

DLV11-E

OFF LINE TEST
CVDVAB0

AH-B151B-MC

JAN 1978

COPYRIGHT © 1977

digital

FICHE 1 OF 1

MADE IN USA

.REM 2

IDENTIFICATION

PRODUCT CODE: AC-B150B-MC
PRODUCT NAME: CVDVABO DLV11-E OFFLINE TEST
PRODUCT DATE: APRIL, 1977
AUTHOR: ODES CHOATE
MAINTAINER: DIAGNOSTIC ENGINEERING GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1977 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

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

TABLE OF CONTENTS

1.0	GENERAL PROGRAM INFORMATION.
1.1	PROGRAM PURPOSE (ABSTRACT).
1.2	SYSTEM REQUIREMENTS.
1.3	RELATED DOCUMENTS AND STANDARDS.
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES.
1.5	ASSUMPTIONS.
2.0	OPERATING INSTRUCTIONS.
2.1	LOADING AND STARTING PROCEDURES.
2.2	SPECIAL ENVIRONMENTS.
2.3	OPERATIONAL SWITCH SETTINGS
2.4	PROGRAM OPTIONS.
2.5	EXECUTION TIMES.
3.0	ERROR INFORMATION.
3.1	ERROR REPORTING PROCEDURE.
3.2	ERROR HALTS.
4.0	PERFORMANCE AND PROGRESS REPORTS.
4.1	PERFORMANCE REPORTS.
5.0	DEVICE INFORMATION TABLES.
6.0	SUMMARY OF TESTS AND SPECIAL SUBROUTINES

1.0 GENERAL PROGRAM INFORMATION.

1.1 PROGRAM PURPOSE (ABSTRACT).

THIS DIAGNOSTIC IS A LOGIC TEST TO VERIFY THE OPERATION OF THE DLV11-E SERIAL LINE INTERFACE. THE PROGRAM AS SET INITIALLY DEFAULTS TO ALL OPTIONS, EXCEPT PROGRAMMABLE BAUD RATE, ENABLED AND A WAP CABLE CONNECTED. THE USER CAN SELECTIVELY ENABLE AND DISABLE TESTING OF THE OPTIONS BY ALTERING THE CONTENTS OF 'USER'. THE DIAGNOSTIC IS DESIGNED TO TEST AND DETECT FAULTS TO THE LOGIC LEVEL (NOT TO THE CHIP LEVEL). THIS TEST OPERATES ON UP TO SIXTEEN(16) IDENTICALLY CONFIGURED DLV11-E SERIAL LINE INTERFACES. THE DEFAULT ADDRESSES ARE:

175610 -FIRST SERIAL LINE ADDRESS OF 16 CONSECUTIVE SERIAL LINE DEVICES.

300 - VECTOR FOR FIRST OF 16 DEVICES.

THIS PROGRAM IS DESIGNED TO RUN ON ANY PDP-11 WITH 4K OF MEMORY AND A DLV11-E (LSI-BUS) MODULE. IT CAN RUN UNDER XXDP, APT AND ACT MONITORS, AND ON PROCESSORS WITH NO HARDWARE SWITCH REGISTER. A POWER FAILURE WILL CAUSE THE DIAGNOSTIC TO RESTART.

1.2 SYSTEM REQUIREMENTS.

1. HARDWARE REQUIREMENTS:

ANY PDP-11 FAMILY PROCESSOR
4K MEMORY - MINIMUM
H315 - CABLE TURN AROUND PLUG (OR EQUIVALENT)
MODEM CABLE - BC01V-X OR BC05C-X

SOFTWARE REQUIREMENTS:

THIS DIAGNOSTIC IS DESIGNED TO RUN IN ANY OF THE FOLLOWING WAYS:

STAND ALONE
WITH APT MONITOR
WITH ACT MONITOR
WITH XXDP MONITOR (CHAINABLE)

118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163

1.3 RELATED DOCUMENTS AND STANDARDS.

DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS
 APT
 ACT
 SYSMAC

175-003-009-02
 MD-11-DZZMA
 AUTOCAT-11-QZAUB
 MD-11-DZQAC

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES.
 NO SPECIAL DIAGNOSTICS ARE REQUIRED TO RUN BEFORE THIS, BUT
 THE PROCESSOR, MEMORY, AND BUS ARE ASSUMED TO BE FULLY
 OPERATIONAL.

1.5 ASSUMPTIONS.

THIS DIAGNOSTIC ASSUMES THAT THE OPERATOR HAS INITIALIZED
 LOCATION 'SUSWR' AND 'SDEVM' TO THE PROPER VALUES.
 THE (H) JUMPER MUST BE REMOVED FROM ALL DLV11-E'S UNDER TEST.

2.0 OPERATING INSTRUCTIONS.

2.1 LOADING AND STARTING PROCEDURES.

USE STANDARD PROCEDURE FOR PDP-11 ABSOLUTE BINARY FORMATTED
 MEDIA.

THIS DIAGNOSTIC HAS ONLY ONE (1) STARTING ADDRESS. 200 FOR
 START AND RESTART.

THE USER CAN SELECT A SPECIFIC TEST TO BE EXECUTED BY SETTING
 SWITCH B IN THE SWITCH REGISTER AND THE TEST NUMBER (IN OCTAL)
 IN THE LOWER BYTE. (NOTE: ALL TESTS PREVIOUS TO THE SELECTED
 ONE ARE EXECUTED WITHOUT ITERATIONS.)

2.2 SPECIAL ENVIRONMENTS.

THIS DIAGNOSTIC FOLLOWS THE STANDARD PROCEDURE FOR RUNNING
 UDER APT, ACT, XXDP MONITORS, AS DESCRIBED IN THEIR RESPECTIVE
 PROCEDURES MANUAL AND SYSMAC PACKAGE.

2.3 OPERATIONAL SWITCH SETTINGS

IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER. IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH REGISTER (LOC. 176) IS USED.

CONTROL:

THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY DOING THE FOLLOWING:

- 1) TYPE CONTROL G (<↑G>); THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.
- 2) THE MACHINE WILL THEN TYPE: 'SWR=XXXXXX NEW=' (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)
- 3) AFTER THE 'NEW=' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE OF THE FOLLOWING AT THE TTY:
 - A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>. (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED). LEADING ZEROS NEED NOT BE TYPED, AND IF MORE THAN 6 DIGITS ARE TYPED THE LAST 6 WILL BE USED. IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.
 - B) IF A CONTROL U (<↑U>) IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 3.
 - C) IF THE INPUT CHARACTER IS NOT ONE OF THE CHARACTERS MENTIONED ABOVE THEN A QUESTION MARK (?) WILL BE TYPED FOLLOWED BY A CARRAGE RETURN AND A LINE FEED THEN PROCEED FROM STEP 3 (ERASING ALL PREVIOUS INPUT.)

DYNAMIC SWITCH REGISTER

- BIT 15 - HALT ON ERROR
 14 - LOOP ON TEST
 13 - INHIBIT ERROR TYPEOUTS
 12 - (UNUSED)
 11 - INHIBIT ITERATIONS
 10 - BELL ON ERROR
 9 - LOOP ON ERROR
 8 - LOOP ON TEST IN SWR<7:0>
 7:0 - TEST NUMBER TO LOOP ON (USED WITH BIT 8)

2.4 PROGRAM OPTIONS.

THIS PROGRAM WILL SUPPORT TESTING OF MULTIPLE DLV11-E'S. IT
REQUIRES THE ADDRESS OF THE FIRST RCSR (STORED AT '\$BASE'), AND
ITS INTERRUPT VECTOR (STORED AT '\$VECT1'). AND WILL BE ABLE
TO ADDRESS ANY DLV11-E STARTING AT THE SPECIFIED BASE ADDRESS
UP TO 16 CONSECUTIVE DEVICES.
EXAMPLES: \$BASE: 175610
\$VECT1: 300

THE PROGRAM WILL BE ABLE TO TEST ANY DLV11-E WITHIN THE
ADDRESS RANGE 175610 --> 176000

\$BASE AND \$VECT1 DEFAULT TO 175610 AND 300 RESPECTIVELY.
THE PROGRAM ASSOCIATES UNIT NUMBERS AS FOLLOWS: (NUMBERS IN
PARENTHESES ARE OCTAL)

UNIT#0 -- BASE ADDRESS STORED AT '\$BASE'
ASSOCIATED BASE VECTOR STORED AT '\$VECT1'
UNIT#1 -- BASE ADDRESS + (10)
BASE VECTOR + (10)

UP TO

UNIT#15 -- BASE ADDRESS + (170)
BASE VECTOR + (170)

LOCATION '\$DEVN' IS USED AS A BIT MAP TO INDICATE WHICH UNIT
NUMBERS ARE PRESENT AND WILL BE TESTED.

BIT 15	BIT 1	BIT 0
!UNIT!	!UNIT!	!UNIT!
! 15 !	! #1 !	! #0 !

A BIT MAP CAN BE ENTERED AT '\$DEVN' PRIOR TO STARTING THE
PROGRAM.

EXAMPLE:
\$BASE: 175610
\$VECTOR: 300
\$DEVN: 13

THE PROGRAM WILL TEST-

UNIT#0	175610	300
UNIT#1	175620	310
UNIT#3	175640	330

H01

SEQ 0007

OPTIONS

LOCATION \$USWR CONTAINS ALL THE USER SELECTABLE OPTIONS. THE
 VALUES IN THIS WORD MUST CONFORM TO THE ACTUAL BOARD
 CONFIGURATION.
 THE DEFAULT VALUE OF \$USWR IS AS FOLLOWS:

BIT POSITION	DEFINITION	DEFAULT VALUE
-----	-----	-----
0-3	#OF DATA BITS	10(8) = 8
4	PARITY ENABLED	0 = NO
5	EVEN ODD PARITY	0 = ODD
6	COMMON SPEED	1 = YES
7	PROGRAMMABLE BAUD RATE	0 = NO
8-11	BAUD RATE OFFSET (SEE FOLLOWING NOTE)	05(8) = 110 BAUD
12	BREAK GENERATION ENABLED	1 = YES
13	CABLE TERMINATED (H315)	1 = YES
14	(-FR) AND (-FD) JUMPERS IN	1 = YES
15	(NOT DEFINED)	

NOTE

THIS DIAGNOSTIC DOES NOT TEST THE PARITY LOGIC.

WHEN THE PROGRAMMABLE BAUD RATE OPTION IS
 ENABLED THE PROGRAMMABLE BAUD RATE TEST
 WILL EXIT WITH THE BAUD RATE SET TO THE
 SELECTED VALUE. TO CHANGE THE DEFAULT
 VALUE OF 110 BAUD REPLACE BITS <11:8>
 WITH THE OFFSET INDICATED IN THE TABLE
 AT THE END OF THE PBR TEST.

2.5 EXECUTION TIMES.

EXECUTION TIMES ARE FOR AN LSI-11 PROCESSOR WITH ALL OPTIONS
 ENABLED ON THE DLV11-E (EXCEPT FOR PROGRAMMABLE BAUD RATE), AT
 110 BAUD.

FIRST PASS- 90 SECONDS
 ADDITIONAL PASSES 95 SECONDS
 ADDITIONAL DEVICES 95 SECONDS

THE TEST TIME IS BAUD RATE DEPENDANT; HIGHER BAUD GIVES
 SHORTER PASS TIMES.

267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321

3.0 ERROR INFORMATION.

3.1 ERROR REPORTING PROCEDURE.

SINCE THIS DIAGNOSTIC WAS DESIGNED TO FIT IN 4-K OF MEMORY THE ERROR TYPEOUT IS VERY BRIEF. THE FORMAT OF THE ERROR TYPEOUT IS AS FOLLOWS:

TEST#+++++,ERROR#+++++,PC=+++++,ADDRESS=+++++,VECTOR=+++++

WHERE ALL VALUES TYPED ARE OCTAL.
 THE ADDRESS AND VECTOR REFER TO THE FAILING DLV11-E.
 FOR FURTHER INFORMATION THE LISTING MUST BE CONSULTED.
 BITS 15,13,10 AND 9 OF THE SWITCH REGISTER CONTROL THE SEQUENCE OF EVENTS AFTER AN ERROR IS CAUGHT.

BIT 15 - CAUSES THE PROGRAM TO HALT IN THE ERROR ROUTINE. CONTINUEING THE PROGRAM CAUSES IT TO PROCEED.

BIT 13 - DISABLES THE PRINTING OF THE ERROR MESSAGE.

BIT 10 - CAUSES THE BELL TO RING ON ERROR.

BIT 9 - CAUSES THE DIAGNOSTIC TO LOOP FROM BEGINNING OF TEST TO ERROR.

THE ERROR ROUTINE SUPPORTS THE CONTROL G FUNCTION.

3.2 ERROR HALTS.

THE ONLY HALT IN THIS DIAGNOSTIC IS IN THE ERROR ROUTINE, AND IS EXECUTED ONLY IF BIT 15 OF THE SWITCH REGISTER IS A ONE WHEN AN ERROR OCCURS.

4.0 PERFORMANCE AND PROGRESS REPORTS.

4.1 PERFORMANCE REPORTS.

AS EACH DEVICE COMPLETES ONE PASS OF THE DIAGNOSTIC THE FOLLOWING WILL BE TYPED:

CSR:+++++,VECTOR:+++++,ERRORS:+++++

WHERE. 'CSR:+++++' IS THE DEVICE CSR UNDER TEST
 'VECTOR:++' IS THE ASSOCIATED VECTOR
 AND 'ERRORS:++' IS THE TOTAL NUMBER OF ERRORS ON THIS DEVICE ON THIS PASS.

NOTE

THIS IS TYPED AFTER THE DEVICE HAS COMPLETED ITS PASS.

J01

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 10
CVDVAB.P11 15-DEC-77 08:58

SEQ 0009

378
379
380
381

AFTER ALL DEVICES HAVE BEEN EXERCISED AN END PASS STATEMENT IS
TYPED: "ENDPASS+++++."

5.0 DEVICE INFORMATION TABLES.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCSR	DATA	RING	CLR	CAR	RCVR	REC			RCVR	RCVR	DATA		SEC	REQ	DTR	
	INT		SEND	DET	ACT	REC			DONE	IE	IE		XMIT	SEND		
RBUF	ERRO	OR	FR	P												
	R	ERR	ERR	ERR												
TCSR	PROGRAMMABLE	BAUD	PBR						XMIT	XMIT				MAIN		BREA
	RATE	SELECT	ENAB						RDY	IE				T		K
TBUF																

NOTE

BLANK BOXES INDICATE UNUSED AND RESERVED
BIT POSITIONS. SEE THE LISTING FOR AN
EXPLANATION OF THE BITS.

6.0 SUMMARY OF TESTS AND SPECIAL SUBROUTINES.

TEST 1 ADDRESSABILITY

THIS TEST VERIFIES THAT THE ADDRESS AS PLACED IN THE
HARDWARE P-TABLE TO BE CORRECT AND THE DLV11-E
RESPONDS TO THAT ADDRESS SPACE.

THE FOLLOWING 8 TESTS TEST ALL 'READ WRITE' BITS

TEST 2 BREAK - TCSR0 SET, CLEAR, RESET

TEST 3 MAINT - TCSR2 SET, CLEAR, RESET

TEST 4 XMITIE - TCSR6 SET, CLEAR, RESET

TEST 5 DTR - RCSR1 SET, CLEAR

NOTE

RESET DOES NOT CLEAR THIS BIT. WE CANNOT TEST
FOR AN INITIAL CONDITION AS THIS BIT IS
UNDEFINED UPON POWER UP AND INIT DOESN'T
AFFECT IT.

TEST 6 REQSEND - RCSR2 SET, CLEAR, RESET

THIS TEST ASSUMES THAT JUMPER FR IS IN.

TEST 7 SECXMIT - RCSR3 SET, CLEAR, RESET

TEST 10 DATAIE - RCSR5 SET, CLEAR, RESET

TEST 11 RCVRIE - RCSR6 SET, CLEAR, RESET

410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 13
CVDVAB.P11 15-DEC-77 08:58

MO1

SEQ 0012

466

NO1

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 14
CVDVAB.P11 15-DEC-77 08:58

SEQ 0013

467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522

THE FOLLOWING 4 TESTS VERIFY THAT RESET (INIT) INITIALIZES
READ ONLY BITS.

TEST 12 RCVRDONE - RCSR 7 - IS CLEARED BY INIT
---- --

TEST 13 XMITRDY - TCSR 7 - IS SET BY INIT
---- --

TEST 14 DATAINT - RCSR 15 - IS CLEARED BY INIT.
---- --

TEST 15 RCVRACK - RCSR 1: - IS CLEARED BY INIT
---- --

THE FOLLOWING 4 TESTS VERIFY THAT THE EIA SIGNALS CAN BE
TRANSMITTED AND RECEIVED THROUGH THE CABLE.

TEST 16 CAPDET SETS AND CLEARS AS DTR SETS AND CLEARS
---- --

TEST 17 CLREND SETS AND CLEARS AS DTR SETS AND CLEARS
---- --

TEST 20 RING SETS AND CLEARS AS REQSEND SETS AND CLEARS
---- --

TEST 21 SECRC SETS AND CLEARS AS SECXMIT SETS AND CLEARS
---- --

TEST 22 DATAINT (RCSR-15) SETS WHEN DTR CHANGES STATE AND THAT
DATAINT IS CLEARED AFTER READING RCSR

NOTE

DTR IS TIED TO BOTH CAPDET AND CLREND BY THE
H315.

TEST 23 DATAINT SETS WHEN RING SETS AND THAT DATAINT
DOES NOT SET WHEN RING CLEARS

TEST 24 DATAINT SETS WHEN SECRC CHANGES STATE
---- --

802

MAINDEC-22-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 15
CVDVAB.P11 15-DEC-77 08:58

SEQ 0014

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
565
566
567
568
569
570
571
572
573
574
575
576

TEST 25 XMIT RDY - TCSF 7 - CLEARS WHEN TBUF IS LOADED
---- --
WITH A CHARACTER AND THAT IT SETS WITHIN A
REASONABLE AMOUNT OF TIME.

TEST 26 OUTPUTTING A CHAR FROM TBUF (WITH MAINT SET)
---- --
RESULTS IN RCVRDONE SETTING WITHIN A
REASONABLE AMOUNT OF TIME AND THAT RESET
CLEARS THE BIT.

TEST 27 RCVRDONE IS CLEARED BY READING RBUF
---- --

TEST 30 RCVRACT - RCSR 11 - SETS WHEN A START BIT IS
---- --
RECEIVED AND CLEARS WHEN RCVRDONE - RCSR 7 -
SETS

TEST 31 OVERRUN BIT - RBUF 14
---- --

TEST 32 PROGRAMMABLE BAUD RATE TEST TEST AT ALL SPEEDS
---- --
AVAILABLE A COMPARISON WILL BE MADE TO SEE IF
NEW TIME IS LESS THAN PREVIOUS.

TEST 33 TRANSMITTER INTERRUPT LOGIC TEST
---- --
LOGICALLY THIS IS 4 SEPARATE TESTS
A) DOES TRANSMITTER INTERRUPT LOGIC WORK
B) AT PRIORITY OF 0
C) AND ONLY ONCE
D) BUT NOT WITH INTERRUPT ENABLE CLEAR

TEST 34 RECEIVER INTERRUPT LOGIC TEST THIS TEST COVERS ALL
---- --
OF THE RECEIVER SIDE OF THE INTERRUPT LOGIC, BOTH
DATASET AND CHARACTER MODES.

TEST 35 TEST ACTUAL DATA TRANSFERED NON-INTERRUPT
---- --
MAINTENANCE BIT SET

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
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619

TEST 36 TEST DATA THROUGH CABLE
---- --

TEST 37 FULL DATA TRANSFER WITH INTERRUPTS AND MAINTENANCE
---- --
MODE.

TEST 40 TEST BREAK GENERATION LOGIC TRANSMIT KNOWN CHAR
---- --
WITH BREAK SET AND COMPARE RECEIVED WITH 0.

TEST 41 NOT A TEST - SEND BACK TO LOOP
---- --

NOTE

FOR ALL OF THE FOLLOWING ROUTINES THE USE
OF (RS) IS PART OF THE LINKAGE MECHANISM
BETWEEN THE CALLER AND THE CALLED.

ROUTINE:TIMER

THIS ROUTINE IS USED TO TEST THE STATUS OF
ANY BIT IN ANY REGISTER.

INPUTS: HOWLONG THE MAXIMUM AMOUNT OF TIME TO
SPEND IN THIS ROUTINE.
WHICHBIT A MASK WITH THE BIT(S) SET THAT
ARE TO BE CHECKED
REG A POINTER TO THE REGISTER TO BE
CHECKED
SETCLR THE DESIRED RESULTS -- EITHER SET
OR CLEAR

OUTPUT: THE 'C' BIT IS SET TO INDICATE AN ERROR BUT IT
IS TESTED BY THE IF.ERROR STATEMENT.

620
 621
 622
 623
 624
 625
 626
 627
 628
 629
 630
 631
 632
 633
 634
 635
 636
 637
 638
 639
 640
 641
 642
 643
 644
 645
 646
 647
 648
 649
 650
 651
 652
 653
 654
 655
 656
 657
 658
 659
 660
 661

ROUTINE:DATLNG

 THIS ROUTINE SETS UP A MASK FOR DATA, WITH -
 INPUT: NOTHING IS PASSED TO THIS ROUTINE BUT GLOBAL
 INFORMATION IS ASSUMED TO EXIST:
 \$USWR-- THE WORD FOR SOFTWARE PARAMETERS
 DATA-- A MASK FOR THE LOCATION OF THE OCTAL
 NUMBER OF DATA BITS

OUTPUT----

 MASK-- A MASK OF BINARY ZEROS RIGHT-JUSTIFIED
 THE NUMBER OF WHICH IS DEFINED IN \$USWR WORD.

ROUTINE:WAIT

 THIS ROUTINE IS USED TO DELAY EXECUTION OF THE
 MAIN PROGRAM FOR A SPECIFIED AMOUNT OF TIME.
 THIS IS ACCOMPLISHED BY INCREMENTING A
 REGISTER UP TO A LIMIT. THE INNER LOOP IS SET
 TO APPROXIMATE 1 MILLI SEC.

SERVICE ROUTINE: INTSRV

 THIS GLOBAL ROUTINE DOES NOTHING BUT INCREMENT
 'INTFLAG' EACH TIME IT IS CALLED. IT ASSUMES
 THAT THE MAIN CALLING ROUTINE WILL KNOW WHAT
 TO LOOK FOR.

ROUTINE:CYCLE

 THIS ROUTINE CAUSES ADRS TO POINT TO THE
 ADDRESS OF DLV11-E UNDER TEST. ADRS +2 TO
 POINT TO THE VECTOR OF THE DLV11-E UNDER TEST.
 IT KEEPS TRACK OF THE CURRENT DEVICE AND BIT
 MASKS.

```

662      2
663      .TITLE MAINDEC-ZZ-CVDVA-B
664      .*COPYRIGHT (C) 1977
665      .*DIGITAL EQUIPMENT CORP.
666      .*MAYNARD, MASS. 01754
667      .*
668      .*PROGRAM BY ODES CHOATE
669      .*
670      .*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
671      .*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
672      .*
673      .SBTTL OPERATIONAL SWITCH SETTINGS
674      .*
675      .*          SWITCH          USE
676      .*          -----
677      .*          15          HALT ON ERROR
678      .*          14          LOOP ON TEST
679      .*          13          INHIBIT ERROR TYPEOUTS
680      .*          11          INHIBIT ITERATIONS
681      .*          10          BELL ON ERROR
682      .*          9          LOOP ON ERROR
683      .*          8          LOOP ON TEST IN SWR<7:0>
684
685      .SBTTL BASIC DEFINITIONS
686
687      .*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
688      STACK= 1100
689      .EQUIV EMT,ERROR          ;;BASIC DEFINITION OF ERROR CALL
690      .EQUIV IOT,SCOPE          ;;BASIC DEFINITION OF SCOPE CALL
691
692      .*MISCELLANEOUS DEFINITIONS
693      HT= 11          ;;CODE FOR HORIZONTAL TAB
694      LF= 12          ;;CODE FOR LINE FEED
695      CR= 15          ;;CODE FOR CARRIAGE RETURN
696      CRLF= 200        ;;CODE FOR CARRIAGE RETURN-LINE FEED
697      PS= 177776       ;;PROCESSOR STATUS WORD
698      .EQUIV PS,PSW
699      STKLM= 177774     ;;STACK LIMIT REGISTER
700      PIRQ= 177772      ;;PROGRAM INTERRUPT REQUEST REGISTER
701      DSWR= 177570     ;;HARDWARE SWITCH REGISTER
702      DDISP= 177570    ;;HARDWARE DISPLAY REGISTER
703
704      .*GENERAL PURPOSE REGISTER DEFINITIONS
705      R0= %0          ;;GENERAL REGISTER
706      R1= %1          ;;GENERAL REGISTER
707      R2= %2          ;;GENERAL REGISTER
708      R3= %3          ;;GENERAL REGISTER
709      R4= %4          ;;GENERAL REGISTER
710      R5= %5          ;;GENERAL REGISTER
711      R6= %6          ;;GENERAL REGISTER
712      R7= %7          ;;GENERAL REGISTER
713      SP= %6          ;;STACK POINTER
714      PC= %7          ;;PROGRAM COUNTER
715
716      .*PRIORITY LEVEL DEFINITIONS
717      PRO= 0          ;;PRIORITY LEVEL 0

```

718	000040	PR1=	40	:: PRIORITY LEVEL 1
719	000100	PR2=	100	:: PRIORITY LEVEL 2
720	000140	PR3=	140	:: PRIORITY LEVEL 3
721	000200	PR4=	200	:: PRIORITY LEVEL 4
722	000240	PR5=	240	:: PRIORITY LEVEL 5
723	000300	PR6=	300	:: PRIORITY LEVEL 6
724	000340	PR7=	340	:: PRIORITY LEVEL 7
725		: *SWITCH REGISTER* SWITCH DEFINITIONS		
726		SW15=	100000	
727	100000	SW14=	40000	
728	040000	SW13=	20000	
729	020000	SW12=	10000	
730	010000	SW11=	4000	
731	004000	SW10=	2000	
732	002000	SW09=	1000	
733	001000	SW08=	400	
734	000400	SW07=	200	
735	000200	SW06=	100	
736	000100	SW05=	40	
737	000040	SW04=	20	
738	000020	SW03=	10	
739	000010	SW02=	4	
740	000004	SW01=	2	
741	000002	SW00=	1	
742	000001	.EQUIV	SW09, SW9	
743		.EQUIV	SW08, SW8	
744		.EQUIV	SW07, SW7	
745		.EQUIV	SW06, SW6	
746		.EQUIV	SW05, SW5	
747		.EQUIV	SW04, SW4	
748		.EQUIV	SW03, SW3	
749		.EQUIV	SW02, SW2	
750		.EQUIV	SW01, SW1	
751		.EQUIV	SW00, SW0	
752		: *DATA BIT DEFINITIONS (BIT00 TO BIT15)		
753		BIT15=	100000	
754		BIT14=	40000	
755	100000	BIT13=	20000	
756	040000	BIT12=	10000	
757	020000	BIT11=	4000	
758	010000	BIT10=	2000	
759	004000	BIT09=	1000	
760	002000	BIT08=	400	
761	001000	BIT07=	200	
762	000400	BIT06=	100	
763	000200	BIT05=	40	
764	000100	BIT04=	20	
765	000040	BIT03=	10	
766	000020	BIT02=	4	
767	000010	BIT01=	2	
768	000004	BIT00=	1	
769	000002	.EQUIV	BIT09, BIT9	
770	000001	.EQUIV	BIT08, BIT8	
771		.EQUIV	BIT07, BIT7	
772				
773				

```

774      .EQUIV BIT06,BIT6
775      .EQUIV BIT05,BIT5
776      .EQUIV BIT04,BIT4
777      .EQUIV BIT03,BIT3
778      .EQUIV BIT02,BIT2
779      .EQUIV BIT01,BIT1
780      .EQUIV BIT00,BIT0
781
782      ;*BASIC "CPU" TRAP VECTOR ADDRESSES
783      ERRVEC= 4      ; TIME OUT AND OTHER ERRORS
784      RESVEC= 10     ; RESERVED AND ILLEGAL INSTRUCTIONS
785      TBITVEC= 14    ; "T" BIT
786      TRIVEC= 14    ; TRACE TRAP
787      BPTVEC= 14    ; BREAKPOINT TRAP (BPT)
788      IOTVEC= 20     ; INPUT/OUTPUT TRAP (IOT) **SCOPE**
789      PWRVEC= 24     ; POWER FAIL
790      EMTVEC= 30     ; EMULATOR TRAP (EMT) **ERROR**
791      TRAPVEC= 34    ; "TRAP" TRAP
792      TKVEC= 60      ; TTY KEYBOARD VECTOR
793      TPVEC= 64      ; TTY PRINTER VECTOR
794      PIRQVEC= 240   ; PROGRAM INTERRUPT REQUEST VECTOR
795
796      ILLMEM= 4
797      ADRS= R1
798      GOOD= R2
799      BAD= R3
800      REGISTER=R1
801      BIT= R2
802      FUNCT= R3
803      LEAD= R2
804      FOLLOW= R4
805      DLADDR= 175610
806
807      ; THE FOLLOWING DEFINITIONS APPLY TO THE GLOBAL SUBS
808      SET= -1
809      CLR= 0
810
811      ;*****
812      ; RCSR REGISTER BIT NAMES
813      ;*****
814      DATAINT= BIT15 ; DATASET INTERRUPT
815      RING= BIT14 ; RINGING SIGNAL INDICATOR
816      CLSEND= BIT13 ; CLEAR TO SEND FROM DATASET
817      CARDET= BIT12 ; CARRIER DETECT
818      RCVRACT= BIT11 ; RECEIVER ACTIVE INDICATOR
819      SECREC= BIT10 ; SECONDARY RECEIVE
820      ; UNUSED BIT09
821      ; UNUSED BIT08
822      RCVROONE= BIT07 ; RECEIVER DONE
823      RCVRIE= BIT06 ; RECEIVER INTERRUPT ENABLE
824      DATAIE= BIT05 ; DATASET INTERRUPT ENABLE
825      ; UNUSED BIT04
826      SECXMIT= BIT03 ; SECONDARY TRANSMIT DATA
827      REQSEND= BIT02 ; REQUEST TO SEND
828      DTR= BIT01 ; DATA TERMINAL READY
829      ; UNUSED BIT00

```



```

877      : UNUSED      BIT14
878      : UNUSED      BIT13
879      : UNUSED      BIT12
880      : UNUSED      BIT11
881      : UNUSED      BIT10
882      : UNUSED      BIT09
883      : UNUSED      BIT08
884      000200      TDATA7=      BIT07      : \
885      000100      TDATA6=      BIT06      : |
886      000040      TDATA5=      BIT05      : |
887      000020      TDATA4=      BIT04      : | \ TRANSMITTER DATA BUFFER
888      000010      TDATA3=      BIT03      : |
889      000004      TDATA2=      BIT02      : |
890      000002      TDATA1=      BIT01      : |
891      000001      TDATA0=      BIT00      : /
892
893
894      ;;*****
895      ; FLAG BITS TO BE USE OR CLEARED IN $USWR.
896
897      000017      DATA =      17
898      000020      PARITY =      20
899      000040      EVENODD =      40
900      000100      COMSPD =      100
901      000200      PBR =      200
902
903      ; BAUDE MUST BE ON THE UPPER
904      ; BYTE BOUNDARY OF $USWR.--4 BITS
905      007400      BAUD =      7400
906      010000      BRK =      10000
907      020000      CABLE =      20000
908      040000      FRFD =      40000
909      ;;*****
910      .SBTTL TRAP CATCHER
911
912      000000      .=0
913      ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
914      ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
915      ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
916      000174      .=174
917      000174      000000      DISPREG: .WORD 0      ;;SOFTWARE DISPLAY REGISTER
918      000176      000000      SWREG: .WORD 0      ;;SOFTWARE SWITCH REGISTER
919      .SBTTL STARTING ADDRESS(ES)
920      000200      000137      001336      JMP 2*START ;;JUMP TO STARTING ADDRESS OF PROGRAM
921      .SBTTL ACT11 HOOKS
922
923      ;;*****
924      ;HOOKS REQUIRED BY ACT11
925      000204      $SVPC=.      ;SAVE PC
926      000046      .=46
927      000046      012362      $ENDAD      ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
928      000052      .=52
929      000052      000000      .WORD 0      ;;2)SET LOC.52 TO ZERO
930      000204      .=$SVPC      ;; RESTORE PC
931      001000      .=1000
932      .SBTTL APT PARAMETER BLOCK

```

K02

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 24
 CVDVAB.P11 15-DEC-77 08:58 APT PARAMETER BLOCK

SEQ 0023

```

933
934
935
936
937      001000
938      000024
939      000024 000200
940      000044 000044
941      000044 001000
942      001000
943
944
945
946
947      001000
948      001000 000000
949      001002 001174
950      001004 000005
951      001006 000055
952      001010 000036
953      001012 000030

```

```

*****
SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
*****
.SX=.      SAVE CURRENT LOCATION
.=24      SET POWER FAIL TO POINT TO START OF PROGRAM
200      FOR APT START UP
.=44      POINT TO APT INDIRECT ADDRESS PNTR.
$APTHOR   POINT TO APT HEADER BLOCK
.=.SX     RESET LOCATION COUNTER
*****
SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
INTERFACE SPEC.

```

```

$APTHO:
$HIBTS: .WORD 0      ;; TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MBADR: .WORD $MAIL  ;; ADDRESS OF APT MAILBOX (BITS 0-15)
$STMT:  .WORD 5      ;; RUN TIM OF LONGEST TEST
$PASTM: .WORD 45.    ;; RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
$UNITM: .WORD 30.    ;; ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
          .WORD $ETEND-$MAIL/2 ;; LENGTH MAILBOX-ETABLE(WORDS)

```

```

954 .SBTTL COMMON TAGS
955
956 ;*****
957 ;THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
958 ;*USED IN THE PROGRAM.
959
960 001100 .=1100 ;;START OF COMMON TAGS
961 001100 SCMTAG: .WORD 0
962 001100 000000 ;;CONTAINS THE TEST NUMBER
963 001102 000 STSTNM: .BYTE 0 ;;CONTAINS ERROR FLAG
964 001103 000 SERFLG: .BYTE 0 ;;CONTAINS SUBTEST ITERATION COUNT
965 001104 000000 $ICNT: .WORD 0 ;;CONTAINS SCOPE LOOP ADDRESS
966 001106 000000 $LPADR: .WORD 0 ;;CONTAINS SCOPE RETURN FOR ERRORS
967 001110 000000 $LPERR: .WORD 0 ;;CONTAINS TOTAL ERRORS DETECTED
968 001112 000000 $ERTTL: .WORD 0 ;;CONTAINS ITEM CONTROL BYTE
969 001114 000 $ITEMB: .BYTE 0 ;;CONTAINS MAX. ERRORS PER TEST
970 001115 001 $ERMAX: .BYTE 1 ;;CONTAINS PC OF LAST ERROR INSTRUCTION
971 001116 000000 $ERRPC: .WORD 0 ;;CONTAINS ADDRESS OF 'GOOD' DATA
972 001120 000000 $GADDR: .WORD 0 ;;CONTAINS ADDRESS OF 'BAD' DATA
973 001122 000000 $BADDR: .WORD 0 ;;CONTAINS 'GOOD' DATA
974 001124 000000 $GDATA: .WORD 0 ;;CONTAINS 'BAD' DATA
975 001126 000000 $BDATA: .WORD 0 ;;RESERVED--NOT TO BE USED
976 001130 000000 .WORD 0
977 001132 000000 .WORD 0
978 001134 000 $AUTOB: .BYTE 0 ;;AUTOMATIC MODE INDICATOR
979 001135 000 $INTAG: .BYTE 0 ;;INTERRUPT MODE INDICATOR
980 001136 000000 .WORD 0
981 001140 177570 $SWR: .WORD DSWR ;;ADDRESS OF SWITCH REGISTER
982 001142 177570 $DISPLAY: .WORD DDISP ;;ADDRESS OF DISPLAY REGISTER
983 001144 177560 $TKS: 177560 ;;TTY KBD STATUS
984 001146 177562 $TKB: 177562 ;;TTY KBD BUFFER
985 001150 177564 $TPS: 177564 ;;TTY PRINTER STATUS REG. ADDRESS
986 001152 177566 $TPB: 177566 ;;TTY PRINTER BUFFER REG. ADDRESS
987 001154 000 $NULL: .BYTE 0 ;;CONTAINS NULL CHARACTER FOR FILLS
988 001155 002 $FILLS: .BYTE 2 ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
989 001156 012 $FILLC: .BYTE 12 ;;INSERT FILL CHARS. AFTER A "LINE FEED"
990 001157 000 $TPFLG: .BYTE 0 ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
991 001160 000000 $TIMES: 0 ;;MAX. NUMBER OF ITERATIONS
992 001162 000000 $ESCAPE: 0 ;;ESCAPE ON ERROR ADDRESS
993 001164 177607 000377 $BELL: .ASCIZ <207><377><377> ;;CODE FOR BELL
994 001170 077 $QUES: .ASCII /?/ ;;QUESTION MARK
995 001171 015 $CRLF: .ASCII <15> ;;CARRIAGE RETURN
996 001172 000012 $LF: .ASCIZ <12> ;;LINE FEED
997 ;*****
998 .SBTTL APT MAILBOX-ETABLE
999
1000 ;*****
1001 .EVEN
1002 001174 $MAIL: .WORD 0 ;;APT MAILBOX
1003 001174 000000 $MSGTY: .WORD 0 ;;MESSAGE TYPE CODE
1004 001176 000000 $FATAL: .WORD 0 ;;FATAL ERROR NUMBER
1005 001200 000000 $TESTN: .WORD 0 ;;TEST NUMBER
1006 001202 000000 $PASS: .WORD 0 ;;PASS COUNT
1007 001204 000000 $DEVCT: .WORD 0 ;;DEVICE COUNT
1008 001206 000000 $UNIT: .WORD 0 ;;I/O UNIT NUMBER
1009 001210 000000 $MSGAD: .WORD 0 ;;MESSAGE ADDRESS

```

M02

MAINDEC-22-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 26
 CVDVAB.P11 15-DEC-77 08:58 APT MAILBOX-ETABLE

SEQ 0025

1010	001212	000000	\$MSGLG: .WORD	AMSGLG	:: MESSAGE LENGTH
1011	001214		\$ETABLE:		:: APT ENVIRONMENT TABLE
1012	001214	000	\$ENV: .BYTE	AENV	:: ENVIRONMENT BYTE
1013	001215	000	\$ENVM: .BYTE	AENVM	:: ENVIRONMENT MODE BITS
1014	001216	000000	\$SWREG: .WORD	ASWREG	:: APT SWITCH REGISTER
1015	001220	071110	\$USWR: .WORD	AUSWR	:: USER SWITCHES
1016	001222	000000	\$CPUOP: .WORD	ACPUOP	:: CPU TYPE, OPTIONS
1017			::		BITS 15-11=CPU TYPE
1018			::		11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
1019			::		11/70=06, P00=07, 0=10
1020			::		BIT 10=REAL TIME CLOCK
1021			::		BIT 9=FLOATING POINT PROCESSOR
1022			::		BIT 8=MEMORY MANAGEMENT
1023	001224	000	\$MAMS1: .BYTE	AMAMS1	:: HIGH ADDRESS, M.S. BYTE
1024	001225	000	\$MTYP1: .BYTE	AMTYP1	:: MEM. TYPE, BLK#1
1025			::		MEM. TYPE BYTE -- (HIGH BYTE)
1026			::		900 NSEC CORE=001
1027			::		300 NSEC BIPOLAR=002
1028			::		500 NSEC MOS=003
1029	001226	000000	\$MAOR1: .WORD	AMAOR1	:: HIGH ADDRESS, BLK#1
1030			::		MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
1031	001230	000	\$MAMS2: .BYTE	AMAMS2	:: HIGH ADDRESS, M.S. BYTE
1032	001231	000	\$MTYP2: .BYTE	AMTYP2	:: MEM. TYPE, BLK#2
1033	001232	000000	\$MAOR2: .WORD	AMAOR2	:: MEM. LAST ADDRESS, BLK#2
1034	001234	000	\$MAMS3: .BYTE	AMAMS3	:: HIGH ADDRESS, M.S. BYTE
1035	001235	000	\$MTYP3: .BYTE	AMTYP3	:: MEM. TYPE, BLK#3
1036	001236	000000	\$MAOR3: .WORD	AMAOR3	:: MEM. LAST ADDRESS, BLK#3
1037	001240	000	\$MAMS4: .BYTE	AMAMS4	:: HIGH ADDRESS, M.S. BYTE
1038	001241	000	\$MTYP4: .BYTE	AMTYP4	:: MEM. TYPE, BLK#4
1039	001242	000000	\$MAOR4: .WORD	AMAOR4	:: MEM. LAST ADDRESS, BLK#4
1040	001244	000300	\$VECT1: .WORD	AVECT1	:: INTERRUPT VECTOR#1, BUS PRIORITY#1
1041	001246	000000	\$VECT2: .WORD	AVECT2	:: INTERRUPT VECTOR#2, BUS PRIORITY#2
1042	001250	175610	\$BASE: .WORD	ABASE	:: BASE ADDRESS OF EQUIPMENT UNDER TEST
1043	001252	000001	\$DEVN: .WORD	ADEVN	:: DEVICE MAP
1044	001254		\$ETEND:		
1045			.MEXIT		

.SBTTL ERROR POINTER TABLE

;;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
 ;;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
 ;;*LOCATION SITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
 ;;*NOTE1: IF SITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).
 ;;*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

;;* EM ;;POINTS TO THE ERROR MESSAGE
 ;;* D1 ;;POINTS TO THE DATA HEADER
 ;;* DT ;;POINTS TO THE DATA
 ;;* DF ;;POINTS TO THE DATA FORMAT

\$ERRTB:

;; GLOBAL DATA

DLADD: DLADDR
 DLVEC: 300
 RCSR: DLADDR + 0
 RBUF: DLADDR + 2
 TCSR: DLADDR + 4
 TCSRHI: DLADDR + 5
 TBUF: DLADDR + 6

I: 0
 .BLKW 20 ;FOR R5 STACK
 RSSTACK: .WORD 0

START:

.SBTTL INITIALIZE THE COMMON TAGS

;;CLEAR THE COMMON TAGS (\$CMTAG) AREA

MOV #SCMTAG,R6 ;;FIRST LOCATION TO BE CLEARED
 CLR (R6)+ ;;CLEAR MEMORY LOCATION
 CMP #SWR,R6 ;;DONE?
 BNE -6 ;;LOOP BACK IF NO
 MOV #STACK,SP ;;SETUP THE STACK POINTER

;;INITIALIZE A FEW VECTORS

MOV #SCOPE,2#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
 MOV #340,2#IOTVEC+2 ;;LEVEL 7
 MOV #ERROR,2#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
 MOV #340,2#EMTVEC+2 ;;LEVEL 7
 MOV #STRAP,2#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
 MOV #340,2#TRAPVEC+2 ;;LEVEL 7
 MOV #SPWRON,2#PWRVEC ;;POWER FAILURE VECTOR
 MOV #340,2#PWRVEC+2 ;;LEVEL 7
 MOV #ENDCT,SEOPCT ;;SETUP END-OF-PROGRAM COUNTER
 CLR \$TIMES ;;INITIALIZE NUMBER OF ITERATIONS
 CLR \$ESCAPE ;;CLEAR THE ESCAPE ON ERROR ADDRESS
 MOV #1,\$ERMAX ;;ALLOW ONE ERROR PER TEST
 MOV #,\$LPAOR ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
 MOV #,\$LPERR ;;SETUP THE ERROR LOOP ADDRESS

;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS

;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.

