

KWV11-A

DIAGNOSTIC
MD-11-DVKWA-B

EP DVKWA B DL B

MAR 1977

COPYRIGHT 1977

digital

FICHE 1 OF 1

MADE IN USA

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	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

B01

.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DVKWA-B-D
PRODUCT NAME: KVV11A DIAGNOSTIC
DATE CREATED: OCTOBER 1976
DATE REVISED: JANUARY 1977
MAINTAINER: DIAGNOSTIC ENGINEERING

COPYRIGHT (C) 1976, 1977
DIGITAL EQUIPMENT CORPORATION

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY 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 EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

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

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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

TABLE OF CONTENTS

1.0	ABSTRACT
2.0	REQUIREMENTS
2.1	EQUIPMENT
2.2	STORAGE
3.0	LOADING PROCEDURE
3.1	METHOD
3.2	NON-STANDARD ADDRESS, VECTOR, OR USE OF SOFTWARE SWITCH REGISTER
4.0	STARTING PROCEDURE
4.1	CONTROL SWITCH SETTINGS
4.2	STARTING ADDRESS
4.3	PROGRAM AND/OR OPERATOR ACTION
5.0	OPERATING PROCEDURE
5.1	SWITCH REGISTER FUNCTION
5.2	SCOPE LOOPS
5.3	PROGRAM AND/OR OPERATOR ACTION
5.3.1	LOGIC TEST
5.4	INHIBITING AUTO-SIZE FEATURE
6.0	ERRORS
6.1	ERROR PRINTOUT
6.1.1	EXAMPLE
6.2	NON-STANDARD ERROR HALTS
7.0	RESTRICTIONS
7.1	EXTERNAL INPUTS
7.2	STARTING RESTRICTION
7.3	POSSIBLE PROGRAM "BOMBS"
8.0	MISCELLANEOUS
8.1	POWER FAIL
8.2	XXDP, ACT, APT
8.3	EXECUTION TIME
8.4	LSI-11 "ODT" COMMANDS
8.5	ENTERING LSI-11 "ODT"
8.6	USE OF PROGRAM SOFTWARE SWR
8.7	SPECIAL I/O SIGNAL TESTS
8.8	PRODUCTION STARTING ADDRESS
8.9	TESTOR STARTING ADDRESS
8.10	TRAP CATCHER

46
47
48
49
50
51
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

1.0 ABSTRACT

THIS PROGRAM ALLOWS THE USER CHECK-OUT OR DEBUG THE KVV11A, PROGRAMMABLE REAL-TIME CLOCK. THE LOGIC TEST IS SELF CONTAINED AND NEEDS NO EXTERNAL MAINTENANCE HARDWARE OR OPERATOR INTERVENTION WITH ONLY ONE EXCEPTION: IF THE CUSTOMER HARDWARE CONNECTED TO THE KVV11 COULD INJECT SIGNALS ON ST2, ST1, OR SLAVE IN INPUTS, IT MUST BE DISCONNECTED.

EVEN THOUGH THE KVV11 IS A Q BUS OPTION, THIS PROGRAM WAS DESIGNED TO RUN ON ANY PDP-11 FAMILY COMPUTER. IF THE USER IS UNFAMILAR WITH AN LSI-11 HE SHOULD REVIEW SECTIONS 8.4 AND 8.5. A SOFTWARE SWITCH REGISTER IS INCLUDED WITH THIS PROGRAM.

EVERY EFFORT WAS MADE TO MAKE THIS PROGRAM CONFORM TO LSI-11 PROGRAMMING RESTRICTIONS, HOWEVER; THE USER SHOULD READ SECTIONS 7.2 AND 7.3.

2.0 REQUIREMENTS

2.1 EQUIPMENT

1. PDP-11 FAMILY COMPUTER WITH 8K OF MEMORY (OR MORE) AND I/O FACILITIES (I.E., TTY).
2. KVV11 UNDER TEST.

2.2 STORAGE

THIS PROGRAM OCCUPIES AND USES 8K OF MEMORY.

3.0 LOADING PROCEDURE

3.1 METHOD

STANDARDS PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED.

1. ABSOLUTE LOADER MUST BE IN MEMORY.
2. PLACE BINARY TAPE IN READER.
3. TYPE ADDRESS *7500 (5* DETERMINE BY LOCATION OF LOADER).
4. TYPE "G" (PROGRAM WILL BE LOADED INTO MEMORY).

THE PROGRAM CAN ALSO BE LOADED BY XXDP, ACT, OR APT.

3.2 NON-STANDARD ADDRESS, VECTOR, OR USE OF SOFTWARE SWITCH REGISTER

THIS PROGRAM IS SET TO TEST A KVV11 WITH A STANDARD ADDRESS AND VECTOR. IF ANY OF THESE ARE DIFFERENT ON THE KVV11K YOU ARE TESTING, CHANGE THE CORRESPONDING LOCATION IN MEMORY BEFORE STARTING THIS TEST.

LOCATION	TAG	CURRENT CONTENTS	COMMENTS
1250	\$BASE:	170420	;; BASE ADDRESS OF EQUIPMENT
1244	\$VECT1:	000440	;; UNDER TEST
176	\$SWREG:	000000	;; INTERRUPT VECTOR #1
1157	\$TPFLG:	.BYTE 0	;; MANUAL SWR.
			;; "TERMINAL AVAILABLE"
			;; FLAG (BIT<0:7>=0=YES)

179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
2154.0 STARTING PROCEDURE

4.1 CONTROL SWITCH SETTING

BEFORE STARTING THE DIAGNOSTIC, SET ALL SWITCH REGISTER BITS AS
DESIRED, SEE SECTION 5.1.

4.2 STARTING ADDRESSES

200 START OF LOGIC TESTS
204 RESTART ADDRESS FOR LOGIC TEST
210 I/O SIGNAL TEST #1
214 I/O SIGNAL TEST #2
220 I/O SIGNAL TEST #3
230 PRODUCTION STARTING ADDRESS
240 TESTOR STARTING ADDRESS

4.3 PROGRAM AND/OR OPERATOR ACTION

1. LOAD PROGRAM INTO MEMORY.
2. ENTER KEYBOARD "ODT".
3. ALTER LOCATION "\$SWREG" (ADDRESS 176) TO REFLECT DESIRED
OPTIONS OF A SWITCH REGISTER - SEE SECTION 5.1.
4. TYPE STARTING ADDRESS, FOLLOWED BY "G" TO START PROGRAM.

5.0 OPERATING PROCEDURE

5.1 SWITCH REGISTER FUNCTION

SWR BIT	OCTAL	FUNCTION WHEN SET
15	100000	HALT ON ERROR
14	040000	LOOP ON TEST
13	020000	INHIBIT ERROR TYPEOUT
12	010000	ENABLE LINE FREQ. RATE TESTING
11	004000	INHIBIT ITERATIONS (SHORT PASS)
10	002000	BELL ON ERROR
09	001000	LOOP ON ERROR
08	000400	LOOP ON TEST IN SWR <7:0>

5.2 SCOPE LOOPS

IF AN ERROR OCCURS AND THE USER WISHES TO SCOPE THE ERROR, "\$SWREG" SHOULD BE ALTERED TO "100000" AT THE START OF THE TEST TO HALT ON ERROR, THEN WHEN THE PROGRAM HALTS ON ERROR AND THE CPU ENTERS "ODT", "\$SWREG" SHOULD BE ALTERED TO "060000" TO LOOP ON CURRENT TEST AND INHIBIT ERROR TYPEOUT, THEN TYPE "P" TO CONTINUE PROGRAM EXECUTION.

5.3 PROGRAM AND/OR OPERATOR ACTION

5.3.1 LOGIC TEST

THE FIRST PASS THROUGH THE PROGRAM WILL BE MADE WITH ITERATIONS INHIBITED. SUCCESSIVE PASSES WILL ENABLE ITERATIONS IF SWR11=0.

IF NOT INHIBITED BY APT, THE PROGRAM WILL LOOK FOR MORE K WV11'S TO EXERCISE, ONE PASS WILL EXERCISE ALL K WV11'S.

IF FOUR UNITS ARE DETECTED, THE FOLLOWING WILL BE TYPED:

UNIT #000001 COMPLETED TESTING UNIT #000002
UNIT #000002 COMPLETED TESTING UNIT #000003
UNIT #000003 COMPLETED TESTING UNIT #000004
UNIT #000004 COMPLETED

AT END OF PASS WHEN ALL UNITS HAVE BEEN TESTED, THE FOLLOWING TYPEOUT WILL OCCUR:

"ENDPASS 12 - TOTAL ERRORS 4 THERE ARE 4 (OCTAL) UNITS -
GOOD UNITS (L TO R) 0000000000001011".

THIS INDICATES THAT THE PROGRAM HAS COMPLETED 12 OCTAL (10 DECIMAL) PASSES. DURING THAT TIME 4(OCTAL) ERRORS WERE DETECTED. ALSO WE TESTED 4 UNITS AND THE THIRD UNIT WAS THE ONLY UNIT TO FAIL.

5.4 INHIBITING AUTO-SIZE FEATURE

THIS PROGRAM WILL AUTOMATICALLY AUTO-SIZE AND TEST EACH K WV11 IT DETECTS ON THE SYSTEM. TO INHIBIT THIS FEATURE, SET BIT 15 OF LOCATION "\$ENVN". ALSO, TO TEST AN INDIVIDUAL K WV11 IN A GROUP, SET THIS BIT AND REFER TO SECTION 3.2 FOR CHANGING THE BASE ADDRESS OF THE K WV11 UNDER TEST.

6.0 ERRORS

6.1 ERROR PRINTOUT

PRINTOUT VARIES WITH THE ERROR DETECTED. THE ERROR PC TYPED OUT IS THE ACTUAL LOCATION OF THE ERROR CALL.

A HALT AT LOCATION "\$TYPE"+10 WHEN RUNNING WITH NO TERMINAL INDICATES AN ERROR HAS OCCURRED. TO FIND OUT THE NUMBER OF THE ERROR, EXAMINE LOCATION "\$STNM". THIS IS THE ITEM NUMBER OF THE ERROR. TO FIND OUT WHAT THE ERROR TYPEOUT WOULD HAVE BEEN GOTO TO THE ERROR POINTER TABLE BEGINNING AT LOCATION "ERRTB".

6.1.1 EXAMPLE

IF WE EXAMINED LOCATION "\$STNM" AND FOUND A 5(101) WE GO TO LOCATION "\$ERRTB" AND LOOK THROUGH THE ERROR POINTER TABLE UNTIL WE FOUND ITEM 5. THE INFORMATION WOULD LOOK LIKE:

;ITEM 5

EMS	;CLOCK SR DATA ERROR
DHS	;ERRPC ASR WAS S/B
DTS	;SERRPC,ASR,\$BDDAT,\$GDDAT
DFO	;ALL NUMBERS ARE IN OCTAL FORM

TO FIND OUT THE INFORMATION SPECIFIED BY DTS (\$ERRPC,BSR,\$BDDAT,\$GDDAT) FOLLOW THESE STEPS:

1. LOOK UP THE ADDRESS OF THE LABEL (I.E., \$ERRPC) IN THE SYMBOL TABLE WHICH FOLLOWS THE LISTING.
2. * PUT THIS ADDRESS IN THE SWITCH REGISTER AND DEPRESS THE LOAD ADDRESS SWITCH ON THE PROCESSOR'S CONSOLE.
3. * NOW DEPRESS THE EXAMINE SWITCH.
4. * THE DATA DISPLAYED IN THE DATA LIGHTS IS THE INFORMATION THAT WOULD HAVE BEEN PRINTED FOR HIS LABEL IF YOU HAD A INPUT/OUTPUT TERMINAL.

* SEE SECTION 8.4 FOR LSI-11 ODT COMMANDS.

6.2 NON-STANDARD ERROR HALTS

BUS ERRORS WILL CAUSE A HALT TO THE ROUTINE "IOTRD". THE ADDRESS THAT CAUSED THIS TRAP WILL BE IN ADDRESS "TRTO".

7.0 RESTRICTIONS

7.1 EXTERNAL INPUTS

EXTERNAL INPUTS SUCH AS "SLAVE IN", "ST1" AND "ST2" MUST NOT BE CONNECTED TO ANY CUSTOMER HARDWARE THAT MIGHT GENERATE THESE SIGNAL WHILE THE DIAGNOSTIC IS RUNNING.

7.2 STARTING RESTRICTION

IF A FREE-RUNNING CLOCK, SUCH AS 60HZ FROM THE POWER SUPPLY, IS ATTACHED TO THE "BEVNT" BUS LINE ON BOTH REV LEVEL C/D AND E SYSTEMS, AN INTERRUPT TO LOCATION 100 WILL OCCUR WHEN USING THE "G" AND "L" COMMANDS PRIOR TO EXECUTING THE FIRST INSTRUCTION. THEREFORE THIS PROGRAM CAN NOT DISABLE THE BEVNT BUS LINE BY INHIBITING INTERRUPTS.

USER SYSTEMS REQUIRING A FREE-RUNNING CLOCK ATTACHED TO THE BEVNT BUS LINE CAN TEMPORARILY AVOID THIS SITUATION BY SETTING THE PSW(R5) TO 200, LOADING THE PC WITH THE STARTING ADDRESS INSTEAD OF USING THE "G" COMMAND, AND THEN USING THE "P" COMMAND. BEFORE USING THE "L" COMMAND, THE PSW(R5) CAN BE SET TO 200, THEREBY INHIBITING INTERRUPTS, TO AVOID RECEIVING THE EVENT INTERRUPT AFTER LOADING THE ABS LOADER.

340
341
342
343
344
345
346
347
348
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

7.3 POSSIBLE PROGRAM "BOMBS"

THE FIRST TWO TESTS OF THIS PROGRAM CHECK TO SEE IF THE KVV11 RESPONDS TO THE ADDRESS THE PROGRAM THINKS ITS AT. IF THE KVV11 DOES NOT RESPOND, A BUS ERROR OCCURS. ALSO BUS ERRORS CAN OCCUR DURING THE TIME THE PROGRAM SIZES TO SEE HOW MANY KVV11 ARE ON YOUR SYSTEM.

FOR MORE INFORMATION ON THE NEXT SUBJECT, SEE JAN. 1976 LSI-11 ENGINEERING BULLETIN ISSUED BY THE DIGITAL COMPONENTS GROUP.

BUS ERRORS MAY ALTER THE PRESET CONTENTS OF LOCATION 4 BEFORE THE TRAP IS EXECUTED, THEREBY TRANSFERRING PROGRAM CONTROL TO AREA IN THE PROGRAM THAT WAS NOT SET UP TO HANDLE THE TRAP. IF THIS HAPPENS, THE PROGRAM WILL "BOMB" AND POSSIBLY REWRITE PARTS OF ITSELF.

8.0 MISCELLANEOUS

8.1 POWER FAIL

AFTER A POWER FAILURE OCCURS, THE PROGRAM EXECUTION WILL CONTINUE AT THE POINT WHERE THE POWER OCCURRED. THE PROGRAM WILL TYPE "POWER".

8.2 XXDP, ACT, APT

THE PROGRAM IS CHAINABLE UNDER XXDP, ACT, OR APT. ALTHOUGH "APT HOOKS" HAVE BEEN INSTALLED, THEY HAVE NOT BEEN TESTED.

8.3 EXECUTION TIME

0.5 MINUTES (30 SEC) ITERATION INHIBITED - NO ERRORS
2.5 MINUTES (150 SEC) WITH ITERATIONS - NO ERRORS

8.4 LSI-11 "ODT" COMMANDS

FORMAT -----	DESCRIPTION -----
<CR> RETURN	CLOSE OPENED LOCATION AND ACCEPT NEXT COMMAND.
<LF> LINE FEED	CLOSE CURRENT LOCATION; OPEN NEXT SEQUENTIAL LOCATION.
↑(UPARROW)	OPEN PREVIOUS LOCATION.
+ (LEFT ARROW)	TAKE CONTENTS OF OPENED LOCATION, INDEXED BY CONTENTS OF PC, AND OPEN THAT LOCATION.
@	TAKE CONTENTS OF OPENED LOCATION AS ABSOLUTE ADDRESS AND OPEN THAT LOCATION.
R/	OPEN THE WORD AT LOCATION R.
/	REOPEN THE LAST LOCATION.
\$N/ OR RN/	OPEN GENERAL REGISTER N(0-7) OR S(PS REGISTER).
R;G OR RG	GOTO LOCATION R AND START PROGRAM.
NL	EXECUTE BOOTSTRAP LOADER USING N AS DEVICE CSR. CONSOLE DEVICE IS 177560.
;P OR P	PROCEED WITH PROGRAM EXECUTION.
RUBOUT	ERASES PREVIOUS NUMERIC CHARACTER. RESPONSE IS A BACKSLASH ().

8.5 ENTERING LSI-11 "ODT"

THE HALT OR ODT MICROCODE STATE OF THE KD11F (LSI-11 MODULE) CAN BE ENTERED IN FIVE DIFFERENT WAYS (OTHERS ARE A SUBSET OF THESE) FROM THE RUN STATE:

1. EXECUTION OF A LSI-11 HALT INSTRUCTION,
2. A DOUBLE BUS ERROR,
3. AS A POWER UP OPTION,
4. ASCII BREAK WITH DLV11 FRAMING ERROR ASSERTING THE B HALT

MO1

MAINDEC-11-DVKWA-B
DVKWAB.P11

MACY11 27(665) 21-FEB-77 14:43 PAGE 12

479
480
481
482

LINE (ENABLED BY JUMPER OF DLV11).

UPON ENTERING THE HALT STATE, THE KD11F RESPONDS THROUGH THE SET
OF COMMAND LISTED IN SECTION 8.4.

8.6 USE OF PROGRAM SOFTWARE SWR

THE PROGRAM SOFTWARE SWITCH REGISTER IS ENABLED IF

1. NO HARDWARE SWR EXISTS;
2. IF YOU START WITH ALL ONES (SWR=177777) IN THE SWITCH REGISTER.

THE SOFTWARE SWITCH REGISTER MAY BE CHANGED BY TYPING $\uparrow$ G (CONTROL AND LETTER G KEYS TYPED SIMULTANEOUSLY). WHEN $\uparrow$ G IS TYPED, THE PROGRAM RESPONDS BY TYPING "SWR=XXXXXX" WHERE XXXXXX EQUALS THE FORMER CONTENTS OF THE SWITCH REGISTER.

IF YOU WISH TO KEEP THE CURRENT VALUE, TYPE <CR>. IF YOU WISH TO CHANGE THE VALUE, TYPE THE NEW VALUE FOLLOWED BY A <CR>.

IT IS IMPORTANT TO NOTE THAT THE DIAGNOSTIC IS NOT RUNNING AFTER THE $\uparrow$ G UNTIL A <CR> IS TYPED.

8.7 SPECIAL I/O SIGNAL TESTS

THREE TESTS WERE INCLUDED TO ENABLE CHECKOUT OF I/O SIGNALS: ST1, ST2, AND CLOCK OVERFLOW. THESE TESTS HAVE A SPECIAL STARTING ADDRESS. SINCE END-PASSES ARE IMMEDIATE, NO "END OF PASS" MESSAGE IS REPORTED. ERRORS ARE REPORTED BY TYPING OUT THE PC WHERE THE ERROR WAS DETECTED. WHEN STARTED, THE PROGRAM REMAINS IN A LOOP GENERATING AND DETECTING THE SPECIFIED SIGNALS. HALT ON ERROR AND INHIBIT ERROR TIMEOUT OPTIONS MAY BE USED.

LOGIC TEST MUST HAVE ALREADY BEEN RUN ON THE KVV11.

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

8.7.1 I/O SIGNAL TEST #1 ST1 IN, ST2 OUT

SWITCH PACK S2 MUST BE SET UP AS FOLLOWS:

SWITCH STATE

1	OFF
2	ON
3	OFF
4	OFF
5	ON
6	ON
7	NOT USED

THE FOLLOWING JUMPER MUST BE INSTALLED.

J1-SS (ST2 OUT) TO J1-VV (ST1 IN)

LOAD AND START THE PROGRAM AT 210.

8.7.2 I/O SIGNAL TEST #2 CLOCK OVERFLOW TEST

SWITCH PACK S2 MUST BE SET UP AS FOLLOWS:

SWITCH STATE

1	OFF
2	OFF
3	OFF
4	ON
5	ON
6	OFF
7	NOT USED

THE FOLLOWING JUMPER MUST BE INSTALLED.

J1-RR (CLOCK OVERFLOW) TO J1-TT (ST2 IN)

LOAD AND START AT LOCATION 214.

519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564

8.7.3 I/O SIGNAL TEST #3 ST1 OUT AND ST2 IN
SWITCH PACK S2 MUST BE SET UP AS FOLLOWS:

SWITCH STATE

1	OFF
2	OFF
3	OFF
4	ON
5	ON
6	ON
7	NOT USED

THE FOLLOWING JUMPER MUST BE INSTALLED:

J1-UU (ST1 OUT) TO J1-TT (ST2 IN)

LOAD AND START AT LOCATION 220.

8.8 PRODUCTION STARTING ADDRESS

A SPECIAL STARTING ADDRESS HAS BEEN PROVIDED FOR IN-HOUSE PRODUCTION TO USE TO START THE LOGIC DIAGNOSTIC AND INFORM THE TEST THAT PRODUCTION IS USING IT.

IN THE FIELD ONLY ENOUGH ADDRESSES WERE ALLOTTED FOR 4 SEQUENTIAL KVV11S. WHEN THE LOGIC TESTS ARE STARTED AT LOCATION 200, WE ONLY AUTO-SIZE UP TO 4 KVV11S.

IN HOUSE TESTING MAY WISH TO EXERCISE UP TO 16 KVV11S AT ONE TIME. THE LOGIC TESTS MAY BE STARTED AT LOCATION 230 AND THE PROGRAM WILL AUTO SIZE UP TO 16 KVV11S.

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
601
602
603

8.9 TESTOR STARTING ADDRESS

A SPECIAL STARTING ADDRESS HAS BEEN PROVIDED FOR MANUFACTURING TO USE TO START THE LOGIC DIAGNOSTIC AND INFORM THE PROGRAM THAT THE CLOCK MODULE IS CABLED TO AN IN-HOUSE TESTOR.

MANUAL INTERVENTION IS NEEDED IN THIS SEQUENCE OF TESTING. THE PROGRAM WILL TYPE OUT ALL INSTRUCTIONS. A CABLE SHOULD CONNECT J1 ON THE CLOCK MODULE TO J10 ON THE TESTOR. SWITCHES 1 AND 3 OF S2 (ON THE CLOCK MODULE) SHOULD BE ON, ALL OTHER SWITCHES ON S2 SHOULD BE OFF.

8.10 TRAP CATCHER

THE TRAP CATCHER IN THIS DIAGNOSTIC EMPLOYS A NEW CONCEPT. THIS CONCEPT WILL ENABLE THE USER OF THIS DIAGNOSTIC TO GAIN MORE KNOWLEDGE OF THE EVENTS THAT LEAD THE PROGRAM TO THIS AREA.

THE TRAP CATCH CONSISTS OF PC+2 AND JSR PC,RO. (I.E., LOCATION 300 WOULD CONTAIN 302 AND LOCATION 302 WOULD CONTAIN 4700.)

WHEN A DEVICE INTERRUPTS UNEXPECTEDLY TO THE TRAP CATCHER, IT WOULD PICK UP THE PC+2 OF THE TRAP AS AN ADDRESS OF THE INTERRUPT SERVICE ROUTINE.

THE PROGRAM WOULD THEN PICK UP "4700" AS THE NEW PSW. BIT 7 OF THE NEW PSW HAVING BEEN SET, WOULD CAUSE FURTHER INTERRUPTS FROM HAPPENING. WHEN THE CPU ATTEMPTS TO EXECUTE "4700" (JSR PC,RO), A BUSS-TIME-OUT TRAP WILL OCCUR TO LOCATION 4. LOCATION 4 CONTAINS A POINTER TO "IOTRD", A ROUTINE THAT WILL REPORT THE TRAP AS AN ERROR.

TO GUARD AGAINST "REAL" BUS ERRORS ROUTING US THROUGH LOCATION 4 TO "IOTRD", WE CHECK TO SEE IF THE TRAP THAT BROUGHT US TO LOCATION 4 REALLY CAME FROM THE TRAP CATCHER AREA. IF NOT WE'LL HALT AND LEAVE THE TRAP ADDRESS IN "TRTO".

MORE ABOUT THE INTERRUPT ERROR CAN BE FOUND IN THE DESCRIPTION OF THE ERROR IN THE PROGRAM LISTING IN THE ROUTINE "IOTRD".

```
%
.NLIST MC,MD,CND
.LIST ME
.ENABL ABS
.ENABL AMA
.MCALL .HEADER,.SETUP,.SETTRAP,.TRMTRP,.STRAP,.SRDOCT,.STYPBIN
.MCALL TYPOCS,.SPOWER,.SCATCH,.STYPOCT,.EQUAT,.SCMTAG,.SWRHI
.MCALL .SEOP,.SERROR,.SERRTYP,.STYPDEC,.SSCOPE,.SREAD,.STYPE
.MCALL .SACT11,.S OR,.SAPTYP
$SWR= 167400
```

167400

```

658
659
660
661
662
663
664
665 .TITLE MAINDEC-11-DVKWA-B
666 ;*COPYRIGHT (C) 1976
667 ;*DIGITAL EQUIPMENT CORP.
668 ;*MAYNARD, MASS. 01754
669 ;*
670 ;*PROGRAM BY EDWARD C. BADGER
671 ;*
672 ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
673 ;*PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
674 ;*
675 000001 $TN=1
676
677 .SBTTL OPERATIONAL SWITCH SETTINGS
678 ;*
679 ;* SWITCH USE
680 ;* -----
681 ;* 15 HALT ON ERROR
682 ;* 14 LOOP ON TEST
683 ;* 13 INHIBIT ERROR TYPEOUTS
684 ;* 11 INHIBIT ITERATIONS
685 ;* 10 BELL ON ERROR
686 ;* 9 LOOP ON ERROR
687 ;* 8 LOOP ON TEST IN SWR<7:0>
688
689 .SBTTL TRAP CATCHER
690
691 000000 .=0
692 ;*ALL UNUSED LOCATIONS FROM 4-776 CONTAIN A ".+2"
693 ;*AND "JSR PC,RO" SEQUENCE TO CATCH ILLEGAL INTERRUPTS.
694 ;*AND INTERRUPTS TO THE WRONG VECTOR.
695 ;*LOCATION 0 CONTAINS A 0 TO CATCH IMPROPERLY LOADED
696 ;*VECTORS.
697
698 000004 000004 000200 .=4
699 000174 000174 .WORD IOTRD,200 ;HANDLE BUSS ERROR.
700 000174 000000 DISPREG: .WORD 0 ;SOFTWARE DISPLAY REGISTER.
701 000176 000000 SWREG: .WORD 0 ;SOFTWARE SWITCH REGISTER.
702 000100 000100 .=100
703 000100 000104 000200 000002 .WORD 104,200,2 ;IF "B EVENT"ON Q-BUS IS
704 ;CONNECTED,WE NEED A WAY OF
705 ;IGNORING ITS INTERRUPTS.
706
707 000200 000200 .=200
708 000200 000137 001474 JMP @#START
709 000204 000137 002104 JMP @#RSTART
710 000210 000137 014006 JMP @#IOTST1
711 000214 000137 014064 JMP @#IOTST2

```

MAINDEC-11-DVKWA-B MACY11 27(665) 21-FEB-77 14:43 PAGE 18
 DVKWAB.P11 TRAP CATCHER

```

712 000220 000137 014132      JMP      @#IOTST3
713
714      000230      . =230
715 000230 000137 001460      JMP      @#WSTART      ;WESTFIELD STARTING ADDRESS
716      000240      . =240
717 000240 000137 001444      JMP      @#TSTSTR      ;ALL TESTER TESTS
718                                     ;IF STARTED HERE.
719                                     ;ALLOWS PRODUCTION TO EXERCISE
720                                     ;UP TO 16 CLOCKS.NORMAL=4.
721
722      .SBTTL  BASIC DEFINITIONS
723
724      ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
725      001100      STACK= 1100
726      .EQUIV  EMT,ERROR      ;;BASIC DEFINITION OF ERROR CALL
727      .EQUIV  IOT,SCOPE      ;;BASIC DEFINITION OF SCOPE CALL
728
729      ;*MISCELLANEOUS DEFINITIONS
730      000011      HT= 11      ;;CODE FOR HORIZONTAL TAB
731      000012      LF= 12      ;;CODE FOR LINE FEED
732      000015      CR= 15      ;;CODE FOR CARRIAGE RETURN
733      00C200      CRLF= 200    ;;CODE FOR CARRIAGE RETURN-LINE FEED
734      177776      PS= 177776   ;;PROCESSOR STATUS WORD
735      .EQUIV  PS,PSW
736      177774      STKLMT= 177774 ;;STACK LIMIT REGISTER
737      177772      PIRQ= 177772 ;;PROGRAM INTERRUPT REQUEST REGISTER
738      177570      DSWR= 177570 ;;HARDWARE SWITCH REGISTER
739      177570      DDISP= 177570 ;;HARDWARE DISPLAY REGISTER
740
741      ;*GENERAL PURPOSE REGISTER DEFINITIONS
742      000000      R0= %0      ;;GENERAL REGISTER
743      000001      R1= %1      ;;GENERAL REGISTER
744      000002      R2= %2      ;;GENERAL REGISTER
745      000003      R3= %3      ;;GENERAL REGISTER
746      000004      R4= %4      ;;GENERAL REGISTER
747      000005      R5= %5      ;;GENERAL REGISTER
748      000006      R6= %6      ;;GENERAL REGISTER
749      000007      R7= %7      ;;GENERAL REGISTER
750      000006      SP= %6      ;;STACK POINTER
751      000007      PC= %7      ;;PROGRAM COUNTER
752
753      ;*PRIORITY LEVEL DEFINITIONS
754      000000      PR0= 0      ;;PRIORITY LEVEL 0
755      000040      PR1= 40     ;;PRIORITY LEVEL 1
756      000100      PR2= 100    ;;PRIORITY LEVEL 2
757      000140      PR3= 140    ;;PRIORITY LEVEL 3
758      000200      PR4= 200    ;;PRIORITY LEVEL 4
759      000240      PR5= 240    ;;PRIORITY LEVEL 5
760      000300      PR6= 300    ;;PRIORITY LEVEL 6
761      000340      PR7= 340    ;;PRIORITY LEVEL 7
762
763      ;*"SWITCH REGISTER" SWITCH DEFINITIONS
764      100000      SW15= 100000
765      040000      SW14= 40000

