TOP LEFT-HAND CORNER AND THEN MOVES ALONG THE DIAGONAL LINE TO THE CENTRE LINE, THEN DOWN THE DIAGONAL TO THE BOTTOM LEFT-HAND CORNER, ALONG THE BOTTOM OF THE SCREEN TO THE RIGHT-HAND CORNER, AND ALONG THE DIAGONALS TO THE TOP RIGHT-HAND CORNER, FINALLY ALONG THE TOP OF THE PICTURE TO THE START POINT. THE CURSOR MOVES ROUND THE SCREEN TWICE FOR THIS TEST.

TEST 14

THIS TEST IS A X HATCH, IT IS NOT INTENDED TO TEST THE DISPLAY BUT IS TO AID THE SETTING UP OF THE COLOUR MONITOR.

THE PATTERN IS A WHITE GRID ON A BLACK BACKGROUND, WHICH IS DISPLAYED FOR ABOUT ONE MINUTE. TO HOLD THE DISPLAY ON THIS PICTURE, EITHER HALT THE PROCESSOR OR SET SWITCH REGISTER BIT 08.

477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528

APPENDIX A

PDP-11 DIAGNOSTIC LOOPING FACILITIES VIA
SWITCH REGISTER OPTIONS

N.B.

THE INTENTION OF THIS APPENDIX IS TO EXPLAIN, IN GENERAL, THE SCOPE FACILITIES WITHIN PROGRAM CODING. THE USER MUST FIRST EXAMINE THE CODING ABOUT THE AREA HE WISHES TO USE SCOPING FACILITIES, TO ASCERTAIN THAT THE PARTICULAR FACILITY HE REQUIRES IS, IN FACT, AVAILABLE.

PROGRAM LOOPING CONTROL CAN BE SELECTED BY USING SWR 12 - 07. THE PROGRAM HANDLES THIS BY USE OF THE TRAPSV ROUTINE, WHICH IS ENTERED USING THE TRAP INSTRUCTION. BASICALLY, THE ROUTINE CHECKS EQUALITY BETWEEN BITS 05 - 00 OF THE TRAP INSTRUCTION AND SWR 12 - 07.

THERE ARE THREE DISTINCT FUNCTIONS CONTROLLED BY THE TRAP ROUTINE:-

- A) RJN - (TRAP + 2 INSTRUCTION AND SWR 07 SET)
USUALLY USED TO INHIBIT TEST NUMBER PRINTOUT; USEFUL IN THE CASE OF NON-INTERVENTION TESTS. WHEN SWR 07 IS SET, ALL TEST NUMBER MESSAGES ARE SUPPRESSED.
NOTE: IT DOES NOT SUPPRESS ERROR PRINTOUTS.
- B) LOOP ON A SUB-TEST - (TRAP + 4 INSTRUCTION AND SWR 08 SET)
THIS IS GENERALLY USED TO ALLOW THE OPERATOR, BY SETTING SWR 08, TO CONTINUOUSLY LOOP ON ONE LOGICAL TEST OR GROUP OF TESTS.
- C) SCOPE ON A SUB-TEST - (TRAP + 10 - 70 INSTRUCTIONS AND SWR 09 - 11)
THIS ALLOWS THE OPERATOR TO SELECT SEVEN LEVELS OF LOOPING FACILITY WITHIN A SELECTION OR TEST.

IT IS USED TYPICALLY WITHIN A TEST WHERE ONE SUB-TEST SETS A FLAG AND THE NEXT ONE CLEARS IT. BY USING SCOPE LEVEL 1 (SWR 09) - TRAP + 10), HE COULD LOOP ON THE FIRST SUB-TEST, SCOPE LEVEL 2 (SWR 10) - TRAP + 20), HE COULD LOOP ON THE SECOND SUB-TEST OR SCOPE LEVEL 3 (SWR 10 & 09 - TRAP + 30) TO LOOP CONSTANTLY THROUGH BOTH SUB-TESTS SEQUENTIALLY. THE SCOPE RETURN ADDRESS ALWAYS APPEARS AS THE ARGUMENT OR NEXT WORD AFTER THE TRAP INSTRUCTION.

530
531
532
533
534
535
536
537
538
539
540
541
5.2
543
544
545
546
547
548

TO ALLOW THE SCOPE LEVEL TO BE CHANGED WITHOUT STOPPING THE PROGRAM, E.G. TO CHANGE FROM LEVEL 1 TO 2, WHICH WOULD ALMOST CERTAINLY CAUSE LEVEL 3 OR 0 TO BE SEEN MOMENTARILY), A 'PRESERVE SCOPE' FACILITY IS PROVIDED WITH SWR12. WHEN THIS IS SELECTED, THE PROGRAM NO LONGER INSPECTS SWR 11 - 09 BUT USES THE SETTING MEMORISED FROM BEFORE SWR 12 WAS SELECTED. THE SCOPE LEVEL MAY NOW BE CHANGED WITH NO EFFECT UNTIL SWR 12 IS SET TO 0, WHEN THE NEW SCOPE SETTING APPLIES.

N.B. SETTING SWR 12 SHOULD ONLY BE USED TO PRESERVE AS EXISTING SCOPE LEVEL, AS PREVIOUSLY SET ON THE SWITCH REGISTER.

WHICH SCOPE LEVEL TO SELECT MAY BE DETERMINED THE LISTING, LEVELS 1 THROUGH 7 ARE CALLED BY TRAP + 10 (SWR 09) THROUGH TRAP + 70 (SWR 09, 10 AND 11).

550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600

EXAMPLE OF SCOPE LOOP FACILITIES WITHIN DIAGNOSTIC

```

TEST10: MOV      #40,COUNT      ; ITERATION COUNT.
T101:   BIS      #1,STATUS     ; SET GO BIT.
        BIT      #1,STATUS     ; IS GO BIT SET?
        BNE     T10SCP        ; YES, BRANCH.
        JSR     PC,ERROR      ; ERROR!!! GO NOT SET.
        10                ; ERROR NUMBER.
T10SCP: TRAP+10              ; SCOPE TEST10. IF LEVEL 1 SELECTED.
        T101              ; RETURN LABEL FOR SCOPE.
TEST11: BIC      #1,STATUS     ; CLEAR GO BIT
        BIT      #1,STATUS     ; GO BIT CLEAR ?
        BEQ     T11SCP        ; GO BIT CLEAR BRANCH.
        JSR     PC,ERROR      ; NO. GO BIT FAILED TO CLEAR.
        11                ; ERROR!!! NO. 11.
T11SCP: TRAP+30              ; SCOPE TEST 11 IF LEVEL 3 SELECTED.
        TEST11            ; RETURN LABEL FOR SCOPE.
        TRAP+20           ; SCOPE TESTS 10 & 11.
        T101              ; RETURN LABEL.
TEST12: BIS      #100,STATUS   ; NO SCOPE SELECTED. CARRY ON.
        BIT      #100,STATUS  ; IS DONE BIT SET ?
        (ETC.)
        .
        .
T12SCR: TRAP+10              ; SCOPE INT. EN. SET, LEVEL 1.
        TEST 12            ; RETURN LABEL.
TEST13: BIC      #100,STATUS   ; CLEAR INT ENABLE BIT.
        BIT      #100,STATUS  ; IS BIT CLEAR?
        (ETC.)
        .
        .
T13SCP: TRAP+30              ; SCOPE INT CLEAR LEVEL 3.
        TEST13            ; RETURN LABEL.
        TRAP+20           ; SCOPE INT. SET AND CLEAR.
        TEST12            ; LEVEL 2.
        TRAP+4            ; SCOPE LOOP ON THIS SET OF TESTS.
        T101              ; RETURN LABEL.
        DEC      COUNT      ; DO 40 TIMES ANYWAY
        BGT     T101
TEST14: (ETC.)
        .
        .

```

602		.ENDP
603		.TITLE VTV VISUAL TESTS
604		.SBTTL GENERAL DEFINITIONS
605		
606		
607	0000 00	R0=X00
608	000001	R1=X01
609	000002	R2=X02
610	000003	R3=X03
611	000004	R4=X04
612	000005	R5=X05
613	000006	R6=X06
614	000007	R7=X07
615	000006	SP=X06
616	000007	PC=X07
617	177776	PSW=177776
618	177570	MSWR=177570
619		
620		
621	172340	PAR0=172340
622	172342	PAR1=172342
623	172344	PAR2=172344
624	172346	PAR3=172346
625	172350	PAR4=172350
626	172352	PAR5=172352
627	172354	PAR6=172354
628	172356	PAR7=172356
629		
630	172300	PDR0=172300
631	172302	PDR1=172302
632	172304	PDR2=172304
633	172306	PDR3=172306
634	172310	PDR4=172310
635	172312	PDR5=172312
636	172314	PDR6=172314
637	172316	PDR7=172316
638		
639	177572	SR0=177572
640	177574	SR1=177574
641	177576	SR2=177576
642		
643	177546	LKS=177546
644		
645	177746	CA1170=177746
646		
647		
648	001000	REPC1=1000
649	000100	REPC2=100
650	000002	REPC3=2
651		
652		
653		
654	100000	G=100000
655	040000	D=40000
656	020000	A=20000
657	010000	S=10000

658	004000	C=4000
659		
660		
661	177564	TPS=177564
662	177566	TPE=177566
663	177560	TKS=177560
664	177562	TKB=177562
665		
666		
667		
668		
669		
670		.ENABL ABS
671		.ENABL AMA
672		
673		
674		
675		
676		

678
679
680
681
682
683
684
685
686
687
68
689
690
691
692
693
694
695
696
697
698
699
700

```
.SBTTL MACROS
:
: SET PROCESSOR PRIORITY
:
:
: .MACRO PSWSET $ARG,?LAB
MOV $ARG,-(SP) ; SET UP NEW PSW AS $ARG
MOV #LAB,-(SP) ; SET RETURN ADDRESS
RTI ; RTI TO SET PRIORITY
LAB: NOP ; RETURN ADDRESS
.ENDM
:
: READ PROCESSOR PRIORITY
:
: .MACRO PSWREA $ARG
EMT ; ISSUE EMT TO READ PSW
MOV FSAVPW,$ARG ; READ PSW IN $ARG
.ENDM
:
```

```

702          .SBTTL  INITIALISATION
703          000000  .ASECT
704          000000  .=0
705          000200  .=200
706 000200 000137 001000  JMP  @#START      ; JMP TO START AT 200
707          001000  .=1000
708
709
710 001000 012706 001000  START:  MOV  #,5P      ; INITIALISE STACK POINTER
711 001004          PSWSET #340
712
713 001020 012737 017046 000004  MOV  *SWRSET,4      ; TEST FOR HARDWARE SWR
714 001026 012737 000340 000006  MOV  340,6          ; TRAPS TO 4 IF IT
715 001034 012737 177570 016604  MOV  #HSWR,SWR      ; DOES NOT EXIST
716 001042 005777 015536          TST  @SWR
717 001046 005037 016606          CLR  SSWR            ; INITIALISE SOFTWARE SWR
718
719          :
720          : NOW FIND OUT WHETHER THE PROCESSOR IS A SINGLE INTERRUPT
721          : LEVEL LSI-11 OR NOT.
722 001052 004737 017002          JSR  PC,SILLSI      ; FIND OUT ABOUT PROCESSOR
723          :
724 001056 004737 017056          JSR  PC,VECTOR      ; FILL 0-774 WITH HALT TRAPS
725 001062 004737 021172          JSR  PC,TYPOUT
726 001066 014630          GOMSG      ; DIAGNOSTIC
727
728 001070 004737 014420          JSR  PC,SET56
729 001074 004537 022050          JSR  R5,BUSSET      ; SET UP BUS AND VECTOR ADDRESSES
730 001100          007          .BYTE 7,4
731 001102 016556          .WORD CSR
732 001104          001          .BYTE 1,1
733 001106 016576          .WORD INTVEC
734 001110 016600          .WORD VECLEV
735 001112 000000          .WORD 0
736
737          :
738          :
739 001114 013737 016562 016566  MOV  CAR,CARX
740 001122 013737 016562 016570  MOV  CAR,CARY
741 001130 005237 016570          INC  CARY
742 001134 013737 016564 016572  MOV  CHSR,CHDR
743 001142 013737 016564 016574  MOV  CHSR,CHAR
744 001150 005237 016574          INC  CHAR
745 001154 000137 001200          JMP  RSTART        ; THEN GOTO THE RESTART ADDRESS
746          001200  .=1200
747 001200 012706 001000  RSTART: MOV  #START,SP      ; INITIALISE STACK POINTER
748 001204          PSWSET #340
749
750 001220 004737 021172          JSR  PC,TYPOUT
751 001224 015276          WMSG      ; SELECT DESIRED CONSOLE SWITCHES
752 001226 004737 017704  START1: JSR  PC,MONIT      ; GO TO SWR MONITOR
753
754 001232 012706 001000  START2: MOV  #START,SP      ; INITIALISE STACK POINTER
755 001236          PSWSET #340
756
757 001252 032777 000100 015324  BIT  #100,@SWR      ; CHECK FOR PRE-SELECTED TEST

```

```
758 001260 001002          BNE  START3
759
760 001262 000137 001424          JMP  TEST6          ;TO TEST 6
761 001266 000137 001272  START3: JMP  TABLE          ;TO LOOK-UP TABLE
762
763
764
765
766
767
768
769
770
;INTERFACE REGISTERS AND VECTORS
```

```
772 001272 017700 015306      TABLE:  MOV    @SWR,RO      :GET SELECTED TEST NO
773 001276 042700 177700      BIC    #177700,RO
774 001302 022700 000014      CMP    #14,RO      :CHECK FOR VALID TEST NO
775 001306 002734      BLT    RSTART
776
777 001310 020027 000006      CMP    RO,#6
778 001314 100731      BMI    RSTART
779 001316 006300      ASL    RO
780 001320 000170 001324      JMP    @TABLE1(RO)  :JUMP TO PRE-SELECTED TEST
```

781
782
783
784
785

```
786 001324 001424      TABLE1: TEST6
787 001326 001424      TEST6
788 001330 001424      TEST6
789 001332 001424      TEST6
790 001334 001424      TEST6
791 001336 001424      TEST6
792 001340 001424      TEST6
793 001342 002470      TEST7
794 001344 004064      TEST10
795 001346 004652      TEST11
796 001350 005434      TEST12
797 001352 013576      TEST13
798 001354 014136      TEST14
```


800

```

802 001356 042777 100000 015172 TIME1: BIC #100000,@CSR ;CLEAR TIMER BIT
803 001364 005237 001420 INC INTFLG ;SET INTERRUPT BIT
804 001370 005337 001412 DEC COUNT1 ;
805 001374 001005 BNE TIME1A ;IS 1 SECOND UP
806 001376 005237 001422 INC BELLS ;YES , SO SET BELL
807 001402 013737 001416 001412 MOV SECOND,COUNT1 ;SET UP TIME TO NEXT BELL
808 001410 000002 TIME1A: RTI
809
810 001412 000000 COUNT1: 0
811 001414 000000 COUNT2: 0
812 001416 000000 SECOND: 0
813 001420 000000 INTFLG: 0
814 001422 J00000 BELLS: 0
815
816
817 .SBTTL TEST6
818
819 *****
820 :THIS TEST PATTERN IS FOR
821 :DEMONSTRATING THAT THE
822 :PICTURE STORE COLOUR AND
823 :CHARACTER BITS ARE WORKING.
824 *****
825 001424 012737 000006 016602 TEST6: MOV #6,TESTNO ;SET UP TEST NO.
826 001432 104402 TRAP+2
827 001434 001442 T6000
828 001436 004737 016756 JSR PC,TESTR ;OUTPUT TEST NO
829
830 001442 005037 016706 T6000: CLR ERRDIS
831 001446 012704 000001 MOV #1,R4
832 001452 004737 017444 JSR PC,FASTSW
833 001456 010437 016610 MOV R4,REPCNT ;SET UP TEST ITERATION COUNT
834
835 001462 013700 016576 T6001: MOV INTVEC,R0 ;
836 001466 012720 001356 MOV #TIME1,(R0)+ ;SET UP INTERRUPT VECTOR
837 001472 012710 000340 MOV #340,(R0) ;
838 001476 012777 001400 015052 MOV #1400,@CSR ;SET 6H & 6V BITS
839 001504 012701 000200 MOV #200,R1 ;SET NO. OF CHARACTERS
840 001510 105077 015060 CLRB @CHAR ;CLEAR CHARACTER ADDRESS
841 001514 012700 002462 T6002: MOV #ALL,R0 ;*LOAD ALL CHARACTERS
842 001520 004737 001746 JSR PC,LDCH ;*WITH \ \ PATTERN
843 001524 005301 DEC R1 ;*
844 001526 001372 BNE T6002 ;*
845
846 001530 012702 000026 MOV #22.,R2 ;LOAD ALL SPECIAL CHARACTERS
847 001534 012701 002427 MOV #FOR10,R1 ;
848 001540 012703 002400 MOV #FOR11,R3 ;
849 001544 112377 015024 T6003: MOV (R3)+,@CHAR ;LOAD ADDRESS
850 001550 010100 MOV R1,R0 ;ADDRESS OF DATA
851 001552 005201 INC R1 ;ADDRESS FOR NEXT CHAR.
852 001554 004737 001746 JSR PC,LDCH ;LOAD CHARACTER DATA
853 001560 005302 DEC R2 ;HAVE WE LOADED ALL
854 001562 001370 BNE T6003 ;BRANCH IF NO.
855
856 001564 112777 000302 014766 MOVB #302,@DBUF ;CLEAR BLINK CONTROL
857 001572 112777 000242 014760 MOVB #242,@DBUF ;SET GREEN ON BLUE

```

```

858 001600 012777 001411 014750      MOV    #1411,@CSR      ;SET 6H,6V,PRESET & DISP. ON
859 001606 105777 014744      1$:   TSTB    @CSR      ;TEST FOR READY
860 001612 100375                BPL    1$             ;BRANCH IF NOT READY
861
862 001614 112777 000241 014736  T6004: MOVB    #241,@DBUF    ;SET RED ON BLUE
863 001622 004737 001764        JSR    PC,FORGND     ;CHANGE TO FORGROUND
864 001626 112777 000221 014724        MOVB    #221,@DBUF    ;SET RED ON GREEN
865 001634 004737 002174        JSR    PC,BAKGND     ;CHANGE TO BACKGROUND
866 001640 112777 000224 014712        MOVB    #224,@DBUF    ;SET BLUE ON GREEN
867 001646 004737 001764        JSR    PC,FORGND     ;
868 001652 112777 000214 014700        MOVB    #214,@DBUF    ;SET BLUE ON RED
869 001660 004737 002174        JSR    PC,BAKGND     ;
870 001664 112777 000212 014666        MOVB    #212,@DBUF    ;SET GREEN ON RED
871 001672 004737 001764        JSR    PC,FORGND     ;
872 001676 112777 000242 014654        MOVB    #242,@DBUF    ;SET GREEN ON BLUE
873 001704 004737 002174        JSR    PC,BAKGND     ;
874
875 001710 104404                TRAP+4                ;SET SWR BIT 8 TO
876 001712 001462                T6001                ;LOOP ON TEST
877
878 001714 005337 016610        DEC    REPCNT        ;DONE ENOUGH ?
879 001720 001402                BEQ    T60WW         ;YES
880 001722 000137 001614        JMP    T6004         ;NO
881
882 001726 032777 000100 014650  T60WW: BIT    #100,@SWR     ;CHECK FOR PRE-SELECTED TEST
883 001734 001402                BEQ    1$            ;
884 001736 000137 001232        JMP    START2        ;
885 001742 000137 002470      1$:   JMP    TEST7        ;
886
887
888
889
890
891
892 001746 012705 000006      LDCH:  MOV    #6,R5           ;NUMBER OF LOADS
893 001752 112077 014614      LDCH1: MOVB    (R0)+,@CHDR    ;DO A LOAD
894 001756 005305                DEC    R5             ;ANY MORE ?
895 001760 001374                BNE    LDCH1         ;BRANCH IF YES
896 001762 000207                RTS    PC             ;
897
898 001764 013737 014516 002170  FORGND: MOV    HALF6V,MAX      ;
899 001772 006337 002170        ASL    MAX           ;
900 001776 005337 002170        DEC    MAX           ;SET MAX Y ADDRESS
901 002002 005037 002172        CLR    ADDR         ;
902 002006 012701 000005      F3:   MOV    #5,R1           ;
903 002012 013777 002172 014542  F1:   MOV    ADDR,@CAR      ;LOAD CURSOR ADDRESS
904 002020 127727 014542 000120  F4:   CMPB   @CARX,#80.      ;END OF LINE ?
905 002026 001406                BEQ    F4B           ;YES
906 002030 116177 002406 014522        MOVB    FOR5(R1),@DBUF  ;LOAD ONE CHARACTER
907 002036 105777 014524        TSTB   @CARX         ;HAS X REACHED ZERO
908 002042 001415                BEQ    F2            ;BRANCH IF YES
909 002044 105377 014516      F4B:  DECB   @CARX         ;NO SO DECREMENT X
910 002050 116177 002400 014502        MOVB    FOR11(R1),@DBUF ;LOAD OTHER CHARACTER
911 002056 127737 014506 002170        CMPB   @CARY,MAX      ;ARE WE ON BOTTOM LINE
912 002064 001404                BEQ    F2            ;BRANCH IF YES
913 002066 105277 014476        INCB   @CARY         ;NO SO INCREMENT ADDRESS

```

```

;*****
;SUBROUTINE FOR LOADING CHARACTERS
;ENTER WITH DATA ADDRESS IN R0
;AND CHARACTER ADDRESS IN CHAR
;*****

```

```

914 002072 000137 002020          JMP      F4          ;GO BACK AND DO SOME MORE
915
916 002076 012737 000001 004024 F2:    MOV      #1,TIME     ;WAIT SOME TIME
917 002104 004737 003732          JSR      PC,WAIT    ;
918 002110 005301          DEC      R1         ;NEXT PAIR OF CHARACTERS
919 002112 100337          BPL      F1         ;
920
921 002114 123727 002172 000120          CMPB    ADDR,#80.   ;IS X ADDRESS AT MAX
922 002122 001404          BEQ      F5         ;BRANCH IF YES
923 002124 105237 002172          INCB    ADDR        ;NO SO INCREMENT X
924 002130 000137 002006          JMP      F3         ;
925
926 002134 123737 002173 002170 F5:    CMPB    ADDR+1,MAX  ;IS Y ADDRESS AT MAX
927 002142 001404          BEQ      F6         ;BRANCH IF YES
928 002144 105237 002173          INCB    ADDR+1     ;NO SO INCREMENT Y
929 002150 000137 002006          JMP      F3         ;
930
931 002154 012737 000002 004024 F6:    MOV      #2,TIME ;WAIT SOME TIME
932 002162 004737 003732          JSR      PC,WAIT    ;
933 002166 000207          RTS      PC         ;DONE
934
935 002170 000000          MAX:    0
936 002172 000000          ADDR:   0
937

```

```

939 002174 013737 014516 002170 BAKGND: MOV HALF6V,MAX
940 002202 006337 002170 ASL MAX
941 002206 005337 002170 DEC MAX
942 002212 005037 002172 CLR ADDR
943 002216 012701 000005 BF3: MOV #5,R1
944 002222 013777 002172 014332 BF1: MOV ADDR,@CAR
945 002230 127727 014332 000120 BF4: CMPB @CARX,#80. :END OF LINE ?
946 002236 001406 BEQ BF4B :YES
947 002240 116177 002421 014312 MOVB BAC5(R1),@DBUF
948 002246 105777 014314 TSTB @CARX
949 002252 001415 BEQ BF2
950 002254 105377 014306 BF4B: DECB @CARX
951 002260 116177 002413 014272 MOVB BAC11(R1),@DBUF
952 002266 127737 014276 002170 CMPB @CARY,MAX
953 002274 001404 BEQ BF2
954 002276 105277 014266 INCB @CARY
955 002302 000137 002230 JMP BF4
956 002306 012737 000001 004024 BF2: MOV #1,TIME
957 002314 004737 003732 JSR PC,WAITT
958 002320 005301 DEC R1
959 002322 100337 BPL BF1
960 002324 123727 002172 000120 CMPB ADDR,#80.
961 002332 001404 BEQ BF5
962 002334 105237 002172 INCB ADDR
963 002340 000137 002216 JMP BF3
964 002344 123737 002173 002170 BF5: CMPB ADDR+1,MAX
965 002352 001404 BEQ BF6
966 002354 105237 002173 INCB ADDR+1
967 002360 000137 002216 JMP BF3
968 002364 012737 000002 004024 BF6: MOV #2,TIME ;
969 002372 004737 003732 JSR PC,WAITT
970 002376 000207 RTS PC
971
972
973 002400 177 FOR11: .BYTE 177
974 002401 042 FOR10: .BYTE 42
975 002402 101 FOR9: .BYTE 101
976 002403 052 FOR8: .BYTE 52
977 002404 100 FOR7: .BYTE 100
978 002405 040 FOR6: .BYTE 40
979 002406 020 FOR5: .BYTE 20
980 002407 010 FOR4: .BYTE 10
981 002410 004 FOR3: .BYTE 4
982 002411 002 FOR2: .BYTE 2
983 002412 001 FOR1: .BYTE 1
984 002413 000 FOR0:BAC11: .BYTE 0
985 002414 165 BAC10: .BYTE 165
986 002415 076 BAC9: .BYTE 76
987 002416 125 BAC8: .BYTE 125
988 002417 077 BAC7: .BYTE 77
989 002420 137 BAC6: .BYTE 137
990 002421 157 BAC5: .BYTE 157
991 002422 167 BAC4: .BYTE 167
992 002423 173 BAC3: .BYTE 173
993 002424 175 BAC2: .BYTE 175
994 002425 176 BAC1: .BYTE 176

