

.TITLE CZMSPAO MS11-L/M/P MEMORY DIAG.

IDENTIFICATION

PRODUCT CODE:

AC-T156A-MC

PRODUCT NAME:

CZMSPAO MS11-L/M/P MEMORY DIAG

PRODUCT DATE:

MAY 1982

MAINTAINER:

STORAGE SYSTEMS S/W TEST APPLICATIONS

COPYRIGHT(C):

1982

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

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 2
TABLE OF CONTENTS

110- 4559 DEFINE TRAPS
111- 4675 DEFINE BASIC PDP11 STUFF
```

```
110- 4559
111- 4675
111- 4758
                                   DEFINE
                                                        CACHE REGISTERS
111- 4768

113- 4898

113- 4966

113- 4970

113- 4980

115- 5068

115- 5068

115- 5068

119- 5068

119- 5163

121- 5163

125- 5261

126- 5366

130- 5366

137- 5508

137- 5508

137- 5508

137- 5508

137- 5508

137- 5508

137- 5508

137- 5508

137- 5508

137- 5508

137- 5508

137- 5508

137- 5883

147- 5883

147- 5883

147- 5893

147- 5893

151- 6050

155- 6050

155- 6050

155- 6063

155- 6063

155- 6063

155- 6063

156- 6731

160- 6856
                                   DEFINE
                                                        CPU REGISTERS
                                   DEFINE
                                                        MEMORY MANAGEMENT REGISTERS
                                  DEFINE
                                                        UNIBUS MAP REGISTERS
                                  DEFINE
DEFINE
DEFINE
                                                        SOFTWARE SWITCH & DISPLAY REGISTERS CONTROL STATUS REGISTERS
                                                        PARAMETERS
                                   MACRO
                                                        FATAL
                                   MACRO
                                                        TYPE
                                   MACRO
                                                        NEWTST
                                   MACRO
                                                        $$NEWTEST
                                   MACRO
                                                        SUBTST
                                                        $SUBTST
                                   MACRO
                                   MACRO
MACRO
                                                        TYPOCT
                                                        TYPOCS
                                   MACRO
                                                        TYPDEC
                                   MACRO
                                                        BMOV
                                   MACRO
                                                        MAP
                                   MACRO
                                                        SUPERVISOR
                                   MACRO
                                                        USER
                                   MACRO
                                                        TESTAREA
                                   MACRO
                                                        SET4 & RES4
                                   MACRO
                                                        DLEFT
                                   TRAP CATCHER
                                  ACT11 HOOKS
APT11 HOOKS
                                                                             INITIALIZED TO ZERO INITIALIZED TO NON ZERO
                                   VARIABLES
                                   VARIABLES
                                   CONFIGURATION TABLE
                                 INITIALIZE VARIABLES TO ZERO
CLEAR NON-PROGRAM SPACE
TYPE OF SYSTEM SIZER
INITIALIZE VARIABLES TO NON ZERO
INITIALIZE VECTORS
INITIALIZE PATTERNS
SUBR PLUG IN NULL PATTERNS
CLEAR THE CONFIGURATION TABLE
SIZE FOR A HARDWARE SWITCH REGISTER
SETUP ACT, APT, & XXDP
PROTECT PROGRAM & LOADERS
CHECK SYSTEM FOR CACHE
SETUP USER & SUPERVISOR STACK
GET SOFTWARE SWITCH REGISTER IF NECESSARY
GET MEMORY MANAGEMENT READY
                                   ****************** MAIN *************
                                   GET MEMORY MANAGEMENT READY
                                  TI BIT TEST OF ALL CSR'S DETERMINE TYPE OF ECC MEMORY PRINT CSR REGISTER MAP
                                 READ AND WRITE ALL CSR BITS
CLEAR ALL MEMORY SPACE FROM BANK 2 ON
MATCH ALL CSR'S WITH MEMORY
T2 TEST BANK 0 ACCESSES
ENABLE ECC FOR CORRECT TRAPS
T3 TEST BANKS 1-200 (OCTAL) FOR ZEROS & ONES
FIND SHADOW INHIBIT MODE POINTERS
```

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 2-1 TABLE OF CONTENTS

```
172- 6879
182- 7036
184- 7150
186- 7207
187- 7242
187- 7259
187- 7259
187- 7353
193- 7355
193- 7355
193- 7355
193- 7452
201- 7489
203- 7572
203- 7610
203- 7675
213- 7666
215- 7861
217- 7907
219- 8012
220- 8012
220- 8012
220- 8012
220- 8012
220- 8012
220- 8012
220- 8025
221- 7676
215- 7861
217- 7907
219- 8025
221- 8151
222- 8151
223- 8151
224- 8158
224- 8158
224- 8158
224- 8158
224- 8158
224- 8158
224- 8251
223- 8251
223- 8251
223- 8251
223- 8251
223- 8251
223- 8251
223- 8251
223- 8251
224- 8574
224- 8574
234- 8574
234- 8574
234- 8574
234- 8574
234- 8574
234- 8574
                                                                                                                                      ECC INHIBIT MODE POINTER TEST
                                                                                   LEGAL CONFIGURATION CHECK
                                                                                   PRINT CONFIGURATION DETAILS
                                                                                     CHECK APT SIZING
                                                                                 TS DIAGNOSTIC MODE DISPATCH ROUTINE
TO UNIQUE BANK TEST
FLUSH OUT DBE'S
                                                                                   END OF PASS ROUTINE
                                                                                   WRITE BACKGROUND PATTERNS
MTEST MODES
                                                                               BANKS FORWARD, PATTERNS FORWARD

BANKS FORWARD, PATTERNS REVERSE

BANKS WORST FIRST, PATTERNS FORWARD

BANKS WORST FIRST, PATTERNS REVERSE

PATTERNS FORWARD, BANKS FORWARD

PATTERNS FORWARD, BANKS WORST FIRST

PATTERNS REVERSE, BANKS WORST FIRST

PATTERNS REVERSE, BANKS WORST FIRST

SUBR SETUP MEMORY TEST

SUBR TEST ECC CSR LOGIC DISPATCH

CHECK FOR SBE FREE LOCATIONS

CSR PATTERN CASE STATEMENT

SUBR ECC TEST DISPATCH

SUBR PARITY TEST DISPATCH

PATTERNS
                                                                                   PATTERNS
                                                                                   MEMORY TEST SETUP ROUTINES
MT0000 SETUP DATA PATTERN TEST
                                                                                                                                SETUP DATA PATTERN TEST
SETUP ADDRESS TEST
SETUP COMPLEMENT ADDRESS TEST
SETUP 3 XOR 9 WORST CASE NOISE TEST
SETUP ROTATING ZEROS TEST
SETUP ROTATING ONES TEST
SETUP HITTIAL DATA TEST
SETUP ADDRESS BIT TEST
SETUP ADDRESS BIT TEST
SETUP WRITE BYTE CLEARS SBE TEST
SETUP WRITE BYTE CLEARS SBE TEST
SETUP WRITE BYTE CLEARS SBE TEST
SETUP WRITE INHIBIT OF BYTE WITH DBE
SETUP WRITE INHIBIT OF BYTE WITH DBE
SETUP WRITE INHIBIT OF WORD WITH DBE
SETUP HOLDING 1'S & 0'S
SETUP SYNDROMES TO CSR ON SINGLE BIT ERROR
SETUP MARCHING 0'S & 1'S TEST
SETUP MARCHING 0'S & 1'S TEST
SETUP FAST GALLOPING PATTERN TEST
SETUP FAST GALLOPING PATTERN TEST
SETUP FAST GALLOPING PATTERN TEST
SETUP FANDOM DATA TEST
UNIQUE BANK TEST
SETUP FLUSH OUT DBE'S TEST
SETUP SOB-A-LONG TEST
SETUP SOB-A-LONG TEST
                                                                                 MT0001
MT0002
MT0003
MT0004
MT0005
MT0006
MT0007
MT0010
MT0011
MT0012
MT0013
MT0014
MT0015
                                                                                  MT0016
MT0017
MT0020
MT0021
MT0022
MT0023
MT0024
MT0025
MT0027
MT0030
MT0031
MT0032
MT0033
                                                                                                                                    SETUP SOB-A-LONG TEST
SETUP WRITE RECOVERY TEST
SETUP BRANCH GOBBLE TEST
SOFT ERROR - BACKGROUND PATTERN TEST
SETUP WORST CASE NOISE PARITY TEST
SETUP CORRECTION CODE TEST
                                                                                  MT0034
MT0035
MT0036
```

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 2-2 TABLE OF CONTENTS

```
SETUP ECC DISABLE TEST
248- 8731

248- 8743

249- 8746

250- 8755

251- 8765

251- 8781

251- 8789

251- 8789

251- 8789

251- 8892

253- 8813

255- 8892

259- 8902

259- 8902

259- 8902

261- 8939

261- 8962

263- 8988

265- 9011

265- 9020

267- 9034

265- 9034

267- 9117

273- 9153

277- 9376

282- 9467

283- 9515

283- 9515

285- 9614

291- 9750

292- 9888

297- 9888

297- 9982

301-10026
                                                                MT0037
                                                               MT0040
MT0041
MT0042
MT0043
MT0044
MT0045
MT0046
MT0047
MT0999
                                                                                                        SETUP ADDRESS TO CSR ON DOUBLE BIT ERROR TEST SETUP EXTENDED UNIBUS ADDRESS TO CSR TEST
                                                                                                      SETUP WRITE BYTE CLEARS SBE TEST
SETUP SHIFTING 1/0'S THROUGH THE CHECK BITS TEST
SETUP SYNDROMES TO CSR ON DOUBLE BIT ERROR
SETUP CHECK SINGLE BIT ERRORS WITH ECC DISABLED TET
SETUP NO CSR UPDATE ON SBE WITH EXSISTING DBE TEST
                                                             MT0047 SETUP NO CSR UPDATE ON SBE WITH EXSISTING DBE MT0999 SETUP NULL TEST CHECK FOR KAMIKAZE MODE SUBR EXECUTE PATTERN IN SUPERVISOR MEMORY TEST PATTERN ROUTINES MTP000 BASIC DATA TEST MTP001 ADDRESS TEST (WRITE DOWN, READ UP) MTP002 COMPLEMENT ADDRESS TEST (WRITE DOWN, READ UP) MTP003 3 XOR 9 WORST CASE NOISE TEST (WRITE) MTPB03 3 XOR 9 WORST CASE NOISE TEST (READ) MTPC03 TEST DATA SUBPROGRAM MTP003 TEST DATA SUBPROGRAM MTP004 ROTATING ZEROS TEST MTP004 ROTATING ZEROS TEST MTP005 ROTATION ONES TEST MTP006 INITIAL DATA TEST MTP007 ADDRESS BIT TEST MTP007 ADDRESS BIT TEST MTP010 BYTE ADDRESSING TEST
                                                                                                     SUBR ROTATING BIT
ROTATION ONES TEST
INITIAL DATA TEST
ADDRESS BIT TEST
BYTE ADDRESSING TEST
SINGLE BIT ERROR TEST
WRITE BYTE CLEARS SBE TEST
CREATE DOUBLE BIT ERROR TEST
BASIC DOUBLE BIT ERROR TEST
WRITE INHIBIT OF BYTE WITH DBE
WRITE INHIBIT OF WORD WITH DBE
HOLDING 1'S & O'S TEST
SYNDROMES TO CSR ON SINGLE BIT ERROR TEST
MARCHING 1'S & O'S PATTERN TEST
REFRESH & SHIFTING DIAGONAL TEST
REFRESH DELAY
FAST GALLOPING PATTERN TEST
                                                                   MTP010
                                                                MTP010
MTP011
MTP012
MTP013
MTP014
MTP015
MTP016
MTP017
MTP020
MTPA21
MTP022
SURP
                                                              MTP022 REFRESH & SHIFTING DIA

SUBR REFRESH DELAY

MTPA24 FAST GALLOPING PATTERN

MTPB24 FAST GALLOP PART B

MTPC24 FAST GALLOP PART C

MTP025 INTERRUPT ENABLE TEST

MTPA26 RANDOM DATA (WRITE)

MTPB26 RANDOM DATA (READ)

RANDOM NUMBER SUBPROGRAM

RANDOM NUMBER SUBSUBPROGRAM

MTP031 SOB-A-LONG TEST

MTP031 SOB-A-LONG TEST

MTP032 WRITE RECOVERY TEST

MTP033 BRANCH GOBBLE TEST

MTP034 SOFT ERROR - BACKROUND

MTP035 WORST CASE NOISE PARIT

MTP036 CORRECTION CODE TEST

MTP037 CHECK ECC DISABLE TEST

MTP031 ADDRESS TO CSR ON DOUE
                                                                                                        FAST GALLOPING PATTERN TEST
FAST GALLOP PART B
FAST GALLOP PART C
      301-10034
    303-10034
303-10044
307-10138
307-10163
      307-10176
    309-10184
309-10190
311-10241
313-10261
                                                                                                         SOB-A-LONG TEST
WRITE RECOVERY TEST
BRANCH GOBBLE TEST
SOFT ERROR - BACKROUND PATTERN TEST
WORST CASE NOISE PARITY TEST
CORRECTION CODE TEST
      314-10307
315-10319
      316-10351
      317-10404
319-10425
                                                                                                           CHECK ECC DISABLE TEST
                                                                   MTP041
MTP042
MTP043
MTP044
                                                                                                           ADDRESS TO CSR ON DOUBLE BIT ERROR TEST
      321-10466
322-10519
                                                                                                           EXTENDED ADDRESS TO CSR ON ERROR TEST
                                                                                                           WRITE BYTE CLEARS SINGLE BIT ERROR TEST
      323-10560
                                                                                                           SHIFTING CHECK BITS THROUGH THE CSR TEST
```

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 2-3 TABLE OF CONTENTS

```
324-10640
325-10680
326-10756
327-10800
                               SYNDROMES TO CSR ON DOUBLE BIT ERROR TEST
                   MTP045
                   MTP046
MTP047
                               CHECK SINGLE BIT ERRORS WITH ECC DISABLED
                               NO CSR UPDATE ON SBE WITH EXSISTING DBE
                   MISC SUBROUTINES
327-10802
327-10808
327-10808
328-10835
                   SUBR COPY RO TO R4, R1 TO R3, & R2 TO R5
FLIP WARNING CONSTANTS IN WORST CASE NOISE TESTS
SUBR PITE BACKGROUND
                                 CT CSR INFORMATION FROM CONFIGURATION TABLE
329-10854
                    SUBR
331-10868
333-10920
                                PRINT CONFIGURATION MAP
                    SUBR
                    SUBR
                                TYPE CONFIGURATION
337-11050
                    TRAP
                                PARITY ERROR HANDLER
                               NON-EXISTANT MEMORY (HOLES) HANDLER
TIMEOUT (TRAP TO 4) HANDLER
339-11082
                    TRAP
339-11102
339-11106
                    TRAP
                    TRAP
                                MEMORY MANAGEMENT (TRAP TO 250) HANDLER
                   TRAP
339-11109
                                RESERVED INSTRUCTION HANDLER
                   FIND BAD SP. PC. & PSW FROM STACK
TRAP KERNEL TRAP HANDLER
339-11119
341-11127
341-11135
341-11139
341-11143
                               ENERGIZE TRAP HANDLER
DEENERGIZE TRAP HANDLER
                    TRAP
                    TRAP
                               CACHON TRAP HANDLER
                   TRAP
341-11150
343-11158
                               CACHOFF TRAP HANDLER
                    TRAP
                                LOAD CSR TRAP HANDLER
                    TRAP
343-11177
                   TRAP
                                READ CSR TRAP HANDLER
344-11185
                                TEST (R1) & READ CSR CAREFULLY
                    TRAP
346-11222
                   TRAP
346-1122

346-11236

346-11234

346-11238

346-11242

346-11246

346-11251

348-11258

348-11283

350-11293

350-11318

352-11329

352-11337

352-11342

354-11349
                                ECC DISABLE ALL CSR'S TRAP HANDLER
                                ECC DISABLE OF 1 SELECTED CSR TRAP HANDLER
                   TRAP
                                INITIALIZE ALL CSR'S TRAP HANDLER
                    TRAP
                    TRAP
                                INITIALIZE 1 SELECTED CSR TRAP HANDLER
                               ENABLE SEE PARITY TRAPS ON ALL CSR'S ENABLE SEE PARITY TRAPS ON 1 SELECTED CSR
                    TRAP
                    TRAP
                                WRITE CHECKBITS THRU ALL CSR'S TRAP HANDLER
                    TRAP
                                WRITE CHECKBITS THRU 1 SELECTED CSR TRAP HANDLER
                    TRAP
                               WAS THERE A SBE ON ANY CSR TRAP HANDLER
WAS THERE A SBE IN 1 SELECTED CSR TRAP HANDLER
WAS THERE A DBE ON ANY CSR TRAP HANDLER
WAS THERE A DBE ON 1 SELECTED CSR TRAP HANDLER
CLEAR ALL ECC CSR'S TRAP HANDLER
CLEAR ALL ECC CSR'S TRAP HANDLER
                    TRAP
                    TRAP
                    TRAP
                    TRAP
                    TRAP
                                CLEAR 1 SELECTED CSR TRAP HANDLER
                    TRAP
                               ECC DISABLE, CHECK MODE, & WRITE CHECKBITS IN ALL CSR'S TRAP HANDLER ECC DISABLE, CHECK MODE, & WRITE CHECKBITS IN 1 SELECTED CSR WRITE IN ALL CSR'S
                    TRAP
                    TRAP
354-11349
354-11364
355-11373
                    SUBR
                    TRAP
                                INVALIDATE BACKGROUND PATTERN
                                GENERATE AND TEST ERROR ADDRESS
                    TRAP
 355-11427
357-11434
                                            ENABLE CHECK/SYNDROME BIT REGISTER
                    TRAP
                    SUBR
                                GENERATE CHECK BITS
361-11503
361-11588
                    SUBR
                                MAPPER
                    TRAP
                                MAP KERNEL (ALMOST 1 TO 1) TRAP HANDLER
363-11611
365-11716
365-11761
367-11774
369-11822
                    RELOCATE PROGRAM
                    UNRELOCATE PROGRAM
                    SETUP LOWER 16K OF UNIBUS MAP
                    MOVE BANKS
                                MAP USER TO NEW BANK
                    SUBR
 369-11842
369-11855
                                SETUP KERNEL PAR'S FOR NEW BANK
                    SUBR
                    SUBR
                                MAP KERNAL PARS 4 AND 5 TO A BANK
                                SETUP KERNEL PAR'S FOR NEW LOADER BANK UNMAP KERNAL PAR'S 4 AND 5
 369-11865
                    SUBR
 369-11876
                    SUBR
 371-11883
                    SUBR
                                EXAMINE BANK
```

TABLE OF CONTENTS

```
373-11963
373-11974
                               BANK OK?
                   SUBR
                                INCREMENT PATTERN TESTING
                   SUBR
373-11982
                                SET HIGHEST PATTERN TESTING TYPE
                    SUBR
373-11986
                    SUBR
                                INCREMENT BANK & TEST
375-11986
375-11993
377-12022
377-12031
377-12058
379-12068
380-12095
380-12097
382-12147
                   BOOTSTRAP ROUTINE
                   HALT PROGRAM
                   SHUTDOWN DIAGNOSTIC
                   APT SHUTDOWN SEQUENCE
                   BLOCK MOVE SUBROUTINE
                   FIELD SERVICE MODE
                               FIELD SERVICE COMMAND MODE
                   SUBR
                    COMMAND O
                                           EXIT
382-12169
                                COMMAND
                                                        READ CSR
384-12184

386-12208

388-12250

390-12302

391-12433

393-12439

395-12480

397-12510

399-12551

399-12561

399-12566

399-12572

400-12618

402-12632

404-12638

406-12644

421-13212

421-13226

423-13267

425-13284

430-13472

432-13499

434-13541

449-14334

454-14583

454-14583
                                COMMAND
                                                        LOAD CSR
                                                        EXAMINE MEMORY
                   FS
                                COMMAND
                                COMMAND
                                                        MODIFY MEMORY
                   FS
                                COMMAND 5
                                                        SELECT BANK & PATTERN
                   FS
                                                        TYPE CONFIGURATION MAP
                                COMMAND
                                COMMAND
                                                        SOB-A-LONG TEST
                                COMMAND 8
                                                        ERROR SUMMARY
                               COMMAND 9
COMMAND 10
COMMAND 11
COMMAND 12
                                                        REFRESH TEST
                   FS
                   FS
                                                        SET FILL COUNT
                                                        ENTER KAMIKAZE MODE
                                                        EXIT KAMIKAZE MODE
                   FS
                                COMMAND 13
                                                        TURN CACHE OFF
TURN CACHE ON
                                COMMAND 14
                                COMMAND 15
                                                        TEST ONLY SELECTED BANKS
                                                        RESUME TESTING ALL BANKS
                    FS
                                COMMAND 16
                                COMMAND 17
                                                        ENABLE TRACE
                    FS
                    FS
                                COMMAND 18
                                                        DISABLE TRACE
                    SUBR
                                DETERMINE CORRECT CSR
                    ERROR DATA (SUPERVISOR) SETUP STUFF DATA WAS 3 WORDS
                    GET DATA FROM ABORTED AREA IF POSSIBLE
                    POWER FAIL AUTO RESTART
                    ROUTINE POWER DOWN AND UP
                    POWER FAIL WHILE RELOCATED
                    POWER UP FROM BANK O TO RELOCATION
                    IO SUBROUTINES
                    ROUTINE TYPE
                    ERROR DATA SETUP
                    DATA WAS A WORD
DATA WAS A BYTE
DATA WAS A 7 BIT BYTE
 456-14608
456-14608
456-14623
458-14632
462-14721
463-14791
465-14808
468-14923
476-15150
481-15292
482-15370
483-15427
485-15547
                    DETERMINE XOR OF GOOD & BAD
                    LOG ERROR ON BAD BANK
                    ROUTINE SCOPE HANDLER
                    SUBR
                                DISPLAY
                    ROUTINE ERROR HANDLER
                    ROUTINE ERROR MESSAGE TYPEOUT
                                DETAILED ERROR REPORT
                    SUBR
                    ROUTINE BINARY TO OCTAL (ASCII) AND TYPE ROUTINE CONVERT BINARY TO DECIMAL AND TYPE ROUTINE TTY INPUT
                    CONTROL
                    CONTROL S & CONTROL Q
                    ROUTINE READ AN OCTAL NUMBER FROM THE TTY
```

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 2-5 TABLE OF CONTENTS

ROUTINE READ A DECIMAL NUMBER FROM THE TTY ROUTINE SAVE AND RESTORE RO-R5
ROUTINE RANDOM NUMBER GENERATOR ROUTINE DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT
TABLES
APT MAILBOX-ETABLE
ROUTINE TRAP DECODER
TRAP TABLE
TABLE ERROR POINTER
ERROR DATA TAGS (DT)
ERROR DATA FORMATS (DF)
ERROR MESSAGES (EM)
ERROR DATA HEADERS (DH)
MESSAGES

REVISION HISTORY

REVISION CZMSPA

DATE ======== 1-JUN-82

AUTHOR IRA CHAVIS

CHANGES

NONE - NEW PROGRAM

OPERATIONAL SWITCH SETTINGS SWITCH REGISTER DEFINITIONS

*	SWITCH	USE	
*	15 14	HALT ON ERROR LOOP ON TEST	
*	13	INHIBIT ERROR TYPEOUTS INHIBIT RELOCATION QUICK VERIFY	
* *	10	BELL ON ERROR	
*	8	HALT PROGRAM (UNRELOCATED RESTORE LO	ADERS)
* *	5	INHIBIT CONFIGURATION MAP LIMIT MAX ERRORS PER BANK FAT TERMINAL (132 COLUMNS OR BETTER)	
*	3	TEST MODE - SEE DOCUMENT TEST MODE - SEE DOCUMENT	
*	0	TEST MODE - SEE DOCUMENT DETECT SINGLE BIT ERRORS	

82 83	TABLE OF CONTENTS
84 85	1.0 GENERAL PROGRAM INFORMATION
81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110	1.1 PROGRAM PURPOSE (ABSTRACT) 1.2 SYSTEM REQUIREMENTS 1.3 RELATED DOCUMENTS AND STANDARDS 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES 1.5 ASSUMPTIONS
93	2.0 OPERATING INSTRUCTIONS
95 96 97 98 99	2.1 LOADING AND STARTING PROCEDURES 2.2 DEFAULT TEST SEQUENCE 2.3 SPECIAL ENVIRONMENTS 2.4 PROGRAM OPTIONS 2.5 EXECUTION TIMES
100 101	3.0 ERROR INFORMATION
102 103 104 105	3.1 ERROR REPORTING 3.2 ERROR ABBREVIATIONS 3.3 ERROR HALTS
106 107	4.0 PROGRESS REPORTS
108	5.0 CSR INFORMATION TABLES
112 113	5.1 MS11-P CSR 5.2 MS11-L CSR 5.3 MS11-M CSR
114	6.0 SUB-TEST SUMMARIES
116 117 118	6.1 TESTS 6.2 PATTERNS
120	7.0 PROGRAM FEATURES
118 119 120 121 122 123 124 125	7.1 FAST DATA ACCESS RATES 7.2 BANK ZERO TESTING 7.3 MEMORY CONFIGURATION MAP 7.4 EVERYTHING YOU'VE ALWAYS WANTED TO KNOW ABOUT SUPERMAC 7.5 MEMORY MANAGEMENT MAPPING

128	
120	
154	
130	
131	
173	
155	
133	
134	
175	
133	
136	
137	
138	
170	
134	
140	
141	
1/2	
145	
143	
144	
145	
177	
140	
147	
148	
1/0	
147	
150	
151	
152	
155	
153	
154	
155	
154	
120	
15/	
158	
150	
140	
100	
161	
162	
147	
103	
164	
165	
128 129 131 133 133 133 133 133 134 134 134 134	
100	
167	
168	
160	
109	

- 1.0 GENERAL PROGRAM INFORMATION
- 1.1 PROGRAM PURPOSE (ABSTRACT)
 - A. INTENDED FOR USE ON ALL PDP-11/24/44'S WHICH MEET THE CONDITIONS IN 1.2.1.
 - B. THIS PROGRAM WILL BE USED BY SYSTEM MANAGERS AND OPERATORS TO DETERMINE THE CORRECT OPERATION OF MAIN MEMORY AND ALSO IT WILL BE PRIMARILY USED BY FIELD SERVICE AND MANUFACTURING TO ISOLATE FAILURES TO THE MEMORY AND TO ISOLATE FAILURES WITHIN THE MEMORY TO THE CORRECT CARD.
 - C. THE OBJECT OF THIS SOFTWARE IS TO FUNCTIONALLY TEST AND VERIFY ALL MAIN MEMORY FUNCTIONS AS FAST AS POSSIBLE.
 - D. THERE IS THE CAPABILITY OF TESTING MIXED CONFIGURATIONS (MS11-L, MS11-M AND MS11-P) ON THE SYSTEM.
 - E. IT HAS SPECIAL A MAINTENANCE MODE (FIELD SERVICE MODE) TO PROVIDE SPECIFIC FUNCTIONAL CAPABILITIES.
- 1.2 SYSTEM REQUIREMENTS
- 1.2.1 HARDWARE REQUIREMENTS -

PDP-11-24/44 CPU WITH 22 BIT ADDRESSING AND AT LEAST 64K (16 BIT WORDS) OF MEMORY AND MEMORY MANAGEMENT.

NOTE

- 1. LIKE MEMORY TYPES MUST BE ON 16K WORD BOUNDARIES STARTING AT PHYSICAL ADDRESS 0.
- 2. PDP-11 SERIES 16/18 BIT PROCESSORS ARE NOT SUPPORTED.

1.2.2 SOFTWARE REQUIREMENTS -

THIS PROGRAM IS DESIGNED TO RUN STAND ALONE OR UNDER ANY OF THE FOLLOWING MONITORS:

ACT APT

1.3 RELATED DOCUMENTS AND STANDARDS

- 1. PDP-11/04/24/34/44/70 PROCESSOR HANDBOOK (EB-19402)
- 2. PDP-11/44 USER'S GUIDE (EK-11044-UG)
- 3. MS11-M USER'S GUIDE (EK-MS11M-UG-001)
- 4. MS11-L USERS GUIDE (EK-MS11L-UG-001)
- 5. MS11-P TECHNICAL MANUAL (EK-MS11P-TM-001)

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

IF THE PROGRAM IN ANY WAY MISBEHAVES, THEN:

- 1. TRY IT AGAIN WITH CACHE OFF (REFERENCE SECTION 2.4.3.1)
- 2. INHIBIT RELOCATION (REFERENCE SECTION 2.4.1)
- 3. TRY CPU DIAGNOSTICS
- 4. TRY MEMORY MANAGEMENT DIAGNOSTICS
- 5. TRY CACHE DIAGNOSTICS (WHERE APPLICABLE)
- 6. TRY UNIBUS MAP DIAGNOSTICS (WHERE APPLICABLE)

1.5 ASSUMPTIONS

THIS PROGRAM ASSUMES THE CORRECT OPERATION OF THE CPU, MEMORY MANAGEMENT, CACHE, AND THE UNIBUS MAP. THIS PROGRAM OCCUPIES (INITIALLY) BANK 0 (0-16K). THE XXDP LOADERS ARE IN BANK 1.

- 2.0 OPERATING INSTRUCTIONS
- 2.1 LOADING STARTING PROCEDURES
- 2.1.1 QUICK STARTING -
 - 1. LOAD ADDRESS 200
 - 2. SET SWITCH REGISTER FOR OPTIONS (NORMALLY 0)
 - 3. START

NOTE

IF ON AN 11/24 USING MS11-L MEMORY BE SURE THAT THE PERIPHERAL PAGE JUMPER IS IN PLACE; FAILURE TO DO SO SENDS THE DIAGNOSTIC TO NEVER-NEVER LAND.

- 2.1.2 STOPPING -
 - 1. SET SW8, AND/OR
 - 2. TYPE CONTROL "C" (REFERENCE SECTION 2.4.4.1).
- 2.1.3 RESTARTING (PRESERVE CONFIGURATION TABLE) -
 - 1. LOAD ADDRESS 202
 - 2. SET SWITCH REGISTER FOR OPTIONS (NORMALLY 0)
 - 3. START

270	
210	
271	
611	
272	
212	
273	
27/	
274	
275	
275	
27	
2/0	
277	
211	
279	
210	
272 273 274 275 276 277 278 279 280 281 282	
617	
280	
200	
281	
202	
275 276 277 278 279 280 281 282	
207	
283	
201	
284 285 286 287	
205	
203	
286	
200	
287	
288	
288	
200	
289	
290	
290	
201	
291	
288 289 290 291 292	
646	
203	
673	
20/	
674	

2.1.4 SWITCH REGISTER OPTIONS -

SWITCH	USE
15 14 13 12 11 10	HALT ON ERROR LOOP ON TEST INHIBIT ERROR TYPEOUTS INHIBIT RELOCATION QUICK VERIFY BELL ON ERROR LOOP ON ERROR
98765	HALT PROGRAM (UNRELOCATE RESTORE LOADERS) DETAILED ERROR REPORTS INHIBIT CONFIGURATION MAP LIMIT MAX ERRORS PER BANK
3210	FAT TERMINAL (132 COLUMNS OR BETTER) TEST MODE - SEE DOCUMENT TEST MODE - SEE DOCUMENT TEST MODE - SEE DOCUMENT DETECT SINGLE BIT ERRORS

2.2 DEFAULT TEST SEQUENCE

THE FOLLOWING TWO LISTS GIVE THE TEST PROTOCOL FOR PARITY AND ECC MEMORY. TESTS MARKED WITH A "*" ARE NOT NORMALLY RUN EXCEPT UNDER ACT OR APT. OR THROUGH A FIELD SERVICE COMMAND (REFERENCE SECTION 2.4.4.8).

2.2.1 TEST PROTOCOL FOR MS11-L PARITY MEMORY -

1	EST	TEST NAME	TIME	(SEC/16K)
	34	SOFT ERROR TEST		<1
	17	INITIAL DATA TEST HOLDING 1'S AND 0'S TEST		1 1 1 1
	7	ADDRESS BIT TEST		21
	1	ADDRESS TEST		<1
	2	COMPLEMENT ADDRESS TEST		<1
	3	3 XOR 9 TEST		1
	4	ROTATING O'S TEST		1
	21	ROTATING 1'S TEST		1
	21	MARCHING 1'S AND 0'S TEST WORST CASE NOISE PARITY T	ECT	NI/A
*	33	REFRESH TEST	E21	N/A 10
*	23	SHIFTING DIAGONAL TEST		10
	26	RANDOM DATA TEST		<1
*	24	FAST GALLOPING PATTERN TE	ST	10 <1 20
*	31	SOB-A-LONG TEST		3
*	32	WRITE RECOVERY TEST		<1 35
*	33	BRANCH GOBBLE TEST		55
	34	SOFT ERROR TEST		<1

2.2.2 TEST PROTOCOL FOR MS11-M ECC MEMORY -

TEST	TEST NAME TIME (SEC	(/16K)
35 +0 +0 +0 +0 +0 +0 +0 +0 +0 +0	ROTATING 1'S TEST INTERRUPT ENABLE TEST SINGLE BIT ERROR TEST WRITE BYTE CLEARS SBE TEST CREATE DOUBLE BIT ERROR TEST WRITE INHIBIT OF BYTE W/DBE TEST WRITE INHIBIT OF WORD W/DBE TEST SOFT ERROR TEST INITIAL DATA TEST BYTE ADDRESS TEST HOLDING 1'S AND 0'S TEST ADDRESS BIT TEST ADDRESS TEST COMPLEMENT ADDRESS TEST ROTATING 0'S TEST ROTATING 0'S TEST ROTATING 0'S AND 1'S TEST REFRESH TEST RANDOM DATA TEST REFRESH TEST SOB-A-LONG TEST WRITE RECOVERY TEST BRANCH GOBBLE TEST 35	
34	SOFT ERROR TEST <1	

- a RUN ONLY ON THE FIRST PASS WHEN UNDER ACT OR APT
- + RUN TWICE FOR EACH 16K BANK IF INTERLEAVED

AT THE END OF EACH PASS THE PROGRAM WILL RUN CLEANUP PATTERNS #30, AND #27 FOR ALL BANKS.

395	
390	
398	
399	
400	
401	
402	
404	
405	
406	
407	
408	
409	
411	
412	
413	
414	
415	
407 408 410 411 412 413 414 415 416 421 421 423 424	
418	
419	
420	
421	
422	
423	
425	
426	
427	
428	
429	
427 428 429 430 431	
431	
37890123440567890112345167890122345678901233455640078901123451678901223456789012334556	
434	
435	
436	

2.2.3 TEST PROTOCOL FOR MS11-P ECC MEMEORY

PATTERN	PATTERN NAME T	IME	(SEC/16K)
5 34 44 14 45 320 34 42 43 44 10 17 17	ROTATING 1'S TEST SOFT ERROR TEST INITIAL DATA TEST SHIFTING CHECK BITS THRU BASIC DBE TEST CSR TEST SYNDROMES IN CSR ON DBE TEST CORRECTION CODE TEST SYNDROMES IN CSR ON SBE TEST CHECK ECC DISABLE TEST ADDRESS TO CSR ON DBE TEST EXTENDED ADDRESS TO CSR TEST BYTE WRITE TEST CHECK SBE WITH ECC DISABLE TEST NO CSR UPDATE ON SBE WITH DBE TES BYTE ADDRESS TEST HOLDING 1'S AND O'S TEST ADDRESS BIT TEST ADDRESS TEST COMPLEMENT ADDRESS TEST ROTATING O'S TEST ROTATING O'S TEST ROTATING O'S AND 1'S TEST REFRESH TEST RANDOM DATA TEST		1
* 22 * 26 * 24 * 31 * 32 * 33	FAST GALLOPING PATTERN TEST SOB-A-LONG TEST WRITE RECOVERY TEST BRANCH GOBBLE TEST SOFT ERROR TEST		10 <1 20 3 <1 35 <1

a - RUN ONLY ON THE FIRST PASS WHEN UNDER ACT OR APT

AT THE END OF EACH PASS THE PROGRAM WILL RUN CLEANUP PATTERNS #30, AND #27 FOR ALL BANKS.

2.3 SPECIAL ENVIRONMENTS

2.3.1 XXDP -

THE FIRST PASS WILL BE A QUICK VERIFY PASS IF AND ONLY IF IT IS IN CHAIN MODE.

2.3.2 ACT APT AUTOMATIC MODE -

THE PROGRAM WILL NOT CREATE DOUBLE BIT ERRORS (DBE'S) AFTER THE 1ST PASS.

2.3.2.1 APT EXECUTION TIMES -

HERE ARE SOME MEASURED EXECUTION TIMES FOR AN 11/44 WITH CACHE UNDER

1ST QV PASS 2ND PASS ONWARD

128K MS11-M (NON-INTERLEAVED) 10 MIN 15 SEC 7 MIN 40 SEC

128K MS11-L 9 MIN 50 SEC 7 MIN 30 SEC

256K MS11-M (INTERLEAVED) 19 MIN 50 SEC 14 MIN 45 SEC

512K MS11-P NOT ESTABLISHED AT RELEASE TIME

THE FIRST PASS WILL BE A QUICK VERIFY PASS

NOTE

EVEN THOUGH THE FIRST PASS IS A QV PASS IT TAKES LONGER THAN THE SUBSEQUENT NON-QV PASSES DUE TO THE FACT THAT IT IS RUNNING MORE PATTERNS, SOME OF WHICH (PATTERNS #24 AND #33 FOR EXAMPLE) CAN BE EXTREMEMLY TIME CONSUMING.

2.3.2.2 APT ENVIRONMENT TABLE -

THE FOLLOWING TABLE GIVES SOME OF THE STANDARD SETTINGS FOR THE APT E-TABLE. THEY MAY BE MODIFIED AS NOTED AS THE USER SEES FIT.

THIS PARAMETER SHOULD BE SET ACCORDING TO THE AMOUNT AND TYPE OF MEMORY TO BE TESTED. THE ABOVE TABLE (APT EXECUTION TIMES) GIVES SOME MEASURED TIMES. FOR ANY PATTERNS DELETED (THROUGH USE OF THE DEVICE DESCRIPTOR WORDS) REFERENCE SECTION 2.2 FOR INDIVIDUAL PATTERN TIMES.

NOTE

THE TIMES GIVEN IN SECTION 2.2 ARE FOR 16K CHUNKS OF MEMORY, NOT 128K BOARDS!

LONGEST TEST TIME:
THIS PARAMETER SHOULD BE SET TO THE EXECUTION TIME OF THE LONGEST
PATTERN BEING RUN. FOR THE DEFAULT CASE THIS IS 35 SECONDS FOR
PATTERN #33.

ADDITIONAL RUN TIME: NOT USED BY PROGRAM.

SOFTWARE ENVIRONMENT:

FOR APT AUTO MODE THIS PARAMETER SHOULD BE SET TO A "1". FOR DUMP MODE SET THIS TO A "0".

ENVIRONMENT MODE:
WHEN THIS PARAMETER IS SET TO A 'O' THE PROGRAM DOES IT'S OWN SIZING. IF THE USERS SETS BIT #7 HOWEVER, HE MUST SPECIFY THE TYPES AND AMOUNTS OF MEMORY TO BE TESTED.

THE DEFAULT SETTING OF THIS SWITCH IS "101". APT USES THIS AS THE SWITCH REGISTER FOR THE PROGRAM. REFERENCE SECTION 2.4.1 FOR MORE INFORMATION ON SWITCH SETTINGS.

SWITCH 2:
THIS SWITCH, IF SET TO ANY NON-ZERO NUMBER, IS USED TO LIMIT THE AMOUNT OF PASSES APT WILL MAKE. THE PROGRAM WILL HANG AFTER THIS COUNT HAS BEEN REACHED.

CPU OPTIONS: NOT USED BY PROGRAM.

MEMORY TYPE N (N=1 TO 4)

IF BIT #7 OF ENVIRONMENT MODE IS SET THESE FOUR WORDS ARE USED TO LOG THE DIFFERENT TYPES OF MEMORY TO BE TESTED. IF BIT #7 IS NOT SET THESE LOCATION ARE NOT USED.

MAXIMUM ADDRESS N (N=1 TO 4)

THESE FOUR WORDS ARE USED IN CONJUNTION WITH THE CORRESPONDING

MEMORY TYPE WORDS TO INDICATE THE HIGHEST ADDRESS THAT MEMORY TYPE OCCUPIES.

NOTE

THE ABOVE TWO PARAMETERS DO NOT ACTUALLY HAVE TO REPRESENT AN ACCURATE CONFIGURATION OF MEMORY. ALL THE PROGRAM LOOKS FOR IS AN ACCURATE TALLY OF MEMORY AMOUNT!

INTERRUPT VECTOR N (N=1 TO 2)
NOT USED BY PROGRAM.

BUS PRIORITY N (N=1 TO 2) NOT USED BY PROGRAM.

BASE ADDRESS: NOT USED BY PROGRAM.

DEVICE MAP: NOT USED BY PROGRAM.

CONTROLLER DESCRIPTOR CODE N (N=1 TO 2)
NOT USED BY PROGRAM.

THE DEVICE DESCRIPTOR CODES ARE USED BY THE PROGRAM TO DETERMINE WHICH PATTERNS IT WILL RUN. THE DEFAULT VALUES OF THESE WORDS ARE ALL "1"S, INDICATING THAT ALL OF THE PATTERNS SHOWN IN SECTION 2.2 ARE EXECUTED (SAVE FOR EXCEPTIONS AS NOTED THERE). EACH SET OF WORDS CONTROLS A TABLE IN THE PROGRAM AS FOLLOWS:

DD WORDS PROGRAM TABLE (SYMBOLIC LOCATION)
WORDS 0-1 MKCSRT

WORDS 2-3 MKPAT

WORDS 4-5 MJPAT

BIT #0 SET IN THE FIRST WORD INDICATES THAT THE FIRST PATTERN IN THE TABLE WILL BE EXECUTED, BIT #1 THE SECOND, BIT #2 THE THIRD,... BIT #0 OF THE SECOND WORD INDICATES THAT THE 17TH ENTRY IN THE TABLE WILL BE EXECUTED, AND SO ON.

2.3.3 NO SBE FREE BANKS -

IF THE PROGRAM CANNOT FIND ANY SBE (SINGLE BIT ERROR) FREE LOCATIONS (IN NON-PROTECTED ECC MEMORY) IT WILL PRINT OUT AN ERROR MESSAGE AND CONTINUE TESTING BY-PASSING THE ECC LOGIC TESTS.

2.3.4 MIXED PARITY ECC CONFIGURATIONS -

THE PROGRAM WILL FUNCTION NORMALLY IN MIXED ENVIRONMENTS. THE SEQUENCE OF TESTING MAY SEEM STRANGE DUE TO THE RECURSIVE TEST MODE ALGORITHM (REFERENCE SECTIONS 2.4.1.1, 2.4.1.2, 2.4.1.3).

2.4 PROGRAM OPTIONS

2.4.1 SWITCH REGISTER DETAILS -

IF A HARDWARE SWITCH REGISTER IS NOT AVAILABLE THEN THE SOFTWARE SWITCH REGISTER IS IN LOCATION 176. IF UNDER APT IF BIT7 IS SET IN THE E-TABLE SYMBOLIC LOCATION 'SENVM' THE APT SOFTWARE SWITCH REGISTER WILL BE USED (LOCATION \$SWREG).

TO CHANGE THE SOFTWARE SWITCH REGISTER CONTENTS: TYPE 'CONTROL G'.
THIS WILL CAUSE DISPLAY THE CURRENT VALUE OF THE SWR AND PROMPT FOR
THE OCTAL INPUT OF THE NEW SWR VALUE FROM THE TERMINAL. THIS ROUTINE
WILL IGNORE YOU (NOT RESPOND TO CONTROL 'G') IF YOU HAVE A HARDWARE
SWITCH REGISTER.

SW15 = HALT ON ERROR (100000)

CONTINUING FROM THIS HALT WILL FIRST CHECK FOR A CHANGE IN THE SOFTWARE SWITCH REGISTER ("CONTROL G" IN THE TTY INPUT BUFFER) THEN IT WILL CONTINUE TESTING.

SW14 = LOOP ON TEST (40000)

THIS WILL CAUSE LOOPING ON THE PRESENT TEST OR PATTERN (BACK TO LAST SCOPE TRAP). IF IN A PATTERN THEN THE LOOPING WILL BE FOR AN ENTIRE BANK OF 16K ADDRESSES.

SW13 = INHIBIT ERROR TYPEOUTS (20000)

THIS WILL CAUSE RETURNS FROM THE ERROR ROUTINE WITHOUT THE TYPED MESSAGES. OTHER ON ERROR FUNCTIONS ARE NOT AFFECTED.

SW12 = INHIBIT RELOCATION (10000)

THIS PREVENTS THE PROGRAM FROM MOVING AND CONSEQUENTLY PREVENTS THE PROGRAM FROM TESTING AT LEAST 32K OF MEMORY.

SW11 = QUICK VERIFY (4000)

IF THIS SWITCH IS SELECTED APPROXIMATELY ONE 64TH OF THE POSSIBLE COMBINATIONS OF SBE'S DBE'S ARE TESTED.

EACH PASS COMPLETE TYPEOUT WILL INDICATE THIS MODE BY PRECEDING THE PASS NUMBER WITH "QV".

SW10 = BELL ON ERROR (2000)THIS CAUSES A BELL (OR BEEP OR CLICK) ON EACH ERROR SW9 = LOOP ON ERROR (1000)THIS WILL CAUSE LOOPING FROM FAILURE POINT BACK TO THE LAST CORRECTLY INITIALIZED AREA OF THE CURRENT TEST. = HALT PROGRAM SW8 (400)THIS INITIATES THE FOLLOWING SEQUENCE: 1. IF PROGRAM IS RELOCATED IT MOVES BACK TO BANK ZERO. 2. FLUSH OUT ALL POSSIBLE DBE'S. TURNS OFF MEMORY MANAGEMENT. RESTORE LOADERS. UNMAP THE UNIBUS MAP (IF THERE IS ONE). 6. HALT IF UNDER APT OR ACT BRANCH SEL. = DETAILED ERROR REPORTS (200) SW7 AFTER ANY NORMAL ERROR REPORT IS TYPED THIS OPTION CAUSES THE CONTENTS OF THE FOLLOWING REGISTERS TO BE TYPED: RO, R1, R2, R3, R4, R5, SP, "CONTROL", "CPUERR" = INHIBIT CONFIGURATION MAP SW6 (100)THIS INHIBITS THE PRINTING OF A MAP SHOWING THE MEMORY CONFIGURATION - REFERENCE SECTION 7.3 = LIMIT MAX ERRORS PER BANK SW5 THIS WILL LIMIT THE NUMBER OF ERROR TYPEOUTS PER BANK. THE DEFAULT IS 10. DECIMAL, HOWEVER THIS CAN BE CHANGED BY CHANGING LOCATION "ERRMAX" MANUALLY.

SW4 = FAT TERMINAL (20)

THIS INFORMS THE PROGRAM THAT THE CONSOLE TERMINAL HAS A WIDTH OF AT LEAST 132 COLUMNS (LA36 WITH WIDE PAPER).

SW3-1 = TEST MODE

TEST MODES DETERMINE THE RECURSION ALGORITHM TO BE USED DURING PATTERN TESTS.

MODE NAME DESCRIPTION

(0)	0	BAFPAF	BANKS FORWARD, PATTERNS FORWARD
(0) (2) (4) (6)	1	BAFPAR	BANKS FORWARD, PATTERNS REVERSE
(4)	2	BAWPAF	BANKS WORST FIRST, PATTERNS FORWARD.
(6)	3	BAWPAR	BANKS WORST FIRST, PATTERNS REVERSE.
(10)	4	PAFBAF	PATTERNS FORWARD, BANKS FORWARD
(10)	5	PAFBAW	PATTERNS FORWARD, BANKS WORST FIRST
(14)	6	PARBAF	PATTERNS REVERSE, BANKS FORWARD
(16)	7	PARBAW	PATTERNS REVERSE, BANKS WORST FIRST

FOR MORE DETAILS REFERENCE SECTION 2.4.1.1, 2.4.1.2 AND 2.4.1.3.

SWO = DETECT SINGLE BIT ERRORS (SBI'S)

FOR MANUFACTURING PURPOSES THIS SWITCH SHOULD ALWAYS BE ON. FOR FIELD SERVICE PURPOSES THIS SWITCH SHOULD ALWAYS BE OFF.

THIS SWITCH WILL ALLOW ALL ECC SINGLE BIT ERRORS TO BE REPORTED BY DISABLING ERROR CORRECTION.

ERROR PRINTOUTS OF SBE'S ARE NOT DISTINGUISHABLE FROM DBE'S.

NOTE

IF DOUBLE BIT ERRORS ARE FOUND IN THE MEMORY. THIS SWITCH SHOULD BE SET TO MAKE SURE THAT NEW DATA CAN BE WRITTEN TO THE DBE LOCATIONS.

2.4.1.1 TEST MODE EXAMPLE -

EXAMPLE ANALYSIS OF MODE 5 'PAFBAW'. ASSUME BANKS 0 1 ARE MS11-L AND BANKS 2,3,4, 5 ARE MS11-M.