MOV 2#ERRVEC, -(SP) ;;SAVE ERROR VECTOR
 MOV #64,\$ERRVEC ;;SET UP ERROR VECTOR
 MOV #DSWR,SWR ;;SETUP FOR A HARDWARE SWICH REGISTER
 MOV #DDISP,DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
 CMP #-1,\$SWR ;;TRY TO REFERENCE HARDWARE SWR

1046
 1047
 1048
 1049
 1050
 1051
 1052
 1053
 1054
 1055
 1056
 1057
 1058
 1059
 1060 001254
 1061
 1062 001254 175610
 1063 001256 000300
 1064 001260 175610
 1065 001262 175612
 1066 001264 175614
 1067 001266 175615
 1068 001270 175616
 1069 001272 000000
 1070 001274 000020
 1071 001334 000000
 1072 001336
 1073
 1074
 1075 001336 012706 001100
 1076 001342 005026
 1077 001344 022706 001140
 1078 001350 001374
 1079 001352 012706 001100
 1080
 1081 001356 012737 014304 000020
 1082 001364 012737 000340 000022
 1083 001372 012737 014104 000030
 1084 001400 012737 000340 000032
 1085 001406 012737 015236 000034
 1086 001414 012737 000340 000036
 1087 001422 012737 012416 000024
 1088 001430 012737 000340 000026
 1089 001436 016767 010666 010656
 1090 001444 005067 177510
 1091 001450 005067 177506
 1092 001454 112767 000001 177433
 1093 001462 012767 001462 177416
 1094 001470 012767 001470 177412
 1095
 1096
 1097 001476 013746 000004
 1098 001502 012737 001536 000004
 1099 001510 012767 177570 177422
 1100 001516 012767 177570 177416
 1101 001524 022777 177777 177406

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 28
 CVDVAB.P11 15-DEC-77 08:58 INITIALIZE THE COMMON TAGS

SEQ 0027

```

1102 001532 001012          BNE      66$          ;; BRANCH IF NO TIMEOUT TRAP OCCURRED
1103                                ;; AND THE HARDWARE SWR IS NOT = -1
1104 001534 000403          BR       65$          ;; BRANCH IF NO TIMEOUT
1105 001536 012716 001544    64$:  MOV     #65$, (SP) ;; SET UP FOR TRAP RETURN
1106 001542 000002          RTI
1107 001544 012767 000176 177366 65$:  MOV     #SWREG, SWR ;; POINT TO SOFTWARE SWR
1108 001552 012767 000174 177362  MOV     #DISPREG, DISPLAY
1109 001560 012637 000004    66$:  MOV     (SP)+, #ERRVEC ;; RESTORE ERROR VECTOR
1110
1111 001564 000567 177412          CLR     $PASS          ;; CLEAR PASS COUNT
1112 001570 132767 000200 177417  BITB     #APTSIZE, $ENVM ;; TEST USER SIZE UNDER APT
1113 001576 001403          BEQ      67$          ;; YES, USE NON-APT SWITCH
1114 001600 012767 001216 177332  MOV     #SSWREG, SWR ;; NO, USE APT SWITCH REGISTER
1115 001606
1116                                .SBTTL  TYPE PROGRAM NAME
1117                                ;; TYPE THE NAME OF THE PROGRAM IF FIRST PASS
1118 001606 0005227 177777          INC     #-1          ;; FIRST TIME?
1119 001612 001037          BNE      68$          ;; BRANCH IF NO
1120 001614 022737 012362 000042  CMP     #SENDAD, #42 ;; ACT-11?
1121 001622 001433          BEQ      68$          ;; BRANCH IF YES
1122 001624 104401 001672          TYPE     69$          ;; TYPE ASCIZ STRING
1123                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1124 001630 0005737 000042          TST     #42          ;; ARE WE RUNNING UNDER XXDP/ACT?
1125 001634 001012          BNE      70$          ;; BRANCH IF YES
1126 001636 126727 177352 000001  CMPB    $ENV, #1      ;; ARE WE RUNNING UNDER APT?
1127 001644 001406          BEQ      70$          ;; BRANCH IF YES
1128 001646 026727 177266 000176  CMP     SWR, #SWREG ;; SOFTWARE SWITCH REG SELECTED?
1129 001654 001005          BNE      71$          ;; BRANCH IF NO
1130 001656 104406          GTSWR          ;; GET SOFT-SWR SETTINGS
1131 001660 000403          BR       71$
1132 001662 112767 000001 177244 70$:  MOVB    #1, $AUTOB ;; SET AUTO-MODE INDICATOR
1133 001670          71$:
1134 001670 000410          BR       68$          ;; GET OVER THE ASCIZ
1135                                .ASCIZ  <CRLF>*MD-ZZ-CVDVA-B*<CRLF>
1136                                68$:
1137                                LET  INITFLAG := #1
1138 001712 012767 000001 010344    MOV     #1, INITFLAG
1139 001720          LOOP:
1140 001720          CALL  CYCLE          ; NO ARGUMENTS--ADDRS -> NEXT ADDRESS
1141 001720 004767 010210          JSR     PC, CYCLE
1142                                ;
1143                                ADDR+2 -> NEXT VECTOR
1144                                ; GET UNIT ADDRESS
1145 001724          MOV     (ADRS)+, DLADD      DLADD := (ADRS)+
1146                                ; GET UNIT VECTOR
1147 001730          MOV     (ADRS), DLVEC      DLVEC := (ADRS)
1148 001730 011167 177322          MOV     (ADRS), DLVEC
1149 001734          MOV     DLADD, ADRS      LET  ADRS := DLADD
1150 001734 016701 177314          MOV     DLADD, ADRS
1151                                ; RCSR = DLADD + 0
1152 001740          MOV     DLADD, RCSR      LET  RCSR := DLADD
1153 001740 016767 177310 177312  MOV     DLADD, RCSR
1154 001746          MOV     DLADD, RBUF      LET  RBUF := DLADD + #2
1155 001746 016767 177302 177306  MOV     #2, RBUF
1156 001754 062767 000002 177300  ADD
1157 001762          LET  TCSR := DLADD + #4

```


C03

MAINDEC-22-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 29
CVDVAB.P11 15-DEC-77 06:58 GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0028

1158	001762	016767	177266	177274	MOV	DLADD,TCSR		
1159	001770	062767	000004	177266	ADD	#4,TCSR		
1160	001776						LET	TCSRHI := DLADD + #5
1161	001776	016767	177252	177262	MOV	DLADD,TCSRHI		
1162	002004	062767	000005	177254	ADD	#5,TCSRHI		
1163	002012						LET	TBUF := DLADD + #6
1164	002012	016767	177236	177250	MOV	DLADD,TBUF		
1165	002020	062767	000006	177242	ADD	#6,TBUF		
1166	002026						LET	R5 := #RSSTACK
1167	002026	012705	001334		MOV	#RSSTACK,R5		
1168								::BRESET
1169	002032	000005			RESET			

```

1170 .....
1171 .....
1172 .....
1173 .....
1174 .....
1175 .....
1176 002034 000004 .....
1177 002036 012767 000002 177114 .....
1178 002044 012767 000001 177126 .....
1179 002052 .....
1180 002052 016701 177176 .....
1181 .....
1182 002056 .....
1183 002056 010146 .....
1184 002060 012701 000004 .....
1185 002064 012721 011742 .....
1186 002070 012711 000340 .....
1187 002074 012601 .....
1188 002076 .....
1189 002076 005067 177170 .....
1190 002102 .....
1191 002102 .....
1192 002102 .....
1193 002102 012767 002110 177000 .....
1194 .....
1195 002110 .....
1196 002110 005067 007634 .....
1197 .....
1198 .....
1199 002114 005711 .....
1200 002116 .....
1201 002116 005767 007626 .....
1202 002122 001401 .....
1203 .....
1204 002124 .....
1205 002124 104001 .....
1206 002126 .....
1207 002126 .....
1208 002126 .....
1209 002126 .....
1210 002126 062767 000002 177136 .....
1211 002134 .....
1212 002134 016701 177114 .....
1213 002140 066701 177126 .....
1214 002144 .....
1215 002144 026727 177122 000010 .....
1216 002152 001353 .....
1217 002154 .....
1218 002154 010146 .....
1219 002156 010246 .....
1220 002160 012701 000004 .....
1221 002164 010102 .....
1222 002166 062702 000002 .....
1223 002172 010221 .....
1224 002174 005011 .....
1225 002176 012602 .....

;*****
;TEST 1 ADDRESSABILITY
;THIS TEST VERIFIES THAT THE ADDRESS AS PLACED IN
;THE HARDWARE P-TABLE TO BE CORRECT AND THE DLV11-E RESPONDS
;TO THAT ADDRESS SPACE
;*****
TST1: SCOPE
      MOV #2, $TIMES ;DO 2 ITERATIONS
      MOV #1, $TESTN ;SET TEST NUMBER IN APT MAIL BOX
      LET ADRS := DLADD
      MOV DLADD, ADRS
      SETVEC ;SET UP INTERRUPT
      ;ILLMEM, #INTSRV, #PR7
      MOV R1, -(SP)
      MOV #ILLMEM, R1
      MOV #INTSRV, (R1)+
      MOV #PR7, (R1)
      MOV (SP)+, R1
      LET I := #0
      REPEAT
      BGNSUB
      ;CLEAR FLAG
      LET INTFLAG := #0
      ;READ FLAG
      IF INTFLAG NE #0 THEN
      ; FATAL ERROR
      ERRDF 1,,NODL
      ENDIF
      ENDSUB
      LET I := I + #2
      LET ADRS := DLADD + I
      UNTIL I EQ #8.
      CLRVEC ILLMEM
      ;PUSH R1 ON STACK
      ;PUSH R2 ON STACK
      MOV R1, -(SP)
      MOV R2, -(SP)
      MOV #ILLMEM, R1
      MOV R1, R2
      ADD #2, R2
      MOV R2, (R1)+
      CLR (R1)
      MOV (SP)+, R2 ;POP STACK INTO R2

```


E03

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 31
CVDVAB.P11 15-DEC-77 08:58 T1 ADDRESSABILITY

SEQ 0030

```
1226 002200 012601          MOV      (SP)+,R1          ;;POP STACK INTO R1
1227                                     ;;END OF TEST
1228 002202                                     ENDTST
1229 ;*****
1230 ;* THE FOLLOWING 8 TESTS TEST ALL 'READ WRITE' BITS
1231 ;*****
1232
1233
```

F03

 MAINDEC-22-CVDVA-B
 CVDVAB.P11 15-DEC-77 08:58

MACY11 30(1046)

 19-DEC-77 08:25 PAGE 32
 T2 BREAK - TCSR0 SET, CLEAR, RESET

SEQ 0031

```

1234 ;*****
1235 ;TEST 2 BREAK - TCSR0 SET, CLEAR, RESET
1236 ;* NOTE: THE (H) JUMPER MUST BE REMOVED FOR THIS
1237 ;* TEST TO FUNCTION PROPERLY.
1238 ;*****
1239 002202 000004 000010 176746 1ST2: SCOPE
1240 002204 012767 000010 176746 MOV #10,$TIMES ;;DO 10 ITERATIONS
1241 002212 012767 000002 176760 MOV #2,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
1242 ; SEE IF IT IS CLEAR
1243 BGNSUB
1244 002220 012767 002226 176662 MOV #64,$LPERR
1245 002220 012767 002226 176662 IF #BREAK SETIN @TCSR THEN
1246 002226 032777 000001 177030 BIT #BREAK,@TCSR
1247 002226 032777 000001 177030 BEQ $3
1248 002234 001401 000001 177030 ; BREAK DID NOT RESET IN TCSR
1249 002234 001401 000001 177030 ERRHRD 2,,DIDNOT
1250 002236 104002 000001 177030 ENDIF
1251 002236 104002 000001 177030 $3:
1252 002236 104002 000001 177030 ENDSUB
1253 002240 104002 000001 177030 ; TRY TO SET BREAK BIT
1254 002240 104002 000001 177030 BGNSUB
1255 002240 104002 000001 177030 MOV #64,$LPERR
1256 002240 104002 000001 177030 LET @TCSR := @TCSR SET.BY #BREAK
1257 002240 104002 000001 177030 BIS #BREAK,@TCSR
1258 002240 104002 000001 177030 IF ; STUCK TO 0
1259 002240 104002 000001 177030 IF #BREAK NOTSETIN @TCSR THEN
1260 002246 052777 000001 177010 BIT #BREAK,@TCSR
1261 002246 052777 000001 177010 BNE $4
1262 002254 032777 000001 177002 ; BREAK DID NOT SET IN TCSR
1263 002254 032777 000001 177002 ERRHRD 3,,DIDNOT
1264 002262 001001 000001 177002 $4:
1265 002262 001001 000001 177002 ENDSUB
1266 002264 104003 000001 177002 ; TRY TO CLEAR A SET BIT
1267 002264 104003 000001 177002 BGNSUB
1268 002264 104003 000001 177002 MOV #64,$LPERR
1269 002266 104003 000001 177002 LET @TCSR := @TCSR CLR.BY #BREAK
1270 002266 104003 000001 177002 BIC #BREAK,@TCSR
1271 002266 104003 000001 177002 IF ; SHOULD HAVE CLEARED
1272 002266 104003 000001 177002 IF #BREAK SETIN @TCSR THEN
1273 002274 042777 000001 176762 BIT #BREAK,@TCSR
1274 002274 042777 000001 176762 BEQ $5
1275 002274 042777 000001 176762 ; BREAK DID NOT CLEAR IN TCSR
1276 002274 042777 000001 176762 ERRHRD 4,,DIDNOT
1277 002302 032777 000001 176754 $5:
1278 002302 032777 000001 176754 ENDSUB
1279 002310 001401 000001 176754 ERROR 4
1280 002312 104004 000001 176754 ENDIF
1281 002312 104004 000001 176754 $5:
1282 002312 104004 000001 176754 ENDSUB
1283 002314 104004 000001 176754
1284 002314 104004 000001 176754
1285 002314 104004 000001 176754
1286 002314 104004 000001 176754
1287 002314 104004 000001 176754
1288 002314 104004 000001 176754
1289 002314 104004 000001 176754

```

G03

MAINDEC-22-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 33
CVDVAB.P11 15-DEC-77 08:58 T2 BREAK - TCSRO SET, CLEAR, RESET

SEQ 0032

; NOW SEE IF RESET CLEARS IT
BGNSUB

```
1290
1291 002314
1292 002314 012767 002322 176566      MOV      #64$, $LPERR
1293
1294 002322
1295 002322 052777 000001 176734      BIS      #BREAK, @TCSR      LET      @TCSR := @TCSR SET.BY #BREAK
1296
1297 002330
1298 002330 000005                      RESET      ; ISSUE BUS RESET
1299 002332
1300 002332 032777 000001 176724      BIT      #BREAK, @TCSR      BRESÉT
1301 002340 001401                      BEQ      $6              IF      #BREAK SET IN @TCSR THEN
1302
1303 002342
1304 002342 104005                      ERROR      5              ; BREAK DID NOT RESET IN TCSR
1305 002344
1306 002344                                ERRHRD 5,,DIDNOT
1307 002344                                ENDIF
1308 002344                                $6:
1309
1310
1311                                ENDST
                                ENDSUB
                                ENDTST
                                ; *****
```

H03

MAINDEC-22-CVDVA-B MACY11 30(1046)
CVDVAB.P11 15-DEC-77 08:58

19-DEC-77 08:25 PAGE 34
T3 MAINT - TCSR2 SET, CLEAR, RESET

SEQ 0033

```

1312 *****
1313 :TEST 3 MAINT - TCSR2 SET, CLEAR, RESET
1314 *****
1315 002344 000004 000010 176604 TST3: SCOPE
1316 002346 012767 000003 176616 MOV #10,$TIMES ;;DO 10 ITERATIONS
1317 002354 012767 000003 176616 MOV #3,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
1318
1319 ; SEE IF IT IS CLEAR
1320 002362 BGNSUB
1321 002362 012767 002370 176520 MOV #64$,$LPERR
1322
1323 002370 IF #MAINT SETIN @TCSR THEN
1324 002370 032777 000004 176666 BIT #MAINT,@TCSR
1325 002376 001401 $7 BEQ $7
1326 ; MAINT DID NOT RESET IN TCSR
1327 002400 ERRHRD 6,,DIDNOT
1328 002400 104006 ERROR 6
1329 002402 ENDF
1330 002402 $7: ENDSUB
1331 002402
1332 ; TRY TO SET MAINT BIT
1333 BGNSUB
1334 002402 012767 002410 176500 MOV #64$,$LPERR
1335 002402 052777 000004 176646 BIS #MAINT,@TCSR
1336 002410 LET @TCSR := @TCSR SET.BY #MAINT
1337 002410 IF : STUCK TO 0
1338 002416 IF #MAINT NOTSETIN @TCSR THEN
1339 002416 032777 000004 176640 BIT #MAINT,@TCSR
1340 002416 001001 $10 BNE $10
1341 ; MAINT DID NOT SET IN TCSR
1342 002426 ERRHRD 7,,DIDNOT
1343 002426 104007 ERROR 7
1344 002430 ENDF
1345 002430 $10: ENDSUB
1346 002430
1347 ; TRY TO CLEAR A SET BIT
1348 BGNSUB
1349 002430 012767 002436 176452 MOV #64$,$LPERR
1350 002436 LET @TCSR := @TCSR CLR.BY #MAINT
1351 002436 042777 000004 176620 BIC #MAINT,@TCSR
1352 IF : SHOULD HAVE CLEARED
1353 002444 IF #MAINT SETIN @TCSR THEN
1354 002444 032777 000004 176612 BIT #MAINT,@TCSR
1355 002452 001401 $11 BEQ $11
1356 ; MAINT DID NOT CLEAR INTCSR
1357 002454 ERRHRD 10,,DIDNOT
1358 002454 104010 ERROR 10
1359 002456 ENDF
1360 002456 $11: ENDSUB
1361 002456
1362 ; NOW SEE IF RESET CLEARS IT
1363 BGNSUB
1364 002456
1365
1366
1367

```


J03

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 36
 CVDVAB.P11 15-DEC-77 08:58 T4 XMITIE - TCSR6 SET, CLEAR, RESET

SEQ 0035

```

1389
1390
1391
1392 002506 000004
1393 002510 012767 000010 176442
1394 002516 012767 000004 176454
1395
1396 002524 012746 000340
1397 002530 012746 002536
1398 002534 000002
1399 002536
1400
1401
1402 002536
1403 002536 012767 002544 176344
1404
1405 002544
1406 002544 032777 000100 176512
1407 002552 001401
1408
1409 002554
1410 002554 104012
1411 002556
1412 002556
1413 002556
1414
1415
1416 002556
1417 002556 012767 002564 176324
1418 002564
1419 002564 052777 000100 176472
1420
1421 002572
1422 002572 032777 000100 176464
1423 002600 001001
1424
1425 002602
1426 002602 104013
1427 002604
1428 002604
1429 002604
1430
1431
1432 002604
1433 002604 012767 002612 176276
1434
1435 002612
1436 002612 042777 000100 176444
1437
1438 002620
1439 002620 032777 000100 176436
1440 002626 001401
1441
1442 002630
1443 002630 104014
1444 002632

;*****
;TEST 4 XMITIE - TCSR6 SET, CLEAR, RESET
;*****
TST4: SCOPE
      MOV #10,$TIMES ;;DO 10 ITERATIONS
      MOV #4,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
                          ; USE PRIORITY OF 7
      MOV #PR7,-(SP) ;;PUT NEW PS ON STACK
      MOV #64$,-(SP) ;;PUT NEW PC ON STACK
      RTI ;;POP NEW PC AND PS

64$:
      ; SEE IF IT IS CLEAR
      BGNSUB
      MOV #65$,$LPERR
      IF #XMITIE SETIN @TCSR THEN
      BIT #XMITIE,@TCSR
      BEQ $13
      ; XMITIE DID NOT RESET IN TCSR
      ERRHRD 12,,DIDNOT
      ERROR 12
      ENDIF
$13:
      ENDSUB
      ; TRY TO SET XMITIE BIT
      BGNSUB
      MOV #64$,$LPERR
      LET @TCSR := @TCSR SET.BY #XMITIE
      BIS #XMITIE,@TCSR
      IF ; STUCK TO 0
      IF #XMITIE NOTSETIN @TCSR THEN
      ; XMIT DID NOT RESET IN TCSR
      ERRHRD 13,,DIDNOT
      ERROR 13
      ENDIF
$14:
      ENDSUB
      ; TRY TO CLEAR A SET BIT
      BGNSUB
      MOV #64$,$LPERR
      LET @TCSR := @TCSR CLR.BY #XMITIE
      BIC #XMITIE,@TCSR
      IF ; SHOULD HAVE CLEARED
      IF #XMITIE SETIN @TCSR THEN
      ; XMIT DID NOT CLEAR IN TCSR
      ERRHRD 14,,DIDNOT
      ERROR 14
      ENDIF

```


K03

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 37
CVDVAB.P11 15-DEC-77 08:58 T4 XMITIE - TCSR6 SET, CLEAR, RESET

SEQ 0036

```
1445 002632 $15:
1446 002632
1447
1448 ; NOW SEE IF RESET CLEARS IT
1449 002632 012767 002640 176250 MOV #64$, $LPERR BGNSUB
1450 002632
1451 002640 052777 000100 176416 BIS #XMITIE, @TCSR LET @TCSR := @TCSR SET.BY #XMITIE
1452 002640
1453 002646 000005 RESET BRESE1 ; ISSUE BUS RESET
1454 002646
1455 002650 032777 000100 176406 BIT #XMITIE, @TCSR IF #XMITIE SETIN @TCSR THEN
1456 002650
1457 002656 001401 BEQ $16 ; XMIT DID NOT RESET IN TCSR
1458 002656
1459 002660 104015 ERROR 15 ERRHRD 15,,DIDNOT
1460 002660
1461 002662 $16:
1462 002662
1463 002662
1464 002662
1465 002662
1466 002662
1467
1468
1469
1470 ;*****
```

```

1471 *****
1472 :TEST 5      DTR - RCSR1  SET, CLEAR
1473 :           NOTE: RESET DOES NOT CLEAR THIS BIT
1474 :           WE CANNOT TEST FOR AN INITIAL CONDITION
1475 :           AS THIS BIT IS UNDEFINED UPON POWER UP AND
1476 :           INIT DOESN'T AFFECT IT.
1477 :           THE (-FD) JUMPER MUST BE IN FOR THIS TEST TO WORK.
1478 *****
1479 002662 000004      TS5:  SCOPE
1480 002664 012767 000010 176266      MOV  #10,$TIMES      ;;DO 10 ITERATIONS
1481 002672 012767 000005 176300      MOV  #5,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1482 002700                                IF #FRFD NOTSETIN $USWR THEN
1483 002700 032767 040000 176312      BIT  #FRFD,$USWR
1484 002706 001004      BNE  $17
1485 002710                                EXIT TST
1486 002710 012767 000001 176242      MOV  #1,$TIMES
1487 002716 000441      BR   TST6      ;;;EXIT THIS TEST
1488 002720                                ENDF
1489 002720      $17
1490                                ; TRY TO CLEAR DTR BIT
1491 002720                                BGNSUB
1492 002720 012767 002726 176162      MOV  #64,$SLPERR
1493 002726 042777 000002 176324      BIC  #DTR,$RCSR      LET  @RCSR := @RCSR CLR.BY #DTR
1494 002734                                ; STUCK TO 0
1495 002734 032777 000002 176316      BIT  #DTR,$RCSR      IF  #DTR SETIN @RCSR THEN
1496 002742 001401      BEQ  $20
1497                                ; DTR DID NOT CLEAR IN RCSR
1498                                ERRHRD 16,,DIDNOT
1499 002744      ERROR  16
1500 002744 104016      ENDF
1501 002746      $20:
1502 002746      ENDSUB
1503 002746
1504 002746
1505                                ; TRY TO SET DTR
1506 002746                                BGNSUB
1507 002746 012767 002754 176134      MOV  #64,$SLPERR
1508 002754                                LET  @RCSR := @RCSR SET.BY #DTR
1509 002754 052777 000002 176276      BIS  #DTR,$RCSR      IF  #DTR NOTSETIN @RCSR THEN
1510 002762                                ; DTR DID NOT SET IN RCSR
1511 002762 032777 000002 176270      BIT  #DTR,$RCSR      ERRHRD 17,,DIDNOT
1512 002770 001001      BNE  $21
1513                                ENDF
1514 002772      ERROR  17
1515 002772 104017      ENDF
1516 002774      $21:
1517 002774      ENDSUB
1518 002774
1519 002774
1520 002774
1521                                ; TRY TO CLEAR IT AGAIN
1522 002774                                BGNSUB
1523 002774 012767 003002 176106      MOV  #64,$SLPERR
1524 003002                                LET  @RCSR := @RCSR CLR.BY #DTR
1525 003002 042777 000002 176250      BIC  #DTR,$RCSR
1526 003002

```


MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 39
CVDVA8.P11 15-DEC-77 08:58 TS DTR - RCSR1 SET, CLEAR

SEQ 0038

```
1527
1528 003010
1529 003010 032777 000002 176242 BIT #DTR, @RCSR
1530 003016 001401 BEQ $22
1531
1532 003020
1533 003020 104020 ERROR 20
1534 003022
1535 003022 $22:
1536 003022
1537 003022
1538
1539
1540
1541
```

IF ; SHOULD HAVE CLEARED IT
#DTR SET IN @RCSR THEN

; DTR DID NOT CLEAR IN RCSR
ERRHRD 20,,DIDNOT

ENDIF

ENDSUB

ENDTST

;;*****

MAINDEC-22-CVDVA-B MACY11 30(1046)
CVDVAB.P11 15-DEC-77 08:58

19-DEC-77 08:25 PAGE 40
T6 REQSEND - RCSR2

SET, CLEAR, RESET

SEQ 0039

```

1542 *****
1543 ;TEST 6 REQSEND - RCSR2 SET, CLEAR, RESET
1544 ;* THIS TEST ASSUMES THAT JUMPER -(FR) IS IN
1545 *****
1546 $T6: SCOPE
1547 003022 000004 000010 176126 MOV #10,$TIMES ;;DO 10 ITERATIONS
1548 003024 012767 000006 176140 MOV #6,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
1549 003032 012767 000006 176140 IF #FRFD NOTSETIN $USWP THEN
1550 003040 032767 040000 176152 BIT #FRFD,$USWR
1551 003046 001004 000000 176152 BNE $23
1552 003050 000000 000000 176152 EXIT TST
1553 003050 012767 000001 176102 MOV #1,$TIMES
1554 003056 000452 000000 176102 BR TST7 ;;EXIT THIS TEST
1555 003060 000000 000000 176102 ENDF
1556 003060 $23:
1557 ; SEE IF IT IS CLEAR
1558 BGHSUB
1559 003060 012767 003066 176022 MOV #64,$SLPERR
1560 003060 012767 003066 176022 IF #REQSEND SETIN @RCSR THEN
1561 003066 032777 000004 176164 BIT #REQSEND,@RCSR
1562 003066 001401 000004 176164 BEQ $24
1563 003074 001401 000004 176164 ; REQSEND DID NOT RESET IN RCSR
1564 003074 001401 000004 176164 ERRHRD 21,,DIDNOT
1565 003076 104021 000000 176164 ENDF
1566 003076 104021 000000 176164 $24:
1567 003100 104021 000000 176164 BGHSUB
1568 003100 104021 000000 176164 ENDSUB
1569 003100 104021 000000 176164 ; TRY TO SET REQSEND BIT
1570 003100 104021 000000 176164 BGHSUB
1571 003100 012767 003106 176002 MOV #64,$SLPERR
1572 003100 012767 003106 176002 LET @RCSR := @RCSR SET.BY #REQSEND
1573 003106 052777 000004 176144 BIS #REQSEND,@RCSR
1574 003106 052777 000004 176144 IF ; STUCK TO 0
1575 003114 032777 000004 176136 BIT #REQSEND,@RCSR
1576 003114 001001 000004 176136 BNE $25
1577 003122 001001 000004 176136 ; REQSEND DID NOT SET IN RCSR
1578 003122 001001 000004 176136 ERRHRD 22,,DIDNOT
1579 003124 104022 000000 176136 ENDF
1580 003124 104022 000000 176136 $25:
1581 003126 104022 000000 176136 BGHSUB
1582 003126 104022 000000 176136 ENDSUB
1583 003126 104022 000000 176136 ; TRY TO CLEAR A SET BIT
1584 003126 104022 000000 176136 BGHSUB
1585 003126 012767 003134 175754 MOV #64,$SLPERR
1586 003126 012767 003134 175754 LET @RCSR := @RCSR CLR.BY #REQSEND
1587 003134 042777 000004 176116 BIC #REQSEND,@RCSR
1588 003134 042777 000004 176116 IF ; SHOULD HAVE CLEARED
1589 003142 032777 000004 176110 BIT #REQSEND,@RCSR
1590 003142 001401 000004 176110 BEQ $26
1591 003150 001401 000004 176110
1592 003150 001401 000004 176110
1593 003150 001401 000004 176110
1594 003150 001401 000004 176110
1595 003150 001401 000004 176110
1596 003150 001401 000004 176110
1597 003150 001401 000004 176110

```

```

1598                                     ; REQSEND DID NOT CLEAR IN RCSR
1599 003152                                     ERRHRD 23,,DIDNOT
1600 003152 104023          ERROR 23
1601 003154                                     ENDIF
1602 003154 $26:                                     ENDSUB
1603 003154                                     ; NOW SEE IF RESET CLEARS IT
1604                                     BGNSUB
1605                                     ;
1606 003154 012767 003162 175726          MOV      #64$, $LPERR
1607 003154                                     LET      @RCSR := @RCSR SET.BY #REQSEND
1608                                     ; ISSUE BUS RESET
1609 003162 052777 000004 176070          BIS      #REQSEND, @RCSR
1610 003162                                     BRESET
1611                                     IF      #REQSEND SETIN @RCSR THEN
1612 003170                                     ;
1613 003170 000005          RESET
1614 003172                                     ; REQSEND DID NOT RESET IN RCSR
1615 003172 032777 000004 176060          BIT      #REQSEND, @RCSR
1616 003200 001401          BEQ      $27
1617                                     ERRHRD 24,,DIDNOT
1618 003202                                     ENDIF
1619 003202 104024          ERROR 24
1620 003204                                     ENDSUB
1621 003204 $27:                                     ENDTST
1622 003204
1623
1624
1625
1626
1627 ;*****

```

C04

MAINDEC-22-CVDVA-B
CVDVAB.P11MACY11 30(1046)
15-DEC-77 08:5819-DEC-77 08:25 PAGE 42
T7 SECXMIT - RCSR3

SET, CLEAR, RESET

SEQ 0041

```

1628      ;*****
1629      ;TEST 7      SECXMIT - RCSR3      SET, CLEAR, RESET
1630      ;*****
1631      ST7:  SCOPE
1632      MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1633      MOV     #7,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1634      ; SEE IF IT IS CLEAR
1635      BGNSUB
1636      MOV     #64$,$LPERR
1637
1638      IF      #SECXMIT SETIN @RCSR THEN
1639      BIT     #SECXMIT,@RCSR
1640      BEQ     $30
1641      ; SECXMIT DID NOT RESET IN RCSR
1642      ERRHRD 25,,DIDNOT
1643      ERROR   25
1644      ENDF
1645      $30:
1646      ENDSUB
1647
1648      ; TRY TO SET SECXMIT BIT
1649      BGNSUB
1650      MOV     #64$,$LPERR
1651      LET     @RCSR := @RCSR SET.BY #SECXMIT
1652      BIS     #SECXMIT,@RCSR
1653      ; STUCK TO 0
1654      IF      #SECXMIT NOTSETIN @RCSR THEN
1655      BIT     #SECXMIT,@RCSR
1656      BNE     $31
1657      ; SECXMIT DID NOT SET IN RCSR
1658      ERRHRD 26,,DIDNOT
1659      ERROR   26
1660      ENDF
1661      $31:
1662      ENDSUB
1663
1664      ; TRY TO CLEAR A SET BIT
1665      BGNSUB
1666      MOV     #64$,$LPERR
1667
1668      LET     @RCSR := @RCSR CLR.BY #SECXMIT
1669      BIC     #SECXMIT,@RCSR
1670      ; SHOULD HAVE CLEARED
1671      IF      #SECXMIT SETIN @RCSR THEN
1672      BIT     #SECXMIT,@RCSR
1673      BEQ     $32
1674      ; SECXMIT DID NOT CLEAR IN RCSR
1675      ERRHRD 27,,DIDNOT
1676      ERROR   27
1677      ENDF
1678      $32:
1679      ENDSUB
1680      BGNSUB
1681      MOV     #64$,$LPERR
1682      ; NOW SEE IF RESET CLEARS IT
1683

```

004

MAINDEC-22-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 43
 CVDVAB.P11 15-DEC-77 08:58 T7 SECXMIT - RCSR3 SET, CLEAR, RESET SEQ 0042

```

1684
1685 003324
1686 003324 052777 000010 175726 BIS #SECXMIT, @RCSR LET @RCSR := @RCSR SET.BY #SECXMIT
1687
1688 003332
1689 003332 000005 RESET BRESÉT : ISSUE BUS RESET
1690 003334
1691 003334 032777 000010 175716 BIT #SECXMIT, @RCSR IF #SECXMIT SETIN @RCSR THEN
1692 003342 001401 BEQ $33 : SECXMIT DID NOT RESET IN RCSR
1693
1694 003344
1695 003344 104030 ERROR 30 ERRHRD 30,,DIDNOT
1696 003346
1697 003346 $33:
1698 003346
1699 003346
1700
1701
1702
1703 ;*****
  
```

ENDSUB
 ENDTST

E04

MAINDEC-ZZ-CVDVA-B MACY11 30(1046)
CVDVAB.P11 15-DEC-77 08:5819-DEC-77 08:25 PAGE 44
T10 DATAIE - RCSRS SET, CLEAR, RESET

SEQ 0043

```

1704 ;*****
1705 ;*TEST 10 DATAIE - RCSRS SET, CLEAR, RESET
1706 ;*****
1707 003346 000004 TST10: SCOPE
1708 003350 012767 000010 175602 MOV #10,$TIMES ;DO 10 ITERATIONS
1709 003356 012767 000010 175614 MOV #10,$TESTN ;SET TEST NUMBER IN APT MAIL BOX
1710 ; SEE IF IT IS CLEAR
1711 003364 BGNSUB
1712 003364 012767 003372 175516 MOV #64$,$LPERR
1713 IF #DATAIE SETIN @RCSR THEN
1714 003372
1715 003372 032777 000040 175660 BIT #DATAIE,@RCSR
1716 003400 001401 BEQ $34
1717 ; DATAIE DID NOT RESET IN RCSR
1718 003402 ERRHRD 31,,DIDNOT
1719 003402 104031 ERROR 31
1720 003404
1721 003404 $34:
1722 003404 ENDSUB
1723 ; TRY TO SET DATAIE BIT
1724 BGNSUB
1725 003404
1726 003404 012767 003412 175476 MOV #64$,$LPERR
1727 003412 LET @RCSR := @RCSR SET.BY #DATAIE
1728 003412 052777 000040 175640 BIS #DATAIE,@RCSR
1729 ; STUCK TO 0
1730 003420 IF #DATAIE NOTSETIN @RCSR THEN
1731 003420 032777 000040 175632 BIT #DATAIE,@RCSR
1732 003426 001001 BNE $35
1733 ; DATAIE DID NOT SET IN RCSR
1734 003430 ERRHRD 32,,DIDNOT
1735 003430 104032 ERROR 32
1736 003432
1737 003432 $35:
1738 003432 ENDSUB
1739 ; TRY TO CLEAR A SET BIT
1740 BGNSUB
1741 003432
1742 003432 012767 003440 175450 MOV #64$,$LPERR
1743 LET @RCSR := @RCSR CLR.BY #DATAIE
1744 003440
1745 003440 042777 000040 175612 BIC #DATAIE,@RCSR
1746 ; SHOULD HAVE CLEARED
1747 003446 IF #DATAIE SETIN @RCSR THEN
1748 003446 032777 000040 175604 BIT #DATAIE,@RCSR
1749 003454 001401 BEQ $36
1750 ; DATAIE DID NOT CLEAR IN RCSR
1751 003456 ERRHRD 33,,DIDNOT
1752 003456 104033 ERROR 33
1753 003460
1754 003460 $36:
1755 003460 ENDSUB
1756 ; NOW SEE IF RESET CLEARS IT
1757 BGNSUB
1758 003460
1759 003460 012767 003466 175422 MOV #64$,$LPERR

```


F04

MAINDEC-22-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 45
CVDVAB.P11 15-DEC-77 08:58 T10 DATAIE - RCSR5 SET, CLEAR, RESET

SEQ 0044

```
1760
1761 003466
1762 003466 052777 000040 175564 BIS #DATAIE, @RCSR LET @RCSR := @RCSR SET.BY #DATAIE
1763 : ISSUE BUS RESET
1764 003474 BRESET
1765 003474 000005 RESET
1766 003476 IF #DATAIE SET IN @RCSR THEN
1767 003476 032777 000040 175554 BIT #DATAIE, @RCSR
1768 003504 001401 BEQ $37
1769 : DATAIE DID NOT RESET IN RCSR
1770 003506 ERRHRD 34,, DIDNOT
1771 003506 104034 ERROR 34
1772 003510
1773 003510 $37:
1774 003510
1775 003510
1776
1777
1778
1779 ;*****
```

ENDSUB
ENDTST

G04

 MAINDEC-ZZ-CVDVA-B
 CVDVAB.P11 15-DEC-77 08:58

 MACY11 30(1046)
 T11

 19-DEC-77 08:25 PAGE 46
 RCVRIE - RCSR6 SET, CLEAR, RESET

SEQ 0045

```

1780
1781
1782
1783 003510 000004
1784 003512 012767 000010 175440
1785 003520 012767 000011 175452
1786
1787 003526
1788 003526 012767 003534 175354
1789
1790 003534
1791 003534 032777 000100 175516
1792 003542 001401
1793
1794 003544
1795 003544 104035
1796 003546
1797 003546
1798 003546
1799
1800
1801 003546
1802 003546 012767 003554 175334
1803 003554
1804 003554 052777 000100 175476
1805
1806 003562
1807 003562 032777 000100 175470
1808 003570 001001
1809
1810 003572
1811 003572 104036
1812 003574
1813 003574
1814 003574
1815
1816
1817 003574
1818 003574 012767 003602 175306
1819
1820 003602
1821 003602 042777 000100 175450
1822
1823 003610
1824 003610 032777 000100 175442
1825 003616 001401
1826
1827 003620
1828 003620 104037
1829 003622
1830 003622
1831 003622
1832
1833
1834 003622
1835 003622 012767 003630 175260

;*****
;TEST 11 RCVRIE - RCSR6 SET, CLEAR, RESET
;*****
↑ST11: SCOPE
      MOV #10,$TIMES      ;;DO 10 ITERATIONS
      MOV #11,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                          ; SEE IF IT IS CLEAR
                          BGNSUB
      MOV #64,$SLPERR
      IF #RCVRIE SETIN @RCSR THEN
      BIT #RCVRIE,@RCSR
      BEQ $40
      ; RCVRIE DID NOT RESET IN RCSR
      ERRHRD 35,,DIDNOT
      ENDIF
$40:
      ENDSUB
      ; TRY TO SET RCVRIE BIT
      BGNSUB
      MOV #64,$SLPERR
      BIS #RCVRIE,@RCSR
      LET @RCSR := @RCSR SET.BY #RCVRIE
      ; STUCK TO 0
      IF #RCVRIE NOTSETIN @RCSR THEN
      ; RCVRIE DID NOT SET IN RCSR
      ERRHRD 36,,DIDNOT
      ENDIF
$41:
      ENDSUB
      ; TRY TO CLEAR A SET BIT
      BGNSUB
      MOV #64,$SLPERR
      BIC #RCVRIE,@RCSR
      LET @RCSR := @RCSR CLR.BY #RCVRIE
      ; SHOULD HAVE CLEARED
      IF #RCVRIE SETIN @RCSR THEN
      ; RCVRIE DID NOT CLEAR IN RCSR
      ERRHRD 37,,DIDNOT
      ENDIF
$42:
      ENDSUB
      ; NOW SEE IF RESET CLEARS IT
      BGNSUB
      MOV #64,$SLPERR

```


MAINDEC-22-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 47
 CVDVAB.P11 15-DEC-77 08:58 T11 RCVRIE - RCSR6 SET, CLEAR, RESET

SEQ 0046

```

1836
1837 003630
1838 003630 052777 000100 175422 BIS #RCVRIE,@RCSR LET @RCSR := @RCSR SET.BY #RCVRIE
1839
1840 003636
1841 003636 000005 RESET ; ISSUE BUS RESET
1842 003640
1843 003640 032777 000100 175412 BIT #RCVRIE,@RCSR BRES@T
1844 003646 001401 BEQ $43 IF #RCVRIE SET IN @RCSR THEN
1845
1846 003650
1847 003650 104040 ERROR 40 ; RCVRIE DID NOT RESET IN RCSR
1848 003652 ERRHRD 40,,DIDNOT
1849 003652 $43:
1850 003652
1851 003652
1852 003652
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862

```

CKLOOP
 ENDSUB
 ENDTST

```

;*****
; THE FOLLOWING 4 TESTS VERIFY
; THAT RESET (INIT) INITIALIZES READ ONLY BITS.
;*****

```

ENDSUB
ENDTST

J04

MAINDEC-ZZ-CVDVA-B MACY11 30(1046)
CVDVAB.P11 15-DEC-77 08:58

19-DEC-77 08:25 PAGE 49
T13 TEST THAT XMITRDY - TCSR 7 - IS SET BY INIT

SEQ 0048

```

1896      ;*****
1897      ;TEST 13      TEST THAT XMITRDY - TCSR 7 - IS SET BY INIT
1898      ;*****
1899      TST13:  SCOPE
1900      MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1901      MOV     #13,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1902
1903
1904
1905
1906      003730      BGNSUB
1907      003730      012767  003736  175152      MOV     #64$,$LPERR
1908
1909      003736      IF      #XMITRDY NOTSETIN 2TCSR THEN
1910      003736      032777  000200  175320      BIT     #XMITRDY,2TCSR
1911      003744      001002      BNE     $45
1912
1913      ;RESET SHOULD HAVE SET BIT.
1914      ;XMITRDY DID NOT SET IN TCSR (AFTER RESE
1915      003746      ERRHRD 42,HRESET,DIDNOT
1916      003746      104042      ERROR    42
1917
1918      003750      ;ISSUE ANOTHER RESET
1919      003750      BRESET
1920      003752      ENDIF
1921      003752      $45:
1922
1923      003752      ;ALLOW LOOPING ON ERROR
1924      003752      CKLOOP
1925      003752      ENDSUB
1926
1927      ENDTST
1928

```

K04

MAINDEC-ZZ-CVDVA-B MACY11 30(1046)
 CVDVAB.P11 15-DEC-77 08:58

19-DEC-77 08:25 PAGE 50
 T14 TEST THAT DATAINT - RCSR 15 - IS CLEARED BY INIT.

SEQ 0049

```

1929      ;*****
1930      ;*TEST 14      TEST THAT DATAINT - RCSR 15 - IS CLEARED BY INIT.
1931      ;*****
1932      ST14:  SCOPE
1933      MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1934      MOV     #14,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1935
1936
1937
1938
1939      003770                                BGNSUB
1940      003770      012767      003776      175112      MOV     #64$,$LPERR
1941      003776                                IF      #DATAINT SETIN RCSR THEN
1942      003776      032777      100000      175254      BIT     #DATAINT,RCSR
1943      004004      001402                                BEQ     $46
1944
1945      004006                                ERRHRD 43, HRESET, DIDNOT
1946      004006      104043      ERROR      43
1947
1948                                ;TESTING EFFECT OF RESET ON BIT
1949
1950                                ;DATAINT DID NOT CLEAR IN RCSR
1951                                ;ALLOW A FRESH START
1952                                BRESET
1953      004010      000005      RESET
1954      004012                                ENDIF
1955      004012      $46:
1956      004012                                CKLOOP
1957      004012                                ENDSUB
1958      004012                                ENDTST
1959
1960
1961      ;*****
1962

```

L04

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 51
CVDVAB.P11 15-DEC-77 08:58

T15 TEST THAT RCVRACT - RCSR 11 - 15 CLEARED BY INIT

SEQ 0050

```

1963
1964
1965
1966 004012 000004
1967 004014 012767 000010 175136
1968 004022 012767 000015 175150
1969
1970
1971 004030
1972 004030 032767 020000 175162
1973 004036 001004
1974
1975 004040
1976 004040 012767 000001 175112
1977 004046 000411
1978 004050
1979 004050
1980
1981
1982
1983 004050
1984 004050 012767 004056 175032
1985
1986 004056
1987 004056 032777 004000 175174
1988 004064 001402
1989
1990
1991 004066
1992 004066 104044
1993
1994
1995
1996
1997
1998
1999 004070
2000 004070 000005
2001 004072
2002 004072
2003
2004 004072
2005 004072
2006 004072
2007

;*****
;TEST 15 TEST THAT RCVRACT - RCSR 11 - 15 CLEARED BY INIT
;*****
T15: SCOPE
MOV #10,$TIMES ;;DO 10 ITERATIONS
MOV #15,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX

IF #CABLE NOTSETIN $USWR THEN
    BIT #CABLE,$USWR
    BNE $47
    EXIT TST
; CAN'T TEST WITHOUT BERG OR H315.
;;EXIT THIS TEST
ENDIF

$47:
    MOV #64,$SLPERR
    BGNSUB
    IF #RCVRACT SETIN @RCSR THEN
        ;RESET SHOULD HAVE CLEARED RCVRACT
        ;ERRWRD 44, HRESET, DIDNOT
        ERROR 44
        ;TESTING EFFECT OF RESET ON BIT
        ;RCVRACT DID NOT CLEAR IN RCSR
        ;ALLOW ANOTHER TRY
        BRESET
    ENDIF
    ;ALLOW LOOPING ON ERROR
    CKLOOP
    ENDSUB
ENDTST

```

M04

MAINDEC-22-CVDVA-B MACY11 30(1046)
CVDVAB.P11 15-DEC-77 08:58

19-DEC-77 08:25 PAGE 52
TIS TEST THAT RCVRCT - RCSR 11 - 15 CLEARED BY INIT

SEQ 0051

2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063

004072 000004
004074 012767 000010 175056
004102 012767 000016 175070

004110
004110 032767 060000 175102
004116 001004

004120
004120 012767 000001 175032
004126 000441
004130
004130

004130
004130 012767 004136 174752

004136
004136 042777 000002 175114

004144
004144 032777 010000 175106
004152 001401

004154
004154 104045

004156
004156
004156

004156

```

*****
* THE FOLLOWING 4 TESTS VERIFY
* THAT THE EIA SIGNALS CAN BE TRANSMITTED
* AND RECEIVED THROUGH THE CABLE
*****

*****
* TEST 16 TEST THAT CARDET SETS AND CLEARS
* AS DTR SETS AND CLEARS
* THE (-FD) JUMPER MUST BE IN FOR THIS TEST.
*****
TEST16: SCOPE
MOV #10,STIMES ;;DO 10 ITERATIONS
MOV #16,STESTN ;;SET TEST NUMBER IN APT MAIL BOX
IF #CABLE+FRFD NOTSETIN SUSWR THEN
; CAN WE USE THE WRAPAROUND??
; CAN'T TEST WITHOUT BERG OR H315
; OR WITH (-FD) JUMPER OUT.
; OR WITH (-FR) JUMPER OUT.
EXIT TEST
MOV #1,STIMES
BR TS17 ;;EXIT THIS TEST
ENDIF
$S1:
; DTR AND
; CARDET ARE CONNECTED
; BY THE H315 OR EQUIV.
; CLEAR
BGNSUB
MOV #64$,SLPERR
LET ; CLEAR DTR
; RCSR := RCSR CLR.BY #DTR
BIC #DTR,RCSR
; CARDET SHOULD FOLLOW
IF #CARDET SETIN RCSR THEN
; CARDET DID NOT
ERRHRD 45,,FORCE
; CLEAR WITH DTR
ENDIF
$S2:
ENDSUB
; SET
BGNSUB

```


N04

MAINDEC-22-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 53
CVDVAB.P11 15-DEC-77 08:58 T16 TEST THAT CARDET SETS AND CLEARS

SEQ 0052

2064	004156	012767	004164	174724	MOV	#64\$, \$LPERR	
2065							
2066							
2067	004164						LET ; SET DTR
2068	004164	052777	000002	175066	BIS	#DTR, @RCSR	@RCSR := @RCSR SET.BY #DTR
2069							
2070	004172						IF ; CARDET SHOULD FOLLOW
2071	004172	032777	010000	175060	BIT	#CARDET, @RCSR	#CARDET NOTSETIN @RCSR THEN
2072	004200	001001			BNE	\$53	
2073							; CARDET DID NOT SET
2074	004202						ERRHRD 46,, FORCE
2075	004202	104046			ERROR	46	
2076							
2077							ENDIF ; WITH DTR
2078	004204						
2079	004204						ENDSUB
2080	004204						
2081							
2082							; CLEAR
2083	004204						BGNSUB
2084	004204	012767	004212	174676	MOV	#64\$, \$LPERR	
2085							
2086							
2087	004212						LET ; CLEAR DTR
2088	004212	042777	000002	175040	BIC	#DTR, @RCSR	@RCSR := @RCSR CLR.BY #DTR
2089							
2090	004220						IF ; CARDET SHOULD FOLLOW
2091	004220	032777	010000	175032	BIT	#CARDET, @RCSR	#CARDET SETIN @RCSR THEN
2092	004226	001401			BEQ	\$54	
2093							; CARDET DID NOT
2094	004230						ERRHRD 47,, FORCE
2095	004230	104047			ERROR	47	
2096							
2097							ENDIF ; CLEAR WITH DTR
2098	004232						
2099	004232						ENDSUB
2100	004232						ENDTST
2101	004232						
2102							
2103							
2104							
2105							

```

2106 ;*****
2107 ;*****
2108 ;TEST 17 TEST THAT CLREND SETS AND CLEARS
2109 ; AS DTR SETS AND CLEARS
2110 ; (-FD) JUMPER MUST BE IN FOR THIS TEST TO WORK
2111 ;*****
2112 ST17: SCOPE
2113 MOV #10,$TIMES ; DO 10 ITERATIONS
2114 MOV #17,$TESTN ; SET TEST NUMBER IN APT MAIL BOX
2115 ; CAN WE USE THE WRAPAROUND??
2116 IF #CABLE+FRFD NOTSETIN $USWR THEN
2117 004232 000004 000010 174716 BIT #CABLE+FRFD,$USWR
2118 004234 012767 000017 174730 BNE $55
2119 ; CAN'T TEST WITHOUT BERG OR H315
2120 ; EXIT TEST
2121 004260 012767 000001 174672 MOV #1,$TIMES
2122 004266 000441 TST20 ; EXIT THIS TEST
2123 004270 ENDIF
2124 004270 $55:
2125 ;
2126 ; DTR AND
2127 ; CLREND ARE CONNECTED
2128 ; BY THE H315 OR EQUIV.
2129 ;
2130 ; CLEAR
2131 ; BGNSUB
2132 004270 012767 004276 174612 MOV #64,$LPERR
2133 004276 042777 000002 174754 BIC #DTR,$RCSR
2134 ; CLEAR DTR
2135 LET $RCSR := $RCSR CLR.BY #DTR
2136 004304 032777 020000 174746 BIT #CLREND,$RCSR
2137 004312 001401 BEQ $56 ; CLREND SHOULD FOLLOW
2138 ; #CLREND SETIN $RCSR THEN
2139 ; CLREND DID NOT
2140 ; ERRHRD SO,,FORCE
2141 004314 104050 ERROR 50
2142 ; CLEAR WITH DTR
2143 ENDIF
2144 004316 $56:
2145 004316 ENDSUB
2146 004316 ; SET
2147 ; BGNSUB
2148 004316 012767 004324 174564 MOV #64,$LPERR
2149 004324 052777 000002 174726 BIS #DTR,$RCSR
2150 ; SET DTR
2151 LET $RCSR := $RCSR SET.BY #DTR
2152 004332 032777 020000 174720 BIT #CLREND,$RCSR
2153 004340 001001 BNE $57 ; CLREND SHOULD FOLLOW
2154 ; #CLREND NOTSETIN $RCSR THEN
2155 004332 032777 020000 174720 BIT #CLREND,$RCSR
2156 004340 001001 BNE $57
2157
2158
2159
2160
2161

```



```

2162                                     ; CLRSEND DID NOT SET
2163 004342                                     ERRHRD 51,,FORCE
2164 004342 104051                     ERROR 51
2165
2166                                     ; WITH DTR
2167 004344                     ENDIF
2168 004344                     $57:
2169 004344                     ENDSUB
2170
2171                                     ; CLEAR
2172 004344                     BGNSUB
2173 004344 012767 004352 174536      MOV      #64$, $LPERR
2174
2175                                     ; CLEAR DTR
2176 004352                     LET      @RCSR := @RCSR CLR.BY #DTR
2177 004352 042777 000002 174700      BIC      #DTR, @RCSR
2178
2179                                     ; CLRSEND SHOULD FOLLOW
2180 004360 032777 020000 174672      BIT      #CLRSEND, @RCSR
2181 004366 001401                     BEQ      $60
2182
2183                                     ; CLRSEND DID NOT
2184 004370 104052                     ERROR 52
2185
2186                                     ; CLEAR WITH DTR
2187 004372                     ENDIF
2188 004372                     $60:
2189 004372                     ENDSUB
2190 004372                     ENDTST
2191
2192
2193
2194