```

995	002426	177			BAC0:	.BYTE	177
996							
997	002427	374			FOR11D:	.BYTE	374
998	002430	374			FOR10D:	.BYTE	374
999	002431	374			FOR9D:	.BYTE	374
1000	002432	374			FOR8D:	.BYTE	374
1001	002433	374			FOR7D:	.BYTE	374
1002	002434	374			FOR6D:	.BYTE	374
1003	002435	370			FOR5D:	.BYTE	370
1004	002436	360			FOR4D:	.BYTE	360
1005	002437	340			FOR3D:	.BYTE	340
1006	002440	300			FOR2D:	.BYTE	300
1007	002441	200			FOR1D:	.BYTE	200
1008	002442	000			BAC11D:	.BYTE	0
1009	002443	000			BAC10D:	.BYTE	0
1010	002444	000			BAC9D:	.BYTE	0
1011	002445	000			BAC8D:	.BYTE	0
1012	002446	000			BAC7D:	.BYTE	0
1013	002447	000			BAC6D:	.BYTE	0
1014	002450	004			BAC5D:	.BYTE	4
1015	002451	014			BAC4D:	.BYTE	14
1016	002452	034			BAC3D:	.BYTE	34
1017	002453	074			BAC2D:	.BYTE	74
1018	002454	174	374	374	BAC1D:	.BYTE	174,374,374,374,374,374
	002457	374	374	374			
1019	002462	070	034	214	ALL:	.BYTE	70, 34,214,304,340,160
	002465	304	340	160			
1020						.EVEN	
1021							
1022							

```

1024 .SBI: TEST7
1025
1026 .....
1027 :THIS TEST PATTERN DEMONSTRATES
1028 :THAT ALL THE BITS IN THE
1029 :CHARACTER STORE WILL SET AND
1030 :CLEAR.
1031 .....
1032 002470 012737 000007 016602 TEST7: MOV #7,TESTNO ;SET UP TEST NO.
1033 002476 104402 TRAP+2
1034 002500 002506 T7000
1035 002502 004737 016756 JSR PC,TESTR ;OUTPUT TEST NO
1036
1037 002506 005037 016706 T7000: CLR ERDIS
1038 002512 012704 000002 MOV #2,R4
1039 002516 004737 017444 JSR PC,FASTSW
1040 002522 010437 016610 MOV R4,REPCNT ;SET UP TEST ITERATION COUNT
1041
1042 002526 013737 014520 004062 MOV HALF8V,ROWS8 ;
1043 002531 006337 004062 ASL ROWS8 ;SET UP ROWS8
1044 002540 005077 014012 T7001: CLR @CSR ;CLEAR THE CSR
1045 002544 012777 000212 014006 MOV #212,@DBUF ;LOAD COLOUR GREEN NO RED
1046 002552 012777 000302 014000 MOV #302,@DBUF ;LOAD NO BLINK
1047 002560 012737 000000 004050 MOV #0,TEMPX ;ZERO TEMPX
1048 002566 012737 000000 004052 MOV #0,TEMPY ;ZERO TEMPY
1049 002574 012700 000000 MOV #0,R0 ;SET CHARACTER IN R0
1050 002600 113737 004050 004054 CAR3: MOVB TEMPX,TEMPX1 ;
1051 002606 113737 004052 004056 MOVB TEMPY,TEMPY1 ;
1052 002614 113737 004054 004060 CAR1: MOVB TEMPX1,IMPADD ;•SET UP
1053 002622 113737 004056 004061 MOVB TEMPY1,IMPADD+1 ;•CURSOR ADDRESS
1054 002630 123727 004060 000100 CMPB IMPADD,#64. ;END OF LINE ?
1055 002636 002002 BGE CAR1B ;YES
1056 002640 004737 004026 JSR PC,LDADPS ;LOAD ADDRESS AND PICTURE STORE
1057 002644 105337 004054 CAR1B: DECB TEMPX1 ;•MOD XBY TO MAKE
1058 002650 105237 004056 INCB TEMPY1 ;•DIAGONAL LINE
1059 002654 105737 004054 TSTB TEMPX1 ;•IS IT
1060 002660 100406 BML CAR2 ;•THE END
1061 002662 123737 004056 014520 CMPB TEMPY1,HALF8V ;•OF THE
1062 002670 100002 BPL CAR2 ;•LINE ?
1063 002672 000137 002614 JMP CAR1 ;NO, THEN DO SOME MORE
1064 002676 005237 004050 CAR2: INC TEMPX ;START OF NEXT LINE
1065 002702 005200 INC R0 ;NEXT CHARACTER
1066 002704 022737 000120 004050 CMP #80,TEMPX ;ANY MORE LINES
1067 002712 100332 BPL CAR3 ;YES, THEN BRANCH
1068 002714 112777 000242 013636 MOVB #242,@DBUF ;LOAD COLOUR GREEN ON R.UE
1069 002722 013737 014520 004052 MOV HALF8V,TEMPY ;•SET UP START FOR 90° TURN
1070 002730 012737 000000 004050 MCV #0,TEMPX ;•HALF OF SCREEN
1071 002736 012700 000100 MCV #64,R0 ;NEW CHARACTER
1072 002742 113737 004050 004054 CAR4: MOVB TEMPX,TEMPX1 ;•SET UP
1073 002750 113737 004052 004056 CAR5: MOVB TEMPY,TEMPY1 ;•CURSOR ADDRESS
1074 002756 113737 004054 004060 MOVB TEMPX1,IMPADD ;
1075 002764 113737 004056 004061 MOVB TEMPY1,IMPADD+1 ;
1076 002772 123727 004060 000100 CMPB IMPADD,#64. ;END OF LINE ?
1077 003000 002002 BGE CAR5A ;YES
1078 003002 004737 004026 JSR PC,LDADPS ;LOAD ADDRESS AND PICTURE STORE
1079 003006 105337 004054 CAR5A: DECB TEMPX1 ;•MOD XBY TO MAKE

```

```

1080 003012 105237 004056          INCB  TEMPY1          ; *DIAGONAL LINE
1081 003016 105737 004054          TSTB  TEMPX1          ; *IS IT
1082 003022 100406          BMI   CAR6            ; *THE END
1083 003024 123737 004056 004062  CMPB  TEMPY1,ROWSB    ; *OF THE
1084 003032 100002          BPL   CAR6            ; *LINE ?
1085 003034 000137 002756          JMP   CAR5            ; NO, THEN DO SOME MORE
1086
1087 003040 005237 004050          CAR6: INC  TEMPX        ; START OF NEXT LINE
1088 003044 005200          INC  RO              ; NEXT CHARACTER
1089 003046 042700 000200          BIC  #200,RO         ; CLEAR TOP BIT
1090 003052 022737 000120 004050  CMP   #80.,TEMPX     ; ANY MORE LINES
1091 003060 100330          BPL   CAR4            ; YES, THEN BRANCH
1092
1093
1094
1095 003062 012700 002000          MOV   #2000,RO       ;
1096 003066 105077 013502          CLRB @CHAR           ; SET CHARACTER ADDRESS TO 0
1097 003072 105077 013474          CAR7: CLRB @CHDR      ; SET CHARACTER STORE TO 0
1098 003076 005300          DEC  RO              ; ALL DONE ?
1099 003100 001374          BNE  CAR7            ; NO, THEN BRANCH

```

; THE PICTURE STORE IS NOW LOADED.

1101	0G3102	052777	000001	013446	BIS	#1,@CSR	;TURN ON DISPLAY
1102	0J3110	G13700	016576		MOV	INTVEC,R0	;
1103	003114	012720	001356		MOV	#TIME1,(R0)+	;SET UP INTERRUPT VECTOR
1104	003120	012710	000340		MOV	#340,(R0)	;
1105	003124	005000			CLR	R0	;SET TO CHARACTER ZERO
1106	003126	000240			NOP		
1107	003130	110077	013440		MOV	R0,@CHAR	;LOAD CHARACTER ADDRESS
1108	003134	012702	003532		MOV	#CHAR0,R2	;SET UP POINTER
1109	003140	012701	000017		MOV	#15.,R1	;
1110	003144	112277	013422		MOV	(R2)+,@CHDR	;LOAD CHARACTER DATA
1111	003150	005301			DEC	R1	;HAVE WE DONE
1112	003152	100374			BPL	C0	;BRANCH IF NO
1113	003154	012737	000001	004024	MOV	#1,TIME	;SET UP WAIT
1114	003162	004737	003732		JSR	PC,WAITT	;WAIT SOME TIME
1115							
1116	003166	110077	013402		MOV	R0,@CHAR	;LOAD CHARACTER ADDRESS
1117	003172	012701	000017		MOV	#15.,R1	;
1118	003176	112277	013370		MOV	(R2)+,@CHDR	;LOAD CHARACTER DATA
1119	003202	005301			DEC	R1	;HAVE WE DONE
1120	003204	100374			BPL	C1	;BRANCH IF NO
1121	003206	012737	000002	004024	MOV	#2,TIME	;SET UP WAIT
1122	003214	004737	003732		JSR	PC,WAITT	;WAIT SOME TIME
1123							
1124	003220	110077	013350		MOV	R0,@CHAR	;LOAD CHARACTER ADDRESS
1125	003224	012701	000017		MOV	#15.,R1	;
1126	003230	112277	013336		MOV	(R2)+,@CHDR	;LOAD CHARACTER DATA
1127	003234	005301			DEC	R1	;HAVE WE DONE
1128	003236	100374			BPL	C2	;BRANCH IF NO
1129	003240	012737	000001	004024	MOV	#1,TIME	;SET UP WAIT
1130	003246	004737	003732		JSR	PC,WAITT	;WAIT SOME TIME
1131							
1132	003252	110077	013316		MOV	R0,@CHAR	;LOAD CHARACTER ADDRESS
1133	003256	012701	000017		MOV	#15.,R1	;
1134	003262	112277	013304		MOV	(R2)+,@CHDR	;LOAD CHARACTER DATA
1135	003266	005301			DEC	R1	;HAVE WE DONE
1136	003270	100374			BPL	C3	;BRANCH IF NO
1137	003272	012737	000002	004024	MOV	#2,TIME	;SET UP WAIT
1138	003274	004737	003732		JSR	PC,WAITT	;WAIT SOME TIME
1139							
1140	00330	110077	013264		MOV	R0,@CHAR	;LOAD CHARACTER ADDRESS
1141	003310	012701	000017		MOV	#15.,R1	;
1142	003314	112277	013252		MOV	(R2)+,@CHDR	;LOAD CHARACTER DATA
1143	003320	005301			DEC	R1	;HAVE WE DONE
1144	003322	100374			BPL	C4	;BRANCH IF NO
1145	003324	012737	000001	004024	MOV	#1,TIME	;SET UP WAIT
1146	003332	004737	003732		JSR	PC,WAITT	;WAIT SOME TIME
1147							
1148	003336	110077	013232		MOV	R0,@CHAR	;LOAD CHARACTER ADDRESS
1149	003342	012701	000017		MOV	#15.,R1	;
1150	003346	112277	013220		MOV	(R2)+,@CHDR	;LOAD CHARACTER DATA
1151	003352	005301			DEC	R1	;HAVE WE DONE
1152	003354	100374			BPL	C5	;BRANCH IF NO
1153	003356	012737	000001	004024	MOV	#1,TIME	;SET UP WAIT
1154	003364	004737	003732		JSR	PC,WAITT	;WAIT SOME TIME
1155							
1156	003370	110077	013200		MOV	R0,@CHAR	;LOAD CHARACTER ADDRESS

```

1157 003374 012701 000017          MOV      #15.,R1          ;
1158 003400 112277 013166          C6:     MOVB     (R2)+,@CHDR ;LOAD CHARACTER DATA
1159 003404 005301          DEC      R1              ;HAVE WE DONE
1160 003406 100374          BPL     C6              ;BRANCH IF NO
1161 003410 012737 000001 004024          MOV     #1,TIME        ;SET UP WAIT
1162 003416 004737 003732          JSR     PC,WAITT       ;WAIT SOME TIME
1163
1164 003422 110077 013146          MOVB    R0,@CHAR      ;
1165 003426 012701 000017          MOV     #15.,R1      ;
1166 003432 112277 013134          C7:     MOVB     (R2)+,@CHDR ;
1167 003436 005301          DEC     R1            ;HAVE WE DONE
1168 003440 100374          BPL     C7            ;
1169 003442 012737 000001 004024          MOV     #1,TIME        ;
1170 003450 004737 003732          JSR     PC,WAITT       ;
1171
1172 003454 005200          INC     R0            ;INCREMENT CHARACTER NUMBER
1173 003456 032700 000200          BIT     #200,R0        ;
1174 003462 001002          BNE     1$            ;
1175 003464 000137 003126          JMP     CX            ;
1176 003470 042700 000200          1$:    BIC     #200,R0    ;KEEP AS CHARACTER
1177
1178 003474 104404          TRAP+4                ;SET SWR BIT 8 TO
1179 003476 002540          T7001                 ;LOOP ON TEST
1180 003500 005300 016610          DEC     REPCNT        ;DONE ENOUGH ?
1181 003504 001400          BEQ     T700W         ;YES
1182 003506 000137 003126          JMP     CX            ;NO
1183
1184 003512 032777 000100 013064  T700W:  BIT     #100,@SWR     ;CHECK FOR PRE-SELECTED TEST
1185 003520 001402          BEQ     1$            ;
1186 003522 000137 001232          JMP     START2        ;
1187 003526 000137 004064          1$:    JMP     TEST10       ;
1188
1189
1190
1191 003532          000      001      002  CHAR0:  .BYTE  0,1,2,4,10,20,40,100
1192 003542          200      000      000  .BYTE  200,0,0,0,0,0,0,0
1193 003552          000      000      001  CHAR1:  .BYTE  0,0,1,2,4,10,20,40
1194 003562          100      200      000  .BYTE  100,200,0,0,0,0,0,0
1195 003572          000      000      000  CHAR2:  .BYTE  0,0,0,1,2,4,10,20
1196 003602          040      100      200  .BYTE  40,100,200,0,0,0,0,0
1197 003612          000      000      000  CHAR3:  .BYTE  0,0,0,0,1,2,4,10
1198 003622          020      040      100  .BYTE  20,40,100,200,0,0,0,0
1199 003632          000      000      000  CHAR4:  .BYTE  0,0,0,0,0,1,2,4
1200 003642          010      020      040  .BYTE  10,20,40,100,200,0,0,0
1201 003652          000      000      000  CHAR5:  .BYTE  0,0,0,0,0,0,1,2
1202 003662          004      010      020  .BYTE  4,10,20,40,100,200,0,0
1203 003672          000      000      000  CHAR6:  .BYTE  0,0,0,0,0,0,0,1
1204 003702          002      004      010  .BYTE  2,4,10,20,40,100,200,0
1205 003712          000      000      000  CHAR7:  .BYTE  0,0,0,0,0,0,0,0
1206 003722          001      002      004  .BYTE  1,2,4,10,20,40,100,200
1207
1208
1209
1210
1211 003732 005037 001420          WAITT: CLR     INTFLG    ;CLEAR FLAG
1212 003736 052777 040000 012612          BIS     #40000,@CSR   ;SET INTERRUPT ENABLE
  
```

```
1213 003744          PSWSET #0          ;TURN ON INTERUPTS
1214 003760 005737 001420 1$: TST INTFLG      ;TEST FOR INTERRUPT
1215 003764 001775          BEQ 1$          ;
1216 003766 005037 001420  CLR INTFLG      ;CLEAR FLAG
1217 003772 005337 004024  DEC TIME          ;
1218 003776 001370          BNE 1$          ;BRANCH IF MORE TO DO
1219 004000 042777 040000 012550 BIC #40000,@CSR ;CLEAR ENABLE
1220 004006          PSWSET #200        ;TURN OFF INTERUPTS
1221 004022 000207          RTS PC          ;WAIT OVER
1222
1223 004024 000000          TIME: 0
1224
1225
1226
1227
1228 004026 013777 004060 012526 LDADPS: MOV TMPADD,@CAR ;LOAD ADDRESS
1229 004034 105777 01251E  LDPS: TSTB @CSR ;TEST FOR READY
1230 004040 100375          BPL LDPS ;BRANCH IF NOT
1231 004042 110077 012512  MOVB RO,@DBUF ;LOAD PICTURE STORE
1232 004046 000207          RTS PC ;RETURN
1233
1234 004050 000000          TEMPX: 0
1235 004052 000000          TEMPY: 0
1236 004054 000000          TEMPX1: 0
1237 004056 000000          TEMPY1: 0
1238 004060 000000          TMPADD: 0
1239 004062 000000          ROWS8: 0
```

```

1241          .SBTTL TEST10
1242          :*****
1243          :THIS TEST PATTERN DEMONSTRATE.
1244          :THE OPERATION OF BLINK AT
1245          :EVERY CHARACTER POSITION.
1246          :*****
1247
1248 004064 012737 000010 016602 TEST10: MOV #10,TESTNO ;SET UP TEST NO.
1249 004072 104402          TRAP+2
1250 004074 004102          T10000
1251 004076 004737 016756          JSR PC,TESTR ;OUTPUT TEST NO
1252
1253 004102 005037 016706          T10000: CLR ERRDIS
1254 004106 012704 000004          MOV #4,R4
1255 004112 004737 017444          JSR PC,FASTSW
1256 004116 010437 016610          MOV R4,REPCNT ;SET UP TEST ITERATION COUNT
1257
1258 004122 013700 016576          T10001: MOV INTVEC,RO ;*SET UP
1259 004126 012720 001356          MOV #TIME1,(RO)+ ;*INTERRUPT
1260 004132 012710 000340          MOV #340,(RO) ;*VECTOR
1261 004136 012777 001404 012412          MOV #1404,@CSR ;SET 6V,6H & INC BITS
1262 004144 012701 000200          MOV #200,R1 ;*
1263 004150 105077 012420          CLR @CHAR ;*
1264 004154 012700 005403          T10003: MOV #PRE2,RO ;*LOAD ALL CHARACTERS
1265 004160 004737 001746          JSR PC,LDCH ;*WITH SAME PATTERN
1266 004164 005301          DEC R1 ;*
1267 004166 001372          BNE T10003 ;*
1268
1269 004170 005077 012366          CLR @CAR ;SET CURSOR TO ZERO
1270 004174 012703 000003          MOV #3,R3 ;
1271 004200 012701 000007          T10004: MOV #7,R1 ;
1272 004204 005000          T10005: CLR RO ;
1273 004206 012702 000120          T10006: MOV #80.,R2 ;CHARACTERS PER LINE
1274 004212 116177 004642 012340          T10007: MOV B(KCOL(R1),@DBUF) ;LOAD A COLOUR
1275 004220 110077 012334          MOV B(R0,@DBUF) ;LOAD CHARACTER
1276 004224 005200          INC RO ;NEXT CHARACTER
1277 004226 005302          DEC R2 ;
1278 004230 001370          BNE T10007 ;BRANCH IF MORE ROWS
1279 004232 112777 000315 012320          MOV B(#315,@DBUF) ;DO CR
1280 004240 112777 000312 012312          MOV B(#312,@DBUF) ;LF
1281 004246 005301          DEC R1 ;
1282 004250 100355          BPL T10005 ;BRANCH IF MORE COLOURS
1283 004252 012701 000007          T10008: MOV #7,R1 ;
1284 004256 012700 000100          T10009: MOV #100,RO ;
1285 004262 012702 000120          MOV #80.,R2 ;
1286 004266 116177 004642 012264          T10010: MOV B(KCOL(R1),@DBUF) ;LOAD COLOUR
1287 004274 110077 012260          MOV B(R0,@DBUF) ;LOAD CHARACTER
1288 004300 005200          INC RO ;
1289 004302 042700 000200          BIC #200,RO ;SET UP CHARACTER
1290 004306 005302          DEC R2 ;ANY MORE
1291 004310 001366          BNE #10C10 ;BRANCH IF YES
1292 004312 112777 000315 012240          MOV B(#315,@DBUF) ;CR
1293 004320 112777 000312 012232          MOV B(#312,@DBUF) ;LF
1294 004326 005301          DEC R1 ;
1295 004330 100352          BPL T10009 ;
1296 004332 005303          DEC R3 ;

```

```

1297 004334 001321          BNE      T10004          ;
1298                                ;THE PICTURE STORE IS NOW LOADED
1299                                ;*****
1300                                ;NOW START SETTING & CLEARING BLINK
1301 004336 012777 001401 012212      MOV      #1401,@CSR      ;SET 6V,6H & DISPLAY ON
1302 004344 012737 000303 004640      MOV      #303,BLINK     ;BLINK ON CODE
1303 004352 012737 000057 0C2170      MOV      #47.,MAX       ;MAX NUMBER OF ROWS
1304 004360 113777 004640 012172      MOVB    BLINK,@DBUF     ;LOAD BLINK ON/OFF COMMAND
1305 004366 005037 002172          CLR      ADDR           ;
1306 004372 013777 002172 012162      MOV      ADDR,@CAR      ;LOAD ADDRESS
1307 004400 052777 000020 012150      BIS     #20,@CSR        ;READ STORE
1308 004406 017700 012146          MOV      @DBUF,R0       ;SAVE CONTENTS
1309 004412 042700 00C100          BIC     #100,R0         ;
1310 004416 052700 000200          BIS     #200,R0        ;MAKE COLOUR COMMAND
1311 004422 110077 012132      MOVB    R0,@DBUF       ;LOAD COLOUR
1312 004426 00C300          SWAB    R0              ;GET CHARACTER
1313 004430 110077 012124      MOVB    R0,@DBUF       ;LOAD IT
1314
1315 004434 105777 012126          TSTB    @CARX           ;TEST X ADDRESS
1316 004440 001412          BEQ     BK2             ;BRANCH IF ZERO
1317 004442 105377 012120          DECB    @CARX           ;DECREMENT X ADDRESS
1318 004446 127737 012116 002170      CMPB    @CARX,MAX      ;CHECK Y ADDRESS
1319 004454 001404          BEQ     BK2             ;BRANCH IF =
1320 004456 105277 012106          INCB    @CARX           ;INCREMENT Y ADDRESS
1321 004462 000137 004400          JMP     BK4             ;DO SOME MORE
1322 004466 012737 000004 004024      MOV      #4,TIME        ;SET UP WAIT
1323 004474 004737 003732          JSR     PC,WAITT        ;NOW WAIT
1324 004500 123727 002172 000117      CMPB    ADDR,#79.      ;HAVE WE FINISHED
1325 004506 001404          BEQ     BK5             ;BRANCH IF YES
1326 004510 105237 002172          INCB    ADDR           ;
1327 004514 000137 004372          JMP     BK1             ;
1328 004520 123737 002173 002170      CMPB    ADDR+1,MAX     ;ALL DONE ?
1329 004526 001404          BEQ     BK6             ;BRANCH IF YES
1330 004530 105237 002173          INCB    ADDR+1        ;INCREMENT Y ADDRESS
1331 004534 000137 004372          JMP     BK1             ;GO AND DO SOME MORE
1332 004540 012737 000500 004024      MOV      #500,TIME     ;SET UP FOR WAIT BEFORE
1333 004546 004737 003732          JSR     PC,WAITT        ;NEXT PASS
1334 004552 032737 000001 004640      BIT     #1,BLINK       ;
1335 004560 001404          BEQ     BK7             ;
1336 004562 042737 000001 004640      BIC     #1,BLINK       ;
1337 004570 000403          BR      BK8             ;
1338 004572 052737 000001 004640      BIS     #1,BLINK       ;*MOD BLINK COMMAND
1339
1340                                BK8:   TRAP+4          ;SET SWR BIT 8 TO
1341 004600 104404          BK0              ;LOOP ON TEST
1342 004602 004360          DEC     REPCNT         ;DONE ENOUGH ?
1343 004604 005337 016610          BEQ     T100WW        ;YES
1344 004610 001402          JMP     BK0            ;NO
1345
1346 004616 112777 000300 011734      T100WW: MOVB    #300,@DBUF ;CLEAR BLINK FLAG
1347 004624 032777 000100 011752      BIT     #100,@SWR      ;CHECK FOR PRE-SELECTED TEST
1348 004632 001407          BEQ     TEST11        ;
1349 004634 000137 001232          JMP     START2        ;
1350
1351 004640 000000          BLINK.  0
1352

```

1353 004642 270 267 256 BKCOL: .BYTE 270,267,256,245,234,223,212,201
004645 245 234 223
004650 212 201

1354
1355
1356
1357
1358

```

1360          .SBTTL TEST11
1361          ;*****
1362          ;THIS TEST PATTERN USES PRESET
1363          ;TO DEMONSTRATE THAT THE SCREEN
1364          ;WILL PRESET TO ANY FORGROUND
1365          ;OR BACKGROUND COLOUR.
1366          ;*****
1367
1368 004652 104402          TEST11: TRAP+2
1369 004654 004670          T11000
1370
1371 004656 012737 000011 016602          MOV #11,TESTNO          ;SET UP TEST NO
1372 004664 004737 016756          JSR PC,TESTR          ;OUTPUT TEST NO
1373
1374 004670 005037 016706          T11000: CLR ERRDIS
1375 004674 012704 000002          MOV #REPCNT3,R4
1376 004700 004737 017444          JSR PC,FASTSW
1377 004704 010437 016610          MOV R4,REPCNT          ;SET UP TEST ITERATION COUNT
1378
1379 004710 013700 016576          T11001: MOV INTVEC,R0          ;*SET UP
1380 004714 012720 001356          MOV #TIME1,(R0)+          ;*INTERRUPT
1381 004720 012710 000340          MOV #340,(R0)          ;*VECTOR
1382 004724 012777 001400 011624          MOV #1400,@CSR          ;SET 6V & 6H BITS
1383 004732 012701 000200          MOV #200,R1          ;*
1384 004736 105077 011632          CLRB @CHAR          ;*
1385 004742 012700 005372          T1102: MOV #PRE0,R0          ;*LOAD ALL CHARACTERS
1386 004746 004737 001746          JSR PC,LDCH          ;*WITH ZERO
1387 004752 005301          DEC R1          ;*
1388 004754 001372          BNE T1102          ;*
1389
1390 004756 112777 000204 011574          T1103: MOVB #204,@DBUF          ;LOAD COLOUR
1391 004764 004737 005134          JSR PC,PRESUB          ;CHANGE PICTURE
1392 004770 112777 000246 011562          MOVB #246,@DBUF          ;
1393 004776 004737 005134          JSR PC,PRESUB          ;
1394 005002 112777 000262 011550          MOVB #262,@DBUF          ;
1395 005010 004737 005134          JSR PC,PRESUB          ;
1396 005014 112777 000227 011536          MOVB #227,@DBUF          ;
1397 005022 004737 005134          JSR PC,PRESUB          ;
1398 005026 112777 000275 011524          MOVB #275,@DBUF          ;
1399 005034 004737 005134          JSR PC,PRESUB          ;
1400 005040 112777 000253 011512          MOVB #253,@DBUF          ;
1401 005046 004737 005134          JSR PC,PRESUB          ;
1402 005052 112777 000231 011500          MOVB #231,@DBUF          ;
1403 005060 004737 005134          JSR PC,PRESUB          ;
1404 005064 112777 000210 011466          MOVB #210,@DBUF          ;
1405 005072 004737 005134          JSR PC,PRESUB          ;
1406
1407
1408
1409 005076 104404          TRAP+4          ;SET SWR BIT 8 TO
1410 005100 004756          T1103          ;LOOP ON TEST
1411 005102 005337 016610          DEC REPCNT          ;DONE ENOUGH ?
1412 005106 001402          BEQ T110WW          ;YES
1413 005110 000137 004756          JMP T1103          ;NO
1414
1415 005114 032777 000100 011462          T110WW: BIT #100,@SWR          ;CHECK FOR PRE-SELECTED TEST