```
ASSUME ALSO THAT BANK 3 IS KNOWN BAD BY THE PROGRAM VIA THE SIZING ROUTINE OR PREVIOUS RUNS THE TESTING SEQUENCE WOULD BE AS FOLLOWS:
:TEST MS11-M MEMORY TYPES FIRST :TEST KNOWN BAD MEMORY (BANK 3)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                TEST 17.
TEST 7.
TEST 1.
TEST 2.
TEST 4.
TEST 5.
TEST 21.
TEST 20.
TEST 20.
TEST 20.
TEST 26.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        BANK
BANK
BANK
BANK
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           BANK
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           BANK
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         BANK
BANK
BANK
BANK
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    : TEST PRESUMED GOOD MEMORY (BANKS 2,4,5)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                TEST 17, TEST 1, TEST 20, TEST 26, TEST 27, TEST
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         BANK 2
BANK 4
BANK 4
BANK 4
```

:RELOCATE TEST PROGRAM SPACE (BANK 0 & 1)

TEST 1. BANK 0
TEST 2. BANK 0
TEST 3. BANK 0
TEST 4. BANK 0
TEST 5. BANK 0
TEST 26. BANK 0
TEST 1. BANK 1
TEST 2. BANK 1
TEST 3. BANK 1
TEST 3. BANK 1
TEST 4. BANK 1
TEST 4. BANK 1
TEST 5. BANK 1
TEST 5. BANK 1
TEST 5. BANK 1

NOTE

THIS IS AN EXAMPLE NOT AN ACTUAL SEQUENCE.

THE TEST SEQUENCE WAS FORWARD (THE SIMPLE PATTERNS FIRST, COMPLEX TESTS LAST) SEQUENCE OF PATTERNS (MS11-M = 17, 7, 1, 2, 4, 5, 21, 20, 22, 26) (MS11-L = 1, 2, 3, 4, 5, 26).

IF THE BANK SELECTION IS FORWARD THE BANKS WILL BE TESTED IN THE FOLLOWING ORDER:

- 1. ECC BANKS THAT ARE NOT PROTECTED OR PROGRAM SPACE (FROM 0 TO 167).
- 2. PARITY BANKS THAT ARE NOT PROGRAM SPACE (FROM 0 TO 167).
- 3. THE PROGRAM NOW RELOCATES TESTS:
- 4. ECC BANKS THAT WERE PROTECTED OR PROGRAM SPACE (FROM 0 TO 167).
- 5. PARITY BANKS THAT WERE PROGRAM SPACE (FROM 0 TO 167).

IF BANK SELECTION IS WORST FIRST THE CONFIGURATION TABLE WILL BE CONSULTED AND BANKS WILL BE TESTED IN THE FOLLOWING ORDER.

- 1. ECC BANKS THAT ARE KNOWN BAD AND ARE NOT PROTECTED OR PROGRAM SPACE (FROM 0 TO 167).
- 2. PARITY BANKS THAT ARE KNOWN BAD AND ARE NOT PROGRAM SPACE (FROM 0 TO 167).
- 3. ECC BANKS THAT ARE PRESUMED GOOD AND ARE NOT PROTECTED OR

PROGRAM SPACE (FROM 0 TO 167).

- 4. PARITY BANKS THAT ARE PRESUMED GOOD AND ARE NOT PROGRAM SPACE (FROM 0 TO 167).
- 5. THE PROGRAM NOW RELOCATES TESTS:
- 6. ECC BANKS THAT ARE KNOWN BAD AND WERE PROTECTED OR PROGRAM SPACE (FROM 0 TO 167).
- 7. PARITY BANKS THAT ARE KNOWN BAD AND WERE PROGRAM SPACE (FROM 0 TO 167).
- 8. ECC BANKS THAT ARE PRESUMED GOOD AND WERE PROTECTED OR PROGRAM SPACE (FROM 0 TO 167).
- 9. PARITY BANKS THAT ARE PRESUMED GOOD AND WERE PROGRAM SPACE (FROM 0 TO 167).

2.4.1.2 TEST MODE DETAILS -

MODE 0 = 'BAFPAF' BANKS FORWARD, PATTERNS FORWARD

THIS IS THE DEFAULT AND SIMPLEST MODE.

THIS MODE TESTS EACH BANK COMPLETELY FROM 0 TO 167 EXCEPT THOSE REQUIRING RELOCATION*.

WHILE TESTING EACH BANK THE PATTERNS ARE RUN WITH THE SIMPLE ONES FIRST BUILDING TO THE MORE COMPLEX.

MODE 1 = 'BAFPAR' = BANKS FORWARD, PATTERNS REVERSE

THIS MODE TESTS EACH BANK COMPLETELY FROM 0 TO 167 EXCEPT THOSE REQUIRING RELOCATION*.

WHILE TESTING EACH BANK THE PATTERNS ARE RUN WITH THE MOST COMPLEX ONES FIRST, WORKING TO THE SIMPLE ONES.

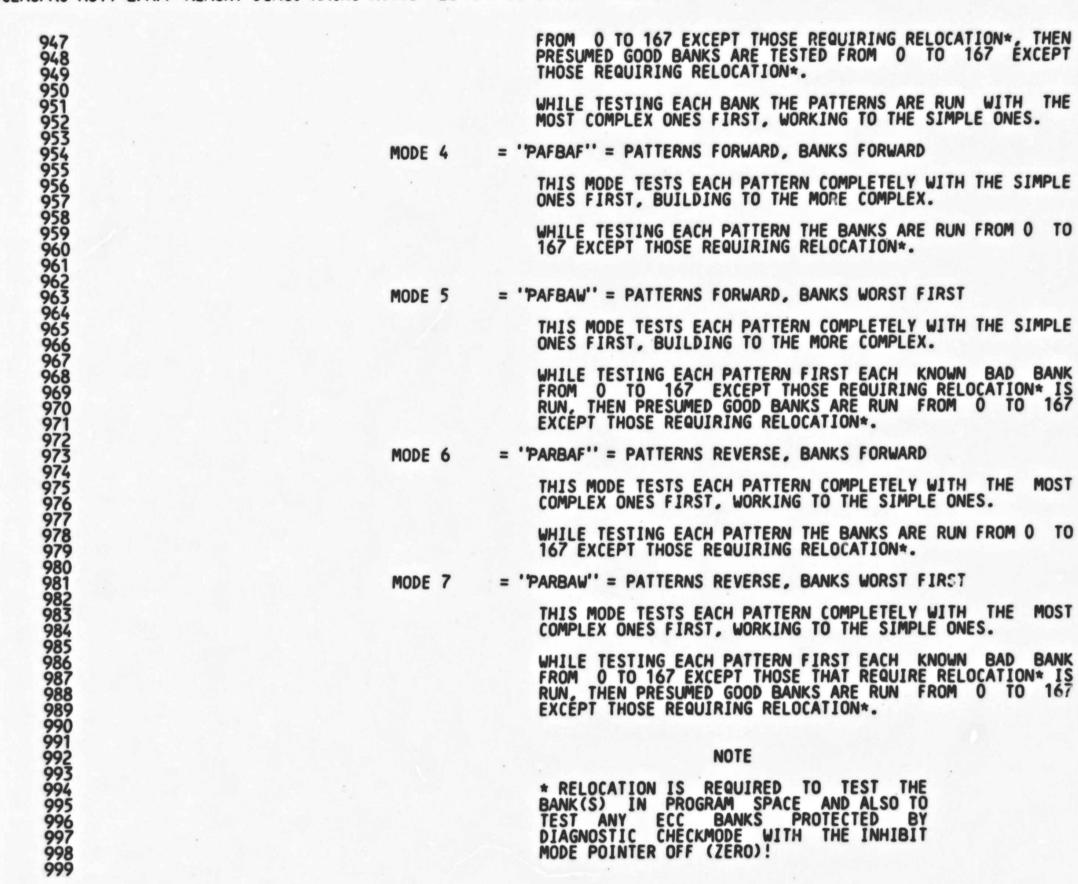
MODE 2 = 'BAWPAF'' = BANKS WORST FIRST, PATTERNS FORWARD

THIS MODE FIRST TESTS EACH KNOWN BAD BANK COMPLETELY FROM 0 TO 167 EXCEPT THOSE REQUIRING RELOCATION*, THEN PRESUMED GOOD BANKS ARE TESTED FROM 0 TO 167 EXCEPT THOSE REQUIRING RELOCATION*.

WHILE TESTING EACH BANK THE PATTERNS ARE RUN WITH THE SIMPLE ONES FIRST, BUILDING TO THE MORE COMPLEX.

MODE 3 = 'BAWPAR'' = BANKS WORST FIRST, PATTERNS REVERSE

THIS MODE FIRST TESTS EACH KNOWN BAD BANK COMPLETELY



2.4.1.3 TEST MODE APPLICATIONS -

1. TO VERIFY CORRECT OPERATION OF THE MEMORY SYSTEM USE MODE 0 'BAFPAF'.

ADVANTAGES: EASY TO UNDERSTAND.

DISADVANTAGES: IN CASE OF A FAILING BANK, IT MAY TAKE A LONG TIME TO FIND THE FAILURE.

2. TO GET DETAILED ERROR INFORMATION ON KNOWN BAD BANKS (FOUND BY SIZING ROUTINE) USE MODE 2 "BAWPAF".

ADVANTAGES: SEEKS BAD BANKS. EASY TO UNDERSTAND.

DISADVANTAGES: FAILURES OTHER THAN ZEROS ONES MAY TAKE A LONG TIME TO FIND.

3. TO GET GOOD ERROR INFO ON ANY MEMORY PROBLEM FAST USE MODE 4

ADVANTAGES: COVERS ALL BANKS FAST. EASY TO UNDERSTAND.

DISADVANTAGES: FAILURES FROM ONLY COMPLEX PATTERNS MAY TAKE A LONG TIME TO FIND.

4. TO FIND ANY PROBLEM FAST USE MODE 7 'PARBAW'.

ADVANTAGES: COVERS ALL BANKS FAST.

DISADVANTAGES: DIFFICULT TO UNDERSTAND FAILURES REPORTED ARE NOT NECESSARILY THE MOST BASIC FAILURE MODES.

2.4.2 DISPLAY REGISTER -

A SOFTWARE DISPLAY REGISTER EXISTS IN LOCATION 174 IN ADDITION TO ANY HARDWARE DISPLAY EXISTENCE.

DISPLAY FIELDS ARE AS FOLLOWS:

RELOCATED | 14 13 12 11 10 9 8 | 7 6 5 | 4 3 2 1 | NOT USED | PATTERN #

PATTERN # = THE NUMBER OF THE PATTERN PRESENTLY BEING RUN. ALL
PATTERNS ARE DESCRIBED IN SECTION 6.2. ANY PATTERN CAN BE
FOUND IN THE DIAGNOSTIC BY LOOKING UP THE SYMBOLIC TAGS
'MTOONN' AND 'MTPONN' - WHERE 'NN' IS THE TEST NUMBER.
MTOONN REFERS TO THE ROUTINE THAT SETS UP FOR THE TEST
PATTERN WHEREAS MTPONN IS THE ACTUAL PATTERN ITSELF.

NOTE

THE PATTERN # IS NOT NECESSARILY AN INDICATION OF DEGREE OF DIFFICULTY.

BANK = THE NUMBER OF THE BANK (16K) OF MEMORY UNDER TEST (0-167).
THESE BITS DIRECTLY MAP TO PHYSICAL ADDRESS BITS (21:15).

RELOCATED = THIS BIT INDICATES THAT THE PROGRAM IS RELOCATED AND NO LONGER IN BANK O. IT WILL BE RELOCATED TO THE FIRST KNOWN GOOD NON-PROTECTED MEMORY BANK INDICATED ON THE CONFIGURATION MAP (REFERENCE SECTION 7.3).

NOTE

ANOTHER WAY TO OBTAIN THIS INFORMATION IS TO TYPE A CONTROL/T AT THE CONSOLE (REFERENCE SECTION 2.4.4.5).

2.4.3 SPECIAL MEMORY LOCATIONS -

2.4.3.1 CACHE CONSTANT -

THE CACHE CONSTANT IS LOCATED AT SYMBOLIC LOCATION "CACHK" AND IS USED TO ENABLE CACHE.

NOTE

BIT O IN THE CACHE CONSTANT HAS NO EFFECT SINCE IT IS UNCONDITIONALLY SET BY THE PROGRAM WHENEVER IT TRIES TO ENABLE CACHE.

2.4.3.2 CONFIGURATION TABLE

THE CONFIGURATION TABLE IS LOCATED AT SYMBOLIC LOCATION "CONFIG" AND HAS THE FOLLOWING FORMAT:

CONFIG: FIRST 16K CONFIGURATION WORDS (2 EACH)
2ND 16K CONFIGURATION WORDS (2 EACH)

200TH 16K CONFIGURATION WORDS (2 EACH)

CONFIGURATION WORDS:

BIT 0 ERRORS PRESENT MEMORY EXISTS LOW: BIT 1 RESERVED

MED:

BIT 2-4
BIT 5
SKIP ECC LOGIC TESTS FLAG (1=SKIP)
BIT 6
PROTECTED REGION OF AN ECC MEMORY
BIT 7
PROTECTED (PROGRAM SPACE)
BIT 8-11
CSR CODE
BIT 12-15 INTERLEAVED CSR CODE
BIT 0-7 NUMBER OF ERRORS
BIT 8-10 MEMORY TYPE
BIT 11 CSR TESTED OK
BIT 12 INTERLEAVE ENABLED
BIT 13 "BACKGROUND PATTERN VALID" FLAG
BIT 14 BANK SELECTED FOR TEST BY FIELD SER BANK SELECTED FOR TEST BY FIELD SERVICE MODE LOADERS HOME BANK

THIS TABLE IS USED AS THE SOURCE FOR THE CONFIGURATION MAP (REFERENCE. SECTION 7.3).

2.4.4 TERMINAL COMMANDS -

2.4.4.1 CONTROL "C"

THIS COMMAND WILL:

- 1. IF SWITCH 8 (HALT PROGRAM) IN THE SWITCH REGISTER IS SET HALT THE PROGRAM.
- 2. IF SWITCH 8 IS NOT SET, UNRELOCATE IF PROGRAM WAS RELOCATED.
- 3. FLUSH OUT ANY DBE'S.
- 4. TURN OFF MEMORY MANAGEMENT.
- 5. ATTEMPT TO BOOT RKOS DRIVE O.
- 6. FAILING 4. ATTEMPT TO BOOT RKO4 DRIVE 1.
- 7. FAILING 5, GO TO 4.

THIS COMMAND WILL ONLY BE RECOGNIZED AT THE COMPLETION OF THE CURRENT TEST OR PATTERN, OR AT THE END OF A LINE OF AN ERROR MESSAGE.

2.4.4.2 CONTROL 'K" (KILL ERROR PRINTOUT AND SKIP PATTERN)

THIS COMMAND WILL ALLOW YOU TO STOP AN ERROR PRINTOUT AND SKIP TO THE NEXT PATTERN. THIS IS HANDY, FOR EXAMPLE, WHEN YOU HAVE A WHOLE BANK FULL OF ERRORS, HAVE GOTTEN ENOUGH INFORMATION, AND WISH TO SKIP TO THE NEXT PATTERN.

2.4.4.3 CONTROL "T" (TELL ME WHAT'S HAPPENING)

THIS COMMAND WILL PRINT OUT THE INFORMATION ENCODED IN THE DISPLAY REGISTER. THIS IS MAINLY INTENDED FOR CPU'S WITHOUT A HARDWARE DISPLAY REGISTER.

EXAMPLE:

BANK = 17 TEST = 46 RELOCATED BANK= 0 PAT= 26

BY USE OF FIELD SERVICE COMMAND 17 "TRACE" CAN BE SET SO THAT IT WILL AUTOMATICALLY TYPE OUT THE BANK AND PATTERN NUMBERS AS EACH PATTERN IS RUN. (REFERENCE SECTION 2.4.4.8.18).

2.4.4.4 CONTROL "S" (STOP)

THIS COMMAND WILL STOP TYPEOUT (SOON) AND WILL WAIT FOR A CONTROL "Q".

2.4.4.5 CONTROL 'Q' (QUINTINUE)

THIS COMMAND WILL CONTINUE TYPING THAT HAS BEEN STOPPED BY CONTROL "S". IF THERE HAS BEEN NO CONTROL "S" TYPED THEN THIS COMMAND IS IGNORED.

2.4.4.6 CONTROL "F" (FIELD SERVICE MODE)

THIS COMMAND WILL CAUSE YOU TO ENTER A MODE WHICH LOOKS FOR SUB COMMANDS.

WHEN THE PROGRAM IS LOOKING FOR A SUB COMMAND ANY NUMBER THAT IS NOT A LEGAL COMMAND WILL CAUSE A MINI HELP MESSAGE TO BE TYPED. THEREFORE WHEN IN DOUBT TYPE 99 (CR) AND YOU WILL GET HELP.

NOTE

TYPING JUST CARRIAGE RETURN IS A DEFAULT COMMAND 0.

2.4.4.7.1 FIELD SERVICE COMMAND 0 (EXIT)

THIS COMMAND WILL EXIT FIELD SERVICES MODE AND RETURN TO WHATEVER TASK IT WAS IN PRIOR TO TYPING CONTROL 'F'. NOTE TYPING JUST CARRIAGE RETURN IS A DEFAULT COMMAND O.

2.4.4.7.2 FIELD SERVICE COMMAND 1 (READ CSR)

THIS COMMAND WILL TYPEOUT THE CONTENTS OF THE CSR.

IF THERE IS MORE THAN ONE CSR ON THE CPU (OR IF THE PROGRAM HAS NOT DETERMINED THE CSR STATUS YET), IT WILL ASK YOU "WHICH CSR(0-F)" TO WHICH YOU MUST RESPOND WITH AN HEXIDECIMAL NUMBER FROM 0 TO F. NOTE TYPING JUST CARRIAGE RETURN IS A DEFAULT 0.

IF THE CSR YOU SELECT CAUSES A TRAP TO 4 THE PROGRAM WILL TYPE "THIS CSR DOES NOT EXIST".

NOTE

CSR REFERENCES ARE DONE IN ACCORDANCE WITH SECTION 5.0.

2.4.4.7.3 FIELD SERVICE COMMAND 2 (LOAD CSR)

THIS COMMAND WILL ENABLE YOU TO LOAD THE CSR.

IF THERE IS MORE THAN ONE CSR ON THE CPU (OR IF THE PROGRAM HAS NOT YET DETERMINED THE CSR STATUS YET) IT WILL ASK YOU "WHICH CSR(0-F)" TO WHICH YOU MUST RESPOND WITH AN HEXIDECIMAL NUMBER FROM 0 TO F. NOTE TYPING JUST CARRIAGE RETURN IS A DEFAULT 0.

IF THE CSR YOU SELECT CAUSES A TRAP TO 4 THE PROGRAM WILL TYPE "THIS CSR DOES NOT EXIST".

THE CSR WILL BE READ AND DISPLAYED AS IN COMMAND 1.

THE PROGRAM WILL THEN ASK YOU FOR THE "CSR?" TO WHICH YOU MUST RESPOND WITH AN OCTAL NUMBER. NOTE TYPING JUST CARRIAGE RETURN IS A DEFAULT O.

THE PROGRAM WILL THEN LOAD THE CSR AND READ IT AGAIN DISPLAYING ITS NEW CONTENTS.

2.4.4.7.4 FIELD SERVICE COMMAND 3 (EXAMINE MEMORY)

THIS COMMAND WILL ALLOW YOU TO EXAMINE ANY PHYSICAL ADDRESS AND DOES THE NECESSARY MEMORY MANAGEMENT MAPPING FOR YOU.

THE PROGRAM WILL ASK YOU FOR THE "PHYSICAL ADDRESS (0-17757776)" TO WHICH YOU MUST RESPOND WITH AN OCTAL NUMBER.

IF THE ADDRESS ACCESS CAUSES A TRAP TO 4 THE PROGRAM WILL TYPE "TIMEOUT TRAP". IF THE ADDRESS ACCESS CAUSES A TRAP TO 114 THE PROGRAM WILL TYPE "PARITY ABORT".

THE CONTENTS OF YOUR PHYSICAL ADDRESS WILL BE TYPED.

2.4.4.7.5 FIELD SERVICE COMMAND 4 (MODIFY MEMORY)

THIS COMMAND ALLOWS YOU TO MODIFY ANY PHYSICAL ADDRESS AND DOES THE NECESSARY MEMORY MANAGEMENT MAPPING FOR YOU.

THE PROGRAM WILL ASK YOU FOR THE 'PHYSICAL ADDRESS (0-17757776)" TO WHICH YOU MUST RESPOND WITH AN OCTAL NUMBER.

IF THE ADDRESS ACCESS CAUSES A TRAP TO 4 THE PROGRAM WILL TYPE 'TIMEOUT TRAP'. IF THE ADDRESS ACCESS CAUSES A TRAP TO 114 THE PROGRAM WILL TYPE 'PARITY ABORT'.

THE PROGRAM WILL TYPE 'OLD DATA WAS' AND THE CONTENTS OF YOUR PHYSICAL ADDRESS.

THE PROGRAM WILL THEN TYPE "INPUT NEW DATA" TO WHICH YOU MUST RESPOND WITH AN OCTAL NUMBER. NOTE TYPING JUST CARRIAGE RETURN IS A DEFAULT O.

THE PROGRAM WILL ATTEMPT TO WRITE THIS NEW DATA INTO YOUR PHYSICAL ADDRESS AFTER WHICH IT WILL READ IT AGAIN AND TYPE 'DATA IS NOW' AND THE NEW CONTENTS OF YOUR PHYSICAL ADDRESS.

NOTE

IF YOU CAN'T CHANGE THE DATA, THAT WOULD INDICATE THAT YOU HAVE A DOUBLE BIT ERROR IN THAT DOUBLE WORD PAIR.

2.4.4.7.6 FIELD SERVICE COMMAND 5 (SELECT BANK TEST)

THIS COMMAND ALLOWS YOU TO RUN ANY BANK WITH ANY PATTERN FOREVER.

THE PROGRAM WILL ASK YOU 'BANK(0-167)" TO WHICH YOU MUST RESPOND WITH AN OCTAL NUMBER. IF THE BANK IS NOT ACCESSIBLE. THE PROGRAM WILL TYPE 'BANK NOT ACCESSIBLE" AND ASK QUESTION OVER.

THE PROGRAM WILL THEN ASK "TEST (0-47)" TO WHICH YOU MUST RESPOND WITH AN OCTAL NUMBER.

NOTE

ANY PATTERN CAN BE RUN INCLUDING THOSE THAT ARE NOT PART OF THE APT E-TABLE DEFAULTS (REFERENCE SECTION 6.2.1). IF YOU SELECT PATTERN 0, THE PROGRAM WILL ASK 'TEST 0 DATA IS?' TO WHICH YOU MUST RESPOND WITH AN OCTAL NUMBER.

'BANK REQUIRES RELOCATION' AND EXIT THIS COMMAND. NOTE NORMALLY THIS IS TRUE FOR BANK 0.

THE PROGRAM WILL THEN ARM THE CONSOLE KEYBOARD FOR INTERRUPTS AND TYPE "TO ESCAPE TYPE ANY KEY!".

THE TEST PATTERN WILL BE ENTERED AND RUN UNTIL A CONSOLE KEY IS DEPRESSED TO ESCAPE THIS LOOP.

2.4.4.7.7 FIELD SERVICE COMMAND 6 (TYPE CONFIGURATION MAP)

THIS COMMAND TYPES THE CONFIGURATION MAP.

THIS IS USEFUL AFTER A LONG RUN (OVERNIGHT) TO SEE ALL THE BANKS THAT ARE MARKED AS BAD. (ESPECIALLY IF YOUR CONSOLE IS A VIDEO TERMINAL).

FOR A DETAILED EXPLANATION OF THE MAP REFERENCE SECTION 7.3.

2.4.4.7.8 FIELD SERVICE COMMAND 7 (SOB-A-LONG TEST)

THIS COMMAND ALLOWS EXECUTION OF THE SOB-A-LONG TEST ON ALL NON-PROTECTED BANKS REFERENCE SECTION 6.2.2.26. OPERATION IS IDENTICAL TO COMMAND 5 EXCEPT THAT NO PATTERN OR BANK IS ENTERED AND EACH PASS CAUSES A BELL.

2.4.4.7.9 FIELD SERVICE COM 4D 8 (ERROR SUMMARY)

THIS COMMAND TYPES OUT THE JMBER OF PASSES AND THE TOTAL NUMBER OF ERRORS. IF THERE WERE / ERRORS IT WILL TYPE OUT THE BANKS AND THE NUMBER OF ERRORS PER BANK JP TO 255 DECIMAL.

THIS BECOMES USEFUL AFTER LONG RUNS (ALL NIGHT) ON SYSTEMS WITH A VIDEO CONSOLE TERMINAL.

2.4.4.7.10 FIELD SERVICE COMMAND 9 (REFRESH TEST)

THIS COMMAND ALLOWS EXECUTION OF THE REFRESH TEST ON ALL NON-PROTECTED BANKS REFERENCE SECTION 6.2.2.19. OPERATION IS IDENTICAL TO COMMAND 5 EXCEPT THAT NO PATTERN OR BANK IS ENTERED AND EACH PASS CAUSES A BELL.

2.4.4.7.11 FIELD SERVICE COMMAND 10 (SET FILL COUNT)

THIS COMMAND ALLOWS SETTING OF THE TERMINAL FILL COUNT (NECESSARY FOR LA30'S, ASR33'S, AND VT05'S). IT IS NORMALLY SET TO ZERO FOR LA36'S, VT52'S, VT100'S, ETC.

2.4.4.7.12 FIELD SERVICE COMMAND 11 (ENTER KAMIKAZE MODE)

THIS COMMAND ALLOWS YOU TO RUN PATTERNS THAT ARE NORMALLY NOT EXECUTED UNLESS UNDER APT OR ACT. THEY ARE USUALLY VERY TIME CONSUMING AND CAN RESULT IN FAILURES THAT ARE FATAL TO THE PROGRAM. IN EFFECT YOU ARE TRYING TO FIND A HARDWARE FAILURE REGARDLESS OF THE CONSEQUENCES. NOTE THAT MOST CRASHES DO NOT WIPE OUT THE DISPLAY INFORMATION WHICH IS TELLING YOU WHAT THE PROGRAM WAS DOING JUST PRIOR TO FAILURE. THERE ARE TWO WAYS TO DIE HERE - IMPATIENCE AND CRASHES.

2.4.4.7.13 FIELD SERVICE COMMAND 12 (EXIT KAMIKAZE MODE)
RETURN TO THE DEFAULT MODE OF TESTING (UNDO COMMAND 12).

2.4.4.7.14 FIELD SERVICE COMMAND 13 (TURN CACHE OFF)

THIS CHANGES THE CACHE CONSTANT TO BYPASS CACHE (REFERENCE SECTION 2.4.3.1).

2.4.4.8.15 FIELD SERVICE COMMAND 14 (TURN CACHE ON)
THIS CHANGES THE CACHE CONSTANT TO USE CACHE (REFERENCE SECTION 2.4.3.1).

2.4.4.7.16 FIELD SERVICE COMMAND 15 (TEST ONLY SELECTED BANKS)

THIS COMMAND ALLOWS YOU TO CENTER THE TEST EFFORT ON ONLY THOSE BANKS
THAT YOU ARE TROUBLESHOOTING. YOU MAY ALSO TEST BANKS THAT REQUIRE
RELOCATION AND WERE INACCESSABLE VIA COMMAND 5.

2.4.4.7.17 FIELD SERVICE COMMAND 16 (RESUME TESTING ALL BANKS)
RETURN TO THE DEFAULT MODE OF TESTING (UNDO COMMAND 15).

2.4.4.7.18 FIELD SERVICE COMMAND 17 (RESUME TESTING ALL BANKS)

ENABLE 'TRACE'. AFTER EXITING FIELD SERVICE MODE, THE PROGRAM WILL
TYPE OUT THE BANK AND PATTERN NUMBERS AS EACH PATTERN IS RUN.

2.4.4.7.19 FIELD SERVICE COMMAND 18 (RESUME TESTING ALL BANKS)
DISABLE "TRACE". (UNDO COMMAND 17).

```
2.5 EXECUTION TIMES
```

2.5.1 TYPICAL (SYSTEM) -

EXECUTION TIME DEPENDS ON MANY VARIABLES; HOWEVER HERE ARE SOME MEASURED TIMES ON AN 11/44 WITH CACHE:

128K WORDS OF MS11-L MEMORY
NORMAL PASS 0 MIN 50 SEC
QUICK VERIFY 0 MIN 50 SEC
KAMIKAZE MODE 10 MIN 5 SEC
KAMIKAZE QV 10 MIN 5 SEC

128K WORDS OF MS11-M MEMORY (NON-INTERLEAVED)
NORMAL PASS 2 MIN 25 SEC
QUICK VERIFY 1 MIN 0 SEC
KAMIKAZE MODE 11 MIN 0 SEC
KAMIKAZE QV 10 MIN 30 SEC

128K WORDS OF MS11-M MEMORY (INTERLEAVED)
NORMAL PASS 3 MIN 55 SEC
QUICK VERIFY 1 MIN 50 SEC
KAMIKAZE MODE 22 MIN 0 SEC
KAMIKAZE QV 20 MIN 5 SEC

512K WORDS OF MS11-P MEMORY NORMAL PASS 3 MIN 55 SEC QUICK VERIFY 3 MIN 25 SEC KAMIKAZE MODE 42 MIN 30 SEC KAMIKAZE QV 37 MIN 0 SEC

2.5.2 CALCULATIONS (SYSTEM)

NORMAL PASS
ADD
18 SEC PER 16K BANK OF NON-INTEREAVED MS11-M
ADD
15 SEC PER 16K BANK OF INTERLEAVED MS11-M
ADD
6 SEC PER 16K BANK OF MS11-L
ADD
22 SEC PER 64K BANK OF MS11-P

ADD 8 SEC PER 16K BANK OF NON-INTERLEAVED MS11-M
ADD 7 SEC PER 16K BANK OF INTERLEAVED MS11-M
ADD 6 SEC PER 16K BANK OF MS11-L
ADD 20 SEC PER 64K BANK OF MS11-P

KAMIKAZE MODE ADD 10 MIN. PER 128K WORDS FOR APPROXIMATE PASS TIMES.

1580		
1581		
1582 1583 1584 1585 1586 1587 1588 1589		
1584		
1585	2.5.3 TYPICAL (TEST	TS)
1586	TPAT TIME	DECCRIPTION
1587	TEST TIME	DESCRIPTION
1589		
1590	MT0000 :<1 SEC	DATA PATTERN TEST
1591	MT0001 :<1 SEC MT0002 :<1 SEC	ADDRESS TEST COMPLEMENT ADDRESS TEST
1592 1593	MT0003 : 1 SEC	3 XOR 9 WORST CASE NOISE TEST
1594 1595	MT0003 : 1 SEC MT0004 : 1 SEC MT0005 : 1 SEC	ROTATING ZEROS TEST
1595	MT0005 ; 1 SEC MT0006 ;<1 SEC	RUIATING UNES TEST
1597	MT0006 :<1 SEC MT0007 :<1 SEC	ADDRESS BIT TEST
1596 1597 1598 1599	MT0010 ;<1 SEC	3 XOR 9 WORST CASE NOISE TEST ROTATING ZEROS TEST ROTATING ONES TEST INITIAL DATA TEST ADDRESS BIT TEST BYTE ADDRESSING TEST
1599	MT0011 :<2 SEC MT0012 :<1 SEC	CKENIE SINGLE DII EKKUK 1ESI
1600	MT0012 :<1 SEC	CREATE DOUBLE BIT ERROR TEST
1601 1602	MT0013 : 1 SEC MT0014 : 1 SEC MT0015 : 1 SEC	BASIC DOUBLE BIT ERROR TEST
1603	MT0000 :<1 SEC MT0001 :<1 SEC MT0002 :<1 SEC MT0004 :1 SEC MT0005 :1 SEC MT0007 :<1 SEC MT0010 :<1 SEC MT0011 :<2 SEC MT0012 :<1 SEC MT0013 :1 SEC MT0014 :1 SEC MT0015 :1 SEC MT0017 :<1 SEC MT0017 :<1 SEC MT0020 :1 SEC MT0021 :1 SEC MT0022 :10 SEC MT0023 :10 SEC MT0025 :<1 SEC	WRITE INHIBIT OF BYTE WITH DBE WRITE INHIBIT OF WORD WITH DBE HOLDING 1'S O'S TEST
1604 1605	MT0016 :<1 SEC MT0017 :<1 SEC	HOLDING 1'S O'S TEST
1606	MT0020 ; 1 SEC	HOLDING 1'S 0'S TEST SYNDROMES TO CSR ON SINGLE BIT ERROR TEST MARCHING 0'S 1'S TEST
1607	MT0021 : 1 SEC	MARCHING O'S 1'S TEST
1608 1609	MT0022 :10 SEC MT0023 :10 SEC	REFRESH TEST SHIFTING DIAGONAL TEST
1610	MT0023 :10 SEC MT0024 :20 SEC	FAST GALLOPING PATTERN TEST
1611	MT0025 :<1 SEC	INTERRUPT ENABLE TEST
1612 1613	MT0026 :<1 SEC MT0027 : 1 SEC	RANDOM DATA TEST
1614	MT0030 : 1 SEC	UNIQUE BANK TEST FLUSH OUT DBE'S TEST
1615		SOB-A-LONG TEST
1616	MT0032 :<1 SEC MT0033 :35 SEC MT0034 :<1 SEC MT0035 :<1 SEC	BRANCH GOBBLE TEST
1618	MT0034 ;<1 SEC	SOFT ERROR TEST
1619	MT0035 :<1 SEC	WORST CASE PARITY TEST
1620	MT0037 :<1 SEC	CHECK ECC DISABLE TEST
1622	MT0041 : 1 SEC	ADDRESS TO CSR ON DBC TEST
1623	MT0042 :<1 SEC MT0043 :<1 SEC	EXTENDED ADDRESS TO CSR ON ERROR TEST
1616 1617 1618 1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1629	MT0031 : 3 SEC MT0032 :<1 SEC MT0033 : 35 SEC MT0034 :<1 SEC MT0035 :<1 SEC MT0037 :<1 SEC MT0041 : 1 SEC MT0042 :<1 SEC MT0043 :<1 SEC MT0044 : 1 SEC MT0045 :<1 SEC MT0047 :<1 SEC	SOB-A-LONG TEST WRITE RECOVERY TEST BRANCH GOBBLE TEST SOFT ERROR TEST WORST CASE PARITY TEST CORRECTION CORE TEST CHECK ECC DISABLE TEST ADDRESS TO CSR ON DBC TEST EXTENDED ADDRESS TO CSR ON ERROR TEST WRITE BYTE TEST SHIFTING CHECKBITS THROUGH CSR TEST SYNDROME BITS TO THE CSR ON A DBE TEST CHECK SINGLE BIT ERRORS WITH ECC DISABLED TEST NO CSR UPDATE WITH EXISTING DBE TEST
1626	MT0045 :<1 SEC	SYNDROME BITS TO THE CSR ON A DBE TEST
1627	MT0046 : 1 SEC	CHECK SINGLE BIT ERRORS WITH ECC DISABLED TEST
1628	MT0047 ;<1 SEC	MO COM OPPAIE MILIA EXTOLING DOE LEGI
1027		

1631	
1632	
1634	
1635	
1636	
1637	
1639	
1640	
1641	
1642	
1644	
1645	
1646	
1648	
1649	
1650	
1651	
1653	
1654	
1655	
1631 1632 1633 1633 1633 1633 1633 1633	
1658	
1659	
1660	
1662	
1663	
1664	
1666	
1667	
1668	
1669	
1671	
1672	
1673	
1674	
1676	
1677	
1678	
1679 1680 1681	
1681	
1682	

- 3.0 ERROR INFORMATION
- 3.1 ERROR REPORTING

MOST ERRORS ARE REPORTED USING THE EMT TRAP AND HANDLER PROVIDED BY SYSMAC.SML. MOST ERRORS WILL BE OF THE 'MEMORY DATA ERROR' TYPE WHICH WILL BE DESCRIBED HERE. MEMORY DATA ERRORS WILL ALSO CAUSE THE BANK TO BE MARKED AS BAD IN THE CONFIGURATION TABLE.

OTHER ERRORS ARE BEST EXPLAINED BY REFERENCING THE SPECIFIC TYPEOUT AND IF NECESSARY THE PROGRAM LISTING.

EXAMPLE 1: MEMORY DATA ERROR PC 022132 022132 022132 022132 022132 XCR CSR MTYP INT PAT BANK VADD BAD PADD 06 06 06 06 000000 000100 000100 03700006 060006 000100 000100 03700006 000000 060006 000000 000100 03700006 000100 060006 000000 000100 000100 03700006 060006

WHILE TESTING BANK 37 AT VIRTUAL ADDRESS 60006 (VIRTUAL ADDRESSES ARE ALWAYS BETWEEN 60000 AND 157776 FOR MAPPING PURPOSES), PHYSICAL ADDRESS 3700006 (THAT'S BANK 37 PHYSICAL 6 WITHIN THE BANK) WITH PATTERN 6 (INITIAL DATA TEST), THE GOOD DATA EXPECTED WAS 0 BUT THE DATA ACTUALLY READ (BAD) WAS 100, THE EXCLUSIVE OR AT GOOD BAD YIELDS 100 WHICH INDICATES ONLY FAILING BIT(S) (BIT 6). IT IS AN MS11-P (ECC) MEMORY AND IT'S NOT INTERLEAVED. THE CSR IS LOCATED AT 172000.

EXAMPLE 2: MEMORY DATA ERROR PC 022132 022132 022132 INT PAT GOOD BAD XOR CSR MTYP BANK VADD PADD 03500000 03500002 000001 000001 06 000000 0 060000 000100 060002 000000 000100 03500006 000000 000100 000100

WHILE TESTING BANK 35, VIRTUAL ADDRESS 60000, PHYSICAL ADDRESS 3700000 WITH PATTERN 6 (INITIAL DATA TEST), THE GOOD DATA EXPECTED WAS 0 BUT THE DATA ACTUALLY READ (BAD) WAS 1, THE EXCLUSIVE OR AT GOOD BAD YIELDS 1 WHICH INDICATES ONLY FAILING BIT(S) (BIT 0). IT IS AN MS11-M (ECC) MEMORY AND IT'S INTERLEAVED; SO SINCE ADDRESS BIT 1 WAS NOT ASSERTED, THE CSR IS LOCATED AT 172000.

WHILE ALSO IN BANK 35, VIRTUAL ADDRESSES 60002 AND 60006 WERE EXPECTED TO HAVE 0, BUT THE DATA READ WAS 100, THE EXCLUSIVE OR OF GOOD BAD YIELDS 100 WHICH INDICATES ONE FAILING BIT (BIT 6). SINCE IT IS INTERLEAVED MS11-M MEMORY, AND ADDRESS BIT 1 IS ASSERTED, THE CSR IS LOCATED AT 172102 (CSR NUMBER 1 UNDER THE INT COLUMN)

NOTE

SUBSEQUENT ERRORS OF THE SAME TEST DO NOT TYPE A NEW HEADING.

