

M8268

PDP11/34 CACHE
CFKKAB0

AH-E111B-MC

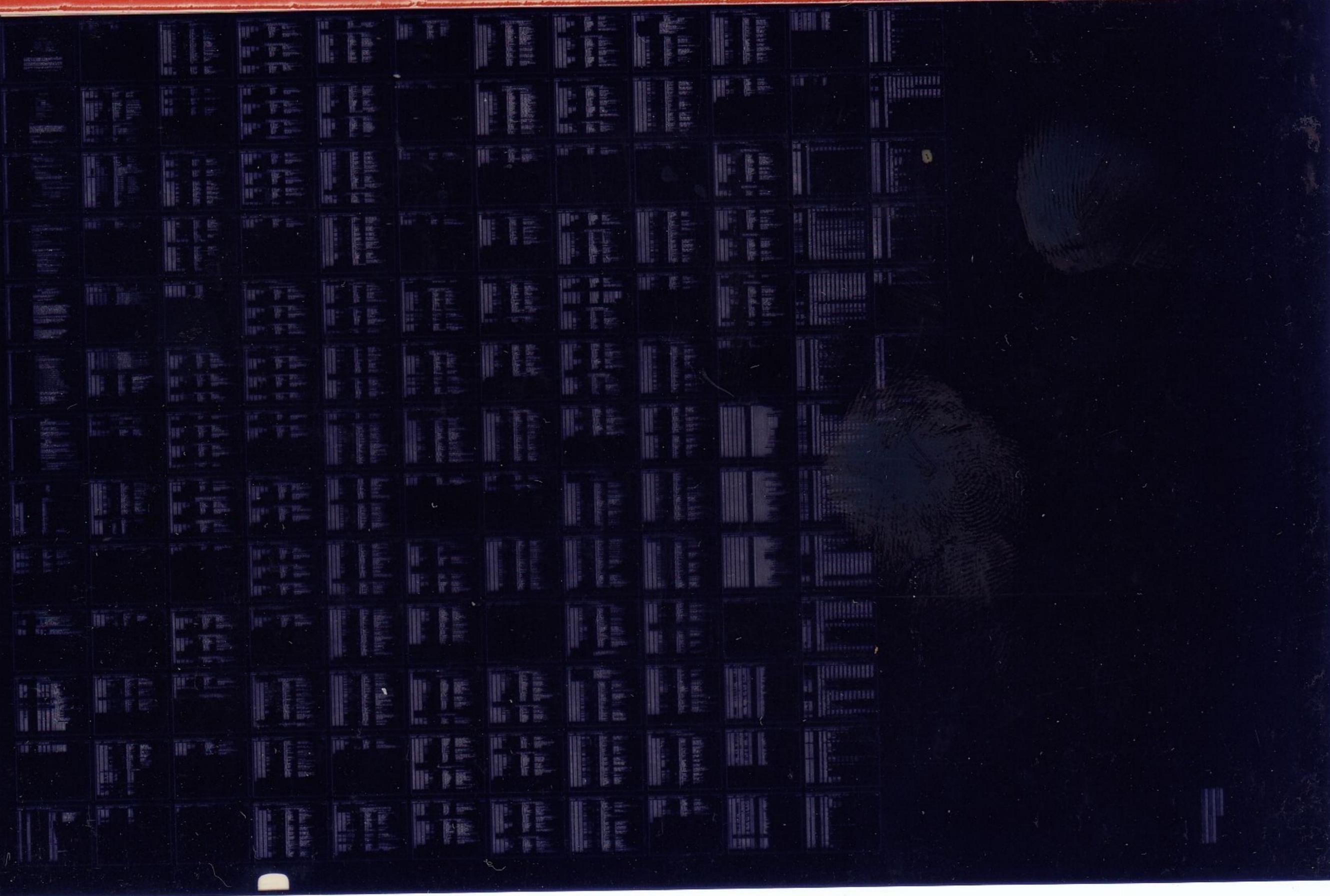
COPYRIGHT 78-80

FICHE 1 OF 1

JAN 1980

digital

MADE IN USA



5540
5541
5542
5543
5544
5545
5546
5547
5548
5549
5550
5551
5552
5553
5554
5555
5556
5557
5558
5559
5560
5561
5562
5563
5564
5565
5566
5567
5568
5569
5570
5571
5572
5573
5574
5575
5576
5577
5578
5579
5580
5581
5582
5583
5584
5585
5586
5587
5588
5589
5590
5591
5592
5593
5594
5595

.REM %

IDENTIFICATION

PRODUCT CODE:	AC-E110B-MC
PRODUCT NAME:	CFKKABO PDP11/34 CACHE
DATE CREATED:	MAY 1979
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	SCOTT GORDON REVISED BY JOHN W. CIUKAJ REV. DATE JAN 1979

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 MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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

COPYRIGHT (C) 1978, 1979 BY DIGITAL EQUIPMENT CORPORATION

5596
5597
5598
5599
5600
5601
5602
5603
5604
5605
5606
5607
5608
5609
5610
5611
5612
5613
5614
5615
5616
5617
5618
5619
5620
5621
5622
5623
5624
5625
5626
5627
5628
5629
5630
5631
5632
5633
5634
5635
5636
5637
5638
5639
5640
5641
5642
5643
5644
5645
5646
5647
5648
5649
5650
5651

TABLE OF CONTENTS

1.0 ABSTRACT

2.0 SYSTEM REQUIREMENTS

 2.1 HARDWARE

 2.2 SOFTWARE

 2.3 ACT & APT SETUP

 2.4 EXECUTION TIME

3.0 DIAGNOSTIC HIERARCHY PREREQUISITES

4.0 STARTING ADDRESS

5.0 PROGRAM CONTROL AND OPERATOR ACTION

6.0 PROGRAM DESCRIPTION

7.0 ERROR REPORTING

8.0 HANDLERS AND COMMON ROUTINES

9.0 REV B CHANGES

1.0 ABSTRACT

THE 11/34 CACHE DIAGNOSTIC IS COMPRISED OF A SERIES OF TESTS WHICH WERE DESIGNED TO CHECK THE CACHE'S DATA PATHS AND ITS CONTROL LOGIC. THE TESTS ARE ARRANGED IN A LOGICAL ORDER SUCH THAT THEY BUILD ON ONE ANOTHER. THAT IS, THE CURRENTLY RUNNING TEST WILL DEPEND ON LOGIC EXERCISED BY PREVIOUS TESTS. THOSE TESTS REQUIRING EXTENSIVE AMOUNTS OF CACHE FUNCTIONING ARE DONE NEAR THE END OF THE PROGRAM. THIS TESTING PROCEDURE SHOULD PROVIDE AN EFFECTIVE DEGREE OF FAULT ISOLATION.

2.0 SYSTEM REQUIREMENTS

2.1 HARDWARE

1. A WORKING 11/34 CPU
2. A MINIMUM OF 16K TO A MAX OF 124K OF MEMORY. 124K IS NEEDED FOR COMPLETE CHECK OF TAG MEMORY.

5652
5653
5654
5655
5656
5657
5658
5659
5660
5661
5662
5663
5664
5665
5666
5667
5668
5669
5670
5671
5672
5673
5674
5675
5676
5677
5678
5679
5680
5681
5682
5683
5684
5685
5686
5687
5688
5689
5690
5691
5692
5693
5694
5695
5696
5697
5698
5699
5700
5701
5702
5703
5704
5705
5706
5707

- 3. A CONSOLE TERMINAL
- 4. A UNIBUS EXERCISER IF NPR DATOS ARE TO BE TESTED.
HARDWARE SETTINGS: ADDRESS = 770000 (ALL SWITCHES ON)
VECTOR NOT USED

2.2 SOFTWARE

THIS DIAGNOSTIC WILL RUN UNDER ACT, XXDP AND STAND ALONE.
IT CAN ALSO BE RUN UNDER APT IN ACT MODE.

2.3 ACT& APT SETUP

2.3.1 RUN TIMES

FIRST PASS RUN TIME = 10
MAXIMUM PASS RUN TIME = 10

2.3.2 50HZ/60HZ SYSTEM CONFIGURATION

PROPER OPERATION OF FORCE MISS TESTS ARE DEPENDANT ON WHETHER LINE FREQU
IS 50HZ OR 60HZ. THE DIAGNOSTIC OPERATES IN THE
FOLLOWING MANNER UNDER THE SPECIFIED ENVIRONMENTS:

1. ACT,XXDP

A. QUICK VERIFY,AUTO ACCEPT,XXDP CHAIN

DIAGNOSTIC RUNS FORCE MISS TESTS FOR 60HZ CONFIGURATION ONLY.

B. DUMP MODE

1ST PASS OF DIAGNOSTIC FOLLOWING LOADING OF PROGRAM
WILL PROMPT USER FOR 60HZ OR 50HZ CONFIGURATION.

2. APT

A. QUICK VERIFY,RUN TIME ,STANDALONE

APT SCRIPT MUST USE BIT0 IN SWITCH 1(CONSOLE SWITCH
REGISTERS)

1= 50HZ CONFIGURATION
0= 60HZ CONFIGURATION

2.4 EXECUTION TIME

FOR AN ERROR FREE, FIRST RUN PASS ON A 11/34 WITH CORE MEMORY,
IT TAKES APPROXIMATELY 10 SECONDS.

5708
5709
5710
5711
5712
5713
5714
5715
5716
5717
5718
5719
5720
5721
5722
5723
5724
5725
5726
5727
5728
5729
5730
5731
5732
5733
5734
5735
5736
5737
5738
5739
5740
5741
5742
5743
5744
5745
5746
5747
5748
5749
5750
5751
5752
5753
5754
5755
5756
5757
5758
5759
5760
5761
5762
5763

3.0 DIAGNOSTIC HIERARCHY PREREQUISITES

IT IS ASSUMED THAT CPU, MEMORY, MEMORY MANAGEMENT AND TTY ARE WORKING PROPERLY FOR THIS PROGRAM TO GIVE CORRECT ERROR REPORTS. IF NOT, THEIR RESPECTIVE DIAGNOSTIC SHOULD BE RUN BEFORE THE CACHE DIAGNOSTIC.

4.0 STARTING ADDRESS

200 FOR NORMAL STARTUP

5.0 PROGRAM CONTROL AND OPERATOR ACTION

5.1 THE STANDARD DIAGNOSTIC LOADING PROCEDURES ARE TO BE FOLLOWED.

5.2 LOAD ADDRESS 200

5.3 START

5.4 DIAG. WILL THEN PRINT ITS NAME AND EXPECTED RUN TIME AFTER WHICH PROGRAM ENTERS COMMAND MODE AND PROMPTS USER WITH 'CACHE=>'.
5.5 THE USER THEN HAS THE OPTION OF USING THE FOLLOWING COMMANDS

5.5.1 'LOT' ENTER LOOP ON ERRORING TEST MODE
PROGRAM WILL LOOP ON ANY TEST IN WHICH AN ERROR HAS OCCURED

5.5.2 'CLOT' (DEFAULT) CANCELS EFFECT OF 'LOT'

5.5.3 'LOE' LOOP ON ERROR
PROGRAM WILL LOOP ON CURRENT ERROR

5.5.4 'CLOE' (DEFAULT) CANCELS EFFECT OF 'LOE'

5.5.5 'HOE' HALT ON ERROR
PROGRAM WILL PRINT TEST NUMBER OF FAILING TEST THEN HALT. TEST MAY THEN BE REENTERED BY USING THE CONSOLE CONTINUE SWITCH.

5.5.6 'CHOE' (DEFAULT) CANCELS EFFECT OF 'HOE'

5.5.7 'CIER' (DEFAULT) ENABLE ERROR PRINTOUT
ENABLES THE PRINTING OF ERROR MESSAGES.

5.5.8 'IER' DISABLE PRINTING ERROR MESSAGES

5764
5765
5766
5767
5768
5769
5770
5771
5772
5773
5774
5775
5776
5777
5778
5779
5780
5781
5782
5783
5784
5785
5786
5787
5788
5789
5790
5791
5792
5793
5794
5795
5796
5797
5798
5799
5800
5801
5802
5803
5804
5805
5806
5807
5808
5809
5810
5811
5812
5813
5814
5815
5816
5817
5818
5819

5.5.9 'LST XXX' LOOP ON SELECTED TEST
WHERE XXX = TEST TO BE LOOPED ON
EXAMPLE: LST 121
WILL EXECUTE ALL TESTS BEFORE TEST 121
THEN HOLD AT TEST 121 IN A LOOP

5.5.10 'CLST' (DEFAULT) CANCELS EFFECT OF 'LST XXX'

5.6 AFTER DESIRED OPTIONS HAVE BEEN SELECTED THE
PROGRAM MAY THEN BE RUN BY TYPING 'RUN'
AFTER WHICH TESTING WILL BEGIN.

5.7 TYPING '^C' AT ANY TIME WILL STOP TESTING
AND RETURN TO COMMAND MODE.

6.0 PROGRAM DESCRIPTION

UPON START OF THE PROGRAM, THE CACHE IS IMMEDIATELY TURNED OFF
(FORCE MISS IS ON FOR BOTH HALVES OF CACHE, INTERRUPTS ARE DISABLED
AND CACHE IS IN BYPASS MODE). THE TESTS THEN PROCEED TO SELECTIVELY
TURN ON ONLY THE HALF OF CACHE THAT IS TO BE EXERCISED.
THIS IS TO ENSURE THAT THE INSTRUCTIONS ARE NOT EXECUTED OUT
OF A POSSIBLY BAD CACHE. IN ORDER TO IMPLEMENT THIS SCHEME,
THE TESTS THAT ENABLE CACHE ARE RELOCATED TO AREAS OF CACHE
THAT ARE NOT ENABLED. THE TESTS ARE STRUCTURED ON A HALF CACHE
BASIS. THAT IS A TEST MAY BE RUN IN LOW CACHE WHILE TESTING
HIGH CACHE AFTER WHICH AN IDENTICAL TEST WILL RUN IN HIGH CACHE
WHILE TESTING LOW CACHE.

TO FACILITATE THE TESTING OF CACHE, A 2K BUFFER IS RESERVED AT THE
END OF THE PROGRAM FOR READ WRITE OPERATIONS AND RELOCATION OF TESTS.

IMMEDIATELY AFTER THE PROGRAM IS STARTED THE PROGRAM
IDENTIFIES ITSELF AND THEN PROMPTS USER TO ENTER COMMANDS
THAT WILL SET THE CONDITIONS FOR TESTING (SEE SEC. 5.4).
THIS IS ONLY DONE ON PROGRAM START AND NOT REPEATED
FOR SUBSEQUENT PROGRAM LOOPS.

7.0 ERROR REPORTING

THE CONTENTS OF THE ERROR REPORTS IDENTIFIES THE HARDWARE
UNDER TEST AT THE TIME OF FAILURE. OTHER PERTINENT INFORMATION
SUCH AS CONTENTS OF CACHE CONTROL FIELDS AND FAILING ADDRESS
, GOOD DATA, BAD DATA ARE ALSO REPORTED. EACH ERROR REPORT
THAT USES THE (ADDRESS, GOOD DATA, BAD DATA) FORMAT
WILL BE PRECEDED WITH AN EXPLANATION OF WHO'S ADDRESS
AND WHAT DATA IS BEING REPORTED.

IF THE PROGRAM SHOULD HANG OR HALT WITHOUT
PRINTING AN ERROR MESSAGE THE NUMBER OF THE LAST
TEST EXECUTED CAN BE FOUND AT \$TESTN: LOCATION 612 .

5820
5821
5822
5823
5824
5825
5826
5827
5828
5829
5830
5831
5832
5833
5834
5835
5836
5837
5838
5839
5840
5841
5842
5843
5844
5845
5846
5847
5848
5849
5850
5851
5852
5853
5854
5855
5856
5857
5858
5859
5860
5861
5862
5863
5864
5865
5866
5867
5868
5869
5870
5871
5872
5873
5874
5875

8.0 HANDLERS AND COMMON ROUTINES

8.1 THE FOLLOWING SECTION OFFERS EXPLANATION OF THE
UTILITY ROUTINES USED BY THE PROGRAM. THESE ROUTINES ARE
LOCATED ON THE FIRST 16 PAGES OF THE LISTING

- 8.1.1 'START:'' PREPARES PROGRAM FOR EXECUTION
- 8.1.2 'PREPARE:'' PREPARES 11/34 AND CACHE FOR
EXECUTION OF TESTS
- 8.1.3 'AHALT:'' HALT ON ERROR HANDLER
PRINTS HALT ON ERROR MESSAGE, THEN HALTS.
- 8.1.3 'PNTNAM:'' PRINT PROGRAM TITLE
- 8.1.4 'LP1:'' LOOP ON TEST COMMAND HANDLER
- 8.1.5 'LP2:'' NO LOOP ON TEST COMMAND HANDLER
- 8.1.6 'LP3:'' LOOP ON ERROR COMMAND HANDLER
- 8.1.7 'LP4:'' NO LOOP ON ERROR COMMAND HANDLER
- 8.1.8 'HL1:'' HALT ON ERROR COMMAND HANDLER
- 8.1.9 'HL2:'' NO HALT ON ERROR COMMAND HANDLER
- 8.1.10 'DIS1:'' DISABLE ERROR PRINTOUT COMMAND HANDLER
- 8.1.11 'DIS2:'' ENABLE ERROR PRINTOUT COMMAND HANDLER
- 8.1.12 'LP5:'' LOOP ON SPECIFIED TEST COMMAND HANDLER
- 8.1.13 'LP6:'' DISABLE LOOP ON SPECIFIED TEST COMMAND HANDLER
- 8.1.14 'PTID:'' USED TO PRINT TEST I.D. ON CURRENT TEST RUNNING
- 8.1.15 'TSTFLG:'' LOOKS FOR KEYBOARD INPUT
- 8.1.16 'GETCHA:'' INPUTS CHAR. FROM KEYBOARD, PREFORMS
LOWER TO UPPER CASE CONVERSION AND CHECKS FOR SPECIAL
CONDITIONS SUCH AS RUBOUT AND CARRIAGE RETURN. THEN
ENTERS CHAR. INTO INPUT BUFFER.
- 8.1.17 'ECHO:'' ECHO'S CHAR. JUST INPUT FROM KEYBOARD
- 8.1.18 'TRAP:'' TIMEOUT TRAP HANDLER
SETS TRAP FLAG AND RETURNS
- 8.1.19 'PARITY:'' PARITY TRAP HANDLER
DISABLES CACHE, SAVES CONTENTS OF ERROR REGISTER,
CLEARS ERROR REGISTER AND RETURNS.
- 8.1.20 'ERTSHI:'' ERROR LOOP HANDLER
HANDLER IS CALLED BY

5876
5877
5878
5879
5880
5881
5882
5883
5884
5885
5886
5887
5888
5889
5890
5891
5892
5893
5894
5895
5896
5897
5898
5899
5900
5901
5902
5903
5904
5905
5906
5907
5908
5909
5910
5911
5912
5913
5914
5915
5916
5917
5918
5919
5920
5921
5922
5923
5924
5925
5926
5927
5928
5929
5930
5931

JSR R0,(R0)
.WORD 1
.WORD 2

WHERE WORD1 IS THE RETURN ADDRESS IF LOOP ON ERROR
ENABLED AND AN ERROR OCCURED.
WORD 2 IS THE RETURN ADDRESS IF LOOP ON TEST WAS ENABLED.
IF NO ERROR, RETURN IS MADE TO INSTRUCTION FOLLOWING
WORD 2.

8.1.21 'LPONTS:'' LOOP ON TEST HANDLER
CHECKS LOOP ON TEST MODE FLAG
IF IN LOOP MODE (LOPERR) WILL BE CHECKED TO SEE IF TEST HAD ANY FAILURE
IF IT DID TEST WILL BE RESTARTED.

8.1.22 'DECODE:'' PROGRAM COMMAND DECODER
READS COMMAND FROM INPUT BUFFER AND COMPAIRS IT
TO COMMAND LIST. IF FOUND IN LIST THE SELECTED COMMAND
HANDLER WILL BE ENTERED . IF NOT FOUND THE MESSAGE
'ILLEGAL COMMAND!'' WILL BE PRINTED.

8.1.23 'SETEN:'' PRINT SELECTED ERROR MESSAGE
CALLED BY
JSR R0,SETEN
.WORD ^D4
.WORD SEN001
.WORD SEN002

WHERE THE FIRST WORD AFTER THE CALL IS USED AS AN OFFSET
FOR RETURN.
THE SECOND WORD IS THE NUMBER OF THE FIRST SENTENCE TO BE PRINTED
THE THIRD WORD IS THE SECOND SENTENCE TO BE PRINTED
THERE ARE NO LIMITS ON THE NUMBER OF SENTENCES THAT CAN
BE PRINTED BY THIS CALL.

8.1.24 'RELCTH:'' AND 'RELCTL:'' TEST RELOCATION HANDLERS
THESE HANDLERS ARE USED TO RELOCATE TESTS TO A
HIGH OR LOW CACHE AREA LOCATED AT THE END OF PROGRAM.
AFTER RELOCATION IS COMPLETE A JUMP WILL BE MADE TO THE RELOCATED
AREA AND TESTING WILL BEGIN.

8.1.25 'GDBD:'' GOOD BAD DATA PRINTER
PRINTS THE OCTAL CONTENTS OF LOCATIONS (ADD),(GOOD) AND (BAD).

8.1.26 'BITNAM:'' BIT NAME PRINTER
PRINTS THE NAME OF ANY BIT LEFT SET IN LOCATION (ERROR).

9.0 REV B CHANGES

REASON: 1.ALLOW FORCE MISS TESTS TO OPERATE UNDER VARIOUS CONFIGURATIONS
OF LINE FREQUENCY/MAIN MEMORY SPEED

2. ALLOW CNTRLC TO ABORT ERROR PRINTOUTS AND RETURN
TO SOFTWARE MONITOR.

FOR EACH OF THE FOLLOWING CODING BLOCKS SPECIFIED, CHECK APPROPRIATE
COMMENTS FOR INDICATION OF CHANGES.

5932
5933
5934
5935
5936
5937
5938
5939
5940
5941
5942
5943
5944
5945
5946
5947
5948
5949
5950
5952 000000
5953 000000
5954 000000 000002
5955 000002 000000
5956 000004 000006
5957 000006 000000
5958 000010 000012
5959 000012 000000
5960 000014 000016
5961 000016 000000
5962 000020 000022
5963 000022 000000
5964 000024 000200
5965 000026 000000
5966 000030 000032
5967 000032 000000
5968 000042
5969 000042 000000
5970 000044 000626
5971 000046
5972 000046 035662
5973 000052 000052
5974 000052 000000
5975 000114 000114
5976 000114 004650
5977 000116 000340
5978 000176 000176
5979 000176 000000
5980 000200 000200
5981 000200 000167 000274
5982 000500 000500
5983 000500 012767 001015 177240
5984 000506 000005
5985 000510 012706 060000
5986 000514 000240
5987 000516 000240
5988

%

CONWRD:

START:

START:
PREPARE:
LINFREQ:
CMDLST:
CMDLS1:
SETEN:
DIC131:
FORCE MISS TESTS

.ENABLE ABS
.TITLE CACHE DIAG.
.LIST ME
.NLIST TTM,BEX,MD,MC,CND
.ASECT
.=0
.WORD 2
.WORD 0
.WORD 6
.WORD 0
.WORD 12
.WORD 0
.WORD 16
.WORD 0
.WORD 22
.WORD 0
.WORD 200
.WORD 0
.WORD 32
.WORD 0
.=42
.WORD 0
.WORD \$APTHDR
.=46
.WORD ENDIT
.=52
.WORD 0
.=114
.WORD PARITY
.WORD 340
.=176
.WORD 0
.=200
JMP START
.=500
MOV #OFF,CCR
RESET
MOV #60000,SP
NOP
NOP

:DISABLE CACHE
:DISABLE ALL INTERRUPTS
:SET STACK POINTER

5989							:REVB		
5990	000520	056767	000140	177450			BIS \$\$WREG,CONWRD		;SET ANY BITS FROM APT LOADING
5991									
5992	000526	105767	000130				TSTB \$ENV		;IS THIS APT MODE
5993	000532	001003					BNE 3\$;YES
5994	000534	005737	000042				TST @#42		;IS THIS DUMP MODE
5995	000540	001407					BEQ 2\$;YES
5996	000542	052767	100000	177426		3\$:	BIS #BIT15,CONWRD		;SET HALT ON ERROR MODE
5997	000550	004767	000610				JSR PC,PNTNAM		;PRINT TITLE
5998	000554	000167	000220				JMP PREPARE		;START TEST
5999									
6000							:REV B		
6001									
6002	000560	005227	177777			2\$:	INC #-1		;PROMPT LINE FREQ. ON FIRST PASS ONLY
6003	000564	001010					BNE TITLE		
6004	000566	004767	000404				JSR PC,LINFRQ		;PRINT LINE FREQ. PROMPT MESSAGE
6005	000572	012767	004430	003624			MOV #INBUF,BUFPNT		;GET ANSWER
6006	000600	004767	003322			4\$:	JSR PC,TSTFLG		
6007	000604	000775					BR 4\$		
6008									
6009							:REV B		
6010									
6011	000606	004767	000552		TITLE:		JSR PC,PNTNAM		;PRINT TITLE
6012	000612	012767	004430	003604			MOV #INBUF,BUFPNT		
6013	000620	004767	003302			1\$:	JSR PC,TSTFLG		
6014	000624	000775					BR 1\$		

```

6016
(1)
(2)
(1)
(2)
(1) 000626
(1) 000024 000024
(1) 000024 000200
(1) 000044 000044
(1) 000044 000626
(1) 000626
(2)
(1)
(1)
(1)
(1) 000626
(1) 000626 000000
(1) 000630 000642
(1) 000632 000000
(1) 000634 000000
(1) 000636 000000
(1) 000640 000052

```

.SBTTL APT PARAMETER BLOCK

```

*****
:SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
*****
.$X=      ;;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.
$APTHDR  ;;POINT TO APT HEADER BLOCK
.=.$X    ;;RESET LOCATION COUNTER
*****
:SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
:INTERFACE SPEC.

```

```

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

```

6018
 (1)
 (2)
 (1)
 (1) 000642
 (1) 000642 000000
 (1) 000644 000000
 (1) 000646 000000
 (1) 000650 000000
 (1) 000652 000000
 (1) 000654 000000
 (1) 000656 000000
 (1) 000660 000000
 (1) 000662
 (1) 000662 000
 (1) 000663 000
 (1) 000664 000000
 (1) 000666 000000
 (1) 000670 000000
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1) 000672 000
 (1) 000673 000
 (1)
 (1)
 (1)
 (1) 000674 000000
 (1)
 (1) 000676 000
 (1) 000677 000
 (1) 000700 000000
 (1) 000702 000
 (1) 000703 000
 (1) 000704 000000
 (1) 000706 000
 (1) 000707 000
 (1) 000710 000000
 (1) 000712 000000
 (1) 000714 000000
 (1) 000716 000000
 (1) 000720 000000
 (1) 000722 000000
 (1) 000724 000000
 (1) 000726 000000
 (1) 000730 000000
 (1) 000732 000000
 (1) 000734 000000
 (1) 000736 000000
 (1) 000740 000000
 (1) 000742 000000
 (1) 000744 000000

.SBTTL APT MAILBOX-ETABLE

```

*****
.EVEN
$MAIL:                ::APT MAILBOX
$MSGTY: .WORD  AMSGTY  ::MESSAGE TYPE CODE
$FATAL: .WORD  AFATAL  ::FATAL ERROR NUMBER
$TESTN: .WORD  ATESTN  ::TEST NUMBER
$PASS:  .WORD  APASS   ::PASS COUNT
$DEVCT: .WORD  ADEVCT  ::DEVICE COUNT
$UNIT:  .WORD  AUNIT   ::I/O UNIT NUMBER
$MSGAD: .WORD  AMSGAD  ::MESSAGE ADDRESS
$MSGLG: .WORD  AMSGLG  ::MESSAGE LENGTH
$ETABLE:              ::APT ENVIRONMENT TABLE
$ENV:    .BYTE  AENV    ::ENVIRONMENT BYTE
$ENVM:   .BYTE  AENVM   ::ENVIRONMENT MODE BITS
$SWREG:  .WORD  ASWREG  ::APT SWITCH REGISTER
$USWR:   .WORD  AUSWR   ::USER SWITCHES
$CPUOP:  .WORD  ACPUOP  ::CPU TYPE,OPTIONS
BITS 15-11=CPU TYPE
          11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
          11/70=06,PDQ=07,Q=10
BIT 10=REAL TIME CLOCK
BIT 9=FLOATING POINT PROCESSOR
BIT 8=MEMORY MANAGEMENT
$MAMS1: .BYTE  AMAMS1  ::HIGH ADDRESS,M.S. BYTE
$MTYP1: .BYTE  AMTYP1  ::MEM. TYPE,BLK#1
MEM.TYPE BYTE -- (HIGH BYTE)
          900 NSEC CORE=001
          300 NSEC BIPOLAR=002
          500 NSEC MOS=003
$MADR1: .WORD  AMADR1  ::HIGH ADDRESS,BLK#1
MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
$MAMS2: .BYTE  AMAMS2  ::HIGH ADDRESS,M.S. BYTE
$MTYP2: .BYTE  AMTYP2  ::MEM. TYPE,BLK#2
$MADR2: .WORD  AMADR2  ::MEM.LAST ADDRESS,BLK#2
$MAMS3: .BYTE  AMAMS3  ::HIGH ADDRESS,M.S.BYTE
$MTYP3: .BYTE  AMTYP3  ::MEM. TYPE,BLK#3
$MADR3: .WORD  AMADR3  ::MEM.LAST ADDRESS,BLK#3
$MAMS4: .BYTE  AMAMS4  ::HIGH ADDRESS,M.S.BYTE
$MTYP4: .BYTE  AMTYP4  ::MEM. TYPE,BLK#4
$MADR4: .WORD  AMADR4  ::MEM.LAST ADDRESS,BLK#4
$VECT1: .WORD  AVECT1  ::INTERRUPT VECTOR#1,BUS PRIORITY#1
$VECT2: .WORD  AVECT2  ::INTERRUPT VECTOR#2BUS PRIORITY#2
$BASE:  .WORD  ABASE   ::BASE ADDRESS OF EQUIPMENT UNDER TEST
$DEVN:  .WORD  ADEVN   ::DEVICE MAP
$CDW1:  .WORD  ACDW1   ::CONTROLLER DESCRIPTION WORD#1
$CDW2:  .WORD  ACDW2   ::CONTROLLER DESCRIPTION WORD#2
$DDW0:  .WORD  ADDW0   ::DEVICE DESCRIPTOR WORD#0
$DDW1:  .WORD  ADDW1   ::DEVICE DESCRIPTOR WORD#1
$DDW2:  .WORD  ADDW2   ::DEVICE DESCRIPTOR WORD#2
$DDW3:  .WORD  ADDW3   ::DEVICE DESCRIPTOR WORD#3
$DDW4:  .WORD  ADDW4   ::DEVICE DESCRIPTOR WORD#4
$DDW5:  .WORD  ADDW5   ::DEVICE DESCRIPTOR WORD#5
$DDW6:  .WORD  ADDW6   ::DEVICE DESCRIPTOR WORD#6
$DDW7:  .WORD  ADDW7   ::DEVICE DESCRIPTOR WORD#7

```

(1)	000746	000000	\$DDW8:	.WORD	ADDW8	::DEVICE	DESCRIPTOR	WORD#8
(1)	000750	000000	\$DDW9:	.WORD	ADDW9	::DEVICE	DESCRIPTOR	WORD#9
(1)	000752	000000	\$DDW10:	.WORD	ADDW10	::DEVICE	DESCRIPTOR	WORD#10
(1)	000754	000000	\$DDW11:	.WORD	ADDW11	::DEVICE	DESCRIPTOR	WORD#11
(1)	000756	000000	\$DDW12:	.WORD	ADDW12	::DEVICE	DESCRIPTOR	WORD#12
(1)	000760	000000	\$DDW13:	.WORD	ADDW13	::DEVICE	DESCRIPTOR	WORD#13
(1)	000762	000000	\$DDW14:	.WORD	ADDW14	::DEVICE	DESCRIPTOR	WORD#14
(1)	000764	000000	\$DDW15:	.WORD	ADDW15	::DEVICE	DESCRIPTOR	WORD#15
(1)								
(1)								
(1)	000766		SETEND:					
(1)								

.SBTTL REGISTER DEFINITIONS

6020	
6021	
6022	
6023	177744
6024	177746
6025	177750
6026	177752
6027	177560
6028	177562
6029	177564
6030	177566
6031	177776
6032	000000
6033	000001
6034	000002
6035	000003
6036	000004
6037	000005
6038	000006
6039	000007
6040	000001
6041	000002
6042	000004
6043	000010
6044	000020
6045	000040
6046	000100
6047	000200
6048	000400
6049	001000
6050	002000
6051	004000
6052	010000
6053	020000
6054	040000
6055	100000
6056	177546
6057	001015
6058	172300
6059	172302
6060	172304
6061	172306
6062	172310
6063	172312
6064	172314
6065	172316
6066	172340
6067	172342
6068	172344
6069	172346
6070	172350
6071	172352
6072	172354
6073	172356
6074	177572
6075	170002

CMPE = 177744
CCR = 177746
CMR = 177750
CHR = 177752
KRS = 177560
KRB = 177562
PPS = 177564
PPB = 177566
PSW = 177776
R0 = %0
R1 = %1
R2 = %2
R3 = %3
R4 = %4
R5 = %5
SP = %6
PC = %7
BIT00 = 1
BIT01 = 2
BIT02 = 4
BIT03 = 10
BIT04 = 20
BIT05 = 40
BIT06 = 100
BIT07 = 200
BIT08 = 400
BIT09 = 1000
BIT10 = 2000
BIT11 = 4000
BIT12 = 10000
BIT13 = 20000
BIT14 = 40000
BIT15 = 100000
KOOK00 = 177546
OFF = 1015
KPDR0 = 172300
KPDR1 = 172302
KPDR2 = 172304
KPDR3 = 172306
KPDR4 = 172310
KPDR5 = 172312
KPDR6 = 172314
KPDR7 = 172316
KPAR0 = 172340
KPAR1 = 172342
KPAR2 = 172344
KPAR3 = 172346
KPAR4 = 172350
KPAR5 = 172352
KPAR6 = 172354
KPAR7 = 172356
SRO = 177572
BECC = 170002

:CACHE MEMORY PARITY ERROR REGISTER
:CACHE CONTROL REGISTER
:CACHE MAINTENANCE REGISTER
:CACHE HIT REGISTER
:KEYBOARD READER STATUS
:KEYBOARD READER BUFFER
:PRINTER PUNCH STATUS
:PRINTER PUNCH BUFFER
:PROCESSOR STATUS WORD
:GENERAL REGISTERS

CACHE DIAG.
CFKAB.P11

MACY11 30A(1052)
25-JUN-79 13:31

31-OCT-79 15:29 PAGE 59-13
REGISTER DEFINITIONS

8 2

SEQ 0014

6076	170004
6077	170000
6078	170006
6079	170016
6080	000646

BEBA =	170004
BEDA =	170000
BECA =	170006
BECA2 =	170016
TID =	STESTN

```

6083      001000 001000
6084 001000 012701 044000      PREPARE:      .=1000
6085 001004 012102
6086 001006 020127 050000
6087 001012 001374
6088 001014 012706 060000
6089 001020 005067 003662
6090 001024 005067 177616
6091
6092
6093
6094 001030 056767 177630 177140
6095
6096 001036 012737 004650 000114
6097 001044 012737 000340 000116
6098 001052 012767 001415 176666
6099 001060 032767 040000 177110
6100 001066 001403
6101 001070 012767 000004 003610
6102 001076 032767 001000 177072
6103 001104 001403
6104 001106 012767 000002 003572
6105 001114 000167 005576
6106 001120 010246
6107 001122 012702 001154
6108 001126 004767 004732
6109 001132 004767 002652
6110 001136 012702 005634
6111 001142 004767 004716
6112 001146 012602
6113 001150 000000
6114 001152 000207
6115 001154 040510 052114 047440  HALONER:
6116
6117
6118
6119
6120
6121 001176 012702 001210
6122 001202 004767 004656
6123 001206 000207
6124 001210 015 012 177
6125 001215 111 020123 044514
6126 001263 015 012 177
6127 001270 044450 020106 047516
6128 001323 015 012 177
6129 001330 054524 042520 023440
6130 001352 040 040 040
6131
6132
6133
6134
6135 001364 105767 177272
6136 001370 001411
6137 001372 000207
6138 001374 005737 000042

```

```

MOV #LOWSP,R1      ;LOW CACHE ADDRESS
MOV (R1)+,R2      ;TAG ALL CACHE
CMP R1,#LOW1      ;TAGGING COMPLETE
BNE 5$            ;NO
MOV #60000,SP     ;SET UP STACK
CLR ACTION        ;ERROR ACTION
CLR TID           ;RESET TEST ID

;REV B
BIS $$WREG,CONWRD ;SET BITS LOADED FROM APT
MOV #PARITY,@#114 ;LOAD PARITY TRAP HANDLER
MOV #340,@#116
MOV #OFF+BIT08,CCR ;DISABLE AND FLUSH CACHE
BIT #BIT14,CONWRD ;IS THIS LOOP ON TEST MODE ?
BEQ 1$           ;NO
MOV #4,ACTION    ;SET ACTION TO LOOP ON TEST
BIT #BIT09,CONWRD ;IS THIS LOOP ON ERROR MODE?
BEQ 4$           ;NO
MOV #2,ACTION    ;SET ACTION TO LOOP ON TEST
JMP TST001      ;START TESTING
MOV R2,-(SP)    ;FREE R2 FOR USE
MOV #HLONER,R2  ;PRINT HALT ON ERROR MESSAGE
JSR PC,TYPE
JSR PC,PTID
MOV #CRLF,R2
JSR PC,TYPE
MOV (SP)+,R2
HALT
RTS PC
.ASCIZ /HALT ON ERROR AT /
.EVEN

;LINE FREQUENCY DETERMINATION(REV B)
MOV #QUESHZ,R2
JSR PC,TYPE
RTS PC
.BYTE 15,12,177,177,177
.ASCII *IS LINE FREQUENCY OF THIS SYSTEM 60HZ?*
.BYTE 15,12,177,177,177
.ASCII *(IF NO, ASSUMPTION IS 50HZ)*
.BYTE 15,12,177,177,177
.ASCII *TYPE 'Y' OR 'N' : *
.BYTE 40,40,40,40,40,40,177,177,177,0
.EVEN

PRINT TITLE
PNTNAM: TSTB $ENV ;IS THIS APT MODE
        BEG 1$   ;NO
        RTS PC  ;YES
        TST @#42 ;IS THIS DUMP MODE

```

6139 001400 001405
6140 001402 023737 000042 000046
6141 001410 001001
6142 001412 000207
6143 001414 005767 177230
6144 001420 001405
6145 001422 012702 001730
6146 001426 004767 004432
6147 001432 000207
6148 001434 012702 001446
6149 001440 004767 004420
6150 001444 000207
6151 001446 043103 045513 041101 NAME:
6152 001510 015 012 177
6153 001515 105 050130 041505
6154 001560 015 012 177
6155 001565 105 052116 051105
6156 001644 015 012 177
6157 001651 106 046117 047514
6158 001723 015 012 177
6159 001730 040503 044103 036505 PROM:
6160 001737 040 177 177
6161
6162 001744 052767 040000 176224 LP1:
6163 001752 000207
6164 001754 042767 040000 176214 LP2:
6165 001762 000207
6166 001764 052767 001000 176204 LP3:
6167 001772 000207
6168 001774 042767 001000 176174 LP4:
6169 002002 000207
6170 002004 052767 100000 176164 HL1:
6171 002012 000207
6172 002014 042767 100000 176154 HL2:
6173 002022 000207
6174 002024 052767 020000 176144 DIS1:
6175 002032 000207
6176 002034 042767 020000 176134 DIS2:
6177 002042 000207
6178 002044 005005 LP5:
6179 002046 121127 000060
6180 002052 103403
6181 002054 121127 000070
6182 002060 103402
6183 002062 005201
6184 002064 000770
6185 002066 006305
6186 002070 006305
6187 002072 006305
6188 002074 111102
6189 002076 042702 177700
6190 002102 162702 000060
6191 002106 060205
6192 002110 005201
6193 002112 121127 000060
6194 002116 103403

```

BEQ 1$           ;YES
CMP @#42,@#46   ;IS THIS ACT MODE
BNE 1$           ;YES
RTS PC           ;RETURN
1$: TST $PASS
BEQ 3$
MOV #PROM,R2    ;PRINT PROMPT
JSR PC,TYPE
RTS PC
3$: MOV #NAME,R2 ;ADDRESS OF LINE TO PRINT
JSR PC,TYPE     ;PRINT IT
RTS PC         ;RETURN
.ASCII *CFKKABO PDP 11/34 CACHE DIAGNOSTIC*
.BYTE 15,12,177,177,177
.ASCII *EXPECTED RUN TIME APROX. 10 SECONDS*
.BYTE 15,12,177,177,177
.ASCII *ENTER 'H' FOR HELP OR 'RUN' TO START DIAGNOSTIC*
.BYTE 15,12,177,177,177
.ASCII *FOLLOW ALL COMMANDS WITH A CARRIAGE RETURN*
.BYTE 15,12,177,177,177
.PROM: .ASCII *CACHE=>*
.BYTE 40,177,177,177,0
.EVEN
LP1: BIS #BIT14,CONWRD ;LOOP ON TEST
RTS PC
LP2: BIC #BIT14,CONWRD ;NO LOOP ON TEST
RTS PC
LP3: BIS #BIT09,CONWRD ;LOOP ON ERROR
RTS PC
LP4: BIC #BIT09,CONWRD ;NO LOOP ON ERROR
RTS PC
HL1: BIS #BIT15,CONWRD ;HALT ON ERROR
RTS PC
HL2: BIC #BIT15,CONWRD ;NO HALT ON ERROR
RTS PC
DIS1: BIS #BIT13,CONWRD ;DISABLE ERROR PRINTOUT
RTS PC
DIS2: BIC #BIT13,CONWRD ;ENABLE ERROR PRINTOUT
RTS PC
LP5: CLR R5 ;LOOP ON THIS TEST
1$: CMPB (R1),#0 ;MUST BE DIGIT
BLO 2$
CMPB (R1),#8
BLO 3$
2$: INC R1 ;NEXT CHAR.
BR 1$
3$: ASL R5 ;POSITION TO ADD DIGIT
ASL R5
ASL R5
4$: MOVB (R1),R2 ;ADD DIGIT
BIC #177700,R2
SUB #60,R2
ADD R2,R5
INC R1 ;NEXT DIGIT
CMPB (R1),#0
BLO 6$

```

CACHE DIAG.
CFKKAB.P11

MACY11 30A(1052)
25-JUN-79 13:31

31-OCT-79 15:29 PAGE 59-16
REGISTER DEFINITIONS

SEQ 0017

6195	002120	121127	000070				CMPB (R1),#8	
6196	002124	103760					BLO 3\$	
6197	002126	052767	000400	176042		6\$:	BIS #BIT08,CONWRD	;SET LOOP ON TEST MODE
6198	002134	110567	176036				MOVB R5,CONWRD	
6199	002140	000207					RTS PC	
6200	002142	042767	000777	176026	LP6:		BIC #BIT08+377,CONWRD	;EXIT LOOP ON TEST MODE
6201	002150	000207					RTS PC	
6202	002152	012702	002210		HELP:		MOV #LOST,R2	;PRINT HELP MESSAGE

6204	002156	004767	003702			JSR PC,TYPE	
6205	002162	000207				RTS PC	
6206	002164	052767	000001	176004	FRQ50:	BIS #1,CONWRD	;SPECIFY 50HZ
6207	002172	000167	176410			JMP TITLE	
6208	002176	042767	000001	175772	FRQ60:	BIC #1,CONWRD	;SPECIFY 60HZ
6209	002204	000167	176376			JMP TITLE	
6210							
6211	002210	047105	042524	020122	LOST:	.ASCII /ENTER ANY OF THE FOLLOWING COMMANDS AFTER THE 'CACHE=>' PROMPT./	
6212	002307	015	012	177		.BYTE 15,12,177,177,177	
6213	002314	044042	042517	020042		.ASCII *'HOE' = HALT ON ERROR (SW/SWR BIT 15)*	
6214	002361	015	012	177		.BYTE 15,12,177,177,177	
6215	002366	046042	052117	020042		.ASCII *'LOT' = LOOP ON FAILING TEST (SW/SWR BIT 14)*	
6216	002442	015	012	177		.BYTE 15,12,177,177,177	
6217	002447	042	042511	021122		.ASCII *'IER' = INHIBIT ERROR TYPEOUTS (SW/SWR BIT 13)*	
6218	002525	015	012	177		.BYTE 15,12,177,177,177	
6219	002532	046042	042517	020042		.ASCII *'LOE' LOOP ON ERROR (SW/SWR BIT 9)*	
6220	002574	015	012	177		.BYTE 15,12,177,177,177	
6221	002601	042	051514	021124		.ASCII *'LST' LOOP ON TEST XXX (SW/SWR BIT 8)*	
6222	002646	015	012	177		.BYTE 15,12,177,177,177	
6223	002653	124	042510	041440		.ASCII *THE COMMAND MUST BE FOLLOWED BY A CARRIDGE RETURN. THE PROGRAM WILL*	

```

6225 002756 015 012 177 .BYTE 15,12,177,177,177
6226 002763 124 042510 020116 .ASCII *THEN RESPOND ANOTHER PROMPT.THE USER MAY ENETER ANOTHER COMMAND*
6227 003062 015 012 177 .BYTE 15,12,177,177,177
6228 003067 117 020122 051047 .ASCII /OR 'RUN' TO START DIAGNOSTIC./
6229 003124 015 012 177 .BYTE 15,12,177,177,177
6230 003131 101 054516 041440 .ASCII ANY COMMAND MAY BE CANCELLED BY TYPING 'C' PLUS THE COMMAND*
6231 003224 015 012 177 .BYTE 15,12,177,177,177
6232 003231 050 054105 020056 .ASCII *(EX. CHOE REMOVES HALTING ON ERROR).*
6233 003275 015 012 177 .BYTE 15,12,177,177,177
6234 003302 054524 044520 043516 .ASCII *TYPING ^C AT ANY TIME WILL STOP TESTING AND RETURN TO COMMAND MODE.*
6235 003405 015 012 177 .BYTE 15,12,177,177,177
6236 003412 040527 047122 047111 .ASCII *WARNING !!! THE HARDWARE SWITCHES ON THE CACHE MODULE MUST BE*
6237 003507 015 012 177 .BYTE 15,12,177,177,177
6238 003514 047111 052040 042510 .ASCII *IN THE ON POSITION I.E. BOTH SWITCHES FACING TOWARDS THE CONSOLE*
6239 003614 015 012 177 .BYTE 15,12,177,177,177
6240 003621 116 052117 035105 .ASCII *NOTE: EACH OF THE ABOVE COMMANDS SETS THE DESIGNATED BIT IN THE*
6241 003720 015 012 177 .BYTE 15,12,177,177,177
6242 003725 123 043117 053524 .ASCII *SOFTWARE SWITCH REGISTER LOCATED AT 000176*
6243 003777 015 012 177 .BYTE 15,12,177,177,177,0,0,0
6244 004010 .EVEN
6245
6246
6247
6248
6249

```

PRINT THE CURRENT TID

```

6250 004010 010246 PTID: MOV R2,-(SP) ;SAVE REGISTERS R2,R3
6251 004012 010346 MOV R3,-(SP)
6252 004014 016702 174626 MOV TID,R2 ;PLACE CURRENT TEST I.D. FOR DISMEMBERING
6253 004020 012703 000057 MOV #57,R3 ;SET UP TO GENERATE THE HUNDREDS DIGIT
6254 004024 005203 1$: INC R3 ;R3=R3+1 FOR EVERY HUNDRED FOUND
6255 004026 162702 000100 SUB #100,R2 ;DIVISION BY SUBTRACTION
6256 004032 100374 BPL 1$
6257 004034 110367 000055 MOV R3,STID+3 ;ADD DIGIT TO STRING
6258 004040 062702 000100 ADD #100,R2 ;CORRECT THE REMAINDER OF NUMBER
6259 004044 012703 000057 MOV #57,R3 ;FIND TENS ASCII DIGIT
6260 004050 005203 2$: INC R3
6261 004052 162702 000010 SUB #10,R2 ;FIND NUMBER OF TENS IN NUMBER
6262 004056 100374 BPL 2$
6263 004060 110367 000032 MOV R3,STID+4 ;ADD DIGIT TO NUMBER
6264 004064 062702 000070 ADD #70,R2 ;CORRECT NUMBER FOR ONES DIGIT
6265 004070 110267 000023 MOV R2,STID+5 ;ADD DIGIT TO STRING
6266 004074 012702 004112 MOV #STID,R2 ;CORRECT DONES DIGIT
6267 004100 004767 001760 JSR PC,TYPE
6268 004104 012603 MOV (SP)+,R3 ;RESTORE R3
6269 004106 012602 MOV (SP)+,R2 ;RESTORE R2
6270 004110 000207 RTS PC
6271 004112 051524 020124 020040 STID: .ASCII /TST /
6272 004120 015 012 001 .BYTE 15,12,1,1,0
6273 004126 .EVEN

```

```

6275                                     .SBTTL KEYBOARD HANDLER
6276                                     : TEST TO SEE IF THE THE KEYBOARD WANTS US FOR ANYTHING
6277                                     :-----
6278                                     :
6279 004126 105767 173426               TSTFLG:          TSTB KRS          ;ANY CHARS. AVAIL
6280 004132 100002                       BPL 1$          ;NO
6281 004134 004767 000002               JSR PC,GETCHA  ;INPUT CHAR.
6282 004140 000207                       1$:          RTS PC
6283                                     :
6284                                     : INPUT CHAR. AND TEST FOR SPECIAL CONDITIONS
6285                                     :-----
6286                                     :
6287                                     :
6288 004142 010246                       GETCHA:          MOV R2,-(SP)          ;FREE R2 FOR USE
6289 004144 116777 173412 000252         MOVB KRB,@BUFNT ;STORE CHAR.
6290 004152 142777 000200 000244         BICB #200,@BUFNT ;NO PARITY ALLOWED
6291 004160 122777 000003 000236         CMPB #3,@BUFNT  ;LOOK FOR ^C
6292 004166 001002                       BNE 10$         ;NOT THIS TIME
6293 004170 000167 174304                       JMP START      ;RESTART PROGRAM
6294 004174 127727 000224 000177         10$:          CMPB @BUFNT,#177  ;IS THIS A RUBOUT,WELL IS IT!!
    
```

6296	004202	001023			BNE 1\$;NO ,NOT AT ALL
6297	004204	005767	000216		TST SLASH		;IS IT THE FIRST RUBOUT
6298	004210	001007			BNE 2\$;NO
6299	004212	112767	000057	000206	MOVB #'/,SLASH		;ITS THE FIRST SO PRINT SLASH
6300	004220	012702	004426		MOV #SLASH,R2		
6301	004224	004767	001634		JSR PC,TYPE		
6302	004230	022767	004430	000166	2\$: CMP #INBUF,BUFPNT		;YOU CAN'T DELETE WHAT YOU DIDN'T TYPE
6303	004236	001445			BEQ 5\$		
6304	004240	005367	000160		DEC BUFPNT		;ADJUST BUFFER POINTER
6305	004244	004767	000136		JSR PC,ECHO		;ECHO DELETED CHAR.
6306	004250	000440			BR 5\$		
6307	004252	005767	000150		1\$: TST SLASH		;WAS LAST CHAR. RUBOUT
6308	004256	001406			BEQ 3\$;NO
6309	004260	012702	004426		MOV #SLASH,R2		;YUP SO PRINT OTHER SLASH
6310	004264	004767	001574		JSR PC,TYPE		
6311	004270	005067	000132		CLR SLASH		;RESET FLAG
6312	004274	127727	000124	000015	3\$: CMPB @BUFPNT,#15		; <CR> OR <LF>
6313	004302	001425			BEQ 6\$;TERMINATE STRING
6314	004304	127727	000114	000012	CMPB @BUFPNT,#12		
6315	004312	001421			BEQ 6\$		
6316	004314	122777	000140	000102	CMPB #140,@BUFPNT		; CONVERT TO UPPER CASE
6317	004322	101003			BHI 4\$		
6318	004324	142777	000040	000072	4\$: BICB #40,@BUFPNT		;CORRECT FOR THAT
6319	004332	004767	000050		JSR PC,ECHO		;ECHO CHAR.
6320	004336	026727	000062	004537	CMP BUFPNT,#INBUF+^D71		;IS BUFFER FULL
6321	004344	103002			BHIS 5\$;YES
6322	004346	005267	000052		INC BUFPNT		;POINT TO NEXT
6323	004352	012602			5\$: MOV (SP)+,R2		;RESTORE R2
6324	004354	000207			RTS PC		;RETURN
6325	004356	105077	000042		6\$: CLRB @BUFPNT		;MARK END OF STRING
6326	004362	012767	004430	000034	MOV #INBUF,BUFPNT		;RESET BUFFER POINTER
6327	004370	012702	005634		MOV #CRLF,R2		;ECHO <CR>,<LF>
6328	004374	004767	001464		JSR PC,TYPE		
6329	004400	004767	000462		JSR PC,DECODE		;DECODE COMMAND
6330	004404	000762			BR 5\$		
6331							
6332							
6333							
6334							
6335							
6336	004406	105767	173152		ECHO: TSTB PPS		;IS PRINTER READY
6337	004412	100375			BPL ECHO		;NO
6338	004414	117767	000004	173144	MOVB @BUFPNT,PPB		;YES SO PRINT IT
6339	004422	000207			RTS PC		
6340	004424	000000			BUF PNT: .WORD 0		
6341	004426	000000			SLASH: .WORD 0		
6342	004430	000044			INBUF: .BLKW ^D36		
6343	004540	000000			ITTRAP: .WORD 0		
6344	004542	000000			ERROR: .WORD 0		
6345	004544	000000			ERR1: .WORD 0		
6346	004546	000000			SAVTAT: .WORD 0		
6347	004550	000000			GOOD: .WORD 0		
6348	004552	000000			BAD: .WORD 0		
6349	004554	000000			ADD: .WORD 0		
6350	004556	000000			GOODBD: .WORD 0		
6351	004560	000000			BITFLG: .WORD 0		