```

```

1416 005122 001402          BEQ    T110WY
1417 005124 000137 001232    JMP    START2
1418
1419 005130 000137 005434    T110WY: JMP    TEST12
1420
1421 005134 042777 000001  J11414 PRESUB: BIC    #1,@CSR    ;TURN DISPLAY OFF
1422 005142 052777 000010  J11406      BIS    #10,@CSR    ;DO PRESET
1423 005150 105777 011402    PRES1:  STB    @CSR    ;TEST FOR READY
1424 005154 100375          BPL    PRES1    ;BRANCH IF NOT READY
1425 005156 052777 000001  011372    BIS    #1,@CSR    ;TURN DISPLAL BACK ON
1426 005164 105077 011404    CLRB   @CHAR    ;ZERO CHARACTER ADDRESS
1427 005170 012700 005372    MOV    #PRE0,RO
1428 005174 004737 001746    JSR    PC,LDCH   ;LOAD CHARACTER
1429 005200 012737 000200    MOV    #200,TIME
1430 005206 004737 003732    JSR    PC,WAITT  ;WAIT SHORT TIME
1431 005212 105077 011356    CLRB   @CHAR    ;ZERO CHARACTER ADDRESS
1432 005216 012700 005376    MOV    #PRE1,RO
1433 005222 004737 001746    JSR    PC,LDCH   ;LOAD CHARACTER
1434 005226 012737 000004  004024    MOV    #4,TIME
1435 005234 004737 003732    JSR    PC,WAITT  ;WAIT SHORT TIME
1436 005240 105077 011330    CLRB   @CHAR    ;ZERO CHARACTER ADDRESS
1437 005244 012700 005403    MOV    #PRE2,RO
1438 005250 004737 001746    JSR    PC,LDCH   ;LOAD CHARACTER
1439 005254 012737 000004  004024    MOV    #4,TIME
1440 005262 004737 003732    JSR    PC,WAITT  ;WAIT SHORT TIME
1441 005266 105077 011302    CLRB   @CHAR    ;ZERO CHARACTER ADDRESS
1442 005272 012700 005411    MOV    #PRE3,RO
1443 005276 004737 001746    JSR    PC,LDCH   ;LOAD CHARACTER
1444 005302 012737 000004  004024    MOV    #4,TIME
1445 005310 004737 003732    JSR    PC,WAITT  ;WAIT SHORT TIME
1446 005314 105077 011254    CLRB   @CHAR    ;ZERO CHARACTER ADDRESS
1447 005320 012700 005417    MOV    #PRE4,RO
1448 005324 004737 001746    JSR    PC,LDCH   ;LOAD CHARACTER
1449 005330 012737 000004  004024    MOV    #4,TIME
1450 005336 004737 003732    JSR    PC,WAITT  ;WAIT SHORT TIME
1451 005342 105077 011226    CLRB   @CHAR    ;ZERO CHARACTER ADDRESS
1452 005346 012700 005425    MOV    #PRE5,RO
1453 005352 004737 001746    JSR    PC,LDCH   ;LOAD CHARACTER
1454 005356 012737 000200  004024    MOV    #200,TIME
1455 005364 004737 003732    JSR    PC,WAITT  ;WAIT SHORT TIME
1456 005370 000207    RTS    PC        ;FINISHED
1457
1458
1459 005372    000    000    000  PRE0:  .NLIST BEX
1460 005376    000    000    060  PRE1:  .BYTE 0,0,0,0
1461 005403    000    060    170  PRE2:  .BYTE 0,0,60,60,0
1462 005411    060    170    374  PRE3:  .BYTE 0,60,170,170,60,0
1463 005417    170    374    374  PRE4:  .BYTE 60,170,374,374,170,60
1464 005425    374    374    374  PRE5:  .BYTE 170,374,374,374,374,170
1465          005434    .EVEN
1466          .LIST BEX

```


1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514

005434 104402
005436 005452
005440 012737 000012 016602
005446 004737 016756
005452 005037 016706
005456 012704 000001
005462 004737 017444
005466 010437 016610
005472 013700 016576
005476 012720 001356
005502 012710 000340
005506 005077 011044
005512 105077 011056
005516 012700 002000
005522 112777 000000 011042
005530 005300
005532 001373
005534 112777 000110 011032
005542 012700 012572
005546 004737 012550
005552 112777 000120 011014
005560 012700 012602
005564 004737 012550
005570 112777 000070 010776
005576 012700 012612
005602 004737 012550
005606 112777 000066 010760
005614 012700 012622
005620 004737 012550
005624 112777 000177 010742
005632 012700 012632
005636 004737 012550

.SBTTL TEST12

TEST12: TRAP+2
T12000

T12000: CLR ERRDIS
MOV #1,R4
JSR PC,FASTSW
MOV R4,REPCNT

T12001: MOV INTVEC,R0 ;*SET UP
MOV #TIME1,(R0)+ ;*INTERRUPT
MOV #340,(R0) ;*VECTOR

T12004: LLR @CSR ;
CLRB @CHAR ;CLEAR CHARACTER ADDRESS
MOV #2000,R0 ;
MOVB #0,@CHDR ;SET ALL CHARACTERS TO ZERO
DEC R0 ;
BNE T12004 ;
MOVB #'H,@CHAR ;LOAD H
MOV #T12H,R0 ;
JSR PC,LDCH8V ;

MOVB #'V,@CHAR ;LOAD V
MOV #T12V,R0 ;
JSR PC,LDCH8V ;

MOVB #'8,@CHAR ;LOAD 8
MOV #T128,R0 ;
JSR PC,LDCH8V ;

MOVB #'6,@CHAR ;LOAD 6
MOV #T126,R0 ;
JSR PC,LDCH8V ;

MOVB #'77,@CHAR ;LOAD #
MOV #T12SLD,R0 ;
JSR PC,LDCH8V ;

:THIS TEST DEMONSTRATES THAT
:THE CONTROLLER TIMING IS WORKING
:FOR ALL COMBINATIONS OF
: 6 B B , V B H.
:*****

```

1516
1517 005642 012777 000400 010706 112005: MOV #400,@CSR ;*****NOW LOAD SPECIAL CHARACTERS
1518 005650 112777 000001 010716 MOVB #1,@CHAR ;SET 6H
1519 005656 012700 012642 MOV #V86H1,R0
1520 005662 004737 012550 JSR PC,LDCH8V ;CHARACTER 1
1521 005666 004737 012550 JSR PC,LDCH8V ; 2
1522 005672 004737 012550 JSR PC,LDCH8V ; 3
1523 005676 004737 012550 JSR PC,LDCH8V ; 4
1524 005702 004737 012550 JSR PC,LDCH8V ; 5
1525 005706 004737 012550 JSR PC,LDCH8V ; 6
1526
1527 005712 112777 000011 010654 MOVB #11,@CHAR ;
1528
1529 005720 004737 012550 JSR PC,LDCH8V ; 11
1530 005724 004737 012550 JSR PC,LDCH8V ; 12
1531 005730 004737 012550 JSR PC,LDCH8V ; 13
1532 005734 004737 012550 JSR PC,LDCH8V ; 14
1533 005740 004737 012550 JSR PC,LDCH8V ; 15
1534 005744 004737 012550 JSR PC,LDCH8V ; 16
1535
1536 005750 112777 000021 010616 MOVB #21,@CHAR ;
1537
1538 005756 004737 012550 JSR PC,LDCH8V ; 21
1539 005762 004737 012550 JSR PC,LDCH8V ; 22
1540 005766 004737 012550 JSR PC,LDCH8V ; 23
1541 005772 004737 012550 JSR PC,LDCH8V ; 24
1542
1543 005776 112777 000041 010570 MOVB #41,@CHAR ;
1544
1545 006004 004737 012550 JSR PC,LDCH8V ; 41
1546 006010 004737 012550 JSR PC,LDCH8V ; 42
1547 006014 004737 012550 JSR PC,LDCH8V ; 43
1548 006020 004737 012550 JSR PC,LDCH8V ; 44
1549
1550 006024 112777 000051 010542 MOVB #51,@CHAR ;
1551
1552 006032 004737 012550 JSR PC,LDCH8V ; 51
1553 006036 004737 012550 JSR PC,LDCH8V ; 52
1554 006042 004737 012550 JSR PC,LDCH8V ; 53
1555 006046 004737 012550 JSR PC,LDCH8V ; 54
1556
1557 ;*****NOW START TO LOAD PICTURE
1558 006052 005737 014514 112006: TST L525 ;
1559 006056 001007 BNE 1$ ;
1560 006060 012737 000006 012566 MOV #6,VCNT ;
1561 006066 012737 000040 012570 MOV #32,HCNT ;
1562 006074 000406 BR 2$ ;
1563 006076 012737 000005 012566 1$: MOV #5,VCNT ;PARAMATERS FOR 525L1.e
1564 006104 012737 000050 012570 MOV #40,HCNT ;
1565 006112 013737 014520 002170 2$: MOV HALF8V,MAX ;
1566 006120 006337 002170 ASL MAX ;
1567 006124 005337 002170 DEC MAX ;
1568 006130 112777 000247 010422 MOVB #247,@DBUF ;COLOUR WHITE ON BLUE
1569 006136 052777 000010 010412 BIS #10,@CSR ;DO PRESET
1570 006144 105777 010406 9$: TSTB @CSR ;
1571 006150 100375 BPL 9$ ;
  
```


1628	006464	105377	010076			DECB	@CARX	:
1629	006470	105277	010074			INCB	@CARY	:
1630	006474	005300				DEC	R0	:
1631	006476	001370				BNE	6S	:
1632								:
1633	006500	013700	012570			MOV	MCNT,R0	:
1634								:
1635	006504	112777	000021	010046	7S:	MOVB	#21,@DBUF	:LOAD ----- LINE ↓
1636	006512	105377	010050			DECB	@CARX	:
1637	006516	005300				DEC	R0	:
1638	006520	001371				BNE	7S	:
1639								:
1640	006522	013700	012566			MOV	VCNT,R0	:
1641	006526	004737	012272		8S:	JSR	PC,BLTRBV	:LOAD / LINE
1642	006532	005300				DEC	R0	:
1643	006534	001405				BEQ	T12008	:
1644	006536	105377	010024			DECB	@CARX	:
1645	006542	105277	010022			INCB	@CARY	:
1646	006546	006767				BR	8S	:
1647								:
1648	006550	052777	000004	010000	T12008:	BIS	#4,@CSR	:SET INCREMENT BIT
1649	006556	005077	010000			CLR	@CAR	:
1650	006562	112777	000041	007770		MOVB	#41,@DBUF	:
1651	006570	112777	000042	007762		MOVB	#42,@DBUF	:
1652	006576	012777	00116	007756		MOV	#78,@CAR	:
1653	006604	112777	000053	007746		MOVB	#53,@DBUF	:
1654	006612	112777	000054	007740		MOVB	#54,@DBUF	:
1655	006620	105077	007742			CLRB	@CARX	:XYZ
1656	006624	113777	002170	007736		MOVB	MAX,@CARY	:
1657	006632	112777	000051	007720		MOVB	#51,@DBUF	:
1658	006640	112777	000052	007712		MOVB	#52,@DBUF	:
1659	006646	112777	000116	007704		MOVB	#78,@CARX	:
1660	006654	112777	000043	007696		MOVB	#43,@DBUF	:
1661	006662	112777	000044	007690		MOVB	#44,@DBUF	:
1662	006670	113777	014520	007682		MOVB	HALF8V,@CARY	:SET UP MESSAGE
1663	006676	162777	001000	007656		S	#1000,@CAR	:
1664	006704	112777	000043	007654		MOVB	#35,@CARX	:
1665	006712	112777	000177	007640		MOVB	#177,@DBUF	:
1666	006720	112777	000000	007632		MOVB	#0,@DBUF	:
1667	006726	112777	000070	007624		MOVB	#8,@DBUF	:
1668	006734	112777	000126	007616		MOVB	#V,@DBUF	:
1669	006742	112777	000000	007610		MOVB	#0,@DBUF	:
1670	006750	112777	000000	007602		MOVB	#0,@DBUF	:
1671	006756	112777	000066	007594		MOVB	#6,@DBUF	:
1672	006764	112777	000110	007586		MOVB	#M,@DBUF	:
1673	006772	112777	000030	007580		MOVB	#0,@DBUF	:
1674	007000	112777	000177	007552		MOVB	#177,@DBUF	:
1675	007006	042777	000004	007542		BIC	#4,@CSR	:CLEAR INC BIT
1676	007014	052777	000001	007534		BIS	#1,@CSR	:TURN ON DISPLAY
1677	007022	012737	000500	004024		MOV	#500,T,ME	:
1678	007030	004737	003732			JSR	PC,WAIT	:WAIT A SHORT TIME

```

1680
1681 007034 012777 001000 007514 T12010: MOV #1000,@CSR ;*****NOW LOAD SPECIAL CHARACTERS 6V 8H
1682 007042 112777 000001 007524 MOVB #1,@CHAR ;SET 6V
1683 007050 012700 013142 MOV #V68H1,R0 ;
1684 007054 004737 001746 JSR PC,LDCM ;CHARACTER 1
1685 007060 004737 001746 JSR PC,LDCM ; 2
1686 007064 004737 001746 JSR PC,LDCM ; 3
1687 007070 004737 001746 JSR PC,LDCM ; 4
1688 007074 004737 001746 JSR PC,LDCM ; 5
1689 007100 004737 001746 JSR PC,LDCM ; 6
1690
1691 007104 112777 000011 007462 MOVB #11,@CHAR ;
1692
1693 007112 004737 001746 JSR PC,LDCM ; 11
1694 007116 004737 001746 JSR PC,LDCM ; 12
1695 007122 004737 001746 JSR PC,LDCM ; 13
1696 007126 004737 001746 JSR PC,LDCM ; 14
1697 007132 004737 001746 JSR PC,LDCM ; 15
1698 007136 004737 001746 JSR PC,LDCM ; 16
1699
1700 007142 112777 000021 007424 MOVB #21,@CHAR ;
1701
1702 007150 004737 001746 JSR PC,LDCM ; 21
1703 007154 004737 001746 JSR PC,LDCM ; 22
1704 007160 004737 001746 JSR PC,LDCM ; 23
1705 007164 004737 001746 JSR PC,LDCM ; 24
1706
1707 007170 112777 000041 007376 MOVB #41,@CHAR ;
1708
1709 007176 004737 001746 JSR PC,LDCM ; 41
1710 007202 004737 001746 JSR PC,LDCM ; 42
1711 007206 004737 001746 JSR PC,LDCM ; 43
1712 007212 004737 001746 JSR PC,LDCM ; 44
1713
1714 007216 112777 000051 007350 MOVB #51,@CHAR ;
1715
1716 007224 004737 001746 JSR PC,LDCM ; 51
1717 007230 004737 001746 JSR PC,LDCM ; 52
1718 007234 004737 001746 JSR PC,LDCM ; 53
1719 007240 004737 001746 JSR PC,LDCM ; 54
1720
1721 ;*****NOW START TO LOAD PICTURE
1722 007244 005737 014514 T12011: TST L525 ;
1723 007250 001007 BNE 'S ;
1724 007252 012737 000006 012566 MOV #6,VCNT ;
1725 007260 012737 000034 012570 MOV #28,HCNT ;
1726 007266 000406 BR 2$ ;
1727 007270 012737 000005 012566 1$: MOV #5,VCNT ;PARAMATERS FOR 525LINE
1728 007276 012737 000042 012570 MOV #34,HCNT ;
1729 007304 013737 014516 002170 2$: MOV HALF@V,MAX ;
1730 007312 006337 002170 ASL MAX ;
1731 007316 005337 002170 DEC MAX ;
1732 007322 112777 000247 007230 MOVB #247,@DB IF ;COLOUR WHITE ON BLUE
1733 007330 052777 000010 007220 BIS #10,@CSR ;DO PRESET
1734 007336 105777 007214 9$: TSTB @CSR ;
1735 007342 100375 BPL 9$ ;
    
```

1736	007344	005077	007212		CLR	@CAR	:ZERO CURSCR	
1737	007350	052777	000004	007200	BIS	#4,@CSR	:SET INC BIT	
1738	007356	012700	000100		MOV	#64.,RO	:	
1739	007362	112777	000024	007170	3\$:	MOVB	#24,@DBUF	:LOAD ----- LINE
1740	007370	005300			DEC	RO	:	
1741	007372	001373			BNE	3\$:	
1742							:	
1743	007374	105077	007166		CLRB	@CARX	:	
1744	007400	113777	002170	007162	MOVB	MAX,@CARY	:	
1745	007406	012700	000100		MOV	#64.,RO	:	
1746	007412	112777	000023	007140	4\$:	MOVB	#23,@DBUF	:LOAD ----- LINE
1747	007420	005300			DEC	RO	:	
1748	007422	001373			BNE	4\$:	
1749							:	
1750	007424	042777	000004	007124	BIC	#4,@CSR	:CLEAR INC BIT	
1751	007432	005077	007124		CLR	@CAR	:	
1752	007436	013700	002170		MOV	MAX,RO	:	
1753	007442	112777	000021	007110	5\$:	MOVB	#21,@DBUF	:LOAD [LINE
1754	007450	112777	000312	007102		MOVB	#312,@DBUF	:
1755	007456	005300			DEC	RO	:	
1756	007460	000370			BPL	5\$:	
1757							:	
1758	007462	012777	000077	007072	MOV	#63.,@CAR	:	
1759	007470	013700	002170		MOV	MAX,RO	:	
1760	007474	112777	000022	007056	6\$:	MOVB	#22,@DBUF	:LOAD] LINE
1761	007502	112777	000312	007050		MOVB	#312,@DBUF	:
1762	007510	005300			DEC	RO	:	
1763	007512	100370			BPL	6\$:	
1764							:	
1765	007514	013700	012566		T12012:	MOV	VCNT,RO	:
1766	007520	012777	000000	007034		MOV	#0,@CAR	:
1767	007526	004737	012364		1\$:	JSR	PC,TLBR6V	:LOAD \ LINE
1768	007532	005300			DEC	RO	:	
1769	007534	001405			BEQ	2\$:	
1770	007536	105277	007024		INCB	@CARX	:	
1771	007542	105277	007022		INCB	@CARY	:	
1772	007546	000767			BR	1\$:	
1773	007550	105277	007012		2\$:	INCB	@CARX	:
1774	007554	013700	012570		MOV	HCNT,RO	:	
1775	007560	052777	000004	006770		BIS	#4,@CSR	:SET INCREMENT BIT
1776	007566	112777	000023	006764	3\$:	MOVB	#23,@DBUF	:LOAD ----- LINE
1777	007574	005300			DEC	RO	:	
1778	007576	001373			BNE	3\$:	
1779	007600	042777	000004	006750		BIC	#4,@CSR	:CLEAR INCREMENT BIT
1780	007606	105277	006756		INCB	@CARY	:	
1781	007612	013700	012566		MOV	VCNT,RO	:	
1782	007616	004737	012364		4\$:	JSR	PC,TLBR6V	:LOAD \ LINE
1783	007622	005300			DEC	RO	:	
1784	007624	001405			BEQ	5\$:	
1785	007626	105277	006734		INCB	@CARX	:	
1786	007632	105277	006732		INCB	@CARY	:	
1787	007636	000767			BR	4\$:	
1788							:	
1789	007640	013700	012566		5\$:	MOV	VCNT,RO	:
1790	007644	012777	000077	006710		MOV	#63.,@CAR	:
1791	007652	004737	012456		6\$:	JSR	PC,BLTR6V	:LOAD / LINE

```

1792 007656 105377 006704      DECB  @CARX      :
1793 007662 105277 006702      INCB  @CARY      :
1794 007666 005300              DEC   RO         :
1795 007670 001370              BNE   CS         :
1796
1797 007672 013700 012570      MOV   MCNT,RO    :
1798
1799 007676 112777 000024 006654 78:  MOVB  #24,@DBUF  :LOAD ----- LINE
1800 007704 105377 006656      DECB  @CARX      :
1801 007710 005300              DEC   RO         :
1802 007712 001371              BNE   78        :
1803
1804 007714 013700 012566      MOV   VCNT,RO    :
1805 007720 004737 012456      JSR   FC,BLTR6V  :LOAD /      LINE
1806 007724 005300              DEC   RO         :
1807 007726 001405              BEQ   T12013     :
1808 007730 105377 006632      DECB  @CARX      :
1809 007734 105277 006630      INCB  @CARY      :
1810 007740 000767              BR    88        :
1811
1812 007742 005077 006614      T12013: CLR   @CAR      :
1813 007746 112777 000054 006604      MOVB  #54,@DBUF  :
1814 007754 112777 000312 006576      MOVB  #312,@DBUF :DO L.F
1815 007762 112777 000053 006570      MOVB  #53,@DBUF  :
1816 007770 012777 000077 006564      MOV   #63.,@CAR  :
1817 007776 112777 000044 006554      MOVB  #44,@DBUF  :
1818 010004 112777 000312 006546      MOVB  #312,@DBUF :
1819 010012 112777 000043 006540      MOVB  #43,@DBUF  :
1820 010020 105077 006542      CLRB  @CARX      :XYZ
1821 010024 113777 002170 006536      MOVB  MAX,@CARY  :
1822 010032 105377 006532      DECB  @CARY      :
1823 010036 112777 000042 006514      MOVB  #42,@DBUF  :
1824 010044 112777 000312 006506      MOVB  #312,@DBUF :
1825 010052 112777 000041 006500      MOVB  #41,@DBUF  :
1826 010060 112777 000077 006500      MOVB  #63.,@CARX :
1827 010066 113777 002170 006474      MOVB  MAX,@CARY  :
1828 010074 105377 006470      DECB  @CARY      :
1829 010100 112777 000052 006452      MOVB  #52,@DBUF  :
1830 010106 112777 000312 006444      MOVB  #312,@DBUF :
1831 010114 112777 000051 006436      MOVB  #51,@DBUF  :
1832 010122 052777 000004 006426      BIS   #4,@CSR    :SET INCREMENT BIT
1833 010130 113777 014516 006432      MOVB  HALF6V,@CARY :SET UP MESSAGE
1834 010136 162777 001000 006416      SUB   #1000,@CAR  :
1835 010144 112777 000033 006414      MOVB  #27.,@CARX  :
1836 010152 112777 000177 006400      MOVB  #177,@DBUF  :
1837 010160 112777 000000 006372      MOVB  #0,@DBUF    :
1838 010166 112777 000066 006364      MOVB  #'6,@DBUF   :
1839 010174 112777 000126 006356      MOVB  #'V,@DBUF   :
1840 010202 112777 000000 006350      MOVB  #0,@DBUF    :
1841 010210 112777 000000 006342      MOVB  #0,@DBUF    :
1842 010216 112777 000070 006334      MOVB  #'8,@DBUF   :
1843 010224 112777 000110 006326      MOVB  #'H,@DBUF   :
1844 010232 112777 000000 006320      MOVB  #0,@DBUF    :
1845 010240 112777 000177 006312      MOVB  #177,@DBUF  :
1846 010246 042777 000004 006302      BIC   #4,@CSR    :CLEAR INC BIT
1847 010254 052777 000001 006274      BIS   #1,@CSR    :TURN ON DISPLAY
  
```

1848	010262	012737	000500	004024	MOV	#500,TIME	:	
1849	010270	004737	003732		JSR	PC,WAIT	:	WAIT SHORT TIME


```

1851
1852 010274 005077 006256          CLR      @CSR      ;*****NOW LOAD SPECIAL CHARACTERS 8V 8H
1853 010300 112777 000001 006266  MOVB    #1,@CHAR  ;SET CHARACTER ADDRESS
1854 010306 012700 013456          MOV     #V88H1,RO
1855 010312 004737 012550          JSR    PC,LDCH8V  ;CHARACTER 1
1856 010316 004737 012550          JSR    PC,LDCH8V  ;          2
1857 010322 004737 012550          JSR    PC,LDCH8V  ;          3
1858 010326 004737 012550          JSR    PC,LDCH8V  ;          4
1859 010332 004737 012550          JSR    PC,LDCH8V  ;          5
1860 010336 004737 012550          JSR    PC,LDCH8V  ;          6
1861 010342 004737 012550          JSR    PC,LDCH8V  ;          7
1862 010346 004737 012550          JSR    PC,LDCH8V  ;         10
1863 010352 004737 012550          JSR    PC,LDCH8V  ;         11
1864 010356 004737 012550          JSR    PC,LDCH8V  ;         12
1865 010362 004737 012550          JSR    PC,LDCH8V  ;         13
1866
1867
1868
1869 010366 005737 014514          112014: TST     L525
1870 010372 001004
1871 010374 012737 000034 012570  BNE     1$
1872 010402 000403
1873 010404 012737 000042 012570  1$: MOV     #28.,HCNT ;SET FOR 625 LINES
1874 010412 013737 014520 002170  2$: MOV     #34.,HCNT ;SET FOR 525 LINES
1875 010420 006337 002170
1876 010424 005337 002170
1877
1878 010430 112777 000247 006122  MOVB    #247,@DBUF ;COLOUR WHITE ON BLUE
1879 010436 052777 000010 006112  BIS     #10,@CSR  ;DO PRESET
1880 010444 105777 006106          3$: TSTB    @CSR
1881 010450 100375
1882 010452 005077 006104
1883 010456 052777 000004 006072  BPL     @CAR
1884 010464 012700 000100
1885 010470 112777 000005 006062  4$: MOVB    #4,@CSR ;
1886 010476 005300
1887 010500 001373
1888 010502 042777 000004 006046  MOVB    #64.,RO ;LOAD ----- LINE
1889 010510 013700 014520
1890 010514 006300
1891 010516 112777 000004 006034  5$: DEC     RO ;CLEAR INC BIT
1892 010524 112777 000312 006026  MOVB    #4,@DBUF ;DO LINE FEED
1893 010532 005300
1894 010534 001370
1895 010536 005077 006020  MOVB    #312,@DBUF ;LOAD ] LINE
1896 010542 013700 014520
1897 010546 006300
1898 010550 112777 000003 006002  6$: BNE     5$ ;ZERO CURSOR ADDRESS
1899 010556 112777 000312 005774  CLR     @CAR
1900 010564 005300
1901 010566 001370
1902 010570 052777 000004 005760  MOV     HALF8V,RO ;DO LINE FEED
1903 010576 012700 000100
1904 010602 112777 000006 005750  7$: DEC     RO ;LOAD [ LINE
1905 010610 005300
1906 010612 001373