```
1685
1686
1687
1688
1689
1690
                                                                      3.2 ERROR ABBREVIATIONS
                                                                      THE FOLLOWING IS A LIST OF ALL ABBREVIATIONS USED IN ERROR REPORTS.
                                                                                                     NUMBER OF ERRORS THAT WERE DETECTED.
1691
1692
                                                                      # OF ERRORS
                                                                                                    FIRST ADDRESS THAT FAILED.
THE ARRAY NUMBER THAT WAS LOCKED UP IN THE MS11-M CSR.
THE # OF CPU'S APT EXPECTS ON THE SYSTEM.
APT CORE SIZE.
                                                                      1ST ADD
1693
                                                                      ARRAY
1694
1695
1696
1697
1698
1699
                                                                      APT#
                                                                       APTCORE
                                                                                                     APT MOS SIZE.
                                                                      APTMOS
                                                                                                     BAD DATA.
                                                                      BAD
                                                                                                     BAD WORD #1 OF A DOUBLE WORD DATA VALUE.
BAD WORD #2 OF A DOUBLE WORD DATA VALUE.
                                                                      BAD-WD1
                                                                      BAD-WD2
1700
1701
1702
1703
1704
1705
                                                                                                     BAD CHECK CODE BITS.
                                                                      BAD-CHK
                                                                                                     THE BANK NUMBER. BANKS ARE 16K WORDS LONG.
                                                                      BANK
                                                                                                    THE BANK NUMBER. BANKS ARE TOK WORDS LO
BAD CHECK CODE BITS.
THE 7 BIT VALUE OF THE CHECK CODE BITS.
THE CACHE CONTROL REGISTER.
CPU ERROR REGISTER.
CONTROL AND STATUS REGISTER.
CSR NUMBER (0-F HEXIDECIMAL).
THE CACHE DATA REGISTER.
DOUBLE BIT ERROR (UNCORRECTABLE ERROR).
DEVICE ADDRESS.
ERROR CORRECTABLE CODE.
GOOD CHECK CODE BITS.
                                                                       BD-CC
                                                                       CHKBITS
                                                                       CONTRL
                                                                       CPUERR
                                                                       CSR
1707
                                                                       CSRNO
                                                                       DATARG
1709
1710
                                                                       DBE
                                                                       DEV ADD
                                                                       ECC
1711
                                                                                                     GOOD CHECK CODE BITS.
                                                                       GD-CC
                                                                                                     GOOD CHECK CODE BITS.
GOOD WORD #1 OF A DOUBLE WORD DATA VALUE.
GOOD WORD #2 OF A DOUBLE WORD DATA VALUE.
                                                                       GD-CHK
                                                                       GD-WD1
1715
                                                                       GD-WD2
1716
1717
1718
1719
                                                                       GOOD
                                                                                                     GOOD DATA.
                                                                                                     INTERLEAVED (ADDRESS BIT 1 ASSERTED) CSR NUMBER.
                                                                       INT
                                                                                                    INTERLEAVED (ADDRESS BIT 1 ASSERTED) CSR NUMBER.
MS11-L SIZE.
MEMORY ERROR REGISTER.
MEMORY MANAGEMENT REGISTER #0.
MEMORY MANAGEMENT REGISTER #1.
MEMORY MANAGEMENT REGISTER #2.
MEMORY MANAGEMENT REGISTER #3.
MS11-M SIZE.
MEMORY TYPE (MS11-L, MS11-M, OR MS11-P).
PHYSICAL ADDRESS (ASSERTED BY THE PROGRAM AFTER MAPPING).
                                                                       LSIZE
                                                                       MEMERR
                                                                       MMRO
MMR1
                                                                       MMR2
MMR3
                                                                       MSIZE
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
                                                                       MTYP
                                                                       PADD
                                                                                                     PATTERN NUMBER.
                                                                       PAT
                                                                                                     PROGRAM COUNTER AT THE TIME THE ERROR OCCURRED.
                                                                       PC
                                                                                                     SINGLE BIT ERROR (CORRECTABLE ERROR).
                                                                                                     VIRTUAL ADDRESS (ASSERTED BY THE PROGRAM BEFORE MAPPING).
                                                                       VADD
                                                                                                     THE DATA THAT WAS WRITTEN INTO THE 1ST HALF OF A DOUBLE WORD. THE DATA THAT WAS WRITTEN INTO THE 2ND HALF OF A DOUBLE WORD.
                                                                       WROTE1
                                                                       WROTE2
                                                                                                     EXCLUSIVE OR OF THE GOOD AND BAD DATA. SHOWS THE BAD BITS.
                                                                       XOR
                                                                       AUT
                                                                                                     ADDRESS UNDER TEST
```

3.3 ERROR HALTS

THERE ARE SEVERAL HALTS IN THE PROGRAM.

ALL UNUSED TRAP VECTORS CONTAIN A TRAP CATCHER (.WORD .+2, HALT).

AN UNDEFINED TRAP INSTRUCTION HALTS AT SYMBOLIC LOCATION 'SHALT2'.

THE APT DOWN LOAD SEQUENCE WILL HALT AT SYMBOLIC LOCATION "APTHLT".

HALT ON ERROR OPTION (SW15 SET) AT SYMBOLIC LOCATION "SHALT".

HALT PROGRAM (SW8 SET) AT SYMBOLIC LOCATION 'SEXHALT'.

POWER FAIL WILL NORMALLY HALT AT THE END OF THE SHUT DOWN SEQUENCE (SYMBOLIC LOCATION 'SDOWN').

POWER FAIL HAS A FATAL HALT AT SYMBOLIC LOCATION "SILLUP" WHICH CAN BE CAUSED BY POWER UP OCCURRING BEFORE POWER DOWN SEQUENCE COMPLETED OR BY POWER DOWN BEFORE A POWER UP SEQUENCE IS COMPLETED.

4.0 PROGRESS REPORTS

PASS COMPLETE TYPEOUTS AS FOLLOWS:

END PASS # 1 END PASS #QV

NOTE

PASS 2 WAS FLAGGED AS A QUICK VERIFY PASS. (BECAUSE OF A CHANGE IN SW5)

TO OBTAIN PROGRESS REPORTS WHILE EXECUTING, TYPING A CONTROL "T" WILL PRINT OUT THE INFORMATION ENCODED IN THE DISPLAY REGISTER. EXAMPLE:

BANK= 2 TEST= 34

REFERENCE SECTION 2.4.4.7.18 FOR MORE INFORMATION ON TRACING.

5.0 CSR INFORMATION TABLES

THE FOLLOWING IS A PICTURE VIEW OF THE CURRENT CONTROL STATUS REGISTERS WHICH CAN BE TESTED BY THIS PROGRAM. IT SHOWS BIT ASSIGNMENTS AND DEFINITIONS TO PROVIDE A HANDY REFERENCE, AND SHOWS THE SIMILARITIES AND DIFFERENCES BETWEEN EACH ONE:

NOTE

ALL UNUSED BITS IN EACH CSR ARE EQUAL TO ZERO.

1815	
1816 1817	
1817	
1818	
1820	
1821	
1822	(
1823	
1819 1820 1821 1822 1823 1824 1825 1826 1827 1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838	
1825	
1820	
1828	
1829	
1830	(
1831	
1832	
1833	
1834	
1836	
1837	
1838	(
1839	
1840	
1841	
1842 1843	
1844	
1845	
1846	
1847	
1848	
1849	
1850	
1852	
1853	
1854	
1855	
1856	
1857	
1850	
1860	
1861	
1851 1852 1853 1854 1855 1856 1857 1858 1859 1860 1861 1862 1863 1864 1865 1866 1867 1868	
1863	
1864	
1865	
1867	
1868	
1000	

	MC	44	1_	0	-	C	0
5.1	MЭ	ш		P	L	3	ĸ

(1)	I I I I I I I I I I I I I I I I I I I
	15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
(11)	IDE EA SI I I I I I I I I I I I I I I I I I I
	15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
(111)	I I I I I I I I I I I I I I I I I I I
	15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00

BIT ASSIGNMENTS ARE DEFINED AS FOLLOWS:

BIT15 UNCORRECTABLE ERROR
ON A READ TO MEMORY (ECC DISABLE
BIT = 0), THIS BIT IS SET IF
A DOUBLE ERROR OCCURS. THE
ERROR ADDRESS IS STORED IN THE
CSR. SETTING THIS BIT ALSO
TURNS ON A RED LED AT THE REAR OF
THE CARD FOR A VISUAL INDICATION.
THIS BIT IS ALSO SET IN ECC
DISABLE MODE IF A SERR OR DERR
OCCURS.

BIT14 EUB ERROR ADDRESS WITH BIT 14 = 1, A READ TO THE CSR WILL FETCH ADDRESS A21 THROUGH A18. WHEN BIT 14 = 1, DIAGNOSTIC DATA MAY NOT BE LOADED INTO THE SYNDROME REGISTER.

BIT13 SET INHIBIT MODE
WHEN THIS BIT IS SET TO
A "1", IT ENABLES THE INHIBIT
MODE POINTER TO INHIBIT
EITHER THE FIRST OR SECOND
16K FROM EVER GOING INTO THE
DIAG CHECK OR ECC DISABLE
MODE.

BITSO5-10 CHECK BIT STORAGE (CSR II)
CHECK BIT STORAGE (DIAG CK
BIT 2 = 1)
WHEN IN THE DIAGNOSTIC CHECK
MODE THESE BITS ARE USED TO STORE
THE CHECK BITS TO BE WRITTEN
INTO MEMORY OR THE CHECK BITS
READ FROM MEMORY. IF A DOUBLE
ERROR OR SINGLE ERROR OCCURS
WHEN IN THE DIAGNOSTIC CHECK
MODE AND ECC DISABLE BIT 1 = 0,
THEN THE CHECK BITS ARE STORED
IN THE CSR TOGETHER WITH
THE DOUBLE OR SINGLE ERROR
BIT. THESE BITS ARE WRITEABLE
IN DIAGNOSTIC MODE. A "1"
IS STORED IN BIT 11 IF CSR
02, CSR 13, AND CSR 14 ARE
SET TO INDICATE THAT THE
MEMORY UNDER TEST IS A MS11-P.

BITS05-11 UNIBUS ADDRESS STORAGE (CSR I) (DIAG CK BITS 2 = 0, ECC DISABLE BIT 1 = 0)

IF A DOUBLE OR SINGLE ERROR
OCCURS ON A READ CYCLE, THEN
ADDRESS BITS ALL THROUGH
A17 ARE STORED IN THESE BITS
THESE BITS ARE READ ONLY ON
THE CONDITION THAT SERR (CSR 4)
OR DERR (CSR 15) IS SET BUT
CSR 14 IS NOT SET.

EUB ADDRESS STORAGE (DIAG CK BIT 2 = 0), ECC DISABLE BIT 1 = 0 OR 1).

IF A DOUBLE OR SINGLE ERROR OCCURS ON A READ CYCLE, ADDRESS BITS A17 THROUGH A11 ARE STORED IN CSR BITS 11 THROUGH 5 AND ADDRESS BITS A21 THROUGH A18 ARE

1978

STORED IN A BACKUP REGISTER.
THE EUB ERROR ADDRESS
RETRIEVAL BIT (CSR 14) IS
USED TO OBTAIN THE TOTAL
ERROR ADDRESS AS FOLLOWS:

WITH CSR BIT 14 = 0 A READ TO THE CSR WILL OBTAIN A17 THROUGH A11 FROM CSR BITS 11 THROUGH 5.

CSR EIT 14 CAN THEN BE SET TO A "1" AND A READ TO THE CSR WILL THEN READ A21 THROUGH A18 FROM CSR BITS 8 THROUGH 5 AND 0'S FROM CSR BITS 11 THROUGH 9.

ADDRESS BITS A21 THROUGH A11 ARE OBTAINED TO LOCATE THE DOUBLE ERROR TO A 1K SEGMENT OF MEMORY.

THE EUB ADDRESS A21 THROUGH A18 IS READ ONLY WHENEVER CSR 14 = 1.

BIT05-10 SYNDROME STORAGE (CSR III)
IF A DOUBLE OR SINGLE ERROR
OCCURS ON A READ OR WRITE
BYTE CYCLE, AND IF CSR BIT
2 IS SET TO A 'O'' SYNDROME
BITS X, 0, 1, 2, 4 AND 8
AND STORED IN CSR BITS 5
THROUGH 10. TO READ THE
SYNDROME BITS FROM CSR, BIT
YOU MUST READ THE ERROR
ADDRESS, THEN SET 2 OF
THE CSR MUST BE ST TO
A '''' (DIAGNOSTIC MODE) AND
THE CSR READ AGAIN. THIS OPERATION
WILL ALLOW SYNDROME BITS
FOR A SINGLE OR DOUBLE
FAILURE TO BE READ INSTEAD
OF THE ADDRESS BITS NORMALLY
READ WHEN CSR 02 IS SET TO 'O''.

BIT04 SINGLE ERROR
IF ON A READ TO MEMORY A
SBE OCCURS, THE ERROR
ADDRESS A21-A11 AND
THE ERROR SYNDROMES WILL

BE LOGGED IN CSR BITS 5-11
UNLESS THE UNCORRECTABLE ERROR
CSR 15 IS SET. THE ERROR
ADDRESS WILL BE LOGGED
UNCONDITIONALLY IN THE ECC
DISABLE MODE. THIS BIT IS NOT
SET IF INHIBIT MODE (BIT
13 = 1) IS SET AND DIAGNOSTIC
MODE (BIT 02 = 1) IS SET.

BIT03 INHIBIT MODE POINTER
THIS BIT WORKS IN CONJUNCTION
WITH THE SET INHIBIT MODE
(BIT 13). WHEN BIT 13 IS SET
TO A 1, A 16K PORTION OF
MEMORY IS INHIBITED FROM
OPERATING IN THE ECC DISABLE
MODE OR DIAGNOSTIC CHECK MODE.

THE INHIBIT MODE POINTER INDICATES WHICH 16K IS BEING INHIBITED, I.E., BIT 3 = 0 THE FIRST 16K OF MEMORY IS INHIBITED, BIT 3 = 1, THE SECOND 16K OF MEMORY IS INHIBITED.

WITH BIT 13 SET TO A 0, BIT 3 BECOMES INOPERATIVE.

BIT03, IN CONJUNCTION WITH
BIT 13, THEREFORE ALLOWS A 16K
CHUNK OF MEMORY TO ALWAYS HAVE
ECC COVERAGE. THE SYSTEMS
DIAGNOSTIC CAN THEREFORE
RESIDE IN THIS PROTECTED
PORTION OF MEMORY AND CAN
DISABLE ECC AND/OR RUN THE
DIAGNOSTIC CHECK MODE IN THE
REST OF MEMORY WITHOUT ITSELF
BECOMING VULNERABLE TO SINGLE
ERRORS. THIS BIT IS A READ/WRITE
BIT RESET BY POWER UP AND BUS
INIT.

BITO2 DIAGNOSTIC CHECK MODE THIS MODE ALLOWS A MEANS OF FORCING A SINGLE OR DOUBLE ERROR IN A DESIRED LOCATION. IT ALSO PROVIDES A MEANS OF EXAMINING THE CHECK BITS AND THE SYNDROME IN A GIVEN LOCATION.

THE CHECK BITS DESIRED FOR A GIVEN DATA PATTERN ARE WRITTEN INTO BITS 5 THROUGH 11 OF THE CSR. A WORD OR WRITE BYTE MEMORY WILL WRITE THE CHECK BITS FROM THE CSR TO THE MOS ARRAY (CSR 2 = 1) INSTEAD OF THE CHECK BITS GENERATED ON THE DATA TO BE WRITTEN. SINGLE ERRORS ON THE READ PORTION OF THE DATOB CYCLE ARE CORRECTED.

A READ TO THE MEMORY WILL READ THE CHECK BITS STORED IN MEMORY AND CLOCK THEM INTO THE CSR.

IF A DOUBLE ERROR OR SINGLE ERROR OCCURS THE DERR OR SERR BIT IN THE CSR IS SET AND THE ERROR SYNDROME BITS READ FROM ECC ARE STORED IN CSR BITS 10-5 AS WELL AS THE ADDRESS BITS. IN DIAGNOSTIC CHECK MODE THE ERROR SYNDROME BITS WILL BE READ WHEN CSR BITS 10-5 ARE READ.

THIS BIT IS A READ/WRITE BIT AND IS RESET ON POWER UP AND BUS INIT.

BIT01 DISABLE CORRECTION MODE
IF THIS BIT IS SET, NO SINGLE
ERRORS WILL BE CORRECTED. A
SINGLE ERROR WILL SET CSR 4
AND CSR 15 OR A DOUBLE ERROR
WILL SET CSR 15 AND ASSERT
BUS PBL IF CSR 00 IS ASSERTED.
THE 1K BLOCK OF ADDRESS WHERE
THE ERROR OCCURS WILL ALSO
RE STORED IN THE CSR. THE
PRIORITY OF A SERR AND DERR
WILL BE THE SAME, I.E., THE
LAST ERROR INFORMATION WILL
ALWAYS BE STORED UNLESS A DERR
PRECEDES A SERR. IF A DOUBLE
ERROR OCCURS DURING A WRITE BYTE
CYCLE, THE WRITE PORTION OF
THE CYCLE WILL NOT BE ABORTED.
THE CHECK BITS WRITTEN WILL

HAVE BEEN GENERATED ON THE DATA WRITTEN. THIS MEANS THAT IF A SINGLE OR DOUBLE ERROR EXISTED IN THE LOCATION ACCESSED, IT WOULD BE CLEARED (UNLESS THE ERRORS WERE HARD).

THIS BIT IS A DIAGNOSTIC AID TO ALLOW WRITING AND READING DATA FROM MEMORY WITHOUT INTERFERENCE FROM THE ERROR CORRECTION LOGIC.

BITOO UNCORRECTABLE ERROR INDICATION ENABLE IF A DOUBLE ERROR OCCURS WITH ECC ENABLED OR A SINGLE ERROR OR DOUBLE ERROR WITH ECC DISABLED, ON A READ CYCLE TO THE MEMORY AND THIS BIT IS SET, THEN BUS PBL WILL BE ASSERTED.

5.2 MS11-L CSR

III	I	I	I	I	I	I	I	I	I	I	I	I	I	I
!PE!EU!	Ţ	Ī		A	DDR	ESS			!	!	! W	P!	!A	E!
III	I	I	I	I	I	I	I	I	I	I	I	I	I	I

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00

BIT ASSIGNMENTS ARE DEFINED AS FOLLOWS:

BIT15 PARITY ERROR

BIT14 EUB ERROR
RETRIEVAL IF THE
MEMORY IS ON AN
EXTENDED UNIBUS, WHEN
BIT14 IS ZERO, THE LOW
ORDER FAILING
ADDRESSES ARE
AVAILABLE (BITS
11-17); WHEN BIT14 IS
ONE, THE HIGH ORDER
FAILING ADDRESSES ARE
AVAILABLE (BITS 18-21
OF ADDRESS). IF THE
MEMORY IS ON A UNIBUS,
A JUMPER DISABLES THIS
BIT SO THAT IT IS READ
ONLY, AND EQUAL TO ZERO.

BITS 11-5 ERROR ADDRESS WITH BIT14 SET, THEY CONTAIN THE HIGH ORDER PARITY ERROR ADDRESS (BITS 21-18 OF ADDRESS); WITH BIT14 CLEARED, THEY CONTAIN THE LOW ORDER PARITY ERROR ADDRESS (BITS 17-11 OF ADDRESS).

BITO2 WRITE WRONG PARITY (ODD) WHEN CLEAR; OTHER PARITY (EVEN) WHEN SET.

BITOO ACTION ENABLE NO ACTION WHEN CLEAR; TRAP TO VECTOR 114 WHEN SET.

5.3 MS11-M CSR

III	1	I	I	I	I	I	I	I	I	I	I	I	1	1
!DE!EU!	SI!	!		A	DDR	ESS			!S	E!I	P!D	CIE	C!E	E!
III	I	I	I	I	I	I	I	I	I	I	I	1	I	1

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00

BIT ASSIGNMENTS ARE DEFINED AS FOLLOWS:

BIT15 UNCORRECTABLE ERROR THIS BIT IS SET IF A DBE OCCURS, AND THE ERROR ADDRESS IS STORED IN THE CSR. THIS BIT IS ALSO SET IN THE ECC DISABLE MODE IF AN SBE OR DBE OCCURS.

BIT14 EUB ERROR
RETRIEVAL IF THE MEMORY IS ON AN EXTENDED
UNIBUS, WHEN BIT14 IS
ZERO AND EITHER BIT4
OR BIT 15 IS A ONE,
THE LOW ORDER FAILING
ADDRESSES ARE AVAILABLE (BITS 11-17); WHEN
BIT14 IS ONE, THE HIGH
ORDER FAILING ADDRESSES ARE AVAILABLE (BITS
18-21 OF ADDRESS). IF
THE MEMORY IS ON A
UNIBUS, A JUMPER DISABLES THIS BIT SO THAT
IT IS READ ONLY, AND
EQUAL TO ZERO.

BIT13 SET INHIBIT MODE
WHEN THIS BIT IS SET
TO A 1, IT ENABLES THE
INH MODE POINTER TO
INHIBIT EITHER THE
FIRST OR SECOND 16K
FROM EVER GOING INTO
THE DIAG. CHECK OR
ECC DISABLE MODE.
WHEN THIS BIT IS SET
TO A 0, IT ALLOWS THE
DIAG. CHECK MODE

AND/OR ECC DISABLE MODE TO OPERATE OVER THE ENTIRE MEMORY ON THE BOARD.

BITS 11-5 ERROR ADDRESS WITH BIT02 CLEARED AND BIT14 SET, THEY CONTAIN THE HIGH ORDER ERROR ADDRESS (BITS 21-18); WHEN BIT02 AND BIT14 ARE CLEARED, THEY CONTAIN THE LOW ORDER ERROR ADDRESS (BITS 17-11); WHEN BIT02 IS SET THEY CONTAINS CHECK BITS FOR ECC.

BIT04 SINGLE ERROR SET WHENEVER SINGLE ERROR OCCURS.

BITO3 INHIBIT MODE
POINTER THE INHIBIT
MODE POINTER WORKS IN
CONJUNCTION WITH THE
SET INHIBIT MODE BIT.
WHEN BIT13 IS SET TO A
1, A 16K PORTION OF
MEMORY IS INHIBITTED
FROM OPERATING IN THE
ECC DISABLE MODE OR
DIAGNOSTIC CHECK MODE.
THE INHIBIT MODE
POINTER INDICATES
WHICH 16K IS BEING INHIBITED; E.G.-IF BIT3
=1, THE SECOND 16K OF
MEMORY IS INHIBITTED.
WHEN BIT13 IS SET TO A
0, BIT3 BECOMES
INOPERATIVE.

BITO2 DIAGNOSTIC CHECK MODE WHEN SET ENABLES READ-WRITE OF CHECK BITS(SEE BITS 11-5). IF A DBE OCCURS IN THIS MODE (WITH BIT1=0), BIT15 IS SET, BUT THE CHECK BITS READ

ARE STORED IN BITS 11-5, NOT THE DBE ADDRESS BITS.

BIT01 DISABLE ERROR CORRECTION WHEN SET NO SINGLE ERROR CORRECTION TAKES PLACE. A SINGLE BIT ERROR WILL SET BIT04 AND BIT15 AND ASSERT BUS PBL LIF BIT00 IS ASSERTED; A DOUBLE ERROR WILL SET SET BIT15 AND ASSERT BUS PBL LIF BIT00 IS ASSERTED. THE ERROR ADDRESS IS STORED IN THE CSR, AND CORRECT CHECK BITS ARE GENERATED AND STORED ON A WRITE.

BITOO UNCORRECTABLE ERROR ENABLE WHEN SET ENABLES TRAP TO VECTOR 114 ON UNCORRECTABLE ERROR.

2711	
2211	
2312	
2317	
2315	
2316	
2317	
2318	
2319	
2320	
2327	
2323	
2324	
2325	
2326	
2327	
2328	
2329	
2330	
2313 2314 2314 2314 2314 2314 2314 2314	
2333	
2334	
2335	
2336	
2337 2338 2339 2340 2341	
2330	
2340	
2341	
2342	
2343	
2344	
2345	
2314 2314 2314 2314 2316 2316 2316 2316 2316 2316 2316 2316	
2342 2343 2344 2345 2346 2347	
2349	
2350	
2351	
2352	
2353	
2324	
2333	

6.0 SUB-TEST SUMMARIES

6.1 TESTS

TEST 1

BIT TEST OF ALL CSR'S/MATCH ALL CSR'S WITH MEMORY (CSR ACCESS MAY CAUSE WRONG TYPE OF TRAPS)

TEST 2

TEST BANK O ACCESSES FAILURES ARE FATAL.

TEST 3

TEST BANKS 1-167 (OCTAL) FOR ZEROS AND ONES ERRORS ARE NOT TYPED HERE - ONLY LOGGED IN THE CONFIGURATION TABLE

TEST 4

ECC INHIBIT MODE POINTER TEST

TEST 5

DIAGNOSTIC MODE DISPATCH ROUTINE THIS TEST RUNS ALL THE PATTERNS IN THE MODE SELECTED.

TEST 6

UNIQUE BANK TEST PATTERN 27 IS RUN

6.2 TESTS

6.2.1 GENERAL TEST INFORMATION

ACTUAL TESTS ARE IDENTIFIED BY SYMBOLIC LOCATIONS 'MTPXYY' WHERE X MAY BE ANY SUB PROGRAM INDICATOR (A,B,C,ETC) OR O AND YY WILL BE THE NUMBER OF THE TEST.

SETUP PROCEDURES FOR EACH TEST ARE IDENTIFIED BY SYMBOLIC LOCATIONS 'MTOOYY' WHERE YY WILL BE THE NUMBER OF THE TEST.

TESTS RESIDE IN 4 SCRIPTS THAT ARE SCANNED FOR EXECUTION. SYMBOLIC LOCATION 'MKCSRT' IS A TABLE OF TESTS THAT CAN RUN ONCE FOR EACH ECC BANK (TWICE FOR INTERLEAVED MS11-M'S). SYMBOLIC LOCATION 'MKPAT' IS A TABLE OF TESTS THAT CAN RUN ON EACH BANK OF ECC MEMORY. SYMBOLIC LOCATION 'MJPAT' IS A TABLE OF TESTS THAT CAN RUN ON EACH BANK OF PARITY MEMORY. SYMBOLIC LOCATION 'FSPAT' IS A TABLE OF TESTS THAT CAN BE RUN IN FIELD SERVICE MODE (COMMAND 5).

THE 1ST 3 SCRIPTS ARE COMPLETELY CONTROLLED BY THE APT E-TABLE (EVEN IF NOT RUNNING UNDER APT). MODIFICATIONS TO THIS TABLE CAN BE MADE (1) WITH APT, OR (2) MANUALLY.

EXAMPLE E-TABLE SEGMENT:

:THE FOLLOWING LOCATIONS SPECIFY WHICH TESTS :ARE TO BE RUN FOR PARTICULAR MEMORIES

REFERENCE THE TABLE LISTED BELOW TO RELATE BITS TO TESTS. BITO SET WILL RUN THE FIRST ENTRY IN THE TABLE, BITO SET IN THE SECOND WORD WILL RUN THE 17TH ENTRY IN THE TABLE...

:NOTE**NULL TESTS DO NOT TAKE ANY TIME

\$DDW0: \$DDW1: \$DDW2: \$DDW3: \$DDW4: \$DDW4:	. WORD . WORD . WORD . WORD	177777 177777 177777 177777 177777	ECC CSR TESTS ECC CSR TESTS ECC TESTS ECC TESTS PARITY TESTS PARITY TESTS	177777 TABLE = MKCSRT: 177777 TABLE = MKCSRT: 103777 TABLE = MKPAT: 177777 TABLE = MKPAT: 003777 TABLE = MJPAT: 177774 TABLE = MJPAT:
\$DDW5:	. WORD	177777	;PARITY TESTS	1////4 TABLE = MJPAT:

RECOMMENDED VALUE

6.2.2 SPECIFIC TESTS

6.2.2.1 TEST 0 BASIC DATA TEST

WRITES READS R2 INTO A 16K BANK.

THIS IS USED FOR ZEROS AND ONES TESTING AND IN FIELD SERVICE MODE FOR ANY CONSOLE SELECTED TEST.

IT CAN EXECUTE OUT OF THE USER INSTRUCTION PAR'S.

NOTE

IT IS FREQUENTLY MODIFIED DYNAMICALLY SUCH THAT (1) IT RETURNS AFTER WRITING ONLY (THE 1ST NOP IS REPLACED WITH A RETURN) OR (2) IT ONLY COUNTS ERRORS (THE CODE PERROZ AND NOP ARE REPLACED WITH INC ampatern).

6.2.2.2 TEST 1 ADDRESS TEST

WRITES READS AN INCREMENTING PATTERN EQUIVALENT TO PHYSICAL ADDRESSED INTO A 16K BANK.

IT CAN EXECUTE OUT OF THE USER INSTRUCTION PAR'S.

6.2.2.3 TEST 2 COMPLEMENT ADDRESS TEST

WRITES THE COMPLEMENT OF THE PHYSICAL ADDRESS FROM HIGH ADDRESSES TO LOW (WRITE DOWN) AND READS FROM LOW ADDRESSES TO HIGH (READ UP).

THIS PROVIDES THE COMPLEMENT OF THE COVERAGE OF TEST 1 IN BOTH DATA PATTERN AND ADDRESSING SEQUENCE.

IT CAN EXECUTE OUT OF THE USER INSTRUCTION PAR'S.

6.2.2.4 TEST 3 3 XOR 9

WRITES READS A TEST THAT COMPLEMENTS AS ADDRESS BITS 3 AND 9 CHANGE.

THIS TEST IS RUN 4 TIMES (1) WITH ZEROS ONES, (2) WITH ONES ZEROS, (3) WITH 401 ONES, AND (4) WITH ONES 401. THE TEST OF THE 401 IS TO FORCE A THE PARITY BITS TO BECOME INVOLVED.

IT CAN EXECUTE OUT OF THE USER DATA PDR'S, THE USER INSTRUCTION PAR'S, THE KERNEL DATA PAR'S AND THE SUPERVISOR DATA PAR'S.

6.2.2.5 TEST 4 ROTATING ZEROS TEST

WRITES A BACKGROUND PATTERN OF ONES. ROTATES A ZERO CARRY BIT LEFT THRU EACH PAR OF BYTES (18 TIMES) AND THEN CHECKS THAT THE CARRY IS ZERO AND THE WORD (2 BYTES) IS STILL ALL ONES.

IT CAN EXECUTE OUT OF THE USER DATA PAR'S AND THE KERNEL DATA PAR'S.

NOTE

IT IS NOT UNCOMMON TO OBSERVE THE GOOD DATA EQUAL TO THE BAD DATA. THIS INDICATES THAT THE CARRY WAS NOT CLEAR AFTER 18 ROLB'S.

6.2.2.6 TEST 5 ROTATING ONES TEST

WRITES A BACKGROUND PATTERN OF ZEROS. ROTATES A ONE CARRY BIT LEFT THRU EACH PAIR OF BYTES (18 TIMES) AND THEN CHECKS THAT THE CARRY IS A ONE AND THE WORD (2 BYTES) IS STILL ALL ZEROS.

THIS PROVIDES THE COMPLEMENT OF THE COVERAGE OF TEST 4 IN DATA.

IT CAN EXECUTE OUT OF THE USER DATA PAR'S AND THE KERNEL DATA PAR'S.

NOTE

IT IS NOT UNCOMMON TO OBSERVE THE GOOD DATA EQUAL THE BAD DATA. THIS INDICATES THAT THE CARRY WAS NOT SET AFTER 18 ROLB'S.

6.2.2.7 TEST 6 INITIAL DATA TEST

WRITES READS A DOUBLE WORD FIRST WITH ALL BITS O EXCEPT 1 (FOR EVERY BIT POSITION). SECOND WITH ALL BITS 1 EXCEPT 1 (FOR EVERY BIT POSITION).

THIS IS A VERY QUICK CHECK OF THE DATA PATHS.

6.2.2.8 TEST 7 ADDRESS BIT TEST

WRITES A BACKGROUND OF ALL ZEROS.
READ ADDRESS 1 FOR A 0 BYTE.
COMPLEMENT ADDRESS 1.
READ ADDRESS 1 FOR A NON 0 BYTE.
FOR EACH ADDRESS BIT POSITION FROM BIT 1:
VIRTUAL (2, 4, 10, 20, 40, 100, 200, 400, 1000, 2000, 4000, 10000,
60000, 20000)
PHYSICAL (60002, 60004, 60010, 60020, 60040, 60100, 60200, 60400,
61000, 62000, 64000, 70000, 140000, 100000)
READ ADDRESS FOR A 0 WORD.
COMPLEMENT ADDRESS CONTENTS.
READ ADDRESS FOR A NON-ZERO WORD.

THIS IS A VERY QUICK CHECK OF THE ADDRESS BIT UNIQUENESS.

2580	
2581	
5701	
2285	
2583	
2584	
2505	
2586	
2587	
2588	
2500	
5304	
2590	
2591	
2591 2592 2593	
2502	
5273	
2594 2595 2596	
2595	
2506	
2507	
5341	
2598	
2599	
2597 2598 2599 2600	
2400	
2601	
26n2	
2603	
2003	

6.2.2.9 TEST 10 BYTE ADDRESSING TEST

WITH ECC DISABLED.
WRITES ALL ONES TO A DOUBLE WORD.
FOR EACH OF THE 4 BYTES IN THE DOUBLE WORD.
CLEARS ONE BYTE.
READS ALL 4 BYTES FROM DOUBLE WORD.
CHECKS FOR ONLY PROPER BYTE CLEAR.
ALL OTHER BYTES SET TO ALL ONES.

THIS IS ONLY DONE ON ONE DOUBLE WORD ADDRESS.

NOTE

THIS IS RUN FOR ECC MEMORY ONLY

2605	
2605 2606 2607 2608 2616 2616 2616 2616 2616 2616 2616 261)
2607	1
2608	
2607	
2611	,
2612	,
2613	Š
2614	
2615	;
2616)
2617	
2618	Š
2630	′
2621	í
262	,
2623	Š
2624	4
2625	5
2626	2
262	
2020	5
2630	7
263	í
263	j
263	3
2634	4
263	5
2630	5
203	6
263	•
264	ń
264	ĭ
264	ż
264	3
264	4
264	5
204	9
264	9
264	ğ
265	Ó
265	ĺ
265	2
265	3
265	4
265	2
200	9
265	9
265	ğ
200	-

6.2.2.10 TEST 11 SINGLE BIT ERROR TEST

- 1. CREATE A SINGLE BIT ERROR.
- 2. READ DATA UNCORRECTED (WITH ECC DISABLE).
- 3. CHECK THAT SBE AND DBE FLAGS ARE SET, AND THE ERROR ADDRESS IS LATCHED.
- 4. READ FIRST WORD OF DATA CORRECTED (WITH ECC ENABLED)
- 5. CHECK THAT THE CSR SINGLE BIT ERROR FLAG WAS SET, AND THE ERROR ADDRESS WAS LATCHED.
- 6. CLEAR SBE FLAG.
- 7. READ SECOND WORD OF DATA CORRECTED (WITH ECC ENABLED).
- 8. CHECK THAT THE CSR SINGLE BIT ERROR FLAG WAS SET.
- 9. DO (1-7) FOR A SINGLE BIT ERROR IN EACH OF 32 POSITIONS OF A DOUBLE WORD.
 1.E. (32 TIMES)
- 10. IF NOT IN QUICK VERIFY MODE THEN DO (1-8) FOR DATA CONSISTING OF 1 BIT SET IN EACH OF 32 POSITIONS OF A DOUBLE WORD.

 I.E. (32 X 32 = 1024 TIMES)
- 11. DO (1-9) FOR COMPLEMENTED DATA (1 BIT CLEAR IN EACH OF 32 POSITIONS OF A DOUBLE WORD).

 1.E. (1024 X 2 = 2048 TIMES)

 OR (32 X 2 = 64 TIMES (QUICK VERIFY))
- 12. DO (1-7) FOR A DOUBLE WORD EQUAL TO (000000,000000), AND ALL POSSIBLE SINGLE BIT ERROR COMBINATIONS FORCED INTO THE CHECK BITS (CSR BITS 5-11).
- 13. CLEAR ANY ERRORS OUT OF TEST LOCATIONS.

THIS INSURES THAT ALL SINGLE BIT ERRORS CAN BE CORRECTED AND DETECTED.

NOTE

THIS TEST IS RUN FOR MS11-M MEMORY ONLY

2661
2662
2662
2003
2004
2665
2666
2667
2449
2000
5003
2670
2671
2672
2673
2474
20/4
5012
2676
2677
2678
2670
2017
2080
2681
2682
2683
2684
2405
2002
2080
2687
2688
2689
2600
2690
5031
2692
2693
2694
2605
2661 2662 2663 2664 2665 2666 2667 2676 2676 2676 2676 2677 2678 2683 2683 2688 2688 2688 2688 2688 268
2070
2697
2698
2699

6.2.2.11 TEST 12 WRITE BYTE CLEARS SBE TEST

- 1. CREATE A SINGLE BIT ERROR.
- 2. WRITE A BYTE OF DOUBLE WORD TO ONES.
- 3. READ A BYTE OF DOUBLE WORD.
- 4. IF THIS IS MS11-M, THE SBE FLAG SHOULD BE SET.
- 5. THE BYTE SHOULD HAVE BEEN EQUAL TO ONES.
- 6. DO (1-5) FOR EACH OF THE 4 BYTES OF THE DOUBLE WORD
- 7. DO (1-6) FOR A SINGLE BIT ERROR IN EACH OF 32 POSITIONS OF A DOUBLE WORD I.E. (32 TIMES)
- 8. IF NOT IN QUICK VERIFY MODE THEN DO (1-7) FOR DATA CONSISTING OF 1 BIT SET IN EACH OF 32 POSITIONS OF A DOUBLE WORD.

 I.E. (32 X 32 = 1024 TIMES)
- 9. CLEAR ANY ERRORS OUT OF TEST LOCATIONS.

THIS INSURES THAT SINGLE BIT ERRORS IN THE DATA PORTION (NOT IN CHECKBITS) CAN BE CLEARED BY WRITING THE CORRESPONDING BYTE AND THAT WRITING ANY OTHER BYTE DOES NOT CHANGE THE EXISTING SINGLE BIT ERROR.

NOTE

THIS TEST IS RUN FOR MS11-M MEMORY ONLY.

6.2.2.12 TEST 13 CREATE DOUBLE BIT ERROR TEST

- 1. CREATE A DOUBLE BIT ERROR.
- 2. ACCESS THE DATA (TST INSTRUCTION).
- 3. CHECK THAT THE CSR DBE FLAG IS SET, AND THE ERROR ADDRESS IS LATCHED.
- 4. INITIALIZE CSR TO ALLOW PARITY TRAPS ON DBE'S.
- 5. ACCESS THE DATA (TST INSTRUCTION).
- 6. CHECK THAT A PARITY TRAP OCCURRED.
- 7. DO (1-6) FOR THE 2ND BIT OF EACH DOUBLE BIT ERROR IN EACH OF 32 POSITIONS OF A DOUBLE WORD LESS THE ONE POSITION OF THE 1ST BAD BIT.

 I.E. (31 TIMES)
- 8. IF NOT IN QUICK VERIFY MODE THEN DO (1-7) FOR THE 1ST BIT OF EACH OF DOUBLE BIT ERROR IN EACH OF 32 POSITIONS OF A DOUBLE WORD.

 I.E. (31 X 32 = 992 TIMES)
- 9. DO (1-8) FOR COMPLEMENTED DATA (ONES VERSUS ZEROS IN DOUBLE WORD)
 I.E. (992 X 2 = 1984 TIMES)
 OR (31 X 2 = 62 TIMES (QUICK VERIFY))
- 10. DO (1-6) FOR A DOUBLE WORD EQUAL TO (000000,000000), AND ALL POSSIBLE DOUBLE BIT ERROR COMBINATIONS FORCED INTO EACH OF THE CHECK BITS (CSR BITS 5-11).
- 11. CLEAR ANY ERRORS OUT OF TEST LOCATIONS.

THIS INSURES THAT ALL DOUBLE BIT ERRORS CAN BE CREATED AND DETECTED AND CAUSE TRAPS.

NOTE

- 1) THIS TEST IS RUN ON THE MS11-M ONLY.
- 2) THIS TEST IS ONLY RUN DURING THE FIRST (QV) PASS WHEN UNDER ACT OR APT, AND IS RUN FOR ECC MEMORY ONLY.

3	75	6	
-	275 275 275 276 276 276 276 276 276 276 277 277 277	8	
-	275	9	
5	76	1	
	276	3	
-	76	2	
	276	5	
-	2/6 76	9	
	276	8	
	276 77	9	
	277	ĭ	
	277 277	2	
-	277	4	
	277	5	
•	277	7	
	277	8	
	278	ó	
	278	1	
	278 278	3	
	278	4	
	2/8 278	16	
	278	7	
	278 278	8	
-	279	Ó	
1	279	11	
	279	3	

6.2.2.13	TEST 14	RASIC	DOUBLE B	IT ERROR	TEST
0.2.2.13	ILSI 14	DUSIC	DOODLE D	TI FULLOW	1 5 3

- 1. WRITE THE CSR TO ENABLE DIAG MODE WITH A DOUBLE BIT ERROR CHECK BITS OF 110011 AND UNCORRECTABLE ERROR INDICATION ENABLED.
- 2. WRITE FIRST AUT IN A 16K BANK WITH DATA OF ALL ZERO'S. THIS WILL WRITE THE CHECK BITS IN (1)
- 3. READ ADDRESS, THIS SHOULD CAUSE A DOUBLE BIT ERROR. BUS PBL IS ASSERTED AND WE CHECK FOR A PARITY TRAP TO OCCUR.
- 4. READ THE CSR FOR CHECK BITS IN (1) AND UNCORRECTABLE ERROR INDICATOR.
- 5. WRITE ONES TO THE HIGH BYTE OF THE ADDRESS UNDER TEST. SINCE A DBE EXSISTS AT THIS ADDRESS THE WRITE SHOULD BE ABORTED.
- 6. READ ADDRESS AND CHECK FOR A PARITY TRAP TO OCCUR AS A RESULT OF (5)
- 7. REPEAT 5 AND 6 FOR DATA OF ONES IN THE LOW BYTE AND CHECK FOR WRITE ABORT AND PARITY TRAP.

THIS TEST CHECKS TO SEE IF A DOUBLE BIT ERROR WILL BE ABORTED AND A BYTE WRITE OF A DOUBLE BIT ERROR WILL BE ABORTED.

NOTE

THIS TEST IS ONLY RUN FOR THE MS11-P

6.2.2.14 TEST 15 WRITE INHIBIT OF BYTE WITH DBE

- 1. CREATE A DOUBLE BIT ERROR.
- 2. DO A MOVB IMMEDIATE TO TEST BYTE.
- 3. CHECK THAT DOUBLE WORD IS STILL BAD (UNCHANGED-WITH DBE).
- 4. DO (2-3) ON ALL 4 BYTES OF DOUBLE WORD.
- 5. DO (1-4) FOR THE 2ND BIT OF EACH DOUBLE BIT ERROR IN EACH OF 32 POSITIONS OF A DOUBLE WORD LESS THE ONE POSITION OF THE 1ST BAD BIT.
 I.E. (31 TIMES)
- 6. IF NOT IN QUICK VERIFY MODE THEN DO (1-5) FOR THE 1ST BIT OF EACH DOUBLE BIT ERROR IN EACH OF 32 POSITIONS OF A DOUBLE WORD.

 I.E. (31 x 32 = 922 TIMES)
- 7. DO (1-6) FOR COMPLEMENTED DATA (ONES VERSUS ZEROS IN DOUBLE WORD).
 I.E. (992 X 2 = 1984 TIMES)
 OR (31 X 2 = 62 TIMES (QUICK VERIFY))
- 8. DO (1-4) FOR A DOUBLE WORD EQUAL TO (000000,000000), AND ALL POSSIBLE DOUBLE BIT ERROR COMBINATIONS FORCED INTO THE CHECK BITS (CSR BITS 5-11).
- 9. CLEAR ANY ERRORS OUT OF TEST LOCATIONS.

THIS INSURES THAT NO DOUBLE BIT ERROR CAN BE CLEARED BY A MOVB TO ANY AFFECTED BYTE.

NOTE

- 1) THIS TEST IS RUN ON THE MS11-M ONLY.
- 2) THIS TEST IS ONLY RUN DURING THE FIRST (QV) PASS WHEN UNDER ACT OR APT, AND IS RUN FOR ECC MEMRY ONLY.

6.2.2.15 TEST 16 WRITE INHIBIT OF WORD WITH DBE TEST

- 1. CREATE A DOUBLE BIT ERROR.
- 2. DO MOV IMMEDIATE ON TEST LOCATION.
- 3. CHECK THAT DOUBLE WORD IS STILL BAD (UNCHANGED-WITH DBE).
- 4. DO (2-3) ON BOTH DOUBLE WORDS.
- 5. DO (1-4) FOR THE 2ND BIT OF EACH DOUBLE BIT ERROR IN EACH OF 32 POSITIONS OF A DOUBLE WORD LESS THE ONE POSITION OF THE 1ST BAD BIT.
 I.E. (31 TIMES)
- 6. IF NOT IN QUICK VERIFY MODE THEN DO (1-5) FOR THE 1ST BIT OF EACH DOUBLE BIT ERROR IN EACH OF 32 POSITIONS OF A DOUBLE WORD.

 I.E. (32 X 32 = 992 TIMES)
- 7. DO (1-6) FOR COMPLEMENTED DATA (ONES VERSUS ZEROS IN DOUBLE WORD).
 I.E. (992 X 2 = 1984 TIMES)
 OR (31 X 2 = 62 TIMES (QUICK VERIFY))
- 8. DO (1-4) FOR A DOUBLE WORD EQUAL TO (000000,000000), AND ALL POSSIBLE DOUBLE BIT ERROR COMBINATIONS FORCED INTO THE CHECK BITS (CSR BITS 5-11).
- 9. CLEAR ANY ERRORS OUT OF TEST LOCATIONS.

THIS INSURES THAT NO DOUBLE BIT ERROR CAN BE CLEARED BY A MOV TO ANY AFFECTED WORD.

NOTE

- 1) THIS TEST IS RUN ON THE MS11-M ONLY
- 2) THIS TEST IS ONLY RUN DURING THE FIRST (QV) PASS WHEN UNDER ACT OR APT, AND IS RUN FOR ECC MEMORY ONLY.

6.2.2.16 TEST 17 HOLDING 1'S O'S TEST

- 1. WRITE A 16K BANK WITH ALTERNATING BYTES OF ZEROS ONES WRITING A BYTE AT A TIME.
- 2. READ EACH WORD FOR CORRECT TEST.
- 3. DO (1-2) AGAIN FOR A COMPLEMENT TEST.

THIS CHECKS THE MEMORY FOR THE CAPABILITY OF HOLDING O'S 1'S.

6.2.2.17 TEST 20 SYNDROME BITS TO THE CSR ON A SBE TEST

- 1. WRITE CSR WITH CHECK BITS TO CORRECT BIT O OF THE FIRST AUT 16K BANK FROM A O TO A 1 WITH DIAG MODE.
- 2. WRITE AUT WITH DATA OF O'S CREATING A SBE.
- 3. CLEAR CSR.
- 4. READ THE AUT TO CLOCK THE ADDRESS AND SYNDROMES INTO THE CSR.
- 5. READ THE CSR FOR THE SBE INDICATOR, BIT 4.
- 6. WRITE THE CSR TO DIAG MODE TO CLOCK THE SYNDROME BITS INTO CSR BITS 5-11.
- 7. READ THE CSR FOR THE PROPER SYNDROME BITS.
- 8. REPEAT 1-7 FOR ALL 16 DATA BITS.
- 9. REPEAT 1-8 FOR DATA OF ONES SO THAT A CORRECTION WILL OCCUR FROM A 1 TO A ZERO.

THIS TEST CHECKS TO SEE THAT THE EDC CHIP CAN DETECT SINGLE BIT ERRORS FOR ALL 16 DATA BITS BY CHECKING FOR CSR BIT#4 AND THAT THE PROPER SYNDROME BITS ARE PLACED IN THE CSR.

NOTE

6.2.2.18 TEST 21 MARCHING O'S 1'S TEST

- 1. WRITE A BACKGROUND OF ALTERNATING BYTES OF ZEROS ONES
- FOR THE 16K BANK ADDRESSING DOWN
 - (A) READ CHECK A WORD
 (B) BYTE SWAP A WORD
 - (C) READ CHECK A WORD
- FOR THE 16K BANK ADDRESSING UP

 - (A) READ CHECK A WORD (B) BYTE SWAP A WORD (C) READ CHECK A WORD
- FOR THE 16K BANK ADDRESSING UP

 - (A) READ CHECK A WORD (B) BYTE SWAP A WORD (C) READ CHECK A WORD
- FOR THE 16K BANK ADDRESSING DOWN (A) READ CHECK A WORD

 - (B) BYTE SWAP A WORD
 - (C) READ CHECK A WORD

THIS CHECKS THE INTEGRITY OF THE 32 BIT DOUBLE WORDS. IT CAN EXECUTE OUT OF THE USER DATA PAR'S.

NOTE

IT IS NOT UNCOMMON TO SEE A MISLEADING ERROR TYPEOUT BECAUSE THE SECOND TEST IN EACH CASE IS BASED UPON A BYTESWAP OF THE FIRST TEST WHICH MAY OR MAY NOT HAVE FAILED. IF THE ERROR REPORT INDICATES ERRORS IN DAILS HITH THE DAY DIT IN THE ERRORS IN PAIRS WITH THE BAD BIT IN THE SECOND REPORT BEING THE SAME BIT POSITION RELATIVE TO A BYTE THEN YOU SHOULD IGNORE THE SECOND ERROR REPORT.

6.2.2.19 TEST 22 REFRESH TEST

1. WRITE A DIAGONAL TEST OF ONES ON EVERY KDIAG(TH) STRIPE WRITE ZEROS ELSEWHERE.

THIS TEST IS ON ADDRESSES NOT BIT POSITIONS.

EXAMPLE:

ADDRESS

LSB'S

PI:	98	2							
	00010001	00100010	01000100	10001000	00010001	00100010	01000100	10001000	

MCDIC

NOTE

EXAMPLE USES KDIAG OF VALUE 4 MORE TYPICAL IS A VALUE OF 8. CONSULT THE SYMBOLIC DEFINITION OF "KDIAG" IN THE PROGRAM LISTING TO BE SURE.

- 2. DISTURB EACH ROW FOR > 3.2MS
- 3. READ CHECK DIAGONAL PATTERN
- 4. DO (1-3) KDIAG TIMES MOVING THE PLACEMENT OF THE DIAGONAL STRIPE TO COVER ALL ADDRESS POSITIONS.
- 5. DO (1-4) FOR A COMPLEMENT PATTERN (ZEROS IN A BACKGROUND OF ONES)

NOTE

6.2.2.20 TEST 23 SHIFTING DIAGONAL TEST

SIMILAR IN OVERALL OPERATION TO TEST 22 EXCEPT IT DOES NOT DELAY FOR REFRESH AND DISTURB ROWS.

NOTE

6.2.2.21 TEST 24 FAST GALLOPING PATTERN TEST

THIS DOES A CLASSICAL GALLOPING PATTERN EXCEPT THAT ADDRESSING IS INCREMENTED BY 400 OCTAL (EVERY 64TH DOUBLE WORD)

NOTE

6.2.2.22 TEST 25 INTERRUPT ENABLE TEST

- 1. SET CSR TO ALLOW UNCORRECTABLE ERROR TRAPS.
- 2. ACCESS TEST DOUBLE WORDS.
- 3. CHECK THAT NO UNCORRECTABLE ERROR TRAP OCCURRED.
- 4. ENABLE CSR FOR SBE TRAPS.
- 5. ACCESS TEST DOUBLE WORDS.
- 6. CHECK THAT NO SBE TRAP OCCURRED.
- 7. WRITE A SBE IN 1 BYTE.
- 8. DISABLE CSR TRAPS.
- 9. ACCESS TEST DOUBLE WORDS.
- 10. CHECK THAT NO TRAPS OCCURRED.
- 11. ENABLE CSR FOR SBE TRAPS.
- 12. ACCESS TEST DOUBLE WORDS.
- 13. CHECK TO INSURE TRAP OCCURRED.
- 14. DO (7-13) FOR THE 3 OTHER BYTES IN THE DOUBLE WORD.
- 15. CREATE A DBE IN 1 BYTE.
- 16. DISABLE CSR TRAPS.
- 17. ACCESS THE TEST DOUBLE WORD.
- 18. CHECK THAT NO TRAPS OCCURRED.
- 19. ENABLE CSR FOR DBE TRAPS.
- 20. ACCESS THE TEST DOUBLE WORD.
- 21. CHECK TO INSURE TRAP OCCURRED.
- 22. ENABLE CSR FOR SBE TRAPS.
- 23. ACCESS THE TEST DOUBLE WORD.
- 24. CHECK TO INSURE TRAP OCCURRED.
- 25. DO (15-24) FOR THE 3 OTHER BYTES IN THE DOUBLE WORD.

THIS INSURES THAT SBE'S DBE'S GIVE THE CORRECT TYPE OF TRAPS.

THIS TEST IS RUN FOR MS11-M MEMORY ONLY.

6.2.2.23 TEST 26 RANDOM DATA TEST

WRITE RANDOM DATA IN A 16K BANK WHILE INCREMENTING THE ADDRESSES.

READ CHECK RANDOM DATA.

THIS ROUTINE REGENERATES THE SAME RANDOM NUMBERS BY USING THE SAME

SEED AS THE WRITE SEQUENCE. AFTER THE READ CHECK THE SEED IS UPDATED SO THAT THE NEXT USE OF THIS PATTERN WILL NOT INVOKE THE SAME SEQUENCE OF RANDOM NUMBERS.

IF YOU WISH TO CHANGE THE RANDOM SEQUENCE SO THAT IT IS DIFFERENT THAN ANY OTHER RUN IN THE SAME CONFIGURATION THEN THERE ARE 2 WAYS OF DOING SO.

- 1. MODIFY SYMBOLIC LOCATIONS "SEEDHI" AND "SEEDLO" TO ANY NUMBER YOU LIKE.
- 2. ENTER FIELD SERVICE MODE AND EXECUTE THIS TEST (COMMAND 5) ON SOME (ANY GOOD) BANK FOR A SHORT TIME (30 SEC OR SO).

THIS CAN EXECUTE OUT OF THE USER DATA PAR'S, THE KERNEL DATA PAR'S, AND THE SUPERVISOR DATA PAR'S.

6.2.2.24 TEST 27 UNIQUE BANK TEST

THIS TEST USES TEST O TO WRITE READ THE BANK NUMBER IN EACH BANK.

IT DOES NOT TEST BANKS THAT REQUIRE RELOCATION TO TEST.

IT DOES NOT RUN AS PART OF ANY SCRIPT BUT RATHER IS ALWAYS RUN AFTER NORMAL PATTERN TESTS ARE COMPLETE.

6.2.2.25 TEST 30 FLUSH OUT DBE'S TEST

THIS READS EACH LOCATION THEN MOVES THE OLD VALUE BACK IN. THIS IS DONE WITH ECC DISABLED AND THEREFORE CORRECTS ANY DBE'S OR SBE'S (IF POSSIBLE).

IT DOES NOT RUN AS PART OF ANY SCRIPT BUT RATHER IS ALWAYS RUN JUST PRIOR TO THE END OF PASS CODE, AS PART OF A CONTROL "C" (BOOT) COMMAND, AS PART OF END OF PASS SHUTDOWN FOR ACT OR XXDP CHAIN MODE, AS PART OF HANGING SEQUENCE AFTER AN ERROR IF UNDER ACT OR APT, AND AS PART OF A SHUTDOWN SEQUENCE DIRECTED BY SWITCH 8 (HALT PROGRAM).

6.2.2.26 TEST 31 SOB-A-LONG TEST

RATIONALIZATION

IN ORDER TO CONCENTRATE THE MEMORY CYCLES OF A TEST INTO A PARTICULAR ADDRESS, WE MUST CUT THE OVERHEAD CYCLES TO A MINIMUM. FREQUENTLY, THE INSTRUCTION ITSELF MAY PROVIDE ADEQUATE DATA OR SET UP A BACKGROUND IN WHICH ANY COMPLEMENTED BIT MAY FIND IT HARD TO SURVIVE.

THE SOB INSTRUCTION IS THE ONLY PDP-11 INSTRUCTION THAT IS (1) A SINGLE OPERAND, (2) CAN BE REPEATEDLY EXECUTED AT THE SAME PC AND, (3) CAN ESCAPE THIS REPETITIOUS LOOP.

HENCE, IT CAN BE POSSIBLE TO SOB A MOS CELL TO DEATH (OR AT LEAST BRAIN WASH HIM), AND TO SOB A CORE INTO OVER-HEATING (OR AT LEAST WARM DISCOMFORT).

THE SOB ROUTINE WILL BE LOADED AND CALLED WITH RO SET EQUAL TO THE SOB CONSTANT 'SOBK', R1 SET EQUAL TO THE COMPLEMENT OF A 'SOB RO..' INSTRUCTION "100776".

SIMPLIFIED SOB EXAMPLE:

1\$:	SOB MOV CMP BEQ	R0.1\$ R1.1\$ R1.1\$:SOB TILL RO UNDERFLOWS :WRITE COMPLEMENT OF SOB :READ CHECK NOT SOB :SKIP IF OK
2\$:	SOBFAIL SOBMOV1 SOBMOV2 SOBMOV3 SOBMOV4		CODE TO GET SELF MOVED FORWARD 1 WORD AND RUN AGAIN

THE VALUE OF THE SOB CONSTANT CAN BE FOUND AT SYMBOLIC LOCATION "SOBK" (TYPICAL 25 DECIMAL).

THIS TEST IS NOT IN THE NORMAL SCRIPT OF EXECUTION BUT MAY BE ADDED VIA THE APT E-TABLE, REFERENCE SYMBOLIC LOCATIONS 'MKPAT', 'MJPAT', 'SDDW2-5'. FIELD SERVICE MODE COMMAND 8 IS THE NORMAL METHOD OF RUNNING THIS PATTERN.

NOTE

6.2.2.27 TEST 32 WRITE RECOVERY TEST

THIS TEST CAUSES A WRITE, READ, WRITE, READ, ... TO OCCUR IN MEMORY AND IF THE 1ST, 3RD, 5TH, ... READ IS BAD THE PROGRAM MAY BOMB OR IF THE 2ND, 4TH, 6TH, ... READ IS BAD THE PROGRAM WILL GRACEFULLY TYPE OUT THE ERROR.

WRITE RECOVERY TEST THIS TEST DIFFERS FROM OTHER TESTS IN THAT IT CONSISTS OF A SMALL TEST PROGRAM ACTUALLY RUNNING IN THE BANK UNDER TEST. THE PROGRAM IS SELF MODIFYING AND MAY BE DIFFICULT TO DEBUG. TO AID IN THE DEBUG, REMEMBER THAT THE BANK AND MARGIN ARE BEING DISPLAYED. THIS WILL ALLOW THE USER TO AT LEAST SEE WHICH MEMORY BANK FAILED.

THE TEST CONSISTS OF 1/2 OF THE BANK STORED WITH 'MOV R2,-(PC)"
AND THE OTHER 1/2 CONTAINING "177667". "177667" IS THE COMPLEMENT
OF "JMP (R0)" INSTRUCTION. R2 CONTAINS "COM -(R1)" INSTRUCTION
ON ENTRY TO THE BANK AND R1 CONTAINS THE HIGHEST TEST ADDRESS IN

- IF YOU UNDERSTAND THIS SO FAR THE REST IS EASY.

 THE TEST EXECUTION IS AS FOLLOWS:

 1. THE 'MOV R2,-(PC)'' INSTRUCTION EXECUTES STORING

 THE CONTENTS OF R2 IN THE ADDRESS IT VACATED (DUE TO -(PC).

 2. SINCE R2 CONTAINS A 'COM -(R1)'' INSTRUCTION IT COMPLEMENTS

 THE HIGHEST ADDRESS UNDER TEST. THIS ADDRESS CONTAINED

 ''177667'' SO AFTER THE COM -(R1) IT EQUALS 110

 CLEVERLY THIS IS THE 'JMP (R0)'' INSTRUCTION.
 - 3. THIS SEQUENCE CONTINUES UNTIL THE 'MOV R2,-(PC)" INSTRUCTIONS REACH THE MIDDLE OF THE TEST BANK. THEN THE 'JMP (RO)" INSTRUCTION IS MET AND EXECUTED. RO CONTAINED THE RETURN ADDRESS BACK TO TEST 13.
 - 4. THESE STEPS ARE REPEATED FOR EACH BANK UNDER TEST.

NOTE

6.2.2.28 TEST 33 BRANCH GOBBLE TEST

THIS TEST LOADS A SMALL ROUTINE INTO THE MEMORY UNDER TEST. THE ROUTINE MOVES ITSELF ALONG IN MEMORY ONE WORD AFTER EACH PASS SO THAT WHEN IT REACHES THE END EVERY INSTRUCTION HAS EXECUTED FROM EVERY LOCATION WITH THE EXCEPTION OF THE BEGINNING AND END OF EACH TEST AREA.

THE BRANCH GOBBLE'S GENERAL FORMAT AFTER YOU ELIMINATE SETUP CODE AND CODE TO MOVE THE PROGRAM ALONG IS AS FOLLOWS.

BGTEST:			;TEST WORD
BRGOBB:	SEC ADCB BMI INCB BR	BGTEST 1\$ BGTEST+1 BRGOBB	:INC LOW BYTE :END LOOP AFTER 128 TIMES ;INC HIGH BYTE :LOOP 128 TIMES
1\$:	BVS ERROR	2\$:BRANCH IF V-BIT SET (SHOULD BE) :ERROR TRAP
2\$:	CLV INCB BCS BVC BMI	BGTEST 3\$ 3\$	CLEAR V-BIT :INC HIGH BYTE ONE LAST TIME :BRANCH IF C-BIT SET (SHOULD NOT BE) :BRANCH IF V-BIT CLEAR (SHOULD NOT BE) :BRANCH IF N-BIT SET (SHOULD BE)
3\$:	ERROR RETURN	4\$	ERROR TRAP

THIS CODE ORIGIONALLY CAME FROM THE PDP-11 FAMILY INSTRUCTION EXERCISER DZQKA-A. THE FIRST MOS MEMORYS FELL SUCCEPTABLE TO THIS SECTION OF THAT DIAGNOSTIC AND IT HAS BEEN AN IMPORTANT MEMORY EXERCISER EVER SINCE.

NOTE

6.2.2.29 TEST 34 SOFT ERROR TEST

RATIONALIZATION

MOS CHIPS HAVE A FAILURE MODE IN WHICH THEY CAN RANDOMLY PICK OR DROP BITS. THIS IS CAUSED BY ALPHA PARTICLES BOMBARDING THE CELL. IF THE CELL IS VERY SMALL (AND THEY ARE) THEN THE ELECTRONS DISPLACED BY THE ALPHA PARTICLE ARE SUFFICIENT TO CAUSE THE CELL TO CHANGE FROM A ONE TO A ZERO OR FROM A ZERO TO A ONE.

THIS TEST IS CONTROLLED BY THE MAIN PROGRAM SO THAT IT IS USED TO CREATE A TEST OF 125252 AND 52525 ON ALTERNATE PASSES OF THE PROGRAM. THE CONFIGURATION TABLE IS USED TO FLAG BANKS THAT HAVE THE TEST INVALIDATED BECAUSE ANOTHER TEST WAS WRITTEN OVER THIS BACKGROUND.

THIS TEST IS NOTHING MORE THAN A CLEVER USE OF TEST O.

34	1	1	
34	1	2	
34	1	34567890	
24	d	ĭ	
34 34	Н	7	
24	ı	Š	
34	1	6	
34 34 34 34 34 34	1	7	
34	1	8	
37	1	ă	
7/	2	ń	
34	5	ď	
34	Ž	12	
34	2	2	
34	2	3	
34	.5	4	
34	ัว	Š	
77	5	6	
24	Ž		
34	1	1	
3434	2	78	
34	.2		
34	3	ó	
37	ž	ĭ	
77	7	2	
24	5	ξ	
54	2	Ş	
34	3	4	
34343434	.3	567	
34	3	6	
3/	E	78	
3,	F	6	
5,	2	O	
34	15	9	

6.2.2.30 TEST 35 WORST CASE PARITY TEST

- 1. FORCE WRITE WRONG PARITY IN EACH 1K WORD BLOCK OF THE MEMORY UNDER TEST.
- 2. READ WITH PARITY TRAPPING ENABLED, MAKING SURE THAT A TRAP OCCURRS.
- 3. MAKE SURE ERROR ADDRESS BITS ARE SET CORRECTLY.
- 4. WRITE GOOD PARITY WITHOUT TRAPPING, AND MAKE SURE NO TRAP OCCURRS WHEN READ.

NOTE

THIS TEST IS RUN FOR PARITY MEMORY WHICH IS NOT CONTROLLED BY THE SAME CSR AS THE PROGRAM.

3441
3442
3445
3444
3446
3447
3448
3449
3450
3452
3453
3454 3455
3455
3456 3457 3457
3458
3459
3460
3461
3463
3463 3464 3465 3466
3465
3466
3467

6.2.2.31 TEST 36 CORRECTION CODE TEST

- 1. WRITE CSR WITH CHECK BITS TO CORRECT BIT O OF THE FIRST ADDRESS IN A 16K BANK FROM A 0 TO A 1 WITH DIAG MODE.
- 2. WRITE AUT WITH DATA OF O'S.
- 3. READ AUT FOR CORRECTION OF BIT 0 FROM A 0 TO A 1.
- 4. REPEAT 1-3 FOR ALL 16 DATA BITS.
- 5. REPEAT 1-4 FOR DATA OF ONES SO THAT A CORRECTION WILL OCCUR FROM A 1 TO A ZERO.

THIS TEST CHECKS TO SEE THAT THE EDC CHIP CAN CORRECT SINGLE BIT ERRORS FOR ALL 16K DATA BITS FROM A 1 TO A 0 AND VISA VERSA.

NOTE

6.2.2.32 TEST 37 CHECK ECC DISABLE TEST

- 1. WRITE CSR WITH ECC DISABLE, DIAG MODE, AND SBE CHECK BITS OF 000010.
- 2. WRITE AUT WITH DATA OF ZERO'S. THIS SHOULD WRITE CHECK BITS TO MEMORY.
- 3. READ AUT FOR DATA OF ZEROS INSURING NO CORRECTION WAS MADE.

NOTE

THIS TEST IS RUN ON THE MS11-P ONLY.

6.2.2.33 TEST 41 ADDRESS TO CSR ON DBE TEST

- 1. WRITE CSR WITH ECC DISABLE, DIAG MODE, AND DOUBLE BIT ERROR CHECK BITS OF 010011
- 2. WRITE AUT WITH DATA OF ZEROS CREATING A DBE.
- 3. READ AUT TO DETECT DBE AND TO CLOCK ADDRESS INTO CSR
- 4. READ CSR FOR CORRECT ADDRESS IN BITS 5-11.
- 5. INCREMENT ADDRESS BY 1K AND REPEAT 1-4 UNTIL 16K IS DONE.

THIS TEST INSURES THAT THE CORRECT ADDRESS APPEARS IN CSR BITS 5-11 ON A DBE

NOTE

THIS TEST IS RUN ON A MS11-P ONLY.

- 6.2.2.34 TEST 42 EXTENDED ADDRESS TO CSR ON ERROR TEST
 - 1. WRITE CSR WITH SBE CHECK BITS OF 000010 WITH DIAGNOSTIC MODE.
 - 2. WRITE LOW ADDRESS IN A 16K BANK WITH DATA OF ZEROS CREATING A SBE.
 - 3. CLEAR THE CSR.
 - 4. READ ADDRESS TO DETECT SBE.
 - 5. READ CSR FOR CORRECT ADDRESS AND THE SBE INDICATOR BIT #4.
 - 6. ENABLE CSR BIT 14 TO CHECK THE EXTENDED ADDRESS BITS.
 - 7. READ CSR FOR CORRECT ADDRESS BITS
 - 8. REPEAT 1-7 WITH A TEST ADDRESS THAT IS THE HIGHEST IN A 16K BANK.

THIS TEST CHECKS TO SEE THAT THE CORRECT ADDRESS BITS APPPEAR IN THE CSR. THIS IS ALSO REPEATED FOR THE EXTENDED ADDRESS FUNCTION IN THE CSR.

NOTE

3560	
3561	
3562	
3563	
3565	
3204	
2202	
3566	
3567	
3568	
3560	
3570	
3570	
25(1	
3572	
3573	
3574	
3575	
3576	
3210	
35((
35/8	
3579	
3580	
3581	
3582	
3502	
2203	
3284	
3585	
3586	
3587	
3588	
3580	
3500	
2290	
3591	
3592	
3593	
3560 3561 3562 3563 3564 3565 35667 35667 3567 3577 3577 3577 35	
3594 3595	
3563 35645 35645 35645 355667 355677 35577 35577 35577 35577 35581 355867 3558	
2270	
3597	
3598	
3599	

6.2.2.35 TEST 43 WRITE BYTE TEST

- 1. WRITE CSR TO DIAG MODE WITH CHECK BITS OF 001100. THESE CORRESPOND TO DATA OF ZEROS.
- 2. WRITE FIRST AUT WITH DATA OF ONE IN BIT ZERO. THE WRITE EFFECTIVELY CREATES A SBE IN BYTE O.
- 3. CLEAR THE CSR
- 4. WRITE BYTE 1 OF THE AUT WITH DATA OF ALL ONES.
- 5. READ CSR TO CHECK FOR SBE INDICATION.
- 6. WRITE THE CSR TO DIAG MODE.
- 7. READ THE AUT TO CHECK FOR THE CORRECT DATA -- ALL ONES IN HIGH BYTE AND ALL ZEROS IN LOW BYTE.
- 8. READ THE CSR TO CHECK FOR CORRECT CHECK BITS CORRESONDING TO THE DATA READ IN (7). THESE CHECK BITS ARE 000110.
- 9. REPEAT (1)-(8) THIS TIME CREATING AN ERROR IN BYTE 1 (2) AND WRITING BYTE 0 IN (4).

THIS TEST CHECKS TO SEE THAT A SBE WILL BE CORRECTED DURING THE READ PORTION OF THE BYTE WRITE AND THAT CORRECT CHECKBITS WILL BE GENERATED ON THE WRITE.

NOTE

- 6.2.2.36 TEST 44 SHIFTING CHECKBITS THROUGH THE CSR TEST
 - 1. WRITE CSR TO DIAG MODE TO ENABLE CHECKBIT REGISTER.
 - 2. WRITE CSR WITH CHECK BITS OF 000001, ECC DISABLE AND DIAG MODE.
 - 3. WRITE MEMORY WITH DATA OF ZEROS. THIS SHOULD WERITE THE CHECK BITS INTO MEMORY.
 - 4. COMPLEMENT CHECK BITS PATTERN AND WRITE CSR AS IN (2).
 - 5. READ CSR FOR COMPLMENT CHECK BIT PATTERN.
 - 6. READ MEMORY TO READ CHECK BITS WRITTEN IN (2) INTO CSR.
 - 7. READ CSR FOR CORRET CHECK BITS WRITTEN IN (2).
 - 8. SHIFT CHECK BIT PATTERN AND REPEAT (1-7) TILL CSR BITS 5-10 ARE DONE.
 - 9. COMPLEMENT CHECK BIT PATTERN IN (2) AND REPEAT (1-8) SHIFTING A ZERO THROUGH A FIELD OF ONES.

THIS TEST CHECKS THE ABILITY TO READ CHECK BITS FROM THE CSR TO MEMORY AND BACK.
THE TEST IS DONE TWICE. ONCE SHIFTING A FIELD OF A ONE THROUGH A FIELD OF ZEROS
AND A ZERO THROUGH A FIELD OF ONES. THIS TESTS THE CHECKBIT/SYNDROME BIT REGISTER
AND CHECK BIT RAM'S

NOTE

6.2.2.37 TEST 45 SYNDROME BITS TO THE CSR ON A DBE TEST

- 1. WRITE CSR WITH DIAG MODE TO ENABLE CHECK/SYNDROME BIT REGISTER.
- 2. WRITE CSR WITH DBE CHECK BITS OF 110011 WITH DIAG MODE.
- 3. WRITE MEMORY WITH DATA OF ZEROS CREATING A DBE.
- 4. CLEAR CSR.
- 5. READ MEMORY TO DETECT DBE.
- 6. READ CSR FOR UNCORRECTABLE ERROR INDICATOR.
- 7. WRITE CSR TO DIAG MODE TO READ SYNDROME BITS INTO CSR.
- 8. READ CSR FOR CORRECT SYNDROME BITS OF 111111.
- 9. REPEAT (1-8) WITH MULIPLE BIT ERROR CHECK BITS OF 111100 AND CORRESPONDING SYNDROME BITS OF 110000.

THIS TEST CHECKS THE ABILITY OF THE CSR TO DETECT A DBE AND READ FOR THE PROPER SYNDROME BITS GENERATED BY THE EDC CHIP. THIS TEST IS THEN REPEATED WITH CHECK BITS CORRESPONDING TO A MULTIPLE BIT ERROR.

NOTE

- 6.2.2.38 TEST 46 CHECK SINGLE BIT ERRORS WITH ECC DISABLED TEST
 - 1. WRITE CSR WITH CHECK BITS TO CORRECT BIT O OF THE FIRST ADDRESS IN A 16K BANK FROM A 0 TO A 1 WITH DIAG MODE AND ECC DISABLED.
 - 2. WRITE AUT WITH DATA OF O'S THUS CREATING A SBE.
 - 3. WRITE THE CSR TO ECC DISABLE.
 - 4. READ AUT TO DETECT SBE.
 - 5. CHECK TO SEE THAT NO TRAP OCCURED.
 - 6. READ CSR TO SEE THAT UNCORRECTABLE ERROR (CSR15) IS SET.
 - 7. REPEAT 1-6 FOR ALL 16 DATA BITS.
 - 8. REPEAT 1-7 FOR DATA OF ONES SO THAT A CORRECTION WILL OCCUR FROM A 1 TO A ZERO.
 - 9. REPEAT 1-8 EXCEPT IN STEPS (3) THE CSR IS WRITTEN TO ECC DISABLE AND BUS PBL ENABLE AND (5) WE CHECK FOR TRAPS.

THIS TEST CHECKS TO SEE THAT SBE ARE TREATED A UNCORRECTABLE ERRORS WITH ECC DISABLE. THE TEST IS REPEATED 2 TIMES, ONCE WITH TRAPS DISABLED AND AGAIN WITH IT ENABLED. THIS IS DONE FOR ALL 16 POSSIBLE SBE CONDITIONS.

NOTE

- 6.2.2.39 TEST 47 NO CSR UPDATE ON SBE WITH EXSISTING DBE TEST
 - 1. WRITE THE CSR TO DIAG MODE TO ENABLE CHECKBIT/SYNDROME BIT REGISTER.
 - 2. WRITE THE CSR WITH DBE CHECK BITS OF 110011 AND DIAG MODE.
 - 3. WRITE MEMORY WITH DATA OF ZEROS CREATING A DBE.
 - 4. WRITE CSR WITH SBE CHECK BITS OF 000010 AND DIAG MODE.
 - 5. WRITE MEMORY 4K ABOVE ADDRESS IN (3) CREATING A SBE.
 - 6. CLEAR CSR.
 - 7. READ MEMORY WITH ADDRESS IN (3) TO DETECT DBE.
 - 8. READ CSR FOR CORRECT ADDRESS AND UNCORRECTABLE ERROR INDICATOR
 - 9. READ MEMORY WITH ADDRESS IN (5) TO DETECT SBE.
 - 10. READ CSR FOR SBE INDICATOR AND NO CHANGE IN DBE STATUS IN CSR IN (8)

THIS TEST CHECKS TO SEE THAT NO UPDATE WILL OCCUR IN THE CSR WITH A SBE IN MEMORY WHEN A DBE ALREADY EXISTS.

NOTE

6.2.2.40 TEST 999 NULL TEST
THIS IS AN INSTANT RETURN ADDED TO PRESERVE THE SOFTWARE STRUCTURE.
THIS TEST REPLACES ANY REAL TESTS WHEN THE APT E-TABLE DOES NOT SPECIFY A TEST TO BE RUN.

7.0 PROGRAM FEATURES

7.1 FAST DATA ACCESS RATES

ONE OF THE MAIN AREAS OF CONCERN IN TESTING MEMORY IN SYSTEMS ENVIRONMENTS IS SPEED. ONE OF THE PRIME REASONS THAT SYSTEM PROGRAMS LIKE RSTS, IAS AND MUMPS CAN CRASH DUE TO MEMORY FAILURES NOT DETECTABLE BY MEMORY DIAGNOSTICS (0-124K,0-2 MEG,ETC.) IS BECAUSE OF MULTIPLE NPR DEVICES CONTENDING FOR THE BUS. AFTER SOME DELAY A NPR DEVICE BECOMES BUS MASTER AND DOES SEVERAL MEMORY TRANSFERS AT MEMORY DATA RATES.

ON THE OTHER HAND MOST DIAGNOSTICS WHEN WRITING READING AND/OR CHECKING PATTERNS SPEND MOST OF THEIR TIME FETCHING INSTRUCTIONS AND OPERANDS OUT OF THEIR PROGRAM SPACE AND PROPORTIONALLY LITTLE TIME ACCESSING THE MEMORY UNDER TEST.

THIS DIAGNOSTIC'S ERROR DETECTING ABILITIES HAVE BEEN OPTIMIZED AROUND THE PRIMARY DESIGN CRITERIA OF SPEED. TO THIS END THE FOLLOWING STEPS HAVE BEEN TAKEN.

7.1.1 FAST CITY

UTILIZATION OF MEMORY MANAGEMENT REGISTERS AS NON MEMORY BUS, NON UNIBUS, BIPOLAR MEMORY. SINCE USER MODE IS ONLY USED FOR RELOCATION AND DATA SPACE IS NEVER USED, THEN SUBROUTINES CAN BE EXECUTED FROM THE UIPAR'S, UDPAR'S, KDPAR'S, SDPAR'S AND WITH SOME BIT PATTERN RESTRICTIONS THE UIPDR'S, UDPDR'S, KDPDR'S, AND SDPDR'S.

THE PROGRAM RUNS IN KERNEL MODE AND PATTERNS ARE EXECUTED IN SUPERVISOR MODE FOR MAPPING PURPOSES. ALL CORE PATTERNS AND SOME MOS PATTERNS ARE SUBROUTINES THAT ARE MOVED TO THIS BIPOLAR REGION REFERRED TO IN THE PROGRAM AS FAST CITY.

NOTE

18-BIT PDP-11'S CANNOT EXECUTE FROM THE PAR'S BECAUSE THEIR PAR'S ARE ONLY 12 BITS WIDE; THEY ALSO HAVE NO SUPERVISOR MODE. THEREFORE, ALL PATTERNS ARE EXECUTED IN MEMORY, USING USER MODE (REFERENCE SECTION 7.5).

7.1.2 SOB'S

UTILIZATION OF THE FULL PDP-11 INSTRUCTION SET TO SPEED PATTERN ALGORITHMS (PRINCIPALLY THE SOB).

7.1.3 CACHE

CACHE IS USED BETWEEN PATTERN TESTS TO DECREASE PROGRAM PASS TIMES. CACHE CAN BE DEFEATED BY THE OPERATOR (REFERENCE SECTION 2.4.3.1).

7.2 BANK ZERO TESTING

BANK ZERO HAS BEEN TRADITIONALLY NEGLECTED BY MEMORY DIAGNOSTICS FOR THE FOLLOWING REASON.

THE VECTOR SPACE EXISTS THERE AND ALL TRAPS MUST NOT ACCESS TEST PATTERN DATA. IF THE AREA IS TESTED THE DIAGNOSTIC MUST NOT USE ANY TRAPS, AND IT IS AGAINST THE RULES FOR POWER TO FAIL.

SYSTEMS WITH MEMORY MANAGEMENT CAN OVERCOME THIS BECAUSE ALL TRAPS ARE TO KERNEL VIRTUAL SPACE EVEN IF THE POWER SHOULD FAIL (CAUTION MUST BE OBSERVED BECAUSE POWER UP GOES TO PHYSICAL ADDRESS 24 (BECAUSE THE MEMORY MANAGEMENT UNIT COMES UP OFF)).

HOWEVER, CATCH 22 IS THAT THE DIAGNOSTIC IS NOT APT COMPATIBLE IN THIS MODE BECAUSE APT ACCESSES PHYSICAL MEMORY LOCATIONS.

THE PDP-11/44 CAN OVER COME THIS BECAUSE THE UNIBUS MAP CAN FOOL APT.

BECAUSE OF THE PREVIOUS ARGUMENTS THIS PROGRAM DOES NOT RELOCATE IN THE TRUE SINCE OF THE WORD (I.E. NO POSITION INDEPENDENT CODE WAS WRITTEN (AT LEAST NOT ON PURPOSE)), BUT RATHER THIS PROGRAM MOVES AND REMAPS (HEREAFTER REFERRED TO AS RELOCATES). THIS ENABLES THE COMPLETE TESTING OF BANK ZERO OR ANY OTHER PROGRAM SPACE OR PRIVILEGED SPACE EXACTLY AS ALL OTHER BANKS ARE TESTED. (THE CONDITIONAL TEST TO SEE IF A BANK IS PROTECTED IS COMPLEMENTED WHEN RELOCATED).

NOTE

THE PROGRAM WILL RELOCATE ONLY IN THE FIRST PASS UNDER APT; AFTER THIS, THE PROGRAM WILL REMAIN FIXED IN BANKS O AND 1.

7.3 MEMORY CONFIGURATION MAP

THIS MAP IS PRINTED OUT IMMEDIATELY AFTER SIZING THE MEMORY UNLESS SW6 IS SET (REFERENCE SECTION 2.4.1). IT CAN ALSO BE PRINTED AT ANY LATER TIME IN FIELD SERVICE MODE (REFERENCE SECTION 2.4.4.8.7)

EXAMPLE:

MEMORY CONFIGURATION MAP 16K BANKS

01234567012345

ERRORS
INTRLV ---MEMTYPE PPPP
CSR 4444
PROTECT

DISPLAYED ARE BANKS 0-73 OCTAL (2 MEG WORDS). IF THE FAT TERMINAL SWITCH WAS SET (REFERENCE SECTION 2.4.1) THEN ALL BANKS (0-167) WOULD BE SHOWN. IF THIS WAS AN 18-BIT PDP-11 (EG - 11/34), ONLY BANKS 0-7 WOULD BE PRINTED. THE FIELDS:

ERRORS:

THE SIZING ROUTINE COULD NOT WRITE ZEROS AND ONES IN BANKS 10 11, HENCE THEY ARE MARKED AS BAD WITH X'S.

INTRLV:

THERE IS INTERLEAVING ON BANKS 20-37, WITH CSR 2 (172104) CONTROLLING THE ADDRESS BIT 1 NON-ASSERTED ADDRESSES, AND CSR 3 (172106) CONTROLLING THE ADDRESS BIT 1 ASSERTED ADDRESSES.

ERRORS:

MEMTYPE:

BANKS 0-7 ARE MEMORY TYPE L (MS-11L), AND BANKS 10-37 ARE MEMORY TYPE M (MS11-M) AND BANKS 40-77 ARE MEMORY TYPE P(MS11-P). BANKS 100-167 DO NOT EXIST.

CSR:

BANKS 0-7 ARE ASSIGNED TO CSR 172100, 10-17 TO CSR 172102, AND 20-37 TO INTERLEAVED CSR'S 172104 AND 172106 AND BANKS 40-77 ARE ASSIGNED CSR 17210.

PROTECT:

BANKS O AND 1 ARE PROTECTED BECAUSE THEY ARE PROGRAM SPACE. BANK O AND 1 CAN ALSO BE PROTECTED BECAUSE THEY ARE IN THE BOTTOM 16K OF AN MS11-M CSR. THE PROTECTION IS HIERARCHICAL AND PROGRAM SPACE OVERSHADOWS MS11-M PROTECTION. BANKS O AND 1 WILL NOT BE TESTED UNTIL THE PROGRAM RELOCATES. IF

ANY BANK IS PROTECTED BY MS11-M AND NOT BECAUSE IT IS IN PROGRAM SPACE IT WILL HAVE AN 'I' TYPED IN THIS ROW. THIS IS TO POINT OUT WHERE THE PROTECTED BANKS START FOR EACH ECC CSR. NOTE THE 'P' AT BANK 30; THIS POINTS OUT THE 'SHADOW' PROTECTION WHICH OCCURRS WHEN TWO MS11-M MEMORIES ARE INTERLEAVED. THEREFORE, BANK 30 WILL NOT BE TESTED UNTIL THE PROGRAM HAS RELOCATED.

7.4 EVERYTHING YOU'VE ALWAYS WANTED TO KNOW ABOUT SUPERMAC ...

SUPER-MAC IS A SET OF STRUCTURED PROGRAMMING MACROS THAT ALLOWS PROGRAMS TO BE WRITTEN IN A HIGH LEVEL, EASILY UNDERSTOOD LANGUAGE.

AS A GENERAL RULE, MOST SUPER-MAC STATEMENTS CAN BE SINGLE-LINE STATEMENTS OR MULTIPLE-LINE (NESTED) BLOCK STATEMENTS. A SINGLE-LINE STATEMENT MUST BE COMPLETED ON ONE SOURCE LINE; NO CONTINUATION LINES ARE ALLOWED. SINGLE-LINE STATEMENTS SHOULD BE AS SHORT AND SIMPLE AS POSSIBLE. COMMENTS MAY ALSO BE INCLUDED ON A SOURCE LINE. ALL THE GENERAL RULES, CONDITIONS, ETC., THAT GOVERN MACRO-11 ALSO GOVERN SUPER-MAC. SPACING ON A SOURCE LNE IS VERY IMPORTANT. THE ELEMENTS SHOULD BE SEPARATED BY A COMMA OR A SPACE. TABS SHOULD NEVER BE USED FOR SPACING. FOR EXAMPLE: THE EXPRESSION A+B IS INTERPRETED DIFFERENT THAN A + B.

ALL THE CONDITIONAL STATEMENTS CAN BE WRITTEN AS MULTIPLE-LINE NESTED BLOCKS. EACH LEVEL OF NESTING WITHIN A BLOCK MUST BE TERMINATED WITH AN ASSOCIATED END STATEMENT. EACH LEVEL OF NESTING SHOULD BE INDENTED TWO SPACES.

USER WRITTEN MACROS OR ASSEMBLY LANGUAGE INSTRUCTIONS MAY BE INCLUDED IN A PROGRAM IF DESIRED. AS A DEBUGGING AID, IF THE SYMBOL LST\$\$ IS DEFINED, IT WILL CAUSE GENERATED CODE AND LABELS TO BE LISTED. ALL PROGRAMS MUST BEGIN WITH THE MACRO CALL SMACIT. THIS CALL INITIALIZES SUPER-MAC. ALL LEGAL PDP-11 SOURCE AND DESTINATION OPERANDS ARE LEGAL IN SUPER-MAC.