```


E05

MAINDEC-22-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 57
CVDVAB.P11 15-DEC-77 08:58 T20 TEST THAT RING SETS AND CLEARS

SEQ 0056

```

2251 004500 001001          BNE      $63
2252
2253 004502                  ; RING DID NOT SET
2254 004502 104054          ERROR    54      ERRHRD 54,,FORCE
2255
2256                                ; WITH REQSEND
2257 004504                  ENDIF
2258 004504          $63:
2259 004504                  ENDSUB
2260
2261                                ; CLEAR
2262 004504                  BGMSUB
2263 004504 012767 004512 174376      MOV      #64$,$LPERR
2264
2265                                ; CLEAR REQSEND
2266 004512                  LET      @RCSR := @RCSR CLR.BY #REQSEND
2267 004512 042777 000004 174540      BIC      #REQSEND,@RCSR
2268
2269                                ; RING SHOULD FOLLOW
2270 004520 032777 040000 174532      BIT      #RING,@RCSR
2271 004526 001401          BEQ      $64      IF      #RING SET IN @RCSR THEN
2272
2273                                ; RING DID NOT
2274 004530 104055          ERROR    55      ERRHRD 55,,FORCE
2275
2276                                ; CLEAR WITH REQSEND
2277 004532                  ENDIF
2278 004532          $64:
2279 004532                  ENDSUB
2280 004532                  ENDTST
2281
2282
2283
2284

```

F05

MAINDEC-ZZ-CVDVA-B MACY11 30(1046)
CVDVAB.P11 15-DEC-77 08:58

19-DEC-77 08:25 PAGE 58
T20 TEST THAT RING SETS AND CLEARS

SEQ 0057

```

2285      ;*****
2286      ;*****
2287      ;TEST 21      TEST THAT SECREC SETS AND CLEARS
2288      ;*          AS SECXMIT SETS AND CLEARS
2289      ;*****
2290      ;ST21: SCOPE
2291      ;MOV      #10,$TIMES      ;;DO 10 ITERATIONS
2292      ;MOV      #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2293      ;          ; CAN WE USE THE WRAPAROUND??
2294      ;          IF      #CABLE NOTSETIN $USWR THEN
2295      ;          ; CAN'T TEST WITHOUT BERG OR H315.
2296      ;          EXIT TST
2297      ;
2298      ;
2299      ;MOV      #1,$TIMES
2300      ;BR       TST22      ;;EXIT THIS TEST
2301      ;          ENDF
2302      ;$65:
2303      ;
2304      ;          ; SECXMIT AND
2305      ;          ; SECRC ARE CONNECTED
2306      ;
2307      ;          ; BY THE H315 OR EQUIV.
2308      ;
2309      ;          ; CLEAR
2310      ;          BGNSUB
2311      ;          MOV      #64$,SLPERR
2312      ;
2313      ;          ; CLEAR SECXMIT
2314      ;          LET      @RCSR := @RCSR CLR.BY #SECXMIT
2315      ;          ; SECRC SHOULD FOLLOW
2316      ;          IF      #SECRC SETIN @RCSR THEN
2317      ;          ; SECRC DID NOT
2318      ;          ; ERRHRD $6,,FORCE
2319      ;          ;
2320      ;          ; CLEAR WITH SECXMIT
2321      ;          ENDF
2322      ;          ENDSUB
2323      ;
2324      ;          ; SET
2325      ;          BGNSUB
2326      ;          MOV      #64$,SLPERR
2327      ;
2328      ;          ; SET SECXMIT
2329      ;          LET      @RCSR := @RCSR SET.BY #SECXMIT
2330      ;          ; SECRC SHOULD FOLLOW
2331      ;          IF      #SECRC NOTSETIN @RCSR THEN
2332      ;          ; SECRC DID NOT SET
2333      ;
2334      ;
2335      ;
2336      ;
2337      ;
2338      ;
2339      ;
2340      ;

```

G05

MAINDEC-22-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 59
 CVDVAB.P11 15-DEC-77 08:58 T21 TEST THAT SECURE SETS AND CLEARS

SEG 0058

```

2341 004642                                ERRHRD 57,,FORCE
2342 004642 104057                        ERROR 57
2343
2344
2345                                ; WITH SECXMIT
2346                                ENDIF
2347 004644                                ENDSUB
2348 004644
2349                                ; CLEAR
2350                                BGNSUB
2351 004644 012767 004652 174236            MOV  #64$,SLPERR
2352
2353                                ; CLEAR SECXMIT
2354 004652                                LET  @RCSR := @RCSR CLR.BY #SECXMIT
2355 004652 042777 000010 174400            BIC  #SECXMIT,@RCSR
2356                                ; SECURE SHOULD FOLLOW
2357 004660                                IF  #SECURE SETIN @RCSR THEN
2358 004660 032777 002000 174372            BIT  #SECURE,@RCSR
2359 004666 001401                                BEQ  $70
2360                                ; SECURE DID NOT
2361 004670                                ERRHRD 60,,FORCE
2362 004670 104060                        ERROR 60
2363
2364                                ; CLEAR WITH SECXMIT
2365 004672                                ENDIF
2366 004672                                ENDSUB
2367 004672                                ENDTST
2368 004672
2369
2370
2371
2372

```

H05

MAINDEC-22-CVOVA-B MACY11 30(1046)
CVOVA8.P11 15-DEC-77 08:58

19-DEC-77 08:25 PAGE 60
T21 TEST THAT SECURE SETS AND CLEARS

SEG 0059

```

2373 .....
2374 .....
2375 *TEST 22      TEST THAT DATAINT (RCSR-15) SETS
2376 *              WHEN DTR CHANGES STATE
2377 *              AND THAT DATAINT IS CLEARED AFTER READING RCSR
2378 *              NOTE DTR IS TIED TO BOTH CARDET AND CLSEND BY THE H315
2379 *              THE (-FD) JUMPER MUST BE IN FOR THIS TEST.
2380 .....
2381 ST22: SCOPE
2382 MOV     #10,$TIMES      ;;DO 10 ITERATIONS
2383 MOV     #22,$TESTIN    ;;SET TEST NUMBER IN APT MAIL BOX
2384                      IF      #CABLE+FRFD NOTSETIN $USWR THEN
2385 004672 000004           ; CAN WE USE THE WRAPAROUND??
2386 004674 012767 000010 174256 BIT     #CABLE+FRFD,$USWR
2387 004702 012767 000022 174270 BNE     $71
2388                      ; CAN'T TEST WITHOUT BERG OR H315
2389                      ; OR WITH (-FD) JUMPER OUT.
2390                      EXIT TST
2391 004710           ;;;EXIT THIS TEST
2392 004720 012767 000001 174232 BR      TST23
2393 004726 000463           ENDF
2394 004730
2395 $71:
2396
2397 004730 012746 000340 MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
2398 004734 012746 004742 MOV     #64$,-(SP)    ;;PUT NEW PC ON STACK
2399 004740 000002 RTI        ;;POP NEW PC AND PS
2400 004742
2401 64$:
2402                      ;READ TWICE - CLEARS
2403                      BGNSUB
2404 004742 012767 004750 174140 MOV     #65$,$LPERR
2405                      ; CLEAR DTR
2406 004750 LET      @RCSR := @RCSR CLR.BY #DTR
2407 004750 042777 000002 174302 BIC     #DTR,@RCSR
2408                      ;WAIT 1 MILLI-SEC FOR CABLE
2409 004756 WAITMS 1
2410 004756
2411 004760 010546 MOV     R5,-(SP)
2412 004760 012745 000001 MOV     #1,-(R5)
2413 004764 004767 004672 JSR     PC,WAIT
2414 004770 012605 MOV     (SP)+,R5
2415                      ; READ RCSR - TO CLEAR DATAINT
2416 004772 LET      R3 := @RCSR
2417 004772 017703 174262 MOV     @RCSR,R3
2418                      ; READ RCSR AGAIN
2419 004776 IF      #DATAINT SETIN @RCSR THEN
2420 005004 001401 100000 174254 BIT     #DATAINT,@RCSR
2421 005006 $72:
2422 005006 ERROR 61
2423 005006
2424 005010
2425 005010
2426
2427 005010
2428 ENDSUB

```


; DTR SETTING SETS DATAINT
 BGNSUB

```

2429 005010
2430 005010 012767 005016 174072 MOV #64$,SLPERR
2431
2432
2433
2434 005016 ;SET DTR
2435 005016 052777 000002 174234 BIS #DTR,@RCSR LET @RCSR := @RCSR SET.BY #DTR
2436 005024 IF #DATAINT NOTSETIN @RCSR THEN
2437 005024 032777 100000 174226 BIT #DATAINT,@RCSR
2438 005032 001001 BNE $73
2439 ;SETTING DTR DID NOT SET DATAINT
2440 005034 ERRHRD 62,, E2DATA
2441 005034 104062 ERROR 62
2442 005036
2443 005036 $73:
2444
2445 005036 IF #DATAINT SETIN @RCSR THEN
2446 005036 032777 100000 174214 BIT #DATAINT,@RCSR
2447 005044 001401 BEQ $74
2448 ;READING RCSR DID NOT CLEAR DATAINT
2449 005046 ERRHRD 63,E2DATA
2450 005046 104063 ERROR 63
2451 005050
2452 005050 $74:
2453 005050
2454
2455 ; DTR CLEARING SETS DATAINT
2456 005050 BGNSUB
2457 005050 012767 005056 174032 MOV #64$,SLPERR
2458
2459
2460 005056 ;CLEAR DTR
2461 005056 042777 000002 174174 BIC #DTR,@RCSR LET @RCSR := @RCSR CLR.BY #DTR
2462 005064 IF #DATAINT NOTSETIN @RCSR THEN
2463 005064 032777 100000 174166 BIT #DATAINT,@RCSR
2464 005072 001001 BNE $75
2465 ;CLEARING DTR DID NOT SET DATAINT
2466 005074 ERRHRD 64,, E2DATA
2467 005074 104064 ERROR 64
2468 005076
2469 005076 $75:
2470 005076
2471 005076
2472
2473
2474
  
```

 ENDST
 ENDSUB

J05

 MAINDEC-22-CVDVA-B MACY11 30(1046)
 CVDVAB.P11 15-DEC-77 08:58

 19-DEC-77 08:25 PAGE 62
 T22 TEST THAT DATAINT (RCSR-15) SETS

SEQ 0061

```

2475 ;*****
2476 ;*****
2477 ;TEST 23      TEST THAT DATAINT SETS WHEN RING SETS
2478 ;            AND THAT DATAINT DOES NOT SET WHEN RING CLEARS
2479 ;            THE (-FR) JUMPER MUST BE IN FOR THIS TEST.
2480 ;*****
2481 005076 000004  ST23: SCOPE
2482 005100 012767  MOV     #10,$TIMES      ;;DO 10 ITERATIONS
2483 005106 012767  MOV     #23,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2484 ;                                     ; CAN WE USE THE WRAPAROUND??
2485 005114 032767 060000 174076      BIT     #CABLE+FRFD,$USWR      IF     #CABLE+FRFD NOTSETIN $USWR THEN
2486 005114 001004  BNE     $76
2487 ;                                     ; CAN'T TEST WITHOUT BERG OR H315
2488 ;                                     ; OR WITH (-FR) JUMPER OUT.
2489 ;                                     EXIT TST
2490 005124 012767 000001 174026      MOV     #1,$TIMES
2491 005132 000473  BR      TST24      ;;EXIT THIS TEST
2492 005134  ENDIF
2493 005134
2494 $76:
2495
2496 ;NO INTERRUPTS
2497 005134 012746 000340      MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
2498 005140 012746 005146      MOV     #64$,-(SP)      ;;PUT NEW PC ON STACK
2499 005144 000002  RTI          ;;POP NEW PC AND PS
2500 005146
2501 64$:
2502 ;START OFF WITH EVERYTHING CLEAR
2503 005146      BGNSUB
2504 005146 012767 005154 173734      MOV     #65$,$LPERR
2505
2506 ;CLEAR RING
2507 005154      LET     @RCSR := @RCSR CLR.BY #REQSEND
2508 005154 042777 000004 174076      BIC     #REQSEND,@RCSR
2509 ;WAIT 1 MILLI-SEC FOR CABLE
2510 005162      WAITMS 1
2511 005162 010546      MOV     R5,-(SP)
2512 005164 012745 000001      MOV     #1,-(R5)
2513 005170 004767 004466      JSR     PC,WAIT
2514 005174 012605      MOV     (SP)+,R5
2515
2516 ;READ ONCE
2517 005176      LET     R3 := @RCSR
2518 005176 017703 174056      MOV     @RCSR,R3
2519 ;READ TWICE
2520 005202      IF     #DATAINT SETIN @RCSR THEN
2521 005202 032777 100000 174050      BIT     #DATAINT,@RCSR
2522 005210 001401      BEQ     $77
2523 ;READING RCSR DID NOT CLEAR DATAINT
2524 005212      ERRHRD 65, EDATAINT
2525 005212 104065      ERROR 65
2526 005214      ENDIF
2527 005214 $77:
2528      ENDSUB
2529 ;
2530 005214      SET RING --> SET DATAINT
      BGNSUB

```


K05

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 63
 CVDVAB.P11 15-DEC-77 08:58 T23 TEST THAT DATAINT SETS WHEN RING SETS

SEQ 0062

```

2531 005214 012767 005222 173666      MOV      #64$, $LPERR
2532
2533                                     ;WE ARE SETTING RING
2534                                     ; SET RING
2535 005222                                     LET      @RCSR := @RCSR SET.BY #REQSEND
2536 005222 052777 000004 174030      BIS      #REQSEND, @RCSR
2537                                     ;WAIT 1 MILLI-SEC FOR CABLE
2538 005230                                     WAITMS 1
2539 005230 010546      MOV      R5, -(SP)
2540 005232 012745 000001      MOV      #1, -(R5)
2541 005236 004767 004420      JSR      PC, WAIT
2542 005242 012605      MOV      (SP)+, R5
2543 005244                                     IF      #DATAINT NOTSETIN @RCSR THEN
2544 005244 032777 100000 174006      BIT      #DATAINT, @RCSR
2545 005252 001001      BNE      $100
2546                                     ;SETTING RING DID NOT SET DATAINT
2547 005254                                     ERRHRD 122,, E2DATA
2548 005254 104122      ERROR      122
2549 005256                                     ENDIF
2550 005256                                     $100:
2551 005256                                     ENDSUB
2552                                     ;CLEAR RING CAUSES NO CHANGE IN DATAINT (CLEAR)
2553                                     BGNSUB
2554 005256      MOV      #64$, $LPERR
2555 005256 012767 005264 173624
2556
2557                                     ;CLEAR RING
2558 005264                                     LET      @RCSR := @RCSR CLR.BY #REQSEND
2559 005264 042777 000004 173766      BIC      #REQSEND, @RCSR
2560                                     ;WAIT 1 MILLI-SEC FOR CABLE
2561 005272                                     WAITMS 1
2562 005272 010546      MOV      R5, -(SP)
2563 005274 012745 000001      MOV      #1, -(R5)
2564 005300 004767 004356      JSR      PC, WAIT
2565 005304 012605      MOV      (SP)+, R5
2566 005306                                     IF      #DATAINT SETIN @RCSR THEN
2567 005306 032777 100000 173744      BIT      #DATAINT, @RCSR
2568 005314 001401      BEQ      $101
2569                                     ;CLEARING RING SET DATAINT
2570 005316                                     ERRHRD 123, CLRANG
2571 005316 104123      ERROR      123
2572 005320                                     ENDIF
2573 005320                                     $101:
2574 005320                                     ENDSUB
2575 005320      EXIT      ;SKIP AROUND MESSAGE
2576 005320 000400      BR       TST24      ;;;EXIT THIS TEST
2577 005322      ENDTST
2578
2579
2580

```

MAINDEC-ZZ-CVDVA-B MACY11 30(1046)
CVDVAB.P11 15-DEC-77 08:58

19-DEC-77 08:25 PAGE 64
T23 TEST THAT DATAINT SETS WHEN RING SETS

SEQ 0063

```

2581      ;*****
2582      ;*****
2583      ;TEST 24      TEST THAT DATAINT SETS WHEN SECRC CHANGES STATE
2584      ;*****
2585      ST24: SCOPE
2586      005322 000004      MOV      #10,STIMES      ;;DO 10 ITERATIONS
2587      005324 012767 000010 173626      MOV      #24,STESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2588      005332 012767 000024 173640      IF          CAN WE USE THE WRAPAROUND??
2589      005340      BIT      #CABLE,$USWR      IF          #CABLE NOTSETIN $USWR THEN
2590      005340 032767 020000 173652      BNE      $102
2591      005346 001004      EXIT      ; CAN'T TEST WITHOUT BERG OR H315.
2592      ;
2593      005350      MOV      #1,STIMES      EXIT TST
2594      005350 012767 000001 173602      BR       TST25      ;;EXIT THIS TEST
2595      005356 000454      ENDF
2596      005360
2597      005360      $102:
2598
2599      005360 012746 000340      MOV      #PR7,-(SP)      ;NO INTERRUPTS
2600      005364 012746 005372      MOV      #64$,-(SP)      ;;PUT NEW PS ON STACK
2601      005370 000002      RTI          ;;PUT NEW PC ON STACK
2602      005372      64$:      ;;POP NEW PC AND PS
2603
2604
2605
2606      ;START FRESH
2607      ;CLEAR SECRC
2608      005372      LET      @RCSR := @RCSR CLR.BY #SECXMIT
2609      005372 042777 000010 173660      BIC      #SECXMIT,@RCSR
2610      005400      LET      R3 := @RCSR
2611      005400 017703 173654      MOV      @RCSR,R3
2612
2613      ;SET SECRC --> DATAINT SET
2614      005404      BGNSUB
2615      005404 012767 005412 173476      MOV      #65$,SLPERR
2616
2617      ;SET SECRC
2618      005412      LET      @RCSR := @RCSR SET.BY #SECXMIT
2619      005412 052777 000010 173640      BIS      #SECXMIT,@RCSR
2620
2621      ;WAIT 1 MILLI-SEC FOR CABLE
2622      005420      WAITMS 1
2623      005420 010546      MOV      R5,-(SP)
2624      005422 012745 000001      MOV      #1,-(R5)
2625      005426 004767 004230      JSR      PC,WAIT
2626      005432 012605      MOV      (SP)+,R5
2627
2628      IF          #DATAINT NOTSETIN @RCSR THEN
2629      005434 032777 100000 173616      BIT      #DATAINT,@RCSR
2630      005442 001001      BNE      $103
2631
2632      ;SETTING SECRC DID NOT SET DATAINT
2633      005444      ERRHRD 124,, E2DATA
2634      005444 104124      ERROR      124
2635      005446      ENDF
2636      005446      ENDSUB

```

;CLEAR SECRC --> DATAINT SET

MOS

```

MAINDEC-22-CVDVA-8      MACY11 30(1046) 19-DEC-77 08:25 PAGE 65
CVDVAB.P11      15-DEC-77 08:58      T24      TEST THAT DATAINT SETS WHEN SECRC CHANGES STATE
                                                    SEQ 0064

2637 005446
2638 005446 012767 005454 173434      MOV      #64$, $LPERR      BGNSUB
2639
2640 005454
2641 005454 042777 000010 173576      BIC      #SECXMIT, @RCR      ;CLEAR SECRC
2642
2643 005462
2644 005462 010546
2645 005464 012745 000001      MOV      R5, -(SP)      LET @RCR := @RCR CLR.BY #SECXMIT
2646 005470 004767 004166      MOV      #1, -(R5)      ;WAIT 1 MILLI-SEC FOR CABLE
2647 005474 012605      JSR      PC, WAIT      WAITMS 1
2648 005476
2649 005476 032777 100000 173554      BIT      #DATAINT, @RCR      IF      #DATAINT NOTSETIN @RCR THEN
2650 005504 001001      BNE      $104      ;CLEARING SECRC DID NOT SET DATAINT
2651
2652 005506
2653 005506 104125      ERROR      125      ERRHRD 125,, E2DATA
2654 005510
2655 005510      $104:
2656 005510
2657 005510      ENDSUB
2658
2659
2660      ENDTST

```

N05

MAINDEC-22-CVDVA-B MACY11 30(1046)
CVDVAB.P11 15-DEC-77 08:58

19-DEC-77 08:25 PAGE 66
T24 TEST THAT DATAINT SETS WHEN SECREC CHANGES STATE

SEQ 0065

```

2661 *****
2662 *****
2663 *****
2664 *TEST 25      TEST THAT XMIT RDY - TCSR 7 - CLEARS
2665 *              WHEN TBUF IS LOADED WITH A CHARACTER
2666 *              AND THAT IT SETS WITHIN A REASONABLE AMOUNT OF TIME.
2667 *****
2668 ST25: SCOPE
2669      MOV      #1,STIMES      ;;DO 1 ITERATION
2670      MOV      #25,STESTN     ;;SET TEST NUMBER IN APT MAIL BOX
2671      ; THIS TEST IS 'BREAK OR HALT' SENSITIVE.
2672      ; IF #APTENV SETIN SENV THEN
2673      BIT      #APTENV,SENV
2674      BEQ      $105
2675      EXIT TEST
2676      MOV      #1,STIMES
2677      BR       TST26          ;;EXIT THIS TEST
2678      ENDF
2679 $105:
2680      BGNSUB
2681      MOV      #64$,$LPERR
2682      ; LOAD TBUF WITH ONE CHARACTER
2683      ; WAIT FOR READY TO SET
2684      ; (SHOULD BE VERY SHORT WAIT
2685      ; SINCE UART DOUBLE BUFFERS ITS INPUT)
2686      ; SEND A CHARACTER
2687      LET      @TBUF :B= #0
2688      ; WAIT A MAXIMUM
2689      ; OF 50 MSEC FOR
2690      ; XMIT RDY TO SET IN TCSR
2691      CALL     TIMER IN (<#500,#XMITRDY,TCSR,#SET>)
2692      MOV      R5,-(SP)
2693      MOV      #SET,-(R5)
2694      MOV      TCSR,-(R5)
2695      MOV      #XMITRDY,-(R5)
2696      MOV      #500,-(R5)
2697      JSR      PC,TIMER
2698      MOV      (SP)+,R5
2699      ; TIMER RETURNS AN ERROR IF BIT DID
2700      ; NOT MEET CONDITION WITHIN TIME LIMIT
2701      ; IF.ERROR THEN
2702      ; XMIT RDY DID NOT SET IN TCSR
2703      ; ERRHRD 66,,DIDNOT
2704      BCC      $106
2705      ;
2706      ;
2707      ;
2708      ;
2709      ;
2710      ;
2711      ;
2712      ;
2713      ;
2714      ;
2715      ;
2716      ;

```

```

2717
2718
2719 005622
2720 005622 105077 173442 CLR B 2TBUF NOP
2721 005626 000240
2722
2723
2724 005630
2725 005630 032777 000200 173426 BIT 2XMITRDY,2TCSR
2726 005636 001401 BEQ $107
2727
2728 005640
2729 005640 104067 ERROR 67
2730 005642
2731 005642 $107:
2732
2733
2734
2735
2736 005642
2737 005642 010546 MOV R5, -(SP)
2738 005644 012745 177777 MOV 2SET, -(R5)
2739 005650 016745 173410 MOV TCSR, -(R5)
2740 005654 012745 000200 MOV 2XMITRDY, -(R5)
2741 005660 012745 000500 MOV 2500, -(R5)
2742 005664 004767 003514 JSR PC, TIMER
2743 005670 012605 MOV (SP)+, R5
2744 005672
2745 005672 103001 BCC $110
2746
2747 005674
2748 005674 104070 ERROR 70
2749 005676
2750 005676 $110:
2751 005676
2752 005676

```

; SEND SECOND CHARACTER
 LET 2TBUF :B= 00
 ; GIVE IT TIME TO CLEAR
 ; XMITRDY SHOULD HAVE CLEARED UPON
 ; RECEIPT OF A CHARACTER
 IF 2XMITRDY SET IN 2TCSR THEN
 ; XMITRDY DID NOT CLEAR IN TCSR
 ERRHRD 67,,DIDNOT
 ENDIF
 ; WAIT A MAXIMUM
 ; OF 50 MSEC FOR
 ; XMIT RDY TO SET IN TCSR
 CALL TIMER IN <2500,2XMITRDY,TCSR,2SET>
 IF .ERROR THEN
 ; XMIT RDY DID NOT SET IN TCSR
 ERRHRD 70,,DIDNOT
 ENDIF
 ENDSUB
 ENDTST

```

2753 *****
2754 *****
2755 *****
2756 TEST 26      TEST THAT OUTPUTTING A CHAR FROM TBUF (WITH MAINT SET)
2757 *           RESULTS IN RCVRDONE SETTING WITHIN A REASONABLE AMOUNT OF TIME
2758 *           AND THAT RESET CLEARS THE BIT.
2759 *****
2759 005676 000004  ST26: SCOPE
2760 005700 012767 000010  MOV    #10,STIMES      ;; DO 10 ITERATIONS
2761 005706 012767 000026 173252 MOV    #26,STESTN      ;; SET TEST NUMBER IN APT MAIL BOX
2762
2763                               ; SET THE MAINTENANCE BIT
2764 005714 052777 000004 173342 BIS     #MAINT,@TCSR      LET @TCSR := @TCSR SET.BY #MAINT
2765
2766                               BGNSUB
2767 005722 012767 005730 173160 MOV     #64$,$LPERR
2768                               ; SEND A CHARACTER AND LET IT WRAP AROUND
2769
2770                               LET @TBUF :B= #0
2771 005730 105077 173334 CLRB    @TBUF
2772
2773                               ; WAIT A MAXIMUM OF 50 MSEC
2774                               ; FOR RCVR DONE TO SET IN
2775                               ; RCSR
2776                               CALL TIMER IN (<#500,#RCVRDONE,RCSR,#SET>)
2777
2778 005734 010546 177777 MOV     RS,-(SP)
2779 005736 012745 173312 MOV     #SET,-(RS)
2780 005742 016745 000200 MOV     RCSR,-(RS)
2781 005746 012745 000500 MOV     #RCVRDONE,-(RS)
2782 005752 012745 003422 MOV     #500,-(RS)
2783 005756 004767 JSR      PC,TIMER
2784 005762 012605 MOV     (SP)+,RS
2785
2786                               ; DIDN'T SET IN TIME
2787 005764 103001 BCC      $1111      IF .ERROR THEN
2788
2789                               ; RCVRDONE DID NOT SET IN RCSR
2790                               ERRHRD 71,,DIDNOT
2791 005766 104071 ERROR    71
2792 005770
2793                               ENDIF
2794
2795                               $1111:
2796                               ENDSUB
2797
2798                               BGNSUB
2799 005770 012767 005776 173112 MOV     #64$,$LPERR
2800                               ; NOW THAT IT IS SET SEE IF IT CAN BE RESET
2801                               ; THIS ALSO WILL CLEAR THE MAINT. BIT
2802 005776 000005 RESET
2803                               BRESET
2804
2805                               IF #RCVRDONE SETIN @RCSR THEN
2806 006000 032777 000200 173252 BIT     #RCVRDONE,@RCSR
2807 006006 001401 BEQ      $112
2808                               ; RCVRDONE DID NOT RESET IN RCSR.
2809                               ERRHRD 72,,DIDNOT

```


D06

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 69
CVDVAB.P11 15-DEC-77 08:58 T26 TEST THAT OUTPUTTING A CHAR FROM TBUF (WITH MAINT SET)

SEQ 0068

2809 006010 104072
2810 006012
2811 006012
2812 006012
2813 006012

ERROR 72

\$112:

ENDIF

ENDSUB

ENDTST

E06

 MAINDEC-22-CVDVA-B MACY11 30(1046)
 CVDVAB.P11 15-DEC-77 08:58

 19-DEC-77 08:25 PAGE 70
 T26 TEST THAT OUTPUTTING A CHAR FROM TBUF (WITH MAINT SET)

SEQ 0069

```

2814 ;*****
2815 ;*****
2816 ;*****
2817 ;*****
2818 006012 000004          ;*****
2819 006014 012767 000010 173136 1ST27: SCOPE
2820 006022 012767 000027 173150      MOV    #10,$TIMES      ;;DO 10 ITERATIONS
2821                                     MOV    #27,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2822                                     ; SET MAINT. BIT
2823 006030                                     LET @TCSR := @TCSR SET.BY #MAINT
2824 006030 052777 000004 173226      BIS    #MAINT,@TCSR
2825 006036                                     BGNSUB
2826 006036 012767 006044 173044      MOV    #64,$SLPERR
2827                                     ; OUTPUT A CHARACTER WITH MAINTENANCE
2828                                     ; SET, AND WAIT FOR XMITRDY TO SET.
2829
2830                                     ; OUTPUT A CHARACTER
2831 006044                                     LET @TBUF :B= #0
2832 006044 105077 173220      CLRB    @TBUF
2833                                     ; WAIT MAXIMUM OF 500 MSEC
2834                                     ; FOR RCVRDONE TO SET IN
2835                                     ; RCSR
2836 006050                                     CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
2837 006050 010546          MOV    R5,-(SP)
2838 006052 012745          MOV    #SET,-(R5)
2839 006056 016745          MOV    RCSR,-(R5)
2840 006062 012745          MOV    #RCVRDONE,-(R5)
2841 006066 012745          MOV    #500,-(R5)
2842 006072 004767          JSR    PC,TIMER
2843 006076 012605          MOV    (SP)+,R5
2844
2845 006100                                     ; DID IT BECAME READY?
2846 006100 103001          BCC     $113      IF.ERROR THEN
2847                                     ;RCVRDONE DID NOT SET IN RCSR
2848 006102                                     ERRHRD 73,, DIDNOT
2849 006102 104073          ERROR    73
2850 006104
2851 006104          $113:
2852 006104
2853                                     ENDIF
2854                                     ENDSUB
2855                                     ; NOW THAT IT IS SET LETS SEE IF READING THE
2856                                     ; BUFFER CLEARS RCVRDONE.
2857
2858 006104                                     ;READ BUFFER
2859 006104 117700 173152          MOVB   @RBUF,R0      LET R0 :B= @RBUF
2860
2861 006110                                     IF #RCVRDONE SETIN @RCSR THEN
2862 006110 032777 000200 173142      BIT    #RCVRDONE,@RCSR
2863 006116 001401          BEQ     $114
2864                                     ;RCVRDONE DID NOT CLEAR IN RCSR
2865 006120                                     ERRHRD 74,DIDNOT
2866 006120 104074          ERROR    74
2867 006122          $114:
2868 006122
2869 006122          ENDTST

```


F06

 MAINDEC-22-CVDVA-B MACY11 30(1046)
 CVDVAB.P11 15-DEC-77 08:58

 19-DEC-77 08:25 PAGE 71
 T27 TEST THAT RCVRDONE IS CLEARED BY READING RBUF

SEQ 0070

```

2870 ;*****
2871 ;*****
2872 ;TEST 30      TEST THAT RCVRACT - RCSR 11 - SETS
2873 ;            WHEN A START BIT IS RECEIVED AND
2874 ;            CLEARS WHEN RCVRDONE - RCSR 7 - SETS
2875 ;*****
2876 006122 000004          ST30: SCOPE
2877 006124 012767 000010 173026      MOV    #10,STIMES      ;;DO 10 ITERATIONS
2878 006132 012767 000030 173040      MOV    #30,STESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2879
2880 ; THIS TEST IS 'BREAK OR HALT' SENSITIVE.
2881 006140          IF #APTENV SETIN #ENV THEN
2882 006140 032767 000001 173046      BIT     #APTENV,#ENV
2883 006146 001404          BEQ     $115
2884 006150          EXIT TEST
2885 006150 012767 000001 173002      MOV    #1,STIMES
2886 006156 000500          BR      TST31      ;;;EXIT THIS TEST
2887 006160          ENDF
2888 006160          $115:
2889 006160          LET #TCSR := #TCSR SET.BY #MAINT
2890 006160 052777 000004 173076      BIS     #MAINT,#TCSR
2891 006166          LET RO := #CLR
2892 006166 012700 000000          MOV    #CLR,RO
2893 006172          LET R1 := #0
2894 006172 005001          CLR     R1
2895          ;LOAD A CHARACTER INTO TBUF
2896          ;WAIT FOR RCVRACT TO SET
2897
2898          ;SEND A CHARACTER
2899 006174          LET #TBUF := #0
2900 006174 105077 173070          CLRB    #TBUF
2901 006200          REPEAT
2902 006200          $116:
2903 006200          IF #RCVRACT SETIN #RCSR THEN
2904 006200 032777 004000 173052      BIT     #RCVRACT,#RCSR
2905 006206 001403          BEQ     $117
2906 006210          LET RO := #SET
2907 006210 012700 177777          MOV    #SET,RO
2908 006214          ELSE
2909 006214 000401          BR      $120
2910 006216          $117:
2911 006216          LET R1 := R1 + #1
2912 006216 005201          INC     R1
2913 006220          ENDF
2914 006220          $120:
2915 006220          UNTIL RO EQ #SET OR R1 HI MAX
2916 006220 020027 177777          CMP     RO,#SET
2917 006224 001403          BEQ     $121
2918 006226 020167 000124          CMP     R1,MAX
2919 006232 101762          BLOS    $116
2920 006234          $121:
2921 006234          IF R1 HI MAX THEN
2922 006234 020167 000116          CMP     R1,MAX
2923 006240 101407          BLOS    $122
2924
2925          ;IT NEVER SET
          ;RCVRACT DID NOT SET IN RCSR.

```


H06

MAINDEC-22-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 73
CVDVAB.P11 15-DEC-77 08:58 T30 TEST THAT RCVRACT - RCSR 11 - SETS

SEQ 0072

```
2982                                     ;READ CHAR.  
2983 006336                               LET RO := ARBUF  
2984 006336 017700 172720             MOV  ARBUF,RO  
2985  
2986 006342                               IF #RCVRDONE SETIN ARCSR THEN  
2987 006342 032777 000200 172710     BIT   #RCVRDONE,ARCSR  
2988 006350 001401                     BEQ   $130  
2989                                     ;RCVRDONE DID NOT CLEAR IN RCSR  
2990 006352                               ERRHRD 100,,DIDNOT  
2991 006352 104100                     ENDIF  
2992 006354  
2993 006354                               $130:  
2994  
2995 006354                               BR     TST31  
2996 006354 000401                     ;;;EXIT THIS TEST  
2997 006356 070000                     MAX:70000  
2998  
2999 006360                               EXIT  
3000                               ENDTST
```

```

3001
3002
3003
3004
3005
3006 006360 000004
3007 006362 012767 000010 172570
3008 006370 012767 000031 172602
3009
3010 006376
3011 006376 012767 006404 172504
3012
3013
3014
3015
3016
3017 006404
3018 006404 105077 172660
3019
3020 006410
3021 006410 010546
3022 006412 012745 000310
3023 006416 004767 003240
3024 006422 012605
3025
3026
3027 006424
3028 006424 105077 172640
3029
3030 006430
3031 006430 010546
3032 006432 012745 000310
3033 006436 004767 003220
3034 006442 012605
3035
3036
3037 006444
3038 006444 017704 172612
3039
3040
3041 006450
3042 006450 032704 040000
3043 006454 001005
3044
3045 006456
3046 006456 104101
3047
3048
3049 006460
3050 006460 012767 000001 172472
3051 006466 000456
3052 006470
3053 006470
3054 006470
3055
3056

```

```

*****
*****
TEST 31      TEST THE OVERRUN BIT - RBUF 14
*****
TST31:  SCOPE
        MOV     #10, $TIMES      ;; DO 10 ITERATIONS
        MOV     #31, $TESTN     ;; SET TEST NUMBER IN APT MAIL BOX
                                BGNSUB
                                : OUTPUT 2 CHARACTERS WITH
                                : AMPL DELAYS BETWEEN FOR RECEPTION.
                                : THIS SHOULD AN CAUSE OVERRUN ERROR.
                                : OUTPUT 1 CHARACTER
                                LET $TBUF :B= #0
                                : GO AWAY FOR 200. M SEC
                                WAITMS 200.
        MOV     R5, -(SP)
        MOV     #200, -(R5)
        JSR     PC, WAIT
        MOV     (SP)+, R5
                                : OUTPUT 2ND CHARACTER
                                LET $TBUF :B= #0
                                : LET OVERRUN HAPPEN
                                WAITMS 200.
        MOV     R5, -(SP)
        MOV     #200, -(R5)
        JSR     PC, WAIT
        MOV     (SP)+, R5
                                : READ BUFFER AND ERROR BITS
                                LET R4 := $RBUF
        MOV     $RBUF, R4
                                : IT DIDN'T SET
                                IF #ORERR NOTSET IN R4 THEN
                                : ORERR DID NOT SET IN RBUF
                                ERRHRD 101,, DIDNOT
        ERROR   101
                                : NO USE COMPOUNDING ERRORS
                                EXIT TST
        MOV     #1, $TIMES
        BR      TST32          ;; EXIT THIS TEST
                                ENDIF
                                ENDSUB
                                : NOW SEE IF ERROR BIT SET WITH OVERRUN ERROR:

```

```

$131:

```

```

3057 006470                                BGNSUB
3058 006470 012767 006476 172412          MOV    #64$, $LPERR
3059 006476                                IF #ERROR NOTSETIN R4 THEN
3060 006476 032704 100000                    BIT    #ERROR, R4
3061 006502 001005                    BNE     $132
3062
3063                                ;ERROR DID NOT SET IN RBUF
3064 006504                                ERRHRD 102,,DIDNOT
3065 006504 104102                    ERROR    102
3066
3067                                ;--WHEN ORERR SET.
3068                                ;GET OUT NOW.
3069                                EXIT TST
3070 006506 012767 000001 172444          MOV    #1, $TIMES
3071 006514 000443                    BR      TST32          ;;;EXIT THIS TEST
3072 006516                                ENDIF
3073 006516                                $132:
3074 006516                                ENDSUB
3075
3076                                BGNSUB
3077 006516 012767 006524 172364          MOV    #64$, $LPERR
3078                                ;CHECK REAL RBUF TO SEE IF ORERR IS STILL SET.
3079                                IF #ORERR NOTSETIN RBUF THEN
3080 006524                                ;READING RBUF CLEARED ORERR.
3081 006524 032777 040000 172530          BIT    #ORERR, RBUF
3082 006532 001002                    BNE     $133
3083
3084                                ;SKIP REST OF TEST
3085 006534                                EXIT
3086 006534 104103                    ERROR    103
3087
3088 006536                                ;;;EXIT THIS TEST
3089 006536 000432                    BR      TST32          ENDIF
3090 006540                                $133:
3091 006540                                ENDSUB
3092 006540                                BGNSUB
3093
3094 006540                                ;NOW SEE IF THEY CLEAR WHEN ANOTHER CHAR. IS RECEIVED
3095 006540 012767 006546 172342          MOV    #64$, $LPERR
3096                                ;SEND A CHARACTER AROUND.
3097                                LET RBUF :B= #0
3098                                ;LET IT CIRCULATE
3099 006546                                WAITMS 200.
3100 006546 105077 172516                    CLRB   RBUF
3101
3102 006552                                IF #ORERR SETIN RBUF THEN
3103 006552 010546                    MOV    R5, -(SP)
3104 006554 012745 000310                    MOV    #200, -(R5)
3105 006560 004767 003076                    JSR    PC, WAIT
3106 006564 012605                    MOV    (SP)+, R5
3107
3108                                IF #ORERR SETIN RBUF THEN
3109 006566 032777 040000 172466          BIT    #ORERR, RBUF
3110 006574 001405                    BEQ     $134
3111
3112 006576                                ;ORERR DID NOT CLEAR IN RBUF
                                ERRHRD 104,,DIDNOT

```

K06

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 76
CVDVAB.P11 15-DEC-77 08:58 T31 TEST THE OVERRUN BIT - RBUF 14

SEQ 0075

```

3113 006576 104104          ERROR 104
3114
3115                                ; - AFTER RECEIVING ANOTHER CHAR
3116                                ; SKIP AROUND REST
3117                                EXIT TST
3118 006600 012767 000001 172352      MOV  #1, $TIMES
3119 006606 000406                      BR   TST32          ;;;EXIT THIS TEST
3120 006610                      $134:                      ENDIF
3121 006610
3122
3123 006610                                IF #ERROR SETIN @RBUF THEN
3124 006610 032777 100000 172444      BIT  #ERROR, @RBUF
3125 006616 001401                      BEQ  $135
3126                                ; ERROR DID NOT CLEAR IN RBUF
3127 006620                                ERRHRD 105,, DIDNOT
3128 006620 104105          ERROR 105
3129
3130                                ENDIF
3131 006622      $135:
3132 006622                                ENDSUB
3133 006622                                EXIT
3134 006622 000400          BR   TST32          ;;;EXIT THIS TEST
3135                                EVEN
3136 006624                                ENDTST
3137

```



```

3138
3139
3140
3141
3142
3143
3144
3145
3146 006624 000004
3147 006626 012767 000010 172324
3148 006634 012767 000032 172336
3149 006642
3150 006642 032767 000200 172350
3151 006650 001004
3152 006652
3153 006652 012767 000001 172300
3154 006660 000552
3155 006662
3156 006662
3157
3158 006662
3159 006662 032767 000001 172324
3160 006670 001404
3161 006672
3162 006672 012767 000001 172260
3163 006700 000542
3164 006702
3165 006702
3166
3167 006702
3168 006702 012767 177777 000272
3169 006710
3170 006710 012767 177777 000266
3171 006716
3172 006716 052777 000004 172340
3173
3174 006724
3175 006724 005003
3176 006726 000401
3177 006730
3178 006730 005203
3179 006732
3180 006732 020327 000017
3181 006736 003060
3182 006740
3183 006740 017700 172316
3184
3185 006744
3186 006744 116377 007124 172314
3187
3188 006752
3189 006752 005002
3190
3191 006754
3192 006754 005077 172310
3193

```

```

*****
*****
TEST 32      PROGRAMMABLE BAUD RATE TEST
              TEST AT ALL SPEEDS AVAILABLE
              A COMPARISON WILL BE MADE TO SEE
              IF NEW TIME IS LESS THAN PREVIOUS.
*****
TST32:  SCOPE
              MOV      #10,$TIMES      ;;DO 10 ITERATIONS
              MOV      #32,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                      IF #PBR NOTSETIN $USWR THEN
              BIT      #PBR,$USWR
              BNE      $136
                                      EXIT TST
              MOV      #1,$TIMES
              BR        TST33          ;;;EXIT THIS TEST
                                      ENDIF
$136:
              ; THIS TEST IS 'BREAK OR HALT' SINSATIVE.
              IF #APTENV SETIN $ENV THEN
              BIT      #APTENV,$ENV
              BEQ      $137
                                      EXIT TEST
              MOV      #1,$TIMES
              BR        TST33          ;;;EXIT THIS TEST
                                      ENDIF
$137:
              LET OLD := #-1
              MOV      #-1,OLD
              LET OLD+2 := #-1
              MOV      #-1,OLD+2
              LET @TCSR := @TCSR SET.BY #MAINT
              ;EACH BAUD RATE
              INCR R3 FROM #0 TO #15. BY #1
              CLR      R3
              BR        $140
$141:
              INC      R3
$140:
              CMP      R3,#15.
              BGT      $142
              LET RO := @RBUF
              ;CHANGE BAUDE RATE
              LET @TCSRHI :=# RATES(R3)
              ;FLAG
              LET BIT := #0
              ;OUTPUT THE CHARACTER
              LET @TBUF := #0
              ;INITIALIZE COUNTER

```

M06

MAINDEC-22-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 78
 CVDVAB.P11 15-DEC-77 08:58 T32 PROGRAMMABLE BAUD RATE TEST

SEQ 0077

3194	006760				
3195	006760	005067	000212	CLR	NEW
3196	006764				
3197	006764	005067	000210	CLR	NEW+2
3198	006770				
3199	006770			\$143:	
3200	006770	005702		TST	BIT
3201	006772	001014		BNE	\$144
3202	006774				
3203	006774	032777	000200 172256	BIT	RCVRDONE, RCSR
3204	007002	001403		BEQ	\$145
3205					
3206	007004				
3207	007004	012702	000001	MOV	#1, BIT
3208	007010				
3209	007010	000404		BR	\$146
3210	007012			\$145:	
3211					
3212	007012				
3213	007012	005267	000160	INC	NEW
3214	007016				
3215	007016	005567	000156	ADC	NEW+2
3216	007022				
3217	007022			\$146:	
3218					
3219	007022				
3220	007022	000762		BR	\$143
3221	007024			\$144:	
3222					
3223	007024				
3224	007024	026767	000150 000152	CMP	NEW+2, OLD+2
3225	007032	103001		BHIS	\$147
3226					
3227	007034				
3228	007034	000412		BR	\$150
3229	007036			\$147:	
3230					
3231	007036				
3232	007036	026767	000136 000140	CMP	NEW+2, OLD+2
3233	007044	001005		BNE	\$151
3234	007046	026767	000124 000126	CMP	NEW, OLD
3235	007054	103001		BHIS	\$151
3236					
3237	007056				
3238	007056	000401		BR	\$152
3239	007060			\$151:	
3240					
3241					
3242					
3243					
3244	007060				
3245	007060	104126		ERROR	126
3246	007062				
3247	007062			\$152:	
3248	007062				
3249	007062			\$150:	

```

LET NEW := #0
LET NEW+2 := #0
WHILE BIT EQ #0 DO

    IF RCVRDONE SETIN RCSR THEN

        ; DONE - ITS READY
        LET BIT := #1

    ELSE

        ; OTHERWISE-INCREMENT TIME
        LET NEW := NEW + #1
        LET NEW+2 := NEW+2 + CARRY

    ENDIF

; SIGNALS DONE
ENDDO

IF NEW+2 LO OLD+2 THEN

    ; OK
ELSE

    ; NEW+2 >= OLD+2
    IF NEW+2 EQ OLD+2 AND NEW LO OLD THEN

        ; OK
    ELSE

        ; NEW+2 > OLD+2 OR
        ; (NEW+2 = OLD+2 AND
        ; NEW >= OLD)
        ; BAUD RATE DIDN'T CHANGE
        ERRHRD 126, BAUDRATE

    ENDIF

ENDIF

```


N06

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 79
CVDVAB.P11 15-DEC-77 08:58 T32 PROGRAMMABLE BAUD RATE TEST

SEQ 0078

```

3250                                     :UPDATE OLD TIME
3251 007062                               LET OLD := NEW
3252 007062 016767 000110 000112         MOV NEW,OLD
3253 007070                               LET OLD+2 := NEW+2
3254 007070 016767 000104 000106         MOV NEW+2,OLD+2
3255                                     ENDINC ;BAUD RATE
3256 007076                               BR $141
3257 007076 000714                       $142:
3258 007100                               LET R3 :B= $USWR+1 AND #17 ; PUT BAUD BACK
3259 007100                               LET R3 :B= $USWR+1 AND #17 ; PUT BAUD BACK
3260 007100 116703 172115                 MOV $USWR+1,R3
3261 007104 110346                       MOV R3, -(SP)
3262 007106 142716 000017                 BICB #17, (SP)
3263 007112 142603                       BICB (SP)+,R3
3264 007114                               LET @TCSRHI :B= RATES(R3) ; LIKE HE WANTED IT
3265 007114 116377 007124 172144         MOV RATES(R3),@TCSRHI
3266
3267 007122                               EXIT ;SKIP TABLE
3268 007122 000431                       BR TST33 ;;;EXIT THIS TEST
3269
3270 007124
3271
3272
3273
3274
3275
3276
3277
3278
3279 007124 010
3280 007125 030
3281 007126 050
3282 007127 070
3283 007130 110
3284 007131 130
3285 007132 150
3286 007133 170
3287 007134 210
3288 007135 230
3289 007136 250
3290 007137 270
3291 007140 310
3292 007141 330
3293 007142 350
3294 007143 370
3295
3296 007144 040502 042125 051040
3297 007152 052101 020105 044504
3298 007160 047104 052047 041440
3299 007166 040510 043516 027105
3300 007174 000
3301 007176
3302 007176 000000 000000
3303 007202 000000 000000
3304 007206
3305

RATES: ; A TABLE OF THE ACTUAL BYTES TO MOVE INTO THE
; UPPER BYTE OF XCSR FOR EACH BAUD RATE
; ** NOTE: : THE VALUE INDICATED IN THE COLUMN 'OFFSET
; ** INTO TABLE' CAN BE PLACED INTO BITS<11:8>
; ** OF LOCATION '$USWR' TO CAUSE THE CORRESPONDING
; ** BAUD TO BE SELECTED IN THE DLV11-E UPON
; ** COMPLETION OF THIS TEST.

BAUD OFFSET INTO TABLE
R0050: .BYTE 010 : 50 0
R0070: .BYTE 030 : 70 1
R0110: .BYTE 050 : 110 2
R0135: .BYTE 070 : 135 3
R0150: .BYTE 110 : 150 4
R0300: .BYTE 130 : 300 5
R0600: .BYTE 150 : 600 6
R0200: .BYTE 170 : 1200 7
R1800: .BYTE 210 : 1800 10
R2000: .BYTE 230 : 2000 11
R2400: .BYTE 250 : 2400 12
R3600: .BYTE 270 : 3600 13
R4800: .BYTE 310 : 4800 14
R7200: .BYTE 330 : 7200 15
R9600: .BYTE 350 : 9600 16
R10000: .BYTE 370 : 19200 17

BAUDRATE: .ASCIZ /BAUD RATE DIDN'T CHANGE./

.EVEN
NEW: 0,0
OLD: 0,0

ENDTST