```

1907	010614	005077	005742			CLR	@CAR	:ZERO CURSOR ADDRESS
1908	010620	112777	000007	005732		MOVB	#7,@DBUF	:
1909	010626	105277	005736			INCB	@CARY	:LOAD TOP LEFT
1910	010632	112777	000001	005720	8\$:	MOVB	#1,@DBUF	:LOAD \ LINE
1911	010640	105277	005724			INCB	@CARY	:
1912	010644	127737	005720	014520		CMPB	@CARY,HALFBV	:
1913	010652	001367				BNE	8\$:
1914								
1915	010654	013700	012570			MOV	HCNT,RO	:
1916	010660	112777	000005	005672	9\$:	MOVB	#5,@DBUF	:LOAD ----- LINE
1917	010666	005300				DEC	RO	:
1918	010670	001373				BNE	9\$:
1919								
1920	010672	112777	000001	005660	10\$:	MOVB	#1,@DBUF	:LOAD \ LINE
1921	010700	105277	005664			INCB	@CARY	:
1922	010704	127737	005660	002170		CMPB	@CARY,MAX	:
1923	010712	001367				BNE	10\$:
1924	010714	112777	000011	005636		MOVB	#11,@DBUF	:LOAD BOTTOM RIGHT
1925								
1926	010722	105077	005640		T12015:	CLRB	@CARY	:ZERO X ADDRESS
1927	010726	113777	002170	005634		MOVB	MAX,@CARY	:SET Y ADDRESS
1928	010734	112777	000010	005616		MOVB	#10,@DBUF	:LOAD BOTTOM LEFT
1929	010742	105377	005622			DECB	@CARY	:
1930	010746	112777	000002	005604	1\$:	MOVB	#2,@DBUF	:LOAD / LINE
1931	010754	127737	005610	014520		CMPB	@CARY,HALFBV	:
1932	010762	001403				BEQ	2\$:
1933	010764	105377	005600			DECB	@CARY	:
1934	010770	000766				BR	1\$:
1935	010772	105377	005572		2\$:	DECB	@CARY	:
1936	010776	013700	012570			MOV	HCNT,RO	:
1937	011002	112777	000006	005550	3\$:	MOVB	#6,@DBUF	:LOAD ----- LINE
1938	011010	005300				DEC	RO	:
1939	011012	001373				BNE	3\$:
1940	011014	112777	000002	005536	4\$:	MOVB	#2,@DBUF	:LOAD / LINE
1941	011022	105377	005542			DECB	@CARY	:
1942	011026	001372				BNE	4\$:
1943	011030	112777	000012	005522		MOVB	#12,@DBUF	:LOAD TOP RIGHT
1944								
1945	011036	113777	014520	005524		MOVB	HALFBV,@CARY	:SET UP MESSAGE
1946	011044	162777	001000	005510		SUB	#1000,@CAR	:
1947	011052	112777	000033	005506		MOVB	#27,@CARX	:
1948	011060	112777	000177	005472		MOVB	#177,@DBUF	:
1949	011066	112777	000000	005464		MOVB	#0,@DBUF	:
1950	011074	112777	000070	005456		MOVB	#8,@DBUF	:
1951	011102	112777	000126	005450		MOVB	#V,@DBUF	:
1952	011110	112777	000000	005442		MOVB	#0,@DBUF	:
1953	011116	112777	000000	005434		MOVB	#0,@DBUF	:
1954	011124	112777	000070	005426		MOVB	#8,@DBUF	:
1955	011132	112777	000110	005420		MOVB	#H,@DBUF	:
1956	011140	112777	000000	005412		MOVB	#0,@DBUF	:
1957	011146	112777	000177	005404		MOVB	#177,@DBUF	:
1958								
1959	011154	042777	000004	005374		BIC	#4,@CSR	:CLEAR INCREMENT BIT
1960	011162	052777	000001	005366		BIS	#1,@CSR	:TURN ON DISPLAY
1961	011170	012737	000500	004024		MOV	#500,TIME	:
1962	011176	004737	003732			JSR	PC,WAIT	:WAIT SOME TIME

1963

```

1965 011202 004737 011224 JSR PC,LD6X6
1966 011206 012737 000500 004024 MOV #500,TIME
1967 011214 004737 003732 JSR PC,WAITT ;WAIT SOME TIME
1968 011220 000137 012142 JMP T12018
1969
1970 011224 012777 001400 005324 LD6X6: MOV #1400,@CSR ;*****NOW LOAD SPECIAL CHARACTERS 6V 6H
1971 011232 112777 000000 005334 MOV #0,@CHAR ;SET 6V & 6H BITS
1972 011240 012700 005372 MOV #PREO,R0 ;CHARACTER ZERO
1973 011244 004737 001746 JSR PC,LDCH ;LOAD WITH ALL 0'S
1974 011250 112777 000001 005316 MOV #1,@CHAR ;SET CHARACTER ADDRESS
1975 011256 012700 013362 MOV #V66H1,R0
1976 011262 004737 001746 JSR PC,LDCH ;CHARACTER 1
1977 011266 004737 001746 JSR PC,LDCH ; 2
1978 011272 004737 001746 JSR PC,LDCH ; 3
1979 011276 004737 001746 JSR PC,LDCH ; 4
1980 011302 004737 001746 JSR PC,LDCH ; 5
1981 011306 004737 001746 JSR PC,LDCH ; 6
1982 011312 004737 001746 JSR PC,LDCH ; 7
1983 011316 004737 001746 JSR PC,LDCH ; 10
1984 011322 004737 001746 JSR PC,LDCH ; 11
1985 011326 004737 001746 JSR PC,LDCH ; 12
1986 011332 004737 001746 JSR PC,LDCH ; 13
1987
1988 ;*****NOW START TO LOAD PICTURE
1989
1990 011336 005737 014514 T12016: TST L525
1991 011342 001004 BNE 1$
1992 011344 012737 000040 012570 MOV #32.,HCNT ;SET FOR 625 LINES
1993 011352 000403 BR 2$
1994 011354 012737 000050 012570 1$: MOV #40.,HCNT ;SET FOR 525 LINES
1995 011362 013737 014516 002170 2$: MOV HALF6V,MAX
1996 011370 006337 002170 ASL MAX
1997 011374 005337 002170 DEC MAX ;LAST Y ADDRESS
1998
1999 011400 112777 000247 005152 MOV #247,@DBUF ;COLOUR WHITE ON BLUE
2000 011406 052777 000010 005142 BIS #10,@CSR ;DO PRESET
2001 011414 105777 005136 3$: TSTB @CSR
2002 011420 100375 BPL 3$
2003 011422 005077 005134 CLR @CAR
2004 011426 052777 000004 005122 BIS #4,@CSR
2005 011434 012700 000120 MOV #80.,R0
2006 011440 112777 000005 005112 4$: MOV #5,@DBUF
2007 011446 005300 DEC R0 ;LOAD ----- LINE
2008 011450 001373 BNE 4$
2009 011452 042777 000004 005076 BIC #4,@CSR ;CLEAR INC BIT
2010 011460 013700 014516 MOV HALF6V,R0
2011 011464 006300 ASL R0
2012 011466 112777 000004 005064 5$: MOV #4,@DBUF
2013 011474 112777 000312 005056 MOV #312,@DBUF ;DO LINE FEED
2014 011502 005300 DEC R0 ;LOAD ] LINE
2015 011504 001370 BNE 5$
2016 011506 005077 005050 CLR @CAR ;ZERO CURSOR ADDRESS
2017 011512 013700 014516 MOV HALF6V,R0
2018 011516 006300 ASL R0
2019 011520 112777 000003 005032 6$: MOV #3,@DBUF
2020 011526 012777 000312 005024 MOV #312,@DBUF ;DO LINE FEED

```

2021	011534	005300				DEC	R0	:LOAD [LINE
2022	011536	001370				BNE	6\$:
2023	011540	052777	000004	005010		BIS	#4,@CSR	:SET INC BIT
2024	011546	012700	000120			MOV	#80.,R0	:
2025	011552	112777	000006	005000	7\$:	MOVB	#6,@DBUF	:
2026	011560	005300				DEC	R0	:LOAD ----- LINE
2027	011562	001373				BNE	7\$:
2028	011564	005077	004772			CLR	@CAR	:ZERO CURSOR ADDRESS
2029	011570	112777	000007	004762		MOVB	#7,@DBUF	:
2030	011576	105277	004766			INCB	@CARY	:LOAD TOP LEFT
2031	011602	112777	000001	004750	8\$:	MOVB	#1,@DBUF	:LOAD \ LINE
2032	011610	105277	004754			INCB	@CARY	:
2033	011614	127737	004750	014516		CMPB	@CARY,HALF6V	:
2034	011622	001367				BNE	8\$:
2035								
2036	011624	013700	012570			MOV	HCNT,R0	:
2037	011630	112777	000005	004722	9\$:	MOVB	#5,@DBUF	:LOAD ----- LINE
2038	011636	005300				DEC	R0	:
2039	011640	001373				BNE	9\$:
2040								
2041	011642	112777	0000C1	004710	10\$:	MOVB	#1,@DBUF	:LOAD \ LINE
2042	011650	105277	004714			INCB	@CARY	:
2043	011654	127737	004711	002170		CMPB	@CARY,MAX	:
2044	011662	001367				BNE	10\$:
2045	011664	112777	000011	004666		MOVB	#11,@DBUF	:LOAD BOTTOM RIGHT
2046								
2047	011672	105077	004670		T12017:	CLRB	@CARX	:ZERO X ADDRESS
2048	011676	113777	002170	004664		MOVB	MAX,@CARY	:SET Y ADDRESS
2049	011704	112777	000010	004646		MOVB	#10,@DBUF	:LOAD BOTTOM LEFT
2050	011712	105377	004652			DFCB	@CARY	:
2051	011716	112777	000002	004634	1\$:	MOVB	#2,@DBUF	:LOAD / LINE
2052	011724	127737	004640	014516		CMPB	@CARY,HALF6V	:
2053	011732	001403				BEQ	2\$:
2054	011734	105377	004630			DECB	@CARY	:
2055	011740	000766				BR	1\$:
2056	011742	105377	004622		2\$:	DECB	@CARY	:
2057	011746	013700	012570			MOV	HCNT,R0	:
2058	011752	112777	000006	004600	3\$:	MOVB	#6,@DBUF	:LOAD ----- LINE
2059	011760	005300				DEC	R0	:
2060	011762	001373				BNE	3\$:
2061	011764	112777	000002	004566	4\$:	MOVB	#2,@DBUF	:LOAD / LINE
2062	011772	105377	004572			DECB	@CARY	:
2063	011776	001372				BNE	4\$:
2064	012000	112777	000012	004552		MOVB	#12,@DBUF	:LOAD TOP RIGHT
2065								
2066	012006	113777	014516	004554		MOVB	HALF6V,@CARY	:SET UP MESSAGE
2067	012014	162777	001000	004540		SUB	#1000,@CAR	:
2068	012022	112777	000043	004536		MOVB	#35.,@CARX	:
2069	012030	112777	000177	004522		MOVB	#177,@DBUF	:
2070	012036	112777	000000	004514		MOVB	#0,@DBUF	:
2071	012044	112777	000066	004506		MOVB	#'6,@DBUF	:
2072	012052	112777	000126	004500		MOVB	#'V,@DBUF	:
2073	012060	112777	000000	004472		MOVB	#0,@DBUF	:
2074	012066	112777	000000	004464		MOVB	#0,@DBUF	:
2075	012074	112777	000066	004456		MOVB	#'6,@DBUF	:
2076	012102	112777	000110	004450		MOVB	#'H,@DBUF	:

```
2077 012110 112777 000000 004442      MOVB   #0,@DBUF      :  
2078 012116 112777 000177 004434      MOVB   #177,@DBUF   :  
2079                                     :  
2080 012124 042777 000004 004424      PIC    #4,@CSR      :CLEAR INCREMENT BIT  
2081 012132 052777 000001 004416      RIS    #1,@CSR      :TURN ON DISPLAY  
2082 012140 000207                                     :  
2083                                     :
```

```

2085 012142 104404          T12018: TRAP+4          ;SET SWR BIT 8 TO
2086 012144 005472          T12001          ;LOOP ON TEST
2087 012146 005337 016610    DEC          REPT I      ;DONE ENOUGH ?
2088 012152 001402          BEQ          T120WW      ;YES
2089 012154 000137 005472    JMP          T12001      ;NO
2090
2091 012160 032777 000100 004416 T120WW: BIT          #100, @SWR      ;CHECK FOR PRE-SELECTED TEST
2092 012166 001402          BIT          18
2093 012170 000137 001232    Jrr          START2
2094 012174 000137 013576    18:         JMP          TEST13
2095
2096
2097
2098 012200 112777 000001 004352 TLBR5BV: MOV B      #1, @DBUF
2099 012206 105277 004354          INCB        @CARX
2100 012212 112777 000002 004340    MOV B      #2, @DBUF
2101 012220 105277 004344          INCB        @CARY
2102 012224 112777 000003 004326    MOV B      #3, @DBUF
2103 012232 105277 004330          INCB        @CARX
2104 012236 112777 000004 004314    MOV B      #4, @DBUF
2105 012244 105277 004320          INCB        @CARY
2106 012250 112777 000005 004302    MOV B      #5, @DBUF
2107 012256 105277 004304          INCB        @CARX
2108 012262 112777 000006 004270    MOV B      #6, @DBUF
2109 012270 000207          RTS          PC
2110
2111 012272 112777 000016 004260 BLTRBV: MOV B      #16, @DBUF
2112 012300 105377 004262          DECB        @CARX
2113 012304 112777 000015 004246    MOV B      #15, @DBUF
2114 012312 105277 004252          INCB        @CARY
2115 012316 112777 000014 004234    MOV B      #14, @DBUF
2116 012324 105377 004236          DECB        @CARX
2117 012330 112777 000013 004222    MOV B      #13, @DBUF
2118 012336 105277 004226          INCB        @CARY
2119 012342 112777 000012 004210    MOV B      #12, @DBUF
2120 012350 105377 004212          DECB        @CARX
2121 012354 112777 000011 004176    MOV B      #11, @DBUF
2122 012362 000207          RTS          PC
2123 012364 112777 000016 004166 TLBR6V: MOV B      #16, @DBUF
2124 012372 105277 004172          INCB        @CARY
2125 012376 112777 000015 004154    MOV B      #15, @DBUF
2126 012404 105277 004156          INCB        @CARX
2127 012410 112777 000014 004142    MOV B      #14, @DBUF
2128 012416 105277 004146          INCB        @CARY
2129 012422 112777 000013 004130    MOV B      #13, @DBUF
2130 012430 105277 004132          INCB        @CARX
2131 012434 112777 000012 004116    MOV B      #12, @DBUF
2132 012442 105277 004122          INCB        @CARY
2133 012446 112777 000011 004104    MOV B      #11, @DBUF
2134 012454 000207          RTS          PC
2135
2136 012456 112777 000006 004074 BLTR6V: MOV B      #6, @DBUF
2137 012464 105277 004100          INCB        @CARY
2138 012470 112777 000005 004062    MOV B      #5, @DBUF
2139 012476 105377 004064          DECB        @CARX
2140 012502 112777 000004 004050    MOV B      #4, @DBUF

```

2141	012510	105277	004054			INCB	@CARY	:
2142	012514	112777	000003	004036		MOVH	#3,@DBUF	:
2143	012522	105377	004040			DECB	@CARX	:
2144	012526	112777	000002	004024		MOVH	#2,@DBUF	:
2145	012534	105277	004030			INCB	@CARY	:
2146	012540	112777	000001	004012		MOVH	#1,@DBUF	:
2147	012546	000207				RTS	PC	:
2148								:
2149	012550	012701	000010			LDCHBV: MOV	#10,R1	:
2150	012554	112077	004012			IS: MOVH	(R0)+,@CHDR	:
2151	012560	005301				DEC	R1	:
2152	012562	001574				BNE	IS	:
2153	012564	000207				RTS	PC	:
2154								:
2155	012566	000000				VCNT:	0	:
2156	012570	000000				HCNT:	0	:
2157								:
2158						.NLIST	BEX	:
2159	012572	104	104	174	TT2H:	.BYTE	104,104,174,104,104,104,0,0	:
2160	012602	104	104	104	T12V:	.BYTE	104,104,104,104,50,20,0,0	:
2161	012612	070	104	070	T128:	.BYTE	70,104,70,104,104,70,0,0	:
2162	012622	070	100	170	T126:	.BYTE	70,100,170,104,104,70,0,0	:
2163	012632	377	377	377	T12SLD:	.BYTE	377,377,377,377,377,377,377,377	:
2164								:
2165							:8V 6M CHARACTER SFT	:
2166	012642	200	100	040	V86M1:	.BYTE	200,100,40,20,10,4,0,0	:
2167	012652	000	000	000		.BYTE	0,0,0,0,0,0,200,100	:
2168	012662	040	020	010		.BYTE	40,20,10,4,0,0,0,0	:
2169	012672	000	000	000		.BYTE	0,0,0,0,200,100,40,20	:
2170	012702	010	004	000		.BYTE	10,4,0,0,0,0,0,0	:
2171	012712	000	000	200		.BYTE	0,0,200,100,40,20,10,4	:
2172								:
2173	012722	000	000	004	V86M11:	.BYTE	0,0,4,10,20,40,100,200	:
2174	012732	100	200	000		.BYTE	100,200,0,0,0,0,0,0	:
2175	012742	000	000	000		.BYTE	0,0,0,0,4,10,20,40	:
2176	012752	020	040	100		.BYTE	20,40,100,200,0,0,0,0	:
2177	012762	000	000	000		.BYTE	0,0,0,0,0,0,0,4,10	:
2178	012772	004	010	020		.BYTE	4,10,20,40,100,200,0,0	:
2179								:
2180	013002	374	000	000	V86M21:	.BYTE	374,0,0,0,0,0,0,0	:
2181	013012	000	000	000		.BYTE	0,0,0,0,0,0,0,374	:
2182	013022	200	200	200		.BYTE	200,200,200,200,200,200,200,200	:
2183	013032	004	004	004		.BYTE	4,4,4,4,4,4,4,4	:
2184								:
2185	013042	374	300	240	V86M41:	.BYTE	374,300,240,220,210,204,200,200	:
2186	013052	374	000	000		.BYTE	374,0,0,0,0,0,200,100	:
2187	013062	010	004	000		.BYTE	10,4,0,0,0,0,0,374	:
2188	013072	004	004	204		.BYTE	4,4,204,104,44,24,14,374	:
2189								:
2190	013102	200	200	204	V86M51:	.BYTE	200,200,204,210,220,240,300,374	:
2191	013112	100	200	000		.BYTE	100,200,0,0,0,0,0,374	:
2192	013122	374	000	000		.BYTE	374,0,0,0,0,0,0,10	:
2193	013132	374	014	024		.BYTE	374,14,24,44,104,204,4,4	:
2194								:
2195								:
2196	013142	004	010	020	V86M1:	.BYTE	4,10,20,40,100,200	:

2197	013150	000	000	000	.BYTE	0, 0, 0, 0, 1, 2
2198	013156	020	040	100	.BYTE	20, 40, 100, 200, 0, 0
2199	013164	000	000	001	.BYTE	0, 0, 1, 2, 4, 10
2200	013172	100	200	000	.BYTE	100, 200, 0, 0, 0, 0
2201	013200	001	002	004	.BYTE	1, 2, 4, 10, 20, 40
2202						
2203	013206	040	020	010	V68M11: .BYTE	40, 20, 10, 4, 2, 1
2204	013214	000	000	000	.BYTE	0, 0, 0, 0, 200, 100
2205	013222	010	004	002	.BYTE	10, 4, 2, 1, 0, 0
2206	013230	000	000	200	.BYTE	0, 0, 200, 100, 40, 20
2207	013236	002	001	000	.BYTE	2, 1, 0, 0, 0, 0
2208	013244	200	100	040	.BYTE	200, 100, 40, 20, 10, 4
2209						
2210	013252	200	200	200	V68M21: .BYTE	200, 200, 200, 200, 200, 200
2211	013260	001	001	001	.BYTE	1, 1, 1, 1, 1, 1
2212	013266	000	000	000	.BYTE	0, 0, 0, 0, 0, 377
2213	013274	377	000	000	.BYTE	377, 0, 0, 0, 0, 0
2214						
2215	013302	204	210	220	V68M41: .BYTE	204, 210, 220, 240, 300, 377
2216	013310	200	200	200	.BYTE	200, 200, 200, 200, 201, 202
2217	013316	101	201	001	.BYTE	101, 201, 1, 1, 1, 1
2218	013324	377	003	005	.BYTE	377, 3, 5, 11, 21, 41
2219	013332	041	021	011	V68M51: .BYTE	41, 21, 11, 5, 3, 377
2220	013340	001	001	001	.BYTE	1, 1, 1, 1, 201, 101
2221	013346	202	201	200	.BYTE	202, 201, 200, 200, 200, 200
2222	013354	377	300	240	.BYTE	377, 300, 240, 220, 210, 204
2223						
2224						:6V 6M CHARACTERS
2225	013362	200	100	040	V66M1: .BYTE	200, 100, 40, 20, 10, 4
2226	013370	004	010	020	.BYTE	4, 10, 20, 40, 100, 200
2227	013376	200	200	200	.BYTE	200, 200, 200, 200, 200, 200
2228	013404	004	004	004	.BYTE	4, 4, 4, 4, 4, 4
2229	013412	374	000	000	.BYTE	374, 0, 0, 0, 0, 0
2230	013420	000	000	000	.BYTE	0, 0, 0, 0, 0, 374
2231						
2232	013426	374	300	240	.BYTE	374, 300, 240, 220, 210, 204
2233	013434	204	210	220	.BYTE	204, 210, 220, 240, 300, 374
2234	013442	204	104	044	.BYTE	204, 104, 44, 24, 14, 374
2235	013450	374	014	024	.BYTE	374, 14, 24, 44, 104, 204
2236						
2237						:8V 8M CHARACTERS
2238	013456	200	100	040	V88M1: .BYTE	200, 100, 40, 20, 10, 4, 2, 1
2239	013466	001	002	004	.BYTE	1, 2, 4, 10, 20, 40, 100, 200
2240	013476	200	200	200	.BYTE	200, 200, 200, 200, 200, 200, 200, 200
2241	013506	001	001	001	.BYTE	1, 1, 1, 1, 1, 1, 1, 1
2242	013516	377	000	000	.BYTE	377, 0, 0, 0, 0, 0, 0, 0
2243	013526	000	000	000	.BYTE	0, 0, 0, 0, 0, 0, 0, 377
2244						
2245	013536	377	300	240	.BYTE	377, 300, 240, 220, 210, 204, 202, 201
2246	013546	201	202	204	.BYTE	201, 202, 204, 210, 220, 240, 300, 377
2247	013556	201	101	041	.BYTE	201, 101, 41, 21, 11, 5, 3, 377
2248	013566	377	003	005	.BYTE	377, 3, 5, 11, 21, 41, 101, 201
2249						
2250						
2251					.EVEN	
2252					.LIST	DEF

2253

```

2255 .SBTTL TEST13
2256
2257
2258
2259
2260
2261
2262 013576 104402 TEST13: TRAP+2
2263 013600 013614 T13000
2264
2265 013602 012737 000013 016602 MOV #13,TESTNO ;SET UP TEST NO
2266 013610 004737 016756 JSR PC,TESTR ;OUTPUT TEST NO
2267
2268 013614 005037 016706 T13000: CLR ERRDIS
2269 013620 012704 000001 MOV #1,R4
2270 013624 004737 017444 JSR PC,FASTSW
2271 013630 010437 016610 MOV R4,REPCNT ;SET UP TEST ITERATION COUNT
2272
2273 013634 013700 016576 T13001: MOV INTVEC,R0 ;*SET UP
2274 013640 012720 001356 MOV #TIME1,(R0)+ ;*INTERRUPT
2275 013644 012710 000340 MOV #340,(R0) ;*VECTOR
2276
2277 013650 004737 011224 JSR PC,LD6X6 ;LOAD PICTURE
2278
2279 013654 005077 002702 CLR @CAR ;ZERO CURSOR
2280 013660 052777 000002 002670 BIS #2,@CSR ;TURN ON CURSOR
2281
2282 013666 012737 000015 004024 T13002: MOV #15,TIME ;
2283 013674 004737 003732 JSR PC,WAITT ;SHORT WAIT
2284 013700 105277 002664 INCB @CARY ;
2285 013704 127737 002660 014516 CMPB @CARY,HALF6V ;
2286 013712 001403 BEQ 2$ ;
2287 013714 105277 002646 INCB @CARX ;
2288 013720 000762 BR T13002 ;
2289
2290 013722 004737 000015 004024 2$: MOV #15,TIME ;
2291 013730 004737 003732 JSR PC,WAITT ;SHORT WAIT
2292 013734 105277 002630 INCB @CARY ;
2293 013740 105377 002622 DECB @CARX ;
2294 013744 001366 BNE 2$ ;
2295
2296 013746 012737 000015 004024 3$: MOV #15,TIME ;
2297 013754 004737 003732 JSR PC,WAITT ;SHORT WAIT
2298 013760 105277 002602 INCB @CARX ;
2299 013764 127727 002576 000117 CMPB @CARX,#79. ;
2300 013772 001365 BNE 3$ ;
2301
2302 013774 012737 000015 004024 4$: MOV #15,TIME ;
2303 014002 004737 003732 JSR PC,WAITT ;SHORT WAIT
2304 014006 127737 002556 014516 CMPB @CARY,HALF6V ;
2305 014014 001405 BEQ 5$ ;
2306 014016 105377 002544 DECB @CARX ;
2307 014022 105377 002542 DECB @CARY ;
2308 014026 000762 BR 4$ ;
2309
2310 014030 105377 002534 5$: DECB @CARY ;
    
```

```

*****
: THIS TEST DEMONSTRATES THAT
: THE CURSOR CAN BE DISPLAYED
: AT ALL X & Y ADDRESSES
*****
    
```

```

2311 014034 012737 000015 004024 68:  MOV  #15,TIME      ;
2312 014042 004737 003732          JSR  PC,WAITT    ;SHORT 6
2313 014046 105277 002514          INCB @CARX       ;
2314 014052 105377 002512          DECB @CARY       ;
2315 014056 001366          BNE  68         ;
2316
2317 014060 012737 000015 004024 78:  MOV  #15,TIME      ;
2318 014066 004737 003732          JSR  PC,WAITT    ;SHORT WAIT
2319 014072 105377 002470          DECB @CARX       ;
2320 014076 001370          BNE  78         ;
2321
2322 014100 104404          TRAP+4          ;SET SWR BIT 8 TO
2323 014102 013666          T13002          ;LOOP ON TEST
2324 014104 005337 016610          DEC  REPCNT     ;DONE ENOUGH
2325 014110 001402          BEQ  T130WW     ;YES
2326 014112 000137 013666          JMP  T13002     ;NO
2327
2328 014116 032777 000100 002460 T130WW: BIT  #100,@SWR  ;CHECK FOR PRESFLECTED TEST
2329 014124 001402          BEQ  T130WY     ;
2330 014126 000137 001232          JMP  START2     ;
2331
2332 014132 000137 014136          T130WY: JMP  TEST14 ;
  