```
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
4000
4001
4002
                                                                                                                                                                        SAMPLE SOURCE FILE -
.ENABL ABS
.ENABL AMA
.MCALL .SUPER
.SUPER
;LST$$=0
                                                                                                                                              7.4.1
                                                                                                                                                                              BIT5=40
                                                                                                                                              BCDEF:
                                                                                                                                                                               000000000
                                                                                                                                               G:
                                                                                                                                               I:
                                                                                                                                                J:
                                                                                                                                                                               .PAGE
4003
4004
4005
4006
4007
4008
4009
4010
4011
4013
4014
4015
4016
                                                                                                                                               :LET EXAMPLES

LET RO := A
                                                                                                                                                                             LET B := C + D

LET E := F + 1

LET G := H + 2

LET J := J + 0

LET A :B= B
                                                                                                                                              LET A :B= B

:IF EXAMPLES

IF A IS TRUE

MOV 23.D

END :OF IF A

IF B IS FALSE

MOV 34.E

END :OF IF B

IF A EQ B THEN LET C := D

IF A LT B

MOV C.D

FLSE
  4017
 4018
4019
4020
4021
4022
4023
                                                                                                                                                                               ELSE
                                                                                                                                                                             ELSE
MOV E,D
END; OF IF A
IF A EQ B AND C NE D
MOV F,G
END; OF IF A
IF A EQ B OR C NE D
MOV F,G
END; OF IF A
IFB A EQ B AND C EQ 1
MOV H,J
ELSE
                                                                                                                                                                       MOV E,J
END :OF IFB A
IFB A EQ B ANDB C EQ 1
MOV H,J
ELSE
  4032
4033
4034
4035
4036
4037
4038
                                                                                                                                                                                                            E.J
                                                                                                                                                                                       MOV
```

```
END OF IFB A
4040
4041
                                                                                         MOV A,B
END ; OF IF RESULT
IF BITS SET. IN A
                                                                         MOV B,C

END; OF IF BIT5

IF BIT5 OFF. IN A

MOV C,D

END; OF IF BIT5

ON.ERROR IS LIKE AN IF STATEMENT ON THE C-BIT

ON.ERROR EXAMPLES
                                                                                          ON. ERROR
                                                                                              MOV A,B
                                                                                          ELSE
                                                                                         MOV C.B
END : OF ON.ERROR
ON.NOERROR
4056
4057
4058
4059
4060
4061
4062
                                                                                              MOV C.B
                                                                                          ELSE
                                                                                          MOV A,B
END ; OF ON.NOERROR
ON.ERROR THEN LET A :B= B
4063
4064
4065
4066
4067
4068
                                                                          : FOR EXAMPLES
                                                                                          FOR I := -5 TO 23
                                                                                          END : OF FOR I
FOR RO := 0 TO 140 BY 4
4069
 4070
4071
4072
                                                                         DEC A(RO)
END; OF FOR RO
FOR I := 133 DOWNTO 3 BY 2
ADD A,B
END; OF FOR I

;BEGIN EXAMPLES
BEGIN ALPHA
FOR RO := 0 TO 167
MOVB A(RO),B
IF B LT 0 THEN LEAVE ALPHA
END; OF FOR RO
FOR RO := 400 TO 567
IF B GE 0 THEN LEAVE ALPHA
END; OF FOR RO
                                                                                             DEC A(RO)
 4073
 4074
 4080
                                                                                          END OF FOR RO
                                                                           : SRETURN EXAMPLES
                                                                                           $RETURN
                                                                                           SRETURN ERROR
                                                                                           SRETURN NOERROR
                                                                           : CASE EXAMPLES
                                                                                                          A,RO
 4090
4091
4092
                                                                                           MOV
                                                                                           CASE RO
```

```
4094
4095
4096
4097
4098
                                                                                                 BCDE
                                                                                              END : OF CASE RO
                                                                                              .END
4103
4104
4105
4106
                                                                             7.4.2 SAMPLE LISTING FILE (WITH NO EXPANDED MACROS) - - .MAIN. MACRO M1111 01-APR-79 16:41 PAGE 2
4107
                                                                                                                                                                                  .ENABL ABS
                                                                                              000000
4108
                                                                                                                                                                                  . ENABL AMA
4109
                                                                                                                                                                                 .MCALL .SUPER
.SUPER
:LST$$=0
BIT5=40
 4110
                                                                                              000000
 4111
                                                                                                               000040
                                                                                              000000
                                                                                              000002
                                                                                                               000000
                                                                                                              000000
000000
000000
000000
                                                                                              000006
                                                                                                                                                                 E:
                                                                                              000010
                                                                                              000014
                                                                                                                                                                 G:
                                                                                        14 000016
15 000020
16 000022
                                                                                                               000000
                                                                                                                                                                 I:
                                                                                                               000000
                                                                              .MAIN. MACRO M1111 01-APR-79 16:41 PAGE 3
                                                                                                                                                                :LET EXAMPLES

LET RO := A

LET B := C +

LET E := F +

LET G := H +

LET J := J +

LET A :B= B
                                                                                        189012234567890123355
3355
                                                                                              000024
000030
000044
000056
000072
                                                                                                                                                                                              := C + D
                                                                                                                                                               LET J := J + 01
LET A :B = B

:IF EXAMPLES
IF A IS TRUE
MOV 23,D
END :OF IF A
IF B IS FALSE
MOV 34,E
END :OF IF B
IF A EQ B THEN LET C := D
IF A LT B
MOV C,D
ELSE
                                                                                              000100
                                                                                              000106
000114
000122
000130
000136
000136
000154
000164
000172
                                                                                                                                                000006
                                                                                                               012737
                                                                                                                                000023
                                                                                                               012737
                                                                                                                                000034
                                                                                                                                                000010
                                                                                                               013737
                                                                                                                                000004
                                                                                                                                                000006
```

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113	26-APR-82	09:41	PAGE 99		SEQ 0104
4148 4149	36 000174 37 000202	013737	7 000010	000006	MOV E,D END : OF IF A
4150 4151 4152	36 000174 37 000202 38 000202 39 000222 40 000230	013737	7 000012	000014	MOV E,D END; OF IF A IF A EQ B AND C NE D MOV F,G END; OF IF A IF A EQ B OR C NE D MOV F,G END; OF IF A IFB A EQ B AND C EQ 1 MOV H,J ELSE MOV E,J
4153 4154 4155	41 000230 42 000250 43 000256	013737	000012	000014	IF A EQ B OR C NE D MOV F,G END; OF IF A
4156	44 000256 45 000276	013737	000016	000022	IFB A EQ B AND C EQ 1
4158 4159 4160	46 000304 47 000306 48 000314	013737	000010	000022	ELSE MOV E.J END : OF IFB A
4161 4162 4162	49 000314 50 000334	013737	000016	000022	END OF IFB A IFB A EQ B ANDB C EQ 1 MOV H,J ELSE
4164 4165	52 000344 53 000352	013737	000010	000022	MOV F.
4150 4151 4152 4153 4154 4155 4156 4157 4158 4159 4160 4161 4162 4163 4164 4165 4166 4167 4168 4169 4170 4171 4172 4173	48 000314 49 000314 50 000334 51 000342 52 000344 53 000352 54 000352 55 000362 57 000362 58 000372 59 000400	013737	000000	000002	END : OF IFB A IF RESULT IS EQ MOV A.B END : OF IF RESULT
4169 4170 4171	57 000362 58 000372 59 000400	013737	000002	000004	
4174	60 000400 61 000410 62 000416	013737	000004	000006	MOV B,C END :OF IF BITS IF BITS OFF.IN A MOV C,D END :OF IF BITS :ON.ERROR IS LIKE AN IF STATEMENT ON THE C-BIT :ON.ERROR EXAMPLES
4175	63				ON. ERROR IS LIKE AN IF STATEMENT ON THE C-BIT
4178 4179	65 000416 66 000420 67 000426	013737	000000	000002	MOV A.R
4180 4181 4182	68 000430 69 000436 70 000436	013737	000004	000002	ELSE MOV C.B END:OF ON.ERROR ON.NOERROR
4183 4184	71 000440	013737	000004	000002	MOV C,B ELSE
4185 4186 4187	65 000416 66 000420 67 000426 68 000430 69 000436 70 000436 71 000440 72 000446 73 000456	013737	000000	000002	MOV A,B END ; OF ON.NOERROR
4176 4177 4178 4179 4180 4181 4182 4183 4184 4185 4186 4187 4188 4189 4190 4191 4192 4193 4194 4195 4196 4197 4198 4199 4200	N. MACRO	11111 0	1-APR-79	16:41 P	AGE 3-1
4191 4192 4103	75 000456				ON. ERROR THEN LET A :B= B
4194 4195 4196	76 77 000466 78 000474 79 000500 80 000514 81 000516 82 000522 83 000534	005237	000000		FOR EXAMPLES FOR I := -5 TO 23 INC A
4197 4198	80 000514 81 000516	005360	000000		END : OF FOR I FOR RO := 0 TO 140 BY 4 DEC A(RO)
4199	82 000522 83 000534				DEC A(RO) END :OF FOR RO FOR I := 133 DOWNTO 3 BY 2

```
84 000542
85 000550
86
87 000566
88 000566
89 000570
90 000576
91 000604
92 000614
93 000620
94 000626
95 000636
96 97 000636
98 000640
99 000644
                                                                                          ADD A,B
END :OF FOR I

:BEGIN EXAMPLES
BEGIN ALPHA
FOR RO := 0 TO 167
MOVB A(RO),B
IF B LT 0 THEN LEAVE ALPHA
END :OF FOR RO
FOR RO := 400 TO 567
IF B GE 0 THEN LEAVE ALPHA
END :OF FOR RO
                              063737 000000 000002
                              116037 000000 000002
                                                                                                                END OF FOR RO
                                                                                           : SRETURN EXAMPLES
                                                                                                                $RETURN
                                                                                                                 SRETURN ERROR
                                                                                                                 SRETURN NOERROR
                                                                                           : CASE EXAMPLES
100
101 000650
102 000654
103 000664
104 000666
105 000670
106 000672
107 000674
108 000676
109 000700
                                                                                                                MOV
CASE RO
                              013700
                                                  000000
                              000000
000002
000004
                              000006
000010
000012
                                                                                                                END ; OF CASE RO
 110
                               000001
                                                                                                                 .END
111
```

7.4.3 SAMPLE LISTING FILE (WITH EXPANDED MACROS) - - .MAIN. MACRO M1111 01-APR-79 16:10 PAGE 2

.MAIN. MACRO M1111 01-APR-79 16:10 PAGE 3

```
.ENABL ABS
000000
                                                                         .MCALL .SUPER
000000
                                                                         .SUPER
               000000
                                                                         LST$$=0
                                                                         BIT5=40
               000040
000000
000002
000004
               000000
               000000
               000000
000000
000000
000000
000000
000006
000010
000012
000014
000016
000020
                                                                         000000
                                                          H:
                                                           1:
```

4255	18				;LET EXAMPLES LET RO := A MOV A,RO LET B := C + D MOV C,B ADD D,B
4256	18 19 000024 000024	013700	000000		LET RO := A MOV A,RO
4258	20 000030			000002	LET B := C + D
4256 4257 4260 4261 4262 4263 4264 4265 4266 4267 4268 4269 4270 4271 4272 4273 4274 4275 4278 4277 4280 4281 4282 4283 4284 4285 4286 4287 4288 4290 4291 4292 4293 4294 4295 4296	000024 20 000030 000030 000036	013737 063737	000004 000006	000002 000002	ADD D.B
4261	21 000044	013737	000012	000010	LET E := F + 1 MOV F,E INC E
4263	000052	013737 005237	000012 000010		INC E LET G := H + 2
4264 4265	22 000056	013737 062737	000016	000014	LET G := H + 2 MOV H, G ADD 2.G
4266	21 000044 000044 000052 22 000056 000056 000064 23 000072 000072	062737	000002	000014	ADD 2.6 LET J := J + 01
4268	000072	062737	000001	000022	LET J := J + 01 ADD 01,J LET A :B= B MOVB B,A ; IF EXAMPLES IF A IS TRUE TST A BED 10
4269 4270	000100	113737	000002	000000	MOVB B.A
4271	25				; IF EXAMPLES
4273	000106	005737	000000		TST A
4274 4275	27 000112	005737 001403 012737	000023	000006	MOV 23,D
4276	27 000114 28 000122				LO: END ; OF IF A
4278	29 000122	******			IF B IS FALSE
4279 4280	000122	005737 001003 012737	000002		TST B BNE L1
4281	30 000130 31 000136	012737	000034	000010	MOV 34,E END; OF IF B
4282 4283	000136				L1:
4284	32 000136	023737	000000	000002	IF A EQ B THEN LET C := D
4286	000144	023737 001003 013737		000004	CMP A.B BNE L2 MOV D.C
4287 4288	000154	013/3/	000006	000004	L2:
4289	33 000154	023737	000000	000002	IF A LT B
4291	000162	023737			BGE 1.3
4292 4293	34 000164 35 000172	013737	000004	000006	MOV C,D ELSE BR L4
4294	000172	000403			L3:
4296	36 000174 37 000202	013737	000010	000006	MOV E,D END ; OF IF A
4297 4298	000202				14.
4299	38 000202	023737	000000	000002	IF A EQ B AND C NE D
4301	000210	001007			BNE L5
4302	000212	023737 001007 023737 001403 013737	000004	000006	BEQ L5
4299 4300 4301 4302 4303 4304 4305	26 000106 000112 27 000114 28 000122 000122 000122 000123 000136 30 000136 000136 000136 000136 000154 000154 33 000154 000154 34 000162 34 000162 35 000172 000172 000172 000174 36 000172 000174 37 000202 000202 000210 000220 39 000230	013737	000012	000014	IF A EQ B AND C NE D CMP A B BNE L5 CMP C D BEQ L5 MOV F G END ; OF IF A
4303	40 000230				CHD , OF 11 A

4307 4308 4309 4310 4311 4312 4313	41 00023 00023 00023 00024 00024	0 0 0 023737 6 001404 0 023737 6 001403	000000 00000 000004 00000		IF A EQ B OR C NE D CMP A,B BEQ LO CMP C,D BEQ L7
4314 4315 4316 4317 4318 4319 4320	.MAIN. MACRO 00025 42 00025 43 00025	0 012727	APR-79 16:10 000012 00001	4 L6:	MOV F,G END ;OF IF A
4318 4319 4320 4321 4322 4323 4324 4325 4326 4327 4328 4329 4330 4331 4332 4333 4334 4335 4336 4337 4338 4339 4340 4341 4342	42 00025 00025 00025 00025 00026 00026 00027 45 00027 46 00030	146	000000 00000 000004 00000 000016 00002	1	IFB A EQ B AND C EQ 1 CMPB A,B BNE L10 CMP C, 1 BNE L10 MOV H,J ELSE BR L11
4330 4331 4332 4333 4334 4335	47 00030 47 00030 48 00031 00031 00031 00032	013737 4 4	000010 00000	L11:	MOV E.J END ; OF IFB A
4337 4338 4339 4340 4341 4342	00032 00033 50 00033 51 00034 00034	123727 32 001004 34 013737 32 000403	000010 00000 000010 00000	112:	IFB A EQ B ANDB C EQ 1 CMPB A,B BNE L12 CMPB C, 1 BNE L12 MOV H,J ELSE BR L13
4344 4345 4346 4347 4348 4349	00035	2	000000 00000	L13:	MOV E,J END; OF IFB A IF RESULT IS EQ BNE L14 MOV A,B END; OF IF RESULT
4343 4344 4345 4346 4347 4348 4349 4350 4351 4352 4353 4354 4355 4356 4357 4358 4358	54 00033 00033 55 00033 56 00036 00036 00037 58 00037 59 00046 60 00046	032737 70 001403 72 013737 00 00	000040 00000 000002 00000	00	IF BITS SET.IN A BIT BITS, A BEQ L15 MOV B, C END; OF IF BITS IF BITS OFF.IN A
4358 4359	00040		000040 00000	00	BIT BITS,A

			E 9		SEQ 010
ZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M111	3 26-APR-8	2 09:41 F	PAGE 103		324 010
4361	61 00041	0 013737	000004	000006	MOV C,D
4362	62 00041	6			END : OF IF BITS
4364	62 00041	6			L16:
4365	63 64 65 00041				ON EDROR EXAMPLES
4367	65 00041 00041	6			ON.ERROR
4368	66 00042	6 103004 0 013737	000000	000002	MOV A.B
4370	67 00042	6	000000	000002	ELSE
4371	00042 00043	6 000403			L17:
43/2	00043	0			
4374	68 00043	0 013737	000004	000002	MOV C,B
4375 4376	69 00043	6			END : OF ON. ERROR
4377	69 00043 00043 70 00043	6			L20:
4378	70 00043	6			ON.NOERROR
4380					사람이 그렇게 하는 것이 그리고 하는 때문에 되었다.
4381 .M	MACRO	M1111 0	1-APR-79	16:10 P	PAGE 3-2
4382 4383					
4384	00043	6 103404 0 013737	00000/	000003	BCS L21
4385 4386	71 00044 72 00044	6 013/3/	000004	000002	MOV C,B
4387	00044	6 000403			BR L22
4388	73 00045	0 013737	000000	000002	L21: MOV A,B
4390	74 00045	6	000000	00000	END ; OF ON.NOERROR
4391	00045				ON. ERROR THEN LET A :B= B
4392 4393	75 00045 00045	6 103003			BCC L23 MOVB B.A
4394	00045 00046 00046	6 103003 0 113737	000002	000000	MOVB B,A
4395 4396	76	0			L23: ;FOR EXAMPLES
4397	76 77 00046	6	42222	000000	FOR I := -5 TO 23 MOV -5,I
4398	00046	6 012737	177773	000020	BO:
4400	78 00047	4 005237	000000		TNC A
4401	79 00050	0 005237			END OF FOR I
4402	00050	005237 4 023727 2 003770	000020 000020	000023	CMP 1 23 BLE BÓ
4404	00051	2 003770			BLE BO
4405	80 00051	2			EO: FOR RO := 0 TO 140 BY 4
4407	00051	4 005000			CLR RO
4408	81 00051	6 005360	000000		B1: DEC A(RO)
4410	82 00052	2			END OF FOR RO
4411	00052	2 062700	000004		CMP RO 140
4361 4363 4364 4365 4366 4367 4368 4370 4371 4372 4373 4375 4376 4377 4378 4379 4380 4381 4382 4383 4384 4385 4386 4387 4388 4389 4390 4391 4392 4393 4394 4395 4396 4397 4398 4399 4400 4401 4402 4403 4404 4405 4406 4407 4408 4409 4410 4411 4411	76 77 00046 00047 78 00047 79 00050 00050 00051 00051 80 00051 00051 81 00051 82 00052 00052	2 062700 6 020027	000004 000140		END OF FOR RO

ZMSPAU MSTI-L/M/P MEMURY DIAG.	MACRU MITTS 20	-APK-02	07.41	PAGE 104			
4414 4415		000532 000534	003771			E1:	BLE B1
4416 4417	83	000534 000534 000542 000542	012737	000133	000020		FOR I := 133 DOWNTO 3 BY 2 MOV 133,I
4418 4419	84 85	000542 000542	063737	000000	000002	B2:	ADD A.B
4418 4419 4420 4421 4422 4423 4424 4425 4426 4427 4428 4429 4430 4431 4432 4435 4436 4437 4438 4439 4440 4441 4442 4443 4444 4445 4446 4447	85	000550 000550 000556 000564 000566	162737 023727 002366	000002	000020 000003		ADD A,B END ; OF FOR I SUB 2,I CMP I, 3 BGE B2
4424	AK					E2: ;BEGIN	
4426	8/	000566				B3:	BEGIN ALPHA
4428 4429	88	000566 000566 000566 000570 000570	005000				FOR RO := 0 TO 167
4430 4431 4432	89	000570 000570 000576	116037	000000	000002	B4:	MOVB A(RO), B IF B LT O THEN LEAVE ALPHA
4433 4434	,	000576 000602	005737 002415	000002			TST R
4435 4436 4437	91	000576 000576 000602 000604 000604 000606 000612	005200 020027 003766	000167			BLT E3 END : OF FOR RO INC RO CMP RO, 167 BLE B4
4438		000612	003766			E4:	BLE 84
4440 4441	92	000614 000614	012700	000400			FOR RO := 400 TO 567 MOV 400,RO
4443	.MAIN.	MACRO I		1-APR-79	16.10	PAGE 3-3	
4445 4446				I-AFR-77	10.10		
4448	93	000620	005777	000003		B5:	IF B GE O THEN LEAVE ALPHA
4449 4450 4451	94	000624 000626	005737 002004				TST B BGE E3 END : OF FOR RO
4449 4450 4451 4452 4453 4454 4455 4456 4457 4458 4459 4460 4461 4462 4463 4464 4465	, and a second	000620 000624 000626 000630 000634 000636 000636	005200 020027 003771	000567			END ; OF FOR RO INC RO CMP RO, 567 BLE B5
4454		000634	003771			E5:	BLE BS
4456	95	000636					END ALPHA
4457 4458	96	000000				;\$RETU	RN EXAMPLES
4459	96 97	000636	000207	91 4.			\$RETURN RTS PC
4461	98	000640	000201				SRETURN ERROR
446Z 4463		000640	000261 000207				SEC RTS PC
4464	99	000636 000636 000640 000642 000644 000644	000241				SRETURN NOERROR
4400		000644	000241 000207				CLC RTS PC

ZMSPAO	MS11-L/M/P MEMORY	DIAG. MACRO MITTS	20-APK-02	09:41 P	AGE 105			
4468 4469 4470			100 101 000650 102 000654 000654	013700 010046	000000	; CASE I	EXAMPLES MOV A,RO CASE RO MOV RO,-(SP) ASL asp	
4468 4477 4477 4477 4477 4477 4477 4487 4488 4488 4488 4491 4493 4495			000656 000660 103 000664 104 000666 105 000670 106 000672 107 000674	010046 006316 004737 000000 000002 000004 000006 000010	000700		ASL aSP JSR PC,L24 A B C D	
4479			108 000676 109 000700	000012		L24:	END ; OF CASE RO	
4481 4482 4483 4484			000700 000700 000702 000704	062616 013646 004736		L24:	ADD (SP)+,aSP MOV a(SP)+,-(SP) JSR PC,a(SP)+	
4485 4486 4487			110 111	000001			.END	
4488 4489		7.5	MEMORY MAN	AGEMENT	MAPPING			
4490 4491		7.5.	1 MEMORY M	ANAGEMEN	T MAPPING FOR	THE 11/44		
4492		PAR		SUPERVI	SOR	KERNEL		US
4494		0		PROGRAM		PROGRA		DS

PAR	SUPERVISOR	KERNEL	USER
0	PROGRAM	PROGRAM	DST BK/FST MEM
1	PROGRAM	PROGRAM	SRC BK/FST MEM
2	PROGRAM	PROGRAM	SRC BK/FST MEM
3	TEST AREA	PROGRAM	SRC BK/FST MEM
4	TEST AREA	PROGRAM	DST BK/FST MEM
5	TEST AREA	PROGRAM	DST BK/FST MEM
6	TEST AREA	MAP TO CSR'S	DST BK/FST MEM
7	PERIF PAGE	PERIF PAGE	DST BK/FST MEM

7.5.2 MEMORY MANAGEMENT MAPPING FOR UNIBUS-11'S WITH SUPERVISOR MODE (EG 11/45) -

PAR	SUPERVISOR	KERNEL	USER
0	PROGRAM	PROGRAM	DST BK
1	PROGRAM	PROGRAM	SRC BK
2	PROGRAM	PROGRAM	SRC BK
3	TEST AREA	PROGRAM	SRC BK
Z	TES AREA	PROGRAM	DST BK
5	TEST AREA	PROGRAM	DST BK
ź	TEST AREA	MAP TO CSR'S	DST BK
7	PERIF PAGE	PERIF PAGE	DST BK

4519	
7520	
7551	
7555	
7558	
7557	
7555	
4262	
4220	
4521	
4220	
4229	
4530	
4551	

7.5.3 MEMORY MANAGEMENT MAPPING FOR UNIBUS-11'S W/O SUPERVISOR MODE (EG 11/34) -

PAR	KERNEL	USER
0	PROGRAM	PROGRAM/DST BK
ĭ	PROGRAM	PROGRAM/SRC BK
5	PROGRAM	PROGRAM/SRC BK
2	PROGRAM	TEST AREA/SRC BK
Ž.	PROGRAM	TEST AREA/DST BK
2	PROGRAM	TEST AREA/DST BK
2	MAP TO CSR'S	TEST AREA/DST BK
7	PERIF PAGE	PERIF PAGE/DST BK"

4535 4536 000000 4537 4538 4539 4540 4541 4542 4543 4544 4545 4546 4547 4548 4549 4550 4551 4552 4553 4554 4555 4556 000000	THIS PROGRAM. ALL THESE MCALL SMACITPUSH,PO MCALL IFOPRISGENBR, MCALL RNE,REQ,RLT,RGE,RG MCALL IFOR,.IFARILEA MCALL FOR,TO,DOWNTO,REPE MCALL \$\$FND.LEAVE.JUMPTO	SUPER.MAC SOURCE AND IS RELEASED WITH .MCALL STATEMENTS REFERENCE THAT FILE. PTAGBRANEMITEMITNEMITLEMITR .OPADDOPSUB.CLEAR.SET.CLEARB.SETB .T.RLE.RPL.RMI.RHI.RLOS.RHIS.RLO.RCS.RCC VEGOTO.OR.AND.THEN.ELSE.WHILE.CASE AT.UNTIL.THRU.END.BEGIN .GOTO.PUSH.POP.LET .ANDB.IFB.UNTILB.WHILEB.ON.ERROR.ON.NOERROR
4549 4550 4551 4552 4553 4554 163000 4555 000000	.NLIST TTM .LIST MC,SYM .NLIST MD,CND,ME LST\$\$= 0 \$SWR= 163000 \$TN= 1 SMACIT	; I WANT FAT PAPER! ;LIST MACRO CALLS, SYMBOL TABLE ;DON'T LIST MACRO DEFS & CONDITIONALS & EXPANSIONS ;DEFINED TO LIST SUPERMAC EXPANSIONS ;USE THESE SYSMAC SWITCHES ;FIRST TEST NUMBER TO ONE(1)

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 110
DEFINE TRAPS
                                                                 SBITL DEFINE TRAPS

;ALL ENTRIES HERE MUST HAVE A CORRESPONDING ENTRY IN THE ;TRAP TABLE 'STRPAD' (NEAR END OF PROGRAM).
     4560
4561
4562
4563
                                                                  :*TRAP DEFINITIONS
                                                                  HERE IS HOW TRAPS WORK IN THIS PROGRAM
     4565
4566
4567
4568
4569
4571
4572
4573
4574
4577
4578
4581
4583
4584
4585
                                                                  :ALL TRAPS EXECUTE A "TRAP" INSTRUCTION WHICH TAKES THE PROGRAM :TO SYMBOLIC LOCATION "STRAP"
                                                                 AT STRAP THE PROGRAM PICKS UP THE RIGHT BYTE OF THE TRAP INSTRUCTION AND INDEXES INTO A TABLE AT LOCATION "STRPAD" WHICH SENDS THE PROGRAM TO
                                                                  THE SPECIFIC ROUTINE TO HANDLE THAT SPECIFIC TRAPS TASK.
                                                                  THE ULTIMATE DESTINATION OF A TRAP INSTRUCTION CAN BE GUESSED AT AS FOLLOWS
                                                                  :EXAMPLE:
                                                                                            NOP
                                                                                            NOP
                                                                                            KERNEL
                                                                                                                                   :ENTER KERNEL MODE
                                                                                            NOP
                                                                               ADD A DOLLAR SIGN TO THE SYMBOLIC NAME AND CHECK THE CRF FOR SOMETHING CLOSE IN THIS CASE THE CRF HAS $KERNE LISTED AS 032546 AT LOCATION 32546 YOU FIND THE ROUTINE $KERNEL
                                                                               NOTE THAT CRF SYMBOLS ARE TRUCNATED TO 6 CHARACTERS SYMBOLIC NAMES GREATER THAT 6 CHARACTERS ARE USED SO I CAN
     4586
4587
4588
4589
4590
4591
4592
                                                                               REMEMBER WHAT THEY MEAN!
                                                                  TYPEIT= 104401 TRAP ROUTINES RETURN VIA AN 'RTI' INSTRUCTION
                           104401
                                                                 TYPOC= 104402
TYPOS= 104403
;TYPON= 104404
TYPDS= 104405
;TYPBN= 104406
                                                                                                         TYPE OCTAL NUMBER (WITH LEADING ZEROS)
TYPE OCTAL NUMBER (NO LEADING ZEROS)
TYPE OCTAL NUMBER (AS PER LAST CALL)
TYPE DECIMAL NUMBER (WITH SIGN)
                           104402
      4593
4594
4595
4596
4597
4598
                           104405
                                                                                                          :: TYPE BINARY (ASCII) NUMBER
                           104407
                                                                  GTSWR= 104407
CKSWR= 104410
                                                                                                          ::GET SOFT-SWR SETTING
                                                                                                          :: TEST FOR CHANGE IN SOFT-SWR
      4600
4601
4602
4603
4604
4605
                                                                               104411
104412
104413
104414
                                                                                                         ::TTY TYPEIN CHARACTER ROUTINE
::TTY TYPEIN STRING ROUTINE
::READ AN OCTAL NUMBER FROM TTY
                           104411
104412
104413
                                                                  RDCHR=
                                                                  RDLIN=
                                                                  RDOCT=
                                                                                                          :: READ A DECIMAL NUMBER FROM TTY
                           104414
                                                                  RDDEC=
                                                                  SAVREG= 104415
RESREG= 104416
                                                                                                          ;; SAVE RO-R5 ROUTINE
                           104415
104416
                                                                                                          :: RESTORE RO-R5 ROUTINE
      4607
4608
4609
4610
4611
4613
                                                                  KERNEL= 104417
                                                                                                          ENTER KERNEL MODE
                           104417
                           104420
104421
104422
                                                                                                          TURN ON MEMORY MANAGEMENT & TRAPS
                                                                  ENERGIZE=104420
                                                                  DEENERGIZE=104421
KMAP= 104422
                                                                                                          TURN OFF MEMORY MANAGEMENT & TRAPS
                                                                                                          :MAP KERNEL 1 TO 1
                                                                                                          :TURN ON CACHE
      4614
                                                                  CACHON= 104423
CACHOFF=104424
                                                                                                          :TURN OFF CACHE
```

DEFINE	TRAPS			
4616 4617 4618 4619		104425 104426	LOADCSR=104425 READCSR=104426	:LOAD CORRECT CSR :READ CORRECT CSR
4623 4623 4623 4623 4623 4623 4623 4623		104427 104431 104432 104433 104434 104435 104436 104437 104444 104442 104443 104444 104445 104445 104451 104451 104453 104454 104455 104456 104461 104462 104463 104463 104464 104465	PERRO1= 104427 PERRO2= 104430 PERRO3= 104431 PERRO4= 104432 PERRO7= 104433 PERR10= 104434 PERR11= 104436 PERR12= 104436 PERR13= 104437 PERR14= 104440 PERR15= 104441 PERR16= 104442 PERR17= 104443 PERR20= 104446 PERR21= 104457 PERR24= 104451 PERR25= 104454 PERR25= 104454 PERR31= 104455 PERR31= 104455 PERR31= 104457 PERR31= 104466 PERR35= 104463 PERR36= 104463 PERR41= 104466 PERR43= 104466 PERR43= 104467	PROGRAM DETECTED ERROR
465 465 465 465 465 465 466 466 466 466	0 1 2 3 4 5 6 7 8 9 0	104470 104471 104472 104473 104474 104475 104476 104477 104500 104501 104502 104503 104504 104505 104506 104507 104510 104511	ECCDIS= 104470 ECC1DIS=104471 ECCINIT=104472 ECC1INIT=104473 CBCSR= 104474 CB1CSR= 104476 WASSBE= 104476 WASSBE=104477 WASDBE= 104500 WAS1DBE=104501 CLRCSR= 104502 CLR1CSR=104503 CHKDIS= 104504 CHK1DIS=104505 ENASBE= 104506 ENA1SBE=104507 TSTREAD=104511 ERRGEN =104512	DISABLE ECC ON ALL CSR'S DISABLE ECC ON 1 SELECTED CSR INITIALIZE ALL ECC CSR'S INITIALIZE 1 SELECTED ECC CSR WRITE GENERATED CHECKBITS IN ALL CSR'S WRITE GENERATED CHECKBITS IN 1 SELECTED CSR WAS THERE A SBE ON ANY CSR? WAS THERE A SBE ON 1 SELECTED CSR? WAS THERE A DBE ON ANY CSR? WAS THERE A DBE ON 1 SELECTED CSR? CLEAR ALL CSR'S CLEAR 1 SELECTED CSR DISABLE ECC & WRITE CHECKBITS FROM ALL CSR'S DISABLE ECC & WRITE CHECKBITS FROM 1 SELECTED CSR ENABLE TRAPS ON SBE'S FROM ALL CSR'S ENABLE TRAPS ON SBE'S FROM 1 SELECTED CSR TEST LOC (R1) & TST FOR SBE (WITHOUT FETCHES) INVALIDATE BACKGROUND PATTERN ON 'BANK' CHECK ERROR ADDRESS

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 110-2 DEFINE TRAPS

104513 4673

CBREG =104513 ; ENABLES CHECK/SYNDROME BIT REGISTER

```
.SBTTL DEFINE BASIC PDP11 STUFF
4675
4676
4677
4678
4679
                                                                  :*INITIAL ADDRESS OF THE STACK POINTER
STACK= 2000 ::FIRST ADDRESS
KERSTK= STACK ::KERNEL STACK
SUPSTK= 740 ::SUPERVISOR ST
                                                                                                              ::FIRST ADDRESS OF THE STACK
                       002000
000740
000700
                                                                                                              SUPERVISOR STACK
SUSER STACK
BASIC DEFINITION OF ERROR CALL
BASIC DEFINITION OF SCOPE CALL
PROCESSOR STATUS WORD
STACK LIMIT REGISTER
PROGRAM INTERRUPT REQUEST REGISTER
4680
4681
                                                                   USESTK= 700
                                                                   ERROR=EMT
4682
                        104000
                                                                   SCOPE=IOT
4683
                        000004
                                                                   PSW= 177776
;STKLMT=177774
;PIRQ= 177772
DSWR= 177570
DDISP= 177570
4684
                        177776
4685
4686
4687
4688
                       177570
177570
177546
                                                                                                              : HARDWARE SWITCH REGISTER : HARDWARE DISPLAY REGISTER
                                                                                                               ::LINE CLOCK (KW11-L) STATUS REGISTER
                                                                                  177546
4689
                                                                   LKS=
4690
                                                                    : *MISCELLANEOUS DEFINITIONS
4691
                                                                                 11
12
15
200
7
                                                                                                              :: CODE FOR HORIZONTAL TAB
:: CODE LINE FEED
4692
4693
                        000011
                                                                   HT=
                       000012
000015
000200
                                                                   LF=
                                                                                                              CODE CARRIAGE RETURN
CODE FOR CARRIAGE RETURN-LINE FEED
CODE FOR PROCESSOR TYPE INSTRUCTION
                                                                   CR=
4694
                                                                   CRLF=
4695
                                                                   MFPT=
4696
4697
                        000007
                                                                    :*GENERAL PURPOSE REGISTER DEFINITIONS
4698
                                                                                                              ::STACK POINTER
::KERNEL STACK POINTER
::SUPERVISOR STACK POINTER
4699
4700
4701
4702
4703
4704
4705
4708
4709
4711
4712
4713
4716
4717
4718
4721
4721
4721
4723
4723
4728
4728
4729
4731
                                                                    :SP=R6
                                                                    :KSP=SP
                                                                    SSP=SP
                        000006
                                                                                                               :: USER STACK POINTER
                                                                   USP=SP
                        000006
                                                                                                               :: PROGRAM COUNTER
                                                                   :PC=R7
                                                                   ;*'SWITCH REGISTER' SWITCH DEFINITIONS
SW15= 100000
                        100000
                        040000
020000
010000
004000
002000
                                                                   SW14=
                                                                                  40000
                                                                    SW13=
                                                                                  20000
                                                                   SW12=
SW11=
                                                                                   10000
                                                                                  4000
                                                                                  2000
1000
400
200
100
                                                                    SW10=
                        001000
                                                                    SW9=
                        000400
000200
                                                                    SW8=
                                                                    SW7=
                        000100
                                                                    SW6=
                        000040
000020
                                                                    SW5=
                                                                    SW4=
                                                                    SW3=
                        000010
                        000004
                                                                    SW2=
                                                                    SW1=
                        000002
                        000001
                                                                    SWO=
                                                                    **DATA BIT DEFINITIONS (BIT00 TO BIT15)
BIT15= 100000
                        100000
040000
020000
                                                                   BIT15=
                                                                                  40000
                                                                   BIT14=
BIT13=
                                                                   BIT12=
BIT11=
                                                                                  10000
                        010000
                                                                                  4000
                        004000
                        002000
                                                                    BIT10=
                         001000
                                                                                   1000
                                                                    BIT9=
                                                                                   400
                                                                    BIT8=
                         000400
```

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 111-1
DEFINE BASIC PDP11 STUFF
                                                                        200
                        000100
000040
000020
                                                            BIT6=
BIT5=
                                                                        40
20
10
                                                            BIT4=
BIT3=
BIT2=
                       000020
000010
000004
000002
000001
                                                            B111=
                                                            BITO=
                                                             *BASIC 'CPU' TRAP VECTOR ADDRESSES
                       000004
                                                            ERRVEC= 4
RESVEC= 10
                                                                                                ::TIME OUT AND OTHER ERRORS
                                                                                                RESERVED AND ILLEGAL INSTRUCTIONS
                                                            :TBITVEC=14
                                                            :TRTVEC=
                                                                                                            ::TRACE TRAP
::BREAKPOINT TRAP (BPT)
    :BPTVEC=
                                                            IOTVEC= 20
PWRVEC= 24
EMTVEC= 30
                                                                                                ::INPUT/OUTPUT TRAP (IOT) **SCOPE**
                        000020
                        000024
                                                                                                : : POWER FAIL
                                                                                                :EMULATOR TRAP (EMT) **ERROR**
:'TRAP'' TRAP
:TTY KEYBOARD VECTOR
:TTY PRINTER VECTOR
:LINE CLOCK (KW11-L) VECTER
:CACHE ERROR INTERRUPT VECTOR
                        000034
                                                            TRAPVEC=34
                                                            TKVEC= 60
                        000060
                                                            :TPVEC= 64
:LKVEC= 100
                        000114
                                                            CACHVEC=114
                                                           PARVEC=CACHVEC
;PIRQVEC=240
MMVEC= 250
                        000114
                                                                                                ;;PROGRAM INTERRUPT REQUEST VECTOR
;;MEMORY MANAGEMENT VECTOR
CACHE REGISTERS
                        000250
                                                            MEMERR = 177744
                                                                                   DEFINE
                                                                                                            :: CACHE ERROR REGISTER
:: MEMORY CONTROL REGISTER
                                                           CONTRL = 177746
MAINT = 177750
;HITMIS = 177752
DATARG = 177754
                        177746
177750
                                                                                                            :: MEMORY MAINTENENCE REGISTER
:: HIT MISS REGISTER '1' IMPLIES HIT IN CACHE
                        177754
                                                                                                            ::DATA REGISTER
                                                           CPUERR = 177766 CPU REGISTERS
                       177766
                                                                                                ;; CPU ERROR REGISTER HOLDS CONDITION THAT CAUSED
                                                                        .SBTTL DEFINE MEMORY MANAGEMENT REGISTERS
                                                                       **MEMORY MANAGEMENT STATUS REGISTER ADDRESSES
177572
177574
177576
172516
                       177572
177574
177576
172516
                                                            MMRG=
                                                            MMR1=
                                                            MMR2=
MMR3=
                                                                                  "I" PAGE DESCRIPTOR REGISTERS
                                                           UIPDR0= 177600
                       177600
                                                            :UIPDR1=
:UIPDR2=
                                                                                    177602
177604
177606
                                                            :UIPDR3=
                                                            :UIPDR4=
                                                            :UIPDR5=
                                                            :UIPDR6=
                                                            :UIPDR7=
                                                                                    177616
                                                                                 'D" PAGE DESCRIPTOR REGISTORS
177620
177622
177624
                                                                        : *USER
                                                            :UDPDR0=
:UDPDR1=
                                                            :UDPDR2=
```

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 111-2
DEFINE MEMORY MANAGEMENT REGISTERS
                                                                                   177626
177630
177632
177634
    4789
4790
4791
                                                            :UDPDR3=
                                                            :UDPDR4=
                                                            :UDPDR5=
                                                            :UDPDR6=
                                                            :UDPDR7=
                                                          ;*USER ''I'' PAGE ADDRESS REGISTERS

FASTCITY=UIPARO
UIPARO= 177640 ;PATTERN PROGRAM S
UIPAR1= 177642 ;PATTERN PROGRAM S
UIPAR2= 177644 ;PATTERN PROGRAM S
UIPAR3= 177646 ;PATTERN PROGRAM S
                       177640
177642
177644
177644
177646
177650
177652
                                                                                               :PATTERN PROGRAM SPACE
:PATTERN PROGRAM SPACE
                                                                                               : PATTERN PROGRAM SPACE
                                                                                                PATTERN PROGRAM SPACE
                                                           UIPAR4= 177650
UIPAR5= 177652
UIPAR6= 177654
                                                                                                PATTERN PROGRAM SPACE
                                                                                                : PATTERN PROGRAM SPACE
                                                                                               PATTERN PROGRAM SPACE
                                                                                                           :PATTERN PROGRAM SPACE
                                                                                   177656
                                                            :UIPAR7=
                                                                                 'D' PAGE ADDRESS REGISTERS ; PATTERN PROGRAM SPACE
                                                           UDPAR0= 177660
                        177660
                                                                                   177662
                                                            :UDPAR1=
                                                                                                           :PATTERN PROGRAM SPACE
                                                                                   177664
177666
177670
177672
177674
                                                            :UDPAR2=
:UDPAR3=
                                                                                                           :PATTERN PROGRAM SPACE
     4809
     4810
4811
4812
4813
4814
4815
                                                                                                           :PATTERN PROGRAM SPACE
                                                                                                           PATTERN PROGRAM SPACE
PATTERN PROGRAM SPACE
PATTERN PROGRAM SPACE
                                                            :UDPAR4=
                                                            :UDPAR5=
                                                            :UDPAR6=
                                                                                               :PATTERN PROGRAM SPACE
                                                            UDPAR7= 177676
                        177676
                                                           SIPDRO= 172200 'I" PAGE DESCRIPTOR REGISTERS
     4816
4817
4818
4819
4820
                        172200
                                                                                   172202
172204
172206
172210
                                                            :SIPDR1=
                                                            :SIPDR2=
                                                            :SIPDR3=
                                                            :SIPDR4=
                                                            SIPDR5=
SIPDR6=
                                                            :SIPDR7=
                                                                        **SUPERVISOR "D" PAGE DESCRIPTOR REGISTERS
                                                            :SDPDRO=
                                                            :SDPDR1=
                                                            :SDPDR2=
                                                            :SDPDR3=
                                                            :SDPDR4=
                                                            :SDPDR5=
                                                            :SDPDR6=
                                                                                    172236
                                                            :SDPDR7=
                                                            SIPARO= 172240 "I" PAGE ADDRESS REGISTERS
                        172240
                                                            ;SIPAR1=
;SIPAR2=
SIPAR3= 172246
                                                                                    172242
172244
                                                                                               ;TEST AREA
                        172246
                                                                                    172250
                                                                                                           TEST AREA
                                                             :SIPAR4=
                                                            SIPARS= 172252
SIPAR6= 172254
                                                                                                            :TEST AREA
                                                                                    172256
                                                            :SIPAR7=
```

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 111-3 DEFINE MEMORY MANAGEMENT REGISTERS

CI TIAL	HEMORI IMMOCREMI NEGIGIENO	
4846 4847	172260	SDPARO= 172260 **SUPERVISOR 'D' PAGE ADDRESS REGISTERS
4848 4849 4850 4851 4852		;SDPAR1= 172262 ;SDPAR2= 172264 ;SDPAR3= 172266 ;SDPAR4= 172270
4851 4852	172272	SDPAR5= 172272
4853 4854 4855	172274 172276	SDPAR6= 172274 SDPAR7= 172276
4856 4857	172300	*KERNEL "I" PAGE DESCRIPTOR REGISTERS KIPDRO= 172300 ;KIPDR1= 172302
4859 4860		;KIPDR2= 172304 :KIPDR3= 172306
4861 4862		KIPDR4= 172310 KIPDR5= 172312 KIPDR6= 172314
4856 4857 4858 4859 4860 4861 4863 4864 4865		;KIPDR7= 1/2316
4866 4867 4868		:*KERNEL 'D'' PAGE DESCRIPTOR REGISTERS :KDPDR0= 172320 :KDPDR1= 172322
4867 4868 4869 4870		:KDPDR2= 172324 :KDPDR3= 172326
4871 4872 4873		KDPDR0
4874 4875 4876		
4877	172340	KIPARO= 172340
4878 4879 4880 4881	172350	;KIPAR1= 172342 ;KIPAR2= 172344 ;KIPAR3= 172346 KIPAR4= 172350
4882 4883	172350 172352 172354	KIPAR4= 172350 KIPAR5= 172352 KIPAR6= 172354 ;KIPAR7= 172356
4885 4886		:*KERNEL 'D' PAGE ADDRESS REGISTERS
4387 4888 4889	172360	KDPAR0= 172360 ;KDPAR1= 172362 ;KDPAR2= 172364
4890 4891		;KDPAR3= 172366 ;KDPAR4= 172370
4882 4883 4884 4885 4886 4889 4891 4892 4893 4893	172374 172376	*KDPAR5= 172372 KDPAR6= 172374 KDPAR7= 172376
4895		

```
.SBTTL DEFINE UNIBUS MAP REGISTERS
;*THE LOWER 16 BITS OF THE MAP REGISTERS ARE LABELED 'MAPLXX'
;*THE UPPER 6 BITS OF THE MAP REGISTERS ARE LABELED 'MAPHXX'

MAPL0 = 170200
MAPH0 = 170202
MAPL1 = 170204
;MAPH1 = 170210
;MAPH2 = 170212
;MAPH3 = 170216
;MAPH3 = 170216
;MAPL4 = 170220
4898
4899
4900
4901
4902
4903
                                            170200
170202
170204
4908
4909
                                                                                                                                 :MAPL4 = 170220
                                                                                                                                 :MAPH4 = 170222
:MAPL5 = 170224
4911
                                                                                                                                 MAPHS = 170226
MAPL6 = 170230
4912
 4913
                                                                                                                                 MAPH6 = 170232
MAPL7 = 170234
MAPH7 = 170236
4914
 4915
 4916
                                                                                                                                  :MAPL10 = 170240
 4917
                                                                                                                                  MAPH10 = 170242
MAPL11 = 170244
 4918
 4919
                                                                                                                                   :MAPH11 = 170246
                                                                                                                                 :MAPL12 = 170250
:MAPH12 = 170252
:MAPL13 = 170254
                                                                                                                                  :MAPH13 = 170256
                                                                                                                                  :MAPL14 = 170260
:MAPH14 = 170262
                                                                                                                                 MAPL14 = 170260
MAPH14 = 170262
MAPH15 = 170264
MAPH15 = 170270
MAPH16 = 170272
MAPH17 = 170274
MAPH17 = 170276
MAPH20 = 170300
MAPH20 = 170300
MAPH21 = 170304
MAPH21 = 170310
MAPH22 = 170312
MAPH23 = 170314
MAPH24 = 170320
MAPH25 = 170320
MAPH26 = 170320
MAPH26 = 170330
MAPH27 = 170336
MAPH27 = 170336
MAPH28 = 170336
MAPH29 = 170346
MAPH30 = 170346
MAPH31 = 170346
                                                                                                                                  :MAPH31 = 170346
:MAPL32 = 170350
:MAPH32 = 170352
```

17.45			E 10	SEQ 0121
CZMSPAO DEFINE	MS11-L/M/P MEMORY DIAG. UNIBUS MAP REGISTERS	MACRO M1113 26-APR-82 09:41	PAGE 113-1	
4955 4956 4957 4958 4959 4960 4961 4962 4963 4964 4965 4966 4969 4970 4971 4973 4974 4975		:MAPL33 = 170354 :MAPH33 = 170356 :MAPL34 = 170360 :MAPH34 = 170362 :MAPL35 = 170364 :MAPH35 = 170366 :MAPL36 = 170370 :MAPH36 = 170372 :MAPH37 = 170374 :MAPH37 = 170376		
4966 4967 4968	000174 000176	DISPREG=174 SWREG= 176	NE SOFTWARE SWITCH & DISPLAY REGISTERS	
4969 4970 4971	172100	CSRADD=172100 DEFI	NE CONTROL STATUS REGISTERS	
4973 4974 4975 4976	060000 157776 040000	SBTTL DEFI FIRST=60000 LAST=157776 SIZE=40000	PARAMETERS START OF THE 16K TEST PATTERN AREA END OF THE 16K TEST PATTERN AREA SIZE OF THE 16K TEST PATTERN ARE	EA A (FOR SOB INSTRUCTIONS)

```
.LIST MD ;BE NICE TO SEE M
.SBTTL MACRO FATAL
4979
4980
4981
4982
4983
4984
                                                                                              BE NICE TO SEE MY DEFINITIONS
                                             FATAL IS USED TO REPORT FATAL ERRORS (ERRORS THAT PREVENT THE PROGRAM FROM CONTINUING).
.MACRO FATAL ARG
                                                                                             : ***MACRO***MACRO***
                                                       .NLIST
                                                       .DSABL CRF
.IIF DF LST$$ .LIST ME
.ENABL CRF
                                                       LIST
                                                                                 ;SET FATAL INDICATOR
                                                                FATALS
                                                       ERROR +ARG
                                                       .DSABL CRF
                                                       .IIF DF LST$$ .NLIST ME .ENABL CRF
                                                       .ENDM
                                                               FATAL
                                                       .SBTTL MACRO
                                                                         TYPE
                                                                          ARG
                                                        .NLIST
                                                       .DSABL CRF
.IIF DF LST$$ .LIST ME
.ENABL CRF
                                                       .LIST
.IF B ARG
                                                       TYPEIT
                                                        .IFF
                                                       TYPEIT ,ARG
                                                       .ENDC
                                                       .DSABL CRF
.IIF DF LST$$ .NLIST ME
.ENABL CRF
.ENDM TYPE
```

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 117
MACRO NEWTST
   5018
5019
5020
5021
                                                             .SBTTL MACRO
                                                                                NEWTST
                                                   ****************** NEWIST ******
                                                  NEWTST IS USED AS THE FIRST INSTRUCTION OF A TEST.
                                                  : II WILL:
                                                        GENERATE A TEST NUMBER FOR THE LABEL OF THIS TEST
                                                  :2) PUT STARS BEFORE AND AFTER A MESSAGE
                                                  : ARGUMENTS
                                                                      THIS IS THE MESSAGE THAT WILL APPEAR
                                                  ;1) ASCII
                                                                      ON THE LISTING
                                                                       IF NON-BLANK AND BIT 11 OF $SWR = 1 IT WILL BE
                                                  :2) ICOUNT --
                                                                      THE NUMBER OF ITERATIONS TO MAKE ON THIS TEST

IF NON-BLANK WILL BE THE ADDRESS TO

WHICH THE NEXT SCOPE STATEMENT WILL

LOOP BACK TO.