```

807

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 80
CVDVAB.P11 15-DEC-77 08:58 T32 PROGRAMMABLE BAUD RATE TEST

SEQ 0079

3306
3307

```

3308 .....
3309 .....
3310 .....
3311 *TEST 33 TRANSMITTER INTERRUPT LOGIC TEST
3312 * LOGICALLY THIS IS 4 SEPARATE TESTS
3313 * A) DOES TRANSMITTER INTERRUPT LOGIC WORK
3314 * B) AT PRIORITY OF 0
3315 * C) AND ONLY ONCE
3316 * D) BUT NOT WITH INTERRUPT ENABLE CLEAR
3317 .....
3318 007206 000004 TST33: SCOPE
3319 007210 012767 000010 MOV #10,$TIMES ;;DO 10 ITERATIONS
3320 007216 012767 000033 171742 MOV #33,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
3321 007224 005067 002520 CLR INTFLAG ;CLEAR 'INTERRUPT OCCURED' FLAG
3322 007224 005067 002520 CLR INTFLAG LET INTFLAG := #0
3323 .....
3324 .....
3325 007230 016703 172022 MOV DLVEC,R3 ;GET VECTOR ADDRESS
3326 007230 016703 172022 MOV DLVEC,R3 LET R3 := DLVEC
3327 .....
3328 007234 062703 000004 ADD #4,R3 ;FOR THE TRANSMITTER
3329 007234 062703 000004 ADD #4,R3 LET R3 := R3 + #4
3330 .....
3331 007240 010146 SETVEC R3, #INTSRV, #PR7 ;SET VECTOR TO POINT TO TRANS.SRV AT PRI
3332 007240 010146 SETVEC R3, #INTSRV, #PR7
3333 007242 010301 MOV R1,-(SP)
3334 007244 012721 011742 MOV R3,R1
3335 007250 012711 000340 MOV #INTSRV,(R1)+
3336 007254 012601 000340 MOV #PR7,(R1)
3337 007256 012601 000340 MOV (SP)+,R1
3338 007256 012767 007264 171624 MOV #64,$SLPERR BGNSUB
3339 .....
3340 007264 042777 000100 171772 BIC #XMITIE,@TCSR ;CLEAR INTERRUPT ENABLE
3341 007264 042777 000100 171772 BIC #XMITIE,@TCSR LET @TCSR := @TCSR CLR.BY #XMITIE
3342 .....
3343 .....
3344 007272 012746 000000 MOV #PRO,-(SP) ;SET IT TO 0
3345 007276 012746 007304 MOV #65$,-(SP) ;;PUT NEW PS ON STACK
3346 007302 000002 RTI ;;PUT NEW PC ON STACK
3347 007304 000002 RTI ;;POP NEW PC AND PS
3348 .....
3349 .....
3350 007304 052777 000100 171752 BIS #XMITIE,@TCSR ;NOW SET I.E. BIT
3351 007304 052777 000100 171752 BIS #XMITIE,@TCSR LET @TCSR := @TCSR SET.BY #XMITIE
3352 .....
3353 .....
3354 007312 010546 WAITMS 200. ;LET INTERRUPT HAVE TIME TO OCCUR
3355 007312 010546 WAITMS 200.
3356 007314 012745 000310 MOV R5,-(SP)
3357 007320 004767 002336 MOV #200,-(R5)
3358 007324 012605 JSR PC WAIT
3359 007324 012605 MOV (SP)+,R5
3360 .....
3361 007326 026727 002416 000001 CMP INTFLAG,#1 ;DID EXACTLY 1 INTERRUPT OCCUR
3362 007326 026727 002416 000001 CMP INTFLAG,#1 IF INTFLAG NE #1 THEN
3363 007334 001406 BEQ $153

```

D07

MAINDEC-22-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 82
 CVDVAB.P11 15-DEC-77 08:58 T33 TRANSMITTER INTERRUPT LOGIC TEST

SEQ 0081

```

3364
3365 007336
3366 007336 005767 002406          TST      INTFLAG
3367 007342 001002                      BNE      $154
3368
3369 007344
3370 007344 104106          ERROR    106
3371 007346
3372 007346 000401          BR       $155
3373 007350          $154:
3374
3375
3376 007350
3377 007350 104107          ERROR    107
3378 007352
3379 007352          $155:
3380 007352
3381 007352          $153:
3382 007352
3383
3384 007352
3385 007352 012767 007360 171530      MOV      #64$,SLPERR
3386
3387 007360
3388 007360 005067 002364          CLR      INTFLAG
3389
3390 007364
3391 007364 042777 000100 171672      BIC      #XMITIE,ATCSR
3392
3393 007372 012746 000000          MOV      #PRO,-(SP)      ;; PUT NEW PS ON STACK
3394 007376 012746 007404          MOV      #65$,-(SP)      ;; PUT NEW PC ON STACK
3395 007402 000002          RTI          ;; POP NEW PC AND PS
3396 007404          65$:
3397
3398 007404
3399 007404 010546          MOV      R5,-(SP)
3400 007406 012745 000002          MOV      #2,-(R5)
3401 007412 004767 002244          JSR      PC,WAIT
3402 007416 012605          MOV      (SP)+,R5
3403 007420
3404 007420 005767 002324          TST      INTFLAG
3405 007424 001401          BEQ      $156
3406
3407 007426
3408 007426 104110          ERROR    110
3409 007430
3410 007430          $156:
3411 007430
3412 007430 000005          RESET
3413 007432
3414
3415 007432
3416 007432 010146          MOV      R1,-(SP)      ;; PUSH R1 ON STACK
3417 007434 010246          MOV      R2,-(SP)      ;; PUSH R2 ON STACK
3418 007436 012701 000003          MOV      #R3,R1
3419 007442 010102          MOV      R1,R2

```

; NO - WAS IT 0 OR MORE THAN ONCE
 IF INTFLAG EQ #0 THEN
 ; TRANSMITTER DID NOT INTERRUPT IN TIME
 ERRHRD 106,,DIDNOT
 ELSE
 ; TWICE
 ; TRANSMITTER INTERRUPTED TWICE
 ERRHRD 107,,TWICE
 ENDIF
 ENDIF
 ENDSUB
 ; INTERRUPT WITHOUT INTERRUPT ENABLE SET
 BGNSUB
 ; CLEAR 'INTERRUPT OCCURED' FLAG
 LET INTFLAG := #0
 ; CLEAR INTERRUPT ENABLE
 LET ATCSR := ATCSR CLR.BY #XMITIE
 ; NO INTERRUPTS SHOULD OCCUR.
 ; DARE IT TO HAPPEN
 WAITMS 2
 IF INTFLAG NE #0 THEN
 ; INTERRUPT OCCURED WITH I E CLEARED
 ERRHRD 110,NOTENAB
 ENDIF
 BRESET
 ENDSUB
 ; RESTORE VECTOR AREA
 CLRVEC R3

E07

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 83
CVDVAB.P11 15-DEC-77 08:58 T33 TRANSMITTER INTERRUPT LOGIC TEST

SEQ 0082

3420	007444	062702	000002	ADD	#2,R2	
3421	007450	010221		MOV	R2,(R1)+	
3422	007452	005011		CLR	(R1)	
3423	007454	012602		MOV	(SP)+,R2	::POP STACK INTO R2
3424	007456	012601		MOV	(SP)+,R1	::POP STACK INTO R1
3425						
3426	007460				ENDTST	
3427						
3428						
3429						
3430						
3431						
3432						

```

3433      ;*****
3434      ;*****
3435      ;TEST 34      RECEIVER INTERRUPT LOGIC TEST
3436      ;            THIS TEST COVERS ALL OF THE RECEIVER
3437      ;            SIDE OF THE INTERRUPT LOGIC, BOTH DATASET
3438      ;            AND CHARACTER MODES.
3439      ;*****
3440      007460 000004      ST34: SCOPE
3441      007462 012767 000010 171470      MOV      #10, $TIMES      ;; DO 10 ITERATIONS
3442      007470 012767 000034 171502      MOV      #34, $TESTN      ;; SET TEST NUMBER IN APT MAIL BOX
3443      ;                                           ; CLEAR INTERRUPT OCCURED FLAG
3444      ;                                           ; SET UP RECEIVER INTER. VECTOR
3445      ;                                           SETVEC DLVEC, #INTSRV, #PR7
3446      007476      MOV      R1, -(SP)
3447      007476 010146      MOV      DLVEC, R1
3448      007500 016701 171552      MOV      #INTSRV (R1)+
3449      007504 012721 011742      MOV      #PR7, (R1)
3450      007510 012711 000340      MOV      (SP)+, R1
3451      007514 012601      MOV      ; PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-RCVRIE
3452      007516      BGNSUB
3453      007516 012767 007524 171364      MOV      #64$, $LPERR
3454      007524      LET INTFLAG := #0
3455      007524 005067 002220      CLR      INTFLAG
3456      ; SET MAINT. BIT
3457      007530      LET @TCSR := @TCSR SET.BY #MAINT
3458      007530 052777 000004 171526      BIS      #MAINT, @TCSR
3459      ; CLEAR INTERRUPTS
3460      007536      LET @RCSR := @RCSR CLR.BY #RCVRIE
3461      007536 042777 000100 171514      BIC      #RCVRIE, RCSR
3462      ; CHANGE PRIORITY
3463      ; TO 0
3464      007544 012746 000000      MOV      #PRO, -(SP)      ;; PUT NEW PS ON STACK
3465      007550 012746 007556      MOV      #65$, -(SP)      ;; PUT NEW PC ON STACK
3466      007554 000002      RTI      ;; POP NEW PC AND PS
3467      007556      65$:
3468      ; SEND A CHARACTER
3469      LET @TBUF := #0
3470      007556 105077 171506      CLRB      @TBUF
3471      ; WAIT A MAXIMUM
3472      ; OF 500 MSEC FOR
3473      ; RCVR RDY TO SET IN RCSR
3474      CALL TIMER IN (<#500, #RCVRDONE, RCSR, #SET>)
3475      007562
3476      007562 010546      MOV      R5, -(SP)
3477      007564 012745 177777      MOV      #SET, -(R5)
3478      007570 016745 171464      MOV      RCSR, -(R5)
3479      007574 012745 000200      MOV      #RCVRDONE, -(R5)
3480      007600 012745 000500      MOV      #500, -(R5)
3481      007604 004767 001574      JSR      PC, TIMER
3482      007610 012605      MOV      (SP)+, R5
3483      ; SET INTERRUPT ENABLE
3484      007612      LET @RCSR := @RCSR SET.BY #RCVRIE
3485      007612 052777 000100 171440      BIS      #RCVRIE, @RCSR
3486      ; LET IT COME IN.
3487      007620      WAITMS 1
3488      007620 010546      MOV      R5, -(SP)

```


G07

MAINDEC-22-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 85
 CVDVAB.P11 15-DEC-77 08:58 T34 RECEIVER INTERRUPT LOGIC TEST

SEQ 0084

```

3489 007622 012745 000001      MOV      #1,-(R5)
3490 007626 004767 002030      JSR      PC, WAIT
3491 007632 012605              MOV      (SP)+, R5
3492 007634              LET RO := ARBUF ; CLEAR RCVRDONE
3493 007634 017700 171422      MOV      ARBUF, RO
3494
3495
3496 007640              ; DID WE DO IT RIGHT?
3497 007640 026727 002104 000001  CMP      INTFLAG, #1
3498 007646 001406              BEQ      $157
3499
3500 007650              ; NONE OCCURED
3501 007650 005767 002074      TST      INTFLAG
3502 007654 001002              BNE      $160
3503
3504 007656              ; RECEIVER DID NOT INTERRUPT IN TIME
3505 007656 104111              ERROR    111
3506
3507 007660              ; TWICE OR MORE
3508 007660 000401              BR       $161
3509 007662              ELSE
3510
3511 007662              ; RECEIVER INTERRUPTED TWICE
3512 007662 104112              ERROR    112
3513
3514 007664              ENDIF
3515 007664              $161:
3516 007664              $157:
3517
3518 007664              ; RESET MAINT. BIT.
3519 007664 042777 000004 171372  BIC      #MAINT, ATCSR
3520
3521 007672              ; CLEAR INTERRUPT ENABLE
3522 007672 042777 000100 171360  BIC      #RCVRIE, ARCSR
3523 007700              ENDSUB
3524
3525
3526
3527
3528
3529
3530
3531
3532              ; PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-DATAIE
3533 007700              BGNSUB
3534 007700 012767 007706 171202  MOV      #64$, $LPERR
3535 007706 032767 020000 171304  BIT      #CABLE, $USWR
3536 007714 001004              BNE      $162
3537
3538 007716              ; CAN'T TEST WITHOUT A CABLE
3539 007716 012767 000001 171234  MOV      #1, $TIMES
3540 007724 000466              BR       TST35
3541 007726              ;;;EXIT THIS TEST
3542 007726              $162:
3543
3544 007726              ; CLEAR 'INTFLAG'
3545              LET INTFLAG := #0

```

H07

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 86
 CVDVAB.P11 15-DEC-77 08:58 T34 RECEIVER INTERRUPT LOGIC TEST

SEQ 0085

```

3545 007726 005067 002016          CLR      INTFLAG
3546                                     ;CLEAR INTERRUPTS
3547 007732                                     LET  @RCSR := @RCSR CLR.BY #DATAIE
3548 007732 042777 000040 171320      BIC      #DATAIE,@RCSR
3549                                     ;CHANGE PRIORITY
3550                                     ;...TO 0
3551 007740 012746 000000          MOV      #PRO,-(SP)      ;;PUT NEW PS ON STACK
3552 007744 012746 007752          MOV      #64$,-(SP)      ;;PUT NEW PC ON STACK
3553 007750 000002          RTI                          ;;POP NEW PC AND PS
3554 007752                                     64$:
3555 007752                                     LET  @RCSR := @RCSR CLR.BY #REQSEND
3556 007752 042777 000004 171300      BIC      #REQSEND,@RCSR
3557                                     ;SET INTERRUPT ENABLE
3558 007760                                     LET  @RCSR := @RCSR SET.BY #DATAIE
3559 007760 052777 000040 171272      BIS      #DATAIE,@RCSR
3560 007766                                     LET  @RCSR := @RCSR SET.BY #REQSEND
3561 007766 052777 000004 171264      BIS      #REQSEND,@RCSR
3562                                     ;LET IT COME IN.
3563 007774                                     WAITMS 1
3564 007774 010546          MOV      R5,-(SP)
3565 007776 012745 000001          MOV      #1,-(R5)
3566 010002 004767 001654          JSR      PC,WAIT
3567 010006 012605          MOV      (SP)+,R5
3568
3569                                     ; DID IT DO IT RIGHT?
3570 010010                                     IF INTFLAG NE #1 THEN
3571 010010 026727 001734 000001      CMP      INTFLAG,#1
3572 010016 001406          BEQ      $163
3573                                     ;NONE OCCURED
3574 010020                                     IF INTFLAG EQ #0 THEN
3575 010020 005767 001724          TST      INTFLAG
3576 010024 001002          PNE      $164
3577                                     ;DATAINT DID NOT INTERRUPT IN TIME
3578 010026                                     ERRHRD 113,,DIDNOT
3579 010026 104113          ERROR      113
3580                                     ;TWICE OR MORE
3581 010030                                     ELSE
3582 010030 000401          BR      $165
3583 010032                                     ; DATAINT INTERRUPTED TWICE
3584                                     ERRHRD 114,,TWICE
3585 010032 104114          ERROR      114
3586 010032                                     ENDIF
3587 010034                                     ENDIF
3588 010034                                     $165:
3589 010034                                     $163:
3590 010034
3591 010034          BIC      #DATAIE,@RCSR
3592 010034 042777 000040 171216      BIC      #DATAIE,@RCSR
3593 010042          BIC      #REQSEND,@RCSR
3594 010042 042777 000004 171210      BIC      #REQSEND,@RCSR
3595 010050          ENDSUB
3596
3597 010050          LET R4 := @DLVEC
3598 010050 017704 171202          MOV      @DLVEC,R4
3599 010054          CLRVEC R4
3600 010054 010146          MOV      R1,-(SP)      ;;PUSH R1 ON STACK

```


MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 87
CVDVAB.P11 15-DEC-77 08:58 T34 RECEIVER INTERRUPT LOGIC TEST

SEQ 0086

3601	010056	010246		MOV	R2, -(SP)	;; PUSH R2 ON STACK
3602	010060	012701	000004	MOV	#R4, R1	
3603	010064	010102		MOV	R1, R2	
3604	010066	062702	000002	ADD	#2, R2	
3605	010072	010221		MOV	R2, (R1)+	
3606	010074	005011		CLR	(R1)	
3607	010076	012602		MOV	(SP)+, R2	;; POP STACK INTO R2
3608	010100	012601		MOV	(SP)+, R1	;; POP STACK INTO R1
3609	010102				ENDTST	

J07

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 88
 CVDVAB.P11 15-DEC-77 08:58 T34 RECEIVER INTERRUPT LOGIC TEST

SEQ 0087

```

3610 ;*****
3611 ;*****
3612 ;TEST 35 TEST ACTUAL DATA TRANSFERED
3613 ;NON-INTERRUPT MAINTENANCE BIT SET
3614 ;*****
3615 010102 000004 000001 171046 ST35: SCOPE
3616 010104 012767 000035 171060 MOV #1,$TIMES ;;DO 1 ITERATION
3617 010112 012767 000035 171060 MOV #35,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
3618 ;SET MAINT. BIT
3619 010120 052777 000004 171136 BIS #MAINT,$TCSR LET $TCSR := $TCSR SET BY #MAINT
3620 010120 052777 000004 171136
3621 ;CHANGE PRIORITY
3622 ;TO 0
3623 010126 012746 000000 MOV #PRO,-($P) ;;PUT NEW PS ON STACK
3624 010132 012746 010140 MOV #64$,-($P) ;;PUT NEW PC ON STACK
3625 010136 000002 RTI ;;POP NEW PC AND PS
3626 010140 64$:
3627 ;GET DATA MASK.
3628 CALL DATLNG OUT ($R1)
3629 010140 162705 000002 SUB #1*2,$R5
3630 010140 004767 001412 JSR PC,DATLNG
3631 010144 012501 MOV ($R5)+,$R1
3632 010150 017700 171104 MOV $RBUF,$R0
3633 ;START CLEAN
3634 LET $R0 := $RBUF
3635 010152 017700 171104
3636 ;ALL BINARY CHAR.
3637 INCR $R2 FROM #0 TO #377 BY #1
3638 010156 005002 CLR $R2
3639 010156 000401 BR $166
3640 010160 005202 INC $R2
3641 010162 020227 000377 CMP $R2,$377
3642 010162 003047 BGT $170
3643 010164
3644 010170
3645 ;TRANSMIT CHAR IN $R2
3646 CALL TIMER IN (<$500,$XMITRDY,$TCSR,$SET>)
3647 010172 010546 MOV $R5,-($P)
3648 010172 012745 177777 MOV #SET,-($R5)
3649 010174 016745 171060 MOV $TCSR,-($R5)
3650 010200 012745 000200 MOV #XMITRDY,-($R5)
3651 010204 012745 000500 MOV #500,-($R5)
3652 010210 004767 001164 JSR PC,TIMER
3653 010214 012605 MOV ($P)+,$R5
3654 ;TRANSMIT IT
3655 LET $TBUF := $R2
3656 010222 110277 171042 MOVB $R2,$TBUF
3657 010222
3658 CALL TIMER IN (<$500,$RCVRDONE,$RCSR,$SET>)
3659 010226 010546 MOV $R5,-($P)
3660 010226 012745 177777 MOV #SET,-($R5)
3661 010230
3662
3663
3664
3665

```

K07

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 89
 CVDVAB.P11 15-DEC-77 08:58 T35 TEST ACTUAL DATA TRANSFERED

SEQ 0088

```

3666 010234 016745 171020      MOV      RCSR, -(R5)
3667 010240 012745 000200      MOV      #RCVADONE, -(R5)
3668 010244 012745 000500      MOV      #500, -(R5)
3669 010250 004767 001130      JSR      PC, TIMER
3670 010254 012605              MOV      (SP)+, R5
3671                                ;AND SAVE IT
3672 010256              LET R3 := @RBUF
3673 010256 017703 171000      MOV      @RBUF, R3
3674
3675                                ;COMPARE TO SEE IF WE RECEIVED IT ALL
3676                                ;CLEAN OFF NON-DATA BITS
3677                                ;ON BOTH TRANSMITTED AND
3678                                LET R4 := R2 CLR.BY R1
3679
3680 010262              LET R3 := R3 CLR.BY R1
3681 010262 010204      MOV      R2, R4
3682 010264 040104      BIC      R1, R4
3683 010266              LET R3 := R3 CLR.BY R1
3684 010266 040103      BIC      R1, R3
3685
3686                                ;RECEIVED DATA
3687 010270              IF R4 NE R3 THEN
3688 010270 020403      CMP      R4, R3
3689 010272 001405      BEQ      $171
3690
3691                                ;DATA COMPARE ERROR
3692 010274 104116      ERRHRD 116, COMP, SBWAS
3693 010276              EXIT TST ; ON ERROR
3694 010276 012767 000001 170654      MOV      #1, $TIMES
3695 010304 000404      BR      TST36      ;;;EXIT THIS TEST
3696 010306              ENDIF
3697 010306      $171:              ENDINC ; R2
3698 010306              BR      $167
3699 010306 000725      $170:
3700 010310              $170:
3701
3702                                ;RESET MAINT. BIT.
3703 010310              LET @TCSR := @TCSR CLR.BY #MAINT
3704 010310 042777 000004 170746      BIC      #MAINT, @TCSR
3705 010316              ENDTST
3706
3707
3708

```

L07

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 90
 CVDVAB.P11 15-DEC-77 08:58 T35 TEST ACTUAL DATA TRANSFERED

SEQ 0089

```

3709          ;*****
3710          ;*****
3711          ;TEST 36      TEST DATA THROUGH CABLE
3712          ;*****
3713          TST36: SCOPE
3714          MOV      #1,STIMES      ;;DO 1 ITERATION
3715          MOV      #36,STESTN     ;;SET TEST NUMBER IN APT MAIL BOX
3716          ;*****
3717          BIT      #CABLE,$USWR   IF #CABLE NOTSETIN $USWR THEN
3718          BNE      $172
3719          ;*****
3720          ;CAN'T TEST WITHOUT A CABLE
3721          EXIT TST
3722          MOV      #1,STIMES
3723          BR       TST37           ;;EXIT THIS TEST
3724          ;*****
3725          $172:
3726          ;DON'T USE MAINT.
3727          LET @TCSR := @TCSR CLR.BY #MAINT
3728          ;CHANGE PRIORITY
3729          ;...TO 0
3730          MOV      #PRO,-(SP)      ;;PUT NEW PS ON STACK
3731          MOV      #64$,-(SP)     ;;PUT NEW PC ON STACK
3732          RTI                    ;;POP NEW PC AND PS
3733          64$:
3734          ;*****
3735          ;GET DATA MASK
3736          CALL DATLNG OUT <R1>
3737          SUB      #1*2,R5
3738          JSR      PC,DATLNG
3739          MOV      (R5)+,R1
3740          MOV      @RBUF,R0
3741          LET R0 := @RBUF ; START CLEAN
3742          ;*****
3743          ;BINARY COUNT PATTERN
3744          INCR R2 FROM #0 TO #377 BY #1
3745          CLR      R2
3746          BR       $173
3747          $174:
3748          INC      R2
3749          $173:
3750          CMP      R2,#377
3751          BGT      $175
3752          ;*****
3753          ;TRANSMIT THE CHAR. IN R2.
3754          CALL TIMER IN (<#500,#XMITRDY,TCSR,#SET>)
3755          MOV      R5,-(SP)
3756          MOV      #SET,-(R5)
3757          MOV      TCSR,-(R5)
3758          MOV      #XMITRDY,-(R5)
3759          MOV      #500,-(R5)
3760          JSR      PC,TIMER
3761          MOV      (SP)+,R5
3762          ;*****
3763          ;START IT ON ITS WAY
3764          LET @TBUF :B= R2

```

M07

MAINDEC-22-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 91
 CVDVAB.P11 15-DEC-77 08:58 T36 TEST DATA THROUGH CABLE

SEQ 0090

```

3765 010456 110277 170606      MOV      R2,2TBUF
3766 010462
3767 010462 010546      MOV      R5, -(SP)
3768 010464 012745 177777      MOV      #SET, -(R5)
3769 010470 016745 170564      MOV      RCSR, -(R5)
3770 010474 012745 000200      MOV      #RCVRDONE, -(R5)
3771 010500 012745 000500      MOV      #500, -(R5)
3772 010504 004767 000674      JSR      PC_TIMER
3773 010510 012605      MOV      (SP)+, R5
3774
3775
3776 010512
3777 010512 017703 170544      MOV      2RBUF, R3
3778
3779
3780 010516
3781 010516 010204      MOV      R2, R4
3782 010520 040104      BIC      R1, R4
3783 010522
3784 010522 040103      BIC      R1, R3
3785
3786
3787 010524
3788 010524 020403      CMP      R4, R3
3789 010526 001405      BEQ      $176
3790
3791 010530
3792 010530 104117      ERROR    117
3793 010532
3794 010532 012767 000001 170420      MOV      #1, $TIMES
3795 010540 000401      BR       TST37
3796 010542
3797 010542      $176:
3798
3799 010542
3800 010542 000725      $175:
3801 010544      BR       $174
3802
3803
3804
3805 010544
3806
3807
3808
3809

```

CALL TIMEP IN <#500, #RCVRDONE, RCSR, #SET>

:RETRIEVE
 LET R3 := 2RBUF

:STRIP OFF JUNK ON BOTH
 LET R4 := R2 CLR.BY R1

LET R3 := R3 CLR.BY R1

:WE HAVE TROUBLE
 IF R4 NE R3 THEN

:DATA COMPARE ERROR
 ERRHRD 117,COMP,SBWAS
 EXIT TST ; ON ERROR

;;;EXIT THIS TEST
 ENDIF

ENDINC ; R2

ENDTST

```

3810      ; *****  

3811      ; *****  

3812      ; *****  

3813      ; *****  

3814      ; *****  

3815      ; *****  

3816      ; *****  

3817      ; *****  

3818      ; *****  

3819      ; *****  

3820      ; *****  

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      ; *****  

3875      ; *****  

3876      ; *****  

3877      ; *****  

3878      ; *****  

3879      ; *****  

3880      ; *****  

3881      ; *****  

3882      ; *****  

3883      ; *****  

3884      ; *****  

3885      ; *****  

3886      ; *****  

3887      ; *****  

3888      ; *****  

3889      ; *****  

3890      ; *****  

3891      ; *****  

3892      ; *****  

3893      ; *****  

3894      ; *****  

3895      ; *****  

3896      ; *****  

3897      ; *****  

3898      ; *****  

3899      ; *****  

3900      ; *****  

3901      ; *****  

3902      ; *****  

3903      ; *****  

3904      ; *****  

3905      ; *****  

3906      ; *****  

3907      ; *****  

3908      ; *****  

3909      ; *****  

3910      ; *****  

3911      ; *****  

3912      ; *****  

3913      ; *****  

3914      ; *****  

3915      ; *****  

3916      ; *****  

3917      ; *****  

3918      ; *****  

3919      ; *****  

3920      ; *****  

3921      ; *****  

3922      ; *****  

3923      ; *****  

3924      ; *****  

3925      ; *****  

3926      ; *****  

3927      ; *****  

3928      ; *****  

3929      ; *****  

3930      ; *****  

3931      ; *****  

3932      ; *****  

3933      ; *****  

3934      ; *****  

3935      ; *****  

3936      ; *****  

3937      ; *****  

3938      ; *****  

3939      ; *****  

3940      ; *****  

3941      ; *****  

3942      ; *****  

3943      ; *****  

3944      ; *****  

3945      ; *****  

3946      ; *****  

3947      ; *****  

3948      ; *****  

3949      ; *****  

3950      ; *****  

3951      ; *****  

3952      ; *****  

3953      ; *****  

3954      ; *****  

3955      ; *****  

3956      ; *****  

3957      ; *****  

3958      ; *****  

3959      ; *****  

3960      ; *****  

3961      ; *****  

3962      ; *****  

3963      ; *****  

3964      ; *****  

3965      ; *****  

3966      ; *****  

3967      ; *****  

3968      ; *****  

3969      ; *****  

3970      ; *****  

3971      ; *****  

3972      ; *****  

3973      ; *****  

3974      ; *****  

3975      ; *****  

3976      ; *****  

3977      ; *****  

3978      ; *****  

3979      ; *****  

3980      ; *****  

3981      ; *****  

3982      ; *****  

3983      ; *****  

3984      ; *****  

3985      ; *****  

3986      ; *****  

3987      ; *****  

3988      ; *****  

3989      ; *****  

3990      ; *****  

3991      ; *****  

3992      ; *****  

3993      ; *****  

3994      ; *****  

3995      ; *****  

3996      ; *****  

3997      ; *****  

3998      ; *****  

3999      ; *****  

4000      ; *****

```


3866	010646	012711	000340	MOV	#PR7,(R1)	
3867						
3868						; INITIALIZE COUNTERS
3869	010652			LET R1 :=	#-1	
3870	010652	012701	177777	MOV	#-1,R1	
3871						; RECEIVER STORAGE
3872	010656			LET R2 :=	#0	
3873	010656	005002		CLR	R2	
3874						; # OF RECEIVED CHAR. COUNT.
3875	010660			LET R4 :=	#-1	
3876	010660	012704	177777	MOV	#-1,R4	
3877						
3878						; CLEAR ERROR COUNT.
3879	010664			LET ERRCNT :=	#0	
3880	010664	005067	000066	CLR	ERRCNT	
3881						
3882	010670			BRESET		; SET UP ALL REGISTERS
3883	010670	000005		RESET		
3884						; SET UP MAINTENANCE
3885	010672			LET ATCSR :=	ATCSR SET.BY #MAINT	
3886	010672	052777	000004 170364	BIS	#MAINT,ATCSR	
3887						
3888						; SET I.E. IN TRANSMITTER
3889	010700			LET ATCSR :=	ATCSR SET.BY #XMITIE	
3890	010700	052777	000100 170356	BIS	#XMITIE,ATCSR	
3891						; AND RECEIVER
3892	010706			LET ARCSR :=	ARCSR SET.BY #RCVRIE	
3893	010706	052777	000100 170344	BIS	#RCVRIE,ARCSR	
3894						
3895						; NOW WE WAIT UNTIL R4 COUNT (RECEIVED) IS EQUAL
3896				REPEAT		
3897	010714					UNTIL R4 EQ NUMBER OR ERRCNT GT #0
3898	010714					
3899	010714					
3900	010714	020467	000040	CMP	R4,NUMBER	
3901	010720	001403		BEQ	\$201	
3902	010722	005767	000030	TST	ERRCNT	
3903	010726	003772		BLE	\$200	
3904	010730					
3905						
3906						; DATA COMPARE ERRORS.
3907	010730			IF ERRCNT NE	#0 THEN	
3908	010730	005767	000022	TST	ERRCNT	
3909	010734	001401		BEQ	\$202	
3910						; DATA COMPARE ERROR
3911	010736			ERRHRD	120,COMP,FIRST	
3912	010736	104120		ERROR	120	
3913	010740					ENDIF
3914	010740					
3915						
3916	010740			LET ATCSR :=	ATCSR CLR.BY #XMITIE	
3917	010740	042777	000100 170316	BIC	#XMITIE,ATCSR	
3918	010746			LET ARCSR :=	ARCSR CLR.BY #XMITIE	
3919	010746	042777	000100 170304	BIC	#XMITIE,ARCSR	
3920						
3921	010754			EXIT		; SKIP OVER SUPPORT ROUTINES & STORAGE

```

3922 010754 000462          BR      TST40          ;;;EXIT THIS TEST
3923
3924 010756 000000          ERRCNT: 0
3925 010760 000400          NUMBER: 400
3926 010762          000          SB: .BYTE 0
3927 010763          000          WAS: .BYTE 0
3928
3929
3930
3931 ;*****
3932 ;TRANSMIT INTERRUPT HANDLER
3933
3934 010764          BGNSRV  TRAN
3935 010764
3936 TRAN:
3937 ;*****
3938
3939 010764          ;INCREMENT CHAR COUNT
3940 010764 005201          INC      R1          LET R1 := R1 + #1
3941
3942 010766          ;SET UP FOR TRANSFER
3943 010766 010167 000030          MOV      R1,HOLD          LET HOLD := R1      CLR.BY R3
3944 010772 040367 000024          BIC      R3,HOLD
3945
3946 010776          ;AND SEND.
3947 010776 016777 000020 170264          MOV      HOLD,ATBUF          LET ATBUF := HOLD
3948
3949 011004          ;ALL DONE
3950 011004 020167 177750          CMP      R1,NUMBER          IF R1 EQ NUMBER THEN
3951 011010 001003          BNE      $203
3952
3953 011012          ;STOP INTERRUPT PROCESSING
3954 011012 042777 000100 170244          BIC      #XMITIE,ATCSR          LET ATCSR := ATCSR CLR.BY #XMITIE
3955
3956 011020          ENDIF
3957 $203:
3958 011020 000401          BR      ZZZ          ; EXIT SRV
3959
3960 011022 000000          HOLD:0
3961
3962 011024          ENDSRV
3963 011024 000002          ZZZ:      RTI
3964
3965
3966 ;*****
3967
3968 ;RECEIVER INTERRUPT HANDLER
3969 011026          BGNSRV  REC
3970 011026
3971 REC:
3972 ;*****
3973
3974 011026          ;COUNT THIS CHAR.
3975 011026 005204          INC      R4          LET R4 := R4 + #1
3976
3977 011030          ;GET CHAR IN + MASK IT
          LET R2 := ATBUF CLR.BY R3

```


3978	011030	017702	170226	MOV	ARBUF,R2	
3979	011034	040302		BIC	R3,R2	
3980						:RHLD WILL CONTAIN EXPECTED INPUT
3981	011036					LET RHLD := R4 CLR.BY R3
3982	011036	010467	000054	MOV	R4,RHLD	
3983	011042	040367	000050	BIC	R3,RHLD	
3984						
3985						:DO THEY COMPARE
3986	011046					IF R2 NE RHLD THEN
3987	011046	020267	000044	CMP	R2,RHLD	
3988	011052	001412		BEQ	\$204	
3989						:FIRST ERROR
3990	011054					IF ERRCNT EQ #0 THEN
3991	011054	005767	177676	TST	ERRCNT	
3992	011060	001005		BNE	\$205	
3993						:SAVE RECORD OF FIRST MISS
3994	011062					LET SB :B= RHLD
3995	011062	116767	000030 177672	MOVB	RHLD,SB	
3996	011070					LET WAS :B= R2
3997	011070	110267	177667	MOVB	R2,WAS	
3998	011074					ENDIF
3999	011074					:COUNT IT.
4000						LET ERRCNT := ERRCNT + #1
4001	011074					ENDIF
4002	011074	005267	177656	INC	ERRCNT	
4003	011100					
4004	011100					:ALL DONE?
4005						IF R4 EQ NUMBER THEN
4006						
4007	011100					:STOP RECEIVER INTERRUPTS
4008	011100	020467	177654	CMP	R4,NUMBER	
4009	011104	001003		BNE	\$206	
4010						LET ARCSR := ARCSR CLR.BY ARCVRIE
4011	011106					:MAIN REPEAT LOOP IS CHECKING
4012	011106	042777	000100 170144	BIC	ARCVRIE,ARCSR	
4013						ENDIF
4014	011114					
4015	011114					:FOR 'R4 = NUMBER' ALSO
4016						; EXIT SRV
4017						RHLD:0
4018	011114	000401		BR	ZZZZ	
4019						ENDSRV
4020	011116	000000				
4021	011120					ENDTST
4022	011120					
4023	011120	000002		RTI		
4024						
4025	011122					
4026						
4027						
4028						

E08

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 96
CVDVAB.P11 15-DEC-77 08:58

T37 FULL DATA TRANSFER WITH INTERRUPTS

SEQ 0095

```

4029 .....
4030 .....
4031 .....
4032 .....
4033 .....
4034 .....
4035 011122 000004 .....
4036 011124 012767 000010 170026 .....
4037 011132 012767 000040 170040 .....
4038 .....
4039 011140 .....
4040 011140 052777 000004 170116 .....
4041 .....
4042 .....
4043 011146 .....
4044 011146 017700 170110 .....
4045 .....
4046 .....
4047 .....
4048 011152 .....
4049 011152 052777 000001 170104 .....
4050 .....
4051 011160 .....
4052 011160 012777 000252 170102 .....
4053 011166 .....
4054 011166 010546 .....
4055 011170 012745 177777 .....
4056 011174 016745 170060 .....
4057 011200 012745 000200 .....
4058 011204 012745 000500 .....
4059 011210 004767 000170 .....
4060 011214 012605 .....
4061 011216 .....
4062 011216 103001 .....
4063 .....
4064 011220 .....
4065 011220 104115 .....
4066 011222 .....
4067 011222 .....
4068 .....
4069 011222 .....
4070 011222 105777 170034 .....
4071 011226 001401 .....
4072 .....
4073 011230 .....
4074 011230 104121 .....
4075 011232 .....
4076 011232 .....
4077 011232 .....
4078 011232 000005 .....
4079 011234 .....
4080 011234 000413 .....
4081 011236 051102 040505 020113 .....
4082 011244 044504 020104 047516 .....
4083 011252 020124 050505 040525 .....
4084 011260 020114 000060 .....

;*****
;*****
;TEST 40 TEST BREAK GENERATION LOGIC
; TRANSMIT KNOWN CHAR WITH BREAK SET
; AND COMPARE RECEIVED WITH D.
;*****
TST40: SCOPE
MOV #10,$TIMES ;;DO 10 ITERATIONS
MOV #40,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
;;SET MAINTENANCE BIT
LET @TCSR := @TCSR SET.BY #MAINT
LET RO := @RBUF ; CLEAR RCVRDONE JUST IN CASE
MOV @RBUF,RO
;;SET BREAK BIT
LET @TCSR := @TCSR SET.BY #BREAK
LET @TBUF := #252 ;NON-ZERO CHAR. '*'
CALL TIMER IN (<#500,#RCVRDONE,RCSR,#SET)
IF.ERROR THEN
ERRHRD 115
ENDIF
IFB @RBUF NE #0 THEN
;; BREAK DID NOT EQUAL 0
ERRHRD 121 ,BADBRK
ENDIF
BRESET ;CLEAN UP
EXIT
THIS TEST
BADBRK: .ASCIZ 'BREAK DID NOT EQUAL 0'

```

F08

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 97
CVDVAB.P11 15-DEC-77 08:58 T40 TEST BREAK GENERATION LOGIC

SEQ 0096

4085
4086 011264

ENDTST

```

4087      ;*****
4088      ;*TEST 41          NOT A TEST - SEND BACK TO LOOP
4089      ;*****
4090      ;*****
4091      011264 000004      ;ST41: SCOPE
4092      011266 012767 000001 167664      MOV      #1,STIMES      ;;DO 1 ITERATION
4093      011274 104401 011302      TYPE      65$      ;;TYPE ASCIZ STRING
4094      011300 000404      BR      64$      ;;GET OVER THE ASCIZ
4095      ;65$: .ASCIZ <CRLF>*CSR: *
4096      011312      64$:
4097      011312 016746 167736      MOV      DLADD,-(SP)      ;;SAVE DLADD FOR TYPEOUT
4098      011316 104402      TYP0C      ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
4099      011320 104401 011326      TYPE      67$      ;;TYPE ASCIZ STRING
4100      011324 000405      BR      66$      ;;GET OVER THE ASCIZ
4101      ;67$: .ASCIZ *,VECTOR: *
4102      011340      66$:
4103      011340 016746 167712      MOV      DLVEC,-(SP)      ;;SAVE DLVEC FOR TYPEOUT
4104      011344 104402      TYP0C      ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
4105      011346 104401 011354      TYPE      69$      ;;TYPE ASCIZ STRING
4106      011352 000405      BR      68$      ;;GET OVER THE ASCIZ
4107      ;69$: .ASCIZ *,ERRORS: *
4108      011366      68$:
4109      011366 016746 167520      MOV      $ERTTL,-(SP)      ;;SAVE $ERTTL FOR TYPEOUT
4110      011372 104405      TYPDS      ;;GO TYPE--DECIMAL ASCII WITH SIGN
4111      011374 005067 167512      CLR      $ERTTL      ;;RESET FOR NEXT DEVICE/PASS
4112      011400 000167 170314      JMP      LOOP      ;;BACK UP TO THE BEGINNING

```

H08

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 99
 CVDVAB.P11 15-DEC-77 08:58 T41 NOT A TEST - SEND BACK TO LOOP

SEG 0098

```

4113
4114
4115 ;;BGNMOD SUBS
4116 011404 ;;*****ROUTINE TIMER (HOWLONG,WHICHBIT,REG,SETCLR)
4117 011404
4118
4119 * ROUTINE:TIMER
4120 * THIS ROUTINE IS USED TO TEST THE STATUS OF ANY BIT
4121 * IN ANY REGISTER.
4122 *
4123 * INPUTS:
4124 * HOWLONG THE MAXIMUM AMOUNT OF TIME TO SPEND IN
4125 * THIS ROUTINE.
4126 * WHICHBIT A MASK WITH THE BIT(S) SET THAT ARE
4127 * TO BE CHECKED.
4128 * REG A POINTER TO THE REGISTER TO BE CHECKED
4129 * SETCLR THE DESIRED RESULTS
4130 * EITHER #SET OR #CLEAR
4131 *
4132 * OUTPUT:
4133 * THE 'C' BIT IS SET TO INDICATE AN ERROR
4134 * BUT IT IS TESTED BY THE IF.ERROR STATEMENT
4135 *
4136 * NOTE:: THE USE OF (R5) IS PART OF THE LINKAGE
4137 * MECHANISM BETWEEN THE CALLER AND THE CALLED
4138 ;;*****
4139
4140 000001 TRUE= 1
4141 000000 FALSE= 0
4142
4143 011404 016567 000004 000136 MOV REG(R5),REGSAV LET REGSAV := REG(R5) ; GET POINTER TO REGIST
4144 011412 016567 000000 000132 MOV HOWLONG(R5),TIMSAV LET TIMSAV := HOWLONG(R5) ; SAVE HOWLONG FOR
4145 011420 112767 000000 000126 MOVB #FALSE,FLAG LET FLAG :B= #FALSE ; INITIALIZE THE EXIT FLA
4146
4147
4148 ; START OF AN INFINITE LOOP
4149
4150 LOOP
4151 011426
4152 011426 $213:
4153
4154 011426 036577 000002 000114 BIT WHICHBIT(R5),@REGSAV IF ; TEST TO SEE IF WHICHBIT IS SET
4155 011434 001004 BNE $215 IF WHICHBIT(R5) NOTSETIN @REGSAV THEN
4156
4157 011436 112767 000000 000111 MOVB #CLR,HOLDSC LET HOLDSC :B= #CLR
4158 011444 000403 BR $216 ELSE
4159
4160 011446 112767 177777 000101 MOVB #SET,HOLDSC LET HOLDSC :B= #SET ; REMEMBER THIS
4161
4162
4163 $215:
4164
4165 $216:
4166
4167
4168 011454 ; NOW SEE IF THAT WAS WHAT WE WANTED
IFB HOLDSC EQ SETCLR(R5) THEN

```

```

4169 011454 126765 000075 000006      CMPB  HOLDSC,SETCLR(R5)
4170 011462 001003      BNE    $217
4171                                     ; JUST THE THING WE NEEDED
4172 011464                                     LET    FLAG :B= #TRUE
4173 011464 112767 000001 000062      MOVB  #TRUE,FLAG
4174 011472                                     ENDIF
4175 011472      $217:
4176
4177 011472                                     EXIFB  FLAG EQ #TRUE OR TIMSAV LE #0
4178 011472 126727 000056 000001      CMPB  FLAG,#TRUE
4179 011500 001414      BEQ    $214
4180 011502 005767 000044      TST    TIMSAV
4181 011506 003411      BLE    $214
4182                                     ; ONE WAY OR THE OTHER, WE ARE DONE
4183                                     ; IF WE ARE STILL HERE THEN HANG AROUND A WHILE
4184
4185 011510                                     WAITMS 1          ;WAIT FOR 10 MILLI-SECONDS
4186 011510 010546
4187 011512 012745 000001      MOV    R5,-(SP)
4188 011516 004767 000140      MOV    #1,-(R5)
4189 011522 012605      JSR    PC,WAIT
4190 011524      MOV    (SP)+,R5
4191 011524 005367 000022      DEC    TIMSAV
4192 011530      LET    TIMSAV := TIMSAV - #1 ; COUNTING DOWN
4193 011530 000736      ENDLOOP      ; CONTINUED AT THE TOP
4194 011532      BR      $213
4195      $214:
4196      ; ONLY 2 WAYS TO GET HERE
4197      ; 1). WE RAN OUT OF TIME---ERROR !!
4198      ; 2). THE BIT IS IN THE CORRECT CONDITION--GOOD !!
4199
4200 011532                                     IFB    FLAG EQ #TRUE THEN
4201 011532 126727 000016 000001      CMPB  FLAG,#TRUE
4202 011540 001001      BNE    $220
4203 011542                                     RETURN NO.ERROR      ; GOOD
4204 011542 000405      BR      $211
4205 011544                                     ENDIF
4206 011544      $220:
4207 011544                                     RETURN ERROR          ; BAD
4208 011544 000261
4209 011546 000404      SEC
4210      BR      $212
4211 011550 000000
4212 011552 000000
4213 011554 000
4214 011555 000
4215                                     REGSAV: .WORD 0
4216                                     TIMSAV: .WORD 0
4217                                     FLAG: .BYTE 0
4218                                     HOLDSC: .BYTE 0
4219                                     ; WE ARE DONE GO BACK HOME
4220 011556      ENDRTN
4221 011556      $211:
4222 011556 000241      CLC
4223 011560      $212:
4224 011560 000207      RTS    PC

```



```

4221
4222
4223 011562
4224 011562
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238 011562
4239 011562 005065 000000
4240 011566
4241 011566 016767 167426 000062
4242 011574 016746 000056
4243 011600 042716 000017
4244 011604 042667 000046
4245
4246 011610
4247 011610 012767 000001 167454
4248 011616 000402
4249 011620
4250 011620 005267 167446
4251 011624
4252 011624 026767 167442 000024
4253 011632 003006
4254 011634
4255 011634 006365 000000
4256 011640
4257 011640 052765 000001 000000
4258 011646
4259 011646 000764
4260 011650
4261 011650
4262 011650 005165 000000
4263 011654
4264 011654 000401
4265 011656 000000
4266 011660
4267 011660
4268 011660
4269 011660 000207

*****
ROUTINE DATLNG <MASK>
DATLNG:
; ROUTINE:DATLNG
; THIS ROUTINE SETS UP A MASK FOR DATA, WITH
; INPUT - NOTHING IS PASSED TO THIS ROUTINE
; BUT GLOBAL INFORMATION IS ASSUMED TO EXIST:
; SUSWR-- THE WORD FOR SOFTWARE PARAMETERS
; DATA-- A MASK FOR THE LOCATION OF THE OCTAL
;         NUMBER OF DATA BITS
; OUTPUT----
; MASK-- A MASK OF BINARY ONES RIGHT-JUSTIFIED
;        THE NUMBER OF WHICH IS DEFINED IN SUSWR WORD.
*****
; ;
                                LET MASK(R5) := #0 ; START
                                LET NUMBR := SUSWR AND #DATA
                                MOV $SUSWR, NUMBR
                                MOV NUMBR, -(SP)
                                BIC #DATA, (SP)
                                BIC (SP)+, NUMBR
                                INCR I FROM #1 TO NUMBR BY #1
                                MOV #1, I
                                BR $223
                                INC I
                                $223: CMP I, NUMBR
                                BGT $225
                                ASL MASK(R5)
                                LET MASK(R5) := MASK(R5) SHIFT #1
                                LET MASK(R5) := MASK(R5) SET.BY #1
                                BIS #1, MASK(R5)
                                BR $224
                                $225: COM MASK(R5)
                                RETURN
                                BR NUMBR:0 $221
                                $221:
                                $222:
                                RTS PC
                                ENDRTN