```

```

2334 .SBTTL TEST14
2335 ;*****
2336 ;X-MATCH FOR SETTING UP TV MONITOR
2337 ;*****
2338
2339 014136 104402 TEST14: TRAP+2 ;
2340 014140 014154 T14000 ;
2341 014142 012737 000014 016602 MOV #14,TESTNO ;SET UP TEST NUMBER
2342 014150 004737 016756 JSR PC,TESTR ;OUTPUT TEST NO.
2343
2344 014154 005037 016706 T14000: CLR ERDIS ;
2345 014160 012704 000001 MOV #1,R4 ;
2346 014164 004737 017444 JSR PC,FASTSW ;
2347 014170 010437 016610 MOV R4,REPCNT ;SET UP TEST ITERATION COUNT
2348
2349 014174 013700 016576 T14001: MOV INTVEC,RO ;
2350 014200 012720 001356 MOV #TIME1,(RO)+ ;
2351 014204 012710 000340 MOV #340,(RO) ;
2352
2353 014210 004737 021172 JSR PC,TYPOUT ;
2354 014214 014352 XMES ;
2355
2356 014216 005077 002334 CLR @CSR ;PICTURE SIZE 8 X 8 MATRIX
2357 014222 105077 002346 CLR @CHAR ;
2358 014226 012700 014342 MOV #XMATCH,RO ;
2359 014232 012705 000010 MOV #10,R5 ;
2360 014236 112077 002330 2$: MOV (RO)+,@CHDR ;LOAD CHARACTER ZERO
2361 014242 005305 DEC R5 ;
2362 014244 001374 BNE 2$ ;
2363
2364 014246 112777 000207 002304 MOV #207,@DBUF ;SET WHITE ON BLACK
2365 014254 112777 000300 002276 MOV #300,@DBUF ;CLEAR BLINK FLAG
2366 014262 052777 000010 002266 BIS #10,@CSR ;DO PRESET
2367 014270 105777 002262 1$: TSTB @CSR ;WAIT FOR READY
2368 014274 100375 BPL 1$ ;
2369 014276 052777 000001 002252 BIS #1,@CSR ;TURN ON DISPLAY
2370
2371 014304 012737 004000 004024 T14003: MOV #4000,TIME ;
2372 014312 004737 003732 JSR PC,WAITT ;WAIT SOME TIME
2373
2374 014316 104404 TRAP+4 ;SET SWR BIT 8 TO STAY
2375 014320 014304 T14003 ;ON THIS TEST PATTERN
2376
2377 014322 032777 000100 002254 BIT #100,@SWR ;CHECK FOR PRESELECTED TEST
2378 014330 001402 BEQ T14004 ;
2379 014332 000137 001232 JMP START2 ;
2380
2381 014336 000137 017224 T14004: JMP ENDIT
2382
2383
2384 014342 020 020 020 XMATCH: .BYTE 20,20,20,377,20,20,20,20
014345 377 020 020
014350 020 020
2385
2386
2387 014352 020133 026530 040510 XMES: .NLIST BEX
.ASCII /[ X-MATCH FOR SETTING UP TV MONITOR [a/
  
```

2388
2389
2390

.LIST BEX

.SBTTL UTILITY ROUTINES

```
2392
2393
2394
2395 014420 004737 021172          SET56: JSR      PC, TYP0UT      ;ASK 525 625
2396 014424 014522                ASK56                      ;
2397 014426 004737 020444          JSR      PC, READ         ;READ
2398 014432 020027 000065          CMP      RO, #'5          ;WAS IT 5
2399 014436 001012                BNE     SET56A           ;BRANCH IF NO
2400 014440 012737 177777 014514    MOV     #177777, L525     ;YES, SO SET FLAG
2401 014446 012737 000017 014520    MOV     #15., HALF8V     ;
2402 014454 012737 000024 014516    MOV     #20., HALF6V     ;
2403 014462 000207                RTS     PC                ;DONE
2404 014464 020027 000066          SET56A: CMP     RO, #'6   ;WAS IT 6
2405 014470 001353                BNE     SET56           ;BRANCH IF NO AND ASK AGAIN
2406 014472 005037 014514          CLR     L525             ;CLEAR FLAG
2407 014476 012737 000022 014520    MOV     #18., HALF8V     ;
2408 014504 012737 000030 014516    MOV     #24., HALF6V     ;
2409 014512 000207                RTS     PC                ;DONE
2410
2411 014514 000000          L525:  0
2412 014516 000030          HALF6V: 24.
2413 014520 000022          HALF8V: 18.
```

```

2416 .SBTTL ASCII STRINGS
2417
2418 .MLIST BEX
2419
2420 014522 020133 051511 052040 ASK56: .ASCII /[E IS THE CONTROLLER JUMPERED FOR 525 OR 625/
2421 014575 040 044514 042516 .ASCII / LINES,[ TYPE 5 OR 6 .....@/
2422 014630 053133 053124 030063 GOMSG: .ASCII ![VTV30-H/J OR VT30-H DIAGNOSTIC AND'
2423 014673 105 042530 041522 .ASCII /EXERCISER/
2424 014704 050133 051101 031124 .ASCII /[PART2----- DISPLAY TESTS/
2425 014735 133 052123 051101 .ASCII /[START ADDRESS IS 1000, OR 200/
2426 014773 133 042522 052123 .ASCII /[RESTART ADDRESS IS 1200/
2427 015023 133 047531 020125 .ASCII /[YOU ARE STRONGLY ADVISED TO READ THE/
2428 015070 047533 052120 047511 .ASCII /[OPTION DESCRIPTION BEFORE PROCEEDING,/
2429 015136 040533 054516 043040 .ASCII /[ANY FURTHER./
2430 015153 133 047506 020122 .ASCII /[FOR DETAILS OF BUS AND VECTOR ADDRESSES/
2431 015223 133 046120 040505 .ASCII /[PLEASE REFER TO THE OPTION DESCRIPTION/
2432 015272 055533 040133 .ASCII /[[[@/
2433 015276 051533 046105 041505 WMSG: .ASCII /[SELECT DESIRED SWITCH REGISTER OPTIONS@/
2434 015347 133 042524 052123 TESMSG: .ASCII /[TEST NO@/
2435 015360 042533 042116 047440 PASMSG: .ASCII /[END OF PASS@/
2436 015376 052133 040522 020120 ILVMSG: .ASCII /[TRAP TO ILLEGAL VECTOR @/
2437 015430 043133 047522 020115 FRMMSG: .ASCII /[FROM ADDRESS @/
2438 015450 043133 051117 052040 FIRVMS: .ASCII /[FOR THE FIRST VECTOR GROUP@
2439 015504 043133 051117 052040 NXTVMS: .ASCII /[FOR THE NEXT VECTOR GROUP@
2440 015537 133 047516 020116 NOMEMA: .ASCII /[NON EXISTANT MEMORY HAS NOT BEEN FOUND@/
2441
2442 015607 133 054524 042520 REDMES: .ASCII /[TYPE CNTRL-C TO CONTINUE @/
2443 015642 051533 051127 036440 SWRMSG: .ASCII /[SWR = @/
2444 015654 022133 020040 100 MODADM: .ASCII /[$ @/
2445 015661 040 020057 100 MODSPA: .ASCII / ? / @?
2446 015665 133 020052 040040 MODPRM: .ASCII /[* @/
2447 015672 043133 051111 052123 BAMSG: .ASCII /[FIRST BUS ADDRESS IS .....@/
2448 015726 043133 051111 052123 VAMSG: .ASCII /[FIRST VECTOR ADDRESS IS .....@/
2449 015765 133 047111 040526 UDAMSG: .ASCII /[INVALID ADDRESS @/
2450 016010 040533 042104 042522 OVAMSG: .ASCII /[ADDRESS EXCEEDS 772 @/
2451 016037 133 044506 051522 PRMSG: .ASCII /[FIRST PRIORITY LEVEL IS .....@/
2452 016076 047133 054105 020124 NPRMSG: .ASCII /[NEW PRIORITY LEVEL IS .....@/
2453 016135 133 047516 020116 NXMSG: .ASCII /[NON EXISTANT ADDRESS @/
2454 016165 133 047111 040526 BADPRI: .ASCII /[INVALID PRIORITY, PLEASE RE-ENTER/
2455 016227 133 044124 020105 .ASCII /[THE PRIORITY.....@/
2456 016252 042133 043105 052501 NODEFM: .ASCII /[DEFAULT SETTINGS ARE NOT ALLOWED/
2457 016313 133 046120 040505 .ASCII /[PLEASE RE-ENTER THE VALUE .....@/
2458 016354 042533 040043 EMSG1: .ASCII /[E#@/
2459 016360 020040 052101 050040 EMSG2: .ASCII / AT PC @/
2460 016372 043533 047517 020104 EMSG3: .ASCII /[GOOD : @/
2461 016403 040 020040 040502 EMSG4: .ASCII / BAD :@/
2462 016414 042133 052101 020101 EMSG5: .ASCII /[DATA : @/
2463 016425 040 020040 042101 EMSG6: .ASCII / ADDRESS :@/
2464 016443 133 052123 052101 EMSG7: .ASCII /[STATUS :@/
2465 016456 041533 046101 042514 EMSG8: .ASCII /[CALLED FROM :@/
2466 016475 040 020040 051105 EMSG9: .ASCII / ERROR COUNT = @/
2467 .EVEN
2468 016520 000000 000000 000000 OCTNUM: .WORD 0,0,0
2469 016526 100 000 .BYTE 100,0
2470 016530 040502 042523 030061 BASE11: .ASCII /BASE10@/
2471 016540 .EVEN

```


2472	016540	000000	000000	000000	DECMMSG: .WORD	0,0,0
2473	016546	000000	000000	000000	OCTMSG: .WORD	0,0,C,0
2474					.EVEN	
2475					.LIST	PEX

2477
2478
2479 016556 164000
2480 016560 164002
2481 016562 164004
2482 016564 164006
2483 016566 164004
2484 016570 164005
2485 016572 164006
2486 016574 164007
2487 016576 000170
2488 016600 000004
2489 016602 000000
2490 016604 177570
2491 016606 000000
2492 016610 000000
2493 016612 000137 001000
2494 016616 000001
2495 016620 000000
2496 016622 000000
2497 016624 000000
2498 016626 000000
2499 016630 000000
2500 016632 000000
2501 016634 000000
2502 016636 000000
2503 016640 000000
2504 016642 000000
2505 016644 000000
2506 016646 000000
2507 016650 000000
2508 016652 000000
2509 016654 000000
2510 016656 000000
2511 016660 000000
2512 016662 000000
2513 016664 052525
2514 016666 000001
2515 016670 000006
2516 016672 052525
2517 016674 000000
2518 016676 000000
2519 016700 000000
2520 016702 000000
2521 016704 000000
2522 016706 000000
2523 016710 000000
2524 016712 000000
2525 016714 000000
2526 016716 000000
2527 016720 000000
2528 016722 000000
2529 016724 000000
2530 016726 000000
2531 016730 000000
2532 016732 000000

.SBTTL PROGRAM VARIABLES

CSR: 164000
DBUF: 164002
CAR: 164004
CHSR: 164006
CARX: 164004
CARY: 164005
CMDR: 164006
CHAR: 164007
INTVEC: 170
VECLEV: 4
TESTNO: 0
SWR: HSWR
SSWR: 0
REPCNT: 0
JM600: JMP @#START
FSTCNT: 1
TRPARG: 0
TRPSEL: 0
TRPMEM: 0
SAVPC: 0
SAVPC1: 0
SAVSTA: 0
TYPOTA: 0
RAND: 0
TYPD1: 0
MODADR: 0
MODSAV: 0
MASK: 0
BASADD: 0
TRPERR: 0
CHWORD: 0
ERR: 0
PARITY: 0
BCCHAR: 0
RANDN: 52525
RANSEL: 1
RANMTA: 6
RANST: 52525
GOOD: 0
BAD: 0
DATA: 0
STATUS: 0
ADDRES: 0
ERRDIS: 0
ERRARG: 0
CALLPC: 0
TRXVAD: 0
TRXPAR: 0
TRXEXM: 0
NXMADR: 0
SAVEXM: 0
SAVPAR: 0
SAVLT4: 0
SAVLT0: 0

2533	016734	000000	CNVFLG: 0
2534	016736	000000	STRADD: 0
2535	016740	000000	STRLEN: 0
2536	016742	000000	LOWCHR: 0
2537	016744	000000	UPPCHR: 0
2538	016746	000000	RUBFLG: 0
2539	016750	000000	RANDC: 0
2540	016752	000000	FSVPM: C
2541	016754	000000	BCOUNT: 0

2543
 2544
 2545
 2546
 2547
 2548
 2549
 2550
 2551
 2552
 2553
 2554
 2555
 2556
 2557
 2558
 2559
 2560
 2561
 2562
 2563
 2564
 2565
 2566
 2567
 2568
 2569
 2570
 2571
 2572
 2573
 2574
 2575
 2576
 2577
 2578
 2579
 2580
 2581
 2582
 2583
 2584
 2585
 2586
 2587
 2588
 2589
 2590
 2591
 2592 016756 004737 021172
 2593 016762 015347
 2594 016764 013700 016602
 2595 016770 004737 021262
 2596 016774 004737 021136
 2597 017000 000207

```

.SBTL PRINT TEST NUMBER

DESCRIPTION:
    ROUTINE TO PRINT THE TEST NUMBER IN OCTAL

CALLING SEQUENCE:
    JSR    PC,TESTR

INPUT PARAMETERS:
    TESTNO CONTAINS THE OCTAL TEST NUMBER TO BE
    PRINTED

IMPLICIT INPUT PARAMETERS:
    THE LABEL TESMSG IS THE START ADDRESS OF AN
    ASCII STRING 'TEST NO'

OUTPUT PARAMETERS:
    RO WILL BE CORRUPTED

IMPLICIT OUTPUT PARAMETERS:
    THE MESSAGE 'TEST NO N' WILL BE PRINTED ON THE
    CONSOLE TERMINAL

COMPLETION CODES:
    NONE

POSSIBLE ERROR CODES:
    NONE

TESTR: JSR    PC,TYPEOUT
        TESMSG ;TEST NO
        MOV    TESTNO,RO
        JSR    PC,PROCT ;PRINT OCTAL TEST NO
        JSR    PC,CRLF
        RTS    PC ;EXIT
  
```

2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654

.SBTTL 'SILLSI' SUBROUTINE

DESCRIPTION:

ROUTINE TO ESTABLISH WHETHER OR NOT THE
DIAGNOSTIC IS RUNNING ON A PROCESSOR
WHICH POSSESSES ONLY ONE INTERRUPT BUS
PRIORITY LEVEL.

CALLING SEQUENCE:

JSR PC,SILLSI

INPUT PARAMETERS:

NONE

IMPLICIT INPUT PARAMETERS:

NONE

OUTPUT PARAMETERS:

THE VARIABLE 'LSIFLG' WILL BE SET UP TO
REFLECT WHETHER OR NOT THE PROCESSOR HAS
A SINGLE INTERRUPT PRIORITY LEVEL.

LSIFLG = 0 => MULTIPLE INT. PRIORITIES
<> 0 => SINGLE INT. PRIORITY

IMPLICIT OUTPUT PARAMETERS:

NONE

COMPLETION CODES:

NONE

POSSIBLE ERROR CODES:

NONE

017002 005037 017044
017006 012737 017032 000004
017014 012737 000340 000006

SILLSI: CLR LSIFLG ;SET UP FLAG FOR NON-LSI
MOV #15,4 ;INSTALL TRAP THRU 4 VECTOR
MOV #340,6 ;AND CORRESPONDING PRIORITY

```
2655  
2656 017022 005737 177776          :          TST     PSW          ;TRY ADDRESSING THE PSW --  
2657 017026 000240                  :          NOP                    ;IF NOT THERE, TRAP THRU 4  
2658 017030 000404                  :          BR      28            ;WILL OCCUR, ELSE BRANCH.  
2659  
2660 017032 022626                  :          CMP     (SP)+,(SP)+    ;PERFORM A DUMMY RTI  
2661 017034 012737 177777 017044  :          MOV     #-1,LSIFLG     ;PROCESSOR HAS ONE INT.LEVEL  
2662  
2663 017042 000207                  :          RTS     PC            ;RETURN TO MAINLINE CALL.  
2664  
2665  
2666  
2667 017044 000000                  :          LSIFLG: 0            ;0 => MULTIPLE INT.PRIORITIES  
2668  
2669  
2670  
2671
```

2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726

.SBTTL NON EXISTANT SWR TRAP

DESCRIPTION:

THE TRAP WHEN TESTING FOR THE HARDWARE
SWITCH REGISTER WILL OCCUR MFRE
THE LOCATION SWR WILL BE SET TO CONTAIN THE
ADDRESS OF THE SOFTWARE SWITCH REGISTER SSWR

ENTRY POINT

SWRSET

INPUT PARAMETERS:

OCCURS IF A HARDWARE SWITCH REGISTER IS NOT
PRESENT

IMPLICIT INPUT PARAMETER.:

NONE

OUTPUT PARAMETERS:

THE LOCATION SWR WILL BE SET TO CONTAIN SSWR

IMPLICIT OUTPUT PARAMETERS:

NONE

COMPLETION CODES:

AN RTI IS PERFORMED

POSSIBLE ERROR CODES:

NONE

017046 012737 016606 016604 SWRSET: MOV #SSWR,SWR
017054 000002 RTI

2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783

.SBTTL SET UP ILLEGAL VECTOR TRAPS

DESCRIPTION:

ROUTINE TO SET CATCHES FOR TRAPS TO ILLEGAL VECTORS IN THE RANGE 0 TO 772, DURING THE RUNNING OF THE TESTS.

THE CATCH IS TO FORCE THE EXECUTION OF AN IOT TRAP.

THE VECTOR 14 (ODT VECTOR) IS LEFT FREE, 34 (TRAP VECTOR) IS SET TO THE ADDRESS TRAPSV TO SERVICE THE TRAP INSTRUCTION. THE VECTOR 20 (IOT) IS SET TO ILLVEC TO SERVICE ILLEGAL VECTOR TRAPS, AND THE ADDRESSES 200 AND 202 ARE SET WITH A JUMP TO START, THUS ALLOWING THE BE BE RESTARTED FROM ADDRESS 200. LOCATIONS 30 AND 32 AND SET TO CATCH EMT CALLS AND HENCE READ THE PROCESSOR PRIORITY.

LOCATION 40 IS LEFT FREE TO CONTAIN THE LOAD INDICATORS, LOCATION 42 IS LEFT FREE TO CONTAIN THE XXDP RETURN ADDRESS (IF PRESENT). LOCATION 46 IS SET TO POINT TO A RETURN TO XXDP AND LOCATION 52 IS SET TO ZERO.

CALLING SEQUENCE:

JSR PC,VECTOR

INPUT PARAMETERS:

NONE

IMPLICIT INPUT PARAMETERS:

NONE

OUTPUT PARAMETERS:

R0 AND R1 WILL BE CORRUPTED

ADDRESSES 0 THRU TO 774 WILL BE SET WITH APPROPRIATE VALUES

IMPLICIT OUTPUT PARAMETERS:

NONE

2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822

017056 005000
017060 012701 000002
017064 020027 000014
017070 001002
017072 022020
017074 000410
017076 020027 000040
017102 001002
017104 022020
017106 000403
017110 010120
017112 012720 000004
017116 062701 000004
017122 020027 000774
017126 002756
017130 012737 017464 000034
017136 012737 000340 000036
017144 012737 017314 000030
017152 012737 000340 000032
017160 012737 017364 000020
017166 012737 000340 000022
017174 013737 016612 000200
017202 013737 016614 000202
017210 012737 017300 000046
017216 005037 000052
017222 000207

```

COMPLETION CODES:
MCNE

POSSIBLE ERROR CODES:
NONE

VECTOR: CLR R0 ;FILL 0-572 WITH IOT TRAPS
MOV #2,R1
FILL: CMP R0,#14 ;ODT TRAP?
BNE 1$ ;YES BUMP R0
CMP (R0)+,(R0)+ ;YES BUMP R0
BR FILL1
1$: CMP R0,#40 ;XXDP RETURN ADDRESS
BNE 2$ ;NO
CMP (R0)+,(R0)+ ;YES BUMP R0
BR FILL1
2$: MOV R1,(R0)+ ;'+2'
MOV #4,(R0)+ ;'IOT'
FILL1: ADD #4,R1
CMP R0,#774
BLT FILL
FILL2: MOV #TRAPSV,34 ;TRAP (LOOP CONTROL)
MOV #340,36
MOV #ADR,30 ; PLUG EMT FOR READING
MOV #340,32 ; THE PROCESSOR STATUS
MOV #ILLVEC,20 ;PLUG 20 FOR IOTS
MOV #340,22
MOV JM600,200 ;SET UP JMP START IN LOC 200
MOV JM600+2,202
MOV #SENDAD,46 ;POINT TO RETURN TO XXDP
CLR 52 ;CLEAR 52
RTS PC ;THEN EXIT
    
```

2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879

.SBTTL XXDP END OF PASS HOOKS

DESCRIPTION:

ROUTINE TO SIGNIFY END OF PASS, AND IF THE PROGRAM HAS BEEN LOADED USING AN XXDP MONITOR A CALL WILL BE MADE BACK TO THE MONITOR. THE LOCATIONS USED BY XXDP ARE 40, AND 41 FOR THE LOAD MEDIUM AND 42, 43 FOR THE RETURN ADDRESS.
IF A PRESELECTED TEST IS IN OPERATION THE PROGRAM WILL GO AND SELECT THAT TEST.

ENTRY POINT:

ENDIT

INPUT PARAMETERS:

LOCATION 42/43 CONTAINS THE XXDP RETURN ADDRESS

IMPLICIT INPUT PARAMETERS:

NONE

OUTPUT PARAMETERS:

NONE

IMPLICIT OUTPUT PARAMETERS:

NONE

COMPLETION CODES:

IF LOCATIONS 42/43 ARE NON ZERO THEY ARE ASSUMED TO CONTAIN THE XXDP MONITOR RETURN ADDRESS

POSSIBLE ERROR CODES:

NONE

```
2880  
2881  
2882 017224 012706 001000  
2883 017230  
2884 017244 032777 000100 177332  
2885 017252 001402  
2886 017254 000137 001232  
2887  
2888  
2889  
2890 017260 004737 021172  
2891 017264 015360  
2892 017266 013700 000042  
2893 017272 001002  
2894 017274 000137 001232  
2895  
2896  
2897  
2898 017300 004710  
2899 017302 000240  
2900 017304 000240  
2901 017306 000240  
2902 017310 000137 001232  
2903  
2904  
2905
```

```
ENDIT: MOV #START,SP ; RESET THE STACK  
        PSWSET #340 ; AND PROCESSOR PRIORITY  
        BIT #100,@SWR ; IS THERE A PRESELECT ON  
        BEQ 1$ ; NO  
        JMP START2 ; YES GO SELECT THE TEST  
  
; NO PRESELECT ON SO SIGNAL END OF PASS  
1$: JSR PC,TYPEOUT ; END OF PASS  
    PASMMSG  
    MOV @#42,R0 ; GET RETURN ADDRESS TO XXDP  
    BNE $ENDAD ; IF IT IS ZERO THERE IS NO  
    JMP START2 ; MONITOR SO RESTART DIAG  
  
$ENDAD: JSR PC,(R0) ; CALLED VIA XXDP SO RETURN  
        NOP ; THERE  
        NOP  
        NOP ; ALLOW A SLIGHT PAUSE  
        JMP START2 ; THEN RESTART THE DIAGNOSTIC
```

2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947

.SBTTL READ PROCESSOR PRIORITY

DESCRIPTION:

THIS IS THE EMT HANDLER AND IS USED TO READ
THE PROCESSOR PRIORITY OFF THE STACK AND RETURNING
IT IN FSAVPW

CALLING SEQUENCE:

CALLED BY ISSUING AN EMT

INPUT PARAMETERS:

NONE

IMPLICIT INPUT PARAMETERS:

THE LOCATIONS 30 AND 32 MUST HAVE BEEN SET
UP TO POINT TO THIS ROUTINE

OUTPUT PARAMETERS:

THE CONTENTS OF THE PROCESSOR PRIORITY
ARE RETURNED IN THE LOCATION FSAVPW

COMPLETION CODES:

NONE

POSSIBLE ERROR CODES:

NONE

017314	016637	000002	016752	FADR:	MOV	2(S0),FSAVPW	; READ PRIORITY
017322	000002				RTI		; RETURN TO CALLER

2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998

.SBTTL RING TTY BELL

DESCRIPTION:

ROUTINE TO RING THE BELL ON THE CONSOLE
 TERMINAL, IF BIT 5 IS SET IN THE SWITCH
 REGISTER.

CALLING SEQUENCE:

JSR PC,BELL

INPUT PARAMETERS:

NONE

IMPLICIT INPUT PARAMETERS:

THE SWITCH REGISTER MUST HAVE BEEN
 SET UP

OUTPUT PARAMETERS:

THE TELETYPE BELL WILL BE RUNG IF
 APPROPRIATE

IMPLICIT OUTPUT PARAMETERS:

NONE

COMPLETION CODES:

NONE

POSSIBLE ERROR CODES:

NONE

2999	017324	032777	000040	177252	BELL:	BIT	#40,@SWR	
3000	017332	001401				BEQ	BELL1	
3001	017334	000207				RTS	PC	
3002	017336	004737	017434		BELL1:	JSR	PC,TPREDY	;WAIT FOR PRINTER READY
3003	017342	112737	000007	177566		MOVB	#7,TPB	
3004	017350	004737	017434			JSR	PC,TPREDY	;WAIT FOR PRINTER READY.

3005	017354	11273	000000	177566	MOVB	#0,TPB	:PRINT NULL
3006	017362	000207			RIS	PC	:GO OUT

3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063

.SB*TL ILLEGAL VECTOR TRAP CATCH

DESCRIPTION:

TRAPS TO ILLEGAL VECTORS WILL BE REPORTED HERE. THE VECTOR TO WHICH THE TRAP OCCURRED WILL BE PRINTED AS WELL AS THE ADDRESS IN THE MAIN LINE CODE FROM WHICH THE TRAP OCCURRED. A PROGRAM RESTART IS THEN PERFORMED, UNLESS A NEW TEST HAS BEEN SELECTED WHILE RUNNING UNDER A SOFTWARE SWITCH REGISTER.

ENTRY POINT

ILLVEC

INPUT PARAMETERS:

ENTRY IS CAUSED BY AN ILLEGAL VECTOR TRAP

IMPLICIT INPUT PARAMETERS:

NONE

OUTPUT PARAMETERS:

THE VECTOR AND MAINLINE ADDRESS WILL BE PRINTED

IMPLICIT OUTPUT PARAMETERS:

NONE

COMPLETION CODES:

A PROGRAM RESTART OR A NEW TEST SELECTION .

POSSIBLE ERROR CODES:

NONE

```
3064  
3065  
3066  
3067  
3068  
3069  
3070 017364 004737 021172  
3071 017370 015376  
3072 017372 012600  
3073 017374 162790 000004  
3074 017400 004737 021262  
3075 017404 004737 021172  
3076 017410 015430  
3077 017412 005726  
3078 017414 012600  
3079 017416 004737 021262  
3080 017422 005726  
3081 017424 004737 017704  
3082 017430 000137 001200
```

```
;  
:  
:  
: A TRAP TO AN UNRECOGNISED VECTOR WILL  
: BE REPORTED FROM HERE.  
:  
ILLVEC: JSR PC, TYP0UT  
ILVMSG  
MOV (SP)+, R0  
SUB #4, R0  
JSR PC, PROCT ;PRINT VECTOR  
JSR PC, TYP0UT  
FRMSG ;PRINT MAINLINE ADDRES  
TST (SP)+  
MOV (SP)+, R0  
JSR PC, PROCT  
TST (SP)+  
JSR PC, MONIT  
JMP RSTART
```


3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133

.SBTTL WAIT FOR PRINTER READY

DESCRIPTION:

ROUTINE TO WAIT UNTIL THE PRINTER
ON THE CONSOLE TERMINAL IS READY,
IE: IT IS READY TO PRINT THE NEXT
CHARACTER.