CACHE DIAG.
CFKKAB.P11

MACY11 30A(1052)
25-JUN-79 13:31

31-OCT-79 15:29 PAGE 62-1
KEYBOARD HANDLER

J 2

SEQ 0022

6352 004562 047125 041111 051525 NOUBE:
6353 004634 015 012 177
6354

.ASCII /UNIBUS EXERCISER PRESENT AT ADDRESS 770000/
.BYTE 15,12,177,177,177,0
.EVEN

CACHE DIAG.
CFKKAB.P11

MACY11 30A(1052)
25-JUN-79 13:31

31-OCT-79 15:29 PAGE 62-2
TRAP HANDELERS

SEQ 0023

6356						.SBTTL TRAP HANDELERS	
6357	004642	005267	177672		TRAP:	INC ITTRAP	;IND. TRAP OCCURED
6358	004646	000002				RTI	
6359	004650	012767	001015	173070	PARITY:	MOV #OFF,CCR	;DISABLE CACHE
6360	004656	056767	173062	177660		BIS CMPE,ERR1	;UPDATE ERROR INFO
6361	004664	012767	000000	173052		MOV #0,CMPE	;RESET PARITY ERROR
6362	004672	000002				RTI	;INSTANT RETURN
6363	004674	005267	000002		CLOCK:	INC TIME	;BUMP COUNTER
6364	004700	000002				RTI	;RETURN
6365	004702	000000			TIME:	.WORD 0	
6366	004704	000000			COUNT:	.WORD 0	
6367	004706	000000			ACTION:	.WORD 0	
6368	004710	000000			LAST:	.WORD 0	
6369	004712	000000			NOINC:	.WORD 0	

```

6371          .SBTTL HIGH CACHE ERROR LOOP HANDLER
6372 004714 056767 177622 000142 ERTSHI:  BIS ERROR,LOPERR      ;SET LOOP ON TEST FLAG
6373 004722 105767 173734          TSTB $ENV             ;IS THIS APT MODE
6374 004726 001002          BNE 5$                ;YES
6375 004730 004767 177172          JSR PC,TSTFLG        ;LOOK FOR KEYBOARD INPUT
6376 004734 062700 000004          ADD #4,R0           ;NORMAL RETURN IF NO ERROR
6377 004740 032767 000400 173230 5$:  BIT #BIT08,CONWRD   ;IS THIS LOOP ON TEST MODE
6378 004746 001413          BEQ 1$              ;NO
6379 004750 126767 173222 173670  CMPB CONWRD,TID    ;IS THIS THE TEST TO LOOP ON
6380 004756 001007          BNE 1$              ;YES
6381 004760 005740          TST -(R0)          ;YES
6382 004762 000403          BR 4$              ;SET LOOP FLAG IF LOOPING
6383 004764 005767 177716          3$:  TST ACTION        ;SET LOOP FLAG IF LOOPING
6384 004770 001401          BEQ 2$              ;SET RETURN ADDRESS
6385 004772 011000          4$:  MOV (R0),R0     ;SET RETURN ADDRESS
6386 004774 000200          2$:  RTS R0         ;RETURN
6387 004776 005767 177540          1$:  TST ERROR       ;ANY ERRORS
6388 005002 001774          BEQ 2$              ;NO
6389 005004 166700 177676          SUB ACTION,R0      ;TAKE SELECTED ACTION
6390 005010 000765          BR 3$

6391
6392
6393
6394 005012 032767 000400 173156 LPONTS:  BIT #BIT08,CONWRD   ;IS THIS LOOP ON TEST MODE
6395 005020 001410          BEQ 1$              ;NO
6396 005022 126767 173150 173616  CMPB CONWRD,TID    ;IS THIS TEST TO LOOP ON
6397 005030 001004          BNE 1$              ;NO
6398 005032 011000          2$:  MOV (R0),R0     ;SET RETURN ADDRESS
6399 005034 005067 000024          CLR LOPERR        ;RESET LOOP FLAG
6400 005040 000200          RTS R0             ;RETURN
6401 005042 026727 177640 000004          1$:  CMP ACTION,#4  ;IS THIS LOOP ON ERRORING TEST MODE
6402 005050 001003          BNE 3$              ;NO
6403 005052 005767 000006          TST LOPERR        ;DID TEST FAIL
6404 005056 001365          BNE 2$              ;YES, RESTART TEST
6405 005060 005720          3$:  TST (R0)+      ;RETURN ADDRESS NO LOOP
6406 005062 000200          RTS R0
6407 005064 000000          LOPERR:          .WORD 0
    
```


CACHE DIAG.
CFKKAB.P11

MACY11 30A(1052)
25-JUN-79 13:31

31-OCT-79 15:29 PAGE 62-5
COMMAND DECODER

SEQ 0026

6465 005324 001764
 6466 005326 001774
 6467 005330 002004
 6468 005332 002014
 6469 005334 002024
 6470 005336 002034
 6471 005340 002044
 6472 005342 002142
 6473 005344 002152
 6474
 6475
 6476
 6477 005346 002164
 6478 005350 002176
 6479
 6480
 6481

.WORD LP3
 .WORD LP4
 .WORD HL1
 .WORD HL2
 .WORD DIS1
 .WORD DIS2
 .WORD LP5
 .WORD LP6
 .WORD HELP

;REV B

.WORD FRQ50
 .WORD FRQ60

;REV B

```

        .SBTTL ERROR MESSAGE HANDLERS
6483          SETEN:      MOV (R0)+,LAST      ;FIND RETURN ADDRESS
6484 005352 012067 177332      ADD R0,LAST
6485 005356 060067 177326
6486
6487 005362 004767 176540      JSR PC,TSTFLG      ;REV B
6488
6489 005366 105767 173270      TSTB $ENV          ;IS THIS APT MODE
6490 005372 001410          BEQ 11$            ;NO
6491 005374 005767 177142      TST ERROR          ;ANY ERROR
6492 005400 001411          BEQ 1$             ;NO
6493 005402 005267 173234      INC $MSGTY         ;SET APT ERROR FLAG
6494 005406 004767 173506      JSR PC,AHALT      ;HALT
6495 005412 000404
6496 005414 032767 020000 172554 11$: BIT #BIT13,CONWRD ; ERROR PRINT MODE
6497 005422 001407          BEQ 4$             ;NO
6498 005424 016700 177260      1$: MOV LAST,R0    ;SO RETURN
6499 005430 005067 177122      CLR GOODBD        ;RESET DATA PRINT FLAG
6500 005434 005067 177120      CLR BITFLG        ;RESET BIT PRINT FLAG
6501 005440 000200
6502 005442 005767 177074      4$: TST ERROR      ;ANY ERRORS
6503 005446 001766          BEQ 1$             ;NO SO RETURN
6504 005450 004767 176334      JSR PC,PTID       ;PRINT CURRENT TEST ID
6505 005454 010146          MOV R1,-(SP)      ;FREE R1 FOR USE
6506 005456 012001          3$: MOV (R0)+,R1   ;SENTENCE POINTER
6507 005460 004767 000100      JSR PC,PRINT      ;PRINT SENTENCE
6508 005464 020067 177220      CMP R0,LAST       ;ALL SENTENCES PRINTED YET
6509 005470 001372          BNE 3$            ;NO
6510 005472 005767 177060      TST GOODBD        ;IS THIS DATA PRINT MODE
6511 005476 001402          BEQ 5$            ;NO
6512 005500 004767 000464      JSR PC,GDBD       ;ENTER DATA PRINTER
6513 005504 005767 177050      5$: TST BITFLG    ;IS THIS BIT PRINT MODE
6514 005510 001402          BEQ 6$            ;NO
6515 005512 004767 000736      JSR PC,BITNAM     ;ENTER BIT PRINTER
6516 005516 010246          6$: MOV R2,-(SP)   ;FREE R2 FOR USE
6517 005520 012702 005552      MOV #LFLF,R2     ;ADD LINE FEEDS TO END OF MESSAGE
6518 005524 004767 000334      JSR PC,TYPE
6519 005530 032767 100000 172440 10$: BIT #BIT15,CONWRD ;IS THIS HALT ON ERROR MODE
6520 005536 001402          BEQ 10$          ;NO
6521 005540 004767 173354      JSR PC,AHALT     ;YES !! SO HALT
6522 005544 012602          MOV (SP)+,R2    ;RESTORE R2
6523 005546 012601          MOV (SP)+,R1    ;RESTORE R1
6524 005550 000725          BR 1$           ;EXIT
6525 005552      012      012 LFLF: .BYTE 12,12,12,1,1,1,1,1,0
6526      005564          .EVEN
6527
6528
6529          PRINT:
6530 005564 010346          MOV R3,-(SP)    ;FREE R3 FOR USE
6531 005566 012703 005657      MOV #LINE,R3   ;STRING POINTER
6532 005572 010246          MOV R2,-(SP)   ;SAVE R2
6533 005574 012102          2$: MOV (R1)+,R2  ;WORD POINTER
6534 005576 001013          BNE 1$         ;0=RETURN
6535 005600 012702 005634      MOV #CRLF,R2   ;PRINT CARIDGE RETURN,LINE FEED AND FILL
6536 005604 004767 000032      JSR PC,ADWRD   ;ADD WORD TO STRING
6537 005610 012702 005657      MOV #LINE,R2   ;ADDRESS OF LINE TO PRINT
6538 005614 004767 000244      JSR PC,TYPE    ;PRINT STRING
    
```

6539	005620	012602				MOV (SP)+,R2	;RESTORE R2
6540	005622	012603				MOV (SP)+,R3	;RESTORE R3
6541	005624	000207				RTS PC	
6542	005626	004767	000010		1\$:	JSR PC,ADWRD	;ADD WORD TO STRING
6543	005632	000760				BR 2\$	
6544	005634	015	012	001	CRLF:	.BYTE 15,12,1,1,1,0	
6545						.EVEN	
6546							
6547							
6548							
6549	005642	112763	000040	177777	ADWRD:	MOVB #40,-1(R3)	;ADD SPACE TO START OF WORD
6550	005650	112223			1\$:	MOVB (R2)+,(R3)+	;ADD WORD TO STRING
6551	005652	001376				BNE 1\$	
6552	005654	000207				RTS PC	
6553	005656	000				.BYTE 0	
6554	005657	000204			LINE:	.BLKB ^D132	
6555		006064				.EVEN	
6556							
6557							
6558							
6559	006064	105767	172572		TYPE:	TSTB \$ENV	;IS THIS APT MODE
6560	006070	001401				BEQ 1\$;NO
6561	006072	000207				RTS PC	;NO TYPING IN APT MODE
6562	006074	105767	171464		1\$:	TSTB PPS	;IS PRINTER READY
6563	006100	100375				BPL 1\$;NO
6564	006102	112267	171460			MOVB (R2)+,PPB	;YES, SO PRINT IT
6565	006106	105712				TSTB (R2)	;LAST CHAR.
6566	006110	001365				BNE TYPE	;NO
6567	006112	000207				RTS PC	;YES RETURN

.SBTTL RELOCATION HANDLERS

6569								
6570								
6571								
6572								
6573								
6574	006114	012701	044000	RELCTL:	MOV #LOWSP,R1			;START OF LOW SPACE
6575	006120	012402			MOV (R4)+,R2			;END OF MOVE
6576	006122	012421		1\$:	MOV (R4)+,(R1)+			;TRANSFER TEST
6577	006124	020402			CMP R4,R2			;PROCEED TO STOP MARK
6578	006126	001375			BNE 1\$			
6579	006130	016721	000004		MOV 2\$,(R1)+			;RETURN INSTRUCTION
6580	006134	000167	035640		JMP LOWSP			;START TESTS
6581	006140	000204		2\$:	RTS R4			
6582								
6583								
6584								
6585								
6586	006142	012701	046000	RELCTH:	MOV #HIGHSP,R1			;START OF HIGH CACHE SPACE
6587	006146	012402			MOV (R4)+,R2			;END OF MOVE
6588	006150	012421		1\$:	MOV (R4)+,(R1)+			;TRANSFER TEST
6589	006152	020402			CMP R4,R2			;CONTINUE UNTIL START OF NEXT TEST
6590	006154	001375			BNE 1\$			
6591	006156	016721	000004		MOV 2\$,(R1)+			;RETURN INSTRUCTION
6592	006162	000167	037612		JMP HIGHSP			;START TEST
6593	006166	000204		2\$:	RTS R4			
6594								
6595								
6596								
6597								
6598	006170	010346		GDBD:	MOV R3,-(SP)			;SAVE R3
6599	006172	010246			MOV R2,-(SP)			;SAVE R2
6600	006174	012703	005657		MOV #LINE,R3			;LINE POINTER
6601	006200	012702	006400		MOV #ADDST,R2			;ADDRESS HEADER
6602	006204	112223		5\$:	MOVB (R2)+,(R3)+			;MOVE HEADER TO LINE
6603	006206	001376			BNE 5\$			
6604	006210	016702	176340		MOV ADD,R2			;ADDRESS TO PRINT
6605	006214	004767	000070		JSR PC,OCTASC			;ADD TO LINE
6606	006220	012702	006413	1\$:	MOV #GOODST,R2			;POINTER TO LINE HEADER
6607	006224	112223			MOVB (R2)+,(R3)+			;MOVE HEADER TO LINE
6608	006226	001376			BNE 1\$			
6609	006230	016702	176314		MOV GOOD,R2			;GOOD DATA
6610	006234	004767	000050		JSR PC,OCTASC			;ADD DATA STRING TO LINE
6611	006240	012702	006434		MOV #BADST,R2			;POINTER TO BAD HEADER
6612	006244	112223		3\$:	MOVB (R2)+,(R3)+			;MOVE HEADER TO LINE
6613	006246	001376			BNE 3\$			
6614	006250	016702	176276		MOV BAD,R2			;BAD DATA
6615	006254	004767	000030		JSR PC,OCTASC			;ADD STRING TO LINE
6616	006260	005203			INC R3			;ADJUST LINE POINTER
6617	006262	012702	005634		MOV #CRLF,R2			;ADD <CR> <LF> TO END OF LINE
6618	006266	004767	177350		JSR PC,ADWRD			
6619	006272	012702	005657		MOV #LINE,R2			;LINE POINTER
6620	006276	004767	177562		JSR PC,TYPE			;PRINT LINE
6621	006302	012602			MOV (SP)+,R2			;RESTORE R2
6622	006304	012603			MOV (SP)+,R3			;RESTORE R3
6623	006306	000207			RTS PC			
6624	006310	010446		OCTASC:	MOV R4,-(SP)			;FREE R4 FOR USE

6625	006312	010546				MOV R5,-(SP)	:FREE R5 FOR USE
6626	006314	010146				MOV R1,-(SP)	:FREE R1 FOR USE
6627	006316	012704	000006			MOV #6,R4	:PASS COUNTER
6628	006322	062703	000005			ADD #5,R3	:LAST DIGIT POINTER
6629	006326	012705	000003		3\$:	MOV #3,R5	:PASS COUNTER FOR OCTAL DIGIT
6630	006332	010201				MOV R2,R1	:GRAB LSD
6631	006334	042701	177770			BIC #177770,R1	:MASK OFF OTHER DIGITS
6632	006340	062701	000060			ADD #60,R1	:CONVERT TO ASCII
6633	006344	110143				MOVB R1,-(R3)	:ADD CAHR TO LINE
6634	006346	005304				DEC R4	
6635	006350	001405				BEQ 1\$:LAST DIGIT SO EXIT
6636	006352	000241			2\$:	CLC	:POSITION NEXT DIGIT
6637	006354	006002				ROR R2	
6638	006356	005305				DEC R5	
6639	006360	001374				BNE 2\$	
6640	006362	000751				BR 3\$	
6641	006364	062703	000006		1\$:	ADD #6,R3	:CORRECT LINE POINTER
6642	006370	012601				MOV (SP)+,R1	:RESTORE R1
6643	006372	012605				MOV (SP)+,R5	:RESTORE R5
6644	006374	012604				MOV (SP)+,R4	:RESTORE R4
6645	006376	000207				RTS PC	
6646	006400	042101	051104	051505	ADDST:	.ASCIZ /ADDRESS = /	
6647	006413	040	020040	043440	GOODST:	.ASCIZ / GOOD DATA = /	
6648	006434	020040	020040	040502	BADST:	.ASCIZ / BAD DATA = /	
6649						.EVEN	
6650							
6651							
6652							
6653							
6654	006454	010446			BITNAM:	MOV R4,-(SP)	:FREE R4 FOR USE
6655	006456	016704	176060			MOV ERROR,R4	:ERRORING BIT(S)
6656	006462	012701	006556			MOV #BIT,R1	:BIT TABLE
6657	006466	005704			2\$:	TST R4	:RETURN IF NO BITS SET
6658	006470	001002				BNE 1\$	
6659	006472	012604				MOV (SP)+,R4	:RESTORE R4
6660	006474	000207				RTS PC	
6661	006476	032704	000001		1\$:	BIT #1,R4	:DID BIT FAIL
6662	006502	001420				BEQ 3\$:NO
6663	006504	012703	005657		4\$:	MOV #LINE,R3	:LINE TO PRINT
6664	006510	112123				MOVB (R1)+,(R3)+	
6665	006512	001376				BNE 4\$	
6666	006514	112743	000015			MOVB #15,-(R3)	
6667	006520	112723	000012			MOVB #12,(R3)+	
6668	006524	105023				CLRB (R3)+	
6669	006526	012702	005657			MOV #LINE,R2	:PRINT LINE
6670	006532	004767	177326			JSR PC,TYPE	
6671	006536	000257				CCC	
6672	006540	006004				ROR R4	
6673	006542	000751				BR 2\$	
6674	006544	062701	000006		3\$:	ADD #6,R1	:POINT TO NEXT NAME
6675	006550	000257				CCC	
6676	006552	006004				ROR R4	
6677	006554	000744				BR 2\$	
6678	006556	044502	030124	000060	BIT:	.ASCIZ /BIT00/	
6679	006564	044502	030124	000061		.ASCIZ /BIT01/	
6680	006572	044502	030124	000062		.ASCIZ /BIT02/	

6681	006600	044502	030124	000063	.ASCIZ /BIT03/
6682	006606	044502	030124	000064	.ASCIZ /BIT04/
6683	006614	044502	030124	000065	.ASCIZ /BIT05/
6684	006622	044502	030124	000066	.ASCIZ /BIT06/
6685	006630	044502	030124	000067	.ASCIZ /BIT07/
6686	006636	044502	030124	000070	.ASCIZ /BIT08/
6687	006644	044502	030124	000071	.ASCIZ /BIT09/
6688	006652	044502	030524	000060	.ASCIZ /BIT10/
6689	006660	044502	030524	000061	.ASCIZ /BIT11/
6690	006666	044502	030524	000062	.ASCIZ /BIT12/
6691	006674	044502	030524	000063	.ASCIZ /BIT13/
6692	006702	044502	030524	000064	.ASCIZ /BIT14/
6693	006710	044502	030524	000065	.ASCIZ /BIT15/

.SBTTL CACHE REGISTER RESPONSE TESTS

```

6695
6696
6697
6698
6699
6700 006716 005267 171724
6701 006722 012767 001015 171016
6702 006730 012767 004642 171046
6703 006736 012767 000340 171042
6704 006744 012700 004714
6705 006750 005067 175564
6706 006754 016701 170764
6707 006760 016767 175554 175554
6708 006766 004067 176360
6709 006772 000004
6710 006774 035722
6711 006776 035734
6712 007000 004010
6713 007002 006750
6714 007004 006750
6715
6716
6717
6718
6719
6720
6721
6722
6723 007006 005267 171634
6724 007012 012767 001015 170726
6725 007020 005067 175514
6726 007024 016701 170716
6727 007030 016767 175504 175504
6728 007036 004067 176310
6729 007042 000004
6730 007044 035722
6731 007046 035754
6732 007050 004010
6733 007052 007012
6734 007054 007012
6735
6736
6737
6738
6739
6740
6741
6742
6743 007056 005267 171564
6744 007062 012767 001015 170656
6745 007070 005067 175444
6746 007074 016701 170650
6747 007100 016767 175434 175434
6748 007106 004067 176240
6749 007112 000004
6750 007114 035722

```

```

          :
          :
          :
TST001:  ATTEMPT READ INTO CMPE TO TEST ADDRESS SELECT LOGIC
          IF TIME OUT OCCURES THEN LOGIC IN ERROR
          INC TID           ;UPDATE TEST ID
          MOV #OFF,CCR      ;DISABLE CACHE
          MOV #TRAP,4       ;SETUP TRAP VECTOR
          MOV #340,6
          MOV #ERTSHI,R0    ;ERROR LOOP HANDLER
1$:      CLR ITTRAP        ;RESET TRAP FLAG
          MOV CMPE,R1       ;READ PARITY REGISTER
          MOV ITTRAP,ERROR  ;SET ERROR FLAG IF TRAPPED
          JSR RO,SETEN      ;PRINT LIST OF SENTENCES
          .WORD ^D4
          .WORD SEN1       ;CACHE REGISTER RESPONSE TESTS
          .WORD SEN2       ;READING PARITY ERROR REGISTER CAUSED TIME OUT
          JSR RO,(R0)       ;TAKE SELECTED ACTION IF ERROR
          .WORD 1$         ;LOOP ON ERROR
          .WORD 1$         ;LOOP ON TEST
          :
          :
          :
          :
          :
TST002:  ATTEMPT READ INTO CMPE TO CHECK ADDRESS SELECT LOGIC
          IF TIME OUT OCCURES THEN LOGIC IN ERROR
          INC TID           ;UPDATE TEST ID
1$:      MOV #OFF,CCR      ;DISABLE CACHE
          CLR ITTRAP       ;RESET TRAP FLAG
          MOV CCR,R1        ;ATTEMPT READ TO CONTROL REGISTER
          MOV ITTRAP,ERROR  ;SET ERROR FLAG IF TRAP OCCURED
          JSR RO,SETEN      ;PRINT LIST OF SENTENCES
          .WORD ^D4
          .WORD SEN1       ;CACHE REGISTER RESPONSE TESTS
          .WORD SEN3       ;READING CONTROL REGISTER CAUSED TIME OUT
          JSR RO,(R0)       ;TAKE SELECTED ACTION IF ERROR
          .WORD 1$         ;LOOP ON ERROR
          .WORD 1$         ;LOOP ON TEST
          :
          :
          :
          :
          :
TST003:  ATTEMPT READ INTO CMR TO CHECK ADDRESS SELECT LOGIC
          IF TIME OCCURES THEN LOGIC IN ERROR
          INC TID           ;UPDATE TEST ID
1$:      MOV #OFF,CCR      ;DISABLE CACHE
          CLR ITTRAP       ;RESET TRAP FLAG
          MOV CMR,R1        ;READ MAINTENANCE REGISTER
          MOV ITTRAP,ERROR  ;SET ERROR FLAG IF TRAP OCCURED
          JSR RO,SETEN      ;PRINT LIST OF SENTENCES
          .WORD ^D4
          .WORD SEN1       ;CACHE REGISTER RESPONSE TESTS

```

```

6751 007116 035772 .WORD SEN4 ;READING MAINTENANCE REGISTER CAUSED TIME OUT
6752 007120 004010 JSR RO,(R0) ;TAKE SELECTED ACTION ON ERROR
6753 007122 007062 .WORD 1$ ;LOOP ON ERROR
6754 007124 007062 .WORD 1$ ;LOOP ON TEST
6755
6756
6757
6758
6759
6760
6761
6762
6763 007126 005267 171514 TST004: ATTEMPT READ INTO CHR TO CHECK ADDRESS SELECT LOGIC
6764 007132 012767 001015 170606 IF TIME OUT OCCURES THEN LOGIC IN ERROR
6765 007140 005067 175374 1$: INC TID ;UPDATE TEST ID
6766 007144 016701 170602 CLR ITTRAP ;DISABLE CACHE
6767 007150 016767 175364 175364 MOV CHR,R1 ;RESET TRAP FLAG
6768 007156 004067 176170 MOV ITTRAP,ERROR ;READ HIT REGISTER
6769 007162 000004 JSR RO,SETEN ;SET ERROR FLAG IF TRAP OCCURED
6770 007164 035722 .WORD ^D4 ;PRINT LIST OF SENTENCES
6771 007166 036010 .WORD SEN1 ;CACHE REGISTER RESPONSE TESTS
6772 007170 004010 .WORD SEN5 ;READING HIT REGISTER CAUSED TIME OUT
6773 007172 007132 JSR RO,(R0) ;TAKE SELECTED ACTION ON ERROR
6774 007174 007132 .WORD 1$ ;LOOP ON ERROR
6775 .WORD 1$ ;LOOP ON TEST
6776
6777
6778
6779
6780
6781
6782
6783 007176 005267 171444 TST005: READ ADDRESS BELOW ERROR REGISTER TO CHECK THAT CACHE
6784 007202 012767 001015 170536 WILL NOT RESPOND TO THAT LOCATION
6785 007210 005067 175324 1$: INC TID ;UPDATE TEST ID
6786 007214 016701 170522 CLR ITTRAP ;DISABLE CACHE
6787 007220 016767 175314 175314 MOV CMPE-2,R1 ;RESET TRAP FLAG
6788 007226 005367 175310 MOV ITTRAP,ERROR ;READ ADDRESS BELOW ERROR REGISTER
6789 007232 004067 176114 DEC ERROR ;ERROR IF NO TRAP
6790 007236 000004 JSR RO,SETEN ;PRINT LIST OF SENTENCES
6791 007240 035722 .WORD ^D4 ;CACHE REGISTER RESPONSE TESTS
6792 007242 036026 .WORD SEN1 ;READING INVALID ADDRESS 177740 DID NOT CAUSE TI
6793 007244 004010 .WORD SEN6 ;TAKE SELECTED ACTION ON ERROR
6794 007246 007202 JSR RO,(R0) ;LOOP ON ERROR
6795 007250 007202 .WORD 1$ ;LOOP ON TEST
6796 .WORD 1$
6797
6798
6799
6800
6801
6802
6803
6804 007252 005267 171370 TST006: READ LOCATION ABOVE HIT REGISTER TO CHECK THAT
6805 007256 012767 001015 170462 CACHE WILL NOT RESPOND
6806 007264 005067 175250 1$: INC TID ;UPDATE TEST ID
MOV #OFF,CCR ;DISABLE CACHE
CLR ITTRAP ;RESET TRAP FLAG

```

```

6807 007270 016701 170462
6808 007274 016767 175240 175240
6809 007302 005367 175234
6810
6811 007306 004067 176040
6812 007312 000004
6813 007314 035722
6814 007316 036052
6815 007320 004010
6816 007322 007256
6817 007324 007256
6818
6819
6820
6821
6822
6823
6824
6825
6826
6827
6828
6829 007326 005267 171314
6830 007332 012767 001015 170406
6831 007340 005067 175174
6832 007344 012767 000001 170372
6833 007352 016767 170366 175162
6834 007360 042767 177776 175154
6835 007366 004067 175760
6836 007372 000006
6837 007374 035722
6838 007376 036076
6839 007400 036114
6840 007402 004010
6841 007404 007332
6842 007406 007332
6843
6844
6845
6846
6847
6848
6849
6850
6851
6852
6853 007410 005267 171232
6854 007414 012767 001015 170324
6855 007422 052767 000001 170316
6856 007430 016767 170312 175104
6857 007436 042767 177776 175076
6858 007444 005367 175072
6859 007450 004067 175676
6860 007454 000006
6861 007456 035722
6862 007460 036126
    
```

```

MOV CHR+4,R1 ;READ TWO LOCATIONS ABOVE HIT REGISTER
MOV ITTRAP,ERROR ;IF NO TRAP THEN ERROR
DEC ERROR

JSR RO,SETEN ;PRINT LIST OF SENTENCES
.WORD ^D4
.WORD SEN1 ;CACHE REGISTER RESPONSE TESTS
.WORD SEN7 ;READING INVALID ADDRESS 177756 DID NOT CAUSE T1
JSR RO,(RO) ;TAKE SELECT ACTION ON ERROR
.WORD 1$ ;LOOP ON ERROR
.WORD 1$ ;LOOP ON TEST
    
```

```

TESTING ADDRESS SELECTION LOGIC BY WRITING ONE INTO UNUSED
CMPE REGISTER BIT00 THEN READ CONTENTS OF REGISTER BACK
LOOKING TO SEE IF BIT00 IS SET.
IF BIT00 IS SET IT IS POSSIBLE WE ARE ADDRESSING THE WRONG
REGISTER
TST007: INC TID ;UPDATE TEST ID
1$: MOV #OFF,CCR ;DISABLE CACHE
CLR ITTRAP ;RESET TRAP FLAG
MOV #1,CMPE ;WRITE 1 INTO ERROR REGISTER
MOV CMPE,ERROR ;ERROR IF BIT 0 SET
BIC #177776,ERROR ;DON'T LOOK AT UPPER BITS
JSR RO,SETEN ;PRINT LIST OF SENTENCES
.WORD ^D6
.WORD SEN1 ;CACHE REGISTER RESPONSE TESTS
.WORD SEN8 ;UNUSED CMPE BIT00 READ AS ONE
.WORD SEN9 ;POSSIBLE REGISTER ADDRESSING ERROR
JSR RO,(RO) ;TAKE SELECTED ACTION ON ERROR
.WORD 1$ ;LOOP ON ERROR
.WORD 1$ ;LOOP ON TEST
    
```

```

TESTING ADDRESS SELECTION LOGIC BY WRITING A ONE INTO
CCR BITC THEN READ REGISTER BACK
IF BIT00 READ AS ZERO THEN IT IS POSSIBLE WE ARE
ADDRESSING WRONG REGISTER
TST010: INC TID ;UPDATE TEST ID
1$: MOV #OFF,CCR ;DISABLE CACHE
BIS #1,CCR ;WRITE 1 INTO CONTROL REGISTER
MOV CCR,ERROR ;ERROR IF BIT00 NOT SET
BIC #177776,ERROR
DEC ERROR
JSR RO,SETEN ;PRINT LIST OF SENTENCES
.WORD ^D6
.WORD SEN1 ;CACHE REGISTER RESPONSE TESTS
.WORD SEN10 ;WROTE ONE INTO CCR BIT00 READ BACK ZERO
    
```

CACHE DIAG.
CFKKAB.P11

MACY11 30A(1052)
25-JUN-79 13:31

31-OCT-79 15:29 PAGE 62-14
CACHE REGISTER RESPONSE TESTS

J 3

SEQ 0035

6863 007462 036114
6864 007464 004010
6865 007466 007414
6866 007470 007414
6867
6868
6869
6870

⋮
⋮
⋮
⋮

.WORD SEN9
JSR R0,(R0)
.WORD 1\$
.WORD 1\$

;POSSIBLE REGISTER ADDRESSING ERROR
;TAKE SELECTED ACTION ON ERROR
;LOOP ON ERROR
;LOOP ON TEST

.SBTTL CONTROL REGISTER DATA TEST

```

6872
6873
6874
6875
6876
6877
6878 007472 005267 171150
6879 007476 012767 001415 170242
6880 007504 042767 000001 170234
6881 007512 016767 170230 175022
6882 007520 012767 001015 170220
6883 007526 042767 177776 175006
6884 007534 004067 175612
6885 007540 000004
6886 007542 036150
6887 007544 036162
6888 007546 004010
6889 007550 007504
6890 007552 007504
6891
6892
6893
6894
6895
6896
6897
6898
6899
6900 007554 005267 171066
6901 007560 004467 176356
6902 007564 007640
6903 007566 042737 000004 177746
6904 007574 013737 177746 004542
6905 007602 012737 001015 177746
6906 007610 042737 177773 004542
6907 007616 004037 005352
6908 007622 000006
6909 007624 036150
6910 007626 036204
6911 007630 042166
6912 007632 004010
6913 007634 046000
6914 007636 046000
6915
6916
6917
6918
6919
6920
6921
6922
6923
6924 007640 005267 171002
6925 007644 052737 000004 177746
6926 007652 013737 177746 004542
6927 007660 012737 001015 177746

```

```

WRITE ZERO INTO CCR BIT00 THEN READ CCR
IF CCR IS READ AS ONE THEN CACHE CCR REGISTER MAY BE BAD
OR CACHE REGISTER DATA PATH COULD BE IN ERROR
TST011: INC TID ;UPDATE TEST ID
        MOV #OFF+BIT08,CCR ;DISABLE CACHE
1$: BIC #BIT00,CCR ;WRITE ZERO TO BIT00
    MOV CCR,ERROR ;ERROR IF BIT00 = 1
    MOV #OFF,CCR ;DISABLE CACHE
    BIC #-BIT00-1,ERROR
    JSR RO,SETEN ;PRINT LIST OF SENTENCES
    .WORD ^D4
    .WORD SEN11 ;CONTROL REGISTER DATA TEST
    .WORD SEN12 ;WROTE ZERO INTO CCR BIT00 READ BACK ONE
    JSR RO,(RO) ;TAKE SELECTED ACTION ON ERROR
    .WORD 1$ ;LOOP ON ERROR
    .WORD 1$ ;LOOP ON TEST

WRITE ZERO INTO CCR BIT02 THEN READ CCR
IF BIT02 IS READ AS ONE THEN CCR REGISTER MAY BE BAD
OR CACHE REGISTER DATA PATH MAY BE AT FAULT
TST012: INC TID ;UPDATE TEST ID
        JSR R4,RELCTH ;RELOCATE TEST TO HIGH CACHE
        .WORD TST013
1$: BIC #BIT02,@CCR ;WRITE 0 INTO BIT02
    MOV @CCR,@ERROR ;ERROR IF BIT02 = 1
    MOV #OFF,@CCR ;DISABLE CACHE
    BIC #-BIT02-1,@ERROR
    JSR RO,@SETEN ;PRINT LIST OF SENTENCES
    .WORD ^D6
    .WORD SEN11 ;CONTROL REGISTER DATA TEST
    .WORD SEN13 ;WROTE ZERO INTO CCR BIT02 READ ONE
    .WORD SEN168
    JSR RO,(RO) ;TAKE SELECTED ACTION ON ERROR
    .WORD HIGHSP ;LOOP ON ERROR
    .WORD HIGHSP ;LOOP ON TEST

WRITE ONE INTO CCR BIT02 THEN READ CCR
IF CCR BIT02 READ BACK AS ZERO THEN CCR REGISTER BIT MAY BE BAD
OR CACHE REGISTER DATA PATH MAY BE AT FAULT
TST013: INC TID ;UPDATE TEST ID
1$: BIS #BIT02,@CCR ;WRITE 4 INTO CONTROL REGISTER
    MOV @CCR,@ERROR ;ERROR IF BIT02 <> 1
    MOV #OFF,@CCR ;DISABLE CACHE

```


6952	010034	004467	176054
6953	010040	010114	
6954	010042	042737	000010 177746
6955	010050	013737	177746 004542
6956	010056	012737	001015 177746
6957	010064	042737	177767 004542
6958	010072	004037	005352
6959	010076	000006	
6960	010100	036150	
6961	010102	036244	
6962	010104	042166	
6963	010106	004010	
6964	010110	044000	
6965	010112	044000	

```

JSR R4,RELCTL           ;RELOCATE TEST TO LOW CACHE
.WORD TST015
1$: BIC #BIT03,@#CCR     ;WRITE 0 TO BIT03
MOV @#CCR,@#ERROR      ;ERROR IF BIT = 1
MOV #OFF,@#CCR         ;DISABLE CACHE
BIC #-BIT03-1,@#ERROR
JSR R0,@#SETEN        ;PRINT LIST OF SENTENCES
.WORD ^D6
.WORD SEN11           ;CONTROL REGISTER DATA TEST
.WORD SEN15          ;WROTE ZERO INTO CCR BIT03 READ ONE
.WORD SEN168
JSR R0,(R0)           ;TAKE SELECTED ACTION ON ERROR
.WORD LOWSP          ;LOOP ON ERROR
.WORD LOWSP          ;LOOP ON TEST

```

CACHE DIAG.
CFKKAB.P11

MACY11 30A(1052)
25-JUN-79 13:31

31-OCT-79

15:29 PAGE 64

CONTROL REGISTER DATA TEST

N 3

SEQ 0039

6967
6968
6969
6970
6971

⋮
⋮
⋮
⋮

```

6973
6974
6975
6976
6977 010114 005267 170526
6978 010120 052767 000010 167620
6979 010126 016767 167614 174406
6980 010134 012767 001015 167604
6981 010142 042767 177767 174372
6982 010150 162767 000010 174364
6983 010156 004067 175170
6984 010162 000004
6985 010164 036150
6986 010166 036264
6987 010170 004010
6988 010172 010120
6989 010174 010120

```

```

WRITE ONE INTO CCR BIT03 THEN READ CCR
IF CCR BIT03 READ AS ZERO THEN CCR REGISTER BIT MAY BE BAD
OR CACHE REGISTER DATA PATH MAY BE AT FAULT
TST015:
1$: INC TID ;UPDATE TEST ID
    BIS #BIT03,CCR ;WRITE 1 INTO CONTROL REGISTER BIT03
    MOV CCR,ERROR ;ERROR IF BIT03 = 0
    MOV #OFF,CCR ;DISABLE CACHE
    BIC #-BIT03-1,ERROR
    SUB #10,ERROR
    JSR RO,SETEN ;PRINT LIST OF SENTENCES
    .WORD ^D4
    .WORD SEN11 ;CONTROL REGISTER DATA TEST
    .WORD SEN16 ;WROTE ONE INTO CCR BIT03 READ ZERO
    JSR RO,(RO) ;TAKE SELECTED ACTION ON ERROR
    .WORD 1$ ;LOOP ON ERROR
    .WORD 1$ ;LOOP ON TEST

```

```

6990
6991
6992
6993
6994
6995
6996
6997
6998
6999 010176 005267 170444
7000 010202 042767 000100 167536
7001 010210 016767 167532 174324
7002 010216 012767 001015 167522
7003 010224 042767 177677 174310
7004 010232 004067 175114
7005 010236 000004
7006 010240 036150
7007 010242 036304
7008 010244 004010
7009 010246 010202
7010 010250 010202

```

```

WRITE ONE INTO CCR BIT06 THEN READ CCR
IF BIT06 READ AS ONE THEN CCR REGISTER BIT MAY BE BAD
OR CACHE REGISTER DATA PATH MAY BE AT FAULT
TST016:
1$: INC TID ;UPDATE TEST ID
    BIC #BIT06,CCR ;WRITE 0 INTO CONTROL REGISTER BIT06
    MOV CCR,ERROR ;ERROR IF BIT06 = 1
    MOV #OFF,CCR ;DISABLE CACHE
    BIC #-BIT06-1,ERROR
    JSR RO,SETEN ;PRINT LIST OF SENTENCES
    .WORD ^D4
    .WORD SEN11 ;CONTROL REGISTER DATA TEST
    .WORD SEN17 ;WROTE ZERO INTO CCR BIT06 READ ONE
    JSR RO,(RO) ;TAKE SELECTED ACTION ON ERROR
    .WORD 1$ ;LOOP ON ERROR
    .WORD 1$ ;LOOP ON TEST

```

```

7011
7012
7013
7014
7015
7016
7017
7018
7019
7020 010252 005267 170370
7021 010256 052767 000100 167462
7022 010264 016767 167456 174250
7023 010272 012767 001015 167446
7024 010300 012701 044000
7025 010304 012102
7026 010306 020127 050000
7027 010312 001374
7028 010314 042767 177677 174220

```

```

WRITE ONE INTO CCR BIT06 THEN READ CCR
IF CCR BIT06 READ AS ZERO THEN CCR REGISTER BIT MAY BE BAD
OR CACHE REGISTER DATA PATH MAY BE AT FAULT
TST017:
1$: INC TID ;UPDATE TEST ID
    BIS #BIT06,CCR ;WRITE 1 INPUT CONTROL REGISTER BIT06
    MOV CCR,ERROR ;ERROR IF BIT06 = 0
    MOV #OFF,CCR ;DISABLE CACHE
    MOV #LOWSP,R1 ;RETAG ALL CACHE
    MOV (R1)+,R2 ;READ TO TAG LOCATION
    CMP R1,#LOW1 ;TAG FULL 1K
    BNE 10$
    BIC #-BIT06-1,ERROR
10$:

```

```

7029 010322 162767 000100 174212          SUB #100,ERROR
7030 010330 004067 175016          JSR RO,SETEN          ;PRINT LIST OF SENTENCES
7031 010334 000004          .WORD ^D4
7032 010336 036150          .WORD SEN11          ;CONTROL REGISTER DATA TEST
7033 010340 036324          .WORD SEN18          ;WROTE ONE INTO CCR BIT06 READ ZERO
7034 010342 004010          JSR RO,(RO)          ;TAKE SELECTED ACTION ON ERROR
7035 010344 010256          .WORD 1$            ;LOOP ON ERROR
7036 010346 010256          .WORD 1$            ;LOOP ON TEST
7037
7038
7039
7040
7041
7042
7043
7044
7045
7046 010350 005267 170272          TST020:             WRITE ZERO INTO CCR BIT07 THEN READ CCR
7047 010354 042767 000200 167364          1$:                 IF CCR BIT07 READ AS ONE THEN CCR REGISTER BIT MAY BE BAD
7048 010362 016767 167360 174152          INC TID              OR CACHE REGISTER DATA PATH MAY BE AT FAULT
7049 010370 012767 001015 167350          BIC #BIT07,CCR      ;UPDATE TEST ID
7050 010376 042767 177577 174136          MOV CCR,ERROR       ;WRITE 0 INTO CONTROL REGISTER BIT07
7051 010404 004067 174742          MOV #OFF,CCR        ;ERROR IF BIT07 = 1
7052 010410 000004          BIC #-BIT07-1,ERROR ;DISABLE CACHE
7053 010412 036150          JSR RO,SETEN        ;PRINT LIST OF SENTENCES
7054 010414 036344          .WORD ^D4
7055 010416 004010          .WORD SEN11         ;CONTROL REGISTER DATA TEST
7056 010420 010354          .WORD SEN19         ;WROTE ZERO INTO CCR BIT07 READ ONE
7057 010422 010354          JSR RO,(RO)        ;TAKE SELECTED ACTION ON ERROR
7058
7059
7060
7061
7062
7063
7064
7065
7066
7067 010424 005267 170216          TST021:             WRITE ONE INTO CCR BIT07 THEN READ CCR
7068 010430 052767 000200 167310          2$:                 IF CCR BIT07 READ AS ZERO THEN CCR REGISTER BIT MAY BE BAD
7069 010436 016767 167304 174076          INC TID              OR CACHE REGISTER DATA PATH MAY BE AT FAULT
7070 010444 012767 001015 167274          BIS #BIT07,CCR      ;UPDATE TEST ID
7071 010452 042767 177577 174062          MOV CCR,ERROR       ;WRITE 1 INTO CONTROL REGISTER BIT07
7072 010460 162767 000200 174054          MOV #OFF,CCR        ;ERROR IF BIT07 = 0
7073 010466 004067 174660          BIC #-BIT07-1,ERROR ;DISABLE CACHE
7074 010472 000004          SUB #200,ERROR
7075 010474 036150          JSR RO,SETEN        ;PRINT LIST OF SENTENCES
7076 010476 036364          .WORD ^D4
7077 010500 004010          .WORD SEN11         ;CONTROL REGISTER DATA TEST
7078 010502 010430          .WORD SEN20         ;WROTE ONE INTO CCR BIT07 READ ZERO
7079 010504 010430          JSR RO,(RO)        ;TAKE SELECTED ACTION ON ERROR
7080
7081
7082
7083
7084
WRITE ZERO INTO CCR BIT08 THEN READ CCR
IF CCR BIT08 READ AS ONE THEN CCR REGISTER BIT MAY BE BAD
OR CACHE REGISTER DATA PATH MAY BE AT FAULT

```

7085	010506	005267	170134		TST022:	INC TID	:UPDATE TEST ID
7086	010512	042767	001000	167226	1\$:	BIC #BIT09,CCR	:WRITE 0 INTO CONTROL REGISTER BIT08
7087	010520	016767	167222	174014		MOV CCR,ERROR	:ERROR IF BIT08 = 1
7088	010526	012767	001015	167212		MOV #OFF,CCR	:DISABLE CACHE
7089	010534	042767	177377	174000		BIC #-BIT08-1,ERROR	
7090	010542	004067	174604			JSR RO,SETEN	:PRINT LIST OF SENTENCES
7091	010546	000004				.WORD ^D4	
7092	010550	036150				.WORD SEN11	:CONTROL REGISTER DATA TEST
7093	010552	036404				.WORD SEN21	:WROTE ZERO INTO CCR BIT09 READ ONE
7094	010554	004010				JSR RO,(RO)	:TAKE SELECTED ACTION ON ERROR
7095	010556	010512				.WORD 1\$:LOOP ON ERROR
7096	010560	010512				.WORD 1\$:LOOP ON TEST
7097					:		
7098					:		
7099					:		
7100					:		
7101					:		
7102					:		
7103					:		
7104					:		
7105					:		
7106	010562	005267	170060		TST023:	WRITE ONE INTO CCR BIT09 THEN READ CCR	
7107	010566	052767	001000	167152	1\$:	IF CCR BIT09 READ AS ZERO THEN CCR REGISTER BIT MAY BE BAD	
7108	010574	016767	167146	173740		OR CACHE REGISTER DATA PATH MAY BE AT FAULT	
7109	010602	012767	001015	167136		INC TID	:UPDATE TEST ID
7110	010610	042767	176777	173724		BIS #BIT09,CCR	:WRITE 1 INTO CONTROL REGISTER BIT09
7111	010616	162767	001000	173716		MOV CCR,ERROR	:ERROR IF BIT09 = 0
7112	010624	004067	174522			MOV #OFF,CCR	:DISABLE CACHE
7113	010630	000004				BIC #-BIT09-1,ERROR	
7114	010632	036150				SUB #1000,ERROR	
7115	010634	036424				JSR RO,SETEN	:PRINT LIST OF SENTENCES
7116	010636	004010				.WORD ^D4	
7117	010640	010566				.WORD SEN11	:CONTROL REGISTER DATA TEST
7118	010642	010566				.WORD SEN22	:WROTE ONE INTO CCR BIT09 READ ZERO
7119						JSR RO,(RO)	:TAKE SELECTED ACTION ON ERROR
7120						.WORD 1\$:LOOP ON ERROR
7121						.WORD 1\$:LOOP ON TEST
7122					:		
7123	010644	005267	167776		TST024:	INC TID	:UPDATE TEST ID
7124	010650	042737	002000	177746	1\$:	BIC #BIT10,#CCR	:WRITE ZERO INTO CONTROL REG BIT10
7125	010656	016767	167064	173656		MOV CCR,ERROR	:READ CONTROL REGISTER
7126	010664	042767	175777	173650		BIC #-BIT10-1,ERROR	:MASK BIT10
7127	010672	004067	174454			JSR RO,SETEN	:REPORT ERROR IF ANY
7128	010676	000004				.WORD ^D4	
7129	010700	036150				.WORD SEN11	:CONTROL REGISTER DATA TEST
7130	010702	036444				.WORD SEN23	:WROTE ZERO INTO CCR BIT10 READ ONE
7131	010704	004010				JSR RO,(RO)	:TAKE SELECTED ACTION ON ERROR
7132	010706	010650				.WORD 1\$	
7133	010710	010650				.WORD 1\$	
7134							
7135							
7136							
7137							
7138	010712	005267	167730		TST025:	INC TID	:UPDATE TEST ID
7139	010716	052767	002000	167022	1\$:	BIS #BIT10,CCR	:WRITE ONE TO BIT 10
7140	010724	016767	167016	173610		MOV CCR,ERROR	:SAVE CCR

CACHE DIAG.
CFKKAB.P11

MACY11 30A(1052)
25-JUN-79 13:31

31-OCT-79 15:29 PAGE 65-3
CONTROL REGISTER DATA TEST

SEQ 0043

7141	010732	012767	001015	167006
7142	010740	042767	175777	173574
7143	010746	162767	002000	173566
7144	010754	004067	174372	
7145	010760	000004		
7146	010762	036150		
7147	010764	036464		
7148	010766	004010		
7149	010770	010716		
7150	010772	010716		

```

MOV #OFF,CCR
BIC #-BIT10-1,ERROR
SUB #BIT10,ERROR
JSR RO,SETEN
.WORD ^D4
.WORD SEN11
.WORD SEN24
JSR RO,(RO)
.WORD 1$
.WORD 1$

```

```

;DISABLE CACHE
;MASK BIT10
;REPORT ERROR IF ANY
;CONTROL REGISTER DATA TEST
;WROTE ONE INTO CCR BIT10 READ ZERO
;TAKE SELECTED ACTION ON ERROR
;LOOP ON ERROR
;LOOP ON TEST

```

.SBTTL CONTROL REGISTER UNUSED BIT TEST

```

7152
7153
7154
7155
7156
7157
7158 010774 005267 167646
7159 011000 012767 001015 166740
7160 011006 052767 000002 166732
7161 011014 016767 166726 173520
7162 011022 042767 177775 173512
7163 011030 004067 174316
7164 011034 000004
7165 011036 036504
7166 011040 036520
7167 011042 004010
7168 011044 011000
7169 011046 011000
7170
7171
7172
7173
7174
7175
7176
7177
7178 011050 005267 167572
7179 011054 012767 001015 166664
7180 011062 052767 000020 166656
7181 011070 016767 166652 173444
7182 011076 042767 177757 173436
7183 011104 004067 174242
7184 011110 000004
7185 011112 036504
7186 011114 036536
7187 011116 004010
7188 011120 011054
7189 011122 011054
7190
7191
7192
7193
7194
7195
7196
7197
7198 011124 005267 167516
7199 011130 012767 001015 166610
7200 011136 052767 000040 166602
7201 011144 016767 166576 173370
7202 011152 042767 177757 173362
7203 011160 004067 174166
7204 011164 000004
7205 011166 036504
7206 011170 036554
7207 011172 004010

```

```

WRITE INTO UNUSED CCR REGISTER BIT01 THEN READ CCR
IF CCR BIT01 READ AS ONE THEN CACHE DATA PATH
ERROR
TST026:
1$:  INC TID             ;UPDATE TEST ID
      MOV #OFF,CCR      ;DISABLE CACHE
      BIS #BIT01,CCR    ;WRITE 1 INTO CONTROL REGISTER BIT01
      MOV CCR,ERROR     ;ERROR IF BIT01 = 1
      BIC #-BIT01-1,ERROR
      JSR RO,SETEN      ;PRINT LIST OF SENTENCES
      .WORD ^D4
      .WORD SEN25       ;CONTROL REGISTER UNUSED BIT TEST
      .WORD SEN26       ;READ ONE FROM UNUSED CCR BIT01
      JSR RO,(RO)       ;TAKE SELECTED ACTION ON ERROR
      .WORD 1$          ;LOOP ON ERROR
      .WORD 1$          ;LOOP ON TEST

```

```

WRITE ONE INTO UNUSED CCR BIT04 THEN READ CCR
IF CCR BIT04 READ AS ONE THEN CACHE REGISTER DATA PATH ERROR
TST027:
1$:  INC TID             ;UPDATE TEST ID
      MOV #OFF,CCR      ;DISABLE CACHE
      BIS #BIT04,CCR    ;WRITE 1 INTO CONTROL REGISTER BIT04
      MOV CCR,ERROR     ;ERROR IF BIT04 = 1
      BIC #-BIT04-1,ERROR
      JSR RO,SETEN      ;PRINT LIST OF SENTENCES
      .WORD ^D4
      .WORD SEN25       ;CONTROL REGISTER UNUSED BIT TEST
      .WORD SEN27       ;READ ONE FROM UNUSED CCR BIT04
      JSR RO,(RO)       ;TAKE SELECTED ACTION IF ERROR
      .WORD 1$          ;LOOP ON ERROR
      .WORD 1$          ;LOOP ON TEST