```

766	020000	SW13=	20000
767	010000	SW12=	10000
768	004000	SW11=	4000
769	002000	SW10=	2000
770	001000	SW09=	1000
771	000400	SW08=	400
772	000200	SW07=	200
773	000100	SW06=	100
774	000040	SW05=	40
775	000020	SW04=	20
776	000010	SW03=	10
777	000004	SW02=	4
778	000002	SW01=	2
779	000001	SW00=	1
780		.EQUIV	SW09,SW9
781		.EQUIV	SW08,SW8
782		.EQUIV	SW07,SW7
783		.EQUIV	SW06,SW6
784		.EQUIV	SW05,SW5
785		.EQUIV	SW04,SW4
786		.EQUIV	SW03,SW3
787		.EQUIV	SW02,SW2
788		.EQUIV	SW01,SW1
789		.EQUIV	SW00,SW0
790			
791		;*DATA	BIT DEFINITIONS (BIT00 TO BIT15)
792	100000	BIT15=	100000
793	040000	BIT14=	40000
794	020000	BIT13=	20000
795	010000	BIT12=	10000
796	004000	BIT11=	4000
797	002000	BIT10=	2000
798	001000	BIT09=	1000
799	000400	BIT08=	400
800	000200	BIT07=	200
801	000100	BIT06=	100
802	000040	BIT05=	40
803	000020	BIT04=	20
804	000010	BIT03=	10
805	000004	BIT02=	4
806	000002	BIT01=	2
807	000001	BIT00=	1
808		.EQUIV	BIT09,BIT9
809		.EQUIV	BIT08,BIT8
810		.EQUIV	BIT07,BIT7
811		.EQUIV	BIT06,BIT6
812		.EQUIV	BIT05,BIT5
813		.EQUIV	BIT04,BIT4
814		.EQUIV	BIT03,BIT3
815		.EQUIV	BIT02,BIT2
816		.EQUIV	BIT01,BIT1
817		.EQUIV	BIT00,BIT0
818			
819		;*BASIC	"CPU" TRAP VECTOR ADDRESSES

```

820      000004      ERRVEC= 4      ;; TIME OUT AND OTHER ERRORS
821      000010      RESVEC= 10     ;; RESERVED AND ILLEGAL INSTRUCTIONS
822      000014      TBITVEC=14     ;; "T" BIT
823      000014      TRIVEC= 14     ;; TRACE TRAP
824      000014      BPTVEC= 14     ;; BREAKPOINT TRAP (BPT)
825      000020      IOTVEC= 20     ;; INPUT/OUTPUT TRAP (IOT) **SCOPE**
826      000024      PWRVEC= 24     ;; POWER FAIL
827      000030      EMTVEC= 30     ;; EMULATOR TRAP (EMT) **ERROR**
828      000034      TRAPVEC=34     ;; "TRAP" TRAP
829      000060      TKVEC= 60      ;; TTY KEYBOARD VECTOR
830      000064      TPVEC= 64      ;; TTY PRINTER VECTOR
831      000240      PIRQVEC=240    ;; PROGRAM INTERRUPT REQUEST VECTOR
832
833      170420      ABASE= 170420
834      000440      AVECT1= 440
835      000200      APRIOR= 200
836
837      167400      $SWR= 167400
838      000001      $TN= 1
839
840      .SBTTL ACT11 HOOKS
841
842      ;;*****
843      ;HOOKS REQUIRED BY ACT11
844      $SVPC=.      ;SAVE PC
845      .=46
846      000046      $ENDAD      ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
847      000052      .=52
848      000052      .WORD 0      ;;2)SET LOC.52 TO ZERO
849      000244      .=$SVPC      ;; RESTORE PC
850      001000      .=1000
851      .SBTTL APT PARAMETER BLOCK
852
853      ;;*****
854      ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
855      ;;*****
856      001000      .$X=.      ;;SAVE CURRENT LOCATION
857      000024      .=24      ;;SET POWER FAIL TO POINT TO START OF PROGRAM
858      000024      200      ;;FOR APT START UP
859      000044      .=44      ;;POINT TO APT INDIRECT ADDRESS PNTR.
860      000044      $APTHDR    ;;POINT TO APT HEADER BLOCK
861      001000      .=$X      ;;RESET LOCATION COUNTER
862      ;;*****
863      ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
864      ;INTERFACE SPEC.
865
866      001000      $APTHD:
867      001000      $HIBTS: .WORD 0      ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
868      001002      $MBADR: .WORD $MAIL  ;;ADDRESS OF APT MAILBOX (BITS 0-15)
869      001004      $STMT: .WORD 2      ;;RUN TIM OF LONGEST TEST
870      001006      $PASTM: .WORD 120.  ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
871      001010      $UNITM: .WORD 120.  ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
872      001012      .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)

```

MAINDEC-11-DVKWA-B MACY11 27(665) 21-FEB-77 14:43 PAGE 21
 DVKWAB.P11 COMMON TAGS

```

873      .SBTTL  COMMON TAGS
874
875      ;*****
876      ;THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
877      ;*USED IN THE PROGRAM.
878
879      001100      .=1100
880      001100      SCMTAG:      ;;START OF COMMON TAGS
881      001100      000000      .WORD      0      ;;CONTAINS THE TEST NUMBER
882      001102      000      .BYTE      0      ;;CONTAINS ERROR FLAG
883      001103      000      .BYTE      0      ;;CONTAINS SUBTEST ITERATION COUNT
884      001104      000000      .WORD      0      ;;CONTAINS SCOPE LOOP ADDRESS
885      001106      000000      .WORD      0      ;;CONTAINS SCOPE RETURN FOR ERRORS
886      001110      000000      .WORD      0      ;;CONTAINS TOTAL ERRORS DETECTED
887      001112      000000      .WORD      0      ;;CONTAINS ITEM CONTROL BYTE
888      001114      000      .BYTE      0      ;;CONTAINS MAX. ERRORS PER TEST
889      001115      001      .BYTE      1      ;;CONTAINS PC OF LAST ERROR INSTRUCTION
890      001116      000000      .WORD      0      ;;CONTAINS ADDRESS OF 'GOOD' DATA
891      001120      000000      .WORD      0      ;;CONTAINS ADDRESS OF 'BAD' DATA
892      001122      000000      .WORD      0      ;;CONTAINS 'GOOD' DATA
893      001124      000000      .WORD      0      ;;CONTAINS 'BAD' DATA
894      001126      000000      .WORD      0      ;;RESERVED--NOT TO BE USED
895      001130      000000      .WORD      0
896      001132      000000      .WORD      0
897      001134      000      .BYTE      0      ;;AUTOMATIC MODE INDICATOR
898      001135      000      .BYTE      0      ;;INTERRUPT MODE INDICATOR
899      001136      000000      .WORD      0
900      001140      177570      .WORD      DSWR      ;;ADDRESS OF SWITCH REGISTER
901      001142      177570      .WORD      DDISP      ;;ADDRESS OF DISPLAY REGISTER
902      001144      177560      .WORD      177560      ;;TTY KBD STATUS
903      001146      177562      .WORD      177562      ;;TTY KBD BUFFER
904      001150      177564      .WORD      177564      ;;TTY PRINTER STATUS REG. ADDRESS
905      001152      177566      .WORD      177566      ;;TTY PRINTER BUFFER REG. ADDRESS
906      001154      000      .BYTE      0      ;;CONTAINS NULL CHARACTER FOR FILLS
907      001155      002      .BYTE      2      ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
908      001156      012      .BYTE      12      ;;INSERT FILL CHARS. AFTER A "LINE FEED"
909      001157      000      .BYTE      0      ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
910      001160      000000      .WORD      0      ;;MAX. NUMBER OF ITERATIONS
911      001162      000000      .WORD      0      ;;ESCAPE ON ERROR ADDRESS
912      001164      177607      .ASCIZ      <207><377><377>      ;;CODE FOR BELL
913      001170      077      .ASCII      /?/      ;;QUESTION MARK
914      001171      015      .ASCII      <15>      ;;CARRIAGE RETURN
915      001172      000012      .ASCIZ      <12>      ;;LINE FEED
916      ;*****
917      .SBTTL  APT MAILBOX-ETABLE
918
919      ;*****
920      .EVEN
921      001174      $MAIL:      ;;APT MAILBOX
922      001174      000000      $MSGTY: .WORD      AMSGTY      ;;MESSAGE TYPE CODE
923      001176      000000      $FATAL: .WORD      AFATAL      ;;FATAL ERROR NUMBER
924      001200      000000      $TESTN: .WORD      ATESTN      ;;TEST NUMBER
925      001202      000000      $PASS:  .WORD      APASS      ;;PASS COUNT
926      001204      000000      $DEVCT: .WORD      ADEVCT      ;;DEVICE COUNT

```

J02

MAINDEC-11-DVKWA-B MACY11 27(665) 21-FEB-77 14:43 PAGE 22
 DVKWAB.P11 APT MAILBOX-ETABLE

927	001206	000000	\$UNIT:	.WORD	AUNIT	;; I/O UNIT NUMBER
928	001210	000000	\$MSGAD:	.WORD	AMSGAD	;; MESSAGE ADDRESS
929	001212	000000	\$MSGLG:	.WORD	AMSGLG	;; MESSAGE LENGTH
930	001214		\$ETABLE:			;; APT ENVIRONMENT TABLE
931	001214	000	\$ENV:	.BYTE	AENV	;; ENVIRONMENT BYTE
932	001215	000	\$ENVM:	.BYTE	AENVM	;; ENVIRONMENT MODE BITS
933	001216	000000	\$SWREG:	.WORD	ASWREG	;; APT SWITCH REGISTER
934	001220	000000	\$USWR:	.WORD	AUSWR	;; USER SWITCHES
935	001222	000000	\$CPUOP:	.WORD	ACPUOP	;; CPU TYPE, OPTIONS
936			.*			BITS 15-11=CPU TYPE
937			.*			11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
938			.*			11/70=06, PDQ=07, Q=10
939			.*			BIT 10=REAL TIME CLOCK
940			.*			BIT 9=FLOATING POINT PROCESSOR
941			.*			BIT 8=MEMORY MANAGEMENT
942	001224	000	\$MAMS1:	.BYTE	AMAMS1	;; HIGH ADDRESS, M.S. BYTE
943	001225	000	\$MTYP1:	.BYTE	AMTYP1	;; MEM. TYPE, BLK#1
944			.*			MEM. TYPE BYTE -- (HIGH BYTE)
945			.*			900 NSEC CORE=001
946			.*			300 NSEC BIPOLAR=002
947			.*			500 NSEC MOS=003
948	001226	000000	\$MADR1:	.WORD	AMADR1	;; HIGH ADDRESS, BLK#1
949			.*			MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
950	001230	000	\$MAMS2:	.BYTE	AMAMS2	;; HIGH ADDRESS, M.S. BYTE
951	001231	000	\$MTYP2:	.BYTE	AMTYP2	;; MEM. TYPE, BLK#2
952	001232	000000	\$MADR2:	.WORD	AMADR2	;; MEM. LAST ADDRESS, BLK#2
953	001234	000	\$MAMS3:	.BYTE	AMAMS3	;; HIGH ADDRESS, M.S. BYTE
954	001235	000	\$MTYP3:	.BYTE	AMTYP3	;; MEM. TYPE, BLK#3
955	001236	000000	\$MADR3:	.WORD	AMADR3	;; MEM. LAST ADDRESS, BLK#3
956	001240	000	\$MAMS4:	.BYTE	AMAMS4	;; HIGH ADDRESS, M.S. BYTE
957	001241	000	\$MTYP4:	.BYTE	AMTYP4	;; MEM. TYPE, BLK#4
958	001242	000000	\$MADR4:	.WORD	AMADR4	;; MEM. LAST ADDRESS, BLK#4
959	001244	000440	\$VECT1:	.WORD	AVECT1	;; INTERRUPT VECTOR#1, BUS PRIORITY#1
960	001246	000000	\$VECT2:	.WORD	AVECT2	;; INTERRUPT VECTOR#2, BUS PRIORITY#2
961	001250	170420	\$BASE:	.WORD	ABASE	;; BASE ADDRESS OF EQUIPMENT UNDER TEST
962	001252	000000	\$DEVN:	.WORD	ADEVN	;; DEVICE MAP
963	001254	000000	\$CDW1:	.WORD	ACDW1	;; CONTROLLER DESCRIPTION WORD#1
964	001256		\$ETEND:			
965			.MEXIT			

K02

MAINDEC-11-DVKWA-B MACY11 27(665) 21-FEB-77 14:43 PAGE 23
 DVKWAB.P11 ERROR POINTER TABLE

```

966      .SBTTL  ERROR POINTER TABLE
967
968      ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
969      ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
970      ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
971      ;*NOTE1:      IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
972      ;*NOTE2:      EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
973
974      ;*      EM      ;;POINTS TO THE ERROR MESSAGE
975      ;*      DH      ;;POINTS TO THE DATA HEADER
976      ;*      DT      ;;POINTS TO THE DATA
977      ;*      DF      ;;POINTS TO THE DATA FORMAT
978
979
980      001256      $ERRTB:
981
982
983      ;ITEM      1
984
985      001256      017134      EM1      ;CLOCK      SR FUNCTION ERROR
986      001260      017457      DH1      ;ERRPC      ASR      WAS      S/B
987      001262      017666      DT1      ;$ERRPC,ASR,$BDDAT,$GDDAT
988      001264      017764      DF0      ;ALL NUMBERS ARE IN OCTAL FORM
989
990
991      ;ITEM      2
992
993      001266      017166      EM2      ;CLOCK      SR DATA ERROR
994      001270      017457      DH1      ;ERRPC      ASR      WAS      S/B
995      001272      017666      DT1      ;$ERRPC,ASR,$BDDAT,$GDDAT
996      001274      017764      DF0      ;ALL NUMBERS ARE IN OCTAL FORM
997
998
999      ;ITEM      3
1000
1001      001276      017214      EM3      ;CLOCK      BR DATA ERROR
1002      001300      017503      DH3      ;ERRPC      ABR      WAS
1003      001302      017700      DT3      ;$ERRPC,ABR,$BDDAT,$GDDAT
1004      001304      017764      DF0      ;ALL NUMBERS ARE IN OCTAL FORM
1005
1006
1007      ;ITEM      4
1008
1009      001306      017242      EM4      ;INTERRUPT ERROR.
1010      001310      017527      DH4A     ;ERRPC      TO      ROM ADDR.
1011      001312      017712      DT4      ;$ERRPC,      TRTO,TRFRO
1012      001314      017764      DF0      ;ALL NUMBERS ARE IN OCTAL FORM
1013
1014
1015      ;ITEM      5
1016
1017      001316      017263      EM5      ;CLOCK      COUNT REG ERROR
1018      001320      017457      DH1      ;ERRPC      ASR      WAS      S/B
1019      001322      017666      DT1      ;$ERRPC, ACR, $BDDAT, $GDDAT

```

1020	001324	017764	DF0		;ALL NUMBERS ARE IN OCTAL FORM
1021					
1022					
1023				;ITEM 6	
1024					
1025	001326	017325	EM12		;CLOCK COUNT FUNCTION ERROR
1026	001330	017563	DH12		;ERRPC ASR
1027	001332	017722	DT12		;ERRPC, ASR
1028	001334	017764	DF0		;ALL NUMBERS ARE IN OCTAL FORM
1029					
1030					
1031				;ITEM 7	
1032					
1033	001336	017354	EM16		;CLOCK INTERRUPT ERROR
1034	001340	017563	DH12		;ERRPC ASR
1035	001342	017722	DT12		;ERRPC, ASR
1036	001344	017764	DF0		;ALL NUMBERS ARE IN OCTAL FORM
1037					
1038					
1039				;ITEM 10	
1040					
1041	001346	017405	EM20		;CLOCK REPEATABILITY ERROR
1042	001350	017600	DH20		;ERRPC ASR 2ND CNT 1ST CNT 3RD CNT
1043	001352	017730	DT20		;ERRPC, ASR, SBDDAT, SGDDAT, STMPO
1044	001354	017764	DF0		;ALL NUMBERS ARE IN OCTAL FORM
1045					
1046					
1047				;ITEM 11	
1048					
1049	001356	017306	EM11		;CLOCK COUNT ERROR
1050	001360	017457	DH1		;ERRPC ASR WAS S/B
1051	001362	017744	DT22		;ERRPC, ASR, SBDDAT, STMPO
1052	001364	017764	DF0		;ALL NUMBERS ARE IN OCTAL FORM
1053					
1054					
1055				;ITEM 12	
1056					
1057	001366	017434	EM26		;CLOCK ADDRESSING ERROR
1058	001370	017641	DH26		;ERRPC CLOCK ADDR.
1059	001372	017756	DT26		;ERRPC, STMPO
1060	001374	017764	DF0		;ALL NUMBERS ARE IN OCTAL FORM
1061					
1062					
1063	001376	170420	ASR:	.WORD	ABASE
1064	001400	170422	ABR:	.WORD	ABASE+2
1065	001402	000440	VECT1:	.WORD	AVECT1
1066	001404	000442	VECTP:	.WORD	AVECT1+2
1067	001406	000444	VECT2:	.WORD	AVECT1+4
1068	001410	000446	VECT2P:	.WORD	AVECT1+6
1069	001412	000200	PRIOR:	.WORD	APRIOR
1070	001414	167774	DR:	.WORD	167774
1071	001416	167772	DR2:	.WORD	167772
1072	001420	000000	STMPO:	.WORD	0
1073	001422	000000	STMP1:	.WORD	0

; VECTOR ADDR. OF ST2 INTR.

; TEMP STORAGE.

; TMP STORAGE.

1074	001424	000000			\$TMP3: .WORD 0	
1075	001426	000000			ROTATE: .WORD 0	;POINT TO DEVICE UNDER TEST.
1076	001430	000000			UTEST: .WORD 0	;KEEPS TRACK OF GOOD UNITS.
1077	001432	000000			ERCNT: .WORD 0	;COUNTS ERRORS.
1078	001434	000000			MDEVCT: .WORD 0	;COUNTS DEVICES TESTED.
1079	001436	000000			TSTCNT: .WORD 0	;MAX DEVICES TO BE TESTED.
1080	001440	000000			EXS: .WORD 0	;=0, NORMAL; =1 SPECIAL TESTOR START, BY L+S 2
1081	001442	000000			LCNT: .WORD 0	;TOTAL UNITS TESTED.
1082						
1083						
1084	001444	005237	001440		TSTSTR: INC EXS	;SET FOR TESTOR.
1085	001450	012737	000020	001436	MOV #16., TSTCNT	;ALLOW 16 UNITS
1086	001456	000413			BR 15	
1087		001460			WSTART=.	
1088	001460	012737	000020	001436	MOV #16., TSTCNT	;TEST UP TO 16 UNITS.
1089	001466	005037	001440		CLR EXS	
1090	001472	000405			BR 15	
1091		001474			START=.	
1092	001474	012737	000004	001436	MOV #4, TSTCNT	;TEST UP TO FOUR UNITS.
1093	001502	005037	001440		CLR EXS	
1094	001506				15:	
1095					.SBTTL INITIALIZE THE COMMON TAGS	
1096					;;CLEAR THE COMMON TAGS (\$CMTAG) AREA	
1097	001506	012706	001100		MOV #SCMTAG, R6	;;FIRST LOCATION TO BE CLEARED
1098	001512	005026			CLR (R6)+	;;CLEAR MEMORY LOCATION
1099	001514	022706	001140		CMP #SWR, R6 ;;DONE?	
1100	001520	001374			BNE -6	;;LOOP BACK IF NO
1101	001522	012706	001100		MOV #STACK, SP	;;SETUP THE STACK POINTER
1102					;;INITIALIZE A FEW VECTORS	
1103	001526	012737	015040	000020	MOV #SCOPE, @IOTVEC	;;IOT VECTOR FOR SCOPE ROUTINE
1104	001534	012737	000340	000022	MOV #340, @IOTVEC+2	;;LEVEL 7
1105	001542	012737	014476	000030	MOV #ERROR, @EMTVEC	;;EMT VECTOR FOR ERROR ROUTINE
1106	001550	012737	000340	000032	MOV #340, @EMTVEC+2	;;LEVEL 7
1107	001556	012737	017054	000034	MOV #STRAP, @TRAPVEC	;;TRAP VECTOR FOR TRAP CALLS
1108	001564	012737	000340	000036	MOV #340, @TRAPVEC+2	;;LEVEL 7
1109	001572	012737	016626	000024	MOV #SPWRDN, @PWRVEC	;;POWER FAILURE VECTOR
1110	001600	012737	000340	000026	MOV #340, @PWRVEC+2	;;LEVEL 7
1111	001606	005037	001160		CLR \$TIMES	;;INITIALIZE NUMBER OF ITERATIONS
1112	001612	005037	001162		CLR \$ESCAPE	;;CLEAR THE ESCAPE ON ERROR ADDRESS
1113	001616	112737	000001	001115	MOV #1, \$ERMAX	;;ALLOW ONE ERROR PER TEST
1114	001624	012737	001624	001106	MOV #., \$LPADR	;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1115	001632	012737	001632	001110	MOV #., \$LPERR	;;SETUP THE ERROR LOOP ADDRESS
1116					;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS	
1117					;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.	
1118	001640	013746	000004		MOV @ERRVEC, -(SP)	;;SAVE ERROR VECTOR
1119	001644	012737	001700	000004	MOV #64\$, @ERRVEC	;;SET UP ERROR VECTOR
1120	001652	012737	177570	001140	MOV #DSWR, SWR	;;SETUP FOR A HARDWARE SWICH REGISTER
1121	001660	012737	177570	001142	MOV #DDISP, DISPLAY	;;AND A HARDWARE DISPLAY REGISTER
1122	001666	022777	177777	177244	CMP #-1, @SWR	;;TRY TO REFERENCE HARDWARE SWR
1123	001674	001012			BNE 66\$	;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1124						;;AND THE HARDWARE SWR IS NOT = -1
1125	001676	000403			BR 65\$	;;BRANCH IF NO TIMEOUT
1126	001700	012716	001706		64\$: MOV #65\$, (SP)	;;SET UP FOR TRAP RETURN
1127	001704	000002			RTI	

N02

MAINDEC-11-DVKWA-B MACY11 27(665) 21-FEB-77 14:43 PAGE 26
 DVKWAB.P11 INITIALIZE THE COMMON TAGS

```

1128 001706 012737 000176 001140 65$: MOV #SWREG,SWR ;;POINT TO SOFTWARE SWR
1129 001714 012737 000174 001142 MOV #DISPREG,DISPLAY
1130 001722 012637 000004 66$: MOV (SP)+,2#ERRVEC ;;RESTORE ERROR VECTOR
1131
1132 001726 005037 001202 CLR $PASS ;;CLEAR PASS COUNT
1133 001732 132737 000200 001215 BITB #APTSIZE,$ENVM ;;TEST USER SIZE UNDER APT
1134 001740 001403 BEQ 67$ ;;YES,USE NON-APT SWITCH
1135 001742 012737 001216 001140 MOV #SSWREG,SWR ;;NO,USE APT SWITCH REGISTER
1136 001750 67$:
1137
1138
1139 001750 012746 000340 MOV #340,-(SP) ;SET CPU PRIORITY ON RETERN.
1140 001754 012746 001762 MOV #68$,-(SP) ;SHOW RETURN ADDRESS.
1141 001760 000002 RTI ;CAUSE A RETURN(PUTS STATUS IN STATUS REG.).
1142 001762 68$:
1143
1144 001762 005037 001204 CLR $DEVCT ;ZERO DEVICE COUNT.
1145 001766 012737 017004 000004 MOV #IOTRD,2#ERRVEC ;FIX TRAP CATCHER.
1146 001774 013737 001244 001402 MOV $VECT1,VECT1 ;NOW FIX VECTOR ADDR.
1147 002002 013737 001250 001376 MOV $BASE,ASR ;FIX ADDRESS OF CSR.
1148
1149 .SBTTL TYPE PROGRAM NAME
1150 ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
1151 002010 005227 177777 INC #-1 ;;FIRST TIME?
1152 002014 001033 BNE 69$ ;;BRANCH IF NO
1153 002016 104401 002064 TYPE 70$ ;;TYPE ASCIZ STRING
1154
1155 .SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
1156 002022 005737 000042 TST 2#42 ;;ARE WE RUNNING UNDER XXDP/ACT?
1157 002026 001012 BNE 71$ ;;BRANCH IF YES
1158 002030 123727 001214 000001 CMPB $ENV,#1 ;;ARE WE RUNNING UNDER APT?
1159 002036 001406 BEQ 71$ ;;BRANCH IF YES
1160 002040 023727 001140 000176 CMP SWR,#SWREG ;;SOFTWARE SWITCH REG SELECTED?
1161 002046 001005 BNE 72$ ;;BRANCH IF NO
1162 002050 104406 GTSWR ;;GET SOFT-SWR SETTINGS
1163 002052 000403 BR 72$
1164 002054 112737 000001 001134 71$: MOVB #1,$AUTOB ;;SET AUTO-MODE INDICATOR
1165 002062 000410 72$: BR 69$ ;;GET OVER THE ASCIZ
1166
1167 002104 ;;70$: .ASCIZ <CRLF>#MD11-DVKWA-B#<CRLF>
1168 002104 69$:
1169 002104 005737 001440 RSTART: TST EXS ;TESTOR MODE ENABLED??
1170 002110 001441 BEQ 1$ ;NO DON'T TYPE NEXT MESSAGE.
1171 002112 104401 002120 TYPE 65$ ;;TYPE ASCIZ STRING
1172 002116 000436 BR 64$ ;;GET OVER THE ASCIZ
1173
1174 002214 ;;65$: .ASCIZ <15><12>#TESTOR MODE ENABLED--SEE DOCUMENTATION FOR INSTRUCTIONS.#
1175 002214 64$:
1176 002214 104401 002222 1$: TYPE 67$ ;;TYPE ASCIZ STRING
1177 002220 000411 BR 66$ ;;GET OVER THE ASCIZ
1178
1179 002244 ;;67$: .ASCIZ <15><12>#TEST RUNNING...#
1180 002244 005037 001434 66$: CLR MDEVCT ;TESTING FIRST UNIT.
1181 002250 005037 001432 CLR ERCNT ;NO ERRORS.

```


MAINDEC-11-DVKWA-B
DVKWAB.P11 T3

MACY11 27(665) 21-FEB-77 14:43 PAGE 30
*TEST THAT CLOCK A STATUS REGISTER BIT 14 CAN BE SET AND CLEARED

```

1320 ;/
1321 ;*****
1322 ;*TEST 4 *TEST THAT CLOCK A STATUS REGISTER BIT 13 CAN BE SET AND CLEARED
1323 ;*
1324 ;*CLOCK STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
1325 ;*F/FS OR GATES
1326 ;*
1327 ;*****
1328 ;*****
1329 002614 000004
1330 002616 012737 000100 001160
1331
1332 002624 005077 176546 CLR QASR ;/CLEAR THE STATUS REGISTER.
1333 002630 052777 020000 176540 BIS #BIT13,QASR ;/SET BIT 13.
1334 002636 012737 020000 001124 MOV #BIT13,$GDDAT ;/SET FOR ERROR TYPEOUT S/B.
1335 002644 017737 176526 001126 MOV QASR,$BDDAT ;/READ THE STATUS REGISTER.
1336 002652 023737 001124 001126 CMP $GDDAT,$BDDAT ;/DID BIT 13 AND ONLY BIT 13 SET?
1337 002660 001402 BEQ 1$ ;/IF SO-LETS TRY CLEARING IT.
1338
1339 ;*****
1340 ;*****
1341 002662 104002
1342
1343
1344 ;*****
1345 ;*****
1346 002664 000412 BR 2$ ;/BR TO END SUBTEST.
1347
1348 002666 042777 020000 176502 1$: BIC #BIT13,QASR ;/TRY CLEARING BIT 13.
1349 002674 005037 001124 CLR $GDDAT ;/CLEAR S/B FOR TYPEOUT IF ANY.
1350 002700 017737 176472 001126 MOV QASR,$BDDAT ;/NOW READ IT BACK.
1351 002706 001401 BEQ 2$ ;/IF ZERO - NO ERROR!
1352
1353 ;*****
1354 ;*****
1355 002710 104002
1356
1357
1358
1359 ;*****
1360 ;*****
1361 002712
1362
1363
1364

```

F03

MAINDEC-11-DVKWA-B
DVKWAB.P11 T4

MACY11 27(665) 21-FEB-77 14:43 PAGE 31
*TEST THAT CLOCK A STATUS REGISTER BIT 11 CAN BE SET AND CLEARED

```

1365 ;/8
1366 ;*****
1367 ;*TEST 5 *TEST THAT CLOCK A STATUS REGISTER BIT 11 CAN BE SET AND CLEARED
1368 ;*
1369 ;*CLOCK STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
1370 ;*F/FS OR GATES
1371 ;*
1372
1373 ;*****
1374 002712 000004 1ST5: SCOPE
1375 002714 012737 000100 001160 MOV #100,$TIMES ;;DO 100 ITERATIONS
1376
1377 002722 005077 176450 CLR JASR ;/CLEAR THE STATUS REGISTER.
1378 002726 052777 004000 176442 BIS #BIT11,JASR ;/SET BIT 11.
1379 002734 012737 004000 001124 MOV #BIT11,$GDDAT ;/SET FOR ERROR TYPEOUT S/B.
1380 002742 017737 176430 001126 MOV JASR,$BDDAT ;/READ THE STATUS REGISTER.
1381 002750 023737 001124 001126 CMP $GDDAT,$BDDAT ;/DID BIT 11 AND ONLY BIT 11 SET?
1382 002756 001402 BEQ 15 ;/IF SO-LETS TRY CLEARING IT.
1383
;;*****
1386 002760 104002 ERROR 2 ;/ERROR CLOCK AS STATUS REGISTER
1387 ;/BIT 11 FAILED TO BIT SET.
1388
;;*****
1391 002762 000412 BR 25 ;/BR TO END SUBTEST.
1392
1393 002764 042777 004000 176404 15: BIC #BIT11,JASR ;/TRY CLEARING BIT 11.
1394 002772 005037 001124 CLR $GDDAT ;/CLEAR S/B FOR TYPEOUT IF ANY.
1395 002776 017737 176374 001126 MOV JASR,$BDDAT ;/NOW READ IT BACK.
1396 003004 001401 BEQ 25 ;/IF ZERO - NO ERROR!
1397
1398
;;*****
1401 003006 104002 ERROR 2 ;/ERROR - CLOCK A STATUS REGISTER.
1402 ;/BIT 11 FAILED TO CLEAR.
1403
1404
;;*****
1407 003010 25:
1408
1409

```

MACY11 27(665) 21-FEB-77 14:43 PAGE 32
*TEST THAT CLOCK A STATUS REGISTER BIT 11 CAN BE SET AND CLEARED

[illegible]

MAINDEC-11-DVKWA-B
DVKWAB.P11 T6

MACY11 27(665) 21-FEB-77 14:43 PAGE 33
*TEST THAT CLOCK A STATUS REGISTER BIT 6 CAN BE SET AND CLEARED

```

1455      ;/8  

1456      ;;*****  

1457      *TEST 7          *TEST THAT CLOCK A STATUS REGISTER BIT 5 CAN BE SET AND CLEARED  

1458      *  

1459      *CLOCK STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL  

1460      *F/FS OR GATES  

1461      *  

1462      ;  

1463      ;;*****  

1464      003106 000004  

1465      003110 012737 000100 001160   TST7: SCOPE  

1466                                     MOV     #100,$TIMES           ;;DO 100 ITERATIONS  

1467      003116 005077 176254           CLR     @ASR                ;/CLEAR THE STATUS REGISTER.  

1468      003122 052777 000040 176246   BIS     #BITS,@ASR         ;/SET BIT 5.  

1469      003130 012737 000040 001124   MOV     #BITS,$GDDAT       ;/SET FOR ERROR TYPEOUT S/B.  

1470      003136 017737 176234 001126   MOV     @ASR,$BDDAT        ;/READ THE STATUS REGISTER.  

1471      003144 023737 001124 001126   CMP     $GDDAT,$BDDAT      ;/DID BIT 5 AND ONLY BIT 5 SET?  

1472      003152 001402                   BEQ     1$               ;/IF SO-LETS TRY CLEARING IT.  

1473  

1476      003154 104002  

1477      ERROR    2                      ;/ERROR CLOCK AS STATUS REGISTER  

1478                                      ;/BIT 5 FAILED TO BIT SET.  

1479  

1481      003156 000412  

1482      BR        2$                    ;/BR TO END SUBTEST.  

1483      003160 042777 000040 176210 1$: BIC     #BITS,@ASR         ;/TRY CLEARING BIT 5.  

1484      003166 005037 001124           CLR     $GDDAT             ;/CLEAR S/B FOR TYPEOUT IF ANY.  

1485      003172 017737 176200 001126   MOV     @ASR,$BDDAT        ;/NOW READ IT BACK.  

1486      003200 001401                   BEQ     2$               ;/IF ZERO - NO ERROR!  

1487  

1488      ;;*****  

1491      003202 104002  

1492      ERROR    2                      ;/ERROR - CLOCK A STATUS REGISTER.  

1493                                      ;/BIT 5 FAILED TO CLEAR.  

1494  

1497      003204  

1498      2$:  

1499      ;;*****

```

```

1500 ;/8
1501 ;*****
1502 ;*TEST 10 *TEST THAT CLOCK A STATUS REGISTER BIT 4 CAN BE SET AND CLEARED
1503 ;*
1504 ;*CLOCK STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
1505 ;*F/FS OR GATES
1506 ;*
1507
1508 ;*****
1509 003204 000004
1510 003206 012737 000100 001160
1511
1512 003214 005077 176156
1513 003220 052777 000020 176150
1514 003226 012737 000020 001124
1515 003234 017737 176136 001126
1516 003242 023737 001124 001126
1517 003250 001402
1518
1519
1520
1521 003252 104002
1522
1523
1524
1525
1526 003254 000412
1527
1528 003256 042777 000020 176112
1529 003264 005037 001124
1530 003270 017737 176102 001126
1531 003276 001401
1532
1533
1534
1535
1536 003300 104002
1537
1538
1539
1540
1541
1542 003302
1543
1544