CALLING SEQUENCE:

JSR PC,TPREDY

INPUT PARAMETERS:

NONE

IMPLICIT INPUT PARAMETERS:

NONE

OUTPUT PARAMETERS:

NONE

IMPLICIT OUTPUT PARAMETERS:

NONE

COMPLETION CODES:

NONE

POSSIBLE ERROR CODES:

NONE

017434 105737 177564
017440 100375
017442 000207

TPREDY: TSTB TPS ;ROUTINE TO WAIT FOR PRINTER READY
BPL TPREDY
RTS PC

3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190

.SBTTL SET ITERATION COUNT

DESCRIPTION:

ROUTINE TO SET UP THE TEST ITERATION
COUNT. A PROPOSED COUNT IS SET IN R4
THEN UNLESS BIT 13 IN THE SWITCH REGISTER
IS SET, THE SAME VALUE IS RETURNED.
IF BIT 13 IS SET THEN FAST ITERATION IS
ASSUMED AND A VALUE OF 1 IS RETURNED IN
R4.

CALLING SEQUENCE:

JSR PC,FASTSW

INPUT PARAMETERS:

R4 CONTAINS THE PROPOSED ITERATION
COUNT

IMPLICIT INPUT PARAMETERS:

SETTING BIT 13 IN THE SWR WILL INDICATE
FAST ITERATION, AND A SINGLE PASS
WILL BE REQUESTED

OUTPUT PARAMETERS:

R4 WILL CONTAIN THE ACTUAL ITERATION
COUNT

IMPLICIT OUTPUT PARAMETERS:

NONE

COMPLETION CODES:

NONE

POSSIBLE ERROR CODES:

NONE

017444 032777 020000 177132 FASTSW: BIT #20000,@SWR

3191	017452	001001		BNE	18
3192	017454	000207		RTS	PC
3193	017456	013704	016616	MOV	FSTCNT,R4
3194	017462	000207		RTS	PC

3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251

.SB7L TRAP SERVICE ROUTINE

DESCRIPTION:

TRAP HANDLING ROUTINE. THE TRAP HANDLER IS ENTERED UPON THE EXECUTION OF ANY TRAP INSTRUCTION. IT COMPARES THE LOWER BYTE OF THE TRAP INSTRUCTION WITH THE CONTENTS ON THE SWITCH REGISTER, AND IF A MATCH IS FOUND TAKES THE CONTENTS OF THE ADDRESS FOLLOWING THE TRAP INSTRUCTION AS THE RETURN ADDRESS.

THE EXPECTED FORMAT IS:

TRAP+N
ADDR

WHERE ADDR IS THE ADDRESS TO PROCEED TO IF N MATCHES THE SWITCH REGISTER SETTINGS. THE TRAP ARGUMENT IS RELATED TO THE SWITCH REGISTER SETTINGS THUS:

TRAP ARG	SWR SETTING
2	200
4	400
10	1000
20	2000
30	3000
40	4000
50	5000
60	6000
70	7000

THE SETTING OF SWR BIT 12, WILL FORCE THE TRAP HANDLER TO USE THE SWR SETTINGS THAT WERE IN FORCE WHEN THE LAST TRAP INSTRUCTION WAS EXECUTED.

IF A CNTRL-G IS OUTSTANDING ON THE CONSOLE TERMINAL WHEN THE TRAP WAS EXECUTED, THEN MONIT IS CALLED. IF CNTRL-O WAS OUTSTANDING THEN MODIFY IS CALLED

ENTRY POINT

TRAPSV

INPUT PARAMETERS:

3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307

017464 004737 020076
017470 011600
017472 016037 177776 016620
017500 105737 177560
017504 100015
017506 004737 020444
017512 120027 000017
017516 001003
017520 004737 021522
017524 000405
017526 120027 000007 1\$
017532 001002
017534 004737 017704
017540 017737 177040 016622 3\$
017546 132737 000020 016623
017554 001012
017556 013737 016622 016624
017564 042737 170777 016624
017572 132737 000016 016623
017600 001412
017602 013700 016620
017606 006000
017610 006000
017612 000300

```

ON EXECUTION OF ANY TRAP INSTRUCTION

IMPLICIT INPUT PARAMETERS:

    THE SWR HAS BEEN SET UP

OUTPUT PARAMETERS:

    EXIT TO THE CONTENTS OF THE ADDRESS FOLLOWING
    THE TRAP INSTRUCTION IF A MATCH WAS FOUND, ELSE THE
    PROGRAM WILL CONTINUE FROM THE ADDRESS AFTER THAT.

IMPLICIT OUTPUT PARAMETERS:

    NONE

COMPLETION CODES:

    NONE

POSSIBLE ERROR CODES:

    NONE

TRAPSV: JSR    PC,SAVREG    ;SAVE REGS
        MOV    (SP),R0
        MOV    -2(R0),TRPARG ;GET TRAP CALL
        TSTB   'KS        ;LOOK FOR MONITOR CALL
        BPL   'S
        JSR   FC,READ
        CMPB  R0,#17      ;IS IT CNTRL-O ?
        BNE   'S          ;NO
        JSR   PC,MODIFY   ;YES CALL MODIFIER ROUTINE
        BR    'S
        CMPB  R0,#7      ;IS IT CTRL-G?
        BNE   'S
        JSR   PC,MONIT    ;YES, GO TO SWR MONITOR
        MOV   @SWR,TRPSEL
        BITB  #20,TRPSEL+1 ;IS IT PRESERVE SCOPE
        BNE   TRPSCP      ;YES
        MOV   TRPSEL,TRPMEM ;NO SO SAVE SWITCH SETTING
        BIC   #170777,TRPMEM ;GET IT SO WE CAN COMPARE
        BITB  #16,TRPSEL+1 ;ANY SCOPE LEVELS SET
        BEQ   TOPLP       ;NO
        TRPSCP: MOV   TRPARG,R0 ;YES
        ROR   R0          ;GET TO POSITION FOR COMPARE
        ROR   R0          ;FOR 10 & ABOVE
        SWAB  R0          ;ONLY FOR SCOPE BITS(9-11)
    
```

```

3308 017614 042700 170777          BIC      #170777,RO      ;ARGUMENT SCOPE BITS
3309 017620 020037 016624          CMP      RO,TRPMEM    ;AS SELECTED ?
3310 017624 001422                   BEQ      TRPBAK       ;YES, GO BACK
3311 017626 013700 016622          TRPLP:  MOV      TRPSEL,RO ;NO, TEST LOOP
3312 017632 006100                   ROL      RO          ;FOR BITS 7&8
3313 017634 006100                   ROL      RO
3314 017636 000300                   SWAB     RO
3315 017640 142700 000371          BICB    #371,RO      ;CUT OUT SW06
3316 017644 142737 000370 016620  BICB    #370,TRPARG   ;ONLY SWITCHES 7 OR 8 NOW
3317 017652 130037 016520          BITB    RO,TRPARG    ;ANY SELECTED ?
3318 017656 001005                   BNE     TRPBAK       ;YES
3319 017660 004737 020174          JSR     PC,RSTREG
3320 017664 062716 000002          ADD     #2,(SP)      ;NO SCOPE OR LOOP,SO RUN
3321 017670 000002                   RTI                    ;PAST ARGUMENT AND RETURN
3322 017672 004737 020174          TRPBAK: JSR     FC,RSTREG ;RESTORE REGS
3323 017676 017616 000000          MOV     @2(SP),(SP)  ;RETURN ADDRESS TO STACK
3324 017702 000002                   RTI                    ;EXIT, LOOPING

```

3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381

.SBTTL SWITCH REGISTER MONITOR

DESCRIPTION:

SWITCH REGISTER MONITOR. CALLED BY AN ERROR WITH SWR BIT 15 CLEAR, OR BY TYPING CTRL-G ON THE CONSOLE TELETYPE. IF USING HARDWARE SWR, SIMPLY ASKS FOR CTRL-C TO CONTINUE. OTHERWISE, IT PRINTS THE CURRENT CONTENTS OF THE SOFTWARE SWITCH REGISTER, FOLLOWED BY A PROMPT (>). THE NEW SWITCH REGISTER SETTINGS SHOULD THEN BE ENTERED AS AN OCTAL NUMBER, TERMINATED BY CARRIAGE RETURN. TYPING CARRIAGE RETURN ALONE WILL CAUSE THE SWITCH REGISTER CONTENTS TO REMAIN UNCHANGED. IF THE SWITCH REGISTER IS UPDATED TO SELECT A TEST (BIT 6 SET) THE NEW TEST WILL BE ENTERED IMMEDIATELY.

CALLING SEQUENCE:

JSR PC,MONIT

INPUT PARAMETERS:

BY TYPING CNTRL-G DURING THE RUNNING OF THE TESTS.

IMPLICIT INPUT PARAMETERS:

THE SOFTWARE SWITCH REGISTER, IF BEING USED MUST HAVE BEEN SET UP.

OUTPUT PARAMETERS:

IF RUNNING UNDER SOFTWARE SWITCH REGISTER MODE A NEW SETTING OF THE SWR COULD HAVE BEEN SET UP.

IMPLICIT OUTPUT PARAMETERS:

NONE

COMPLETION CODES:

NONE

POSSIBLE ERROR CODES:

NONE

```

3382
3383
3384
3385 017704 010046
3386 017706 023727 016604 177570
3387 017714 001430
3388 017716 004737 021172
3389 017722 015642
3390 017724 013700 016606
3391 017730 004737 021262
3392 017734 012700 000076
3393 017740 004737 021250
3394 017744 004737 020272
3395 017750 005737 016636
3396 017754 001413
3397 017756 010037 016606
3398 017762 032737 000100 016606
3399 017770 001405
3400 017772 000137 001232
3401 017776 004737 020014
3402 020002 000775
3403 020004 004737 021136
3404 020010 012600
3405 020012 000207

:
:
:
MONIT: MOV RO,-(SP) ;SAVE RO
        CMP SWR,#HSWR ;HARDWARE SWR?
        BEQ MONITA ;IF YES, GO TO END
        JSR PC,TYPCT ;SWR=
        SWMSG
        MOV SSWR,RO
        JSR PC,PROCT
        MOV #76,RO
        JSR PC,PCHR ;PRINT '>'
        JSR PC,OCTIN ;GET NEW SETTING
        TST RAND ;ANY INPUT?
        BEQ MONITX
        MOV RO,SSWR ;YES UPDATE SSWR
        BIT #100,SSWR ;TEST SELECTED?
        BEQ MONITX
        JMP START2 ;YES GO DO IT
MONITA: JSR PC,TYPCT ;CTRL-C TO CONTINUE
        BR MONITA
MONITX: JSR PC,CRLF
        MOV (SP)+,RO ;RESTORE RO
        RTS PC
  
```


3407
 3408
 3409
 3410
 3411
 3412
 3413
 3414
 3415
 3416
 3417
 3418
 3419
 3420
 3421
 3422
 3423
 3424
 3425
 3426
 3427
 3428
 3429
 3430
 3431
 3432
 3433
 3434
 3435
 3436
 3437
 3438
 3439
 3440
 3441
 3442
 3443
 3444
 3445
 3446
 3447
 3448
 3449
 3450
 3451
 3452
 3453
 3454
 3455
 3456
 3457
 3458
 3459
 3460
 3461
 3462

.SBTTL WAIT FOR CNTRL-C

DESCRIPTION:

ROUTINE TO WAIT FOR THE USER TO TYPE
 CNTRL-C ON THE CONSOLE. IF CNTRL-O IS HIT MODIFY
 IS CALLED, AND IF CNTRL-G IS HIT MONIT IS CALLED.

IF NONE OF THESE ARE HIT, THE ROUTINE WILL RETURN
 TO THE NEXT LOCATION AFTER THE CALL. IF CNTRL-C WAS
 HIT THE ROUTINE WILL RETURN TO THE NEXT LOCATION+2.

CALLING SEQUENCE:

JSR PC,TYPCTC

INPUT PARAMETERS:

NONE

IMPLICIT INPUT PARAMETERS:

INPUT IS REQUESTED FROM THE CONSOLE

OUTPUT PARAMETERS:

RO IS CORRUPTED

IMPLICIT OUTPUT PARAMETERS:

NONE

COMPLETION CODES:

A RETURN TO THE FIRST OR SECOND LOCATION AFTER
 THE CALL IS PERFORMED.

POSSIBLE ERROR CODES:

NONE

TYPCTC: JSR PC,TYPOUT ;PRINTS TYPE CTRL/C WHEN READY
 REDMES
 QEXIT: JSR PC,READ ;CTRL/C ENTERED
 CMP RO,#3
 BNE QEXIT2 ;NO

020014 004737 021172
 020020 015607
 020022 004737 020444
 020026 020027 000003
 020032 001005

```
3463 020034 004737 021136          JSR    PC,CRLF
3464 020040 062716 000002          ADD    #2,(SP)      ;YES SO JUMP OVER NO FIND RETURN
3465 020044 000207          QEXIT1: RTS        PC      ;GO BACK
3466 020046 020027 000017          QEXIT2: CMP        RO,#17   ;CNTRL-O ?
3467 020052 001003          BNE    QEXIT3      ;NO
3468 020054 004737 021522          JSR    PC,MODIFY   ;YES CALL MODIFY PROGRAM
3469 020060 000755          BR     TYPCTC      ;THEN GO BACK TO START
3470 020062 020027 000007          QEXIT3: CMP        RO,#7    ;CTRL-G?
3471 020066 001366          BNE    QEXIT1
3472 020070 004737 017704          JSR    PC,MONIT    ;GO TO SWR MONITOR
3473 020074 000747          BR     TYPCTC      ;LOOK FOR CTRL-C AGAIN
3474
3475
```

3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532

.SBTTL SAVE REGISTERS

DESCRIPTION:

ROUTINE TO SAVE ALL THE GENERAL PURPOSE
 REGISTERS ON THE STACK, AND LEAVE THE ADDRESS OF THE
 CALLING ROUTINE ON THE STACK. THE ROUTINE WILL RUN AT
 PRIORITY 7 TO AVOID ANY INTERRUPTS

CALLING SEQUENCE:

JSR PC, SAVREG

INPUT PARAMETERS:

NONE

IMPLICIT INPUT PARAMETERS:

NONE

OUTPUT PARAMETERS:

REGISTERS 0 THRU 5 ARE SAVED ON THE STACK
 AND THE RETURN ADDRESS OF THE CALLING ROUTINE IS
 SET AS THE LAST ENTRY ON THE STACK

IMPLICIT OUTPUT PARAMETERS:

NONE

COMPLETION CODES:

NONE

POSSIBLE ERROR CODES:

NONE

SAVREG: PSWREA SA151A
 PSWSET #34C
 MOV (SP)+, SAVPC ;SAVE PC FOR RETURN FROM THIS ROUTINE
 MOV (SP)+, SAVPC1
 MOV R5, -(SP)

020076
020106
020122 012637 016626
020126 012637 016630
020132 010546

```
3533 020134 010446      MOV     R4,-(SP)
3534 020136 010346      MOV     R3,-(SP)
3535 020140 010246      MOV     R2,-(SP)
3536 020142 010146      MOV     R1,-(SP)
3537 020144 010046      MOV     R0,-(SP)
3538 020146 013746 016630      MOV     SAVPC1,-(SP)
3539 020152 013746 016626      MOV     SAVPC,-(SP)      ;PUT PC READY FOR
3540 020156                      PSWSET SAVSTA
3541 020172 000207      RTS     PC                ;RETURN
3542
3543
3544
```

3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601

020174
020204
020220 012637 016626
020224 012637 016630
020230 012600
020232 012601
020234 012602
020236 012603
020240 012604
020242 012605

.....

.SBTTL RESTORE REGISTERS

DESCRIPTION:

RESTORE TO RESTORE THE GENERAL PURPOSE
REGISTERS. THE STACK IS LEFT IN THE SAME STATE AS IT
WAS WHEN SAVREG WAS CALLED.

CALLING SEQUENCE:

JSR PC,RSTREG

INPUT PARAMETERS:

NONE

IMPLICIT INPUT PARAMETERS:

NONE

OUTPUT PARAMETERS:

R0 THRU R5 RESTORED

IMPLICIT OUTPUT PARAMETERS:

NONE

COMPLETION CODES:

NONE

POSSIBLE ERROR CODES:

NONE

RSTREG: PSWREA SAVSTA
PSWSET #340
MOV (SP)+,SAVPC
MOV (SP)+,SAVPC1
MOV (SP)+,R0
MOV (SP)+,R1
MOV (SP)+,R2
MOV (SP)+,R3
MOV (SP)+,R4
MOV (SP)+,R5

3602	020244	013746	016630	MOV	SAVPC1,-(SP)	
3603	020250	013746	016626	MOV	SAVPC,-(SP)	:PUT PC READY FOR
3604	020254			PSWSET	SAVSTA	
3605	020270	000207		RTS	PC	:RETURN

3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662

.SBTTL ENTER AN OCTAL NUMBER

DESCRIPTION:

ROUTINE TO ENTER AN OCTAL NUMBER ON THE CONSOLE.
THE NUMBER ENTERED IS RETURNED IN R0, AND THE VALUE RAND
IS SET TO NON ZERO IF ANY CHARACTERS WERE ENTERED.
TYPING CARRIAGE RETURN, LINE FEED OR ESCAPE WILL
TERMINATE THE LINE BEING ENTERED.
RUBOUT WILL DELETE THE LAST CHARACTER ENTERED,
CNTRL-U WILL RUBOUT THE WHOLE LINE, AND CNTRL-R
WILL TYPE OUT THE CHARACTERS SO FAR ENTERED.
RAND WILL BE SET TO NON ZERO IF ANY CHARACTERS
WERE HIT AND RANDC WILL CONTAIN THE TERMINATING
CHARACTER.

A '!' WILL BE PRINTED IF AN OVERFLOW CONDITION IS
DETECTED AND A '?' IF AN INVALID CHARACTER WAS
ENTERED.

CALLING SEQUENCE:

JSR PC,OCTIN

INPUT PARAMETERS:

NONE

IMPLICIT INPUT PARAMETERS:

NONE

OUTPUT PARAMETERS:

R0 CONTAINS THE NUMBER ENTERED.
RAND=0 IF NO CHARACTERS WERE ENTERED.

IMPLICIT OUTPUT PARAMETERS:

NONE

COMPLETION CODES:

NONE

POSSIBLE ERROR CODES:

NONE

```

3663
3664
3665
3666 020272 004737 020076
3667 020276 005000
3668 020300 005037 016636
3669 020304 012737 016546 016736
3670 020312 012737 000006 016740
3671 020320 012737 000060 016742
3672 020326 012737 000067 016744
3673 020334 005037 016734
3674 020340 004737 020470
3675 020344 013701 016740
3676 020350 001420
3677 020352 012702 016546
3678 020356 112203
3679 020360 162703 000060
3680 020364 000241
3681 020366 006300
3682 020370 103417
3683 020372 006300
3684 020374 103415
3685 020376 006300
3686 020400 050300
3687 020402 005237 016636
3688 020406 005301
3689 020410 001362
3690 020412 010037 016636
3691 020416 004737 020174
3692 020422 013700 016634
3693 020426 000207
3694 020430 012700 000041
3695 020434 004737 021250
3696 020440 000137 020276
3697

```

OCTIN: JSR PC, SAVREG ; SAVE THE REGISTERS
TYPOTB: CLR R0 ; CLEAR R0
CLR RAND ; CLEAR FLAG WORD
MOV #OCTMSG, STRADD ; SET START ADDRESS OF STRING
MOV #6, STRLEN ; AND ITS SIZE
MOV #60, LOWCHR ; MINIMUM CHAR
MOV #67, UPPCHR ; MAXIMUM CHAR
CLR CNVFLG ; DON'T CONVERT TO UPPER CASE
JSR PC, GETSTR ; GET A STRING
MOV STRLEN, R1 ; ZERO LENGTH ?
BEQ TYPOTD ; YES
MOV #OCTMSG, R2 ; NO GET START ADDRESS OF STRING
TYPOTC: MOVB (R2)+, R3 ; GET A CHARACTER
SUB #60, R3 ; PUT IN RANGE
CLC
ASL R0 ; SHIFT OUT RESULT
BCS TYPOTE ; C BIT SET ERROR
ASL R0 ; TIMES 4
BCS TYPOTE ; C BIT ERROR
ASL R0 ; TIMES 8
BIS R3, R0 ; SET NEW BITS IN
INC RAND ; SET GOOD
DEC R1
BNE TYPOTC ; LOOP IF MORE TO COME
TYPOTD: MOV R0, TYPOTA ; SAVE FINAL NUMBER
JSR PC, RSTREG ; RESTORE REGISTERS
MOV TYPOTA, R0 ; GET RESULT
RTS PC
TYPOTE: MOV #41, R0 ; OVERFLOW
JSR PC, PCHR ; PRINT A
JMP TYPOTB ; THEN GO AGAIN

3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745 020444 105737 177560
3746 020450 100375
3747 020452 013700 177562
3748 020456 042700 177600
3749 020462 004737 021250
3750 020466 000207

```

.SBTTL READ A SINGLE CHARACTER

DESCRIPTION:

ROUTINE TO READ A SINGLE CHARACTER
FROM THE CONSOLE. THE CHARACTER IS RETURN IN R0
AND HAS THE 200 BIT STRIPPED OFF.

CALLING SEQUENCE:

JSR PC,READ

INPUT PARAMETERS:

NONE

IMPLICIT INPUT PARAMETERS:

NONE

OUTPUT PARAMETERS:

R0 CONTAINS THE CHARACTER READ IN.

IMPLICIT OUTPUT PARAMETERS:

THE CHARACTER IS ECHOED ON THE TERMINAL

COMPLETION CODES:

NONE

POSSIBLE ERROR CODES:

NONE

READ: TSTB TKS
      BPL READ
      MOV TKB,R0
      BIC #177600,R0
      JSR PC,PCWR
      RTS PC

```

3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807

.SBTTL ENTERING A CHARACTER STRING

DESCRIPTION:

ROUTINE TO ENTER A STRING OF CHARACTERS ON THE CONSOLE TERMINAL. VARIOUS CONTROL CODES ARE USED TO CONTROL HOW THE CHARACTERS ARE INTERPRETED:

ESCAPE, CARRIAGE RETURN AND LINE FEED ARE USED TO TERMINATE THE STRING. RUBOUT WILL DELETE THE LAST CHARACTER ENTERED, CNTRL-U WILL DELETE THE ALL THE CHARACTERS ENTERED, AND CNTRL-R WILL ECHO THOSE CHARACTERS ALREADY ENTERED.

ON ENTRY THE FOLLOWING POINTERS ARE USED:
STRADD TO INDICATE THE START OF THE CHARACTER STRING
STRLN TO INDICATE ITS SIZE
CVNFLG SET TO CONVERT LOWER CASE TO UPPER CASE
UPPCHR TO INDICATE THE HIGHEST CHARACTER CODE
LOWCHR TO INDICATE THE LOWEST CHARACTER CODE

ON EXIT THE LOCATION STRLEN WILL INDICATE THE NUMBER OF CHARACTERS ENTERED AND RANDC WILL CONTAIN THE TERMINATING CHARACTER

IF AN INVALID CHARACTER WAS HIT A '?' IS PRINTED

CALLING SEQUENCE:

JSR PC,GETSTR

INPUT PARAMETERS:

STRADD THE START ADDRESS OF THE STRING
STRLN THE NUMBER OF CHARACTERS TO READ
UPPCHR THE HIGHEST CHARACTER CODE ALLOWED
LOWCHR THE LOWEST CHARACTER CODE ALLOWED
CVNFLG TO INDICATE LOWER TO UPPER CASE CONVERSION

IMPLICIT INPUT PARAMETERS:

NONE

OUTPUT PARAMETERS:

STRLN THE NUMBERS OF CHARACTERS READ
RANDC THE TERMINATING CHARACTER

3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828 020470 004737 020076
3829 020474 005037 016746
3830 020500 013702 016736
3831 020504 005001
3832 020506 105737 177560
3833 020512 100375
3834 020514 013700 177562
3835 020520 042700 177600
3836
3837
3838
3839 020524 020027 000015
3840 020530 001002
3841 020532 000137 021114
3842 020536 020027 000012
3843 020542 001002
3844 020544 000137 021114
3845 020550 020027 000033
3846 020554 001002
3847 020556 000137 021114
3848 020562 020027 000177
3849 020566 001002
3850 020570 000137 020742
3851 020574 020027 000022
3852 020600 001002
3853 020602 000137 021040
3854 020606 020027 000025
3855 020612 001002
3856 020614 000137 021010
3857
3858
3859
3860
3861
3862 020620 020027 000140
3863 020624 002405

```

: IMPLICIT OUTPUT PARAMETERS:
:
:     NONE
:
: COMPELETION CODES:
:
:     NONE
:
: POSSIBLE ERROR CONDITIONS:
:
:     A '?' IS PRINTED IF AN INVALID CHARACTER IS ENTERED
:
:
: GETSTR: JSR     PC, SAVREG      ; SAVE REGISTERS
: GETCH1: CLR     RUBFLG         ; CLEAR RUBOUT FLAG
:         MOV     STRADD, R2      ; GET STARTING POINTER
:         CLR     R1             ; SET INTITAL LENGTH
: GETCH2: TSTB    TKS           ; ANY TTY INPUT
:         BPL     GETCH2         ; NO WAIT FOR IT
:         MOV     TKB, R0        ; GET CHARACTER
:         BIC     #177600, R0    ; CLEAR RUBBISH
:
: CHECK FOR CONTROL CODES
:
:         CMP     RO, #15        ; WAS IT <CR> ?
:         BNE     1$            ; NO
: 1$:      JMP     CHRFIN
:         CMP     RO, #12        ; WAS IT <LF>
:         BNE     3$            ; NO
: 3$:      JMP     CHRFIN
:         CMP     RO, #33       ; WAS IT <ESC>
:         BNE     5$            ; NO
: 5$:      JMP     CHRFIN
:         CMP     RO, #177      ; RUBOUT ?
:         BNE     7$            ; NO
:         JMP     RUBCHR        ; YES
: 7$:      CMP     RO, #22      ; CNTRL-R
:         BNE     9$            ; NO
:         JMP     LINECH       ; YES EXCO LINE
: 9$:      CMP     RO, #25      ; CNTRL-U
:         BNE     GETCH3       ; NO
:         JMP     LINDEL       ; YES DELETE LINE
:
: IT WAS NOT A CONTROL CODE CHECK
: FOR LEGALITY ?
:
: GETCH3: CMP     RO, #140      ; IS IT LOWER CASE
:         BLT     3$            ; NO
    
```

```

3864 020626 005737 016734          TST      CNVFLG      ; YES DO WE CONVERT TO UPPER
3865 020632 001402                   BEQ      3$         ; NO
3866 020634 042700 000040          BIC      #40,RO     ; YES STRIP 40 OFF
3867 020640 020037 016742          3$: CMP      RO,LOWCHR ; IS THE CHAR TOO SMALL
3868 020644 002002                   BGE      5$         ; NO
3869 020646 000137 021074          JMP      ILLCHR     ; YES
3870 020652 020037 016744          5$: CMP      RO,UPPCHR ; IS THE CHAR TOO BIG ?
3871 020656 003402                   BLE      GETCH4     ; NO
3872 020660 000137 021074          JMP      ILLCHR     ; YES ITS ILLEGAL
3873
3874          ; WE HAVE A LEGAL CHARACTER
3875
3876 020664 010003          GETCH4: MOV     RO,R3      ; SAVE CHAR
3877 020666 005737 016746          TST      RUBFLG     ; IS RUBOUT SET
3878 020672 001406                   BEQ      3$         ; NO
3879 020674 012700 000057          MOV      #57,RO     ; YES PRINT A '/'
3880 020700 004737 021250          JSR      PC,PCHR
3881 020704 005037 016746          CLR      RUBFLG     ; CLEAR RUBOUT FLAG
3882 020710 005201          3$: INC      R1       ; UPDATE CHAR COUNT
3883 020712 020137 016740          CMP      R1,STRLEN  ; END OF STRING SEEN
3884 020716 003403                   BLE      5$         ; NO
3885 020720 005000                   CLR      RO         ; YES, FORCE A NULL TERMINATOR
3886 020722 000137 021114          JMP      CHRFIN     ; AND COMPLETE
3887 020726 010300          5$: MOV      R3,RO    ; RESTORE CHAR
3888 020730 004737 021250          JSR      PC,PCHR   ; ECHO IT
3889 020734 110022          MOVB    RO,(R2)+   ; SAVE IT IN BUFFER
3890 020736 000137 020506          JMP      GETCH2    ; NO GET NEXT ONE
3891
3892          ; RUBOUT WAS HIT
3893
3894 020742 005701          RUBCHR: TST     R1      ; ANY CHARACTERS TO RUBOUT
3895 020744 001002                   BNE      3$         ; YES
3896 020746 000137 021030          JMP      LINDL1     ; NO
3897 020752 005737 016746          3$: TST      RUBFLG  ; IS RUBOUT SET ALREADY ?
3898 020756 001006                   BNE      5$         ; YES
3899 020760 012700 000057          MOV      #57,RO     ; NO PRINT A '/'
3900 020764 004737 021250          JSR      PC,PCHR
3901 020770 005237 016746          INC      RUBFLG     ; SET RUBOUT
3902 020774 114200          5$: MOVB    -(R2),RO  ; GET LAST CHAR ENTERED
3903 020776 004737 021250          JSR      PC,PCHR   ; PRINT IT
3904 021002 005301                   DEC      R1         ; REDUCE COUNT
3905 021004 000137 020506          JMP      GETCH2    ; GET ANOTHER CHAR
3906
3907          ; RUBOUT LINE WAS HIT
3908
3909 021010 012700 000136          LINDEL: MOV     #136,RO ; PRINT A ^
3910 021014 004737 021250          JSR      PC,PCHR
3911 021020 012700 000125          MOV      #125,RO   ; THEN U
3912 021024 004737 021250          JSR      PC,PCHR
3913 021030 004737 021136          LINDL1: JSR    PC,CRLF ; START ON A NEWLINE
3914 021034 000137 020474          JMP      GETCH1    ; A GO BACK TO BEGINNING
3915
3916          ; CNTRL-R WAS HIT
3917
3918 021040 112712 000100          LINECH: MOVB   #'@,(R2) ; PUT IN A TERMINATOR
3919 021044 004737 021136          JSR      PC,CRLF   ; START ON A NEW LINE
  
```

```
3920 021050 013737 016736 021066      MOV    STRADD,38      ; SET START ADDRESS
3921 021056 005037 016746              CLR    RUBFLG        ; CLEAR RUBOUTS
3922 021062 004737 021172              JSR    PC,TYPOUT     ; PRINT LINE
3923 021066 000000              38:    0
3924 021070 000137 020506              JMP    GETCH2        ; AND GET ANOTHER CHAR
3925
3926      ; ILLEGAL CHARACTER ENTERED
3927
3928 021074 004737 021250      ILLCHR: JSR    PC,PCHR ; ECHO IT
3929 021100 012700 000077      MOV    #'?',RO      ; PRINT A ?
3930 021104 004737 021250      JSR    PC,PCHR
3931 021110 000137 021040      JMP    LINECH        ; THE ECHO LINE
3932
3933      ; A TERMINATOR WAS FOUND
3934
3935 021114 010037 016750      CHRFIN: MOV    RO,RANDC ; SAVE TERMINATOR
3936 021120 163702 016736      SUB    STRADD,R2    ; CALCULATE BYTE COUNT
3937 021124 010237 016740      MOV    R2,STLEN     ; SET IT FOR RETURN
3938 021130 004737 020174      JSR    PC,RSTREG    ; RESTORE REGISTERS
3939 021134 000207              RTS    PC            ; AND EXIT
```

3941
 3942
 3943
 3944
 3945
 3946
 3947
 3948
 3949
 3950
 3951
 3952
 3953
 3954
 3955
 3956
 3957
 3958
 3959
 3960
 3961
 3962
 3963
 3964
 3965
 3966
 3967
 3968
 3969
 3970
 3971
 3972
 3973
 3974
 3975
 3976
 3977
 3978
 3979
 3980
 3981
 3982
 3983
 3984
 3985
 3986
 3987
 3988
 3989
 3990
 3991
 3992
 3993
 3994
 3995
 3996

.SBTTL CARRIAGE RETURN LINE FEED

DESCRIPTION

ROUTINE TO DO A <CR> <LF> ON THE
 CONSOLE TERMINAL

CALLING SEQUENCE:

JSR PC,CRLF

INPUT PARAMETERS:

NONE

IMPLICIT INPUT PARAMETERS:

NONE

OUTPUT PARAMETERS:

NONE

IMPLICIT OUTPUT PARAMETERS:

THE CARRIAGE WILL BE PLACED ON A NEW LINE

COMPLETION CODES:

NONE

POSSIBLE ERROR CODES:

NONE

```

CRLF:  MOV    R0,-(SP)
      MOV    #15,R0          ;PRINT A RETURN - LINE FEED
      JSR   PC,R0
      CLR   R0              ;DUMMY
      JSR   PC,PCHR
      MOV   #12,R0
      JSR   PC,PCHR
      MOV   (SP)+,R0
      RTS   PC
  
```

VTV VISUAL TESTS MACY11 30A(1052) 06-SEP-79 15:27 PAGE 55-1^{M 8}
CVVTBA.SRC 06-SEP-79 15:23 CARRIAGE RETURN LINE FEED

SEQ 0098

3997

3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054

.SBTTL PRINT AN ASCII MESSAGE

DESCRIPTION:

ROUTINE TO PRINT A STRING OF ASCII
CHARACTERS ON THE CONSOLE TERMINAL. CERTAIN
CHARACTERS WITHIN THE STRING ARE INTERPRETED
AS CONTROL CODES, THESE ARE:

133 ([) WILL GENERATE A <CR>,<LF>
100 (@) WILL SIGNIFY END OF MESSAGE

THE ADDRESS OF THE MESSAGE STRING TO BE PRINTED
WILL BE HELD IN THE LOCATION FOLLOWING THE CALL
TO THE ROUTINE, IE:

JSR PC, TYP0UT
ADDR

CALLING SEQUENCE:

JSR PC, TYP0UT

INPUT PARAMETERS:

THE ADDRESS OF THE MESSAGE STRING FOLLOWS
THE SUBROUTINE CALL.

IMPLICIT INPUT PARAMETERS:

NONE

OUTPUT PARAMETERS:

NONE

IMPLICIT OUTPUT PARAMETERS:

THE SPECIFIED MESSAGE WILL BE PRINTED

COMPLETION CODES:

NONE

POSSIBLE ERROR CODES:

NONE

4055
4056
4057
4058 021172 004737 020076
4059 021176 017601 000000
4060 021202 062716 000002
4061 021206 111100
4062 021210 022700 000100
4063 021214 001412
4064 021216 022700 000133
4065 021222 001003
4066 021224 004737 021136
4067 021230 000402
4068 021232 004737 021250
4069 021236 005201
4070 021240 000762
4071 021242 004737 020174
4072 021246 000207

```

:
:
:
TYPOUT: JSR    PC, SAVREG      ;SAVE REGS
        MOV    @ (SP), R1      ;R1 POINTS AT STRING
        ADD    #2, (SP)        ;JUMPS OVER ARGUMENT
PMSG1:  MOVB   @R1, R0         ;PRINT THE MESSAGE POINTED
        CMP    #10, R0         ;TO BY R1 UNTIL @, WHICH IS END.
        BEQ    PMSG4          ;[ MEANS CR-LF
        CMP    #133, R0
        BNE   PMSG2
        JSR    PC, CRLF
        BR    PMSG3
PMSG2:  JSR    PC, PCHR
PMSG3:  INC    R1
        BR    PMSG1
PMSG4:  JSR    PC, RSTREG      ;RESTORE REGS
        RTS   PC
```

4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121

.SBTTL PRINT A CHARACTER

DESCRIPTION:

ROUTINE TO PRINT A CHARACTER ON THE
CONSOLE. R0 CONTAINS THE CHARACTER TO BE PRINTED

CALLING SEQUENCE:

JSR PC,PCHR

INPUT PARAMETERS:

R0 CONTAINS THE CHARACTER TO BE PRINTED

IMPLICIT INPUT PARAMETERS:

NONE

OUTPUT PARAMETERS:

NONE

IMPLICIT OUTPUT PARAMETERS:

THE CHARACTER SELECTED WILL BE PRINTED

COMPLETION CODES:

NONE

POSSIBLE ERROR CODES:

NONE

PCHR: JSR PC,TPREDY ;PRINTER READY
MOV R0,IPB
RTS PC

021250 004737 017434
021254 010037 177566
021260 000207

4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178

021262 004737 020076
021266 012701 016520
021272 112721 000040
021276 020127 016526
021302 001373
021304 000241
021306 010002
021310 042702 177770
021314 062702 000060

.SBTTL PRINT AN OCTAL NUMBER

DESCRIPTION:

ROUTINE TO PRINT AN OCTAL NUMBER ON THE CONSOLE TERMINAL. R0 CONTAINS THE BINARY REPRESENTATION ON THE NUMBER THAT IS TO BE PRINTED.

CALLING SEQUENCE:

JSR PC,PROCT

INPUT PARAMETERS:

R0 CONTAINS THE NUMBER THAT IS TO BE PRINTED

IMPLICIT INPUT PARAMETERS:

NONE

OUTPUT PARAMETERS:

NONE

IMPLICIT OUTPUT PARAMETERS:

THE NUMBER SPECIFIED WILL BE PRINTED

COMPLETION CODES:

NONE

POSSIBLE ERROR CODES:

NONE

PROCT: JSR PC,SAVREG ;SAVE REGS
MOV #OCTNUM,R1 ;POINTER TO MESSAGE
PROCT1: MOVB #40,(R1)+ ;FILL WITH SPACES
CMP R1,#OCTNUM+6 ;ALL DONE
BNE PROCT1 ;NO
PRCT1A: CLC ;CLEAR AT START
PROCT2: MOV R0,R2 ;SAVE CHARS
BIC #177770,R2 ;CLEAR ALL BUT BOTTOM 3 BITS
ADD #60,R2 ;NOW ASCII

```

4179 021320 110241          MOVB   R2,-(R1)          ;STORE IT IN MESSAGE
4180 021322 042700 000007   BIC    #7,R0           ;NOW CLEAR BOTTOM 3 BITS
4181 021326 001404          BEQ    PROCT3          ;ALL DONE
4182 021330 006000          ROR    R0              ;ROTATE NEXT 3 BITS
4183 021332 006000          ROR    R0
4184 021334 006000          ROR    R0
4185 021336 000763          BR     PROCT2          ;GO DO NEXT
4186 021340 004737 021172   PROCT3: JSR   PC,TYPEOUT ;TYPE MESSAGE
4187 021344 016520          OCTNUM                ;MESSAGE ADD.
4188 021346 004737 020174   JSR   PC,RSTREG       ;RESTORE REGS
4189 021352 000207          RTS    PC
4190                          :
4191                          :
4192                          :ROUTINE TO PRINT LEAST SIG.9 BITS OF R0
4193                          :
4194                          :
4195 021354 004737 020076   PRNT3: JSR   PC,SAVREG  ;CLEAR UNWANTER
4196 021360 042700 177000   BIC    #177000,R0     ;SET UP MESSAGE
4197 021364 012737 002040 016520  MOV    #2040,OCTNUM   ;WITH END CODE
4198 021372 012737 040040 016522  MOV    #40040,OCTNUM+2 ;& POINTER
4199 021400 012701 016523   MOV    #OCTNUM+3,R1   ;JUMP INTO MAIN ROUTINE
4200 021404 000737          BR     PRCT1A         ;WITH POINTERS SET UP
4201

```

4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258

021406 005700
021410 100005
021412 005400
021414 112737 000055 016530
021422 000403

.....

.SBTTL PRINT A DECIMAL NUMBER

DESCRIPTION:

ROUTINE TO PRINT A SIGNED, OR UNSIGNED
DECIMAL NUMBER ON THE CONSOLE. RO CONTAINS THE
BINARY REPRESENTATION ON THE NUMBER TO BE PRINTED

CALLING SEQUENCE:

JSR PC,BASE10 FOR UNSIGNED
JSR PC,BASM10 FOR SIGNED

INPUT PARAMETERS:

RO CONTAINS THE NUMBER TO BE PRINTED

IMPLICIT INPUT PARAMETERS:

NONE

OUTPUT PARAMETERS:

NONE

IMPLICIT OUTPUT PARAMETERS:

THE NUMBER SPECIFIED WILL BE PRINTED

COMPLETION CODES:

NONE

POSSIBLE ERROR CODES:

NONE

BASM10: TST RO ;IS IT -VE
BPL BASE10 ;NO
NEG RO ;YES SO MAKE +VE
MCLVB #55,BASE11 ;PUT '-' IN MESSAGE
BR BAS10A ;GO DO IT

ROUTINE TO PRINT UNSIGNED DECIMAL NO.

```

4259
4260 021424 112737 000040 016530 BASE10: MOVB #40,BASE11 ;PUT SPACE AS 'S' CHARA
4261 021432 004737 020076 BASE10A: JSR PC,SAVREG ;SAVE REGS
4262 021436 012703 016531 MOV #BASE11+1,R3 ;REST OF MESSAGE WITH SPACES
4263 021442 112723 000040 BASE1A: MOVB #40,(R3)+
4264 021446 022703 016536 CMP #BASE11+6,R3 ;ALL DONE
4265 021452 001373 BNE BASE1A ;NO
4266 021454 005001 BASE1D: CLR R1 ;R1 IS RECEIVER
4267 021456 020027 000012 BASE1B: CMP R0,#12 ;MORE THAN 10
4268 021462 103404 BLO BASE1C ;YES SO DONE THIS TIME
4269 021464 162730 000012 SUB #12,R0 ;NO SO SUB 10 & ADD 1 TO R1
4270 021470 005201 INC R1
4271 021472 000771 BR BASE1B ;GO DO AGAIN
4272 021474 062700 000060 BASE1C: ADD #60,R0 ;MAKE ASCII & STORE
4273 021500 110043 MOVB R0,-(R3)
4274 021502 010100 MOV R1,R0
4275 021504 001363 BNE BASE1D
4276 021506 004737 020174 JSR PC,RSTREG ;RESTORE REGS
4277 021512 004737 021172 JSR PC,TYPOUT ;TYPE MESSAGE
4278 021516 016530 BASE11 ;ADDRESS OF MESSAGE
4279 021520 000207 RTS PC
  
```

4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336

.SBTTL MODIFY THE PROGRAM

DESCRIPTION:

ROUTINE TO MODIFY A LOCATION IN MEMORY
ENTERED BY TYPING CNTRL-O ON THE CONSOLE
TERMINAL.

PROMPTS (\$) FOR AN ADDRESS TO EXAMINE
AND PRINTS IT IN THE FORM

ADDR CONTENTS /

THEN A NEW VALUE CAN BE ENTERED

VIS:

ADDR CONTENTS / NEW VALUE

THE NEW VALUE CAN BE TERMINATED USING
<CR>, <LF>, OR <ESC>

<LF> TO EXAMINE THE NEXT LOC
<CR> TO SELECT ANOTHER ADDRESS
<ESC> TO EXIT

CALLING SEQUENCE:

JSR PC,MODIFY

INPUT PARAMETERS:

NONE

IMPLICIT INPUT PARAMETERS:

ENTERED BY TYPING CNTRL-O DURING
THE RUNNING OF THE TESTS

OUTPUT PARAMETERS:

NONE

4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354 021522 004737 020076
4355 021526
4356 021536
4357 021552 004737 021172
4358 021556 015654
4359 021560 004737 020272
4360 021564 005737 016636
4361 021570 001506
4362 021572 010037 016642
4363 021576 032737 000001 016642
4364 021604 001404
4365 021606 004737 021172
4366 021612 015765
4367 021614 000756
4368 021616 012737 022572 000004
4369 021624 012737 000340 000006
4370 021632 005037 016652
4371 021636 005777 175000
4372 021642 005737 016652
4373 021646 001341
4374 021650 004737 021136
4375 021654 013700 016642
4376 021660 004737 021262
4377 021664 012700 000040
4378 021670 004737 021250
4379 021674 017700 174742
4380 021700 004737 021262
4381 021704 004737 021172
4382 021710 015661
4383 021712 004737 020272
4384 021716 005737 016636
4385 021722 001402
4386 021724 010077 174712
4387 021730 013700 016750
4388 021734 001706
4389 021736 020027 000015
4390 021742 001703
4391 021744 020027 000012
4392 021750 001006

```

:
:      IMPLICIT OUTPUT PARAMETERS:
:
:      THE LOCATIONS SPECIFIED WILL HAVE BEEN
:      MODIFIED
:
:      COMPLETION CODES:
:
:      NONE
:
:      POSSIBLE ERROR CODES:
:
:      NONE
:
:
:      MODIFY: JSR      PC,SAVREG      ;SAVE REGISTERS
:              PSWREA MODSAV
:              PSWSET #340
:
:      MOD11: JSR      PC,TYPEOUT     ;PROMPT $ FOR AN ADDRESS
:              MODADM                ;TO HAVE A LOOK AT
:              JSR      PC,OCTIN      ;READ REPLY
:              TST      RAND          ;ANYTHING READ ?
:              BEQ      MODXIT        ;NO, SO EXIT
:              MOV      RO,MODADR     ;ELSE SAVE OUR ADDRESS
:              BIT      #1,MODADR     ;IS IT EVEN ?
:              BEQ      MOD13        ;YES WE CAN USE IT
:              JSR      PC,TYPEOUT     ;ELSE SAY IT IS AN ODD
:              ODAMSG                ;ADDRESS, AND REPROMPT
:
:              BR      MOD11
:
:      MOD13: MOV      #NXMTRP,4      ;PLUG TRAP THRU 4
:              MOV      #340,6       ;FOR NXM TESTS
:              CLR      TRPERR       ;CLEAR NXM FLAG
:              TST      @MODADR       ;TEST OUR ADDRESS
:              TST      TRPERR       ;DOES IT EXIST
:              BNE      MOD11        ;NO TRY AGAIN
:              JSR      PC,CRLF       ;START ON A NEW LINE
:              MOV      MODADR,RO    ;PRINT OUR ADDRESS
:              JSR      PC,PROCT
:              MOV      #40,RO       ;THEN A SPACE
:              JSR      PC,PCHR       ;AS A SEPARATOR
:              MOV      @MODADR,RO   ;THEN PRINT THE CONTENTS
:              JSR      PC,PROCT
:              JSR      PC,TYPEOUT     ;PROMPT FOR THE NEW
:              MODSPA                ;CONTENTS '/'
:              JSR      PC,OCTIN      ;READ REPLY
:              TST      RAND          ;ANYTHING GIVEN
:              BEQ      MOD14        ;NO, DON'T UPDATE
:              MOV      RO,@MODADR    ;ELSE SET NEW VALUE
:              MOD14: MOV      RANDC,RO ; GET TERMINATOR
:              BFG      MOD11        ; NULL MEANS <CR>
:              CMP      RO,#15       ;WAS IT <CR> ?
:              BEQ      MOD11        ;YES GET NEXT ADDRESS
:              CMP      RO,#12       ;WAS IT <LF> ?
:              BNE      MOD15        ;NO

```


4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464

.SBTTL SET BUS AND VECTOR ADDRESSES

DESCRIPTION:

ROUTINE TO SET UP THE BUS AND VECTOR ADDRESSES AND PRIORITY LEVELS OF THE DEVICE UNDER TEST. THE VALIDITY OF THE VARIOUS ADDRESSES ARE CHECKED. THE ARGUMENTS AFTER THE CALL MUST BE SET TO INDICATE THE NUMBER OF BUS AND VECTOR ADDRESSES AND THE NUMBER OF PRIORITIES REQUIRED, AND THE LOCATIONS WHERE THE VARIOUS PARAMETERS ARE TO BE STORED

IT IS CALLED THUS:

```
JSR    R5,BUSSET  
.BYTE  A,B  
.WORD  C  
.BYTE  D,E  
.WORD  F,G  
.BYTE  H,J  
.WORD  K,L
```

WHERE A IS EITHER 7 IF THERE ARE 4 OR LESS BUS ADDRESSES OR 17 IF THERE ARE 5-8 ADDRESSES, OR 37 IF THERE 9-16, AND SO ON. B IS THE NUMBER OF BUS ADDRESSES. C IS THE ADDRESS INTO WHICH THE FIRST BUS ADDRESS WILL BE LOADED. SUBSEQUENT BUS ADDRESSES WILL BE LOADED INTO THE LOCATION FOLLOWING THE ADDRESS C. HENCE IF THERE A 4 BUS ADDRESSES THE C MUST POINT TO A FOUR WORD BUFFER THAT WILL CONTAIN THE 4 BUS ADDRESSES. D IS THE NUMBER OF VECTOR ADDRESSES E IS THE NUMBER OF PRIORITY LEVELS, AND IF THE VECTOR IS TO BE ON A 10 BYTE BOUNDARY, BIT 7 OF E MUST BE SET AS A FLAG. F IS THE ADDRESS INTO WHICH THE FIRST VECTOR ADDRESS WILL BE LOADED. G IS THE ADDRESS INTO WHICH THE FIRST PRIORITY WILL BE LOADED. H AND J ARE EQUIVALENT TO D AND E IN CASE MORE THAN ONE VECTOR GROUP ARE PRESENT. K AND L ARE THE ADDRESSES EQUIVALENT TO F AND G. THE SEQUENCE H,J,K,L CAN BE REPEATED FOR AS MANY VECTOR PAIRS AS REQUIRED. THE SEQUENCE IS TERMINATED BY SETTING H AND J TO ZERO

CALLING SEQUENCE:

```
JSR    R5,BUSSET
```

INPUT PARAMETERS:

THE ARGUMENTS TO SET UP THE ADDRESSES, VECTORS AND PRIORITIES MUST BE SET UP, AS DESCRIBED ABOVE

IMPLICIT INPUT PARAMETERS:

```

4465 : NONE
4466 :
4467 :
4468 : OUTPUT PARAMETERS:
4469 :
4470 : NONE
4471 :
4472 :
4473 : IMPLICIT OUTPUT PARAMETERS:
4474 :
4475 : THE BUS AND VECTOR ADDRESSES WILL
4476 : BE SET UP AS SPECIFIED BY THE ARGUMENTS
4477 :
4478 :
4479 : COMPLETION CODES:
4480 :
4481 : NONE
4482 :
4483 :
4484 : POSSIBLE ERROR CODES:
4485 :
4486 : NONE
4487 :
4488 :
4489 :
4490 022050 004737 020076 BUSSET: JSR PC, SAVREG ;SAVE REGISTER CONTENTS
4491 022054 012737 022572 000004 MOV #NXMTRP,4 ;SET UP MEMORY ERROR TRAP
4492 022062 012737 000340 000006 MOV #340,6
4493 :
4494 022070 012537 016646 MOV (R5)+, MASK ;GET MASK AND NO OF ADDRESSES
4495 022074 012537 016650 MOV (R5)+, BASADD ;GET BASE BUS ADDRESS
4496 :
4497 022100 004737 021172 BUSSE1: JSR PC, TYP0UT
4498 022104 015672 BMSG ;FIRST BUS ADDRESS IS....
4499 022106 004737 020272 1$: JSR PC, OCTIN ;INPUT OCTAL ADDRESS
4500 022112 005737 016636 TST RAND
4501 022116 001004 BNE 2$
4502 022120 004737 021172 JSR PC, TYP0UT
4503 022124 016252 NODFEM
4504 022126 000767 BR 1$
4505 022130 133700 016646 2$: BITB MASK, R0 ;CHECK INPUT
4506 022134 001406 BEQ BUSSE2
4507 :
4508 022136 004737 021172 JSR PC, TYP0UT
4509 022142 015765 ODAMSG ;INVALID ADDRESS
4510 022144 004737 021262 BUSS1A: JSR PC, PROCT
4511 022150 000753 BR BUSSE1
4512 :
4513 022152 005037 016652 BUSSE2: CLR TRPERR
4514 022156 113701 016647 MOVB MASK+1, R1 ;SET UP COUNT
4515 022162 013702 016650 MOV BASADD, R2 ;SET UP ADDRESS BASE
4516 022166 010022 BUSSE3: MOV R0, (R2)+ ;SET UP ADDRESS
4517 022170 005710 TST (R0) ;CHECK ADDRESS EXISTS
4518 022172 005737 016652 TST TRPERR ;CHECK NON-EXISTANT MEMORY FLAG
4519 022176 001362 BNE BUSS1A
4520 022200 062700 000002 ADD #2, R0 ;UPDATE TO NEXT ADDRESS

```

```

4521
4522 022204 00530i      DEC      R1
4523 022206 001367      BNE     BUSSE3
4524 022210 004737 021172 JSR     PC,TYPOUT      ; PROMPT FOR VECTOR GROUP
4525 022214 015450      FIRVMS
4526
4527 022216 012537 016646 MOV     (R5)+,MASK     ;GET NO OF VECTORS
4528 022222 012537 016650 BUSSE3A: MOV    (R5)+,BASADD ;GET BASE VECTOR ADDRESS
4529
4530 022226 004737 021172 BUSSE4: JSR     PC,TYPOUT
4531 022232 015726      VAMSG
4532 022234 004737 020272 1$:     JSR     PC,OCTIN  ;FIRST VECTOR ADDRESS IS....
4533 022240 005737 016636      TST     RAND          ;INPUT OCTAL ADDRESS
4534 022244 001004      BNE     2$
4535 022246 004737 021172 JSR     PC,TYPOUT
4536 022252 016252      NODEFM
4537 022254 000767      BR      1$
4538 022256 005737 016646 2$:     TST     MASK
4539 022262 1000J7      BPL     BUSSE6        ;CHECK MASK TYPE
4540
4541 022264 032700 000007 BIT     #7,R0          ;CHECK ADDRESS
4542 022270 001404      BEQ     BUSSE6
4543
4544 022272 004737 021172 BUSSE5: JSR     PC,TYPOUT
4545 022276 015765      ODAMSG
4546 022300 000752      BR      BUSSE4        ;INVALID ADDRESS
4547
4548 022302 032700 000003 BUSSE6: BIT     #3,R0
4549 022306 001371      BNE     BUSSE5        ;CHECK ADDRESS
4550
4551 022310 022700 000774 CMP     #774,R0      ;CHECK ADDRESS LESS THAN 772
4552 022314 002004      BGE     BUSSE7
4553
4554 022316 004737 021172 JSR     PC,TYPOUT
4555 022322 016010      OVAMSG
4556 022324 000740      BR      BUSSE4        ;ADDRESS EXCEEDS 772
4557
4558 022326 013701 016646 BUSSE7: MOV     MASK,R1 ;SET UP COUNT
4559 022332 013702 016650 MOV     BASADD,R2    ;SET UP ADDRESS BASE
4560 022336 010022 BUSSE10: MOV    RO,(R2)+ ;SET UP ADDRESS
4561 022340 022020      CMP     (R0)+,(R0)+
4562 022342 005301      DEC     R1
4563 022344 001374      BNE     BUSS10
4564 022346 042737 100000 016640 BIC     #100000,MASK
4565
4566 022354 113701 016647 MOV     MASK+1,R1
4567 022360 012537 016650 MOV     (R5)+,BASADD ;GET BASE PRIORITY ADDRESS
4568
4569 022364 005737 017044 TST     LSIFLG      ;SINGLE INT.LEVEL PROCESSOR ?
4570 022370 001403      BEQ     3$          ;IF NOT THEN BRANCH AWAY.
4571 022372 012700 000004 MOV     #4,R0        ;ELSE SET PRIORITY LEVEL AT 4
4572 022376 000416      BR      BUS11B      ;THEN GO INSTALL IT.
4573
4574 022400 004737 021172 3$:     JSR     PC,TYPOUT
4575 022404 016037      PRMSG
4576 022406 013702 016650 MOV     BASADD,R2    ;FIRST PRIORITY LEVEL IS...