IF NON-BLANK WILL BE THE FIRST

INSTRUCTION OF THE TEST

IF BLANK SCOPE WILL BE THE

FIRST INSTRUCTION
                                                        RETURN --
                                                        COMAND --
                                                                      FIRST INSTRUCTION
                                                             .MACRO NEWTST ASCII, ICOUNT, RETURN, COMAND
                                                             SSTN=1
                                                             SNWTST=0
                                                             .NLIST MC
                                                             .IF B <COMAND>
                                                             $$NEWTEST
                                                                                 \$TN, <ASCII>, SCOPE
                                                             .IFF
                                                             SSNEWTEST
                                                                                 \$TN, <ASCII>, <COMAND>
                                                             .ENDC
                                                             .NLIST
                                                             .LIST
                                                             .LIST
                                                             .1F NE 4000&$SWR
                                                             . IF NB ICOUNT
                                                             .IF LE <ICOUNT-1>
                                                            MOV
                                                                       #1,STIMES
                                                                                           ::DO 1 ITERATION
                                                             .IFF
                                                                       #ICOUNT, STIMES ;; DO ICOUNT ITERATIONS
                                                             MOV
                                                             .ENDC
                                                             .ENDC
                                                            .IF NB RETURN MOV #RETUR
                                                                       #RETURN, $LPADR ;; SET SCOPE LOOP ADDRESS
                                                             .ENDC
                                                             .ENDC
                                                             .NLIST
                                                              LIST
                                                                       MC
                                                             .NLIST
                                                                       NEWTST
                                                             . ENDM
```

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 119
MACRO SSNEWTEST
                                                  .SBTTL MACRO $$NEWTEST
.MACRO $$NEWTEST A.
.IRP ASCI.<ASC>
.IF EQ $NWTST
$NWTST=1
   5068
5069
5070
5071
5072
5073
5074
                                                                            A.ASC.COMND
                                                   .SBTTL T'A'
                                                                    ASCI
                                                   .NLIST
                                                   .LIST
                                                   .LIST
                                          :*TEST A
                                                           ASCI
                                                   ASCI
                                                   .ENDC
                                                   . ENDM
                                          TST'A: COMND
                                                   .NLIST ME
                                                   STN=STN+1
                                                   .ENDM SSNEWTEST
                                                   .SBTTL MACRO SUBTST
                                            ******* SUBTST ******************
                                           THIS MACRO WILL FORMAT A SUBTEST HEADING WITH STARS
                                           :A .SBTTL WILL BE FORCED & .NLISTED FOR THE TABLE OF CONTENTS.
                                           ARGUMENT:
                                                           THIS IS THE MESSAGE THAT WILL APPEAR IN THE TABLE OF CONTENTS & LISTING.
                                                           SUBTST <<THIS IS A FUN SUBTST>>
                                           :EXAMPLE:
                                           ***************
                                                   .MACRO SUBTST ASCII
                                                   .NLIST MC
                                                   $SUBTST <ASCII>
                                                   .LIST
                                                           MC
                                                   . ENDM
                                                           SUBTST
                                                   .SBTTL MACRO $SUBTST
.MACRO $SUBTST ASC
                                                   .IRP
.SBTTL
                                                           ASCI, <ASC>
                                                   .NLIST
                                                   .LIST
                                                   .LIST
                                           *********
                                           : *SUBTEST
                                                            ASCI
                                                   .ENDM
                                                   .NLIST ME
   5120
                                                   .ENDM $SUBTST
```

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 121
MACRO TYPOCT
                                               .SBTTL MACRO TYPOCT
                                            TYPOCT IS USED TO CHANGE A BINARY NUMBER TO A 6 DIGIT OCTAL NUMBER AND TYPE IT
                                            : ARGUMENTS:
                                                             THE NUMBER TO BE TYPED
                                            :1)
                                                    NUM
                                            :2)
                                                    REMARK ALLOWS A COMMENT TO BE MADE
                                            :ROUTINES REQUIRED
                                                    CONVERT BINARY TO OCTAL AND TYPE (.$TYPOCT)
                                            :2)
                                                    TYPE AN ASCIZ STRING (.$TYPE)
                                            :EXAMPLES:
                                                    TYPOCT HILMT, <TYPES THE CONTENTS OF HILMT>
TYPOCT #5, <TYPES ' 000005'>
                                                     .MACRO TYPOCT NUM, REMARK
   5149
5150
5151
5152
5153
5154
5156
5156
                                                     .NLIST
                                                     .DSABL CRF
                                                     .IIF DF LST$$ .LIST ME
                                                     .ENABL CRF
                                                     LIST.
                                                             NUM,-(SP) ;; SAVE NUM FOR TYPEOUT
                                                     MOV
                                                     .IIF NB <REMARK>,
                                                                                                          :: REMARK
                                                                               :: GO TYPE--OCTAL ASCII(ALL DIGITS)
                                                     TYPOC
                                                     .DSABL CRF
                                                     .IIF DF LST$$ .NLIST ME
                                                            TYPOCT
                                                     . ENDM
```

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 123
MACRO TYPOCS
   5163
5164
5165
                                                      .SBTTL MACRO TYPOCS
                                                *************** TYPOCS ***********
                                             TYPOCS IS USED TO CHANGE A BINARY NUMBER TO AN OCTAL NUMBER AND TYPE 1 TO 6 DIGITS
                                                      WITH OR WITHOUT LEADING ZEROS.
                                             : ARGUMENTS:
                                                      NUM
                                                               NUMBER TO BE TYPED
                                             :1)
                                                      REMARK ALLOWS A COMMENT TO BE MADE
                                             :2)
                                             :3)
                                                               NUMBER OF DIGITS (1 TO 6) TO BE TYPED
                                                               BLANK=SUPPRESS LEADING ZEROS (TYPES SPACES)
NON-BLANK=TYPE LEADING ZEROS
                                             ROUTINES REQUIRED
                                                      CONVERT BINARY TO OCTAL AND TYPE (.$TYPOCT)
                                              :1)
                                                      TYPE AN ASCIZ STRING (.STYPE)
                                             :EXAMPLES:
                                                      TYPOCS #12345, <TYPES ''5'>,1
TYPOCS #004, <TYPES ''04'>,2,X
TYPOCS #004, <TYPES ''4'>,2
                                             :1)
:2)
:3)
    5188
   5189
5190
5191
                                             ***************
    5192
                                                      .MACRO TYPOCS NUM, REMARK, N, Z
                                                      .NLIST
                                                      .DSABL CRF
                                                      .IIF DF LST$$ .LIST ME
                                                       .ENABL CRF
                                                       LIST
                                                                                 :: SAVE NUM FOR TYPEOUT
                                                               NUM,-(SP)
                                                      MOV
                                                       .IIF NB <REMARK>,
                                                                                                            :: REMARK
                                                                                 ::GO TYPE--OCTAL ASCII
                                                      TYPOS
                                                       . IF NB N
                                                       .BYTE N
                                                                                 ::TYPE N DIGIT(S)
                                                       .IFF
                                                       BYTE
                                                                                 :: TYPE 6 DIGITS
                                                       .ENDC
                                                       . IF NB Z
                                                       BYTE
                                                                                 :: TYPE LEADING ZEROS
                                                       . IFF
                                                       BYTE
                                                                                 ::SUPPRESS LEADING ZEROS
                                                       .ENDC
                                                       .DSABL
                                                      .IIF DF LST$$ .NLIST ME
.ENABL CRF
.ENDM TYPOCS
```

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 125
MACRO TYPDEC
                                         .SBTTL MACRO TYPDEC
                                        TYPDEC IS USE TO CHANGE A BINARY NUMBER TO A SIGNED
                                                DECIMAL NUMBER AND TYPE IT REPLACING LEADING ZERO
                                        NOTE: IF THE NUMBER IS NEGATIVE A
                                                MINUS SIGN WILL BE TYPED.
                                        : ARGUMENTS:
                                                        NUMBER TO BE TYPED
                                        :1)
                                                MUM
                                         :2)
                                                REMARK ALLOWS A COMMENT TO BE MADE
                                         ROUTINES REQUIRED
                                                CONVERT BINARY TO DECIMAL AND TYPE (.$TYPDEC)
                                         :1)
                                         (2)
                                                TYPE AN ASCIZ STRING (.$TYPE)
                                         :EXAMPLES
                                                TYPDEC SIZE. <TYPE THE CONTENTS OF SIZE>
TYPDEC #-10., <TYPE A MINUS TEN>
                                        :1)
                                         *************
                                                 .MACRO TYPDEC NUM, REMARK
                                                 .NLIST
                                                .DSABL CRF
                                                 .IIF DF LST$$ .LIST ME .ENABL CRF
                                                 .LIST
                                                MOV NUM,-(SP)
.IIF NB <REMARK>,
                                                                        :: SAVE NUM FOR TYPEOUT
                                                                                                 :: REMARK
                                                                         :: GO TYPE--DECIMAL ASCII WITH SIGN
                                                 TYPDS
                                                 .DSABL CRF
                                                .IIF DF LST$$ .NLIST ME
.ENABL CRF
.ENDM TYPDEC
```

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 126
```

```
MACRO BMOV
                    :1)
                    :2)
                    3)
                    *************
```

```
.SBTTL MACRO BMOV
      *********** BMOV **************
  THIS MACRO MOVES A BLOCK OF DATA.
: ARGUEMENTS:
                                  THE FIRST ADDRESS OF THE SOURCE BLOCK.
           FROMHERE
                                   THE FIRST ADDRESS OF THE DESTINATION BLOCK.
           TOHERE
                                  IF BLANK THE 1ST ADDRESS OF THE USER INSTRUCTION PAR'S IS USED (FASTCITY).
                                   THE SIZE OF THE SOURCE BLOCK.
           SIZE
                                  IF BLANK A 16 WORD TRANSFER IS ASSUMED.
'WHY DEFAULT TO 16 WORDS?'' YOU ASK!
'BECAUSE THAT'S HOW MANY WORDS TO THE USER PAR REGISTERS & THAT'S WHERE I INTEND TO MOVE LOTS OF STUFF.'' I REPLY!
```

.MACRO BMOV FROMHERE, TOHERE, SIZE .IF B TOHERE .NLIST CRF .DSABL .IIF DF LST\$\$.LIST ME CRF . ENABL .LIST JSR R5, BLOCK1 FROMHERE .DSABL .IIF DF LST\$\$.NLIST ME .ENABL CRF .MEXIT .ENDC .IF B SIZE .DSABL CRF .IIF DF LST\$\$.LIST ME .ENABL CRF .LIST JSR R5.BLOCK2 TOHERE FROMHERE .DSABL CRF .IIF DF LST\$\$.NLIST ME .ENABL CRF .ENABL .MEXIT . IFF .NLIST .DSABL .IIF DF LST\$\$.LIST ME .ENABL CRF LIST JSR R5, BLOCK3 SIZE

SEQ 0129

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 128 MACRO MAP
```

.SBTTL MACRO MAP

THIS MACRO MAPS A MEMORY BANK (16K) INTO THE TEST PATTERN AREA (SUPERVISOR VIRTUAL (60000-157777)).

ARGUEMENTS:

1) BANK THE BANK OF 16K WORDS TO BE MAPPED. THERE ARE 120 BANKS OF 16K WORDS

EXAMPLES

MAP LOC

;LOCATION 'LOC' CONTAINS THE # OF THE BANK TO MAP

MAP #28.

;BANK 34 (OCTAL) WILL BE MAPPED

.MACRO MAP PUSH R3 BANK .NLIST .DSABL CRF .IIF DF LST\$\$.LIST ME .ENABL CRF .LIST .IF B BANK #120.,R3 MOV .IFF MOV BANK, R3 .ENDC MAPPER CALL .DSABL CRF .IIF DF LST\$\$.NLIST ME .ENABL CRF POP R3 .ENDM MAP

-

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 130
MACRO SUPERVISOR
                                           .SBTTL MACRO SUPERVISOR
                                       ********** SUPERVISOR **********
                                    : THIS MACRO SWITCHES TO SUPERVISOR MODE.
                                    :ARGUEMENTS: NONE.
                                    **************
                                            .MACRO SUPERVISOR
                                            .NLIST
                                           .DSABL CRF
.IIF DF LST$$ .LIST ME
.ENABL CRF
                                            LIST
                                                                               :GO TO SUPERVISOR MODE
                                                   #BIT14,PSW
                                            .DSABL CRF
                                            .IIF DF LST$$ .NLIST ME
.ENABL CRF
                                            .ENDM SUPERVISOR
                                     .SBTTL MACRO USER
                                     THIS MACRO SWITCHES TO USER MODE.
                                     :ARGUEMENTS: NONE.
                                     *************
                                            .MACRO USER
                                            .NLIST
                                            .DSABL CRF
.IIF DF LST$$ .LIST ME
.ENABL CRF
.LIST
                                                                               GO TO USER MODE
                                                   #BIT15!BIT14,PSW
                                            BIS
                                            .DSABL CRF
.IIF DF LST$$ .NLIST ME
.ENABL CRF
                                            .ENDM USER
```

	C 11
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1	113 26-APR-82 09:41 PAGE 131
5409 5410	.SBTTL MACRO TESTAREA
5411 5412 5413	THIS MACRO SWITCHES TO THE SPECIFIED TEST MODE.
5414 5415	ARGUEMENTS: NONE.
5416 5417 5418 5419 5420 5421	.MACRO TESTAREA .NLIST .DSABL CRF .IIF DF LST\$\$.LIST ME
5422 5423 5424 5425 5426 5427 5428	.ENABL CRF .LIST BIS TESTMODE,PSW ;GO TO SYSTEM TEST MODE .DSABL CRF .IIF DF LST\$\$.NLIST ME .ENABL CRF .ENDM TESTAREA

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 133 MACRO SET4 & RES4
```

```
.SBTTL MACRO SET4 & RES4
  THESE MACROS SET & RESTORE VECTOR 4(TIMEOUT TRAP)
  IN IT'S RESTORED MODE TRAPS ARE REPORTED AS SUCH.
                               ; THE LOCATION TO VECTOR TO (ONLY USED IN "SET4" NOT "RES4")
: ARGUEMENTS:
                    LOC
I USE THE SET4 AND RES4 MACROS AROUND CODE THAT I EXPECT TO TRAP TO 4 LIKE LOOKING FOR ALL POSSIBLE CSR'S AND ETC. WHENEVER CODE IS NOT SURROUNDED BY SET4 AND RES4 THEN ANY TRAPS TO 4 WILL CAUSE AN ERROR PRINTOUT THAT SAYS 'UNEXPECTED TRAP TO 4" AND ALL THE ASSOCIATED REGISTER JUNK
           .MACRO SET4
                               ARG
           .NLIST
                     CRF
           .DSABL
          .IIF DF LST$$ .LIST ME .ENABL CRF
           .LIST
           MOV
                     ARG.4
          .DSABL
                     CRF
          .IIF DF LST$$ .NLIST ME
.ENABL CRF
.ENDM SET4
           .MACRO
                     RES4
           .NLIST
           .DSABL
                     CRF
           .IIF DF LST$$ .LIST ME
           .ENABL
                     CRF
           .LIST
           MOV
                     #TIMEOUT,4
                     #1.PROTYP
           CMP
                                          ; IS THIS AN 11/44?
           BNE
                                           BRANCH IF NOT
                     CPUERR
                                          CLEAR OUT THE CPU ERROR REGISTER BITS
           CLR
1015:
                                          THAT A EXPECTED TRAP COULD HAVE SET
           .DSABL CRF
           .IIF DF LST$$ .NLIST ME
           .ENABL CRF
                     RES4
           . ENDM
```

```
E 11
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 135
MACRO DLEFT
                                      .SBTTL MACRO DLEFT
                                       THIS MACRO DOES A DOUBLE WORD LEFT SHIFT
                                                            :THE LOCATION TO BE SHIFTED LEFT (CARRY TO LOC+2)
                                                     LOC
                                      :ARGUEMENTS:
                                      ***********
                                             .MACRO DLEFT ARG
                                             .DSABL CRF
.IIF DF LST$$ .LIST ME
.ENABL CRF
                                              LIST.
                                             ROL
                                                     ARG
                                                     ARG+2
                                             ROL
                                             .DSABL CRF
.IIF DF LST$$ .NLIST ME
.ENABL CRF
.ENDM DLEFT
.NLIST MD ;
                                                                  DON'T NEED TO SEE THEM ANY MORE
```

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 137 TRAP CATCHER
```

```
.SBTTL TRAP CATCHER
                000000
                                                           .=0
                                                                     177
                000000
000177
                                                           . WORD
     000000
                           000000
                                                                                          :. WORD .+2, HALT
                                                           SBTTL ACT11 HOOKS
                                                 ** THE HOOKS REQUIRED BY ACT11 ARE DEFINED AND SETUP BELOW:
                                                           DEFINITIONS:
                                                                                                     "END-OF-PASS" HOOK
                                                                                1)LOC.46
                                                                                          =ADDRESS OF END OF PASS ROUTINE
                                                                                            MODIFIED BY ACTIL.
                                                                                2)LOC.52
                                                                                                     PROGRAM NEEDS HOOK
                                                                                          BIT 15=1 PROGRAM SHOULD BE POWER
                                                                                          FAILED WHILE RUNNING

=0 NO POWER FAIL

BIT 14=1 PROGRAM MEMORY SIZE DEPENDENT
                                                                                          =0 NOT MEMORY SIZE DEPENDENT
BIT 13=1 PROGRAM REQUIRES MANUAL INTERVENTION
=0 MANUAL INTERVENTION NOT REQUIRED
                                                                                           BITS 12-0 MUST BE ZERO'S
                000046
015232
000052
                                                            =46
5524 000046
5525
                                                                                          ::1) SET LOC.46 TO ADDRESS OF SENDAD IN .SEOP
                                                           SENDAD
                                                           .=52
                                                                                           ::2) SET LOC.52 TO INDICATE MEMORY SIZE DEPENDANT
                                                           . WORD
     000052
                 000020
                                                                      BIT4
                                                           .SBTTL
=24
200
                                                                     APT11 HOOKS
                000024
000200
000042
                                                                      :: SET POWER FAIL TO POINT TO START OF PROGRAM
                                                                      :: FOR APT START UP
     000024
                                                            =42
                                                           STACK ;SO RT11 CAN START WITH RUN COMMAND ;POINT TO APT INDIRECT ADDRESS PNTR. SAPTHDR ;;POINT TO APT HEADER BLOCK
      000042
      000044
                                                           .=200
                                                                                           "NORMAL" START (SAVE ERROR ACCOUNTING)
      000200
                                                 START3: BR
                                                                      START1
                                                           BR
                                                                      START2
                                                            =300
                 000300
5538 000300
5539 000304
5540 000310
5541 000316
5542
                                                                      RESTART
START
                                                 START1: CLR
                           002612
                                                            JMP
                                                                      RESTART
                                                 START2: SET
                                                           JMP
                                                                      START
                           003654
                 000137
                 002000
                                                           .=STACK
```

AVITABLES	INTITACTED TO SENO	
5545 5546		.SBTTL VARIABLES INITIALIZED TO ZERO :*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS :*USED IN THE PROGRAM.
5547 5548 002000 5549 002000 5550 002002 5551 002004 5552 002006 5553 5554 002010	000000 000000 000000 000000	**USED IN THE PROGRAM. \$CMTAG: SELONLY:0 DIAGFLAG:0 KAMIKAZE:0 SKIPKAMI:0 :*START OF COMMON TAGS :*SELECT ONLY BANKS MARKED BY FIELD SERVICE MODE FLAG SET FOR SHIFTING DIAGONAL TEST SET FOR KAMIKAZE MODE TESTING :USED TO SKIP RESTORING KAMIKAZE MODE WHEN MODIFIED
5553 5554 002010 5555 002011 5556 002012 5557 002013 5558 002014 5559 002016 5560 002020 5561 002022 5562 002024 5563 002026 5564 002030 5565 002032 5566 002034 5567 002036	000 000 000 000 00000 000000	SPATMAR:.BYTE 0 ;PATTERN NUMBER SBANK: .BYTE 0 ;BANK & SIGN SERFLG: .BYTE 0 ;;CONTAINS ERROR FLAG SITEMB: .BYTE 0 ;;CONTAINS ITEM CONTROL BYTE LASTERROR:.WORD 0 ;NUMBER OF ERRORS ON LAST PASS ERRPC: .WORD 0 ;CONTAINS PC OF ERROR FOR TYPEOUT BADPC: .WORD 0 ;CONTAINS PC OF ERROR
5561 002022 5562 002024 5563 002026 5564 002030 5565 002032 5566 002034 5567 002036	000000 000000 000000 000000 000000 00000	ERRSP: .WORD 0 ;CONTAINS SP OF ERROR FOR TYPEOUT BADSP: .WORD 0 ;CONTAINS SP OF ERROR ERRPSW: .WORD 0 ;CONTAINS PSW OF ERROR FOR TYPEOUT BADPSW: .WORD 0 ;CONTAINS PSW OF ERROR ADDRESS: .WORD 0 ;CONTAINS ADDRESS OF 'BAD' DATA PADDRESS: .WORD 0 ;ADDRESS OF PARITY ERROR PHYADD: .WORD 0,0 ;22 BIT PHYSICAL ADDRESS
5567 002036 5568 002042 5569 002044 5570 002046 5571 002050 5572 002052 5573 002054 5574 002056 5575 002060 5576 002062 5577 002064 5578 002066 5579 002070	000000 000000 000000 000000 000000 00000	**USED IN THE PROGRAM. **USED IN THE PROGRAM. **USED IN THE PROGRAM. **CMTAG: SELONLY:0 DIAGFLAG:0 SELONLY:0 SELFOR SHIFTING DIAGONAL TEST SELONLY:0 SELFOR SHIFTING DIAGONAL TEST SELFOR SHIFTING DIAGONAL TEST SELFOR SHIFTING DIAGONAL TEST SELFOR SHIFTING DIAGONAL TEST SET FOR KAMIKAZE MODE TESTING SEANK: BYTE SEANK: BYTE SEANK: BYTE SEARMS: BYTE SITEMB: BYTE SEARMS: BYTE SITEMB: BYTE SITMB: BYTE SITMB
5576 002062 5577 002064 5578 002066 5579 002070 5580 002072 5581 002074 5582 002076 5583 002100	000000 000000 000000 000000 000000 00000	SKPERR: WORD 0 SKPERR: WORD 0 SKPERR: WORD 0 SKIP ERROR MESSAGE IN 'SERRGEN' NON-EXISTANT MEMORY COUNTER (HOLES) PARCNT: 0 PATERR: 0 PATERR: 0 NOPAR: 0 NOPAR: 0 NONEM: 0 BANK: 0 BANKINDEX: 0 SKIP ERROR MESSAGE IN 'SERRGEN' PATITY ERROR COUNTER PATITY ERROR COUNTER NO PARITY ERROR MODE INDICATOR NO NON-EXISTANT MEMORY (HOLES) MODE INDICATOR MEMORY BANK UNDER TEST SUSED TO INDEX INTO CONFIG TABLE
5585 002104 5586 002106 5587 002110 5588 002112 5589 002114 5590 002116 5591 002120	000000 000000 000000 000000 000000 00000	CPUBIT: 0 ; CONTAINS 1 BIT TO IDENTIFY CPU TO CONFIGURATION TABLE MUT: 0 ; MEMORY UNDER TEST FLAG PATTERN:0 ; PATTERN NUMBER UNDER TEST KPFLAG: .WORD 0 ; BANK IS PROTECTED REGION OF ECC ACFLAG: .WORD 0 ; BANK CAN BE ACCESSED BY THIS CPU MKFLAG: .WORD 0 ; IF SET INDICATES MS11-M OR MF11S-K UNDER TEST PFLAG: .WORD 0 ; BANK IS IN PROGRAM SPACE PRANK IS IN PROGRAM SPACE
5579 002070 5580 002072 5581 002074 5582 002076 5583 002100 5584 002102 5586 002106 5586 002106 5587 002110 5588 002112 5589 002112 5590 002122 5591 002120 5592 002122 5593 002122 5594 002126 5596 002130 5596 002130 5598 002136 5599 002140 5600 002142	000000 000000 000000 000000 000000 00000	PATERR: 0 PATERR: 0 NOPAR: 0 N
5601 002144	000000	CTLKVEC: . WORD 0 :HOLDS OLD KERNAL STACK POINTER IN CASE OF CNTL/K

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 139-1 VARIABLES INITIALIZED TO ZERO
                                                                                                                      CSR: .WORD 0
CSRNO: 0
SAVCSR: .WORD 0
OLDCSR: .WORD 0
                                                                                                                       CSRNO: 0
                                                                                                                                                                                                                         :DATA TO OR FROM CSR
                                              000000
000000
000000
        5602 002146

5603 002150

5604 002152

5605 002154

5606

5607 002156

5608 002160

5609 002162

5610 002164

5611 002166

5612 002170

5613 002172

5614 002174

5615

5616 002176

5617 002200

5618 002202

5619 002204

5620 002206

5621 002210

5623 002214

5624 002216

5625 002220

5626 002222
                                                                                                                                                CSR ADDRESS NUMBER (4 LSB'S)

WORD 0

UDCATION TO SAVE CSRNO DURING FS COMMAND

OUT OF THESE LOCATIONS STORE GPR'S DURING SUPERVISOR TESTS
                                                                                                                        SUPDR1: 0
SUPDR2: 0
SUPDR3: 0
                                               000000
                                                000000
                                                000000
                                                                                                                        SUPDR4: 0
                                                000000
                                               000000
000000
000000
                                                                                                                        SUPDR5: 0
                                                                                                                        SUPDR6: 0
                                                                                                                                                                                                                     DUMMY LOCATION FOR ADDRESS PASSING
                                                                                                                        DUMMY: 0
                                                                                                                        THESE LOCATIONS STORE GPR'S & PSW DURING DETAILED ERROR PRINTOUTS
                                               000000
                                                                                                                        DETRO:
                                                                                                                        DETR1:
                                                                                                                        DETR2:
DETR3:
                                                000000
                                                000000
                                                 000000
                                                                                                                        DETR4:
                                                                                                                         DETRS:
                                                 000000
                                               000000
000000
000000
000000
000000
                                                                                                                         DETSP:
                                                                                                                        DETPSW: 0
                                                                                                                                                                                                                        DETAILED REPORT FLAG
CSR'S HAVE BEEN TESTED FLAG
BIT PER EXISTING CSR, EG-
CSR O REPRESENTED BY BIT 15, ETC.
                                                                                                                        DETFLAG: 0
                                                                                                                         CONTFLAG: 0
                                                                                                                         TOTCSRS: . WORD
                                                000000
000000
000000
000000
         5628 002224

5629 002226

5630 002230

5631 002232

5632 002234

5633 002236

5634 002240

5635 002244

5636 002250

5637 002254

5638 002260

5639 002262

5640 002263

5641 002264

5642 002266

5643 002270

5644 002272

5645 002274

5646 002276

5647 002300

5648 002302

5650 002304

5651 002310

5652 002312

5653 002320

5656 002322

5657 002324

5658 002326
                                                                                                                                                                                                                         FIRST ADDRESS UNDER CONTROL OF THIS CSR
                                                                                                                         CSRFIRST:.WORD
                                                                                                                        CSRLAST: .WORD
CSRFBANK: .WORD
CSRLBANK: .WORD
                                                                                                                       CSRFBANK:.WORD O
CSRLBANK:.WORD O
CSRINT: .WORD O
SPLTCSR: .WORD O
DATBUF: .WORD O.O
TSTDAT: .WORD O.O
SBEMSK: .WORD O.O
SBEMSK: .WORD O.O
SUPDOADD:.WORD O
PASFLG: .BYTE O
UPPFLG: .BYTE O
PASSNO: .WORD O
SAVPAR: .WORD O
SAVPAR: .WORD O
SAVPAR: .WORD O
REALPAT:.WORD O
PARTHERE:.WORD O
PARTHERE:.WORD O
SOURCE: .WORD O
CSTACK:.WORD O
CSTACK:.WORD O
CSTACK:.WORD O
CSTACK:.WORD O
CSRLOOP:.WORD O
CSRLOOP:.WORD O
CSRLOOP:.WORD O
CSRLOOP:.WORD O
                                                                                                                                                                          Ŏ
                                                                                                                                                                                                 TWO WORD DATA BUFFER
TWO WORD TEST DATA
TWO WORD SINGLE BIT ERROR MASK
TWO WORD DOUBLE BIT ERROR MASK
ADDRESS OF SUBROUTINE TO EXECUTE
                                                 000000
                                               000000
                                                                        000000
                                                                        000000
                                                                                                                                                                                                                         : INU WORD DOUBLE BIT ERROR MASK
:ADDRESS OF SUBROUTINE TO EXECUTE IN SUPERVISOR MODE
:LOCAL LOOP PASS CONTROL
:LOCAL LOOP PASS CONTROL
:LOCAL LOOP PASS CONTROL
:USED TO SAVE KERNAL PAR 5
:XXDP MONITOR RETURN ADDRESS
:RETURN TO MONITOR FLAG
                                                                                                                                                                                                                         RETURN TO MONITOR FLAG
REAL PATTERN UNDER TEST
BACKED UP VALUE OF CACHE CONTROL REGISTER
PARITY TRAPS SOMETIMES GO TO ADDRESS STORED HERE
STACK SAVED HERE IF IN FIELD SERVICE MODE
USED FOR RELOCATION TO A NEW BANK
SOURCE OF DATA WORDS FOR CHECKBIT GENERATION SUBROUTINE
CHECKBITS TO BE LOADED INTO CSR
                                                  000000
                                                  000C00
                                                 CHECK BITS TO BE WRITTEN
BIT MASK FOR CSR
CSR ALL 1'S PATTERN
BIT POINTER
                                                                                                                                                                                                                         VALUE TO BUMP THE PC BY TO RECOVER AFTER A PARITY TRAP VALUE TO INCREMENT ADDRESS BY TO REMAIN IN THE SAME CSR LOOP CONTROL FOR CSR TESTING
```

		M/P MEMOR	RY DIAG.	MACRO P	11113 26	-APR-82	09:41	PAGE	139-2		324 0130
5659 5660 5661 5662 5663 5664 5665 5666 5667 5668 5669	002330 002332 002334 002336 002340 002342 002344 002350 002350	000000 000000 000000 000000 000000 00000			SFILLS:	BYTE	WO BYTES	s Mus1	STAY	FLAG SET BY SUCCESSFULL TASK OR SUBROUTINE FOR AID IN 'MOV' INSTRUCTIONS SECONDS THAT BATTERIES SHOULD LAST FLAG TO SKIP MKCONTROL SUBROUTINE SET WHEN RUNNING NULL PATTERNS FLAGS QUICK VERIFY PASS UNDER APT, ACT, OR X FLAGS ACT AUTOMATIC MODE PROGRAMMING RULES FLAGS APT AUTOMATIC MODE PROGRAMMING RULES FLAGS XXDP CHAIN MODE PROGRAMMING RULES TOGETHER CONTAINS NULL CHARACTER FOR FILLS CONTAINS NOF FILL CHARACTERS	XDP CHAIN MODE
5674 5675 5676 5677 5678 5680 5681 5682 5683 5684 5685 5686 5686 5687 5688	002360 002362 002364 002366 002370 002374 002376 002400 002402 002404 002416 002416 002420 002420 002420	000 000 000 000000 000000 000000 000000	000000		SFILLS: STPFLG: SESCAPE EVEN: STRIPES COUNT: NOTAB: BSIZE: KSIZE: LSIZE: MSIZE: PSIZE: TOOMANY READONL TESTADD UNITOP: STOPOK: APTPAR: APTECC: NOFSMOD NOERROR	BYTE BYTE EVEN :0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	000			CONTAINS # OF FILL CHARACTERS: 'TERMINAL NOT AVAILABLE' FLAG	FOR A BANK US MAP
5706 5706 5706 5707 5708 5708 5708	002430 002432 002434 002436 002446 002446 002446 002450 002450 002456 002456 002476 002476 002512 002516 002526 002526	000000 000000 000000 000000 000000 00000	000000 000000 000000 000000 000000			0.0.0.	0.0.0.0	.0		;USED TO STORE INFORMATION ABOUT THE 16 ;POSSIBLE CSR'S :USED TO HOLD LINKS TO PATTERNS WHICH ;CAN EXECUTE IN THE PAR/PDR'S OR NOT ;USED TO STORE CSR VALUES FOR CSR TESTS ;USED TO FLAG MF11S-K MEMORY TO TESTS ;POINTS TO PROGRAM CSR ;FLAGS INHIBIT ECC TESTS ON RELOCATION	
5711	002532	000000			INHECC:	.WORD	0			FLAGS INHIBIT ECC TESTS ON RELOCATION	

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 139-3 VARIABLES INITIALIZED TO ZERO

000000

INHBANK:.WORD 0
FULLREL:.WORD 0
\$CMTGE: ; *END OF COMMON TAGS

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 141 VARIABLES INITIALIZED TO NON ZERO

5717 5718 002540 5719 002546 5720 002546 5721 002550 5722 002552 5723 002554 5724 002556 5725 002560 5726 002562 5727 002564 5728 002566 5729 002570 5730 002572 5731 002574 5732 002576 5733 002600 5734 002602 5735 002604 5736 002606 5737 002610 5738 002612 5739 002614	002540 002544 002546 002550 002552 002554 002560 002560 002562 002564 002566 002570 002572 002574 002576 002604 002604 002604 002610 002612 002614	000001 001415 040000 000012 000167 170000 000031 002000 009001 177777 176543 123456 176543 123456 177777 177777 177777 000003 052525 000000 000000 000000	000000		CACHKN: CACHKF: TESTMODE ERRMAX: LASTBANK LASTBLOC SOBK: KSTACK: LOADHOME WORST: SEEDLO: MSEEDL: HEADER: ONES: FLIPLOC: SOFTPAT: \$LPADR: \$LPERR: RESTART: \$ERTTL:	.SBTTL 1.0 1415 :40000 10. :167 K:170000 25. STACK :1 177777 176543 123456 176543 123456 177777 177777 177777 177777 177777 .WORD .WORD	VARIABLES 0 0	INITIALIZED TO NON ZERO CACHE CONSTANT (MOVED TO CONTRL TO TURN ON CACHE) CACHE CONSTANT (MOVED TO CONTRL TO TURN OFF CACHE) USED TO SELECT THE PROPER TEST MODE FOR A PATTERN RUN MAX N OF ERRORS PER BANK WITH SW11 HIGHEST BANK OF MEMORY HIGHEST BANK OF MEMORY+1 (IN PAR FORMAT) SOB CONSTANT STACK BEGINNING HOME BANK OF LOADERS SET IF TESTING BANKS IN WORST FIRST MODE(1ST PASS) WORKING SEED HI (USED FOR RANDOM NUMBER GENERATOR) WORKING SEED HO (USED FOR RANDOM NUMBER GENERATOR) MASTER SEED HO (USED FOR RANDOM NUMBER GENERATOR) MASTER SEED LO (USED FOR RANDOM NUMBER GENERATOR) COUNTER FOR FLIPING DATA ON WORST CASE NOISE TEST PATTERN FOR SOFT ERROR BACKGROUND TESTS CONTAINS SCOPE LOOP ADDRESS CONTAINS SCOPE LOOP ADDRESS CONTAINS SCOPE RETURN FOR ERRORS RESTART (START ADD 202) FLAG CONTAINS TOTAL ERRORS	•
5740 5741 5742 5743	002616 002620	000377 177400			BAKPAT: SWAPAT:	.WORD	377 177400	;BACKGROUND PATTERN * ;SWAPPED BAKPAT *	
5745 5746 5747 5748 5749 5750 5751 5752 5753	002622 002624 002626 002630 002632 002634 002636 002637	177570 177570 177560 177562 177564 177566 012 207 000 077	377	377	SWR: DISPLAY: \$TKS: \$TKB: \$TPS: \$TPB: \$FILLC: \$BELL:	.WORD 177560 177562 177564 177566 .BYTE .ASCIZ	DSWR DDISP 12 <207><377><377>	::ADDRESS OF SWITCH REGISTER ::ADDRESS OF DISPLAY REGISTER ::TTY KBD STATUS ::TTY KBD BUFFER ::TTY PRINTER STATUS REG. ADDRESS ::TTY PRINTER BUFFER REG. ADDRESS ::INSERT FILL CHARS. AFTER A 'LINE FEED'' ::CODE FOR BELL	
5754 5755	002642 002643 002644 002645	000 077 015 012	000				/?/ <15>	::QUESTION MARK ::CARRIAGE RETURN ::LINE FEED	

5760 5761 5762 5763 5764	SBTTL CONFIGURATION TABLE CONFIG:FIRST 16K CONFIGURATION WORDS (2 EACH) 200TH 16K CONFIGURATION WORDS (2 EACH) 200TH 16K CONFIGURATION WORDS (2 EACH)
5760 5762 5763 5764 5765 5766 5767 5768 5769 5770 5771 5772 5773 5774 5775 5776 5777 5778 5779 5780	CONFIGURATION WORDS: LOW: BIT 0 BIT 1 MEMORY SUCESSFULLY ACCESSED BIT 2-4 RESERVED BIT 5 SKIP ECC LOGIC TESTS FLAG (1=SKIP) BIT 6 PROTECTED REGION OF ECC MEMORY BIT 7 PROTECTED (PROGRAM SPACE) BIT 8-11 CSR CODE
5773 5774 5775 5776 5777 5778 5779 5780 5781	HIGH: BIT 0-7 NUMBER OF ERRORS BIT 8-10 MEMORY TYPE BIT 11 INTERLEAVED BOARD TYPE (0=128K, 1=64K) BIT 12 INTERLEAVE ENABLED BIT 13 'BACKGROUND PATTERN VALID' FLAG BIT 14 BANK SELECTED FOR TEST BY FIELD SERVICE MODE BIT 15 LOADERS HOME BANK
5782 002650 000201 5785 003654	CONFIG: REPT 201 CONFIEND:

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 144

5787 5788 003654		START: SUBTST	**************************************
5789 003654 105737 5790 003660 001001 5791 003662 000005 5792 003664 5793 003670 010637 5794 003674 013706 5795 003700 012700 5796 003704 005020	065660 002270 002560 002000	TSTB BNE RESET NORES: CLEAR MOV MOV	SENV NORES MONFLG ;; CLEAR RETURN TO MONITOR FLAG SP, SAVMON ;; SAVE XXDP MONITOR RESTART ADDRESS KSTACK, SP ;; SETUP THE STACK POINTER #\$CMTAG, RO ;; FIRST LOCATION TO BE CLEARED (RO) + ;; CLEAR MEMORY LOCATION
5797 003706 022700 5798 003712 001374 5799 003714 012737 5800 003722	002540 000167 002552	1\$: CLR CMP BNE MOV SUBTST ;***********************************	#\$CMTGE,RO ;:DONE? 1\$;LOOP BACK IF NO #167,LASTBANK ;RESTORE LASTBANK (THIS MUST BE DONE PRIOR TO SYSTEM SIZING) < <clear non-program="" space="">> CLEAR NON-PROGRAM SPACE</clear>
5801 5802 5803 5804 003722 012737 5805 003730 005000 5806 003732 000241 5807 003734 005520 5808 003736 020027 5809 003742 103773 5810 003744 005037	000001 002074 160000 002074	THIS A	ATTEMPS TO GET RID OF ANY PARITY ERRORS BY WRITING INTO LOCATION THAT IS NOT LOADED INTO BY THE PROGRAM OR ALLOCATED E XXDP LOADERS #1,NOPAR ;PARITY ACTION = COUNT & IGNORE R0 (R0)+ R0,#160000 2\$ NOPAR ;RESTORE DEFAULT PARITY ACTION

5813	003750				******* *SUBTE		<type of="" sizer<="" system="" th=""><th>?>></th></type>	?>>
5814 5815 5816	003750 003752 003754 003762	000401 000000			PROTYP: SYSSIZ:	BR .WORD SET4	**************************************	;SKIP OVER VARIABLE LOCATION
5818	003766	005737	177746			TST SET4	CONTRL #9\$:SEE IF CACHE REGISTER RESPONDS :YES - DO WE HAVE 11/44 TYPE CACHE
5820 5821	004000	005737 000411 012737	177750 000014	002544	9\$:	TST BR MOV	MAINT 5\$ #14,CACHKF	:SEE IF CACHE REGISTER RESPONDS :YES - DO WE HAVE 11/44 TYPE CACHE :OR 11/60 TYPE CACHE? :BRANCH IF 11/44 TYPE CACHE :TURN OFF CONSTANT FOR 11/60 CACHE
5822 5823	004010 004012	000405		002744	4\$:	BR CLR	5\$ CACHKN	NO CACHE ON SYSTEM
5824 5825	004016 004024	005037	002540	067330	5\$:	MOV SET4	#ZEROS,DT14 #6\$;DO NOT PRINT CONTRL ERROR MESSAGES
5821 5822 5823 5824 5825 5826 5827 5828 5829	004032 004036 004042	005737 005037 052737 032737	172516 172516 000020 000020	172516		TST CLR BIS	MMR3 MMR3 #BIT4,MMR3	;DO WE HAVE AN MMR3? ;YES WE DO ;SEE IF THERE IS 22-BIT MODE
5829 5830	004000 004002 004010 004012 004016 004024 004032 004036 004042 004050 004050	032737 001026 000413	000020	172516 172516		BIT	#BIT4,MMR3 10\$	BRANCH IF 22-BIT RELOCATION BRANCH IF AMR3 BUT NO 22-BIT RELOC.
5832 5833			140000	002546	;* 11/3	BR 4 TYPE M	7\$ ACHINES ENTER HERE	;BRANCH IF MMR3 BUT NO 22-BIT RELOC. ;MAKE TESTMODE USER
5834 5835	004062 004070 004074	012737 005237 005037	140000 002452 067200 067340	002340		INC	ACHINES ENTER HERE #140000, TESTMODE NOSUPER DT5+10	PARE TESTHODE USER
5836 5837 5838	004100	005037 005237	067340 002450			CLR INC 5 TYPE M	DT14+10 NO22BIT ACHINES ENTER HERE NO22BIT #7,LASTBANK	
5839 5840	004110 004114 004122 004126 004132 004134 004142	005237 012737	002450 000007 067202 067342	002552	78:	INC	NO22BIT #7,LASTBANK	;
5841 5842 5843	004122 004126	005237 012737 005037 005037 000417	067202 067342			CLR	DT14+12	
5844 5845	004134	000007			10\$:	BR SET4 MFPT	8\$ #8\$:TYPE OF PROCESSOR TEST: THIS INSTRUCTION
5847 5848 5849								: (AVAILABLE ON NEWER PROCESSORS ONLY) PLACES : A CODE IN THE LOWER BYTE OF RO THAT : INDICATES THE PROCESSOR TYPE 1=11/44
5850	004144 004150	110037 022737	003752 000003	003752		MOVB	RO, PROTYP	3=11/24 MOV THE CODE TO PROTYP IS THIS AN 11/24?
5853 5854 5855	004144 004150 004156 004160 004164	110037 022737 001005 005237 012737	002452 140000	002546		INC MOV	NOSUPER #140000, TESTMODE	BRANCH IF NOT - WE HAVE AN 11/44 NO SUPERVISOR MODE MAKE TEST MODE USER
5851 5852 5853 5854 5855 5856 5857 5858 5859	004172 004200 004204 004210 004216	005037 005737 012737	061322 177766		8\$:	SET4 CLR TST	#11\$ CPERRF a#177766	:TRAPS GO TO 11\$:R-C :CLEAR THE FLAG :R-C :IS THERE A CPU ERROR REGISTER? :R-C :YES-TRAPPED
5859 5860	004210 004216	012737	177777	061322	11\$:	MOV RES4	#-1,CPERRF	;YES-TRAPPED ;R-C

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 147 TYPE OF SYSTEM SIZER

5863 004240	SUBTST < <initialize non="" to="" variables="" zero="">> :***********************************</initialize>
5864 004240 5865 004246 012737 000003 002602 5866 004254 5867 004262 012737 176543 002572 5868 004270 012737 123456 002574 5869 004276 013737 002572 002566 5870 004304 013737 002574 002570 5871 004312 012737 000377 002616 5872 004320 012737 177400 002620	SET WORST MOV #3,FLIPLOC SET HEADER MOV #176543,MSEEDH MOV #123456,MSEEDL MOV #123456,MSEEDL MOV MSEEDH,SEEDHI ;PRIME THE RANDOM NUMBER GENERATOR MOV MSEEDL,SEEDLO ;BOTH HIGH AND LOW WORDS MOV #377,BAKPAT MOV #177400,SWAPAT SUBTST < <initialize vectors="">></initialize>
5874 004326 012737 060166 000020 5875 004334 012737 000340 000022 5876 004342 012737 060522 000030 5877 004350 012737 000340 000032 5878 004356 012737 065754 000034 5879 004364 012737 000340 000036 5880 004372 012737 054356 000024	**SUBTEST INITIALIZE VECTORS
5882 004406 012737 042406 000114 5883 004414 012737 000340 000116 5884 004422 012737 042602 000010 5885 004430 012737 000340 000012 5886 004436 012737 042556 000004	MOV #340,TRAPVEC+2;LEVEL 7 MOV #\$PWRDN,PWRVEC;;POWER FAILURE VECTOR MOV #340,PWRVEC+2 ;;LEVEL 7 MOV #PARITY,PARVEC;GÉT READY FOR PARITY ERRORS MOV #340,PARVEC+2 MOV #9DP1105,RESVEC;RESERVED INSTRUCTION TRAP MOV #340,RESVEC+2
5887 004444 012737 000340 000006 5888 004452 012737 042570 000250 5889 004460 012737 000340 000252 5890 004466 104423	MOV #340,ERRVEC+2 ;SET PRIORITY OF ERROR TRAPS MOV #MMTRAP,MMVEC ;VECTOR FOR MEMORY MANAGEMENT MOV #340,MMVEC+2 CACHON ;TURN CACHE ON

```
SUBTST <<INITIALIZE PATTERNS>>
5893 004470
                                                              ************************
                                                                                        INITIALIZE PATTERNS
                                                              *********
                                                                          THE APT E-TABLE DETERMINES WHICH PATTERNS ARE GOING TO BE RUN.

EACH BIT SET REPRESENTS A PATTERN TABLE ENTRY THAT IS TO BE LEFT

ALONE (TO BE RUN). EACH BIT CLEARED REPRESENTS A PATTERN TABLE ENTRY

THAT IS TO BE OVERLAYED WITH THE ADDRESS OF A NULL PATTERN.

MOV #$DDWO,RO
5894
5895
 5897
                     012700
012001
012703
012702
004737
012001
012702
004737
012001
012703
012702
004737
5898 004470

5899 004474

5900 004476

5901 004502

5902 004506

5903 004512

5904 004514

5905 004520

5906 004524

5907 004526

5908 004532

5909 004536

5910 004542

5911 004544

5912 004550

5913 004554

5914 004566

5916 004562

5916 004562

5917 004572

5918 004574

5919 004600

5920 004604
                                   065724
                                                                                         (R0)+,R1
                                                                           MOV
                                                                                        WMKCSRT, R3
                                   020204
                                                                           MOV
                                                                                        #16.,R2
                                                                           MOV
                                                                                        PATPLUG
                                   004606
                                                                           CALL
                                                                                        (R0)+,R1
#8.,R2
                                                                           MOV
                                   000010
                                                                           MOV
                                   004606
                                                                                         PATPLUG
                                                                           CALL
                                                                                        (R0)+,R1
#MKPAT,R3
#16.,R2
PATPLUG
                                                                           MOV
                                   020434
000020
004606
                                                                           MOV
                                                                           MOV
                                                                           CALL
                     004737
012001
012702
004737
012001
012703
012702
004737
012001
012702
004737
000417
                                                                                         (R0)+,R1
#8.,R2
                                                                           MOV
                                                                           MOV
                                   000010
                                                                                         PATPLUG
                                                                           CALL
                                   004606
                                                                                        (RO)+,R1
#MJPAT,R3
                                                                           MOV
                                   020620
                                                                           MOV
                                                                           MOV
                                                                                         #16.,R2
                                                                                         PATPLUG
                                   004606
                                                                           CALL
                                                                                         (R0)+,R1
#8.,R2
                                                                           MOV
                                   000010
                                                                           MOV
                                                                                         PATPLUG
                                   004606
                                                                           CALL
                                                                                         SUBAAA
                                                              PATPLUG:SUBTST <<SUBR PLUG IN NULL PATTERNS>>
5922 004606
                                                              **************
                                                                                         SUBR
                                                                                                   PLUG IN NULL PATTERNS
                                                              :*SUBTEST
                                                              ****************
                                                                           FOR I := #1 TO R2
 5923 004606
5923 004606
5924 004614
5925 004616
5926 004620
5927 004624
5928 004624
5929 004630
5930 004642
                                                                               ROR R1
                      006001
                                                                                                                  ; IF CARRY CLEAR
                                                                               ON. NOERROR
                                                                                  MOV #MT0999, (R3)
                      012713 026760
                                                                               END : OF ON. ERROR ADD #2,R3
                      062703 000002
                                                                           END OF FOR
                      000207
```

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 151
        PLUG IN NULL PATTERNS
SUBR
                                         SUBAAA: SUBTST <<CLEAR THE CONFIGURATION TABLE>>
   5933 004644
                                         *********
                                                         CLEAR THE CONFIGURATION TABLE
                                         :*SUBTEST
                                         **************
                                                 THIS ZEROS (UNLESS WE STARTED AT ADDRESS 202) THE CONFIG TABLE WHICH IS FULLY DISCRIBED AT LOCATION "CONFIG".
                                                  ENABLE LSB
                                                 IF RESTART IS FALSE
                                                         #CONFIG,RO
                012700
                                                   MOV
                        002650
                                                   CLR
                                                          (R0) +
                                         15:
                                                    CMP
                                                          #CONFIEND, RO
```

5937 004644 5938 004652 5939 004656 5940 004660 5941 004664 5942 004666 005020 022700 5940 5941 5942 5943 003654 BNE END OF IF RESTART MOV #BIT1.CPUBIT ;SET ID BIT SUBTST <<SIZE FOR A HARDWARE SWITCH REGISTER>> 012737 000002 002104 004666 *************** :: IF NOT FOUND OR IT IS
:: EQUAL TO A '-1', SETUP FOR A SOFTWARE SWITCH REGISTER.
ENABL LSB
SET4 #3\$ SIZE FOR A HARDWARE SWITCH REGISTER : *SUBTEST 5946 5947 5948 5949 004674 5950 004702 5951 004710 5952 004716 5953 004726 5954 004730 5955 004734 TRAPS TO 4 GOTO 3\$ #DSWR.SWR MOV 012737 012737 : AND A HARDWARE DISPLAY REGISTER #DDISP, DISPLAY MOV :IF NO TRAP FROM REFERENCE TO aSWR AND aSWR = #-1 IF #-1 EQ aSWR 2\$ #2\$,(SP) :: BRANCH IF NO TIMEOUT 000403 012716 000002 SET UP FOR TRAP RETURN 004736 3\$: MOV RTI 5956 004736 5957 004760 5958 004766 5959 004774 RESET TRAPS TO 4 TO DEFAULT MDISPREG, DISPLAY :: POINT TO SOFTWARE SWR 2\$: RES4 MOV 000176 012737 000174 MOV END : OF IF #-1

.DSABL LSB

5960

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 153 SIZE FOR A HARDWARE SWITCH REGISTER

```
SUBAAB: SUBTST <<SETUP ACT, APT, & XXDP>>
5963 004774
                                                                                          SETUP ACT, APT, & XXDP
                                                                :*SUBTEST
                                                                ********
5964
5965
5966 004774
5967 005000
5968 005010
5969 005016
5970 005016
5971 005026
5972 005034
5973 005034
5974 005044
5975 005074
                                                                             :THIS SETS UP A BUNCH OF FLAGS TO TELL THE PROGRAM EVERYTHING :IT CARES TO KNOW ABOUT APT, ACT, & XXDP.

CLR SPASS :CLEAR PASS COUNT
                                                                             CLR SPASS
IFB #BITS SET.IN SENVM
                     005037 065646
                                                                                                                     :INDICATE NO TERMINAL
                                                                                SET STPFLG
                                                                             END : OF IFB #BIT5
IFB #BIT7 SET IN SENVM
                                                                                SET APTSIZE
                                                                             END OF IFB #BIT7
IFB SENV EQ #1
SET APTFLAG, QV
                                                                                        APTFLAG, QVFLAG, $AUTO, QUICK #APTDOWN, PWRVEC
                                   047746
065662
                      012737
012737
                                                 000024
002622
5975 005074
                                                                                MOV
5976 005102
5977 005110
5978 005112
5979 005130
                                                                                                                      :USE APT SWR
                                                                                MOV
                                                                                         #$SWREG, SWR
                                                                                IF 42 NE #STACK AND 42 NE #0
                                                                                   SET OVFLAG, SAUTO
5980 005144
5981 005154
5982 005162
5983 005164
5984 005172
5985 005172
                                                                                       SET
                                                                                                        ACTFLAG
                                                                                                         XXDPCHAIN
                                                                            END : OF IF 42
END : OF IF 42
END : OF IFB SENV
5986 005172
```

5988	005172				*SUBTES	SUBTST	<pre><<protect &="" l="" loa<="" pre="" program="" protect=""></protect></pre>	********
5989 5990 5991 5992 5993	005172 005200 005206 005216 005224	052737 052737	000200	002650 002654	,	BIS BIS IF #\$ENI IF NO	#BIT7,CONFIG #BIT7,CONFIG+4 DAD NE 42 22BIT NE #0 T MONFLG	PROTECT PROGRAM SPACE (BANK 0) PROTECT LOADER SPACE (BANK 1) NOT ACT-11? RETURN TO XXDP MONITOR
5994 5995 5996 5997 5998	005232 005234 005236 005242 005242	104064				ELSE TYI END	ROR +64	TYPE PROGRAM TITLE
5999	005242					SUBTST	<< CHECK SYSTEM FOR CA	ACHE>>
					*SUBTE	ST	CHECK SYSTEM FOR CACH	Æ
6001 6002 6003	005040				*****	* THIS	FIGURES OUT IF THERE TYPE OF SYSTEM IT IS, ISABLED.	IS A CACHE ON THE SYSTEM, , AND WHETHER IT IS ENABLED
6004	005242 005250	005737	177746			SET4	#3\$ CONTRL	:IS THERE A CONTROL REGISTER?
6006	005254 005262	005737	177750			SET4 TST	#2\$ MAINT	:IS THERE A MAINTENANCE REGISTER?
6001 6002 6003 6004 6005 6006 6007 6008 6009 6010	005266 005274 005300	005737	177754			SET4 TST TYPE	M18 DATARG MSG117 48	:IS THERE A DATA REGISTER?
6012 6013	005304 005306 005314	104064			1\$:	BR SET ERROR	MONFLG +64	: 11/34
6014		104004			2\$:	SET	MONFLG	PROCESSOR NOT SUPPORTED BY THIS DIAGNOSTIC
6015 6016 6017	005316 005324	104064			24.	ERROR	+64	PROCESSOR NOT SUPPORTED BY THIS DIAGNOSTIC
		052737 042737 032737	000014 000014 000004	177746 177746 177746	4\$:	BIS BIC BIT	#BIT2!BIT3,CONTRL #BIT2!BIT3,CONTRL #BIT2,CONTRL	SET CACHE DISABLE BITS
6021 6022 6023 6024	005326 005334 005342 005350 005352 005360 005366 005376 005376 005402 005404	042737 032737 001004 032737 001413	000010	177746	7\$:	BNE BIT BEQ TYPE	7\$ #BIT3,CONTRL 6\$ MSG121	CLEAR CACHE DISABLE BITS IS THE BIT SET? BRANCH IF THE BIT IS SET IS THE BIT SET? BRANCH IF THE BIT IS SET CACHE BYPASSED
6025 6026 6027	005366 005370 005376 005402	104424 013737 005037 000404	002540 002540	002542		CACHOFF MOV CLR BR	CACHKN, CACHKN+2 CACHKN 8\$:SAVE INFO ABOUT CACHE :CACHE CANNOT BE USED - IT'S BYPASSED
6029 6030	005404 005410				3\$: 6\$:	TYPE	MSG119 MSG120	CACHE AVAILABLE

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 155 CHECK SYSTEM FOR CACHE

6032	005414				;***** ;*SUBTE	SUBTST	SETUP USER & SU	*******	*****	******
6034	005414 005416 005422	104421 005737 001011	002452		8\$:	DEENERG TST BNE	IZE NOSUPER 5\$;TURN OFF ;IS THERE ;NO-SKIP S	MEMORY MANAGEMENT A SUPERVISOR MODE? UPERVISOR SETUP.	
6035 6036 6037 6038 6039	005424 005432	042737 052737	030000 010000	177776 177776		SET PRI	EVIOUS MODE TO S #BIT13!BIT12,PS #BIT12,PSW	UPERVISOR		
6040 6041 6042	005440 005444	006606				PUSH MTPI	#SUPSTK SSP			
6042 6043 6044 6045	005446	052737	030000	177776	5\$:	SET PRI	EVIOUS MODE TO U #BIT13!BIT12,PS	ISER SW		
6048	005454 005460	006606				PUSH	#USESTK USP			
6050	005462				:***** :*SUBTE		*****	*******	STER IF NECESSARY>	*******
6051 6052 6053 6054	005462 005470 005500 005502 005502	104407			;*****	IF SAUT IF SW GTS END :	O IS FALSE R EQ #SWREG WR OF IF SWR IF \$AUTO	; 1	; IF NOT (APT C F SOFTWARE SWITCH F ;; GET SOFT-SU	REG SELECTED
6057	005502				:***** :*SUBTE	SUBTST ********	< <get mana<="" memory="" td=""><td>*******</td><td>******</td><td>**********</td></get>	*******	******	**********
6058	005502	104422			;*****	KMAP	*****	MAP KERNE	L SPACE 1 TO 1 RVISOR SPACE (TEST A	DEA) 1 TO 1
	005504	104420				MAP ENERGIZ	E	TURN ON	MEMORY MANAGEMENT	MEAN TIOT

063	005522				*TEST	****	COBIT TEST OF ALL	*****	> ***********	********	******
5064 5065 5066 5067 5068 5070 5071 5072 5073	005522	000004			THE	2) TEST 3) FIGU 4) TEST 5) IF	LE, AND STORES AND IS THE CSR BITS COURES OUT IF THE MO IS THE BITS PARTICANY BITS TEST BAD CSR INFORMATION	MMON TO DULE IS DULAR TO IN THE CO	FOR "TOTCSRS ALL CSR'S. A ECC OR PARI THAT TYPE OF SR UNDER TEST CLEARED.	TY MEMORY	N THE
5074 5075					* OF C:	TYPE	NOT USED BIT2	ECC TYPE BIT1	BITO	CODE TOTALS	
6076 6077 6078 6079						MS11-L MS11-M MS11-P	0	0	0	0 1 3	
6080 6081						MEMORY	CODE WILL BE USED	IN THE	SECOND PART O	F THIS ANALYSIS	
6084 6085 6086	005524 005526 005530 005534 005542	005005 005000 012703 012737	172100 000001	002074		CLR CLR MOV MOV SET4 REPEAT	RO #CSRADD.R3	:RO IS A	HE TOTAL CSR TABLE INDEX THE CSR ADDRE PARITY ERRORS	SS	
6090	005550 005550 005552	005713 052705	000001			BIS	#1,R5	:MARK IT	IS CSR RESPON IN CSR MAP HE LAST CSR I		
6091 6092 6093 6094 6095	005556 005560 005566 005574 005576	005004 042760 052760 005013	000004 000030	002456 002456		BIC BIS CLR LET	#4,CSRINFO(RO) #BIT4!BIT3,CSRIN	CLEAR U	NUSED BITS YES-MARK IT I HE CSR UNDER	N CSR INFORMATION	
6096 6097	005602	004737	005714			END	CALL ECCTYPE	FIGURE	OUT WHAT KIND	CC MEMORY??? OF ECC MEMORY WE	HAVE
6099 6100	005614 005616	005013 004737	006206			CLR	(R3) L RWCSR	BIT TES	SR UNDER TEST	IN CSR'S	
6101 6102 6103	005632 005640	016037 104021	002456	002050		11	CALL ECCTYPE (R3) L RWCSR CSRINFO(R0) MI #30 MOV CSRINFO(R0),BA ERROR +21	D':MOVE	IN BAD DATA	CONFIGURATION:	
6104 6105 6106 6107	005574 005576 005602 005610 005614 005614 005622 005632 005640 005642 005642 005646 005652 005654 005656 005660 005660 005670 005672	062700 062703 006305	000002		NXTCSR:	F145	#2 PO	GO TO N			
6109	005656	005204				END		•			
6111	005660 005666 005670	006005 005704				UNTIL ROR TST	RO EQ #40 R5 R4	RESYNC WAS THE	LL CSR'S ARE R5 RE A CSR 0? IF NOT EQUAL IT IN CSR TA	DONE	
6115	005674	052705	100000			RNE BIS	22 \$ #BIT15,R5	YES SET	IT IN CSR TA	ABLE	