```

MAINDEC-11-DVKWA-B
DVKWAB.P11 T10

MACY11 27(665) 21-FEB-77 14:43 PAGE 35
*TEST THAT CLOCK A STATUS REGISTER BIT 4 CAN BE SET AND CLEARED

```

1545      ;/
1546      ;*****
1547      ;*TEST 11      *TEST THAT CLOCK A STATUS REGISTER BIT 3 CAN BE SET AND CLEARED
1548      ;*
1549      ;*CLOCK STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
1550      ;*F/FS OR GATES
1551      ;*
1552      ;*****
1553      ;*****
1554      003302 000004      1ST11: SCOPE
1555      003304 012737 000100 001160      MOV      #100,$TIMES      ;;DO 100 ITERATIONS
1556
1557      003312 005077 176060      CLR      @ASR      ;/CLEAR THE STATUS REGISTER.
1558      003316 052777 000010 176052      BIS      #BIT3,@ASR      ;/SET BIT 3.
1559      003324 012737 000010 001124      MOV      #BIT3,$GDDAT      ;/SET FOR ERROR TYPEOUT S/B.
1560      003332 017737 176040 001126      MOV      @ASR,$BDDAT      ;/READ THE STATUS REGISTER.
1561      003340 023737 001124 001126      CMP      $GDDAT,$BDDAT      ;/DID BIT 3 AND ONLY BIT 3 SET?
1562      003346 001402      BEQ      1$      ;/IF SO-LETS TRY CLEARING IT.
1563
      ;;*****>>> ERROR <<<*****
1566      003350 104002      ERROR      2      ;/ERROR CLOCK AS STATUS REGISTER
1567      ;/BIT 3 FAILED TO BIT SET.
1568
      ;;*****>>> ERROR <<<*****
1571      003352 000412      BR      2$      ;/BR TO END SUBTEST.
1572
1573      003354 042777 000010 176014 1$: BIC      #BIT3,@ASR      ;/TRY CLEARING BIT 3.
1574      003362 005037 001124      CLR      $GDDAT      ;/CLEAR S/B FOR TYPEOUT IF ANY.
1575      003366 017737 176004 001126      MOV      @ASR,$BDDAT      ;/NOW READ IT BACK.
1576      003374 001401      BEQ      2$      ;/IF ZERO - NO ERROR!
1577
      ;;*****>>> ERROR <<<*****
1581      003376 104002      ERROR      2      ;/ERROR - CLOCK A STATUS REGISTER.
1582      ;/BIT 3 FAILED TO CLEAR.
1583
      ;;*****>>> ERROR <<<*****
1587      003400      2$:
1588
1589

```

K03

MAINDEC-11-DVKWA-B
DVKWAB.P11 T11

MACY11 27(665) 21-FEB-77 14:43 PAGE 36
*TEST THAT CLOCK A STATUS REGISTER BIT 3 CAN BE SET AND CLEARED

```

1590
1591
1592
1593
1594
1595
1596
1597
1598
1599 003400 000004
1600 003402 012737 000100 001160
1601
1602 003410 005077 175762
1603 003414 052777 000004 175754
1604 003422 012737 000004 001124
1605 003430 017737 175742 001126
1606 003436 023737 001124 001126
1607 003444 001402
1608
1611 003446 104002
1612
1613
1616 003450 000412
1617
1618 003452 042777 000004 175716 15:
1619 003460 005037 001124
1620 003464 017737 175706 001126
1621 003472 001401
1622
1623
1626 003474 104002
1627
1628
1629
1632 003476
1633
1634

; /#
; *****
; *TEST 12 *TEST THAT CLOCK A STATUS REGISTER BIT 2 CAN BE SET AND CLEARED
; *
; *CLOCK STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
; *F/FS OR GATES
; *
; *****
; *ST12: SCOPE
; MOV #100,$TIMES ;DO 100 ITERATIONS
; CLR @ASR ;CLEAR THE STATUS REGISTER.
; BIS #BIT2,@ASR ;SET BIT 2.
; MOV #BIT2,$GDDAT ;SET FOR ERROR TYPEOUT S/B.
; MOV @ASR,$BDDAT ;READ THE STATUS REGISTER.
; CMP $GDDAT,$BDDAT ;DID BIT 2 AND ONLY BIT 2 SET?
; BEQ 15 ;IF SO-LETS TRY CLEARING IT.
; ;*****
; ;***** ERROR <<<*****
; ERROR 2 ;/ERROR CLOCK AS STATUS REGISTER
; ;/BIT 2 FAILED TO BIT SET.
; ;*****
; ;***** ERROR <<<*****
; BR 25 ;/BR TO END SUBTEST.
; BIC #BIT2,@ASR ;/TRY CLEARING BIT 2.
; CLR $GDDAT ;/CLEAR S/B FOR TYPEOUT IF ANY.
; MOV @ASR,$BDDAT ;/NOW READ IT BACK.
; BEQ 25 ;/IF ZERO - NO ERROR!
; ;*****
; ;***** ERROR <<<*****
; ERROR 2 ;/ERROR - CLOCK A STATUS REGISTER.
; ;/BIT 2 FAILED TO CLEAR.
; ;*****
; ;***** ERROR <<<*****
; 25:

```

MAINDEC-11-DVKWA-B
DVKWAB.P11 T12

MACY11 27(665) 21-FEB-77 14:43 PAGE 37
*TEST THAT CLOCK A STATUS REGISTER BIT 2 CAN BE SET AND CLEARED

```

1635      ;/8
1636      ;*****
1637      ;*TEST 13      *TEST THAT CLOCK A STATUS REGISTER BIT 1 CAN BE SET AND CLEARED
1638      ;*
1639      ;*CLOCK STATUS REGISTER BIT EXERCISE. ON FAILURE-SUSPECT INDIVIDUAL
1640      ;*F/FS OR GATES
1641      ;*
1642
1643      ;*****
1644      003476 000004      ST13: SCOPE
1645      003500 012737 000100 001160      MOV      #100,$TIMES      ;;DO 100 ITERATIONS
1646
1647      003506 005077 175664      CLR      @ASR      ;/CLEAR THE STATUS REGISTER.
1648      003512 052777 000002 175656      BIS      #BIT1,@ASR      ;/SET BIT 1.
1649      003520 012737 000002 001124      MOV      #BIT1,$GDDAT      ;/SET FOR ERROR TYPEOUT S/B.
1650      003526 017737 175644 001126      MOV      @ASR,$BDDAT      ;/READ THE STATUS REGISTER.
1651      003534 023737 001124 001126      CMP      $GDDAT,$BDDAT      ;/DID BIT 1 AND ONLY BIT 1 SET?
1652      003542 001402      BEQ      1$      ;/IF SO-LETS TRY CLEARING IT.
1653
      ;;*****
1656      003544 104002      ERROR      2      ;/ERROR CLOCK AS STATUS REGISTER
1657      ;/BIT 1 FAILED TO BIT SET.
1658
      ;;*****
1661      003546 000412      BR      2$      ;/BR TO END SUBTEST.
1662
1663      003550 042777 000002 175620 1$: BIC      #BIT1,@ASR      ;/TRY CLEARING BIT 1.
1664      003556 005037 001124      CLR      $GDDAT      ;/CLEAR S/B FOR TYPEOUT IF ANY.
1665      003562 017737 175610 001126      MOV      @ASR,$BDDAT      ;/NOW READ IT BACK.
1666      003570 001401      BEQ      2$      ;/IF ZERO - NO ERROR!
1667
      ;;*****
1671      003572 104002      ERROR      2      ;/ERROR - CLOCK A STATUS REGISTER.
1672      ;/BIT 1 FAILED TO CLEAR.
1673
      ;;*****
1677      003574      2$:
1678
1679

```


MAINDEC-11-DVKWA-B
DVKWAB.P11 T17

MACY11 27(665) 21-FEB-77 14:43 PAGE 41
*TEST THE LOW BYTE OPERATION OF CLOCK'S STATUS REGISTER

1842
1843
1844
1845

;CLOCK'S STATUS WHEN
;DOING A DATOB TO THE LOW BYTE.

;;SSSSSSSSSSSSSSSSSSSSSSSSSSSS>>> ERROR <<<SS

1848 004124

15:

1849
1850
1851
1852
1853
1854
1855
1856
1857

;;*****
;*TEST 20 *TEST THE HIGH BYTE OPERATION OF A'S STATUS REGISTER

;*
;*WE CAN SUCCESSFULLY WRITE EVERY BIT IN STATUS REG A
;*NOW LETS CHECK THE BYTE OPERATION OF THIS REGISTER.
;*

1858

;;*****

1859 004124 000004
1860 004126 012737 000050 001160

†ST20: SCOPE
MOV #50,\$TIMES ;;DO 50 ITERATIONS

1861

1862 004134 005077 175236

CLR @ASR ;CLEAR THE STATUS REGISTER.

1863

1864 004140 005237 001376

INC ASR ;ADD #1 TO THE STATUS REGISTER'S ADDRESS
;SO THAT WE WILL BE WRITING INTO
;THE HIGH BYTE.

1865

1866

1867

1868 004144 112777 177313 175224

MOVB #177313,@ASR ;TRY WRITING ALL BITS IN THE STATUS
;REGISTER. LOGIC SHOULD PREVENT THE LOW
;BYTE OF THE STATUS REGISTER FROM
;BEING WRITTEN INTO BECAUSE WE ARE USING
;A DATOB INSTRUCTION WITH ADD SET.

1869

1870

1871

1872

1873

1874 004152 005337 001376

DEC ASR ;FIX ADDRESS OF THE STATUS REGISTER ADDR.
;SO WE CAN LOOK AT THE WHOLE WORD.

1875

1876 004156 017737 175214 001126

MOV @ASR,\$BDDAT ;READ BACK WHAT THE STATUS REG. CONTAINS
MOV \$BDDAT,\$GDDAT ;FIX \$GDDAT FOR ERROR TYPEOUT IF AN ERROR
CLR \$GDDAT ;OCCURRED, LOWER BYTE CLEARED.
TSTB \$BDDAT ;IS LOWER BYTE CLEAR?
BEQ 15 ;BR IF YES TO NEXT SUBTEST.

1877 004164 013737 001126 001124

1878 004172 105037 001124

1879 004176 105737 001126

1880 004202 001401

1881

1882

;;SSSSSSSSSSSSSSSSSSSSSSSSSSSS>>> ERROR <<<SS

1885 004204 104001

ERROR 1 ;ERROR - WROTE INTO LOWER BYTE
;OF CLOCKS STATUS REGISTER WHEN
;DOING A DATOB TO THE HIGH BYTE.

1886

1887

1888

;;SSSSSSSSSSSSSSSSSSSSSSSSSSSS>>> ERROR <<<SS

1891 004206

15:

1892

1893

1894

1895

;;*****
;*TEST 21 *TEST CLOCK'S COUNT REGISTER WITH 125252 PATTERN
;*****

1950								
1951	004364	012737	052525	001124	MOV	#052525,\$GDDAT	;SET EXPECTED TO PATTERN IN CASE OF	
1952							;NEED OF ERROR TYPEOUT.	
1953	004372	01746	175000		MOV	@ASR,-(6)	; /SAVE CSR	
1954	004376	011537	001424		MOV	(6),\$TMP3	; /GET CSR.	
1955	004402	04737	177707	001424	BIC	#177707,\$TMP3	; /SAVE RATE BITS.	
1956	004410	02737	004005	001424	BIS	#BIT11!BIT2!BIT0,\$TMP3	; /SET MODE 2, NO RATE,DISABLE INTERNAL OSC	
1957	004416	013777	001424	174752	MOV	\$TMP3,@ASR	; /LOAD CSR.	
1958							; /THIS MUST BE DONE IN	
1959							; /ORDER TO XFERR COUNTER	
1960							; /TO BUFFER ON ST2.	
1961	004424	052777	001000	174744	BIS	#BIT9,@ASR	; /GENERATE ON ST2 PULSE	
1962	004432	017737	174742	001126	MOV	@ABR,\$BDDAT	; /READ THE PRESET BUFFER,	
1963							; /PREVIOUS COUNTER	
1964	004440	012677	174732		MOV	(6)+@ASR	; /CONTENTS ARE IN \$BDDAT.	
1965	004444	005737	001126		TST	\$BDDAT	; /RESTORE CSR	
1966								
1967	004450	023737	001124	001126	CMP	\$GDDAT,\$BDDAT	;DID ALL THE BITS AND NO OTHER BITS	
1968							;COME THROUGH?	
1969	004456	001401			BEQ	1\$	;BR IF YES TO NEXT TEST.	
1970								
1971								

G04

MAINDEC-11-DVKWA-B
DVKWAB.P11 T24MACY11 27(665) 21-FEB-77 14:43 PAGE 45
*TEST THAT INIT CLEARS BUFFER REGISTER

```

2058 004606 1S:
2059 004606 005737 001440 TST EXS ;TEST EXTERNAL SIGS?
2060 004612 001403 BEQ TST25 ;
2061 004614 052777 016000 174574 BIS #BIT11!BIT12!BIT10,ADR2 ;ENABLE THEM
2062
2063 ;*****
2064 ;*TEST 25 *TEST THE SETTING OF MAINTENANCE ST2 IN CLOCK BIT 15 TO SET
2065 ;*****
2066 004622 000004 TST25: SCOPE
2067
2068 004624 012777 000001 174544 MOV #1,ADR ;SET THE GO BIT(ENABLES ST2 TO SET FLG).
2069 004632 052777 001000 174536 BIS #BIT9,ADR ;SET MAINTENANCE ST2.
2070
2071 004640 005777 174532 TST ADR ;DID BIT15 (ST2 FLAG) SET?
2072 004644 100402 BMI 1S ;BR IF YES - NEXT TEST
2073
2074 ;;*****
2077 004646 104001 ERROR 1 ;ERROR - MAINTENANCE ST2 (BIT9)
2078 ;DID NOT SET BIT15 (ST2 FLAG).
2079
2080 ;;*****
2083 004650 000410 1S: BR TST26 ;
2084 004652 005737 TST EXS ;TEST EXTERNAL SIGNALS?
2085 004656 001405 BEQ TST26 ;
2086 004660 032777 000004 174526 BIT #BIT2,ADR ;DID EXTERNAL ST2 GET SET?
2087 004666 001001 BNE TST26 ;
2088 ;;*****
2091 004670 104006 ERROR 6 ;ST2 OUT NOT DETECTED
2092 ;BY TESTOR
2093 ;;*****
2096
2097 ;*****
2098 ;*TEST 26 *TEST THAT BIT00 IN CLOCK STATUS REG. WILL SET WHEN BIT13 AND MAIN. ST2
2099 ;*****
2100 004672 000004 TST26: SCOPE
2101
2102 004674 012777 020000 174474 MOV #BIT13,ADR ;SET "ST2 ENB COUNTER" IN CLK STATUS REG.
2103 004702 052777 001000 174466 BIS #BIT9,ADR ;GENERATE A MAINTENANCE ST2.
2104 004710 032777 000001 174460 BIT #BIT00,ADR ;DID BIT00 (GO) SET?
2105 004716 001001 BNE 1S ;BR IF YES - NEXT TEST.
2106
2107 ;;*****
2108 ;*****
2111 004720 104001 ERROR 1 ;ERROR - BIT00 OF CLOCK'S STATUS REGISTER

```

H04

MAINDEC-11-DVKWA-B
DVKWAB.F11 T26MACY11 27(665) 21-FEB-77 14:43 PAGE 46
*TEST THAT BIT00 IN CLOCK STATUS REG. WILL SET WHEN BIT13 AND MAIN. ST2; FAILED TO SET WHEN BIT13 WAS SET
; AND A MAINTENANCE ST2 GENERATED.2112
2113
2114
2115

;; \$ ERROR <<< \$

2118 004722 005077 174450

15: CLR QASR ; LEAVE SUBTEST WITH CLOCK CLEAR.

2119
2120
2121
2122
2123.SBTTL *
.SBTTL *PHASE 3 COUNT TESTS
.SBTTL *2124
2125
2126

;; \$

*TEST 27 *TEST TO SEE IF THE COUNTER WILL INCREMENT

;; \$

2127 004726 000004

*ST27: SCOPE

2128
2129
2130
2131
2132004730 005077 174442
004734 005077 174440
004740 052777 000061 174430
004746 052777 000400 174422CLR QASR ; CLEAR THE CSR.
CLR QABR ; CLEAR THE BUFFER
BIS #BITS!BIT4!BIT0, QASR ; SET RATE:ST1, GO.
BIS #BIT8, QASR ; GENERATE A MAINTENANCE ST1.
; DID THE CLOCK COUNT?
MOV QASR, -(6) ; /SAVE CSR
MOV (6), STMP3 ; /GET CSR.
BIC #177707, STMP3 ; /SAVE RATE BITS.
BIS #BIT11!BIT2!BIT0, STMP3 ; /SET MODE 2, NO RATE, DISABLE INTERNAL OSC
MOV STMP3, QASR ; /LOAD CSR.
; /THIS MUST BE DONE IN
; /ORDER TO XFERR COUNTER
; /TO BUFFER ON ST2.
BIS #BIT9, QASR ; /GENERATE ON ST2 PULSE
MOV QABR, \$BDDAT ; /READ THE PRESET BUFFER,
; /PREVIOUS COUNTER
MOV (6)+, QASR ; /CONTENTS ARE IN \$BDDAT.
TST \$BDDAT ; /RESTORE CSR
BNE 15 ; YES, NEXT TEST.2133
2134
2135
2136
2137004754 017746 174416
004760 011637 001424
004764 042737 177707 001424
004772 052737 004005 001424
005000 013777 001424 1743702138
2139
2140
2141005006 052777 001000 174362
005014 017737 174360 0011262142
2143
2144
2145
2146
2147
2148005022 012677 174350
005026 005737 001126
005032 001001

;; \$ ERROR <<< \$

2151 005034 104006

ERROR 6 ; CLOCK FAILED TO INCREMENT.

2152
2153

;; \$ ERROR <<< \$

2156 005036

15:

2157
2158
2159
2160;; \$
*TEST 30 *SEE IF CLOCK WILL COUNT UP FROM A ZERO BASE. RATE:ST12161
2162
2163
2164
2165*
* NOTE: IN THIS TEST, LOOP ON ERROR WILL CAUSE A LOOP
* ON THE FAILING COUT PATTERN;
* WHILE LOOP ON TEST WILL START THE TEST
* AND THE CLOCK FROM ZERO TO THE FAILING COUNT.
*

J04

MAINDEC-11-DVKWA-B
DVKWAB.P11 T30MACY11 27(665) 21-FEB-77 14:43 PAGE 48
*SEE IF CLOCK WILL COUNT UP FROM A ZERO BASE, RATE:ST1

```

2220 005240 005737 001124          TST      $GDDAT          ;ALL DONE?
2221 005244 001312          BNE      1$                    ;NO DO NEXT INCREMENT.
2222 005246          3$:
2223
2224          ;*****
2225          ;*TEST 31          *TEST THAT OVERFLOW (CSR BIT07) WILL SET ON OVERFLOW
2226          ;*****
2227 005246 000004          †ST31: SCOPE
2228
2229 005250 005737 001440          TST      EXS                    ;TESTOR MODE ENABLED??
2230 005254 001411          BEQ      2$                    ;NO-THEN SKIP NEXT SECTION OF CODE.
2231
2232 005256 005037 170500          CLR      @#170500          ;CLEAR TESTOR A/D
2233 005262 005737 170502          TST      @#170502          ;DUMB READ OF A/D BUFFER.
2234 005266 052737 000040 170500  BIS      @#170500          ;ENABLE EXTRENAL START OF A/D.
2235 005274 012700 000010          MOV      @#.,R0           ;SET TIME OUT NUMBER.
2236 005300          2$:
2237 005300 005077 174072          CLR      @ASR              ;CLEAR THE CSR
2238 005304 012777 177777 174066  MOV      #-1,@ABR          ;SET PRESET BUFFER TO ALL ONES.
2239
2240 005312 052777 000061 174056  BIS      @#BIT5!BIT4!BIT0,@ASR ;START CLOCK, RATE ST1.
2241
2242 005320 052777 000400 174050  BIS      @#BIT8,@ASR          ;COUNT CLOCK ONCE, OVERFLOW
2243                                ;SHOULD OCCUR.
2244 005326 105777 174044          TSTB     @ASR              ;DID OVERFLOW SET?
2245 005332 100402          BMI      1$                    ;YES - THEN NEXT TEST
2246
2247          ;,; $$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<< $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
2250 005334 104006          ERROR      6                    ;ERROR - OVERFLOW, CSR BIT0?
2251                                ;FAILED TO SET ON OVERFLOW
2252 005336 000411          BR      TST32
2253 005340 005737 001440          1$:  TST      EXS                    ;TEST EXTERNAL SIGNALS?
2254 005344 001406          BEQ      TST32
2255 005346 105737 1, J500          TSTB     @#170500          ;IF OVERFLOW GOT OUT IT GAVE A/D START
2256                                ;WE'RE LOOKING FOR A/D DONE-DID IT GET SET?
2257 005352 100403          BMI      TST32
2258 005354 005300          DEC      R0
2259 005356 001370          BNE      1$                    ;DID WE ALLOW ENOUGH TIME??
2260                                ;NO-THEN WAIT.
2263 005360 104006          ERROR      6                    ;OVERFLOW OUT NOT DETECTED
2264                                ;BY TESTOR
2265
2268          ;,; $$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<< $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
2269
2270          ;*****
2271          ;*TEST 32          *TEST THAT OVERFLOW WILL CLEAR THE GO BIT
2272          ;*****
2273 005362 000004          †ST32: SCOPE

```

K04

MAINDEC-11-DVKWA-8
DVKWAB.P11 T32MACY11 27(665) 21-FEB-77 14:43 PAGE 49
*TEST THAT OVERFLOW WILL CLEAR THE GO BIT

```

2274
2275 005364 005077 174006          CLR    2ASR          ;CLEAR THE CSR.
2276
2277 005370 012777 177777 174002    MOV    #-1,2ABR      ;PRESET CLOCK TO -1.
2278
2279 005376 052777 000061 173772    BIS     #BIT5!BIT4!BIT0,2ASR ;START CLOCK, RATE:ST1
2280
2281 005404 052777 000400 173764    BIS     #BIT8,2ASR      ;COUNT ONCE, OVERFLOW
2282                                ;SHOULD OCCUR CLEARING
2283                                ;ENABLE (CSR BIT00)
2284
2285 005412 032777 000001 173756    BIT     #BIT0,2ASR      ;DID THE ENABLE CLEAR?
2286 005420 001401                                BEQ     1$          ;YES - NEXT TEST.
2287
2288
                ;;SSSSSSSSSSSSSSSSSSSSSSSSSS>>> ERROR <<<SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
2291 005422 104006          ERROR    6          ;ERROR - OVERFLOW FAILED
2292                                ;TO CLEAR ENABLE (CSR BIT00)
2293
2294 005424          1$:
2295
2296                ;;*****
2297                ;*TEST 33          *TEST THAT GO BIT DOES NOT CLEAR ON OVERFLOW, IF MODE 1
2298                ;******
2299 005424 000004    TST33: SCOPE
2300
2301 005426 005077 173744          CLR    2ASR          ;CLEAR THE CSR.
2302 005432 012777 177777 173740    MOV    #-1,2ABR      ;PRESET BUFFER=ONE COUNT FROM OVERFLOW.
2303 005440 052777 000063 173730    BIS     #63,2ASR      ;MODE 1, RATE:ST1, GO.
2304
2305 005446 052777 000400 173722    BIS     #BIT8,2ASR      ;GENERATE MAINTENANCE ST1.
2306
2307 005454 032777 000001 173714    BIT     #BIT0,2ASR      ;DID ENABLE (GO BIT) CLEAR?
2308 005462 001001                                BNE     1$          ;NO (GOOD) NEXT TEST.
2309
                ;;SSSSSSSSSSSSSSSSSSSSSSSSSS>>> ERROR <<<SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
2312 005464 104006          ERROR    6          ;GO BIT CLEARED ON OVERFLOW
2313                                ;WHEN MODE 1 WAS SELECTED
2314
                ;;SSSSSSSSSSSSSSSSSSSSSSSSSS>>> ERROR <<<SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
2317
2318 005466 005077 173704    1$:    CLR    2ASR          ;CLEAR THE CLOCK.
2319
2320
2321                ;;*****
2322                ;*TEST 34          *TEST THE ABILITY OF CLOCK TO COUNT AT 1MHZ RATE
2323                ;*
2324                ;*THIS TEST IS DESIGNED TO TEST THE CLOCK'S ABILITY
2325
2326
2327

```


MAINDEC-11-DVKWA-B
DVKWAB.P11 T35

MACY11 27(665) 21-FEB-77 14:43 PAGE 51
*TEST THE ABILITY OF CLOCK TO COUNT AT 100KHZ RATE

```
2415 005744 104006          ERROR 6          ;/CLOCK FAILED TO COUNT AT
2416                                     ;/RATE:100KHZ
2417
```

```

2420
2421 005746 005077 173424      25:      CLR      JASR      ;/CLEAR THE CLOCK.

```

```

;*****
;*TEST 36      *TEST THE ABILITY OF CLOCK TO COUNT AT 10KHZ RATE

```

```

;*
;*THIS TEST IS DESIGNED TO TEST THE CLOCK'S ABILITY
;*TO COUNT AT 10KHZ RATE.

```

.....

```

2433 005752 000004          +ST36: SCOPE
2434 005754 012737 000005 001160      MOV      #5,$TIMES      ;;DO 5 ITERATIONS
2435

```

N04

MAINDEC-11-DVKWA-B
DVKWAB.P11 T36

MACY11 27(665) 21-FEB-77 14:43 PAGE 52
*TEST THE ABILITY OF CLOCK TO COUNT AT 10KHZ RATE

```

2436
2437 005762 005077 173410 CLR QASR ;/CLEAR CLOCK
2438 005766 005077 173406 CLR QABR ;/CLEAR PRESET BUFFER
2439 005772 012777 000031 173376 MOV #BIT0!30,QASR ;/START CLOCK, MODE0, RATE:10KHZ
2440 006000 005000 CLR RO ;/NOW WE'LL DO A LITTLE DELAY. THIS DELAY
2441
2442 006002 005200 15: INC RO ;/WILL AMOUNT TO APPROXIMATELY
2443 006004 J01376 BNE 15 ;/369 MS.
2444
2445 006006 017746 173364 MOV QASR, -(6) ;/SAVE CSR
2446 006012 011637 001424 MOV (6), $TMP3 ;/GET CSR.
2447 006016 042737 177707 001424 BIC #177707, $TMP3 ;/SAVE RATE BITS.
2448 006024 052737 004005 001424 BIS #BIT11!BIT2!BIT0, $TMP3 ;/SET MODE 2, NO RATE, DISABLE INTERNAL OSC
2449 006032 013777 001424 173336 MOV $TMP3, QASR ;/LOAD CSR.
2450
2451 ;/THIS MUST BE DONE IN
2452 ;/ORDER TO XFERR COUNTER
2453 006040 052777 001000 173330 BIS #BIT9, QASR ;/GENERATE ON ST2 PULSE
2454 006046 017737 173326 001126 MOV QABR, $BDDAT ;/READ THE PRESET BUFFER,
2455 ;/PREVIOUS COUNTER
2456 006054 012677 173316 MOV (6)+, QASR ;/CONTENTS ARE IN $BDDAT.
2457 006060 005737 001126 TST $BDDAT ;/RESTORE CSR
2458 006064 001004 BNE 25 ;/YES - NEXT TEST.
2459 006066 105766 177776 TSTB -2(6) ;/AT HIGH RATE MAY HAVE HAD OVERFLOW
2460 ;/NOTE: CSR HAD BEEN PUT ON STACK.
2461 006072 100401 BMI 25 ;/NEXT TEST IF OVERFLOW.
2462
2463

```

;; \$ ERROR <<< \$

```

2466 006074 104006 ERROR 6 ;/CLOCK FAILED TO COUNT AT
2467 ;/RATE:10KHZ
2468

```

;; \$ ERROR <<< \$

```

2471
2472 006076 005077 173274 25: CLR QASR ;/CLEAR THE CLOCK.
2473
2474
2475
2476
2477
2478
2479

```

;; *****
*TEST 37 *TEST THE ABILITY OF CLOCK TO COUNT AT 1KHZ RATE

;; *
*THIS TEST IS DESIGNED TO TEST THE CLOCK'S ABILITY
*TO COUNT AT 1KHZ RATE.
*
;; *****

```

2484 006102 000004 TST37: SCOPE
2485 006104 012737 000005 001160 MOV #5, $TIMES ;;DO 5 ITERATIONS
2486
2487
2488 006112 005077 173260 CLR QASR ;/CLEAR CLOCK
2489 006116 005077 173256 CLR QABR ;/CLEAR PRESET BUFFER

```

MAINDEC-11-DVKWA-B
DVKWAB.P11 T37

MACY11 27(665) 21-FEB-77 14:43 PAGE 53
*TEST THE ABILITY OF CLOCK TO COUNT AT 1KHZ RATE

```

2490 006122 012777 000041 173246      MOV    #BIT0!40, @ASR    ;/START CLOCK, MODE0, RATE:1KHZ
2491 006130 005000                      CLR    RO              ;/NOW WE'LL DO A LITTLE DELAY. THIS DELAY
2492                                     1$:   INC    RO              ;/WILL AMOUNT TO APPROXIMATELY
2493 006132 005200                      BNE    1$              ;/369 MS.
2494 006134 001376
2495
2496 006136 017746 173234      MOV    @ASR, -(6)        ;/SAVE CSR
2497 006142 011637 001424      MOV    (6), $TMP3        ;/GET CSR.
2498 006146 042737 177707 001424      BIC    #177707, $TMP3    ;/SAVE RATE BITS.
2499 006154 052737 004005 001424      BIS    #BIT1!BIT2!BIT0, $TMP3 ;/SET MODE 2, NO RATE, DISABLE INTERNAL OSC
2500 006162 013777 001424 173206      MOV    $TMP3, @ASR      ;/LOAD CSR.
2501                                     ;/THIS MUST BE DONE IN
2502                                     ;/ORDER TO XFERR COUNTER
2503                                     ;/TO BUFFER ON ST2.
2504 006170 052777 001000 173200      BIS    #BIT9, @ASR      ;/GENERATE ON ST2 PULSE
2505 006176 017737 173176 001126      MOV    @ABR, $BDDAT    ;/READ THE PRESET BUFFER,
2506                                     ;/PREVIOUS COUNTER
2507 006204 012677 173166      MOV    (6)+, @ASR      ;/CONTENTS ARE IN $BDDAT.
2508 006210 005737 001126      TST    $BDDAT          ;/RESTORE CSR
2509 006214 001004                      BNE    2$              ;/YES - NEXT TEST.
2510 006216 105766 177776      TSTB   -2(6)           ;/AT HIGH RATE MAY HAVE HAD OVERFLOW
2511                                     ;/NOTE: CSR HAD BEEN PUT ON STACK.
2512 006222 100401                      BMI    2$              ;/NEXT TEST IF OVERFLOW.
2513
2514

```

;; \$>>> ERROR <<< \$

```

2517 006224 104006      ERROR    6              ;/CLOCK FAILED TO COUNT AT
2518                                     ;/RATE:1KHZ
2519

```

;; \$>>> ERROR <<< \$

```

2522
2523 006226 005077 173144      2$:   CLR    @ASR              ;/CLEAR THE CLOCK.
2524
2525
2526
2527

```

;; *****
; *TEST 40 *TEST THE ABILITY OF CLOCK TO COUNT AT 100HZ RATE

;; *
; *THIS TEST IS DESIGNED TO TEST THE CLOCK'S ABILITY
; *TO COUNT AT 100HZ RATE.
; *

;; *****

```

2530
2531
2532
2533
2534
2535 006232 000004      ST40:  SCOPE
2536 006234 012737 000005 001160      MOV    #5, $TIMES      ;;DO 5 ITERATIONS
2537
2538
2539 006242 005077 173130      CLR    @ASR              ;/CLEAR CLOCK
2540 006246 005077 173126      CLR    @ABR              ;/CLEAR PRESET BUFFER
2541 006252 012777 000051 173116      MOV    #BIT0!50, @ASR  ;/START CLOCK, MODE0, RATE:100HZ
2542 006260 005000                      CLR    RO              ;/NOW WE'LL DO A LITTLE DELAY. THIS DELAY
2543

```

C05

MAINDEC-11-DVKWA-B
DVKWAB.P11 T40MACY11 27(665) 21-FEB-77 14:43 PAGE 54
*TEST THE ABILITY OF CLOCK TO COUNT AT 100HZ RATE

```

2544 006262 005200      1$: INC R0      ;/WILL AMOUNT TO APPROXIMATELY
2545 006264 001376      BNE 1$      ;/369 MS.
2546
2547 006266 017746 173104      MOV @ASR, -(6)      ;/SAVE CSR
2548 006272 011637 001424      MOV (6), $TMP3      ;/GET CSR.
2549 006276 042737 177707 001424      BIC #177707, $TMP3      ;/SAVE RATE BITS.
2550 006304 052737 004005 001424      BIS #BIT11!BIT2!BIT0, $TMP3      ;/SET MODE 2, NO RATE, DISABLE INTERNAL OSC
2551 006312 013777 001424 173056      MOV $TMP3, @ASR      ;/LOAD CSR.
2552
2553
2554
2555 005320 052777 001000 173050      BIS #BIT9, @ASR      ;/THIS MUST BE DONE IN
2556 006326 017737 173046 001126      MOV @ABR, $BDDAT      ;/ORDER TO XFERR COUNTER
2557
2558 006334 012677 173036      MOV (6)+, @ASR      ;/TO BUFFER ON ST2.
2559 006340 005737 001126      TST $BDDAT      ;/GENERATE ON ST2 PULSE
2560 006344 001004      BNE 2$      ;/READ THE PRESET BUFFER,
2561 006345 105766 177776      TSTB -2(6)      ;/PREVIOUS COUNTER
2562
2563 006352 100401      BMI 2$      ;/CONTENTS ARE IN $BDDAT.
2564
2565