```

```

4577 022412 004737 020272          BUSS11: JSR      PC,OC11M      ;INPUT OCTAL PRIORITY
4578 022416 005737 016636          TST      RAND
4579 022422 001004          BNE      BUS11B
4580 022424 004737 021172          JSR      PC,TYPOUT
4581 022430 016252          NODEFM
4582 022432 000767          BR       BUSS11
4583 022434 020027 000007          BUS11B: CMP      R0,#7          ;IS THE PRIORITY LEGAL ?
4584 022440 101404          BLOS    2$          ;YES
4585 022442 004737 021172          JSR      PC,TYPOUT
4586 022446 016165          BADPRI          ;NO, RE-ENTER VALUE
4587 022450 000760          BR       BUSS11
4588 022452 042700 177770          2$:      BIC      #-10,R0      ;CLEAR UNWANTED BITS
4589 022456 000300          SWAB     R0
4590 022460 006200          ASR     R0
4591 022462 006200          ASR     R0
4592 022464 006200          ASR     R0
4593 022466 010022          MOV     R0,(R2)+    ;SET UP PRIORITY
4594
4595 022470 005301          DEC     R1
4596 022472 001412          BEQ     BUSS12
4597
4598 022474 005737 017044          ;      TST      LSIFLG      ;SINGLE INT.LEVEL PROCESSOR ?
4599 022500 001403          BEQ     4$          ;IF NOT THEN BRANCH AWAY
4600 022502 012700 000004          MOV     #4,R0      ;ELSE SET UP PRIORITY AS 4
4601 022506 000752          BR      BUS11B     ;AND GO INSTALL IT.
4602
4603 022510 004737 021172          4$:      JSR      PC,TYPOUT
4604 022514 016076          NPRMSG
4605 022516 000735          BR      BUSS11     ;NEXT PRIORITY LEVEL IS....
4606
4607 022520 012537 016646          BUSS12: MOV     (R5)+,MASK    ; GET NEXT VECTOR PAIR
4608 022524 001405          BEQ     BUSS13     ; ZERO MEANS END
4609 022526 004737 021172          JSR      PC,TYPOUT
4610 022532 015504          NXTVMS          ; ELSE PROMPT FOR NEXT GROUP
4611 022534 000137 022222          JMP     BUS 3A
4612 022540 010537 016646          BUSS13: MOV     R5,MASK
4613 022544 012737 000006 000004          MOV     #6,1
4614 022552 012737 000004 000006          MOV     #4,5
4615
4616 022560 004737 020174          JSR      PC,RSTREG  ;RESTORE REGISTER
4617 022564 013705 016646          MOV     MASK,R5
4618 022570 000205          RTS     R5
4619
4620
4621
4622
4623 022572 004737 021172          NXMTRP: JSR      PC,TYPOUT
4624 022576 016135          NXMSG
4625 022600 005237 016652          INC     TRPERR
4626 022604 000002          RTI
4627
4628
4629
4630
4631

```

4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688

.SBTTL NUMBER GENERATOR

DESCRIPTION:

ROUTINE TO GENERATE DATA PATTERNS,
THE TYPE OF PATTERN IS SELECTED BY R3, AND THE
PATTERN GENERATED IS RETURNED IN R0 AND LOCATION
GOOD.

CALLING SEQUENCE:

JSR PC,GENER

INPUT PARAMETERS:

R3 CONTAINS THE PATTERN NUMBER

R3=0 ~ ALL ZEROES
1 ALL ONES
2 010101 ETC BIT PATTERN
3 101010 ETC BIT PATTERN
4 ROTATING 1 IN A ZERO WORD
5 ROTATING 0 IN AN ALL ONE WORD
6 PSEUDO RANDOM NUMBER
7 INCREMENTING DATA PATTERN, GOOD
CONTAINS THE VALUE TO BE UPDATED

IMPLICIT INPUT PARAMETERS:

NONE

OUTPUT PARAMETERS:

THE NUMBER GENERATED IS HELD IN
R0 AND GOOD.

IMPLICIT OUTPUT PARAMETERS:

NONE

COMPLETION CODES:

NONE

POSSIBLE ERROR CODES:

NONE

```

4689
4690
4691
4692 022606 042703 177770 GENER: BIC #177770,R3
4693 022612 004737 020076 JSR PC,SAVREG
4694 022616 006303 ASL R3
4695 022620 000173 022624 JMP @GENSEL(R3)
4696 022624 022644 GENSEL: GEN0 :ALL ZERO WORD
4697 022626 022650 GEN1 :ALL ONE WORD
4698 022630 022656 GEN52 :52 PATTERN
4699 022632 022664 GEN25 :25 PATTERN
4700 022634 022672 GENR1 :ROTATE '1' EACH CALL
4701 022636 022702 GENRO :ROTATE '0' EACH CALL
4702 022640 022740 GENRAN :RANDOM NUMBER
4703 022642 023060 GENINC :INCREMENTING COUNT
4704 022644 005000 GENO: CLR RO :0>RO
4705 022646 000507 BR GENEX
4706 022650 005000 GEN1: CLR RO :NOTC>RO
4707 022652 005100 COM RO
4708 022654 000504 BR GENEX
4709 022656 012700 052525 GEN52: MOV #52525,R0 :5252>RO
4710 022662 000501 BR GENEX
4711 022664 012700 125252 GEN25: MOV #125252,R0 :125252>RO
4712 022670 000476 BR GENEX
4713 022672 000241 GENR1: CLC
4714 022674 004737 022714 JSR PC,GENROT :SHIFT 1 > RO
4715 022700 000472 BR GENEX
4716 022702 000241 GENRO: CLC
4717 022704 004737 022714 JSR PC,GENROT
4718 022710 005100 COM RO :SHIFT 0 > RO
4719 022712 000465 BR GENEX
4720 022714 006037 022736 GENROT: ROR GENISH :ROTATE 1 PATTERN
4721 022720 001003 BNE GENERI : = 0?
4722 022722 012737 100000 022736 MOV #100000,GENISH :YES, SET MSB
4723 022730 013700 022736: GENR1: MOV GENISH,R0 :PUT 1 IN RO
4724 022734 000207 RTS PC :AND EXIT
4725 022736 000001 GENISH: 1
4726 022740 012737 000005 016666 GENRAN: MOV #5,RANSEL :SET SELECT VALUE TO 5
4727 022746 004737 022760 JSR PC,RANGEN :GENERATE RANDOM NUMBER IN RO
4728 022752 013700 016664 MOV RANDN,R0
4729 022756 000443 BR GENEX
4730 022760 013702 016664 RANGEN: MOV RANDN,R2
4731 022764 001002 BNE RAN1 :IS RANDOM = 0
4732 022766 013702 016672 MOV RANST,R2 :YES, PUT RANDOM START VALUE IN
4733 022772 032737 000777 016666 RAN1: BIT #777,RANSEL :NO; IS RANSEL SELECT VALUE - 0
4734 023000 001003 BNE RAN2 :NO
4735 023002 012737 000001 016666 MOV #1,RANSEL :YES: SET RANSEL = 1
4736 023010 013703 016666 RAN2: MOV RANSEL,R3
4737 023014 013702 016664 MOV RANDN,R2
4738 023020 033702 016670 BIT RANMTA,R2 :GET R2 <0 AND 1>
4739 023024 001405 BEQ RANCLC
4740 023026 005102 COM R2
4741 023030 033702 016670 BIT RANMTA,R2
4742 023034 001401 BEQ RANCLC
4743 023036 000402 BR RANSEC
4744 023040 000241 RANCLC: CLC

```

4745 023042 000401
4746 023044 000261
4747 023046 006037 016664
4748 023052 005303
4749 023054 001357
4750 023056 000207
4751 023060 013700 016674
4752 023064 005200
4753 023066 010037 016674
4754 023072 010066 000002
4755 023076 004737 020174
4756 023102 000207

BR RAN4
RANSEC: SEC
RAN4: ROR RANDN
DEC R3
BNE RAN2+4
RANEX: RTS PC
GENINC: MOV GOOD,RO
INC RO
GENEX: MOV RO,GOOD
MOV RO,2(SP)
JSR PC,RSTREG
RTS PC

:ROTATE C TO B15
:IS THIS NUMBER REQUIRED?
:NO, GET ANOTHER
:YES, EXIT
:INCREMENTS LOC. 'GOOD'

:LOAD RO MAINROUTINE

4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813

.SBTTL PRINT ERROR MESSAGES

DESCRIPTION:

ROUTINE TO PRINT ERROR MESSAGES
IF BIT 14 IN THE SWR IS SET NO MESSAGES WILL BE
PRINTED, IF BIT 15 IS SET THE PROGRAM WILL NO WAIT
AFTER AN ERROR HAS BEEN PRINTED. IT IS CALLED THUS:

JSR PC,ERROR
ARG

WHERE ARG CONTAINS THE ERROR CODE AND IS OF THE FORM
X*N, WHERE N IS THE ERROR NUMBER IN THE RANGE 0-177
AND X IS A COMBINATION OF FLAGS THAT INDICATE WHAT
VALUES ARE TO BE PRINTED. THESE VALUES SHOULD BE LOADED
BEFORE THE ERROR ROUTINE IS CALLED AND ARE DEFINED
AS FOLLOWS:

FLAG SETTING	NUMBER	LOCATION	MESSAGE
-----	-----	-----	-----
C	4000	CALLPC	CALLED FROM
S	10000	STATUS	STATUS
A	20000	ADDRESS	ADDRESS
D	40000	DATA	DATA
G	100000	GOOD, BAD	GOOD= BAD =

IN ADDITION THE ERROR NUMBER WILL BE COMBINED WITH
THE TEST NUMBER TO INDICATE IN WHICH TEST THE ERROR
OCCURRED.

AN ERROR COUNT IS MAINTAINED AN ON EACH ERROR THE
COUNT IS UPDATED. IF RUNNING UNDER A SOFTWARE SWITCH
REGISTER, IT IS POSSIBLE TO SELECT NEW OPTIONS

CALLING SEQUENCE:

JSR PC,ERROR

INPUT PARAMETERS:

THE LOACTION FOLLOWING THE CALL
CONTAINS THE ERROR CODE AND FLAG SETTINGS

IMPLICIT INPUT PARAMETERS:

NONE

OUTPUT PARAMETERS:

4814
 4815
 4816
 4817
 4818
 4819
 4820
 4821
 4822
 4823
 4824
 4825
 4826
 4827
 4828
 4829
 4830
 4831
 4832
 4833
 4834
 4835 023104 004737 020076
 4836 023110 005237 016706
 4837 023114 032777 040000 173462
 4838 023122 001133
 4839 023124 017637 000000 016710
 4840 023132 004737 021172
 4841 023136 016354
 4842 023140 013700 016602
 4843 023144 000300
 4844 023146 006300
 4845 023150 153700 016710
 4846 023154 004737 021262
 4847 023160 004737 021172
 4848 023164 016360
 4849 023166 011600
 4850 023170 004737 021262
 4851 023174 132737 000200 016711
 4852 023202 001416
 4853 023204 004737 021172
 4854 023210 016372
 4855 023212 013700 016674
 4856 023216 004737 021262
 4857 023222 004737 021172
 4858 023226 016403
 4859 023230 013700 016676
 4860 023234 004737 021262
 4861 023240 132737 000100 016711
 4862 023246 001407
 4863 023250 004737 021172
 4864 023254 016414
 4865 023256 013700 016700
 4866 023262 004737 021262
 4867 023266 132737 000040 016711
 4868 023274 001407
 4869 023276 004737 021172

NONE

IMPLICIT OUTPUT PARAMETERS:

THE APPROPRAITE ERROR MESSAGE WILL BE PRINTED
 IF PERMITTED.

COMPLETION CODES:

NONE

POSSIBLE ERROR CODES:

NONE

ERROR: 15/11/78
 JSR PC,SAVREG
 INC ERDIS ;INC COUNT
 BIT #40000,@SWR ;SUPPRESS PRINTOUT
 BNF ERHALT ;YES
 MOV @(SP),ERPARG ;NO
 JSR PC,TYPOUT ;TYPE MESSAGE
 MSG1
 MOV TESTNO,R0
 SWAB R0
 ASL R0
 BISB ERRARG,R0
 JSR PC,PROCT
 JSR PC,TYPOUT
 MSG2
 MOV (SP),R0
 JSR PC,PROCT ;PRINT 6 DIGITS
 BITB #200,ERRARG+1
 BEQ ERRSV1
 JSR PC,TYPOUT
 MSG3
 MOV GOOD,R0
 JSR PC,PROCT
 JSR PC,TYPOUT
 MSG4
 MOV BAD,R0
 JSR PC,PROCT
 BITB #100,ERRARG+1 ;D SET ?
 BEQ ERRSV2 ;NO
 JSR PC,TYPOUT
 MSG5
 MOV DATA,R0
 JSR PC,PROCT
 BITB #40,ERRARG+1 ;A SET ?
 BEQ ERRSV3 ;NO
 JSR PC,TYPOUT

4870	023302	016425				EMSG6			
4871	023304	013700	016704			MOV	ADDRES,RO		
4872	023310	004737	021262			JSR	PC,PROCT		
4873	023314	132737	000020	016711	ERRSV3:	BITB	#20,ERRARG+1	:S SET ?	
4874	023322	001407				BEQ	ERRSV4	:NO	
4875	023324	004737	021172			JSR	PC,TYPOUT		
4876	023330	016443				EMSG7			
4877	023332	013700	016702			MOV	STATUS,RO		
4878	023336	004737	021262			JSR	PC,PROCT		
4879	023342	132737	000010	016711	ERRSV4:	BITB #10,	ERRARG+1	:C SET	
4880	023350	001407				BEQ	ERRSV5	:NO	
4881	023352	004737	021172			JSR	PC,TYPOUT		
4882	023356	016456				EMSG8			
4883	023360	013700	016712			MOV	CALLPC,RO		
4884	023364	004737	021262			JSR	PC,PROCT		
4885	023370	004737	021136		ERRSV5:	JSR	PC,CRLF		
4886	023374	004737	021172			JSR	PC,TYPOUT		
4887	023400	016475				EMSG9			
4888	023402	013700	016706			MOV	ERRDIS,RO		
4889	023406	004737	021424			JSR	PC,BASE10		
4890	023412	032777	100000	173164	ERHALT:	BIT	#10,000,@SWR		
4891	023420	001004				BNE	NOHALT		
4892	023422	013700	016706			MOV	ERRD S,RO	:DISPLAY ERROR COUNT	
4893	023426	004737	017704			JSR	PC,MONIT	:GO TO SWR	
4894	023432	004737	020174		NOHALT:	JSR	PC,RSTREG		
4895	023436	062716	000002			ADD	#2,(SP)		
4896	023442	000207				RTS	PC		
4897	023444	000000			BUFF1:	0			
4898		001000				.END	START	:PROGRAM END, SELF-START.	

A = 020000	BKCOL 004642	CHRFIN 021114	FOR11 002400	HSWR = 177570
ADDR 002172	BK0 004360	CHSR 016564	FOR11D 002427	ILLCHR 021074
ADDRES 016704	BK1 004372	CHWORD 016654	FOR2 002411	ILLVEC 017364
ALL 002462	BK2 004466	CNVFLG 016734	FOR2D 002440	ILVMSG 015376
ASK56 014522	BK4 004400	COUNT1 001412	FOP3 002410	INTFLG 001420
BACO 002426	BK5 004520	COUNT2 001414	FOR3D 002437	INTVEC 016576
BAC1 002425	BK6 004540	CRLF 021136	FOR4 002407	JM600 016612
BAC1D 002454	BK7 004572	CSP 016556	FOR4D 002436	LDADPS 004026
BAC10 002414	BK8 004600	CX 003126	FOR5 002406	LDCH 001746
BAC10D 002443	BLINK 004640	CO 003144	FOR5D 002435	LDCH1 001752
BAC11 002413	BLTR6V 012456	C1 003176	FOR6 002405	LDCH8V 012550
BAC11D 002442	BLTR8V 012272	C2 003230	FOR6D 002434	LDPS 004034
BAC2 002424	BUFF1 023444	C3 003262	FOR7 002404	LD6X6 011224
BAC2D 002453	BUSGET 022050	C4 003314	FOR7D 002433	INDEX 021010
BAC3 002423	BUSSE1 022100	C5 003346	FOR8 002403	LINDL1 021030
BAC3D 002452	BUSSE2 022152	C6 003400	FOR8D 002432	LINECH 021040
BAC4 002422	BUSSE3 022166	C7 003432	FOR9 002402	LKS = 177546
BAC4D 002451	BUSSE4 022226	D = 040000	FOR9D 002431	LOWCHR 016742
BAC5 002421	BUSSE5 022272	DATA 016700	FRMMSG 015430	LSIFLG 017044
BAC5D 002450	BUSSE6 022302	DBUF 016560	FSAVPW 016752	L525 014514
BAC6 002420	BUSSE7 022326	DECMMSG 016540	FSTCNT 016616	MASK 016646
BAC6D 002447	BUSS1A 022144	EMSG1 016354	F1 002012	MAX 002170
BAC7 002417	BUSS10 022336	EMSG2 016360	F2 002076	MODADM 015654
BAC7D 002446	BUSS11 022412	EMSG3 016372	F3 002006	MODADR 016642
BAC8 002416	BUSS12 022520	EMSG4 016403	F4 002020	MODIFY 021522
BAC8D 002445	BUSS13 022540	EMSG5 016414	F4B 002044	MODI1 021552
BAC9 002415	BUSS3A 022222	EMSG6 016425	F5 002134	MODI2 021576
BAC9D 002444	BUS11B 022434	EMSG7 016443	F6 002154	MODI3 021616
BAD 016676	C = 004000	EMSG8 016456	G = 100000	MODI4 021730
BADPRI 016165	CALLPC 016712	EMSG9 016475	GENER 022606	MODI5 021766
BAKND 002174	CAR 016562	ENDIT 017224	GENER1 022730	MODPRM 015665
BAMSG 015672	CARX 016566	ERHALT 023412	GENEX 023066	MODSAV 016644
BASADD 016650	CARY 016570	ERR 016656	GENINC 023060	MODSPA 015661
BASE1A 021442	CAR1 002614	ERRARG 016710	GENISH 022736	MODXIT 022006
BASE1B 01456	CAR1B 002644	ERRDIS 016706	GENRAN 022740	MONIT 017704
BASE1C 021474	CAR2 002676	ERROR 023104	GENROT 022714	MONITA 017776
BASE1D 021454	CAR3 002600	EPRSV1 023240	GENRO 022702	MONITX 020004
BASE10 021424	CAR4 002742	ERRSV2 023266	GENR1 022672	NODEFM 016252
BASE11 016530	CAR5 002756	ERRSV3 023314	GENSEL 022624	NOHALT 023432
BASM10 021406	CAR5A 003006	ERRSV4 023342	GENO 022644	NOMEMA 015537
BAS10A 021432	CAR6 003040	ERRSV5 023370	GFN1 022650	NPRMSG 016076
BCCHAR 016662	CAR7 003072	FADR 017314	GEN25 022664	NXMADR 016722
BCOUNT 016754	CA1170= 177716	FASTSW 017444	GEN52 022656	NXMSG 016135
BELL 017324	CHAR 016574	FILL 017064	GETCH1 020474	NXMTRP 022572
BELLS 001422	CHARC 003532	FILL1 017116	GETCH2 020506	NXTVMS 015504
BELL1 017336	CHAR1 003552	FILL2 017130	GETCH3 020620	OCTIN 020272
BF1 002222	CHAR2 003572	FIRVMS 015450	GETCH4 020664	OCTMSG 016546
BF2 002306	CHAR3 003612	FORGND 001764	GETSTR 020470	OCTNUM 016520
BF3 002216	CHAR4 003632	FORO 002413	GOMSG 014630	ODAMSG 015765
BF4 002230	CHAR5 003652	FOR1 002412	GOOD 016674	OVAMSG 016010
BF4B 002254	CHAR6 003672	FOR1D 002441	HALF6V 014516	PARITY 016660
BF5 002344	CHAR7 003712	FOR10 002401	HALF8V 014520	PARO = 172340
BF6 002364	CHDR 016572	FOR10D 002430	HCNT 012570	PAR1 = 172342

PAR2 = 172344	RANEX 023056	STRADD 016736	TYPD1 016640	T126 012622
PAR3 = 172346	RANGEN 022760	STRLEN 016740	TYPOTA 016634	T128 012612
PAR4 = 172350	RANMTA 016670	SWR 016604	TYPOTB 020276	T130WW 014116
PAR5 = 172352	RANSEC 023044	SWRMSG 015642	TYPOTC 020356	T130WY 014132
PAR6 = 172354	RANSEL 016666	SWRSET 017046	TYPOTD 020412	T13000 013614
PAR7 = 172356	RANST 016672	TABLE 001272	TYPOTE 020430	T13001 013634
PASMSG 015360	RAN1 022772	TABLE1 001324	TYPOTF 021172	T13002 013666
PCHR 021250	RAN2 023010	TEMPX 004050	T100WW 004616	T14000 014154
PDR0 = 172300	RAN4 023046	TEMPX1 004054	T10000 004102	T14001 014174
PDR1 = 172302	READ 020444	TEMPY 004052	T10001 004122	T14003 014304
PDR2 = 172304	REDMES 015607	TEMPY1 004056	T10003 004154	T14004 014336
PDR3 = 172306	REPCNT 016610	TESMSG 015347	T10004 004200	T60WW 001726
PDR4 = 172310	REPCT1= 001000	TESTNO 016602	T10005 004204	T6000 001442
PDR5 = 172312	REPCT2= 000100	TESTR 016756	T10006 004206	T6001 001462
PDR6 = 172314	REPCT3= 000002	TEST10 004064	T10007 004212	T6002 001514
PDR7 = 172316	ROWS 004062	TEST11 004652	T10008 004252	T6003 001544
PMSG1 021206	RSTART 001200	TEST12 005434	T10009 004256	T6004 001614
PMSG2 021232	RSTREG 020174	TEST13 013576	T10010 004266	T70WW 003512
PMSG3 021236	RUBCHR 020742	TEST14 014136	T110WW 005114	T7000 002506
PMSG4 021242	RUBFLG 016746	TEST6 001424	T110WY 005130	T7001 002540
PRCT1A 021304	R6 = %000006	TEST7 002470	T11000 004670	UPPCHR 016744
PRESUB 005134	R7 = %000007	TIME 004024	T11001 004710	VAMSG 015726
PRES1 005150	S = 010000	TIME1 001356	T1102 004742	VCNT 012566
PRE0 005372	SAVEXM 016724	TIME1A 001410	T1103 004756	VECLEV 016600
PRE1 005376	SAVLT4 016730	TKB = 177562	T12H 012572	VECTOR 017056
PRE2 005403	SAVLT6 016732	TKS = 177560	T12SLD 012632	V66H1 013362
PRE3 005411	SAVPAR 016726	TLBR6V 012364	T12V 012602	V68H1 013142
PRE4 005417	SAVPC 016626	TLBR8V 012200	T120WW 012160	V68H11 013206
PRE5 005425	SAVPC1 016630	TMPADD 004060	T12000 005452	V68H21 013252
PRMSG 016037	SAVREG 020076	TPB = 177566	T12001 005472	V68H41 013302
PRNT3 021354	SAVSTA 016632	TPREDY 017434	T12004 005522	V68H51 013332
PROCT 021262	SECOND 001416	TPS = 177564	T12005 005642	V86H1 012642
PROCT1 021272	SE156 014420	TRAPSV 017464	T12006 006052	V86H11 012722
PROCT2 021306	SET56A 014464	TRPARG 016620	T12007 006322	V86H21 013002
PROCT3 021340	SILLSI 017002	TRPBAK 017672	T12008 006550	V86H41 013042
PSW = 177776	SRO = 177572	TRPERR 016652	T12010 007034	V86H51 013102
QEXIT 020022	SR1 = 177574	TRPLP 017626	T12011 007244	V88H1 013456
QEXIT1 020044	SR2 = 177576	TRPMEM 016624	T12012 007514	WAITT 003732
QEXIT2 020046	SSWR 016606	TRPSCP 017602	T12013 007742	WMSG 015276
QEXIT3 020062	START 001000	TRPSEL 016622	T12014 010366	XHATCH 014342
RANCLC 023040	START1 001226	TRXEXM 016720	T12015 010722	XMES 014352
RAND 016636	START2 001232	TRXPAR 016716	T12016 011336	SENDAD 017300
RANDC 016750	START3 001266	TRXVAD 016714	T12017 011672	= 023446
RANDN 016664	STATUS 016702	TYPCTC 020014	T12018 012142	

. ABS. 023446 000

ERRORS DETECTED: 0

CVVTBA.BIN, CVVTBA.SEQ=CVVTBA.SRC
 RUN-TIME: 15 29 1 SECONDS
 RUN-TIME RATIO: 197/45=4.3
 CORE USED: 12K (23 PAGES)