```

4270
 4271
 4272 011662
 4273 011662
 4274
 4275
 4276
 4277
 4278
 4279
 4280
 4281 011662 010146
 4282 011664 010246
 4283 011666 010346
 4284 011670
 4285 011670 016501 000000
 4286 011674
 4287 011674 012702 000001
 4288 011700 000402
 4289 011702
 4290 011702 062702 000001
 4291 011706
 4292 011706 020201
 4293 011710 101010
 4294 011712
 4295 011712 005003
 4296 011714 000401
 4297 011716
 4298 011716 005203
 4299 011720
 4300 011720 020327 000100
 4301 011724 003001
 4302 011726
 4303 011726 000773
 4304 011730
 4305 011730
 4306 011730 000764
 4307 011732
 4308 011732 012603
 4309 011734 012602
 4310 011736 012601
 4311 011740
 4312 011740
 4313 011740
 4314 011740 000207

```

*****
ROUTINE WAIT      (TIME)
WAIT:
; * ROUTINE:WAIT
; *   THIS ROUTINE IS USED TO DELAY EXECUTION OF THE
; *   MAIN PROGRAM FOR A SPECIFIED AMOUNT OF TIME.
; *   THIS IS ACCOMPLISHED BY INCREMENTING A
; *   REGISTER UP TO A LIMIT. THE INNER LOOP IS SET
; *   TO APPROXIMATE 1 MILLI SEC.
*****
      MOV     R1,-(SP)      ;; PUSH R1 ON STACK
      MOV     R2,-(SP)      ;; PUSH R2 ON STACK
      MOV     R3,-(SP)      ;; PUSH R3 ON STACK
      LET R1 := TIME(R5)
      MOV     TIME(R5),R1
      INCRU R2 FROM #1 TO R1 BY #1
      MOV     #1,R2
      BR      $230
$231:  ADD     #01,R2
$230:  CMP     R2,R1
      BHI     $232
      INCR R3 FROM #0 TO #100 BY #1
      CLR     R3
      BR      $233
$234:  INC     R3
$233:  CMP     R3,#100
      BGT     $235
      ENDINC
      BR      $234
$235:  ENDINC
      BR      $231
$232:  MOV     (SP)+,R3      ;; POP STACK INTO R3
      MOV     (SP)+,R2      ;; POP STACK INTO R2
      MOV     (SP)+,R1      ;; POP STACK INTO R1
ENDRTN
$226:
$227:  RTS      PC
  
```


MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 103
 CVDVAB.P11 15-DEC-77 08:58 ROUTINE - WAIT

SEQ 0102

```

4315
4316
4317
4318 011742
4319
4320
4321
4322
4323
4324
4325
4326
4327 011742
4328 011742 005267 000002
4329 011746
4330 011746 000002
4331 011750 000000

      .SBTTL INTSRV  INTERRUPT SERVICE ROUTINE
      ;*****
      INTSRV:
      ;* SERVICE ROUTINE: INTSRV
      ;*   THIS GLOBAL ROUTINE DOES NOTHING BUT INCREMENT
      ;*   'INTFLAG' EACH TIME IT IS CALLED.  IT ASSUMES
      ;*   THAT THE MAIN CALLING ROUTINE WILL KNOW WHAT
      ;*   TO LOOK FOR.
      ;*****
      ;ADD 1 TO 'INTERRUPT OCCURED' FLAG
      LET INTFLAG := INTFLAG + #1
      INC INTFLAG
      ENDSRV
      RTI
      INTFLAG: 0
  
```

```

4332
4333 011752 ROUTINE MYTYPE
4334 011752 MYTYPE:
4335 ;*****
4336 011752 104401 011760 ; TYPE 65$ ; TYPE ASCII STRING
4337 011756 000405 ; BR 64$ ; GET OVER THE ASCII
4338 ; 65$: .ASCIIZ <CRLF>*TEST * *
4339 64$: MOV $TESTN,-(SP) ; SAVE $TESTN FOR TYPEOUT
4340 011772 016746 167202 ; TPOC ; GO TYPE--OCTAL ASCII(ALL DIGITS)
4341 011776 104402 ; TYPE 67$ ; TYPE ASCII STRING
4342 012000 104401 012006 ; BR 66$ ; GET OVER THE ASCII
4343 012004 000405 ; 67$: .ASCIIZ *,ERROR * *
4344 ; 66$:
4345 012020 ; MOV $ITEMB,$FATAL ; APT FATAL ERROR NUMBER
4346 012020 116767 167070 167150 ; MOV $FATAL,-(SP) ; SAVE $FATAL FOR TYPEOUT
4347 012026 016746 167144 ; TPOC ; GO TYPE--OCTAL ASCII(ALL DIGITS)
4348 012032 104402 ; TYPE 69$ ; TYPE ASCII STRING
4349 012034 104401 012042 ; BR 68$ ; GET OVER THE ASCII
4350 012040 000404 ; 69$: .ASCIIZ *,PC = *
4351 ; 68$:
4352 012052 ; MOV $ERRPC,-(SP) ; SAVE $ERRPC FOR TYPEOUT
4353 012052 016746 167040 ; TPOC ; GO TYPE--OCTAL ASCII(ALL DIGITS)
4354 012056 104402 ; TYPE 71$ ; TYPE ASCII STRING
4355 012060 104401 012066 ; BR 70$ ; GET OVER THE ASCII
4356 012064 000404 ; 71$: .ASCIIZ *,CSR: *
4357 ; 70$:
4358 012076 ; MOV DLADD,-(SP) ; SAVE DLADD FOR TYPEOUT
4359 012076 016746 167152 ; TPOC ; GO TYPE--OCTAL ASCII(ALL DIGITS)
4360 012102 104402 ; TYPE 73$ ; TYPE ASCII STRING
4361 012104 104401 012112 ; BR 72$ ; GET OVER THE ASCII
4362 012110 000405 ; 73$: .ASCIIZ *,VECTOR: *
4363 ; 72$:
4364 012124 ; MOV DLVEC,-(SP) ; SAVE DLVEC FOR TYPEOUT
4365 012124 016746 167126 ; TPOC ; GO TYPE--OCTAL ASCII(ALL DIGITS)
4366 012130 104402
4367 012132 ENORTN
4368 012132 $236:
4369 012132 $237:
4370 012132 000207 RTS PC

```

```

4371 012134 ROUTINE CYCLE
4372 012134 CYCLE:
4373 *****
4374 * ROUTINE: CYCLE
4375 * THIS ROUTINE CAUSES ADRS TO POINT TO THE
4376 * ADDRESS OF DLV11-E UNDER TEST, ADRS +2 TO
4377 * POINT TO THE VECTOR OF THE DLV11-E UNDER TEST.
4378 * IT KEEPS TRACK OF THE CURRENT DEVICE AND BIT
4379 * MASKS.
4380 *****
4381 012134 REPEAT
4382 012134
4383 012134
4384 012134 005767 000122 TST BITMASK
4385 012140 001027 BNE $243 IF BITMASK EQ #0 THEN
4386 012142 IF INITFLAG EQ #1 THEN
4387 012142 026727 000116 000001 CMP INITFLAG,#1
4388 012150 001003 BNE $244
4389 012152 LET INITFLAG := #0
4390 012152 005067 000106 CLR INITFLAG
4391 012156 ELSE
4392 012156 000403 BR $245
4393 012160 $244: CALL $EOP ; AS A SUBROUTINE
4394 012160 JSR PC,$EOP
4395 012160 004767 000110
4396 SPECIALADDRESS: ; BECAUSE $EOP RETURNS AS A JUMP
4397 012164 LET RO := POP
4398 012164
4399 012164 012600 MOV (SP)+,RO
4400 012166 ENDF
4401 012166 $245: LET BITMASK := #1
4402 012166 LET $DEVCT := #1
4403 012166 012767 000001 000066 MOV #1,BITMASK
4404 012174 012767 000001 167002 MOV #1,$DEVCT
4405 012202 LET ADDRESS := $BASE
4406 012202 016767 167042 000056 MOV $BASE,ADDRESS
4407 012210 LET VECTOR := $VECT1
4408 012210 016767 167030 000052 MOV $VECT1,VECTOR
4409 012216 ELSE
4410 012216 000410 BR $246
4411 012220 $243: LET R4 := #10
4412 012220 012704 000010 MOV #10,R4
4413 012224 LET BITMASK := BITMASK ROTATE 1
4414 012224 006167 000032 ROL BITMASK
4415 012230 LET ADDRESS := ADDRESS + R4
4416 012230 060467 000032 ADD R4,ADDRESS
4417 012234 LET VECTOR := VECTOR + R4
4418 012234 060467 000030 ADD R4,VECTOR
4419 012240 ENDF
4420 012240 $246: UNTIL BITMASK SETIN $DEVM
4421 012240 036767 000016 167004 BIT BITMASK,$DEVM
4422 012240 001732 BEQ $242
4423
4424
4425
4426

```

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 106
CVDVAB.P11 15-DEC-77 08:58 ROUTINE, - CYCLE

SEQ 0105

```
4427 012250          LET ADRS := #ADDRESS
4428 012250 012701 012266      MOV  #ADDRESS,ADRS
4429 012254          LET $DEVCT := $DEVCT + #1
4430 012254 005267 166724      INC  $DEVCT
4431 012260          RETURN
4432 012260 000404          BR    $240
4433 012262 000000      BITMASK: 0
4434 012264 000001      INITFLAG: 1
4435 012266 000000      ADDRESS: 0
4436 012270 000000      VECTOR: 0
4437
4438 012272          ENDRTN
4439 012272
4440 012272
4441 012272 000207      $240:
4442                      $241:
4443                      RTS    PC
```

MAINDEC-ZZ-CVDVA-8 MACY11 30(1046)
CVDVAB.P11 15-DEC-77 08:58

19-DEC-77 08:25 PAGE 107
END OF PASS ROUTINE

SEQ 0106

```

4444 .SBTTL END OF PASS ROUTINE
4445
4446 *****
4447 *INCREMENT THE PASS NUMBER ($PASS)
4448 *INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
4449 *TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
4450 *IF THERES A MONITOR GO TO IT
4451 *IF THERE ISN'T JUMP TO SPECIALADDRESS
4452
4453 $EOP:
4454 SCOPE
4455 CLR $TSTNM ;; ZERO THE TEST NUMBER
4456 CLR $TIMES ;; ZERO THE NUMBER OF ITERATIONS
4457 INC $PASS ;; INCREMENT THE PASS NUMBER
4458 BIC #100000,$PASS ;; DON'T ALLOW A NEG. NUMBER
4459 DEC (PC)+ ;; LOOP?
4460 $EOPCT: .WORD 1
4461 BGT $DOAGN ;; YES
4462 MOV (PC)+,$(PC)+ ;; RESTORE COUNTER
4463 $ENDCT: .WORD 1
4464 $EOPCT
4465 TYPE $ENOMG ;; TYPE "END PASS #"
4466 MOV $PASS,-(SP) ;; SAVE $PASS FOR TYPEOUT
4467 TYPDS ;; GO TYPE--DECIMAL ASCII WITH SIGN
4468 TYPE $ENULL ;; TYPE A NULL CHARACTER
4469 $GET42: MOV #42,R0 ;; GET MONITOR ADDRESS
4470 BEQ $DOAGN ;; BRANCH IF NO MONITOR
4471 RESET ;; CLEAR THE WORLD
4472 $ENDAD: JSR PC,(R0) ;; GO TO MONITOR
4473 NOP ;; SAVE ROOM
4474 NOP ;; FOR
4475 NOP ;; ACT11
4476 $DOAGN:
4477 JMP $(PC)+ ;; RETURN
4478 $RTNAD: .WORD SPECIALADDRESS
4479 $ENULL: .BYTE -1,-1,0 ;; NULL CHARACTER STRING
4480 $ENDMG: .ASCIIZ <15><12>/END PASS #/
4481
4482

```

012274	000004			
012274	005067	166600		
012302	005067	166652		
012306	005267	166670		
012312	042767	100000	166662	
012320	005327			
012322	000001			
012324	003022			
012326	012737			
012330	000001			
012332	012322			
012334	104401	012401		
012340	016746	166636		
012344	104405			
012346	104401	012376		
012352	013700	000042		
012356	001405			
012360	000005			
012362	004710			
012364	000240			
012366	000240			
012370	000240			
012372				
012372	000137			
012374	012164			
012376	377	377	000	
012401	015	042412	042116	
012406	050040	051501	020123	
012414	000043			

.SBTTL POWER DOWN AND UP ROUTINES

```

4483
4484
4485
4486
4487 012416 012737 012562 000024 $PWRDN: MOV $SILLUP,2#PWRVEC ;;SET FOR FAST UP
4488 012424 012737 000340 000026 MOV #340,2#PWRVEC+2 ;;PRIO:7
4489 012432 010046 MOV R0,-(SP) ;;PUSH R0 ON STACK
4490 012434 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
4491 012436 010246 MOV R2,-(SP) ;;PUSH R2 ON STACK
4492 012440 010346 MOV R3,-(SP) ;;PUSH R3 ON STACK
4493 012442 010446 MOV R4,-(SP) ;;PUSH R4 ON STACK
4494 012444 010546 MOV R5,-(SP) ;;PUSH R5 ON STACK
4495 012446 017746 166466 MOV @SWR,-(SP) ;;PUSH @SWR ON STACK
4496 012452 010667 000110 MOV SP,$SAVR6 ;;SAVE SP
4497 012456 012737 012470 000024 MOV $PWRUP,2#PWRVEC ;;SET UP VECTOR
4498 012464 000000 HALT
4499 012466 000776 BR .-2 ;;HANG UP
4500
4501
4502
4503 012470 012737 012562 000024 $PWRUP: MOV $SILLUP,2#PWRVEC ;;SET FOR FAST DOWN
4504 012476 016706 000064 MOV $SAVR6,SP ;;GET SP
4505 012502 005067 000060 CLR $SAVR6 ;;WAIT LOOP FOR THE TTY
4506 012506 005267 000054 1$: INC $SAVR6 ;;WAIT FOR THE INC
4507 012512 001375 BNE 1$ ;;OF WORD
4508 012514 012677 166420 MOV (SP)+,@SWR ;;POP STACK INTO @SWR
4509 012520 012605 MOV (SP)+,R5 ;;POP STACK INTO R5
4510 012522 012604 MOV (SP)+,R4 ;;POP STACK INTO R4
4511 012524 012603 MOV (SP)+,R3 ;;POP STACK INTO R3
4512 012526 012602 MOV (SP)+,R2 ;;POP STACK INTO R2
4513 012530 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
4514 012532 012600 MOV (SP)+,R0 ;;POP STACK INTO R0
4515 012534 012737 012416 000024 MOV $PWRDN,2#PWRVEC ;;SET UP THE POWER DOWN VECTOR
4516 012542 012737 000340 000026 MOV #340,2#PWRVEC+2 ;;PRIO:7
4517 012550 104401 TYPE $PWRMG: .WORD $POWER ;;REPORT THE POWER FAILURE
4518 012552 012570 $PWRMG: MOV (PC)+,(SP) ;;POWER FAIL MESSAGE POINTER
4519 012554 012716 $PWRAD: .WORD START ;;RESTART AT START
4520 012556 001336 RTI ;;RESTART ADDRESS
4521 012560 000002 $SILLUP: HALT
4522 012562 000000 BR .-2 ;;THE POWER UP SEQUENCE WAS STARTED
4523 012564 000776 $SAVR6: 0 ;;BEFORE THE POWER DOWN WAS COMPLETE
4524 012566 000000 $POWER: .ASCIIZ <15><12>"POWER" ;;PUT THE SP HERE
4525 012570 005015 047520 042527
4526 012576 000122
4527

```



```

4528 .SBTTL TYPE ROUTINE
4529
4530 *****
4531 *ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
4532 *THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
4533 *NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
4534 *NOTE2: $FILLC CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
4535 *NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
4536
4537 *CALL:
4538 *1) USING A TRAP INSTRUCTION
4539 *      TYPE      ,MESADR      ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
4540 *OR
4541 *      TYPE
4542 *      MESADR
4543
4544
4545 012600 105767 166353 $TYPE: TSTB $TPFLG      ;; IS THERE A TERMINAL?
4546 012604 100002      BPL 1$          ;; BR IF YES
4547 012606 000000      HALT          ;; HALT HERE IF NO TERMINAL
4548 012610 000430      BR 3$          ;; LEAVE
4549 012612 010046 1$:  MOV RO, -(SP)  ;; SAVE RO
4550 012614 017600      MOV 22(SP), RO ;; GET ADDRESS OF ASCIZ STRING
4551 012620 122767 000001 166366  CMPB #APTENV, $ENV  ;; RUNNING IN APT MODE
4552 012626 001011      BNE 62$      ;; NO GO CHECK FOR APT CONSOLE
4553 012630 132767 000100 166357  BITB #APTSPool, $ENVM ;; SPOOL MESSAGE TO APT
4554 012636 001405      BEQ 62$      ;; NO GO CHECK FOR CONSOLE
4555 012640 010067 000004      MOV RO, 61$  ;; SETUP MESSAGE ADDRESS FOR APT
4556 012644 004767 000774      JSR PC, $ATY3 ;; SPOOL MESSAGE TO APT
4557 012650 000000 61$:  .WORD 0      ;; MESSAGE ADDRESS
4558 012652 132767 000040 166335 62$:  BITB #APTCSUP, $ENVM ;; APT CONSOLE SUPPRESSED
4559 012660 001003      BNE 60$      ;; YES, SKIP TYPE OUT
4560 012662 112046 2$:  MOVB (RO)+, -(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
4561 012664 001005      BNE 4$      ;; BR IF IT ISN'T THE TERMINATOR
4562 012666 005726      TST (SP)+    ;; IF TERMINATOR POP IT OFF THE STACK
4563 012670 012600 60$:  MOV (SP)+, RO  ;; RESTORE RO
4564 012672 062716 000002 3$:  ADD #2, (SP) ;; ADJUST RETURN PC
4565 012676 000002      RTI          ;; RETURN
4566 012700 122716 000011 4$:  CMPB #HT, (SP) ;; BRANCH IF <HT>
4567 012704 001430      BEQ 8$      ;;
4568 012706 122716 000200      CMPB #CRLF, (SP) ;; BRANCH IF NOT <CRLF>
4569 012712 001006      BNE 5$      ;;
4570 012714 005726      TST (SP)+    ;; POP <CR><LF> EQUIV
4571 012716 104401      TYPE          ;; TYPE A CR AND LF
4572 012720 001171      $CRLF
4573 012722 105067 000130      CLRB $CHARCNT ;; CLEAR CHARACTER COUNT
4574 012726 000755      BR 2$      ;; GET NEXT CHARACTER
4575 012730 004767 000056 5$:  JSR PC, $TYPEC ;; GO TYPE THIS CHARACTER
4576 012734 126726 166216 6$:  CMPB $FILLC, (SP)+ ;; IS IT TIME FOR FILLER CHARS.?
4577 012740 001350      BNE 2$      ;; IF NO GO GET NEXT CHAR.
4578 012742 016746 166206      MOV $NULL, -(SP) ;; GET # OF FILLER CHARS. NEEDED
4579                                     ;; AND THE NULL CHAR.
4580 012746 105366 000001 7$:  DECB 1(SP) ;; DOES A NULL NEED TO BE TYPED?
4581 012752 002770      BLT 6$      ;; BR IF NO--GO POP THE NULL OFF OF STACK
4582 012754 004767 000032      JSR PC, $TYPEC ;; GO TYPE A NULL
4583 012760 105367 000072      DECB $CHARCNT ;; DO NOT COUNT AS A COUNT

```

MAINDEC-22-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 110
 CVDVAB.P11 15-DEC-77 08:58 TYPE ROUTINE

SEQ 0109

```

4584 012764 000770          BR      7$          ;;LOOP
4585
4586          ;HORIZONTAL TAB PROCESSOR
4587
4588 012766 112716 000040      8$:      MOVB      #' (SP)          ;; REPLACE TAB WITH SPACE
4589 012772 004767 000014      9$:      JSR      PC,$TYPEC          ;; TYPE A SPACE
4590 012776 132767 000007 000052      BITB      #7,$CHARCNT          ;; BRANCH IF NOT AT
4591 013004 001372          BNE      9$          ;; TAB STOP
4592 013006 005726          TST      (SP)+          ;; POP SPACE OFF STACK
4593 013010 000724          BR      2$          ;; GET NEXT CHARACTER
4594 013012 105777 166132      $TYPEC: TSTB      2$TPS          ;; WAIT UNTIL PRINTER IS READY
4595 013016 100375          BPL      $TYPEC
4596 013020 116677 000002 166124      MOVB      2(SP),2$TPB          ;; LOAD CHAR TO BE TYPED INTO DATA REG.
4597 013026 122766 000015 000002      CMPB      #CR,2(SP)          ;; IS CHARACTER A CARRIAGE RETURN?
4598 013034 001003          BNE      1$          ;; BRANCH IF NO
4599 013036 105067 000014          CLRB      $CHARCNT          ;; YES--CLEAR CHARACTER COUNT
4600 013042 000406          BR      $TYPEX          ;; EXIT
4601 013044 122766 000012 000002      1$:      CMPB      #LF,2(SP)          ;; IS CHARACTER A LINE FEED?
4602 013052 001402          BEQ      $TYPEX          ;; BRANCH IF YES
4603 013054 105227          INCB      (PC)+          ;; COUNT THE CHARACTER
4604 013056 000000      $CHARCNT: .WORD      0          ;; CHARACTER COUNT STORAGE
4605 013060 000207      $TYPEX: RTS      PC
4606

```



```

4607 .SBTTL TTY INPUT ROUTINE
4608
4609 ;*****
4610 ;ENABL LSB
4611
4612 ;*****
4613 ;SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
4614 ;ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
4615 ;SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
4616 ;WHEN OPERATING IN TTY FLAG MODE.
4617 013062 022767 000176 166050 $CKSWR: CMP      #SWREG,SWR      ;; IS THE SOFT-SWR SELECTED?
4618 013070 001074          BNE      15$      ;; BRANCH IF NO
4619 013072 105777 166046          TSTB     2$TKS      ;; CHAR THERE?
4620 013076 100071          BPL      15$      ;; IF NO, DON'T WAIT AROUND
4621 013100 117746 166042          MOVB     2$TKB, -(SP)      ;; SAVE THE CHAR
4622 013104 042716 177600          BIC      #177, (SP)      ;; STRIP-OFF THE ASCII
4623 013110 022726 000007          CMP      #7, (SP)+      ;; IS IT A CONTROL G?
4624 013114 001067          BNE      15$      ;; NO RETURN TO USER
4625 013116 126727 166012 000001          CMPB     $AUTOB, #1      ;; ARE WE RUNNING IN AUTO-MODE?
4626 013124 001456          BEQ      15$      ;; BRANCH IF YES
4627
4628 013126 104401 013607          $GTSWR: TYPE     , $CNTLG      ;; ECHO THE CONTROL-G (1G)
4629 013132 104401 013614          TYPE     , $MSWR      ;; TYPE CURRENT CONTENTS
4630 013136 016746 165034          MOV      SWREG, -(SP)      ;; SAVE SWREG FOR TYPEOUT
4631 013142 104402          TYPEOC      ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
4632 013144 104401 013625          TYPE     , $MNEW      ;; PROMPT FOR NEW SWR
4633 013150 005046          19$: CLR      -(SP)      ;; CLEAR COUNTER
4634 013152 005046          CLR      -(SP)      ;; THE NEW SWR
4635 013154 105777 165764          7$: TSTB     2$TKS      ;; CHAR THERE?
4636 013160 100375          BPL      7$      ;; IF NOT TRY AGAIN
4637
4638 013162 117746 165760          MOVB     2$TKB, -(SP)      ;; PICK UP CHAR
4639 013166 042716 177600          BIC      #177, (SP)      ;; MAKE IT 7-BIT ASCII
4640
4641
4642
4643 013172 021627 000025          9$: CMP      (SP), #25      ;; IS IT A CONTROL-U?
4644 013176 001005          BNE      10$      ;; BRANCH IF NOT
4645 013200 104401 013602          TYPE     , $CNTLU      ;; YES, ECHO CONTROL-U (1U)
4646 013204 062706 000006          20$: ADD      #6, SP      ;; IGNORE PREVIOUS INPUT
4647 013210 000757          BR      19$      ;; LET'S TRY IT AGAIN
4648
4649
4650 013212 021627 000015          10$: CMP      (SP), #15      ;; IS IT A <CR>?
4651 013216 001022          BNE      16$      ;; BRANCH IF NO
4652 013220 005766 000004          TST      4(SP)      ;; YES, IS IT THE FIRST CHAR?
4653 013224 001403          BEQ      11$      ;; BRANCH IF YES
4654 013226 016677 000002 165704          MOV      2(SP), 2$SWR      ;; SAVE NEW SWR
4655 013234 062706 000006          11$: ADD      #6, SP      ;; CLEAR UP STACK
4656 013240 104401 001171          14$: TYPE     , $CRLF      ;; ECHO <CR> AND <LF>
4657 013244 126727 165665 000001          CMPB     $INTAG, #1      ;; RE-ENABLE TTY KBD INTERRUPTS?
4658 013252 001003          BNE      15$      ;; BRANCH IF NOT
4659 013254 012777 000100 165662          MOV      #100, 2$TKS      ;; RE-ENABLE TTY KBD INTERRUPTS
4660 013262 000002          15$: RTI      ;; RETURN
4661 013264 004767 177522          16$: JSR      PC, $TYPEC      ;; ECHO CHAR
4662 013270 021627 000060          CMP      (SP), #60      ;; CHAR < 0?

```

4663 013274 002420
4664 013276 021627 000067
4665 013302 003015
4666 013304 042726 000060
4667 013310 005766 000002
4668 013314 001403
4669 013316 006316
4670 013320 006316
4671 013322 006316
4672 013324 005266 000002
4673 013330 056616 177776
4674 013334 000707
4675 013336 104401 001170
4676 013342 000720

BLT 18\$
CMP (SP),#67
BGT 18\$
BIC #60,(SP)+
TST 2(SP)
BEQ 17\$
ASL (SP)
ASL (SP)
ASL (SP)
17\$: INC 2(SP)
BIS -2(SP),(SP)
BR 7\$
18\$: TYPE \$QUES
BR 20\$
.DSABL L\$B

:: BRANCH IF YES
:: CHAR > 7?
:: BRANCH IF YES
:: STRIP-OFF ASCII
:: IS THIS THE FIRST CHAR
:: BRANCH IF YES
:: NO, SHIFT PRESENT
:: CHAR OVER TO MAKE
:: ROOM FOR NEW ONE.
:: KEEP COUNT OF CHAR
:: SET IN NEW CHAR
:: GET THE NEXT ONE
:: TYPE ?(CR)(LF)
:: SIMULATE CONTROL-U

:: *****

:: 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

4688 013344 011646
4689 013346 016666 000004 000002
4690 013354 105777 165564
4691 013360 100375
4692 013362 117766 165560 000004
4693 013370 042766 177600 000004
4694 013376 026627 000004 000023
4695 013404 001013
4696 013406 105777 165532
4697 013412 100375
4698 013414 117746 165526
4699 013420 042716 177600
4700 013424 022627 000021
4701 013430 001366
4702 013432 000750
4703 013434 026627 000004 000140
4704 013442 002407
4705 013444 026627 000004 000175
4706 013452 003003
4707 013454 042766 000040 000004
4708 013462 000002

\$RDCHR: MOV (SP),-(SP)
MOV 4(SP),2(SP)
1\$: TSTB 2\$TKS
BPL 1\$
MOV 2\$TKB,4(SP)
BIC #177,4(SP)
CMP 4(SP),#23
BNE 3\$
2\$: TSTB 2\$TKS
BPL 2\$
MOV 2\$TKB,-(SP)
BIC #177,4(SP)
CMP (SP)+,#21
BNE 2\$
BR 1\$
3\$: CMP 4(SP),#140
BLT 4\$
CMP 4(SP),#175
BGT 4\$
BIC #40,4(SP)
4\$: RTI

:: PUSH DOWN THE PC
:: SAVE THE PS
:: WAIT FOR
:: A CHARACTER
:: READ THE TTY
:: GET RID OF JUNK IF ANY
:: IS IT A CONTROL-S?
:: BRANCH IF NO
:: WAIT FOR A CHARACTER
:: LOOP UNTIL ITS THERE
:: GET CHARACTER
:: MAKE IT 7-BIT ASCII
:: IS IT A CONTROL-Q?
:: IF NOT DISCARD IT
:: YES, RESUME
:: IS IT UPPER CASE?
:: BRANCH IF YES
:: IS IT A SPECIAL CHAR?
:: BRANCH IF YES
:: MAKE IT UPPER CASE
:: GO BACK TO USER

:: *****

:: THIS ROUTINE WILL INPUT A STRING FROM THE TTY

:: CALL:

:: RDLIN
:: RETURN HERE

:: INPUT A STRING FROM THE TTY
:: ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
:: TERMINATOR WILL BE A BYTE OF ALL 0'S

4716 013464 010346
4717 013466 012703 013572
4718 013472 022703 013602

\$RDLIN: MOV R3, -(SP)
1\$: MOV #TTYIN, R3
2\$: CMP #TTYIN+8, R3

:: SAVE R3
:: GET ADDRESS
:: BUFFER FULL?

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 113
 CVDVAB.P11 15-DEC-77 08:58 TTY INPUT ROUTINE

SEQ 0112

4719	013476	101405			BLOS	4\$		;; BR IF YES
4720	013500	104410			RDCHR			;; GO READ ONE CHARACTER FROM THE TTY
4721	013502	112613			MOV8	(SP)+, (R3)		;; GET CHARACTER
4722	013504	122713	000177		CMPB	#177, (R3)		;; IS IT A RUBOUT
4723	013510	001003			BNE	3\$		;; SKIP IF NOT
4724	013512	104401	001170		TYPE	\$QUES		;; TYPE A '?'
4725	013516	000763			BR	1\$		;; CLEAR THE BUFFER AND LOOP
4726	013520	111367	000044		MOV8	(R3), 9\$		;; ECHO THE CHARACTER
4727	013524	104401	013570		TYPE	9\$		
4728	013530	122723	000015		CMPB	#15, (R3)+		;; CHECK FOR RETURN
4729	013534	001356			BNE	2\$		;; LOOP IF NOT RETURN
4730	013536	105063	177777		CLRB	-1(R3)		;; CLEAR RETURN (THE 15)
4731	013542	104401	001172		TYPE	\$LF		;; TYPE A LINE FEED
4732	013546	012603			MOV	(SP)+, R3		;; RESTORE R3
4733	013550	011646			MOV	(SP), -(SP)		;; ADJUST THE STACK AND PUT ADDRESS OF THE
4734	013552	016666	000004	000002	MOV	4(SP), 2(SP)		;; FIRST ASCII CHARACTER ON IT
4735	013560	012766	013572	000004	MOV	#TTYIN, 4(SP)		
4736	013566	000002			RTI			;; RETURN
4737	013570	000			9\$: .BYTE	0		;; STORAGE FOR ASCII CHAR. TO TYPE
4738	013571	000			.BYTE	0		;; TERMINATOR
4739	013572	000010			TTYIN: .BLKB	8		;; RESERVE 8 BYTES FOR TTY INPUT
4740	013602	052536	005015	000	SCNTLU: .ASCIZ	/U/<15><12>		;; CONTROL "U"
4741	013607	136	006507	000012	SCNTLG: .ASCIZ	/G/<15><12>		;; CONTROL "G"
4742	013614	005015	053523	020122	MSWR: .ASCIZ	<15><12>/SWR = /		
4743	013622	020075	000					
4744	013625	040	047040	053505	SMNEW: .ASCIZ	/ NEW = /		
4745	013632	036440	000040					

.SBTTL APT COMMUNICATIONS ROUTINE

```

4746
4747
4748
4749 013636 112767 000001 000236 $ATY1: MOVB #1,$FFLG ;;TO REPORT FATAL ERROR
4750 013644 112767 000001 000226 $ATY3: MOVB #1,$MFLG ;;TO TYPE A MESSAGE
4751 013652 000403 $ATYC BR $ATYC
4752 013654 112767 000001 000220 $ATY4: MOVB #1,$FFLG ;;TO ONLY REPORT FATAL ERROR
4753 013662 $ATYC:
4754 013662 010046 MOV RO,-(SP) ;;PUSH RO ON STACK
4755 013664 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
4756 013666 105767 000206 TSTB $MFLG ;;SHOULD TYPE A MESSAGE?
4757 013672 001450 BEQ $S IF NOT: BR
4758 013674 122767 000001 165312 CMPB #APTENV,$ENV ;;OPERATING UNDER APT?
4759 013702 001031 BNE $S IF NOT: BR
4760 013704 132767 000100 165303 BITB #APTSPool,$ENVM ;;SHOULD SPOOL MESSAGES?
4761 013712 001425 BEQ $S IF NOT: BR
4762 013714 017600 000004 MOV #4(SP),RO ;;GET MESSAGE ADDR.
4763 013720 062766 000002 000004 ADD #2,4(SP) ;;BUMP RETURN ADDR.
4764 013726 005767 165242 1$: TST $MSGTYPE ;;SEE IF DONE W/ LAST XMISSION?
4765 013732 001375 BNE $S IF NOT: WAIT
4766 013734 010067 165250 MOV RO,$MSGAD ;;PUT ADDR IN MAILBOX
4767 013740 105720 2$: TSTB (RO)+ ;;FIND END OF MESSAGE
4768 013742 001376 BNE $S
4769 013744 166700 165240 SUB $MSGAD,RO ;;SUB START OF MESSAGE
4770 013750 006200 ASR RO ;;GET MESSAGE LGTH IN WORDS
4771 013752 010067 165234 MOV RO,$MSGLG ;;PUT LENGTH IN MAILBOX
4772 013756 012767 000004 165210 MOV #4,$MSGTYPE ;;TELL APT TO TAKE MSG.
4773 013764 000413 BR $S
4774 013766 017667 000004 000016 3$: MOV #4(SP),4$ ;;PUT MSG ADDR IN JSR LINKAGE
4775 013774 062766 000002 000004 ADD #2,4(SP) ;;BUMP RETURN ADDRESS
4776 014002 016746 163770 MOV 177776,-(SP) ;;PUSH 177776 ON STACK
4777 014006 004767 176566 JSR PC,$TYPE ;;CALL TYPE MACRO
4778 014012 000000 4$: .WORD 0
4779 014014 5$:
4780 014014 105767 000062 10$: TSTB $FFLG ;;SHOULD REPORT FATAL ERROR?
4781 014020 001416 BEQ $S IF NOT: BR
4782 014022 005767 165166 TST $ENV ;;RUNNING UNDER APT?
4783 014026 001413 BEQ $S IF NOT: BR
4784 014030 005767 165140 11$: TST $MSGTYPE ;;FINISHED LAST MESSAGE?
4785 014034 001375 BNE $S IF NOT: WAIT
4786 014036 017667 000004 165132 MOV #4(SP),$FATAL ;;GET ERROR #
4787 014044 062766 000002 000004 ADD #2,4(SP) ;;BUMP RETURN ADDR.
4788 014052 005267 165116 INC $MSGTYPE ;;TELL APT TO TAKE ERROR
4789 014056 105067 000020 12$: CLRB $FFLG ;;CLEAR FATAL FLAG
4790 014062 105067 000013 CLRB $LFLG ;;CLEAR LOG FLAG
4791 014066 105067 000006 CLRB $MFLG ;;CLEAR MESSAGE FLAG
4792 014072 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
4793 014074 012600 MOV (SP)+,RO ;;POP STACK INTO RO
4794 014076 000207 RTS PC ;;RETURN
4795 014100 000 $MFLG: .BYTE 0 ;;MESSG. FLAG
4796 014101 000 $LFLG: .BYTE 0 ;;LOG FLAG
4797 014102 000 $FFLG: .BYTE 0 ;;FATAL FLAG
4798 014104 .EVEN
4799 000200 APTSIZE=200
4800 000001 APTENV=001
4801 000100 APTSPool=100

```

K09

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 115
CVDVAB.P11 15-DEC-77 08:58 APT COMMUNICATIONS ROUTINE

SEQ 0114

4802

000040

APTC SUP=040

.SBTTL ERROR HANDLER ROUTINE

```

*****
*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
*AND GO TO MYTYPE ON ERROR
*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
*SW15=1      HALT ON ERROR
*SW13=1      INHIBIT ERROR TYPEOUTS
*SW10=1      BELL ON ERROR
*SW09=1      LOOP ON ERROR
*CALL
*          ERROR      N          ;;ERROR=EMT AND N=ERROR ITEM NUMBER

```

\$ERROR:

```

7$:      CKSWR          ;; TEST FOR CHANGE IN SOFT-SWR
        INCB           SET THE ERROR FLAG
        BEQ            7$      DON'T LET THE FLAG GO TO ZERO
        MOV            $STNM,$DISPLAY  DISPLAY TEST NUMBER AND ERROR FLAG
        BIT            $BIT10,$SWR    BELL ON ERROR?
        BEQ            1$      NO - SKIP
        TYPE          $BELL          RING BELL
        INC           $ERTL          COUNT THE NUMBER OF ERRORS
        MOV           (SP),$ERRPC    GET ADDRESS OF ERROR INSTRUCTION
        SUB           $2,$ERRPC
        MOV           $2,$ERRPC
        MOV           $2,$ERRPC,$ITEMB  STRIP AND SAVE THE ERROR ITEM CODE
        BIT           $BIT13,$SWR    SKIP TYPEOUT IF SET
        BNE           20$          SKIP TYPEOUTS
        JSR           PC,MYTYPE      GO TO USER ERROR ROUTINE
        TYPE          $CRLF

20$:      CMPB          $APTENV,$ENV  ;; RUNNING IN APT MODE
        BNE           2$          NO SKIP APT ERROR REPORT
        MOV           $ITEMB,$21$    SET ITEM NUMBER AS ERROR NUMBER
        JSR           PC,$ATY4      REPORT FATAL ERROR TO APT

21$:      .BYTE        0
        .BYTE        0

22$:      BR           22$          ;; APT ERROR LOOP
2$:      TST           $SWR          HALT ON ERROR
        BPL           3$          SKIP IF CONTINUE
        HALT          HALT ON ERROR!
        CKSWR         TEST FOR CHANGE IN SOFT-SWR
        BIT           $BIT09,$SWR   LOOP ON ERROR SWITCH SET?
        BEQ           4$          BR IF NO
        MOV           $LPERR,(SP)   FUDGE RETURN FOR LOOPING
        TST           $ESCAPE      CHECK FOR AN ESCAPE ADDRESS
        BEQ           5$          BR IF NONE
        MOV           $ESCAPE,(SP)  FUDGE RETURN ADDRESS FOR ESCAPE

5$:      CMP           $SENDAD,$#42  ;; ACT-11 AUTO-ACCEPT?
        BNE           6$          BRANCH IF NO
        HALT          YES

6$:      RTI                ;; RETURN

```

```

4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817 014104
4818 014104 104407
4819 014106 105267 164771
4820 014112 001775
4821 014114 016777 164762 165020
4822 014122 032777 002000 165010
4823 014130 001402
4824 014132 104401 001164
4825 014136 005267 164750
4826 014142 011667 164750
4827 014146 162767 000002 164742
4828 014154 117767 164736 164732
4829 014162 032777 020000 164750
4830 014170 001004
4831 014172 004767 175554
4832 014176 104401 001171
4833 014202
4834 014202 122767 000001 165004
4835 014210 001007
4836 014212 116767 164676 000004
4837 014220 004767 177430
4838 014224 000
4839 014225 000
4840 014226 000777
4841 014230 005777 164704
4842 014234 100002
4843 014236 000000
4844 014240 104407
4845 014242 032777 001000 164670
4846 014250 001402
4847 014252 016716 164632
4848 014256 005767 164700
4849 014262 001402
4850 014264 016716 164672
4851 014270
4852 014270 022737 012362 000042
4853 014276 001001
4854 014300 000000
4855 014302
4856 014302 000002

```


.SBTTL SCOPE HANDLER ROUTINE

```

*****
*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
*AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
*SW14=1      LOOP ON TEST
*SW11=1      INHIBIT ITERATIONS
*SW09=1      LOOP ON ERROR
*SW08=1      LOOP ON TEST IN SWR<7:0>
*CALL
*          SCOPE          ;;SCOPE=IOT

```

```

$SCOPE:
1$:      CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
          BIT          $BIT14,$SWR      ;;LOOP ON PRESENT TEST?
          BNE          $OVER          ;;YES IF SW14=1
          ;;*****START OF CODE FOR THE XOR TESTER*****
          $XTSTR: BR    6$
          MOV          $ERRVEC, -(SP)    ;;IF RUNNING ON THE "XOR" TESTER CHANGE
          MOV          $SS,$ERRVEC      ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
          TST          $177060         ;;SAVE THE CONTENTS OF THE ERROR VECTOR
          MOV          (SP)+,$ERRVEC    ;;SET FOR TIMEOUT
          BR          $SVLAD           ;;TIME OUT ON XOR?
          ;;          ;;RESTORE THE ERROR VECTOR
          5$: CMP      (SP)+,(SP)+      ;;GO TO THE NEXT TEST
          MOV          (SP)+,$ERRVEC    ;;CLEAR THE STACK AFTER A TIME OUT
          BR          7$              ;;RESTORE THE ERROR VECTOR
          6$:;*****END OF CODE FOR THE XOR TESTER*****
          BIT          $BIT08,$SWR      ;;LOOP ON SPEC. TEST?
          BEQ          2$              ;;BR IF NO
          CMPB        $SWR,$STNM       ;;ON THE RIGHT TEST? SWR<7:0>
          BEQ          $OVER          ;;BR IF YES
          2$: TSTB     $ERFLG          ;;HAS AN ERROR OCCURRED?
          BEQ          3$              ;;BR IF NO
          CMPB        $ERMAX,$ERFLG    ;;MAX. ERRORS FOR THIS TEST OCCURRED?
          BHI          3$              ;;BR IF NO
          BIT          $BIT09,$SWR      ;;LOOP ON ERROR?
          BEQ          4$              ;;BR IF NO
          7$: MOV      $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
          BR          $OVER
          4$: CLRB     $ERFLG          ;;ZERO THE ERROR FLAG
          CLR          $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
          BR          1$              ;;ESCAPE TO THE NEXT TEST
          3$: BIT      $BIT11,$SWR     ;;INHIBIT ITERATIONS?
          BNE          1$              ;;BR IF YES
          TST          $PASS          ;;IF FIRST PASS OF PROGRAM
          BEQ          1$              ;;INHIBIT ITERATIONS
          INC          $ICNT          ;;INCREMENT ITERATION COUNT
          CMP          $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
          BGE          $OVER          ;;BR IF MORE ITERATION REQUIRED
          1$: MOV      $1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
          MOV          $MXCNT,$TIMES  ;;SET NUMBER OF ITERATIONS TO DO
          $SVLAD: INCB  $STNM          ;;COUNT TEST NUMBERS
          MOVB        $STNM,$TESTN    ;;SET TEST NUMBER IN APT MAILBOX

```

```

4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871 014304
4872 014304 104407
4873 014306 032777 040000 164624
4874 014314 001114
4875
4876 014316 000416
4877
4878 014320 013746 000004
4879 014324 012737 014344 000004
4880 014332 005737 177060
4881 014336 012637 000004
4882 014342 000463
4883 014344 022626
4884 014346 012637 000004
4885 014352 000423
4886 014354
4887 014354 032777 000400 164556
4888 014362 001404
4889 014364 127767 164550 164510
4890 014372 001465
4891 014374 105767 164503
4892 014400 001421
4893 014402 126767 164507 164473
4894 014410 101015
4895 014412 032777 001000 164520
4896 014420 001404
4897 014422 016767 164462 164456
4898 014430 000446
4899 014432 105067 164445
4900 014436 005067 164516
4901 014442 000415
4902 014444 032777 004000 164466
4903 014452 001011
4904 014454 005767 164522
4905 014460 001406
4906 014462 005267 164416
4907 014466 026767 164466 164410
4908 014474 002024
4909 014476 012767 000001 164400
4910 014504 016767 000052 164446
4911 014512 105267 164364
4912 014516 116767 164360 164454

```

N09

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 118
CVDVAB.P11 15-DEC-77 08:58 SCOPE HANDLER ROUTINE

SEQ 0117

4913	014524	011667	164356		MOV	(SP), \$LPADR	:: SAVE SCOPE LOOP ADDRESS
4914	014530	011667	164354		MOV	(SP), \$LPERR	:: SAVE ERROR LOOP ADDRESS
4915	014534	005067	164422		CLR	\$ESCAPE	:: CLEAR THE ESCAPE FROM ERROR ADDRESS
4916	014540	112767	000001	164347	MOVB	#1, \$ERMAX	:: ONLY ALLOW ONE(1) ERROR ON NEXT TEST
4917	014546	016777	164330	164366	\$OVER: MOV	\$TSTNM, @DISPLAY	:: DISPLAY TEST NUMBER
4918	014554	016716	164326		MOV	\$LPADR, (SP)	:: FUDGE RETURN ADDRESS
4919	014560	000002			RTI		:: FIXES PS
4920	014562	003720			\$MXCNT: 2000.		:: MAX. NUMBER OF ITERATIONS

.SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

```

*****
THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
REPLACED WITH SPACES.