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 157-1

6116 005700 6117 005704 004737 005774 6118 005710

LET TOTCSRS := R5 CALL CSRMAP JUMPTO CTEST 22\$:

STORE CSR MAP IN TOTCSRS

6120 005714				SUBTST < <determine ecc="" memory="" of="" type="">> :***********************************</determine>
6121 6122 6123 6124 6125 005714				THIS ROUTINE WILL DETERMINE IF THE ECC MEMORY UNDER TEST IS A MS11-M OR A MS11-P
6125 005714 6126 005714 6127 005722	052760	000001	002456	ECCTYPE: BIS #BITO,CSRINFO(RO) ; MARK IT IN THE TABLE AS BEING A ECC MEMORY LET (R3) := #60004 ; IS THIS A MS11-P???
6128 005726 6129 005734 6130 005736 6131 005742	005013			LET (R3) := #60004 ; IS THIS A MS11-P??? IF #BIT11 SET.IN (R3) ; IS BIT 11 SET??? CLR (R3) ; CLEAR CSR LET (R3) := #20004 ; ENABLE CHECK/SYNDROME BIT REGISTER LET (R3) := #27744 ; IT IS BUT MAKE SURE AGAIN
6132 005746 6133 005754 6134 005762	052760	000002	002456	IF (R3) EQ #23744 ;DO WE HAVE 6 OR 7 CHECK BITS IN CSR BIS #BIT1,CSRINFO(R0) ;6 CHECK BITS-MARK IT A MS11-P END ;
6135 005762 6136 005764 6137 005772	042760	000002	002456	BIC #BIT1, CSRINFO(RO); MARK IT IN THE TABLE
6138 005772	000207			RETURN

614	0 005774				CSRMAP:	STST < <print csr="" map="" register="">></print>	********
					*SUBTE		******
614	1 005774 2 005776 3 006002	005000			;*****	RO CLEAR CSR INFO POINT PE MSG008 PRINT TITLE PRINT CSR NUMBERS R1	
614	4 006006 5 006010	005001				PEAT :	
614	6 006010 7 006012 8 006016	010102 022702 100002	000011			MOV R1,R2 CMP #9.R2 BPL 1\$: JUMP AROUND NEXT INS	TRUCTION
41	0 006020	010102 022702 100002 062702 062702 110237	000007 000060 074704		1\$:	ADD #7 R2 ADD #60 R2 MOVB R2.MSG015 TYPE MSG015 TYPE MSG014 TYPE MSG014 TYPE SINGLE SPACE	
61	0 006024 0 006024 1 006030 2 006034 3 006040 4 006044 5 006046 6 006054	005201				TYPE MSG014 :TYPE SINGLE SPACE INC R1	
61 61	66 006054 67 006060					INC R1 IIL R1 EQ #16. PE MSG009 : TYPE MEMTYPE PEAT IF CSRINFO(RO) NF #0 :IS CSR NONEXSISTANT?	777
61	8 006060 9 006066					IF CSRINFO(RO) NE #0 :IS CSR NONEXSISTANT? IF #BITO SET.IN CSRINFO(RO) IF #BIT1 SET.IN CSRINFO(RO)	
61	006076 006106	112737	000120	074704		MOVB #'P,MSG015 ; 11 15 A MS11-P	
61 61	006106 02 006114 03 006116 04 006124	112737	000115	074704		MOVB #'M, MSG015 ; IT IS A MS11-M	
61	5 006124					IF #BIT1!BITO OFF.IN CSRINFO(RO)	
61	006114 03 006116 04 006124 05 006124 06 006124 07 006134 08 006142 09 006142 09 006142 09 006142 09 006152 09 006152	112737	000114	074704		MOVB #'L,MSG015 ; IT IS A MS11-L	
61	0 006142	112737	000040	074704		MOVB #' ,MSG015	
61	72 006152 73 006156					TYPE MSG015 TYPE MSG014 ;TYPE MEMORY ;TYPE SPACE	TYPE
61	75 006164 76 006170	000240 062700	000002			NOP ADD #2.RO POINT TO NEX TIL RO EQ #40	
61 61 61	78 006202	000207			TRACE:	PE MSG129 TURN ORD 0	

PR	INT CSR REGISTE	R MAP						
	6181 006206				*SUBTES	SUBTST < <read a<="" and="" read="" st="" td="" write=""><td>*********</td><td>******</td></read>	*********	******
	6182 6183 6184				THIS P	ROUTINE 'RWCSR' CHECK TO	SEE THAT THE CSR CAN BEWRITTEN ON CORRE FOLLOWING PATTERNS:	CTLY
	6182 6183 6184 6185 6186 6187 6188 6189 6190					1-ZEROS 2-ONES 3-SHIFTING A ONE 4-SHIFTING A ZERO	THROUGH A FIELD OF ZEROS OS THROUGH A FIELD OF ONES	
	6191 006206 6192 006206 6193 006216 6194 006220 6195 006222 6196 006226 6197 006232 6198 006242 6199 006246	006205			ŔWCSR:	ASR R5 LET CSRNO := R5	SAVE R4,R5, AND UIPARO ON STACK GET CSR NUMBER FOR POSSIBLE ERROR GET ADDRESS FOR POSSIBLE ERROR R0); WHAT KIND OF MEMORY IS THIS??? MASK FOR MS11-M/P IT IS A MS11-L MASK FOR MS11-L	GET BIT MASKS FOR D
	6200 006250 6201 006254 6202 006254 6203 006262 6204 006266 6205 006270		002316			LET CSR1S := #177777 BIC R5,CSR1S LET (R3) := #0	SET CSR1S TO ALL ONES CLEAR BITS FOR GOOD DATA O>CSR MASK OUT UNWANTED BITS	
	6207 006274 6208 006300 6209 006304 6210 006310 6211 006312	104035 042760	000010	002456		LET GOOD := #0 LET CSR := R4 ERROR +35 BIC #BIT3,CSRINFO(RO)	DO WE HAVE A CORRECT READ GOOD DATA=0'S BAD DATA=CSR BIT SET ERROR CLEAR CSR OK BIT	
	6212 006320 6213 006320 6214 006324 6215 006326 6216 006330 6217 006332 6218 006340	005013 040504				LET R4 := (R3) CLR (R3) BIC R5,R4	:ONES>(R3) :MASK OUT CORRECT FIELD :CLEAR OUT CSR :WAS PATTERN WRITTEN CORRECTLY? :GOOD DATA = ALL LEGAL BITS SET IN CSR	
	6213 006320 6214 006324 6215 006326 6216 006330 6217 006332 6218 006340 6219 006346 6220 006352 6221 006354 6222 006362 6223 006362 6224 006366	104010 042760	000010	002456		LET CSR := R4 ERROR +10 BIC #BIT3,CSRINFO(R0 END LET PASFLG := #0 REPEAT	GOOD DATA = ALL LEGAL BITS SET IN CSR BAD DATA=CSR BIT CLEAR ERROR CLEAR CSR OK BIT SET UP LOOP COUNTER REPEAT WITH A FIELD OF 1'S THROUGH O'S	
	0227	005237	002262			INC PASFLG LET UIPARO := #-1 IF PASFLG EQ #1 LET R2 := #1 ELSE LET R2 := #177776	O'S THROUGH 1'S :INCREMENT LOOP COUNTER :USE USER PAR FOR BIT COUNTER :PASS 1 :1>FIELD OF ZEROS :PASS 2 :0>FIELD OF ONES	
	6232 006422 6233 006422 6234 006422	005237	177640			REPEAT INC UIPARO	DO BITS 0-4 AND 13-15 INCREMENT BIT POINTER	

```
6235 006426
6236 006446
6237 006452
6238 006452
6239 006454
6240 006456
6241 006460
6242 006462
6243 006464
6244 006470
6245 006474
6246 006500
6247 006510
6248 006512
                                                                     IF PASFLG EQ #2 AND #BITO OFF.IN CSRINFO(RO) ;
                                                                         BIC #BIT14!BIT2,R2 ; IF THIS IS PASS 2 ON A MS11-L, CLEAR EUB BIT AND WRITE
                  042702 040004
                                                                     END
                                                                                                ; WRITE DATA
                                                                     LET (R3) := R2
                                                                                                GET GOOD DATA AND MASK IT OUT
                                                                     LET R1 := R2
                                                                     BIC R5,R1
                                                                                                GET GOOD DATA
                  040501
                                                                                                GET DATA THAT IS READ
                                                                     LET R4 := (R3)
                                                                     BIC R5,R4
                                                                                                :MASK OUT CSR BITS
                  040504
                                                                                                :IS DATA CORRECT???
                                                                     IF R1 NE R4
                                                                                                :BAD DATA = CSR CONTENTS
                                                                         LET BAD := R4
                                                                                                GET GOOD DATA
                                                                         LET CSR := R1
                                                                          IF PASFLG EQ #1
ERROR +35
                                                                                               SELECT ERROR DEPENDING ON PASS
                                                                                                BIT SET ERROR
                  104035
                                                                                                :PASS 2
                                                                          ELSE
       006512
                                                                            ERROR +10
                                                                                                :BIT CLEAR ERROR
       006514
                  104010
6250 006516
                                                                          END
6250 006516
6251 006516
6252 006524
6253 006524
6254 006534
6255 006536
6256 006540
6257 006542
6258 006544
6259 006544
6260 006554
6261 006564
6262 006600
6263 006604
                                                                         BIC #BIT3, CSRINFO(RO) ; CLEAR CSR OK BIT
                  042760
                             000010 002456
                                                                     END
                                                                      IF PASFLG EQ #1
                                                                                                GET DATA FOR NEXT LOOP
                                                                                                SHIFT 1 ACROSS 0'S
                                                                          ASL R2
                  006302
                                                                     ELSE
                                                                         SEC
                                                                                                :SET CARRY
                  000261
                                                                                                ROTATE A O ACROSS A FIELD OF ONES
                                                                          ROL R2
                  006102
                                                                     END
                                                                   UNTIL UIPARO EQ #15. ;UNTIL ALL BITS ARE DONE
                                                              UNTIL PASFLG EQ #2 ; DONE WITH 2 PASSES
IF #BITO SET.IN CSRINFO(RO) THEN JUMPTO DONE ; IF MS11-L DO ONE LAST WRITE
                                                                                                ; WRITE ONES TO CSR WITH EUB BIT ENABLED ; READ CSR FOR CORRECT BITS
                                                               LET (R3) := #140005
                                                               LET R2 := (R3)
BIC #37772,R2
                                                                                                CLEAR UNWANTED BITS
6264 006606
                  042702 037772
                                                                                                : WAS WRITE CORRECT
6264 006606
6265 006612
6266 006620
6267 006626
6268 006632
6269 006634
                                                               IF R2 NE #140005
                                                                   LET GOOD := #140005
                                                                                                : GOOD DATA
                                                                                                :BAD DATA
:BIT CLEAR ERROR
                                                                   LET CSR := R2
                                                                   ERROR +10
                   104010
                                                                   BIC #BIT3, CSRINFO(RO) ; CLEAR CSR OK BIT!
                  042760
                             000010 002456
                                                               END
       006642
                                                               LET (R3) := #0
                                                                                                :CLEAR OUT CSR
       00664
                                                   DONE:
                                                                                                RESTORE UIPARO, R4, AND R5
       006644
                                                               POP UIPARO, R5, R4
       006654
                  000207
                                                               RETURN
```

6277 6278	THE FOLLOWING	ROUTINE DETERMINES WHICH	CSR CONTROLS PROGRAM SPACE
6277 6278 6279 006656 104424 6280 006660 012737 002000 172350 6281 006666 C12737 002000 172350 6282 006674 012701 002406 6283 006700 012737 100000 002406 6284 006706 012737 100000 002406 6285 006714 005000 6286 006716 005037 002150 6287 006722 013703 002222 6288 006726 000240 6289 006730 006303 6290 006732 00303 6290 006734 006200 6292 006740 010037 002150 6293 006744 005070 6291 006734 0062700 000002 6292 006740 001037 002150 000002 6292 006740 001037 002150 000002 6292 006740 001037 002150 000002 6292 006740 001037 002150 000002 6292 006740 001014 6303 006766 032760 000003 002456 6299 006764 001014 6300 006766 052760 000004 172100 6301 006774 012771 123456 0000002 6303 007010 005060 172100 6304 007014 000414 6305 007016 012760 000000 172100 6306 007024 012771 123456 0000002 172100 6306 007024 012771 123456 0000002 172100 6306 007024 012771 123456 0000002 172100 6306 007024 012771 123456 0000002 172100 6307 007032 012771 123456 0000002 172100 6308 007040 012760 020006 172100 6311 007056 032760 000003 002456	CTEST: CACHOFF MOV MOV MOV MOV	#177777,PGMCSR #2000,KIPAR4 #TESTADD,R1 #100000,TESTADD #100002,TESTADD+2	;SET UP MAP REGISTER
6283 006700 012737 100000 002406 6284 006706 012737 100002 002410 6285 006714 005000 6286 006716 005037 002150	MOV CLR CLR	KU	CLEAR CSR COUNTER
6286 006716 005037 002150 6287 006722 013703 002222 6288 006726 000240	MOV NOP	CSRNO TOTCSRS,R3	:OBTAIN CSR MAP :DEBUG AID
6289 006730 006303 6290 006732 103407 6291 006734 062700 000002	4\$: ASL BCS 1\$: ADD	R3 2\$ #2,R0	OBTAIN CSR MAP DEBUG AID PUT HIGH ORDER BIT INTO C BIT BRANCH IF CSR EXISTS UPDATE CSR COUNTER
6291 006734 062700 000002 6292 006740 010037 002150 6293 006744 005703 6294 006746 001474	MOV TST BEQ	RO CSRNO R3 3\$:IS MAP EMPTY? :BRANCH IF SO
6295 006750 000767 6296 006752 000240 6297 006754 000241	2\$: BR	45	
6298 006756 032760 000003 002456 6299 006764 001014	CLC BIT BNE	#BIT1!BIT0,CSRINFO(RO)	:IS THIS PARITY MEMORY?
6300 006766 052760 000004 172100 6301 006774 012771 123456 000000 6302 007002 012771 123456 000002	BIS MOV MOV	#BIT2, CSRADD (RO) #123456, a(R1) #123456, a2(R1) CSRADD (RO)	DEBUG AID CLEAR CARRY IS THIS PARITY MEMORY? BRACH IF NOT SET WRITE WRONG PARITY WRITE DATA
6302 007002 012771 123456 000002 6303 007010 005060 172100 6304 007014 000414	CLR BR		;RESTORE CSR
6305 007016 012760 000000 172100 6306 007024 012771 123456 000000 6307 007032 012771 123456 000002 6308 007040 012760 020006 172100 6309 007046 005771 000000 6310 007052 016004 172100	5\$: MOV MOV MOV	#0, CSRADD (R0) #123456, a (R1) #123456, a 2 (R1) #20006, CSRADD (R0) a (R1)	CLEAR THE CSR UNDER TEST
6308 007040 012760 020006 172100 6309 007046 005771 000000	6\$: MOV	#20006, CSRADD (RO) a(R1)	SET DIAG CHECK MODE : WRITE CHECKBITS TO CSR
6311 007056 032760 000003 002456 6312 007064 001003	MOV BIT BNE	CSRADD(RO),R4 #BIT1!BITO,CSRINFO(RO) 7\$	SET DIAG CHECK MODE WRITE CHECKBITS TO CSR WRITE CSR TO R4 PARITY MEMORY? BRANCH IF NOT
6313 007066 005704 6314 007070 100421	TST	R4 8\$ 1\$:PARITY ERROR? :BRACH IF SO :TRY NEXT CSR :DEBUG AID
6316 007074 000240 6317 007076 072427 177773	7\$: BR NOP ASH		: TRY NEXT CSR : DEBUG AID
6317 007076 072427 177773 6318 007102 042704 177600 6319 007106 032760 000002 002456	ASH BIC BIT	#-5.R4 #^C177.R4 #BIT1.CSRINFO(RO) 10\$ #157.R2	WHAT KIND OF ECC MEMORY IS THIS
6321 007116 012702 000157 6322 007122 000402	BNE MOV BR	#157.R2	; WHAT KIND OF ECC MEMORY IS THIS ; BRANCH IF MS11-P ; LOAD IN CORRECT CHECK BITS FOR MS11-M
6323 007124 012702 000040 6324 007130 020204 6325 007132 001300	10\$: MOV 11\$: CMP	11\$ #40.R2 R2.R4 1\$	CORRECT CHECK BITS FOR MS11-P CORRECT CHECKBITS? BRANCH IF NOT
6326 007134 010037 002526 6327 007140 000240	8\$: MOV 3\$: NOP	RO, PGMCSR	
6314 007070 100421 6315 007072 000720 6316 007074 000240 6317 007076 072427 177773 6318 007102 042704 177600 6319 007106 032760 000002 002456 6320 007114 001003 6321 007116 012702 000157 6322 007122 000402 6323 007124 012702 000040 6324 007130 020204 6325 007132 001300 6326 007134 010037 002526 6327 007140 000240 6328 007142 104502 6329 007144 012771 000000 000000 6330 007152 012771 000000 000000 6331 007160 023727 002526 177777 6332 007166 001402 6333 007170 000137 007642	CLRCSR MOV MOV	#0,a(R1) #0,a2(R1) PGMCSR,#177777	: DEBUG AID : CLEAR ALL CSR'S : RESTORE TEST LOCATIONS
6332 007160 023727 002326 177777 6332 007166 001402 6333 007170 000137 007642	EQ JMP	PGMCSR,#177777 FINT CLRMEM	:IF PROGRAM CSR NOT FOUND GO TO FINT :GO TO SIZING ROUTINE IF FOUND

6335 6336 6337 6338				DING ROUTINE, THIS ROUTINE TRIES
6339 007174 6340 007202 012771 1234 6341 007210 012771 1234 6342 007216 062737 0106 6343 007224 000766 6344 007226 012700 1773 6345 007232 013703 0023			#2\$ #123456,a(R1) #123456,a2(R1) #10000,KIPAR4 1\$; NE MEMORY TRAPS GO TO 2\$; WRITE DATA AT FIRST LOCATION OF BANK 2 IN BOARD ; WRITE DATA AT SECOND LOCATION OF BANK 2 IN BOARD ; UPDATE PAR4 TO POINT TO UPPER BOARDS ; KEEP GOING TILL NO MORE MEMORY
6340 007202 012771 1236 6341 007210 012771 1236 6342 007216 062737 0106 6343 007224 000766 6344 007226 012700 1773 6345 007232 013703 0023 6346 007236 062700 0006 6347 007242 010037 0023 6348 007246 006303 6349 007250 103403 6350 007252 005703 6351 007254 001405 6352 007256 000767	776 222 002 150	2\$: MOV MOV ASL	#-2,R0 TOTCSRS,R3 #2,R0 RO,CSRNO R3	:PUT CSR MAP IN R3 :UPDATE CSR COUNTER :UPDATE CSRNO
6349 007250 103403 6350 007252 005703 6351 007254 001405 6352 007256 000767 6353 007260 012760 0200 6354 007266 000763 6355 007270	006 172100	BCS TST BEQ BR 4\$: MOV	4\$ R3 5\$ 3\$ #20006, CSRADD (R0)	:BRANCH IF CSR EXISTS :ANY CSR'S LEFT? :BRANCH IF NOT :LOOK FOR NEXT CSR :SET DIAGNOSTIC CHECK MODE IN CSR
6339 007174 6340 007202 012771 1236 6341 007210 012771 1236 6342 007216 062737 0106 6343 007224 000766 6344 007226 012700 1777 6345 007232 013703 0027 6346 007236 062700 0006 6347 007242 010037 0027 6348 007246 006303 6349 007250 103403 6350 007252 005703 6351 007254 001405 6352 007256 000767 6353 007260 012760 0206 6354 007266 000767 6355 007270 6356 007276 012700 1777 6357 007302 012737 0026 6358 007310 005771 0006 6359 007314 062700 000	776 000 172350 000 002 150 040	5\$: BR SET4 MOV MOV TST	3\$ #6\$ #-2,R0 #2000,KIPAR4 a(R1)	BRANCH IF CSR EXISTS ANY CSR'S LEFT? BRANCH IF NOT LOOK FOR NEXT CSR SET DIAGNOSTIC CHECK MODE IN CSR LOOK FOR NEXT CSR NE MEMORY TRAPS NOW GO TO 6\$ RESET CSR POINTER REMAP PAR4 TO POINT TO BANK 2 TEST NONASSERTED LOCATIONS UPDTAE CSR POINTER
6360 007320 010037 002 6361 007324 022700 000 6362 007330 001535 6363 007332 032760 000	002 150 040 002 002456	6\$: ADD MOV CMP BEQ BIT BNE	#40,R0 10\$ #BIT1_CSRINFO(R0)	:NOT FOUND? :BRANCH IF NOT :GET TYPE OF ECC MEMORY :BRANCH IF MS11-P :MS11-M CHECK BITS
6356 007276 012700 177 6357 007302 012737 002 6358 007310 005771 000 6359 007314 062700 000 6360 007320 010037 002 6361 007324 022700 000 6362 007330 001535 6363 007332 032760 000 6364 007340 001003 6365 007342 012702 000 6366 007346 000402 6367 007350 012702 000 6368 007354 016004 172 6369 007360 072427 177		MOV BR 55\$: MOV 56\$: MOV ASH	55\$ #157,R2 56\$ #40,R2 CSRADD(R0),R4 #-5,R4	:MS11-P CHECK BITS :GET CSR CONTENTS
6370 007364 042704 177 6371 007370 020204 6372 007372 001401 6373 007374 000747 6374 007376 110037 002	600	BIC CMP BEQ BR	#-5,R4 #^C177,R4 R2,R4 7\$ 6\$ R0,PGMCSR	CLEAR ALL BUT CHECKBITS PROPER CHECKBITS? BRANCH IF SO TRY NEXT CSR IF NOT WRITE NON-ASSERTED CSR # IN PGMCSR NE TRAPS GO TO 8\$
6375 007402 6376 007410 012700 177 6377 007414 013703 002 6378 007420 062700 000 6379 007424 010037 002	776 222 002 150	7\$: MOVB SET4 MOV MOV 23\$: ADD MOV	#8\$ #-2,R0 TOTCSRS,R3 #2,R0 RO,CSRNO	; NE TRAPS GO TO 8\$; PUT CSR MAP IN R3 ; UPDATE CSR COUNTER ; UPDATE CSRNO
6380 007430 006303 6381 007432 103403 6382 007434 005703 6383 007436 001405 6384 007440 000767		ASL BCS TST BEQ BR	R3 24\$ R3 25\$ 23\$:BRANCH IF CSR EXISTS ;ANY CSR'S LEFT? :BRANCH IF NOT ;LOOK FOR NEXT CSR ;SET DIAGNOSTIC CHECK MODE IN CSR ;LOOK FOR NEXT CSR
4384 007450 000763	776 9002 9002 9150 9040	24\$: MOV BR 25\$: MOV TST 8\$: ADD MOV	#20006,CSRADD(R0) 23\$ #-2,R0 a2(R1) #2,R0 R0,CSRNO #40,R0	; LOOK FOR NEXT CSR ; TEST ASSERTED LOCATIONS
6391 007472 022700 000	040	CMP	#40,R0	

CZMSPAO MS11-L	/M/P	MEMORY DIAG.	MACRO M1113	26-APR-82 09:41	PAGE 163-1
----------------	------	--------------	-------------	-----------------	------------

6393 007500 0 6394 007506 0 6395 007510 0	001452 032760 001003 012702	000002	002456		BEQ BIT BNE MOV BR	10\$ #BIT1,CSRINFO(RO) 76\$ #157,R2 77\$
6397 007516 (6398 007522 (6399 007526 (6400 007532 (6401 007536 (6401 007500 (6401 007500 (6401 007500 (6401 007500 (6401 007500 (6401 007500 (6401 007500 (6401 007500 (6401 007500 (6401 007500 (6401 007500 (6401 007500 (6401 007500 (6401 007500 (6401 007500 (6401 007500 (6401 007500 (6401 007500 (6401 007500 (6401000 (6401 007500 (6401 007500 (6401 007500 (6401 0075	000402 012702 016004 072427 042704 020204	000040 172100 177773 177600		76\$: 77\$:	MOV MOV ASH BIC CMP BEQ	#40,R2 CSRADD(R0),R4 #-5,R4 #^C177,R4 R2,R4 9\$
6403 007542 6404 007544 6405 007550	000747 110037 052737	002527 100000	002526	9\$:	BR MOVB BIS CLRCSR	8\$ RO,PGMCSR+1 #BIT15,PGMCSR
6407 007560	104502 012737	002000	172350		MOV SET4	#2000,KIPAR4
6409 007574 6410 007602 6411 007610	012771 012771 062737 000766	000000 000000 010000	000000 000002 172350	11\$:	MOV MOV ADD BR	#0,a(R1) #0,a2(R1) #10000,KIPAR4
6413 007620	104423			12\$:	CACHON	CLRMEM
6415 007624	012737	001000	172350	10\$:	MOV TYPE	#1000 KIPAR4 MSG126
6416 007632 6417 007636	005037	002526			CLR	PGMCSR

:CHECK FOR TYPE OF ECC MEMORY :BRANCH IF MS11-P :CHECK BITS FOR MS11-M :CHECK BITS FOR MS11-P

;PROPER CHECKBITS? ;BRANCH IF SO ;TRY NEXT CSR IF NOT ;WRITE ASSERTED CSR # IN PGMCSR ;SET INTERLEAVED INDICATOR IN PGMCSR

; NE MEMORY TRAPS GO TO 12\$
; WRITE DATA AT FIRST LOCATION OF BANK 2 IN BOARD
; WRITE DATA AT SECOND LOCATION OF BANK 2 IN BOARD
; UPDATE PAR4 TO POINT TO UPPER BOARDS

:ERROR - PROGRAM CSR NOT FOUND!

6419	007642					SUBTST	< <clear all="" mem<="" th=""><th>ORY SPACE FROM BANK 2 ON>></th></clear>	ORY SPACE FROM BANK 2 ON>>
					*SUBTES	T	CLEAR ALL MEMOR	Y SPACE FROM BANK 2 ON
6420					:			
6420 6421 6422 6423 6424 6425					CONTINU	DUTINE CI	LEARS ALL MEMORY L THERE IS NO ME	SPACE BEGINNING AT ADDRESS 200,000 AND MORY LEFT. IT SHOULD CLEAR ANY PARITY ERRORS NO CLEAN UP ANY JUNK LEFT HANGING AROUND IN
6423					; CREATED;	BY THE MEMORY.	LAST ROUTINE, A	ND CLEAN UP ANY JUNK LEFT HANGING AROUND IN
6425 6426	007642 007650				CLRMEM:		#CLREX	; NONEM TRAPS GO TO CLREX
6427	007650	005037 012737 012737 012701 020127 001003 012737 005021 005737 001001 000765 062737 022737	006204 000001 002000 100000 117776	002074		CLR	TRACE #1,NOPAR	· IGNORE PARITY FRRORS
6429	007654 007662 007670 007674 007700 007702 007710 007712	012737	002000	002074 172350		MOV	#2000 KIPAR4	:IGNORE PARITY ERRORS :SET UP MAP TO START AT BANK 2 :R1 MAPS TO KIPAR4 :WHOLE 16K BANK DONE?
6430	007670	012701	100000			MOV	#100000_R1	R1 MAPS TO KIPAR4
6431	007674	020127	117776		1\$:	CMP	R1,#117776	WHOLE TOK BANK DONE?
6432	007700	001003	177777	006204		BNE	2\$ #-1,TRACE	LISE TRACE FLAG TO FLAG END OF BANK
6434	007710	005021	1,,,,,	000204	2\$:	CLR	(R1)+	:KEEP GOING IF NOT :USE TRACE FLAG TO FLAG END OF BANK :CLEAR CONTENTS & INCREMENT :EOB FLAG SET?
6434	007712	005737	006204			TST	TRACE	;EOB FLAG SET?
6436 6437	007716	001001				BNE	3\$ 1\$	GO TO NEXT BANK IF SO
6438	007720	062737	000200	172350	3\$:	BR ADD	#200,KIPAR4	SET MAP FOR NEXT BANK
6439	007720 007722 007730	022737	000200 170000	172350	34.	CMP	#170000_KIPAR4	SET MAP FOR NEXT BANK ARE WE AT THE PERPHERIAL PAGE YES-GO ON
6440	007736	001405				BEQ	CLREX	:YES-GO ON
6441	007740	005037 012701 000751	006204			CLR	TRACE	RESET FLAG
6442	007744	000751	100000			MOV BR	#100000,R1	CLEAR NEXT BANK
6444	007752	000240			CLREX:	NOP		, ceern nen en en
6445	007750 007752 007754	000240 005037	006204			CLR	TRACE	
6446	007760					RES4		

6449	010002				ANA2	******	< <match all="" csr's="" td="" with<=""><td>********</td></match>	********
					:*SU	BTEST	MATCH ALL CSR'S WITH ME	MORY
6450 6451						HE SECOND P	ART OF THE ANALYSIS MATC	HES UP THE CSR'S WITH THE MEMORY, AND N THE CONFIGURATION TABLE. FOR ECC.
6452					* T	HIS IS DONE	RY TAKING FACH CSR FOUN	D IN THE PREVIOUS SECTION SEQUENITALLY
6452 6453					:* A	ND CHECKING	THROUGH ALL OF MEMORY,	ONE BANK AT AT TIME, TO SEE WHICH BANKS FIRST DOUBLE WORD PAIR IN EACH BANK ARE
6455					* * W	IRITTEN WITH	DATA AND DIAGNOSTIC CHE	CK MODE SET IN THE CSR IN URDER TO AC-
6454 6455 6456 6457					.* (OMPLISH THIS	S. ALL POSSIBLE CONFIGUR	ATIONS OF DOUBLE WORD PAIRS (NON-INTER- ERLEAVED) ARE CHECKED FOR EACH BANK
6457					. + T	HADDIICH LICE	NE TESTADO AND KERNEL IN	STRUCTION PAGE ADDRESS REGISTERS 4 AND
6458 6459					: * 5	. IF WE GET	THE PROPER CHECKBITS BA	CK, WE HAVE A MAICH. IF NUI, THE RUUT-
6460 6461					** 1	NE CHECKS P	OR SINGLE OR DOUBLE BIT IF ONE OR THE OTHER I	S FOUND, THE ERROR ADDRESS IS CHECKED
6462					* I	O SEE IF IT	IS THAT BANK. IF IT IS	S FOUND, THE ERROR ADDRESS IS CHECKED , WE HAVE A MATCH. AT THE END OF EACH
6463					*	'I'', WHICH D	ENOTES THE FOLLOWING:	OGRAM COMES UP WITH A NUMBER, STORED IN
6465					:*		MEMORY DESCRIPTION	
6466 6467					*	-		
6468					*	0	NON-EXISTANT MEMORY NON-INTERLEAVED MEMORY	MS11-L_MS11-P
6470					*	2	64K INTERLEAVED, A1 NOT	ASSERTED MEMORY
6469 6470 6471 6472					*	3	128K INTERLEAVED, A1 NO 64K INTERLEAVED, A1 ASS	FRIED MEMORY
6473					;*	5	128K INTERLEAVED, AT AS	SERTED MEMORY
6474						NOTE - I=2	THROUGH I=5 CAN ONLY OCC	UR WITH MS11-M MEMORY.
6476 6477 6478					* 1	OTE THAT PA	RITY MEMORY WRITES WRONG TY ERROR BIT TO BE SET.	PARITY TO THE DOUBLE WORDS, THEN LOOKS IF THE BIT IS SET, WE HAVE A MATCH.
	010002				;*	SET4	#100\$; NE MEMORY TRAPS GO TO 100\$
6481	010010	005037	002310			MOV	CHECK #TESTADD_R1	CLEAR CHECK SET UP THE VIRTUAL ADDR. POINTER
6483	010020	013703	002222			MOV	#TESTADD R1 TOTCSRS R3	MOVE CSR MAP INTO R3
6484	010024	005000				CLR	RO R5	CLEAR THE CSR PUINTER
6486	010030	005737	002450			TST	NO22BIT	; IS THIS AN 11/44 OR 11/24?
6488	010010 010014 010020 010024 010026 010030 010034 010036 010042 010052 010054 010060 010062 010060 010076 010076	005037 012701 013703 005000 005005 005737 001403 005037 000413 022737 001407 013702 072227 012737 012737 012737 012737 012737 012737 012737 012737 012737 012737 012737	002554			BEQ CLR	7\$ LASTBLOCK	SET UP THE VIRTUAL ADDR. POINTER MOVE CSR MAP INTO R3 CLEAR THE CSR POINTER CLEAR THE PROGRAM CSR STATUS POINTER IS THIS AN 11/44 OR 11/24? BRANCH IF IT IS ADJUST LASTBLOCK INDICATOR FOR 124K MACHINE BRANCH OVER NEXT PIECE OF CODE IS THERE UNIBUS MEMORY ABOVE 17000000?
6489	010042	000413	000167	002552	75:	BR CMP	1\$ #167,LASTBANK	BRANCH OVER NEXT PIECE OF CODE :IS THERE UNIBUS MEMORY ABOVE 17000000?
6491	010052	001407	000101	002772		BEQ	1\$	BRANCH IF NOT SET UP A NEW LAST BLOCK INDICATOR
6493	010054	005202	002552			MOV	LASTBANK,R2 R2 M9.,R2	SET OF A NEW EAST BEOCK INDICATOR
6494	010062	072227	000011			MOV	#9. R2 R2,LASTBLOCK	
6496	010072	012702	000004		15:	MOV	#4,R2	R2 IS INDEX FOR CONFIG TABLE
6497	010076	012737	001000	172350 172352		MOV	#1000,KIPAR4 #1000,KIPAR5	SET KIPAR4 FUR BANK I
6498	010112	006303	001000		2\$:	ASL	R3 3\$	DOES THIS CSR EXIST?
6500 6501		062700	000002			BCS	12,R0	SET KIPAR4 FOR BANK 1 SET KIPAR5 FOR BANK 1 DOES THIS CSR EXIST? BRANCH IF IT DOES EXIST INCREMENT THE CSR POINTER STORE IT IN CSRNO ALSO
6502		010037	002150			MOV	RO, CSRNO	STORE IT IN CSRNO ALSO

	CZMSPAO MS11-L/M/P MEMORY DIAG.	MACRO M1113	26-APR-82 09:41	PAGE 166-1
1	IMATCH ALL CSR'S WITH MEMORY			

HATCH MEE CON S WITH HEHON			
6503 010126 005703 6504 010130 001370 6505 010132 012737 001000 172353 6506 010140 012737 001200 172353 6507 010146 013706 002560 6508 010152 000137 011400 6509 010156 010037 002150 6510 010162 104424 6511 010164 000240 6512 010166 012737 100000 002404 6513 010174 012737 120002 002414 6514 010202 032762 000040 002654 6515 010210 001402 6516 010212 000137 011314 6517 010216 005037 002446 6518 010222 005771 0000000 6520 010232 6521 010260 012771 123456 000000 6524 010260 012771 123456 000000 6525 010266 012771 123456 000000 6526 010274 005060 172100 6527 010300 000411 6528 010302 012771 123456 000000 6529 010310 012771 123456 000000 6529 010310 012771 123456 000000 6530 010316 104503 6531 010320 104475 6532 010322 000240 6533 010324 005771 0000000 6534 010330 104475 6535 010332 000240 6536 010334 005771 0000000 6536 010334 005771 0000000 6537 010340 000240 6538 010342 010437 002430 6539 010346 104503	TST BNE MOV MOV MOV JMP 3\$: MOV 13\$: CACHOFF	R3 2\$ #1000,KIPAR4 #1200,KIPAR5 KSTACK,SP SUBAAS R0,CSRNO	:ARE THERE ANY MORE CSR'S TO DO? :BRANCH IF ALL CSRS NOT DONE :RESTORE KIPAR4 :RESTORE KIPAR5 :RESTORE STACK :JUMP TO SUBAAS IF ALL CSR'S ARE DONE :MAKE SURE CSRNO IS UPDATED :TURN THE CACHE OFF
6512 010166 012737 100000 002406 6513 010174 012737 120002 002416 6514 010202 032762 000040 00265 6515 010210 001402	6 45\$: MOV 0 MOV 0 BIT BEQ	#100000, TESTADD #120002, TESTADD+2 #BIT5, CONFIG(R2) 43\$	SET UP VIRTUAL ADDRESS TO KIPAR4 SET UP VIRTUAL ADDRESS TO KIPAR5 IS THIS A BANK TO SKIP? NO - BRANCH AROUND NEXT INSTRUCTION YES - GO TO END OF BANK CLEAR THE MEMORY CONFIGURATION COUNTER TEST TO SEE THAT THERE IS MEMORY PRESENT
6516 010212 000137 011314 6517 010216 005037 002446 6518 010222 005771 000000 6519 010226 005237 002446	43\$: CLR 4\$: TST INC	6\$ I a(R1)	:YES - GO TO END OF BANK :CLEAR THE MEMORY CONFIGURATION COUNTER :TEST TO SEE THAT THERE IS MEMORY PRESENT
6520 010232 6521 010242 032760 000003 00245	6 BIT	a(R1),a2(R1) #BIT1!BIT0,CSRINFO(R0) 34\$; IS THIS PARITY MEMORY?
6521 010242 032760 000003 00245 6522 010250 001014 6523 010252 052760 000004 17210 6524 010260 012771 123456 00000 6525 010266 012771 123456 00000 6526 010274 005060 172100 6527 010300 000411 6528 010302 012771 123456 00000 6529 010310 012771 123456 00000 6530 010316 104503 6531 010320 104475	0 MOV 2 MOV CLR	#8IT2,CSRADD(R0) #123456,a(R1) #123456,a2(R1) CSRADD(R0)	SAVE THE LOCATIONS UNDER TEST IS THIS PARITY MEMORY? NO - BRANCH SET WRITE WRONG PARITY SET THE FIRST LOCATION UNDER TEST SET THE SECOND LUT CLEAR THE CSR TEST LOCATIONS SET THE FIRST LOCATION UNDER TEST SET THE SECOND LUT RESET CSR SET DIAG. CHECK MODE IN CSR UNDER TEST DEBUG AID READ THE FIRST LUT TO WRITE CKBITS. INTO CSR READ THE CSR UNDER TEST DEBUG AID GET THE CHECKBITS FROM THE CSR DEBUG AID SAVE IN TEMP FOR LATER
6527 010300 000411 6528 010302 012771 123456 00000 6529 010310 012771 123456 00000 6530 010316 104503 6531 010320 104475	2 MOV CLR1CSR CB1CSR	41\$ #123456,a(R1) #123456,a2(R1)	SET THE FIRST LOCATION UNDER TEST SET THE SECOND LUT RESET CSR SET DIAG. CHECK MODE IN CSR UNDER TEST
6532 010322 000240 6533 010324 005771 000000 6534 010330 104426	41\$: NOP TST READCSR	a(R1)	READ THE FIRST LUT TO WRITE CKBITS. INTO CSR READ THE CSR UNDER TEST
6536 010334 013704 002146	MOV	CSR,R4	GET THE CHECKBITS FROM THE CSR
6535 010332 000240 6536 010334 013704 002146 6537 010340 000240 6538 010342 010437 002430 6539 010346 104503	MOV CLR1CSR	R4,TEMP	
6540 010350	POP	a2(R1),a(R1) #BIT1!BIT0,CSRINFO(R0) 42\$	RESTORE LOCATIONS UNDER TEST IS THIS PARITY MEMORY?
6541 010360 032760 000003 00245 6542 010366 001004 6543 010370 005704 6544 010372 100431 6545 010374 000137 011314 6546 010400 072427 177773 6547 010404 042704 177600 6548 010410 032760 000002 00245 6549 010416 001005 6550 010420 012737 000157 00231 6551 010426 000240 6552 010430 000404 6553 010432 012737 000040 00231 6554 010440 000240 6555 010442 023704 002312 6556 010446 000240 6557 010450 001402 6558 010452 000137 011002	TST BMI JMP 42\$: ASH	R4 25\$ 6\$ #-5,R4 #^C177,R4	RESET CSR RESTORE LOCATIONS UNDER TEST IS THIS PARITY MEMORY? NO - BRANCH DID WE GET A PARITY ERROR? YES - FILL IN CONFIG TABLE NO - JUMP TO END OF BANK MANIPULATE THE CSR BITS INTO A USABLE FORM. WHAT KIND OF ECC MEMORY IS THIS?? BRANCH IF MS11-P MS11-M CHECK BITS DEBBUG AIDE
6547 010404 042704 177600 6548 010410 032760 000002 00245	6 BIC	#BITT_CSRINFO(RU)	WHAT KIND OF ECC MEMORY IS THIS??
6549 010416 001005 6550 010420 012737 000157 00231 6551 010426 000240	NOP	76\$ #157,CBITS	; MS11-M CHECK BITS ; DEBBUG AIDE
6552 010430 000404 6553 010432 012737 000040 00231	2 76\$: BR	77 \$ #40,CBITS	MS11-P CHECK RITS
6554 010440 000240 6555 010442 023704 002312	77\$: NOP CMP NOP	CBITS,R4	DEBBUGGING AIDE DO THE CHECKBITS COMPARE TO WHAT WAS WRITTEN? DEBBUG AIDE BRANCH IF THERE IS A MATCH
6556 010446 000240 6557 010450 001402 6558 010452 000137 011002 6559	BEQ JMP ;*	25\$ 22\$	BRANCH IF THERE IS A MATCH ELSE BRANCH IF NOT THE SAME

6560 6561			* WE COME HERE IF THERE I	S A MATCH
6562 010456 010004		25\$:	MOV RO,R4	GET THE CSR NUMBER
6564 010462 006204 6565 010464 000304 6566 010466 042704 6567 010472 032737	170377 000004 00244	6	NOP ASR R4 SWAB R4 BIC #170377,R4 BIT #BIT2,I BEQ 15\$	SET IT UP FOR USE IN THE CONFIGURATION TABLE. CLEAR OFF EXTRANEOUS BITS INTERLEAVED A1 ASSERTED MEMORY FOUND? BRANCH IF NOT PUT CSR NUMBER IN INTERLEAVED CSR SLOT PUT CSR NUMBER IN CONFIG. TABLE GET MEMORY TYPE CLEAR OFF THE EXTRANEOUS BITS MOVE INTO PROPER POSITION SET IT INTO THE CONFIG TABLE WAS THIS NON-INTERLEAVED MEMORY? BRANCH IF IT WAS SET THE INTERLEAVED BIT SAVE THE CURRENT BANK INDEX WAS THIS 128K INTERLEAVED? BRANCH IF TRUE SET 64K INTERLEAVED FLAG IN CONFIG SET NEW BANK POINTER TO 4 BANKS AHEAD JUMP OVER NEXT INSTRUCTION SET NEW BANK POINTER 8 BANKS AHEAD SET SKIP ECC LOGIC TESTS FLAG (R4) IG+2(R4)
6568 010500 001402 6569 010502 072427 6570 010506 050462 6571 010512 016004 6572 010516 042704 6573 010522 000304	000004 002650 002456 177770	15\$:	BEQ 15\$ ASH #4.R4 BIS R4.CONFIG(R2) MOV CSRINFO(RO),R4 BIC #^C7.R4 SWAB R4	PUT CSR NUMBER IN INTERLEAVED CSR SLOT PUT CSR NUMBER IN CONFIG. TABLE GET MEMORY TYPE CLEAR OFF THE EXTRANEOUS BITS MOVE INTO PROPER POSITION
6574 010524 050462 6575 010530 022737	002652 000001 00244	6	BIS R4, CONFIG+2(R2)	SET IT INTO THE CONFIG TABLE WAS THIS NON-INTERLEAVED MEMORY?
6576 010536 001431 6577 010540 052762	010000 00265	2	BEQ 24\$ BIS #BIT12, CONFIG+2(R2 MOV R2,R4	(2) SET THE INTERLEAVED BIT SAVE THE CURRENT BANK INDEX
6578 010546 010204 6579 010550 032737	000001 00244	6	BIT #BITO,I	WAS THIS 128K INTERLEAVED?
6580 010556 001006 6581 010560 052762 6582 010566 062704 6583 010572 000402	004000 00265 000020	2	BNE 5\$ BIS #BIT11, CONFIG+2(R2 ADD #20,R4 BR 16\$	SET 64K INTERLEAVED FLAG IN CONFIG SET NEW BANK POINTER TO 4 BANKS AHEAD JUMP OVER NEXT INSTRUCTION
6580 010556 001006 6581 010560 052762 6582 010566 062704 6583 010572 000402 6584 010574 062704 6585 010600 052764 6586 010606 056264 6587 010614 056264 6588 6589 6590	000040 000040 00265 002650 00265 002652 00265	0	ADD #40,R4 BIS #BIT5,CONFIG(R4) BIS CONFIG(R2),CONFIG BIS CONFIG(R2),CONFIG	SET NEW BANK POINTER 8 BANKS AHEAD SET SKIP ECC LOGIC TESTS FLAG (R4) SET OTHER INFO INTO THAT BANK IG+2(R4)
6589			THIS SECTION IS EXECUTE	ED ONLY WHEN THE BANK=1
6590 6591 010622 022737 6592 010630 001402	001000 17235	0 24\$:	CMP #1000,KIPAR4 BEQ 30\$: IS THIS BANK 1 ? :BRANCH IF TRUE
6591 010622 022737 6592 010630 001402 6593 010632 000137 6594 010636 032737 6595 010644 001417	011154 100020 00243	30\$:	BEQ 30\$ JMP 33\$ BIT #BIT15!BIT4,TEMP BEQ 10\$	ELSE JUMP TO END OF THIS BANK
6596 010646 013704 6597 010652 072427 6598 010656 022704	002430 177767 000001		MOV TEMP,R4 ASH #-9.,R4 CMP #1,R4 BGT 10\$	GET CSR CONTENTS MAKE ERROR ADDRESS INTO BANK # ERROR IN BANKS 0 OR 1? BRANCH IF NOT SET ERROR FLAG IN CONFIG TABLE
6600 010664 052762 6601 010672 105262	000001 00265 002652	0	BIS #BITO, CONFIG(R2) INCB CONFIG+2(R2)	SET ERROR FLAG IN CONFIG TABLE ADD ONE TO BANK ERROR COUNT PRINT CONFIG TABLE
6603 010704 053737 6604 010712 053737	002654 00265 002656 00265	0 10\$:	BIS CONFIG+4, CONFIG BIS CONFIG+6, CONFIG+2	;SET UP INFORMATION IN BANK ZERO
6605 010720 000240 6606 010722 022737	000001 00244	6	NOP CMP #1.I	DEBUG AID WAS THIS NON-INTERLEAVED MEMORY
6599 010662 003010 6600 010664 052762 6601 010672 105262 6602 010676 6603 010704 053737 6604 010712 053737 6605 010720 000240 6606 010722 022737 6607 010730 001002 6608 010732 000137 6609 010736 012704 6610 010742 032737 6611 010750 001402 6612 010752 062704 6613 010756 053764	011314 000020 000001 00244	46\$:	BNE 46\$ JMP 6\$ MOV #20,R4 BIT #BITO,I BEQ 26\$; NO - BRANCH OVER NEXT STMT. ; YES - JUMP TO END OF THIS BANK ; SET UP COUNTER FOR 64K INTERLEAVED ; WAS IT 128K INTERLEAVED? ; BRANCH IF NOT ; SET UP COUNTER FOR 128K INTERLEAVED
6603 010704 053737 6604 010712 053737 6605 010720 000240 6606 010722 022737 6607 010730 001002 6608 010732 000137 6609 010736 012704 6610 010742 032737 6611 010750 001402 6612 010752 062704 6613 010756 053764 6614 010764 053764 6615 010772 052764 6616 011000 000465	000020 002650 0026 002652 0026 000040 0026	50 26\$: 52	BEQ 26\$ ADD #20,R4 BIS CONFIG.CONFIG(R4) BIS CONFIG+2.CONFIG+2 BIS #BIT5.CONFIG(R4) BR 33\$	SET UP COUNTER FOR 128K INTERLEAVED SET OTHER BANK WITH SAME INFORMATION (R4) AS IN BANK 0 SET SKIP ECC LOGIC TESTS FLAG BRANCH

519 520 011002 521 011010 522 011012	032737 001001	100020	002146	22\$:	BIT BNE BR	#BIT15!BIT4,CSR 8\$ 33\$;SBE OR DBE FLAGS SET? ;BRANCH IF TRUE ;CHECK TO SEE IF IT IS MS11-M
522 011012 523 011014 524 011020 525 011026	001001 000460 013704 042764	002150 000006	172100	8\$:	MOV BIC PUSH	CSRNO,R4 #6,CSRADD(R4) RO,R1	GET CSRNO TURN OFF DIAG CHECK & ECC DISABLE SAVE RO & R1
526 011032 527 011036	016401 072127	172100 177773			MOV ASH	CSRADD(R4),R1 #-5,R1 #^c177,R1	GET CSR INFORMATION SET UP ERROR ADDRESS
528 011042 529 011046 530 011054	042701 052764 016400	177600 040000 172100	172100		BIC BIS MOV	#BIT14,CSRADD(R4) CSRADD(R4),R0	GET EXTENDED ERROR ADDRESS BITS
631 011060 632 011066 633 011072	042764	040000 177037	172100		BIC BIC ASL	#81114,CSRADD(R4) #^C740,R0 R0	TURN OFF EUB BIT
634 011074 635 011076	006300 006300 060001 010104			27\$:	ASL ADD MOV	RO RO,R1 R1,R4	SET UP TOTAL ERROR ADDRESS
636 011100 637 011102 638 011106 639 011112	072427 020437	000005 172350			POP ASH CMP	R1,R0 #5,R4 R4,KIPAR4	:SAVE IN R4 :RESTORE RO & R1 :SET ERROR ADDRESS UP IN PAR NOTATION
640 011116 641 011120	001001				BNE BR	28 \$ 35 \$:DOES IT EQUAL KIPAR4? :BRANCH IF FALSE :YES - MARK INFO IN CONFIG TABLE
642 011122 643 011126	020437	172352		28\$:	BNE	R4,KIPAR5	BRANCH IF FALSE
644 011130 645 011136	052762 105262	000001 002652	002650	35\$:	BIS INCB SET	WEITO, CONFIG(R2) CONFIG+2(R2) CONFGERROR	YES - MARK INFO IN CONFIG TABLE DOES IT EQUAL KIPARS? BRANCH IF FALSE SET BANK ERROR FLAG INCREMENT BANK ERROR COUNTER PRINT CONFIG TABLE
646 011142 647 011150	000137	010456			JMP	25\$	YES - MARK INFO IN CONFIG TABLE
648 649 650					* TH	IS SECTION SETS UP ALL TH	E POSSIBLE CONFIGURATIONS OF
651 652 011154	032760	000001	002456	33\$:	BIT	#BITO,CSRINFO(RO)	:IS THIS MS11-M MEMORY? :NO - GO TO END OF BANK
653 011162 654 011164 655 011172		000002	002456		BEQ	#BIT1,CSRINFO(RO)	, NO - GO TO END OF BANK
655 011172 656 011174 657 011202 658 011204 659 011212 660 011220 661 011224 662 011232 663 011234 664 011242 665 011244 666 011252 667 011256 668 011264 669 011264 669 011274 671 011302 672 011310	032760 001050 022737 103410 162737 062737 000137 022737 103405 062737 000137 062737 103413 062737 162737 162737	000001	002446		CMP	6\$ #1, I	:IS THIS 1ST TIME THROUGH?
658 011204	162737	000002	002410 172352		SUB	18\$ #2.TESTADD+2	:IS THIS 1ST TIME THROUGH? :BRANCH IF NOT :TRY AS 64K INTERLEAVED :A1 NON-ASSERTED MEMORY :TRY TO MATCH AGAIN :4TH TIME THROUGH? :YES - BRANCH :2ND TIME THROUGH :NO - BRANCH :TRY AS 128K INTERLEAVED :TRY TO MATCH AGAIN :THIRD TIME THROUGH? :NO - BRANCH :TRY TESTING THE BANK :AS A1 ASSERTED :64K INTERLEAVED MEMORY :TRY TO MATCH AGAIN
660 011220	000137	000002 004000 010222 000004		100.	JMP	#4000,KIPAR5	TRY TO MATCH AGAIN
662 011232	001404		002446	18\$:	CMP BEQ	20\$	YES - BRANCH
664 011242	103405	000002			BLO	20\$ #2.I 12\$; NO - BRANCH
665 011244 666 011252	062737	004000 010222 000003	172352	20\$:	JMP	#4000,KIPAR5	TRY TO MATCH AGAIN
667 011256	022737	000003	002446	12\$:	CMP BLO	4\$ #3,1 6\$;THIRD TIME THROUGH? ;NO - BRANCH
669 011266	062737	000002	002406 002410 172352		ADD	#2,TESTADD #2,TESTADD+2 #4000,KIPAR5	TRY TESTING THE BANK
671 011302	162737	004000	1/2352		SUB	#4000 KIPAR5	:64K INTERLEAVED MEMORY

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 166-4 MATCH ALL CSR'S WITH MEMORY

6674 6675 6676 011314 6677 011316 6678 011322 6679 011330 6680 011336 6681 011340 6682 011346 6683 011350 6684 011354 6685 011360 6686 011362 6687 011364	104503 062702 062737 013737 000240 023737 101402 000137 062700 000240 104423 000137	000004 001000 172350 002554 010166 000002	172350 172352 172350	6\$: 19\$:	CLR1CSR ADD ADD	#4,R2 #1000,KIPAR4 KIPAR4,KIPAR5 LASTBLOCK,KIPAR4 19\$ 45\$ #2,R0	CLEAR THE CSR UNDER TEST UPDATE CONFIGURATION POINTER UPDATE KIPAR4 TO NEXT BANK AND UPDATE KIPAR5 DEBUG AID HAVE WE DONE THE WHOLE MEMORY SP BRANCH IF DONE JUMP IF NOT DONE INCREMENT CSR POINTER DEBUG AID TURN ON THE CACHE JUMP TO TRY NEXT CSR	ACE? R-C
6688 6689 011370 6690 011374	062706 000137	000004 011314		100\$:	ADD JMP	#4,SP 6\$	RESTORE STACK : GO TO END OF BANK ROUTINE :	R-C R-C

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 167 MATCH ALL CSR'S WITH MEMORY