```

```

WRITE ONE INTO UNUSED CCR BIT05 THEN READ CCR
IF CCR BIT05 READ AS ONE THEN CACHE REGISTER DATA PATH ERROR
TST030:
1$:  INC TID             ;UPDATE TEST ID
      MOV #OFF,CCR      ;DISABLE CACHE
      BIS #BIT05,CCR    ;WRITE 1 INTO CONTROL REGISTER BIT05
      MOV CCR,ERROR     ;ERROR IF BIT05 = 1
      BIC #-BIT05-1,ERROR
      JSR RO,SETEN      ;PRINT LIST OF SENTENCES
      .WORD ^D4
      .WORD SEN25       ;CONTROL REGISTER UNUSED BIT TEST
      .WORD SEN28       ;READ ONE FROM UNUSED CCR BIT05
      JSR RO,(RO)       ;TAKE SELECTED ACTION ON ERROR

```

```

7208 011174 011130          .WORD 1$          ;LOOP ON ERROR
7209 011176 011130          .WORD 1$          ;LOOP ON TEST
7210
7211
7212
7213
7214
7215
7216
7217
7218 011200 005267 167442    TST031:          WRITE ONE INTO UNUSED CCR BIT08 THEN READ CCR
7219 011204 012767 001015 166534    1$:             IF CCR BIT08 READ AS ONE THEN CACHE REGISTER DATA PATH ERROR
7220 011212 052767 000400 166526    .INC TID        ;UPDATE TEST ID
7221 011220 016767 166522 173314    .MOV #OFF,CCR   ;DISABLE CACHE
7222 011226 042767 177377 173306    .BIS #BIT08,CCR;WRITE 1 INTO CONTROL REGISTER BIT08
7223 011234 004067 174112          .MOV CCR,ERROR ;ERROR IF BIT08 = 1
7224 011240 000004          .BIC #-BIT08-1,ERROR
7225 011242 036504          .JSR RO,SETEN  ;PRINT LIST OF SENTENCES
7226 011244 036572          .WORD ^D4
7227 011246 004010          .WORD SEN25    ;CONTROL REGISTER UNUSED BIT TEST
7228 011250 011204          .WORD SEN29    ;READ ONE FROM UNUSED CCR BIT08
7229 011252 011204          .JSR RO,(RO)   ;TAKE SELECTED ACTION ON ERROR
7230
7231
7232
7233
7234
7235
7236
7237
7238 011254 005267 167366    TST032:          WRITE ONE INTO UNUSED CCR BIT11 THEN READ CCR
7239 011260 012767 001015 166460    1$:             IF CCR BIT11 READ AS ONE THEN CACHE REGISTER DATA PATH ERROR
7240 011266 052767 004000 166452    .INC TID        ;UPDATE TEST ID
7241 011274 016767 166446 173240    .MOV #OFF,CCR   ;DISABLE CACHE
7242 011302 042767 173777 173232    .BIS #BIT11,CCR;WRITE 1 INTO CONTROL REGISTER BIT11
7243 011310 004067 174036    .MOV CCR,ERROR ;ERROR IF BIT11 = 1
7244 011314 000004          .BIC #-BIT11-1,ERROR
7245 011316 036504          .JSR RO,SETEN  ;PRINT LIST OF SENTENCES
7246 011320 036610          .WORD ^D4
7247 011322 004010          .WORD SEN25    ;CONTROL REGISTER UNUSED BIT TEST
7248 011324 011260          .WORD SEN30    ;READ ONE FROM UNUSED CCR BIT11
7249 011326 011260          .JSR RO,(RO)   ;TAKE SELECTED ACTION ON ERROR
7250
7251
7252
7253
7254
7255
7256
7257
7258 011330 005267 167312    TST033:          ;WRITE ONE INTO UNUSED CCR BIT14 THEN READ CCR
7259 011334 012767 001015 166404    1$:             IF CCR BIT14 READ AS ONE THEN CACHE REGISTER DATA PATH ERROR
7260 011342 052767 040000 166376    .INC TID        ;UPDATE TEST ID
7261 011350 016767 166372 173164    .MOV #OFF,CCR   ;DISABLE CACHE
7262 011356 042767 137777 173156    .BIS #BIT14,CCR;WRITE 1 INTO CONTROL REGISTER BIT14
7263 011364 004067 173762    .MOV CCR,ERROR ;ERROR IF BIT14 = 1
          .BIC #-BIT14-1,ERROR
          .JSR RO,SETEN  ;PRINT LIST OF SENTENCES

```

```

7264 011370 000004           .WORD ^D4
7265 011372 036504           .WORD SEN25
7266 011374 036626           .WORD SEN31
7267 011376 004010           JSR RO,(RO)
7268 011400 011334           .WORD 1$
7269 011402 011334           .WORD 1$
7270
7271
7272
7273
7274
7275
7276
7277
7278 011404 005267 167236     TST034:
7279 011410 012767 001015 166330 1$:
7280 011416 052767 100000 166322
7281 011424 016767 166316 173110
7282 011432 042767 077777 173102
7283 011440 004067 173706
7284 011444 000004           .WORD ^D4
7285 011446 036504           .WORD SEN25
7286 011450 036644           .WORD SEN32
7287 011452 004010           JSR RO,(RO)
7288 011454 011410           .WORD 1$
7289 011456 011410           .WORD 1$
7290
7291
7292
7293

```

```

:CONTROL REGISTER UNUSED BIT TEST
:READ ONE FROM UNUSED CCR BIT14
:TAKE SELECTED ACTION ON ERROR
:LOOP ON ERROR
:LOOP ON TEST

```

```

WRITE ONE INTO UNUSED CCR BIT15 THEN READ CCR
IF CCR BIT15 READ AS ONE THEN CACHE REGISTER DATA PATH ERROR
:UPDATE TEST ID
:DISABLE CACHE
:WRITE 1 INTO CONTROL REGISTER BIT15
:ERROR IF BIT15 = 1
:PRINT LIST OF SENTENCES
:CONTROL REGISTER UNUSED BIT TEST
:READ ONE FROM UNUSED CCR BIT15
:TAKE SELECTED ACTION ON ERROR
:LOOP ON ERROR
:LOOP ON TEST

```

.SBTTL CACHE CONTROL REGISTER BYTE TESTS

```

7295
7296
7297
7298
7299
7300
7301 011460 005267 167162
7302 011464 012767 001015 166254
7303 011472 142767 000002 166246
7304 011500 152767 000002 166241
7305 011506 016767 166234 173026
7306 011514 042767 176775 173020
7307 011522 162767 001000 173012
7308 011530 004067 173616
7309 011534 000010
7310 011536 036662
7311 011540 036676
7312 011542 036714
7313 011544 036732
7314 011546 004010
7315 011550 011464
7316 011552 011464
7317
7318
7319
7320
7321
7322
7323
7324
7325 011554 005267 167066
7326 011560 012767 001015 166160
7327 011566 142767 000004 166153
7328 011574 152767 000004 166144
7329 011602 016767 166140 172732
7330 011610 042767 177773 172724
7331 011616 162767 000004 172716
7332 011624 004067 173522
7333 011630 000010
7334 011632 036662
7335 011634 036744
7336 011636 036762
7337 011640 037000
7338 011642 004010
7339 011644 011560
7340 011646 011560
7341
7342
7343
7344

```

```

REGISTER BYTE SELECTION LOGIC TEST
WRITE ZERO INTO LOW BYTE WRITE ONE INTO HIGH BYTE
VERIFY THAT LOW BYTE DATA IS NOT EFFECTED BY WRITE TO HIGH BYTE
TST035:
1$: INC TID ;UPDATE TEST ID
MOV #OFF,CCR ;DISABLE CACHE
BICB #BIT01,CCR ;WRITE 0 INTO CONTROL REGISTER BIT01
BISB #BIT01,CCR+1 ;WRITE 1 INTO CONTROL REGISTER BIT09
MOV CCR,ERROR ;ERROR IF BIT01 = 1
BIC #-<BIT01+BIT09>-1,ERROR ;OR BIT09 = 0
SUB #1000,ERROR
JSR RO,SETEN ;PRINT LIST OF SENTENCES
WORD ^D8
WORD SEN33 ;CACHE CONTROL REGISTER BYTE TESTS
WORD SEN34 ;WROTE ZERO INTO LOW BYTE BIT01
WORD SEN35 ;WROTE ONE INTO HIGH BYTE BIT09
WORD SEN36 ;READ ZERO FROM BIT09
JSR RO,(R0) ;TAKE SELECTED ACTION ON ERROR
WORD 1$ ;LOOP ON ERROR
WORD 1$ ;LOOP ON TEST

```

```

WRITE ZERO INTO HIGH BYTE WRITE ONE INTO LOW BYTE
VERIFY HIGH BYTE NOT EFFECTED BY WRITE INTO LOW BYTE
TST036:
1$: INC TID ;UPDATE TEST ID
MOV #OFF,CCR ;DISABLE CACHE
BICB #BIT02,CCR+1 ;WRITE 0 INTO CONTROL REGISTER BIT10
BISB #BIT02,CCR ;WRITE 1 INTO CONTROL REGISTER BIT02
MOV CCR,ERROR ;ERROR IF BIT01 = 0 OR IF BIT09 = 1
BIC #-BIT02-1,ERROR
SUB #BIT02,ERROR
JSR RO,SETEN ;PRINT LIST OF SENTENCES
WORD ^D8
WORD SEN33 ;CACHE CONTROL REGISTER BYTE TESTS
WORD SEN37 ;WROTE ZERO INTO HIGH BYTE BIT10
WORD SEN38 ;WROTE ONE INTO LOW BYTE BIT02
WORD SEN39 ;READ ZERO FROM BIT02 OR READ ONE FROM BIT10
JSR RO,(R0) ;TAKE SELECTED ACTION ON ERROR
WORD 1$ ;LOOP ON ERROR
WORD 1$ ;LOOP ON TEST

```


7402
7403
7404
7405
7406
7407
7408
7409
7410
7411
7412
7413
7414
7415
7416
7417
7418
7419
7420
7421
7422
7423

012050	005267	166572	
012054	052767	000004	165666
012062	016767	165662	172452
012070	042767	177773	172444
012076	004067	173250	
012102	000004		
012104	037024		
012106	037110		
012110	004010		
012112	012054		
012114	012054		

```

:
:
:
:
TST042: 1$:      ATTEMPT TO WRITE ONE INTO UNUSED MAINT. REGISTER BIT02
:              READ INTO CMR SHOULD RESULT IN ZERO FROM BIT02
:              INC TID
:              BIS #BIT02,CMR
:              MOV CMR,ERROR
:              BIC #-BIT02-1,ERROR
:              JSR R0,SETEN
:              .WORD ^D4
:              .WORD SEN40
:              .WORD SEN44
:              JSR R0,(R0)
:              .WORD 1$
:              .WORD 1$
:              ;UPDATE TEST ID
:              ;WRITE 1 INTO MAINTENANCE REGISTER BIT02
:              ;ERROR IF BIT02 = 1
:              ;PRINT LIST OF SENTENCES
:              ;MAINTENANCE REGISTER DATA TEST
:              ;READ ONE FROM UNUSED CMR BIT02
:              ;TAKE SELECTED ACTION ON ERROR
:              ;LOOP ON ERROR
:              ;LOOP ON TEST
:
:
:
:

```

.SBTTL DATA PATH TEST

7425
7426
7427
7428
7429
7430
7431
7432 012116 005267 166524
7433 012122 004467 173766
7434 012126 012310
7435 012130 012701 046000
7436 012134 012737 000005 177746
7437 012142 012102
7438 012144 020127 050000
7439 012150 001374
7440 012152 012702 177777
7441 012156 010241
7442 012160 020127 046000
7443 012164 001374
7444 012166 005037 004542
7445 012172 005003
7446 012174 012102
7447 012176 032737 000010 177752
7448 012204 001403
7449 012206 005203
7450 012210 050237 004542
7451 012214 020127 050000
7452 012220 001365
7453 012222 012737 001015 177746
7454 012230 005703
7455 012232 001011
7456 012234 005237 004542
7457 012240 004037 005352
7458 012244 000006
7459 012246 037126
7460 012250 037136
7461 012252 037154
7462 012254 000412
7463 012256 005137 004542
7464 012262 005237 004560
7465 012266 004037 005352
7466 012272 000006
7467 012274 037126
7468 012276 037136
7469 012300 037170
7470 012302 004010
7471 012304 044000
7472 012306 044000
7473
7474
7475
7476
7477
7478
7479 012310 005267 166332
7480 012314 004467 173574

TEST WRITES ALL ONES TO HIGH CACHE THEN READS
SAME LOCATIONS. IF READ WAS A HIT FORM CACHE
THEN TEST DATA FOR ALL ONES . ANY BIT READ AS ZERO
CAUSES ERROR REPORT.

...
TST043:
INC TID ;UPDATE TEST ID
JSR R4,RELCTL ;RELOCATE TEST TO LOW CACHE
.WORD TST044
MOV #HIGHSP,R1 ;POINT TO HIGH CACHE
MOV #5,@WCCR ;ENABLE HIGH CACHE
1\$: MOV (R1)+,R2 ;TAG ALL HIGH CACHE LOCATIONS
CMP R1,#HIGHSP+2000
BNE 1\$
MOV #177777,R2 ;DATA FOR TEST
2\$: MOV R2,-(R1) ;WRITE ALL ONES TO HIGH CACHE
CMP R1,#HIGHSP
BNE 2\$
CLR @WERROR
CLR R3 ;DATA READ FROM CACHE FLAG
3\$: MOV (R1)+,R2 ;READ DATA
BIT #BIT03,@WCHR ;VERIFY DATA READ FROM CACHE
BEQ 4\$
INC R3
BIS R2,@WERROR ;OR READ DATA
4\$: CMP R1,#HIGHSP+2000 ;END OF PASS YET
BNE 3\$;NO
MOV #OFF,@WCCR ;DISABLE CACHE
TST R3
BNE 5\$;SEE IF ANY DATA READ FROM CACHE
INC @WERROR
5\$: JSR R0,@WSETEN ;PRINT LIST OF SENTENCES
.WORD *D6
.WORD SEN45 ;DATA PATH TEST
.WORD SEN46 ;WRITE ALL ONES TO HIGH CACHE
.WORD SEN47 ;NO HITS ON DATA READ
BR 6\$;
COM @WERROR
6\$: INC @WBITFLG ;PRINT FAILING BIT NUMBER
JSR R0,@WSETEN ;PRINT LIST OF SENTENCES
.WORD *D6
.WORD SEN45 ;DATA PATH TEST
.WORD SEN46 ;WRITE ALL ONES TO HIGH CACHE
.WORD SEN48 ;DATA BIT(S) READ AS ZERO
7\$: JSR R0,(R0) ;TAKE SELECTED ACTION ON ERROR
.WORD LOWSP ;LOOP ON ERROR
.WORD LOWSP ;LOOP ON TEST
...
TST044:
WRITE ALL ZEROS INTO LOW CACHE. READ AND VERIFY
ZEROS READ FORM CACHE. IF READ HIT AND BIT READ AS ONE THEN ERROR
INC TID ;UPDATE TEST ID
JSR R4,RELCTL ;RELOCATE TEST TO LOW CACHE

```

7481 012320 012502
7482 012322 012701 046000
7483 012326 012737 000005 177746
7484 012334 012102
7485 012336 020127 050000
7486 012342 001374
7487 012344 005002
7488 012346 010241
7489 012350 020127 046000
7490 012354 001374
7491 012356 005037 004542
7492 012362 005003
7493 012364 012102
7494 012366 032737 000004 177744
7495 012374 001404
7496 012376 005203
7497 012400 005102
7498 012402 050237 004542
7499 012406 020127 050000
7500 012412 001364
7501 012414 012737 001015 177746
7502 012422 005137 004542
7503 012426 005703
7504 012430 001011
7505 012432 005237 004542
7506 012436 004037 005352
7507 012442 000006
7508 012444 037126
7509 012446 037204
7510 012450 037154
7511 012452 000410
7512 012454 005237 004560
7513 012460 004037 005352
7514 012464 000006
7515 012466 037126
7516 012470 037204
7517 012472 037222
7518 012474 004010
7519 012476 044000
7520 012500 044000
7521
7522
7523
7524

```

```

      .WORD TST045
1$:  MOV #HIGHSP,R1      ;POINT TO HIGH CACHE
      MOV #5,@#CCR      ;ENABLE HIGH CACHE
      MOV (R1)+,R2      ;TAG ALL HIGH CACHE LOCATIONS
      CMP R1,#HIGHSP+2000
      BNE 1$
      CLR R2
2$:  MOV R2,-(R1)      ;DATA FOR TEST
      CMP R1,#HIGHSP    ;WRITE ALL ZEROS TO HIGH CACHE
      BNE 2$
      CLR @#ERROR
      CLR R3
3$:  MOV (R1)+,R2      ;DATA READ FROM CACHE ,FLAG
      BIT #BIT02,@#CMPE ;READ DATA
      BEQ 4$           ;VERIFY DATA READ FROM CACHE
      INC R3           ;DATA READ FROM CACHE INDICATOR
      COM R2
4$:  BIS R2,@#ERROR    ;OR ALL READ DATA
      CMP R1,#HIGHSP+2000
      BNE 3$
      MOV #OFF,@#CCR   ;DISABLE CACHE
      COM @#ERROR
      TST R3           ;SEE IF ANY DATA READ FROM CACHE
      BNE 5$
      INC @#ERROR      ;ERROR FLAG
      JSR R0,@#SETEN   ;PRINT LIST OF SENTENCES
      .WORD ^D6
      .WORD SEN45
      .WORD SEN49
      .WORD SEN47
5$:  BR 6$
      INC @#BITFLG
      JSR R0,@#SETEN
      .WORD ^D6
      .WORD SEN45
      .WORD SEN49
      .WORD SEN50
6$:  JSR R0,(R0)
      .WORD LOWSP
      .WORD LOWSP

```

```

;POINT TO HIGH CACHE
;ENABLE HIGH CACHE
;TAG ALL HIGH CACHE LOCATIONS

;DATA FOR TEST
;WRITE ALL ZEROS TO HIGH CACHE
;
;DATA READ FROM CACHE ,FLAG
;READ DATA
;VERIFY DATA READ FROM CACHE
;DATA READ FROM CACHE INDICATOR
;OR ALL READ DATA
;DISABLE CACHE
;SEE IF ANY DATA READ FROM CACHE
;ERROR FLAG
;PRINT LIST OF SENTENCES
;DATA PATH TEST
;WRITE ALL ZEROS TO HIGH CACHE
;NO HITS ON DATA READ
;PRINT ERRORING BIT(S)
;DATA PATH TEST
;WRITE ALL ZEROS TO HIGH CACHE
;DATA BIT(S) READ AS ONE
;TAKE SELECTED ACTION ON ERROR
;LOOP ON ERROR
;LOOP ON TEST

```

.SBTTL DATA PATH BIT SHORT TEST

```

7526
7527
7528
7529
7530 012502 005267 166140
7531 012506 005067 172030
7532 012512 004467 173424
7533 012516 013034
7534 012520 012703 000001
7535 012524 012737 000011 177746
7536 012532 013702 050000
7537 012536 013702 044000
7538 012542 010337 044000
7539 012546 005037 177744
7540 012552 013702 044000
7541 012556 032737 000010 177752
7542 012564 001410
7543 012566 020203
7544 012570 001071
7545 012572 006303
7546 012574 001353
7547 012576 012737 001015 177746
7548 012604 000513
7549 012606 052737 000200 177746
7550 012614 013705 177744
7551 012620 042737 000200 177746
7552 012626 032705 000040
7553 012632 001407
7554 012634 013702 050000
7555 012640 052737 002000 177746
7556 012646 013702 044000
7557 012652 032705 000100
7558 012656 001403
7559 012660 052737 000100 177746
7560 012666 110337 044000
7561 012672 042737 000100 177746
7562 012700 032705 000200
7563 012704 001403
7564 012706 052737 000100 177746
7565 012714 000303
7566 012716 110337 044001
7567 012722 000303
7568 012724 042737 002100 177746
7569 012732 013702 044000
7570 012736 010705
7571 012740 014515
7572 012742 020527 046132
7573 012746 001374
7574 012750 020203
7575 012752 001707
7576 012754 012737 001015 177746
7577 012762 012737 000001 004542
7578 012770 010337 004550
7579 012774 005037 004554
7580 013000 010237 004552
7581 013004 005237 004556

```

: ROTATE A ONE ACROSS THE DATA PATH TO VERIFY THAT EACH
: BIT CAN BE WRITTEN TO A ONE INDIVIDUALLY.
TST045: INC TID ;UPDATE TEST ID
CLR ERROR ;RESET ERROR FLAG
JSR R4,RELCTH ;RELOCATE TEST TO HIGH CACHE
.WORD TST046
1\$: MOV #1,R3 ;DATA FOR TEST
MOV #11,@#CCR ;ENABLE LOW CACHE
MOV @#LOW1,R2 ;TAG LOW CACHE BLOCK #2
MOV @#LOWSP,R2 ;TAG LOW CACHE LOCATION
MOV R3,@#LOWSP ;WRITE DATA
CLR @#CMPE ;RESET ERROR REGISTER
MOV @#LOWSP,R2 ;READ DATA
BIT #BIT03,@#CHR ;VERIFY A HIT
BEQ 2\$;NO HIT
CMP R2,R3 ;VERIFY DATA
BNE 7\$;BAD DATA
10\$: ASL R3
BNE 1\$
MOV #OFF,@#CCR ;DISABLE CACHE
BR TST046
2\$: BIS #BIT07,@#CCR ;ENABLE ABORT FOR ERROR READ
MOV @#CMPE,R5 ;SAVE ERROR IN R5
BIC #BIT07,@#CCR ;DISABLE ABORT
BIT #BIT05,R5 ;ANY TAG FAILURE
BEQ 3\$
MOV @#LOW1,R2 ;TAG LOW BLOCK #2
BIS #BIT10,@#CCR ;WRITE WRONG TAG
MOV @#LOWSP,R2 ;WRITE WRONG TAG
3\$: BIT #BIT06,R5 ;ANY LOW BYTE ERROR
BEQ 4\$
BIS #BIT06,@#CCR ;SET WRITE WRONG DATA
MOVB R3,@#LOWSP ;WRITE LOW BYTE
4\$: BIC #BIT06,@#CCR ;DISABLE WRITE WRONG DATA
BIT #BIT07,R5 ;DID BYTE FAIL
BEQ 5\$;YES
BIS #BIT06,@#CCR ;ENABLE WRITE WRONG DATA
5\$: SWAB R3 ;POS DATA FOR MOVB
MOVB R3,@#LOWSP+1 ;WRITE HIGH BYTE
SWAB R3 ;RESTORE DATA
BIC #BIT06+BIT10,@#CCR ;DISABLE WRITE WRONG
MOV @#LOWSP,R2 ;READ DATA
MOV PC,R5 ;CORRECT WRONG PARITY
6\$: MOV -(R5),(R5)
CMP R5,#3\$-TST045-16+HIGHSP
BNE 6\$
CMP R2,R3 ;VERIFY DATA
BEQ 10\$
7\$: MOV #OFF,@#CCR ;DISABLE CACHE
MOV #1,@#ERROR ;SET ERROR FLAG
MOV R3,@#GOOD ;GOOD DATA
CLR @#ADD ;CACHE ADDRESS
MOV R2,@#BAD ;BAD DATA
9\$: INC @#GOODBD ;BIT PRINT MODE

```

7582 013010 004037 005352      JSR R0,@#SETEN
7583 013014 000006      .WORD *D6
7584 013016 037236      .WORD SEN51      ;DATA PATH BIT SHORT TEST
7585 013020 037252      .WORD SEN52      ;ROTATE ONE THROUGH FIELD OF ZEROS
7586 013022 037270      .WORD SEN53      ;TESTING LOW CACHE
7587 013024 004010      JSR R0,(R0)      ;TAKE SELECTED ACTION ON ERROR
7588 013026 046004      .WORD 1$-TST045-16+HIGHSP ;LOOP ON ERROR
7589 013030 046004      .WORD 1$-TST045-16+HIGHSP ;LOOP ON TEST
7590 013032 000657      BR 10$
7591
7592
7593
7594
7595
7596
7597 013034 005267 165606      :
7598 013040 005067 171476      :
7599 013044 004467 173044      :
7600 013050 013366      :
7601 013052 012703 000001      :
7602 013056 012737 000005 177746      :
7603 013064 013702 052000      :
7604 013070 013702 046000      :
7605 013074 010337 046000      :
7606 013100 005037 177744      :
7607 013104 013702 046000      :
7608 013110 032737 000010 177752      :
7609 013116 001410      :
7610 013120 020203      :
7611 013122 001071      :
7612 013124 006303      :
7613 013126 001353      :
7614 013130 012737 001015 177746      :
7615 013136 000513      :
7616 013140 052737 000215 177746      :
7617 013146 013705 177744      :
7618 013152 012737 000005 177746      :
7619 013160 032705 000040      :
7620 013164 001407      :
7621 013166 013702 052000      :
7622 013172 052737 002000 177746      :
7623 013200 013702 046000      :
7624 013204 032705 000100      :
7625 013210 001403      :
7626 013212 052737 000100 177746      :
7627 013220 110337 046000      :

```

```

: ROTATE ONE ACROSS DATA PATH TO VERIFY THAT EACH
: BIT CAN BE WRITTEN TO A ONE INDIVIDUALLY.
TST046: INC TID ;UPDATE TEST ID
CLR ERROR ;RESET ERROR FLAG
JSR R4,RELCTL ;RELOCATE TEST TO LOW CACHE
.WORD TST047
MOV #1,R3 ;DATA FOR TEST
MOV #5,@#CCR ;ENABLE HIGH CACHE
MOV @#HIGH1,R2 ;TAG HIGH BLOCK #2
MOV @#HIGHSP,R2 ;TAG HIGH BLOCK #1
MOV R3,@#HIGHSP ;WRITE DATA TO CACHE
CLR @#CMPE ;RESET ERROR REGISTER
MOV @#HIGHSP,R2 ;READ DATA FROM CACHE
BIT #BIT03,@#CHR ;VERIFY DATA READ FROM CACHE
BEQ 2$ ;NOT READ FROM CACHE
CMP R2,R3 ;VERIFY DATA CORRECT
BNE 7$
10$: ASL R3 ;DATA FOR NEXT TEST
BNE 1$
MOV #OFF,@#CCR ;DISABLE CACHE
BR TST047 ;TEST COMPLETE
2$: BIS #215,@#CCR ;ENABLE ABORT FOR DATA READ
MOV @#CMPE,R5 ;SAVE ERROR IN R5
MOV #5,@#CCR ;DISABLE ABORT
BIT #BIT05,R5 ;ANY TAG FAILURE
BEQ 3$ ;NO
MOV @#HIGH1,R2 ;TAG HIGH BLOCK #2
BIS #BIT10,@#CCR ;ENABLE WRITE WRONG TAG
MOV @#HIGHSP,R2 ;WRITE WRONG TAG
BIT #BIT06,R5 ;ANY LOW BYTE FAILURE
BEQ 4$ ;NO
BIS #BIT06,@#CCR ;WRITE WRONG DATA
MOVB R3,@#HIGHSP ;WRITE LOW BYTE

```

```

7629 013224 042737 000100 177746      BIC #BIT06,@#CCR      ;DISABLE WRITE WRONG DATA
7630 013232 032705 000200              BIT #BIT07,R5        ;DID HIGH BYTE FAIL
7631 013236 001403              BEQ 8$                ;NO
7632 013240 052737 000100 177746      BIS #BIT06,@#CCR      ;WRITE WRONG DATA
7633 013246 000303              SWAB R3               ;POS. DATA FOR HIGH WRITE
7634 013250 110367 032525              MOV B R3,HIGHSP+1     ;WRITE HIGH BYTE
7635 013254 042737 002100 177746      BIC #BIT06+BIT10,@#CCR ;DISABLE WRITE WRONG
7636 013262 000303              SWAB R3               ;RESTORE DATA
7637 013264 013702 046000              MOV @#HIGHSP,R2      ;READ DATA
7638 013270 010705              MOV PC,R5             ;CORRECT WRONG PARITY
7639 013272 014515              MOV -(R5),(R5)        ;CAUSE TAG AND DATA WRITE
7640 013274 020527 044132              CMP R5,#3$-TST046-16+LOWSP
7641 013300 001374              BNE 6$                ;
7642 013302 020203              CMP R2,R3             ;VERIFY DATA
7643 013304 001707              BEQ 10$               ;
7644 013306 012737 001015 177746      MOV #OFF,@#CCR        ;DISABLE CACHE
7645 013314 012737 001000 004554      MOV #1000,@#ADD       ;CACHE ADDRESS
7646 013322 012737 000001 004542      MOV #1,@#ERROR        ;SET ERROR FLAG
7647 013330 010337 004550              MOV R3,@#GOOD         ;GOOD DATA
7648 013334 010237 004552              MOV R2,@#BAD          ;BAD DATA
7649 013340 005237 004556              INC @#GOODBD          ;SET DATA PRINT MODE
7650 013344 004037 005352              JSR R0,@#SETEN        ;REPORT ERROR
7651 013350 000006              .WORD ^D6
7652 013352 037236              .WORD SEN51           ;DATA PATH BIT SHORT TEST
7653 013354 037252              .WORD SEN52           ;ROTATE ONE THROUGH FIELD OF ZEROS
7654 013356 037300              .WORD SEN54           ;TESTING HIGH CACHE
7655 013360 004010              JSR R0,(R0)           ;TAKE SELECTED ACTION ON ERROR
7656 013362 044004              .WORD 1$-TST046-16+LOWSP ;LOOP ON ERROR
7657 013364 044004              .WORD 1$-TST046-16+LOWSP ;LOOP ON TEST
7658
7659
7660
7661
7662
7663
7664 013366 005267 165254      :
7665 013372 005067 171144      :
7666 013376 004467 172540      :
7667 013402 013724              :
7668 013404 012703 177776      :
7669 013410 012737 000011 177746      :
7670 013416 013702 050000              :
7671 013422 013702 044000              :
7672 013426 010337 044000              :
7673 013432 005037 177744              :
7674 013436 013702 044000              :
7675 013442 032737 000010 177752      :
7676 013450 001413              :
7677 013452 020203              :
7678 013454 001074              :
7679 013456 000261              :
7680 013460 006103              :
7681 013462 022703 177777              :
7682 013466 001350              :
7683 013470 012737 001015 177746      :
7684 013476 000512              :

```

ROTATE ZERO ACROSS DATA PATH TO VERIFY THAT EACH BIT CAN BE WRITTEN TO A ZERO INDIVIDUALLY.

TST047:

```

INC TID ;UPDATE TEST ID
CLR ERROR ;RESET ERROR FLAG
JSR R4,RELCTH ;RELOCATE TEST TO HIGH CACHE
.WORD TST050
MOV #177776,R3 ;DATA FOR TEST
MOV #11,@#CCR ;ENABLE LOW CACHE
MOV @#LOW1,R2 ;TAG LOW BLOCK #2
MOV @#LOWSP,R2 ;TAG LOW BLOCK #1
MOV R3,@#LOWSP ;WRITE TEST DATA TO CACHE
CLR @#CMPE ;RESET ERROR REGISTER
MOV @#LOWSP,R2 ;READ DATA FROM CACHE
BIT #BIT03,@#CHR ;VERIFY READ FROM CACHE
BEQ 2$ ;NOT READ FROM CACHE
CMP R2,R3 ;VERIFY DATA CORRECT
BNE 7$ ;DATA INCORRECT
SEC ;DATA FOR NEXT TEST
ROL R3
CMP #177777,R3
BNE 1$
MOV #OFF,@#CCR ;DISABLE CACHE
BR TST050 ;TEST COMPLETE

```

```

7685 013500 052737 000200 177746      2$:  BIS #BIT07,@#CCR      ;ENABLE ABORT FOR ERROR READ
7686 013506 013705 177744              MOV @#CMPE,R5          ;SAVE ERROR IN R5
7687 013512 042737 000200 177746      BIC #BIT07,@#CCR      ;DISABLE ABORT
7688 013520 032705 000040              BIT #BIT05,R5         ;ANY TAG FAILURE
7689 013524 001407                      BEQ 3$                ;NO
7690 013526 013702 050000              MOV @#LOW1,R2        ;TAG LOW BLOCK #2
7691 013532 052737 002000 177746      BIS #BIT10,@#CCR     ;WRITE WRONG TAG
7692 013540 013702 044000              MOV @#LOWSP,R2      ;WRITE WRONG TAG
7693 013544 032705 000100      3$:  BIT #BIT06,R5         ;ANY LOW BYTE ERRORS
7694 013550 001403                      BEQ 4$                ;NO
7695 013552 052737 000100 177746      BIS #BIT06,@#CCR     ;WRITE WRONG DATA
7696 013560 110337 044000      4$:  MOVB R3,@#LOWSP      ;WRITE LOW BYTE
7697 013564 042737 000100 177746      BIC #BIT06,@#CCR     ;DISABLE WRITE WRONG DATA
7698 013572 032705 000200              BIT #BIT07,R5         ;DID HIGH BYTE FAIL
7699 013576 001403                      BEQ 8$                ;NO
7700 013600 052737 000100 177746      BIS #BIT06,@#CCR     ;ENABLE WRITE WRONG DATA
7701 013606 000303      8$:  SWAB R3             ;POS. DATA FOR HIGH WRITE
7702 013610 110337 044001              MOVB R3,@#LOWSP+1   ;WRITE HIGH BYTE
7703 013614 042737 002100 177746      BIC #BIT06+BIT10,@#CCR ;DISABLE WRITE WRONG
7704 013622 000303                      SWAB R3
7705 013624 013702 044000              MOV @#LOWSP,R2      ;READ DATA FROM CACHE
7706 013630 010705              MOV PC,R5           ;CORRECT WRONG WRITTEN PARITY
7707 013632 014515      6$:  MOV -(R5),(R5)      ;CAUSE WRITE TAG AND DATA
7708 013634 020527 044140              CMP R5,#3$-TST047-16+LOWSP
7709 013640 001374                      BNE 6$
7710 013642 020203              CMP R2,R3           ;VERIFY DATA
7711 013644 001704                      BEQ 10$
7712 013646 012737 001015 177746      7$:  MOV #OFF,@#CCR      ;DISABLE CACHE
7713 013654 005037 004554              CLR @#ADD           ;CACHE ADDRESS
7714 013660 012737 000001 004542      MOV #1,@#ERROR      ;SET ERROR FLAG
7715 013666 010337 004550              MOV R3,@#GOOD       ;GOOD DATA
7716 013672 010237 004552              MOV R2,@#BAD        ;BAD DATA
7717 013676 005237 004556              INC @#GOODBD        ;SET DATA PRINT MODE
7718 013702 004037 005352              JSR R0,@#SETEN      ;REPORT ERROR
7719 013706 000006              .WORD 'D6
7720 013710 037236              .WORD SEN51         ;DATA PATH BIT SHORT TEST
7721 013712 037310              .WORD SEN55         ;ROTATE ZERO THROUGH A FIELD OF ONES
7722 013714 037270              .WORD SEN53         ;TESTING LOW CACHE
7723 013716 004010              JSR R0,(R0)         ;TAKE SELECTED ACTION ON ERROR
7724 013720 046004              .WORD 1$-TST047-16+HIGHSP ;LOOP ON ERROR
7725 013722 046052              .WORD 10$-TST047-16+HIGHSP
7726
7727
7728
7729
7730
7731
7732 013724 005267 164716      :
7733 013730 005067 170606      :
7734 013734 004467 172154      :
7735 013740 014262              :
7736 013742 012703 177776      :
7737 013746 012737 000005 177746      :
7738 013754 013702 052000              :
7739 013760 013702 046000              :
7740 013764 010337 046000              :
:
:
:
TST050:
INC TID ;UPDATE TEST ID
CLR ERROR ;RESET ERROR FLAG
JSR R4,RELCTL ;RELOCATE TEST TO LOW CACHE
.WORD TST051
MOV #177776,R3 ;DATA FOR TEST
MOV #5,@#CCR ;ENABLE HIGH CACHE
MOV @#HIGH1,R2 ;TAG HIGH BLOCK #2
MOV @#HIGHSP,R2 ;TAG HIGH BLOCK #1
MOV R3,@#HIGHSP ;WRITE TEST DATA INTO CACHE

```

7741	013770	005037	177744		CLR @#CMPE	:RESET ERROR REGISTER
7742	013774	013702	046000		MOV @#HIGHSP,R2	:READ DATA FROM CACHE
7743	014000	032737	000010	177752	BIT #BIT03,@#CHR	:VERIFY DATA READ FROM CACHE
7744	014006	001413			BEQ 2\$:NOT READ FROM CACHE
7745	014010	020203			CMP R2,R3	:VERIFY DATA READ IS CORRECT
7746	014012	001074			BNE 7\$:DATA IN ERROR
7747	014014	000261			10\$: SEC	:DATA FOR NEXT TEST
7748	014016	006103			ROL R3	
7749	014020	022703	177777		CMP #177777,R3	:INDICATES TEST COMPLETE
7750	014024	001350			BNE 1\$	
7751	014026	012737	001015	177746	MOV #OFF,@#CCR	:DISABLE CACHE
7752	014034	000512			BR TST051	:TEST COMPLETE
7753	014036	052737	000215	177746	2\$: BIS #215,@#CCR	:ENABLE ABORT FOR ERROR READ
7754	014044	013705	177744		MOV @#CMPE,R5	:SAVE ERROR IN R5
7755	014050	012737	000005	177746	MOV #5,@#CCR	:DISABLE ABORT
7756	014056	032705	000040		BIT #BIT05,R5	:ANY TAG FAILURE
7757	014062	001407			BEQ 3\$	
7758	014064	013702	052000		MOV @#HIGH1,R2	:TAG HIGH BLOCK #2
7759	014070	052737	002000	177746	BIS #BIT10,@#CCR	:WRITE WRONG TAG
7760	014076	013702	046000		MOV @#HIGHSP,R2	:WRITE WRONG TAG
7761	014102	032705	000100		3\$: BIT #BIT06,R5	:ANY LOW BYTE ERRORS
7762	014106	001403			BEQ 4\$	
7763	014110	052737	000100	177746	BIS #BIT06,@#CCR	:WRITE WRONG DATA
7764	014116	110337	046000		4\$: MOVB R3,@#HIGHSP	:WRITE LOW BYTE
7765	014122	042737	000100	177746	BIC #BIT06,@#CCR	:DISABLE WRITE WRONG DATA
7766	014130	032705	000200		BIT #BIT07,R5	:DID HIGH BYTE FAIL
7767	014134	001403			BEQ 8\$	
7768	014136	052737	000100	177746	BIS #BIT06,@#CCR	:DISABLE WRITE WRONG DATA
7769	014144	000303			8\$: SWAB R3	:POS DATA FOR HIGH WRITE
7770	014146	110337	046001		MOVB R3,@#HIGHSP+1	:WRITE HIGH BYTE
7771	014152	042737	002100	177746	BIC #BIT06+BIT10,@#CCR	:DISABLE WRITE WRONG
7772	014160	000303			SWAB R3	:RESTORE DATA
7773	014162	013702	046000		MOV @#HIGHSP,R2	:READ DATA
7774	014166	010705			MOV PC,R5	:CORRECT WRONG WRITTEN PARITY
7775	014170	014515			6\$: MOV -(R5),(R5)	:CAUSE WRITE TAG AND DATA
7776	014172	020527	044140		CMP R5,#3\$-TST050-16+LOWSP	
7777	014176	001374			BNE 6\$	
7778	014200	020203			CMP R2,R3	:VERIFY DATA
7779	014202	001704			BEQ 10\$	
7780	014204	012737	001015	177746	7\$: MOV #OFF,@#CCR	:DISABLE CACHE
7781	014212	005037	004554		CLR @#ADD	:CACHE ADDRESS
7782	014216	012737	000002	004542	MOV #2,@#ERROR	:SET ERROR FLAG
7783	014224	010337	004550		MOV R3,@#GOOD	:GOOD DATA
7784	014230	010237	004552		MOV R2,@#BAD	:BAD DATA
7785	014234	005237	004556		INC @#GOODBD	:SET DATA PRINT MODE
7786	014240	004037	005352		JSR R0,@#SETEN	:REPORT ERROR
7787	014244	000006			.WORD ^D6	
7788	014246	037236			.WORD SEN51	:DATA PATH BIT SHORT TEST
7789	014250	037310			.WORD SEN55	:ROTATE ZERO THROUGH A FIELD OF ONES
7790	014252	037330			.WORD SEN56	:TESTING IN HIGH CACHE
7791	014254	004010			JSR R0,(R0)	:TAKE SELECTED ACTION ON ERROR
7792	014256	044004			.WORD 1\$-TST050-16+LOWSP	
7793	014260	044056			.WORD 10\$-TST050-12+LOWSP	
7794						
7795						
7796						

```

7798                                     .SBTTL CACHE FLUSH TESTS
7799                                     VERIFY FLUSH IN PROGRESS BIT WILL SET AS RESULT
7800                                     OF FLUSH
7801 014262 005267 164360                TST051: INC TID                ;UPDATE TEST ID
7802 014266 052737 000400 177746        4$:   BIS #BIT08,@#CCR        ;CAUSE CACHE FLUSH
7803 014274 032737 010000 177746        BIT #BIT12,@#CCR        ;VERIFY FLUSH IN PROGRESS
7804 014302 001413                        BEQ 1$                    ;
7805 014304 005067 170232                CLR ERROR                ;RESET ERROR FLAG
7806 014310 005002                        3$:   CLR R2                ;WAIT FOR FLUSH TO COMPLETE
7807 014312 005302                        2$:   DEC R2                ;WAIT LOOP
7808 014314 001376                        BNE 2$                    ;
7809 014316 004010                        JSR RO,(R0)              ;TAKE SELECTED ACTION ON ERROR
7810 014320 014266                        .WORD 4$                 ;LOOP ON ERROR
7811 014322 014266                        .WORD 4$                 ;LOOP ON TEST
7812 014324 005067 170534                CLR LOPERR               ;RESET LOOP ON TEST FLAG
7813 014330 000412                        BR TST052               ;NEXT TEST
7814 014332 012767 000001 170202        1$:   MOV #1,ERROR          ;SET ERROR FLAG
7815 014340 004067 171006                JSR RO,SETEN            ;REPORT ERROR
7816 014344 000006                        .WORD ^D6                ;
7817 014346 037342                        .WORD SEN57              ;CACHE FLUSH TESTS
7818 014350 037352                        .WORD SEN58              ;FLUSH IN PROGRESS BIT FAILED TO SET
7819 014352 037372                        .WORD SEN59              ;AS RESULT OF SETTING CACHE FLUSH BIT
7820 014354 000755                        BR 3$                    ;
7821
7822
7823
7824
7825                                     :
7826                                     VERIFY FLUSH IN PROGRESS BIT WILL RESET ON COMPLETION
7827                                     OF FLUSH.
7828 014356 005267 164264                TST052: INC TID                ;UPDATE TEST ID
7829 014362 005002                        3$:   CLR R2                ;RESET DELAY COUNTER
7830 014370 052767 000400 163350        CLR ERROR                ;RESET ERROR FLAG
7831 014376 032767 010000 163342        2$:   BIS #BIT08,CCR        ;START FLUSH CYCLE
7832 014404 001413                        BIT #BIT12,CCR          ;SEE IF FLUSH COMPLETE
7833 014406 005302                        BEQ 1$                    ;FLUSH COMPLETE
7834 014410 001372                        DEC R2                    ;SEE IF TIME HAS RUN OUT
7835 014412 012767 000001 170122        BNE 2$                    ;NOT YET
7836 014420 004067 170726                MOV #1,ERROR            ;SET ERROR FLAG
7837 014424 000006                        JSR RO,SETEN            ;REPORT ERROR
7838 014426 037342                        .WORD ^D6                ;
7839 014430 037412                        .WORD SEN57              ;CACHE FLUSH TESTS
7840 014432 037430                        .WORD SEN60              ;FLUSH IN PROGRESS FAILED TO CLEAR
7841 014434 004010                        .WORD SEN61              ;TIME FOR FLUSH TO COMPLETE RAN OUT
7842 014436 014362                        1$:   JSR RO,(R0)          ;TAKE SELECTED ACTION ON ERROR
7843 014440 014362                        .WORD 3$                 ;LOOP ON ERROR
7844 014442 005067 170416                .WORD 3$                 ;LOOP ON TEST
7845                                     CLR LOPERR               ;RESET LOOP ON ERROR FLAG
7846
7847
7848
7849                                     :
7850                                     VERIFY VALID SET IN USE BIT WILL CHANGE AS RESULT
7851                                     OF CACHE FLUSH.
7852 014446 005267 164174                TST053: INC TID                ;UPDATE TEST ID FLAG
7853 014452 005067 170064                6$:   CLR ERROR            ;RESET ERROR FLAG
7853 014456 032767 020000 163262        BIT #BIT13,CCR          ;SELECT VALID BITS SET A

```

7854	014464	001412				BEQ 1\$:ON SET A NOW
7855	014466	005002				CLR R2	:RESET TIME OUT COUNTER
7856	014470	052767	000400	163250		BIS #BIT08,CCR	:CAUSE FLUSH
7857	014476	032767	010000	163242	3\$:	BIT #BIT12,CCR	:WAIT FOR FLUSH TO COMPLETE
7858	014504	001402				BEQ 1\$:FLUSH COMPLETE
7859	014506	005302				DEC R2	:WAIT LOOP
7860	014510	001372				BNE 3\$	
7861	014512	052767	000400	163226	1\$:	BIS #BIT08,CCR	:CAUSE FLUSH
7862	014520	005002				CLR R2	:RESET WAIT LOOP COUNTER
7863	014522	032767	010000	163216	4\$:	BIT #BIT12,CCR	:WAIT FOR FLUSH TO COMPLETE
7864	014530	001402				BEQ 2\$	
7865	014532	005302				DEC R2	:OR TIME TO RUN OUT
7866	014534	001372				BNE 4\$	
7867	014536	016767	163204	167776	2\$:	MOV CCR,ERROR	:VALID SET B SELECTED
7868	014544	032767	020000	167770		BIT #BIT13,ERROR	:SHOULD BE SET
7869	014552	001406				BEQ 5\$:ERROR
7870	014554	005067	167762			CLR ERROR	
7871	014560	004010			7\$:	JSR R0,(R0)	:TAKE SELECTED ACTION ON ERROR
7872	014562	014452				.WORD 6\$:LOOP ON ERROR
7873	014564	014452				.WORD 6\$:LOOP ON TEST
7874	014566	000413				BR TST054	:NEXT TEST
7875	014570	005267	167746		5\$:	INC ERROR	:SET ERROR FLAG
7876	014574	004067	170552			JSR R0,SETEN	:REPORT ERROR
7877	014600	000006				.WORD ^D6	
7878	014602	037342				.WORD SEN57	:CACHE FLUSH TESTS
7879	014604	037450				.WORD SEN62	:VALID BIT IN USE BIT DID NOT SET AS
7880	014606	037474				.WORD SEN63	:RESULT OF FLUSH
7881	014610	005067	170250			CLR LOPERR	:RESET LOOP ON TEST FLAG
7882	014614	000761				BR 7\$	
7883							
7884							
7885							
7886							
7887							
7888							
7889	014616	005267	164024		:	VERIFY VALID SET IN USE BIT WILL CHANGE AS RESULT	
7890	014622	005002			:TST054:	OF CACHE FLUSH.	
7891	014624	005067	167712		4\$:	INC TID	:UPDATE TEST ID
7892	014630	052767	000400	163110		CLR R2	:RESET FLUSH TIME OUT COUNTER
7893	014636	032767	010000	163102	1\$:	CLR ERROR	:RESET ERROR FLAG
7894	014644	001402				BIS #BIT08,CCR	:FLUSH CACHE
7895	014646	005302				BIT #BIT12,CCR	:WAIT TILL COMPLETE
7896	014650	001372				BEQ 2\$:FLUSH COMPLETE
7897	014652	032767	020000	163066	2\$:	DEC R2	:FLUSH TIME OUT COUNTER
7898	014660	001410				BNE 1\$	
7899	014662	005267	167654			BIT #BIT13,CCR	:VALID SET A SHOULD BE INUSE
7900	014666	004067	170460			BEQ 3\$	
7901	014672	000006				INC ERROR	:SET ERROR FLAG
7902	014674	037342				JSR R0,SETEN	:REPORT ERROR
7903	014676	037504				.WORD ^D6	
7904	014700	037524				.WORD SEN57	:CACHE FLUSH TESTS
7905	014702	004010				.WORD SEN64	:VALID SET INUSE BIT FAILED TO CLEAR
7906	014704	014712				.WORD SEN65	:AFTER FLUSH
7907	014706	014712			3\$:	JSR R0,(R0)	:TAKE SELECTED ACTION ON ERROR
7908	014710	000416				.WORD 5\$:LOOP ON ERROR
7909	014712	032767	020000	163026	5\$:	.WORD 5\$:LOOP ON TEST
						BR TST055	
						BIT #BIT13,CCR	:SELECT VALID SET B


```

7966 015210 037342
7967 015212 037532
7968 015214 037546
7969 015216 004010
7970 015220 046000
7971 015222 046000
7972 015224 000426
7973 015226 012702 052000
7974 015232 005722
7975 015234 001776
7976 015236 005037 004542
7977 015242 005237 004542
7978 015246 005722
7979 015250 001374
7980 015252 006337 004542
7981 015256 005237 004560
7982 015262 004037 005352
7983 015266 000010
7984 015270 037342
7985 015272 037532
7986 015274 037546
7987 015276 037570
7988 015300 000746
7989
7990
7991
7992
7993

```

```

14$: .WORD SEN57
      .WORD SEN66
      .WORD SEN67
      JSR R0,(R0)
      .WORD HIGHSP
      .WORD HIGHSP
      BR TST056
11$: MOV #HIGH1,R2
12$: TST (R2)+
      BEQ 12$
13$: CLR @#ERROR
      INC @#ERROR
      TST (R2)+
      BNE 13$
      ASL @#ERROR
      INC @#BITFLG
      JSR R0,@#SETEN
      .WORD ^D8
      .WORD SEN57
      .WORD SEN66
      .WORD SEN67
      .WORD SEN68
      BR 14$

```

```

: CACHE FLUSH TESTS
: FLUSH FAILED TO INVALIDATE CACHE
: TESTING LOW CACHE USING VALID BITS SET A
: TAKE SELECTED ACTION ON ERROR
: LOOP ON ERROR
: LOOP ON TEST
: NO ACTION
: ERROR FLAGS POINTER
: LOOK FOR FIRST ERROR
: RESET ERROR FLAG
: FIND ERRORING BIT
: FIND NO. OF BITS
: FAILING BIT
: BIT PRINT MODE
: REPORT ERROR
: CACHE FLUSH TESTS
: FLUSH FAILED TO INVALIDATE CACHE
: TESTING LOW CACHE USING VALID BITS SET A
: POSSIBLE FLUSH COUNTER BIT FAILURE

```

```

7994 015302 005267 163340
7995 015306 004467 170602
7996 015312 015636
7997 015314 005037 004542
7998 015320 032737 020000 177746
7999 015326 001407
8000 015330 052737 000400 177746
8001 015336 032737 010000 177746
8002 015344 001374
8003 015346 012737 000005 177746
8004 015354 012702 046000
8005 015360 012203
8006 015362 020227 050000
8007 015366 001374
8008 015370 052737 000400 177746
8009 015376 032737 010000 177746
8010 015404 001374
8011 015406 052737 000400 177746
8012 015414 032737 010000 177746
8013 015422 001374
8014 015424 012702 050000
8015 015430 012703 046000
8016 015434 005037 004550
8017 015440 005037 004552
8018 015444 012305
8019 015446 032737 000010 177752
8020 015454 001410
8021 015456 005237 004552

```

: TST056: VERIFY FLUSH COUNTER LOGIC BY EXAMINING FLUSH FAILURE

```

1$: INC TID
      JSR R4,RELCTL
      .WORD TST057
      CLR @#ERROR
      BIT #BIT13,@#CCR
      BEQ 3$
      BIS #BIT08,@#CCR
      BIT #BIT12,@#CCR
      BNE 2$
2$: MOV #5,@#CCR
      MOV #HIGHSP,R2
4$: MOV (R2)+,R3
      CMP R2,#HIGHSP+2000
      BNE 4$
5$: BIS #BIT08,@#CCR
      BIT #BIT12,@#CCR
      BNE 5$
6$: BIS #BIT08,@#CCR
      BIT #BIT12,@#CCR
      BNE 6$
7$: MOV #LOW1,R2
      MOV #HIGHSP,R3
      CLR @#GOOD
      CLR @#BAD
      MOV (R3)+,R5
      BIT #BIT03,@#CHR
      BEQ 8$
      INC @#BAD

```

```

: UPDATE TEST ID
: RELOCATE TEST TO LOW CACHE
: RESET ERPOR FLAG
: SELECT VALID BITS SET A
: SET A INUSE NOW
: FLUSH CACHE
: WAIT TILL COMPLETE
: ENABLE HIGH CACHE
: HIGH BLOCK #1 POINTER
: TAG LOCATION
: TAG ALL OF HIGH BLOCK
: FLUSH CACHE SELECT VALID SET B
: WAIT TILL COMPLETE
: FLUSH CACHE SELECT VALID A
: WAIT TILL COMPLETE
: ERROR LOG BLOCK
: PREV. TAGGED LOW BLOCK
: SUCCESSFUL VALID BIT CLEAR COUNT
: UNSUCCESSFUL VALID BIT CLEAR COUNT
: READ FROM HIGH BLOCK #1
: LOOK FOR READ HIT
: NO HIT
: READ HIT FROM FLUSHED CACHE