```

```

;; $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ ERROR <<< $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

```

```

2568 006354 104006      ERROR 6      ;/CLOCK FAILED TO COUNT AT
2569
2570

```

```

;; $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ ERROR <<< $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

```

```

2573
2574 006356 005077 173014      2$: CLR @ASR      ;/CLEAR THE CLOCK.
2575
2576
2577

```

```

;; *****
; *TEST 41      *TEST THE ABILITY OF CLOCK TO COUNT AT LINEFREQ RATE

```

```

; *
; *THIS TEST IS DESIGNED TO TEST THE CLOCK'S ABILITY
; *TO COUNT AT LINEFREQ RATE.
; *

```

```

; *****

```

```

2586 006362 000004      TST41: SCOPE
2587 006364 012737 000005 001160      MOV #5, $TIMES      ;;DO 5 ITERATIONS
2588
2589 006372 032777 010000 172540      BIT #BIT12, @SWR      ;/SHALL WE TEST
2590
2591
2592 006400 001450      BEQ TST42      ;/LINE FREQ?
2593
2594 006402 005077 172770      CLR @ASR      ;/SW12=1
2595 006406 005077 172766      CLR @ABR      ;/
2596 006412 012777 000071 172756      MOV #BIT0!70, @ASR      ;/CLEAR CLOCK
2597 006420 005000      CLR R0      ;/CLEAR PRESET BUFFER

```

```

; /START CLOCK, MODE0, RATE:LINEFREQ
; /NOW WE'LL DO A LITTLE DELAY. THIS DELAY

```



```

2706                      .SBTTL *PHASE 4 CLOCK INTERRUPT TEST.
2707                      .SBTTL *
2708
2709
2710                      ;;*****
2711                      ;;*TEST 44      *TEST THAT THE CLOCK WILL INTERRUPT ON OVERFLOW
2712                      ;;*****
2713 006724 000004          †ST44: SCOPE
2714 006726 012737 000020 001160      MOV      #20,$TIMES      ;;DO 20 ITERATIONS
2715
2716
2717 006734 012746 000340      MOV      #340,-(SP)      ;PUT PRIORITY ON STACK.
2718 006740 012746 006746      MOV      #64$,-(SP)      ;PUT RETURN ADDRESS ON STACK
2719 006744 000002      RTI      ;DO AN RTI, PUTS PRIORITY IN CPU.
2720 006746
2721
2722 006746 005077 172424      CLR      @ASR      ;CLEAR CLOCK'S CSR.
2723 006752 012777 177777 172420      MOV      #-1,@ABR      ;SET PRESET BUFFER TO ALL ONES.
2724
2725 006760 012777 000161 172410      MOV      #161,@ASR      ;START CLOCK, RATE:ST1.
2726 006766 052777 000400 172402      BIS      #BIT8,@ASR      ;GENERATE A MAINTENANCE ST1.
2727 006774 012777 007034 172400      MOV      #1$,@VECT1      ;SET INTERRUPT ADDR.
2728
2729 007002 012746 000000      MOV      #0,-(SP)      ;PUT PRIORITY ON STACK.
2730 007006 012746 007014      MOV      #65$,-(SP)      ;PUT RETURN ADDRESS ON STACK
2731 007012 000002      RTI      ;DO AN RTI, PUTS PRIORITY IN CPU.
2732 007014
2733
2734 007014 000240      NOP      ;STALL TIME
2735
2736 007016 012746 000340      MOV      #340,-(SP)      ;PUT PRIORITY ON STACK.
2737 007022 012746 007030      MOV      #66$,-(SP)      ;PUT RETURN ADDRESS ON STACK
2738 007026 000002      RTI      ;DO AN RTI, PUTS PRIORITY IN CPU.
2739 007030
2740
                      ;;*****
                      ;;*****>>> ERROR <<<*****
2743 007030 104007          ERROR 7      ;CLOCK FAILED TO INTERRUPT.
2744
                      ;;*****
                      ;;*****>>> ERROR <<<*****
2747 007032 000402          BR      2$
2748 007034
2749 007034 062706 000004      1$: ADD      #4,SP      ;/ADD #4 TO STACK POINTER.
2750 007040 005077 172332      2$: CLR      @ASR      ;CLEAR THE CLOCK.
2751
2752                      ;;*****
2753                      ;;*TEST 45      *TEST THAT ST2 WILL CAUSE AN INTERRUPT
2754                      ;;*****
2755 007044 000004          †ST45: SCOPE
2756
2757
2758 007046 012746 000340      MOV      #340,-(SP)      ;PUT PRIORITY ON STACK.
2759 007052 012746 007060      MOV      #64$,-(SP)      ;PUT RETURN ADDRESS ON STACK

```

G05

MAINDEC-11-DVKWA-8
DVKWAB.P11 T45MACY11 27(665) 21-FEB-77 14:43 PAGE 58
*TEST THAT ST2 WILL CAUSE AN INTERRUPT

```

2760 007056 000002          RTI          ;DO AN RTI, PUTS PRIORITY IN CPU.
2761 007060          64$:
2762
2763 007060 005077 172312      CLR      @ASR      ;CLEAR CLOCKS CSR.
2764 007064 012777 007140 172314  MOV      #15,@VECT2 ;SET UP INTERRUPT VECTOR.
2765 007072 012777 040001 172276  MOV      #BIT14:BIT0,@ASR ;SET "INT2",AND GO BIT.
2766 007100 052777 001000 172270  BIS      #BIT9,@ASR ;GENERATE A MAINTENANCE ST2.
2767
2768 007106 012746 000000      MOV      #0,-(SP) ;PUT PRIORITY ON STACK.
2769 007112 012746 007120      MOV      #65$,-(SP) ;PUT RETURN ADDRESS ON STACK
2770 007116 000002          RTI          ;DO AN RTI, PUTS PRIORITY IN CPU.
2771 007120          65$:
2772
2773 007120 000240          NOP          ;STALL TIME
2774
2775 007122 012746 000340      MOV      #340,-(SP) ;PUT PRIORITY ON STACK.
2776 007126 012746 007134      MOV      #66$,-(SP) ;PUT RETURN ADDRESS ON STACK
2777 007132 000002          RTI          ;DO AN RTI, PUTS PRIORITY IN CPU.
2778 007134          66$:
2779
      ;; $$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<< $$$$$$$$$$$$$$$$$$$$$$$$$$$$
2782 007134 104007          ERROR 7          ;CLOCK FAILED TO INTERRUPT ON ST2.
2783
      ;; $$$$$$$$$$$$$$$$$$$$$$$$$$$$>>> ERROR <<< $$$$$$$$$$$$$$$$$$$$$$$$$$$$

2786 007136 000402          BR      2$
2787 007140          1$:
2788 007140 062706 000004      ADD      #4,SP          ;/ADD #4 TO STACK POINTER.
2789 007144 005077 172226      CLR      @ASR          ;CLEAR CLOCK'S CSR.
2790
2791          .SBTTL *
2792          .SBTTL *PHASE 5 ADVANCED TESTING
2793          .SBTTL *
2794

```

H05

MAINDEC-11-DVKWA-B
DVKWAB.P11 T46MACY11 27(665) 21-FEB-77 14:43 PAGE 59
*TEST THAT THE "FOR" BIT WILL SET ON 2 ST2'S

```

2795                                     ;*****
2796                                     ;*TEST 46      *TEST THAT THE "FOR" BIT WILL SET ON 2 ST2'S
2797                                     ;*****
2798 007150 000004                       †ST46: SCOPE
2799
2800 007152 005077 172220                 CLR    @ASR           ;START WITH CSR CLEAR.
2801 007156 005277 172214                 INC    @ASR           ;SET GO BIT.
2802 007162 052777 001000 172206          BIS    @BIT9,@ASR      ;GENERATE THE 1ST ST2 PULSE.
2803 007170 052777 001000 172200          BIS    @BIT9,@ASR      ;GENERATE 2ND ST2 PULSE.
2804                                     ;THIS SHOULD CAUSE FOR BIT TO SET.
2805 007176 032777 010000 172172          BIT     @BIT12,@ASR     ;DID FOR BIT SET?
2806 007204 001007                                     ;YES-THEN NEXT TEST.
2807
2808 007206 017737 172164 001126          MOV     @ASR,$BDDAT      ;RECORD CSR.
2809 007214 012737 110001 001124          MOV     @BIT15!BIT12!BIT0,$GDDAT ;RECORD S/B.
2810
                                     ;;*****
2813 007222 104001                       ERROR 1                ;ERROR-"FOR" BIT FAILED TO SET ON
2814                                     ;ON TWO SUCCESSIVE ST2 PULSES.
2815
                                     ;;*****
2818 007224                               1$:
2819
2820                                     ;*****
2821                                     ;*TEST 47      *TEST THAT FOR BIT WILL SET ON TWO OVERFLOWS
2822                                     ;*****
2823 007224 000004                       †ST47: SCOPE
2824 007226 012737 000002 001160          MOV     #2,$TIMES        ;;DO 2 ITERATIONS
2825
2826 007234 005077 172136                 CLR    @ASR           ;START WITH CSR CLEAR.
2827 007240 012777 177777 172132          MOV     #-1,@ABR        ;PRELOAD BUFFER REG.
2828 007246 052777 000013 172122          BIS    @13,@ASR      ;START CLOCK MODE 1, 1MHZ.
2829 007254 005000                                     ;NOW DO A SHORT DELAY.
2830 007256 105200 1$: INCB @R0          ;DURING THIS DELAY, THE CLOCK WILL
2831 007260 001376 BNE 1$                ;HAVE OVERFLOWED TWICE-
2832                                     ;THIS SHOULD CAUSE THE FOR BIT TO SET.
2833
2834 007262 032777 010000 172106          BIT     @BIT12,@ASR     ;DID FOR SET?
2835 007270 001007                                     ;YES-NEXT TEST.
2836
2837 007272 017737 172100 001126          MOV     @ASR,$BDDAT      ;NO-RECORD THE CSR.
2838 007300 012737 010213 001124          MOV     @010213,$GDDAT ;RECORD S/B.
2839
                                     ;;*****
2842 007306 104001                       ERROR 1                ;ERROR FOR BIT FAILED TO SET ON
2843                                     ;TWO SUCCESSIVE OVERFLOWS.
2844
                                     ;;*****
2847 007310                               2$:
2848                                     ;*****

```

MAINDEC-11-DVKWA-B
DVKWAB.P11 T50

MACY11 27(665) 21-FEB-77 14:43 PAGE 60
*TEST THAT FOR BIT WILL CLEAR IF GO BIT IS SET

```

2849 ;*TEST 50      *TEST THAT FOR BIT WILL CLEAR IF GO BIT IS SET
2850 ;*****
2851 007310 000004      †ST50: SCOPE
2852
2853 007312 012777 000001 172056      MOV      #BIT0,‡ASR      ;CLEAR CSR,SET GO BIT.
2854 007320 052777 001000 172050      BIS      #BIT9,‡ASR      ;SET 1ST S2 PULSE.
2855 007326 052777 001000 172042      BIS      #BIT9,‡ASR      ;GENERATE 2ND ST2 PULSE.
2856                                ;FOR BIT SETS HERE.
2857 007334 042777 000001 172034      BIC      #BIT0,‡ASR      ;CLEAR GO BIT.
2858
2859 007342 052777 000001 172026      BIS      #BIT0,‡ASR      ;SET THE "GO" BIT AGAIN -
2860                                ;SHOULD CLEAR FOR BIT.
2861
2862 007350 017737 172022 001126      MOV      ‡ASR,$BDDAT      ;READ THE CSR.
2863
2864 007356 012737 100001 001124      MOV      #100001,$GDDAT      ;RECORD WHAT CSR S/B.
2865 007364 032737 010000 001126      BIT      #BIT12,$BDDAT      ;DID FOR BIT CLEAR?
2866 007372 001401                                BEQ      1$      ;YES NEXT TEST.
2867
                ;; $$$$$$$$$$$$$$$$$$$$$$$$$$$$ ERROR <<< $$$$$$$$$$$$$$$$$$$$$$$$$$$$
2870 007374 104001                                ERROR      1      ;FOR BIT FAILED TO CLEAR
2871                                ;WHEN "GO" BIT WAS SET.
2872
                ;; $$$$$$$$$$$$$$$$$$$$$$$$$$$$ ERROR <<< $$$$$$$$$$$$$$$$$$$$$$$$$$$$
2875 007376 005077 171774      1$:      CLR      ‡ASR      ;CLEAR THE CSR.
2876
2877 ;*****
2878 ;*TEST 51      *TEST THAT WE CAN DISABLE THE INTERNAL OSC
2879 ;*****
2880 007402 000004      †ST51: SCOPE
2881 007404 012737 000005 001160      MOV      #5,$TIMES      ;;DO 5 ITERATIONS
2882
2883 007412 005077 171760      CLR      ‡ASR      ;CLEAR THE CSR
2884 007416 005077 171756      CLR      ‡ABR      ;CLEAR THE PRESET BUFFER
2885 007422 005037 001124      CLR      $GDDAT      ;CLEAR EXPED.
2886
2887 007426 012777 004000 171742      MOV      #BIT11,‡ASR      ;DISABLE THE INTERNAL OSC.
2888 007434 052777 000011 171734      BIS      #BIT3!BIT0,‡ASR ;START CLOCK:RATE 1MHZ.
2889 007442 005000                                CLR      RD
2890 007444 105200      1$:      INCB      RD      ;DELAY A SHORT TIME.
2891 007446 001376                                BNE      1$
2892 007450 017746 171722      MOV      ‡ASR,-(6)      ;/SAVE CSR
2893 007454 011637 001424      MOV      (6),$TMP3      ;/GET CSR.
2894 007460 042737 177707 001424      BIC      #177707,$TMP3      ;/SAVE RATE BITS.
2895 007466 052737 004005 001424      BIS      #BIT11!BIT2!BIT0,$TMP3 ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
2896 007474 013777 001424 171674      MOV      $TMP3,‡ASR      ;/LOAD CSR.
2897                                ;/THIS MUST BE DONE IN
2898                                ;/ORDER TO XFERR COUNTER
2899                                ;/TO BUFFER ON ST2.
2900 007502 052777 001000 171666      BIS      #BIT9,‡ASR      ;/GENERATE ON ST2 PULSE
2901 007510 017737 171664 001126      MOV      ‡ABR,$BDDAT      ;/READ THE PRESET BUFFER,
2902                                ;/PREVIOUS COUNTER

```


MAINDEC-11-DVKWA-B MACY11 27(665) 21-FEB-77 14:43 PAGE 63
 DVKWAB.P11 T53 *TEST THE CLOCK'S 1MHZ DIVIDER

```

3011 010054 052777 002000 171314 4$: BIS #BIT10, @ASR ;/GENERATE ANOTHER OSC PULSE.
3012 010062 005300 DEC RO ;/WHAT WE WANT TO CHECK
3013 010064 001373 BNE 4$ ;/1MHZ PULSE ON 9 OSC PULSES.
3014
3015
3016 010066 017746 171304 MOV @ASR, -(6) ;/SAVE CSR
3017 010072 011637 001424 MOV (6), $TMP3 ;/GET CSR.
3018 010076 042737 177707 001424 BIC #177707, $TMP3 ;/SAVE RATE BITS.
3019 010104 052737 004005 001424 BIS #BIT11!BIT2!BIT0, $TMP3 ;/SET MODE 2, NO RATE, DISABLE INTERNAL OSC
3020 010112 013777 001424 171256 MOV $TMP3, @ASR ;/LOAD CSR.
3021 ;/THIS MUST BE DONE IN
3022 ;/ORDER TO XFERR COUNTER
3023 ;/TO BUFFER ON ST2.
3024 010120 052777 001000 171250 BIS #BIT9, @ASR ;/GENERATE ON ST2 PULSE
3025 010126 017737 171246 001126 MOV @ABR, $BDDAT ;/READ THE PRESET BUFFER,
3026 ;/PREVIOUS COUNTER
3027 010134 012677 171236 MOV (6)+, @ASR ;/CONTENTS ARE IN $BDDAT.
3028 010140 005737 001126 TST $BDDAT ;/RESTORE CSR
3029 010144 023737 001124 001126 CMP $GDDAT, $BDDAT ;/WAS ANOTHER 1MHZ PULSE GENERATED?
3030 010152 001401 BEQ 5$ ;/NO - NEXT TEST.
3031

```

;; \$ ERROR <<< \$

```

3034 010154 104011 ERROR 11 ;/WE SEEM TO HAVE GENERATED
3035 ;/ANOTHER 1MHZ PULSE ON
3036 ;/ONLY 9 MAINTENANCE
3037 ;/OSC PULSES.
3038

```

;; \$ ERROR <<< \$

```

3041 010156 005077 171214 5$: CLR @ASR ;/CLEAR THE CSR.
3042
3043
3044
3045 ;/*****
3046 ;/TEST 54 *TEST THE CLOCK'S 100KHZ DIVIDER
3047 ;/*****
3048 010162 000004 STS4: SCOPE
3049 010164 012737 000005 001160 MOV #5, $TIMES ;/DO 5 ITERATIONS
3050 010172 005077 171200 CLR @ASR ;/CLEAR THE CSR.
3051 010176 005077 171176 CLR @ABR ;/CLEAR THE PRESET BUFFER.
3052 010202 052777 004000 171166 BIS #BIT11, @ASR ;/DISABLE THE INTERNAL OSC.
3053 010210 052777 000021 171160 BIS #1!20, @ASR ;/ENABLE CLOCK, RATE:100KHZ
3054
3055
3056 010216 012700 177634 10$: MOV #-100., RO ;/SET TO GENERATE 100 OSC PULSES.
3057
3058 010222 052777 002000 171146 1$: BIS #BIT10, @ASR ;/GENERATE ONE OSC PULSE.
3059 010230 005200 INC RO ;/DONE 100 OSC PULSES?
3060 010232 001373 BNE 1$ ;/NO - DO ANOTHER ONE.
3061
3062
3063 010234 012737 000001 001124 2$: MOV #1, $GDDAT ;/SET FOR ERROR TYPEOUT - IF ANY.
3064 010242 017746 171130 MOV @ASR, -(6) ;/SAVE CSR

```

MAINDEC-11-DVKWA-B MACY11 27(665) 21-FEB-77 14:43 PAGE 64
DVKWAB.P11 T54 *TEST THE CLOCK'S 100KHZ DIVIDER

Address	Hex	Hex	Hex	Hex	Assembly	Comment
3065	010246	011637	001424		MOV	(6),STMP3 ;/GET CSR.
3066	010252	042737	177707	001424	BIC	#177707,STMP3 ;/SAVE RATE BITS.
3067	010260	052737	004005	001424	BIS	#BIT11!BIT2!BIT0,STMP3 ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
3068	010266	013777	001424	171102	MOV	STMP3,2ASR ;/LOAD CSR.
3069						;/THIS MUST BE DONE IN
3070						;/ORDER TO XFERR COUNTER
3071						;/TO BUFFER ON ST2.
3072	010274	052777	001000	171074	BIS	#BIT9,2ASR ;/GENERATE ON ST2 PULSE
3073	010302	017737	171072	001126	MOV	2ABR,\$BDDAT ;/READ THE PRESET BUFFER,
3074						;/PREVIOUS COUNTER
3075	010310	012677	171062		MOV	(6)+,2ASR ;/CONTENTS ARE IN \$BDDAT.
3076	010314	005737	001126		TST	\$BDDAT ;/RESTORE CSR
3077	010320	013737	001124	001420	MOV	\$GDDAT,STMP0 ;/STMP0 USED IN ERROR TYPEOUT.
3078						
3079	010326	023737	001124	001126	CMP	\$GDDAT,\$BDDAT ;/DID CLOCK ADVANCE ONCE?
3080						
3081	010334	001402			BEQ	35 ;/YES - NEXT TEST.
3082						

```
;:SSSSSSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
```

[illegible]

```
;:SSSSSSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
```

```

3091 010340 000443          BR      5$
3092 010342 012700 000143 3$:  MOV    $99.,RO ;/GET THE NUMBER OF MORE OSC PULSES
3093                                     ;/TO BE GENERATED.
3094
3095 010346 052777 002000 171022 4$:  BIS    #BIT10,2ASR ;/GENERATE ANOTHER OSC PULSE.
3096 010354 005300          DEC    RO ;/WHAT WE WANT TO CHECK
3097 010356 001373          BNE    4$ ;/100KHZ PULSE ON 99 OSC PULSES.

```

ADDRESS	HEX	ASSEMBLY	COMMENT
3100	010360	017746	171012
3101	010364	011637	001424
3102	010370	042737	177707 001424
3103	010376	052737	004005 001424
3104	010404	013777	001424 170764
3105			
3106			
3107			
3108	010412	052777	001000 170756
3109	010420	017737	170754 001126
3110			
3111	010426	012677	170744
3112	010432	005737	001126
3113	010436	023737	001124 001126
3114	010444	001401	
3115			

ADDRESS	HEX	ASSEMBLY	COMMENT
		MOV	2ASR, -(6) ;/SAVE CSR
		(6), \$TMP3 ;/GET CSR.	
		BIC	#177707, \$TMP3 ;/SAVE RATE BITS.
		BIS	#BIT11!BIT2!BIT0, \$TMP3 ;/SET MODE 2, NO RATE, DISABLE INTERNAL OSC
		MOV	\$TMP3, 2ASR ;/LOAD CSR.
			;/THIS MUST BE DONE IN
			;/ORDER TO XFERR COUNTER
			;/TO BUFFER ON ST2.
		BIS	#BIT9, 2ASR ;/GENERATE ON ST2 PULSE
		MOV	2ABR, \$BDDAT ;/READ THE PRESET BUFFER,
			;/PREVIOUS COUNTER
		MOV	(6)+, 2ASR ;/CONTENTS ARE IN \$BDDAT.
		TST	\$BDDAT ;/RESTORE CSR
		CMP	\$GDDAT, \$BDDAT ;/WAS ANOTHER 100KHZ PULSE GENERATED?
		BEQ	\$S ;/NO - NEXT TEST.

```
::SSSSSSSSSSSSSSSSSSSSSSSS>> ERROR (<<SSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
```

3118 010446 104011 ERROR 11 :/WE SEEM TO HAVE GENERATED


```
3202 010740 104011 ERROR 11 ;/WE SEEM TO HAVE GENERATED
3203 ;/ANOTHER 10KHZ PULSE ON
3204 ;/ONLY 999 MAINTENANCE
3205 ;/OSC PULSES.
3206
```

```
3209 010742 005077 170430          SS:  CLR  DADR          ;/CLEAR THE CSR.
```

```

;*****
;TEST 56      *TEST THE CLOCK'S 1KHZ DIVIDER
;*****

```

3216	010746	000004			↑\$T56:	SCOPE		
3217	010750	012737	000005	001160		MOV	#5,\$TIMES	;;DO 5 ITERATIONS
3218								
3219	010756	005077	170414			CLR	\$ASR	;/CLEAR THE CSR.
3220	010762	005077	170412			CLR	\$ABR	;/CLEAR THE PRESET BUFFER.
3221	010766	052777	004000	170402		BIS	#BIT11,\$ASR	;/DISABLE THE INTERNAL OSC.
3222	010774	052777	000041	170374		BIS	#1!40,\$ASR	;/ENABLE CLOCK, RATE:1KHZ
3223								
3224								
3225	011002	012700	154360		10\$:	MOV	#-10000.,RO	;/SET TO GENERATE 10000 OSC PULSES.
3226								

.....

```

3389          ;*TEST 60          *TEST THE CLOCK'S MODE 2 OPERATION
3390          ;*
3391          ;*IN THIS TEST WE'LL CHECK MODE 2 OPERATION
3392          ;*MODE 2: EXTERNAL EVENTS TIMING MODE
3393          ;*SETTING THE GO BIT CAUSES THE COUNTER TO BEGIN COUNTING FROM
3394          ;*ZERO AND TO FREE-RUN UNTIL THE GO BIT IS WRITTEN
3395          ;*TO A ZERO THE COUNTER WILL CONTINUE COUNTING AFTER
3396          ;*OVERFLOW. AN EXTERNAL PULSE FROM SCHMITZ TRIGGER 2
3397          ;*(WHEN ST2 GO ENABLE IS A "0") CAUSES DATA TO
3398          ;*TRANSFER FROM THE COUNTER TO THE BUFFER/PRESET REG.
3399          ;*WHILE THE COUNTER CONTINUES TO RUN.
3400          ;*
3401          ;*TO TEST THIS MODE, WE'LL DISABLE THE INTERNAL OSC AND USE
3402          ;*MAINTENANCE OSC PULSES AS WELL AS A MAINTENANCE
3403          ;*ST2.
3404          ;*
3405          ;*****
3406 011552 000004          †ST60: SCOPE
3407 011554 012737 000020 001160      MOV      #20,$TIMES          ;;DO 20 ITERATIONS
3408
3409 011562 005077 167610          CLR      @ASR          ;CLEAR THE CSR.
3410 011566 005077 167606          CLR      @ABR          ;CLEAR THE PRESET REG.
3411 011572 012777 004015 167576      MOV      #004015,@ASR      ;START CLOCK.
3412
3413 011600 012700 177754          1$:      MOV      #-20,$RO          ;SET TO GIVE 20 MAINTENANCE OSC.
3414 011604 052777 002000 167564      2$:      BIS      #BIT10,@ASR      ;GENERATE A MAINTENANCE OSC.
3415 011612 005200
3416 011614 001373          INC      $RO
3417          BNE      2$          ;IF NOT DONE 20 TIMES, LOOP.
3418 011616 052777 001000 167552      3$:      BIS      #BIT9,@ASR          ;HERE'S THE BIGGIE! AN ST2 HAS BEEN GENERATED
3419 011624 012737 000002 001124      MOV      #2,$GDDAT          ;THE PRESET BUFFER SHOULD BE 2.
3420 011632 017737 167542 001126      MOV      @ABR,$BDDAT          ;READ THE PRESET BUFFER.
3421 011640 023737 001126 001124      CMP      $BDDAT,$GDDAT      ;DID A COUNTER TO PRESET BUFFER OCCUR?
3422 011646 001402          BEQ      4$          ;YES - NEXT SUBTEST.
3423
          ;; $$$$$$$$$$$$$$$$$$$$$$$$$$ ERROR <<< $$$$$$$$$$$$$$$$$$$$$$$$$$
          ERROR      5          ;A COUNTER TO PRESET BUFFER DID NOT
          ;HAPPEN PROPERLY.
          ;; $$$$$$$$$$$$$$$$$$$$$$$$$$ ERROR <<< $$$$$$$$$$$$$$$$$$$$$$$$$$
          BR      5$
3431 011652 000434          4$:      MOV      @ASR,-(6)          ;/SAVE CSR
3432
3433 011654          MOV      (6),$TMP3          ;/GET CSR.
3434 011654 017746 167516          BIC      #177707,$TMP3          ;/SAVE RATE BITS.
3435 011660 011637 001424          BIS      #BIT11!BIT2!BIT0,$TMP3      ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
3436 011664 042737 177707 001424      MOV      $TMP3,@ASR          ;/LOAD CSR.
3437 011672 052737 004005 001424          ;/THIS MUST BE DONE IN
3438 011700 013777 001424 167470          ;/ORDER TO XFERR COUNTER
3439          ;/TO BUFFER ON ST2.
3440          ;/GENERATE ON ST2 PULSE
3441
3442 011706 052777 001000 167462      BIS      #BIT9,@ASR

```

MAINDEC-11-DVKWA-B
DVKWAB.P11 T60

MACY11 27(665) 21-FEB-77 14:43 PAGE 71
*TEST THE CLOCK'S MODE 2 OPERATION

```

3443 011714 017737 167460 001126      MOV      JABR,$BDDAT      ;/READ THE PRESET BUFFER,
3444                                     ;/PREVIOUS COUNTER
3445 011722 012677 167450      MOV      (6)+,JASR      ;/CONTENTS ARE IN $BDDAT.
3446 011726 005737 001126      TST      $BDDAT      ;/RESTORE CSR
3447 011732 023737 001124 001126      CMP      $GDDAT,$BDDAT      ;WAS THE COUNTER ACCIDENTLY ZEROED?
3448 011740 001401      BEQ      5$      ;NO - NEXT TEST.
3449

```

;; \$>>> ERROR <<< \$

```

3452 011742 104005      ERROR      5      ;THE COUNT REGISTER SHOULD NOT
3453                                     ;HAVE BEEN EFFECTED BY THE ST2
3454                                     ;IN MODE 2.
3455

```

;; \$>>> ERROR <<< \$

3458 011744
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468

5\$:

;; \$

;; *TEST 61 *TEST THE CLOCK'S MODE 3 OPERATION

```

; *
; *IN THIS TEST WE'LL CHECK MODE 3 OPERATION.
; *MODE 3 IS JUST LIKE MODE 2 EXCEPT THAT THE COUNT
; *REG IS ZEROED AFTER AN ST2.
; *

```

;; \$

†ST61: SCOPE

```

3469 011744 000004      MOV      #20,$TIMES      ;:DO 20 ITERATIONS
3470 011746 012737 000020 001160      CLR      JASR      ;CLEAR THE CSR.
3471                                     CLR      JABR      ;CLEAR THE BUFFER REG.
3472 011754 005077 167416      MOV      #4017,JASR      ;START CLOCK.
3473 011760 005077 167414
3474 011764 012777 004017 167404      MOV      #4017,JASR
3475
3476 011772 012700 177754 1$:      MOV      #-20,R0      ;SET TO GIVE 20 MAINTENANCE OSC.
3477 011776 052777 002000 167372 2$:      BIS      #BIT10,JASR      ;GENERATE A MAINTENANCE OSC.
3478 012004 005200      INC      R0
3479 012006 001373      BNE      2$      ;IF NOT DONE 20 TIMES, LOOP.
3480
3481 012010 052777 001000 167360 3$:      BIS      #BIT9,JASR      ;HERE'S THE BIGGIE! AN ST2 HAS BEEN GENERATED
3482 012016 012737 000002 001124      MOV      #2,$GDDAT      ;THE PRESET BUFFER SHOULD BE 2.
3483 012024 017737 167350 001126      MOV      JABR,$BDDAT      ;READ THE PRESET BUFFER.
3484 012032 023737 001126 001124      CMP      $BDDAT,$GDDAT      ;DID A COUNTER TO PRESET BUFFER OCCUR?
3485 012040 001402      BEQ      4$      ;YES - NEXT SUBTEST.
3486

```

;; \$>>> ERROR <<< \$

```

3489 012042 104005      ERROR      5      ;A COUNTER TO PRESET BUFFER DID NOT
3490                                     ;HAPPEN PROPERLY.
3491

```

;; \$>>> ERROR <<< \$

```

3494 012044 000445      BR      TST62      ;:
3495
3496 012046 005037 001124 4$:      CLR      $GDDAT      ;EXPECT ZERO BACK FROM COUNT REG.