```

```

*CALL:
*      MOV      NUM,-(SP)      ;;PUT THE BINARY NUMBER ON THE STACK
*      TYPDS      ;;GO TO THE ROUTINE

```

STYPDS:

```

MOV      R0,-(SP)      ;;PUSH R0 ON STACK
MOV      R1,-(SP)      ;;PUSH R1 ON STACK
MOV      R2,-(SP)      ;;PUSH R2 ON STACK
MOV      R3,-(SP)      ;;PUSH R3 ON STACK
MOV      R5,-(SP)      ;;PUSH R5 ON STACK
MOV      #20200,-(SP)  ;;SET BLANK SWITCH AND SIGN
MOV      20(SP),R5     ;;GET THE INPUT NUMBER
BPL      1$            ;;BR IF INPUT IS POS.
NEG      R5            ;;MAKE THE BINARY NUMBER POS.
MOVB     #'-,1(SP)     ;;MAKE THE ASCII NUMBER NEG.
CLR      R0            ;;ZERO THE CONSTANTS INDEX
MOV      #SDBLK,R3     ;;SETUP THE OUTPUT POINTER
MOVB     #'',(R3)+     ;;SET THE FIRST CHARACTER TO A BLANK
CLR      R2            ;;CLEAR THE BCD NUMBER
MOV      $DTBL(R0),R1  ;;GET THE CONSTANT
SUB      R1,R5         ;;FORM THIS BCD DIGIT
BLT      4$            ;;BR IF DONE
INC      R2            ;;INCREASE THE BCD DIGIT BY 1
BR       3$
4$:      ADD      R1,R5  ;;ADD BACK THE CONSTANT
TST      R2            ;;CHECK IF BCD DIGIT=0
BNE      5$            ;;FALL THROUGH IF 0
TSTB     (SP)          ;;STILL DOING LEADING 0'S?
BMI      7$            ;;BR IF YES
ASLB     (SP)          ;;MSD?
BCC      6$            ;;BR IF NO
MOVB     1(SP),-1(R3)  ;;YES--SET THE SIGN
BIS      #'0,R2        ;;MAKE THE BCD DIGIT ASCII
BIS      #' ,R2        ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
MOVB     R2,(R3)+     ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
TST      (R0)+        ;;JUST INCREMENTING
CMP      R0,#10       ;;CHECK THE TABLE INDEX
BLT      2$            ;;GO DO THE NEXT DIGIT
BGT      8$            ;;GO TO EXIT
MOV      R5,R2        ;;GET THE LSD
BR       6$            ;;GO CHANGE TO ASCII
TSTB     (SP)+        ;;WAS THE LSD THE FIRST NON-ZERO?
BPL      9$            ;;BR IF NO
MOVB     -1(SP),-2(R3) ;;YES--SET THE SIGN FOR TYPING
CLRB     (R3)          ;;SET THE TERMINATOR
MOV      (SP)+,R5      ;;POP STACK INTO R5
MOV      (SP)+,R3      ;;POP STACK INTO R3
MOV      (SP)+,R2      ;;POP STACK INTO R2

```

```

4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933 014564
4934 014564 010046
4935 014566 010146
4936 014570 010246
4937 014572 010346
4938 014574 010546
4939 014576 012746 020200
4940 014602 0166J5 000020
4941 014606 100004
4942 014610 005405
4943 014612 112766 000055 000001
4944 014620 005000 1$:
4945 014622 012703 015000
4946 014626 112723 000040
4947 014632 005002 2$:
4948 014634 016001 014770
4949 014640 160105 3$:
4950 014642 002402
4951 014644 005202
4952 014646 000774
4953 014650 060105 4$:
4954 014652 005702
4955 014654 001002
4956 014656 105716
4957 014660 100407
4958 014662 106316 5$:
4959 014664 103003
4960 014666 116663 000001 177777
4961 014674 052702 000060
4962 014700 052702 000040 6$:
4963 014704 110223 7$:
4964 014706 005720
4965 014710 020027 000010
4966 014714 002746
4967 014716 003002
4968 014720 010502
4969 014722 000764
4970 014724 105726 8$:
4971 014726 100003
4972 014730 116663 177777 177776 9$:
4973 014736 105013
4974 014740 012605
4975 014742 012603
4976 014744 012602

```

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 120
CVDVAB.P11 15-DEC-77 08:58 CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

SEQ 0119

4977	014746	012601		MOV	(SP)+,R1	::POP STACK INTO R1
4978	014750	012600		MOV	(SP)+,R0	::POP STACK INTO R0
4979	014752	104401	015000	TYPE	\$DBLK	::NOW TYPE THE NUMBER
4980	014756	016666	000002 000004	MOV	2(SP),4(SP)	::ADJUST THE STACK
4981	014764	012616		MOV	(SP)+,(SP)	
4982	014766	000002		RTI		::RETURN TO USER
4983	014770	023420		\$DTBL:	10000.	
4984	014772	001750			1000.	
4985	014774	000144			100.	
4986	014776	000012			10.	
4987	015000	000004		\$DBLK:	.BLKW 4	

```

4988 .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
4989
4990 *****
4991 *THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
4992 *OCTAL (ASCII) NUMBER AND TYPE IT.
4993 *$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
4994 *CALL:
4995 *      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
4996 *      TYPOS    ;;CALL FOR TYPEOUT
4997 *      .BYTE    N              ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
4998 *      .BYTE    M              ;;M=1 OR 0
4999 *                               ;;1=TYPE LEADING ZEROS
5000 *                               ;;0=SUPPRESS LEADING ZEROS
5001
5002 *$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
5003 *$TYPOS OR $TYPOC
5004 *CALL:
5005 *      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
5006 *      TYPON    ;;CALL FOR TYPEOUT
5007
5008 *$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
5009 *CALL:
5010 *      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
5011 *      TYPOC    ;;CALL FOR TYPEOUT
5012
5013 015010 017646 000000 000211 $TYPOS: MOV      2(SP),-(SP)      ;;PICKUP THE MODE
5014 015014 116667 000001      MOV      1(SP),%0FILL      ;;LOAD ZERO FILL SWITCH
5015 015022 112667 000207      MOV      (SP)+,%MODE+1      ;;NUMBER OF DIGITS TO TYPE
5016 015026 062716 000002      ADD      #2,(SP)      ;;ADJUST RETURN ADDRESS
5017 015032 000406      BR      $TYPON
5018 015034 112767 000001 000171 $TYPOC: MOV      #1,%0FILL      ;;SET THE ZERO FILL SWITCH
5019 015042 112767 000006 000165      MOV      #6,%MODE+1      ;;SET FOR SIX(6) DIGITS
5020 015050 112767 000005 000154 $TYPON: MOV      #5,%CNT      ;;SET THE ITERATION COUNT
5021 015056 010346      MOV      R3,-(SP)      ;;SAVE R3
5022 015060 010446      MOV      R4,-(SP)      ;;SAVE R4
5023 015062 010546      MOV      R5,-(SP)      ;;SAVE R5
5024 015064 116704 000145      MOV      %MODE+1,R4      ;;GET THE NUMBER OF DIGITS TO TYPE
5025 015070 005404      NEG      R4
5026 015072 062704 000006      ADD      #6,R4      ;;SUBTRACT IT FOR MAX. ALLOWED
5027 015076 110467 000132      MOV      R4,%MODE      ;;SAVE IT FOR USE
5028 015102 116704 000125      MOV      %0FILL,R4      ;;GET THE ZERO FILL SWITCH
5029 015106 016605 000012      MOV      12(SP),R5      ;;PICKUP THE INPUT NUMBER
5030 015112 005003      CLR      R3      ;;CLEAR THE OUTPUT WORD
5031 015114 006105 1$:      ROL      R5      ;;ROTATE MSB INTO "C"
5032 015116 000404      BR      3$      ;;GO DO MSB
5033 015120 006105 2$:      ROL      R5      ;;FORM THIS DIGIT
5034 015122 006105      ROL      R5
5035 015124 006105      ROL      R5
5036 015126 010503      MOV      R5,R3
5037 015130 006103 3$:      ROL      R3      ;;GET LSB OF THIS DIGIT
5038 015132 105367 000076      DECB      %MODE      ;;TYPE THIS DIGIT?
5039 015136 100016      BPL      7$      ;;BR IF NO
5040 015140 042703 177770      BIC      #177770,R3      ;;GET RID OF JUNK
5041 015144 001002      BNE      4$      ;;TEST FOR 0
5042 015146 005704      TST      R4      ;;SUPPRESS THIS 0?
5043 015150 001403      BEQ      5$      ;;BR IF YES

```

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 122
 CVDVAB.P11 15-DEC-77 08:58 BINARY TO OCTAL (ASCII) AND TYPE

SEQ 0121

5044	015152	005204		4\$:	INC	R4	;; DON'T SUPPRESS ANYMORE 0'S
5045	015154	052703	000060		BIS	#'0,R3	;; MAKE THIS DIGIT ASCII
5046	015160	052703	000040	5\$:	BIS	#' ,R3	;; MAKE ASCII IF NOT ALREADY
5047	015164	110367	000040		MOVB	R3,8\$	;; SAVE FOR TYPING
5048	015170	104401	015230		TYPE	8\$	;; GO TYPE THIS DIGIT
5049	015174	105367	000032	7\$:	DECB	\$OCNT	;; COUNT BY 1
5050	015200	003347			BGT	2\$	;; BR IF MORE TO DO
5051	015202	002402			BLT	6\$	;; BR IF DONE
5052	015204	005204			INC	R4	;; INSURE LAST DIGIT ISN'T A BLANK
5053	015206	000744			BR	2\$	;; GO DO THE LAST DIGIT
5054	015210	012605		6\$:	MOV	(SP)+,R5	;; RESTORE R5
5055	015212	012604			MOV	(SP)+,R4	;; RESTORE R4
5056	015214	012603			MOV	(SP)+,R3	;; RESTORE R3
5057	015216	016666	000002 000004		MOV	2(SP),4(SP)	;; SET THE STACK FOR RETURNING
5058	015224	012616			MOV	(SP)+,(SP)	
5059	015226	000002			RTI		;; RETURN
5060	015230	000		8\$:	.BYTE	0	;; STORAGE FOR ASCII DIGIT
5061	015231	000			.BYTE	0	;; TERMINATOR FOR TYPE ROUTINE
5062	015232	000		\$OCNT:	.BYTE	0	;; OCTAL DIGIT COUNTER
5063	015233	000		\$OFILL:	.BYTE	0	;; ZERO FILL SWITCH
5064	015234	000000		\$OMODE:	.WORD	0	;; NUMBER OF DIGITS TO TYPE

```

5065 .SBTTL TRAP DECODER
5066
5067 ;*****
5068 ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
5069 ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
5070 ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
5071 ;*GO TO THAT ROUTINE.
5072
5073 $TRAP: MOV RO, -(SP) ;:SAVE RO
5074 MOV 2(SP), RO ;:GET TRAP ADDRESS
5075 TST -(RO) ;:BACKUP BY 2
5076 MOVB (RO), 0 ;:GET RIGHT BYTE OF TRAP
5077 ASL RO ;:POSITION FOR INDEXING
5078 MOV $TRPAD(RO), RO ;:INDEX TO TABLE
5079 RTS RO ;:GO TO ROUTINE
5080
5081
5082 ;:THIS IS USE TO HANDLE THE "GETPRI" MACRO
5083
5084 $TRAP2: MOV (SP), -(SP) ;:MOVE THE PC DOWN
5085 MOV 4(SP), 2(SP) ;:MOVE THE PSW DOWN
5086 RTI ;:RESTORE THE PSW
5087
5088 .SBTTL TRAP TABLE
5089
5090 ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
5091 ;*BY THE "TRAP" INSTRUCTION.
5092
5093 ; ROUTINE
5094 ;-----
5095 $TRPAD: .WORD $TRAP2
5096 $TYPE ;:CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE
5097 $TYPOC ;:CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
5098 $TYPOS ;:CALL=TYPOS TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
5099 $TYPON ;:CALL=TYPON TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
5100 $TYPDS ;:CALL=TYPDS TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
5101
5102 $GTSWR ;:CALL=GTSWR TRAP+6(104406) GET SOFT-SWR SETTING
5103
5104 $CKSWR ;:CALL=CKSWR TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR
5105 $RDCHR ;:CALL=RDCHR TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
5106 $RDLIN ;:CALL=RDLIN TRAP+11(104411) TTY TYPEIN STRING ROUTINE
5107 .END

```

RBASE = 175610	1#	1001	1042					
ACDW1 = 000000	1001							
ACDW2 = 000000	1001							
ACPUOP = 000000	1001	1016						
ADDRES = 012266	4407*	4418*	4428	4435#				
ADDW0 = 000000	1001							
ADDW1 = 000000	1001							
ADDW10 = 000000	1001							
ADDW11 = 000000	1001							
ADDW12 = 000000	1001							
ADDW13 = 000000	1001							
ADDW14 = 000000	1001							
ADDW15 = 000000	1001							
ADDW2 = 000000	1001							
ADDW3 = 000000	1001							
ADDW4 = 000000	1001							
ADDW5 = 000000	1001							
ADDW6 = 000000	1001							
ADDW7 = 000000	1001							
ADDW8 = 000000	1001							
ADDW9 = 000000	1001							
ADEVCT = 000000	1001	1007						
ADEVN = 000001	1#	1001	1043					
AENV = 000000	1001	1012						
AENVN = 000000	1001	1013						
AFATAL = 000000	1001	1004						
AMADR1 = 000000	1001	1029						
AMADR2 = 000000	1001	1033						
AMADR3 = 000000	1001	1036						
AMADR4 = 000000	1001	1039						
AMAMS1 = 000000	1001	1023						
AMAMS2 = 000000	1001	1031						
AMAMS3 = 000000	1001	1034						
AMAMS4 = 000000	1001	1037						
AMSGAD = 000000	1001	1009						
AMSGLG = 000000	1001	1010						
AMSGTY = 000000	1001	1003						
AMTYP1 = 000000	1001	1024						
AMTYP2 = 000000	1001	1032						
AMTYP3 = 000000	1001	1035						
AMTYP4 = 000000	1001	1038						
APASS = 000000	1001	1006						
APRIOR = 000000	1001							
APTCU = 000040	4558	4802#						
APTENV = 000001	2673	2882	3159	3822	4551	4758	4800#	4834
APTSIZ = 000200	1112	4799#						
APTSPO = 000100	4553	4760	4801#					
ASWREG = 000000	1001	1014						
ATESTN = 000000	1001	1005						
AUNIT = 000000	1001	1008						
AUSWR = 071110	1#	1001	1015					
AVECT1 = 000300	1#	1001	1040					
AVECT2 = 000000	1001	1041						
BAOBRK = 011236	4081#							
BAUD = 007400	905#							
BAUDRA = 007144	3296#							

[illegible]

MAINDEC-22-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 127
 CVDV48.P11 15-DEC-77 08:58 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0125

EVENOD=	000040	899#												
FALSE =	000000	4139#	4146											
FLAG =	011554	4146#	4173*	4178	4201	4213#								
FRERR =	020000	836#												
FRFD =	040000	908#	1483	1550	2026	2117	2206	2386	2486					
GNS =	***** U	916	1135	4095	4101	4107	4338	4344	4351	4357	4363	5096	5097	5098
		5099	5100	5102	5104	5105	5106							
GTSWR =	104406	1130	5102#											
HOLD =	011022	3943#	3944#	3947	3960#									
HOLDSC	011555	4158#	4163#	4169	4214#									
HOWLON=	000000	4118#	4144											
HT =	000011	693#	4566	4607										
I =	001272	1069#	1189#	1210*	1213	1215	4247*	4250*	4252					
ILLMEM=	000004	795#	1184	1220										
INITFL	012264	1138#	4387	4390*	4434#									
INTFLA	011750	1196#	1201	3322*	3362	3366	3388*	3404	3455*	3497	3501	3545*	3571	3575
		4328#	4331#											
INTSRV	011742	1185	3334	3448	4318#									
IOTVEC=	000020	788#	1081#	1082*										
LF =	000012	694#	4601	4607										
LOOP =	001720	1139#	4112											
MAINT =	000004	868#	1324	1337	1340	1354	1357	1371	1376	2765	2824	2890	3172	3453
		3519	3620	3704	3727	3886	4040							
MASK =	000000	4225#	4239*	4255*	4257*	4262*								
MAX =	006356	2918	2922	2997#										
MYTYPE	011752	4334#	4831											
NEW =	007176	3195*	3197*	3213*	3215*	3224	3232	3234	3252	3254	3302#			
NUMBER	010760	3900	3925#	3950	4008									
NUMBR	011656	4241*	4242	4244*	4252	4265#								
OLD =	007202	3168*	3170*	3224	3232	3234	3252*	3254*	3303#					
ORERR =	040000	835#	3042	3081	3109									
PARITY=	000020	898#												
PBAUDS=	004000	858#												
PBAUD0=	010000	857#												
PBAUD1=	020000	856#												
PBAUD2=	040000	855#												
PBAUD3=	100000	854#												
PBR =	000200	901#	3150											
PERR =	010000	837#												
PIRQ =	177772	700#												
PIRQVE=	000240	794#												
PR0 =	000000	717#	3344	3393	3464	3551	3624	3730	3849					
PR1 =	000040	718#												
PR2 =	000100	719#												
PR3 =	000140	720#												
PR4 =	000200	721#												
PR5 =	000240	722#												
PR6 =	000300	723#												
PR7 =	000340	724#	1186	1396	2397	2497	2600	3335	3449	3860	3866			
PS =	177776	697#	698											
PSW =	177776	698#												
PWRVEC=	000024	789#	1087*	1088*	4487*	4488*	4497*	4503*	4515*	4516*				
RATES	007124	3186	3265	3270#										
RBUF =	001262	1065#	1155*	1156*	2859	2929	2984	3038	3081	3109	3124	3183	3493	3635
		3673	3740	3777	3978	4044	4070							
RCER =	001260	1064#	1153*	1494*	1497	1511*	1513	1526*	1529	1563	1576*	1579	1593*	1596


```
SETCLR= 000006
SPECIA 012164
STACK  = 001100
```

1610*	1615	1639	1652*	1655	1669*	1672	1686*	1691	1715	1728*	1731	1745*
1748	1762*	1767	1791	1804*	1807	1821*	1824	1838*	1843	1877	1942	1987
2048*	2051	2068*	2071	2088*	2091	2137*	2140	2157*	2160	2177*	2180	2227*
2230	2247*	2250	2267*	2270	2315*	2318	2335*	2338	2355*	2358	2407*	2416
2419	2435*	2437	2446	2461*	2463	2508*	2517	2520	2536*	2544	2559*	2567
2609*	2611	2619*	2627	2641*	2649	2780	2805	2839	2862	2904	2943	2947
2950	2970	2987	3203	3461*	3478	3485*	3522*	3548*	3556*	3559*	3561*	3592*
3594*	3666	3769	3893*	3919*	4012*	4056						
818#	1987	2904	2943	2950								
822#	1877	2781	2805	2840	2862	2947	2970	2987	3203	3479	3667	3770
4057												
823#	1791	1804	1807	1821	1824	1838	1843	3461	3485	3522	3893	4012
849#												
848#												
847#												
846#												
845#												
844#												
843#												
842#												
4720	5105#											
5106#												
3858	3970#											
4118#	4142											
4142*	4155	4211#										
827#	1563	1576	1579	1593	1596	1610	1615	2227	2247	2267	2508	2536
2559	3556	3561	3594									
784#												
3982*	3983*	3987	3995	4020#								
815#	2230	2250	2270									
3279#												
3280#												
3281#												
3282#												
3283#												
3286#												
3284#												
3285#												
3294#												
3287#												
3288#												
3289#												
3290#												
3291#												
1071#	1167											
3292#												
3293#												
3926#	3995*											
819#	2318	2338	2358									
826#	1639	1652	1655	1669	1672	1686	1691	2315	2335	2355	2609	2619
2641												
807#	2695	2738	2779	2838	2907	2916	3477	3652	3665	3756	3768	4055
4163												

[illegible]

[illegible]

[illegible]

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 132
CVDVAB.P11 15-DEC-77 08:58 CROSS REFERENCE TABLE -- USER SYMBOLS

SF\$BLA= 000170
SF\$CAS= 000150
SF\$DEC= 000220

3118#	3119#	3153#	3154#	3162#	3163#	3168#	3169#	3170#	3171#	3172#	3173#	3175#
3176#	3178#	3182#	3183#	3184#	3186#	3187#	3189#	3190#	3192#	3193#	3195#	3196#
3197#	3198#	3207#	3208#	3213#	3214#	3215#	3216#	3252#	3253#	3254#	3255#	3260#
3264#	3265#	3266#	3322#	3323#	3326#	3327#	3329#	3330#	3332#	3333#	3334#	3335#
3336#	3337#	3338#	3339#	3341#	3342#	3351#	3352#	3355#	3385#	3386#	3388#	3389#
3391#	3392#	3399#	3418#	3419#	3421#	3422#	3423#	3446#	3447#	3448#	3449#	3450#
3451#	3453#	3454#	3455#	3456#	3458#	3459#	3461#	3462#	3471#	3472#	3476#	3485#
3486#	3488#	3493#	3494#	3519#	3520#	3522#	3523#	3533#	3534#	3539#	3540#	3545#
3546#	3548#	3549#	3556#	3557#	3559#	3560#	3561#	3562#	3564#	3592#	3593#	3594#
3595#	3598#	3599#	3602#	3603#	3605#	3606#	3607#	3620#	3621#	3630#	3635#	3636#
3639#	3640#	3642#	3646#	3651#	3661#	3662#	3664#	3673#	3674#	3681#	3683#	3684#
3685#	3694#	3695#	3704#	3705#	3721#	3722#	3727#	3728#	3736#	3740#	3741#	3743#
3744#	3746#	3750#	3755#	3765#	3766#	3767#	3777#	3778#	3781#	3783#	3784#	3785#
3794#	3795#	3825#	3826#	3831#	3855#	3856#	3858#	3859#	3860#	3861#	3864#	3865#
3866#	3867#	3870#	3871#	3873#	3874#	3876#	3877#	3880#	3881#	3896#	3887#	3890#
3891#	3893#	3894#	3917#	3918#	3919#	3920#	3940#	3941#	3943#	3945#	3947#	3948#
3954#	3955#	3975#	3976#	3978#	3980#	3982#	3984#	3995#	3996#	3997#	3998#	4002#
4003#	4012#	4013#	4040#	4041#	4044#	4045#	4049#	4050#	4052#	4053#	4054#	4142#
4143#	4144#	4145#	4146#	4147#	4158#	4159#	4163#	4164#	4173#	4174#	4186#	4191#
4192#	4204#	4205#	4208#	4210#	4239#	4240#	4241#	4245#	4247#	4248#	4250#	4254#
4255#	4256#	4257#	4258#	4262#	4263#	4285#	4286#	4287#	4288#	4290#	4294#	4295#
4296#	4298#	4302#	4328#	4329#	4390#	4391#	4395#	4399#	4400#	4403#	4404#	4405#
4406#	4407#	4408#	4409#	4410#	4414#	4415#	4416#	4417#	4418#	4419#	4420#	4421#
4428#	4429#	4430#	4431#									
1083#	4817#											
971#	4353#	4826#	4827#	4828#	4857#							
1060#												
968#	4109#	4111#	4825#	4857#								
992#	1091#	4848#	4850#	4857#	4915#							
1011#												
953#	1044#											
1004#	4346#	4347#	4786#									
4749#	4752#	4780#	4789#	4797#								
989#	4576#	4607#										
988#	4607#											
1#	1203#	1250#	1266#	1283#	1302#	1326#	1342#	1359#	1378#			

\$FSG00= 000400

1458	1201	1248	1264	1281	1300	1324	1340	1357	1376	1406	1422	1439
1691	1483	1497	1513	1529	1550	1563	1579	1596	1615	1639	1655	1672
1987	1715	1731	1748	1767	1791	1807	1824	1843	1877	1910	1942	1972
2295	2026	2051	2071	2091	2117	2140	2160	2180	2206	2230	2250	2270
2590	2318	2338	2358	2386	2419	2437	2446	2463	2486	2520	2544	2567
2970	2627	2649	2673	2704	2725	2745	2788	2805	2846	2862	2882	2904
3234	2942	2943	2947	2950	2970	2987	3042	3060	3081	3109	3124	3150
3575	3199	3200	3203	3224	3232	3362	3366	3404	3497	3501	3535	3571
4169	3688	3717	3788	3822	3908	3950	3987	3991	4008	4062	4070	4155

\$FSIF = 000110

1340	1201	1207	1248	1254	1264	1270	1281	1287	1300	1306	1324	1330
1464	1346	1357	1363	1376	1382	1406	1412	1422	1428	1439	1445	1458
1579	1483	1489	1497	1503	1513	1519	1529	1535	1550	1556	1563	1569
1697	1585	1596	1602	1615	1621	1639	1645	1655	1661	1672	1678	1691
1824	1715	1721	1731	1737	1748	1754	1767	1773	1791	1797	1807	1813
2002	1830	1843	1849	1877	1888	1910	1921	1942	1955	1972	1979	1987
2160	2026	2035	2051	2059	2071	2079	2091	2099	2117	2124	2140	2148
2302	2168	2180	2188	2206	2214	2230	2238	2250	2258	2270	2278	2295
2446	2318	2326	2338	2346	2358	2366	2386	2394	2419	2425	2437	2443
2597	2452	2463	2469	2486	2494	2520	2526	2544	2550	2567	2573	2590
2788	2627	2633	2649	2655	2673	2679	2704	2709	2725	2731	2745	2750
2922	2793	2805	2811	2846	2851	2862	2868	2882	2888	2904	2909	2914
3073	2934	2947	2950	2961	2963	2970	2977	2987	2993	3042	3053	3060
3217	3081	3091	3109	3121	3124	3131	3150	3156	3159	3165	3203	3209
3404	3224	3228	3232	3234	3238	3247	3249	3362	3366	3372	3379	3381
3590	3410	3497	3501	3508	3514	3516	3535	3542	3571	3575	3582	3593
3987	3688	3697	3717	3724	3732	3737	3822	3828	3908	3914	3950	3956
4169	3991	3999	4004	4008	4015	4062	4067	4070	4076	4155	4160	4165
	4175	4201	4206	4384	4387	4392	4401	4411	4422			
	3175	3257	3639	3699	3743	3800	4247	4259	4287	4295	4303	4306
	4152	4178	4193									

\$FSINC= 000210

\$FSL00= 000200

\$FSNAM= 000160

\$FSNO = 000403

1458	1201	1248	1264	1281	1300	1324	1340	1357	1376	1406	1422	1439
1691	1483	1497	1513	1529	1550	1563	1579	1596	1615	1639	1655	1672
1987	1715	1731	1748	1767	1791	1807	1824	1843	1877	1910	1942	1972
2295	2026	2051	2071	2091	2117	2140	2160	2180	2206	2230	2250	2270
2590	2318	2338	2358	2386	2419	2437	2446	2463	2486	2520	2544	2567
2970	2627	2649	2673	2704	2725	2745	2788	2805	2846	2862	2882	2904
3234	2987	3042	3060	3081	3109	3124	3150	3159	3200	3203	3224	3232
3908	3362	3366	3404	3497	3501	3535	3571	3575	3688	3717	3788	3822
	3950	3987	3991	4008	4155	4384	4387					

\$FSOP = 000320

1460	1203	1250	1266	1283	1302	1326	1342	1359	1378	1408	1424	1441
1693	1485	1499	1515	1531	1552	1565	1581	1598	1617	1641	1657	1674
1989	1717	1733	1750	1769	1793	1809	1826	1845	1879	1912	1944	1974
2297	2028	2053	2073	2093	2119	2142	2162	2182	2208	2232	2252	2272
2592	2320	2340	2360	2388	2421	2439	2448	2465	2488	2522	2546	2569
2972	2629	2651	2675	2727	2807	2864	2884	2906	2924	2945	2949	2952
3236	2989	3044	3062	3083	3111	3126	3152	3161	3202	3205	3226	3234
3910	3364	3368	3406	3499	3503	3537	3573	3577	3690	3719	3790	3824
	3952	3989	3993	4010	4072	4157	4171	4203	4386	4389		
	4118	4217	4225	4267	4274	4312	4335	4368	4373	4439		

\$FSRTN= 000300

\$FSEL= 000140

\$FSUNT= 000130

\$FSWHI= 000120

\$FSYES= 000402

1191	1215	2902	2916	3898	3900	4382	4424					
2942	2943	2965	3199	3200	3220	3234						
1201	1248	1264	1281	1300	1324	1340	1357	1376	1406	1422	1439	
1458	1483	1497	1513	1529	1550	1563	1579	1596	1615	1639	1655	1672

MAINDEC-ZZ-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 134
CVDVAB.P11 15-DEC-77 08:58 CROSS REFERENCE TABLE -- USER SYMBOLS

```
SGDADR      001120
SGDOAT      001124
SGET42      012352
SGTSHR      013132
SHD         = 000000
SHIBTS      001000
SICNT       001104
SIFLEV=     177777
```

SILLUP 012562
SINTAG 001135
SISKO = 000001

SISK1 = 000001

SITEMB	001114
SLF	001172
SLFLG	014101

1691#	1715	1731	1748	1767	1791	1807	1824	1843	1877	1910	1942	1972
1987	2026	2051	2071	2091	2117	2140	2160	2180	2206	2230	2250	2270
2295	2318	2338	2358	2386	2419	2437	2446	2463	2486	2520	2544	2567
2590	2627	2649	2673	2725	2805	2862	2882	2904	2922	2943	2947	2950
2970	2987	3042	3060	3081	3109	3124	3150	3159	3200	3203	3224	3232
3234	3362	3366	3404	3497	3501	3535	3571	3575	3688	3717	3788	3822
3908	3950	3987	3991	4008	4070	4155	4169	4201	4384	4387		
972#												
974#												
4469#												
4629#	5102											
673												
948#												
965#	4906#	4907	4909#	4920								
1#	1201#	1207#	1248#	1254#	1264#	1270#	1281#	1287#	1300#	1306#	1324#	1330#
1340#	1346#	1357#	1363#	1376#	1382#	1406#	1412#	1422#	1428#	1439#	1445#	1458#
1464#	1483#	1489#	1497#	1503#	1513#	1519#	1529#	1535#	1550#	1556#	1563#	1569#
1579#	1585#	1596#	1602#	1615#	1621#	1634#	1645#	1655#	1661#	1672#	1678#	1691#
1697#	1715#	1721#	1731#	1737#	1748#	1754#	1767#	1773#	1791#	1797#	1807#	1813#
1824#	1830#	1843#	1849#	1877#	1888#	1910#	1921#	1942#	1955#	1972#	1979#	1987#
2002#	2026#	2035#	2051#	2059#	2071#	2079#	2091#	2099#	2117#	2124#	2140#	2148#
2160#	2168#	2180#	2188#	2206#	2214#	2230#	2238#	2250#	2258#	2270#	2278#	2295#
2302#	2318#	2326#	2338#	2346#	2358#	2366#	2386#	2394#	2419#	2425#	2437#	2443#
2446#	2452#	2463#	2469#	2486#	2494#	2520#	2526#	2544#	2550#	2567#	2573#	2590#
2597#	2627#	2633#	2649#	2655#	2673#	2679#	2704#	2709#	2725#	2731#	2745#	2750#
2788#	2793#	2805#	2811#	2846#	2851#	2868#	2882#	2888#	2888#	2904#	2914#	2922#
2934#	2947#	2950#	2961#	2987#	2993#	3042#	3042#	3060#	3042#	3053#	3060#	3073#
3081#	3091#	3109#	3121#	3124#	3131#	3150#	3156#	3159#	3165#	3203#	3217#	3224#
3232#	3247#	3249#	3262#	3266#	3279#	3381#	3404#	3410#	3497#	3501#	3514#	3516#
3535#	3542#	3571#	3575#	3588#	3590#	3688#	3697#	3717#	3724#	3788#	3797#	3822#
3828#	3908#	3914#	3950#	3956#	3987#	3991#	3999#	4004#	4008#	4015#	4062#	4067#
4070#	4076#	4155#	4165#	4169#	4175#	4201#	4206#	4384#	4387#	4401#	4422#	
4487	4503	4522#										
979#	4657	4746										
1201#	1207	1248#	1254	1264#	1270	1281#	1287	1300#	1306	1324#	1330	1340#
1346	1357#	1363	1376#	1382	1406#	1412	1422#	1428	1439#	1445	1458#	1464

SLOCTA= 177777

1#	1191	1192	1202	1203	1207	1208	1216	1217	1249	1250	1254	1255
1265	1266	1270	1271	1282	1283	1287	1288	1301	1302	1306	1307	1325
1326	1330	1331	1341	1342	1346	1347	1358	1359	1363	1364	1377	1378
1382	1383	1407	1408	1412	1413	1423	1424	1428	1429	1440	1441	1445
1446	1459	1460	1464	1465	1484	1485	1489	1490	1498	1499	1503	1504
1514	1515	1519	1520	1530	1531	1535	1536	1551	1552	1556	1557	1564
1565	1569	1570	1580	1581	1585	1586	1597	1598	1602	1603	1616	1617
1621	1622	1640	1641	1645	1646	1656	1657	1661	1662	1673	1674	1678
1679	1692	1693	1697	1698	1716	1717	1721	1722	1732	1733	1737	1738
1749	1750	1754	1755	1768	1769	1773	1774	1792	1793	1797	1798	1808
1809	1813	1814	1825	1826	1830	1831	1844	1845	1849	1850	1878	1879
1888	1889	1911	1912	1921	1922	1943	1944	1955	1956	1973	1974	1979
1980	1988	1989	2002	2003	2027	2028	2035	2036	2052	2053	2059	2060
2072	2073	2079	2080	2092	2093	2099	2100	2118	2119	2124	2125	2141
2142	2148	2149	2161	2162	2168	2169	2181	2182	2188	2189	2207	2208
2214	2215	2231	2232	2238	2239	2251	2252	2258	2259	2271	2272	2278
2279	2296	2297	2302	2303	2319	2320	2326	2327	2339	2340	2346	2347
2359	2360	2366	2367	2387	2388	2394	2395	2420	2421	2425	2426	2438
2439	2443	2444	2447	2448	2452	2453	2464	2465	2469	2470	2487	2488
2494	2495	2521	2522	2526	2527	2545	2546	2550	2551	2568	2569	2573
2574	2591	2592	2597	2598	2628	2629	2633	2634	2650	2651	2655	2656
2674	2675	2679	2680	2704	2705	2709	2710	2726	2727	2731	2732	2745
2746	2750	2751	2788	2789	2793	2794	2806	2807	2811	2812	2846	2847
2851	2852	2863	2864	2868	2869	2883	2884	2888	2889	2902	2903	2905
2906	2909	2910	2911	2914	2915	2917	2918	2919	2920	2921	2923	2924
2934	2935	2942	2943	2944	2945	2948	2949	2951	2952	2961	2962	2963
2964	2965	2966	2967	2971	2972	2977	2978	2988	2989	2993	2994	3043
3044	3053	3054	3061	3062	3073	3074	3082	3083	3091	3092	3110	3111
3121	3122	3125	3126	3131	3132	3151	3152	3156	3157	3160	3161	3165
3166	3176	3177	3178	3179	3180	3181	3182	3199	3200	3201	3202	3204
3205	3209	3210	3211	3217	3218	3220	3221	3222	3225	3226	3228	3229
3230	3233	3234	3235	3236	3238	3239	3240	3247	3248	3249	3258	3257
3258	3259	3263	3264	3267	3268	3272	3273	3274	3279	3280	3281	3282
3405	3406	3410	3411	3498	3499	3502	3503	3508	3509	3510	3514	3515
3516	3517	3536	3537	3542	3543	3572	3573	3576	3577	3582	3583	3584
3588	3589	3590	3591	3640	3641	3642	3643	3644	3645	3646	3689	3690
3697	3698	3699	3700	3701	3718	3719	3724	3725	3744	3745	3746	3747
3748	3749	3750	3789	3790	3797	3798	3800	3801	3802	3823	3824	3828
3829	3898	3899	3901	3902	3903	3904	3905	3909	3910	3914	3915	3951
3952	3956	3957	3988	3989	3992	3993	3999	4000	4004	4005	4009	4010
4015	4016	4062	4063	4067	4068	4071	4072	4076	4077	4117	4152	4153
4156	4157	4160	4161	4162	4165	4166	4170	4171	4175	4176	4179	4180
4181	4182	4193	4194	4195	4202	4203	4204	4205	4206	4207	4209	4210
4217	4218	4219	4220	4224	4225	4249	4250	4251	4252	4253	4254	4259
4260	4261	4264	4265	4267	4268	4269	4273	4288	4289	4290	4291	4292
4293	4294	4296	4297	4298	4299	4300	4301	4302	4303	4304	4305	4306
4307	4308	4312	4313	4314	4334	4368	4369	4370	4372	4382	4383	4385
4386	4388	4389	4392	4393	4394	4401	4402	4411	4412	4413	4422	4423
4425	4426	4432	4433	4439	4440	4441						
966#	1093#	4897#	4913#	4918	4920							
967#	1094#	1193#	1245#	1259#	1275#	1292#	1321#	1335#	1351#	1368#	1403#	1417#
1433#	1450#	1492#	1508#	1524#	1560#	1574#	1590#	1607#	1636#	1650#	1666#	1682#
1712#	1726#	1742#	1759#	1788#	1802#	1818#	1835#	1875#	1907#	1940#	1984#	2044#
2064#	2084#	2133#	2153#	2173#	2223#	2243#	2263#	2311#	2331#	2351#	2404#	2431#
2457#	2504#	2531#	2555#	2615#	2638#	2681#	2713#	2768#	2798#	2826#	3011#	3058#
3077#	3095#	3338#	3385#	3453#	3533#	4847	4897	4914#	4920			

 SLPADR 001106
 SLPERR 001110

\$LSTCN= 177777

1281	1283	1287	1300	1302	1306	1324	1326	1330	1340	1342	1346	1357
1359	1363	1376	1378	1382	1406	1408	1412	1422	1424	1428	1439	1441
1445	1458	1460	1464	1483	1485	1489	1497	1499	1503	1513	1515	1519
1529	1531	1535	1550	1552	1556	1563	1565	1569	1579	1581	1585	1596
1598	1602	1615	1617	1621	1639	1641	1645	1655	1657	1661	1672	1674
1678	1691	1693	1697	1715	1717	1721	1731	1733	1737	1748	1750	1754
1767	1769	1773	1791	1793	1797	1807	1809	1813	1824	1826	1830	1843
1845	1849	1877	1879	1888	1910	1912	1921	1942	1944	1955	1972	1974
1979	1987	1989	2002	2026	2028	2035	2051	2053	2059	2071	2073	2079
2091	2093	2099	2117	2119	2124	2140	2142	2148	2160	2162	2168	2180
2182	2188	2206	2208	2214	2230	2232	2238	2250	2252	2258	2270	2272
2278	2295	2297	2302	2318	2320	2326	2338	2340	2346	2358	2360	2366
2386	2388	2394	2419	2421	2425	2437	2439	2443	2446	2448	2452	2463
2465	2469	2486	2488	2494	2520	2522	2526	2544	2546	2550	2567	2569
2573	2590	2592	2597	2627	2629	2633	2649	2651	2655	2673	2675	2679
2704	2705	2709	2725	2727	2731	2745	2746	2750	2788	2789	2793	2805
2807	2811	2846	2847	2851	2862	2864	2868	2882	2884	2888	2902	2903
2904	2906	2910	2911	2914	2916	2921	2922	2924	2934	2942	2943	2945
2947	2949	2950	2952	2961	2963	2965	2966	2970	2972	2977	2987	2989
2993	3042	3044	3053	3060	3062	3073	3081	3083	3091	3109	3111	3121
3124	3126	3131	3150	3152	3156	3159	3161	3165	3175	3177	3178	3179
3182	3199	3200	3202	3203	3205	3210	3211	3217	3220	3221	3224	3226
3229	3230	3232	3236	3239	3240	3247	3249	3257	3258	3362	3364	3366
3368	3373	3374	3379	3381	3404	3406	3410	3497	3499	3501	3503	3509
3510	3514	3516	3535	3537	3542	3571	3573	3575	3577	3583	3584	3588
3590	3639	3641	3642	3643	3646	3688	3690	3697	3699	3700	3717	3719
3724	3743	3745	3746	3747	3750	3788	3790	3797	3800	3801	3822	3824
3828	3898	3899	3900	3905	3908	3910	3914	3950	3952	3956	3987	3989
3991	3993	3999	4004	4008	4010	4015	4062	4063	4067	4070	4072	4076
4118	4152	4153	4155	4157	4161	4162	4165	4169	4171	4175	4193	4194
4201	4203	4206	4217	4225	4247	4249	4250	4251	4254	4259	4260	4267
4274	4287	4289	4290	4291	4294	4295	4297	4298	4299	4302	4303	4304
4306	4307	4312	4335	4368	4373	4382	4383	4384	4386	4387	4389	4393
4394	4401	4412	4413	4422	4424	4439						
	1138	1139	1141	1142	1145	1146	1148	1149	1150	1151	1153	1154
1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167
1168	1180	1181	1183	1184	1185	1186	1187	1188	1189	1190	1193	1194
1196	1197	1201	1202	1203	1210	1211	1212	1213	1214	1215	1216	1217
1220	1221	1222	1223	1224	1225	1245	1246	1248	1249	1250	1259	1260
1261	1262	1264	1265	1266	1275	1276	1278	1279	1281	1282	1283	1292
1293	1295	1296	1300	1301	1302	1321	1322	1324	1325	1326	1335	1336
1337	1338	1340	1341	1342	1351	1352	1354	1355	1357	1358	1359	1368
1369	1371	1372	1376	1377	1378	1403	1404	1406	1407	1408	1417	1418
1419	1420	1422	1423	1424	1433	1434	1436	1437	1439	1440	1441	1450
1451	1453	1454	1458	1459	1460	1483	1484	1485	1486	1487	1492	1493
1494	1495	1497	1498	1499	1508	1509	1511	1512	1513	1514	1515	1524
1525	1526	1527	1529	1530	1531	1550	1551	1552	1553	1554	1560	1561
1563	1564	1565	1574	1575	1576	1577	1579	1580	1581	1590	1591	1593
1594	1596	1597	1598	1607	1608	1610	1611	1615	1616	1617	1636	1637
1639	1640	1641	1650	1651	1652	1653	1655	1656	1657	1666	1667	1669
1670	1672	1673	1674	1682	1683	1686	1687	1691	1692	1693	1712	1713
1715	1716	1717	1726	1727	1728	1729	1731	1732	1733	1742	1743	1745
1746	1748	1749	1750	1759	1760	1762	1763	1767	1768	1769	1788	1789
1791	1792	1793	1802	1803	1804	1805	1807	1808	1809	1818	1819	1821
1822	1824	1825	1826	1835	1836	1838	1839	1843	1844	1845	1875	1876

\$LSTIN= 000000

1877	1878	1879	1907	1908	1910	1911	1912	1940	1941	1942	1943	1944
1972	1973	1974	1976	1977	1984	1985	1987	1988	1989	2026	2027	2028
2032	2033	2044	2045	2048	2049	2051	2052	2053	2064	2065	2068	2069
2071	2072	2073	2084	2085	2088	2089	2091	2092	2093	2117	2118	2119
2121	2122	2133	2134	2137	2138	2140	2141	2142	2153	2154	2157	2158
2160	2161	2162	2173	2174	2177	2178	2180	2181	2182	2206	2207	2208
2211	2212	2233	2234	2237	2238	2230	2231	2232	2243	2244	2247	2248
2250	2251	2252	2253	2254	2256	2258	2259	2260	2261	2262	2263	2264
2299	2299	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310
2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350
2391	2392	2393	2394	2395	2396	2397	2398	2399	2400	2401	2402	2403
2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431
2457	2458	2459	2460	2461	2462	2463	2464	2465	2466	2467	2468	2469
2505	2506	2507	2508	2509	2510	2511	2512	2513	2514	2515	2516	2517
2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2541	2542	2543
2556	2557	2558	2559	2560	2561	2562	2563	2564	2565	2566	2567	2568
2592	2593	2594	2595	2596	2597	2598	2599	2600	2601	2602	2603	2604
2624	2625	2626	2627	2628	2629	2630	2631	2632	2633	2634	2635	2636
2647	2648	2649	2650	2651	2652	2653	2654	2655	2656	2657	2658	2659
2690	2691	2692	2693	2694	2695	2696	2697	2698	2699	2700	2701	2702
2720	2721	2722	2723	2724	2725	2726	2727	2728	2729	2730	2731	2732
2745	2746	2747	2748	2749	2750	2751	2752	2753	2754	2755	2756	2757
2783	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794	2795
2827	2828	2829	2830	2831	2832	2833	2834	2835	2836	2837	2838	2839
2859	2860	2861	2862	2863	2864	2865	2866	2867	2868	2869	2870	2871
2893	2894	2895	2896	2897	2898	2899	2900	2901	2902	2903	2904	2905
2913	2914	2915	2916	2917	2918	2919	2920	2921	2922	2923	2924	2925
2943	2944	2945	2946	2947	2948	2949	2950	2951	2952	2953	2954	2955
2970	2971	2972	2973	2974	2975	2976	2977	2978	2979	2980	2981	2982
3022	3023	3024	3025	3026	3027	3028	3029	3030	3031	3032	3033	3034
3042	3043	3044	3045	3046	3047	3048	3049	3050	3051	3052	3053	3054
3078	3079	3080	3081	3082	3083	3084	3085	3086	3087	3088	3089	3090
3109	3110	3111	3112	3113	3114	3115	3116	3117	3118	3119	3120	3121
3159	3160	3161	3162	3163	3164	3165	3166	3167	3168	3169	3170	3171
3177	3178	3179	3180	3181	3182	3183	3184	3185	3186	3187	3188	3189
3193	3194	3195	3196	3197	3198	3199	3200	3201	3202	3203	3204	3205
3209	3210	3211	3212	3213	3214	3215	3216	3217	3218	3219	3220	3221
3232	3233	3234	3235	3236	3237	3238	3239	3240	3241	3242	3243	3244
3260	3261	3262	3263	3264	3265	3266	3267	3268	3269	3270	3271	3272
3332	3333	3334	3335	3336	3337	3338	3339	3340	3341	3342	3343	3344
3356	3357	3358	3359	3360	3361	3362	3363	3364	3365	3366	3367	3368
3386	3387	3388	3389	3390	3391	3392	3393	3394	3395	3396	3397	3398
3418	3419	3420	3421	3422	3423	3424	3425	3426	3427	3428	3429	3430
3454	3455	3456	3457	3458	3459	3460	3461	3462	3463	3464	3465	3466
3480	3481	3482	3483	3484	3485	3486	3487	3488	3489	3490	3491	3492
3497	3498	3499	3500	3501	3502	3503	3504	3505	3506	3507	3508	3509
3534	3535	3536	3537	3538	3539	3540	3541	3542	3543	3544	3545	3546
3560	3561	3562	3563	3564	3565	3566	3567	3568	3569	3570	3571	3572
3577	3578	3579	3580	3581	3582	3583	3584	3585	3586	3587	3588	3589
3606	3607	3608	3609	3610	3611	3612	3613	3614	3615	3616	3617	3618
3642	3643	3644	3645	3646	3647	3648	3649	3650	3651	3652	3653	3654
3661	3662	3663	3664	3665	3666	3667	3668	3669	3670	3671	3672	3673
3682	3683	3684	3685	3686	3687	3688	3689	3690	3691	3692	3693	3694
3717	3718	3719	3720	3721	3722	3723	3724	3725	3726	3727	3728	3729
3743	3744	3745	3746	3747	3748	3749	3750	3751	3752	3753	3754	3755
3760	3761	3762	3763	3764	3765	3766	3767	3768	3769	3770	3771	3772

MAINDEC-22-CVDVA-B
CVDVAB.P11MACY11 30(1046)
15-DEC-77 08:5819-DEC-77 08:25 PAGE 138
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0136

SLSTST= 177777

3777	3778	3781	3782	3783	3784	3785	3788	3789	3790	3794	3795	3800
3801	3822	3823	3824	3825	3826	3831	3832	3833	3834	3855	3856	3858
3859	3860	3861	3864	3865	3866	3867	3870	3871	3873	3874	3876	3877
3880	3881	3886	3887	3890	3891	3893	3894	3900	3901	3902	3903	3904
3908	3909	3910	3917	3918	3919	3920	3940	3941	3943	3944	3945	3947
3948	3950	3951	3952	3954	3955	3975	3976	3978	3979	3980	3982	3983
3984	3987	3988	3989	3991	3992	3993	3995	3996	3997	3998	4002	4003
4004	4009	4010	4012	4013	4014	4041	4044	4045	4049	4070	4052	4053
4054	4055	4056	4057	4058	4059	4060	4061	4062	4063	4070	4071	4072
4142	4143	4144	4145	4146	4147	4155	4156	4157	4158	4159	4160	4161
4163	4164	4169	4170	4171	4173	4174	4178	4179	4180	4181	4182	4186
4187	4188	4189	4190	4191	4193	4193	4194	4201	4202	4203	4204	4205
4208	4209	4210	4218	4219	4220	4221	4239	4240	4241	4242	4243	4244
4247	4247	4248	4249	4250	4251	4252	4253	4254	4255	4256	4257	4258
4259	4260	4262	4263	4264	4265	4269	4270	4285	4286	4287	4288	4289
4290	4291	4292	4293	4294	4295	4296	4297	4298	4299	4300	4301	4302
4303	4304	4306	4307	4314	4315	4328	4329	4370	4371	4384	4385	4386
4387	4388	4389	4390	4391	4392	4393	4395	4396	4399	4400	4403	4404
4405	4406	4407	4408	4409	4410	4411	4412	4414	4415	4416	4417	4418
4419	4420	4421	4424	4425	4426	4428	4429	4430	4431	4432	4433	4441
4442												
1281	1191	1192	1201	1203	1207	1215	1248	1250	1254	1264	1266	1270
1282	1283	1287	1300	1302	1306	1324	1326	1330	1340	1342	1346	1357
1359	1363	1376	1378	1382	1406	1408	1412	1422	1424	1428	1439	1441
1445	1458	1460	1464	1483	1485	1489	1497	1499	1503	1513	1515	1519
1529	1531	1535	1550	1552	1556	1563	1565	1569	1579	1581	1585	1596
1598	1602	1615	1617	1621	1639	1641	1645	1655	1657	1661	1672	1674
1678	1691	1693	1697	1715	1717	1721	1731	1733	1737	1748	1750	1754
1767	1769	1773	1791	1793	1797	1807	1809	1813	1824	1826	1830	1843
1845	1849	1877	1879	1888	1910	1912	1921	1942	1944	1955	1972	1974
1979	1987	1989	2002	2026	2028	2035	2051	2053	2059	2071	2073	2079
2091	2093	2099	2117	2119	2124	2140	2142	2148	2160	2162	2168	2180
2182	2188	2206	2208	2214	2230	2232	2238	2250	2252	2258	2270	2272
2278	2295	2297	2302	2318	2320	2326	2338	2340	2346	2358	2360	2366
2386	2388	2394	2419	2421	2425	2437	2438	2443	2446	2448	2452	2463
2465	2469	2486	2488	2494	2500	2522	2526	2544	2546	2550	2567	2569
2573	2590	2592	2597	2627	2629	2633	2649	2651	2655	2673	2675	2679
2704	2705	2709	2725	2727	2731	2745	2746	2750	2788	2789	2793	2805
2807	2811	2846	2847	2851	2862	2864	2868	2882	2884	2888	2902	2903
2904	2906	2909	2910	2911	2914	2916	2922	2924	2934	2942	2943	2945
2947	2949	2950	2952	2961	2963	2965	2966	2970	2972	2977	2987	2989
2993	3042	3044	3053	3060	3062	3073	3081	3083	3091	3109	3111	3121
3124	3126	3131	3150	3152	3156	3159	3161	3165	3175	3177	3178	3179
3182	3199	3200	3202	3203	3205	3209	3210	3211	3217	3220	3221	3224
3226	3228	3229	3230	3232	3236	3238	3239	3240	3247	3249	3257	3258
3262	3264	3266	3268	3272	3273	3274	3279	3281	3294	3406	3410	3497
3299	3501	3503	3508	3509	3510	3514	3516	3535	3537	3542	3571	3573
3575	3577	3582	3583	3584	3588	3590	3639	3641	3642	3643	3646	3688
3690	3697	3699	3700	3717	3719	3724	3743	3745	3746	3747	3750	3788
3790	3797	3800	3801	3822	3824	3828	3898	3899	3900	3908	3910	3914
3950	3952	3956	3987	3989	3991	3993	3999	4004	4008	4010	4015	4062
4063	4067	4070	4072	4076	4118	4152	4153	4155	4157	4160	4161	4162
4165	4169	4171	4175	4178	4193	4194	4201	4203	4206	4217	4225	4247
4249	4250	4251	4254	4259	4260	4267	4274	4287	4289	4290	4291	4294
4295	4297	4298	4299	4302	4303	4304	4306	4307	4312	4335	4368	4373
4382	4383	4384	4386	4387	4389	4392	4393	4394	4401	4411	4412	4413