```

8022	015462	012722	000001		MOV #1,(R2)+	:SET ERROR FLAG FOR LOCATION
8023	015466	020327	050000		9\$: CMP R3,#HIGHSP+2000	:REPEAT FOR ALL OF HIGH BLOCK
8024	015472	001364			BNE 7\$:NOT COMPLETE
8025	015474	000404			BR 10\$:COMPLETE
8026	015476	005237	004550		8\$: INC @#GOOD	:NO READ FROM FLUSHED CACHE
8027	015502	005022			CLR (R2)+	:CLEAR ERROR FLAG FOR LOCATION
8028	015504	000770			BR 9\$	
8029	015506	012737	001015	177746	10\$: MOV #OFF,@#CCR	:DISABLE CACHE
8030	015514	005737	004552		TST @#BAD	:ANY ERRORS
8031	015520	001446			BEQ TST057	:NO
8032	015522	023737	004550	004552	CMP @#GOOD,@#BAD	:IS IT FLUSH COUNTER ERROR
8033	015530	001414			BEQ 11\$:LOOKS LIKE
8034	015532	005237	004542		INC @#ERROR	:SET ERROR FLAG
8035	015536	004037	005352		JSR R0,@#SETEN	:REPORT ERROR
8036	015542	000006			.WORD ^D6	
8037	015544	037342			.WORD SEN57	:CACHE FLUSH TESTS
8038	015546	037532			.WORD SEN66	:FLUSH FAILED TO INVALIDATE CACHE
8039	015550	037604			.WORD SEN69	:TESTING HIGH CACHE USING VALID BITS SET A
8040	015552	004010			14\$: JSR R0,(R0)	:TAKE SELECTED ACTION ON ERROR
8041	015554	044000			.WORD LOWSP	:LOOP ON ERROR
8042	015556	044000			.WORD LOWSP	:LOOP ON TEST
8043	015560	000426			BR TST057	:NO ACTION
8044	015562	012702	050000		11\$: MOV #LOW1,R2	:ERROR FLAGS POINTER
8045	015566	005722			12\$: TST (R2)+	:LOOK FOR FIRST ERROR
8046	015570	001776			BEQ 12\$	
8047	015572	005037	004542		CLR @#ERROR	:RESET ERROR FLAG
8048	015576	005237	004542		13\$: INC @#ERROR	:FIND ERRORING BIT
8049	015602	005722			TST (R2)+	:FIND NO. OF BITS
8050	015604	001374			BNE 13\$	
8051	015606	006337	004542		ASL @#ERROR	:FAILING BIT
8052	015612	005237	004560		INC @#BITFLG	:BIT PRINT MODE
8053	015616	004037	005352		JSR R0,@#SETEN	:REPORT ERROR
8054	015622	000010			.WORD ^D8	
8055	015624	037342			.WORD SEN57	:CACHE FLUSH TESTS
8056	015626	037532			.WORD SEN66	:FLUSH FAILED TO INVALIDATE CACHE
8057	015630	037626			.WORD SEN70	:TESTING HIGH CACHE USING LID BITS SET A
8058	015632	037570			.WORD SEN68	:POSSIBLE FLUSH COUNTER BIT FAILURE
8059	015634	000746			BR 14\$	
8060						
8061						
8062						
8063						
8064						
8065	015636	005267	163004		VERIFY FLUSH COUNTER LOGIC BY EXAMINING FLUSH FAILURE	
8066	015642	004467	170274		INC TID ;UPDATE TEST ID	
8067	015646	016170			1\$: JSR R4,RELCH	:RELOCATE TEST TO HIGH CACHE
8068	015650	005037	004542		.WORD TST060	
8069	015654	032737	020000	177746	CLR @#ERROR	:RESET ERROR FLAG
8070	015662	001007			BIT #BIT13,@#CCR	:SELECT VALID BITS SET B
8071	015664	052737	000400	177746	BNE 3\$:SET B IN USE NOW
8072	015672	032737	010000	177746	BIS #BIT08,@#CCR	:FLUSH CACHE
8073	015700	001374			2\$: BIT #BIT12,@#CCR	:WAIT TILL COMPLETE
8074	015702	012737	000011	177746	BNE 2\$	
8075	015710	012702	044000		3\$: MOV #11,@#CCR	:ENABLE LOW CACHE
8076	015714	012203			MOV #LOWSP,R2	:LOW BLOCK #1 POINTER
8077	015716	020227	046000		4\$: MOV (R2)+,R3	:TAG LOCATION
					CMP R2,#LOWSP+2000	:TAG ALL OF LOW BLOCK

8078	015722	001374				BNE 4\$	
8079	015724	052737	000400	177746		BIS #BIT08,@#CCR	:FLUSH CACHE SELECT VALID A
8080	015732	032737	010000	177746	5\$:	BIT #BIT12,@#CCR	:WAIT TILL COMPLETE
8081	015740	001374				BNE 5\$	
8082	015742	052737	000400	177746		BIS #BIT08,@#CCR	:FLUSH CACHE SELECT VALID B
8083	015750	032737	010000	177746	6\$:	BIT #BIT12,@#CCR	:WAIT TILL COMPLETE
8084	015756	001374				BNE 6\$	
8085	015760	012702	052000			MOV #HIGH1,R2	:ERROR LOG BLOCK
8086	015764	012703	044000			MOV #LOWSP,R3	:PREV. TAGGED LOW BLOCK
8087	015770	005037	004550			CLR @#GOOD	:SUCCESSFUL VALID BIT CLEAR COUNT
8088	015774	005037	004552			CLR @#BAD	:UNSUCCESSFUL VALID BIT CLEAR COUNT
8089	016000	012305			7\$:	MOV (R3)+,R5	:READ FROM LOW BLOCK #1
8090	016002	032737	000010	177752		BIT #BIT03,@#CHR	:LOOK FOR READ HIT
8091	016010	001410				BEQ 8\$:NO HIT
8092	016012	005237	004552			INC @#BAD	:READ HIT FROM FLUSH CACHE
8093	016016	012722	000001			MOV #1,(R2)+	:SET ERROR FLAG FOR LOCATION
8094	016022	020327	046000		9\$:	CMP R3,#LOWSP+2000	:REPEAT FOR ALL LOW CACHE
8095	016026	001364				BNE 7\$:NOT COMPLETE
8096	016030	000404				BR 10\$:COMPLETE
8097	016032	005237	004550		8\$:	INC @#GOOD	:NO READ HIT FROM FLUSHED CACHE
8098	016036	005022				CLR (R2)+	:SET ERROR FLAG FOR LOCATION
8099	016040	000770				BR 9\$	
8100	016042	012737	001015	177746	10\$:	MOV #OFF,@#CCR	:DISABLE CACHE
8101	016050	005737	004552			TST @#BAD	:ANY ERRORS
8102	016054	001445				BEQ TST060	
8103	016056	023737	004550	004552		CMP @#GOOD,@#BAD	:IS IT FLUSH COUNTER ERROR
8104	016064	001414				BEQ 11\$:LOOKS LIKE
8105	016066	005237	004542			INC @#ERROR	:SET ERROR FLAG
8106	016072	004037	005352			JSR R0,@#SETEN	:REPORT ERROR
8107	016076	000006				.WORD ^D6	
8108	016100	037342				.WORD SEN57	:CACHE FLUSH TESTS
8109	016102	037532				.WORD SEN66	:FLUSH FAILED TO INVALIDATE CACHE
8110	016104	037650				.WORD SEN71	:TESTING LOW CACHE USING VALID BITS SET B
8111	016106	004010			14\$:	JSR R0,(R0)	:TAKE SELECTED ACTION ON ERROR
8112	016110	046000				.WORD HIGHSP	:LOOP ON ERROR
8113	016112	046000				.WORD HIGHSP	:LOOP ON TEST
8114	016114	000425				BR TST060	
8115	016116	012702	052000		11\$:	MOV #HIGH1,R2	:ERROR FLAGS POINTER
8116	016122	005722			12\$:	TST (R2)+	:LOOK FOR FIRST ERROR
8117	016124	001776				BEQ 12\$	
8118	016126	005037	004542			CLR @#ERROR	:RESET ERROR FLAG
8119	016132	005237	004542		13\$:	INC @#ERROR	:FIND ERRORING BIT
8120	016136	005722				TST (R2)+	:FIND NO. OF BITS
8121	016140	001374				BNE 13\$	
8122	016142	006337	004542			ASL @#ERROR	:FAILING BIT
8123	016146	005237	004560			INC @#BITFLG	:BIT PRINT MODE
8124	016152	004037	005352			JSR R0,@#SETEN	:REPORT ERROR
8125	016156	000006				.WORD ^D6	
8126	016160	037342				.WORD SEN57	:CACHE FLUSH TESTS
8127	016162	037532				.WORD SEN66	:FLUSH FAILED TO INVALIDATE CACHE
8128	016164	037570				.WORD SEN68	:POSSIBLE FLUSH COUNTER BIT FAILURE
8129	016166	000747				BR 14\$	
8130							
8131							
8132							
8133							

8134					TST060: VERIFY FLUSH COUNTER LOGIC BY EXAMINING FLUSH FAILURE
8135	016170	005267	162452		INC TID ;UPDATE TEST ID
8136	016174	004467	167714		JSR R4,RELCTL ;RELOCATE CACHE TO LOW CACHE
8137	016200	016524			.WORD TST061
8138	016202	005037	004542		CLR @#ERROR ;RESET ERROR FLAG
8139	016206	032737	020000	177746	BIT #BIT13,@#CCR ;SELECT VALID BITS SET B
8140	016214	001007			BNE 3\$
8141	016216	052737	000400	177746	BIS #BIT08,@#CCR ;FLUSH CACHE
8142	016224	052737	010000	177746	2\$: BIS #BIT12,@#CCR ;WAIT TILL COMPLETE
8143	016232	001374			BNE 2\$
8144	016234	012737	000005	177746	3\$: MOV #5,@#CCR ;ENABLE HIGH CACHE
8145	016242	012702	046000		MOV #HIGHSP,R2 ;HIGH BLOCK #1 POINTER
8146	016246	012203			4\$: MOV (R2)+,R3 ;TAG LOCATION
8147	016250	020227	050000		CMP R2,#HIGHSP+2000 ;TAG ALL OF HIGH BLOCK
8148	016254	001374			BNE 4\$
8149	016256	052737	000400	177746	BIS #BIT08,@#CCR ;FLUSH CACHE SELECT VALID A
8150	016264	032737	010000	177746	5\$: BIT #BIT12,@#CCR ;WAIT TILL COMPLETE
8151	016272	001374			BNE 5\$
8152	016274	052737	000400	177746	BIS #BIT08,@#CCR ;FLUSH CACHE SELECT VALID B
8153	016302	032737	010000	177746	6\$: BIT #BIT12,@#CCR ;WAIT TILL COMPLETE
8154	016310	001374			BNE 6\$
8155	016312	012702	050000		MOV #LOW1,R2 ;ERROR LOG BLOCK
8156	016316	012703	046000		MOV #HIGHSP,R3 ;PREV TAGGED LOW BLOCK
8157	016322	005037	004550		CLR @#GOOD ;SUCCESSFUL VALID BIT CLEAR COUNT
8158	016326	005037	004552		CLR @#BAD ;UNSUCCESSFUL VALID BIT CLEAR COUNT
8159	016332	012305			7\$: MOV (R3)+,R5 ;READ FROM HIGH BLOCK #1
8160	016334	032737	000010	177752	BIT #BIT03,@#CHR ;LOOK FOR READ HIT
8161	016342	001410			BEQ 8\$;NO HIT
8162	016344	005237	004552		INC @#BAD ;READ HIT FROM FLUSHED CACHE
8163	016350	012722	000001		MOV #1,(R2)+ ;SET ERROR FLAG FOR LOCATION
8164	016354	020327	050000		9\$: CMP R3,#HIGHSP+2000 ;REPEAT FOR ALL OF HIGH BLOCK
8165	016360	001364			BNE 7\$;NOT COMPLETE
8166	016362	000404			BR 10\$;COMPLETE
8167	016364	005237	004550		8\$: INC @#GOOD ;NO READ HIT FROM FLUSHED CACHE
8168	016370	005022			CLR (R2)+ ;CLEAR ERROR FLAGS FOR LOCATION
8169	016372	000770			BR 9\$
8170	016374	012737	001015	177746	10\$: MOV #OFF,@#CCR ;DISABLE CACHE
8171	016402	005737	004552		TST @#BAD ;ANY ERRORS
8172	016406	001446			BEQ TST061
8173	016410	023737	004550	004552	CMP @#GOOD,@#BAD ;IS IT FLUSH IN ERROR
8174	016416	001414			BEQ 11\$
8175	016420	005237	004542		INC @#ERROR ;LOOKS LIKE
8176	016424	004037	005352		JSR R0,@#SETEN ;REPORT ERROR
8177	016430	000006			.WORD *D6
8178	016432	037342			.WORD SEN57
8179	016434	037532			.WORD SEN66
8180	016436	037672			.WORD SEN72
8181	016440	004010			14\$: JSR R0,(R0)
8182	016442	044000			.WORD LOWSP
8183	016444	044000			.WORD LOWSP
8184	016446	000426			BR TST061
8185	016450	012702	050000		11\$: MOV #LOW1,R2
8186	016454	005722			12\$: TST (R2)+ ;ERROR FLAGS POINTER
8187	016456	001776			BEQ 12\$;LOOK FOR FIRST ERROR
8188	016460	005037	004542		CLR @#ERROR ;RESET ERROR FLAG
8189	016464	005237	004542		13\$: INC @#ERROR ;FIND ERRORING BIT

8190 016470 005722
8191 016472 001374
8192 016474 006337 004542
8193 016500 005237 004560
8194 016504 004037 005352
8195 016510 000010
8196 016512 037342
8197 016514 037532
8198 016516 037672
8199 016520 037570
8200 016522 000746
8201
8202
8203
8204

TST (R2)+
BNE 13\$
ASL @#ERROR ;
INC @#BITFLG
JSR RO,@#SETEN
.WORD ^D8
.WORD SEN57
.WORD SEN66
.WORD SEN72
.WORD SEN68
BR 14\$

;FIND NO. OF BITS
;FAILING BIT
;BIT PRINT MODE
;REPORT ERROR
;CACHE FLUSH TESTS
;FLUSH FAILED TO INVALIDATE CACHE
;TESTING HIGH CACHE USING VALID BITS SET B
;POSSIBLE FLUSH COUNTER BIT FAILURE


```

8262
8263
8264
8265
8266
8267 016712 005267 161730
8268 016716 004467 167220
8269 016722 017100
8270 016724 005037 004704
8271 016730 005037 004702
8272 016734 012702 052000
8273 016740 005001
8274 016742 005037 177546
8275 016746 042737 001004 177746
8276 016754 105737 177546
8277 016760 001404
8278 016762 005237 004702
8279 016766 005037 177546
8280 016772 021212
8281 016774 021212
8282 016776 021212
8283 017000 021212

:
:
:
:
:
TST062:
      INC TID
      JSR R4,RELCTH
      .WORD TST063
3$:   CLR @#COUNT
      CLR @#TIME
      MOV #HIGH1,R2
      CLR R1
      CLR @#KOOKOO
      BIC #BIT02+BIT09,@#CCR
4$:   YSTB @#KOOKOO
      BEQ 10$
      INC @#TIME
      CLR @#KOOKOO
10$:  CMP (R2),(R2)
      CMP (R2),(R2)
      CMP (R2),(R2)
      CMP (R2),(R2)

      ;UPDATE TEST ID
      ;RELOCATE THIS TEST TO HIGH CACHE
      ;RESET LOOP COUNTER
      ;RESET CLOCK
      ;HIGH CACHE ADDRESS
      ;LOOP COUNTER
      ;RESET TICK
      ;ENABLE LOW CACHE
      ;LOOK FOR TICK
      ;NO TICK
      ;CLOCK TICK
      ;RESET TICK
      ;READ FROM CACHE

```

8285	017002	005201			INC R1	
8286	017004	001363			BNE 4\$	
8287	017006	012737	001015	177746	MOV #OFF,@#CCR	;DISABLE CACHE
8288	017014	013737	004702	004542	MOV @#TIME,@#ERROR	;SAVE TIME TO COMPLETE LOOP

8290
 8291
 8292
 8293 017022 032737 000001 000176
 8294 017030 001405
 8295 017032 023727 004542 000103
 8296
 8297 017040 103407
 8298 017042 000404
 8299 017044 023727 004542 000117

;REV B
 BIT #1,@#CONWRD
 BEQ 5\$
 CMP @#ERROR,#103
 BLO 2\$
 BR 6\$
 5\$: CMP @#ERROR,#117

;LINE 60HZ OR 50HZ
 ;BIT 0 CLEARED MEANS 60HZ
 ;IT IS 50 HZ
 ;DID LOOP COMPLETE IN TIME(50HZ)
 ;NO
 ;DID LOOP COMPLETE IN TIME(60HZ)

8301 017052 103402

BLO 2\$

;NO

8302

;REV B

8303

8304

8305 017054 005037 004542

6\$:

CLR @#ERROR

;RESET ERROR FLAG

8306 017060 004037 005352

2\$:

JSR R0,@#SETEN

;PRINT LIST OF SENTENCES

8307 017064 000004

.WORD ^D4

8308 017066 037714

.WORD SEN73

;FORCE MISS TESTS

8309 017070 037742

.WORD SEN75

;HIGH CACHE LOOKS ENABLED WHEN DISABLED

8310 017072 004010

JSR R0,(R0)

;TAKE SELECTED ACTION ON ERROR

```

8312 017074 046000          .WORD 3$-TST062-12+HIGHSP      ;LOOP ON ERROR
8313 017076 046000          .WORD 3$-TST062-12+HIGHSP      ;LOOP ON TEST
8314          :
8315          :
8316          :
8317          :
8318          :
8319          :
8320          :
8321          :
8322          :
8323          :
8324 017100 005267 161542    TST063: INC TID                ;UPDATE TEST ID
8325 017104 004467 167004    JSR R4,RELCTL              ;RELOCATE TEST TO LOW CACHE
8326 017110 017266          .WORD TST064
8327 017112 005037 004704    3$: CLR @#COUNT            ;RESET COUNTER
8328 017116 005037 004702    CLR @#TIME                 ;RESET CLOCK
8329 017122 012702 050000    MOV #LOW1,R2              ;LOW CACHE ADDRESS
8330 017126 005001          CLR R1
8331 017130 005037 177546    CLR @#KOOKOO              ;RESET TICK
8332 017134 042737 001010    BIC #BIT03+BIT09,@#CCR    ;ENABLE HIGH CACHE
8333 017142 105737 177546    4$: TSTB @#KOOKOO          ;LOOK FOR TICK
8334 017146 001404          BEQ 10$                    ;NO TICK
8335 017150 005237 004702    INC @#TIME                 ;CLOCK TICK
8336 017154 005037 177546    CLR @#KOOKOO              ;RESET TICK
8337 017160 021212          10$: CMP (R2),(R2)            ;READ FROM MEMORY
8338 017162 021212          CMP (R2),(R2)
8339 017164 021212          CMP (R2),(R2)
8340 017166 021212          CMP (R2),(R2)
8341 017170 005201          INC R1
8342 017172 001363          BNE 4$
8343 017174 012737 001015    MOV #OFF,@#CCR            ;DISABLE CACHE
8344 017202 013737 004702    MOV @#TIME,@#ERROR        ;SAVE TIME TO COMPLETE LOOP
8345          :
8346          :REV B
8347          :
8348 017210 032737 000001    BIT #1,@#CONWRD           ;LINE 60HZ OR 50HZ
8349 017216 001405          BEQ 5$                    ;BIT 0 CLEARED MEANS 60HZ
8350 017220 023727 004542    CMP @#ERROR,#103         ;IT IS 50 HZ
8351          :
8352 017226 103407          BLO 2$                    ;DID LOOP COMPLETE IN TIME(50HZ)
8353 017230 000404          BR 6$                     ;NO
8354 017232 023727 004542    5$: CMP @#ERROR,#117      ;DID LOOP COMPLETE IN TIME(60HZ)
8355 017240 103402          BLO 2$                    ;NO
8356          :
8357          :REV B
8358          :
8359 017242 005037 004542    6$: CLR @#ERROR            ;RESET ERROR
8360 017246 004037 005352    2$: JSR R0,@#SETEN          ;PRINT LIST OF SENTENCES
8361 017252 000004          .WORD ^D4
8362 017254 037714          .WORD SEN73               ;FORCE MISS TESTS
8363 017256 037760          .WORD SEN76               ;LOW CACHE LOOKS ENABLED WHEN DISABLED
8364 017260 004010          JSR R0,(R0)               ;TAKE SELECTED ACTION ON ERROR
8365 017262 044000          .WORD 3$-TST063-12+LOWSP  ;LOOP ON ERROR
8366 017264 044000          .WORD 3$-TST063-12+LOWSP  ;LOOP ON TEST
8367          :

```

8368
8369
8370
8371
8372
8373
8374
8375
8376
8377
8378 017266 005267 161354
8379 017272 004467 166616
8380 017276 017454
8381 017300 005037 004704
8382 017304 005037 004702
8383 017310 012702 046000
8384 017314 005001
8385 017316 005037 177546
8386 017322 042737 001010 177746
8387 017330 105737 177546
8388 017334 001404
8389 017336 005237 004702
8390 017342 005037 177546
8391 017346 021212
8392 017350 021212
8393 017352 021212
8394 017354 021212
8395 017356 005201
8396 017360 001363
8397 017362 012737 001015 177746
8398 017370 013737 004702 004542
8399
8400
8401
8402 017376 032737 000001 000176
8403 017404 001405
8404 017406 023727 004542 000103
8405
8406 017414 103007
8407 017416 000404
8408 017420 023727 004542 000117
8409 017426 103002
8410
8411
8412
8413 017430 005037 004542
8414 017434 004037 005352
8415 017440 000004
8416 017442 037714
8417 017444 037776
8418 017446 004010
8419 017450 044000
8420 017452 044000
8421
8422
8423

```
TEST ENABLES HIGH CACHE AND CAUSES MULTIPLE READS TO  
HIGH CACHE.  
READ LOOP IS TIMED USING SYSTEM CLOCK  
IF READ LOOP COMPLETES TO SLOW THEN  
IT IS ASSUMED THAT HIGH CACHE DID NOT ENABLE  
TST064: INC TID ;UPDATE TEST ID  
JSR R4,RELCTL ;RELOCATE TEST TO LOW CACHE  
.WORD TST065  
3$: CLR @#COUNT ;RESET LOOP COUNTER  
CLR @#TIME ;RESET CLOCK  
MOV #HIGHSP,R2 ;HIGH CACHE ADDRESS  
CLR R1 ;LOOP COUNTER  
CLR @#KOOKOO ;RESET KL11W  
BIC #BIT03+BIT09,@#CCR ;ENABLE LOW CACHE  
4$: TSTB @#KOOKOO ;LOOK FOR TICK  
BEQ 10$ ;NO TICK  
INC @#TIME ;CLOCK TICK  
CLR @#KOOKOO ;LOOK FOR TICK  
10$: CMP (R2),(R2) ;READ FROM CACHE  
CMP (R2),(R2)  
CMP (R2),(R2)  
CMP (R2),(R2)  
INC R1 ;CHECK PASS COUNTER  
BNE 4$ ;LOOP NOT COMPLETE  
1$: MOV #OFF,@#CCR ;DISABLE CACHE  
MOV @#TIME,@#ERROR ;SAVE LOOP TIME  
;REV B  
BIT #1,@#CONWRD ;LINE 60HZ OR 50HZ  
BEQ 5$ ;BIT 0 CLEARED MEANS 60HZ  
CMP @#ERROR,#103 ;IT IS 50 HZ  
;DID LOOP COMPLETE IN TIME(50HZ)  
;NO  
BHIS 2$  
BR 6$  
5$: CMP @#ERROR,#117 ;DID LOOP COMPLETE IN TIME(60HZ)  
BHIS 2$ ;NO  
;REV B  
6$: CLR @#ERROR ;RESET ERROR FLAG  
2$: JSR R0,@#SETEN ;PRINT LIST OF SENTENCES  
.WORD ^D4  
.WORD SEN73 ;FORCE MISS TESTS  
.WORD SEN77 ;HIGH CACHE LOOKS DISABLED WHEN ENABLED  
JSR R0,(R0) ;TAKE SELECTED ACTION ON ERROR  
.WORD 3$-TST064-12+LOWSP ;LOOP ON ERROR  
.WORD 3$-TST064-12+LOWSP ;LOOP ON TEST
```

.SBTTL WRITE WRONG PARITY TESTS

8425
8426
8427
8428
8429
8430
8431
8432
8433
8434 017454 005267 161166
8435 017460 004467 166430
8436 017464 017750
8437 017466 005001
8438 017470 012737 001415 177746
8439 017476 005301
8440 017500 001404
8441 017502 032737 010000 177746
8442 017510 001372
8443 017512 012701 046000
8444 017516 012737 002105 177746
8445 017524 012102
8446 017526 020127 046014
8447 017532 001371
8448 017534 012737 000005 177746
8449 017542 005037 177744
8450 017546 005037 004542
8451 017552 014105
8452 017554 052737 000215 177746
8453 017562 053737 177744 004542
8454 017570 005037 177744
8455 017574 012737 000005 177746
8456 017602 020127 046000
8457 017606 001361
8458 017610 010702
8459 017612 014201
8460 017614 020227 044000
8461 017620 001374
8462 017622 005137 004542
8463 017626 042737 177437 004542
8464 017634 005037 177744
8465 017640 012737 001015 177746
8466 017646 032737 000100 004542
8467 017654 001406
8468 017656 004037 005352
8469 017662 000006
8470 017664 040014
8471 017666 040026
8472 017670 040054
8473 017672 032737 000200 004542
8474 017700 001406
8475 017702 004037 005352
8476 017706 000006
8477 017710 040014
8478 017712 040026
8479 017714 040100
8480 017716 032737 000040 004542

.....
:TST065:

```

ENABLE WRITE WRONG PARITY TO TAG AND DATA.          ENABLE HIGH
CACHE.          VERIFY PARITY ERROR TO TAG AND DATA.
:UPDATE TEST ID
:RELOCATE TEST TO LOW CACHE
INC TID
JSR R4,RELCTL
.WORD TST066
CLR R1          ;TIME OUT COUNTER
MOV #OFF+BIT08,@#CCR          ;FLUSH CACHE
10$: DEC R1          ;CHECK FOR TIME OUT
      BEQ 1$
      BIT #BIT12,@#CCR          ;WAIT TILL COMPLETE
      BNE 10$
1$: MOV #HIGHSP,R1          ;WRITE TO HIGH CACHE LOCATIONS
2$: MOV #2105,@#CCR          ;WRITE WRONG ALL TO HIGH CACHE
      MOV (R1)+,R2          ;CAUSE WRITES TO 6 HIGH CACHE LOCATIONS
      CMP R1,#HIGHSP+14          ;LOOP
      BNE 2$          ;NOT COMPLETE
      MOV #BIT02+BIT00,@#CCR          ;ENABLE HIGH CACHE ONLY
      CLR @#CMPE          ;RESET PARITY ERRORS
      CLR @#ERROR          ;RESET ERROR FLAG
3$: MOV -(R1),R5          ;READ TO CAUSE ERROR
      BIS #215,@#CCR          ;SET ABORT FOR ERROR READ
      BIS @#CMPE,@#ERROR          ;SAVE ERROR INFO
      CLR @#CMPE          ;RESET ERROR REGISTER
      MOV #5,@#CCR          ;DISABLE ABORT
CMP R1,#HIGHSP          ;LOOP SIX TIMES
      BNE 3$
      MOV PC,R2          ;CORRECT ANY WRONG PARITY TO LOW CACHE
11$: MOV -(R2),R1          ;READ LOW CACHE
      CMP R2,#LOWSP          ;CONTINUE TO START OF BLOCK
      BNE 11$
      COM @#ERROR          ;PARITY ERRORS HIGH LOW AND TAG
      BIC #177437,@#ERROR          ;RESET ERROR REGISTER
      CLR @#CMPE          ;DISABLE CACHE
      MOV #OFF,@#CCR          ;IF NOT SET THEN ERROR
      BIT #BIT06,@#ERROR
      BEQ 4$
      JSR R0,@#SETEN          ;PRINT LIST OF SENTENCES
      .WORD *D6
      .WORD SEN78          ;WRITE WRONG PARITY TESTS
      .WORD SEN79          ;WROTE WRONG PARITY TO LOW BYTE HIGH BYTE AND TA
      .WORD SEN80          ;LOW BYTE PARITY ERROR BIT06 NOT SET IN CMPE
4$: BIT #BIT07,@#ERROR          ;IF NOT SET THEN ERROR
      BEQ 5$
      JSR R0,@#SETEN          ;PRINT LIST OF SENTENCES
      .WORD *D6
      .WORD SEN78          ;WRITE WRONG PARITY TESTS
      .WORD SEN79          ;WROTE WRONG PARITY TO LOW BYTE HIGH BYTE AND TA
      .WORD SEN81          ;HIGH BYTE PARITY ERROR BIT07 NOT SET IN CMPE
5$: BIT #BIT05,@#ERROR          ;IF NOT SET THEN ERROR

```

CACHE DIAG.
CFKKAB.P11

MACY11 30A(1052)
25-JUN-79 13:31

31-OCT-79 15:29 PAGE 70-3
WRITE WRONG PARITY TESTS

SEQ 0073

8481	017724	001406	
8482	017726	004037	005352
8483	017732	000006	
8484	017734	040014	
8485	017736	040026	
8486	017740	040124	
8487	017742	004010	
8488	017744	044024	
8489	017746	044024	

6\$:

```

REQ 6$
JSR R0,@#SETEN          ;PRINT LIST OF SENTENCES
.WORD ^D6
.WORD SEN78             ;WRITE WRONG PARITY TESTS
.WORD SEN79             ;WROTE WRONG PARITY TO LOW BYTE HIGH BYTE AND TA
.WORD SEN82             ;TAG PARITY ERROR BIT05 NOT SET IN CMPE
JSR R0,(R0)             ;TAKE SELECTED ACTION ON ERROR
.WORD 1$-TST065-12+LOWSP ;LOOP ON ERROR
.WORD 1$-TST065-12+LOWSP ;LOOP ON TEST

```

```

8491          .SBTTL HIT REGISTER TESTS
8492          :          CHECK THAT ALL SIX HIT BITS CAN CONTAIN ZEROS
8493
8494
8495 017750 005267 160672          TST066:          INC TID          ;UPDATE TEST ID
8496 017754 004467 166134          JSR R4,RELCTL          ;RELOCATE TEST TO LOW CACHE
8497 017760 020046          .WORD TST067
8498 017762 010701          MOV PC,R1          ; LOW TEST AREA
8499 017764 011121          2$:          MOV (R1),(R1)+
8500 017766 020127 044032          CMP R1,#1$-TST066-12+LOWSP
8501 017772 001374          BNE 2$
8502 017774 000240          NOP
8503 017776 000240          NOP
8504 020000 000240          NOP
8505 020002 000240          NOP
8506 020004 000240          NOP
8507 020006 013737 177752 004542          1$:          MOV @#CHR,@#ERROR          ;READ AND SAVE HIT REGISTER CONTENTS
8508 020014 042737 177700 004542          BIC #177700,@#ERROR          ;MASK FOR HIT BITS
8509 020022 005237 004560          INC @#BITFLG          ;BIT PRINT MODE
8510 020026 004037 005352          JSR R0,@#SETEN          ;REPORT ERROR IF ANY
8511 020032 000004          .WORD ^D4
8512 020034 040146          .WORD SEN83          ;HIT REGISTER TESTS
8513 020036 040156          .WORD SEN84          ;CACHE HIT REGISTER BIT(S) STUCK HIGH
8514 020040 004010          JSR R0,(R0)          ;TAKE SELECTED ACTION ON ERROR
8515 020042 044000          .WORD LOWSP
8516 020044 044000          .WORD LOWSP

```

```

8517
8518
8519
8520
8521          :          CHECK THAT HIT BIT05 CAN CONTAIN A ONE
8522
8523
8524
8525 020046 005267 160574          TST067:          INC TID          ;UPDATE TEST ID
8526 020052 004467 166036          JSR R4,RELCTL          ;RELOCATE TEST TO LOW CACHE
8527 020056 020160          .WORD TST070
8528 020060 012737 000005 177746          MOV #5,@#CCR          ;ENABLE HIGH CACHE
8529 020066 013702 046000          MOV @#HIGHSP,R2          ;CAUSE READ MISS
8530 020072 013702 046000          MOV @#HIGHSP,R2          ;CAUSE READ HIT
8531 020076 000240          NOP
8532 020100 000240          NOP
8533 020102 000240          NOP
8534 020104 013737 177752 004542          MOV @#CHR,@#ERROR          ;READ AND SAVE HIT REGISTER
8535 020112 042737 177737 004542          BIC #177737,@#ERROR          ;MASK FOR BIT 5
8536 020120 162737 000040 004542          SUB #40,@#ERROR          ;BIT 5 SHOULD BE SET
8537 020126 012737 001015 177746          MOV #OFF,@#CCR          ;DISABLE CACHE
8538 020134 004037 005352          JSR R0,@#SETEN          ;REPORT ERROR IF ANY
8539 020140 000010          .WORD ^D8
8540 020142 040146          .WORD SEN83          ;HIT REGISTER TESTS
8541 020144 040174          .WORD SEN85          ;HIT BIT ERROR
8542 020146 040204          .WORD SEN86          ;ATTEMPT TO WRITE HIT REGISTER BIT05
8543 020150 040222          .WORD SEN87          ;TO A ONE , VAI READ HIT , FAILED
8544 020152 004010          JSR R0,(R0)          ;TAKE SELECTED ACTION ON ERROR
8545 020154 044000          .WORD LOWSP          ;LOOP ON ERROR
8546 020156 044000          .WORD LOWSP          ;LOOP ON TEST

```

```

8548
8549 020160 005267 160462
8550 020164 004467 165724
8551 020170 020374
8552 020172 012737 000005 177746
8553 020200 013701 046000
8554 020204 012737 000000 046000
8555 020212 112737 000023 046000
8556 020220 112737 000023 046001
8557 020226 005037 177744
8558 020232 013703 046000
8559 020236 012737 000215 177746
8560 020244 013737 177744 004542
8561 020252 012737 001015 177746
8562 020260 032737 000100 004542
8563 020266 001407
8564 020270 004037 005352
8565 020274 000006
8566 020276 040246
8567 020300 040262
8568 020302 040276
8569 020304 000430
8570 020306 032737 000200 004542
8571 020314 001407
8572 020316 004037 005352
8573 020322 000006
8574 020324 040246
8575 020326 040262
8576 020330 040310
8577 020332 000415
8578 020334 010337 004542
8579 020340 012703 011423
8580 020344 162737 011423 004542
8581 020352 105737 004542
8582 020356 001344
8583 020360 105737 004543
8584 020364 001350
8585 020366 004010
8586 020370 044000
8587 020372 044000

```

TST070:

```

.SBTTL CACHE DATA WRITE BYTE TEST
INC TID ;UPDATE TEST ID
JSR R4,RELCTL ;RELOCATE TEST TO LOW CACHE
.WORD TST071
MOV #5,@#CCR ;ENABLE HIGH CACHE
MOV @#HIGHSP,R1 ;READ MISS
MOV #0,@#HIGHSP ;WRITE WORD HIT
MOV# #23,@#HIGHSP ;WRITE BYTE HIT
MOV# #23,@#HIGHSP+1 ;WRITE BYTE HIT
CLR @#CMPE ;RESET ERROR REGISTER
MOV @#HIGHSP,R3 ;READ AND LOOK FOR ERROR
MOV #215,@#CCR ;SET ABORT FOR ERROR READ
MOV @#CMPE,@#ERROR ;SAVE ERROR
MOV #OFF,@#CCR ;DISABLE CACHE
BIT #BIT06,@#ERROR ;ANY LOW BYTE ERROR
BEQ 1$
3$: JSR R0,@#SETEN ;REPORT ERROR
.WORD ^D6
.WORD SEN88 ;CACHE DATA WRITE BYTE TEST
.WORD SEN89 ;CACHE DATA WRITE BYTE ERROR
.WORD SEN90 ;CANT WRITE LOW BYTE
BR 2$
1$: BIT #BIT07,@#ERROR ;ANY HIGH BYTE ERROR
BEQ 5$
JSR R0,@#SETEN ;REPORT ERROR
.WORD ^D6
.WORD SEN88 ;CACHE DATA WRITE BYTE TEST
.WORD SEN89 ;CACHE DATA WRITE BYTE ERROR
.WORD SEN91 ;CANT WRITE HIGH BYTE
BR 2$
5$: MOV R3,@#ERROR ;WAS DATA READ BACK OK
MOV #11423,R3 ;PREVENT LOOP
SUB #11423,@#ERROR ;DATA THAT SHOULD BE
TSTB @#ERROR ;ANY LOW BYTE ERROR
BNE 3$ ;YES
TSTB @#ERROR+1 ;ANY HIGH BYTE ERROR
BNE 1$ ;YES
2$: JSR R0,(R0) ;TAKE SELECTED ACTION ON ERROR
.WORD LOWSP
.WORD LOWSP

```

.SBTTL PARITY ERROR REGISTER TESTS

```

8589
8590
8591
8592
8593
8594
8595
8596 020374 005267 160246
8597 020400 004467 165510
8598 020404 020560
8599 020406 005037 177744
8600 020412 012737 000005 177746
8601 020420 013702 052000
8602 020424 013702 046000
8603 020430 052737 002000 177746
8604 020436 013702 052000
8605 020442 042737 002000 177746
8606 020450 013702 052000
8607 020454 052737 000215 177746
8608 020462 013702 177744
8609 020466 012737 001015 177746
8610 020474 005037 004542
8611 020500 010701
8612 020502 014111
8613 020504 020127 044000
8614 020510 001374
8615 020512 032702 000040
8616 020516 001415
8617 020520 010237 004542
8618 020524 100412
8619 020526 004037 005352
8620 020532 000006
8621 020534 040322
8622 020536 040334
8623 020540 040356
8624 020542 004010
8625 020544 044000
8626 020546 044000
8627 020550 000403
8628 020552 005037 004542
8629 020556 000771

```

: VERIFY PARITY ERROR REGISTER BIT15 SETS AS RESULT
: OF TAG ERROR

```

TST071:      INC TID      ;UPDATE TEST ID
              JSR R4,RELCTL ;RELOCATE TEST TO LOW CACHE
              .WORD TST072
              CLR @#CMPE    ;RESET CACHE ERROR REGISTER
              MOV #5,@#CCR  ;ENABLE HIGH CACHE
              MOV @#HIGH1,R2 ;TAG HIGH LOCATION BLOCK #2
              MOV @#HIGHSP,R2 ;TAG HIGH LOCATION BLOCK #1
              BIS #BIT10,@#CCR ;WRITE WRONG PARITY TO TAG
              MOV @#HIGH1,R2 ;TAG LOCATION
              BIC #BIT10,@#CCR ;DISABLE WRITE WRONG
              MOV @#HIGH1,R2 ;CAUSE PARITY ERROR TAG
              BIS #215,@#CCR ;ENABLE ABORT FOR ERROR READ
              MOV @#CMPE,R2 ;READ AND SAVE ERROR REGISTER
              MOV #OFF,@#CCR ;DISABLE CACHE
              CLR @#ERROR    ;RESET ERROR FLAG
              MOV PC,R1      ;UNTAG LOW CACHE
              3$: MOV -(R1),(R1) ;CAUSE CACHE WRITE IN BYPASS MODE
              CMP R1,#LOWSP
              BNE 3$
              BIT #BIT05,R2 ;ANY TAG ERROR
              BEQ 1$        ;NO ERROR SO ABORT TEST
              MOV R2,@#ERROR ;SET ERROR FLAG
              BMI 1$        ;BIT 15 SET SO NO ERROR
              JSR R0,@#SETEN ;REPORT ERROR
              .WORD *D6
              .WORD SEN92   ;PARITY ERROR REGISTER TESTS
              .WORD SEN93   ;BIT15 OF CMPE WAS NOT SET AS RESULT
              .WORD SEN94   ;OF TAG PARITY ERROR
              2$: JSR R0,(R0) ;TAKE SELECTED ACTION ON ERROR
              .WORD LOWSP   ;LOOP ON ERROR
              .WORD LOWSP   ;LOOP ON TEST
              BR TST072
              1$: CLR @#ERROR ;RESET ERROR FLAG
              BR 2$

```

: VERIFY PARITY ERROR REGISTER BIT15 SETS
: AS RESULT OF LOW BYTE PARITY ERROR

```

8630
8631
8632
8633
8634
8635
8636
8637
8638 020560 005267 160062
8639 020564 004467 165324
8640 020570 020746
8641 020572 005037 177744
8642 020576 012737 000005 177746
8643 020604 013702 052000
8644 020610 013702 046000

```

```

TST072:      INC TID      ;UPDATE TEST ID
              JSR R4,RELCTL ;RELOCATE TEST TO LOW CACHE
              .WORD TST073
              CLR @#CMPE    ;RESET CACHE PARITY ERROR REGISTER
              MOV #5,@#CCR  ;ENABLE HIGH CACHE
              MOV @#HIGH1,R2 ;TAG HIGH LOCATION BLOCK #2
              MOV @#HIGHSP,R2 ;TAG HIGH LOCATION BLOCK #1

```

8645	020614	052737	000100	177746		BIS #BIT06,@#CCR	:ENABLE WRITE WRONG DATA
8646	020622	112737	000005	046000		MOVB #5,@#HIGHSP	:WRITE WRONG PARITY LOW BYTE
8647	020630	042737	000100	177746		BIC #BIT06,@#CCR	:DISABLE WRITE WRONG
8648	020636	013702	046000			MOV @#HIGHSP,R2	:READ AND CAUSE LOW BYTE PARITY ERROR
8649	020642	052737	000215	177746		BIS #215,@#CCR	:ENABLE ABORT FOR ERROR READ
8650	020650	013702	177744			MOV @#CMPE,R2	:READ AND SAVE ERROR REGISTER
8651	020654	012737	001015	177746		MOV #OFF,@#CCR	:DISABLE CACHE
8652	020662	005037	004542			CLR @#ERROR	:RESET ERROR REGISTER
8653	020666	010703				MOV PC,R3	:UNTAG LOW CACHE PARITY
8654	020670	014313			5\$:	MOV -(R3),(R3)	:CAUSE WRITE IN BYPASS MODE
8655	020672	020327	044000			CMP R3,#LOWSP	
8656	020676	001374				BNE 5\$	
8657	020700	032702	000100			BIT #BIT06,R2	:ANY LOW BYTE PARITY ERROR
8658	020704	001415				BEQ 1\$:NO ERROR ABORT TEST
8659	020706	010237	004542			MOV R2,@#ERROR	:WAS BIT15 SET AS RESULT OF ERROR
8660	020712	100412				BMI 1\$:YES
8661	020714	004037	005352			JSR R0,@#SETEN	:REPORT ERROR
8662	020720	000006				.WORD *D6	
8663	020722	040322				.WORD SEN92	:PARITY ERROR REGISTER TESTS
8664	020724	040370				.WORD SEN95	:BIT15 OF CMPE DID NOT SET
8665	020726	040406				.WORD SEN96	:AS RESULT OF LOW BYTE PARITY ERROR
8666	020730	004010			2\$:	JSR R0,(R0)	:TAKE SELECTED ACTION ON ERROR
8667	020732	044000				.WORD LOWSP	:LOOP ON ERROR
8668	020734	044000				.WORD LOWSP	:LOOP ON TEST
8669	020736	000403				BR TST073	
8670	020740	005037	004542		1\$:	CLR @#ERROR	:RESET ERROR FLAG
8671	020744	000771				BR 2\$	
8672							
8673							
8674							
8675							
8676					:	VERIFY PARITY ERROR REGISTER BIT15 SETS AS RESULT	
8677					:	OF HIGH BYTE PARITY ERROR	
8678							
8679							
8680	020746	005267	157674		TST073:	INC TID	:UPDATE TEST ID
8681	020752	004467	165136			JSR R4,RELCTL	:RELOCATE TEST TO LOW CACHE
8682	020756	021134				.WORD TST074	
8683	020760	005037	177744			CLR @#CMPE	:RESET PARITY ERROR REGISTER
8684	020764	012737	000005	177746		MOV #5,@#CCR	:ENABLE HIGH CACHE
8685	020772	013702	052000			MOV @#HIGH1,R2	:TAG HIGH BLOCK #2
8686	020776	013702	046000			MOV @#HIGHSP,R2	:TAG HIGH BLOCK #1
8687	021002	052737	000100	177746		BIS #BIT06,@#CCR	:ENABLE WRITE WRONG PARITY
8688	021010	112737	000005	046001		MOVB #5,@#HIGHSP+1	:WRITE WRONG PARITY HIGH BYTE
8689	021016	042737	000100	177746		BIC #BIT06,@#CCR	:DISABLE WRITE WRONG
8690	021024	013702	046000			MOV @#HIGHSP,R2	:READ TO CAUSE PARITY ERROR
8691	021030	052737	000215	177746		BIS #215,@#CCR	:ENABLE ABORT FOR ERROR READ
8692	021036	013702	177744			MOV @#CMPE,R2	:READ ERROR
8693	021042	012737	001015	177746		MOV #OFF,@#CCR	:DISABLE CACHE
8694	021050	005037	004542			CLR @#ERROR	:RESET ERROR
8695	021054	010703				MOV PC,R3	:UNTAG LOW CACHE
8696	021056	014313			5\$:	MOV -(R3),(R3)	:CAUSE WRITE TO CACHE IN BYPASS MODE
8697	021060	020327	044000			CMP R3,#LOWSP	
8698	021064	001374				BNE 5\$	
8699	021066	032702	000200			BIT #BIT07,R2	:ANY HIGH BYTE PARITY ERROR
8700	021072	001415				BEQ 1\$:NO ERROR ABORT TEST

8701 021074 010237 004542
 8702 021100 100412
 8703 021102 004037 005352
 8704 021106 000006
 8705 021110 040322
 8706 021112 040426
 8707 021114 040444
 8708 021116 004010
 8709 021120 044000
 8710 021122 044000
 8711 021124 000403
 8712 021126 005037 004542
 8713 021132 000771

```

MOV R2,@#ERROR ;WAS BIT15 SET AS RESULT OF ERROR
BMI 1$ ;YES
JSR R0,@#SETEN ;REPORT ERROR
.WORD ^D6
.WORD SEN92 ;PARITY ERROR REGISTER TESTS
.WORD SEN97 ;BIT15 OF CMPE WAS NOT SET
.WORD SEN98 ;AS RESULT OF HIGH BYTE PARITY ERROR
2$: JSR R0,(R0) ;TAKE SELECTED ACTION ON ERROR
.WORD LOWSP ;LOOP ON ERROR
.WORD LOWSP ;LOOP ON TEST
1$: BR TST074 ;RESET ERROR FLAG
BR 2$

```

8714
8715
8716
8717
8718
8719
8720 021134 005267 157506
8721 021140 004467 164750
8722 021144 021276
8723 021146 005037 177744
8724 021152 012737 000005 177746
8725 021160 013702 052000
8726 021164 012737 002105 177746
8727 021172 013702 046000
8728 021176 012737 000005 177746
8729 021204 013702 046000
8730 021210 005037 177744
8731 021214 012737 001215 177746
8732 021222 013737 177744 004542
8733 021230 012737 000015 177746
8734 021236 010702
8735 021240 014212
8736 021242 020227 044000
8737 021244 001174
8738 021250 012737 001015 177746
8739 021256 004037 005352
8740 021262 000004
8741 021264 040322
8742 021266 040464
8743 021270 004010
8744 021272 044000
8745 021274 044000
8746
8747
8748
8749

```

:
:
: TST074: VERIFY THAT ERROR STATUS LATCHED IN ERROR REGISTER
: CAN BE CLEARED BY WRITE TO THAT REGISTER
INC TID ;UPDATE TEST ID
JSR R4,RELCTL ;RELOCATE TEST TO LOW CACHE
.WORD TST075
1$: CLR @#CMPE ;RESET ERROR REGISTER
MOV #5,@#CCR ;ENABLE HIGH CACHE
MOV @#HIGH1,R2 ;POINT TAG TO HIGH BLOCK #2
MOV #2105,@#CCR ;WRITE WRONG TO HIGH CACHE
MOV @#HIGHSP,R2 ;CAUSE WRONG WRITE TO HIGH CACHE
MOV #5,@#CCR ;DISABLE WRITE WRONG
MOV @#HIGHSP,R2 ;CAUSE WRONG READ FROM CACHE
CLR @#CMPE ;RESET ERROR REGISTER
MOV #OFF+BIT07,@#CCR ;ENABLE ABORT FOR ERROR READ
MOV @#CMPE,@#ERROR ;SAVE ERROR INFO.
MOV #OFF-BIT09,@#CCR ;DISABLE CACHE
MOV PC,R2 ;CORRECT ANY WRONG PARITY IN LOW CACHE
2$: MOV -(R2),(R2) ;CAUSE READ TO LOW CACHE
CMP R2,#LOWSP ;FOR ALL LOW CACHE USED
BNE 2$
MOV #OFF,@#CCR ;DISABLE CACHE
JSR R0,@#SETEN ;PRINT LIST OF SENTENCES
.WORD ^D4
.WORD SEN92 ;PARITY ERROR REGISTER TESTS
.WORD SEN99 ;WRITE TO CMPE FAILED TO CLEAR REGISTER
JSR R0,(R0) ;TAKE SELECTED ACTION ON ERROR
.WORD LOWSP ;LOOP ON ERROR
.WORD LOWSP ;LOOP ON TEST

```

.SBTTL PARITY ERROR LOGIC TESTS

```

8751
8752
8753
8754
8755
8756
8757
8758 021276 005267 157344
8759 021302 004467 164606
8760 021306 021546
8761 021310 012737 000005 177746
8762 021316 005037 177744
8763 021322 013702 052000
8764 021326 013702 046000
8765 021332 012737 000006 046000
8766 021340 013702 046000
8767 021344 052737 000215 177746
8768 021352 013737 177744 004542
8769 021360 012737 000005 177746
8770 021366 013702 052000
8771 021372 012737 000004 052000
8772 021400 005037 177744
8773 021404 013702 052000
8774 021410 052737 000215 177746
8775 021416 013702 177744
8776 021422 042737 000200 177746
8777 021430 012737 001015 177746
8778 021436 005102
8779 021440 040237 004542
8780 021444 032737 000100 004542
8781 021452 001406
8782 021454 004037 005352
8783 021460 000006
8784 021462 040504
8785 021464 040516
8786 021466 040530
8787 021470 032737 000200 004542
8788 021476 001406
8789 021500 004037 005352
8790 021504 000006
8791 021506 040504
8792 021510 040546
8793 021512 040530
8794 021514 032737 000040 004542
8795 021522 001406
8796 021524 004037 005352
8797 021530 000006
8798 021532 040504
8799 021534 040560
8800 021536 040530
8801 021540 004010
8802 021542 044000
8803 021544 044000
8804
8805
8806

```

...
TST075:

```

VERIFY PARITY ERROR LOGIC BY EVEN AND ODD DATA INTO CACHE
TAG AND DATA. IF A PARITY ERROR OCCURES ON EVEN DATA
BUT NOT ON ODD THEN LOGIC IS ASSUMED GOOD.
ONLY IF BOTH ODD AND EVEN FAIL IS THE LOGIC ASSUMED BAD.
INC TID ;UPDATE TEST ID
JSR R4,RELCTL ;RELOCATE TEST TO LOW CACHE
.WORD TST076
1$: MOV #5,@#CCR ;ENABLE HIGH CACHE
CLR @#CMPE ;RESET ERROR REGISTER
MOV @#HIGH1,R2 ;POINT TAG TO HIGH BLOCK #2
MOV @#HIGHSP,R2 ;POINT TAG TO HIGH BLOCK #1
MOV #6,@#HIGHSP ;WRITE DATA INTO IT
MOV @#HIGHSP,R2 ;READ EVEN DATA WITH EVEN TAG
BIS #215,@#CCR ;SET ABORT TO READ ERROR
MOV @#CMPE,@#ERROR ;SAVE ERROR INFO
MOV #5,@#CCR ;DISABLE ABORT
MOV @#HIGH1,R2 ;POINT TAG TO HIGH BLOCK #2
MOV #4,@#HIGH1 ;WRITE ODD DATA INTO IT
CLR @#CMPE ;RESET ERROR REGISTER
MOV @#HIGH1,R2 ;READ ODD DATA AND TAG
BIS #215,@#CCR ;ENABLE ABORT FOR ERROR READ
MOV @#CMPE,R2 ;READ ERROR REGISTER
BIC #BIT07,@#CCR ;DISABLE ABORT
MOV #OFF,@#CCR ;DISABLE CACHE
COM R2 ;AND RESULTS OF ODD AND EVEN
BIC R2,@#ERROR ;ANY BITS SET = ERROR
BIT #BIT06,@#ERROR ;WAS LOW PARITY SET
BEQ 2$
JSR R0,@#SETEN ;PRINT LIST OF SENTENCES
.WORD ^D6
.WORD SEN100 ;PARITY ERROR LOGIC TESTS
.WORD SEN101 ;LOW BYTE PARITY FAILURE
.WORD SEN102 ;FAILED ON ODD AND EVEN READ
2$: BIT #BIT07,@#ERROR ;WAS IT HIGH BYTE ERROR
BEQ 3$
JSR R0,@#SETEN ;PRINT LIST OF SENTENCES
.WORD ^D6
.WORD SEN100 ;PARITY ERROR LOGIC TESTS
.WORD SEN103 ;HIGH BYTE PARITY FAILURE
.WORD SEN102 ;FAILED ON ODD AND EVEN READ
3$: BIT #BIT05,@#ERROR ;WAS IT TAG PARITY ERROR
BEQ 4$
JSR R0,@#SETEN ;PRINT LIST OF SENTENCES
.WORD ^D6
.WORD SEN100 ;PARITY ERROR LOGIC TESTS
.WORD SEN104 ;TAG PARITY FAILURE
.WORD SEN102 ;FAILED ON ODD AND EVEN READ
4$: JSR R0,(R0) ;TAKE SELECTED ACTION ON ERROR
.WORD LOWSP ;LOOP ON ERROR
.WORD LOWSP ;LOOP ON TEST

```

...


```
8863 022046 040570          .WORD SEN105          ;PARITY ERROR BIT FAILED TO SET ON WRONG
8864 022050 040612          .WORD SEN106          ;WRITE ODD AND EVEN
8865 022052 032737 000040 004542 3$: BIT #BIT05,@#ERROR ;ANY TAG ERRORS
8866 022060 001407          BEQ 4$
8867 022062 004037 005352   JSR R0,@#SETEN        ;PRINT LIST OF SENTENCES
8868 022066 000010          .WORD ^D8
8869 022070 040504          .WORD SEN100          ;PARITY ERROR LOGIC TESTS
8870 022072 040560          .WORD SEN104          ;TAG PARITY FAILURE
8871 022074 040570          .WORD SEN105          ;PARITY ERROR BIT FAILED TO SET ON WRONG
8872 022076 040612          .WORD SEN106          ;WRITE ODD AND EVEN
8873 022100 042737 000140 177746 4$: BIC #BIT06+BIT05,@#CCR ;DISABLE WRITE WRONG PARITY
8874 022106 052737 001010 177746   BIS #BIT03+BIT09,@#CCR ;DISABLE CACHE
8875 022114 004010          JSR R0,(R0)           ;TAKE SELECTED ACTION ON ERROR
8876 022116 044000          .WORD LOWSP          ;LOOP ON ERROR
8877 022120 044000          .WORD LOWSP          ;LOOP ON TEST
8878
8879
8880
8881
```

.SBTTL READ WRITE HIT TESTS

```

8883
8884
8885
8886
8887
8888 022122 005267 156520
8889 022126 004467 164010
8890 022132 022262
8891 022134 005037 004542
8892 022140 012701 050000
8893 022144 012737 000011 177746
8894 022152 012102
8895 022154 020127 050014
8896 022160 001371
8897 022162 012701 050000
8898 022166 012721 000000
8899 022172 020127 050014
8900 022176 001373
8901 022200 014102
8902 022202 013703 177752
8903 022206 032703 000004
8904 022212 001402
8905 022214 005237 004542
8906 022220 022701 050000
8907 022224 001365
8908 022226 012737 001015 177746
8909 022234 162737 000006 004542
8910 022242 004037 005352
8911 022246 000004
8912 022250 040624
8913 022252 040636
8914 022254 004010
8915 022256 046000
8916 022260 046000
8917
8918

```

: TEST VERIFYS THAT SIX READS INTO TAGGED CACHE LOCATIONS
: WILL YEALD SIX HITS.

```

TST077: INC TID ;UPDATE TEST ID
        JSR R4,RELCTH ;RELOCATE THIS TEST TO HIGH CACHE
        .WORD TST100
        CLR @WERROR ;RESET ERROR REGISTER
        MOV #LOW1,R1 ;POINTER TO LOW BLOCK #2
1$: MOV #11,@WCCR ;ENABLE LOW CACHE
    MOV (R1)+,R2 ;TAG SIX LOW CACHE LOCATIONS
    CMP R1,#LOW1+14
    BNE 1$
    MOV #LOW1,R1 ;RESET LOW CACHE POINTER
2$: MOV #0,(R1)+ ;CAUSE WRITES TO 6 LOW CACHE LOCATIONS
    CMP R1,#LOW1+14 ;COMPLETE YET
    BNE 2$ ;NO
3$: MOV -(R1),R2 ;READ PREV. WRITTEN LOCATIONS
    MOV @WCHR,R3 ;SAVE HIT REGISTER
    BIT #BIT02,R3 ;WAS LAST ACCESS A HIT
    BEQ 30$ ;NO
    INC @WERROR ;YES
30$: CMP #LOW1,R1 ;LOOP TILL COMPLETE
    BNE 3$
    MOV #OFF,@WCCR ;DISABLE CACHE
    SUB #6,@WERROR ;LESS THEN 6 HITS = ERROR
    JSR R0,@WSETEN ;PRINT LIST OF SENTENCES
        .WORD ^D4 ;READ WRITE HIT TESTS
        .WORD SEN107 ;NO HITS AS RESULT OF READS TO SIX TAGGED LOCATI
        .WORD SEN108 ;TAKE SELECTED ACTION ON ERROR
        JSR R0,(R0) ;LOOP ON ERROR
        .WORD HIGHSP ;LOOP ON TEST
        .WORD HIGHSP

```

.SBTTL I/O PAGE TESTS

```

8920
8921
8922
8923
8924 022262 005267 156360
8925 022266 004467 163622
8926 022272 022452
8927 022274 005037 004542
8928 022300 012702 000006
8929 022304 012737 000005 177746
8930 022312 013703 047776
8931 022316 013703 177776
8932 022322 013703 177776
8933 022326 032737 000010 177752
8934 022334 001414
8935 022336 012737 001015 177746
8936 022344 012737 000001 004542
8937 022352 004037 005352
8938 022356 000004
8939 022360 040664
8940 022362 040674
8941 022364 000421
8942 022366 013703 047776
8943 022372 032737 000010 177752
8944 022400 001013
8945 022402 012737 001015 177746
8946 022410 012737 000001 004542
8947 022416 004037 005352
8948 022422 000004
8949 022424 040664
8950 022426 040712
8951 022430 012737 001015 177746
8952 022436 004010
8953 022440 044010
8954 022442 044150
8955 022444 004037 005012
8956 022450 044004
8957
8958
8959
8960
8961
8962
8963 022452 005267 156170
8964 022456 004467 163432
8965 022462 022670
8966 022464 005037 004542

```

: VERIFY THAT I/O PAGE CAN NOT BE TAGGED OR CAUSE
: A READ HIT.