```

MAINDEC-11-DVKWA-B
DVKWAB.P11 T61

MACY11 27(665) 21-FEB-77 14:43 PAGE 72
*TEST THE CLOCK'S MODE 3 OPERATION

```

3497 012052 017746 167320      MOV    QASR, -(6)      ;/SAVE CSR
3498 012056 011637 001424      MOV    (6), $TMP3      ;/GET CSR.
3499 012062 042737 177707 001424  BIC    #177707, $TMP3      ;/SAVE RATE BITS.
3500 012070 052737 004005 001424  BIS    #BIT11!BIT2!BIT0, $TMP3 ;/SET MODE 2, NO RATE, DISABLE INTERNAL OSC
3501 012076 013777 001424 167272  MOV    $TMP3, QASR      ;/LOAD CSR.
3502                                     ;/THIS MUST BE DONE IN
3503                                     ;/ORDER TO XFERR COUNTER
3504                                     ;/TO BUFFER ON ST2.
3505 012104 052777 001000 167264  BIS    #BIT9, QASR      ;/GENERATE ON ST2 PULSE
3506 012112 017737 167262 001126  MOV    QABR, $BDDAT      ;/READ THE PRESET BUFFER,
3507                                     ;/PREVIOUS COUNTER
3508 012120 012677 167252      MOV    (6)+, QASR      ;/CONTENTS ARE IN $BDDAT.
3509 012124 005737 001126      TST    $BDDAT      ;/RESTORE CSR
3510 012130 001402      BEQ    SS      ;IF SO - NEXT TEST.
3511

```

;; \$ ERROR <<< \$

```

3514 012132 104005      ERROR    5      ;THE CLOCK FORGOT TO ZERO THE COUNT
3515                                     ;REG. AFTER AN ST2 OCCURRED ON
3516                                     ;A MODE 3 COUNT.
3517

```

;; \$ ERROR <<< \$

```

3520 012134 000411      BR        TST62      ;;
3521 012136      SS:      CLR        QASR      ;NOW TRY CLEARING THE CSR.
3522 012136 005077 167234  MOV    QASR, $BDDAT ;READ THE CSR - DID IT CLEAR?
3523 012142 017737 167230 001126  BEQ    TST62      ;
3524 012150 001403      CLR    $BDDAT      ;NO - RECORD S/B.
3525 012152 005037 001124
3526

```

;; \$ ERROR <<< \$

```

3529 012156 104002      ERROR    2      ;CSR FAILED TO CLEAR
3530

```

;; \$ ERROR <<< \$

```

3533
3534
3535      ;*****
3536      ;*TEST 62      *IF ENABLED, CHECK THRESHOLD ST1 FROM TESTOR
3537      ;*****
3538      ;ST62: SCOPE

```

```

3539 012162 012737 000002 001160  MOV    #2, $TIMES      ;DO 2 ITERATIONS.
3540 012170 005737 001440      TST    EXS      ;OPERATING IN TESTOR MODE?
3541 012174 001002      BNE    4$      ;YES DO THIS TEST.
3542 012176 000137 013016      JMP    ENDP      ;NO-END PASS
3543 012202
3544

```

```

3545 012202 005077 167170      CLR    QASR      ;YES-CLEAR CSR.
3546 012206 012777 177775 167164  MOV    #-3, QABR      ;SET TO COUNT THREE TIMES.
3547 012214 012777 000061 167154  MOV    #61, QASR      ;SET RATE: ST1 MODE 1 GO.
3548 012222 104401 012230      TYPE    65$      ;TYPE ASCII STRING
3549 012226 000432      BR        64$      ;GET OVER THE ASCIIZ
3550      ;65$: .ASCIIZ <200><7><7> *SET ST1 THRESHOLD POT FULLY COUNTERCLOCKWISE...#(7)

```

MAINDEC-11-DVKWA-B
DVKWAB.P11 T62

MACY11 27(665) 21-FEB-77 14:43 PAGE 73
*IF ENABLED,CHECK THRESHOLD ST1 FROM TESTOR

```

3551 012314      64$:
3552 012314 004737 013700      JSR    PC,ANYKEY      ;TYPE LAST MESSAGE AND WAIT FOR OPERATER.
3553 012320 012737 177775 001124  MOV    #-3,$BDDAT      ;DON'T EXPECT COUNT TO CHANGE.
3554 012326 017746 167044      MOV    @ASR,-(6)      ;/SAVE CSR
3555 012332 011637 001424      MOV    (6),$TMP3      ;/GET CSR.
3556 012336 042737 177707 001424  BIC    #177707,$TMP3      ;/SAVE RATE BITS.
3557 012344 052737 004005 001424  BIS    #BIT11!BIT2!BIT0,$TMP3 ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
3558 012352 013777 001424 167016  MOV    $TMP3,@ASR      ;/LOAD CSR.
3559                                     ;/THIS MUST BE DONE IN
3560                                     ;/ORDER TO XFERR COUNTER
3561                                     ;/TO BUFFER ON ST2.
3562 012360 052777 001000 167010  BIS    #BIT9,@ASR      ;/GENERATE ON ST2 PULSE
3563 012366 017737 167006 001126  MOV    @ABR,$BDDAT      ;/READ THE PRESET BUFFER,
3564                                     ;/PREVIOUS COUNTER
3565 012374 012677 166776      MOV    (6)+,@ASR      ;/CONTENTS ARE IN $BDDAT.
3566 012400 005737 001126      TST    $BDDAT      ;/RESTORE CSR
3567 012404 001002      BNE    2$      ;IF NON-ZERO,WE'RE OK.
3568

```

;; \$ ERROR <<< \$

```

3571 012406 104002      ERROR    2      ;COUNTERCLOCKWISE THRESHOLD SETTING
3572                                     ;OF ST1 SHOULD HAVE INHIBITED
3573                                     ;ST1 FROM CAUSEING CLOCK TO COUNT.
3574

```

;; \$ ERROR <<< \$

```

3577 012410 000463      BR        TST63      ;;
3578
3579 012412      2$:
3580 012412 104401 012420      TYPE    67$      ;;TYPE ASCIZ STRING
3581 012416 000425      BR        66$      ;/GET OVER THE ASCIZ
3582                                     ;/67$: .ASCIZ <200><7><7>#SET ST1 THRESHOLD POT FULLY CLOCKWISE#
3583                                     ;/66$:
3584 012472 004737 013700      3$: JSR    PC,ANYKEY      ;TYPE LAST MESS. AND WAIT FOR OPERATOR.
3585 012476 017746 166674      MOV    @ASR,-(6)      ;/SAVE CSR
3586 012502 011637 001424      MOV    (6),$TMP3      ;/GET CSR.
3587 012506 042737 177707 001424  BIC    #177707,$TMP3      ;/SAVE RATE BITS.
3588 012514 052737 004005 001424  BIS    #BIT11!BIT2!BIT0,$TMP3 ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC
3589 012522 013777 001424 166646  MOV    $TMP3,@ASR      ;/LOAD CSR.
3590                                     ;/THIS MUST BE DONE IN
3591                                     ;/ORDER TO XFERR COUNTER
3592                                     ;/TO BUFFER ON ST2.
3593 012530 052777 001000 166640  BIS    #BIT9,@ASR      ;/GENERATE ON ST2 PULSE
3594 012536 017737 166636 001126  MOV    @ABR,$BDDAT      ;/READ THE PRESET BUFFER,
3595                                     ;/PREVIOUS COUNTER
3596 012544 012677 166626      MOV    (6)+,@ASR      ;/CONTENTS ARE IN $BDDAT.
3597 012550 005737 001126      TST    $BDDAT      ;/RESTORE CSR
3598 012554 001001      BNE    TST63      ;;
3599

```

;; \$ ERROR <<< \$

```

3602 012556 104002      ERROR    2      ;CLOCKWISE THRESHOLD SETTING OF
3603                                     ;ST1 SHOULD HAVE INHIBITED CLOCK
3604                                     ;FROM COUNTING.

```

3605

;; \$>>> ERROR <<< \$

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

012560 000004
012562 012737 000002 001160012570 005737 001440
012574 001510012576 005077 166574
012602 012777 177775 166570
012610 012777 000061 166560
012616 101401 012624
012622 000433012712
012712 004737 013700
012716 005037 001124
012722 017746 166450
012726 011637 001424
012732 042737 177707 001424
012740 052737 004005 001424
012746 013777 001424 166422012754 052777 001000 166414
012762 017737 166412 001126012770 012677 166402
012774 005737 001126

013000 001402

;; \$>>> ERROR <<< \$

*TEST 63 *IF ENABLED, CHECK ST1,ST2 IN FROM TESTOR

;; \$>>> ERROR <<< \$

†ST63: SCOPE

MOV #2,STIMES ;DO 2 ITERATIONS

TST EXS ;OPERATING IN TESTOR MODE?
BEQ ENDP ;NO-EXIT THIS TEST.

CLR QASR ;CLEAR THE CSR

MOV #-3,QABR ;SET TO COUNT THREE TIMES

MOV #61,QASR ;SET RATE:ST1, MODE1, GO

TYPE 65\$;TYPE ASCII STRING

BR 64\$;GET OVER THE ASCII

;;65\$: .ASCIIZ <200><7><7>*SET ST1 THRESHOLD POT IN THE MIDDLE OF ITS RANGE.*
64\$:

JSR PC,ANYKEY ;TYPE LAST MESS WAITE FOR OPERATOR.

CLR \$GDDAT ;EXPECT COUNTER TO BE CLEAR

MOV QASR,-(6) ;/SAVE CSR

MOV (6),STMP3 ;/GET CSR.

BIC #177707,STMP3 ;/SAVE RATE BITS.

BIS #BIT11!BIT2!BIT0,STMP3 ;/SET MODE 2, NO RATE,DISABLE INTERNAL OSC

MOV STMP3,QASR ;/LOAD CSR.

;/THIS MUST BE DONE IN

;/ORDER TO XFERR COUNTER

;/TO BUFFER ON ST2.

;/GENERATE ON ST2 PULSE

;/READ THE PRESET BUFFER,

;/PREVIOUS COUNTER

;/CONTENTS ARE IN \$BDDAT.

;/RESTORE CSR

;/OF NON-ZERO - REPORT ERROR

;; \$>>> ERROR <<< \$

ERROR 2 ;INCORRECT # OF ST1'S
;RECEIVED BY CLOCK

;; \$>>> ERROR <<< \$

BR ENDP

2\$:

TST QASR ;DID ST2 FLG SET?
BMI ENDP

;; \$>>> ERROR <<< \$

ERROR 6 ;ST2 FLAG DID SET EVEN THOUGH

3650 013004 000404

3651

3652

3653

3654

3655

013006
013006 005777 166364
013012 100401

3658 013014 104006

MACY11 27(665) 21-FEB-77 14:43 PAGE 75
*IF ENABLED, CHECK ST1,ST2 IN FROM TESTOR

[illegible]

```

      JMP      LOOP      :TEST NEW UNIT.

```

MAINDEC-11-DVKWA-B
DVKWAB.P11 T63

MACY11 27(665) 21-FEB-77 14:43 PAGE 76
*IF ENABLED, CHECK ST1,ST2 IN FROM TESTOR

```

3713
3714 013234
3715 013234 062706 000004
3716 013240 012637 000004
3717 013244 022737 000000 001204
3718 013252 001424
3719
3720 013254
3721 013254 104401 013262
3722 013260 000405
3723
3724 013274
3725 013274 013746 001204
3726 013300 104402
3727 013302 104401 013310
3728 013306 000406
3729
3730 013324
3731
3732 013324 013737 001250 001376
3733 013332 013737 001244 001402
3734 013340 013737 001204 001442
3735 013346 005237 001442
3736 013352 012737 000000 001204
3737
3738 013363 005037 001434
3739 013364 012737 000001 001426
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749 013372
3750 013372 000240
3751 013374 005037 001102
3752 013400 005037 001160
3753 013404 005237 001202
3754 013410 042737 100000 001202
3755 013416 005327
3756 013420 000001
3757 013422 003122
3758 013424 012737
3759 013426 000001
3760 013430 013420
3761 013432 104401 013440
3762 013436 000406
3763
3764 013454
3765 013454 013746 001202
3766 013460 104402

15:      ADD      #4,SP          ;/ADD #4 TO STACK POINTER.
          MOV      (6)+,ERRVEC    ;RESTORE LOC 4
          CMP      #0,$DEVCT      ;TESTED ONLY ONE UNIT?
          BEQ      25             ;YES - NO NEED FOR TYPEOUT.

45:      TYPE      715           ;:TYPE ASCIIZ STRING
          BR       705           ;:GET OVER THE ASCIIZ
::715:   .ASCIIZ   <15><12>"UNIT #"
705:      MOV      $DEVCT,-(SP)    ;:SAVE $DEVCT FOR TYPEOUT
          TYP0C      ;:GO TYPE--OCTAL ASCII(ALL DIGITS)
          TYPE      735           ;:TYPE ASCIIZ STRING
          BR       725           ;:GET OVER THE ASCIIZ
::735:   .ASCIIZ   " COMPLETED "
725:

25:      MOV      $BASE,ASR
          MOV      $VECT1,VECT1
          MOV      $DEVCT,LCNT
          INC      LCNT
          MOV      #0,$DEVCT

          CLR      MDEVCT          ;BEGIN TESTING 1ST UNIT.
          MOV      #1,ROTATE       ;POINT TO IT.

.SBTTL   END OF PASS ROUTINE

;*****
;INCREMENT THE PASS NUMBER ($PASS)
;IF THERES A MONITOR GO TO IT
;IF THERE ISN'T JUMP TO LOOP

$EOP:
          NOP
          CLR      $STNM           ;:ZERO THE TEST NUMBER
          CLR      $TIMES          ;:ZERO THE NUMBER OF ITERATIONS
          INC      $PASS           ;:INCREMENT THE PASS NUMBER
          BIC      #100000,$PASS   ;:DON'T ALLOW A NEG. NUMBER
          DEC      (PC)+           ;:LOOP?
$EOPCT:   .WORD    1
          BGT      $DOAGN          ;:YES
          MOV      (PC)+,2(PC)+    ;:RESTORE COUNTER
$ENDCT:   .WORD    1
          $EOPCT
          TYPE      655           ;:TYPE ASCIIZ STRING
          BR       645           ;:GET OVER THE ASCIIZ
::655:   .ASCIIZ   <15><12>#ENDPASS #
645:      MOV      $PASS,-(SP)     ;:SAVE $PASS FOR TYPEOUT
          TYP0C      ;:GO TYPE--OCTAL ASCII(ALL DIGITS)

```

```

3767 013462 104401 013470      TYPE      67$      ;;TYPE ASCIZ STRING
3768 013466 000411              BR      66$      ;;GET OVER THE ASCIZ
3769                      ;;67$: .ASCIZ # TOTAL ERRORS #
3770 013512              66$: MOV      ERCNT, -(SP)      ;;SAVE ERCNT FOR TYPEOUT
3771 013512 013746 001432      TYP0C      ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3772 013516 104402              TYPE      69$      ;;TYPE ASCIZ STRING
3773 013520 104401 013526      BR      68$      ;;GET OVER THE ASCIZ
3774 013524 000407              ;;69$: .ASCIZ #; THERE ARE #
3775                      68$: MOV      LCNT, -(SP)      ;;SAVE LCNT FOR TYPEOUT
3776 013544              TYP0C      ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3777 013544 013746 001442      TYPE      71$      ;;TYPE ASCIZ STRING
3778 013550 104402              BR      70$      ;;GET OVER THE ASCIZ
3779 013552 104401 013560      ;;71$: .ASCIZ # (OCTAL) UNITS. #
3780 013556 000411              70$: TYPE      73$      ;;TYPE ASCIZ STRING
3781                      BR      72$      ;;GET OVER THE ASCIZ
3782 013602              ;;73$: .ASCIZ <200>#THE GOOD UNITS (L TO R) #
3783 013602 104401 013610      72$: MOV      UTEST, -(SP)      ;;SAVE UTEST FOR TYPEOUT
3784 013606 000415              TYPBN      ;;GO TYPE--BINARY ASCII
3785                      SGET42: MOV      2*42, R0      ;;GET MONITOR ADDRESS
3786 013642              BEQ      SDOAGN      ;;BRANCH IF NO MONITOR
3787 013642 013746 001430      RESET      ;;CLEAR THE WORLD
3788 013646 104405              SENDAD: JSR      PC, (R0)      ;;GO TO MONITOR
3789 013650 013700 000042      NOP      ;;SAVE ROOM
3790 013654 001405              NOP      ;;FOR
3791 013656 000005              NOP      ;;ACT11
3792 013660 004710              SDOAGN: JMP      2(PC)+      ;;RETURN
3793 013662 000240      SRTNAD: .WORD      LOOP
3794 013664 000240      SNULL: .BYTE      -1, -1, 0      ;;NULL CHARACTER STRING
3795 013666 000240              .EVEN
3796 013670
3797 013670 000137
3798 013672 002274
3799 013674      377      377      000
3800      013700
3801
3802
3803      ;;THIS ROUTINE TYPES LAST MESSAGE AND WAITS FOR AN OPERATOR
3804      ;;RESPONCE.
3805
3806
3807 013700 105777 165242      ANYKEY: TSTB      2$TKB      ;;CLEAR TTY READY FLAG.
3808 013704 104401 013712      TYPE      65$      ;;TYPE ASCIZ STRING
3809 013710 000430      BR      64$      ;;GET OVER THE ASCIZ
3810                      ;;65$: .ASCIZ <200><7>#SWITCH ST1 3 TIMES, TYPE ANY KEY WHEN DONE...#<7>
3811 013772              64$:
3812
3813 013772 105777 165146      1$: TSTB      2$TKS      ;;WAIT FOR OPERATOR.
3814 013776 100375              BPL      1$
3815 014000 105777 165142      TSTB      2$TKB      ;;CLEAR TTY READY FLAG.
3816 014004 000207      RTS PC
3817
3818      .SBTTL      ;;I/O SIGNAL TEST #1 ST1 IN AND ST2 OUT IN AND OUT
3819
3820

```

N06

3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874

014006	104407		
014010	005077	165362	
014014	005077	165360	
014020	012777	000061	165350
014026	052777	001000	165342
014034	012777	000005	165334
014042	052777	001000	165326
014050	027727	165324	000001
014056	001753		
014060	104000		
014062	000751		

IOTST1: CKSWR
1\$: CLR JASR
CLR JABR
MOV #61, JASR
BIS #BIT9, JASR
MOV #5, JASR
BIS #BIT9, JASR
CMP JABR, #1
BEQ IOTST1
ERROR
BR IOTST1

;CHECK THE SWR
;CLEAR THE CSR
;CLEAR THE BUFFER REG.
;RATE ST1, MODE 0, GO.
;GENERATE A MAINTENANCE ST2.
;NOW SET TO READ COUNT REG
;FORCE COUNT -> BUFFER REG.
;DID COUNT REG ADVANCE ONCE?
;YES - LOOP.
;ST2 OUT TO ST1 IN FAILED.

.SBTTL

;I/O SIGNAL TEST #2 CLOCK OVFLOW OUT TEST.

SWITCH PACK S2 MUST BE SET UP AS FOLLOWS:

SWITCH 1 - OFF
2 - ON
3 - OFF
4 - OFF
5 - ON
6 - ON
7 - NOT USED

THIS SELECTS TTL THRESHOLDS AND POSITIVE SLOPE FOR SCHMITT TRIGGER 1.

PLEASE REMOVE ANY PREVIOUS JUMPER.

JUMPER THE FOLLOWING PINS TOGETHER:

J1 - SS (ST2 OUT) TO J1 - VV (ST1-IN)

LOAD AND START AT LOCATION 210
END PASSES OCCUR IMMEDIATELY AND ARE NOT REPORTED
ERRORS ARE REPORTED AS IN THE REGULAR LOGIC TEST AND
THEIR PRINTOUT MAY BE INHIBITED

SWITCH PACK S2 MUST BE SET UP AS FOLLOWS:

SWITCH 1 - OFF
2 - OFF
3 - OFF
4 - ON
5 - ON
6 - OFF
7 - NOT USED

THIS SELECTS TTL THRESHOLDS AND POSITIVE SLOPE FOR

MAINDEC-11-DVKWA-B
DVKWAB.P11

MACY11 27(665) 21-FEB-77 14:43 PAGE 79
;I/O SIGNAL TEST #2 CLOCK OVFLOW OUT TEST.

```

3875      ; SCHMITT TRIGGER 2.
3876
3877      ; PLEASE REMOVE ANY PREVIOUS JUMPER.
3878
3879      ; JUMPER THE FOLLOWING PINS TOGETHER:
3880
3881      ; J1 - RR (CLK OV) TO J1 - TT (ST2-IN)
3882
3883      ; LOAD AND START AT LOCATION 214.
3884      ; END PASSES OCCUR IMMEDIATELY AND ARE NOT REPORTED.
3885      ; ERRORS ARE REPORTED AS IN TH REGULAR LOGIC TEST AND
3886      ; THEIR PRINTOUT MAY BE INHIBITED.
3887
3888      014064 104407      IOTST2: CKSWR      ; CHECK THE SWR.
3889      014066 005077      CLR      QASR      ; CLEAR THE CSR.
3890      014072 012777 165304      MOV      #-1,QABR      ; PRELOAD PRESET BUFFER.
3891      014100 012777 000063 165270      MOV      #63,QASR      ; RATE ST1, MODE 1, GO.
3892      014106 052777 000400 165262      BIS      #BIT8,QASR      ; GENERATE A MAIN. ST1.
3893      014114 000240
3894      014116 000240
3895      014120 005777 165252      TST      QASR      ; DID OVERFLOW SET ST2 FLAG?
3896      014124 100757      BMI      IOTST2      ; YES - LOOP
3897      014126 104000      ERROR
3898      014130 000755      BR      IOTST2      ; CLK OV OUT TO ST2 IN FAILED.
3899      ; LOOP
3900
3901      .SBTTL      ; I/O SIGNAL TEST #3 ST1 OUT AND ST2 IN
3902
3903
3904
3905      ; SWITCH PACK S2 MUST BE SET UP AS FOLLOWS:
3906      ; SWITCH 1 - OFF
3907      ; 2 - OFF
3908      ; 3 - OFF
3909      ; 4 - ON
3910      ; 5 - ON
3911      ; 6 - ON
3912      ; 7 - NOT USED
3913      ; THIS SELECTS TTL THRESHOLD AND POSITIVE SLOPE FOR
3914      ; SCHMITT TRIGGER 2.
3915
3916      ; PLEASE REMOVE ANY PREVIOUS JUMPERS.
3917
3918      ; JUMPER THE FOLLOWING PINS TOGETHER:
3919      ; J1 - UU (ST1 OUT) TO J1 - TT (ST2-IN)
3920
3921      ; LOAD AND START AT LOCATION 220
3922      ; END PASSES OCCUR IMMEDIATELY AND ARE NOT REPORTED
3923      ; ERRORS ARE REPORTED AS IN THE REGULAR LOGIC TEST AND
3924      ; THEIR PRINTOUT MAY BE INHIBITED
3925
3926      014132 104407      IOTST3: CKSWR      ; CHECK THE SWR
3927      014134 012777 000001 165234      MOV      #1,QASR      ; SET GO BIT.
3928      014142 052777 000400 165226      BIS      #BIT8,QASR      ; GENERATE A MAIN. ST1.

```

```

3929 014150 005777 165222          TST      QASR          ;DID ST2 FLAG SET?
3930 014154 100401          BMI      IS              ;
3931 014156 104000          ERROR          ;ST1 OUT TO ST2 IN FAILED
3932
3933 014160 032777 010000 165210 15:  BIT      #BIT12,QASR      ;DID "FOR" BIT SET?
3934 014166 001761          BEQ      IOTST3          ;NO - GOOD!
3935 014170 104000          ERROR          ;"FOR" BIT SET ON ONLY 1 ST2.
3936 014172 000757          BR       IOTST3          ;LOOP
3937

```

```

3938          .SBTTL
3939          .SBTTL  *SYSMAC ROUTINES
3940          .SBTTL

```

```

3941          .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
3942

```

```

3943          ;*****
3944          ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
3945          ;OCTAL (ASCII) NUMBER AND TYPE IT.
3946          ;*STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3947          ;*CALL:
3948          ;*      MOV      NUM,-(SP)          ;;NUMBER TO BE TYPED
3949          ;*      TYPOS          ;;CALL FOR TYPEOUT
3950          ;*      .BYTE  N          ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3951          ;*      .BYTE  M          ;;M=1 OR 0
3952          ;*
3953          ;*          ;;1=TYPE LEADING ZEROS
3954          ;*          ;;0=SUPPRESS LEADING ZEROS
3955          ;*

```

```

3956          ;*STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
3957          ;*STYPOS OR STYPOC
3958          ;*CALL:

```

```

3959          ;*      MOV      NUM,-(SP)          ;;NUMBER TO BE TYPED
3960          ;*      TYPON          ;;CALL FOR TYPEOUT
3961          ;*

```

```

3962          ;*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3963          ;*CALL:

```

```

3964          ;*      MOV      NUM,-(SP)          ;;NUMBER TO BE TYPED
3965          ;*      TYPOC          ;;CALL FOR TYPEOUT
3966          ;*

```

```

3967 014174 017646 000000          STYPOS: MOV      Q(SP),-(SP)      ;;PICKUP THE MODE
3968 014200 116637 000001 014417  MOVB      1(SP),SOFILL      ;;LOAD ZERO FILL SWITCH
3969 014206 112637 014421          MOVB      (SP)+,SOMODE+1      ;;NUMBER OF DIGITS TO TYPE
3970 014212 062716 000002          ADD      #2,(SP)          ;;ADJUST RETURN ADDRESS
3971 014216 000406          BR       STYPON
3972 014220 112737 000001 014417  STYPOC: MOVB      #1,SOFILL      ;;SET THE ZERO FILL SWITCH
3973 014226 112737 000006 014421  MOVB      #6,SOMODE+1      ;;SET FOR SIX(6) DIGITS
3974 014234 112737 000005 014416  STYPON: MOVB      #5,SOCNT      ;;SET THE ITERATION COUNT
3975 014242 010346          MOV      R3,-(SP)          ;;SAVE R3
3976 014244 010446          MOV      R4,-(SP)          ;;SAVE R4
3977 014246 010546          MOV      R5,-(SP)          ;;SAVE R5
3978 014250 113704 014421          MOVB      $OMODE+1,R4      ;;GET THE NUMBER OF DIGITS TO TYPE
3979 014254 005404          NEG      R4
3980 014256 062704 000006          ADD      #6,R4          ;;SUBTRACT IT FOR MAX. ALLOWED
3981 014262 110437 014420          MOVB      R4,SOMODE      ;;SAVE IT FOR USE
3982 014266 113704 014417          MOVB      $OFILL,R4      ;;GET THE ZERO FILL SWITCH

```

MAINDEC-11-DVKWA-B MACY11 27(665) 21-FE3-77 14:43 PAGE 81
 DVKWAB.P11 BINARY TO OCTAL (ASCII) AND TYPE

```

3983 014272 016605 000012      MOV      12(SP),R5      ;; PICKUP THE INPUT NUMBER
3984 014276 005003              CLR      R3              ;; CLEAR THE OUTPUT WORD
3985 014300 006105              1$:    ROL      R5              ;; ROTATE MSB INTO "C"
3986 014302 000404              BR       3$              ;; GO DO MSB
3987 014304 006105              2$:    ROL      R5              ;; FORM THIS DIGIT
3988 014306 006105              ROL      R5
3989 014310 006105              ROL      R5
3990 014312 010503              MOV      R5,R3
3991 014314 006103              3$:    ROL      R3              ;; GET LSB OF THIS DIGIT
3992 014316 105337 014420      DECB     $OMODE          ;; TYPE THIS DIGIT?
3993 014322 100016              BPL      7$              ;; BR IF NO
3994 014324 042703 177770      BIC      #177770,R3      ;; GET RID OF JUNK
3995 014330 001002              BNE      4$              ;; TEST FOR 0
3996 014332 005704              TST      R4              ;; SUPPRESS THIS 0?
3997 014334 001403              BEQ      5$              ;; BR IF YES
3998 014336 005204              4$:    INC      R4              ;; DON'T SUPPRESS ANYMORE 0'S
3999 014340 052703 000060      BIS      #'0,R3          ;; MAKE THIS DIGIT ASCII
4000 014344 052703 000040      5$:    BIS      #' ,R3          ;; MAKE ASCII IF NOT ALREADY
4001 014350 110337 014414      MOVB     R3,$$          ;; SAVE FOR TYPING
4002 014354 104401 014414      TYPE     8$              ;; GO TYPE THIS DIGIT
4003 014360 105337 014416      7$:    DECB     $OCNT          ;; COUNT BY 1
4004 014364 003347              BGT      2$              ;; BR IF MORE TO DO
4005 014366 002402              BLT      6$              ;; BR IF DONE
4006 014370 005204              INC      R4              ;; INSURE LAST DIGIT ISN'T A BLANK
4007 014372 000744              BR       2$              ;; GO DO THE LAST DIGIT
4008 014374 012605              6$:    MOV      (SP)+,R5      ;; RESTORE R5
4009 014376 012604              MOV      (SP)+,R4          ;; RESTORE R4
4010 014400 012603              MOV      (SP)+,R3          ;; RESTORE R3
4011 014402 016666 000002 000004 MOV      2(SP),4(SP)      ;; SET THE STACK FOR RETURNING
4012 014410 012616              MOV      (SP)+,(SP)
4013 014412 000002              RTI
4014 014414 000              8$:    .BYTE     0              ;; RETURN
4015 014415 000              .BYTE     0              ;; STORAGE FOR ASCII DIGIT
4016 014416 000              $OCNT:   .BYTE     0              ;; TERMINATOR FOR TYPE ROUTINE
4017 014417 000              $OFILL:  .BYTE     0              ;; OCTAL DIGIT COUNTER
4018 014420 000000              $OMODE:  .WORD      0              ;; ZERO FILL SWITCH
4019                                .SBTTL   BINARY TO ASCII AND TYPE ROUTINE
4020
4021                                ;; *****
4022                                ;; *THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 16-BIT
4023                                ;; *BINARY-ASCII NUMBER AND TYPE IT.
4024                                ;; *CALL:
4025                                ;; *      MOV      NUMBER,-(SP)      ;; NUMBER TO BE TYPED
4026                                ;; *      TYPBN
4027                                ;; *
4028 014422 010146              $TYPBN: MOV      R1,-(SP)      ;; SAVE R1 ON THE STACK
4029 014424 016601 000006      MOV      6(SP),R1          ;; GET THE INPUT NUMBER
4030 014430 000261              SEC
4031 014432 112737 000060 014474 1$:    MOVB     #'0,$BIN          ;; SET "C" SO CAN KEEP TRACK OF THE NUMBER OF BITS
4032 014440 006101              ROL      R1              ;; SET CHARACTER TO AN ASCII "0".
4033 014442 001406              BEQ      2$              ;; GET THIS BIT
4034 014444 105537 014474      ADCB     $BIN              ;; DONE?
4035 014450 104401 014474      TYPE     , $BIN          ;; NO--SET THE CHARACTER EQUAL TO THIS BIT
4036 014454 000241              CLC              ;; GO TYPE THIS BIT
                                ;; CLEAR "C" SO CAN KEEP TRACK OF BITS

```

```

4037 014456 000765          BR      1$          ;;GO DO THE NEXT BIT
4038 014460 012601          2$:  MOV      (SP)+,R1      ;;POP THE STACK INTO R1
4039 014462 016666 000002 000004  MOV      2(SP),4(SP)  ;;ADJUST THE STACK
4040 014470 012616          MOV      (SP)+,(SP)
4041 014472 000002          RTI              ;;RETURN TO USER
4042 014474      000      000  $BIN:  .BYTE  0,0      ;;STORAGE FOR ASCII CHAR. AND TERMINATOR
4043
4044
4045          .SBTTL  ERROR HANDLER ROUTINE
4046
4047          ;*****
4048          ;THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT.
4049          ;SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
4050          ;AND GO TO $ERRTYP ON ERROR
4051          ;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
4052          ;$SW15=1      HALT ON ERROR
4053          ;$SW13=1      INHIBIT ERROR TYPEOUTS
4054          ;$SW10=1      BELL ON ERROR
4055          ;$SW09=1      LOOP ON ERROR
4056          ;$CALL
4057          ;*      ERROR  N      ;;ERROR=EMT AND N=ERROR ITEM NUMBER
4058
4059          $ERROR:
4060 014476 014476 104407          CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
4061 014500 105237 001103          7$:  INCB      $ERFLG      ;;SET THE ERROR FLAG
4062 014504 001775          BEQ      7$          ;;DON'T LET THE FLAG GO TO ZERO
4063 014506 013777 001102 164426  MOV      $TSTNM,$DISPLAY  ;;DISPLAY TEST NUMBER AND ERROR FLAG
4064 014514 032777 002000 164416  BIT      #BIT10,$SWR      ;;BELL ON ERROR?
4065 014522 001402          BEQ      1$          ;;NO - SKIP
4066 014524 104401 001164          TYPE      $BELL      ;;RING BELL
4067 014530 005237 001112          1$:  INC      $ERTTL      ;;COUNT THE NUMBER OF ERRORS
4068 014534 011637 001116          MOV      (SP), $ERRPC  ;;GET ADDRESS OF ERROR INSTRUCTION
4069 014540 162737 000002 001116  SUB      #2,$ERRPC
4070 014546 117737 164344 001114  MOVB     $ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
4071 014554 032777 020000 164356  BIT      #BIT13,$SWR      ;;SKIP TYPEOUT IF SET
4072 014562 001004          BNE      20$          ;;SKIP TYPEOUTS
4073 014564 004737 014704          JSR      PC,$ERRTYP  ;;GO TO USER ERROR ROUTINE
4074 014570 104401 001171          TYPE      , $CRLF
4075 014574
4076 014574 122737 000001 001214  20$:  CMPB     #APTENV,$ENV      ;;RUNNING IN APT MODE
4077 014602 001007          BNE      2$          ;;NO, SKIP APT ERROR REPORT
4078 014604 113737 001114 014616  MOVB     $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
4079 014612 004737 016376          JSR      PC,$ATY4      ;;REPORT FATAL ERROR TO APT
4080 014616      000          21$:  .BYTE  0
4081 014617      000          .BYTE  0
4082 014620 000777          22$:  BR      22$          ;;APT ERROR LOOP
4083 014622 005777 164312          2$:  TST      $SWR          ;;HALT ON ERROR
4084 014626 100002          BPL      3$          ;;SKIP IF CONTINUE
4085 014630 000000          HALT          ;;HALT ON ERROR!
4086 014632 104407          CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
4087 014634 032777 001000 164276  3$:  BIT      #BIT09,$SWR      ;;LOOP ON ERROR SWITCH SET?
4088 014642 001402          BEQ      4$          ;;BR IF NO
4089 014644 013716 001110          MOV      $LPERR,(SP)  ;;FUDGE RETURN FOR LOOPING
4090 014650 005737 001162          4$:  TST      $ESCAPE      ;;CHECK FOR AN ESCAPE ADDRESS