6692 6693 6694	011400 011402 011404	104423 104472			SUBAAS:	CACHON ECCINIT NEWTST	:MAKE :TRAP < <test 0<="" bank="" th=""><th>SURE THE CON DOUBLE ACCESSES>></th><th>ACHE IS ON BIT ERRORS</th><th>(NORMAL)</th><th></th></test>	SURE THE CON DOUBLE ACCESSES>>	ACHE IS ON BIT ERRORS	(NORMAL)	
					*TEST	****	TEST BANK 0 AC	******	******	******	
	011404	000004			TST2:	SCOPE	*****	*****	******	*******	
6695	011404	000004				: THIS DO				CATION IN BANK #0 TO SEE	
6696 6697 6698 6699 6700						SINCE E	GETS ANY PARITY EVERY LOCATION TO THIS POINT -	IS EITHER THEN A PA	LOADED OR W	IRITTEN INTO BY THE PROGRAM IMPLIES THAT THERE IS A ON IS TAKEN ITY ERROR COUNTER O PARITY ERROR FLAG EXISTANT MEMORY ERROR COUNTER ON-EXISTANT MEMORY ERROR MODE TO COUNT OF GOTO NONEXIST	
6700						THESE	RRORS ARE COUN	ITED AND A	FATAL ACTIO	ON IS TAKEN	
6/01	011406	005037 012737 005037 012737	002070 000001	002074		CLR MOV	PARCNT #1_NOPAR		SET THE NO	PARITY ERROR FLAG	
6703	011412	005037	002066	002076		CLR	NEMCNT #1 NONEM		CLEAR NON-	EXISTANT MEMORY ERROR COUNTER ON-EXISTANT MEMORY ERROR MODE TO COUNT	
6705	011432		000001	002076		SET4	#NONEXIST		TRAPS TO 4	GOTO NONEXIST	
6706	011440 011442	005000 012701	040000			CLR MOV	RO #SIZE,R1				
6708	011446	104424	0.0000		10.	CACHOFF			TURN CACHE	OFF CAN DO A READ ACCESS WITHOUT A PARITY TRAF	,
6710	011452	005720 077102			1\$:	TST SOB	(R0)+ R1,1\$				
6711 6712	011454	104423				CACHON ; SEE IF	ANY FAILURES		; TURN CACHE	ON	
6713	011456	005737	002070			TST	PARCNT		: ANY PARITY	ERRORS?	
6714	011462 011464	001403				BEQ FATAL	2 \$				
6716	011472 011476	005737 001406	002066		2\$:	TST BEQ	NEMCNT 3\$		SKIP IF E	(ISTANT MEMORY (HOLES)?	
6718	011500	162737	000002	002032		SUB	#2.ADDRESS		:UPDATE 151	ADDRESS FAILURE FROM AUTO INCREMENT #	
6719 6720	011506 011514	053737	002104	002650	3\$:	FATAL BIS	CPUBIT, CONFIG		:SET CORREC	CT ACCESSED BIT ON BANK OPS TO 4 TO DEFAULT	
6721	011522					RES4			RESET TRAF	PS TO 4 TO DEFAULT	
6722 6723	011544					SUBTST	< <enable ecc<="" td=""><td>OR CORRECT</td><td>TRAPS>></td><td></td><td>i</td></enable>	OR CORRECT	TRAPS>>		i
					******		ENABLE ECC FOR	R CORRECT T	RAPS		
/70/	0115//				*****	*****	*****	******	******	********	
6725	011544 011562	104506				ENASB	SET.IN aSWR OF	A ACIPLAG I	TRAP ON S	INGLE BIT ERRORS	
6726	011564 011566	104472				ELSE	IT		:TRAP ON DO	OUBLE BIT ERRORS (NORMAL)	
6728	011570	104412				END : OF	IF #SWO				

ENABLE E	CC FUR	CURRECT	IRAPS					
6731	011570				*TEST	*****	< <test (007)<="" 1-200="" banks="" td=""><td>CTAL) FOR ZEROS & ONES>> ALLY FOR ZEROS & ONES ALLY FOR ZEROS & ONES</td></test>	CTAL) FOR ZEROS & ONES>> ALLY FOR ZEROS & ONES ALLY FOR ZEROS & ONES
6732 6733	011570	000004			fst3:	· FYCEPT	•	TANCE AND IF IT EXISTS
6732 6733 6734 6735 6736 6737						ANY RAI	PROTECTED BANKS (WHERE "TST" INSTRUCTIONS LIP RANKS ARE LOGGED IN	E THE PROGRAM IS) ARE ONLY TESTED BY KE BANK #0 THE CONFIGURATION TABLE. SMART SIZE - NOT ACTUAL TESTING!
6738	011572	005077	002100			CLR	BANK	START SIZE - NOT ACTUAL TESTING.
6739 6740 6741	011572 011576 011604 011612 011620 011626 011636 011636 011644 011654 011654 011670 011702 011704 011710 011712 011714 011720 011730 011730 011730	005037 012737 012737	002100 000001 000002	002074 002076		MOV	#1 NOPAR	SET NO PARITY ERROR FLAG
6742	011612	022737	000001	003752		SET4 CMP	#2,NONEM #NONEXIST #1,PROTYP	TRAPS TO 4 GOTO NONEXIST IS THIS AN 11/44? BRANCH IF TRUE
6744	011626 011630	022737 001407 012737	012426	002516		BEQ MOV	MMTST3+4,LINK1	; BRANCH IF TRUE ; SET UP LINKS
6746	011636	012737	012426 012430	002520		MOV BR	#MTST3+6.LINK2	
6748	011646	000411			15:	BMOV	TAG9\$ MTST3	; PUT IN FAST MEMORY
6749	011654	012737 012737 005237 023737 103457 013701 006301 006301 010137	177644	002516		MOV	WUIPAR2,LINK1	SET UP LINKS
6750	011662	012737	177644 177646 002100 002552	002520		MOV	#UIPAR3,LINK2	
6751	011670	005237	002100		TAG9\$:	INC	BANK	
6752	011674	023737	002552	002100		CMP	LASTBANK, BANK	:DONE?
6753	011702	103457				BLO	TAG2\$:YES - SKIP TO NEXT TEST
6754	011704	013701	002100			MOV	BANK,R1	
6/22	011710	006301				ASL	R1 R1	:BANK * 4
6757	011712	010137	002102			ASL MOV	R1,BANKINDEX	, DAIN - 4
6758	011720	005037	002102 002072			CLR	PATERR	:CLEAR PATTERN ERROR COUNTER
6759	011724	005037	002070			CLR	PARCNT	CLEAR PARITY ERROR COUNTER
6760	011730	005037	002066			CLR	NEMCNT	CLEAR PATTERN ERROR COUNTER CLEAR PARITY ERROR COUNTER CLEAR NON-EXISTANT MEMORY COUNTER (HOLES) MAP SUPERVISOR SPACE (TEST AREA) TO BANK
6761	011734					MAP	BANK	MAP SUPERVISOR SPACE (TEST AREA) TO BANK
6762	011750	105761	002650			TSTB	CONFIG(R1)	IS THIS BANK PROTECTED?
6763	011754	100555	000007	170572	LIADAIT .	BMI	ISTBANK	:YES - GO TEST BANK SPECIAL :PUT 'RETURN' INSTRUCTION AFTER WRITE ROUTINE
6764	011766	012777	000207 060000	170532	WARN1:	MOV	WEIDST DO	PUT RETURN INSTRUCTION AFTER WATTE ROOTTINE
6766	011704	012700	060000			MOV	RO R4	
6767	011772	100555 012777 012700 010004 012701 010103 005002 104424	040000			MOV	TSTBANK #207, aLINK1 #FIRST, RO RO, R4 #SIZE, R1	
6768	011776	010103	0.000			MOV	R1,R3	
6769	012000	005002				CLR	R1,R3 R2	DATA IS ZEROS TURN CACHE OFF ENTER SUPERVISOR MODE IS THIS AN 11/44? BRANCH IF TRUE
6770	012002	104424				CACHOFF TESTARE		TURN CACHE OFF
6771	012004	022777		007753		TESTARE	41 DDOTYD	TENTER SUPERVISOR MODE
6//2	012012	022/3/	000001	003752		CMP	#1,PROTYP	- PRANCH IF TRUE
6774	012020	001403	012422			CALL	MTST3	, DRANCH IT THOL
6775	012026	000402	012422			BR	2\$	
6776	012030	004737	177640		15:	CALL	FASTCITY	CALL TO THE USER INSTRUCTION PAR'S
6777	012034	104417			2\$:	KERNEL		ENTER KERNEL MODE
6778	012036	104423				CALL KERNEL CACHON		CALL TO THE USER INSTRUCTION PAR'S ENTER KERNEL MODE TURN CACHE ON
6779	012040	000416			*****	BR	TAG3\$	SKIP NEXT INSTRUCTION
6780	012042	022737 001403 004737 000402 004737 104417 104423 000416 005037	002100		TAG2\$:	CLR	BANK	DESET TRADS TO A TO DEFAULT
6/81	011750 011754 011756 011764 011770 011772 011776 012002 012002 012004 012012 012020 012020 012034 012036 012036 012036 012040 012040 012040 012040 012040	005077				RES4	NOPAR	RESET TRAPS TO 4 TO DEFAULT :INDICATE DEFAULT PARITY ACTION
4797	012070	005037 000564	002074			CLR BR	SUBAAI	, and the periods that it her told
0/03	012014	000004				UK.	3007412	

MSPAO MS11-L/I	M/P MEMO NKS 1-20	RY DIAG.	MACRO M FOR ZE	1113 26 ROS & ON	-APR-82	09:41 PAGE 169-1	SEQ 016
6784 012076 6785 012102 6786 012104 6787 012106 6788 012110 6789 012116 6790 012122 6791 012124 6792 012126 6793 012132 6794 012134 6795 012140 6796 012142 6797 012146 6798 012150 6799 012154 6800 012156 6801 012162 6802 012164 6803 012170 6804 012176 6805 012200 6806 012206 6807 012214 6808 012216 6809 012222 6810 012224 6811 012230 6812 012232 6813 012234 6814 012240 6815 012244 6816 012246 6817 012252	005737 001401 000671 104424			TAG3\$:	TST	NEMCNT 1\$ TAG9\$	ANY TRAPS? NO - SKIP NOW - TRY NEXT BANK TURN CACHE OFF ENTER SUPERVISOR MODE FINISH PATTERN ENTER KERNEL MODE TURN CACHE ON ANY PATTERN ERRORS YES - SKIP ANY PARITY ERRORS YES - SKIP ANY NON EXISTANT MEMORY YES - SKIP
6789 012110 6789 012116 6790 012122	004777 104417 104423 005737 001040 005737 001035 005737 001032 012700 010004 012701 010103 013702 012777 104424	170376			CALL KERNEL CACHON	aLINK2	FINISH PATTERN ENTER KERNEL MODE
6792 012126	005737	002072			TST	PATERR 2\$:ANY PATTERN ERRORS
6794 012134	005737	002070			TST	PARCNT 2\$	ANY PARITY ERRORS
6796 012142	001033	002066			TST	NEMCNT	ANY NON EXISTANT MEMORY
6798 012150	012700	060000			MOV	#FIRST,RO	, 1E3 - 3KIF
6800 012156	010004	040000			MOV	WFIRST,RO RO,R4 WSIZE,R1 R1,R3 ONES,R2 WOO0240,@LINK1	
6801 012162 6802 012164	010103	002600	170320		MOV	ONES,R2	DATA IS ONES
6803 012170 6804 012176 6805 012200			170320		CACHON TST BNE TST BNE TST BNE MOV MOV MOV MOV CACHOFF TESTARE CMP BEQ CALL	#000240,@LINK1	: TURN CACHE OFF :ENTER TEST MODE
6806 012206 6807 012214	022737	000001	003752		CMP BEQ	#1,PROTYP	:IS THIS AN 11/44? :BRANCH IF IT IS
6808 012216	004737	012422			CALL	24	DO IN MEMORY IF NOT
6810 012224 6811 012230	022737 001403 004737 000402 004737 104417 104423 013700 005737 001006 005737 001003	177640		5\$: 6\$:	CALL KERNEL CACHON	FASTCITY	DATA IS ONES PUT 'NOP' INSTRUCTION BACK IN SUBROUTINE TURN CACHE OFF ENTER TEST MODE IS THIS AN 11/44? BRANCH IF IT IS DO IN MEMORY IF NOT JUMP OVER NEXT INSTRUCTION CALL TO THE USER INSTRUCTION PAR'S ENTER KERNEL MODE TURN CACHE ON
6813 012234 6814 012240	013700 005737	002102 002072		2\$:	MOV TST BNE TST	BANKINDEX,RO PATERR	
6815 012244 6816 012246	001006	002070			TST	3\$ PARCNT	:ANY PATTERN ERRORS? :YES - SKIP :ANY PARITY ERRORS?
6818 012254	001003	002066			BNE	NEMCNT	:YES - SKIP :ANY HOLES? :NONE - SKIP
6819 012260 6820 012262	001406 052760	000001	002650	3\$:	BEQ	4\$ #BITO, CONFIG(RO)	; NONE - SKIP ; SET ERROR BIT IN THIS BANK
4931 N1337N	053760 000137	002104 011670	002650		SET BIS JMP	CONFGERROR CPUBIT, CONFIG(RO) TAG9\$	SET ERROR BIT IN THIS BANK FORCE PRINTING OF CONFIGURATION TABLE SET ACCESSED BIT
6824 6825					;TEST A	PROTECTED BANK	
6822 012276 6823 012304 6824 6825 6826 012310 6827 012312 6828 012320 6829 012324	012737	000001 060000	002076	TSTBANI	K:PUSH MOV MOV	R1 #1,NONEM #FIRST,RO	SET NON-EXISTANT MEMORY TO COUNT
6828 012320 6829 012324 6830 012330	012700 012701 104424	020000			MOV	#20000,R1	TURN CACHE OFF
					CACHOFF TESTARE	A	ENTER TEST MODE
6830 012330 6831 012332 6832 012340 6833 012342 6834 012344 6835 012346 6836 012350	005720 077102			4\$:	TST S08	(R0)+ R1,4\$	FAITED MEDALEL MODE
6834 012344 6835 012346	104417 104423 012737				CACHON		;ENTER KERNEL MODE ;TURN CACHE ON
003/ U1/330	012737	000002	002076		MOV POP	#2,NONEM R1	RESET NON-EXISTANT MEMORY TO EXIT TEST LOOP
6838 012360 6839 012366 6840 012374	052761	000001	002650		IF PARC BIS SET	NT NE #0 #BITO, CONFIG(R1) CONFGERROR	;ERROR BANK

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-32 09:41 PAGE 169-2 T3 TEST BANKS 1-200 (OCTAL) FOR ZEROS & ONES

6841 012402					END ; OF	IF PARCNT NT EQ #0	
6843 012410	053761	002104	002650		BIS END ; OF	CPUBIT, CONFIG(R1) IF NEMCNT	:ACCESSED BANK
6845 012416	000137	011670		MTST3:	JMP	TAG9\$ R2,(R0)+	;v177640
6847 012424 6848 012426	077102			111313.	SOB	R1,MTST3	V177642 V177644
6849 012430	012401			2\$:	MOV CMP	(R4)+,R1	V177646 V177650
6850 012432 6851 012434	020102	002072			BEQ	R1,R2 3\$ PATERR	: V177652 : V177654
6852 012436 6853 012442	077306	002072		3\$:	SOB RETURN	R3,2\$	V177660 V177662
6854 012444	000207				KE I OKIA		, 1111002

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 170 TEST BANKS 1-200 (OCTAL) FOR ZEROS & ONES

6856	012446				SUBAAI: SUBTST < <find inhibit="" mo<="" shadow="" th=""><th>************</th></find>	************
6857 6858 6859					* THIS SECTION LOOKS FOR INTERLEAVED FOR THE SHADOW INHIBIT MODE POINTS	MS11-M MEMORIES AND FIGURES OUT
6860 6861 6862	012446 012452 012456 012462	005037 004737 013700	002100 047020 002102		CLR BANK SHADL1: CALL EXBANK MOV BANKINDEX.RO IF ACFLAG IS TRUE AND INTFLAG	RESET BANK TO ZERO SET BANK PARAMETERS
6864	012476 012504 012510 012516	062700 062737	000020 000010	002100	ADD #20.RO ADD #10.BANK	:POINT TO BANKINDEX + 4 :POINT TO BANK + 8
6868 6869 6870	012520 012524 012532	062702 062737	000040 000020	002100	ELSE ADD #40.R2 ADD #20.BANK END; OF IF INT64K	:POINT TO BANKINDEX + 8 :POINT TO BANK + 16
6871	012532 012540	052760	000200	002650	BIS #BIT7, CONFIG(RO)	MAKE NEW BANK PROGRAM SPACE
6873	012542	005237	002100		INC BANK END: OF IF ACFLAG	GO TO NEXT BANK
6875 6876	012546 012554	023737 002336	002552	002100	CMP LASTBANK, BANK BGE SHADL1	HAVE WE DONE ALL THE BANKS?

```
NEWTST <<ECC INHIBIT MODE POINTER TEST>>
6879 012556
                                                               ************************
                                                              :*TEST 4
                                                                                      ECC INHIBIT MODE POINTER TEST
                                                               *********************
                                                               TST4: SCOPE
        012556 000004
                                                                           THE MS11-M OR MF11S-K INHIBIT ECC DISABLE AND DIAGNOSTIC CHECK MODE ON THE BOTTOM FIRST OR SECOND 16K WORDS CONTROLLED BY A CSR. THIS IS CONSIDERED TO BE A PROTECTED BANK BY THE PROGRAM. IT MAY BE QUITE COMPLEX TO DETERMINE ON A GIVEN SYSTEM CONFIGURATION WHICH
                                                                             BANKS CAN BE PROTECTED;
                                                                            THIS ROUTINE ATTEMPS TO CREATE A DOUBLE BIT ERROR IN ADDRESS 0 & 2 OF EVERY ECC BANK. ECC HARDWARE WILL PREVENT THIS FROM HAPPENING IN PROTECTED BANKS WHICH SHOULD ALWAYS INCLUDE BANK ZERO - WHERE
                                                                             THE PROGRAM IS.
6891
6892
6893
                                                                            ; WARNING: !!!!!!!!!!!
                                                                            IN CASE OF HARDWARE FAILURE IT IS COMMON THAT A DOUBLE BIT ERROR WILL BE CREATED ON THE KERNEL STACK & "CRASH" THE DIAGNOSTIC DURING THIS ROUTINE. YOUR ONLY CLUE IS THAT YOU CAN GET AS FAR AS
                                                                             THIS ROUTINE BUT NOT PAST IT!
6896
6897 012560
6898 012562
6899 012570
6900 012574
6901 012600
6902 012604
6903 012610
6904 012616
6905 012624
6906 012632
6907 012640
6908 012644
6909 012652
6910 012660
6912 012660
6913 012664
6914 012666
6915 012672
6916 012722
6917 012706
6918 012714
6919 012722
6920 012730
                      104424
012737 177777 002154
                                                                                                                                                :TURN CACHE OFF
                                                                             CACHOFF
                                                                                         #-1,OLDCSR
                                                                            MOV
                                                                            FOR BANK := #0 TO LASTBANK
                     012701
004737
013700
                                   060000
047020
                                                                                                                                                :SET UP VIRT ADDR POINTER
                                                                               MOV #FIRST_R1
                                                                               CALL EXBANK
                                                                                MOV BANKINDEX, RO
                                   002102
                                                                              IF ACFLAG IS TRUE

IF MKFLAG IS TRUE

IF SKIPMK IS FALSE

IF INTFLAG IS TRUE

MOV #40000,R3
                                                                                                                                                SET INDEX COUNTER
                      012703
012737
                                   040000
                                                                                                                                                :MAP AS INTERLEAVED BANK
                                                 002236
                                                                                                        #1.SPLTCSR
                                   000001
                                                                                          MOV
                                                                                      ELSE
                                                                                                                                                :SET INDEX COUNTER
                      012703 000002
                                                                                         MOV
                                                                                      END: OF IF INTFLAG
                      116002
006302
042702
010237
                                                                                                       CONFIG+1(RO),R2
                                   002651
                                                                                      ASL
                                                                                                        #*C36,R2
                                   177741
                                                                                                        R2, CSRNO
                                   002150
                                                                                      MOV
                                                                                      IF CSRNO NE OLDCSR
                                                                                                       CSRNO, OLDCSR
                      013737
                                   002150 002154
                                                                                         IF PFLAG IS FALSE
BIS #BIT6, CONFIG(RO)
END: OF IF PFLAG
CALL IMPTEST
                      052760 000100
                                                 002650
 6921 012730
                      004737 013064
```

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 173

6925 012746 07 6926 012752 04 6927 012756 01 6928 012762 06 6929 012766 00	6002 2227 2702 0237 2701 04737 05037	002651 177775 177741 002150 000002 013064 002236	IF INTFLAG IS TRUE MOVB CONFIG+1(RO),R ASH #-3,R2 BIC #^C36,R2 MOV R2,CSRNO ADD #2,R1 CALL IMPTEST CLR SPLTCSR END; OF IF INTFLAG END; OF IF CSRNO END; OF IF SKIPMK END; OF IF MKFLAG END; OF IF ACFLAG END; OF FOR BANK	FIX POINTER FOR AT ASSERTED HALF
6937 013012	5037	002100	CLR BANK	:MAP TEST SPACE TO BANK 0
6939 013020 00	15051	002100	IF #SWO SET. IN aSWR OR ACTFLAG	IS TRUE
6940 013050 10	4506		ENASBE	TRAP ON SINGLE BIT ERRORS
6941 013052			ELSE	:TRAP ON DOUBLE BIT ERRORS (NORMAL)
6940 013050 10 6941 013052 6942 013054 10 6943 013056)4472		ECCINIT END; OF IF #SWO	THAT OH DOODLE BIT ERRORS (HORPAL)
6944 013056 10)4423		CACHON	TURN THE CACHE BACK ON
	0137	013372	JMP SUBAAR	JUMP OVER THE SUBROUTINE

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 174
          ECC INHIBIT MODE POINTER TEST
   6947 013064
6948 013066
6949 013102
6950 013104
6951 013112
6952 013120
6953 013122
6954 013124
6955 013126
6956 013130
6957 013132
6958 013134
6959 013136
6960 013140
6961 013142
                                                       IMPTEST: CLR
                     005004
                                                                                                                       :MAP SUPERVISOR SPACE (TEST AREA) TO BANK
                                                                  MAP BANK
                     005005
012737
                                                                  CLR
                                020000 002146
                                                                              WBIT13,CSR
                                                                  NOV
                                                                   TESTAREA
                                                                                                                          ENTER TEST MODE
                                                                                                                           SAVE TEST LOCATION
                                                                  PUSH
                                                                                                               :INDEX TO NEXT LOCATION
                                                                             R3,R1
(R1)
                                                                   ADD
                      060301
                                                                                                               SAVE TEST LOCATION
DISABLE ECC & WRITE CHECKBITS FOR 1 CSR
WRITE CHECKBITS (ALL ZEROS)
                                                                   PUSH
                                                                   CHK1DIS
                      104505
                                                                             R4.(R1)
R3.R1
                                                                   MOV
                                                                   SUB
                      160301
                                                                  MOV
                                                                             R4, (R1)
                      010411
                                                                  CLR1CSR
                                                                                                                          :CLEAR CSR
                      104503
                                                                                                                          READ CHECKBITS INTO REAL CSR
                                                                   TST
                                                                                                                          : WAS THERE A DOUBLE BIT ERROR
                      104501
                                                                   WAS1DBE
                                                                  :THIS MAKES SURE THAT SBE'S DON'T LOOK LIKE PROTECTED AREAS
    6965 013144
6966 013146
6967 013154
6968 013156
6969 013162
6970 013164
6971 013170
                                                                  ON.NOERROR :1
MOV #BIT13,CSR
                      012737
104505
013711
                                 020000
                                            002146
                                                                                                                        :DISABLE ECC & WRITE CHECKBITS FOR 1 CSR
                                                                      CHK1DIS
                                                                             ONES, (R1)
R3,R1
                                                                      MOV
                                 002600
                      060301
013711
                                                                      ADD
                                                                             ONES, (R1)
R3,R1
                                                                      MOV
                                 002600
                      160301
                                                                      SUB
                                                                                                                          :CLEAR CSR
    6972 013172
6973 013174
                                                                      CLR1CSR
                      104503
                                                                     TST (R1)
                      005711
                                                                      WAS1DBE
                                                                                                                          :WAS THERE A DOUBLE BIT ERROR
    6974 013176
                      104501
                                                                     ON.NOERROR :2

IF #BIT9 SET.IN CONFIG+2(RO) ; IS THIS A MS11-P

; ENABLE CHECK/SYNDROME BIT REGISTER

; ENABLE CHECK/SYNDROME BIT REGISTER
    6975 013200
6976 013202
6977 013212
                      104513
012737
                                                                            MOV #23140, CSR
                                                                                                               ; WRITE DBE'S IN CSR
                                 023140 002146
    6978 013214
                                                                                                               : OR A MS11-M
                                                                        ELSE
    6979 013222
                                                                            MOV #27400, CSR
                                                                                                                :WRITE DBE'S IN CSR
    6980 013224
                      012737 027400 002146
                                                                        END
    6981
                                                                                                               :DISABLE ECC & WRITE CHECKBITS FOR 1 CSR
                                                                        CHK1DIS
                      104505
                      010411
060301
                                                                        MOV R4, (R1)
                                                                        ADD R3,R1
                                                                                                               ADD INDEX TO GET TO SECOND WORD
                                                                        MOV R4.(R1)
SUB R3.R1
CLR1CSR
                      010411
                                                                                                               SUBTRACT INDEX TO FIRST WORD
                       160301
104503
                                                                                                                          :CLEAR CSR
                                                                        TST (R1)
                                                                                                                          :WAS THERE A DOUBLE BIT ERROR
                                                                        WAS1DBE
    6990
6991
6992
6993
                                                                        ON.NOERROR :3
IF #BIT9 SET.IN CONFIG+2(RO)
                                                                                                               ; ENABLE CHECK/SYNDROME BIT REGISTER
                      104513
012737
                                                                               CBREG
                                                                               MOV #23604, CSR
                                  023604
                                             002146
    6994
6995
6996
6997
6998
6999
                                                                                                                :IS IT A MS11-P
                                                                               MOV #74000,CSR
                                                                                                                :WRITE DBE'S IN CSR
                      012737
                                  074000
                                            002146
                                                                            END
           013304
013306
013310
013312
013314
                                                                                                                :DISABLE ECC & WRITE CHECKBITS FOR 1 CSR
                                                                            CHK1DIS
                      010411
060301
010411
104503
160301
                                                                                         R4,(R1)
R3,R1
                                                                           MOV
                                                                            ADD
                                                                                                               :INDEX TO SECOND WORD
                                                                            MOV
                                                                                         R4, (R1)
     7000
                                                                            CLR1CSR
                                                                                                                           :CLEAR CSR
     7001
                                                                                                               GO BACK TO FIRST WORD
                                                                            SUB
                                                                            TST
```

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 174-1
```

14		ECC INN	TOT! HON	POINIER IESI							
	7005	013322 013324 013324 013324	104501			END : OF	S1DBE; OF ON.NOERROR OF ON.NOERROR; 2 ON.NOERROR; 1	;3	:	WAS THERE A DOU	BLE BIT ERROR
	7007 7008 7009 7010	013324 013326	005205			ON. ERROR	R5		:	IDENTIFY AS BAD	BANK
	7011 7012 7013 7014 7015 7016 7017 7018 7019 7020 7021 7022	013330 013330 013332 013334 013340 013342 013344 013346 013352 013356 013360 013362	104471 010411 060301 010411 104503 005705 001405 050560 105260 104036	002650 002652	1\$:	ECCIDIS MOV ADD MOV CLRICSR TST BEQ BIS INCB ERROR POP SUB	ON.ERROR R4,(R1) R3,R1 R4,(R1) R5 1\$ R5,CONFIG(R0) CONFIG+2(R0) +36 (R1) R3,R1		CLEAR OU INDEX TO CLEAR OU	DISABLE ERROR C T DOUBLE BIT ER SECOND WORD T DOUBLE BIT ER TEST LOCATION (TO FIRST WORD TEST LOCATION (ROR!
	7023 7024 7025	013366 013370	104417 000207			POP KERNEL RETURN	(R1)	•	RESTURE	TEST EUCHTION	131 WORD
	7026	013372			SUBAAR:	SET	STOPOK	;PROGRAM	CAN NOW	BE HALTED	

```
ECC INHIBIT MODE POINTER TEST
                                                                                            SUBTST <<LEGAL CONFIGURATION CHECK>>
7036 013400
                                                                            :*SUBTEST
                                                                                                            LEGAL CONFIGURATION CHECK
                                                                            ******************************
7037 013400
7038 013404
7039 013410
7040 013412
7041 013414
7042 013420
7043 013424
7044
7045 013430
7046 013436
7047 013442
7048 013446
7049 013450
7050 013454
                        012700 000020
012701 002456
005021
077002
                                                                                                            #16. RO
#CSRINFO,R1
                                                                                            MOV
                                                                           15:
                                                                                                             (R1) +
                                                                                            CLR
                                                                                            SOB
                                                                                                            RO,1$
                                                                                               FOR BANK := #0 TO LASTBANK
                                                                                                                            EXBANK
                          004737 047020
013700 002102
                                                                                                    CALL
                                                                                                    MOV BANKINDEX RO
                                                                                                    IF ACFLAG IS TRUE
                                                                                                       MOVB CONFIG+1(RO),R3
BIC #^C17,R3
ASL R3
                          116003 002651
042703 177760
006303
005263 002456
                                                                                                        ASL
                                                                                                       INC CSRINFO(R3)

IF MKFLAG IS TRUE

; MAKE SURE THAT EACH BANK HAS NO MORE THAN 2 CSRS

BEGIN LEGALCSR

IF INTFLAG IS TRUE
7050 013454

7051

7052 013462

7053 013462

7054 013470

7055 013474

7056 013476

7057 013502

7058 013506

7059 013512

7060 013516

7061 013524

7062 013532

7063 013534

7064 013534
                         116003 002651
010304
042703 177760
072427 177774
042704 177760
                                                                                                                                             CONFIG+1(RO),R3
                                                                                                                     MOVB
                                                                                                                    MOV R3,R4
BIC #^C17,R3
ASH #-4,R4
BIC #^C17,R4
IF R3 EQ R4
                                                                                                                   BIC
                           042760 014000
042760 170000
                                                                                                                                             #BIT11!BIT12, CONFIG+2(RO)
                                                                                                                                             #170000, CONFIG(RO)
                                                                                                                    LEAVE LEGALCSR
                                                                                                                 ELSE
                                                                                                                LEAVE LEGAL
END; OF IF INTFLAG
T CONFGERROR
 7065 013536
                                                                                                                                             LEGALCSR
 7065 013536
7066 013540
7067 013540
7068 013546
7069 013546
7070 013546
7071 013546
7072 013562
7073 013566
7074 013570
7075 013572
7076 013574
7077 013600
7078 013606
7079 013610
                                                                                                                             LEGALCSR
                                                                                                END : OF IF MKFLAG
END : OF IF ACFLAG
END: OF FOR BANK
                                                                                                             R5,R0
R0
                                                                                                                                                                              ; SAVE CONTENTS OF R5, RO
                                                                                                                                                                              :CLEAR REGISTERS
                                                                                             CLR
                           005000
                           005000
005001
005005
005037
022761
002043
022761
                                                                                             CLR
                                                                                             CLR
CLR
CMP
BGE
                                                                                                                                                                              CLEAR ERROR INDICATOR SIS CURRENT CSR <= 40
                                           014002
                                                                                                             MBERR
                                                                                                             #40,CSRINFO(R1)
                                            000040
                                                            002456 2$:
                                                                                                                                                                              BRANCH IF SO
IS CURRENT CSR < 10
BRANCH IF SO
CALL ERROR ROUTINE
TRY NEXT CSR
MOVE LOW WORD TO R5
DOES MEMORY EXIST HERE?
          013606
013610
013616
013620
013624
013626
013632
013636
013640
013644
013650
013652
                                                                                             CMP
                                                                                                             #10,CSRINFO(R1)
 7079
                                            000010 002456
 7080
7081
7082
7083
7084
7085
                                                                                                              3$
                                                                                             BGE
                           002003
004737
000434
016005
032705
001415
042705
072527
020501
001007
                                                                                                              ILLCSR
                                                                                             CALL
                                            014124
                                                                                             BR
                                                                                                             CONFIG(RO),R5
                                            002650
                                                                                             MOV
                                                                            3$:
                                                                                             BIT
                                                                                                              #BIT1,R5
                                                                                                                                                                               BRANCH IF NOT
                                                                                                             #^C7400,R5
#-7,R5
R5,R1
                                                                                                                                                                              : ISOLATE CSR NUMBER IN : REGISTER 5
  7086
7087
7088
                                                                                                                                                                              : IS IT THE CURRENT CSR?
                                                                                                                                                                               TRY NEXT WORD OF CONFIG IF NOT
```

CZMSPAO MS11-L/M/P MEMORY DIAG. LEGAL CONFIGURATION CHECK	MACRO M1113 26-	APR-82 09:41 PAGE 182-1		
7090 013654 032760 010000 7091 013662 001003 7092 013664 012737 000001 7093 013672 062700 000004 7094 013676 022700 000340 7095 013702 001351 7096 013704 005737 014002 7098 013712 004737 014124 7099 013716 005000 7100 013720 005037 014002 7101 013724 062701 000002 7102 013730 022701 000040 7103 013734 001321 7104 013736 7105 013742 005037 014002 7106 013746 012700 000734 7107 013752 032760 000002 7108 013760 001003 7109 013762 162700 000002 7110 013766 000771 7111 013770 006200 7112 013772 006200 7113 013774 010037 002552 7114 014000 000402 7115 014002 000000 7116 014004 000000 7117 014006 005000 7118 014014 032760 000002 7120 014024 032760 010000 7121 014024 032760 010000 7122 014032 001005 7123 014034 005237 014004	014002	BIT #BIT12,CONFIG+2(RO) BNE 4\$ MOV #1,MBERR ADD #4,RO CMP #340,RO BNE 3\$ TST MBERR BEQ 5\$ CALL ILLCSR	;BRANCH IF SO	
7099 013716 005000 7100 013720 005037 014002 7101 013724 062701 000002 7102 013730 022701 000040 7103 013734 001321 7104 013736 7105 013742 005037 014002 7106 013746 012700 000734		MOV #1,MBERR ADD #4,RO CMP #340,RO BNE 3\$ TST MBERR BEQ 5\$ CALL ILLCSR CLR RO CLR MBERR ADD #2,R1 CMP #40,R1 BNE 2\$ POP RO,R5 CLR MBERR MOV #734,RO BIT #BIT1,CONFIG(RO)	; REINITIALIZE CONFIG COUNTER ; CLEAR ERROR INDICATOR ; UPDATE CSR COUNTER ; ALL CSR'S DONE? ; BRANCH IF NOT ; RESTORE REGISTERS ; RESET ERROR INDICATOR ; INDEX TO TOP OF CONFIG TABLE ; R-C	
7106 013746 012700 000734 7107 013752 032760 000002 7108 013760 001003 7109 013762 162700 000004 7110 013766 000771 7111 013770 006200	7\$:	BIT #BIT1, CONFIG(RO) BNE 7\$ SUB #4,RO BR 6\$ ASR RO ASR RO MOV RO, LASTBANK	:MEMORY PRESENT? ;R-C :BRANCH IF SO ;R-C ;TRY NEXT LOWER ENTRY IN CONFIG TABLE :R-C :R-C	;R-C
7105 013742 005037 014002 7106 013746 012700 000734 7107 013752 032760 000002 7108 013760 001003 7109 013762 162700 000004 7110 013766 000771 7111 013770 006200 7112 013772 006200 7113 013774 010037 002552 7114 014000 000402 7115 014002 000000 7116 014004 000000 7117 014006 005000 7118 014010 005037 014004 7119 014014 032760 000002 7120 014022 001431 7121 014024 032760 010000 7122 014032 001005 7123 014034 005237 014004	MBERR: PHEBE: SKUJ:	MOV ROLASTBANK BR SKUJ .WORD 0 .WORD 0 CLR RO CLR PHEBE	;DIVIDE INDEX BY 4 TO GET BANK #;R-C ;STORE IN LASTBANK ;R-C ;SAVE SPACE FOR ERROR INDICATOR ;SAVE SPACE FOR ODD BOUNDARY INTERLEAVE ;CLEAR CONFIG COUNTER ;CLEAR COUNTER	ED INDICATOR
7118 014010 005037 014004 7119 014014 032760 000002 7120 014022 001431 7121 014024 032760 010000 7122 014032 001005 7123 014034 005237 014004 7124 014040 062700 000004	002652	BIT #BIT1, CONFIG(RO) BEQ 3\$ BIT #BIT12, CONFIG+2(RO) BNE 2\$ INC PHEBE ADD #4,RO	;BRANCH IF NOT ;IS IT INTERLEAVED? ;BRANCH IF SO	
7124 014040 062700 000004 7125 014044 000763 7126 014046 023727 014004 7127 014054 001417 7128 014056 023727 014004 7129 014064 001413 7130 014066 023727 014004 7131 014074 001407 7132 014076 023727 014004 7133 014104 001403 7134 014106 005037 014004 7135 014112 000403 7136 014114 012737 000001 7137 014122 000421 7138 014124 010102 7139 014126 006202 7140 014130 022702 000001 7141 014134 100002 7142 014136 062702 000007 7143 014142 062702 0000060 7144 014146 110237 100014 7145 014152 7146 014156	000010 2\$: 000030 000050	BR 1\$ CMP PHEBE .#10 BEQ 4\$ CMP PHEBE .#30 BEQ 4\$ CMP PHEBE .#50 BEQ 4\$	INCREMENT COUNTER INCREMENT CONFIG COUNTER TRY NEXT BANK IS THE COUNTER EQUAL TO ONE OF THE SPECIAL VALUES. IF IT IS BRANCH TO 4\$	
7132 014076 023727 014004 7133 014104 001403 7134 014106 005037 014004 7135 014112 000403 7136 014114 012737 000001 7137 014122 000421	000070 3\$: 014004 4\$: 5\$:	CMP PHEBE,#70 BEQ 4\$ CLR PHEBE	;CLEAR INDICATOR ;SET INDICATOR RANCH TO NEXT SUBTEST 2 HAS CSR NUMBER AKE ACCEPTABLE FOR PRINTING	
7138 014124 010102 7139 014126 006202 7140 014130 022702 000012 7141 014134 100002 7142 014136 062702 000007 7143 014142 062702 000060 7144 014146 110237 100014		CMP #10.,R2 BPL 1\$ ADD #7,R2 ADD #60,R2		
7144 014146 110237 100014 7145 014152 7146 014156		MOVB R2.MSGA122 ;PI TYPE MSG122 SET CONFGERROR	JT NUMBER INTO ERROR MESSAGE	

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 182-2 LEGAL CONFIGURATION CHECK

7147 014164 000207

RETURN

```
SUBAAP: SUBTST <<PRINT CONFIGURATION DETAILS>>
7150 014166
                                                          ****************
                                                           **SUBTEST
                                                                                   PRINT CONFIGURATION DETAILS
7151 014166
7152 014202
7153 014206
7154 014210
7155 014212
7156 014214
7157 014224
7158 014234
7159 014244
                                                                       CLEAR
                                                                                  LSIZE, MSIZE, PSIZE
                    013702 002552
006302
006302
                                                                                   LASTBANK, R2
                                                                       MOV
                                                                       ASL
                                                                       ASL
                                                                       FOR R1 := #0 TO R2 BY #4
                                                                        IF CPUBIT SET. IN CONFIG(R1)
IF #BIT8 SET. IN CONFIG+2(R1)
IF #BIT9 SET. IN CONFIG+2(R1)
                                                                                     LET PSIZE := PSIZE + #1
7159 014244
7160 014250
7161 014252
7162 014256
7163 014256
7164 014260
7165 014264
7166 014264
7167 014264
                                                                                     LET MSIZE := MSIZE + #1
                                                                              END; IF BIT9
                                                                           ELSE
                                                                        LET LSIZE := LSIZE + #1
END; IF BIT8
END; OF IF CPUBIT
                                                                       END OF FOR ALL BANKS IN TABLE
7168
7169 014274
7170 014300
7171 014302
7172 014306
7173 014312
                                 002446
                    005037
                                                                       FOR R1 := #0 TO #10 BY #2
                                 002370
002370
002370
002370
                    006361
006361
006361
006361
                                                                          ASL
                                                                                   BSIZE (R1)
                                                                                   BSIZE (R1)
                                                                          ASL
                                                                                   BSIZE(R1)
BSIZE(R1)
                                                                          ASL
                                                                                                                         ;BSIZE(R1) := BSIZE(R1) * 16.
7174 014316
7175 014322
7176 014330
                                                                          ASL
                                                                                                                         :1 <- I + BSIZE(R1)
                                 002370
                                             002446
                                                                          ADD
                                                                                   BSIZE(R1),I
                                                                       END; FOR R1
7176
7177
                                                                       FOR R1 := #0 TO #200 BY #4
                                                                            IF CPUBIT SET. IN CONFIG(R1)
                                                                               LET UNITOP := UNITOP + #1
                                                                       END; OF IF CPUBIT
       014360
014372
014376
7181
                    006337
006337
006337
006337
                                 002412
002412
002412
                                                                       ASL
                                                                                    UNITOP
                                                                                    UNITOP
                                                                       ASL
       014402
                                                                                    UNITOP
                                                                       ASL
                                                                       ASL UNITOP := UNITOP * 16.

IF I LT UNITOP THEN LET I := UNITOP
       014406
7186 014412
7187 014430
                                                                       TYPE
                                                                                    SCRLF
                                                                                    LSIZE
                    005737
001405
 7188 014434
                                                          2$:
                                                                       TST
                                 002374
7189 014440
7190 014442
7191 014450
                                                                       BEQ
                                                                       TYPDEC
                                                                                   LSIZE
                                                                       TYPE
                                                                                    MSG112
7191 014450
7192 014454
7193 014460
7194 014462
7195 014470
7196 014474
7197 014500
7198 014502
7199 014510
                    005737
001405
                                                                                    MSIZE
                                                           3$:
                                 002376
                                                                       TST
                                                                       BEQ
                                                                       TYPDEC
                                                                                   MSIZE
                                                                                    MSG113
                                                                       TYPE
                                                                                    PSIZE
5$
                     005737
001405
                                                          45:
                                                                        TST
                                 002400
                                                                       BEQ
                                                                        TYPDEC
                                                                                    MSG114
                                                                       TYPE
 7199 014510
                                                           5$:
                                                                                   MSG070
                                                                        TYPE
                                                                        IF #SW6 OFF.IN @SWR
                                                                          CALL PCONFIG
                     004737 041352
```

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 184-1 PRINT CONFIGURATION DETAILS

7204 014542

END; OF IF #SW6

```
SUBTST <<CHECK APT SIZING>>
7207 014542
                                                         ****************
                                                         : *SUBTEST
                                                                                 CHECK APT SIZING
                                                         ***************
                                                                     IF APTFLAG IS TRUE AND APTSIZE IS TRUE
7208 014542
7209 014556
7210 014562
7211 014566
7212 014570
7213 014576
7214 014600
7215 014614
7216 014612
7217 014614
7218 014616
7219 014620
7220 014620
7221 014622
7222 014624
7223 014626
7224 014630
7225 014632
7226 014634
7227 014640
7238 014660
7231 014660
7232 014670
7233 014674
7233 014674
7234 014674
7235 014700
7237 014710
7238 014742
7239 014742
7239 014742
7230 014742
                                                                        CLR TEMP
                   005037 002430
012700 065670
                                                                       MOV #$MAMS1.R0

FOR R2 := #0 TO #4

IFB 1(R0) NE #0

MOVB (R0)
                                                                                             (RO),R1
#177400,R1
                    111001
042701 177400
                                                                              IF 2(RO) LT #0
                    000261
                                                                               ELSE
                                                                                 CLC
                    000241
                                                                              END : OF IF 2(RO)
                                                                               ROL
                    006101
                                                                                                                      ; TO COMPENSATE FOR 4 BANKS BEING (0-3)
                    005201
006301
                                                                               INC
                                                                                              R1
                                                                               ASL
                                                                                              R1
                    006301
                                                                               ASL
                    006301
                                                                               ASL
                                                                                              R1
                    006301
163701
010137
                                                                               ASL
                                002430
002430
                                                                                              TEMP,R1
                                                                               SUB
                                                                                              R1, TEMP
                                                                               MOV
                                                                               IFB 1(R0) EQ #3
                                                                              ADD R1, APTPAR
END : OF IFB 1 (RO)
IFB 1 (RO) EQ #4
                    060137 002416
                                                                              END : OF IFB 1 (RO)
                    060137 002420
                                                                               ADD
                    062700 000004
                                                                            END : OF IFB 1(RO)
                                                                        END : OF FOR R2
                                                                        IF APTPAR NE LSIZE OR APTECC NE MSIZE OR APTECC NE PSIZE
                                                                            ERROR
                    104046
                                                                                              +46
                                                                     END : OF IF APTPAR END : OF IF APTFLAG
```

7242 014742	LOOP: NEWTST	< <diagnostic dispatch="" mode="" rol<="" th=""><th>ITINE>></th></diagnostic>	ITINE>>
	*TEST 5	DIAGNOSTIC MODE DISPATCH ROUT	NE
7244 014750 017700 165646 7245 014754 042700 177761 7246 014760 004770 014770 7247 014764 000137 015010 7248 014770 015452 7249 014772 015560 7250 014774 015666 7251 014776 016016 7252 015000 016146 7253 015002 016276 7254 015004 016450 7255 015006 016600	TST5: SCOPE CLR MOV BIC CALL JMP DISPTBL:BAFPAF BAFPAR BAWPAF BAWPAF BAWPAR PAFBAF PAFBAW PARBAW	CONTFLAG aSWR,RO ;GET SWITCHES "C16,RO ;MASK TO ONLY aDISPTBL(RO) ;DISPATCH TO I MEMDONE ;GO TO NEXT TI ;MODE 0;BANKS FORWARD, PATTERI ;MODE 1;BANKS FORWARD, PATTERI ;MODE 2;BANKS WORST FIRST, PA ;MODE 3;BANKS WORST FIRST, PA ;MODE 3;BANKS WORST FIRST, PA ;MODE 4;PATTERNS FORWARD, BANI ;MODE 5;PATTERNS FORWARD, BANI ;MODE 6;PATTERNS REVERSE, BANI ;MODE 7;PATTERNS REVERSE, BANI	MODE BITS ROUTINE THROUGH NEXT TABLE ST WS FORWARD WS REVERSE TERNS FORWARD TERNS REVERSE (S FORWARD (S WORST FIRST (S FORWARD (S WORST FIRST
7257 015010 004737 015352			CHECK BACKGROUND PATTERN
7258 7259 015014	NEWTST<	<unique bank="" test="">></unique>	
	TEST 6	UNIQUE BANK TEST	******
015014 000004 7260 7261 7262 015016 7263 015024 7264 015040 004737 024102 7265 015044 7266 015052 005037 002106	TST6: SCOPE		IQUE DATA CH BANK (EXCEPT WHERE THE PROGRAM IS)
7264 015040 004737 024102 7265 015044	CALL	MT0027 HEADER	
7266 015052 005037 002106	CLR	MUT	
7267 015056 7268 015056 004737 015352 7272	END ; OF	DOBACK ; RESTORE BACK	ROUND PATTERN
7273 015062	FLUSH: SUBTST :***********************************	< <flush dbe's="" out="">> FLUSH OUT DBE'S</flush>	*******
7274 015062 004737 024566	CALL	MT0030	

END UF	PASS KUU	IINE				
7277						.SBTTL END OF PASS ROUTINE
7278	015066 015072 015076 015102 015106 015112 015126 015126 015134 015134 015136 015176 015176 015210 015210				******	ENT THE PASS NUMBER (\$PASS)
7280					:*INDIC	THE PASS NUMBER (\$PASS) TE END-OF-PROGRAM AFTER EACH PASSES THRU THE PROGRAM END PASS #XXXXXX'' (WHERE XXXXX IS A DECIMAL NUMBER) RES A MONITOR GO TO IT RE ISN'T JUMP TO LOOP CLR FSINFLAG MOV #CONFIG+2,RO ; MOVE 2ND WORD OF CONFIG TO RO BIC #BIT13,(RO) ; CLEAR BACKGROUND VALID BIT ADD #4 RO :INCREMENT TO NEXT BANK
7281					:*TYPE	END PASS "XXXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
7282					*IF THE	RES A MUNITUR GU TU IT
7284	015066	005037	002436		SEOP:	CLR FSINFLAG
7285	015072	012700	002652			MOV #CONFIG+2, RO : MOVE 2ND WORD OF CONFIG TO RO
7286	015076	042710	002436 002652 020000 000004		15:	BIC #BIT13,(RO) ; CLEAR BACKGROUND VALID BIT ADD #4,RO ; INCREMENT TO NEXT BANK
7288	015106	020027	003620			CMP R0.#3620 : DONE?
7289	015112	005037 012700 042710 062700 020027 003771 013737		00201/		BLE 1\$;NO - BRANCH
7290	015114	005237	002614 065646	002014		MOV SERTTL, LASTERROR ;: INCREMENT THE PASS NUMBER
7292	015126	005237 042737	100000	065646		BIC #100000.\$PASS ::DON'T ALLOW A NEG. NUMBER
7293	015134					TYPE MSG077 :: TYPE "END PASS #"
729	015166					IF #SW11 SET.IN ASWR OR QVFLAG IS TRUE OR \$PASS EQ #1
7296	015172	005037	002342			CLR QVFLAG
7297	015176					END : OF IF SW11 TYPDEC \$PASS
7299	015204	013700	000042			MOV 42 ROGET MONITOR ADDRESS
7300	015210	001456	002000		67AD/2.	BEQ \$DOAGAIN ::BRANCH IF NO MONITOR CMP #STACK.RO :ARE WE UNDER RT11
730	015212	022700	002000		\$ZAP42:	CMP #STACK.RO : ARE WE UNDER RT11 BEQ \$DOAGAIN : YES - BRANCH
7303	3					; WE ARE UNDER (HEAVEN HELP US) XXDP!
7304	015220	004737	047664			PUSH RO CALL SHUTUP
730	015220 5 015222 6 015226 7 015230 8 015232 9 015234 0 015236 1 015240 2 015242	004737	047004			POP RO
7307	015230	000005			CENDAD.	RESET :: CLEAR THE WORLD
7300	015234	004710			SENDAD:	CALL (RO) ::GO TO MONITOR NOP ::SAVE ROOM
731	015236	000240				NOP ;;FOR
731	015240	000240			SDOAGN.	NOP :: ACT11 : UNDO SHUTUP STUFF
731	3				apondit.	RESTORE STACK ENERGIZE UNIBUS MAP & 22 BIT ADDRESSING ENERGIZE MEMORY MANAGEMENT PUT LOADERS BACK HOME MOV KSTACK, SP TST NO22BIT ;IS THIS AN 11/44 OR 11/24?
731. 731. 731.	4					ENERGIZE UNIBUS MAP & 22 BIT ADDRESSING
731	2					: PUT LOADERS BACK HOME
731	7 015242	013706 005737 001003 052737 104420 013700 012701 004737	002560 002450			MOV KSTACK, SP
731	8 015246	005737	002450			TST NO22BIT ; IS THIS AN 11/44 OR 11/24? BNE 1\$
732	0 015254	052737	000060	172516		RIC MRITSIRITA MMRZ
732	1 015262	104420	0025/2		1\$:	ENERGIZE ; TURN ON MEMORY MANAGEMENT
732	3 015270	013700	002562 000001 046410			MOV LOADHOME, RO ; DESTINATION BANK MOV #1, R1 ; SOURCE BANK
732	4 015274	004737	046410			CALL DANIEMOV
732	5 015300					IF APTFLAG IS TRUE
732	7 015316	012701	000050		APTHANG	: MOV #50,R1
732	67 015242 8 015246 9 015252 0 015254 1 015262 2 015264 3 015270 4 015274 5 015306 6 015306 7 015316 8 015322 9 015324 0 015336 2 015340 3 015344	012701 077001 062737 005537 077107 005237 000764	000004	0/5/50	2\$:	IF APTFLAG IS TRUE IF \$USWR EQ \$PASS MOV #50,R1 SOB R0,2\$ ADD #1,\$DEVCT
752	0 015332	005537	000001 065652	065650		ADC SUNIT
733	1 015336	077107	007076			ADC SUNIT SOB R1.2S INC SPASS
733	2 015340	005237	065646			INC SPASS BR APTHANG
(33	3 013344	000/04				DA AFTANO

SEQ 0182

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 189-1 END OF PASS ROUTINE

7334 015346 7335 015346 7336 015346 000137 014742

END : OF IF \$USWR END : OF IF APTFLAG \$DOAGAIN: JMP LOOP

; RETURN

...