TST100: INC TID ;UPDATE TEST ID
JSR R4,RELCTL ;RELOCATE TEST TO LOW CACHE
.WORD TST101
CLR @#ERROR ;RESET ERROR FLAG
5\$: MOV #6,R2 ;PASS COUNTER
3\$: MOV #5,@#CCR ;ENABLE HIGH CACHE
MOV @#HIGHSP+1776,R3 ;TAG HIGH CACHE LOCATION
MOV @#PSW,R3 ;ATTEMPT TO TAG I/O PAGE
MOV @#PSW,R3 ;ATTEMPT HIT FROM I/O PAGE
BIT #BIT03,@#CHR ;CHECK FOR HIT
BEQ 1\$;NO HIT
MOV #OFF,@#CCR ;DISABLE CACHE
MOV #1,@#ERROR ;SET ERROR FLAG
JSR R0,@#SETEN ;REPORT ERROR
.WORD ^D4
.WORD SEN109 ;I/O PAGE TESTS
.WORD SEN110 ;READ INTO I/O PAGE CAUSED HIT
BR 2\$
1\$: MOV @#HIGHSP+1776,R3 ;READ PREV TAGGED LOCATION
BIT #BIT03,@#CHR ;VERIFY HIT
BNE 2\$;THEIR WAS A HIT
MOV #OFF,@#CCR ;DISABLE CACHE
MOV #1,@#ERROR ;SET ERROR FLAG
JSR R0,@#SETEN ;REPORT ERROR
.WORD ^D4
.WORD SEN109 ;I/O PAGE TESTS
.WORD SEN111 ;READ INTO I/O PAGE INVALIDATED TAGGED LOCATION
2\$: MOV #OFF,@#CCR ;DISABLE CACHE
JSR R0,(R0) ;TAKE SELECTED ACTION ON ERROR
.WORD 3\$-TST100-12+LOWSP ;LOOP ON ERROR
.WORD 4\$-TST100-12+LOWSP ;LOOP ON TEST
4\$: JSR R0,@#LPONTS ;LOOP ON TEST CHECK
.WORD 5\$-TST100-12+LOWSP ;RESTART TEST

: VERIFY THAT I/O PAGE CAN NOT BE TAGGED OR CAUSE
: A READ HIT . OPERATING IN MAINTENANCE MODE.

TST101: INC TID ;UPDATE TEST ID
JSR R4,RELCTL ;RELOCATE TEST TO LOW CACHE
.WORD TST102
CLR @#ERROR ;RESET ERROR FLAG

8968	022470	012702	000006		5\$:	MOV #6,R2	:PASS COUNTER
8969	022474	012737	000001	177750	3\$:	MOV #1,@#CMR	:ENABLE MAINT. MODE
8970	022502	012737	000005	177746		MOV #5,@#CCR	:ENABLE HIGH CACHE
8971	022510	013703	047776			MOV @#HIGHSP+1776,R3	:TAG HIGH BLOCK #1
8972	022514	013703	177776			MOV @#PSW,R3	:ATTEMPT TO TAG I/O PAGE
8973	022520	013703	177776			MOV @#PSW,R3	:ATTEMPT HIT FROM I/O PAGE
8974	022524	032737	000010	177752		BIT #BIT03,@#CHR	:VERIFY NO HIT
8975	022532	001417				BEQ 1\$:NO HIT
8976	022534	012737	000001	004542		MOV #1,@#ERROR	:SET ERROR FLAG
8977	022542	005037	177750			CLR @#CMR	:EXIT MAINT MODE
8978	022546	012737	001015	177746		MOV #OFF,@#CCR	:DISABLE CACHE
8979	022554	004037	005352			JSR R0,@#SETEN	:REPORT ERROR
8980	022560	000006				.WORD ^D6	
8981	022562	040664				.WORD SEN109	:I/O PAGE TESTS
8982	022564	040674				.WORD SEN110	:READ INTO I/O PAGE CAUSED HIT
8983	022566	040732				.WORD SEN112	:WHILE IN MAINTENANCE MODE
8984	022570	000424				BR 2\$	
8985	022572	013703	047776		1\$:	MOV @#HIGHSP+1776,R3	:READ PREV. TAGGED LOCATION
8986	022576	032737	000010	177752		BIT #BIT03,@#CHR	:VERIFY HIT
8987	022604	001016				BNE 2\$:VALID HIT
8988	022606	005037	177750			CLR @#CMR	:EXIT MAINT. MODE
8989	022612	012737	001015	177746		MOV #OFF,@#CCR	:DISABLE CACHE
8990	022620	012737	000001	004542		MOV #1,@#ERROR	:SET ERROR FLAG
8991	022626	004037	005352			JSR R0,@#SETEN	:REPORT ERROR
8992	022632	000006				.WORD ^D6	
8993	022634	040664				.WORD SEN109	:I/O PAGE TESTS
8994	022636	040712				.WORD SEN111	:READ INTO I/O PAGE INVALIDATED TAGGED LOCATION
8995	022640	040732				.WORD SEN112	:WHILE IN MAINTENANCE MODE
8996	022642	005037	177750		2\$:	CLR @#CMR	:EXIT MAINT MODE
8997	022646	012737	001015	177746		MOV #OFF,@#CCR	:DISABLE CACHE
8998	022654	004010				JSR R0,(R0)	:TAKE SELECTED ACTION ON ERROR
8999	022656	044010				.WORD 3\$-TST101-12+LOWSP	:LOOP ON ERROR
9000	022660	044176				.WORD 4\$-TST101-12+LOWSP	:LOOP ON TEST
9001	022662	004037	005012		4\$:	JSR R0,@#LPONTS	:LOOP ON TEST CHECK
9002	022666	044004				.WORD 5\$-TST101-12+LOWSP	:RESTART TEST
9003							
9004							
9005							
9006							

```

9008          .SBTTL VALID BIT STORAGE TESTS
9009          : VERIFY ADDRESS LINES TO CACHE MEMORY CHIPS
9010          : BY WRITING AND READING DATA PATTERNS
9011 022670 005267 155752          TST102: INC TID          ;UPDATE TEST ID
9012 022674 004467 163242          JSR R4,RELCTH    ;RELOCATE TEST TO HIGH CACHE
9013 022700 023170          .WORD TST103
9014 022702 012702 000001          MOV #1,R2       ;PATTERN SELECTOR
9015 022706 005037 005064          CLR @#LOPERR    ;RESET LOOP ON ERROR FLAG
9016 022712 012703 044000          11$: MOV #LOWSP,R3 ;TESTING IN LOW CACHE
9017 022716 005037 052000          CLR @#HIGH1     ;HIT COUNTER
9018 022722 005037 052002          CLR @#HIGH1+2  ;MISS COUNTER
9019 022726 052737 000400 177746 3$: BIS #BIT08,@#CCR ;FLUSH CACHE
9020 022734 032737 010000 177746 1$: BIT #BIT12,@#CCR ;WAIT TILL COMPLETE
9021 022742 001374          BNE 1$
9022 022744 032737 020000 177746  BIT #BIT13,@#CCR ;SELECT VALID BITS SET A
9023 022752 001365          BNE 3$
9024 022754 012737 000011 177746  MOV #11,@#CCR   ;ENABLE LOW CACHE LOCATION
9025 022762 010201          6$: MOV R2,R1     ;PASS COUNTER
9026 022764 012305          2$: MOV (R3)+,R5  ;TAG LOW CACHE LOCATION
9027 022766 005301          DEC R1          ;NO. OF HITS TO CAUSE
9028 022770 001375          BNE 2$
9029 022772 020327 046000          CMP R3,#LOWSP+2000 ;PATTERN COMPLETE
9030 022776 001410          BEQ 7$
9031 023000 010201          MOV R2,R1     ;PASS COUNTER
9032 023002 062703 000002          5$: ADD #2,R3     ;MISS LOCATION
9033 023006 005301          DEC R1        ;NO. OF MISSES TO CAUSE
9034 023010 001374          BNE 5$
9035 023012 020327 046000          CMP R3,#LOWSP+2000 ;PATTERN COMPLETE
9036 023016 001361          BNE 6$        ;NO
9037 023020 012703 044000          7$: MOV #LOWSP,R3 ;CHECK PATTERN
9038 023024 012305          9$: MOV (R3)+,R5  ;LOOK FOR HIT
9039 023026 032737 000010 177752  BIT #BIT03,@#CHR ;WAS THIS A HIT
9040 023034 001006          BNE 8$        ;YES
9041 023036 005237 052002          INC @#HIGH1+2  ;NO
9042 023042 020327 046000          CMP R3,#LOWSP+2000 ;COMPLETE YET
9043 023046 001366          BNE 9$        ;NO
9044 023050 000405          BR 10$       ;YES
9045 023052 005237 052000          8$: INC @#HIGH1   ;HIT COUNTER
9046 023056 020327 046000          CMP R3,#LOWSP+2000 ;COMPLETE YET
9047 023062 001360          BNE 9$        ;NO
9048 023064 012737 001015 177746 10$: MOV #OFF,@#CCR ;DISABLE CACHE
9049 023072 023737 052000 052002  CMP @#HIGH1,@#HIGH1+2 ;IF EQUAL NO ERROR
9050 023100 001406          BEQ 12$      ;NO ERROR
9051 023102 005737 052002          TST @#HIGH1+2 ;ERROR IF ZERO
9052 023106 001413          BEQ 13$
9053 023110 005737 052000          TST @#HIGH1   ;ERROR IF ZERO
9054 023114 001410          BEQ 13$
9055 023116 006302          12$: ASL R2        ;REPAIR FOR NEXT TEST
9056 023120 032702 001000          BIT #BIT09,R2 ;TEST COMPLETE YET
9057 023124 001672          BEQ 11$      ;NO
9058 023126 004037 005012          JSR R0,@#LPONTS ;LOOP ON TEST MODE CHECK
9059 023132 046000          .WORD HIGHSP  ;RESTART TEST
9060 023134 000415          BR TST103    ;NEXT TEST
9061 023136 010237 004542          13$: MOV R2,@#ERROR ;SET ERROR FLAG
9062 023142 006337 004542          ASL @#ERROR   ;ERRORING ADDRESS LINE
9063 023146 005237 004560          INC @#BITFLG ;BIT PRINT MODE

```

9064 023152 004037 005352
9065 023156 000006
9066 023160 040744
9067 023162 040756
9068 023164 041000
9069 023166 000753
9070
9071
9072
9073
9074
9075
9076 023170 005267 155452
9077 023174 004467 162714
9078 023200 023470
9079 023202 012702 000001

JSR R0,@#SETEN
.WORD ^D6
.WORD SEN113
.WORD SEN114
.WORD SEN115
BR 12\$

:REPORT ERROR
:VALID BIT STORAGE TESTS
:OPERATING IN LOW CACHE WITH VALID SET A
:ADDRESS BIT ERROR

:
: VERIFY ADDRESS LINES TO CACHE MEMORY CHIPS
: BY WRITING AND READING DATA PATTERNS
TST103:

INC TID
JSR R4,RELCTL
.WORD TST104
MOV #1,R2

:UPDATE TEST ID
:RELOCATE TEST TO LOW CACHE
:PATTERN SELECTOR

9081	023206	005037	005064			CLR @#LOPERR	:RESET LOOP ON TEST FLAG
9082	023212	012703	046000		11\$:	MOV #HIGHSP,R3	:TESTING IN HIGH CACHE
9083	023216	005037	004550			CLR @#GOOD	:HIT COUNTER
9084	023222	005037	004552			CLR @#BAD	:MISS COUNTER
9085	023226	052737	000400	177746	3\$:	BIS #BIT08,@#CCR	:FLUSH CACHE
9086	023234	032737	010000	177746	1\$:	BIT #BIT12,@#CCR	:WAIT TILL COMPLETE
9087	023242	001374				BNE 1\$	
9088	023244	032737	020000	177746		BIT #BIT13,@#CCR	:SELECT VALID BITS SET B
9089	023252	001765				BEQ 3\$	
9090	023254	012737	000005	177746		MOV #5,@#CCR	:ENABLE HIGH CACHE
9091	023262	010201			6\$:	MOV R2,R1	:PASS COUNTER
9092	023264	012305			2\$:	MOV (R3)+,R5	:TAG HIGH CACHE LOCATION
9093	023266	005301				DEC R1	:NO. OF HITS TO CAUSE
9094	023270	001375				BNE 2\$	
9095	023272	020327	050000			CMP R3,#HIGHSP+2000	:PATTERN COMPLETE
9096	023276	001410				BEQ 7\$	
9097	023300	010201				MOV R2,R1	:PASS COUNTER
9098	023302	062703	000002		5\$:	ADD #2,R3	:MISS LOCATION
9099	023306	005301				DEC R1	:NO. OF MISSES TO CAUSE
9100	023310	001374				BNE 5\$	
9101	023312	020327	050000			CMP R3,#HIGHSP+2000	:PATTERN COMPLETE
9102	023316	001361				BNE 6\$:NO
9103	023320	012703	046000		7\$:	MOV #HIGHSP,R3	:CHECK PATTERN
9104	023324	012305			9\$:	MOV (R3)+,R5	:LOOK FOR HIT
9105	023326	032737	000010	177752		BIT #BIT03,@#CHR	:WAS THIS A HIT
9106	023334	001006				BNE 8\$:YES
9107	023336	005237	004552			INC @#BAD	:NO
9108	023342	020327	050000			CMP R3,#HIGHSP+2000	:COMPLETE YET
9109	023346	001366				BNE 9\$:YES
9110	023350	000405				BR 10\$:NO
9111	023352	005237	004550		8\$:	INC @#GOOD	:HIT COUNTER
9112	023356	020327	050000			CMP R3,#HIGHSP+2000	:COMPLETE YET
9113	023362	001360				BNE 9\$:NO
9114	023364	012737	001015	177746	10\$:	MOV #OFF,@#CCR	:DISABLE CACHE
9115	023372	023737	004550	004552		CMP @#GOOD,@#BAD	:IF EQUAL NO ERROR
9116	023400	001406				BEQ 12\$	
9117	023402	005737	004552			TST @#BAD	:ERROR IF ZERO
9118	023406	001413				BEQ 13\$	
9119	023410	005737	004550			TST @#GOOD	:ERROR IF ZERO
9120	023414	001410				BEQ 13\$	
9121	023416	006302			12\$:	ASL R2	:PREPARE FOR NEXT TEST
9122	023420	032702	001000			BIT #BIT09,R2	:TEST COMPLETE YET
9123	023424	001672				BEQ 11\$:NO
9124	023426	004037	005012			JSR R0,@#LPONTS	:LOOP ON TEST MODE CHECK
9125	023432	044000				.WORD LOWSP	:NEXT TEST
9126	023434	000415				BR TST104	
9127	023436	010237	004542		13\$:	MOV R2,@#ERROR	:SET ERROR FLAG
9128	023442	006337	004542			ASL @#ERROR	:SET ERROR FLAG
9129	023446	005237	004560			INC @#BITFLG	:BIT PRINT MODE
9130	023452	004037	005352			JSR R0,@#SETEN	:REPORT ERROR
9131	023456	000006				.WORD ^D6	
9132	023460	040744				.WORD SEN113	:VALID BIT STORAGE TESTS
9133	023462	041010				.WORD SEN116	:HIGH CACHE ENABLED USING VALID SET B
9134	023464	041000				.WORD SEN115	:ADDRESS BIT ERROR
9135	023466	000753				BR 12\$	
9136							

CACHE DIAG.
CFKKAB.P11

MACY11 30A(1052)
25-JUN-79 13:31

31-OCT-79 15:29 PAGE 72-1
VALID BIT STORAGE TESTS

K 7

SEQ 0088

9137
9138
9139

Line	Addr1	Addr2	Addr3	Addr4	Code	Comments
9141						.SBTTL BYPASS MODE TESTS
9142						VERIFY THAT A WRITE TO MEMORY WHILE CACHE IS IN
9143						BYPASS MODE WILL INVALIDATE THAT LOCATION
9144	023470	005267	155152		TST104:	INC TID ;UPDATE TEST ID
9145	023474	004467	162414			JSR R4,RELCTL ;RELOCATE TEST TO LOW CACHE
9146	023500	023650				.WORD TST105
9147	023502	032737	020000	177746		BIT #BIT13,@#CCR ;SELECT VALID SET A
9148	023510	001407				BEQ 1\$;SET A INUSE NOW
9149	023512	052737	000400	177746		BIS #BIT08,@#CCR ;SELECT VALID SET A
9150	023520	032737	010000	177746	2\$:	BIT #BIT12,@#CCR ;WAIT FOR FLUSH TO COMPLETE
9151	023526	001374				BNE 2\$
9152	023530	012737	000005	177746	1\$:	MOV #5,@#CCR ;ENABLE HIGH CACHE
9153	023536	013703	046000			MOV @#HIGHSP,R3 ;TAG HIGH LOCATION
9154	023542	052737	001000	177746		BIS #BIT09,@#CCR ;ENTER BYPASS MODE
9155	023550	010337	046000			MOV R3,@#HIGHSP ;INVALIDATE LOCATION
9156	023554	042737	001000	177746		BIC #BIT09,@#CCR ;EXIT BYPASS MODE
9157	023562	013703	046000			MOV @#HIGHSP,R3 ;VERIFY LOCATION INVALIDATED
9158	023566	013703	177752			MOV @#CHR,R3
9159	023572	012737	001015	177746		MOV #OFF,@#CCR ;DISABLE CACHE
9160	023600	005037	004542			CLR @#ERROR ;RESET ERROR FLAG
9161	023604	032703	000004			BIT #BIT02,R3 ;READ HIT REGISTER
9162	023610	001411				BEQ 3\$;NO HIT MEANS NO ERROR
9163	023612	005237	004542			INC @#ERROR ;SET ERROR FLAG
9164	023616	004037	005352			JSR R0,@#SETEN ;REPORT ERROR
9165	023622	000010				.WORD ^D8
9166	023624	041030				.WORD SEN117 ;BYPASS MODE TESTS
9167	023626	041040				.WORD SEN118 ;HIGH CACHE LOCATION NOT INVALIDATED
9168	023630	041054				.WORD SEN119 ;BY WRITE TO LOCATION IN BYPASS MODE
9169	023632	041074				.WORD SEN120 ;TEST RUN WITH VALID BITS SET A
9170	023634	004010			3\$:	JSR R0,(R0) ;TAKE SELECTED ACTION ON ERROR
9171	023636	044000				.WORD LOWSP ;LOOP ON ERROR
9172	023640	044000				.WORD LOWSP ;LOOP ON TEST
9173	023642	012737	001015	177746		MOV #OFF,@#CCR ;DISABLE CACHE
9174						
9175						
9176						
9177						
9178						VERIFY THAT A WRITE TO MEMORY WHILE CACHE IS IN
9179						BYPASS MODE WILL INVALIDATE THAT LOCATION
9180	023650	005267	154772		TST105:	INC TID ;UPDATE TEST ID
9181	023654	004467	162234			JSR R4,RELCTL ;RELOCATE TEST TO LOW CACHE
9182	023660	024030				.WORD TST106
9183	023662	032737	020000	177746		BIT #BIT13,@#CCR ;SELECT VALID BITS SET B
9184	023670	001007				BNE 1\$;IN USE NOW
9185	023672	052737	000400	177746		BIS #BIT08,@#CCR ;SELECT VALID BITS B
9186	023700	032737	010000	177746	2\$:	BIT #BIT12,@#CCR ;WAIT TILL COMPLETE
9187	023706	001374				BNE 2\$
9188	023710	012737	000005	177746	1\$:	MOV #5,@#CCR ;ENABLE HIGH CACHE
9189	023716	013703	046000			MOV @#HIGHSP,R3 ;TAG HIGH CACHE LOCATION
9190	023722	052737	001000	177746		BIS #BIT09,@#CCR ;ENTER BYPASS MODE
9191	023730	010337	046000			MOV R3,@#HIGHSP ;INVALIDATE HIGH LOCATION
9192	023734	042737	001000	177746		BIC #BIT09,@#CCR ;EXIT BYPASS MODE
9193	023742	013703	046000			MOV @#HIGHSP,R3 ;VERIFY LOCATION INVALIDATED
9194	023746	013703	177752			MOV @#CHR,R3
9195	023752	012737	001015	177746		MOV #OFF,@#CCR ;DISABLE CACHE
9196	023760	005037	004542			CLR @#ERROR ;RESET ERROR FLAG

```

9197 023764 032703 000004
9198 023770 001411
9199 023772 005237 004542
9200 023776 004037 005352
9201 024002 000010
9202 024004 041030
9203 024006 041040
9204 024010 041054
9205 024012 041114
9206 024014 004010
9207 024016 044000
9208 024020 044000
9209 024022 012737 001015 177746
9210
9211
9212
9213
9214
9215
9216 024030 005267 154612
9217 024034 004467 162054
9218 024040 024162
9219 024042 012737 000005 177746
9220 024050 013703 046000
9221 024054 052737 001000 177746
9222 024062 013703 046000
9223 024066 042737 001000 177746
9224 024074 013703 046000
9225 024100 013703 177752
9226 024104 012737 001015 177746
9227 024112 005037 004542
9228 024116 032703 000004
9229 024122 001411
9230 024124 012737 000001 004542
9231 024132 004037 005352
9232 024136 000006
9233 024140 041030
9234 024142 041130
9235 024144 041146
9236 024146 004010
9237 024150 044000
9238 024152 044000
9239 024154 012737 001015 177746
9240
9241
9242
9243
    
```

```

BIT #BIT02,R3 ;LOOK FOR HIT
BEQ 3$ ;NO HIT MEANS NO ERRORS
INC @#ERROR ;SET ERROR FLAG
JSR R0,@#SETEN ;REPORT ERROR
.WORD ^D8
.WORD SEN117 ;BYPASS MODE TESTS
.WORD SEN118 ;HIGH CACHE LOCATION NOT INVALIDATED
.WORD SEN119 ;BY WRITE TO LOCATION IN BYPASS MODE
.WORD SEN121 ;USING VALID BITS SET B
3$: JSR R0,(R0) ;TAKE SELECTED ACTION ON ERROR
.WORD LOWSP ;LOOP ON ERROR
.WORD LOWSP ;LOOP ON TEST
MOV #OFF,@#CCR ;DISABLE CACHE

:
:
:
TST106: INC TID ;UPDATE TEST ID
JSR R4,RELCTL ;RELOCATE TEST TO LOW CACHE
.WORD TST107
MOV #5,@#CCR ;ENABLE HIGH CACHE
MOV @#HIGHSP,R3 ;TAG HIGH LOCATION
BIS #BIT09,@#CCR ;ENTER BYPASS MODE
MOV @#HIGHSP,R3 ;INVALIDATE TAGGED LOCATION
BIC #BIT09,@#CCR ;EXIT BYPASS MODE
MOV @#HIGHSP,R3 ;LOOK FOR READ MISS
MOV @#CHR,R3
MOV #OFF,@#CCR ;DISABLE CACHE
CLR @#ERROR ;RESET ERROR
BIT #BIT02,R3 ;VERIFY MISS
BEQ 1$ ;YES A MISS *YUCK A PUCK*
MOV #1,@#ERROR ;SET ERROR FLAG
JSR R0,@#SETEN ;REPORT ERROR
.WORD ^D6
.WORD SEN117 ;BYPASS MODE TESTS
.WORD SEN122 ;READING TAGGED LOCATION IN BYPASS MODE
.WORD SEN123 ;CAUSED LOCATION TO INVALIDATE
1$: JSR R0,(R0) ;TAKE SELECTED ACTION ON ERROR
.WORD LOWSP ;LOOP ON ERROR
.WORD LOWSP ;LOOP ON TEST
MOV #OFF,@#CCR ;DISABLE CACHE
    
```

```

9245
9246
9247 024162 005267 154460
9248 024166 004467 161750
9249 024172 024346
9250 024174 012737 000011 177746
9251 024202 013703 044000
9252 024206 013703 000000
9253 024212 013703 000000
9254 024216 013737 177752 004542
9255 024224 012737 001015 177746
9256 024232 032737 000004 004542
9257 024240 001015
9258 024242 012737 000001 004542
9259 024250 004037 005352
9260 024254 000006
9261 024256 041160
9262 024260 041176
9263 024262 041216
9264 024264 004010
9265 024266 046000
9266 024270 046000
9267 024272 000425
9268 024274 042737 000777 004542
9269 024302 012737 000000 004554
9270 024310 013737 004542 004552
9271 024316 012737 000000 004550
9272 024324 005237 004556
9273 024330 004037 005352
927' 024334 000006
9275 024336 041160
9276 024340 041224
9277 024342 041242
9278 024344 000747
9279
9280
9281
9282
9283
9284
9285 024346 005267 154274
9286 024352 004467 161564
9287 024356 024540
9288 024360 012737 000011 177746
9289 024366 013703 044000
9290 024372 013703 004000
9291 024376 013703 004000
9292 024402 013737 177752 004542
9293 024410 012737 001015 177746
9294 024416 032737 000004 004542
9295 024424 001015
9296 024426 012737 000001 004542
9297 024434 004037 005352
9298 024440 000006
9299 024442 041160
9300 024444 041176

```

```

: .SBTTL TAG BIT BASIC READ WRITE TESTS
: TEST WRITES AND READS ZERO FROM TAG STORE CHIPS
TST107: INC TID ;UPDATE TEST ID
JSR R4,RELCTH ;RELOCATE TEST TO HIGH CACHE
.WORD TST110
MOV #11,@#CCR ;ENABLE LOW CACHE
MOV @#LOWSP,R3 ;TAG LOW CACHE LOCATION
MOV @#0,R3 ;TAG LOW 1K
MOV @#0,R3 ;READ HIT TO LOW 1K
MOV @#CHR,@#ERROR ;SAVE HIT REGISTER CONTENTS
MOV #OFF,@#CCR ;DISABLE CACHE
BIT #BIT02,@#ERROR ;VERIFY HIT
BNE 1$ ;LOCATION WAS HIT
MOV #1,@#ERROR ;SET ERROR FLAG
JSR R0,@#SETEN ;REPORT ERROR
.WORD ^D6
.WORD SEN124 ;TAG BIT BASIC READ WRITE TESTS
.WORD SEN125 ;NO HIT FROM READ INTO MEMORY LOCATION
.WORD SEN126 ;ADDRESS 000000
2$: JSR R0,(R0) ;TAKE SELECTED ACTION ON ERROR
.WORD HIGHSP ;LOOP ON ERROR
.WORD HIGHSP ;LOOP ON TEST
BR TST110
1$: BIC #777,@#ERROR ;MASK OFF HIT BITS
MOV #0,@#ADD ;FAILING ADDRESS
MOV @#ERROR,@#BAD ;BAD DATA
MOV #0,@#GOOD ;GOOD DATA
INC @#GOODBD ;ERROR PRINT MODE
JSR R0,@#SETEN ;REPORT ERROR
.WORD ^D6
.WORD SEN124 ;TAG BIT BASIC READ WRITE TESTS
.WORD SEN127 ;WRONG TAG READ FROM HIT REGISTER
.WORD SEN128 ;AS RESULT OF MEMORY READ
BR 2$

```

```

: VERIFY TAG BIT11 CAN BE WRITTEN A ONE
: WITH ALL OTHER TAG BITS WRITTEN ZERO
TST110: INC TID ;UPDATE TEST ID
JSR R4,RELCTH ;RELOCATE TEST TO HIGH CACHE
.WORD TST111
MOV #11,@#CCR ;ENABLE LOW CACHE
MOV @#LOWSP,R3 ;TAG LOW CACHE LOCATION
MOV @#4000,R3 ;TAG LOW 2K
MOV @#4000,R3 ;READ HIT TO LOW 2K
MOV @#CHR,@#ERROR ;SAVE HIT REGISTER CONTENTS
MOV #OFF,@#CCR ;DISABLE CACHE
BIT #BIT02,@#ERROR ;SAVE HIT REGISTER CONTENTS
BNE 1$
MOV #1,@#ERROR ;SET ERROR FLAG
JSR R0,@#SETEN ;REPORT ERROR
.WORD ^D6
.WORD SEN124 ;TAG BIT BASIC READ WRITE TESTS
.WORD SEN125 ;NO HIT FROM READ INTO MEMORY LOCATION

```

9301	024446	041256			.WORD SEN129	:ADDRESS 004000
9302	024450	004010			JSR R0,(R0)	:TAKE SELECTED ACTION ON ERROR
9303	024452	046000			.WORD HIGHSP	:LOOP ON ERROR
9304	024454	046000			.WORD HIGHSP	:LOOP ON TEST
9305	024456	000430			BR TST111	
9306	024460	042737	000777	004542	BIC #777,@#ERROR	:MASK OFF HIT BITS
9307	024466	012737	004000	004554	MOV #4000,@#ADD	:FAILING ADDRESS
9308	024474	013737	004542	004552	MOV @#ERROR,@#BAD	:BAD DATA
9309	024502	012737	001000	004550	MOV #1700,@#GOOD	:GOOD DATA
9310	024510	162737	001000	004542	SUB #1000,@#ERROR	:IF ANY BITS LEFT SET THEN ERROR
9311	024516	005237	004556		INC @#GOODBD	:BIT PRINT MODE
9312	024522	004037	005352		JSR R0,@#SETEN	:REPORT ERROR
9313	024526	000006			.WORD ^D6	
9314	024530	041160			.WORD SEN124	:TAG BIT BASIC READ WRITE TESTS
9315	024532	041224			.WORD SEN127	:WRONG TAG READ FROM HIT REGISTER
9316	024534	041264			.WORD SEN130	:MEMORY ADDRESS AND HIT REGISTER DATA
9317	024536	000744			BR 2\$	
9318						
9319						
9320						
9321						
9322						
9323						
9324	024540	005267	154102		TST111: INC TID	:UPDATE TEST ID
9325	024544	004467	161372		JSR R4,RELCH	:RELOCATE TEST TO HIGH CACHE
9326	024550	024732			.WORD TST112	
9327	024552	012737	000011	177746	MOV #11,@#CCR	:ENABLE LOW CACHE
9328	024560	013703	044000		MOV @#LOWSP,R3	:TAG LOW CACHE LOCATION
9329	024564	013703	010000		MOV @#10000,R3	:TAG LOW 4K
9330	024570	013703	010000		MOV @#10000,R3	:READ HIT INTO LOW 4K
9331	024574	013737	177752	004542	MOV @#CHR,@#ERROR	:SAVE HIT REGISTER CONTENTS
9332	024602	012737	001015	177746	MOV #OFF,@#CCR	:DISABLE CACHE
9333	024610	032737	000004	004542	BIT #BIT02,@#ERROR	:VERIFY HIT
9334	024616	001015			BNE 1\$	
9335	024620	012737	000001	004542	MOV #1,@#ERROR	:SET ERROR FLAG
9336	024626	004037	005352		JSR R0,@#SETEN	:REPORT ERROR
9337	024632	000006			.WORD ^D6	
9338	024634	041160			.WORD SEN124	:TAG BIT BASIC READ WRITE TESTS
9339	024636	041176			.WORD SEN125	:NO HIT FROM READ INTO MEMORY LOCATION
9340	024640	041302			.WORD SEN131	:ADDRESS 010000
9341	024642	004010			JSR R0,(R0)	:TAKE SELECTED ACTION ON ERROR
9342	024644	046000			.WORD HIGHSP	:LOOP ON ERROR
9343	024646	046000			.WORD HIGHSP	:LOOP ON TEST
9344	024650	000430			BR TST112	
9345	024652	042737	000777	004542	BIC #777,@#ERROR	:MASK OFF HIT BITS
9346	024660	013737	004542	004552	MOV @#ERROR,@#BAD	:BAD DATA
9347	024666	012737	010000	004554	MOV #10000,@#ADD	:ADDRESS
9348	024674	012737	002000	004550	MOV #2000,@#GOOD	:GOOD DATA
9349	024702	162737	002000	004542	SUB #2000,@#ERROR	:IF NAY BITS LEFT SET THEN ERROR
9350	024710	005237	004556		INC @#GOODBD	:BIT PRINT MODE
9351	024714	004037	005352		JSR R0,@#SETEN	:REPORT ERROR
9352	024720	000006			.WORD ^D6	
9353	024722	041160			.WORD SEN124	:TAG BIT BASIC READ WRITE TESTS
9354	024724	041224			.WORD SEN127	:WRONG TAG READ FROM HIT REGISTER
9355	024726	041264			.WORD SEN130	:MEMORY ADDRESS AND HIT REGISTER DATA
9356	024730	000744			BR 2\$	

9413	025204	012737	000001	004542
9414	025212	004037	005352	
9415	025216	000006		
9416	025220	041160		
9417	025222	041176		
9418	025224	041316		
9419	025226	004010		
9420	025230	046000		

2\$:

```
MOV #1,@#ERROR
JSR RO,@#SETEN
.WORD ^D6
.WORD SEN124
.WORD SEN125
.WORD SEN133
JSR RO,(RO)
.WORD HIGHSP
```

```
;SET ERROR FLAG
;REPORT ERROR
```

```
;TAG BIT BASIC READ WRITE TESTS
;NO HIT FROM READ INTO MEMORY LOCATION
;ADDRESS 040000
;TAKE SELECTED ACTION ON ERROR
;LOOP ON ERROR
```

```

9422 025232 046000
9423 025234 000430
9424 025236 042737 000777 004542
9425 025244 012737 040000 004554
9426 025252 013737 004542 004552
9427 025260 012737 010000 004550
9428 025266 162737 010000 004542
9429 025274 005237 004556
9430 025300 004037 005352
9431 025304 000006
9432 025306 041160
9433 025310 041224
9434 025312 041264
9435 025314 000744
9436
9437
9438
9439
9440
9441
9442
9443 025316 005267 153324
9444 025322 105767 153334
9445 025326 001414
9446 025330 105767 153327
9447 025334 100011
9448 025336 105767 153330
9449 025342 001006
9450 025344 026737 153324 100000
9451 025352 101002
9452 025354 000167 000204
9453 025360 005067 157154
9454 025364 013703 100000
9455 025370 005767 157144
9456 025374 001402
9457 025376 000167 000162
9458 025402 004467 160534
9459 025406 025564
9460 025410 012737 000011 177746
9461 025416 013703 044000
9462 025422 013703 100000
9463 025426 013703 100000
9464 025432 013737 177752 004542
9465 025440 012737 001015 177746
9466 025446 032737 000004 004542
9467 025454 001015
9468 025456 012737 000001 004542
9469 025464 004037 005352
9470 025470 000006
9471 025472 041160
9472 025474 041176
9473 025476 041324
9474 025500 004010
9475 025502 046000
9476 025504 046000
9477 025506 000426

```

```

1$: .WORD HIGHSP ;LOOP ON TEST
BR TST114
BIC #777,@#ERROR ;MASK OFF HIT BITS
MOV #40000,@#ADD ;MEMORY ADDRESS
MOV @#ERROR,@#BAD ;BAD DATA
MOV #10000,@#GOOD ;GOOD DATA
SUB #10000,@#ERROR ;ANY BITS LEFT SET THEN ERROR
INC @#GOODBD ;BIT PRINT MODE
JSR R0,@#SETEN ;REPORT ERROR
.WORD ^D6
.WORD SEN124 ;TAG BIT BASIC READ WRITE TESTS
.WORD SEN127 ;WRONG TAG READ FROM HIT REGISTER
.WORD SEN130 ;MEMORY ADDRESS AND HIT REGISTER DATA
BR 2$

```

```

: THIS TEST WILL ABORT IF LESS THAN 20K OF MEMORY
: VERIFY TAG BIT15 CAN BE WRITTEN A ONE
: WITH ALL OTHER TAG BIT WRITTEN TO ZERO
TST114: INC TID ;UPDATE TEST ID
TSTB SENV ;IS THIS APT MODE
BEQ 11$ ;NO
TSTB SENVM ;IS THIS AUTO SIZE
BPL 11$ ;NO
TSTB SMAMS1 ;MORE THEN 32K ?
BNE 11$ ;YES
CMP SMADR1,@#100000 ;MORE THEN 16K ?
BHI 11$ ;YES
JMP MAGPRE ;CAN'T RUN TEST
11$: CLR ITTRAP ;RESET TRAP FLAG
MOV @#100000,R3 ;IS MEMORY AVAILABLE
TST ITTRAP ;NO MEMORY IF FLAG SET
BEQ 10$
JMP MAGPRE ;NO MEMORY
10$: JSR R4,RELCTH ;RELOCATE TEST TO HIGH CACHE
.WORD MAGPRE
MOV #11,@#CCR ;ENABLE LOW CACHE
MOV @#LOWSP,R3 ;TAG LOW CACHE
MOV @#100000,R3 ;TAG LOW 16K
MOV @#100000,R3 ;READ HIT TO LOW CACHE
MOV @#CHR,@#ERROR ;SAVE HIT REGISTER CONTENTS
MOV #OFF,@#CCR ;DISABLE CACHE
BIT #BIT02,@#ERROR ;VERIFY HIT
BNE 1$
MOV #1,@#ERROR ;SET ERROR FLAG
JSR R0,@#SETEN ;REPORT ERROR
.WORD ^D6
.WORD SEN124 ;TAG BIT BASIC READ WRITE TESTS
.WORD SEN125 ;NO HIT FROM READ INTO MEMORY LOCATION
.WORD SEN134 ;ADDRESS 100000
2$: JSR R0,(R0) ;TAKE SELECTED ACTION ON ERROR
.WORD HIGHSP ;LOOP ON ERROR
.WORD HIGHSP ;LOOP ON TEST
BR MAGPRE

```

```

9478 025510 042737 000777 004542
9479 025516 012737 100000 004554
9480 025524 013737 004542 004552
9481 025532 012737 020000 004550
9482 025540 162737 020000 004542
9483 025546 004037 005352
9484 025552 000006
9485 025554 041160
9486 025556 041224
9487 025560 041264
9488 025562 000746
9489
9490
9491
9492
9493 025564 012767 077406 144506
9494 025572 012767 077406 144502
9495 025600 012767 077406 144476
9496 025606 012767 077406 144472
9497 025614 012767 077406 144466
9498 025622 012767 077406 144462
9499 025630 012767 077406 144456
9500 025636 012767 077406 144452
9501 025644 012767 000000 144466
9502 025652 012767 000200 144462
9503 025660 012767 000400 144456
9504 025666 012767 000600 144452
9505 025674 012767 007600 144454
9506
9507
9508
9509
9510
9511
9512
9513
9514 025702 005267 152740
9515 025706 105767 152750
9516 025712 001410
9517 025714 105767 152743
9518 025720 100005
9519 025722 105767 152744
9520 025726 001002
9521 025730 000167 000232
9522 025734 012767 000001 151630
9523 025742 012767 002000 144400
9524 025750 005067 156564
9525 025754 013703 100000
9526 025760 005767 156554
9527 025764 001404
9528 025766 005067 151600
9529 025772 000167 000170
9530 025776 004467 160140
9531 026002 026166
9532 026004 012737 000011 177746
9533 026012 013703 044000

```

1\$:

```

BIC #777,@#ERROR
MOV #100000,@#ADD
MOV @#ERROR,@#BAD
MOV #20000,@#GOOD
SUB #20000,@#ERROR
JSR R0,@#SETEN
.WORD ^D6
.WORD SEN124
.WORD SEN127
.WORD SEN130
BR 2$

```

```

;MASK OFF HIT BITS
;MEMORY ADDRESS
;BAD DATA
;GOOD DATA
;IF ANY BITS LEFT SET THEN ERROR
;REPORT ERROR
;TAG BIT BASIC READ WRITE TESTS
;WRONG TAG READ FROM HIT REGISTER
;MEMORY ADDRESS AND HIT REGISTER DATA

```

MAGPRE:

```

MOV #77406,KPDR0
MOV #77406,KPDR1
MOV #77406,KPDR2
MOV #77406,KPDR3
MOV #77406,KPDR4
MOV #77406,KPDR5
MOV #77406,KPDR6
MOV #77406,KPDR7
MOV #0,KPAR0
MOV #200,KPAR1
MOV #400,KPAR2
MOV #600,KPAR3
MOV #7600,KPAR7

```

```

;ALLOW ALL ACCESS TO KERNEL PAGE 0
;ALLOW ALL ACCESS TO KERNEL PAGE 1
;ALLOW ALL ACCESS TO KERNEL PAGE 2
;ALLOW ALL ACCESS TO KERNEL PAGE 3
;ALLOW ALL ACCESS TO KERNEL PAGE 4
;ALLOW ALL ACCESS TO KERNEL PAGE 5
;ALLOW ALL ACCESS TO KERNEL PAGE 6
;ALLOW ALL ACCESS TO KERNEL PAGE 7
;MAP PAGE 0 FOR 0-4K
;MAP PAGE 1 FOR 4-8K
;MAP PAGE 2 FOR 8-8K
;MAP PAGE 3 FOR 12-16K
;MAP PAGE 7 FOR 124-128K

```

...
TST115:

```

THIS TEST USES MEMORY MANAGEMENT
TEST WILL ABORT IF LESS THAN 36K OF MEMORY
VERIFY TAG BIT16 CAN BE WRITTEN TO A ONE
WITH ALL OTHER TAG BITS WRITTEN TO ZERO

```

```

INC TID
TSTB $ENV
BEQ 11$
TSTB $ENVM
BPL 11$
TSTB $MAMS1
BNE 11$
JMP TST116
MOV #1,SRO
MOV #2000,KPAR4
CLR ITTRAP
MOV @#100000,R3
TST ITTRAP
BEQ 5$
CLR SRO
JMP TST116
JSR R4,RELCTH
.WORD TST116
MOV #11,@#CCR
MOV @#LOWSP,R3

```

:NO

```

;UPDATE TEST ID
;IS THIS APT MODE
:NO
;IS THIS AUTO SIZE
;MORE THEN 32K
:YES
;ENABLE MEMORY MANAGEMENT
;MAP PAGE 4 FOR 32-36K
;RESET TRAP FLAG
;ANY MEMORY IN THIS LOCATION
;LOOK FOR TRAP
:NO TRAP
;DISABLE MEMORY MANAGEMENT
;RELOCATE TEST TO HIGH CACHE
;ENABLE LOW CACHE
;TAG LOW CACHE

```

```

9534 026016 013703 100000      MOV @#100000,R3      ;TAG LOW 32K
9535 026022 013703 100000      MOV @#100000,R3      ;READ HIT TO LOW 32K
9536 026026 013737 177752 004542 MOV @#CHR,@#ERROR    ;SAVE HIT REGISTER CONTENTS
9537 026034 012737 001015 177746 MOV #OFF,@#CCR       ;DISABLE CACHE
9538 026042 032737 000004 004542 BIT #BIT02,@#ERROR   ;VERIFY HIT
9539 026050 001017                        BNE 1$
9540 026052 012737 000001 004542 MOV #1,@#ERROR      ;SET ERROR FLAG
9541 026060 004037 005352      JSR R0,@#SETEN      ;REPORT ERROR
9542 026064 000006                        .WORD ^D6
9543 026066 041160                        .WORD SEN124        ;TAG BIT BASIC READ WRITE TESTS
9544 026070 041176                        .WORD SEN125        ;NO HIT FROM READ INTO MEMORY LOCATION
9545 026072 041332                        .WORD SEN135        ;ADDRESS 200000
9546 026074 004010      2$: JSR R0,(R0)         ;TAKE SELECTED ACTION ON ERROR
9547 026076 046000                        .WORD HIGHSP       ;LOOP ON ERROR
9548 026100 046000                        .WORD HIGHSP       ;LOOP ON TEST
9549 026102 005037 177572      CLR @#SRO          ;DISABLE MEMORY MANAGEMENT
9550 026106 000427      BR TST116
9551 026110 042737 000777 004542      BIC #777,@#ERROR   ;MASK OFF HIT BITS
9552 026116 012737 000000 004554      MOV #0,@#ADD       ;MEMORY ADDRESS
9553 026124 013737 004542 004552      MOV @#ERROR,@#BAD  ;BAD DATA
9554 026132 012737 040000 004550      MOV #40000,@#GOOD  ;GOOD DATA
9555 026140 162737 040000 004542      SUB #40000,@#ERROR ;IF ANY BITS LEFT SET THEN ERROR
9556 026146 004037 005352      JSR R0,@#SETEN     ;REPORT ERROR
9557 026152 000010                        .WORD ^D8
9558 026154 041160                        .WORD SEN124        ;TAG BIT BASIC READ WRITE TESTS
9559 026156 041224                        .WORD SEN127        ;WRONG TAG READ FROM HIT REGISTER
9560 026160 041264                        .WORD SEN130        ;MEMORY ADDRESS AND HIT REGISTER DATA
9561 026162 041332                        .WORD SEN135        ;ADDRESS 200000
9562 026164 000743      BR 2$

```

```

:
:
:
:
:
9570 :
9571 026166 005267 152454      TST116: INC TID           ;UPDATE TEST ID
9572 026172 105767 152464      TSTB $ENV          ;IS THIS APT MODE
9573 026176 001411      BEQ 11$            ;NO
9574 026200 105767 152457      TSTB $ENVM         ;IS THIS AUTO SIZE
9575 026204 100006      BPL 11$            ;NO
9576 026206 122767 000002 152456 CMPB #2,$MAMS1     ;MORE THEN 64K ?
9577 026214 101002      BHI 11$            ;YES
9578 026216 000167 000230      JMP TST117
9579 026222 012767 004000 144120      MOV #4000,KPAR4    ;MAP PAGE 4 FOR 64-64K
9580 026230 012767 000001 151334      MOV #1,SRO         ;ENABLE MEMORY MANAGEMENT
9581 026236 005067 156276      CLR ITTRAP         ;RESET TRAP FLAG
9582 026242 013703 100000      MOV @#100000,R3    ;ANY MEMORY IN THIS LOCATION
9583 026246 005767 156266      TST ITTRAP         ;LOOK FOR TRAP
9584 026252 001403      BEQ 5$             ;NO TRAP
9585 026254 005067 151312      CLR SRO            ;DISABLE MEMORY MANAGEMENT
9586 026260 000474      BR TST117
9587 026262 004467 157654      JSR R4,RELCTH      ;RELOCATE TEST TO HIGH CACHE
9588 026266 026452      .WORD TST117
9589 026270 012737 000011 177746      MOV #11,@#CCR      ;ENABLE LOW CACHE