```

```

4091 014654 001402      BEQ      5$      ;;BR IF NONE
4092 014656 013716 001162      MOV      $ESCAPE,(SP)  ;;FUDGE RETURN ADDRESS FOR ESCAPE
4093 014662      5$:
4094
4095 014662 005237 001432      INC      ERCNT      ;/UPDATE ERROR COUNT.
4096 014666 001002      BNE      10$      ;/BUT DON'T LET IT OVERFLOW.
4097 014670 005337 001432      DEC      ERCNT      ;/KEEP AT 177777 IF OVERFLOW.
4098 014674      10$:
4099 014674 043737 001426 001430      BIC      ROTATE,UTEST  ;/REMOVE UNIT FROM LIST OF GOOD ONES.
4100 014702 000002      RTI      ;/EXIT.
4101
4102      .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
4103
4104      ;*****
4105      ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
4106      ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
4107      ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
4108
4109      $ERRTYP:
4110 014704 104401 001171      TYPE      $CRLF      ;;"CARRIAGE RETURN" & "LINE FEED"
4111 014710 010046      MOV      RO,-(SP)      ;;SAVE RO
4112 014712 005000      CLR      RO      ;;PICKUP THE ITEM INDEX
4113 014714 153700 001114      BISB      2*$ITEMB,RO
4114 014720 001004      BNE      1$
4115
4116 014722 013746 001116      MOV      $ERRPC,-(SP)      ;;IF ITEM NUMBER IS ZERO, JUST
4117
4118 014726 104402      TYPEOC      ;;TYPE THE PC OF THE ERROR
4119 014730 000426      BR      6$      ;;SAVE $ERRPC FOR TYPEOUT
4120 014732 005300      1$:      DEC      RO      ;;ERROR ADDRESS
4121 014734 006300      ASL      RO      ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
4122 014736 006300      ASL      RO      ;;GET OUT
4123 014740 006300      ASL      RO      ;;ADJUST THE INDEX SO THAT IT WILL
4124 014742 062700 001256      ADD      $ERRTB,RO      ;;WORK FOR THE ERROR TABLE
4125 014746 012037 014756      MOV      (RO)+,2$
4126 014752 001404      BEQ      3$
4127 014754 104401      TYPE      ;;FORM TABLE POINTER
4128 014756 000000      .WORD      0      ;;PICKUP "ERROR MESSAGE" POINTER
4129 014760 104401 001171      TYPE      $CRLF      ;;SKIP TYPEOUT IF NO POINTER
4130 014764 012037 014774      3$:      MOV      (RO)+,4$      ;;TYPE THE "ERROR MESSAGE"
4131 014770 001404      BEQ      5$      ;;"ERROR MESSAGE" POINTER GOES HERE
4132 014772 104401      TYPE      ;;"CARRIAGE RETURN" & "LINE FEED"
4133 014774 000000      .WORD      0      ;;PICKUP "DATA HEADER" POINTER
4134 014776 104401 001171      TYPE      $CRLF      ;;SKIP TYPEOUT IF 0
4135 015002 011000      5$:      MOV      (RO),RO      ;;TYPE THE "DATA HEADER"
4136 015004 001004      BNE      7$      ;;"DATA HEADER" POINTER GOES HERE
4137 015006 012600      6$:      MOV      (SP)+,RO      ;;"CARRIAGE RETURN" & "LINE FEED"
4138 015010 104401 001171      TYPE      $CRLF      ;;PICKUP "DATA TABLE" POINTER
4139 015014 000207      RTS      PC      ;;GO TYPE THE DATA
4140
4141 015016 013046      7$:      MOV      2(RO)+,-(SP)      ;;RESTORE RO
4142 015020 104402      TYPEOC      ;;"CARRIAGE RETURN" & "LINE FEED"
4143 015022 005710      TST      (RO)      ;;RETURN
4144 015024 001770      BEQ      6$      ;;SAVE 2(RO)+ FOR TYPEOUT
      ;GO TYPE--OCTAL ASCII(ALL DIGITS)
      ;IS THERE ANOTHER NUMBER?
      ;BR IF NO

```

```

4145 015026 104401 015034 TYPE BS ;TYPE TWO(2) SPACES
4146 015032 000771 BR 7S ;LOOP
4147 015034 020040 000 BS: .ASCIZ / / ;TWO(2) SPACES
4148 015040 .EVEN
4149 .SBTTL SCOPE HANDLER ROUTINE
4150
4151 ;*****
4152 ;THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
4153 ;AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
4154 ;AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
4155 ;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
4156 ;*SW14=1 LOOP ON TEST
4157 ;*SW11=1 INHIBIT ITERATIONS
4158 ;*SW09=1 LOOP ON ERROR
4159 ;*SW08=1 LOOP ON TEST IN SWR<7:0>
4160 ;*CALL
4161 ;* SCOPE ;SCOPE=IOT
4162
4163 015040 $SCOPE:
4164 015040 104407 CKSWR ;TEST FOR CHANGE IN SOFT-SWR
4165 015042 104407 CKSWR
4166 015044 032777 040000 164066 1S: BIT #BIT14,$SWR ;LOOP ON PRESENT TEST?
4167 015052 001114 BNE $OVER ;YES IF SW14=1
4168 ;*****START OF CODE FOR THE XOR TESTER*****
4169 015054 000416 $XTSTR: BR 6S ;IF RUNNING ON THE "XOR" TESTER CHANGE
4170 ;THIS INSTRUCTION TO A "NOP" (NOP=240)
4171 015056 013746 000004 MOV $ERRVEC,-(SP) ;SAVE THE CONTENTS OF THE ERROR VECTOR
4172 015062 012737 015102 000004 MOV $S,$ERRVEC ;SET FOR TIMEOUT
4173 015070 005737 177060 TST $#177060 ;TIME OUT ON XOR?
4174 015074 012637 000004 MOV (SP)+,$ERRVEC ;RESTORE THE ERROR VECTOR
4175 015100 000463 BR $SVLAD ;GO TO THE NEXT TEST
4176 015102 022626 5S: CMP (SP)+,(SP)+ ;CLEAR THE STACK AFTER A TIME OUT
4177 015104 012637 000004 MOV (SP)+,$ERRVEC ;RESTORE THE ERROR VECTOR
4178 015110 000423 BR 7S ;LOOP ON THE PRESENT TEST
4179 015112 6S: ;*****END OF CODE FOR THE XOR TESTER*****
4180 015112 032777 000400 164020 BIT #BIT08,$SWR ;LOOP ON SPEC. TEST?
4181 015120 001404 BEQ 2S ;BR IF NO
4182 015122 127737 164012 001102 CMPB $SWR,$TSTNM ;ON THE RIGHT TEST? SWR<7:0>
4183 015130 001465 BEQ $OVER ;BR IF YES
4184 015132 105737 001103 2S: TSTB $ERFLG ;HAS AN ERROR OCCURRED?
4185 015136 001421 BEQ 3S ;BR IF NO
4186 015140 123737 001115 001103 CMPB $ERMAX,$ERFLG ;MAX. ERRORS FOR THIS TEST OCCURRED?
4187 015146 101015 BHI 3S ;BR IF NO
4188 015150 032777 001000 163762 BIT #BIT09,$SWR ;LOOP ON ERROR?
4189 015156 001404 BEQ 4S ;BR IF NO
4190 015160 013737 001110 001106 7S: MOV $LPERR,$LPADR ;SET LOOP ADDRESS TO LAST SCOPE
4191 015166 000446 BR $OVER
4192 015170 105037 001103 4S: CLRB $ERFLG ;ZERO THE ERROR FLAG
4193 015174 005037 001160 CLR $TIMES ;CLEAR THE NUMBER OF ITERATIONS TO MAKE
4194 015200 000415 BR 1S ;ESCAPE TO THE NEXT TEST
4195 015202 032777 004000 163730 3S: BIT #BIT11,$SWR ;INHIBIT ITERATIONS?
4196 015210 001011 BNE 1S ;BR IF YES
4197 015212 005737 001202 TST $PASS ;IF FIRST PASS OF PROGRAM
4198 015216 001406 BEQ 1S ;INHIBIT ITERATIONS

```

```

4199 015220 005237 001104      INC      $ICNT      ;; INCREMENT ITERATION COUNT
4200 015224 023737 001160 001104    CMP      $TIMES,$ICNT  ;; CHECK THE NUMBER OF ITERATIONS MADE
4201 015232 002024      BGE      $OVER      ;; BR IF MORE ITERATION REQUIRED
4202 015234 012737 000001 001104 1$:  MOV      #1,$ICNT  ;; REINITIALIZE THE ITERATION COUNTER
4203 015242 013737 015320 001160    MOV      $MXCNT,$TIMES  ;; SET NUMBER OF ITERATIONS TO DO
4204 015250 105237 001102      $SVLAD: INCB    $STNM      ;; COUNT TEST NUMBERS
4205 015254 113737 001102 001200    MOVB    $STNM,$STNM  ;; SET TEST NUMBER IN APT MAILBOX
4206 015262 011637 001106      MOV      (SP),$LPADR  ;; SAVE SCOPE LOOP ADDRESS
4207 015266 011637 001110      MOV      (SP),$LPERR  ;; SAVE ERROR LOOP ADDRESS
4208 015272 005037 001162      CLR      $ESCAPE  ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
4209 015276 112737 000001 001115    MOVB    #1,$ERMAX  ;; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
4210 015304 013777 001102 163630 $OVER: MOV      $STNM,$DISPLAY  ;; DISPLAY TEST NUMBER
4211 015312 013716 001106      MOV      $LPADR,(SP)  ;; FUDGE RETURN ADDRESS
4212 015316 000002      RTI      ;; FIXES PS
4213 015320 003720      $MXCNT: 2000.  ;; MAX. NUMBER OF ITERATIONS
4214      .SBTTL  TTY INPUT ROUTINE
4215
4216      ;;*****
4217      .ENABL  LSB
4218
4219      ;;*****
4220      ;;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
4221      ;;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
4222      ;;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
4223      ;;*WHEN OPERATING IN TTY FLAG MODE.
4224 015322 022737 000176 001140 $CKSWR: CMP      #SWREG,SWR  ;; IS THE SOFT-SWR SELECTED?
4225 015330 001074      BNE      1$      ;; BRANCH IF NO
4226 015332 105777 163606      TSTB    @STKS      ;; CHAR THERE?
4227 015336 100071      BPL      1$      ;; IF NO, DON'T WAIT AROUND
4228 015340 117746 163602      MOVB    @STKB,-(SP)  ;; SAVE THE CHAR
4229 015344 042716 177600      BIC      #177,(SP)  ;; STRIP-OFF THE ASCII
4230 015350 022726 000007      CMP      #7,(SP)+  ;; IS IT A CONTROL G?
4231 015354 001062      BNE      1$      ;; NO, RETURN TO USER
4232 015356 123727 001134 000001    CMPB    $AUTOB,#1  ;; ARE WE RUNNING IN AUTO-MODE?
4233 015364 001456      BEQ      1$      ;; BRANCH IF YES
4234
4235 015366 104401 016047      $GTSWR: TYPE    ,SCNTLG  ;; ECHO THE CONTROL-G (↑G)
4236 015372 104401 016054      TYPE    ,SMSWR      ;; TYPE CURRENT CONTENTS
4237 015376 013746 000176      MOV      SWREG,-(SP)  ;; SAVE SWREG FOR TYPEOUT
4238 015402 104402      TYPOC    ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
4239 015404 104401 016065      TYPE    ,SMNEW      ;; PROMPT FOR NEW SWR
4240 015410 005046      1$:  CLR      -(SP)  ;; CLEAR COUNTER
4241 015412 005046      CLR      -(SP)  ;; THE NEW SWR
4242 015414 105777 163524      7$:  TSTB    @STKS      ;; CHAR THERE?
4243 015420 100375      BPL      7$      ;; IF NOT TRY AGAIN
4244
4245 015422 117746 163520      MOVB    @STKB,-(SP)  ;; PICK UP CHAR
4246 015426 042716 177600      BIC      #177,(SP)  ;; MAKE IT 7-BIT ASCII
4247
4248
4249
4250 015432 021627 000025      9$:  CMP      (SP),#25  ;; IS IT A CONTROL-U?
4251 015436 001005      BNE      10$      ;; BRANCH IF NOT
4252 015440 104401 016042      TYPE    ,SCNTLU  ;; YES, ECHO CONTROL-U (↑U)

```

MAINDEC-11-DVKWA-8 MACY11 27(665) 21-FEB-77 14:43 PAGE 86
 DVKWAB.P11 TTY INPUT ROUTINE

```

4253 015444 062706 000006      20$: ADD    #6,SP      ;; IGNORE PREVIOUS INPUT
4254 015450 000757              BR      19$      ;; LET'S TRY IT AGAIN
4255
4256
4257 015452 021627 000015      10$: CMP      (SP),#15      ;; IS IT A <CR>?
4258 015456 001022              BNE      16$      ;; BRANCH IF NO
4259 015460 005766 000004              TST      4(SP)      ;; YES, IS IT THE FIRST CHAR?
4260 015464 001403              BEQ      11$      ;; BRANCH IF YES
4261 015466 016677 000002 163444      MOV      2(SP),@SWR      ;; SAVE NEW SWR
4262 015474 062706 000006      11$: ADD      #6,SP      ;; CLEAR UP STACK
4263 015500 104401 001171      14$: TYPE    $CRLF      ;; ECHO <CR> AND <LF>
4264 015504 123727 001135 000001      CMPB     $INTAG,#1      ;; RE-ENABLE TTY KBD INTERRUPTS?
4265 015512 001003              BNE      15$      ;; BRANCH IF NOT
4266 015514 012777 000100 163422      MOV      #100,@STKS      ;; RE-ENABLE TTY KBD INTERRUPTS
4267 015522 000002      15$: RTI              ;; RETURN
4268 015524 004737 016310      16$: JSR      PC,$TYPEC      ;; ECHO CHAR
4269 015530 021627 000060              CMP      (SP),#60      ;; CHAR < 0?
4270 015534 002420              BLT      18$      ;; BRANCH IF YES
4271 015536 021627 000067              CMP      (SP),#67      ;; CHAR > 7?
4272 015542 003015              BGT      18$      ;; BRANCH IF YES
4273 015544 042726 000060              BIC      #60,(SP)+      ;; STRIP-OFF ASCII
4274 015550 005766 000002              TST      2(SP)      ;; IS THIS THE FIRST CHAR
4275 015554 001403              BEQ      17$      ;; BRANCH IF YES
4276 015556 006316              ASL      (SP)      ;; NO, SHIFT PRESENT
4277 015560 006316              ASL      (SP)      ;; CHAR OVER TO MAKE
4278 015562 006316              ASL      (SP)      ;; ROOM FOR NEW ONE.
4279 015564 005266 000002      17$: INC      2(SP)      ;; KEEP COUNT OF CHAR
4280 015570 056616 177776              BIS      -2(SP),(SP)      ;; SET IN NEW CHAR
4281 015574 000707              BR      7$      ;; GET THE NEXT ONE
4282 015576 104401 001170      18$: TYPE    $QUES      ;; TYPE ?<CR><LF>
4283 015602 000720              BR      20$      ;; SIMULATE CONTROL-U
4284 .DSABL  LSB
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295 015604 011646 000004 000002 $RDCHR: MOV      (SP),-(SP)      ;; PUSH DOWN THE PC
4296 015606 016666 000004 000002      MOV      4(SP),2(SP)      ;; SAVE THE PS
4297 015614 105777 163324      1$: TSTB     @STKS      ;; WAIT FOR
4298 015620 100375              BPL      1$      ;; A CHARACTER
4299 015622 117766 163320 000004      MOVB     @STKB,4(SP)      ;; READ THE TTY
4300 015630 042766 177600 000004      BIC      #177,4(SP)      ;; GET RID OF JUNK IF ANY
4301 015636 026627 000004 000023      CMP      4(SP),#23      ;; IS IT A CONTROL-S?
4302 015644 001013              BNE      3$      ;; BRANCH IF NO
4303 015646 105777 163272      2$: TSTB     @STKS      ;; WAIT FOR A CHARACTER
4304 015652 100375              BPL      2$      ;; LOOP UNTIL ITS THERE
4305 015654 117746 163266      MOVB     @STKB,-(SP)      ;; GET CHARACTER
4306 015660 042716 177600              BIC      #177,(SP)      ;; MAKE IT 7-BIT ASCII

```

```

;*****
;THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
;CALL:
;* RDCHR
;* RETURN HERE
;*****

```

```

;; INPUT A SINGLE CHARACTER FROM THE TTY
;; CHARACTER IS ON THE STACK
;; WITH PARITY BIT STRIPPED OFF

```

```

4307 015664 022627 000021      CMP      (SP)+, #21      ;; IS IT A CONTROL-0?
4308 015670 001366      BNE      2$      ;; IF NOT DISCARD IT
4309 015672 000750      BR       1$      ;; YES, RESUME
4310 015674 026627 000004 000140 3$:  CMP      4(SP), #140      ;; IS IT UPPER CASE?
4311 015702 002407      BLT      4$      ;; BRANCH IF YES
4312 015704 026627 000004 000175      CMP      4(SP), #175      ;; IS IT A SPECIAL CHAR?
4313 015712 003003      BGT      4$      ;; BRANCH IF YES
4314 015714 042766 000040 000004      BIC      #40, 4(SP)      ;; MAKE IT UPPER CASE
4315 015722 000002      RTI      ;; GO BACK TO USER
4316      ;; *****
4317      ;; *THIS ROUTINE WILL INPUT A STRING FROM THE TTY
4318      ;; *CALL:
4319      ;; *      RDLIN      ;; INPUT A STRING FROM THE TTY
4320      ;; *      RETURN HERE      ;; ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
4321      ;; *      ;; TERMINATOR WILL BE A BYTE OF ALL 0'S
4322
4323 015724 010346      $RDLIN: MOV      R3, -(SP)      ;; SAVE R3
4324 015726 012703 016032 1$:  MOV      #STTYIN, R3      ;; GET ADDRESS
4325 015732 022703 016042 2$:  CMP      #STTYIN+8., R3      ;; BUFFER FULL?
4326 015736 101405      BLOS      4$      ;; BR IF YES
4327 015740 104410      RDCHR      ;; GO READ ONE CHARACTER FROM THE TTY
4328 015742 112613      MOV      (SP)+, (R3)      ;; GET CHARACTER
4329 015744 122713 000177 10$:  CMP      #177, (R3)      ;; IS IT A RUBOUT
4330 015750 001003      BNE      3$      ;; SKIP IF NOT
4331 015752 104401 001170 4$:  TYPE      $QUES      ;; TYPE A '?'
4332 015756 000763      BR       1$      ;; CLEAR THE BUFFER AND LOOP
4333 015760 111337 016030 3$:  MOV      (R3), 9$      ;; ECHO THE CHARACTER
4334 015764 104401 016030      TYPE      9$
4335 015770 122723 000015      CMP      #15, (R3)+      ;; CHECK FOR RETURN
4336 015774 001356      BNE      2$      ;; LOOP IF NOT RETURN
4337 015776 105063 177777      CLRB      -1(R3)      ;; CLEAR RETURN (THE 15)
4338 016002 104401 001172      TYPE      $LF      ;; TYPE A LINE FEED
4339 016006 012603      MOV      (SP)+, R3      ;; RESTORE R3
4340 016010 011646      MOV      (SP), -(SP)      ;; ADJUST THE STACK AND PUT ADDRESS OF THE
4341 016012 016666 000004 000002      MOV      4(SP), 2(SP)      ;; FIRST ASCII CHARACTER ON IT
4342 016020 012766 016032 000004      MOV      #STTYIN, 4(SP)
4343 016026 000002      RTI      ;; RETURN
4344 016030 000      9$:  .BYTE      0      ;; STORAGE FOR ASCII CHAR. TO TYPE
4345 016031 000      .BYTE      0      ;; TERMINATOR
4346 016032 000010      $TTYIN: .BLKB      8.      ;; RESERVE 8 BYTES FOR TTY INPUT
4347 016042 052536 005015 000      $CNTLU: .ASCIZ  /↑U/<15><12>      ;; CONTROL "U"
4348 016047 136 006507 000012      $CNTLG: .ASCIZ  /↑G/<15><12>      ;; CONTROL "G"
4349 016054 005015 053523 020122      $MSWR: .ASCIZ  <15><12>/SWR = /
4350 016062 020075 000
4351 016065 040 047040 053505      $MNEW: .ASCIZ  / NEW = /
4352 016072 036440 000040
4353      .SBTTL TYPE ROUTINE
4354
4355      ;; *****
4356      ;; *ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
4357      ;; *THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
4358      ;; *NOTE1:      $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
4359      ;; *NOTE2:      $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
4360      ;; *NOTE3:      $FILLC CONTAINS THE CHARACTER TO FILL AFTER.

```

K07

MAINDEC-11-DVKWA-B MACY11 27(665) 21-FEB-77 14:43 PAGE 88
 DVKWA8.P11 TYPE ROUTINE

```

4361
4362
4363
4364
4365
4366
4367
4368
4369
4370 016076 105737 001157 $TYPE: TSTB $TPFLG ;; IS THERE A TERMINAL?
4371 016102 100002 BPL 1$ ;; BR IF YES
4372 016104 000000 HALT ;; HALT HERE IF NO TERMINAL
4373 016106 000430 BR 3$ ;; LEAVE
4374 016110 010046 1$: MOV RO, -(SP) ;; SAVE RO
4375 016112 017600 000002 MOV 22(SP), RO ;; GET ADDRESS OF ASCIZ STRING
4376 016116 122737 000001 001214 CMPB #APTENV, $ENV ;; RUNNING IN APT MODE
4377 016124 001011 BNE 62$ ;; NO GO CHECK FOR APT CONSOLE
4378 016126 132737 000100 001215 BITB #APTPOOL, $ENVM ;; SPOOL MESSAGE TO APT
4379 016134 001405 BEQ 62$ ;; NO GO CHECK FOR CONSOLE
4380 016136 010037 016146 MOV RO, 61$ ;; SETUP MESSAGE ADDRESS FOR APT
4381 016142 004737 016366 JSR PC, $ATY3 ;; SPOOL MESSAGE TO APT
4382 016146 000000 61$: .WORD 0 ;; MESSAGE ADDRESS
4383 016150 132737 000040 001215 62$: BITB #APTCSUP, $ENVM ;; APT CONSOLE SUPPRESSED
4384 016156 001003 BNE 60$ ;; YES, SKIP TYPE OUT
4385 016160 112046 2$: MOVB (RO)+, -(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
4386 016162 001005 BNE 4$ ;; BR IF IT ISN'T THE TERMINATOR
4387 016164 005726 TST (SP)+ ;; IF TERMINATOR POP IT OFF THE STACK
4388 016166 012600 60$: MOV (SP)+, RO ;; RESTORE RO
4389 016170 062716 000002 3$: ADD #2, (SP) ;; ADJUST RETURN PC
4390 016174 000002 RTI ;; RETURN
4391 016176 122716 000011 4$: CMPB #HT, (SP) ;; BRANCH IF <HT>
4392 016202 001430 BEQ 8$
4393 016204 122716 000200 CMPB #CRLF, (SP) ;; BRANCH IF NOT <CRLF>
4394 016210 001006 BNE 5$
4395 016212 005726 TST (SP)+ ;; POP <CR><LF> EQUIV
4396 016214 104401 TYPE ;; TYPE A CR AND LF
4397 016216 001171 $CRLF
4398 016220 105037 016354 CLRB $CHARCNT ;; CLEAR CHARACTER COUNT
4399 016224 000755 BR 2$ ;; GET NEXT CHARACTER
4400 016226 004737 016310 5$: JSR PC, $TYPEC ;; GO TYPE THIS CHARACTER
4401 016232 123726 001156 6$: CMPB $FILLC, (SP)+ ;; IS IT TIME FOR FILLER CHARS.?
4402 016236 001350 BNE 2$ ;; IF NO GO GET NEXT CHAR.
4403 016240 013746 001154 MOV $NULL, -(SP) ;; GET # OF FILLER CHARS. NEEDED
4404 AND THE NULL CHAR.
4405 016244 105366 000001 7$: DECB 1(SP) ;; DOES A NULL NEED TO BE TYPED?
4406 016250 002770 BLT 6$ ;; BR IF NO--GO POP THE NULL OFF OF STACK
4407 016252 004737 016310 JSR PC, $TYPEC ;; GO TYPE A NULL
4408 016256 105337 016354 DECB $CHARCNT ;; DO NOT COUNT AS A COUNT
4409 016262 000770 BR 7$ ;; LOOP
4410
4411 ;HORIZONTAL TAB PROCESSOR
4412
4413 016264 112716 000040 8$: MOVB #' (SP) ;; REPLACE TAB WITH SPACE
4414 016270 004737 016310 9$: JSR PC, $TYPEC ;; TYPE A SPACE

```

```

4415 016274 132737 000007 016354      BITB      #7,$SCHARCNT      ;;BRANCH IF NOT AT
4416 016302 001372                      BNE        9$          ;;TAB STOP
4417 016304 005726                      TST        (SP)+      ;;POP SPACE OFF STACK
4418 016306 000724                      BR         2$          ;;GET NEXT CHARACTER
4419 016310 105777 162634      $TYPEC: TSTB      @STPS      ;;WAIT UNTIL PRINTER IS READY
4420 016314 100375                      BPL        $TYPEC
4421 016316 116677 000002 162626      MOVB      2(SP),@STPB      ;;LOAD CHAR TO BE TYPED INTO DATA REG.
4422 016324 122766 000015 000002      CMPB      #CR,2(SP)      ;;IS CHARACTER A CARRIAGE RETURN?
4423 016332 001003                      BNE        1$          ;;BRANCH IF NO
4424 016334 105037 016354      CLRB      $SCHARCNT      ;;YES--CLEAR CHARACTER COUNT
4425 016340 000406                      BR         $TYPEX      ;;EXIT
4426 016342 122766 000012 000002 1$:  CMPB      #LF,2(SP)      ;;IS CHARACTER A LINE FEED?
4427 016350 001402                      BEQ        $TYPEX      ;;BRANCH IF YES
4428 016352 105227                      INCB      (PC)+      ;;COUNT THE CHARACTER
4429 016354 000000      $SCHARCNT: .WORD 0      ;;CHARACTER COUNT STORAGE
4430 016356 000207      $TYPEX: RTS        PC

4431
4432      .SBTTL  APT COMMUNICATIONS ROUTINE
4433
4434      ;*****
4435 016360 112737 000001 016624  $ATY1:  MOVB      #1,$FFLG      ;;TO REPORT FATAL ERROR
4436 016366 112737 000001 016622  $ATY3:  MOVB      #1,$MFLG      ;;TO TYPE A MESSAGE
4437 016374 000403                      BR         $ATYC
4438 016376 112737 000001 016624  $ATY4:  MOVB      #1,$FFLG      ;;TO ONLY REPORT FATAL ERROR
4439 016404                      $ATYC:
4440 016404 010046                      MOV        RO,-(SP)      ;;PUSH RO ON STACK
4441 016406 010146                      MOV        R1,-(SP)      ;;PUSH R1 ON STACK
4442 016410 105737 016622                      TSTB      $MFLG      ;;SHOULD TYPE A MESSAGE?
4443 016414 001450                      BEQ        5$          ;;IF NOT: BR
4444 016416 122737 000001 001214      CMPB      #APTENV,$ENV      ;;OPERATING UNDER APT?
4445 016424 001031                      BNE        3$          ;;IF NOT: BR
4446 016426 132737 000100 001215      BITB      #APTSPool,$ENVM      ;;SHOULD SPOOL MESSAGES?
4447 016434 001425                      BEQ        3$          ;;IF NOT: BR
4448 016436 017600 000004                      MOV        @4(SP),RO      ;;GET MESSAGE ADDR.
4449 016442 062766 000002 000004      ADD        #2,4(SP)      ;;BUMP RETURN ADDR.
4450 016450 005737 001174      1$:  TST        $MSGTYPE      ;;SEE IF DONE W/ LAST XMISSION?
4451 016454 001375                      BNE        1$          ;;IF NOT: WAIT
4452 016456 010037 001210                      MOV        RO,$MSGAD      ;;PUT ADDR IN MAILBOX
4453 016462 105720      2$:  TSTB      (RO)+      ;;FIND END OF MESSAGE
4454 016464 001376                      BNE        2$
4455 016466 163700 001210                      SUB        $MSGAD,RO      ;;SUB START OF MESSAGE
4456 016472 006200                      ASR        RO          ;;GET MESSAGE LGTH IN WORDS
4457 016474 010037 001212                      MOV        RO,$MSGGLT      ;;PUT LENGTH IN MAILBOX
4458 016500 012737 000004 001174      MOV        #4,$MSGTYPE      ;;TELL APT TO TAKE MSG.
4459 016506 000413                      BR         5$
4460 016510 017637 000004 016534 3$:  MOV        @4(SP),4$      ;;PUT MSG ADDR IN JSR LINKAGE
4461 016516 062766 000002 000004      ADD        #2,4(SP)      ;;BUMP RETURN ADDRESS
4462 016524 013745 177776                      MOV        177776,-(SP)      ;;PUSH 177776 ON STACK
4463 016530 004737 016076                      JSR        PC,$TYPE      ;;CALL TYPE MACRO
4464 016534 000000      4$:  .WORD 0
4465 016536      5$:
4466 016536 105737 016624      10$: TSTB      $FFLG      ;;SHOULD REPORT FATAL ERROR?
4467 016542 001416                      BEQ        12$      ;;IF NOT: BR
4468 016544 005737 001214                      TST        $ENV      ;;RUNNING UNDER APT?

```

```

4469 016550 001413      BEQ      12S
4470 016552 005737 001174 11S:   TST      $MSGTYPE
4471 016556 001375      BNE      11S
4472 016560 017637 000004 001176 MOV      24(SP), $FATAL
4473 016566 062766 000002 000004 ADD      2, 4(SP)
4474 016574 005237 001174      INC      $MSGTYPE
4475 016600 105037 016624 12S:   CLRB     $FFL
4476 016604 105037 016623      CLRB     $LFLG
4477 016610 105037 016622      CLRB     $MFLG
4478 016614 012601      MOV      (SP)+, R1
4479 016616 012600      MOV      (SP)+, R0
4480 016620 000207      RTS      PC
4481 016622      000      $MFLG: .BYTE 0
4482 016623      000      $LFLG: .BYTE 0
4483 016624      000      $FFLG: .BYTE 0
4484      016626      .EVEN
4485      000200      APTSIZE=200
4486      000001      APTENV=001
4487      000100      APTSPool=100
4488      000040      APTCSUP=040
4489      .SBTTL     POWER DOWN AND UP ROUTINES
4490
4491      ;*****
4492      ;POWER DOWN ROUTINE
4493 016626 012737 016766 000024 $PWRDN: MOV      $SILLUP, 2*$PWRVEC ;SET FOR FAST UP
4494 016634 012737 000340 000026      MOV      340, 2*$PWRVEC+2 ;PRIO:7
4495 016642 010046      MOV      R0, -(SP) ;PUSH R0 ON STACK
4496 016644 010146      MOV      R1, -(SP) ;PUSH R1 ON STACK
4497 016646 010246      MOV      R2, -(SP) ;PUSH R2 ON STACK
4498 016650 010346      MOV      R3, -(SP) ;PUSH R3 ON STACK
4499 016652 010446      MOV      R4, -(SP) ;PUSH R4 ON STACK
4500 016654 010546      MOV      R5, -(SP) ;PUSH R5 ON STACK
4501 016656 017746 162256      MOV      2$SWR, -(SP) ;PUSH 2$SWR ON STACK
4502 016662 010637 016772      MOV      SP, $SAVR6 ;SAVE SP
4503 016666 012737 016700 000024      MOV      $PWRUP, 2*$PWRVEC ;SET UP VECTOR
4504 016674 000000      HALT
4505 016676 000776      BR      .-2 ;;HANG UP
4506
4507      ;*****
4508      ;POWER UP ROUTINE
4509 016700 012737 016766 000024 $PWRUP: MOV      $SILLUP, 2*$PWRVEC ;SET FOR FAST DOWN
4510 016706 013706 016772      MOV      $SAVR6, SP ;GET SP
4511 016712 005037 016772      CLR      $SAVR6 ;WAIT LOOP FOR THE TTY
4512 016716 005237 016772 1S:   INC      $SAVR6 ;WAIT FOR THE INC
4513 016722 001375      BNE      1S ;OF WORD
4514 016724 012677 162210      MOV      (SP)+, 2$SWR ;POP STACK INTO 2$SWR
4515 016730 012605      MOV      (SP)+, R5 ;POP STACK INTO R5
4516 016732 012604      MOV      (SP)+, R4 ;POP STACK INTO R4
4517 016734 012603      MOV      (SP)+, R3 ;POP STACK INTO R3
4518 016736 012602      MOV      (SP)+, R2 ;POP STACK INTO R2
4519 016740 012601      MOV      (SP)+, R1 ;POP STACK INTO R1
4520 016742 012600      MOV      (SP)+, R0 ;POP STACK INTO R0
4521 016744 012737 016626 000024      MOV      $PWRDN, 2*$PWRVEC ;SET UP THE POWER DOWN VECTOR
4522 016752 012737 000340 000026      MOV      340, 2*$PWRVEC+2 ;PRIO:7

```

[illegible]

```

; VECTORED TO IS WITHIN THE RANGE OF
; VECTORS ASSIGNED TO THE DEVICE,
; THEN I'D SUSPECT THE DEVICE
; INTERRUPTD ILLEGALLY.
; IF THE ADDRESS THE INTERRUPT
; VECTORED TO IS OUTSIDE OF THE
; RANGE ASSIGNED TO THE DEVICE
; I'D SUSPECT THAT THE
; DEVICE PUT THE WRONG INTERRUPT
; VECTOR ON THE BUS DURING THE INTERRUPT
; PROCESS.