SNSKO = 000300

SNSK1 = 000130

SNK2 = 000110

SNSK3 = 000110

SNULL 001154

SNWTST= 000001

SOCNT 015232

SOMODE 015234

SOVER 014546

SPASS 001202

SPASTM 001006
SPALFD 012530

SPOWER 012570
BUBROD 012556

SPWRHO 012556
SPURON 013416

SPWRDN 012416
SPWRMG 012552

SPWFTG 012332

1569#	1579#	1585#	1596#	1602#	1615#	1621#	1639#	1645#	1655#	1661#	1672#
1678#	1691#	1697#	1715#	1721#	1731#	1737#	1748#	1754#	1767#	1773#	1791#
1807#	1813#	1824#	1830#	1843#	1849#	1877#	1888#	1910#	1921#	1942#	1955#
1979#	1987#	2002#	2026#	2035#	2051#	2059#	2071#	2079#	2091#	2099#	2117#
2140#	2148#	2160#	2168#	2180#	2188#	2206#	2214#	2230#	2238#	2250#	2258#
2278#	2295#	2302#	2318#	2326#	2338#	2346#	2358#	2366#	2386#	2394#	2419#
2437#	2443#	2446#	2452#	2463#	2469#	2486#	2494#	2520#	2526#	2544#	2550#
2573#	2590#	2597#	2627#	2633#	2649#	2655#	2673#	2679#	2704#	2709#	2725#
2745#	2750#	2788#	2793#	2805#	2811#	2846#	2851#	2862#	2868#	2882#	2888#
2904#	2909#	2914#	2916#	2922#	2934#	2942#	2947#	2950#	2961#	2963#	2965#
2977#	2987#	2993#	3042#	3053#	3060#	3073#	3081#	3091#	3109#	3121#	3124#
3150#	3156#	3159#	3165#	3175#	3199#	3203#	3209#	3217#	3220#	3224#	3228#
3238#	3247#	3249#	3257#	3362#	3366#	3372#	3379#	3381#	3404#	3410#	3497#
3508#	3514#	3516#	3535#	3542#	3571#	3575#	3582#	3588#	3590#	3639#	3688#
3699#	3717#	3724#	3743#	3788#	3797#	3800#	3822#	3828#	3898#	3900#	3908#
3950#	3956#	3987#	3991#	3999#	4004#	4008#	4015#	4062#	4067#	4070#	4076#
4152#	4155#	4160#	4165#	4169#	4175#	4178#	4193#	4201#	4206#	4217#	4225#
4259#	4267#	4274#	4287#	4295#	4303#	4306#	4312#	4335#	4368#	4373#	4382#
4387#	4392#	4401#	4411#	4422#	4424#	4439#					
1191#	1215#	1248#	1254#	1264#	1270#	1281#	1287#	1300#	1306#	1324#	1330#
1346#	1357#	1363#	1376#	1382#	1406#	1412#	1422#	1428#	1439#	1445#	1458#
1483#	1489#	1497#	1503#	1513#	1519#	1529#	1535#	1550#	1556#	1563#	1569#
1585#	1596#	1602#	1615#	1621#	1639#	1645#	1655#	1661#	1672#	1678#	1691#
1715#	1721#	1731#	1737#	1748#	1754#	1767#	1773#	1791#	1797#	1807#	1813#
1830#	1843#	1849#	1877#	1888#	1910#	1921#	1942#	1955#	1972#	1979#	1987#
2026#	2035#	2051#	2059#	2071#	2079#	2091#	2099#	2117#	2124#	2140#	2148#
2168#	2180#	2188#	2206#	2214#	2230#	2238#	2250#	2258#	2278#	2295#	2302#
2318#	2326#	2338#	2346#	2358#	2366#	2386#	2394#	2419#	2425#	2437#	2443#
2452#	2463#	2469#	2486#	2494#	2520#	2526#	2544#	2550#	2567#	2573#	2590#
2627#	2633#	2649#	2655#	2673#	2679#	2704#	2709#	2725#	2731#	2745#	2750#
2793#	2805#	2811#	2846#	2851#	2862#	2868#	2882#	2888#	2902#	2916#	2922#
2942#	2965#	2970#	2977#	2987#	2993#	3042#	3053#	3060#	3073#	3081#	3091#
3121#	3124#	3131#	3150#	3156#	3159#	3165#	3175#	3257#	3362#	3381#	3404#
3497#	3516#	3535#									

MAINDEC-22-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 141
 CVDVAB.P11 15-DEC-77 08:58 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0139

SPWRUP 012470	4497	4503#												
\$QUES 001170	994#	4607	4675	4724	4740	4857								
\$RDCHR 013344	4688#	5105												
\$RDOEC= ***** U	5107													
\$RDLIN 013464	4716#	5106												
\$RDOCT= ***** U	5107													
\$RDSZ = 000010	4709#													
\$RTNAD 012374	4478#													
\$R2A = ***** U	5107													
\$SAVLE= 177777	1#	2965#	2966#	3178#	3182#	3220#	3221#	3642#	3646#	3746#	3750#	4193#	4194#	
	4250#	4254#	4290#	4294#	4298#	4302#								
\$SAVRE= ***** U	5107													
\$SAVR6 012566	4496#	4504	4505#	4506#	4524#									
\$SCOPE 014304	1081	4871#												
\$SETUP= 000137	1072#	1060	1081	1083	1085	1087	1089	1090	1091	1093	1120	1123	4455	
	4612	4746	4818	4844	4852	4872								
\$SSKO = 000234	2965#	2966	3178#	3182	3220#	3221	3642#	3646	3746#	3750	4193#	4194	4250#	
	4254	4290#	4294	4298#	4302									
\$STUP = 177777	1072#													
\$SVLAD 014512	4882	4911#												
\$SVPC = 000204	925#	930												
\$SWR = 167400	1#	673	677	678	679	680	681	682	683	991	992	993	1090	
	1091	1093	1094	1177	1240	1316	1393	1480	1547	1632	1708	1784	1090	
	1900	1933	1967	2022	2113	2202	2291	2382	2482	2586	2668	2760	2819	
	2877	3007	3147	3318	3441	3616	3714	3816	4036	4092	4450	4456	4471	
	4477	4479	4521	4809	4810	4811	4812	4813	4822	4829	4841	4845	4857	
	4863	4864	4865	4866	4867	4873	4885	4887	4888	4891	4892	4893	4900	
	4901	4902	4914	4917	4920									
\$SWREG 001216	1014#	1114												
\$SWRMK= 000000	683	684	4867	4868	4889									
\$TAGLE= 177777	1#	1192#	1203#	1207#	1215#	1250#	1254#	1266#	1270#	1283#	1287#	1302#	1306#	
	1326#	1330#	1342#	1346#	1359#	1363#	1378#	1382#	1408#	1412#	1424#	1428#	1441#	
	1445#	1460#	1464#	1485#	1489#	1499#	1503#	1515#	1519#	1531#	1535#	1552#	1556#	
	1565#	1569#	1581#	1585#	1598#	1602#	1617#	1621#	1641#	1645#	1657#	1661#	1674#	
	1678#	1693#	1697#	1717#	1721#	1733#	1737#	1750#	1754#	1769#	1773#	1793#	1797#	
	1809#	1813#	1826#	1830#	1845#	1849#	1879#	1888#	1912#	1921#	1944#	1955#	1974#	
	1979#	1989#	2002#	2028#	2035#	2053#	2059#	2073#	2079#	2093#	2099#	2119#	2124#	
	2142#	2148#	2162#	2168#	2182#	2188#	2208#	2214#	2232#	2238#	2252#	2258#	2272#	
	2278#	2297#	2302#	2320#	2326#	2340#	2346#	2360#	2366#	2388#	2394#	2421#	2425#	
	2439#	2443#	2448#	2452#	2455#	2469#	2488#	2494#	2522#	2526#	2546#	2550#	2569#	
	2573#	2592#	2597#	2629#	2633#	2651#	2655#	2675#	2679#	2705#	2709#	2727#	2731#	
	2746#	2750#	2789#	2793#	2807#	2811#	2847#	2851#	2864#	2868#	2884#	2888#	2903#	
	2906#	2910#	2911#	2914#	2916#	2924#	2934#	2943#	2945#	2949#	2952#	2961#	2963#	
	2965#	2972#	2977#	2989#	2993#	3044#	3053#	3062#	3073#	3083#	3091#	3111#	3121#	
	3126#	3131#	3152#	3156#	3161#	3165#	3177#	3179#	3182#	3200#	3202#	3205#	3210#	
	3211#	3217#	3220#	3226#	3229#	3230#	3236#	3239#	3240#	3247#	3249#	3257#	3258#	
	3364#	3368#	3373#	3374#	3379#	3381#	3406#	3410#	3499#	3503#	3509#	3510#	3514#	
	3516#	3537#	3542#	3573#	3577#	3583#	3584#	3588#	3590#	3641#	3643#	3646#	3690#	
	3697#	3699#	3700#	3719#	3724#	3745#	3747#	3750#	3790#	3797#	3800#	3801#	3824#	
	3828#	3899#	3900#	3910#	3914#	3952#	3956#	3989#	3993#	3999#	4004#	4010#	4015#	
	4063#	4067#	4072#	4076#	4153#	4157#	4161#	4162#	4165#	4171#	4175#	4178#	4193#	
	4203#	4206#	4249#	4251#	4254#	4259#	4260#	4289#	4291#	4294#	4297#	4299#	4302#	
	4303#	4304#	4306#	4307#	4383#	4386#	4389#	4393#	4394#	4401#	4412#	4413#	4422#	
	4424#													
\$TAGNU= 000247	1#	1191	1192#	1202	1203#	1249	1250#	1265	1266#	1282	1283#	1301	1302#	
	1325	1326#	1341	1342#	1358	1359#	1377	1378#	1407	1408#	1423	1424#	1440	

1441#	1459	1460#	1484	1485#	1498	1499#	1514	1515#	1530	1531#	1551	1552#
1564	1565#	1580	1581#	1597	1598#	1616	1617#	1640	1641#	1656	1657#	1673
1674#	1692	1693#	1716	1717#	1732	1733#	1749	1750#	1768	1769#	1792	1793#
1808	1809#	1825	1826#	1844	1845#	1878	1879#	1911	1912#	1943	1944#	1973
1974#	1988	1989#	2027	2028#	2052	2053#	2072	2073#	2092	2093#	2118	2119#
2141	2142#	2161	2162#	2181	2182#	2207	2208#	2231	2232#	2251	2252#	2271
2272#	2296	2297#	2319	2320#	2339	2340#	2359	2360#	2387	2388#	2420	2421#
2438	2439#	2447	2448#	2464	2465#	2487	2488#	2521	2522#	2545	2546#	2568
2569#	2591	2592#	2628	2629#	2650	2651#	2674	2675#	2704	2705#	2726	2727#
2745	2746#	2788	2789#	2806	2807#	2846	2847#	2863	2864#	2883	2884#	2902
2903#	2905	2906#	2909	2911#	2917	2920	2921#	2923	2924#	2942	2943#	2944
2945#	2948	2949#	2951	2952#	2971	2972#	2988	2989#	3043	3044#	3061	3062#
3082	3083#	3110	3111#	3125	3126#	3151	3152#	3160	3161#	3176	3177#	3178
3181	3182#	3199	3200#	3201	3202#	3204	3205#	3209	3211#	3225	3226#	3228
3230#	3233	3235#	3236	3238	3240#	3263	3264#	3267	3268#	3272	3274#	3405
3406#	3498	3499#	3502	3503#	3508	3510	3536	3537#	3572	3573#	3576	3577#
3582	3584#	3640	3641#	3642	3645	3646#	3689	3690#	3718	3719#	3744	3745#
3746#	3749	3750#	3789	3790#	3823	3824#	3898	3899#	3901	3904#	3935	3909
3910#	3951	3952#	3988	3989#	3992	3993#	4009	4010#	4062	4063#	4071	4072#
4118#	4152	4153#	4156	4157#	4160	4162#	4170	4171#	4202	4203#	4225	4248
4249#	4250	4253#	4254	4274#	4288	4289#	4290	4293#	4294	4296#	4297	4298#
4301	4302#	4335	4373#	4382	4383#	4385	4386#	4388	4389#	4392	4394	4411
4413#	1139	1145#	1146	1148#	1149	1150#	1151	1153#	1154	1155#	1157	1158#
1138#	1161	1163#	1164	1166#	1167	1168#	1173	1181#	1183	1184#	1185	1186#
1160#	1188	1189#	1190	1193#	1194	1196#	1197	1207#	1210	1211#	1212	1214#
1187#	1216	1220#	1221	1223#	1224	1225#	1245	1246#	1254	1259#	1260	1261#
1215#	1270	1275#	1276	1278#	1279	1287#	1292	1293#	1295	1296#	1306	1321#
1262#	1330	1335#	1336	1337#	1338	1346#	1351	1352#	1354	1355#	1363	1368#
1322#	1371	1372#	1382	1403#	1404	1412#	1417	1418#	1419	1420#	1428	1433#
1369#	1436	1437#	1445	1450#	1451	1453#	1454	1464#	1486	1487#	1489	1492#
1434#	1494	1495#	1503	1508#	1509	1511#	1512	1519#	1524	1525#	1526	1527#
1493#	1553	1554#	1556	1560#	1561	1569#	1574	1575#	1576	1577#	1585	1590#
1535#	1593	1594#	1602	1607#	1608	1610#	1611	1621#	1636	1637#	1645	1650#
1591#	1652	1653#	1661	1666#	1667	1669#	1670	1678#	1682	1683#	1686	1687#
1651#	1712	1713#	1721	1726#	1727	1728#	1729	1737#	1742	1743#	1745	1746#
1697#	1759	1760#	1762	1763#	1773	1783#	1789	1797#	1802	1803#	1804	1805#
1754#	1818	1819#	1821	1822#	1830	1835#	1836	1838#	1839	1849#	1875	1876#
1813#	1907	1908#	1921	1940#	1941	1955#	1976	1977#	1979	1984#	1985	2002#
1888#	2033	2035#	2044	2045#	2048	2049#	2059	2064#	2065	2068#	2069	2079#
2032#	2085	2088#	2089	2099#	2121	2122#	2124	2133#	2134	2137#	2138	2148#
2084#	2154	2157#	2158	2168#	2173	2174#	2177	2178#	2188	2211	2212#	2214#
2153#	2224	2227#	2228	2238#	2243	2244#	2247	2248#	2258	2263	2264#	2267#
2223#	2278	2299#	2300	2302#	2311	2312#	2315	2316#	2326	2331	2332#	2335#
2268#	2346	2351#	2352	2355#	2356	2366#	2391	2392#	2394	2404#	2405	2407#
2336#	2416	2417#	2425	2431#	2432	2435#	2436	2443#	2452	2457#	2458	2461#
2408#	2469	2491#	2492	2494#	2504	2505#	2508	2509#	2517	2518#	2526	2531#
2462#	2536	2537#	2550	2555#	2556	2559#	2560	2573#	2594	2595#	2597	2609#
2532#	2611	2612#	2615	2616#	2619	2620#	2633	2638#	2639	2641#	2642	2655#
2610#	2677	2679#	2681	2682#	2689	2690#	2709	2713#	2714	2720#	2721	2731#
2676#	2765	2766#	2768	2769#	2772	2773#	2793	2798#	2799	2811#	2824	2825#
2750#	2827	2832#	2833	2851#	2859	2860#	2868	2885#	2886	2888#	2890	2891#
2826#	2893	2894#	2895	2900#	2901	2907#	2908	2909#	2910	2912#	2913	2914#
2892#	2919	2929#	2930	2931#	2932	2934#	2958	2959#	2961	2963#	2965	2966#
2916#	2984	2985#	2993	3011#	3012	3018#	3019	3028#	3029	3038#	3039	3050#
2977#	3053	3058#	3059	3070#	3071	3073#	3077	3078#	3091	3095#	3096	3100#
3051#												

\$TEMP = 000300

\$TESTN 001200

\$TIMES 001160

\$TKB 001146
\$TKS 001144
\$TN = 000042

\$TPB 001152
\$TPFLG 001157
\$TPS 001150
\$TRAP 015236
\$TRAP2 015260
\$TRP = 000012
\$TRPAD 015272
\$TSKO = 000242

3101#	3118#	3119#	3121#	3131#	3153#	3154#	3156#	3162#	3163#	3165#	3168#	3169#
3170#	3171#	3172#	3173#	3179#	3183#	3184#	3186#	3187#	3189#	3190#	3192#	
3193#	3195#	3196#	3197#	3198#	3207#	3208#	3209#	3210#	3213#	3214#	3215#	3216#
3217#	3220#	3221#	3228#	3229#	3238#	3239#	3247#	3249#	3252#	3253#	3254#	3255#
3257#	3258#	3260#	3264#	3265#	3266#	3267#	3273#	3276#	3277#	3279#	3280#	3281#
3333#	3334#	3335#	3336#	3337#	3338#	3339#	3341#	3342#	3351#	3352#	3372#	3373#
3379#	3381#	3385#	3386#	3388#	3389#	3391#	3392#	3410#	3418#	3419#	3421#	3422#
3423#	3446#	3447#	3448#	3449#	3450#	3451#	3453#	3454#	3455#	3456#	3458#	3459#
3461#	3462#	3471#	3472#	3475#	3476#	3477#	3478#	3508#	3509#	3514#	3516#	3519#
3520#	3522#	3523#	3523#	3534#	3535#	3536#	3542#	3545#	3546#	3548#	3549#	3556#
3557#	3559#	3560#	3561#	3562#	3563#	3564#	3568#	3590#	3592#	3593#	3594#	3595#
3598#	3599#	3602#	3603#	3605#	3606#	3607#	3620#	3621#	3635#	3636#	3643#	3646#
3661#	3662#	3673#	3674#	3681#	3683#	3684#	3685#	3694#	3695#	3697#	3699#	3700#
3704#	3705#	3721#	3722#	3724#	3727#	3728#	3740#	3741#	3747#	3750#	3765#	3766#
3777#	3778#	3781#	3783#	3784#	3785#	3794#	3795#	3797#	3800#	3801#	3825#	3826#
3828#	3855#	3856#	3858#	3859#	3860#	3861#	3864#	3865#	3866#	3867#	3870#	3871#
3873#	3874#	3876#	3877#	3880#	3881#	3886#	3887#	3890#	3891#	3893#	3894#	3900#
3903#	3914#	3917#	3918#	3919#	3920#	3940#	3941#	3943#	3945#	3947#	3948#	3954#
3955#	3956#	3975#	3976#	3978#	3980#	3982#	3984#	3995#	3996#	3997#	3998#	3999#
4002#	4003#	4004#	4012#	4013#	4015#	4040#	4041#	4044#	4045#	4049#	4050#	4052#
4053#	4067#	4076#	4142#	4143#	4144#	4145#	4146#	4147#	4158#	4159#	4160#	4161#
4163#	4164#	4165#	4173#	4174#	4175#	4178#	4179#	4181#	4191#	4192#	4193#	4194#
4206#	4217#	4239#	4240#	4241#	4245#	4251#	4254#	4255#	4256#	4257#	4258#	4259#
4260#	4262#	4263#	4267#	4285#	4286#	4291#	4294#	4299#	4302#	4303#	4304#	4306#
4307#	4312#	4328#	4329#	4368#	4390#	4391#	4392#	4393#	4399#	4400#	4401#	4403#
4404#	4405#	4406#	4407#	4408#	4409#	4410#	4411#	4412#	4414#	4415#	4416#	4417#
4418#	4419#	4420#	4421#	4422#	4424#	4425#	4428#	4429#	4430#	4431#	4439#	
1005#	1178#	1241#	1317#	1394#	1481#	1548#	1633#	1709#	1785#	1868#	1901#	1934#
1968#	2023#	2114#	2203#	2292#	2383#	2483#	2587#	2669#	2761#	2820#	2878#	3008#
3148#	3319#	3442#	3617#	3715#	3817#	4037#	4340#	4912#				
991#	1090#	1177#	1240#	1316#	1393#	1480#	1486#	1547#	1553#	1632#	1708#	1784#
1867#	1900#	1933#	1967#	1976#	2022#	2032#	2113#	2121#	2202#	2211#	2291#	2299#
2382#	2391#	2482#	2491#	2586#	2594#	2668#	2676#	2760#	2819#	2877#	2885#	2931#
2958#	3007#	3050#	3070#	3118#	3147#	3153#	3162#	3318#	3441#	3539#	3616#	3694#
3714#	3721#	3794#	3816#	3825#	4036#	4092#	4456#	4900#	4907#	4910#	4920#	
984#	4610#	4621#	4638#	4692#	4698#							
983#	4610#	4619#	4635#	4659#	4690#	4696#						
1#	673#	1170#	1177#	1178#	1234#	1240#	1241#	1312#	1316#	1317#	1389#	1393#
1394#	1471#	1480#	1481#	1487#	1542#	1547#	1548#	1554#	1628#	1632#	1633#	1704#
1708#	1709#	1780#	1784#	1785#	1863#	1867#	1868#	1896#	1900#	1901#	1929#	1933#
1934#	1962#	1967#	1968#	1977#	2016#	2022#	2023#	2033#	2107#	2113#	2114#	2122#
2196#	2202#	2203#	2212#	2286#	2291#	2292#	2300#	2374#	2382#	2383#	2392#	2476#
2482#	2483#	2492#	2576#	2582#	2586#	2587#	2595#	2662#	2668#	2669#	2677#	2754#
2760#	2761#	2815#	2819#	2820#	2871#	2877#	2878#	2886#	2932#	2959#	2996#	3003#
3007#	3008#	3051#	3071#	3089#	3119#	3134#	3140#	3147#	3148#	3154#	3163#	3268#
3309#	3318#	3319#	3434#	3441#	3442#	3540#	3611#	3616#	3617#	3695#	3710#	3714#
3715#	3722#	3795#	3811#	3816#	3817#	3826#	3922#	4030#	4036#	4037#	4080#	4088#
4092#												
986#	4596#	4607#										
990#	4545#	4607#										
985#	4594#	4607#										
1085#	5073#											
5084#	5095#											
5088#	5097#	5098#	5099#	5100#	5101#	5102#	5103#	5104#	5105#	5106#	5107#	
5078#	5095#											
1192#	1215#	1250#	1254#	1266#	1270#	1283#	1287#	1302#	1306#	1326#	1330#	1342#

\$TSK1 = 000246

\$TSK2 = 000245

\$TSK3 = 000234

\$TSK4 = 000146

\$STM 001004

\$STNM 001102

\$ITYIN 013572

\$TYPB= ***** U

\$TYPOS 014564

\$TYPE 012600

\$YPEC 013012

\$YPEX 013060

\$YPOC 015034

\$YPON 015050

\$YPOS 015010

\$UNIT 001206

\$UNITM 001010

\$USWR 001220

\$VECT1 001244

\$VECT2 001246

\$XTSTR 014316

\$YESNO= 000001

1346#	1359#	1363#	1378#	1382#	1408#	1412#	1424#	1428#	1441#	1445#	1460#	1464#
1485#	1489#	1499#	1503#	1515#	1519#	1531#	1535#	1552#	1556#	1565#	1569#	1581#
1585#	1598#	1602#	1617#	1621#	1641#	1645#	1657#	1661#	1674#	1678#	1693#	1697#
1717#	1721#	1733#	1737#	1750#	1754#	1769#	1773#	1793#	1797#	1809#	1813#	1826#
1830#	1845#	1849#	1879#	1888#	1912#	1921#	1944#	1955#	1974#	1979#	1989#	2002#
2028#	2035#	2053#	2059#	2073#	2079#	2093#	2099#	2119#	2124#	2142#	2148#	2162#
2168#	2182#	2188#	2208#	2214#	2232#	2238#	2252#	2258#	2272#	2278#	2297#	2302#
2320#	2326#	2340#	2346#	2360#	2366#	2388#	2394#	2421#	2425#	2439#	2443#	2448#
2452#	2465#	2469#	2488#	2494#	2522#	2526#	2546#	2550#	2569#	2573#	2592#	2597#
2629#	2633#	2651#	2655#	2675#	2679#	2705#	2709#	2727#	2731#	2746#	2750#	2789#
2793#	2807#	2811#	2847#	2851#	2864#	2868#	2884#	2888#	2903#	2916#	2924#	2934#
2943#	2965#	2972#	2977#	2989#	2993#	3044#	3053#	3062#	3073#	3083#	3091#	3111#
3121#	3126#	3131#	3152#	3156#	3161#	3165#	3177#	3179#	3182#	3258#	3364#	3381#
3406#	3410#	3499#	3516#	3537#	3542#	3573#	3590#	3641#	3643#	3646#	3700#	3719#
3724#	3745#	3747#	3750#	3801#	3824#	3828#	3899#	3900#	3910#	3914#	3952#	3956#
3989#	4004#	4010#	4015#	4063#	4067#	4072#	4076#	4153#	4193#	4203#	4206#	4249#
4251#	4254#	4260#	4289#	4291#	4294#	4307#	4383#	4424#				
1203#	1207#	2906#	2910#	2911#	2914#	2945#	2965#	3182#	3257#	3368#	3373#	3374#
3379#	3503#	3509#	3510#	3514#	3577#	3583#	3584#	3588#	3646#	3699#	3750#	3800#
3993#	3999#	4153#	4178#	4193#	4254#	4259#	4294#	4306#	4386#	4412#	4413#	4422#
2949#	2963#	3200#	3220#	3226#	3229#	3230#	3249#	3690#	3697#	3790#	3797#	4157#
4161#	4162#	4165#	4171#	4175#	4297#	4299#	4302#	4304#	4389#	4393#	4394#	4401#
2952#	2961#	3202#	3220#	3236#	3239#	3240#	3247#	4302#	4303#			
3205#	3210#	3211#	3217#									
950#												
963#	4455#	4821#	4857#	4862#	4889#	4911#	4912#	4917#	4921#			
4717#	4718#	4735#	4739#									
5101#												
4933#	5100#											
4545#	4777#	5088#	5096#									
4575#	4582#	4589#	4594#	4595#	4661#							
4600#	4602#	4605#										
5018#	5097#											
5017#	5020#	5099#										
5013#	5098#											
1008#												
952#												
1015#	1483#	1550#	1972#	2026#	2117#	2206#	2295#	2386#	2486#	2590#	3150#	3260#
3535#	3717#	4241#										
1040#	4409#											
1041#												
4876#												
1156#	1157#	1159#	1160#	1162#	1163#	1165#	1166#	1210#	1211#	1213#	1214#	1222#
1223#	1261#	1262#	1278#	1279#	1295#	1296#	1337#	1338#	1354#	1355#	1371#	1372#
1419#	1420#	1436#	1437#	1453#	1454#	1494#	1495#	1511#	1512#	1526#	1527#	1576#
1577#	1593#	1594#	1610#	1611#	1652#	1653#	1669#	1670#	1686#	1687#	1728#	1729#
1745#	1746#	1762#	1763#	1804#	1805#	1821#	1822#	1838#	1839#	2048#	2049#	2068#
2069#	2088#	2089#	2137#	2138#	2157#	2158#	2177#	2178#	2227#	2228#	2247#	2248#
2267#	2268#	2315#	2316#	2335#	2336#	2355#	2356#	2407#	2408#	2435#	2436#	2461#
2462#	2508#	2509#	2536#	2537#	2559#	2560#	2609#	2610#	2619#	2620#	2641#	2642#
2765#	2766#	2824#	2825#	2890#	2891#	2912#	2913#	3172#	3173#	3178#	3179#	3213#
3214#	3215#	3216#	3261#	3264#	3329#	3330#	3341#	3342#	3351#	3352#	3391#	3392#
3420#	3421#	3458#	3459#	3461#	3462#	3485#	3486#	3519#	3520#	3522#	3523#	3548#
3549#	3556#	3557#	3559#	3560#	3561#	3562#	3592#	3593#	3594#	3595#	3604#	3605#
3620#	3621#	3642#	3643#	3682#	3683#	3684#	3685#	3704#	3705#	3727#	3728#	3746#
3747#	3782#	3783#	3784#	3785#	3886#	3887#	3890#	3891#	3893#	3894#	3917#	3918#

N11

MAINDEC-ZZ-CVDVA-B
CVDVAB.P11

MACY11 30(1046)
15-DEC-77 08:58

19-DEC-77 08:25 PAGE 145
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0143

SSARGC= 000000
SSBYTE= 000403

SSDST = 000067
SSFLAG= 000001

SSFROM= 000000

SSGET4= 000000
SSLLOC = 012246

3919#	3920#	3940#	3941#	3944#	3945#	3954#	3955#	3975#	3976#	3979#	3980#	3983#
3984#	4002#	4003#	4012#	4013#	4040#	4041#	4049#	4050#	4191#	4192#	4242#	4245#
4250#	4251#	4255#	4256#	4257#	4258#	4262#	4263#	4290#	4291#	4298#	4299#	4328#
4329#	4416#	4417#	4418#	4419#	4420#	4421#	4430#	4431#				
4118#	4225#	4274#	4335#	4373#								
1201#	1248#	1264#	1281#	1300#	1324#	1340#	1357#	1376#	1406#	1422#	1439#	1458#
1483#	1497#	1513#	1529#	1550#	1563#	1579#	1596#	1615#	1639#	1655#	1672#	1691#
1715#	1731#	1748#	1767#	1791#	1807#	1824#	1843#	1877#	1910#	1942#	1972#	1987#
2026#	2051#	2071#	2091#	2117#	2140#	2160#	2180#	2206#	2230#	2250#	2270#	2295#
2318#	2338#	2358#	2386#	2419#	2437#	2446#	2463#	2486#	2520#	2544#	2567#	2590#
2627#	2649#	2673#	2725#	2805#	2862#	2882#	2904#	2922#	2943#	2947#	2950#	2970#
2987#	3042#	3060#	3081#	3109#	3124#	3150#	3159#	3200#	3203#	3224#	3232#	3234#
3362#	3366#	3404#	3497#	3501#	3535#	3571#	3575#	3688#	3717#	3788#	3822#	3908#
3950#	3987#	3991#	4008#	4070#	4155#	4169#	4201#	4384#	4387#			
3261#	4242#											
1201#	1203#	1207#	1246#	1250#	1254#	1264#	1266#	1270#	1281#	1283#	1287#	1300#
1302#	1306#	1324#	1326#	1330#	1340#	1342#	1346#	1357#	1359#	1363#	1376#	1378#
1382#	1406#	1408#	1412#	1422#	1424#	1428#	1439#	1441#	1445#	1458#	1460#	1464#
1483#	1485#	1489#	1497#	1499#	1503#	1513#	1515#	1519#	1529#	1531#	1535#	1550#
1552#	1556#	1563#	1565#	1569#	1579#	1581#	1585#	1596#	1598#	1602#	1615#	1617#
1621#	1639#	1641#	1645#	1655#	1657#	1661#	1672#	1674#	1678#	1691#	1693#	1697#
1715#	1717#	1721#	1731#	1733#	1737#	1748#	1750#	1754#	1767#	1769#	1773#	1791#
1793#	1797#	1807#	1809#	1813#	1824#	1826#	1830#	1843#	1845#	1849#	1877#	1879#
1888#	1910#	1912#	1921#	1942#	1944#	1955#	1972#	1974#	1979#	1987#	1989#	2002#
2026#	2028#	2035#	2051#	2053#	2059#	2071#	2073#	2079#	2091#	2093#	2099#	2117#
2119#	2124#	2140#	2142#	2148#	2160#	2162#	2168#	2180#	2182#	2188#	2206#	2208#
2214#	2230#	2232#	2238#	2250#	2252#	2258#	2270#	2272#	2278#	2295#	2297#	2302#
2318#	2320#	2326#	2338#	2340#	2346#	2358#	2360#	2366#	2386#	2388#	2394#	2419#
2421#	2425#	2437#	2439#	2443#	2446#	2448#	2452#	2463#	2465#	2469#	2486#	2488#
2494#	2520#	2532#	2546#	2548#	2554#	2567#	2569#	2573#	2575#	2590#	2592#	2597#
2627#	2629#	2633#	2649#	2651#	2655#	2673#	2675#	2679#	2704#	2709#	2725#	2727#
2731#	2745#	2750#	2788#	2793#	2805#	2807#	2811#	2846#	2851#	2862#	2864#	2868#
2882#	2884#	2888#	2904#	2911#	2914#	2922#	2924#	2934#	2942#	2943#	2945#	2947#
2949#	2950#	2952#	2961#	2963#	2970#	2972#	2977#	2987#	2989#	2993#	3042#	3044#
3053#	3060#	3062#	3073#	3081#	3083#	3091#	3109#	3111#	3121#	3124#	3126#	3131#
3150#	3152#	3156#	3159#	3161#	3165#	3199#	3200#	3202#	3203#	3205#	3217#	3224#
3226#	3232#	3234#	3236#	3247#	3249#	3262#	3264#	3266#	3268#	3279#	3281#	3404#
3406#	3410#	3497#	3499#	3501#	3503#	3514#	3516#	3535#	3537#	3542#	3571#	3573#
3575#	3577#	3588#	3590#	3688#	3690#	3697#	3717#	3719#	3724#	3788#	3790#	3797#
3822#	3824#	3828#	3908#	3910#	3914#	3950#	3952#	3956#	3987#	3989#	3991#	3993#
3999#	4004#	4008#	4010#	4015#	4062#	4067#	4070#	4072#	4076#	4155#	4157#	4165#
4169#	4171#	4175#	4201#	4203#	4206#	4384#	4386#	4387#	4389#	4401#	4422#	
1141#	2410#	2511#	2539#	2562#	2622#	2644#	2694#	2737#	2778#	2837#	3021#	3031#
3103#	3355#	3399#	3476#	3488#	3564#	3630#	3651#	3664#	3736#	3755#	3767#	3831#
4054#	4186#	4395#										
4471#												
1202#	1203#	1216#	1217#	1249#	1250#	1265#	1266#	1282#	1283#	1301#	1302#	1325#
1326#	1341#	1342#	1358#	1359#	1377#	1378#	1407#	1408#	1423#	1424#	1440#	1441#
1459#	1460#	1484#	1485#	1498#	1499#	1514#	1515#	1530#	1531#	1551#	1552#	1564#
1565#	1580#	1581#	1597#	1598#	1616#	1617#	1640#	1641#	1656#	1657#	1673#	1674#
1692#	1693#	1716#	1717#	1732#	1733#	1749#	1750#	1768#	1769#	1792#	1793#	1808#
1809#	1825#	1826#	1844#	1845#	1878#	1879#	1911#	1912#	1943#	1944#	1973#	1974#
1988#	1989#	2027#	2028#	2052#	2053#	2072#	2073#	2092#	2093#	2118#	2119#	2141#
2142#	2161#	2162#	2181#	2182#	2207#	2208#	2231#	2232#	2251#	2252#	2271#	2272#
2296#	2297#	2319#	2320#	2339#	2340#	2359#	2360#	2387#	2388#	2420#	2421#	2438#
2439#	2447#	2448#	2464#	2465#	2487#	2488#	2521#	2522#	2545#	2546#	2568#	2569#

\$OFILL	015233
\$1	002102
\$10	002430
\$100	005256
\$101	005320
\$102	005360
\$103	005446
\$104	005510
\$105	005546
\$106	005614
\$107	005642
\$11	002456
\$110	005676
\$111	005770
\$112	006012
\$113	006104
\$114	005122
\$115	005160
\$116	006200
\$117	006216
\$12	002506
\$120	006220
\$121	006234
\$122	006260
\$123	006260
\$124	006324
\$125	006322
\$126	006322
\$127	006336
\$13	002556
\$130	006354
\$131	006470
\$132	006516
\$133	006540
\$134	006610
\$135	006622
\$136	006662

2591#	2592#	2628#	2629#	2650#	2651#	2674#	2675#	2704#	2705#	2726#	2727#	2745#
2746#	2788#	2789#	2806#	2807#	2816#	2847#	2862#	2864#	2883#	2884#	2905#	2906#
2919#	2920#	2923#	2924#	2944#	2945#	2948#	2949#	2951#	2952#	2971#	2972#	2988#
2989#	3043#	3044#	3061#	3062#	3082#	3083#	3110#	3111#	3125#	3126#	3151#	3152#
3160#	3161#	3201#	3202#	3204#	3205#	3225#	3226#	3233#	3234#	3235#	3236#	3263#
3364#	3367#	3368#	3405#	3406#	3498#	3499#	3502#	3503#	3536#	3537#	3572#	3573#
3576#	3577#	3689#	3690#	3718#	3719#	3789#	3790#	3823#	3824#	3903#	3904#	3909#
3910#	3951#	3952#	3988#	3989#	3992#	3993#	4009#	4010#	4062#	4063#	4071#	4072#
4156#	4157#	4170#	4171#	4202#	4203#	4385#	4386#	4388#	4389#	4425#	4426#	
4118#	4220#	4225#	4269#	4274#	4314#	4335#	4370#	4373#	4441#			
4118#	4204#	4208#	4218#	4225#	4268#	4274#	4313#	4335#	4369#	4373#	4440#	
4118#	4204#	4217#	4225#	4264#	4267#	4274#	4312#	4335#	4368#	4373#	4432#	4439#
4118#	4209#	4219#	4225#	4268#	4274#	4313#	4335#	4369#	4373#	4440#		
3261#	4242#											
1141#	1142#	2410#	2413#	2511#	2514#	2539#	2542#	2562#	2565#	2622#	2625#	2644#
2647#	2694#	2700#	2737#	2743#	2778#	2784#	2837#	2843#	3021#	3024#	3031#	3034#
3103#	3106#	3355#	3358#	3399#	3402#	3476#	3482#	3488#	3491#	3564#	3567#	3630#
3631#	3632#	3651#	3657#	3664#	3670#	3736#	3737#	3738#	3755#	3761#	3767#	3773#
3831#	3832#	3833#	4054#	4060#	4186#	4189#	4395#	4396#				
5014#	5018#	5028#	5063#									
1191#	1216#											
1341#	1346#											
2545#	2550#											
2568#	2573#											
2591#	2597#											
2628#	2633#											
2650#	2655#											
2674#	2679#											
2704#	2709#											
2726#	2731#											
1358#	1363#											
2745#	2750#											
2788#	2793#											
2806#	2811#											
2846#	2851#											
2863#	2868#											
2883#	2888#											
2902#	2919#											
2905#	2910#											
1377#	1382#											
2909#	2914#											
2917#	2920#											
2923#	2934#											
2942#	2965#				</							

MAINDEC-22-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 147
 CVDVAB.P11 15-DEC-77 08:58 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0145

\$137	006702	3160	3165#
\$14	002604	1423	1428#
\$140	006732	3176	3179#
\$141	006730	3177#	3257
\$142	007100	3181	3258#
\$143	006770	3199#	3220
\$144	007024	3201	3221#
\$145	007012	3204	3210#
\$146	007022	3209	3217#
\$147	007036	3225	3229#
\$15	002632	1440	1445#
\$150	007062	3228	3249#
\$151	007060	3233	3235
\$152	007062	3238	3247#
\$153	007352	3363	3381#
\$154	007350	3367	3373#
\$155	007352	3372	3379#
\$156	007430	3405	3410#
\$157	007664	3498	3516#
\$16	002662	1459	1464#
\$160	007662	3502	3509#
\$161	007664	3508	3514#
\$162	007726	3536	3542#
\$163	010034	3572	3590#
\$164	010032	3576	3583#
\$165	010034	3582	3588#
\$166	010164	3640	3643#
\$167	010162	3641#	3699
\$17	002720	1484	1489#
\$170	010310	3645	3700#
\$171	010306	3689	3697#
\$172	010354	3718	3724#
\$173	010420	3744	3747#
\$174	010416	3745#	3800
\$175	010544	3749	3801#
\$176	010542	3789	3797#
\$177	010602	3823	3828#
\$2	002126	1202	1207#
\$20	002746	1498	1503#
\$200	010714	3898#	3903
\$201	010730	3901	3904#
\$202	010740	3909	3914#
\$203	011020	3951	3956#
\$204	011100	3988	4004#
\$205	011074	3983	3999#
\$206	011114	4009	4015#
\$207	011222	4062	4067#
\$21	002774	1514	1519#
\$210	011232	4071	4076#
\$211	011556	4204	4217#
\$212	011560	4209	4219#
\$213	011426	4152#	4193
\$214	011532	4179#	4181
\$215	011446	4156	4161#
\$216	011454	4160	4165#
\$217	011472	4170	4175#

3239#

4194#

\$22	003022	1530	1535#
\$220	011544	4202	4206#
\$221	011660	4264	4267#
\$222	011660	4268#	
\$223	011624	4248	4251#
\$224	011620	4249#	4259#
\$225	011650	4253	4260#
\$226	011740	4312#	
\$227	011740	4313#	
\$23	003060	1551	1556#
\$230	011706	4288	4291#
\$231	011702	4289#	4306
\$232	011732	4293	4307#
\$233	011720	4296	4299#
\$234	011716	4297#	4303
\$235	011730	4301	4304#
\$236	012132	4368#	
\$237	012132	4369#	
\$24	003100	1564	1569#
\$240	012272	4432	4439#
\$241	012272	4440#	
\$242	012134	4382#	4425
\$243	012220	4385	4412#
\$244	012160	4388	4393#
\$245	012166	4392	4401#
\$246	012240	4411	4422#
\$25	003126	1580	1585#
\$26	003154	1597	1602#
\$27	003204	1616	1621#
\$3	002240	1249	1254#
\$30	003242	1640	1645#
\$31	003270	1656	1661#
\$32	003316	1673	1678#
\$33	003346	1692	1697#
\$34	003404	1716	1721#
\$35	003432	1732	1737#
\$36	003460	1749	1754#
\$37	0035.0	1768	1773#
\$4	002266	1265	1270#
\$40	003546	1792	1797#
\$40CAT=	***** U	4831	4873
\$41	003574	1808	1813#
\$42	003622	1825	1830#
\$43	003652	1844	1849#
\$44	003712	1878	1888#
\$45	003752	1911	1921#
\$46	004012	1943	1955#
\$47	004050	1973	1979#
\$5	002314	1282	1287#
\$50	004072	1988	2002#
\$51	004130	2027	2035#
\$52	004156	2052	2059#
\$53	004204	2072	2079#
\$54	004232	2092	2099#
\$55	004270	2118	2124#
\$56	004316	2141	2148#

MAINDEC-ZZ-CVDVA-B MACY11 30(1046)
CVDVAB.P11 15-DEC-77 08:58

19-DEC-77 08:25 PAGE 149
CROSS REFERENCE TABLE -- USER SYMBOLS

[illegible]

BEGIN	1#														
BGNHRD	1#														
BGNHW	1#														
BGNINI	1#														
BGNMOD	1#	4114													
BGNMSG	1#														
BGNSFT	1#														
BGNSRV	1#	3934	3969	4318											
BGNSUB	1#	1192	1244	1258	1274	1291	1320	1334	1350	1367	1402	1416	1432	1449	1491
	1507	1523	1559	1573	1589	1606	1635	1649	1665	1681	1711	1725	1741	1758	1787
	1801	1817	1834	1874	1906	1939	1983	2043	2063	2083	2132	2152	2172	2222	2242
	2262	2310	2330	2350	2403	2430	2456	2503	2530	2554	2614	2637	2680	2712	2767
	2797	2825	3010	3057	3076	3094	3337	3384	3452	3532					
BGNSW	1#														
BRESET	1#	1168	1297	1373	1455	1612	1688	1764	1840	1885	1918	1952	1999	2801	3411
	3882	4077													
CALL	1#	1140	2410	2511	2539	2562	2622	2644	2693	2736	2777	2836	3021	3031	3103
	3355	3399	3475	3488	3564	3629	3650	3663	3735	3754	3766	3830	4053	4186	4394
CASE	1#														
CKLOOP	1#	1850	1890	1923	1956	2004									
CLRVEC	1#	1217	3415	3599											
COMMEN	795#														
DECR	1#														
DECRU	1#														
DEFAULT	1#														
DEVREG	1#														
DEVTYP	1#														
DISPAT	1#														
ELSE	1#	2908	3208	3227	3237	3371	3507	3581	4159	4391	4410				
END	1#														
ENDCLM	1#														
ENDCOM	795#														
ENDDEC	1#														
ENDDO	1#	2964	3219												
ENDHRD	1#														
ENDHW	1#														
ENDIF	1#	1206	1253	1269	1286	1305	1329	1345	1362	1381	1411	1427	1444	1463	1488
	1502	1518	1534	1555	1568	1584	1601	1620	1644	1660	1677	1696	1720	1736	1753
	1772	1796	1812	1829	1848	1887	1920	1954	1978	2001	2034	2058	2078	2098	2123
	2147	2167	2187	2213	2237	2257	2277	2301	2325	2345	2365	2393	2424	2442	2451
	2468	2493	2525	2549	2572	2596	2632	2654	2678	2708	2730	2749	2752	2810	2850
	2867	2887	2913	2933	2960	2962	2976	2992	3052	3072	3090	3120	3130	3155	3164
	3216	3246	3248	3378	3380	3409	3513	3515	3541	3587	3589	3696	3723	3796	3827
	3913	3955	3998	4003	4014	4066	4075	4164	4174	4205	4400	4421			
ENDINC	1#	3256	3698	3799	4258	4302	4305								
ENDINI	1#														
ENDLOO	1#	4192													
ENDMOD	1#														
ENDMSG	1#														
ENDRTN	1#	4216	4266	4311	4367	4438									
ENDSEL	1#														
ENDSFT	1#														
ENDSRV	1#	3962	4022	4329											
ENDSUB	1#	1208	1255	1271	1288	1307	1331	1347	1364	1383	1413	1429	1446	1465	1504
	1520	1536	1570	1586	1603	1622	1646	1662	1679	1698	1722	1738	1755	1774	1798
	1814	1831	1851	1891	1924	1957	2005	2060	2080	2100	2149	2169	2189	2239	2259

	2279	2327	2347	2367	2427	2453	2470	2527	2551	2574	2634	2656	2710	2751	2795
	2812	2852	3054	3074	3092	3132	3382	3413	3523	3595					
ENDSW	1#														
ENDTST	1#	1228	1308	1384	1466	1537	1623	1699	1775	1852	1892	1925	1958	2006	2101
	2190	2280	2368	2471	2577	2657	2752	2813	2869	2999	3136	3304	3426	3609	3705
	3805	4025	4086												
EQUALS	1#														
ERRDF	1#	1204													
ERRHRD	1#	1251	1267	1284	1303	1327	1343	1360	1379	1409	1425	1442	1461	1500	1516
	1532	1566	1582	1599	1618	1642	1658	1675	1694	1718	1734	1751	1770	1794	1810
	1827	1846	1882	1915	1945	1991	2054	2074	2094	2143	2163	2183	2233	2253	2273
	2321	2341	2361	2422	2440	2449	2466	2523	2547	2570	2630	2652	2706	2728	2747
	2790	2808	2848	2865	2926	2954	2973	2990	3045	3064	3085	3112	3127	3244	3369
	3376	3407	3504	3511	3578	3585	3691	3791	3911	4064	4073				
ERROR	689#	1205	1252	1268	1285	1304	1328	1344	1361	1380	1410	1426	1443	1462	1501
	1517	1533	1567	1583	1600	1619	1643	1659	1676	1695	1719	1735	1752	1771	1795
	1811	1828	1847	1883	1916	1946	1992	2055	2075	2095	2144	2164	2184	2234	2254
	2274	2322	2342	2362	2423	2441	2450	2467	2524	2548	2571	2631	2653	2707	2729
	2748	2791	2809	2849	2866	2927	2955	2974	2991	3046	3065	3086	3113	3128	3245
	3370	3377	3408	3505	3512	3579	3586	3692	3792	3912	4065	4074			
ESCAPE	795#														
EXIF	1#														
EXIFB	1#	4177													
EXIT	1#	1485	1552	1975	2031	2120	2210	2298	2390	2490	2575	2593	2675	2884	2930
	2957	2995	3049	3069	3088	3117	3133	3152	3161	3267	3538	3693	3720	3793	3824
	3921	4079													
GETPRI	795#														
GETSWR	795#	1123#													
GPHARD	1#														
GPRMA	1#														
GPRMD	1#														
GPRML	1#														
HEADER	1#														
IF	1#	1200	1247	1263	1280	1299	1323	1339	1356	1375	1405	1421	1438	1457	1482
	1496	1512	1528	1549	1562	1578	1595	1614	1638	1654	1671	1690	1714	1730	1747
	1766	1790	1806	1823	1842	1876	1909	1941	1971	1986	2025	2050	2070	2090	2116
	2139	2159	2179	2205	2229	2249	2269	2294	2317	2337	2357	2385	2418	2436	2445
	2462	2485	2519	2543	2566	2589	2626	2648	2672	2724	2804	2861	2881	2903	2921
	2946	2949	2969	2986	3041	3059	3080	3108	3123	3149	3158	3202	3223	3231	3361
	3365	3403	3496	3500	3534	3570	3574	3687	3716	3787	3821	3907	3949	3986	3990
	4007	4154	4383	4386											
IFB	1#	4069	4168	4200											
IFCONO	1#														
IF.ERP	1#	2703	2744	2787	2845	4061									
IF.NO.	1#														
INCR	1#	3174	3638	3742	4246	4294									
INCRU	1#	4286													
INLINE	1#														
LASTAD	1#														
LEAVE	1#														
LET	1#	1137	1144	1147	1149	1152	1154	1157	1160	1163	1166	1179	1183	1184	1185
	1186	1187	1188	1193	1195	1209	1211	1220	1221	1223	1224	1245	1259	1260	1275
	1277	1292	1294	1321	1335	1336	1351	1353	1368	1370	1403	1417	1418	1433	1435
	1450	1452	1486	1492	1493	1508	1510	1524	1525	1553	1560	1574	1575	1590	1592
	1607	1609	1636	1650	1651	1666	1668	1682	1685	1712	1726	1727	1742	1744	1759
	1761	1788	1802	1803	1818	1820	1835	1837	1875	1907	1940	1976	1984	2032	2044