```

```

9590 026276 013703 044000
9591 026302 013703 100000
9592 026306 013703 100000
9593 026312 013737 177752 004542
9594 026320 012737 001015 177746
9595 026326 032737 000004 004542
9596 026334 001017
9597 026336 012737 000001 004542
9598 026344 004037 005352
9599 026350 000006
9600 026352 041160
9601 026354 041176
9602 026356 041340
9603 026360 004010
9604 026362 046000
9605 026364 046000
9606 026366 005037 177572
9607 026372 000427
9608 026374 042737 000777 004542
9609 026402 012737 000000 004554
9610 026410 013737 004542 004552
9611 026416 012737 100000 004550
9612 026424 162737 100000 004542
9613 026432 004037 005352
9614 026436 000010
9615 026440 041160
9616 026442 041224
9617 026444 041264
9618 026446 041340
9619 026450 000743
9620
9621
9622
9623
    
```

2\$:

1\$:

```

MOV @#LOWSP,R3
MOV @#100000,R3
MOV @#100000,R3
MOV @#CHR,@#ERROR
MOV #OFF,@#CCR
BIT #BIT02,@#ERROR
BNE 1$
MOV #1,@#ERROR
JSR RO,@#SETEN
.WORD ^D6
.WORD SEN124
.WORD SEN125
.WORD SEN136
JSR RO,(RO)
.WORD HIGHSP
.WORD HIGHSP
CLR @#SR0
BR TST117
BIC #777,@#ERROR
MOV #0,@#ADD
MOV @#ERROR,@#BAD
MOV #100000,@#GOOD
SUB #100000,@#ERROR
JSR RO,@#SETEN
.WORD ^D8
.WORD SEN124
.WORD SEN127
.WORD SEN130
.WORD SEN136
BR 2$
    
```

```

;TAG LOW CACHE
;TAG LOW 64K
;READ HIT TO LOW 64K
;SAVE HIT REGISTER CONTENTS
;DISABLE CACHE
;VERIFY HIT

;SET ERROR FLAG
;REPORT ERROR

;TAG BIT BASIC READ WRITE TESTS
;NO HIT FROM READ INTO MEMORY LOCATION
;ADDRESS 40000
;TAKE SELECTED ACTION ON ERROR
;LOOP ON ERROR
;LOOP ON TEST
;DISABLE MEMORY MANAGEMENT

;MASK OFF HIT BITS
;MEMORY ADDRESS
;BAD DATA
;GOOD DATA
;IF NAY BITS LEFT SET THEN ERROR
;REPORT ERROR

;TAG BIT BASIC READ WRITE TESTS
;WRONG TAG READ FROM HIT REGISTER
;MEMORY ADDRESS AND HIT REGISTER DATA
;ADDRESS 40000
    
```

.SBTTL CACHE MEMORY ADDRESS TEST

9625
9626
9627
9628
9629
9630
9631
9632
9633
9634 026452 005267 152170
9635 026456 004467 157432
9636 026462 027204
9637 026464 012701 002000
9638 026470 012702 046000
9639 026474 010122
9640 026476 062701 000002
9641 026502 020227 050000
9642 026506 001372
9643 026510 012737 000005 177746
9644 026516 012702 046000
9645 026522 012203
9646 026524 020227 050000
9647 026530 001374
9648 026532 005037 177744
9649 026536 012701 050000
9650 026542 012702 046000
9651 026546 000413
9652 026550 005037 177744
9653 026554 012203
9654 026556 052737 000210 177746
9655 026564 013721 177744
9656 026570 042737 000200 177746
9657 026576 012705 044064
9658 026602 012737 001015 177746
9659 026610 011525
9660 026612 020527 044124
9661 026616 001374
9662 026620 012737 000005 177746
9663 026626 020227 050000
9664 026632 001346
9665 026634 012701 050000
9666 026640 012703 002000
9667 026644 012702 046000
9668 026650 042711 100000
9669 026654 005711
9670 026656 001012
9671 026660 062701 000002
9672 026664 062702 000002
9673 026670 062703 000002
9674 026674 020227 050000
9675 026700 001363
9676 026702 000450
9677 026704 032711 000040
9678 026710 001411
9679 026712 016205 004000
9680 026716 052737 002000 177746

```

:      VERIFY CACHE MEMORY ADDRESS LINES BY WRITING
:      CACHE ADDRESS INTO CACHE DATA

TST117:  INC TID           ;UPDATE TEST ID
        JSR R4,RELCTL    ;RELOCATE TEST TO LOW CACHE
        .WORD TST120
        MOV #2000,R1     ;CACHE ADDRESS FOR DATA
        MOV #HIGHSP,R2  ;HIGH CACHE PATTERN BLOCK
2$:     MOV R1,(R2)+     ;GENERATE PATTERN
        ADD #2,R1
        CMP R2,#HIGHSP+2000
        BNE 2$
        MOV #5,@#CCR    ;ENABLE HIGH CACHE
        MOV #HIGHSP,R2  ;TAG HIGH BLOCK #2
3$:     MOV (R2)+,R3     ;CAUSE HIGH LOCATION TO TAG
        CMP R2,#HIGHSP+2000
        BNE 3$
        CLR @#CMPE      ;RESET CACHE ERROR REGISTER
        MOV #LOW1,R1    ;POINTER TO WRITE WRONG FLAG BLOCK
        MOV #HIGHSP,R2  ;POINTER TO TAGGED CACHE
        BR 22$
4$:     CLR @#CMPE      ;RESET ERROR REGISTER
        MOV (R2)+,R3    ;CAUSE READ TO CACHE
        BIS #BIT07+10,@#CCR ;ENABLE ABORT FOR ERROR READ
        MOV @#CMPE,(R1)+ ;SAVE ERROR INFO
        BIC #BIT07,@#CCR ;DISABLE ABORT
22$:    MOV #4$-TST117-12+LOWSP,R5 ;UNTAG LOW CACHE
        MOV #OFF,@#CCR  ;BYPASS MODE TO INVALIDATE
20$:    MOV (R5),(R5)+
        CMP R5,#20$-TST117-12+LOWSP
        BNE 20$
        MOV #5,@#CCR    ;ENABLE HIGH CACHE
        CMP R2,#HIGHSP+2000
        BNE 4$
        MOV #LOW1,R1    ;POINTER TO WRITE WRONG FLAG
        MOV #2000,R3    ;EXPECTED DATA
        MOV #HIGHSP,R2  ;POINTER TO TAGGED CACHE
5$:     BIC #BIT15,(R1)  ;MASK OFF BIT 15
        TST (R1)        ;ANY ERROR
        BNE 6$         ;YES
9$:     ADD #2,R1        ;UPDATE POINTERS
        ADD #2,R2
        ADD #2,R3
        CMP R2,#HIGHSP+2000 ;PASS COMPLETE YET
        BNE 5$         ;NO
        BR 12$
6$:     BIT #BIT05,(R1)  ;ANY TAG PARITY ERROR
        BEQ 7$
        MOV 4000(R2),R5 ;UNTAG LOCATION
        BIS #BIT10,@#CCR ;ENABLE WRITE WRONG TAG
    
```

9681	026724	011205			MOV (R2),R5	:TAG LOCATION WRONG PARITY
9682	026726	042737	002000	177746	BIC #BIT10,@#CCR	:DISABLE WRITE WRONG
9683	026734	022711	000300		CMP #300,(R1)	:IF BOTH BYTES FAILED WRITE WRONG WORD
9684	026740	101010			BHI 8\$	
9685	026742	052737	000100	177746	BIS #BIT06,@#CCR	:ENABLE WRITE WRONG DATA
9686	026750	010312			MOV R3,(R2)	
9687	026752	012737	000005	177746	MOV #5,@#CCR	:DISABLE WRITE WRONG
9688	026760	000737			BR 9\$	
9689	026762	032711	000100		BIT #BIT06,(R1)	:ANY LOW BYTE ERROR
9690	026766	001405			BEQ 19\$:NO
9691	026770	052737	000100	177746	BIS #BIT06,@#CCR	:ENABLE WRITE WRONG DATA
9692	026776	110312			MOVB R3,(R2)	
9693	027000	000764			BR 10\$	
9694	027002	032711	000200		BIT #BIT07,(R1)	:ANY HIGH BYTE ERROR
9695	027006	001761			BEQ 10\$	
9696	027010	052737	000100	177746	BIS #BIT06,@#CCR	:ENABLE WRITE WRONG DATA
9697	027016	110362	000001		MOVB R3,1(R2)	
9698	027022	000753			BR 10\$	
9699	027024	010703			MOV PC,R3	:UNTAG LOW CACHE
9700	027026	014313			MOV -(R3),(R3)	:CAUSE WRITE IN BYPASS MODE
9701	027030	020327	044000		CMP R3,#LOWSP	
9702	027034	001374			BNE 13\$	
9703	027036	012701	050000		MOV #LOW1,R1	:DATA FOR COMPARE
9704	027042	012702	046000		MOV #HIGHSP,R2	:POINTER TO TAGGED CACHE
9705	027046	012737	000005	177746	MOV #5,@#CCR	:ENABLE HIGH CACHE
9706	027054	005037	177744		CLR @#CMPE	:RESET ERROR REGISTER
9707	027060	012221			MOV (R2)+,(R1)+	:READ PATTERN FROM CACHE
9708	027062	020227	050000		CMP R2,#HIGHSP+2000	
9709	027066	001374			BNE 30\$	
9710	027070	012737	001015	177746	MOV #OFF,@#CCR	:DISABLE CACHE
9711	027076	012701	050000		MOV #LOW1,R1	:DATA READ
9712	027102	012702	002000		MOV #2000,R2	:DATA WRITTEN
9713	027106	021102			CMP (R1),R2	
9714	027110	001010			BNE 23\$	
9715	027112	062701	000002		ADD #2,R1	
9716	027116	062702	000002		ADD #2,R2	
9717	027122	020127	052000		CMP R1,#LOW1+2000	
9718	027126	001367			BNE 21\$	
9719	027130	000422			BR 25\$	
9720	027132	012737	000001	004542	MOV #1,@#ERROR	:SET ERROR FLAG
9721	027140	010237	004554		MOV R2,@#ADD	:FAILING CACHE ADDRESS
9722	027144	010237	004550		MOV R2,@#GOOD	:EXPECTED DATA
9723	027150	011137	004552		MOV (R1),@#BAD	:DATA READ
9724	027154	005237	004556		INC @#GOODBD	:ENTER DATA PRINT MODE
9725	027160	004037	005352		JSR R0,@#SETEN	:REPORT ERROR
9726	027164	000006			.WORD ^D6	
9727	027166	041346			.WORD SEN137	:CACHE MEMORY ADDRESS TEST
9728	027170	041360			.WORD SEN138	:WRITING CACHE ADDRESS INTO CACHE DATA
9729	027172	041376			.WORD SEN139	:CACHE MEMORY LOCATION AND DATA
9730	027174	000746			BR 24\$	
9731	027176	004010			JSR R0,(R0)	:TAKE SELECTED ACTION ON ERROR
9732	027200	044362			.WORD 14\$-TST117-12+LOWSP	:LOOP ON ERROR
9733	027202	044362			.WORD 14\$-TST117-12+LOWSP	:LOOP ON TEST
9734						
9735						
9736						

```

9738          .SBTTL UNIBUS EXERCISER DMA TEST
9739          : CHECK THAT DMA WRITE INVALIDATES CACHE
9740
9741 027204 005267 151436      TST120:      INC TID          ;UPDATE TEST ID
9742 027210 004467 156700      JSR R4,RELCTL   ;RELOCATE TEST TO HIGH CACHE
9743 027214 027442              .WORD TST121
9744 027216 005037 004540      CLR @#ITTRAP   ;RESET TRAP FLAG
9745 027222 005737 170006      TST @#BECR1    ;LOOK FOR UNIBUS EXERCISER
9746 027226 005737 004540      TST @#ITTRAP   ;LOOK FOR TRAP
9747 027232 001402              BEQ 10$
9748 027234 000167 000202      JMP TST121
9749 027240 005737 000650      10$: TST @#SPASS    ;PRINT ON FIRST PASS ONLY
9750 027244 001004              BNE 11$
9751 027246 012702 004562      MOV #NOUBE,R2
9752 027252 004737 006064      JSR PC,@#TYPE
9753 027256 012737 000005 177746 11$: MOV #5,@#CCR    ;ENABLE HIGH CACHE
9754 027264 012701 052000      MOV #HIGH1,R1  ;TAG HIGH CACHE BLOCK #2
9755 027270 012102              MOV (R1)+,R2   ;CAUSE WRITE TO CACHE
9756 027272 020127 054000      CMP R1,#HIGH1+2000
9757 027276 001374              BNE 1$
9758 027300 012701 046000      MOV #HIGHSP,R1 ;TAG HIGH BLOCK #1
9759 027304 012102              MOV (R1)+,R2   ;CAUSE WRITE TO CACHE
9760 027306 020127 050000      CMP R1,#HIGHSP+2000
9761 027312 001374              BNE 2$
9762 027314 012737 046000 170004 MOV #HIGHSP,@#BEBA ;EXER. ADDRESS
9763 027322 012737 177000 170002 MOV #-1000,@#BECC ;EXER. TRANSFER COUNT
9764 027330 012737 177777 170000 MOV #177777,@#BEDA ;DATA FOR WRITE
9765 027336 012737 000000 170016 MOV #0,@#BECR2   ;CONTROL REGISTER #2
9766 027344 012737 003045 170006 MOV #3045,@#BECR1 ;CONTROL REGISTER #1
9767 027352 105737 170006      3$: TSTB @#BECR1  ;WAIT FOR EXER. TO COMPLETE
9768 027356 100375              BPL 3$
9769 027360 012701 046000      MOV #HIGHSP,R1 ;LOOK FOR READ HITS NOW
9770 027364 012102              MOV (R1)+,R2   ;READ PREV TAGGED LOCATIONS
9771 027366 053737 177752 004542 BIS @#CHR,@#ERROR ;SAVE HIT REGISTER
9772 027374 020127 050000      CMP R1,#HIGHSP+2000
9773 027400 001371              BNE 4$
9774 027402 042737 177700 004542 BIC #177700,@#ERROR ;MASK FOR HITS ONLY
9775 027410 012737 001015 177746 MOV #OFF,@#CCR  ;DISABLE CACHE
9776 027416 004037 005352      JSR R0,@#SETEN ;REPORT ERROR IF ANY
9777 027422 000010              .WORD ^D8
9778 027424 041412              .WORD SEN140   ;UNIBUS EXERCISER DMA TEST
9779 027426 041424              .WORD SEN141   ;TAGGED CACHE DID NOT INVALIDATE BY DMA WRITE
9780 027430 037330              .WORD SEN56    ;TESTING IN HIGH CACHE
9781 027432 041446              .WORD SEN142   ;USING UNIBUS EXERCISER MODULE
9782 027434 004010              JSR R0,(R0)    ;TAKE SELECTED ACTION ON ERROR
9783 027436 044000              .WORD LOWSP   ;LOOP ON ERROR
9784 027440 044000              .WORD LOWSP   ;LOOP ON TEST
9785
9786
9787
9788
9789          .SBTTL DATA BIT MARCH PATTERN TEST
9790          : TEST OPERATES IN HIGH CACHE FOR TESTING LOW CACHE
9791          : TEST WRITES BACKGROUND OF ALL ZEROS
9792          : READ A LOCATION STARTING AT CACHE ADDRESS 0000
9793          : COMPLEMENTS DATA

```

CACHE DIAG.
CFKKAB.P11

MACY11 30A(1052)
25-JUN-79 13:31

31-OCT-79 15:29 PAGE 73-7
DATA BIT MARCH PATTERN TEST

SEQ 0102

9794
9795
9796
9797
9798
9799
9800
9801
9802
9803
9804
9805
9806
9807
9808
9809
9810
9811
9812
9813
9814
9815
9816
9817
9818
9819
9820
9821
9822
9823
9824
9825
9826
9827
9828
9829
9830
9831
9832
9833
9834
9835
9836
9837
9838
9839
9840
9841
9842
9843
9844
9845
9846
9847
9848
9849

027442 005267 151200
027446 004467 156470
027452 030564
027454 005037 004542
027460 005037 005064
027464 012737 000011
027472 012701 050000
027476 012103
027500 022701 052000
027504 001374
027506 012701 044000
027512 005021
027514 020127 046000
027520 001374
027522 012701 044000
027526 012702 052000
027532 005037 177744
027536 012103
027540 052737 000215
027546 013722 177744
027552 012737 000011
027560 020127 046000
027564 001362
027566 012701 044000
027572 012702 052000
027576 005712
027600 001010
027602 062702 000002
027606 062701 000002
027612 020127 046000
027616 001367
027620 000454
027622 032712 000040
027626 001411
027630 016103 004000
027634 052737 002000
027642 011103
027644 042737 002000
027652 032712 000100
027656 001410
027660 052737 000100
027666 112711 000000
027672 042737 000100
027700 032712 000200

177746

177746

177746

177746

177746
177746
177746
177746
177746
177746

.....

WRITES THE COMPLEMENT TO CACHE
READS COMPLEMENTED DATA AND ERROR CHECKS
PROCEEDS TO NEXT CACHE LOCATION
AFTER ALL HIGH CACHE HAS BIN WRITTEN TO COMPLEMENT
TEST THEN STARTS READING FROM LAST LOCATION
COMPLEMENTS DATA
WRITES COMPLEMENTED DATA TO CACHE
READS AND ERROR CHECKS DATA
PROCEEDS TO NEXT LOWER LOCATION
UNTIL START OF HIGH CACHE

```

TST121:    INC TID                                ;UPDATE TEST ID
           JSR R4,RELCTH                        ;RELOCATE TEST TO HIGH CACHE
           .WORD TST122
           CLR @WERROR                            ;RESET ERROR FLAG
           CLR @WLOPERR                          ;RESET LOOP ON TEST FLAG
           MOV #11,@WCCR                          ;ENABLE LOW CACHE
           MOV #LOW1,R1                          ;TAG LOW BLOCK #1
1$:        MOV (R1)+,R3                            ;CAUSE READ TO LOW BLOCK
           CMP #LOW1+2000,R1
           BNE 1$
           MOV #LOWSP,R1                          ;WRITE BACKGROUND TO LOW CACHE
2$:        CLR (R1)+                              ;CAUSE READ TO TAG ,THEN WRITE BACKGROUND
           CMP R1,#LOWSP+2000
           BNE 2$
           MOV #LOWSP,R1                          ;ADDRESS OF TEST BLOCK
           MOV #HIGH1,R2                         ;ADDRESS OF ERROR BLOCK
3$:        CLR @WCMPE                             ;RESET ERROR REGISTER
           MOV (R1)+,R3                           ;READ BACKGROUND SAVE ERROR INFO
           BIS #215,@WCCR                         ;SET ABORT FOR ERROR READ
           MOV @WCMPE,(R2)+                       ;SAVE ERROR INFO.
           MOV #11,@WCCR                          ;DISABLE ABORT
           CMP R1,#LOWSP+2000
           BNE 3$
           MOV #LOWSP,R1                          ;WRITE BACKGROUND
           MOV #HIGH1,R2
33$:       TST (R2)                               ;ANY ERROR FOR LOCATION
           BNE 4$                                  ;YES
9$:        ADD #2,R2                               ;POINT TO NEXT
           ADD #2,R1
           CMP R1,#LOWSP+2000
           BNE 33$
           BR 10$
4$:        BIT #BIT05,(R2)                       ;ANY TAG ERROR
           BEQ 5$
           MOV 4000(R1),R3                         ;UNTAG LOCATION
           BIS #BIT10,@WCCR                       ;ENABLE WRITE WRONG TAG
           MOV (R1),R3                             ;TAG LOCATION WRONG
           BIC #BIT10,@WCCR                       ;DISABLE WRITE WRONG TAG
5$:        BIT #BIT06,(R2)                       ;ANY LOW BYTE FAILURE
           BEQ 6$                                  ;NO
           BIS #BIT06,@WCCR                       ;ENABLE WRITE WRONG DATA
           MOVB #0,(R1)                            ;WRITE WRONG DATA LOW BYTE
           BIC #BIT06,@WCCR                       ;DISABLE WRITE WRONG DATA
6$:        BIT #BIT07,(R2)                       ;ANY HIGH BYTE FAILURE

```

9850	027704	001406				BEQ 7\$	
9851	027706	052737	000100	177746		BIS #BIT06,@#CCR	:WRITE WRONG DATA
9852	027714	112761	000000	000001		MOVB #0,1(R1)	:WRITE WRONG DATA HIGH BYTE
9853	027722	012737	001015	177746	7\$:	MOV #OFF,@#CCR	:DISABLE CACHE
9854	027730	010703				MOV PC,R3	:CORRECT WRONG PARITY HIGH
9855	027732	014313			8\$:	MOV -(R3),(R3)	
9856	027734	020327	046146			CMP R3,#4\$-TST121-12+HIGHSP	
9857	027740	001374				BNE 8\$	
9858	027742	012737	000011	177746		MOV #11,@#CCR	:ENABLE LOW CACHE
9859	027750	000714				BR 9\$	
9860	027752	012701	044000		10\$:	MOV #LOWSP,R1	:START TEST
9861	027756	012702	052000			MOV #HIGH1,R2	
9862	027762	011103			11\$:	MOV (R1),R3	:READ DATA
9863	027764	005103				COM R3	:COMPLEMENT DATA
9864	027766	010311				MOV R3,(R1)	:WRITE DATA
9865	027770	005037	177744			CLR @#CMPE	:RESET ERROR REGISTER
9866	027774	011112				MOV (R1),(R2)	:READ COMP. DATA , SAVE
9867	027776	005737	177744			TST @#CMPE	:ANY ERRORS
9868	030002	001010				BNE 12\$:YES
9869	030004	062701	000002		16\$:	ADD #2,R1	:ADJ. POINTER
9870	030010	062702	000002			ADD #2,R2	:ADJ. POINTER
9871	030014	020127	046000			CMP R1,#LOWSP+2000	
9872	030020	001360				BNE 11\$	
9873	030022	000453				BR 21\$	
9874	030024	052737	000215	177746	12\$:	BIS #215,@#CCR	:SET ABORT FOR ERROR READ
9875	030032	013705	177744			MOV @#CMPE,R5	:SAVE ERROR
9876	030036	012737	000011	177746		MOV #11,@#CCR	:DISABLE ABORT
9877	030044	032705	000100			BIT #BIT06,R5	:ANY LOW BYTE ERROR
9878	030050	001407				BEQ 13\$:NO
9879	030052	052737	000100	177746		BIS #BIT06,@#CCR	:ENABLE WRITE WRONG DATA
9880	030060	110311				MOVB R3,(R1)	:WRITE WRONG LOW BYTE
9881	030062	042737	000100	177746		BIC #BIT06,@#CCR	:DISABLE WRITE WRONG
9882	030070	032705	000200		13\$:	BIT #BIT07,R5	:ANY HIGH BYTE ERROR
9883	030074	001411				BEQ 14\$:NO
9884	030076	000303				SWAB R3	
9885	030100	052737	000100	177746		BIS #BIT06,@#CCR	:ENABLE WRITE WRONG DATA
9886	030106	110361	000001			MOVB R3,1(R1)	:WRITE WRONG DATA HIGH BYTE
9887	030112	042737	000100	177746		BIC #BIT06,@#CCR	:DISABLE WRITE WRONG
9888	030120	012737	001015	177746	14\$:	MOV #OFF,@#CCR	:DISABLE CACHE
9889	030126	010705				MOV PC,R5	:CORRECT WRONG PARITY HIGH CACHE
9890	030130	014515			15\$:	MOV -(R5),(R5)	
9891	030132	020527	046350			CMP R5,#12\$-TST121-12+HIGHSP	
9892	030136	001374				BNE 15\$	
9893	030140	012737	000011	177746		MOV #11,@#CCR	:ENABLE LOW CACHE
9894	030146	011112				MOV (R1),(R2)	:READ DATA
9895	030150	000715				BR 16\$	
9896	030152	012737	001015	177746	21\$:	MOV #OFF,@#CCR	:DISABLE CACHE
9897	030160	012701	052000			MOV #HIGH1,R1	:READ AND REPORT ERROR
9898	030164	012737	177777	004550	17\$:	MOV #177777,@#GOOD	:GOOD DATA
9899	030172	010137	004554			MOV R1,@#ADD	:MEMORY ADDRESS
9900	030176	162737	052000	004554		SUB #HIGH1,@#ADD	:CACHE ADDRESS
9901	030204	012137	004552			MOV (R1)+,@#BAD	:READ DATA
9902	030210	013737	004552	004542		MOV @#BAD,@#ERROR	:ERROR CHECK
9903	030216	005237	004542			INC @#ERROR	:NO ERROR IF ZERO
9904	030222	001004				BNE 18\$	
9905	030224	020127	054000		19\$:	CMP R1,#HIGH1+2000	

9906	030230	001355				BNE 17\$	
9907	030232	000413				BR 20\$	
9908	030234	005237	004556		18\$:	INC @#GOODBD ;DATA PRINT MODE	
9909	030240	004037	005352			JSR R0,@#SETEN ;REPORT ERROR	
9910	030244	000004				.WORD ^D4	
9911	030246	041460				.WORD SEN143 ;DATA BIT MARCH PATTERN TEST	
9912	030250	041474				.WORD SEN144 ;CACHE ADDRESS , DATA EXPECTED , DATA READ	
9913	030252	053737	004542	005064		BIS @#ERROR,@#LOPERR ;SET HAD ERROR FLAG	
9914	030260	000761				BR 19\$	
9915	030262	012701	046000		20\$:	MOV #LOWSP+2000,R1	
9916	030266	014103			22\$:	MOV -(R1),R3 ;READ DATA	
9917	030270	005103				COM R3 ;COMPLEMENT DATA	
9918	030272	010311				MOV R3,(R1) ;WRITE COMPLEMENTED DATA	
9919	030274	020127	044000			CMP R1,#LOWSP	
9920	030300	001372				BNE 22\$	
9921	030302	012701	046000			MOV #LOWSP+2000,R1	
9922	030306	012702	054000			MOV #HIGH1+2000,R2	
9923	030312	014142			24\$:	MOV -(R1),-(R2) ;READ AND SAVE DATA	
9924	030314	005737	177744			TST @#CMPE	
9925	030320	001004				BNE 23\$	
9926	030322	020127	044000		27\$:	CMP R1,#LOWSP	
9927	030326	001371				BNE 24\$	
9928	030330	000450				BR 28\$	
9929	030332	052737	000215	177746	23\$:	BIS #215,@#CCR ;SET ABORT FOR ERROR READ	
9930	030340	013705	177744			MOV @#CMPE,R5 ;READ ERROR REGISTER	
9931	030344	012737	000011	177746		MOV #11,@#CCR ;DISABLE ABORT	
9932	030352	032705	000100			BIT #BIT06,R5 ;DID HIGH BYTE FAIL	
9933	030356	001407				BEQ 25\$;NO	
9934	030360	052737	000100	177746		BIS #BIT06,@#CCR ;SET WRITE WRONG DATA	
9935	030366	111211				MOV R2,(R1) ;WRITE WRONG LOW BYTE	
9936	030370	042737	000100	177746		BIC #BIT06,@#CCR ;DISABLE WRITE WRONG	
9937	030376	032705	000200		25\$:	BIT #BIT07,R5 ;DID HIGH BYTE FAIL	
9938	030402	001406				BEQ 46\$;NO	
9939	030404	052737	000100	177746		BIS #BIT06,@#CCR ;ENABLE WRITE WRONG DATA	
9940	030412	116261	000001	000001		MOV R2,R1 ;WRITE WRONG HIGH BYTE	
9941	030420	012737	001015	177746	46\$:	MOV #OFF,@#CCR ;DISABLE CACHE	
9942	030426	010703				MOV PC,R3 ;CORRECT WRONG PARITY	
9943	030430	014313			26\$:	MOV -(R3),(R3)	
9944	030432	020327	046656			CMP R3,#23\$-TST121-12+HIGHSP	
9945	030436	001374				BNE 26\$	
9946	030440	012737	000011	177746		MOV #11,@#CCR ;ENABLE CACHE	
9947	030446	011112				MOV (R1),(R2) ;READ DATA	
9948	030450	000724				BR 27\$	
9949	030452	012737	001015	177746	28\$:	MOV #OFF,@#CCR ;DISABLE CACHE	
9950	030460	012701	054000			MOV #HIGH1+2000,R1 ;CHECK FOR ERRORS	
9951	030464	012737	000000	004550		MOV #0,@#GOOD ;GOOD DATA	
9952	030472	014137	004542		29\$:	MOV -(R1),@#ERROR ;READ DATA	
9953	030476	010137	004554			MOV R1,@#ADD ;MEMORY ADDRESS	
9954	030502	013737	004542	004552		MOV @#ERROR,@#BAD ;BAD DATA	
9955	030510	005237	004556			INC @#GOODBD ;SET DATA PRINT MODE	
9956	030514	162737	052000	004554		SUB #HIGH1,@#ADD ;CACHE ADDRESS	
9957	030522	004037	005352			JSR R0,@#SETEN ;REPORT ERROR IF ANY	
9958	030526	000004				.WORD ^D4	
9959	030530	041460				.WORD SEN143 ;DATA BIT MARCH PATTERN TEST	
9960	030532	041474				.WORD SEN144 ;CACHE ADDRESS , DATA EXPECTED , DATA READ	
9961	030534	053737	004542	005064		BIS @#ERROR,@#LOPERR ;DEL ERROR LOCATION	

9962	030542	020127	052000	
9963	030546	001351		
9964	030550	013737	005064	004542
9965	030556	004010		
9966	030560	046000		
9967	030562	046000		
9968				
9969				
9970				
9971				
9972				
9973				
9974				
9975				
9976				
9977				
9978				
9979				
9980				
9981				
9982				
9983				
9984				
9985	030564	005267	150056	
9986	030570	004467	155320	
9987	030574	031706		
9988	030576	005037	004542	
9989	030602	005037	005064	
9990	030606	012737	000005	177746
9991	030614	012701	052000	
9992	030620	012103		
9993	030622	022701	054000	
9994	030626	001374		
9995	030630	012701	046000	
9996	030634	005021		
9997	030636	020127	050000	
9998	030642	001374		
9999	030644	012701	046000	
10000	030650	012702	050000	
10001	030654	005037	177744	
10002	030660	012103		
10003	030662	052737	000215	177746
10004	030670	013722	177744	
10005	030674	012737	000005	177746
10006	030702	020127	050000	
10007	030706	001362		
10008	030710	012701	046000	
10009	030714	012702	050000	
10010	030720	005712		
10011	030722	001010		
10012	030724	062702	000002	
10013	030730	062701	000002	
10014	030734	020127	052000	
10015	030740	001367		
10016	030742	000454		
10017	030744	032712	000040	

```

CMP R1,#HIGH1
BNE 29$
MOV @#LOPERR,@#ERROR
JSR R0,(R0) ;TAKE SELECTED ACTION ON ERROR
;WORD HIGHSP ;LOOP ON ERROR
;WORD HIGHSP ;LOOP ON TEST
;SBTTL DATA BIT MARCH PATTERN TEST
TEST OPERATES IN LOW CACHE FOR TESTING HIGH CACHE
TEST WRITES BACKGROUND OF ALL ZEROS
READ A LOCATION STARTING AT CACHE ADDRESS 0000
COMPLEMENTS DATA
WRITES THE COMPLEMENT TO CACHE
READS COMPLEMENTED DATA AND ERROR CHECKS
PROCEEDS TO NEXT CACHE LOCATION
AFTER ALL LOW CACHE HAS BIN WRITTEN TO COMPLEMENT
TEST THEN STARTS READING FROM LAST LOCATION
COMPLEMENTS DATA
WRITES COMPLEMENTED DATA TO CACHE
READS AND ERROR CHECKS DATA
PROCEEDS TO NEXT HIGHER LOCATION
UNTIL START OF LOW CACHE

TST122: INC TID ;UPDATE TEST ID
JSR R4,RELCTL ;RELOCATE TEST TO LOW CACHE
;WORD TST123
CLR @#ERROR ;RESET ERROR FLAG
CLR @#LOPERR ;RESET LOOP ON TEST FLAG
MOV #5,@#CCR ;ENABLE HIGH CACHE
MOV #HIGH1,R1 ;TAG HIGH BLOCK #1
1$: MOV (R1)+,R3 ;CAUSE READ TO HIGH BLOCK
CMP #HIGH1+2000,R1
BNE 1$
MOV #HIGHSP,R1 ;WRITE BACKGROUND TO HIGH CACHE
2$: CLR (R1)+ ;CAUSE READ TO TAG ,THEN WRITE BACKGROUND
CMP R1,#HIGHSP+2000
BNE 2$
MOV #HIGHSP,R1 ;ADDRESS OF TEST BLOCK
MOV #LOW1,R2 ;ADDRESS OF ERROR BLOCK
3$: CLR @#CMPE ;RESET ERROR REGISTER
MOV (R1)+,R3 ;READ BACKGROUND SAVE ERROR INFO
BIS #215,@#CCR ;SET ABORT FOR ERROR READ
MOV @#CMPE,(R2)+ ;SAVE ERROR INFO.
MOV #5,@#CCR ;DISABLE ABORT
CMP R1,#HIGHSP+2000
BNE 3$
MOV #HIGHSP,R1 ;WRITE BACKGROUND
MOV #LOW1,R2
33$: TST (R2) ;ANY ERROR FOR LOCATION
BNE 4$ ;YES
9$: ADD #2,R2 ;POINT TO NEXT
ADD #2,R1
CMP R1,#LOW1+2000
BNE 33$
BR 10$
4$: BIT #BIT05,(R2) ;ANY TAG ERROR
    
```

10018	030750	001411			BEQ 5\$	
10019	030752	016103	004000		MOV 4000(R1),R3	:UNTAG LOCATION
10020	030756	052737	002000	177746	BIS #BIT10,@#CCR	:ENABLE WRITE WRONG TAG
10021	030764	011103			MOV (R1),R3	:TAG LOCATION WRONG
10022	030766	042737	002000	177746	BIC #BIT10,@#CCR	:DISABLE WRITE WRONG TAG
10023	030774	032712	000100		BIT #BIT06,(R2)	:ANY LOW BYTE FAILURE
10024	031000	001410			BEQ 6\$:NO
10025	031002	052737	000100	177746	BIS #BIT06,@#CCR	:ENABLE WRITE WRONG DATA
10026	031010	112711	000000		MOVB #0,(R1)	:WRITE WRONG DATA LOW BYTE
10027	031014	042737	000100	177746	BIC #BIT06,@#CCR	:DISABLE WRITE WRONG DATA
10028	031022	032712	000200		BIT #BIT07,(R2)	:ANY HIGH BYTE FAILURE
10029	031026	001406			BEQ 7\$	
10030	031030	052737	000100	177746	BIS #BIT06,@#CCR	:WRITE WRONG DATA
10031	031036	112761	000000	000001	MOVB #0,1(R1)	:WRITE WRONG DATA HIGH BYTE
10032	031044	012737	001015	177746	MOV #OFF,@#CCR	:DISABLE CACHE
10033	031052	010703			MOV PC,R3	:CORRECT WRONG PARITY LOW
10034	031054	014313			MOV -(R3),(R3)	
10035	031056	020327	044146		CMP R3,#4\$-TST122-12+LOWSP	
10036	031062	001374			BNE 8\$	
10037	031064	012737	000005	177746	MOV #5,@#CCR	:ENABLE HIGH CACHE
10038	031072	000714			BR 9\$	
10039	031074	012701	046000		MOV #HIGHSP,R1	:START TEST
10040	031100	012702	050000		MOV #LOW1,R2	
10041	031104	011103			MOV (R1),R3	:READ DATA
10042	031106	005103			COM R3	:COMPLEMENT DATA
10043	031110	010311			MOV R3,(R1)	:WRITE DATA
10044	031112	005037	177744		CLR @#CMPE	:RESET ERROR REGISTER
10045	031116	011112			MOV (R1),(R2)	:READ COMP. DATA , SAVE
10046	031120	005737	177744		TST @#CMPE	:ANY ERRORS
10047	031124	001010			BNE 12\$:YES
10048	031126	062701	000002		ADD #2,R1	:ADJ. POINTER
10049	031132	062702	000002		ADD #2,R2	:ADJ. POINTER
10050	031136	020127	050000		CMP R1,#HIGHSP+2000	
10051	031142	001360			BNE 11\$	
10052	031144	000453			BR 21\$	
10053	031146	052737	000215	177746	BIS #215,@#CCR	:SET ABORT FOR ERROR READ
10054	031154	013705	177744		MOV @#CMPE,R5	:SAVE ERROR
10055	031160	012737	000005	177746	MOV #5,@#CCR	:DISABLE ABORT
10056	031166	032705	000100		BIT #BIT06,R5	:ANY LOW BYTE ERROR
10057	031172	001407			BEQ 13\$:NO
10058	031174	052737	000100	177746	BIS #BIT06,@#CCR	:ENABLE WRITE WRONG DATA
10059	031202	110311			MOVB R3,(R1)	:WRITE WRONG LOW BYTE
10060	031204	042737	000100	177746	BIC #BIT06,@#CCR	:DISABLE WRITE WRONG
10061	031212	032705	000200		BIT #BIT07,R5	:ANY HIGH BYTE ERROR
10062	031216	001411			BEQ 14\$:NO
10063	031220	000303			SWAB R3	
10064	031222	052737	000100	177746	BIS #BIT06,@#CCR	:ENABLE WRITE WRONG DATA
10065	031230	110361	000001		MOVB R3,1(R1)	:WRITE WRONG DATA HIGH BYTE
10066	031234	042737	000100	177746	BIC #BIT06,@#CCR	:DISABLE WRITE WRONG
10067	031242	012737	001015	177746	MOV #OFF,@#CCR	:DISABLE CACHE
10068	031250	010705			MOV PC,R5	:CORRECT WRONG PARITY LOW CACHE
10069	031252	014515			MOV -(R5),(R5)	
10070	031254	020527	044350		CMP R5,#12\$-TST122-12+LOWSP	
10071	031260	001374			BNE 15\$	
10072	031262	012737	000005	177746	MOV #5,@#CCR	:ENABLE HIGH CACHE
10073	031270	011112			MOV (R1),(R2)	:READ DATA

CACHE DIAG.
CFKKAB.P11

MACY11 30A(1052)
25-JUN-79 13:31

31-OCT-79 15:29 PAGE 73-12
DATA BIT MARCH PATTERN TEST

SEQ 0107

10074	031272	000715		
10075	031274	012737	001015	177746
10076	031302	012701	050000	

21\$: BR 16\$
 MOV #OFF,@#CCR
 MOV #LOW1,R1

:DISABLE CACHE
 :READ AND REPORT ERROR

10078	031306	012737	177777	004550	17\$:	MOV #177777, @#GOOD	;GOOD DATA
10079	031314	010137	004554			MOV R1, @#ADD	;MEMORY ADDRESS
10080	031320	162737	052000	004554		SUB #HIGH1, @#ADD	;CACHE ADDRESS
10081	031326	012137	004552			MOV (R1)+, @#BAD	;READ DATA
10082	031332	013737	004552	004542		MOV @#BAD, @#ERROR	;ERROR CHECK
10083	031340	005237	004542			INC @#ERROR	;NO ERROR IF ZERO
10084	031344	001004				BNE 18\$	
10085	031346	020127	052000		19\$:	CMP R1, #LOW1+2000	
10086	031352	001355				BNE 17\$	
10087	031354	000413				BR 20\$	
10088	031356	005237	004556		18\$:	INC @#GOODBD	;DATA PRINT MODE
10089	031362	004037	005352			JSR R0, @#SETEN	;REPORT ERROR
10090	031366	000004				.WORD *D4	
10091	031370	041460				.WORD SEN143	;DATA BIT MARCH PATTERN TEST
10092	031372	041474				.WORD SEN144	;CACHE ADDRESS , DATA EXPECTED , DATA READ
10093	031374	053737	004542	005064		BIS @#ERROR, @#LOPERR	;SET HAD ERROR FLAG
10094	031402	000761				BR 19\$	
10095	031404	012701	050000		20\$:	MOV #HIGHSP+2000, R1	
10096	031410	014103			22\$:	MOV -(R1), R3	;READ DATA
10097	031412	005103				COM R3	;COMPLEMENT DATA
10098	031414	010311				MOV R3, (R1)	;WRITE COMPLEMENTED DATA
10099	031416	020127	046000			CMP R1, #HIGHSP	
10100	031422	001372				BNE 22\$	
10101	031424	012701	050000			MOV #HIGHSP+2000, R1	
10102	031430	012702	052000			MOV #LOW1+2000, R2	
10103	031434	014142			24\$:	MOV -(R1), -(R2) ;	READ AND SAVE DATA
10104	031436	005737	177744			TST @#CMPE	
10105	031442	001004				BNE 23\$	
10106	031444	020127	046000		27\$:	CMP R1, #HIGHSP	
10107	031450	001371				BNE 24\$	
10108	031452	000450				BR 28\$	
10109	031454	052737	000215	177746	23\$:	BIS #215, @#CCR	;SET ABORT FOR ERROR READ
10110	031462	013705	177744			MOV @#CMPE, R5	;READ ERROR REGISTER
10111	031466	012737	000005	177746		MOV #5, @#CCR	;DISABLE ABORT
10112	031474	032705	000100			BIT #BIT06, R5	;DID LOW BYTE FAIL
10113	031500	001407				BEQ 25\$;NO
10114	031502	052737	000100	177746		BIS #BIT06, @#CCR	;SET WRITE WRONG DATA
10115	031510	111211				MOVB (R2), (R1)	;WRITE WRONG LOW BYTE
10116	031512	042737	000100	177746		BIC #BIT06, @#CCR	;DISABLE WRITE WRONG
10117	031520	032705	000200		25\$:	BIT #BIT07, R5	;DID HIGH BYTE FAIL
10118	031524	001406				BEQ 46\$;NO
10119	031526	052737	000100	177746		BIS #BIT06, @#CCR	;ENABLE WRITE WRONG DATA
10120	031534	116261	000001	000001		MOVB 1(R2), 1(R1)	;WRITE WRONG HIGH BYTE
10121	031542	012737	001015	177746	46\$:	MOV #OFF, @#CCR	;DISABLE CACHE
10122	031550	010703				MOV PC, R3	;CORRECT WRONG PARITY
10123	031552	014313			26\$:	MOV -(R3), (R3)	
10124	031554	020327	044656			CMP R3, #23\$-TST122-12+LOWSP	
10125	031560	001374				BNE 26\$	
10126	031562	012737	000005	177746		MOV #5, @#CCR	;ENABLE CACHE
10127	031570	011112				MOV (R1), (R2)	;READ DATA
10128	031572	000724				BR 27\$	
10129	031574	012737	001015	177746	28\$:	MOV #OFF, @#CCR	;DISABLE CACHE
10130	031602	012701	052000			MOV #LOW1+2000, R1	;CHECK FOR ERRORS
10131	031606	012737	000000	004550		MOV #0, @#GOOD	;GOOD DATA
10132	031614	014137	004542		29\$:	MOV -(R1), @#ERROR	;READ DATA
10133	031620	010137	004554			MOV R1, @#ADD	;MEMORY ADDRESS

10134	031624	013737	004542	004552	MOV @#ERROR,@#BAD	;BAD DATA
10135	031632	005237	004556		INC @#GOODBD	;SET DATA PRINT MODE
10136	031636	162737	052000	004554	SUB #HIGH1,@#ADD	;CACHE ADDRESS
10137	031644	004037	005352		JSR R0,@#SETEN	;REPORT ERROR IF ANY
10138	031650	000004			.WORD ^D4	
10139	031652	041460			.WORD SEN143	;DATA BIT MARCH PATTERN TEST
10140	031654	041474			.WORD SEN144	;CACHE ADDRESS , DATA EXPECTED , DATA READ
10141	031656	053737	004542	005064	BIS @#ERROR,@#LOPERR	;DEL ERROR LOCATION
10142	031664	020127	050000		CMP R1,#LOW1	
10143	031670	001351			BNE 29\$	
10144	031672	013737	005064	004542	MOV @#LOPERR,@#ERROR	;TAKE SELECTED ACTION ON ERROR
10145	031700	004010			JSR R0,(R0)	;LOOP ON ERROR
10146	031702	044000			.WORD LOWSP	;LOOP ON TEST
10147	031704	044000			.WORD LOWSP	

```

10150          .SBTTL DATA PARITY MARCH PATTERN TEST
10151          : PATTERN RUNNING IN LOW CACHE PARITY
10152
10153
10154 031706 005267 146734      TST123:      INC TID          ;UPDATE TEST ID
10155 031712 004467 154224      JSR R4,RELCTH   ;RELOCATE TEST TO HIGH CACHE
10156 031716 032324      .WORD TST124
10157 031720 005037 005064      CLR @#LOPERR
10158 031724 005037 004542      CLR @#ERROR    ;RESET ERROR FLAG
10159 031730 012701 044000      MOV #LOWSP,R1  ;FIRST ADDRESS OF TEST BLOCK
10160 031734 012737 000011 177746  MOV #11,@#CCR  ;ENABLE LOW CACHE
10161 031742 012701 044000      MOV #LOWSP,R1  ;TEST BLOCK
10162 031746 005021      1$:          CLR (R1)+      ;WRITE ZERO TO PARITY
10163 031750 020127 046000      CMP R1,#LOWSP+2000
10164 031754 001374      BNE 1$
10165 031756 012701 044000      MOV #LOWSP,R1  ;FIRST ADDRESS OF TEST BLOCK
10166 031762 012703 052000      MOV #HIGH1,R3  ;ERROR BLOCK POINTER
10167 031766 005037 177744      CLR @#CMPE     ;RESET ERROR REGISTER
10168 031772 011102      MOV (R1),R2    ;READ PARITY
10169 031774 005737 177744      TST @#CMPE    ;ANY ERROR
10170 032000 001052      BNE 3$        ;YES
10171 032002 005013      CLR (R3)
10172 032004 012711 004001      MOV #4001,(R1) ;WRITE COMP. PARITY
10173 032010 011102      MOV (R1),R2    ;READ COMP
10174 032012 005737 177744      TST @#CMPE    ;ANY ERROR
10175 032016 001043      BNE 3$        ;YES
10176 032020 062701 000002      4$:          ADD #2,R1      ;ADJ. POINTER
10177 032024 062703 000002      ADD #2,R3
10178 032030 012737 001015 177746  MOV #OFF,@#CCR ;INVALIDATE HIGH CACHE
10179 032036 010702      MOV PC,R2
10180 032040 014212      30$:         MOV -(R2),(R2)
10181 032042 020227 046046      CMP R2,#2$-TST123-12+HIGHSP
10182 032046 001374      BNE 30$
10183 032050 012737 000011 177746  MOV #11,@#CCR
10184 032056 020127 046000      CMP R1,#LOWSP+2000
10185 032062 001341      BNE 2$
10186 032064 012701 044000      MOV #LOWSP,R1  ;DATA ADDRESS
10187 032070 012703 052000      MOV #HIGH1,R3  ;ERROR REGISTER SAVE BLOCK
10188 032074 012737 001015 177746  MOV #OFF,@#CCR ;DISABLE CACHE
10189 032102 005713      21$:         TST (R3)      ;ANY ERROR FOR LOCATION
10190 032104 001021      BNE 22$       ;YES
10191 032106 062701 000002      23$:         ADD #2,R1      ;UPDATE POINTERS
10192 032112 062703 000002      ADD #2,R3
10193 032116 020127 046000      CMP R1,#LOWSP+2000
10194 032122 001367      BNE 21$
10195 032124 000471      BR 10$
10196 032126 052737 000215 177746  3$:          BIS #215,@#CCR ;ENABLE ABORT FOR ERROR READ
10197 032134 013713 177744      MOV @#CMPE,(R3) ;SAVE ERROR
10198 032140 012737 000011 177746  MOV #11,@#CCR  ;DISABLE ABORT
10199 032146 000724      BR 4$
10200 032150 005037 004550      22$:         CLR @#GOOD
10201 032154 005037 004552      CLR @#BAD
10202 032160 005711      TST (R1)
10203 032162 001404      BEQ 5$
10204 032164 012737 000001 004550  MOV #1,@#GOOD ;SHOULD PARITY BIT BE SET
10205 032172 000403      BR 6$        ;NO
                ;PARITY BIT SHOULD HAVE BIN SET

```

10206	032174	012737	000001	004552	5\$:	MOV #1,@#BAD	:PARITY BIT SHOULDN'T BE SET
10207	032202	010137	004554		6\$:	MOV R1,@#ADD	:MEMORY ADDRESS
10208	032206	162737	040000	004554		SUB #40000,@#ADD	:CACHE ADDRESS
10209	032214	005237	004542			INC @#ERROR	:SET ERROR FLAG
10210	032220	005237	004556			INC @#GOODBD	:DATA PRINT MODE
10211	032224	032713	000100			BIT #BIT06,(R3)	:WAS IT LOW BYTE ERROR
10212	032230	001015				BNE 7\$	
10213	032232	032713	000200			BIT #BIT07,(R3)	:WAS IT HIGH BYTE
10214	032236	001670				BEQ 4\$:NO
10215	032240	004037	005352			JSR R0,@#SETEN ;REPORT	ERROR
10216	032244	000006				.WORD *D6	
10217	032246	041516				.WORD SEN145	:DATA PARITY MARCH PATTERN TEST
10218	032250	041532				.WORD SEN146	:HIGH BYTE PARITY ERROR
10219	032252	041544				.WORD SEN147	:CACHE ADDRESS , PARITY WRITTEN , PARITY READ
10220	032254	053737	004542	005064		BIS @#ERROR,@#LOPERR	:HAD ERROR FLAG
10221	032262	000711				BR 23\$	
10222	032264	004037	005352		7\$:	JSR R0,@#SETEN	:REPORT ERROR
10223	032270	000006				.WORD *D6	
10224	032272	041516				.WORD SEN145	:DATA PARITY MARCH PATTERN TEST
10225	032274	041566				.WORD SEN148	:LOW BYTE PARITY ERROR
10226	032276	041544				.WORD SEN147	:CACHE ADDRESS , PARITY WRITTEN , PARITY READ
10227	032300	053737	004542	005064		BIS @#ERROR,@#LOPERR	:HAD ERROR FLAG
10228	032306	000677				BR 23\$	
10229	032310	013737	005064	004542	10\$:	MOV @#LOPERR,@#ERROR	:CHECK HAD ERROR FLAG
10230	032316	004010				JSR R0,(R0)	:TAKE SELECTED ACTION ON ERROR
10231	032320	046000				.WORD HIGHSP	:LOOP ON ERROR
10232	032322	046000				.WORD HIGHSP	:LOOP ON TEST
10233						:	PATTERN RUNNING IN HIGH CACHE
10234							
10235							
10236	032324	005267	146316		TST124:	INC TID	:UPDATE TEST ID
10237	032330	004467	153560			JSR R4,RELCTL	:RELOCATE TEST TO LOW CACHE
10238	032334	032742				.WORD TST125	
10239	032336	005037	005064			CLR @#LOPERR	
10240	032342	005037	004542			CLR @#ERROR	:RESET ERROR FLAG
10241	032346	012701	046000			MOV #HIGHSP,R1	:FIRST ADDRESS OF TEST BLOCK
10242	032352	012737	000005	177746		MOV #5,@#CCR	:ENABLE HIGH CACHE
10243	032360	012701	046000			MOV #HIGHSP,R1	:TEST BLOCK
10244	032364	005021			1\$:	CLR (R1)+	:WRITE ZERO TO PARITY
10245	032366	020127	050000			CMP R1,#HIGHSP+2000	
10246	032372	001374				BNE 1\$	
10247	032374	012701	046000			MOV #HIGHSP,R1	:FIRST ADDRESS OF TEST BLOCK
10248	032400	012703	050000			MOV #LOW1,R3	:ERROR BLOCK POINTER
10249	032404	005037	177744		2\$:	CLR @#CMPE	:RESET ERROR REGISTER
10250	032410	011102				MOV (R1),R2	:READ PARITY
10251	032412	005737	177744			TST @#CMPE	:ANY ERROR
10252	032416	001052				BNE 3\$:YES
10253	032420	005013				CLR (R3)	
10254	032422	012711	004001			MOV #4001,(R1)	:WRITE COMP. PARITY
10255	032426	011102				MOV (R1),R2	:READ COMP.
10256	032430	005737	177744			TST @#CMPE	:ANY ERROR
10257	032434	001043				BNE 3\$:YES
10258	032436	062701	000002		4\$:	ADD #2,R1	:ADJ. POINTER
10259	032442	062703	000002			ADD #2,R3	
10260	032446	012737	001015	177746		MOV #OFF,@#CCR	:INVALIDATE LOW CACHE
10261	032454	010702				MOV PC,R2	