```

```

; PROCESS NOTE:
; FOR THIS ERROR - DON'T USE
; "LOOP ON ERROR" OPTION.
; ALSO EXPECT THAT THE INTERRUPT TEST TO
; WILL REPORT THAT THE DEVICE DIDN'T
; INTERRUPT.
; FOLLOW THE RECOMMENDED PROCEDURE
; IN THE DOCUMENT (ON THIS DIAGNOSTIC)
; FOR LOOPING ON TEST.

```

```
;:SSSSSSSSSSSSSSSSSSSSSSSSS)) ERROR ((SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
```

```

RTI
TRTO:  .WORD 0
TRFR0: .WORD 0
.SBTL  TRAP DECODER

```

```

;CONTAINS ADDR. WE TRAPPED OR INTERRUPTED TO.
;CONTAINS ADDR. WE TRAPPED OR INTR. FROM.

```

```

;*****
;THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
;AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
;OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
;GO TO THAT ROUTINE.

```

```

$TRAP:  MOV      RO,-(SP)          ;;SAVE RO
        MOV      2(SP),RO         ;;GET TRAP ADDRESS
        TST      -(RO)            ;;BACKUP BY 2
        MOVB     (RO),RO          ;;GET RIGHT BYTE OF TRAP
        ASL      RO               ;;POSITION FOR INDEXING
        MOV      $TRPAD(RO),RO    ;;INDEX TO TABLE
        RTS      RO               ;;GO TO ROUTINE

```

```
;; THIS IS USE TO HANDLE THE "GETPRI" MACRO
```

```
STRAP2: MOV      (SP), -(SP)      ;; MOVE THE PC DOWN
        MOV      4(SP), 2(SP)    ;; MOVE THE PSW DOWN
        RTI                      ;; RESTORE THE PSW
```

```
.SBTTL  TRAP TABLE
```

```

; *THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
; *BY THE "TRAP" INSTRUCTION:

```

					ROUTINE

4631					
4632					
4633					
4634	017110	017076			\$TRPAD: .WORD \$TRAP2
4635	017112	016076			;;CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE
4636	017114	014220			;;CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
4637	017116	014174			;;CALL=TYPOS TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
4638	017120	014234			;;CALL=TYPON TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
4639	017122	014422			;;CALL=TYPNB TRAP+5(104405) TYPE BINARY (ASCII) NUMBER
4640					
4641	017124	015372			\$GTSWR ;;CALL=GTSWR TRAP+6(104406) GET SOFT-SWR SETTING
4642					
4643	017126	015322			\$CKSWR ;;CALL=CKSWR TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR
4644	017130	015604			;;CALL=RDCHR TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
4645	017132	015724			;;CALL=RDLIN TRAP+11(104411) TTY TYPEIN STRING ROUTINE
4646	017134	005015	046103	041517	EM1: .ASCIZ <15><12>/CLOCK SR FUNCTION ERROR/
4647	017142	020113	051123	043040	
4648	017150	047125	052103	047511	
4649	017156	020116	051105	047522	
4650	017164	000122			
4651	017166	005015	046103	041517	EM2: .ASCIZ <15><12>/CLOCK SR DATA ERROR/
4652	017174	020113	051123	042040	
4653	017202	052101	020101	051105	
4654	017210	047522	000122		
4655	017214	005015	046103	041517	EM3: .ASCIZ <15><12>/CLOCK BR DATA ERROR/
4656	017222	020113	051102	042040	
4657	017230	052101	020101	051105	
4658	017236	047522	000122		
4659	017242	044600	052116	051105	EM4: .ASCIZ <200>/INTERRUPT ERROR/
4660	017250	052522	052120	042440	
4661	017256	051122	051117	000	
4662	017263	015	041412	052517	EM5: .ASCIZ <15><12>/COUNT REG. ERROR/
4663	017270	052116	051040	043505	
4664	017276	020056	051105	047522	
4665	017304	000122			
4666	017306	005015	047503	047125	EM11: .ASCIZ <15><12>#COUNT ERROR #
4667	017314	020124	051105	047522	
4668	017322	020122	000		
4669	017325	015	041412	052517	EM12: .ASCIZ <15><12>#COUNT FUNCTION ERROR#
4670	017332	052116	043040	047125	
4671	017340	052103	047511	020116	
4672	017346	051105	047522	000122	
4673	017354	005015	046103	041517	EM16: .ASCIZ <15><12>#CLOCK INTERRUPT ERROR #
4674	017362	020113	047111	042524	
4675	017370	051122	050125	020124	
4676	017376	051105	047522	020122	
4677	017404	000			
4678	017405	015	051012	050105	EM20: .ASCIZ <15><12>#REPEATABILITY ERROR #
4679	017412	040505	040524	044502	
4680	017420	044514	054524	042440	
4681	017426	051122	051117	000040	
4682	017434	005015	042101	051104	EM26: .ASCIZ <15><12>#ADDRESSING ERROR#
4683	017442	051505	044523	043516	
4684	017450	042440	051122	051117	

4685	017456	000							
4686									
4687	017457	015	042412	051122	DH1:	.ASCIZ	<15><12>#ERRPC	ASR	WAS S/B#
4688	017464	041520	040411	051123					
4689	017472	053411	051501	051411					
4690	017500	041057	000						
4691	017503	015	042412	051122	DH3:	.ASCIZ	<15><12>#ERRPC	ABR	WAS S/B#
4692	017510	041520	040411	051102					
4693	017516	053411	051501	051411					
4694	017524	041057	000						
4695	017527	200	051105	050122	DH4A:	.ASCIZ	<200>#ERRPC	TO	FROM ADDR. #
4696	017534	020103	020040	047524					
4697	017542	020040	020040	020040					
4698	017550	051106	046517	040440					
4699	017556	042104	027122	000					
4700	017563	015	042412	051122	DH12:	.ASCIZ	<15><12>#ERRPC	ASR	#
4701	017570	041520	040411	051123					
4702	017576	000011							
4703	017600	005015	051105	050122	DH20:	.ASCIZ	<15><12>#ERRPC	ASR	2NDCNT 1STNCT 3RDCNT#
4704	017606	004503	051501	004522					
4705	017614	047062	041504	052116					
4706	017622	030411	052123	041516					
4707	017630	004524	051063	041504					
4708	017636	052116	000						
4709	017641	015	042412	051122	DH26:	.ASCIZ	<15><12>#ERRPC	CLOCK ADDR. #	
4710	017646	041520	041411	047514					
4711	017654	045503	040440	042104					
4712	017662	027122	000						
4713									
4714		017666					.EVEN		
4715									
4716	017666	001116	001376	001126	DT1:	.WORD	\$ERRPC, ASR, \$BDDAT, \$GDDAT, 0		
4717	017674	001124	000000						
4718	017700	001116	001400	001126	DT3:	.WORD	\$ERRPC, ABR, \$BDDAT, \$GDDAT, 0		
4719	017706	001124	000000						
4720	017712	001116	017050	017052	DT4:	.WORD	\$ERRPC, TRTO, TRFRO, 0		
4721	017720	000000							
4722	017722	001116	001376	000000	DT12:	.WORD	\$ERRPC, ASR, 0		
4723	017730	001116	001376	001126	DT20:	.WORD	\$ERRPC, ASR, \$BDDAT, \$GDDAT, \$TMPO, 0		
4724	017736	001124	001420	000000					
4725	017744	001116	001376	001126	DT22:	.WORD	\$ERRPC, ASR, \$BDDAT, \$TMPO, 0		
4726	017752	001420	000000						
4727	017756	001116	001420	000000	DT26:	.WORD	\$ERRPC, \$TMPO, 0		
4728									
4729	017764	000000	000000		DFO:	.WORD	0, 0		
4730									
4731		000001					.END		

833#	920	961	1063	1064								
1064#	1196*	1197*	1248	1255	1732*	1734*	1735	1750*	1752	1771*	1773*	1774
1789*	1791	1900*	1917	1945*	1962	2041*	2045	2130*	2143	2171*	2202	2219*
2238*	2277*	2302*	2336*	2352	2387*	2403	2438*	2454	2489*	2505	2540*	2556
2595*	2611	2640*	2659	2685*	2723*	2827*	2884*	2901	2922*	2942	2967*	2989
3025	3051*	3073	3109	3135*	3157	3193	3220*	3242	3278	3304*	3329	3368
3410*	3420	3443	3473*	3483	3506	3546*	3563	3594	3619*	3636	3849*	3854

3890*	4718		
920	963		
920			
920	935		
920			
920			
920			
920			
920			
920			
920			
920			
920			
920			
920			
920			
920			
920			
920			
920			
920			
920	926		
920	962		
920	931		
920	932		
920	923		
920	948		
920	952		
920	955		
920	958		
920	942		
920	950		
920	953		
920	956		
920	928		
920	929		
920	922		
920	943		
920	951		
920	954		
920	957		
3552	3584	3625	3807*
920	925		
835*	920	1069	
4383	4488*		
4076	4376	4444	4486*
1133	4485*		
4378	4446	4487*	

ASR 001376

1063*	1147*	1196	1214	1221	1287*	1288*	1290	1303*	1305	1332*	1333*	1335
1348*	1350	1377*	1378*	1380	1393*	1395	1422*	1423*	1425	1438*	1440	1467*
1468*	1470	1483*	1485	1512*	1513*	1515	1528*	1530	1557*	1558*	1560	1573*
1575	1602*	1603*	1605	1618*	1620	1647*	1648*	1650	1663*	1665	1692*	1693*
1695	1708*	1710	1823*	1824*	1829	1862*	1864*	1868*	1874*	1876	1899*	1904*
1908	1912*	1916*	1919*	1944*	1949*	1953	1957*	1961*	1964*	1995*	1999	2068*
2069*	2071	2103*	2104*	2105	2118*	2129*	2131*	2132*	2134	2138*	2142*	2145*
2170*	2175*	2177*	2193	2197*	2201*	2204*	2218*	2237*	2240*	2242*	2244	2275*
2279*	2281*	2285	2301*	2303*	2305*	2307	2318*	2335*	2337*	2343	2347*	2351*
2354*	2370*	2386*	2388*	2394	2398*	2402*	2405*	2421*	2437*	2439*	2445	2449*
2453*	2456*	2472*	2488*	2490*	2496	2500*	2504*	2507*	2523*	2539*	2541*	2547
2551*	2555*	2558*	2574*	2594*	2596*	2602	2606*	2610*	2613*	2629*	2639*	2642*
2644*	2650	2654*	2658*	2661*	2664	2676*	2684*	2686*	2691	2722*	2725*	2726*
2750*	2763*	2765*	2766*	2789*	2800*	2801*	2802*	2803*	2805	2808	2826*	2828*
2834	2837	2853*	2854*	2855*	2857*	2859*	2862	2875*	2883*	2887*	2888*	2892
2896*	2900*	2903*	2914*	2921*	2923*	2924*	2927*	2933	2937*	2941*	2944*	2956*
2966*	2968*	2969*	2974*	2980	2984*	2988*	2991*	3011*	3016	3020*	3024*	3027*
3041*	3050*	3052*	3053*	3058*	3064	3068*	3072*	3075*	3095*	3100	3104*	3108*
3111*	3125*	3134*	3136*	3137*	3142*	3148	3152*	3156*	3159*	3179*	3184	3188*
3192*	3195*	3209*	3219*	3221*	3222*	3227*	3233	3237*	3241*	3244*	3264*	3269
3273*	3277*	3280*	3294*	3303*	3305*	3306*	3312*	3320	3324*	3328*	3331*	3352*
3359	3363*	3367*	3370*	3384*	3409*	3411*	3414*	3418*	3434	3438*	3442*	3445*
3472*	3474*	3477*	3481*	3497	3501*	3505*	3508*	3522*	3523	3545*	3547*	3554
3558*	3562*	3565*	3585	3589*	3593*	3596*	3618*	3620*	3627	3631*	3635*	3638*
3653	3682*	3686	3732*	3848*	3850*	3851*	3852*	3853*	3889*	3891*	3892*	3895
3927*	3928*	3929	3933	4716	4722	4723	4725					

ASWREG= 000000
 ATESTN= 000000
 AUNIT = 000000
 AUSWR = 000000
 AVECT1= 000440
 AVECT2= 000000
 BIT0 = 000001

920	933											
920	924											
920	927											
920	934											
834*	920	959	1065	1066	1067	1068						
920	960											
817*	1693	1694	1708	1904	1911	1949	1956	2131	2137	2175	2196	2240
2279	2285	2307	2337	2346	2388	2397	2439	2448	2490	2499	2541	2550
2596	2605	2642	2653	2765	2809	2853	2857	2859	2888	2895	2924	2936
2983	3019	3067	3103	3151	3187	3236	3272	3323	3362	3437	3500	3557
3588	3630											

BIT00 = 000001
 BIT01 = 000002
 BIT02 = 000004
 BIT03 = 000010
 BIT04 = 000020
 BIT05 = 000040
 BIT06 = 000100
 BIT07 = 000200
 BIT08 = 000400
 BIT09 = 001000
 BIT1 = 000002
 BIT10 = 002000

807*	817	2105										
806*	816											
805*	815											
804*	814											
803*	813											
802*	812											
801*	811											
800*	810											
799*	809	4180										
798*	808	4087	4188									
816*	1648	1649	1663	2181								
797*	2017	2061	2927	2974	3011	3058	3095	3142	3179	3227	3264	3312
3352	3414	3477	4064									
796*	1378	1379	1393	1911	1956	2017	2061	2137	2196	2346	2397	2448
2499	2550	2605	2653	2887	2895	2923	2936	2968	2983	3019	3052	3067
3103	3136	3151	3187	3221	3236	3272	3305	3323	3362	3437	3500	3557
3588	3630	4195										

BIT11 = 004000

[illegible]

MAINDEC-11-DVKWA-B MACY11 27(665) 21-FEB-77 14:43 PAGE 98
 DVKWAB.P11 CROSS REFERENCE TABLE

ERRVEC= 000004	820*	1118	1119*	1130*	1145*	1211	1212*	1236*	1245	1246*	1270*	3683	3684*
EXS = 001440	3710*	3716*	4171	4172*	4174*	4177*							
GNS = ***** U	1080*	1084*	1089*	1093*	1169	2015	2059	2084	2179	2229	2253	3540	3615
	1166	1173	1178	3550	3582	3623	3695	3701	3706	3723	3729	3763	3769
GTSWR = 104406	3775	3781	3785	3810	4635	4636	4637	4638	4639	4641	4643	4644	4645
HT = 000011	1161	4641*											
IOTRD = 017004	730*	4391	4432										
IOTST1 = 014006	698	1145	4552*										
IOTST2 = 014064	710	3847*	3855	3857									
IOTST3 = 014132	711	3888*	3896	3898									
IOTVEC= 000020	712	3926*	3934	3936									
LCNT = 001442	825*	1103*	1104*										
LF = 000012	1081*	3734*	3735*	3777									
LOOP = 002274	731*	4426	4432										
MDEVCT = 001434	1185*	3712	3798										
PC = %000007	1078*	1180*	3677	3680*	3738*								
	751*	3552*	3584*	3625*	3755*	3758*	3792*	3797	3816*	4073*	4079*	4139*	4268*
	4381*	4400*	4407*	4414*	4428*	4430*	4463*	4480*					
PIRQ = 177772	737*												
PIRQVE= 000240	831*												
PRIOR = 001412	1069*												
PR0 = 000000	754*												
PR1 = 000040	755*												
PR2 = 000100	756*												
PR3 = 000140	757*												
PR4 = 000200	758*												
PR5 = 000240	759*												
PR6 = 000300	760*												
PR7 = 000340	761*												
PS = 177776	734*	735											
PSW = 177776	735*												
PWRVEC= 000024	826*	1109*	1110*	4493*	4494*	4503*	4509*	4521*	4522*				
RDCHR = 104410	4327	4644*											
RDLIN = 104411	4645*												
RESVEC= 000010	821*												
ROTATE = 001426	1075*	1183*	1184	3679*	3691	3739*	4099						
RSTART = 002104	709	1168*											
RO = %000000	742*	2235*	2258*	2338*	2340*	2389*	2391*	2440*	2442*	2491*	2493*	2542*	2544*
	2597*	2599*	2645*	2647*	2687*	2688*	2829*	2830*	2889*	2890*	2925*	2930*	2972*
	2975*	3008*	3012*	3056*	3059*	3092*	3096*	3140*	3143*	3176*	3180*	3225*	3228*
	3261*	3265*	3310*	3313*	3349*	3353*	3413*	3415*	3476*	3478*	3789*	3792	4111
	4112*	4113*	4120*	4121*	4122*	4123*	4124*	4125	4130	4135*	4137*	4141	4143
	4374	4375*	4380	4385	4388*	4440	4448*	4452	4453	4455*	4456*	4457	4479*
	4495	4520*	4612	4613*	4614	4615*	4616*	4617*	4618*				
R1 = %000001	743*	3308*	3316*	3348*	3355*	4028	4029*	4032*	4038*	4441	4478*	4496	4519*
R2 = %000002	744*	4497	4518*										
R3 = %000003	745*	3975	3984*	3990*	3991*	3994*	3999*	4000*	4001	4010*	4323	4324*	4325
	4328*	4329	4333	4335	4337*	4339*	4498	4517*					
R4 = %000004	746*	3976	3978*	3979*	3980*	3981	3982*	3996	3998*	4006*	4009*	4499	4516*
R5 = %000005	747*	3977	3983*	3985*	3987*	3988*	3989*	3990	4008*	4500	4515*		
R6 = %000006	748*	1097*	1098*	1099									
R7 = %000007	749*												
SF = %000006	750*	1101*	1118*	1126*	1130	1139*	1140*	1211*	1220*	1236	1245*	1254*	1270
	2717*	2718*	2729*	2730*	2736*	2737*	2749*	2758*	2759*	2768*	2769*	2775*	2776*

MAINDEC-11-DVKWA-8 MACY11 27(665) 21-FEB-77 14:43 PAGE 99
 DVKWAB.P11 CROSS REFERENCE TABLE

	2788*	3697*	3708*	3715*	3725*	3765*	3771*	3777*	3787*	3967*	3968	3969	3970*
	3975*	3976*	3977*	3983	4008	4009	4010	4011*	4012*	4028*	4029	4038	4039*
	4040*	4068	4089*	4092*	4111*	4116*	4137	4141*	4171*	4174	4176	4177	4206
	4207	4211*	4228*	4229*	4230	4237*	4240*	4241*	4245*	4246*	4250	4253*	4257
	4259	4261	4262*	4269	4271	4273*	4274	4276*	4277*	4278*	4279*	4280*	4295*
	4296*	4299*	4300*	4301	4305*	4306*	4307	4310	4312	4314*	4323*	4328	4339
	4340*	4341*	4342*	4374*	4375	4385*	4387	4388	4389*	4391	4393	4395	4401
	4403*	4405*	4413*	4417	4421	4422	4426	4440*	4441*	4448	4449*	4460	4461*
	4462*	4472	4473*	4478	4479	4495*	4496*	4497*	4498*	4499*	4500*	4501*	4502
	4510*	4514	4515	4516	4517	4518	4519	4520	4564*	4612*	4613	4623*	4624*
	725*	1101											
	708	1091*											
	736*												
	900*	1099	1120*	1122	1128*	1135*	1159	2589	4064	4071	4083	4087	4166
	4180	4182	4188	4195	4224	4261*	4501	4514*					
	701*	1128	1159	4224	4237								
	789*												
	779*	789											
	778*	788											
	777*	787											
	776*	786											
	775*	785											
	774*	784											
	773*	783											
	772*	782											
	771*	781											
	770*	780											
	788*												
	769*												
	768*												
	767*												
	766*												
	765*												
	764*												
	787*												
	786*												
	785*												
	784*												
	783*												
	782*												
	781*												
	780*												
	822*												
	829*												
	830*												
	828*	1107*	1108*										
	4563*	4603*	4720										
	4552*	4553*	4555	4602*	4720								
	823*												
	1079*	1085*	1088*	1092*	3677								
	717	1084*											
	1203*												
	1509*												
	1554*												

STACK = 001100
 START = 001474
 STKLMT= 177774
 SWR 001140

SWREG 000176
 SW0 = 000001
 SW00 = 000001
 SW01 = 000002
 SW02 = 000004
 SW03 = 000010
 SW04 = 000020
 SW05 = 000040
 SW06 = 000100
 SW07 = 000200
 SW08 = 000400
 SW09 = 001000
 SW1 = 000002
 SW10 = 002000
 SW11 = 004000
 SW12 = 010000
 SW13 = 020000
 SW14 = 040000
 SW15 = 100000
 SW2 = 000004
 SW3 = 000010
 SW4 = 000020
 SW5 = 000040
 SW6 = 000100
 SW7 = 000200
 SW8 = 000400
 SW9 = 001000
 TBITVE= 000014
 TKVEC = 000060
 TPVEC = 000064
 TRAPVE= 000034
 TRFR0 017052
 TRTO 017050
 TRIVEC= 000014
 TSTCNT 001436
 TSTSTR 001444
 TST1 002362
 TST10 003204
 TST11 003302

[illegible]

[illegible]

MAINDEC-11-DVKWA-B MACY11 27(665) 21-FEB-77 14:43 PAGE 102
 DVKWAB.P11 CROSS REFERENCE TABLE

\$ERTTL	001112	887#	4067*	4102										
\$ESCAP	001162	911#	1112*	4090	4092	4102	4208*							
\$ETABL	001214	930#												
\$ETEND	001256	872	964#											
\$FATAL	001176	923#	4472*											
\$FFLG	016624	4435*	4438*	4466	4475*	4483*								
\$FILLC	001156	908#	4401	4432										
\$FILLS	001155	907#	4432											
\$GDADR	001120	891#												
\$GDDAT	001124	893#	1289*	1291	1304*	1334*	1336	1349*	1379*	1381	1394*	1424*	1426	1439*
		1469*	1471	1484*	1514*	1516	1529*	1559*	1561	1574*	1604*	1606	1619*	1649*
		1651	1664*	1694*	1696	1709*	1733*	1734	1737	1751*	1772*	1773	1776	1790*
		1831*	1832*	1877*	1878*	1906*	1922	1951*	1967	1994*	2040*	2172*	2206*	2207
		2219	2220	2667*	2809*	2838*	2864*	2885*	2979*	2993	2995	3029	3063*	3077
		3079	3113	3147*	3161	3163	3197	3232*	3246	3248	3282	3319*	3333	3335
		3372	3419*	3421	3447	3482*	3484	3496*	3525*	3553*	3626*	4716	4718	4723
		3789#												
\$GET42	013650	4236#	4641											
\$GTSWR	015372	675	676											
\$HD =	000001	867#												
\$HIBTS	001000	884#	4199*	4200	4202*	4213								
\$ICNT	001104	4493	4509	4526#										
\$ILLUP	016766	898#	4264	4353										
\$INTAG	001135	888#	4070*	4078	4102	4113								
\$ITEMB	001114	915#	4102	4338	4347	4432								
\$LF	001172	4476*	4482#											
\$LFLG	016623	885#	1114*	1205*	4190*	4206*	4211	4213						
\$LPADR	001106	886#	1115*	1207*	2173*	4089	4190	4207*	4213					
\$LPERR	001110	948#												
\$MADR1	001226	952#												
\$MADR2	001232	955#												
\$MADR3	001236	958#												
\$MADR4	001242	868	872	921#	1132	1157	4076	4205	4376					
\$MAIL	001174	942#												
\$MAMS1	001224	950#												
\$MAMS2	001230	953#												
\$MAMS3	001234	956#												
\$MAMS4	001240	868#												
\$MBADR	001002	4436*	4442	4477*	4481#									
\$MFLG	016622	4239	4351#											
\$MNEW	016065	928#	4452*	4455										
\$MSGAD	001210	929#	4457*											
\$MSGLG	001212	922#	4450	4458*	4470	4474*								
\$MSGTY	001174	4236	4349#											
\$MSWR	016054	943#												
\$MTYP1	001225	951#												
\$MTYP2	001231	954#												
\$MTYP3	001235	57#												
\$MTYP4	001241	4203	4213#											
\$MXCNT	015320	906#	4403	4432										
\$NULL	001154	1200#	1239#	1276#	1278	1321#	1323	1366#	1368	1411#	1413	1456#	1458	1501#
\$NWTST=	000001	1503	1546#	1548	1591#	1593	1636#	1638	1681#	1683	1727#	1766#	1811#	1813
		1850#	1852	1893#	1938#	1984#	1986	2029#	2031	2063#	2098#	2124#	2158#	2160
		2224#	2270#	2296#	2323#	2325	2374#	2376	2425#	2427	2476#	2478	2527#	2529

M08

MAINDEC-11-DVKWA-B MACY11 27(665) 21-FEB-77 14:43 PAGE 103
 DVKWAB.P11 CROSS REFERENCE TABLE

		2578*	2580	2633*	2679*	2710*	2752*	2795*	2820*	2848*	2877*	2916*	2960*	3044*
		3128*	3213*	3297*	3388*	3390	3461*	3463	3534*	3609*				
\$OCNT	014416	3974*	4003*	4016*										
\$OMODE	014420	3969*	3973*	3978	3981*	3992*	4018*							
\$OVER	015304	4167	4183	4191	4201	4210*								
\$PASS	001202	925*	1132*	1182*	3689	3753*	3754*	3765	3799	4197	4214			
\$PASTH	001006	870*												
\$POWER	016774	4524	4529*											
\$PWRDN	016626	1109	4493*	4521										
\$PWRMG	016762	4524*												
\$PWRUP	016700	4503	4509*											
\$QUES	001170	913*	4102	4282	4331	4347	4432							
\$RDCHR	015604	4295*	4644											
\$RDDEC=	***** U	4646												
\$RDLIN	015724	4323*	4645											
\$RDOCT=	***** U	4646												
\$RDSZ =	000010	4316*												
\$RTNAD	013672	3798*												
\$R2A =	***** U	4646												
\$SAVRE=	***** U	4646												
\$SAVR6	016772	4502*	4510	4511*	4512*	4528*								
\$SCOPE	015040	1103	4163*											
\$SETUP=	000117	1084*	1102	1103	1105	1107	1109	1111	1112	1114	1153	1154	3751	4060
		4086	4094	4164	4219	4353								
\$STUP =	177777	1084*												
\$SVLAD	015250	4175	4204*											
\$SVPC =	000244	844*	849											
\$SWR =	167400	657*	675	681	682	683	684	685	686	687	837*	910	911	912
		1111	1112	1114	1115	1204	1243	1285	1330	1375	1420	1465	1510	1555
		1600	1645	1690	1731	1770	1821	1860	1897	1942	1992	2038	2067	2102
		2128	2168	2228	2274	2300	2332	2383	2434	2485	2536	2587	2637	2683
		2714	2756	2799	2824	2852	2881	2920	2964	3048	3132	3217	3301	3407
		3470	3538	3613	3746	3752	3791	3797	3799	4051	4052	4053	4054	4055
		4064	4071	4083	4087	4102	4155	4156	4157	4158	4159	4166	4178	4180
		4181	4184	4185	4186	4193	4194	4195	4207	4210	4213	4525		
\$SWREG	001216	933*	1135											
\$SWRMK=	000000	687	688	4159	4160	4182								
\$TESTN	001200	924*	4205*											
\$TIMES	001160	910*	1111*	1204*	1285*	1330*	1375*	1420*	1465*	1510*	1555*	1600*	1645*	1690*
		1821*	1860*	1897*	1942*	1992*	2038*	2168*	2332*	2383*	2434*	2485*	2536*	2587*
		2637*	2714*	2824*	2881*	2964*	3048*	3132*	3217*	3301*	3407*	3470*	3539*	3613*
		3752*	4193*	4200	4203*	4213								
\$TKB	001146	903*	3807	3815	4217	4228	4245	4299	4305					
\$TKS	001144	902*	3813	4217	4226	4242	4266*	4297	4303					
\$TMP0	001420	1072*	1221*	1255*	2993*	3077*	3161*	3246*	3333*	4723	4725	4727		
\$TMP1	001422	1073*												
\$TMP3	001424	1074*	1909*	1910*	1911*	1912	1954*	1955*	1956*	1957	2135*	2136*	2137*	2138
		2194*	2195*	2196*	2197	2344*	2345*	2346*	2347	2395*	2396*	2397*	2398	2446*
		2447*	2448*	2449	2497*	2498*	2499*	2500	2548*	2549*	2550*	2551	2603*	2604*
		2605*	2606	2651*	2652*	2653*	2654	2893*	2894*	2895*	2896	2934*	2935*	2936*
		2937	2981*	2982*	2983*	2984	3017*	3018*	3019*	3020	3065*	3066*	3067*	3068
		3101*	3102*	3103*	3104	3149*	3150*	3151*	3152	3185*	3186*	3187*	3188	3234*
		3235*	3236*	3237	3270*	3271*	3272*	3273	3321*	3322*	3323*	3324	3360*	3361*
		3362*	3363	3435*	3436*	3437*	3438	3498*	3499*	3500*	3501	3555*	3556*	3557*

.SASTA= ***** U
.SX = 001000

[illegible]

ADTST	1198#	1238													
BUFLO	1725#	1726	1765												
COMMEN	1#	832#													
COUNTM	2320#	2322	2373	2424	2475	2526	2577								
CSRDTA	1273#	1274	1319	1364	1409	1454	1499	1544	1589	1634	1679				
DFC	982#	988	996	1004	1012	1020	1028	1036	1044	1052	1060				
DIVCH	2958#	2959	3043	3127	3212	3296									
ECB	660#	1223	1233	1257	1267	1293	1298	1308	1314	1338	1343	1353	1359	1383	1388
	1398	1404	1428	1433	1443	1449	1473	1478	1488	1494	1518	1523	1533	1539	1563
	1568	1578	1584	1608	1613	1623	1629	1653	1658	1668	1674	1698	1703	1713	1719
	1740	1745	1755	1760	1779	1784	1794	1799	1838	1845	1882	1888	1926	1933	1971
	1978	2003	2010	2020	2025	2049	2055	2074	2080	2088	2093	2108	2115	2148	2153
	2183	2188	2209	2213	2247	2260	2265	2288	2309	2314	2361	2366	2412	2417	2463
	2468	2514	2519	2565	2570	2620	2625	2668	2673	2694	2699	2740	2744	2779	2783
	2810	2815	2839	2844	2867	2872	2906	2911	2948	2953	2998	3004	3031	3038	3082
	3088	3115	3122	3166	3172	3199	3206	3251	3257	3284	3291	3338	3344	3374	3381
	3423	3428	3449	3455	3486	3491	3511	3517	3526	3530	3568	3574	3599	3605	3642
	3647	3655	3660	4565	4598										
ENDCOM	1#	832#													
ENDPAS	3740#	3761													
ERROR	726#	1226	1260	1296	1311	1341	1356	1386	1401	1431	1446	1476	1491	1521	1536
	1566	1581	1611	1626	1656	1671	1701	1716	1743	1758	1782	1797	1841	1885	1929
	1974	2007	2023	2052	2077	2091	2111	2151	2186	2212	2250	2263	2291	2312	2364
	2415	2466	2517	2568	2623	2671	2697	2743	2782	2813	2842	2870	2909	2951	3001
	3034	3085	3118	3169	3202	3254	3287	3341	3377	3426	3452	3489	3514	3529	3571
	3602	3645	3658	3856	3897	3931	3935	4568							
ESCAPE	1#	832#													
GETPRI	1#	832#													
GETSWR	1#	832#	1154#												
INSTR2	4044#	4094													
LOCKM	2709#	2716	2728	2735	2757	2767	2774								
MULT	1#	832#													
NEWTST	1#	832#	1200	1239	1276	1321	1366	1411	1456	1501	1546	1591	1636	1681	1727
	1766	1811	1850	1893	1938	1984	2029	2063	2098	2124	2158	2224	2270	2296	2323
	2374	2425	2476	2527	2578	2633	2679								