	2047	2064	2067	2084	2087	2121	2133	2136	2153	2156	2173	2176	2211	2223	2226
	2243	2246	2263	2266	2299	2311	2314	2331	2334	2351	2354	2391	2404	2406	2415
	2431	2434	2457	2460	2491	2504	2507	2516	2531	2535	2555	2558	2594	2608	2610
	2615	2618	2638	2640	2676	2681	2688	2713	2719	2764	2768	2771	2798	2823	2826
	2831	2858	2885	2889	2891	2893	2899	2906	2911	2928	2931	2958	2983	3011	3017
	3027	3037	3050	3058	3070	3077	3095	3099	3118	3153	3162	3167	3169	3171	3182
	3185	3188	3191	3194	3196	3206	3213	3214	3251	3253	3259	3264	3281	3325	3328
	3332	3333	3334	3335	3336	3338	3340	3350	3385	3387	3390	3418	3419	3421	3422
	3446	3447	3448	3449	3450	3453	3454	3457	3460	3470	3484	3492	3518	3521	3533
	3539	3544	3547	3555	3558	3560	3591	3593	3597	3602	3603	3605	3606	3619	3634
	3660	3672	3680	3683	3694	3703	3721	3726	3739	3764	3776	3780	3783	3794	3825
	3854	3857	3859	3863	3865	3869	3872	3875	3879	3885	3889	3892	3916	3918	3939
	3942	3946	3953	3974	3977	3981	3994	3996	4001	4011	4039	4043	4048	4051	4141
	4143	4145	4157	4162	4172	4190	4238	4240	4254	4256	4261	4284	4327	4389	4398
	4402	4404	4406	4408	4413	4415	4417	4419	4427	4429					
LOCAL	1#														
LOOP	1#	4151													
MSG	1170#	1172	1234#	1236	1471#	1473	1542#	1544	2016#	2018	2107#	2109	2196#	2198	2286#
	2288	2374#	2376	2476#	2478	2662#	2664	2754#	2756	2871#	2873	3140#	3142	3309#	3311
	3434#	3436	3611#	3613	3811#	3813	4030#	4032							
MULT	795#														
NEWTST	795#	1170	1234	1312	1389	1471	1542	1628	1704	1780	1863	1896	1929	1963	2016
	2107	2196	2286	2374	2476	2582	2662	2754	2815	2871	3003	3140	3309	3434	3611
	3710	3811	4030	4088											
NOLOCA	1#														
POINTE	1#														
POP	795#	1225	3423	3607	4308	4508	4509	4792	4793	4974					
PRINTB	1#														
PUSH	795#	1218	3416	3600	4281	4489	4495	4753	4755	4776	4933				
REPEAT	1#	1190	2901	3897	4381										
REPORT	1#	795#													
RETURN	1#	4203	4207	4263	4431										
ROUTIN	1#	4116	4223	4272	4333	4371									
SAVR14	1#														
SCOPE	690#	1176	1239	1315	1392	1479	1546	1631	1707	1783	1866	1899	1932	1966	2021
	2112	2201	2290	2381	2481	2585	2667	2759	2818	2876	3006	3146	3317	3440	3615
	3713	3815	4035	4091	4454										
SELECT	1#														
SETPRI	795#	1396	2397	2497	2600	3344	3393	3464	3551	3624	3730	3849			
SETTRA	5088#	5097	5098	5099	5100	5102	5104	5105	5106						
SETUP	795#	1072													
SETVEC	1#	1182	3331	3445											
SKIP	795#	1487	1554	1977	2033	2122	2212	2300	2392	2492	2576	2595	2677	2886	2932
	2959	2996	3051	3071	3089	3119	3134	3154	3163	3268	3540	3695	3722	3795	3826
	3922	4080													
SLASH	795#														
SPACE	795#														
STARS	795#														
	956	811	813	831	833	851	853	873	875	894	909	923	934	936	943
	1470	997	1000	1170	1175	1229	1231	1234	1238	1311	1312	1314	1388	1389	1391
	1857	1471	1478	1541	1542	1545	1627	1628	1630	1703	1704	1706	1779	1780	1782
	2106	1860	1863	1865	1896	1898	1929	1931	1962	1963	1965	2008	2012	2016	2020
	2581	2107	2111	2195	2196	2200	2285	2286	2289	2373	2374	2380	2475	2476	2480
	3002	2582	2584	2661	2662	2666	2753	2754	2758	2814	2815	2817	2870	2871	2875
	3709	3003	3005	3139	3140	3145	3308	3309	3316	3433	3434	3439	3610	3611	3614
	4271	3710	3712	3810	3811	3814	4029	4030	4034	4088	4090	4115	4136	4222	4236
		4280	4317	4324	4335	4373	4380	4446	4485	4501	4530	4609	4612	4680	4709

MAINDEC-22-CVDVA-B MACY11 30(1046) 19-DEC-77 08:25 PAGE 154
 CVDVAB.P11 15-DEC-77 08:58 CROSS REFERENCE TABLE --- MACRO NAMES

SEQ 0151

	4748	4805	4859	4923	4990	5067									
STRUCT	1#														
SWRSU	795#	1095#													
TRMTRP	5088#														
TYPBIN	795#														
TYPDEC	795#	4109	4466												
TYPNAM	795#	1116													
TYPNUM	795#														
TYPOCS	795#														
TYPOCT	795#	4097	4103	4340	4347	4353	4359	4365	4630						
TYPTXT	795#	4093	4099	4105	4336	4342	4349	4355	4361						
UNTIL	1#	1214	2915	3899	4423										
UNTILB	1#														
WAITMS	1#	2409	2510	2538	2561	2621	2643	3020	3030	3102	3354	3398	3487	3563	4185
WHILE	1#	2941	3198												
WHILEB	1#														
\$ADDON	1#														
	1340	1191	1192	1201	1203	1248	1250	1264	1266	1281	1283	1300	1302	1324	1326
	1485	1342	1357	1359	1376	1378	1406	1408	1422	1424	1439	1441	1458	1460	1483
	1615	1497	1499	1513	1515	1529	1531	1550	1552	1563	1565	1579	1581	1596	1598
	1750	1617	1639	1641	1655	1657	1672	1674	1691	1693	1715	1717	1731	1733	1748
	1942	1767	1769	1791	1793	1807	1809	1824	1826	1843	1845	1877	1879	1910	1912
	2119	1944	1972	1974	1987	1989	2026	2028	2051	2053	2071	2073	2091	2093	2117
	2295	2140	2142	2160	2162	2180	2182	2206	2208	2230	2232	2250	2252	2270	2272
	2448	2297	2318	2320	2338	2340	2358	2360	2386	2388	2419	2421	2437	2439	2446
	2649	2463	2465	2486	2488	2520	2522	2544	2546	2567	2569	2590	2592	2627	2629
	2847	2651	2673	2675	2704	2705	2725	2727	2745	2746	2788	2789	2805	2807	2846
	2945	2862	2864	2882	2884	2902	2903	2904	2906	2911	2921	2922	2924	2942	2943
	3083	2947	2949	2950	2952	2965	2970	2972	2987	2989	3042	3044	3060	3062	3081
	3202	3109	3111	3124	3126	3150	3152	3159	3161	3175	3177	3178	3182	3199	3200
	3374	3203	3205	3211	3220	3224	3226	3230	3232	3236	3240	3362	3364	3366	3368
	3639	3404	3406	3497	3499	3501	3503	3510	3535	3537	3571	3573	3575	3577	3584
	3824	3641	3642	3646	3688	3690	3717	3719	3743	3745	3746	3750	3788	3790	3822
	4063	3898	3899	3905	3908	3910	3950	3952	3987	3989	3991	3993	4008	4010	4062
	4247	4070	4072	4118	4152	4153	4155	4157	4162	4169	4171	4193	4201	4203	4225
	4382	4249	4250	4254	4274	4287	4289	4290	4294	4295	4297	4298	4302	4335	4373
		4383	4384	4386	4387	4389	4394	4413							
\$AND	1#														
\$BRANC	1#														
	1484	3232													
	1749	1202	1216	1249	1265	1282	1301	1325	1341	1358	1377	1407	1423	1440	1459
	2118	1498	1514	1530	1551	1564	1580	1597	1616	1640	1656	1673	1692	1716	1732
	2447	1768	1792	1808	1825	1844	1878	1911	1943	1973	1988	2027	2052	2072	2092
	2846	2141	2161	2181	2207	2231	2251	2271	2296	2319	2339	2359	2387	2420	2438
	3061	2464	2487	2521	2545	2568	2591	2628	2650	2674	2704	2726	2745	2788	2806
	3235	2863	2883	2905	2909	2917	2919	2923	2944	2948	2951	2965	2971	2988	3043
	3645	3082	3110	3125	3151	3160	3176	3181	3201	3204	3209	3220	3225	3228	3233
	4009	3238	3257	3363	3367	3372	3405	3498	3502	3508	3536	3572	3576	3582	3640
	4264	3689	3699	3718	3744	3749	3789	3800	3823	3901	3903	3909	3951	3988	3992
\$BRCOD	1#	4062	4071	4156	4160	4170	4179	4181	4193	4202	4204	4209	4248	4253	4259
\$CALL	1#	4288	4293	4296	4301	4303	4306	4385	4388	4392	4411	4425	4432		
		2916	3180	3644	3748	3900	4178	4180	4252	4292	4300				
\$CHECK	1#														
	3355	1141	2410	2511	2539	2562	2622	2644	2694	2737	2778	2837	3021	3031	3103
		3399	3476	3488	3564	3630	3651	3664	3736	3755	3767	3831	4054	4186	4395
		1201	1248	1264	1281	1300	1324	1340	1357	1376	1406	1422	1439	1458	1483
	1497	1513	1529	1550	1563	1579	1596	1615	1639	1655	1672	1691	1715	1731	1748
	1767	1791	1807	1824	1843	1877	1910	1942	1972	1987	2026	2051	2071	2091	2117
	2140	2160	2180	2206	2230	2250	2270	2295	2318	2338	2358	2386	2419	2437	2446
	2463	2486	2520	2544	2567	2590	2627	2649	2673	2725	2805	2862	2882	2904	2922

	2943	2947	2950	2970	2987	3042	3060	3081	3109	3124	3150	3159	3200	3203	3224
	3232	3362	3366	3404	3497	3501	3535	3571	3575	3688	3717	3788	3822	3908	3950
\$CHK1	3987	3991	4008	4070	4155	4169	4201	4384	4387						
	1#	1138	1145	1148	1150	1153	1167	1180	1183	1184	1185	1186	1187	1189	1193
	1196	1220	1223	1224	1245	1259	1275	1292	1321	1335	1351	1368	1403	1417	1433
	1450	1486	1492	1508	1524	1553	1560	1574	1590	1607	1636	1650	1666	1682	1712
	1726	1742	1759	1788	1802	1818	1835	1875	1907	1940	1976	1984	2032	2044	2064
	2084	2121	2133	2153	2173	2211	2223	2243	2263	2299	2311	2331	2351	2391	2404
	2416	2431	2457	2491	2504	2517	2531	2555	2594	2611	2615	2638	2676	2681	2689
	2713	2720	2768	2772	2798	2826	2832	2859	2885	2892	2894	2900	2907	2929	2931
	2958	2984	3011	3018	3028	3038	3050	3058	3070	3077	3095	3100	3118	3153	3162
	3168	3170	3175	3178	3183	3186	3189	3192	3195	3197	3207	3252	3254	3265	3322
	3326	3332	3333	3334	3335	3336	3338	3365	3388	3418	3421	3422	3446	3447	3448
	3449	3450	3453	3455	3471	3493	3533	3539	3545	3598	3602	3605	3606	3635	3639
	3642	3661	3673	3694	3721	3740	3743	3746	3765	3777	3794	3825	3855	3858	3860
	3864	3866	3870	3873	3876	3880	3947	3995	3997	4044	4052	4142	4144	4146	4158
	4163	4173	4239	4247	4250	4262	4285	4287	4290	4295	4298	4390	4399	4403	4405
	4407	4409	4414	4428											
\$CKOP2	1#	1155	1158	1161	1164	1210	1212	1221	1261	1278	1295	1337	1354	1371	1419
	1436	1453	1494	1511	1526	1576	1593	1610	1652	1669	1686	1728	1745	1762	1804
	1821	1838	2048	2068	2088	2137	2157	2177	2227	2247	2267	2315	2335	2355	2407
	2435	2461	2508	2536	2559	2609	2619	2641	2765	2824	2890	2912	3172	3213	3215
	3260	3329	3341	3351	3391	3419	3458	3461	3485	3519	3522	3549	3556	3559	3561
	3592	3594	3603	3620	3681	3684	3704	3727	3781	3784	3886	3890	3893	3917	3919
	3940	3943	3954	3975	3978	3982	4002	4012	4040	4049	4191	4241	4255	4257	4328
	4416	4418	4420	4430											
\$CKR6	1#	3261	4242												
\$CMND	1#	1201	1248	1264	1281	1300	1324	1340	1357	1376	1406	1422	1439	1458	1483
	1497	1513	1529	1550	1563	1579	1596	1615	1639	1655	1672	1691	1715	1731	1748
	1767	1791	1807	1824	1843	1877	1910	1942	1972	1987	2026	2051	2071	2091	2117
	2140	2160	2180	2206	2230	2250	2270	2295	2318	2338	2358	2386	2419	2437	2446
	2463	2486	2520	2544	2567	2590	2627	2649	2673	2725	2805	2862	2882	2904	2922
	2943	2947	2950	2970	2987	3042	3060	3081	3109	3124	3150	3159	3200	3203	3224
	3232	3234	3362	3366	3404	3497	3501	3535	3571	3575	3688	3717	3788	3822	3908
	3950	3987	3991	4008	4070	4155	4169	4201	4384	4387					
\$COMPA	1#	1201	1248	1264	1281	1300	1324	1340	1357	1376	1406	1422	1439	1458	1483
	1497	1513	1529	1550	1563	1579	1596	1615	1639	1655	1672	1691	1715	1731	1748
	1767	1791	1807	1824	1843	1877	1910	1942	1972	1987	2026	2051	2071	2091	2117
	2140	2160	2180	2206	2230	2250	2270	2295	2318	2338	2358	2386	2419	2437	2446
	2463	2486	2520	2544	2567	2590	2627	2649	2673	2725	2805	2862	2882	2904	2922
	2862	2882	2904	2922	2943	2947	2950	2970	2987	3042	3060	3081	3109	3124	3150
	3159	3175	3200	3203	3224	3232	3362	3366	3404	3497	3501	3535	3571	3575	3639
	3688	3717	3743	3788	3822	3908	3950	3987	3991	4008	4062	4070	4155	4169	4201
	4247	4287	4295	4384	4387										
\$COUNT	1#	1141	2410	2511	2539	2562	2622	2644	2694	2737	2778	2837	3021	3031	3103
	3355	3399	3476	3488	3564	3630	3651	3664	3736	3755	3767	3831	4054	4186	4395
\$DO	1#	2943	3200												
\$ELSE	1#														
\$ERRMS	1#														
\$EXIFA	1#														
\$EXIFO	1#														
\$EXIF2	1#	4178													
\$EXIF3	1#														
\$GENBR	1#	1202	1216	1249	1265	1282	1301	1325	1341	1358	1377	1407	1423	1440	1459
	1484	1498	1514	1530	1551	1564	1580	1597	1616	1640	1656	1673	1692	1716	1732
	1749	1768	1792	1808	1825	1844	1878	1911	1943	1973	1988	2027	2052	2072	2092

	2118	2141	2161	2181	2207	2231	2251	2271	2296	2319	2339	2359	2387	2420	2438
	2447	2464	2487	2521	2545	2568	2591	2628	2650	2674	2704	2726	2745	2788	2806
	2846	2863	2883	2905	2909	2917	2919	2923	2944	2948	2951	2965	2971	2988	3043
	3061	3082	3110	3125	3151	3160	3176	3181	3201	3204	3209	3220	3225	3228	3233
	3235	3238	3257	3363	3367	3372	3405	3498	3502	3508	3536	3572	3576	3582	3640
	3645	3689	3699	3718	3744	3749	3789	3800	3823	3901	3903	3909	3951	3988	3992
	4009	4062	4071	4156	4160	4170	4179	4181	4193	4202	4204	4209	4248	4253	4259
	4264	4288	4293	4296	4301	4303	4306	4385	4388	4392	4411	4425	4432		
\$GENTA	1#	1191	1207	1254	1270	1287	1306	1330	1346	1363	1382	1412	1428	1445	1464
	1489	1503	1519	1535	1556	1569	1585	1602	1621	1645	1661	1678	1697	1721	1737
	1754	1773	1797	1813	1830	1849	1888	1921	1955	1979	2002	2035	2059	2079	2099
	2124	2148	2168	2188	2214	2238	2258	2278	2302	2326	2346	2366	2394	2425	2443
	2452	2469	2494	2526	2550	2573	2597	2633	2655	2679	2709	2731	2750	2793	2811
	2851	2868	2888	2902	2910	2914	2920	2934	2942	2961	2963	2966	2977	2993	3053
	3073	3091	3121	3131	3156	3165	3177	3179	3199	3210	3217	3221	3229	3239	3247
	3249	3258	3373	3379	3381	3410	3509	3514	3516	3542	3583	3588	3590	3641	3643
	3697	3700	3724	3745	3747	3797	3801	3828	3898	3904	3914	3956	3999	4004	4015
	4067	4076	4152	4161	4165	4175	4194	4206	4217	4219	4249	4251	4260	4267	4268
	4289	4291	4297	4299	4304	4307	4312	4313	4368	4369	4382	4393	4401	4412	4422
	4439	4440													
\$IF	1#	1201	1248	1264	1281	1300	1324	1340	1357	1376	1406	1422	1439	1458	1483
	1497	1513	1529	1550	1563	1579	1596	1615	1639	1655	1672	1691	1715	1731	1748
	1767	1791	1807	1824	1843	1877	1910	1942	1972	1987	2026	2051	2071	2091	2117
	2140	2160	2180	2206	2230	2250	2270	2295	2318	2338	2358	2386	2419	2437	2446
	2463	2486	2520	2544	2567	2590	2627	2649	2673	2725	2805	2862	2882	2904	2922
	2947	2950	2970	2987	3042	3060	3081	3109	3124	3150	3159	3203	3224	3232	3362
	3366	3404	3497	3501	3535	3571	3575	3688	3717	3788	3822	3908	3950	3987	3991
	4008	4070	4155	4169	4201	4384	4387								
\$IFCOD	1#	1201	1215	1248	1264	1281	1300	1324	1340	1357	1376	1406	1422	1439	1458
	1483	1497	1513	1529	1550	1563	1579	1596	1615	1639	1655	1672	1691	1715	1731
	1748	1767	1791	1807	1824	1843	1877	1910	1942	1972	1987	2026	2051	2071	2091
	2117	2140	2160	2180	2206	2230	2250	2270	2295	2318	2338	2358	2386	2419	2437
	2446	2463	2486	2520	2544	2567	2590	2627	2649	2673	2725	2805	2862	2882	2904
	2918	2922	2943	2947	2950	2970	2987	3042	3060	3081	3109	3124	3150	3159	3200
	3203	3224	3232	3234	3362	3366	3404	3497	3501	3535	3571	3575	3688	3717	3788
	3822	3902	3908	3950	3987	3991	4008	4070	4155	4169	4201	4384	4387	4424	
\$IFCON	1#	2704	2745	2788	2846	4062									
\$IFOPR	1#	1202	1216	1249	1265	1282	1301	1325	1341	1358	1377	1407	1423	1440	1459
	1484	1498	1514	1530	1551	1564	1580	1597	1616	1640	1656	1673	1692	1716	1732
	1749	1768	1792	1808	1825	1844	1878	1911	1943	1973	1988	2027	2052	2072	2092
	2118	2141	2161	2181	2207	2231	2251	2271	2296	2319	2339	2359	2387	2420	2438
	2447	2464	2487	2521	2545	2568	2591	2628	2650	2674	2704	2726	2745	2788	2806
	2846	2863	2883	2905	2909	2917	2919	2923	2944	2948	2951	2971	2988	3043	3110
	3125	3151	3160	3201	3204	3225	3233	3235	3363	3367	3405	3498	3502	3536	3572
	3576	3689	3718	3789	3823	3903	3909	3951	3988	3992	4009	4062	4071	4156	4170
	4202	4385	4388	4425											
\$LET	1#	1138	1145	1148	1150	1153	1155	1158	1161	1164	1167	1180	1183	1184	1185
	1186	1187	1189	1193	1196	1210	1212	1220	1221	1223	1224	1245	1259	1261	1275
	1278	1292	1295	1321	1335	1337	1351	1354	1368	1371	1403	1417	1419	1433	1436
	1450	1453	1486	1492	1494	1508	1511	1524	1526	1553	1560	1574	1576	1590	1593
	1607	1610	1636	1650	1652	1666	1669	1682	1686	1712	1726	1728	1742	1745	1759
	1762	1788	1802	1804	1818	1821	1835	1838	1875	1907	1940	1976	1984	2032	2044
	2048	2064	2068	2084	2088	2121	2133	2137	2153	2157	2173	2177	2211	2223	2227
	2243	2247	2263	2267	2299	2311	2315	2331	2335	2351	2355	2391	2404	2407	2416
	2431	2435	2457	2461	2491	2504	2508	2517	2531	2536	2555	2559	2594	2609	2611
	2615	2619	2638	2641	2676	2681	2689	2713	2720	2765	2768	2772	2798	2824	2826

MAINDEC-22-CVDVA-8
CVDVAB.P11MACY11 30(1046)
15-DEC-77 08:5819-DEC-77 08:25 PAGE 157
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0154

	2832	2859	2885	2890	2892	2894	2900	2907	2912	2929	2931	2958	2984	3011	3018
	3028	3038	3050	3058	3070	3077	3095	3100	3118	3153	3162	3168	3170	3172	3183
	3186	3189	3192	3195	3197	3207	3213	3215	3252	3254	3260	3265	3322	3326	3329
	3332	3333	3334	3335	3336	3338	3341	3351	3385	3388	3391	3418	3419	3421	3422
	3446	3447	3448	3449	3450	3453	3455	3458	3461	3471	3485	3493	3519	3522	3533
	3539	3545	3548	3556	3559	3561	3592	3594	3598	3602	3603	3605	3606	3620	3635
	3661	3673	3681	3684	3694	3704	3721	3727	3740	3765	3777	3781	3784	3794	3825
	3855	3858	3860	3864	3866	3870	3873	3876	3880	3886	3890	3893	3917	3919	3940
	3943	3947	3954	3975	3978	3982	3995	3997	4002	4012	4040	4044	4043	4052	4142
	4144	4146	4158	4163	4173	4191	4239	4241	4255	4257	4262	4285	4328	4390	4399
	4403	4405	4407	4409	4414	4416	4418	4420	4428	4430					
SLPCNT	1#	3175	3639	3743	4247	4287	4295								
SOPADD	1#	1156	1159	1162	1165	1210	1213	1222	2912	3178	3213	3215	3329	3420	3604
	3642	3746	3940	3975	4002	4250	4290	4298	4328	4418	4420	4430			
SOPAND	1#	3261	4242												
SOPCD1	1#	1156	1159	1162	1165	1210	1213	1222	1261	1278	1295	1337	1354	1371	1419
	1436	1453	1494	1511	1526	1576	1593	1610	1652	1669	1686	1728	1745	1762	1804
	1821	1838	2048	2068	2088	2137	2157	2177	2227	2247	2267	2315	2335	2355	2407
	2435	2461	2508	2536	2559	2609	2619	2641	2765	2824	2890	2912	3172	3178	3213
	3215	3261	3329	3341	3351	3391	3420	3458	3461	3485	3519	3522	3548	3556	3559
	3561	3592	3594	3604	3620	3642	3682	3684	3704	3727	3746	3782	3784	3886	3890
	3893	3917	3919	3940	3944	3954	3975	3979	3983	4002	4012	4040	4049	4191	4242
	4250	4255	4257	4262	4290	4298	4328	4416	4418	4420	4430				
SOPCD2	1#	4255	4262	4416											
SOPCOD	1#	1156	1159	1162	1165	1210	1213	1222	1261	1278	1295	1337	1354	1371	1419
	1436	1453	1494	1511	1526	1576	1593	1610	1652	1669	1686	1728	1745	1762	1804
	1821	1838	2048	2068	2088	2137	2157	2177	2227	2247	2267	2315	2335	2355	2407
	2435	2461	2508	2536	2559	2609	2619	2641	2765	2824	2890	2912	3172	3178	3213
	3215	3261	3329	3341	3351	3391	3420	3458	3461	3485	3519	3522	3548	3556	3559
	3561	3592	3594	3604	3620	3642	3682	3684	3704	3727	3746	3782	3784	3886	3890
	3893	3917	3919	3940	3944	3954	3975	3979	3983	4002	4012	4040	4049	4191	4242
	4250	4255	4257	4262	4290	4298	4328	4416	4418	4420	4430				
SOPCOM	1#	4262													
SOPDEF	1#	1138	1141	1145	1148	1150	1153	1155	1156	1158	1159	1161	1162	1164	1165
	1167	1180	1183	1184	1185	1186	1187	1189	1193	1196	1201	1202	1210	1212	1213
	1215	1216	1220	1221	1222	1223	1224	1245	1248	1249	1259	1261	1264	1265	1275
	1278	1281	1282	1292	1295	1300	1301	1321	1324	1325	1335	1337	1340	1341	1351
	1354	1357	1358	1368	1371	1376	1377	1403	1406	1407	1417	1419	1422	1423	1433
	1436	1439	1440	1450	1453	1458	1459	1483	1484	1486	1492	1494	1497	1498	1508
	1511	1513	1514	1524	1526	1529	1530	1550	1551	1553	1560	1563	1564	1574	1576
	1579	1580	1590	1593	1596	1597	1607	1610	1615	1616	1636	1639	1640	1650	1652
	1655	1656	1666	1669	1672	1673	1682	1686	1691	1692	1712	1715	1716	1726	1728
	1731	1732	1742	1745	1748	1749	1759	1762	1767	1768	1788	1791	1792	1802	1804
	1807	1808	1818	1821	1824	1825	1835	1838	1843	1844	1875	1877	1878	1907	1910
	1911	1940	1942	1943	1972	1973	1976	1984	1987	1988	2026	2027	2032	2044	2048
	2051	2052	2064	2068	2071	2072	2084	2088	2091	2092	2117	2118	2121	2133	2137
	2140	2141	2153	2157	2160	2173	2177	2180	2181	2206	2207	2211	2211	2223	2227
	2230	2231	2243	2247	2250	2251	2263	2267	2270	2271	2295	2296	2299	2311	2315
	2318	2319	2331	2335	2338	2339	2351	2355	2358	2359	2386	2387	2391	2404	2407
	2410	2411	2412	2413	2416	2419	2420	2431	2435	2437	2438	2446	2447	2457	2461
	2463	2464	2486	2487	2491	2504	2508	2511	2512	2513	2514	2517	2520	2521	2531
	2536	2539	2540	2541	2542	2544	2545	2555	2559	2562	2563	2564	2565	2567	2568
	2590	2591	2594	2609	2611	2615	2619	2622	2623	2624	2625	2627	2628	2638	2641
	2644	2645	2646	2647	2649	2650	2673	2674	2676	2681	2689	2694	2695	2696	2697
	2698	2699	2700	2704	2713	2720	2725	2726	2737	2738	2739	2740	2741	2742	2743
	2745	2765	2768	2772	2778	2779	2780	2781	2782	2783	2784	2788	2798	2805	2806

MAINDEC-ZZ-CVDVA-9
CVDVAB.P11MACY11 30(1046)
15-DEC-77 08:5819-DEC-77 08:25 PAGE 158
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0155

	2824	2826	2832	2837	2838	2839	2840	2841	2842	2843	2846	2859	2862	2863	2882
	2883	2885	2890	2892	2894	2900	2904	2905	2907	2909	2912	2916	2917	2918	2919
	2922	2923	2929	2931	2943	2944	2947	2948	2950	2951	2958	2965	2970	2971	2984
	2987	2988	3011	3018	3021	3022	3023	3024	3028	3031	3032	3033	3034	3038	3042
	3043	3050	3058	3060	3061	3070	3077	3081	3082	3095	3100	3103	3104	3105	3106
	3109	3110	3118	3124	3125	3150	3151	3153	3159	3160	3162	3168	3170	3172	3175
	3176	3178	3180	3181	3183	3186	3189	3193	3195	3197	3200	3201	3203	3204	3207
	3209	3213	3215	3220	3224	3225	3228	3232	3233	3234	3235	3238	3252	3254	3257
	3260	3261	3262	3263	3265	3322	3326	3332	3333	3334	3335	3338	3339	3339	3341
	3351	3355	3356	3357	3358	3362	3363	3366	3367	3372	3385	3388	3391	3399	3400
	3401	3402	3404	3405	3418	3419	3420	3421	3422	3446	3447	3448	3449	3450	3453
	3455	3458	3461	3471	3476	3477	3478	3479	3480	3481	3482	3485	3488	3489	3490
	3491	3493	3497	3498	3501	3502	3508	3519	3522	3533	3535	3536	3539	3545	3548
	3556	3559	3561	3564	3565	3566	3567	3571	3572	3575	3576	3582	3592	3594	3598
	3602	3603	3604	3605	3606	3620	3630	3631	3632	3635	3639	3640	3642	3644	3645
	3651	3652	3653	3654	3655	3656	3657	3661	3664	3665	3666	3667	3668	3669	3670
	3673	3681	3682	3684	3688	3689	3694	3699	3704	3717	3718	3721	3727	3736	3737
	3738	3740	3743	3744	3746	3748	3749	3755	3756	3757	3758	3759	3760	3761	3765
	3767	3768	3769	3770	3771	3772	3773	3777	3781	3782	3784	3788	3789	3794	3800
	3822	3823	3825	3831	3832	3833	3855	3858	3860	3864	3866	3870	3873	3876	3880
	3886	3890	3893	3900	3901	3902	3903	3908	3909	3917	3919	3940	3943	3944	3947
	3950	3951	3954	3975	3978	3979	3982	3983	3987	3988	3991	3992	3995	3997	4002
	4008	4009	4012	4040	4044	4049	4052	4054	4055	4056	4057	4058	4059	4060	4062
	4070	4071	4142	4144	4146	4155	4156	4158	4160	4163	4169	4170	4173	4178	4179
	4180	4181	4186	4187	4188	4189	4191	4193	4201	4202	4204	4208	4209	4218	4220
	4239	4241	4242	4243	4244	4247	4248	4250	4252	4253	4255	4257	4259	4262	4264
	4269	4285	4287	4288	4290	4292	4293	4295	4296	4298	4300	4301	4303	4306	4314
	4328	4370	4384	4385	4387	4388	4390	4392	4395	4399	4403	4405	4407	4409	4411
	4414	4416	4418	4420	4424	4425	4428	4430	4432	4441					
\$OPEQU	1#														
\$OPNAM	1#														
\$OPNEG	1#														
\$OPNOR	1#														
\$OPNOT	1#														
	2267	1278	1354	1436	1494	1526	1593	1669	1745	1821	2048	2088	2137	2177	2227
	3556	2315	2355	2407	2461	2508	2559	2609	2641	3341	3391	3461	3519	3522	3548
	4012	3592	3594	3682	3684	3704	3727	3782	3784	3917	3919	3944	3954	3979	3983
\$OPOR	1#														
	1838	1261	1295	1337	1371	1419	1453	1511	1576	1610	1652	1686	1728	1762	1804
	3559	2068	2157	2247	2335	2435	2536	2619	2765	2824	2890	3172	3351	3458	3485
\$OPROT	1#														
\$OPRO	1#														
	1196	4416													
	1450	1138	1145	1148	1150	1153	1167	1180	1183	1184	1185	1186	1187	1189	1193
	1726	1220	1223	1234	1245	1259	1275	1292	1321	1335	1351	1368	1403	1417	1433
	2084	1486	1492	1508	1524	1553	1560	1574	1590	1607	1636	1650	1666	1682	1712
	2416	1742	1759	1788	1802	1818	1835	1875	1907	1940	1976	1984	2032	2044	2064
	2713	2121	2133	2153	2173	2211	2223	2243	2263	2299	2311	2331	2351	2391	2404
	2958	2431	2457	2491	2504	2517	2531	2555	2594	2611	2615	2638	2676	2681	2689
	3168	2720	2768	2772	2798	2826	2832	2859	2885	2892	2894	2900	2907	2929	2931
	3332	2984	3011	3018	3028	3038	3050	3058	3070	3077	3095	3100	3118	3153	3162
	3450	3170	3175	3183	3186	3189	3192	3195	3197	3207	3252	3254	3265	3322	3326
	3673	3333	3334	3335	3336	3338	3385	3388	3418	3421	3422	3446	3447	3448	3449
	3873	3453	3455	3471	3493	3533	3539	3545	3598	3602	3605	3606	3635	3639	3661
	4247	3694	3721	3740	3743	3765	3777	3794	3825	3855	3858	3860	3864	3866	3870
\$OPR1	1#														
		3876	3880	3947	3995	3997	4044	4052	4142	4144	4146	4158	4163	4173	4239
		4285	4287	4295	4390	4399	4403	4405	4407	4409	4414	4428			
		3178	3642	3746	4250	4262	4290	4298							

\$OPR2	1#	1155	1158	1161	1164	1210	1212	1221	1261	1278	1295	1337	1354	1371	1419
	1436	1453	1494	1511	1526	1576	1593	1610	1652	1669	1686	1728	1745	1762	1804
	1821	1838	2048	2068	2088	2137	2157	2177	2227	2247	2267	2315	2335	2355	2407
	2435	2461	2508	2536	2559	2609	2619	2641	2765	2824	2890	2912	3172	3213	3215
	3260	3329	3341	3351	3391	3419	3458	3461	3485	3519	3522	3548	3556	3559	3561
	3592	3594	3603	3620	3681	3684	3704	3727	3781	3784	3886	3890	3893	3917	3919
	3940	3943	3954	3975	3978	3982	4002	4012	4040	4049	4191	4241	4255	4257	4328
	4416	4418	4420	4430											
\$OPSHF	1#	4255													
\$OPSUB	1#	4191													
\$OPSW8	1#														
\$OPXOR	1#														
\$OR	1#														
\$PUT	1#	2411	2512	2540	2563	2623	2645	2695	2738	2779	2838	3022	3032	3104	3356
	3400	3477	3489	3565	3652	3665	3756	3768	4055	4187					
\$STRUC	1#														
\$SUBON	1#	1207	1215	1254	1270	1287	1306	1330	1346	1363	1382	1412	1428	1445	1464
	1489	1503	1519	1535	1556	1569	1585	1602	1621	1645	1661	1678	1697	1721	1737
	1754	1773	1797	1813	1830	1849	1888	1921	1955	1979	2002	2035	2059	2079	2099
	2124	2148	2168	2188	2214	2238	2258	2278	2302	2326	2346	2366	2394	2425	2443
	2452	2469	2494	2526	2550	2573	2597	2633	2655	2679	2709	2731	2750	2793	2811
	2851	2868	2888	2910	2914	2916	2934	2961	2963	2965	2966	2977	2993	3053	3073
	3091	3121	3131	3156	3165	3179	3182	3210	3217	3220	3221	3229	3239	3247	3249
	3257	3258	3373	3379	3381	3410	3509	3514	3516	3542	3583	3588	3590	3643	3646
	3697	3699	3700	3724	3747	3750	3797	3800	3801	3828	3900	3914	3956	3999	4004
	4015	4067	4076	4161	4165	4175	4193	4194	4206	4217	4251	4254	4259	4260	4267
	4291	4294	4299	4302	4303	4304	4306	4307	4312	4368	4393	4401	4412	4422	4424
	4439														
\$THEN	1#	1201	1248	1264	1281	1300	1324	1340	1357	1376	1406	1422	1439	1458	1483
	1497	1513	1529	1550	1563	1579	1596	1615	1639	1655	1672	1691	1715	1731	1748
	1767	1791	1807	1824	1843	1877	1910	1942	1972	1987	2026	2051	2071	2091	2117
	2140	2160	2180	2206	2230	2250	2270	2295	2318	2338	2358	2386	2419	2437	2446
	2463	2486	2520	2544	2567	2590	2627	2649	2673	2725	2805	2862	2882	2904	2922
	2947	2950	2970	2987	3042	3060	3081	3109	3124	3150	3159	3203	3224	3234	3362
	3366	3404	3497	3501	3535	3571	3575	3688	3717	3788	3822	3908	3950	3987	3991
	4008	4070	4155	4169	4201	4384	4387								
\$TILA	1#														
\$TILO	1#														
\$UNTIL2	1#	2916	3900												
\$UNTIL3	1#														
\$WHILE	1#	2942	3199												
\$SCMRE	954#														
\$SCMTM	954#														
\$SDEFA	1#														
\$SENDS	1#														
\$SERRO	1#														
\$SESCA	795#														
\$SGEN	1#	1191	1207	1254	1270	1287	1306	1330	1346	1363	1382	1412	1428	1445	1464
	1489	1503	1519	1535	1556	1569	1585	1602	1621	1645	1661	1678	1697	1721	1737
	1754	1773	1797	1813	1830	1849	1888	1921	1955	1979	2002	2035	2059	2079	2099
	2124	2148	2168	2188	2214	2238	2258	2278	2302	2326	2346	2366	2394	2425	2443
	2452	2469	2494	2526	2550	2573	2597	2633	2655	2679	2709	2731	2750	2793	2811
	2851	2868	2888	2902	2910	2914	2920	2934	2942	2961	2963	2966	2977	2993	3053
	3073	3091	3121	3131	3156	3165	3177	3179	3199	3210	3217	3221	3229	3239	3247
	3249	3258	3373	3379	3381	3410	3509	3514	3516	3542	3583	3588	3590	3641	3643
	3697	3700	3724	3745	3747	3797	3801	3828	3898	3904	3914	3956	3999	4004	4015

	4067	4076	4117	4152	4161	4165	4175	4194	4206	4217	4219	4224	4243	4251	4260
	4267	4268	4273	4289	4291	4297	4299	4304	4307	4312	4313	4334	4368	4369	4372
	4382	4393	4401	4412	4422	4439	4440								
\$\$GETS	1#	1207	1215	1254	1270	1287	1306	1330	1346	1363	1382	1412	1428	1445	1464
	1489	1503	1519	1535	1556	1569	1585	1602	1621	1645	1661	1678	1697	1721	1737
	1754	1773	1797	1813	1830	1849	1888	1921	1955	1979	2002	2035	2059	2079	2099
	2124	2148	2168	2188	2214	2238	2258	2278	2302	2326	2346	2366	2394	2425	2443
	2452	2469	2494	2526	2550	2573	2597	2633	2655	2679	2709	2731	2750	2793	2811
	2851	2868	2888	2909	2910	2914	2916	2934	2961	2963	2965	2966	2977	2993	3053
	3073	3091	3121	3131	3156	3165	3179	3182	3209	3210	3217	3220	3221	3228	3229
	3238	3239	3247	3249	3257	3258	3372	3373	3379	3381	3410	3508	3509	3514	3516
	3542	3582	3583	3588	3590	3643	3646	3697	3699	3700	3724	3747	3750	3797	3800
	3801	3828	3900	3914	3956	3999	4004	4015	4067	4076	4160	4161	4165	4175	4179
	4193	4194	4206	4217	4251	4254	4259	4260	4267	4291	4294	4299	4302	4303	4304
	4306	4307	4312	4368	4392	4393	4401	4411	4412	4422	4424	4439			
\$\$GETT	1#	2909	3209	3228	3238	3372	3508	3582	4160	4178	4392	4411			
\$\$LPCN	1#	3178	3642	3746	4250	4290	4298								
\$\$NEWT	795#	1170	1234	1312	1389	1471	1542	1628	1704	1780	1863	1896	1929	1963	2016
	2107	2196	2216	2374	2476	2582	2662	2754	2815	2871	3003	3140	3309	3434	3611
\$\$POP	1#	1207	1215	1254	1270	1287	1306	1330	1346	1363	1382	1412	1428	1445	1464
	1489	1503	1519	1535	1556	1569	1585	1602	1621	1645	1661	1678	1697	1721	1737
	1754	1773	1797	1813	1830	1849	1888	1921	1955	1979	2002	2035	2059	2079	2099
	2124	2148	2168	2188	2214	2238	2258	2278	2302	2326	2346	2366	2394	2425	2443
	2452	2469	2494	2526	2550	2573	2597	2633	2655	2679	2709	2731	2750	2793	2811
	2851	2868	2888	2910	2914	2916	2934	2961	2963	2965	2966	2977	2993	3053	3073
	3091	3121	3131	3156	3165	3179	3182	3210	3217	3220	3221	3229	3239	3247	3249
	3257	3258	3273	3379	3381	3410	3509	3514	3516	3542	3583	3588	3590	3643	3646
	3697	3699	3700	3724	3747	3750	3797	3800	3801	3828	3900	3914	3956	3999	4004
	4015	4067	4076	4161	4165	4175	4193	4194	4206	4217	4251	4254	4259	4260	4267
	4291	4294	4299	4302	4303	4304	4306	4307	4312	4368	4393	4401	4412	4422	4424
\$\$PUSH	1#	1191	1192	1201	1203	1248	1250	1264	1266	1281	1283	1300	1302	1324	1326
	1340	1342	1357	1359	1376	1378	1406	1408	1422	1424	1439	1441	1458	1460	1483
	1485	1497	1499	1513	1515	1529	1531	1550	1552	1563	1565	1579	1581	1596	1598
	1615	1617	1639	1641	1655	1657	1672	1674	1691	1693	1715	1717	1731	1733	1748
	1750	1767	1769	1791	1793	1807	1809	1824	1826	1843	1845	1877	1879	1910	1912
	1942	1944	1972	1974	1987	1989	2026	2028	2051	2053	2071	2073	2091	2093	2117
	2119	2140	2142	2160	2162	2180	2182	2206	2208	2230	2232	2250	2252	2270	2272
	2295	2297	2318	2320	2338	2340	2358	2360	2386	2388	2419	2421	2437	2439	2446
	2448	2463	2465	2486	2488	2520	2522	2544	2546	2567	2569	2590	2592	2627	2629
	2649	2651	2673	2675	2704	2705	2725	2727	2745	2746	2788	2789	2805	2807	2846
	2847	2862	2864	2882	2884	2902	2903	2904	2906	2911	2922	2924	2942	2943	2945
	2947	2949	2950	2952	2965	2970	2972	2987	2989	3042	3044	3060	3062	3081	3083
	3109	3111	3124	3126	3150	3152	3159	3161	3175	3177	3178	3182	3199	3200	3202
	3203	3205	3211	3220	3224	3226	3230	3232	3236	3240	3362	3364	3366	3368	3374
	3404	3406	3497	3499	3501	3503	3510	3535	3537	3571	3573	3575	3577	3584	3639
	3641	3642	3646	3688	3690	3717	3719	3743	3745	3746	3750	3788	3790	3822	3824
	3898	3899	3908	3910	3950	3952	3987	3989	3991	3993	4008	4010	4062	4063	4070
	4072	4118	4152	4153	4155	4157	4162	4169	4171	4193	4201	4203	4225	4247	4249
	4250	4254	4274	4287	4289	4290	4294	4295	4297	4298	4302	4335	4373	4382	4383
	4384	4386	4387	4389	4394	4413									
\$\$SELE	1#														
\$\$SET	5088#	5097	5098	5099	5100	5102	5104	5105	5106						
\$\$SETM	1111#														
\$\$SETS	1#	1191	1192	1201	1203	1248	1250	1264	1266	1281	1283	1300	1302	1324	1326

	1340	1342	1357	1359	1376	1378	1406	1408	1422	1424	1439	1441	1458	1460	1483
	1485	1497	1499	1513	1515	1529	1531	1550	1552	1563	1565	1579	1581	1596	1598
	1615	1617	1639	1641	1655	1657	1672	1674	1691	1693	1715	1717	1731	1733	1748
	1750	1767	1769	1791	1793	1807	1809	1824	1826	1843	1845	1877	1879	1910	1912
	1942	1944	1972	1974	1987	1989	2026	2028	2051	2053	2071	2073	2091	2093	2117
	2119	2140	2142	2160	2162	2180	2182	2206	2208	2230	2232	2250	2252	2270	2272
	2295	2297	2318	2320	2338	2340	2358	2360	2386	2388	2419	2421	2437	2439	2446
	2448	2463	2465	2486	2488	2520	2522	2544	2546	2567	2569	2590	2592	2627	2629
	2649	2651	2673	2675	2704	2705	2723	2727	2745	2746	2788	2789	2805	2807	2846
	2847	2862	2864	2882	2894	2902	2903	2904	2906	2911	2922	2924	2942	2943	2945
	2947	2949	2950	2952	2965	2970	2972	2987	2989	3042	3044	3060	3062	3081	3093
	3109	3111	3124	3126	3150	3152	3159	3161	3175	3177	3178	3182	3199	3200	3202
	3203	3205	3211	3220	3224	3226	3230	3232	3236	3240	3262	3264	3266	3268	3274
	3404	3406	3497	3499	3501	3503	3510	3535	3537	3571	3573	3575	3577	3584	3639
	3641	3642	3646	3688	3690	3717	3719	3743	3745	3746	3750	3788	3790	3822	3824
	3898	3899	3908	3910	3950	3952	3987	3989	3991	3993	4008	4010	4062	4063	4070
	4072	4118	4152	4153	4155	4157	4162	4169	4171	4193	4201	4203	4225	4247	4249
	4250	4254	4274	4287	4289	4290	4294	4295	4297	4298	4302	4335	4373	4382	4383
	4384	4386	4387	4389	4394	4413									
\$\$SETT	1#														
\$\$SKIP	795#	1487	1554	1977	2033	2122	2212	2300	2392	2492	2576	2595	2677	2886	2932
	2959	2996	3051	3071	3089	3119	3134	3154	3163	3268	3540	3695	3722	3795	3826
	3922	4080													
.EQUAT	1#	685													
.HEADE	1#	663													
.SETUP	1#	1072													
.SWRHI	1#	673													
.SWRLO	684#														
.\$ACT1	1#	921													
.\$APT8	1#	998#													
.\$APTH	1#	932													
.\$APTY	1#	4746													
.\$CATC	1#	910													
.\$CHTA	1#	954													
.\$EOP	1#	4444													
.\$ERRO	1#	4803													
.\$POWE	1#	4483													
.\$READ	1#	4607													
.\$SCOP	1#	4857													
.\$STRAP	1#	5065													
.\$TYPD	1#	4921													
.\$TYPE	1#	4528													
.\$TYPO	1#	4988													

. ABS. 015316 000

ERRORS DETECTED: 0

 CVDVAB,CVDVAB.SEQ=CVDVAB.MAC,CVDVAB.P11
 RUN-TIME: 101 93 6 SECONDS
 RUN-TIME RATIO: 391/202=1.9
 CORE USED: 34K (67 PAGES)