10262	032456	014212			30\$:	MOV -(R2), (R2)	
10263	032460	020227	044046			CMP R2, #2\$-TST124-12+LOWSP	
10264	032464	001374				BNE 30\$	
10265	032466	012737	000005	177746		MOV #5, @#CCR	
10266	032474	020127	050000			CMP R1, #HIGHSP+2000	
10267	032500	001341				BNE 2\$	
10268	032502	012701	046000			MOV #HIGHSP, R1	:DATA BLOCK
10269	032506	012703	050000			MOV #LOW1, R3	:ERROR BLOCK
10270	032512	012737	001015	177746		MOV #OFF, @#CCR	
10271	032520	005713			21\$:	TST (R3)	
10272	032522	001021				BNE 22\$	
10273	032524	062701	000002		23\$:	ADD #2, R1	:UPDATE POINTERS
10274	032530	062703	000002			ADD #2, R3	
10275	032534	020127	050000			CMP R1, #HIGHSP+2000	
10276	032540	001367				BNE 21\$	
10277	032542	000471				BR 10\$	
10278	032544	052737	000215	177746	3\$:	BIS #215, @#CCR	:ENABLE ABORT FOR ERROR READ
10279	032552	013713	177744			MOV @#CMPE, (R3)	:SAVE ERROR
10280	032556	012737	000005	177746		MOV #5, @#CCR	:DISABLE ABORT
10281	032564	000724				BR 4\$	
10282	032566	005037	004550		22\$:	CLR @#GOOD	:
10283	032572	005037	004552			CLR @#BAD	:
10284	032576	005711				TST (R1)	:SHOULD PARITY BIT BE SET
10285	032600	001404				BEQ 5\$:NO
10286	032602	012737	000001	004550		MOV #1, @#GOOD	:PARITY BIT SHOULD HAVE BIN SET
10287	032610	000403				BR 6\$	
10288	032612	012737	000001	004552	5\$:	MOV #1, @#BAD	:PARITY BIT SHOULDN'T BE SET
10289	032620	010137	004554		6\$:	MOV R1, @#ADD	:MEMORY ADDRESS
10290	032624	162737	040000	004554		SUB #40000, @#ADD	:CACHE ADDRESS
10291	032632	005237	004542			INC @#ERROR	:SET ERROR FLAG
10292	032636	005237	004556			INC @#GOODBD	:DATA PRINT MODE
10293	032642	032713	000100			BIT #BIT06, (R3)	:WAS IT LOW BYTE ERROR
10294	032646	001015				BNE 7\$	
10295	032650	032713	000200			BIT #BIT07, (R3)	:WAS IT HIGH BYTE
10296	032654	001670				BEQ 4\$:NO
10297	032656	004037	005352			JSR R0, @#SETEN	:REPORT ERROR
10298	032662	000006				.WORD ^D6	
10299	032664	041516				.WORD SEN145	:DATA PARITY MARCH PATTERN TEST
10300	032666	041532				.WORD SEN146	:HIGH BYTE PARITY ERROR
10301	032670	041544				.WORD SEN147	:CACHE ADDRESS, PARITY WRITTEN, PARITY READ
10302	032672	053737	004542	005064		BIS @#ERROR, @#LOPERR	:HAD ERROR FLAG
10303	032700	000711				BR 23\$	
10304	032702	004037	005352		7\$:	JSR R0, @#SETEN	:REPORT ERROR
10305	032706	000006				.WORD ^D6	
10306	032710	041516				.WORD SEN145	:DATA PARITY MARCH PATTERN TEST
10307	032712	041566				.WORD SEN148	:LOW BYTE PARITY ERROR
10308	032714	041544				.WORD SEN147	:CACHE ADDRESS, PARITY WRITTEN, PARITY READ
10309	032716	053737	004542	005064		BIS @#ERROR, @#LOPERR	:HAD ERROR FLAG
10310	032724	000677				BR 23\$	
10311	032726	013737	005064	004542	10\$:	MOV @#LOPERR, @#ERROR	:CHECK HAD ERROR FLAG
10312	032734	004010				JSR R0, (R0)	:TAKE SELECTED ACTION ON ERROR
10313	032736	044000				.WORD LOWSP	:LOOP ON ERROR
10314	032740	044000				.WORD LOWSP	:LOOP ON TEST

```

10316          .SBTTL VALID BIT MARCH PATTERN TEST
10317          : TESTING LOW CACHE USING BOTH SETS OF VALID BITS
10318
10319
10320 032742 005267 145700      TST125:      INC TID          ;UPDATE TEST ID
10321 032746 004467 153170      JSR R4,RELCTH   ;RELOCATE TEST TO HIGH CACHE
10322 032752 033334
10323 032754 005005
10324 032756 005037 005064      20$:      CLR @#LOPERR    ;RESET HAD ERROR FLAG
10325 032762 005037 004542      CLR @#ERROR     ;RESET ERROR FLAG
10326 032766 012701 044000      MOV #LOWSP,R1   ;WRITE VALID BIT TO ZERO
10327 032772 012721 000000      1$:      MOV #0,(R1)+
10328 032776 020127 046000      CMP R1,#LOWSP+2000
10329 033002 001373      BNE 1$
10330 033004 012703 052000      MOV #HIGH1,R3
10331 033010 012701 044000      MOV #LOWSP,R1
10332 033014 012737 000011 177746      2$:      MOV #11,@#CCR   ;ENABLE LOW CACHE
10333 033022 011102      MOV (R1),R2     ;READ VALID WRITE COMP.
10334 033024 032737 000010 177752      BIT #BIT03,@#CHR ;LOOK FOR HIT
10335 033032 001012      BNE 7$          ;READ DID CAUSE HIT, IT SHOULDN'T HAVE
10336 033034 012102      MOV (R1)+,R2   ;READ COMP
10337 033036 032737 000010 177752      BIT #BIT03,@#CHR ;IS VALID SET
10338 033044 001414      BEQ 3$          ;NO
10339 033046 005023
10340 033050 020127 046000      5$:      CLR (R3)+
10341 033054 001357      CMP R1,#LOWSP+2000
10342 033056 000416      BNE 2$
10343 033060 113763 177747 000001      7$:      BR 11$
10344 033066 112713 000000      MOV @#CCR+1,1(R3) ;SAVE VALID SET IN USE BIT
10345 033072 005723      MOV @#0,(R3)    ;SAVE DATA WRITTEN
10346 033074 000765      TST (R3)+       ;ADJ POINTER
10347 033076 113763 177747 000001      3$:      BR 5$          ;CONTINUE
10348 033104 112713 000001      MOV @#CCR+1,1(R3) ;SAVE VALID SET IN USE
10349 033110 005723      MOV @#1,(R3)    ;SAVE DATA WRITTEN
10350 033112 000756      TST (R3)+
10351 033114 012737 001015 177746      11$:     BR 5$          ;CONTINUE
10352 033122 012703 052000      MOV #OFF,@#CCR  ;DISABLE CACHE
10353 033126 005723      MOV #HIGH1,R3   ;ERROR DATA BLOCK
10354 033130 001004      TST (R3)+       ;ANY ERROR
10355 033132 020327 054000      16$:     BNE 13$        ;YES
10356 033136 001373      CMP R3,#HIGH1+2000
10357 033140 000455      BNE 12$
10358 033142 005037 004550      13$:     BR 10$
10359 033146 005037 004552      CLR @#GOOD      ;FIND WHAT WENT WRONG
10360 033152 105763 177776      CLR @#BAD
10361 033156 001004      TSTB -2(R3)     ;FIND DATA WRITTEN
10362 033160 012737 000001 004552      BNE 14$        ;DATA WRITTEN WAS ONE
10363 033166 000403      MOV #1,@#BAD    ;DATA READ WAS ONE
10364 033170 012737 000001 004550      BR 15$
10365 033176 010337 004554      14$:     MOV #1,@#GOOD   ;DATA READ WAS ZERO
10366 033202 162737 052002 004554      15$:     MOV R3,@#ADD    ;MEMORY ADDRESS
10367 033210 012737 000001 004542      SUB #HIGH1+2,@#ADD ;CACHE ADDRESS
10368 033216 005237 004556      MOV #1,@#ERROR  ;SET ERROR FLAG
10369 033222 053737 004542 005064      INC @#GOODBD    ;SET DATA PRINT MODE
10370 033230 032763 020000 177776      BIS @#ERROR,@#LOPERR ;SET DELAY ERROR FLAG
10371 033236 001407      BIT #BIT13,-2(R3) ;IS THIS VALID SET B
10371 033236 001407      BEQ 4$          ;NO

```

```

10372 033240 004037 005352      JSR R0,@#SETEN      ;REPORT ERROR
10373 033244 000006              .WORD ^D6           ;VALID BIT MARCH PATTERN TEST
10374 033246 041600              .WORD SEN149        ;VALID BIT SET B INUSE
10375 033250 041614              .WORD SEN150        ;CACHE ADDRESS , VALID BIT WRITTEN , VALID BIT R
10376 033252 041630              .WORD SEN151
10377 033254 000726              BR 16$
10378 033256 004037 005352      4$: JSR R0,@#SETEN      ;REPORT ERROR
10379 033262 000006              .WORD ^D6           ;VALID BIT MARCH PATTERN TEST
10380 033264 041600              .WORD SEN149        ;VALID BIT SET A INUSE
10381 033266 041656              .WORD SEN152        ;CACHE ADDRESS , VALID BIT WRITEN , VALID BIT RE
10382 033270 041672              .WORD SEN153
10383 033272 000717              BR 16$
10384 033274 012737 001015 177746 10$: MOV #OFF,@#CCR      ;DISABLE CACHE
10385 033302 013737 005064 004542  MOV @#LOPERR,@#ERROR ;
10386 033310 004010              JSR R0,(R0)         ;TAKE SELECTED ACTION ON ERROR
10387 033312 046000              .WORD HIGHSP        ;LOOP ON ERROR
10388 033314 046000              .WORD HIGHSP        ;LOOP ON TEST
10389 033316 005705              TST R5              ;IS THIS FIRST PASS
10390 033320 001005              BNE TST126         ;NO CONTINUE TO NEXT TEST
10391 033322 052737 000400 177746  BIS #BIT108,@#CCR   ;FLUSH CACHE SELECT OTHER VALID SET
10392 033330 005205
10393 033332 000611
10394
10395
10396
10397
10398
10399
10400
10401

```

: TESTING HIGH CACHE USING BOTH SETS OF VALID BITS

```

10402 033334 005267 145306      TST126: INC TID          ;UPDATE TEST ID
10403 033340 004467 152550      JSR R4,RELCTL       ;RELOCATE TEST TO LOW CACHE
10404 033344 033730              .WORD TST127
10405 033346 005005              CLR R5              ;VALID CHANGE FLAG
10406 033350 005037 005064      20$: CLR @#LOPERR     ;RESET HAD ERROR FLAG
10407 033354 005037 004542      CLR @#ERROR        ;RESET ERROR FLAG
10408 033360 012701 046000      MOV #HIGHSP,R1     ;WRITE VALID BIT TO ZERO
10409 033364 012721 000000      1$: MOV #0,(R1)+
10410 033370 020127 050000      CMP R1,#HIGHSP+2000
10411 033374 001373              BNE 1$
10412 033376 012703 050000      MOV #LOW1,R3
10413 033402 012701 046000      MOV #HIGHSP,R1
10414 033406 012737 000005 177746 2$: MOV #5,@#CCR      ;ENABLE HIGH CACHE
10415 033414 011102              MOV (R1),R2        ;READ VALID WRITE COMP.
10416 033416 032737 000010 177746  BIT #BIT03,@#CCR   ;SHOULD HAVE NO HIT
10417 033424 001012              BNE 7$
10418 033426 012102              MOV (R1)+,R2       ;READ COMP
10419 033430 032737 000010 177752  BIT #BIT03,@#CHR   ;IS VALID SET
10420 033436 001414              BEQ 3$              ;NO
10421 033440 005023              CLR (R3)+
10422 033442 020127 050000      5$: CMP R1,#HIGHSP+2000
10423 033446 001357              BNE 2$
10424 033450 000416              BR 11$
10425 033452 113763 177747 000001 7$: MOV @#CCR+1,1(R3) ;SAVE VALID SET INUSE BIT
10426 033460 112713 000000      MOV #0,(R3)        ;SAVE DATA WRITTEN

```

10428	033464	005723				TST (R3)+	
10429	033466	000765				BR 5\$	
10430	033470	113763	177747	000001	3\$:	MOVB @#CCR+1,1(R3)	;SAVE VALID SET INUSE BIT
10431	033476	112713	000001			MOVB #1,(R3)	;SAVE DATA WRITTEN
10432	033502	005723				TST (R3)+	
10433	033504	000756				BR 5\$	
10434	033506	012737	001015	177746	11\$:	MOV #OFF,@#CCR	;DISABLE CACHE
10435	033514	012703	050000			MOV #LOW1,R3	;ERROR DATA BLOCK
10436	033520	005723			12\$:	TST (R3)+	;ANY ERROR
10437	033522	001004				BNE 13\$;YES
10438	033524	020327	052000		16\$:	CMP R3,#LOW1+2000	
10439	033530	001373				BNE 12\$	
10440	033532	000456				BR 10\$	
10441	033534	005037	004550		13\$:	CLR @#GOOD	;FIND WHAT WENT WRONG
10442	033540	005037	004552			CLR @#BAD	
10443	033544	105763	177776			TSTB -2(R3)	;FIND DATA WRITTEN
10444	033550	001004				BNE 14\$;DATA WRITTEN WAS A ONE
10445	033552	012737	000001	004552		MOV #1,@#BAD	;DATA READ WAS A ONE
10446	033560	000403				BR 15\$	
10447	033562	012737	000001	004550	14\$:	MOV #1,@#GOOD	;DATA READ WAS ZERO
10448	033570	010337	004554		15\$:	MOV R3,@#ADD	;MEMORY ADDRESS
10449	033574	162737	046002	004554		SUB #HIGHSP+2,@#ADD	;CACHE ADDRESS
10450	033602	012737	000001	004542		MOV #1,@#ERROR	;SET ERROR FLAG
10451	033610	005237	004556			INC @#GOODBD	;SET DATA PRINT MODE
10452	033614	053737	004542	005064		BIS @#ERROR,@#LOPERR	;SET DELAY ERROR FLAG
10453	033622	032763	020000	177776		BIT #BIT13,-2(R3)	;IS THIS VALID SET B
10454	033630	001410				BEQ 4\$;NO
10455	033632	004037	005352			JSR R0,@#SETEN	;REPORT ERROR
10456	033636	000006				.WORD ^D6	
10457	033640	041600				.WORD SEN149	;VALID BIT MARCH PATTERN TEST
10458	033642	041614				.WORD SEN150	;VALID BIT SET B INUSE
10459	033644	041630				.WORD SEN151	;CACHE ADDRESS , VALID BIT WRITTEN , VALID BIT R
10460	033646	005205				INC R5	
10461	033650	000725				BR 16\$	
10462	033652	004037	005352		4\$:	JSR R0,@#SETEN	;REPORT ERROR
10463	033656	000006				.WORD ^D6	
10464	033660	041600				.WORD SEN149	;VALID BIT MARCH PATTERN TEST
10465	033662	041656				.WORD SEN152	;VALID BIT SET A INUSE
10466	033664	041672				.WORD SEN153	;CACHE ADDRESS , VALID BIT WRITTEN , VALID BIT RE
10467	033666	000716				BR 16\$	
10468	033670	012737	001015	177746	10\$:	MOV #OFF,@#CCR	;DISABLE CACHE
10469	033676	013737	005064	004542		MOV @#LOPERR,@#ERROR	
10470	033704	004010				JSR R0,(R0)	;TAKE SELECTED ACTION ON ERROR
10471	033706	044000				.WORD LOWSP	;LOOP ON ERROR
10472	033710	044000				.WORD LOWSP	;LOOP ON TEST
10473	033712	005705				TST R5	;IS THIS FIRST PASS
10474	033714	001005				BNE TST127	;NO CONTINUE TO NEXT TEST
10475	033716	052737	000400	177746		BIS #BIT08,@#CCR	;FLUSH CACHE SELECT OTHER VALID SET
10476	033724	005205				INC R5	
10477	033726	000610				BR 20\$	

.SBTTL TAG MARCH PATTERN TESTS
:OPERATING IN HIGH CACHE TESTING LOW CACHE

```

10479
10480
10481
10482
10483 033730 005267 144712
10484 033734 004467 152202
10485 033740 034330
10486 033742 005037 005064
10487 033746 005037 004542
10488 033752 012737 000011 177746
10489 033760 005001
10490 033762 012102
10491 033764 020127 002000
10492 033770 001374
10493 033772 005001
10494 033774 011102
10495 033776 032737 000010 177752
10496 034004 001422
10497 034006 012737 000001 177750
10498 034014 011102
10499 034016 011102
10500 034020 032737 000010 177752
10501 034026 001411
10502 034030 012737 000000 177750
10503 034036 062701 000002
10504 034042 020127 002000
10505 034046 001352
10506 034050 000435
10507 034052 012737 000000 177750
10508 034060 012737 001015 177746
10509 034066 005037 004552
10510 034072 005037 004550
10511 034076 005237 004556
10512 034102 005237 004542
10513 034106 010137 004554
10514 034112 004037 005352
10515 034116 000006
10516 034120 041720
10517 034122 041732
10518 034124 041744
10519 034126 053737 004542 005064
10520 034134 012737 000011 177746
10521 034142 000735
10522 034144 012701 002000
10523 034150 012737 000001 177750
10524 034156 014102
10525 034160 032737 000010 177752
10526 034166 001414
10527 034170 012737 000000 177750
10528 034176 011102
10529 034200 011102
10530 034202 032737 000010 177752
10531 034210 001403
10532 034212 005701
10533 034214 001355
10534 034216 000433

```

```

TST127:      INC TID      ;UPDATE TEST ID
              JSR R4,RELCTH ;RELOCATE TEST TO HIGH CACHE
              .WORD TST130
              CLR @WLOPERR   ;RESET DELAY ERROR
              CLR @WERROR    ;RESET ERROR FLAG
              MOV #11,@WCCR   ;ENABLE LOW CACHE
              CLR R1         ;TAGGING START ADDRESS
1$:          MOV (R1)+,R2     ;WRITE ZERO TO TAG
              CMP R1,#2000   ;FOR ALL LOW LOCATIONS
              BNE 1$
              CLR R1         ;READ TAG START ADDRESS
2$:          MOV (R1),R2     ;READ BACKGROUND
              BIT #BIT03,@WCHR ;SHOULD HAVE HIT
              BEQ 3$        ;NO HIT
              MOV #1,@WCMR   ;ENABLE MAINT. MODE
              MOV (R1),R2   ;WRITE COMP TAG
              MOV (R1),R2   ;READ COMP TAG
              BIT #BIT03,@WCHR ;SHOULD HAVE HIT
              BEQ 3$
              MOV #0,@WCMR   ;DISABLE MAINT MODE
4$:          ADD #2,R1      ;NEXT ADDRESS
              CMP R1,#2000   ;LOOP COMPLETE YET
              BNE 2$        ;NO
              BR 6$
3$:          MOV #0,@WCMR   ;DISABLE MAINT MODE
              MOV #OFF,@WCCR ;DISABLE CACHE
              CLR @WBAD      ;DATA UNKNOWN
              CLR @WGOOD
              INC @WGOODBD
              INC @WERROR    ;SET ERROR FLAG
              MOV R1,@WADD   ;CACHE ADDRESS
              JSR R0,@WSETEN ;REPORT ERROR
              .WORD *D6
              .WORD SEN154
              .WORD SEN155
              .WORD SEN156
              BIS @WERROR,@WLOPERR ;SET DELAYED ERROR FLAG
              MOV #11,@WCCR   ;ENABLE CACHE
              BR 4$
6$:          MOV #2000,R1    ;END ADDRESS
7$:          MOV #1,@WCMR   ;ENABLE MAINT. MODE
              MOV -(R1),R2  ;READ BACKGROUND
              BIT #BIT03,@WCHR ;ANY HIT
              BEQ 8$        ;NO
              MOV #0,@WCMR   ;DISABLE MAINT. MODE
              MOV (R1),R2   ;WRITE COMP
              MOV (R1),R2   ;READ COMP
              BIT #BIT03,@WCHR ;SHOULD HAVE HIT
              BEQ 8$
9$:          TST R1         ;NO HIT
              BNE 7$        ;LOOP COMPLETE YET
              BR 10$       ;NO

```

10535	034220	012737	001015	177746	8\$:	MOV #OFF,@#CCR	;DISABLE CACHE
10536	034226	005037	004550			CLR @#GOOD	;DATA UNKNOWN
10537	034232	005037	004552			CLR @#BAD	
10538	034236	010137	004554			MOV R1,@#ADD	;CACHE ADDRESS
10539	034242	012737	000001	004542		MOV #1,@#ERROR	;SET ERROR FLAG
10540	034250	005237	004556			INC @#GOODBD	;SET BIT PRINT MODE
10541	034254	004037	005352			JSR RO,@#SETEN	;REPORT ERROR
10542	034260	000006				.WORD ^D6	
10543	034262	041720				.WORD SEN154	;TAG MARCH PATTERN TESTS
10544	034264	041732				.WORD SEN155	;LOW CACHE TAG FAILURE
10545	034266	041744				.WORD SEN156	;CACHE ADDRESS , DATA UNKNOWN
10546	034270	012737	000011	177746		MOV #11,@#CCR	;ENABLE CACHE
10547	034276	053737	004542	005064		BIS @#ERROR,@#LOPERR	;SET DELAYED ERROR FLAG
10548	034304	000742				BR 9\$	
10549	034306	012737	001015	177746	10\$:	MOV #OFF,@#CCR	;DISABLE CACHE
10550	034314	013737	005064	004542		MOV @#LOPERR,@#ERROR	
10551	034322	004010				JSR RO,(RO)	;TAKE SELECTED ACTION ON ERROR
10552	034324	046000				.WORD HIGHSP	
10553	034326	046000				.WORD HIGHSP	
10554							
10555							
10556							
10557							
10558							
10559							

```

10561
10562
10563
10564
10565 034330 005267 144312
10566 034334 004467 151554
10567 034340 034736
10568 034342 005037 005064
10569 034346 005037 004542
10570 034352 012737 000005 177746
10571 034360 012701 002000
10572 034364 012102
10573 034366 020127 004000
10574 034372 001374
10575 034374 012701 002000
10576 034400 011102
10577 034402 032737 000010 177752
10578 034410 001422
10579 034412 012737 000001 177750
10580 034420 011102
10581 034422 011102
10582 034424 032737 000010 177752
10583 034432 001411
10584 034434 012737 000000 177750
10585 034442 062701 000002
10586 034446 020127 004000
10587 034452 001352
10588 034454 000435
10589 034456 012737 000000 177750
10590 034464 012737 001015 177746
10591 034472 005037 004552
10592 034476 005037 004550
10593 034502 005237 004556
10594 034506 005237 004542
10595 034512 010137 004554
10596 034516 004037 005352
10597 034522 000006
10598 034524 041720
10599 034526 041760
10600 034530 041744
10601 034532 053737 004542 005064
10602 034540 012737 000005 177746
10603 034546 000735
10604 034550 012701 004000
10605 034554 012737 000001 177750
10606 034562 014102
10607 034564 032737 000010 177752
10608 034572 001415
10609 034574 012737 000000 177750
10610 034602 011102
10611 034604 011102
10612 034606 032737 000010 177752
10613 034614 001404
10614 034616 022701 002000
10615 034622 001354
10616 034624 000433

```

```

.SBTTL TAG MARCH PATTERN TESTS
:OPERATING IN LOW CACHE TESTING HIGH CACHE

```

```

TST130:      INC TID      ;UPDATE TEST ID
              JSR R4,RELCTL ;RELOCATE TEST TO LOW CACHE
              .WORD TST131
              CLR @WLOPERR  ;RESET DELAY ERROR
              CLR @WERROR   ;RESET ERROR FLAG
              MOV #5,@WCCR   ;ENABLE HIGH CACHE
              MOV #2000,R1  ;TAGGING START ADDRESS
1$:          MOV (R1)+,R2   ;WRITE ZERO TO TAG
              CMP R1,#4000 ;FOR ALL HIGH LOCATIONS
              BNE 1$
              MOV #2000,R1  ;READ TAG , START ADDRESS
2$:          MOV (R1),R2   ;READ BACKGROUND
              BIT #BIT03,@WCHR ;SHOULD HAVE HIT
              BEQ 3$      ;NO HIT
              MOV #1,@WCMR  ;ENABLE MAINT. MODE
              MOV (R1),R2   ;WRITE COMP TAG
              MOV (R1),R2   ;READ COMP TAG
              BIT #BIT03,@WCHR ;SHOULD HAVE HIT
              BEQ 3$
              MOV #0,@WCMR  ;DISABLE MAINT MODE
4$:          ADD #2,R1     ;NEXT ADDRESS
              CMP R1,#4000 ;LOOP COMPLETE YET
              BNE 2$      ;NO
              BR 6$
3$:          MOV #0,@WCMR  ;DISABLE MAINT MODE
              MOV #OFF,@WCCR ;DISABLE CACHE
              CLR @WBAD    ;DATA UNKNOWN
              CLR @WGOOD
              INC @WGOODBD
              INC @WERROR  ;SET ERROR FLAG
              MOV R1,@WADD  ;CACHE ADDRESS
              JSR R0,@WSETEN ;REPORT ERROR
              .WORD *D6
              .WORD SEN154 ;TAG MARCH PATTERN TESTS
              .WORD SEN157 ;HIGH CACHE TAG FAILURE
              .WORD SEN156 ;CACHE ADDRESS , DATA UNKNOWN
              BIS @WERROR,@WLOPERR ;SET DELAYED ERROR FLAG
              MOV #5,@WCCR  ;ENABLE CACHE
              BR 4$
6$:          MOV #4000,R1  ;END ADDRESS
7$:          MOV #1,@WCMR  ;ENABLE MAINT. MODE
              MOV -(R1),R2 ;READ BACKGROUND
              BIT #BIT03,@WCHR ;ANY HIT
              BEQ 8$      ;NO
              MOV #0,@WCMR ;DISABLE MAINT. MODE
              MOV (R1),R2  ;WRITE COMP
              MOV (R1),R2  ;READ COMP
              BIT #BIT03,@WCHR ;SHOULD HAVE HIT
              BEQ 8$      ;NO HIT
9$:          CMP #2000,R1 ;LOOP COMPLETE YET
              BNE 7$      ;NO
              BR 10$

```

10617	034626	012737	001015	177746
10618	034634	005037	004550	
10619	034640	005037	004552	
10620	034644	010137	004554	
10621	034650	012737	000001	004542
10622	034656	005237	004556	
10623	034662	004037	005352	
10624	034666	000006		
10625	034670	041720		
10626	034672	041760		
10627	034674	041744		
10628	034676	012737	000005	177746
10629	034704	053737	004542	005064
10630	034712	000741		
10631	034714	012737	001015	177746
10632	034722	013737	005064	004542
10633	034730	004010		
10634	034732	044000		
10635	034734	044000		
10636				
10637				
10638				
10639				
10640				
10641				

8\$:

```

MOV #OFF,@#CCR
CLR @#GOOD
CLR @#BAD
MOV R1,@#ADD
MOV #1,@#ERROR
INC @#GOODBD
JSR RO,@#SETEN
.WORD ^D6
.WORD SEN154
.WORD SEN157
.WORD SEN156
MOV #5,@#CCR
BIS @#ERROR,@#LOPERR
BR 9$

```

```

;DISABLE CACHE
;DATA UNKNOWN
;CACHE ADDRESS
;SET ERROR FLAG
;SET BIT PRINT MODE
;REPORT ERROR
;TAG MARCH PATTERN TESTS
;HIGH CACHE TAG FAILURE
;CACHE ADDRESS , DATA UNKNOWN
;ENABLE CACHE
;SET DELAYED ERROR FLAG

```

10\$:

```

MOV #OFF,@#CCR
MOV @#LOPERR,@#ERROR
JSR RO,(R0)
.WORD LOWSP
.WORD LOWSP

```

```

;DISABLE CACHE
;TAKE SELECTED ACTION ON ERROR

```

.SBTTL PARITY INTERRUPT TESTS

10643
10644
10645
10646
10647
10648
10649
10650
10651
10652
10653 034736 005267 143704
10654 034742 004467 151146
10655 034746 035126
10656 034750 012737 000000 177776
10657 034756 005037 177744
10658 034762 012737 000005 177746
10659 034770 013702 052000
10660 034774 013702 046000
10661 035000 052737 000100 177746
10662 035006 012737 000077 046000
10663 035014 012737 001015 177746
10664 035022 010702
10665 035024 014212
10666 035026 020227 044000
10667 035032 001374
10668 035034 012737 000005 177746
10669 035042 013702 046000
10670 035046 042737 000001 177746
10671 035054 013737 177746 004542
10672 035062 012737 001015 177746
10673 035070 005137 004542
10674 035074 042737 177776 004542
10675 035102 004037 005352
10676 035106 000010
10677 035110 041772
10678 035112 042002
10679 035114 042022
10680 035116 042034
10681 035120 004010
10682 035122 044000
10683 035124 044000
10684
10685
10686
10687
10688
10689
10690
10691
10692
10693 035126 005267 143514
10694 035132 004467 150756
10695 035136 035276
10696 035140 005037 177744
10697 035144 012737 000005 177746
10698 035152 013702 052000

:
: WRITE WRONG PARITY TO CACHE
: READ CACHE LOCATION , CAUSE PARITY ERROR
:

TST131:

```

INC TID ;UPDATE TEST ID
JSR R4,RELCTL ;RELOCATE TEST TO LOW CACHE
.WORD TST132
MOV #0,@#PSW ;LOWER PRIOR.
CLR @#CMPE ;RESET ERROR REGISTER
MOV #5,@#CCR ;ENABLE HIGH CACHE
MOV @#HIGH1,R2 ;TAG HIGH LOCATION BLOCK #2
MOV @#HIGHSP,R2 ;TAG LOCATION
BIS #BIT06,@#CCR ;SET WRITE WRONG DATA MODE
MOV #77,@#HIGHSP ;WRITE WRONG PARITY DATA
MOV #OFF,@#CCR ;DISABLE WRITE WRONG
MOV PC,R2 ;UNTAG LOW CACHE
1$: MOV -(R2),(R2) ;CAUSE WRITE IN BYPASS MODE
CMP R2,#LOWSP
BNE 1$
MOV #5,@#CCR ;ENABLE HIGH CACHE
MOV @#HIGHSP,R2 ;CAUSE PARITY ERROR
BIC #BIT00,@#CCR ;ENABLE PARITY INTERRUPT
MOV @#CCR,@#ERROR
MOV #OFF,@#CCR ;DISABLE CACHE
COM @#ERROR ;FOR BIT00 ERROR CHECK
BIC #177776,@#ERROR ;MASK ALL OTHERS
JSR R0,@#SETEN ;REPORT ERROR IF ANY
.WORD *DB
.WORD SEN158 ;PARITY INTERRUPT TESTS
.WORD SEN159 ;CACHE FAILED TO INTERRUPT ON PARITY ERROR
.WORD SEN160 ;USING CACHE INTERRUPT LOGIC
.WORD SEN161 ;BIT00 OF CCR USED
JSR R0,(R0) ;TAKE SELECTED ACTION ON ERROR
.WORD LOWSP
.WORD LOWSP
    
```

:
: WRITE WRONG PARITY TO CACHE
: ENABLE PARITY ERROR ABORT FOR HIGH CACHE
: READ AND CAUSE PARITY ERROR

TST132:

```

INC TID ;UPDATE TEST ID
JSR R4,RELCTL ;RELOCATE TEST TO LOW CACHE
.WORD TST133
CLR @#CMPE ;RESET ERROR REGISTER
MOV #5,@#CCR ;ENABLE HIGH CACHE
MOV @#HIGH1,R2 ;TAG HIGH LOCATION BLOCK #2
    
```

CACHE DIAG.
CFKKAB.P11MACY11 30A(1052)
25-JUN-79 13:3131-OCT-79 15:29 PAGE 75-6
PARITY INTERRUPT TESTS

SEQ 0121

10699	035156	013702	046000		MOV @HIGHSP,R2	:TAG HIGH LOCATION BLOCK #1
10700	035162	052737	000100	177746	BIS #BIT06,@CCR	:SET WRITE WRONG DATA MODE
10701	035170	012737	000077	046000	MOV #77,@HIGHSP	:WRITE WRONG DATA PARITY
10702	035176	012737	001015	177746	MOV #OFF,@CCR	:UNTAG LOW CACHE
10703	035204	010702			MOV PC,R2	
10704	035206	014212			1\$: MOV -(R2),(R2)	:CAUSE WRITE IN BYPASS MODE
10705	035210	022702	044000		CMP #LOWSP,R2	
10706	035214	001374			BNE 1\$	
10707	035216	012737	000205	177746	MOV #5+BIT07,@CCR	:ENABLE ABORT
10708	035224	013702	046000		MOV @HIGHSP,R2	:CAUSE PARITY ERROR
10709	035230	013737	177746	004542	MOV @CCR,@ERROR	:
10710	035236	042737	177577	004542	BIC #-BIT07-1,@ERROR	:VERIFY INTERRUPT
10711	035244	012737	001015	177746	MOV #OFF,@CCR	:DISABLE CACHE
10712	035252	004037	005352		JSR R0,@SETEN	:REPORT ERROR IF ANY
10713	035256	000010			.WORD ^D8	
10714	035260	041772			.WORD SEN158	:PARITY INTERRUPT TESTS
10715	035262	042002			.WORD SEN159	:CACHE FAILED TO INTERRUPT ON PARITY ERROR
10716	035264	042046			.WORD SEN162	:USING CACHE ABORT LOGIC IN HIGH CACHE
10717	035266	042066			.WORD SEN163	:BIT07 IN CCR
10718	035270	004010			JSR R0,(R0)	:TAKE SELECTED ACTION ON ERROR
10719	035272	044000			.WORD LOWSP	:LOOP ON ERROR
10720	035274	044000			.WORD LOWSP	:LOOP ON TEST
10721						
10722						
10723						
10724						
10725						
10726						
10727						
10728						
10729						
10730	035276	005267	143344		TST133: INC TID	:UPDATE TEST ID
10731	035302	004467	150634		JSR R4,RELCTH	:RELOCATE TEST TO HIGH CACHE
10732	035306	035446			.WORD TST134	
10733	035310	005037	177744		CLR @CMPE	:RESET ERROR REGISTER
10734	035314	012737	000011	177746	MOV #11,@CCR	:ENABLE LOW CACHE
10735	035322	013702	050000		MOV @LOW1,R2	:TAG LOW BLOCK #2
10736	035326	013702	044000		MOV @LOWSP,R2	:TAG LOW BLOCK #1
10737	035332	052737	000100	177746	BIS #BIT06,@CCR	:ENABLE WRITE WRONG DATA
10738	035340	012737	000077	044000	MOV #77,@LOWSP	:WRITE WRONG DATA
10739	035346	012737	001015	177746	MOV #OFF,@CCR	:DISABLE CACHE
10740	035354	010702			MOV PC,R2	:UNTAG HIGH CACHE
10741	035356	014212			1\$: MOV -(R2),(R2)	
10742	035360	020227	046000		CMP R2,#HIGHSP	
10743	035364	001374			BNE 1\$	
10744	035366	012737	000211	177746	MOV #11+BIT07,@CCR	:ENABLE ABORT
10745	035374	013702	044000		MOV @LOWSP,R2	:CAUSE PARITY ERROR
10746	035400	013737	177746	004542	MOV @CCR,@ERROR	:SAVE ERROR REGISTER
10747	035406	012737	001015	177746	MOV #OFF,@CCR	:DISABLE CACHE
10748	035414	042737	177577	004542	BIC #-BIT07-1,@ERROR	
10749	035422	004037	005352		JSR R0,@SETEN	:REPORT ERROR IF ANY
10750	035426	000010			.WORD ^D8	
10751	035430	041772			.WORD SEN158	:PARITY INTERRUPT TESTS
10752	035432	042002			.WORD SEN159	:CACHE FAILED TO INTERRUPT ON PARITY ERROR
10753	035434	042076			.WORD SEN164	:USING CACHE ABORT LOGIC IN LOW CACHE
10754	035436	042066			.WORD SEN163	:BIT07 IN CCR

```

:
: WRITE WRONG PARITY TO CACHE
: ENABLE PARITY ERROR ABORT FOR LOW CACHE
: READ AND CAUSE PARITY ERROR
:

```

```

10755 035440 004010          JSR R0,(R0)          ;TAKE SELECTED ACTION ON ERROR
10756 035442 046000          .WORD HIGHSP        ;LOOP ON ERROR
10757 035444 046000          .WORD HIGHSP        ;LOOP ON TEST
10758
10759
10760
10761
10762          :          CAUSE PARITY INTERRUPT WITH
10763          :          ABORT AND PARITY INTERRUPT ENABLED
10764          :          VERIFY THAT CACHE ONLY ABORTS
10765
10766
10767 035446 005267 143174      TST134:          INC TID              ;UPDATE TEST ID
10768 035452 004467 150464          JSR R4,RELCTH      ;RELOCATE TEST TO HIGH CACHE
10769 035456 035636          .WORD TST135
10770 035460 012737 000011 177746      MOV #11,@#CCR      ;ENABLE LOW CACHE
10771 035466 013702 050000          MOV @#LOW1,R2     ;TAG LOW LOCATION BLOCK #2
10772 035472 013702 044000          MOV @#LOWSP,R2    ;TAG LOW LOCATION BLOCK #1
10773 035476 052737 000100 177746      BIS #BIT06,@#CCR  ;WRITE WRONG DATA MODE
10774 035504 012737 000077 044000      MOV #77,@#LOWSP   ;WRITE WRONG DATA PARITY
10775 035512 012737 001015 177746      MOV #OFF,@#CCR
10776 035520 010702          MOV PC,R2         ;UNTAG HIGH CACHE
10777 035522 014212          1$: MOV -(R2),(R2)    ;CAUSE WRITE WITH BYPASS
10778 035524 020227 046000          CMP R2,#HIGHSP
10779 035530 001374          BNE 1$
10780 035532 005002          CLR R2            ;REGISTER FOR ABORT CHECK
10781 035534 005037 177744          CLR @#CMPE        ;RESET ERROR REGISTER
10782 035540 012737 000210 177746      MOV #210,@#CCR    ;ENABLE ABORT AND TRAP
10783 035546 013702 044000          MOV @#LOWSP,R2   ;CAUSE ABORT
10784 035552 013737 177746 004542      MOV @#CCR,@#ERROR ;DID PARITY ERROR OCCUR
10785 035560 012737 001015 177746      MOV #OFF,@#CCR   ;DISABLE CACHE
10786 035566 042737 177776 004542      BIC #-BIT00-1,@#ERROR ;LOOK AT BIT00 ONLY
10787 035574 001402          BEQ 2$            ;NO PARITY ERROR
10788 035576 005702          TST R2           ;DID CACHE ABORT
10789 035600 001006          BNE 3$            ;NO
10790 035602 005037 004542          2$: CLR @#ERROR     ;RESET ERROR FLAG
10791 035606 004010          4$: JSR R0,(R0)      ;TAKE SELECTED ACTION ON ERROR
10792 035610 046000          .WORD HIGHSP      ;LOOP ON ERROR
10793 035612 046000          .WORD HIGHSP      ;LOOP ON TEST
10794 035614 000410          BR TST135
10795 035616 004037 005352          3$: JSR R0,@#SETEN   ;REPORT ERROR
10796 035622 000010          .WORD ^DB
10797 035624 041772          .WORD SEN158      ;PARITY INTERRUPT TESTS
10798 035626 042116          .WORD SEN165      ;PARITY ERROR ABORT AND TRAP BOTH ENABLED
10799 035630 042136          .WORD SEN166      ;PARITY ERROR OCCURED AND CACHE TRAPPED
10800 035632 042154          .WORD SEN167      ;CACHE SHOULD HAVE ABORTED
10801 035634 000764          BR 4$
10802
10803
10804
10805
10806 035636 005267 143006      TST135:          INC $PASS           ;UPDATE PASS COUNT
10807 035642 012702 035676          MOV #DEAD,R2     ;PRINT END ON PASS
10808 035646 004767 150212          JSR PC,TYPE
10809 035652 013700 000042          MOV @#42,R0
10810 035656 001405          BEQ REST         ;GIT RETURN ADDRESS
                    ;RESTART TEST

```

CACHE DIAG.
CFKKAB.P11

MACY11 30A(1052)
25-JUN-79 13:31

31-OCT-79 15:29 PAGE 75-8
PARITY INTERRUPT TESTS

SEQ 0123

10811	035660	000005				RESET	
10812	035662	004710			ENDIT:	JSR PC,(R0)	;GO WHERE YOU GO
10813	035664	000240				NOP	
10814	035666	000240				NOP	
10815	035670	000240				NOP	
10816	035672	000167	143102		REST:	JMP PREPARE	;RESTART DIAG.
10817	035676	040520	051523	041440	DEAD:	.ASCII /PASS COMPLETE/	
10818	035713	015	012	177		.BYTE 15,12,177,177,177,0	
10819		035722				.EVEN	

Line	Code	Key	Key	Key	Label	Text
10821						.SBTTL TEXT SENTENCECES
10822	035722	042210	042216	042227	SEN1:	.WORD DIC1,DIC2,DIC3,DIC4,0
10823	035734	042246	042256	042265	SEN2:	.WORD DIC5,DIC6,DIC7,DIC2,DIC8,DIC9,DIC10,0
10824	035754	042246	042313	042216	SEN3:	.WORD DIC5,DIC11,DIC2,DIC8,DIC9,DIC10,0
10825	035772	042246	042323	042216	SEN4:	.WORD DIC5,DIC12,DIC2,DIC8,DIC9,DIC10,0
10826	036010	042246	042337	042216	SEN5:	.WORD DIC5,DIC13,DIC2,DIC8,DIC9,DIC10,0
10827	036026	042246	042343	042353	SEN6:	.WORD DIC5,DIC14,DIC15,DIC16,DIC17,DIC18,DIC19,DIC9,DIC10,0
10828	036052	042246	042343	042353	SEN7:	.WORD DIC5,DIC14,DIC15,DIC20,DIC17,DIC18,DIC19,DIC9,DIC10,0
10829	036076	042417	042426	042433	SEN8:	.WORD DIC21,DIC22,DIC23,DIC24,DIC25,DIC26,0
10830	036114	042455	042216	042466	SEN9:	.WORD DIC27,DIC2,DIC28,DIC7,0
10831	036126	042501	042451	042507	SEN10:	.WORD DIC29,DIC26,DIC30,DIC31,DIC23,DIC24,DIC32,DIC33,0
10832	036150	042313	042216	042532	SEN11:	.WORD DIC11,DIC2,DIC34,DIC35,0
10833	036162	042501	042525	042507	SEN12:	.WORD DIC29,DIC33,DIC30,DIC31,DIC23,DIC24,DIC32,DIC26,0
10834	036204	042501	042525	042507	SEN13:	.WORD DIC29,DIC33,DIC30,DIC31,DIC36,DIC24,DIC26,0
10835	036224	042501	042451	042507	SEN14:	.WORD DIC29,DIC26,DIC30,DIC31,DIC36,DIC24,DIC33,0
10836	036244	042501	042525	042507	SEN15:	.WORD DIC29,DIC33,DIC30,DIC31,DIC37,DIC24,DIC26,0
10837	036264	042501	042451	042507	SEN16:	.WORD DIC29,DIC26,DIC30,DIC31,DIC37,DIC24,DIC33,0
10838	036304	042501	042525	042507	SEN17:	.WORD DIC29,DIC33,DIC30,DIC31,DIC38,DIC24,DIC26,0
10839	036324	042501	042451	042507	SEN18:	.WORD DIC29,DIC26,DIC30,DIC31,DIC38,DIC24,DIC33,0
10840	036344	042501	042525	042507	SEN19:	.WORD DIC29,DIC33,DIC30,DIC31,DIC39,DIC24,DIC26,0
10841	036364	042501	042451	042507	SEN20:	.WORD DIC29,DIC26,DIC30,DIC31,DIC39,DIC24,DIC33,0
10842	036404	042501	042525	042507	SEN21:	.WORD DIC29,DIC33,DIC30,DIC31,DIC40,DIC24,DIC26,0
10843	036424	042501	042451	042507	SEN22:	.WORD DIC29,DIC26,DIC30,DIC31,DIC40,DIC24,DIC33,0
10844	036444	042501	042525	042507	SEN23:	.WORD DIC29,DIC33,DIC30,DIC31,DIC41,DIC24,DIC26,0
10845	036464	042501	042451	042507	SEN24:	.WORD DIC29,DIC26,DIC30,DIC31,DIC41,DIC24,DIC33,0
10846	036504	042313	042216	042417	SEN25:	.WORD DIC11,DIC2,DIC21,DIC42,DIC35,0
10847	036520	042441	042451	042614	SEN26:	.WORD DIC24,DIC26,DIC43,DIC21,DIC31,DIC44,0
10848	036536	042441	042451	042614	SEN27:	.WORD DIC24,DIC26,DIC43,DIC21,DIC31,DIC45,0
10849	036554	042441	042451	042614	SEN28:	.WORD DIC24,DIC26,DIC43,DIC21,DIC31,DIC46,0
10850	036572	042441	042451	042614	SEN29:	.WORD DIC24,DIC26,DIC43,DIC21,DIC31,DIC47,0
10851	036610	042441	042451	042614	SEN30:	.WORD DIC24,DIC26,DIC43,DIC21,DIC31,DIC48,0
10852	036626	042441	042451	042614	SEN31:	.WORD DIC24,DIC26,DIC43,DIC21,DIC31,DIC49,0
10853	036644	042441	042451	042614	SEN32:	.WORD DIC24,DIC26,DIC43,DIC21,DIC31,DIC50,0
10854	036662	042210	042313	042216	SEN33:	.WORD DIC1,DIC11,DIC2,DIC51,DIC4,0
10855	036676	042501	042525	042507	SEN34:	.WORD DIC29,DIC33,DIC30,DIC52,DIC51,DIC44,0
10856	036714	042501	042451	042507	SEN35:	.WORD DIC29,DIC26,DIC30,DIC53,DIC51,DIC40,0
10857	036732	042441	042525	042614	SEN36:	.WORD DIC24,DIC33,DIC43,DIC40,0
10858	036744	042501	042525	042507	SEN37:	.WORD DIC29,DIC33,DIC30,DIC53,DIC51,DIC41,0
10859	036762	042501	042451	042507	SEN38:	.WORD DIC29,DIC26,DIC30,DIC52,DIC51,DIC36,0
10860	037000	042441	042525	042614	SEN39:	.WORD DIC24,DIC33,DIC43,DIC36,DIC54,DIC24,DIC26,DIC43,DIC41,0
10861	037024	042323	042216	042532	SEN40:	.WORD DIC12,DIC2,DIC34,DIC35,0
10862	037036	042501	042525	042507	SEN41:	.WORD DIC29,DIC33,DIC30,DIC55,DIC24,DIC26,0
10863	037054	042501	042451	042507	SEN42:	.WORD DIC29,DIC26,DIC30,DIC55,DIC24,DIC33,0
10864	037072	042441	042451	042614	SEN43:	.WORD DIC24,DIC26,DIC43,DIC21,DIC55,DIC44,0
10865	037110	042441	042451	042614	SEN44:	.WORD DIC24,DIC26,DIC43,DIC21,DIC55,DIC36,0
10866	037126	042532	042720	042537	SEN45:	.WORD DIC34,DIC56,DIC35,0
10867	037136	042725	042733	042737	SEN46:	.WORD DIC57,DIC58,DIC59,DIC60,DIC53,DIC1,0
10868	037154	042747	042752	042757	SEN47:	.WORD DIC61,DIC62,DIC63,DIC34,DIC24,0
10869	037170	042532	042762	042441	SEN48:	.WORD DIC34,DIC64,DIC24,DIC25,DIC33,0
10870	037204	042725	042733	042771	SEN49:	.WORD DIC57,DIC58,DIC65,DIC60,DIC53,DIC1,0
10871	037222	042532	042762	042441	SEN50:	.WORD DIC34,DIC64,DIC24,DIC25,DIC26,0
10872	037236	042532	042720	042610	SEN51:	.WORD DIC34,DIC56,DIC42,DIC66,DIC35,0
10873	037252	043005	042451	043014	SEN52:	.WORD DIC67,DIC26,DIC68,DIC69,DIC70,DIC65,0
10874	037270	043035	042700	042210	SEN53:	.WORD DIC71,DIC52,DIC1,0
10875	037300	043035	042704	042210	SEN54:	.WORD DIC71,DIC53,DIC1,0
10876	037310	043005	042525	043014	SEN55:	.WORD DIC67,DIC33,DIC68,DIC72,DIC69,DIC70,DIC59,0

10877	037330	043035	043047	042704	SEN56:	.WORD DIC71,DIC73,DIC53,DIC1,0
10878	037342	042210	043052	042240	SEN57:	.WORD DIC1,DIC74,DIC4,0
10879	037352	043052	043047	043060	SEN58:	.WORD DIC74,DIC73,DIC75,DIC42,DIC76,DIC60,DIC77,0
10880	037372	042446	043104	043032	SEN59:	.WORD DIC25,DIC78,DIC70,DIC79,DIC1,DIC74,DIC42,0
10881	037412	043052	043047	043060	SEN60:	.WORD DIC74,DIC73,DIC75,DIC76,DIC60,DIC80,0
10882	037430	042302	043131	043052	SEN61:	.WORD DIC9,DIC81,DIC74,DIC60,DIC82,DIC83,DIC10,0
10883	037450	043152	042610	043047	SEN62:	.WORD DIC84,DIC42,DIC73,DIC85,DIC42,DIC17,DIC18,DIC77,DIC25,0
10884	037474	043104	043032	043052	SEN63:	.WORD DIC78,DIC70,DIC74,0
10885	037504	043152	043100	043164	SEN64:	.WORD DIC84,DIC77,DIC86,DIC42,DIC76,DIC60,DIC80,0
10886	037524	043172	043052	000000	SEN65:	.WORD DIC87,DIC74,0
10887	037532	043052	043071	042744	SEN66:	.WORD DIC74,DIC76,DIC60,DIC88,DIC1,0
10888	037546	043035	042700	042210	SEN67:	.WORD DIC71,DIC52,DIC1,DIC89,DIC84,DIC90,DIC77,DIC72,0
10889	037570	042455	043052	043226	SEN68:	.WORD DIC27,DIC74,DIC91,DIC42,DIC92,0
10890	037604	043035	042704	042210	SEN69:	.WORD DIC71,DIC53,DIC1,DIC89,DIC84,DIC90,DIC77,DIC72,0
10891	037626	043035	042704	042210	SEN70:	.WORD DIC71,DIC53,DIC1,DIC89,DIC93,DIC90,DIC77,DIC72,0
10892	037650	043035	042700	042210	SEN71:	.WORD DIC71,DIC52,DIC1,DIC89,DIC84,DIC90,DIC77,DIC94,0
10893	037672	043035	042704	042210	SEN72:	.WORD DIC71,DIC53,DIC1,DIC89,DIC84,DIC90,DIC77,DIC94,0
10894	037714	043254	043262	042240	SEN73:	.WORD DIC95,DIC96,DIC4,0
10895	037724	042700	042210	043267	SEN74:	.WORD DIC52,DIC1,DIC97,DIC98,DIC99,DIC100,0
10896	037742	042704	042210	043267	SEN75:	.WORD DIC53,DIC1,DIC97,DIC100,DIC99,DIC98,0
10897	037760	042700	042210	043267	SEN76:	.WORD DIC52,DIC1,DIC97,DIC100,DIC99,DIC98,0
10898	037776	042704	042210	043267	SEN77:	.WORD DIC53,DIC1,DIC97,DIC98,DIC99,DIC100,0
10899	040014	042725	043323	042256	SEN78:	.WORD DIC57,DIC101,DIC6,DIC4,0
10900	040026	042501	043323	042256	SEN79:	.WORD DIC29,DIC101,DIC6,DIC60,DIC52,DIC51,DIC53,DIC51,DIC102,DIC103,0
10901	040054	042700	042673	042256	SEN80:	.WORD DIC52,DIC51,DIC6,DIC7,DIC38,DIC18,DIC77,DIC73,DIC22,0
10902	040100	042704	042673	042256	SEN81:	.WORD DIC53,DIC51,DIC6,DIC7,DIC39,DIC18,DIC77,DIC73,DIC22,0
10903	040124	043335	042256	042265	SEN82:	.WORD DIC103,DIC6,DIC7,DIC46,DIC18,DIC77,DIC73,DIC22,0
10904	040146	042337	042216	042240	SEN83:	.WORD DIC13,DIC2,DIC4,0
10905	040156	042210	042337	042216	SEN84:	.WORD DIC1,DIC13,DIC2,DIC64,DIC104,DIC53,0
10906	040174	042337	042610	042265	SEN85:	.WORD DIC13,DIC42,DIC7,0
10907	040204	043347	042744	042725	SEN86:	.WORD DIC105,DIC60,DIC57,DIC13,DIC2,DIC46,0
10908	040222	042744	043045	042451	SEN87:	.WORD DIC60,DIC72,DIC26,DIC106,DIC107,DIC24,DIC13,DIC106,DIC76,0
10909	040246	042210	042532	042725	SEN88:	.WORD DIC1,DIC34,DIC57,DIC51,DIC35,0
10910	040262	042210	042532	042725	SEN89:	.WORD DIC1,DIC34,DIC57,DIC51,DIC7,0
10911	040276	043365	042725	042700	SEN90:	.WORD DIC108,DIC57,DIC52,DIC51,0
10912	040310	043365	042725	042704	SEN91:	.WORD DIC108,DIC57,DIC53,DIC51,0
10913	040322	042256	042265	042216	SEN92:	.WORD DIC6,DIC7,DIC2,DIC4,0
10914	040334	042665	043032	042426	SEN93:	.WORD DIC50,DIC70,DIC22,DIC109,DIC18,DIC77,DIC25,DIC78,0
10915	040356	043032	043335	042256	SEN94:	.WORD DIC70,DIC103,DIC6,DIC7,0
10916	040370	042665	043032	042426	SEN95:	.WORD DIC50,DIC70,DIC22,DIC17,DIC18,DIC77,0
10917	040406	042446	043104	043032	SEN96:	.WORD DIC25,DIC78,DIC70,DIC52,DIC51,DIC6,DIC7,0
10918	040426	042665	043032	042426	SEN97:	.WORD DIC50,DIC70,DIC22,DIC109,DIC18,DIC77,0
10919	040444	042446	043104	043032	SEN98:	.WORD DIC25,DIC78,DIC70,DIC53,DIC51,DIC6,DIC7,0
10920	040464	042725	042744	042426	SEN99:	.WORD DIC57,DIC60,DIC22,DIC76,DIC60,DIC80,DIC2,0
10921	040504	042256	042265	043376	SEN100:	.WORD DIC6,DIC7,DIC110,DIC4,0
10922	040516	042700	042673	042256	SEN101:	.WORD DIC52,DIC51,DIC6,DIC92,0
10923	040530	043071	042757	043404	SEN102:	.WORD DIC76,DIC63,DIC111,DIC102,DIC112,DIC24,0
10924	040546	042704	042673	042256	SEN103:	.WORD DIC53,DIC51,DIC6,DIC92,0
10925	040560	043335	042256	043236	SEN104:	.WORD DIC103,DIC6,DIC92,0
10926	040570	042256	042265	042610	SEN105:	.WORD DIC6,DIC7,DIC42,DIC76,DIC60,DIC77,DIC63,DIC101,0
10927	040612	042725	043404	043331	SEN106:	.WORD DIC57,DIC111,DIC102,DIC112,0
10928	040624	042441	042725	042337	SEN107:	.WORD DIC24,DIC57,DIC13,DIC4,0
10929	040636	042747	042752	042446	SEN108:	.WORD DIC61,DIC62,DIC25,DIC78,DIC70,DIC113,DIC60,DIC114,DIC115,DIC116,0
10930	040664	043450	043454	042240	SEN109:	.WORD DIC117,DIC118,DIC4,0
10931	040674	042441	042507	043450	SEN110:	.WORD DIC24,DIC30,DIC117,DIC118,DIC8,DIC13,0
10932	040712	042441	042507	043450	SEN111:	.WORD DIC24,DIC30,DIC117,DIC118,DIC119,DIC115,DIC120,0