[illegible]

.SEOP	1#	655#	3742
.SERR0	1#	655#	4045
.SERRT	1#	655#	4102
.SMULT	1#		
.SPJWE	1#	654#	4489
.SRAND	1#		
.SRDDE	1#		
.SRDOC	1#	653#	
.SREAD	1#	655#	4214
.SR2AZ	1#		
.SSAVE	1#		
.SSB2D	1#		
.SSB2O	1#		
.SSCOP	1#	655#	4149
.SSIZE	1#		
.SSUPR	1#		
.STRAP	1#	653#	4604
.STYPB	1#	653#	4019
.STYPD	1#	655#	
.STYPE	1#	655#	4353
.STYPO	1#	654#	3942
.S4OCA	1#		
.1170	1#		

MAINDEC-11-DVKWA-B MACY11 27(665) 21-FEB-77 14:43 PAGE 108
 DVKWAB.P11 CROSS REFERENCE TABLE

ADCB	4034														
ADD	1189	1191	1193	1197	1220	1254	2749	2788	3682	3711	3715	3970	3980	4124	4253
	4262	4389	4449	4461	4473	4564									
ASL	3679	4121	4122	4123	4276	4277	4278	4616							
ASR	4456														
BEQ	1134	1158	1170	1292	1306	1337	1351	1382	1396	1427	1441	1472	1486	1517	1531
	1562	1576	1607	1621	1652	1666	1697	1711	1738	1753	1777	1792	1836	1880	1924
	1969	2001	2016	2019	2047	2060	2085	2180	2208	2230	2254	2286	2592	2866	2905
	2997	3030	3081	3114	3165	3198	3250	3283	3337	3373	3422	3448	3485	3510	3524
	3616	3641	3678	3718	3790	3855	3934	3997	4033	4062	4065	4088	4091	4126	4131
	4144	4181	4183	4185	4189	4198	4233	4260	4275	4379	4392	4427	4443	4447	4467
	4469														
BGE	4201														
BGT	3757	4004	4272	4313											
BHI	4187														
BIC	1187	1303	1348	1393	1438	1483	1528	1573	1618	1663	1708	1750	1789	1910	1955
	2136	2195	2345	2396	2447	2498	2549	2604	2652	2857	2894	2935	2982	3018	3066
	3102	3150	3186	3235	3271	3322	3361	3436	3499	3556	3587	3629	3754	3994	4099
	4229	4246	4273	4300	4306	4314									
BIS	1288	1333	1378	1423	1468	1513	1558	1603	1649	1693	1904	1911	1916	1949	1956
	1961	2017	2061	2069	2104	2131	2132	2137	2142	2175	2177	2196	2201	2234	2240
	2242	2279	2281	2303	2305	2346	2351	2397	2402	2448	2453	2499	2504	2550	2555
	2605	2610	2642	2644	2653	2658	2686	2726	2766	2802	2803	2828	2854	2855	2859
	2888	2895	2900	2923	2924	2927	2936	2941	2968	2969	2974	2983	2988	3011	3019
	3024	3052	3053	3058	3067	3072	3095	3103	3108	3136	3137	3142	3151	3156	3179
	3187	3192	3221	3222	3227	3236	3241	3264	3272	3277	3305	3306	3312	3323	3328
	3352	3362	3367	3414	3418	3437	3442	3477	3481	3500	3505	3557	3562	3588	3593
	3630	3635	3691	3851	3853	3892	3928	3999	4000	4280					
BISB	4113														
BIT	2018	2086	2105	2181	2285	2307	2589	2805	2834	2865	3933	4064	4071	4087	4166
	4180	4188	4195												
BITB	1133	4378	4383	4415	4446										
BLE	4556														
BLOS	4326														
BLT	4005	4270	4311	4406											
BMI	2072	2245	2257	2359	2410	2461	2512	2563	2618	2692	3654	3674	3896	3930	
BNE	1100	1123	1152	1156	1160	2087	2106	2147	2182	2221	2259	2308	2341	2356	2392
	2407	2443	2458	2494	2509	2545	2560	2600	2615	2648	2663	2689	2806	2831	2835
	2891	2931	2946	2976	3013	3060	3097	3144	3181	3229	3266	3314	3317	3354	3356
	3416	3479	3541	3567	3598	3690	3995	4072	4077	4096	4114	4136	4167	4196	4225
	4231	4251	4258	4265	4302	4308	4330	4336	4377	4384	4386	4394	4402	4416	4423
	4445	4451	4454	4471	4513										
BPL	2665	3814	3993	4084	4227	4243	4298	4304	4371	4420					
BR	1086	1090	1125	1162	1165	1172	1177	1218	1252	1301	1346	1391	1436	1481	1526
	1571	1616	1661	1706	1748	1787	2013	2083	2191	2216	2252	2747	2786	3007	3091
	3175	3260	3347	3431	3494	3520	3549	3577	3581	3622	3650	3694	3700	3705	3722
	3728	3762	3768	3774	3780	3784	3809	3857	3898	3936	3971	3986	4007	4037	4082
	4119	4146	4169	4175	4178	4191	4194	4254	4281	4283	4309	4332	4373	4399	4409
	4418	4425	4437	4459	4505	4527	4561								
CLC	4036														
CLR	1089	1093	1098	1111	1112	1132	1144	1180	1181	1182	1287	1304	1332	1349	1377
	1394	1422	1439	1467	1484	1512	1529	1557	1574	1602	1619	1647	1664	1692	1709
	1732	1751	1771	1790	1823	1862	1899	1944	1994	2040	2118	2129	2130	2170	2171
	2218	2232	2237	2275	2301	2318	2335	2336	2338	2370	2386	2387	2389	2421	2437

	2438	2440	2472	2488	2489	2491	2523	2539	2540	2542	2574	2594	2595	2597	2629
	2639	2640	2645	2667	2676	2684	2687	2722	2750	2763	2789	2800	2826	2829	2875
	2883	2884	2885	2889	2914	2921	2922	2956	2966	2967	3041	3050	3051	3125	3134
	3135	3209	3219	3220	3294	3303	3304	3384	3409	3410	3472	3473	3496	3522	3525
	3545	3618	3626	3738	3751	3752	3848	3849	3889	3984	4112	4193	4208	4240	4241
	4511														
CLRB	1832	1878	4192	4337	4398	4424	4475	4476	4477						
CMP	1099	1122	1159	1291	1336	1381	1426	1471	1516	1561	1606	1651	1696	1737	1776
	1922	1967	2207	2995	3029	3079	3113	3163	3197	3248	3282	3335	3372	3421	3447
	3484	3677	3717	3854	4176	4200	4224	4230	4250	4257	4269	4271	4301	4307	4310
	4312	4325	4555												
CMPB	1157	4076	4182	4186	4232	4264	4329	4335	4376	4391	4393	4401	4422	4426	4444
DEC	1874	2258	3012	3096	3180	3265	3316	3353	3355	3755	4097	4120			
DECB	3992	4003	4405	4408											
EMT	726														
HALT	4085	4372	4504	4526	4558										
INC	1084	1151	1864	2206	2340	2391	2442	2493	2544	2599	2647	2801	2930	2975	3059
	3143	3228	3313	3415	3478	3680	3703	3735	3753	3998	4006	4067	4095	4199	4279
	4474	4512													
INCB	2688	2830	2890	4061	4204	4428									
IOT	727														
JMP	708	709	710	711	712	715	717	3542	3712	3797					
JSR	3552	3584	3625	3792	4073	4079	4268	4381	4400	4407	4414	4463			
MCV	1085	1088	1092	1097	1101	1103	1104	1105	1106	1107	1108	1109	1110	1114	1115
	1118	1119	1120	1121	1126	1128	1129	1130	1135	1139	1140	1145	1146	1147	1183
	1184	1188	1190	1192	1196	1204	1205	1207	1211	1212	1221	1236	1245	1246	1255
	1270	1285	1289	1290	1305	1330	1334	1335	1350	1375	1379	1380	1395	1420	1424
	1425	1440	1465	1469	1470	1485	1510	1514	1515	1530	1555	1559	1560	1575	1600
	1604	1605	1620	1645	1649	1650	1665	1690	1694	1695	1710	1733	1734	1735	1752
	1772	1773	1774	1791	1821	1829	1831	1860	1876	1877	1897	1900	1906	1908	1909
	1912	1917	1919	1942	1945	1951	1953	1954	1957	1962	1964	1992	1995	1999	2038
	2041	2045	2068	2103	2134	2135	2138	2143	2145	2168	2172	2173	2193	2194	2197
	2202	2204	2219	2235	2238	2277	2302	2332	2337	2343	2344	2347	2352	2354	2383
	2388	2394	2395	2398	2403	2405	2434	2439	2445	2446	2449	2454	2456	2485	2490
	2496	2497	2500	2505	2507	2536	2541	2547	2549	2551	2556	2558	2587	2596	2602
	2603	2606	2611	2613	2637	2650	2651	2654	2659	2661	2685	2714	2717	2718	2723
	2725	2727	2729	2730	2736	2737	2758	2759	2764	2765	2768	2769	2775	2776	2808
	2809	2824	2827	2837	2838	2853	2862	2864	2881	2887	2892	2893	2896	2901	2903
	2925	2933	2934	2937	2942	2944	2964	2972	2979	2980	2981	2984	2989	2991	2993
	3008	3016	3017	3020	3025	3027	3048	3056	3063	3064	3065	3068	3073	3075	3077
	3092	3100	3101	3104	3109	3111	3132	3140	3147	3148	3149	3152	3157	3159	3161
	3176	3184	3185	3188	3193	3195	3217	3225	3232	3233	3234	3237	3242	3244	3246
	3261	3269	3270	3273	3278	3280	3301	3308	3310	3319	3320	3321	3324	3329	3331
	3333	3348	3349	3359	3360	3363	3368	3370	3407	3411	3413	3419	3420	3434	3435
	3438	3443	3445	3470	3474	3476	3482	3483	3497	3498	3501	3506	3508	3523	3539
	3546	3547	3553	3554	3555	3558	3563	3565	3585	3586	3589	3594	3596	3613	3619
	3620	3627	3628	3631	3636	3638	3683	3684	3697	3708	3710	3716	3725	3732	3733
	3734	3736	3739	3758	3765	3771	3777	3787	3789	3850	3852	3890	3891	3927	3967
	3975	3976	3977	3983	3990	4008	4009	4010	4011	4012	4028	4029	4038	4039	4040
	4063	4068	4089	4092	4111	4116	4125	4130	4135	4137	4141	4171	4172	4174	4177
	4190	4202	4203	4206	4207	4210	4211	4237	4261	4266	4295	4296	4323	4324	4339
	4340	4341	4342	4374	4375	4380	4388	4403	4440	4441	4448	4452	4457	4458	4460
	4462	4472	4478	4479	4493	4494	4495	4496	4497	4498	4499	4500	4501	4502	4503
	4509	4510	4514	4515	4516	4517	4518	4519	4520	4521	4522	4552	4563	4612	4613

MOV8	4617 1113 4070 4438	4623 1163 4078 4615	4624 1206 4205	1824 4209	1868 4228	3968 4245	3969 4299	3972 4305	3973 4328	3974 4333	3978 4385	3981 4413	3982 4421	4001 4435	4031 4436
NEG	3979														
NOP	1203	2734	2773	3750	3793	3794	3795	3893	3894						
RES	1997	2043	3791												
ROL	3985	3987	3988	3989	3991	4032									
RTI	1127 4390	1141 4525	2719 4601	2731 4625	2738	2760	2770	2777	4013	4041	4100	4212	4267	4315	4343
RTS	3816	4139	4430	4480	4618										
SEC	4030														
SUB	4069	4455	4553												
TRAP	4627	4636	4637	4638	4639	4641	4643	4644	4645						
TST	1155 2233 3160 3689 4468	1169 2253 3196 3895 4470	1214 2355 3245 3929 4614	1248 2406 3281 3996	1920 2457 3332 4083	1965 2508 3371 4090	2015 2559 3446 4143	2059 2614 3509 4173	2071 2662 3540 4197	2084 2904 3566 4259	2146 2945 3597 4274	2179 2992 3615 4387	2205 3028 3639 4395	2220 3076 3653 4417	2229 3112 3686 4450
TSTB	1834 3815	1879 4184	2244 4226	2255 4242	2357 4297	2408 4303	2459 4370	2510 4419	2561 4442	2616 4453	2664 4466	2691	3673	3807	3813
.ASCII	913	914													
.ASCIIZ	912	915	1167	1174	1179	3551	3583	3624	3696	3702	3707	3724	3730	3764	3770
	3776 4666	3782 4669	3786 4673	3811 4678	4147 4682	4347 4687	4348 4691	4349 4695	4351 4700	4529 4703	4646 4709	4651	4655	4659	4662
.BLKB	4346														
.BYTE	882 951 4481	883 953 4482	888 954 4483	889 956	897 957	898 3799	906 4014	907 4015	908 4016	909 4017	931 4042	932 4080	942 4081	943 4344	950 4345
.DSABL	4284														
.ENABL	1	651	652	4217											
.END	4731														
.ENDC	670 880 962 1159 1243 1373 1466 1598 1691 1851 1943 2039 2127 2258 2334 2435 2536 2637 2796 2880 3008 3132	684 882 963 1165 1277 1374 1502 1599 1728 1852 1985 2061 2128 2271 2343 2436 2537 2638 2797 2881 3014 3133	686 910 966 1167 1278 1375 1503 1600 1729 1858 1986 2064 2159 2272 2375 2445 2538 2680 2798 2882 3045 3139	687 911 1084 1174 1283 1376 1508 1601 1730 1859 1990 2065 2160 2273 2376 2477 2547 2681 2799 2917 3046 3146	688 912 1101 1179 1284 1412 1509 1637 1731 1860 1991 2066 2166 2274 2381 2478 2579 2682 2821 2918 3047 3166	726 913 1102 1201 1285 1413 1510 1638 1767 1861 1992 2067 2167 2297 2382 2483 2580 2683 2822 2919 3048 3176	818 917 1105 1202 1286 1418 1511 1643 1768 1894 1993 2084 2168 2298 2383 2484 2585 2711 2823 2920 3049 3182	832 920 1107 1203 1286 1419 1547 1644 1769 1895 1993 2086 2169 2299 2384 2485 2586 2712 2824 2961 3055 3214	843 942 1109 1204 1322 1420 1548 1645 1770 1896 1996 2088 2192 2299 2385 2486 2587 2713 2825 2962 3062 3215	847 950 1111 1205 1328 1421 1553 1646 1812 1897 2020 2099 2225 2324 2394 2487 2588 2714 2849 2963 3082 3216	849 953 1112 1206 1329 1457 1554 1682 1813 1898 2030 2099 2226 2325 2394 2496 2593 2715 2850 2964 3092 3217	854 956 1114 1209 1330 1458 1555 1683 1819 1939 2031 2099 2227 2325 2427 2528 2602 2753 2851 2965 3098 3218	856 959 1116 1240 1331 1463 1556 1688 1820 1940 2036 2101 2228 2330 2432 2529 2634 2754 2852 2971 3098 3224	863 960 1137 1241 1367 1464 1592 1689 1821 1941 2037 2125 2253 2332 2433 2534 2635 2755 2878 2978 3129 3231	876 961 1153 1242 1368 1465 1593 1690 1822 1942 2038 2126 2255 2333 2434 2535 2636 2756 2879 2998 3131 3251

	3261	3267	3298	3299	3300	3301	3302	3309	3318	3338	3349	3357	3389	3390	3405
	3406	3407	3408	3462	3463	3468	3469	3470	3471	3495	3521	3525	3535	3536	3537
	3538	3551	3578	3583	3599	3610	3611	3612	3613	3614	3624	3696	3702	3707	3724
	3730	3745	3746	3748	3751	3757	3760	3761	3764	3770	3776	3782	3786	3789	3791
	3797	3799	3800	3811	3945	4022	4048	4051	4061	4068	4073	4074	4075	4083	4094
	4102	4105	4120	4149	4152	4155	4160	4166	4168	4179	4182	4183	4184	4186	4188
	4195	4199	4204	4206	4210	4213	4214	4217	4218	4220	4248	4284	4288	4316	4317
	4324	4326	4329	4331	4347	4353	4356	4385	4435	4436	4439	4466	4481	4492	4501
	4502	4508	4514	4515	4525	4532	4607	4613	4616	4635	4636	4637	4638	4639	4640
	4641	4642	4643	4644	4645	4646									
.EQUIV	726	727	735	780	781	782	783	784	785	786	787	788	789	808	809
	810	811	812	813	814	815	816	817							
.EVEN	920	1167	1174	1179	3551	3583	3624	3696	3702	3707	3724	3730	3764	3770	3776
	3782	3786	3800	3811	4148	4484	4531	4714							
.IF	666	684	685	686	687	688	724	790	818	842	845	847	853	855	862
	875	879	881	910	911	912	916	917	919	942	950	953	956	959	960
	961	962	963	964	966	1084	1096	1101	1103	1105	1107	1109	1111	1112	1114
	1132	1152	1153	1154	1157	1166	1173	1178	1199	1200	1202	1204	1205	1206	1239
	1241	1243	1276	1278	1283	1285	1286	1321	1323	1328	1330	1331	1366	1368	1373
	1375	1376	1411	1413	1418	1420	1421	1456	1458	1463	1465	1466	1501	1503	1508
	1510	1511	1546	1548	1553	1555	1556	1591	1593	1598	1600	1601	1636	1638	1643
	1645	1646	1681	1683	1688	1690	1691	1727	1729	1731	1766	1768	1770	1811	1813
	1819	1821	1822	1850	1852	1858	1860	1861	1893	1895	1897	1898	1938	1940	1942
	1943	1984	1986	1990	1992	1993	2013	2016	2019	2029	2031	2036	2038	2039	2060
	2063	2065	2067	2083	2085	2087	2098	2100	2102	2124	2126	2128	2158	2160	2166
	2168	2169	2191	2224	2226	2228	2252	2254	2257	2270	2272	2274	2296	2298	2300
	2323	2325	2330	2332	2333	2334	2338	2374	2376	2381	2383	2384	2385	2389	2425
	2427	2432	2434	2435	2436	2440	2476	2478	2483	2485	2486	2487	2491	2527	2529
	2534	2536	2537	2538	2542	2578	2580	2585	2587	2588	2589	2592	2597	2633	2635
	2637	2638	2679	2681	2683	2710	2712	2714	2715	2752	2754	2756	2795	2797	2799
	2820	2822	2824	2825	2848	2850	2852	2877	2879	2881	2882	2916	2918	2920	2960
	2962	2964	2965	2971	2978	2997	3008	3014	3044	3046	3048	3049	3055	3062	3081
	3092	3098	3128	3130	3132	3133	3139	3146	3165	3176	3182	3213	3215	3217	3218
	3224	3231	3250	3261	3267	3297	3299	3301	3302	3308	3316	3337	3348	3355	3388
	3390	3405	3407	3408	3461	3463	3468	3470	3471	3494	3520	3524	3534	3536	3538
	3550	3577	3582	3598	3609	3611	3613	3614	3623	3695	3701	3706	3723	3729	3744
	3745	3746	3747	3748	3750	3756	3759	3761	3763	3769	3775	3781	3785	3789	3791
	3797	3799	3800	3810	3944	4021	4047	4050	4061	4064	4071	4073	4074	4076	4083
	4087	4094	4102	4104	4119	4135	4151	4154	4159	4165	4166	4178	4180	4181	4182
	4184	4185	4186	4195	4197	4205	4207	4212	4213	4214	4216	4218	4219	4220	4248
	4287	4288	4316	4324	4325	4329	4330	4346	4347	4353	4355	4376	4434	4436	4439
	4466	4481	4491	4501	4502	4507	4514	4515	4523	4525	4529	4606	4612	4616	4627
	4636	4637	4638	4639	4641	4643	4644	4645	4646						
.IFF	684	686	687	688	724	843	847	849	854	856	863	876	879	882	910
	917	920	1101	1152	1153	1200	1201	1202	1203	1204	1209	1239	1240	1241	1242
	1243	1277	1278	1284	1285	1322	1323	1329	1330	1367	1368	1374	1375	1412	1413
	1419	1420	1457	1458	1464	1465	1502	1503	1509	1510	1547	1548	1554	1555	1592
	1593	1599	1600	1637	1638	1644	1645	1682	1683	1689	1690	1728	1729	1730	1731
	1767	1768	1769	1770	1812	1813	1820	1821	1851	1852	1859	1860	1894	1895	1896
	1897	1939	1940	1941	1942	1985	1986	1991	1992	2014	2017	2020	2030	2031	2037
	2038	2061	2064	2065	2066	2067	2084	2086	2088	2099	2100	2101	2102	2125	2126
	2127	2128	2159	2160	2167	2168	2192	2225	2226	2227	2228	2253	2255	2258	2271
	2272	2273	2274	2297	2298	2299	2300	2324	2325	2331	2332	2338	2375	2376	2382
	2383	2389	2426	2427	2433	2434	2440	2477	2478	2484	2485	2491	2528	2529	2535

	2536	2542	2579	2580	2586	2587	2593	2597	2634	2635	2636	2637	2680	2681	2682
	2683	2711	2712	2713	2714	2753	2754	2755	2756	2796	2797	2798	2799	2821	2822
	2823	2824	2849	2850	2851	2852	2878	2879	2880	2881	2917	2918	2919	2920	2961
	2962	2963	2964	2997	3045	3046	3047	3048	3081	3129	3130	3131	3132	3165	3214
	3215	3216	3217	3250	3298	3299	3300	3301	3338	3389	3390	3406	3407	3462	3463
	3469	3470	3495	3521	3525	3535	3536	3537	3538	3578	3599	3610	3611	3612	3613
	3745	3747	3750	3757	3760	3799	3945	4022	4048	4050	4064	4094	4105	4120	4149
	4152	4179	4182	4183	4186	4213	4214	4217	4220	4288	4290	4295	4316	4317	4326
	4330	4347	4356	4435	4492	4508	4525	4607	4613						
.IFT	1167	1174	1179	3551	3583	3624	3696	3702	3707	3724	3730	3764	3770	3776	3782
.IFTF	3786	3811	4074	4194	4290	4295									
	1167	1174	1179	3551	3583	3624	3696	3702	3707	3724	3730	3764	3770	3776	3782
	3786	3811	4073	4192	4235	4288	4291								
.IIF	665	670	675	676	681	682	683	684	687	688	916	920	1102	1105	1111
	1112	1114	1115	1153	3698	3709	3726	3746	3751	3752	3766	3772	3778	3788	3799
	3800	4051	4052	4053	4054	4055	4060	4086	4094	4102	4117	4142	4155	4156	4157
	4158	4159	4160	4164	4193	4194	4210	4213	4214	4217	4238	4339	4347	4353	4432
	4635	4636	4637	4638	4639	4641	4643	4644	4645						
.IRP	1084	1200	1239	1276	1321	1366	1411	1456	1501	1546	1591	1636	1681	1727	1766
	1811	1850	1893	1938	1984	2029	2063	2098	2124	2158	2224	2270	2296	2323	2374
	2425	2476	2527	2578	2633	2679	2710	2752	2795	2820	2848	2877	2916	2960	3044
	3128	3213	3297	3388	3461	3534	3609	3750	4094	4165	4440	4441	4462	4478	4479
	4495	4501	4514	4515											
.LIST	1	650	687	697	832	910	917	920	1084	1116	1153	1154	1167	1174	1179
	1200	1204	1224	1226	1234	1236	1239	1243	1258	1260	1268	1270	1275	1276	1285
	1287	1294	1296	1299	1301	1309	1311	1315	1317	1320	1321	1330	1332	1339	1341
	1344	1346	1354	1356	1360	1362	1365	1366	1375	1377	1384	1386	1389	1391	1399
	1401	1405	1407	1410	1411	1420	1422	1429	1431	1434	1436	1444	1446	1450	1452
	1455	1456	1465	1467	1474	1476	1479	1481	1489	1491	1495	1497	1500	1501	1510
	1512	1519	1521	1524	1526	1534	1536	1540	1542	1545	1546	1555	1557	1564	1566
	1569	1571	1579	1581	1585	1587	1590	1591	1600	1602	1609	1611	1614	1616	1624
	1626	1630	1632	1635	1636	1645	1647	1654	1656	1659	1661	1669	1671	1675	1677
	1680	1681	1690	1692	1699	1701	1704	1706	1714	1716	1720	1722	1727	1731	1741
	1743	1746	1748	1756	1758	1761	1763	1766	1770	1780	1782	1785	1787	1795	1797
	1800	1802	1811	1821	1839	1841	1846	1848	1850	1860	1883	1885	1889	1891	1893
	1897	1927	1929	1934	1936	1938	1942	1972	1974	1979	1981	1984	1992	2004	2006
	2011	2013	2021	2023	2026	2028	2029	2038	2050	2052	2056	2058	2063	2067	2075
	2077	2081	2083	2089	2091	2094	2096	2098	2102	2109	2111	2116	2118	2124	2128
	2149	2151	2154	2156	2158	2168	2184	2186	2189	2191	2210	2212	2214	2216	2224
	2228	2248	2250	2261	2263	2266	2268	2270	2274	2289	2291	2296	2300	2310	2312
	2315	2317	2323	2332	2362	2364	2367	2369	2374	2383	2413	2415	2418	2420	2425
	2434	2464	2466	2469	2471	2476	2485	2515	2517	2520	2522	2527	2536	2566	2568
	2571	2573	2578	2587	2621	2623	2626	2628	2633	2637	2669	2671	2674	2676	2679
	2683	2695	2697	2700	2702	2710	2714	2741	2743	2745	2747	2752	2756	2780	2782
	2784	2786	2795	2799	2811	2813	2816	2818	2820	2824	2840	2842	2845	2847	2848
	2852	2868	2870	2873	2875	2877	2881	2907	2909	2912	2914	2916	2920	2949	2951
	2954	2956	2960	2964	2999	3001	3005	3007	3032	3034	3039	3041	3044	3048	3083
	3085	3089	3091	3116	3118	3123	3125	3128	3132	3167	3169	3173	3175	3200	3202
	3207	3209	3213	3217	3252	3254	3258	3260	3285	3287	3292	3294	3297	3301	3339
	3341	3345	3347	3375	3377	3382	3384	3388	3407	3424	3426	3429	3431	3450	3452
	3456	3458	3461	3470	3487	3489	3492	3494	3512	3514	3518	3520	3527	3529	3531
	3533	3534	3538	3551	3569	3571	3575	3577	3583	3600	3602	3606	3608	3609	3613
	3624	3643	3645	3648	3650	3656	3658	3661	3663	3696	3702	3707	3724	3730	3751
	3764	3770	3776	3782	3786	3791	3811	4094	4159	4316	4566	4568	4599	4601	4627

.MACRO	4635 1 2028 4044	4636 660 2158 4627	4637 661 2320	4638 662 2322	4639 663 2373	4640 688 2424	4641 873 2475	4642 982 2526	4643 1132 2577	4644 1198 2709	4645 1272 2958	4646 1273 3387	1725 3460	1810 3740	1983 3817
.MCALL	653	654	655	656	832	917	1116	1154							
.MEXIT	965														
.NLIST	1 1200 1287 1344 1401 1455 1512 1569 1626 1680 1743 1800 1897 2011 2077 2149 2228 2315 2434 2571 2683 2784 2852 2954 3085 3207 3341 3456 3533 3624 3764 4635 873 1275 1602	649 1204 1294 1346 1405 1456 1519 1571 1630 1681 1746 1802 1927 2013 2081 2151 2248 2317 2464 2573 2695 2786 2868 2956 3089 3209 3345 3458 3534 3643 3770 4636 966 1287 1635	687 1224 1296 1354 1407 1465 1521 1579 1632 1690 1748 1811 1929 2021 2083 2154 2250 2323 2466 2578 2697 2795 2870 2960 3091 3213 3347 3461 3538 3645 3776 4637 1275 1320 1647	697 1226 1299 1356 1410 1467 1524 1581 1635 1692 1756 1821 1934 2023 2089 2156 2261 2332 2469 2587 2700 2799 2873 2964 3116 3217 3375 3470 3551 3648 3782 4638 1320 1332 1680	832 1234 1301 1360 1411 1474 1526 1585 1636 1699 1758 1839 1936 2026 2091 2158 2263 2362 2471 2621 2702 2811 2875 2999 3001 3118 3252 3377 3487 3569 3650 3786 4639 1365 1365 1692	910 1236 1309 1362 1420 1476 1534 1587 1645 1701 1761 1841 1938 2028 2094 2168 2266 2364 2476 2623 2710 2813 2877 3005 3123 3254 3382 3489 3571 3656 3791 4640 1410 1377 1724	917 1239 1311 1365 1422 1479 1536 1590 1647 1704 1763 1846 1942 2029 2096 2184 2268 2367 2485 2626 2714 2816 2881 3007 3125 3258 3384 3492 3575 3658 3791 4641 1455 1410 1455	920 1243 1315 1366 1429 1481 1540 1591 1654 1706 1766 1848 1972 2038 2098 2186 2270 2369 2515 2628 2741 2818 2907 3032 3128 3260 3388 3494 3577 3663 4094 4642 1500 1422 1455	1084 1258 1317 1375 1431 1489 1542 1600 1656 1714 1770 1850 1974 2050 2102 2189 2274 2374 2517 2633 2743 2820 2909 3034 3167 3287 3407 3512 3600 3696 4316 4643 1545 1467 1455	1116 1260 1320 1377 1434 1491 1545 1602 1659 1716 1780 1860 1979 2052 2109 2191 2289 2383 2520 2637 2745 2824 2912 3039 3169 3292 3424 3514 3600 3696 4316 4644 1590 1467 1455	1153 1268 1321 1384 1436 1495 1546 1609 1661 1720 1782 1883 1981 2056 2111 2210 2291 2413 2522 2669 2747 2840 2914 3039 3169 3294 3426 3518 3602 3702 4566 4645 1635 1500 1500	1154 1270 1330 1386 1444 1497 1555 1611 1669 1722 1785 1885 1984 2058 2116 2212 2296 2415 2527 2671 2752 2842 2916 3041 3173 3294 3429 3520 3606 3707 4568 4646 1680 1512 1512	1167 1275 1332 1389 1446 1500 1557 1614 1671 1727 1787 1889 1992 2063 2118 2214 2300 2418 2536 2674 2756 2845 2920 3044 3175 3297 3431 3527 3608 3724 4599 1545 1545	1174 1276 1339 1391 1450 1501 1564 1616 1675 1731 1795 1891 2004 2067 2124 2216 2310 2420 2566 2676 2780 2847 2949 3048 3200 3301 3450 3529 3609 3730 4601 1557 1557	1179 1285 1341 1399 1452 1510 1566 1624 1677 1741 1797 1893 2006 2075 2128 2224 2312 2425 2568 2679 2782 2848 2951 3083 3202 3339 3452 3531 3613 3751 4627 1590 1590
.PAGE	873	966	1275	1320	1365	1410	1455	1500	1545	1590	1635	1680			
.RADIX	1275	1287	1320	1332	1365	1377	1410	1422	1455	1467	1500	1512	1545	1557	1590
.REM	1602	1635	1647	1680	1692	1724									
.REPT	1														
.SBTTL	697 677 1366 1893 2374 2795 3819 4604	2795 689 1411 1938 2425 2820 3860 4627	722 1456 1984 2476 2848 3901	840 1501 2029 2527 2877 3938	851 1546 2063 2578 2916 3939	873 1591 2098 2633 2960 3940	917 1636 2120 2679 3044 3942	966 1681 2121 2705 3128 4019	1095 1727 2122 2706 3213 4045	1149 1766 2124 2707 3297 4102	1154 1806 2158 2710 3388 4149	1200 1807 2224 2752 3461 4214	1239 1808 2270 2791 3534 4353	1276 1811 2296 2792 3609 4432	1321 1850 2323 2793 3742 4489
.TITLE	665														
.WORD	697 886 925	698 887 926	700 890 927	701 891 928	703 892 929	848 893 933	867 894 934	868 895 935	869 896 948	870 899 952	871 900 955	872 901 958	881 922 959	884 923 960	885 924 961

K09

MAINDEC-11-DVKWA-B MACY11 27(665) 21-FEB-77 14:43 PAGE 114
DVKWAB.P11 CROSS REFERENCE TABLE

962	963	1053	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074	1075
1076	1077	1078	1079	1080	1081	3756	3759	3798	4018	4128	4133	4382	4429	4464
4524	4602	4603	4634	4716	4718	4720	4722	4723	4725	4727	4729			

ERRORS DETECTED: 0

* ,DVKWAB.SEG/CRF/SOL/NL:TOC=DSKZ:DVKWAB.SML,DVKWAB.P11
RUN-TIME: 20 24 2 SECONDS
CORE USED: 34K