10933	040732	043506	043047	042323	SEN112:	.WORD	DIC121,DIC73,DIC12,DIC122,0
10934	040744	043152	042610	043521	SEN113:	.WORD	DIC84,DIC42,DIC123,DIC4,0
10935	040756	043531	043047	042700	SEN114:	.WORD	DIC124,DIC73,DIC52,DIC1,DIC125,DIC84,DIC77,DIC72,0
10936	041000	042353	042610	042265	SEN115:	.WORD	DIC15,DIC42,DIC7,0
10937	041010	042704	042210	043313	SEN116:	.WORD	DIC53,DIC1,DIC100,DIC89,DIC84,DIC77,DIC94,0
10938	041030	043550	043514	042240	SEN117:	.WORD	DIC126,DIC122,DIC4,0
10939	041040	042704	042210	043475	SEN118:	.WORD	DIC53,DIC1,DIC120,DIC18,DIC119,0
10940	041054	043557	042725	042744	SEN119:	.WORD	DIC127,DIC57,DIC60,DIC120,DIC73,DIC126,DIC122,0
10941	041074	042537	043562	043543	SEN120:	.WORD	DIC35,DIC128,DIC125,DIC84,DIC90,DIC77,DIC72,0
10942	041114	043213	043152	043221	SEN121:	.WORD	DIC89,DIC84,DIC90,DIC77,DIC94,0
10943	041130	042246	043427	043475	SEN122:	.WORD	DIC5,DIC115,DIC120,DIC73,DIC126,DIC122,0
10944	041146	042273	043475	042744	SEN123:	.WORD	DIC8,DIC120,DIC60,DIC88,0
10945	041160	043335	042610	043566	SEN124:	.WORD	DIC103,DIC42,DIC129,DIC24,DIC57,DIC4,0
10946	041176	042747	042337	042614	SEN125:	.WORD	DIC61,DIC13,DIC43,DIC24,DIC30,DIC130,DIC120,0
10947	041216	042353	043603	000000	SEN126:	.WORD	DIC15,DIC131,0
10948	041224	043323	043335	042441	SEN127:	.WORD	DIC101,DIC103,DIC24,DIC43,DIC13,DIC2,0
10949	041242	042446	043104	043032	SEN128:	.WORD	DIC25,DIC78,DIC70,DIC130,DIC24,0
10950	041256	042353	054000	000000	SEN129:	.WORD	DIC15,DIC132,0
10951	041264	043574	042353	043331	SEN130:	.WORD	DIC130,DIC15,DIC102,DIC13,DIC2,DIC34,0
10952	041302	042353	054007	000000	SEN131:	.WORD	DIC15,DIC133,0
10953	041310	042353	054016	000000	SEN132:	.WORD	DIC15,DIC134,0
10954	041316	042353	054025	000000	SEN133:	.WORD	DIC15,DIC135,0
10955	041324	042353	054034	000000	SEN134:	.WORD	DIC15,DIC136,0
10956	041332	042353	054043	000000	SEN135:	.WORD	DIC15,DIC137,0
10957	041340	042353	054052	000000	SEN136:	.WORD	DIC15,DIC138,0
10958	041346	042210	043574	042353	SEN137:	.WORD	DIC1,DIC130,DIC15,DIC35,0
10959	041360	054061	042210	042353	SEN138:	.WORD	DIC139,DIC1,DIC15,DIC30,DIC1,DIC34,0
10960	041376	042210	043574	043475	SEN139:	.WORD	DIC1,DIC130,DIC120,DIC102,DIC34,0
10961	041412	054071	054100	054112	SEN140:	.WORD	DIC140,DIC141,DIC142,DIC35,0
10962	041424	043427	042210	042372	SEN141:	.WORD	DIC115,DIC1,DIC17,DIC18,DIC88,DIC127,DIC142,DIC57,0
10963	041446	043213	054071	054100	SEN142:	.WORD	DIC89,DIC140,DIC141,DIC143,0
10964	041460	042532	042610	054125	SEN143:	.WORD	DIC34,DIC42,DIC144,DIC145,DIC35,0
10965	041474	042210	042353	043357	SEN144:	.WORD	DIC1,DIC15,DIC106,DIC34,DIC146,DIC106,DIC34,DIC24,0
10966	041516	042532	042256	054125	SEN145:	.WORD	DIC34,DIC6,DIC144,DIC145,DIC35,0
10967	041532	042704	042673	042256	SEN146:	.WORD	DIC53,DIC51,DIC6,DIC7,0
10968	041544	042210	042353	043357	SEN147:	.WORD	DIC1,DIC15,DIC106,DIC6,DIC147,DIC106,DIC6,DIC24,0
10969	041566	042700	042673	042256	SEN148:	.WORD	DIC52,DIC51,DIC6,DIC7,0
10970	041600	043152	042610	054125	SEN149:	.WORD	DIC84,DIC42,DIC144,DIC145,DIC35,0
10971	041614	043152	042610	043100	SEN150:	.WORD	DIC84,DIC42,DIC77,DIC94,DIC86,0
10972	041630	042210	042353	043357	SEN151:	.WORD	DIC1,DIC15,DIC106,DIC84,DIC42,DIC147,DIC106,DIC84,DIC42,DIC24,0
10973	041656	043152	042610	043100	SEN152:	.WORD	DIC84,DIC42,DIC77,DIC72,DIC86,0
10974	041672	042210	042353	043357	SEN153:	.WORD	DIC1,DIC15,DIC106,DIC84,DIC42,DIC148,DIC106,DIC84,DIC42,DIC24,0
10975	041720	043335	054125	054133	SEN154:	.WORD	DIC103,DIC144,DIC145,DIC4,0
10976	041732	042700	042210	043335	SEN155:	.WORD	DIC52,DIC1,DIC103,DIC92,0
10977	041744	042210	042353	043357	SEN156:	.WORD	DIC1,DIC15,DIC106,DIC34,DIC149,0
10978	041760	042704	042210	043335	SEN157:	.WORD	DIC53,DIC1,DIC103,DIC92,0
10979	041772	042256	054203	042240	SEN158:	.WORD	DIC6,DIC150,DIC4,0
10980	042002	042210	043071	042744	SEN159:	.WORD	DIC1,DIC76,DIC60,DIC150,DIC63,DIC6,DIC7,0
10981	042022	043213	042210	054203	SEN160:	.WORD	DIC89,DIC1,DIC150,DIC110,0
10982	042034	042433	043032	042514	SEN161:	.WORD	DIC23,DIC70,DIC31,DIC151,0
10983	042046	043213	042210	054222	SEN162:	.WORD	DIC89,DIC1,DIC152,DIC110,DIC73,DIC53,DIC1,0
10984	042066	042566	043047	042514	SEN163:	.WORD	DIC39,DIC73,DIC31,0
10985	042076	043213	042210	054222	SEN164:	.WORD	DIC89,DIC1,DIC152,DIC110,DIC73,DIC52,DIC1,0
10986	042116	042256	042265	054222	SEN165:	.WORD	DIC6,DIC7,DIC152,DIC102,DIC153,DIC154,DIC100,0
10987	042136	042256	042265	054242	SEN166:	.WORD	DIC6,DIC7,DIC155,DIC102,DIC1,DIC156,0
10988	042154	042210	054262	054271	SEN167:	.WORD	DIC1,DIC157,DIC158,DIC159,0

CACHE DIAG. MACY11 30A(1052) 31-OCT-79 15:29 PAGE 75-12
CFKKAB.P11 25-JUN-79 13:31 TEXT SENTENCECES

SEQ 0127

10989 042166 042455 043254 043262 SEN168: .WORD DIC27,DIC95,DIC96,DIC160,DIC161,DIC162,DIC160,DIC163,0

10991						.SBTTL TEXT WORDS
10992	042210	040503	044103	000105	DIC1:	.ASCIZ *CACHE*
10993	042216	042522	044507	052123	DIC2:	.ASCIZ *REGISTER*
10994	042227	122	051505	047520	DIC3:	.ASCIZ *RESPONSE*
10995	042240	042524	052123	000123	DIC4:	.ASCIZ *TESTS*
10996	042246	042522	042101	047111	DIC5:	.ASCIZ *READING*
10997	042256	040520	044522	054524	DIC6:	.ASCIZ *PARITY*
10998	042265	105	051122	051117	DIC7:	.ASCIZ *ERROR*
10999	042273	103	052501	042523	DIC8:	.ASCIZ *CAUSED*
11000	042302	044524	042515	000	DIC9:	.ASCIZ *TIME*
11001	042307	117	052125	000	DIC10:	.ASCIZ *OUT*
11002	042313	103	047117	051124	DIC11:	.ASCIZ *CONTROL*
11003	042323	115	044501	052116	DIC12:	.ASCIZ *MAINTENANCE*
11004	042337	110	052111	000	DIC13:	.ASCIZ *HIT*
11005	042343	111	053116	046101	DIC14:	.ASCIZ *INVALID*
11006	042353	101	042104	042522	DIC15:	.ASCIZ *ADDRESS*
11007	042363	061	033467	032067	DIC16:	.ASCIZ *177740*
11008	042372	044504	000104		DIC17:	.ASCIZ *DID*
11009	042376	047516	000124		DIC18:	.ASCIZ *NOT*
11010	042402	040503	051525	000105	DIC19:	.ASCIZ *CAUSE*
11011	042410	033461	033467	033065	DIC20:	.ASCIZ *177756*
11012	042417	125	052516	042523	DIC21:	.ASCIZ *UNUSED*
11013	042426	046503	042520	000	DIC22:	.ASCIZ *CMPE*
11014	042433	102	052111	030060	DIC23:	.ASCIZ *BIT00*
11015	042441	122	040505	000104	DIC24:	.ASCIZ *READ*
11016	042446	051501	000		DIC25:	.ASCIZ *AS*
11017	042451	117	042516	000	DIC26:	.ASCIZ *ONE*
11018	042455	120	051517	044523	DIC27:	.ASCIZ *POSSIBLE*
11019	042466	042101	051104	051505	DIC28:	.ASCIZ *ADDRESSING*
11020	042501	127	047522	042524	DIC29:	.ASCIZ *WROTE*
11021	042507	111	052116	000117	DIC30:	.ASCIZ *INTO*
11022	042514	041503	000122		DIC31:	.ASCIZ *CCR*
11023	042520	040502	045503	000	DIC32:	.ASCIZ *BACK*
11024	042525	132	051105	000117	DIC33:	.ASCIZ *ZERO*
11025	042532	040504	040524	000	DIC34:	.ASCIZ *DATA*
11026	042537	124	051505	000124	DIC35:	.ASCIZ *TEST*
11027	042544	044502	030124	000062	DIC36:	.ASCIZ *BIT02*
11028	042552	044502	030124	000063	DIC37:	.ASCIZ *BIT03*
11029	042560	044502	030124	000066	DIC38:	.ASCIZ *BIT06*
11030	042566	044502	030124	000067	DIC39:	.ASCIZ *BIT07*
11031	042574	044502	030124	000071	DIC40:	.ASCIZ *BIT09*
11032	042602	044502	030524	000060	DIC41:	.ASCIZ *BIT10*
11033	042610	044502	000124		DIC42:	.ASCIZ *BIT*
11034	042614	051106	046517	000	DIC43:	.ASCIZ *FROM*
11035	042621	102	052111	030460	DIC44:	.ASCIZ *BIT01*
11036	042627	102	052111	032060	DIC45:	.ASCIZ *BIT04*
11037	042635	102	052111	032460	DIC46:	.ASCIZ *BIT05*
11038	042643	102	052111	034060	DIC47:	.ASCIZ *BIT08*
11039	042651	102	052111	030461	DIC48:	.ASCIZ *BIT11*
11040	042657	102	052111	032061	DIC49:	.ASCIZ *BIT14*
11041	042665	102	052111	032461	DIC50:	.ASCIZ *BIT15*
11042	042673	102	052131	000105	DIC51:	.ASCIZ *BYTE*
11043	042700	047514	000127		DIC52:	.ASCIZ *LOW*
11044	042704	044510	044107	000	DIC53:	.ASCIZ *HIGH*
11045	042711	117	000122		DIC54:	.ASCIZ *OR*
11046	042714	046503	000122		DIC55:	.ASCIZ *CMR*

11047	042720	040520	044124	000	DIC56:	.ASCIZ	*PATH*
11048	042725	127	044522	042524	DIC57:	.ASCIZ	*WRITE*
11049	042733	101	046117	000	DIC58:	.ASCIZ	*ALL*
11050	042737	117	042516	000123	DIC59:	.ASCIZ	*ONES*
11051	042744	047524	000		DIC60:	.ASCIZ	*TO*
11052	042747	116	000117		DIC61:	.ASCIZ	*NO*
11053	042752	044510	051524	000	DIC62:	.ASCIZ	*HITS*
11054	042757	117	000116		DIC63:	.ASCIZ	*ON*
11055	042762	044502	024124	024523	DIC64:	.ASCIZ	*BIT(S)*
11056	042771	132	051105	051517	DIC65:	.ASCIZ	*ZEROS*
11057	042777	123	047510	052122	DIC66:	.ASCIZ	*SHORT*
11058	043005	122	052117	052101	DIC67:	.ASCIZ	*ROTATE*
11059	043014	044124	047522	043525	DIC68:	.ASCIZ	*THROUGH*
11060	043024	044506	046105	000104	DIC69:	.ASCIZ	*FIELD*
11061	043032	043117	000		DIC70:	.ASCIZ	*OF*
11062	043035	124	051505	044524	DIC71:	.ASCIZ	*TESTING*
11063	043045	101	000		DIC72:	.ASCIZ	*A*
11064	043047	111	000116		DIC73:	.ASCIZ	*IN*
11065	043052	046106	051525	000110	DIC74:	.ASCIZ	*FLUSH*
11066	043060	051120	043517	042522	DIC75:	.ASCIZ	*PROGRESS*
11067	043071	106	044501	042514	DIC76:	.ASCIZ	*FAILED*
11068	043100	042523	000124		DIC77:	.ASCIZ	*SET*
11069	043104	042522	052523	052114	DIC78:	.ASCIZ	*RESULT*
11070	043113	123	052105	044524	DIC79:	.ASCIZ	*SETTING*
11071	043123	103	042514	051101	DIC80:	.ASCIZ	*CLEAR*
11072	043131	106	051117	000	DIC81:	.ASCIZ	*FOR*
11073	043135	103	046517	046120	DIC82:	.ASCIZ	*COMPLETE*
11074	043146	040522	000116		DIC83:	.ASCIZ	*RAN*
11075	043152	040526	044514	000104	DIC84:	.ASCIZ	*VALID*
11076	043160	051525	000105		DIC85:	.ASCIZ	*USE*
11077	043164	047111	051525	000105	DIC86:	.ASCIZ	*INUSE*
11078	043172	043101	042524	000122	DIC87:	.ASCIZ	*AFTER*
11079	043200	047111	040526	044514	DIC88:	.ASCIZ	*INVALIDATE*
11080	043213	125	044523	043516	DIC89:	.ASCIZ	*USING*
11081	043221	102	052111	000123	DIC90:	.ASCIZ	*BITS*
11082	043226	047503	047125	042524	DIC91:	.ASCIZ	*COUNTER*

11084	043236	040506	046111	051125	DIC92:	.ASCIZ	*FAILURE*
11085	043246	044514	000104		DIC93:	.ASCIZ	*LID*
11086	043252	000102			DIC94:	.ASCIZ	*B*
11087	043254	047506	041522	000105	DIC95:	.ASCIZ	*FORCE*
11088	043262	044515	051523	000	DIC96:	.ASCIZ	*MISS*
11089	043267	114	047517	051513	DIC97:	.ASCIZ	*LOOKS*
11090	043275	104	051511	041101	DIC98:	.ASCIZ	*DISABLED*
11091	043306	044127	047105	000	DIC99:	.ASCIZ	*WHEN*
11092	043313	105	040516	046102	DIC100:	.ASCIZ	*ENABLED*
11093	043323	127	047522	043516	DIC101:	.ASCIZ	*WRONG*
11094	043331	101	042116	000	DIC102:	.ASCIZ	*AND*
11095	043335	124	043501	000	DIC103:	.ASCIZ	*TAG*
11096	043341	123	052524	045503	DIC104:	.ASCIZ	*STUCK*
11097	043347	101	052124	046505	DIC105:	.ASCIZ	*ATTEMPT*
11098	043357	054	000		DIC106:	.ASCIZ	*,*
11099	043361	126	040511	000	DIC107:	.ASCIZ	*VIA*
11100	043365	103	047101	000124	DIC108:	.ASCIZ	*CANT*
11101	043372	040527	000123		DIC109:	.ASCIZ	*WAS*
11102	043376	047514	044507	000103	DIC110:	.ASCIZ	*LOGIC*
11103	043404	042117	000104		DIC111:	.ASCIZ	*ODD*
11104	043410	053105	047105	000	DIC112:	.ASCIZ	*EVEN*
11105	043415	122	040505	051504	DIC113:	.ASCIZ	*READS*
11106	043423	123	054111	000	DIC114:	.ASCIZ	*SIX*
11107	043427	124	043501	042507	DIC115:	.ASCIZ	*TAGGED*
11108	043436	047514	040503	044524	DIC116:	.ASCIZ	*LOCATIONS*
11109	043450	027511	000117		DIC117:	.ASCIZ	*I/O*
11110	043454	040520	042507	000	DIC118:	.ASCIZ	*PAGE*
11111	043461	111	053116	046101	DIC119:	.ASCIZ	*INVALIDATED*
11112	043475	114	041517	052101	DIC120:	.ASCIZ	*LOCATION*
11113	043506	044127	046111	000105	DIC121:	.ASCIZ	*WHILE*
11114	043514	047515	042504	000	DIC122:	.ASCIZ	*MODE*
11115	043521	123	047524	040522	DIC123:	.ASCIZ	*STORAGE*
11116	043531	117	042520	040522	DIC124:	.ASCIZ	*OPERATING*
11117	043543	127	052111	000110	DIC125:	.ASCIZ	*WITH*
11118	043550	054502	040520	051523	DIC126:	.ASCIZ	*BYPASS*
11119	043557	102	000131		DIC127:	.ASCIZ	*BY*
11120	043562	052522	000116		DIC128:	.ASCIZ	*RUN*
11121	043566	040502	044523	000103	DIC129:	.ASCIZ	*BASIC*
11122	043574	042515	047515	054522	DIC130:	.ASCIZ	*MEMORY*
11123	043603	060	030060	030060	DIC131:	.ASCIZ	*000000*
11124							
11125		054000				.=54000	:REV B
11126							
11127							
11128	054000	030060	030064	030060	DIC132:	.ASCIZ	*004000*
11129	054007	060	030061	030060	DIC133:	.ASCIZ	*010000*
11130	054016	031060	030060	030060	DIC134:	.ASCIZ	*020000*
11131	054025	060	030064	030060	DIC135:	.ASCIZ	*040000*
11132	054034	030061	030060	030060	DIC136:	.ASCIZ	*100000*
11133	054043	062	030060	030060	DIC137:	.ASCIZ	*200000*
11134	054052	030064	030060	030060	DIC138:	.ASCIZ	*400000*
11135	054061	127	044522	044524	DIC139:	.ASCIZ	*WRITING*
11136	054071	125	044516	052502	DIC140:	.ASCIZ	*UNIBUS*
11137	054100	054105	051105	044503	DIC141:	.ASCIZ	*EXERCISER*
11138	054112	046504	000101		DIC142:	.ASCIZ	*DMA*
11139	054116	047515	052504	042514	DIC143:	.ASCIZ	*MODULE*

11140	054125	115	051101	044103	DIC144:	.ASCIZ	*MARCH*
11141	054133	120	052101	042524	DIC145:	.ASCIZ	*PATTERN*
11142	054143	105	050130	041505	DIC146:	.ASCIZ	*EXPECTED*
11143	054154	051127	052111	042524	DIC147:	.ASCIZ	*WRITTEN*
11144	054164	051127	052111	047105	DIC148:	.ASCIZ	*WRITEN*
11145	054173	125	045516	047516	DIC149:	.ASCIZ	*UNKNOWN*
11146	054203	111	052116	051105	DIC150:	.ASCIZ	*INTERRUPT*
11147	054215	125	042523	000104	DIC151:	.ASCIZ	*USED*
11148	054222	041101	051117	000124	DIC152:	.ASCIZ	*ABORT*
11149	054230	051124	050101	000	DIC153:	.ASCIZ	*TRAP*
11150	054235	102	052117	000110	DIC154:	.ASCIZ	*BOTH*
11151	054242	041517	052503	042522	DIC155:	.ASCIZ	*OCCURED*
11152	054252	051124	050101	042520	DIC156:	.ASCIZ	*TRAPPED*
11153	054262	044123	052517	042114	DIC157:	.ASCIZ	*SHOULD*
11154	054271	110	053101	000105	DIC158:	.ASCIZ	*HAVE*
11155	054276	041101	051117	042524	DIC159:	.ASCIZ	*ABORTED*
11156	054306	053523	052111	044103	DIC160:	.ASCIZ	*SWITCH*
11157	054315	105	051122	051117	DIC161:	.ASCIZ	*ERROR!!*
11158	054325	126	051105	043111	DIC162:	.ASCIZ	*VERIFY*
11159	054334	047520	044523	044524	DIC163:	.ASCIZ	*POSITIONS*

11161		044000			.=44000
11162	044000	000000	LOWSP:		.WORD 0
11163		046000			.=46000
11164	046000	000000	HIGHSP:		.WORD 0
11165		050000			.=50000
11166	050000	000000	LOW1:		.WORD 0
11167		052000			.=52000
11168	052000	000000	HIGH1:		.WORD 0
11169		000001			.END

ABASE = 000000	6018																		
ACDW1 = 000000	6018																		
ACDW2 = 000000	6018																		
ACPUOP= 000000	6018																		
ACTION 004706	6089*	6101*	6104*	6367#	6383	6389	6401												
ADD 004554	6349#	6604	7579*	7645*	7713*	7781*	9269*	9307*	9347*	9385*	9425*	9479*	9552*						
	9609*	9721*	9899*	9900*	9953*	9956*	10079*	10080*	10133*	10136*	10207*	10208*	10289*						
	10290*	10365*	10366*	10448*	10449*	10513*	10538*	10595*	10620*										
ADDST 006400	6601	6646#																	
ADDW0 = 000000	6018																		
ADDW1 = 000000	6018																		
ADDW10= 000000	6018																		
ADDW11= 000000	6018																		
ADDW12= 000000	6018																		
ADDW13= 000000	6018																		
ADDW14= 000000	6018																		
ADDW15= 000000	6018																		
ADDW2 = 000000	6018																		
ADDW3 = 000000	6018																		
ADDW4 = 000000	6018																		
ADDW5 = 000000	6018																		
ADDW6 = 000000	6018																		
ADDW7 = 000000	6018																		
ADDW8 = 000000	6018																		
ADDW9 = 000000	6018																		
ADEVCT= 000000	6018																		
ADEVN = 000000	6018																		
ADWRD 005642	6536	6542	6549#	6618															
AENV = 000000	6018																		
AENVN = 000000	6018																		
AFATAL= 000000	6018																		
AHALT 001120	6106#	6494	6521																
AMADR1= 000000	6018																		
AMADR2= 000000	6018																		
AMADR3= 000000	6018																		
AMADR4= 000000	6018																		
AMAMS1= 000000	6018																		
AMAMS2= 000000	6018																		
AMAMS3= 000000	6018																		
AMAMS4= 000000	6018																		
AMSGAD= 000000	6018																		
AMSGLG= 000000	6018																		
AMSGTY= 000000	6018																		
AMTYP1= 000000	6018																		
AMTYP2= 000000	6018																		
AMTYP3= 000000	6018																		
AMTYP4= 000000	6018																		
APASS = 000000	6018																		
APRIOR= 000000	6018																		
ASWREG= 000000	6018																		
ATESTN= 000000	6018																		
AUNIT = 000000	6018																		
AUSWR = 000000	6018																		
AVECT1= 000000	6018																		
AVECT2= 000000	6018																		
BAD 004552	6348#	6614	7580*	7648*	7716*	7784*	7946*	7950*	7959	7961	8017*	8021*	8030						

	8032	8088*	8092*	8101	8103	8158*	8162*	8171	8173	9084*	9107*	9115	9117
	9270*	9308*	9346*	9386*	9426*	9480*	9553*	9610*	9723*	9901*	9902	9954*	10081*
	10082	10134*	10201*	10206*	10283*	10288*	10359*	10362*	10442*	10445*	10509*	10537*	10591*
BADST = 006434	6611	6648#											
BEBA = 170004	6076#	9762*											
BECC = 170002	6075#	9763*											
BE CR1 = 170006	6078#	9745	9766*	9767									
BE CR2 = 170016	6079#	9765*											
BEDA = 170000	6077#	9764*											
BIT = 006556	6656	6678#											
BITFLG = 004560	6351#	6500*	6513	7464*	7512*	7981*	8052*	8123*	8193*	8509*	9063*	9129*	
BITNAM = 006454	6515	6654#											
BIT00 = 000001	6040#	6880	6883	7352	7354	7370	7373	8448	10670	10786			
BIT01 = 000002	6041#	7160	7162	7303	7304	7306	7391	7393					
BIT02 = 000004	6042#	6903	6906	6925	6928	7327	7328	7330	7331	7410	7412	7494	8222
	8275	8448	8903	9161	9197	9228	9256	9294	9333	9372	9411	9466	9538
	9595												
BIT03 = 000010	6043#	6954	6957	6978	6981	7447	7541	7608	7675	7743	7948	8019	8090
	8160	8332	8386	8874	8933	8943	8974	8986	9039	9105	10334	10337	10416
	10419	10495	10500	10525	10530	10577	10582	10607	10612				
BIT04 = 000020	6044#	7180	7182	7202									
BIT05 = 000040	6045#	7200	7552	7619	7688	7756	8480	8615	8794	8848	8865	8873	9677
	9838	10017											
BIT06 = 000100	6046#	7000	7003	7021	7028	7557	7559	7561	7564	7568	7624	7626	7629
	7632	7635	7693	7695	7697	7700	7703	7761	7763	7765	7768	7771	8466
	8562	8645	8647	8657	8687	8689	8780	8848	8849	8873	9685	9689	9691
	9696	9844	9846	9848	9851	9877	9879	9881	9885	9887	9932	9934	9936
	9939	10023	10025	10027	10030	10056	10058	10060	10064	10066	10112	10114	10116
	10119	10211	10293	10661	10700	10737	10773						
BIT07 = 000200	6047#	7047	7050	7068	7071	7549	7551	7562	7630	7685	7687	7698	7766
	8473	8570	8699	8731	8776	8787	8840	8848	8857	9654	9656	9694	9849
	9882	9937	10028	10061	10117	10213	10295	10707	10710	10744	10748		
BIT08 = 000400	6048#	6098	6197	6200	6377	6394	6879	7089	7220	7222	7802	7830	7856
	7861	7892	7911	7929	7937	7940	8000	8008	8011	8071	8079	8082	8141
	8149	8152	8438	9019	9085	9149	9185	10391	10475				
BIT09 = 001000	6049#	6102	6166	6168	7086	7107	7110	7306	8222	8275	8332	8386	8733
	8874	9056	9122	9154	9156	9190	9192	9221	9223				
BIT10 = 002000	6050#	7124	7126	7139	7142	7143	7555	7568	7622	7635	7691	7703	7759
	7771	8603	8605	9680	9682	9841	9843	10020	10022				
BIT11 = 004000	6051#	7240	7242										
BIT12 = 010000	6052#	7803	7831	7857	7863	7893	7913	7930	7938	7941	8001	8009	8012
	8072	8080	8083	8142	8150	8153	8441	9020	9086	9150	9186		
BIT13 = 020000	6053#	6174	6176	6496	7853	7868	7897	7909	7927	7998	8069	8139	9022
	9088	9147	9183	10370	10453								
BIT14 = 040000	6054#	6099	6162	6164	7260	7262							
BIT15 = 100000	5996	6055#	6170	6172	6519	7280	7282	9668					
BUFPNT = 004424	6005*	6012*	6289*	6290*	6291	6294	6302	6304*	6312	6314	6316	6318*	6320
	6322*	6325*	6326*	6338	6340#								
CCR = 177746	5983*	6024#	6098*	6359*	6701*	6724*	6726	6744*	6764*	6784*	6805*	6830*	6854*
	6855*	6856	6879*	6880*	6881	6882*	6903*	6904	6905*	6925*	6926	6927*	6954*
	6955	6956*	6978*	6979	6980*	7000*	7001	7002*	7021*	7022	7023*	7047*	7048
	7049*	7068*	7069	7070*	7086*	7087	7088*	7107*	7108	7109*	7124*	7125	7139*
	7140	7141*	7159*	7160*	7161	7179*	7180*	7181	7199*	7200*	7201	7219*	7220*
	7221	7239*	7240*	7241	7259*	7260*	7261	7279*	7280*	7281	7302*	7303*	7304*
	7305	7326*	7327*	7328*	7329	7351*	7436*	7453*	7483*	7501*	7535*	7547*	7549*

7551*	7555*	7559*	7561*	7564*	7568*	7576*	7602*	7614*	7616*	7618*	7622*	7626*	
7629*	7632*	7635*	7644*	7669*	7683*	7685*	7687*	7691*	7695*	7697*	7700*	7703*	
7712*	7737*	7751*	7753*	7755*	7759*	7763*	7765*	7768*	7771*	7780*	7802*	7803	
7830*	7831	7853	7856*	7857	7861*	7863	7867	7892*	7893	7897	7909	7911*	
7913	7927	7929*	7930	7932*	7937*	7938	7940*	7941	7958*	7998	8000*	8001	
8003*	8008*	8009	8011*	8012	8029*	8069	8071*	8072	8074*	8079*	8080	8082*	
8083	8100*	8139	8141*	8142*	8144*	8149*	8150	8152*	8153	8170*	8222*	8233*	
8275*	8287*	8332*	8343*	8386*	8397*	8438*	8441	8444*	8448*	8452*	8455*	8465*	
8528*	8537*	8552*	8559*	8561*	8600*	8603*	8605*	8607*	8609*	8642*	8645*	8647*	
8649*	8651*	8684*	8687*	8689*	8691*	8693*	8724*	8726*	8728*	8731*	8733*	8738*	
8761*	8767*	8769*	8774*	8776*	8777*	8820*	8822*	8825*	8828*	8830*	8832*	8835*	
8838*	8840*	8873*	8874*	8893*	8908*	8929*	8935*	8945*	8951*	8970*	8978*	8989*	
8997*	9019*	9020	9022	9024*	9048*	9085*	9086	9088	9090*	9114*	9147	9149*	
9150	9152*	9154*	9156*	9159*	9173*	9183	9185*	9186	9188*	9190*	9192*	9195*	
9209*	9219*	9221*	9223*	9226*	9239*	9250*	9255*	9288*	9293*	9327*	9332*	9366*	
9371*	9405*	9410*	9460*	9465*	9532*	9537*	9589*	9594*	9643*	9654*	9656*	9658*	
9662*	9680*	9682*	9685*	9687*	9691*	9696*	9705*	9710*	9753*	9775*	9811*	9824*	
9826*	9841*	9843*	9846*	9848*	9851*	9853*	9858*	9874*	9876*	9879*	9881*	9885*	
9887*	9888*	9893*	9896*	9929*	9931*	9934*	9936*	9939*	9941*	9946*	9949*	9990*	
10003*	10005*	10020*	10022*	10025*	10027*	10030*	10032*	10037*	10053*	10055*	10058*	10060*	
10064*	10066*	10067*	10072*	10075*	10109*	10111*	10114*	10116*	10119*	10121*	10126*	10129*	
10160*	10178*	10183*	10188*	10196*	10198*	10242*	10260*	10265*	10270*	10278*	10280*	10332*	
10343	10347	10351*	10384*	10391*	10414*	10416	10425	10430	10434*	10468*	10475*	10488*	
10508*	10520*	10535*	10546*	10549*	10570*	10590*	10602*	10617*	10628*	10631*	10658*	10661*	
10663*	10668*	10670*	10671	10672*	10697*	10700*	10702*	10707*	10709	10711*	10734*	10737*	
10739*	10744*	10746	10747*	10770*	10773*	10775*	10782*	10784	10785*				
CHR = 177752	6026#	6766	6807	7447	7541	7608	7675	7743	7948	8019	8090	8160	8507
	8534	8902	8933	8943	8974	8986	9039	9105	9158	9194	9225	9254	9292
	9331	9370	9409	9464	9536	9593	9771	10334	10337	10419	10495	10500	10525
	10530	10577	10582	10607	10612								
CLOCK 004674	6363#												
CMDLST 005224	6413	6440#											
CMDLS1 005316	6412	6462#											
CMPE = 177744	6023#	6360	6361*	6706	6786	6832*	6833	7494	7539*	7550	7606*	7617	7673*
	7686	7741*	7754	8449*	8453	8454*	8464*	8557*	8560	8599*	8608	8641*	8650
	8683*	8692	8723*	8730*	8732	8762*	8768	8772*	8775	8819*	8826*	8829	8836*
	8839	8846*	9648*	9652*	9655	9706*	9822*	9825	9865*	9867	9875	9924	9930
	10001*	10004	10044*	10046	10054	10104	10110	10167*	10169	10174	10197	10249*	10251
	10256	10279	10657*	10696*	10733*	10781*							
CMR = 177750	6025#	6746	7352*	7353	7370*	7371	7372*	7391*	7392	7410*	7411	8969*	8977*
	8988*	8996*	10497*	10502*	10507*	10523*	10527*	10579*	10584*	10589*	10605*	10609*	
CONWRD 000176	5979#	5990*	5996*	6094*	6099	6102	6162*	6164*	6166*	6168*	6170*	6172*	6174*
	6176*	6197*	6198*	6200*	6206*	6208*	6377	6379	6394	6396	6496	6519	8238
	8293	8348	8402										
COUNT 004704	6366#	8217*	8270*	8327*	8381*								
CRLF 005634	6110	6327	6535	6544#	6617								
DEAD 035676	10807	10817#											
DECODE 005066	6329	6410#											
DIC1 042210	10822	10854	10867	10870	10874	10875	10877	10878	10880	10887	10888	10890	10891
	10892	10893	10895	10896	10897	10898	10905	10909	10910	10935	10937	10939	10958
	10959	10960	10962	10965	10968	10972	10974	10976	10977	10978	10980	10981	10983
	10985	10987	10988	10992#									
DIC10 042307	10823	10824	10825	10826	10827	10828	10882	11001#					
DIC100 043313	10895	10896	10897	10898	10937	10986	11092#						
DIC101 043323	10899	10900	10926	10948	11093#								
DIC102 043331	10900	10923	10927	10951	10960	10986	10987	11094#					

CACHE DIAG.
CFKKAB.P11

MACY11 30A(1052)
25-JUN-79 13:31

31-OCT-79

15:29 PAGE 77-3

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0136

DIC103	043335	10900	10903	10915	10925	10945	10948	10975	10976	10978	11095#
DIC104	043341	10905	11096#								
DIC105	043347	10907	11097#								
DIC106	043357	10908	10965	10968	10972	10974	10977	11098#			
DIC107	043361	10908	11099#								
DIC108	043365	10911	10912	11100#							
DIC109	043372	10914	10918	11101#							
DIC11	042313	10824	10832	10846	10854	11002#					
DIC110	043376	10921	10981	10983	10985	11102#					
DIC111	043404	10923	10927	11103#							
DIC112	043410	10923	10927	11104#							
DIC113	043415	10929	11105#								
DIC114	043423	10929	11106#								
DIC115	043427	10929	10932	10943	10962	11107#					
DIC116	043436	10929	11108#								
DIC117	043450	10930	10931	10932	11109#						
DIC118	043454	10930	10931	10932	11110#						
DIC119	043461	10932	10939	11111#							
DIC12	042323	10825	10861	10933	11003#						
DIC120	043475	10932	10939	10940	10943	10944	10946	10960	11112#		
DIC121	043506	10933	11113#								
DIC122	043514	10933	10938	10940	10943	11114#					
DIC123	043521	10934	11115#								
DIC124	043531	10935	11116#								
DIC125	043543	10935	10941	11117#							
DIC126	043550	10938	10940	10943	11118#						
DIC127	043557	10940	10962	11119#							
DIC128	043562	10941	11120#								
DIC129	043566	10945	11121#								
DIC13	042337	10826	10904	10905	10906	10907	10908	10928	10931	10946	10948
DIC130	043574	10946	10949	10951	10958	10960	11122#				10951
DIC131	043603	10947	11123#								
DIC132	054000	10950	11128#								
DIC133	054007	10952	11129#								
DIC134	054016	10953	11130#								
DIC135	054025	10954	11131#								
DIC136	054034	10955	11132#								
DIC137	054043	10956	11133#								
DIC138	054052	10957	11134#								
DIC139	054061	10959	11135#								
DIC14	042343	10827	10828	11005#							
DIC140	054071	10961	10963	11136#							
DIC141	054100	10961	10963	11137#							
DIC142	054112	10961	10962	11138#							
DIC143	054116	10963	11139#								
DIC144	054125	10964	10966	10970	10975	11140#					
DIC145	054133	10964	10966	10970	10975	11141#					
DIC146	054143	10965	11142#								
DIC147	054154	10968	10972	11143#							
DIC148	054164	10974	11144#								
DIC149	054173	10977	11145#								
DIC15	042353	10827	10828	10936	10947	10950	10951	10952	10953	10954	10955
		10959	10965	10968	10972	10974	10977	11006#			10956
DIC150	054203	10979	10980	10981	11146#						10957
DIC151	054215	10982	11147#								10958
DIC152	054222	10983	10985	10986	11148#						

GOODST	006413	6606	6647#												
HELP	002152	6202#	6473												
HIGHSP	046000	6586	6592	6913	6914	7435	7438	7442	7451	7482	7485	7489	7499	7572	
		7588	7589	7604	7605*	7607	7623	7627*	7634*	7637	7724	7725	7739	7740*	
		7742	7760	7764*	7770*	7773	7970	7971	8004	8006	8015	8023	8112	8113	
		8145	8147	8156	8164	8255	8256	8312	8313	8383	8443	8446	8456	8529	
		8530	8553	8554*	8555*	8556*	8558	8602	8644	8646*	8648	8686	8688*	8690	
		8727	8729	8764	8765*	8766	8823	8824*	8827	8831	8915	8916	8930	8942	
		8971	8985	9059	9082	9095	9101	9103	9108	9112	9153	9155*	9157	9189	
		9191*	9193	9220	9222	9224	9265	9266	9303	9304	9342	9343	9381	9382	
		9420	9422	9475	9476	9547	9548	9604	9605	9638	9641	9644	9646	9650	
		9663	9667	9674	9704	9708	9758	9760	9762	9769	9772	9856	9891	9944	
		9966	9967	9995	9997	9999	10006	10008	10039	10050	10095	10099	10101	10106	
		10181	10231	10232	10241	10243	10245	10247	10266	10268	10275	10387	10388	10408	
		10410	10413	10422	10449	10552	10553	10660	10662*	10669	10699	10701*	10708	10742	
		10756	10757	10778	10792	10793	11164#								
HIGH1	052000	7603	7621	7738	7758	7943	7973	8085	8115	8272	8601	8604	8606	8643	
		8685	8725	8763	8770	8771*	8773	8821	8833	8834*	8837	8841*	9017*	9018*	
		9041*	9045*	9049	9051	9053	9754	9756	9821	9830	9861	9897	9900	9905	
		9922	9950	9956	9962	9991	9993	10080	10136	10166	10187	10330	10352	10355	
		10366	10659	10698	11168#										
HLONER	001154	6107	6115#												
HL1	002004	6170#	6467												
HL2	002014	6172#	6468												
ILL	005174	6432	6437#												
INBUF	004430	6005	6012	6302	6320	6326	6342#	6414							
ITTRAP	004540	6343#	6357*	6705*	6707	6725*	6727	6745*	6747	6765*	6767	6785*	6787	6806*	
		6808	6831*	9453*	9455	9524*	9526	9581*	9583	9744*	9746				
		6056#	8221*	8223	8226*	8274*	8276	8279*	8331*	8333	8336*	8385*	8387	8390*	
KOOKOO=	177546	6066#	9501*												
KPAR0 =	172340	6067#	9502*												
KPAR1 =	172342	6068#	9503*												
KPAR2 =	172344	6069#	9504*												
KPAR3 =	172346	6070#	9523*	9579*											
KPAR4 =	172350	6071#													
KPAR5 =	172352	6072#													
KPAR6 =	172354	6073#	9505*												
KPAR7 =	172356	6058#	9493*												
KPDR0 =	172300	6059#	9494*												
KPDR1 =	172302	6060#	9495*												
KPDR2 =	172304	6061#	9496*												
KPDR3 =	172306	6062#	9497*												
KPDR4 =	172310	6063#	9498*												
KPDR5 =	172312	6064#	9499*												
KPDR6 =	172314	6065#	9500*												
KPDR7 =	172316	6028#	6289												
KRB =	177562	6027#	6279												
KRS =	177560	6368#	6484*	6485*	6498	6508									
LAST	004710	6517	6525#												
LFLF	005552	6531	6537	6554#	6600	6619	6663	6669							
LINE	005657	6004	6121#												
LINFRQ	001176	6372*	6399*	6403	6407#	7812*	7844*	7881*	9015*	9081*	9810*	9913*	9961*	9964	
LOPERR	005064	9989*	10093*	10141*	10144	10157*	10220*	10227*	10229	10239*	10302*	10309*	10311	10324*	
		10369*	10385	10406*	10452*	10469	10486*	10519*	10547*	10550	10568*	10601*	10629*	10632	
LOST	002210	6202	6211#												
LOWSP	044000	6084	6574	6580	6964	6965	7024	7471	7472	7519	7520	7537	7538*	7540	

CACHE DIAG.
CFKKAB.P11

MACY11 30A(1052)
25-JUN-79 13:31

31-OCT-79 15:29 PAGE 77-8

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0141

		7556	7560*	7566*	7569	7640	7656	7657	7671	7672*	7674	7692	7696*	7702*
		7705	7708	7776	7792	7793	7933	7935	7944	7952	8041	8042	8075	8077
		8086	8094	8182	8183	8219	8365	8366	8419	8420	8460	8488	8489	8500
		8515	8516	8545	8546	8586	8587	8613	8625	8626	8655	8667	8668	8697
		8709	8710	8736	8744	8745	8802	8803	8844	8876	8877	8953	8954	8956
		8999	9000	9002	9016	9029	9035	9037	9042	9046	9125	9171	9172	9207
		9208	9237	9238	9251	9289	9328	9367	9461	9533	9590	9657	9660	9701
		9732	9733	9783	9784	9816	9818	9820	9827	9829	9835	9860	9871	9915
		9919	9921	9926	10035	10070	10124	10146	10147	10159	10161	10163	10165	10184
		10186	10193	10263	10313	10314	10326	10328	10331	10340	10471	10472	10634	10635
		10666	10682	10683	10705	10719	10720	10736	10738*	10745	10772	10774*	10783	11162#
LOW1	050000	6086	7026	7536	7554	7670	7690	8014	8044	8155	8185	8329	8892	8895
		8897	8899	8906	9649	9665	9703	9711	9717	9812	9814	10000	10009	10014
		10040	10076	10085	10102	10130	10142	10248	10269	10412	10435	10438	10735	10771
		11166#												
LPONTS	005012	6394#	8955	9001	9058	9124								
LP1	001744	6162#	6463											
LP2	001754	6164#	6464											
LP3	001764	6166#	6465											
LP4	001774	6168#	6466											
LP5	002044	6178#	6471											
LP6	002142	6200#	6472											
MAGPRE	025564	9452	9457	9459	9477	9493#								
MAYBE	007726	6939#												
NAME	001446	6148	6151#											
NOINC	004712	6369#												
NOUBE	004562	6352#	9751											
OCTASC	006310	6605	6610	6615	6624#									
OFF =	001015	5983	6057#	6098	6359	6701	6724	6744	6764	6784	6805	6830	6854	6879
		6882	6905	6927	6956	6980	7002	7023	7049	7070	7088	7109	7141	7159
		7179	7199	7219	7239	7259	7279	7302	7326	7351	7453	7501	7547	7576
		7614	7644	7683	7712	7751	7780	7958	8029	8100	8170	8233	8287	8343
		8397	8438	8465	8537	8561	8609	8651	8693	8731	8733	8738	8777	8835
		8908	8935	8945	8951	8978	8989	8997	9048	9114	9159	9173	9195	9209
		9226	9239	9255	9293	9332	9371	9410	9465	9537	9594	9658	9710	9775
		9853	9888	9896	9941	9949	10032	10067	10075	10121	10129	10178	10188	10260
		10270	10351	10384	10434	10468	10508	10535	10549	10590	10617	10631	10663	10672
		10702	10711	10739	10747	10775	10785							
PARITY	004650	5976	6096	6359#										
PNTNAM	001364	5997	6011	6135#										
PPB =	177566	6030#	6338*	6564*										
PPS =	177564	6029#	6336	6562										
PREPAR	001000	5998	6084#	6462	10816									
PRINT	005564	6507	6530#											
PROM	001730	6145	6159#	6423										
PSW =	177776	6031#	8931	8932	8972	8973	10656*							
PTID	004010	6109	6250#	6504										
QUESHZ	001210	6121	6124#											
RELCTH	006142	6586#	6901	7532	7666	7924	8066	8215	8268	8889	9012	9248	9286	9325
		9364	9403	9458	9530	9587	9807	10155	10321	10484	10731	10768		
RELCTL	006114	6574#	6952	7433	7480	7599	7734	7995	8136	8325	8379	8435	8496	8526
		8550	8597	8639	8681	8721	8759	8817	8925	8964	9077	9145	9181	9217
		9635	9742	9986	10237	10403	10566	10654	10694					
REST	035672	10810	10816#											
SAVTAT	004546	6346#												
SENT	035722	6710	6730	6750	6770	6791	6813	6837	6861	10822#				

TST035	011460	7301#							
TST036	011554	7325#							
TST037	011650	7350#							
TST040	011724	7369#							
TST041	012002	7390#							
TST042	012050	7409#							
TST043	012116	7432#							
TST044	012310	7434	7479#						
TST045	012502	7481	7530#	7572	7588	7589			
TST046	013034	7533	7548	7597#	7640	7656	7657		
TST047	013366	7600	7615	7664#	7708	7724	7725		
TST050	013724	7667	7684	7732#	7776	7792	7793		
TST051	014262	7735	7752	7801#					
TST052	014356	7813	7827#						
TST053	014446	7851#							
TST054	014616	7874	7889#						
TST055	014746	7908	7923#						
TST056	015302	7925	7960	7972	7994#				
TST057	015636	7996	8031	8043	8065#				
TST060	016170	8067	8102	8114	8135#				
TST061	016524	8137	8172	8184	8214#	8255	8256		
TST062	016712	8216	8267#	8312	8313				
TST063	017100	8269	8324#	8365	8366				
TST064	017266	8326	8378#	8419	8420				
TST065	017454	8380	8434#	8488	8489				
TST066	017750	8436	8495#	8500					
TST067	020046	8497	8525#						
TST070	020160	8527	8549#						
TST071	020374	8551	8596#						
TST072	020560	8598	8627	8638#					
TST073	020746	8640	8669	8680#					
TST074	021134	8682	8711	8720#					
TST075	021276	8722	8758#						
TST076	021546	8760	8816#						
TST077	022122	8818	8888#						
TST100	022262	8890	8924#	8953	8954	8956			
TST101	022452	8926	8963#	8999	9000	9002			
TST102	022670	8965	9011#						
TST103	023170	9013	9060	9076#					
TST104	023470	9078	9126	9144#					
TST105	023650	9146	9180#						
TST106	024030	9182	9216#						
TST107	024162	9218	9247#						
TST110	024346	9249	9267	9285#					
TST111	024540	9287	9305	9324#					
TST112	024732	9326	9344	9363#					
TST113	025124	9365	9383	9402#					
TST114	025316	9404	9423	9443#					
TST115	025702	9514#							
TST116	026166	9521	9529	9531	9550	9571#			
TST117	026452	9578	9586	9588	9607	9634#	9657	9660	9732 9733
TST120	027204	9636	9741#						
TST121	027442	9743	9748	9806#	9856	9891	9944		
TST122	030564	9808	9985#	10035	10070	10124			
TST123	031706	9987	10154#	10181					
TST124	032324	10156	10236#	10263					

COMMEN	1528#			
ENDCOM	1540#			
ESCAPE	1656#			
GETPRI	1270#			
GETSWR	1727#			
MULT	4387#			
NEWST	1587#			
POP	2105#			
PUSH	2097#			
REPORT	5346#			
SETPRI	1238#			
SETUP	1304#			
SKIP	1690#			
SLASH	1480#			
STARS	1449#	5951#	6016	6018
SWRSU	1418#			
TYPBIN	2041#			
TYPDEC	2011#			
TYPNAM	1781#			
TYPNUM	1978#			
TYPOCS	1931#			
TYPOCT	1894#			
TYPTXT	1848#			
\$\$ESCA	1669#			
\$\$NEWT	1623#			
\$\$SKIP	1703#			
.EQUAT	168#			
.HEADE	42#			
.KT11	311#			
.SETUP	1172#			
.SWRHI	84#			
.SACT1	4955#			
.SAPT8	4999#	5951#	6018	
.SAPTH	5255#	5951#	6016	
.SAPTY	5430#			
.SASTA	5301#			
.SCATC	897#			
.SCMTA	1008#			
.SDB2D	4585#			
.SDB20	4708#			
.SDIV	4488#			
.SEOP	2164#			
.SERRO	2644#			
.SERRT	2839#			
.SMULT	4425#			
.SPOWE	4137#			
.SRAND	4212#			
.SRDDE	3815#			
.SRDOC	3724#			
.SREAD	3329#			
.SR2AZ	4852#			
.\$SAVE	3890#			
.\$SB2D	4669#			
.\$SB20	4770#			
.\$SCOP	2398#			
.\$SIZE	4265#			

.\$SUFR 4808#
.\$STRAP 3992#
.\$STYPB 3222#
.\$STYPD 3145#
.\$STYPE 2926#
.\$STYPO 3049#
.\$4OCA 936#
.\$1170 490#

. ABS. 054346 000

ERRORS DETECTED: 0

.\$SKZ:CFKKAB.SEQ/CRF/NL:TOC=DSKZ:CFKKAB.SML,DSKZ:CFKKAB.P11
RUN-TIME: 41 62 6 SECONDS
RUN-TIME RATIO: 985/110=8.9
CORE USED: 33K (65 PAGES)