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 191
```

7353 7354	.SBTTL MTEST MODES	
7355 015452	BAFPAF: SUBTST < <banks **subtest="" banks="" forward,="" forward:="" forward<="" patterns="" td=""><td>********</td></banks>	********
7356 015452 005037 002100	CLR BANK ;SET BANK TO 0	
7357 7358 015456 004737 047020 7359 015462 005737 002114 7360 015466 001412 7361 015470 005737 002122 7362 015474 001007 7363 015476 005037 002110	AA PUBANU PUBANU	THIS BANK?
7361 015470 005737 002122 7362 015474 001007	BEQ 4\$:NO - GO TO BATTST RRFLAG :RELOCATION REBNE 4\$:YES - GO TO B	THIS BANK? NK LOOP TERMINATION QUIRED? ANK LOOP TERMINATION 0 0
7363 015476 005037 002110 7364	CLR PATTERN ;SET PATTERN T	0 0
7365 015502 004737 016752 7366	2\$: CALL MTEST :GO TEST CORRE	
7367 015506 004737 047464	CALL INCPAT ; GO SEE IF THI RNF 28 :NO - LOOP ON	S IS THE LAST PATTERN THIS PATTERN
7369 7370 015514 005037 002220 7371 015520 004737 047510 7372 015524 002354 7373	; TERMINATION OF BANK LOOP 4\$: CLR CONTFLAG CALL INCBNK ; NEXT HIGHER B BGE 1\$; IF NOT DONE -	ANK LOOP ON THIS BANK
7375 015532 001401	; END OF LOOPS TST RLFLAG ; HAVE WE BEEN BEQ 5\$; NO - SKIP RETURN ; YES - RETURN	RELOCATED?
7376 015534 000207 7377 015536 004737 045172 7378 015542 7379	5\$: CALL RELOCATE ; MOVE & MAP PRON. ERROR THEN \$RETURN ;**NOTE** RECURSIVE CALL	OGRAM
7380 015546 004737 015452 7381 015552 004737 046056 7382 015556 000207	CALL BAFPAF ; CALL SELF CALL UNRELOCATE ; UNMOVE & UNMA RETURN	P PROGRAM

7385 0	15560			*SUBTE	*****	< <banks banks="" forward,="" pa<="" th=""><th>PATTERNS REVERSE **RECURSIVE**>> TTERNS REVERSE **RECURSIVE**</th></banks>	PATTERNS REVERSE **RECURSIVE**>> TTERNS REVERSE **RECURSIVE**
7386	015560	005037	002100	,	CLR	BANK F BANK LOOP	; SET BANK TO 0
7388 0 7389 0	015564 015570 015574 015576	004737 005737 001412 005737 001007 004737	047020 002114	15:	CALL	EXBANK ACFLAG	: EXAMINE BANK : CAN WE ACCESS THIS BANK?
7390 (015576	001412	002122		TST	RRFLAG 4\$:NO - GO TO BANK LOOP TERMINATION :RELOCATION REQUIRED? :YES - GO TO BANK LOOP TERMINATION
7393 (7394	015604	004737	047500		CALL	SETPAT F PATTERN LOOP	SET HIGH PATTERN FOR CORRECT MEMORY
7395 (7396	015610	004737	016752	2\$:	CALL	MTEST TION OF PATTERN	GO TEST CORRECT MEMORY
7397 (7398 (015614 015620	005337 100373	002110		DEC BPL	PATTERN 2\$ TION OF BANK LOO	; IS THIS THE LAST PATTERN? ;NO - LOOP ON THIS PATTERN
7399 7400 (7401 (7402 (7403	015626	005037 004737 002354	002220 047510	4\$:	CLR CALL BGE	CONTFLAG INCBNK	:NEXT HIGHER BANK :IF NOT DONE - LOOP ON THIS BANK
7404 (7405 (015634 015640 015642	005737 001401 000207 004737	002124		TST BEQ	RLFLAG	; HAVE WE BEEN RELOCATED? ; NO - SKIP ; YES - RETURN
7407 (015642 015644 015650	000207	045172	5\$:	ON. ERROR	RELOCATE THEN SRETURN * RECURSIVE CALL	;MOVE & MAP PROGRAM
7409 7410 7411 7412	015654 015660 015664	004737 004737 000207	015560 046056		CALL	BAFPAR	; CALL SELF ; UNMOVE & UNMAP PROGRAM

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 197 BANKS FORWARD, PATTERNS REVERSE **RECURSIVE**

7415 015666			*SUBTE	
7416 015666 7417	005037	002100	;*****	CLR BANK ;SET BANK TO 0 ;START OF BANK LOOP
7418 015672 7419 015676 7420 015702 7421 015704 7422 015710 7423 015712	004737 005737 001415 005737 001412 005737	047020 002114	1\$:	CALL EXBANK : EXAMINE BANK TST ACFLAG : CAN WE ACCESS THIS BANK?
7420 015702 7421 015704	001415 005737	002126		BEQ 48 ;NO - GO TO BANK LOOP TERMINATION TST BMFLAG ;IS THIS BAD MEMORY (WORST FIRST)?
7422 015710 7423 015712	001412	002122		BEQ 4\$:NO - GO TO BANK LOOP TERMINATION TST RRFLAG :RELOCATION REQUIRED? BNE 4\$:YES - GO TO BANK LOOP TERMINATION CLR PATTERN :SET PATTERN TO 0
7424 013710	001007	002110		BNE 4\$;YES - GO TO BANK LOOP TERMINATION ;SET PATTERN TO 0 ;START OF PATTERN LOOP
7427 015724	004737	016752	2\$:	CALL MTEST : GO TEST CORRECT MEMORY : TERMINATION OF PATTERN LOOP
7426 7427 015724 7428 7429 015730 7430 015734	004737 001373	047464		CALL INCPAT ; GO SEE IF THIS IS THE LAST PATTERN BNE 2\$; NO - LOOP ON THIS PATTERN ; TERMINATION OF BANK LOOP
7432 015736 7433 015742 7434 015746	005037 004737	002220 047510	4\$:	CALL INCONK :NEXT HIGHER BANK BGE 1\$:IF NOT DONE - LOOP ON THIS BANK
7435 7436 015750 7437 015754	005137 001003	002564		END OF LOOPS COM WORST :IS THIS AN EVEN NUMBERED PASS? BNE 5\$:YES - SKIP
7438 015756	004737	015666		:**NOTE** RECURSIVE CALL CALL BAWPAF ; CALL SELF RETURN
7438 7439 015756 7440 015766 7441 015766 7442 015776 7443 015776	000207 005737 001401	002124	5\$:	TST RLFLAG ; HAVE WE BEEN RELOCATED? BEQ 6\$; NO - SKIP
7443 015774 7444 015774 7445 016000 7446	000207	045172	6\$:	RETURN ;YES - RETURN CALL RELOCATE ;MOVE & MAP PROGRAM ON.ERROR THEN \$RETURN ;**NOTE** RECURSIVE CALL
7447 016004 7448 016016 7449 016014	004737	015666 046056		CALL BAWPAF ; CALL SELF CALL UNRELOCATE ; UNMOVE & UNMAP PROGRAM RETURN

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 199 BANKS WORST FIRST, PATTERNS FORWARD **RECURSIVE**

7452 016016	BAWPAR: SUBTST < <banks **recursive**="" first,="" patterns="" reverse="" worst="">> ;*SUBTEST BANKS WORST FIRST, PATTERNS REVERSE **RECURSIVE**</banks>	
7453 016016 005037 002100 7454	CLR BANK ;SET BANK TO 0 ;START OF BANK LOOP	
7455 016022 004737 047020 7456 016026 005737 002114 7457 016032 001415 7458 016034 005737 002126 7459 016040 001412 7460 016042 005737 002122	18: CALL EXBANK ; EXAMINE BANK TST ACFLAG ; CAN WE ACCESS THIS BANK?	
7457 016032 001415 7458 016034 005737 002126	BEQ 4\$;NO - GO TO BANK LOOP TERMINATION TST BMFLAG ;IS THIS BAD MEMORY (WORST FIRST) BEQ 4\$;NO - GO TO BANK LOOP TERMINATION	
7460 016042 005737 002122 7461 016046 001607	BEQ 4\$;NO - GO TO BANK LOOP TERMINATION TST RRFLAG :RELOCATION REQUIRED? BNE 4\$;YES - GO TO BANK LOOP TERMINATION CALL SETPAT ;SET HIGH PATTERN FOR CORRECT MEMORY	
7461 016046 001007 7462 016050 004737 047500 7463	START OF PATTERN LOOP	
7464 016054 004737 016752 7465	2\$: CALL MTEST ;GO TEST CORRECT MEMORY :TERMINATION OF PATTERN LOOP	
7466 016060 005337 002110 7467 016064 100373	DEC PATTERN ; IS THIS THE LAST PATTERN? BPL 2\$;NO - LOOP ON THIS PATTERN ; TERMINATION OF BANK LOOP	
7468 7469 016066 005037 002220 7470 016072 004737 047510 7471 016076 002351	4\$: CLR CONTFLAG CALL INCBNK ;NEXT HIGHER BANK BGE 1\$;IF NOT DONE - LOOP ON THIS BANK ;END OF LOOPS	
7472 7473 016100 005137 002564 7474 016104 001003	COM WORST ; IS THIS AN EVEN NUMBERED PASS? BNE 5\$; YES - SKIP ;**NOTE** RECURSIVE CALL	
7475 7476 016106 004737 016016 7477 016112 000207	CALL BAWPAR ; CALL SELF	
7478 016114 005737 002124 7479 016120 001401	5\$: TST RLFLAG ;HAVE WE BEEN RELOCATED? BEQ 6\$;NO - SKIP RETURN ;YES - RETURN	
7480 016122 000207 7481 016124 004737 045172 7482 016130	6\$: CALL RELOCATE ; MOVE 3 MAP PROGRAM ON.ERROR THEN \$RETURN	
7483 7484 016134 004737 016016 7485 016140 004737 046056 7486 016144 000207	;**NOTE** RECURSIVE CALL CALL BAWPAR ; CALL SELF CALL UNRELOCATE ; UNMOVE & UNMAP PROGRAM RETURN	

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 201 BANKS WORST FIRST, PATTERNS REVERSE **RECURSIVE**

7489	016146			*****	SUBTST < <patterns **recursive**="" banks="" forward="" forward,="">> *********************************</patterns>
				**SUBTE	
7490 7491	016146	005037	002110	*****	CLR PATTERN ;SET PATTERN TO 0 ;START OF PATTERN LOOP
7492	016152	005037	002100	1\$:	CLR BANK ;SET BANK TO 0 :START OF BANK LOOP
7494 7495	016156 016162 016166 016170 016174 016176	004737 004737	047020 047446	2\$:	CALL EXBANK ; EXAMINE BANK CALL BANKOP : CORRECT MEMORY FOR THIS BANK?
7496 7497 7498	016166 016170 016174	001010 005737 001405 005737	002114		BNE 4\$:NO - GO TO BANK LOOP TERMINATOR TST ACFLAG :CAN WE ACCESS THIS BANK? BEQ 4\$:NO - GO TO BANK LOOP TERMINATION TST RRFLAG :RELOCATION REQUIRED? BNE 4\$:YES - GO TO BANK LOOP TERMINATION CALL MTEST :GO TEST CORRECT MEMORY
7499 7500	016176	005737	002122		TST RRFLAG : RELOCATION REQUIRED? BNE 4\$;YES - GO TO BANK LOOP TERMINATION
7501 7502	016204	001002 004737	016752		BNE 4\$;YES - GO TO BANK LOOP TERMINATION CALL MIEST ;GO TEST CORRECT MEMORY :TERMINATION OF BANK LOOP CLR CONTFLAG
7503 7504 7505	016202 016204 016210 016214 016220 016222	005037 004737 002356	002220 047510	4\$:	CLR CONTFLAG CALL INCBNK ;NEXT HIGHER BANK BGE 2\$:IF NOT DONE - LOOP ON THIS BANK ;TERMINATION OF PATTERN LOOP
7507 7508 7508	016222 016226	004737 001351	047464		CALL INCRPT ; NEXT HIGHER PATTERN BNE 1\$; OK - LOOP; ELSE CONTINUE ; END OF LOOPS
7510 7511	016230	005137	002132		COM TMFLAG ; COMPLEMENT TYPE OF MEMORY ; IS THIS AN EVEN NUMBER PASS?
7512	016234	001403			BEQ 5\$;YES - SKIP :**NOTE** RECURSIVE CALL
7514 7515	016236 016242	004737 000207	016146		CALL PAFBAF ; CALL SELF RETURN
7516 7517	016244	004737 000207 005737 001401 000207 004737	002124	5\$:	TST RLFLAG : HAVE WE BEEN RELOCATED? BEQ 6\$:NO - SKIP RETURN : YES - RETURN
7519 7520 7521	016254 016260	004737	045172	6\$:	CALL RELOCATE ; MOVE & MAP PROGRAM ON.ERROR THEN \$RETURN ; **NOTE** RECURSIVE CALL
7522 7523 7524	016236 016242 016244 016250 016252 016254 016260 016270 016274	004737 004737 000207	016146 046056		CALL PAFBAF ; CALL SELF CALL UNRELOCATE ; UNMOVE & UNMAP PROGRAM RETURN

7527 016276			*SUBTE	SUBTST < <patterns **recursive**="" banks="" first="" forward,="" worst="">> ST PATTERNS FORWARD, BANKS WORST FIRST **RECURSIVE**</patterns>
7528 016276	005037	002110	;*****	CLR PATTERN ;SET PATTERN TO 0 ;START OF PATTERN LOOP
7529 7530 016302		002100	1\$:	CLR BANK ;SET BANK TO 0 ;START OF BANK LOOP
7531 7532 016306 7533 016316 7534 016316 7535 016326 7536 016326 7537 016326 7538 016336 7539 016336 7540 016346	004737 004737	047020 047446	2\$:	CALL EXBANK : EXAMINE BANK CALL BANKOK : CORRECT MEMORY FOR THIS BANK?
7534 016316 7535 016320	001013	002114		TST ACFLAG ; CAN WE ACCESS THIS BANK?
7536 016324 7537 016324	001410 005737 001405 005737	002126		BEQ 4\$:NO - GO TO BANK LOOP TERMINATION TST BMFLAG :IS THIS BAD MEMORY (WORST FIRST)
7538 01633	001405			
7540 01634	001002	002122		BNE 4\$:YES - GO TO BANK LOOP TERMINATION
		016752		:TERMINATION OF BANK LOOP
7543 016346 7544 016356 7545 016356 7546	005037 004737 002353	002220 047510	4\$:	BEQ 4\$;NO - GO TO BANK LOOP TERMINATION TST RRFLAG ;RELOCATION REQUIRED? BNE 4\$;YES - GO TO BANK LOOP TERMINATION CALL MTEST ;GO TEST CORRECT MEMORY ;TERMINATION OF BANK LOOP CLR CONTFLAG CALL INCBNK ;NEXT HIGHER BANK BGE 2\$;IF NOT DONE - LOOP ON THIS BANK
7546 7547 016366 7548 016366 7549	004737 001346			; TERMINATION OF PATTERN LOOP CALL INCRPT ; NEXT HIGHER PATTERN BNE 1\$; OK - LOOP; ELSE CONTINUE ; END OF LOOPS
7550 016366 7551	005137	002132		COM TMFLAG ; COMPLEMENT TYPE OF MEMORY ; IS THIS AN EVEN NUMBER PASS?
7552 016373 7553	001403			BEQ 5\$; YES - SKIP
7555 016400	004/5/	010276		;**NOTE** RECURSIVE CALL CALL PAFBAW ;CALL SELF RETURN
7556 016400 7557 016400 7558	000207 005137 001003	002564	5\$:	COM WORST ;4TH PASS? BNE 6\$;YES - SKIP ;**NOTE** RECURSIVE CALL
7559 01641	004737	016276		CALL PAFBAW ; CALL SELF
7561 016410 7562 01642	000207	016276 002124	6\$:	RETURN TST RLFLAG ;HAVE WE BEEN RELOCATED? BEQ 7\$;NO - SKIP
7559 016410 7560 016410 7561 016410 7562 016420 7563 016420 7564 016420 7565 016430	001401 000207 004737	045172	7\$:	RETURN CALL RELOCATE ; MOVE & MAP PROGRAM ON.ERROR THEN SRETURN
7565 01643 7566 7567 01643 7568 01644 7569 01644	004737	016276 046056		;**NOTE** RECURSIVE CALL CALL PAFBAW ; CALL SELF CALL UNRELOCATE ; UNMOVE & UNMAP PROGRAM RETURN

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 205 PATTERNS FORWARD, BANKS WORST FIRST **RECURSIVE**

7572 016450	*SUBTEST	PATTERNS REVERS	RSE,BANKS FORWARD **RECURSIVE**>> E,BANKS FORWARD **RECURSIVE**
7573 016450 004737 047500 7574	;*************************************	L HIPAT ART OF PATTERN LOOP	;SET HIGHTEST PATTERNS
	1\$: CLR	BANK	;SET BANK TO 0
7577 016460 004737 047020 7578 016464 004737 047446 7579 016470 001010	25: CAL	L EXBANK	:EXAMINE BANK :CORRECT MEMORY FOR THIS BANK? :NO - GO TO BANK LOOP TERMINATOR
7579 016470 001010 7580 016472 005737 002114 7581 016476 001405 7582 016500 005737 002122	BNE TST BEG	ACFLAG	CAN WE ACCESS THIS BANK?
7582 016500 005737 002122	TST	RRFLAG	RELOCATION REQUIRED?
7583 016504 001002 7584 016506 004737 016752	CAL	L MTEST	; CORRECT MEMORY FOR THIS BANK? ;NO - GO TO BANK LOOP TERMINATOR ;CAN WE ACCESS THIS BANK? ;NO - GO TO BANK LOOP TERMINATION ;RELOCATION REQUIRED? ;YES - GO TO BANK LOOP TERMINATION ;GO TEST CORRECT MEMORY
7575 016454 005037 002100 7576 7577 016460 004737 047020 7578 016464 004737 047446 7579 016470 001010 7580 016472 005737 002114 7581 016476 001405 7582 016500 005737 002122 7583 016504 001002 7584 016506 004737 016752 7585 7586 016512 005037 002220 7587 016516 004737 047510 7588 016522 002356 7589 7590 016524 005337 002110 7591 016530 100351	CAL	CONTFLAG L INCBNK	:NEXT HIGHER BANK :IF NOT DONE - LOOP ON THIS BANK
7589 7590 016524 005337 002110 7591 016530 100351	DE C	RMINATION OF PATTERN PATTERN 1\$:NEXT LOWER PATTERN :OK - LOOP; ELSE CONTINUE
7592 7593 016532 005137 002132	COM	D OF LOOPS	COMPLEMENT TYPE OF MEMORY IS THIS AN EVEN NUMBER PASS?
7593 016532 005137 002132 7594 7595 016536 001403	BEG	5\$;YES - SKIP
7596 7597 016540 004737 016450	ČAL	NOTE** RECURSIVE CAL L PARBAF TURN	; CALL SELF
7597 016540 004737 016450 7598 016544 000207 7599 016546 005737 002124 7600 016552 001401 7601 016554 000207 7602 016556 004737 045172	5\$: TS1	RLFLAG	; HAVE WE BEEN RELOCATED? ; NO - SKIP
7601 016554 000207 7602 016556 004737 045172 7603 016562	6\$: CAL	TURN L RELOCATE ERROR THEN \$RETURN NOTE** RECURSIVE CAL	:NO - SKIP :YES - RETURN :MOVE & MAP PROGRAM
7595 016536 001403 7596 7597 016540 004737 016450 7598 016544 000207 7599 016546 005737 002124 7600 016552 001401 7601 016554 000207 7602 016556 004737 045172 7603 016562 7604 7605 016566 004737 016450 7606 016572 004737 046056 7607 016576 000207	CAL	I PARBAF	CALL SELF ;UNMOVE & UNMAP PROGRAM

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 207 PATTERNS REVERSE, BANKS FORWARD **RECURSIVE**

7610	016600			******	SUBTST < <patterns **recursive**="" banks="" first="" reverse,="" worst="">> T PATTERNS REVERSE, BANKS WORST FIRST **RECURSIVE**</patterns>
7611 7612	016600	004737	047500	,	CALL HIPAT ;SET HIGHTEST PATTERN ;START OF PATTERN LOOP
7613			002100	1\$:	CLR BANK ;SET BANK TO 0 ;START OF BANK LOOP
7615 7616 7617	016610 016614 016620 016622 016626 016630 016634 016636	004737 004737 001013 005737 001410 005737 001405 005737	047020 047446	2\$:	CALL EXBANK ; EXAMINE BANK
7618	016622	005737	002114		TST ACFLAG ; CAN WE ACCESS THIS BANK?
7620	016630	001410	002126		SEQ 4\$:NO - GO TO BANK LOOP TERMINATION TST BMFLAG :IS THIS BAD MEMORY (WORST FIRST) BEQ 4\$:NO - GO TO BANK LOOP TERMINATION
7621 7622	016634 016636	001405	002122		BEQ 4\$:NO - GO TO BANK LOOP TERMINATION TST RRFLAG :RELOCATION REQUIRED?
7623	016642		016752		TST RRFLAG :RELOCATION REQUIRED? BNE 4\$:YES - GO TO BANK LOOP TERMINATION CALL MTEST :GO TEST CORRECT MEMORY :TERMINATION OF BANK LOOP
7625	010044	004737			I ENTINATION OF DANK LOOP
7626 7627 7628	016644 016650 016654 016660 016662	005037 004737 002353	002220 047510	4\$:	CALL INCBNK :NEXT HIGHER BANK BGE 2\$:IF NOT DONE - LOOP ON THIS BANK
7629 7630 7631	016662 016666	005337 100346	002110		TERMINATION OF PATTERN LOOP DEC PATTERN ;NEXT LOWER PATTERN BPL 1\$;OK - LOOP; ELSE CONTINUE TEMP OF LOOPS
7633	016670	005137	002132		COM TMFLAG COMPLEMENT TYPE OF MEMORY ; IS THIS AN EVEN NUMBER PASS?
7635	016674	001403			BEQ 5\$;YES - SKIP ;**NOTE** RECURSIVE CALL
7637 7637	016674 016676 016702 016704	004737	016600		CALL PARBAW ; CALL SELF
7639 7640	016704 016710	004737 000207 005137 001003	002564	5\$:	COM WORST ;4TH PASS? BNE 6\$;YES - SKIP ;**NOTE** RECURSIVE CALL
7642	016712	004737	016600		CALL PARBAW ; CALL SELF
7643 7644 7645	016716 016720 016724	000207 005737 001401	002124	6\$:	RETURN TST RLFLAG :HAVE WE BEEN RELOCATED? BEQ 7\$:NO - SKIP
7646 7647 7648	016726 016730 016734	004737 000207 005737 001401 000207 004737	045172	7\$:	RETURN CALL RELOCATE ; MOVE & MAP PROGRAM ON. ERROR THEN \$RETURN
7650	016740 016744 016750	004737 004737 000207	016600 046056		:**NOTE** RECURSIVE CALL CALL PARBAW ; CALL SELF CALL UNRELOCATE ; UNMOVE & UNMAP PROGRAM RETURN

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 209 PATTERNS REVERSE, BANKS WORST FIRST **RECURSIVE**

7655 016752		MTEST: SUBTST	< <subr memor<="" setup="" th=""><th>Y TEST>></th></subr>	Y TEST>>
		:*SUBTEST	SUBR SETUP MEMOR	Y TEST
7656 016752 7657 016760 7658 016766 00503	7 002242	SET	HEADER MUT PASFLG	:INITIALIZE HEADER MESSAGE TYPEOUT :INDICATE THERE IS A MEMORY UNDER TEST
7659 016772 00573 7660 016776 00141	7 002116	CLR TST BEQ BEGIN	MKFLAG MT1 HOLDLOOP	:ECC?
7661 017000 7662 017000 7663 017006		IF CO	ONTFLAG IS TRUE THEN CIPMK IS FALSE	LEAVE HOLDLOOP
7663 017006 7664 017014 00473 7665 017020 7666 017020	37 017046	END:	OF IF SKIPMK HOLDLOOP	
7667 017020 00473	7 020274	CALL BR	MKTEST MT2	;YES - DO ECC TESTS
7667 017020 00473 7668 017024 00040 7669 017026 00473 7670 017032 00503 7671 017036	7 020514 37 002106	MT1: CALL MT2: CLR SET	MJTEST MUT HEADER	:DO PARITY TESTS :NOW - NO MEMORY UNDER TEST :ALLOW HEADERS NORMAL
7671 017036 7672 017044 00020)7	RETURN	HEADEN	

```
<<SUBR TEST ECC CSR LOGIC DISPATCH>>
                                                        MKCONTROL: SUBTST
7675 017046
                                                        ********
                                                                                            TEST ECC CSR LOGIC DISPATCH
                                                        : *SUBTEST
                                                                    THE NEXT TWO MODULES SOLVE THE PROBLEM OF
                                                                    HOW TO RUN THE CSR TESTS ON EACH ECC MEMORY
7677
7678
7678
7679 017046
7680 017056
7681 017066
7682 017102
7683 017110
7684 017116
7685 017122
7686 017126
7687 017132
7688 017136
7689 017142
7690 017144
7691 017150
7692 017152
                                                                    IF SELONLY IS TRUE THEN SRETURN
IF INHECC IS TRUE THEN SRETURN
                                                                                BANK, RO.R1, R2, R3
#FIRST, CSRFBANK
#LAST, CSRLBANK
                                                                    PUSH
                   012737
012737
005037
                               060000
                                                                    MOV
                                                                                                                    SET FIRST TEST ADDRESS TO FIRST ADDR.
                                                                    MOV
                               002234
002236
002326
002102
                                                                                CSRINT
                                                                    CLR
                   005037
005037
013700
                                                                                SPLTCSR
CSRLOOP
                                                                    CLR
                                                                                                                    ; AND ZERO THE LOOP COUNTER
                                                                    CLR
                                                                                                                    GET THE BANK INDEX
                                                                                BANKINDEX, RO
                                                                    MOV
                                                                    MOV
                                                                                CONFIG(RO),R1
                                                                                                                    GET CSR NUMBER
                    016001
                                002650
                    000301
                                                                    SWAB
                    042701
006301
                                                                    BIC
                                                                                #^C17_R1
                                177760
                                                                    ASL
                                                                                R1, CSRHOLD
7692
7693
                                                                                                                    STORE IN THE LOW BYTE
       017152 017156
                                002522
                    010137
                                                                    MOV
                                                                                                                    : IS THIS BANK INTERLEAVED?
                    005737
                                                                    TST
                                                                                INTFLAG
                                                                                                                    :BRANCH IF NOT INTERLEAVED
 7694
                                                                    BEQ
       017162
                               002236
120000
002326
002234
002650
177775
 7695
7696
7697
       017164
017170
017176
                                                                    INC
                                                                                SPLTCSR
                                                                                #120000, CSRLBANK
                                            002232
                                                                    MOV
                                                                                                                    :WE MUST LOOP TWICE FOR AN INTERLEAVED BANK
                                                                    INC
                                                                                CSRLOOP
       017202
017206
017212
                                                                    INC
                                                                                CSRINT
 7698
                    016001
072127
042701
                                                                                CONFIG(RO) R1
                                                                                                                    GET THE INTERLEAVE CSR NUMBER
 7699
                                                                    MOV
                                                                                #-3,R1
#^C17000,R1
                                                                    ASH
 7701
       017216
                                160777
                                                                    BIC
7701 017216
7702 017222
7703 017226
7704 017230
7705 017236
7706 017244
7707 017250
7708 017256
7709 017272
7710 017274
7711 017274
                                                                                R1, CSRHOLD
                    050137
                                002522
                                                                                                                    STORE IT IN CSRHOLD'S UPPER BYTE
                                                                    BIS
                    005003
                                                                    CLR
                    116337
                                                        MKLOOP: MOVB
                                                                                CSRHOLD (R3), CSRNO
                                                                                #^C36, CSRNO
                                                                                                                    :CLEAR ANY UNNECESSARY BITS
                                                                    BIC
                                                                       FOR MKCNT := #0 TO CSRINT
                                                                          FOR CSRFIRST := CSRFBANK TO CSRLBANK BY #4000
                                                                                                                    MAP TEST SPACE TO BANK
                                                                          MAP BANK
                                                                                                                    :INVALIDATE BACKROUND PATTERN
                    104511
                                                                          INVALIDATE
                                                                             BEGIN CSRSTUFF
                                                                                            SUCCESS
                    005037 002330
                                                                               IF ACFLAG IS TRUE AND RRFLAG IS FALSE
MOV CSRFIRST, CSRLAST
ADD #4000, CSRLAST
FOR TESTADD := CSRFIRST TO CSRLAST BY #4
MOV TESTADD, TESTADD+2
 7712 017300
7713 017314
                                002224
                                            002226
                    013737
062737
 7714 017322
7715 017330
                    013737
005737
001404
062737
000403
062737
004737
7716 017336
7717 017344
7718 017350
                                002406
002236
                                            002410
                                                                                          TST
                                                                                                         SPLTCSR
 7718
7719
                                                                                                         #40000, TESTADD+2
        017352
                                040000
                                            002410
                                                                                          ADD
        017360
017362
017370
                                                                                          BR
                                000002
                                                                                                         #2.TESTADD+2
                                                                                          ADD
                                            002410
 7722
7723
7724
7725
                                                                                                         SBETEST
                                                                                          CALL
                                                                                          ON. NOERROR
        017374
        017376
017400
                                                                                            CACHOFF
                                                                                                                                :TURN CACHE OFF
                    104424
005037
                                                                                                                                :INDICATE PARITY ACTION
                                002074
                                                                                             CLR
                                                                                                        NOPAR
                                                                                            FOR I := #0 TO #27
        017404
                                                                                               SET
                                                                                                        HEADER
        017410
                    005037 002262
 7728 017416
                                                                                                        PASFLG
```

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 211-1 SUBR TEST ECC CSR LOGIC DISPATCH

```
7729 017422
7730 017426
7731 017430
7732 017434
7733 017442
7734 017446
7735 017466
                                                                                                                       LET RO := I
                                                                                                                       PUSH R3
                                                                                                                                                                ; SAVE LOOP COUNTER
                                                                                                                      MOV SP.CTLKVEC
SUB #2.CTLKVEC
                        010637
162737
004737
                                       002144
000002
020174
                                                                                                                                                                : SAVE VECTOR IN CSR OF AK
                                                      002144
                                                                                                                      CALL CSRCASE
POP R3
                                                                                                                                                                :RESTORE LOOP COUNTER
                                                                                                                   END OF FOR I
7736 017464
7737 017466
7738 017474
7739 017476
                                                                                                                                                                ; TURN CACHE ON
                        104423
                                                                                                 LEAVE CSRSTUFF
END ; OF ON. NOERROR
END ; OF FOR TESTADD
END ; OF IF
7740 017476
7741 017514
7742 017514
7743 017514
                                                                                                 END CSRSTUFF
                                                                                                IF SUCCESS IS FALSE
7743 017514
7744 017522
7745 017526
7746 017536
7747 017542
7748 017546
7749 017546
7750 017564
7751 017570
7752 017604
7753 017612
7754 017620
7755 017622
7756 017630
7758 017634
7759 017636
7760 017644
                                                                                                    TYPE
                                                                                                                   MSGA34
                                                                                                    TYPOCS BANK, <TYPES BANK NUMBER>,3
                                                                                                                   MSGB34
                                                                                                    TYPE
                                                                                                    CALL
                                                                                                                   PERBNK
                        004737 057476
                                                                                            END OF IF SUCCESS
END: OF FOR CSRFIRST
INC SPLTCSR
                         005237
                                       002236
                                                                                        END: OF FOR MKCNT
                        062737
012737
005203
020337
003002
000137
104472
                                                                                                    #2.CSRFBANK
#1.SPLTCSR
                                        000002
                                                                                     ADD
                                                                                     MOV
                                                                                     INC
                                                                                                    R3,CSRLOOP
                                                                                     CMP
                                        002326
                                                                                     BGT
                                                                                                    CONTFLAG ON DOUBLE BIT ERRORS (NORMAL)
                                        017230
                                                                                     JMP
                                                                                     ECCINIT
                                                                      15:
7760 017644
7761 017650
7762 017664
7763 017666
                                                                                                    SPLTCSR
R3,R2,R1,R0,BANK
                         005037
                                       002236
                                                                                     CLR
                                                                                     POP
                         000207
                                                                                     RETURN
                                                                      MKCNT:
                                                                                    . WORD
                                                                                                    0
                                                                                                                                                 COUNTER FOR MKLOOP
```

```
7766 017670
                                              SBETEST:SUBTST <<CHECK FOR SBE FREE LOCATIONS>>
                                              **************
                                              :*SUBTEST
                                                                  CHECK FOR SBE FREE LOCATIONS
                                              *************************
7767
7768
7769
                                                        :IN ORDER TO DETERMINE IF A LOCATION IS SBE FREE I DO THIS
                                                        WRITE ZEROS WITH ECC DISABLE READ ZEROS BACK
7770
                                                        : IF NOT ZEROS THEN RETURN ERROR
                                                        WRITE ZEROS WITH ECC ENABLED BUT TRAPS DISABLED
                                                        :READ ZEROS BACK
                                                        IF NOT ZEROS THEN RETURN ERROR
                                                        :TEST THE LOCATION FROM THE PAR'S (WITH NO PROGRAM FETCHES)
:IF THERE WERE ANY SBE'S OR DBE'S THEN RETURN ERROR
                                                        COMPLIMENT ZEROS TO ONES WITH ECC DISABLE
                                                        READ ONES BACK
                                                        IF NOT ONES THEN RETURN ERROR
                                                        WRITE 100.,100000,00000 (CHECKBITS COMPLIMENT OF BEFORE)
                                                         TEST THE LOCATION FROM THE PAR'S (WITH NO PROGRAM FETCHES)
                                                        IF THERE WERE ANY SBE'S OR DBE'S THEN RETURN ERROR
7789
                                                        IF NONE OF THE ABOVE FORCES A RETURN ERROR THEN RETURN NO. ERROR
7790
                                                        .ENABL LSB
                                                                 RO,R1,R4
      017670
                                                        PUSH
                                                                                               :PUSH RO,R1,R4 ONTO STACK
                013701
      017676
                                                        MOV
                                                                  TESTADD, R1
      017702
                                                        MOV
                                                                TESTADD+2,R4
7794 017706
7795 017714
                                                        TESTAREA
                                                                                                ENTER TEST MODE
                                                        CACHOFF
                                                                                               :TURN CACHE OFF
7796
      017716
                                                        ECC1DIS
                                                                                               :DISABLE ECC ON 1 SELECTED CSR
7797 017720
7798 017724
7799 017726
7800 017730
7801 017732
                                                                  (R1),(R4)
(R1)
                                                        CLEAR
                005711
001107
005714
                                                        TST
                                                        BNE
                                                                  SBENT
                                                        TST
                                                                  (R4)
                001105
                                                        BNE
                                                                  SBENT
7802
7803
7804
7805
7806
7807
7808
7809
7810
7811
7812
7813
7816
7817
7818
     017734
017736
017742
017744
017746
017750
                104503
                                                        CLR1CSR
                                                                                               :CLEAR 1 SELECTED CSR
                                                                  (R1),(R4)
(R1)
                                                        CLEAR
                005711
                                                        TST
                001100
                                                                 SBENT
                                                        BNE
                                                                  (R4)
                001076
                                                        BNE
                                                                  SBENT
     017752
017754
017764
017772
017774
020000
020004
020010
020016
020016
                104510
                                                        TSTREAD
                                                                                      :TEST LOC (R1) & TST FOR SBE (WITHOUT FETCHES)
                                                        IF #BIT15!BIT4 SET.IN CSR
                                                          SET SKPERR
                                                                                               ; DISABLE ERRGEN'S ERROR PRINTOUT
                104512
013700
072027
042700
                                                          ERRGEN
                                                          MOV ERRADD, RO
ASH #-4, RO
BIC #^C177, RO
                                                       BIC M^C177, RO
IF BANK EQ RO THEN GOTO SBENT
END: OF IF M9IT15
ECC1DIS
                104471
                                                                                               ; DISABLE ECC ON 1 SELECTED CSR
```

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 213-1 CHECK FOR SBE FREE LOCATIONS

7820 7821 7822 7823 7824 7825	020020 020022 020024 020030 020032 020036	005111 005114 023711 001046 023714 001043	002600 002600		COM COM CMP BNE CMP BNE	(R1) (R4) ONES, (R1) SBENT ONES, (R4) SBENT	
7826 7827 7828 7829 7830 7831 7832 7833	020040 020042 020044 020050 020052 020054 020060	104503 005011 012714 005711 001035 022714 001032	100000		CLR1CSR CLR MOV TST BNE CMP BNE	(R1) #BIT15,(R4) (R1) SBENT #BIT15,(R4) SBENT	CLEAR 1 SELECTED CSR
7834 7835 7836 7837 7838 7839 7840 7841 7842 7843	020024 020030 020032 020036 020042 020044 020050 020052 020054 020060 020064 020064 020074 020102 020104 020110 020110 020110 020126 020130 020132 020132 020134 020142	104512 013700 072027 042700	002454 177774 177600		TSTREAD IF #BIT SET SI ERRGEI MOV ASH BIC IF BAI END; OF	15 PITA SET. IN CSR	C (R1) & TST FOR SBE (WITHOUT FETCHES) ;DISABLE ERRGEN'S ERROR PRINTOUT
7845 7845 7846 7847 7848 7849	020126 020130 020132 020134 020142	104417 104473 104423			KERNEL ECC1INI CACHON POP \$RETURN	R4,R1,R0 NOERROR	:ENTER KERNEL MODE :INITIALIZE 1 SELECTED CSR :TURN CACHE ON :POP RO,R1 & R4 FROM STACK
7850 7851 7852 7853 7854 7855 7856 7856	020146 020150 020154 020156 020160 020162 020170	104503 104417 104473 104423		SBENT:	CLR1CSR CLEAR KERNEL ECC1INI CACHON POP \$RETURN .DSABL	(R1),(R4) T R4,R1,R0	CLEAR 1 SELECTED CSR ENTER KERNEL MODE INITIALIZE 1 SELECTED CSR TURN CACHE ON POP RO,R1 & R4 FROM STACK

7861	020174		CSRCASE:	SUBTST < <csr patt<="" th=""><th>TTERN CASE STATE</th><th>**************************************</th></csr>	TTERN CASE STATE	**************************************
7862 7863 7864 7865	020174		;******	CASE RO ; WARNING IF YO ; CHANGE 'SDDWO	U CHANGE THIS	TABLE ALSO HE PATTERN BIT MAP)
7866	020204	021750	MKCSRT:	PAT MT0006	TIME ;<1 SEC	DESCRIPTION INITIAL DATA TEST
7867 7868 7869 7870 7871					MS11-P ECC TES	STS
7871 7872 7873 7874 7875 7876 7877 7878 7879 7881 7882 7883 7884	020206 020210 020212 020214 020216 020220 020222	026554 022350 026614 026264 022616 026336 026406 026450 026514 026654		MT0014 MT0045 MT0036 MT0020 MT0037 MT0041	: 1 SEC : 1 SEC	SHIFTING 1/0'S THROUGH CHECK BITS BASIC DOUBLE ERROR TEST SYNDROMES TO CSR ON DOUBLE BIT ERROR TEST CORRECTION CODE TEST SYNDROMES TO CSR UN SINGLE BIT ERROR TEST ECC DISABLE TEST ADDRESS TO CSR ON DOUBLE BIT ERROR EXTENDED ADDRESS TO CSR ON ERROR TEST WRITE BYTE CLEARS SBE TEST CHECK SINGLE BIT ERRORS WITH ECC DISABLED TEST NO CSR UPDATE ON SBE WITH EXSISTING DBE
7883 7884 7885	020234	022046		MT0010	;<1 SEC	BYTE ADDRESSING TEST
7886 7887				1	MS11-M ECC TES	STS
7894 7895		023522 022102 022160 022264 022440 022516		MT0025 MT0011 MT0012 MT0013 MT0015 MT0016	;<1 SEC ;<2 SEC ;<1 SEC ; 1 SEC ;<1 SEC ;<1 SEC	INTERRUPT ENABLE TEST CREATE SINGLE BIT ERROR TEST WRITE BYTE CLEARS SBE TEST CREATE DOUBLE BIT ERROR TEST WRITE INHIBIT OF BYTE WITH DBE WRITE INHIBIT OF WORD WITH DBE
7896 7897 7898 7899 7900 7901 7902 7903 7904	020204	026760 026760 026760 026760 026760 000207		MT0999 MT0999 MT0999 MT0999 MT0999 END ; OF CASE RO RETURN	O SEC O SEC O SEC O SEC	NULL TEST NULL TEST NULL TEST NULL TEST NULL TEST NULL TEST

```
MKTEST: SUBTST <<SUBR ECC TEST DISPATCH>>
7907 020274
                                                                                                                                     *************************
                                                                                                                                      **SUBTEST
                                                                                                                                                                                                                             ECC TEST DISPATCH
                                                                                                                                      **********************
                                                                                                                                                                  IF #SWO SET.IN ASWR OR ACTFLAG IS TRUE
7908 020274
7909 020312
7910 020314
7911 020316
7912 020320
7913 020320
7914 020326
7915 020334
7916 020340
7917 020342
7918 020364
7920 020364
7921 020370
7922 020376
7923 020402
7924 020420
7925 020422
7926 020424
7927 020426
7928 020426
7929 020432
7930
7931
                                                                                                                                                                                                                                                                                    :DISABLE ERROR CORRECTION
                                                                                                                                                                          ECCDIS
                                               104470
                                                                                                                                                                CLRCSR
END ; OF IF
MOV #2, NOPAR
MOV #2, PCBUMP
                                                                                                                                                                       CLRCSR
                                                                                                                                                                                                                                                                          :CLEAR ALL CSR'S
                                                104502
                                              012737 000002 002074
012737 000002 002322
013700 002110
                                                                                                                                                                                                                                                                                     :INDICATE PARITY ACTION
:TRAPS ADD 2 TO PC
                                                                                                                                                                  MOV PATTERN, RO :GET PATTERN NUMBER
ASL RO :MAKE IT A WORD ADDRESS
IF MKPAT(RO) NE #MT0034 AND MKPAT(RO) NE #MT0999
INVALIDATE ;INVALIDATE BACKGROUND PATTERN ON 'BANK''
                                                006300
                                                104511
                                                                                                                                                                   END ; OF IF MKPAT(RO)
MOV SP, CTLKVEC
SUB #2, CTLKVEC
                                              010637
162737
004770
                                                                           002144 000002 002144
                                                                                                                                                                                                                                                                                       :SAVE VECTOR IN CASE OF AK
                 020376
020376
020402
020420
020422
020424
020426
020432
                                                                                                                                                                                                aMKPAT (RO)
                                                                                                                                                                                                                                                                                         : INDEX OFF TABLE
                                                                                                                                                                    CALL
                                                                                                                                                                   IF #SWO SET. IN OSWR OR ACTFLAG IS TRUE
                                                                                                                                                                                                                                                                                     :TRAP ON SINGLE BIT ERRORS
                                                                                                                                                                         ENASBE
                                                104506
                                                                                                                                                                    ELSE
                                                                                                                                                                                                                                                                               :TRAP ON DOUBLE BIT ERRORS (NORMAL)
                                                                                                                                                                          ECCINIT
                                                104472
                                                                                                                                                                   END ; OF IF #SWO
CLR NOPAR
                                                                                                                                                                                                                                                                       ;INDICATE PARITY ACTION
                                                                           002074
                                                005037
                                                                                                                                                                   RETURN
                                                000207
                                                                                                                                                                   ;WARNING IF YOU CHANGE THIS TABLE ALSO

;CHANGE 'SDDWO' - 'SDDW5'' (THE PATTERN BIT MAP)

;PAT TIME DISCRIPTION

;NOTE MT0034 MUST BE FIRST & LAST

MT0034 :<1 SEC ;SOFT ERROR - BACKGROUND PATTERN TEST
  7932
 7933
7934 020434
7935 020434
7936 020436
7937 020440
7938 020442
7939 020444
7940 020450
7942 020452
7943 020454
7944 020456
7945 020460
7946 020464
7948 020466
7949 020470
7950
7951 020472
  7933
                                                                                                                                                 ### TEST

### OF STEST

### OF STEST

### COMPLEMENT ADDRESS TEST

### ROTATING ONES TEST

### ROTATIN
                                                                                                                                                                    MT0034 :<1 SEC
MT0017 :<1 SEC
                                               026000
                                                022004
020760
021100
                 020472
020474
020476
020500
020502
   7956 020504
7957 020506
```

50011											
7962	020514					******	*****	*****		DISPATCH>> DISPATCH	******
					; *SUBTES	Τ	SUBR	PARITY	TEST	DISPATCH	
7963 7964 7965 7966 7967 7968 7970 7971 7972 7973	020514 020522 020530 020536 020544 020550 020552 020572 020574 020574 020600 020606 020616	012737 012737 012737 012737 012737 013700 006300 104511 010637 162737 004770 005037 000207	000002 000002 060000 060002 002110	002074 002322 002406 002410	;*****	MOV MOV MOV MOV ASL IF MJPAT INVALI END ; OF MOV SUB CALL	#2,NOPA #2,PCBU #FIRST, #FIRST+ PATTERN RO (RO) NE IDATE IF MJPA SP,CTLK #2,CTLK @MJPAT(R MP TESTADD 2.TESTAI ,RO #MTOO34 T(RO) VEC VEC RO)	DD+2 4 AND ;INV	; INDICATE PARITY ACTION ; TRAPS ADD 2 TO PC ; GET PATTERN NUMBER ; MAKE IT A WORD ADDRESS MJPAT(RO) NE #MT0999 VALIDATE BACKGROUND PATTERN ON 'BAF ; SAVE VECTOR IN CASE OF "K ; INDEX OFF TABLE ; INDICATE PARITY ACTION	ık"
7975	020612	005037	002074			CLR	NOPAR			;INDICATE PARITY ACTION	
7977 7978 7979 7980	0200.0	000207				; WARNING	IF YOU	CHANGE	THIS	TABLE ALSO	
7981 7982 7983 7984 7985 7986 7987 7988 7989 7991 7992 7993 7994 7995 7996 7997 7998 8001 8002 8003 8004	020620 020620 020622 020624 020626 020630 020632 020634 020636 020640 020642 020644 020646 020650 020652 020654 020656 020660 020662 020662	026000 021750 022574 022004 020760 021100 021240 021472 021614 022706 025160 023212 023600 023256 025070 025260 025612 026000 026760 026760 026760 026760			MJPAT:	: PAT : NOTE M MT0034 MT0006 MT0007 MT0001 MT0002 MT0003 MT0005 MT0021 MT0023 MT0023 MT0026 MT0024 MT0033 MT0034 : NOTE M MT0999 MT0999 MT0999 MT0999 MT0999	TIME 10034 MU 10034 MU	ST BE F	DISC IRST (SOF ; IN) ; HOL ; ADD ; ADD ; ADD ; ROT ; ROT ; ROT ; ROT ; SOF ; SOF ; SOF ; SOF ; SOF ; NUI ; N	CRIPTION LAST FT ERROR - BACKGROUND PATTERN TEST ITIAL DATA TEST LDING 1'S & O'S TEST DRESS BIT TEST DRESS TEST MPLEMENT ADDRESS TEST XOR 9 WORST CASE NOISE TEST TATING ZEROS TEST TATING ONES TEST RCHING O'S & 1'S TEST RSE CASE NOISE PARITY TEST FRESH TEST IFTING DIAGONAL TEST NDOM DATA TEST ST GALLOPING PATTERN TEST B-A-LONG TEST ITE RECOVERY TEST ANCH GOBBLE TEST FT ERROR - BACKGROUND PATTERN TEST LL TEST LL TEST LL TEST LL TEST	

PATTERNS	•							
8009					.SBT	TL	PATTERNS	
8010 8011					.SBT	TL	MEMORY TEST SETUP ROUTI < <mt0000 d<="" setup="" td=""><td>NES</td></mt0000>	NES
8012	020700				MT0000: SUBT	21	< <m10000 d<="" setup="" td=""><td>****</td></m10000>	****
					*SUBTEST	***	MT0000 SETUP DATA PATT	*********
8013	020700	005037	002274		CLR		REAL PAT	SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
8015	020704	012701	060000		MOV		WFIRST,RO WSIZE,R1 REGCOPY	
8016 8017	020714	005037 012700 012701 004737 022737	041064	003752	CALL		#1,PROTYP	:ARE WE ON AN 11/44?
8018	020726	001406 012737 004737	027400	002260	BEQ		#MTP000.SUPDOADD	; ARE WE ON AN 11/44? ; BRANCH IF YES ; ELSE DO PATTERN IN MAIN MEMORY
8020	020700 020704 020710 020714 020720 020726 020730 020736 020736	004737	027206	***************************************	CALL		SUPD03	
8022	020744		027070		1\$: BMOV	/	MTP000	;DO IT IN SUPERVISOR MODE
8024	020756	000207	027030		CALL	JRN		
8025	020760				MT0001: SUBT	ST	<mt0001 a<="" setup="" td=""><td>**********</td></mt0001>	**********
					*SUBTEST			
8026	020760	012737 012700 012701 005737 001005 023737 001007 000404 023737	000001	002274	MOV		#1 REALPAT	SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
8027	020760 020766 020772 020776	012700	060000 040000 002452		MOV		#SIZE,R1	
8029 8030	020776 021002	005737	002452		TST		NOSUPER 2\$	
8031	021004	023737	172252	172254	CMP		SIPAR5, SIPAR6	
8033	021014	000404	177452	177654	BR		3\$ UIPAR5,UIPAR6	
8035	021014 021016 021024 021026 021032	023737 001002 012701 005002 004737 022737 001406 012737 004737 000207		177034	BNE		4\$	
8036 8037	021026	005002	030000		3\$: MOV 4\$: CLR		#30000,R1 R2	
8038	021034	004737	041064 000001	003752	CALL	-	REGCOPY #1,PROTYP	:IS THIS AN 11/44?
8040	021046	001406	027424	002260	BEQ		1\$ #MTP001,SUPDOADD	:IS THIS AN 11/44? :BRANCH IF IT IS :SET UP CALLING ADDRESS
8042	021034 021040 021046 021050 021056 021062 021064 021072	004737	027206	002200	CALL	L	SUPD03	
8043 8044	021062				1\$: RETU	V	M001	AG IT IN CUREOUSCO MODE
8045 8046	021072 021076	004737 000207	027030		CALL	URN	001	:DO IT IN SUPERVISOR MODE
8047	021100				MT0002: SUBT	IST	< <mt0002 (<="" setup="" td=""><td>COMPLEMENT ADDRESS TEST>></td></mt0002>	COMPLEMENT ADDRESS TEST>>
					*SUBTEST		MT0002 SETUP COMPLEMEN	NT ADDRESS TEST
8048	021100	012737	000002	002274	MOV		#2.REALPAT	SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
8049 8050	021106	012700	040000		MOV		#LAST+2,RO #SIZE,R1 #FIRST,R4	
8051 8052	021116	012704 012705	060000 100001 002452		MOV		#FIRST_R4 #100001_R5	
8050 8051 8052 8053 8054	021112 021116 021122 021126 021132 021134	012737 012700 012701 012704 012705 005737 001005 023737 001013	002452		TST BNE		NOSUPER 2\$	
8055	021134	023737	172252	172254	CMP		SIPAR5, SIPAR6	
8056	021142	001013			BNE		45	

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 2	26-APR-82 09:41	PAGE 220-1
---	-----------------	------------

8057 021144	000404			•	BR	3\$
8058 021146 8059 021154	023737	177652	177654	2\$:	CMP BNE	UIPAR5, UIPAR6
8060 021156	012701	030000		3\$:	MOV	#30000,R1
8061 021162	012700	140000			MOV	#140000,R0
8062 021166 8063 021172	012705 012702	120001		45:	MOV	#120001,R5 #1,R2
8064 021176	010103				MOV	R1,R3
8065 021200	022737	000001	003752		CMP	#1,PROTYP
8066 021206 8067 021210	001406 012737	027456	002260		BEQ	#MTP002, SUPDOADD
8068 021216	004737	027206	002200		CALL	SUPD03
8069 021222	000207			15:	RETURN	MTP002
8070 021224 8071 021232	004737	027030		19:	CALL	SUPD01
8072 021236	000207				RETURN	

; IS THIS AN 11/44? ; BRANCH IF TRUE ; SET UP CALLING ADDRESS

C	ZMSPAO	MS11-L/M/P	MEMORY	DIAG.	MACRO M1113	26-APR-82	09:41	PAGE 222	
---	--------	------------	--------	-------	-------------	-----------	-------	----------	--

MT0002	SETUP CO	OMPLEMEN	T ADDRES	STEST				
8075	021240				MT0003:	ST	MT0003 SETUP 3 XOR 9 W	XOR 9 WORST CASE NOISE TEST>> **********************************
8076 8077 8078 8079 8080 8081	021240 021250 021256 021262 021266 021272	012737 005037 004737 012701 012703	000003 002322 041074 060000 020000	002274	1\$: 2\$:	MOV CLR CALL MOV MOV	AG IS TRUE THEN \$RETURN #3.REALPAT PCBUMP FLIPWARN #FIRST,R1 #20000,R3 #-8.R3	:SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY :TRAPS DO NOT ADD TO PC :SETUP WARNING CONSTANTS & R2 :R1 < STARTING ADDRESS
8078 8079 8080 8081 8082 8083 8084 8085 8086 8087 8088 8089 8090 8091 8092 8093 8094 8095 8096 8097 8098	021240 021256 021256 021262 021266 021272 021276 021302 021306 021312 021320 021324 021332 021336 021346 021352 021354 021352 021354 021356 021370 021370	012701 012703 072327 012702 012705 022737 001415 104415	020000 177770 000004 000100 000001	003752		ASH MOV MOV CMP BEQ SAVREG	#4.R2 #64R5 #1.PROTYP 3\$:R3 < R3 / 256. :SMALL LOOP SIZE :MEDIUM LOOP SIZE :IS THIS AN 11/44? :BRANCH IF IT IS
8088	021324	012737	027510 027206	002260		MOV CALL RESREG	WMTPA03, SUPDOADD SUPDO3	;DO IT IN MAIN MEMORY
8090 8091 8092 8093	021346 021346 021352	104416 012737 004737 000442	027550 027222	002260	3\$:	MOV CALL BR BMOV	#MTPB03, SUPDOADD SUPD04 4\$ MTPA03	
8095 8096 8097 8098 8098	021410	104415 004737	027030		30.	SAVREG CALL BMOV BMOV BMOV	SUPDO1	
8100 8101 8102 8103	021422 021430 021436 021444 021452 021454	012737 012737 012737 012737	172360 172260 177644 001032	177642 172374 172276 172272		MOV MOV MOV	MTPC03,KDPAR0,8. MTPD03,SDPAR0,8. WKDPAR0,UIPAR1 WSDPAR0,KDPAR6 WUIPAR2,SDPAR7 W1032,SDPAR5	; SET UP PAR LINKS ; CHANGE INST TO BR .+66 (BR TO KDPAR1)
8107	021460	104416 004737 022737	027044 000003	002602	45:	RESREG CALL CMP	SUPDO2 #3,FLIPLOC	;DONE WITH 4 PATTERNS ;[(0,177777);(177777,0);(401,177777);(177777,401)]? ;NO - LOOP
8109 8110	021466 021470	001275 000207				BNE RETURN		, NO - LOOP
8111	021472				MT0004:		MT0004 SETUP ROTATING	******
8112 8113 8114	021472 021500 021506	012737 012737 013702 004737 012700	000004 000004 002600 041224 060000 040000 000001	002274 002322	•	MOV MOV MOV	#4.REALPAT #4.PCBUMP ONES.R2 BACKGND #FIRST.RO #SIZE.R1 #1.PROTYP	SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY TRAPS ADD 4 TO PC WRITE BACKGROUND OF ONES
8116 8117 8118 8119	02 516 0 522 021526	012700 012701 012737	060000 040000 000001	003752		MOV MOV CMP	#FIRST,RO #SIZE,R1 #1,PROTYP	
8119 8120 8121 812	021534 021536 021544 021550	012701 022737 001406 012737 004737 000207	027646 027222	002260		BEQ MOV CALL RETURN	MMTPA04, SUPDOADD SUPDO4	; IS THIS AN 11/44? ; BRANCH IF IT IS ; SET UP LINKS
8123 8124 8125	021472 021500 021506 021512 021516 0 522 021526 021536 021536 021550 021552 021560 021572	012737	172360	177652	15:	BMOV BMOV MOV	MTPA04 MTPB04,KDPAR0,8. #KDPAR0,UIPAR5	

CZMSPAO MT0004	MS11-L SETUP	/M/P MEMO	RY DIAG. ZEROS TE	MACRO M	1113 26	-APR-82	09:41 PAG	E 222-	-1	
8126 8127	021600	012737 004737 000207	177654 027044	172376		MOV CALL	#UIPAR6,	DPAR7		
8128	021614	000207			MT0005:		< <mt0005< td=""><td>*****</td><td></td><td>****</td></mt0005<>	*****		****

00020.			MT0005: SUBTST	< <mt0005 r<="" setup="" th=""><th>OTATING ONES TEST>></th></mt0005>	OTATING ONES TEST>>
			*SUBTEST	MT0005 SETUP ROTATING	ONES TEST
012737	000005 000004	002274 002322	MOV MOV	#5,REALPAT #4,PCBUMP	SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY TRAPS ADD 4 TO PC
004737 012700	041224		CALL	BACKGND #FIRST,RO	;WRITE BACKGROUND OF ZEROS
022737	000001	003752	CMP	#1,PROTYP	; IS THIS AN 11/44? ; BRANCH IF IT IS
012/3/	027722 027736	002260 027720	MOV MOV	#MTP005, SUPDOADD #MTP005+14, MTPB04+16	SET UP LINKS
012737 000207	027662	027720	MOV RETURN	#MTPA04+14, MTPB04+16	RESET TEST'S ORIGINAL VALUE
012737 012737 004737 000207	172360 177654 027044	177652 172376	15: BMOV BMOV MOV MOV CALL RETURN	MTP005 MTPB04,KDPAR0,8. #KDPAR0,UIPAR5 #UIPAR6,KDPAR7 SUPD02	
	012737 012737 005002 004737 012700 012701 022737 001414 012737 012737 004737 012737 0012737 0012737 0012737	012737 000005 012737 000004 005002 004737 041224 012700 060000 012701 040000 022737 000001 012737 027722 012737 027736 004737 027662 012737 172360 012737 172360 012737 177654 004737 027044	012737 000005 002274 012737 000004 002322 005002 004737 041224 012700 060000 012701 040000 022737 000001 003752 001414 012737 027722 002260 012737 027736 027720 012737 027662 027720 012737 027662 027720 012737 172360 177652 012737 177654 172376 004737 027044	MT0005: SUBTST **********************************	MT0005: SUBTST

CZMSPAO	MS11-L/M/P MEMORY DIAG. MACRO M11	13 26-APR-82 09:41	PAGE 224
MTOOOS	CETUP POTATING ONES TEST		

8151	021750				MT0006: SUBTST	*****	SETUP INITIAL D	*****	******
815 815 815	2 021750 3 021756 4 021764	012737 012737 012701	000006 000004 002406 027756	002274 002322 002260	MOV MOV MOV MOV	#6,REALPAT #4,PCBUMP #TÉSTADD,R1 #MTP006,SUPDOADD SUPDO3	; SETUP ; TRAPS	PATTERN NUMBER FOR ADD 4 TO PC	TYPEOUT & DISPLAY
8156 8156 8156	6 021776 7 022002 8 022004	012701 012737 004737 000207	027206	002200	CALL RETURN MT0007: SUBTST		;DO IT SETUP ADDRESS E	IN SUPERVISOR MODE	oj.
					*SUBTEST	MT0007 SETUP AD	DRESS BIT TEST		797
815 816		012737 005002 004737	000007	002274	MOV	#7,REALPAT	; SETUP	PATTERN NUMBER FOR	TYPEOUT & DISPLAY
816 816 816	022014 2 022020 3 022024	012701 012702	041224 060000 000001		CALL MOV MOV BIS MOV	BACKGND #FIRST R1	;OF ZER	ROS	
816 816 816 816	5 022032 6 022040 7 022044	050201 012737 004737 000207	030156 027206	002260	CALL	#1,R2 R2,R1 #MTP007,SUPDOADD SUPDO3		IN SUPERVISOR MODE	
816	8 022046				MT0010: SUBTST	******	SETUP BYTE ADDR	********	******
					:*SUBTEST	MT0010 SETUP BY	TE ADDRESSING	************	*******
816 817 817	0 022054	012737 012737 013704	000010 000004 002406	002274	MOV MOV MOV	#10,REALPAT #4,PCBUMP TESTADD,R4	:SETUP :TRAPS	PATTERN NUMBER FOR ADD 4 TO PC	TYPEOUT & DISPLAY
817 817 817	2 022066 3 022074 4 022100	012737 004737 000207	030256 027206	002260	MOV CALL RETURN	TESTADD,R4 #MTP010,SUPDOADD SUPDO3	;DO IT	IN SUPERVISOR MODE	

	CZMSPA0	MS11-L/M/P	MEMORY DIAG. MACRO M1113	26-APR-82 09:41	PAGE 226
ı	MTOO10	CETUD DYTE	ADDDECCING TEST		

	00.0.0				
817	7 022102				MT0011: SUBTST < <mt0011 bit="" create="" error="" setup="" single="" test="">> :*SUBTEST MT0011 SETUP CREATE SINGLE BIT ERROR TEST</mt0011>
818 818 818 818 818 818	31 022126 32 022136 33 022144	012737 012737 004737 000207	000011 030364 027206	002274 002260	IF ACTFLAG IS TRUE OR APTFLAG IS TRUE IF SPASS NE #0 THEN \$RETURN END; OF IF ACTFLAG IF PMEMFLG IS TRUE THEN \$RETURN ; EXIT IF NOT MS11-M MOV #11, REALPAT ; SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY MOV #MTP011, SUPDOADD CALL SUPDO3 ; DO IT IN SUPERVISOR MODE RETURN MT0012: SUBTST < <mt0012 byte="" clears="" sbe="" setup="" test="" write="">> **SUBTEST MT0012 SETUP WRITE BYTE CLEARS SBE TEST</mt0012>
818 818 819 819 819 819 819 819	94 022236	012737 013700 012705	000012 002102 040000	002274	IF ACTFLAG IS TRUE OR APTFLAG IS TRUE IF \$PASS NE #0 THEN \$RETURN END: OF IF ACTFLAG IF PMEMFLG IS TRUE THEN \$RETURN ; IS THIS A MS11-M? MOV #12,REALPAT ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY MOV BANKINDEX,RO IF #BIT12 SET.IN CONFIG+2(RO) MOV #40000,R5 ELSE
819 819 819 819 829	96 022244 97 022250 98 022250 99 022256 00 022262 01 022264	012705 012737 004737 000207	000002 031162 027206	002260	MOV #2,R5 END; OF IF #BIT12 MOV #MTP012,SUPDOADD CALL SUPDO3 ;DO IT IN SUPERVISOR MODE RETURN MT0013: SUBTST < <mt0013 bit="" create="" double="" error="" setup="" test="">> ;***********************************</mt0013>
82 82 82 82 82 82 82 82 82	02 022264 03 022300 04 022310 05 022310 06 022320 07 022326 08 022334 09 022342 10 022346	012737 012737 012737 012737 004737 000207	000013 031550 000003 027206	002274 002260 002074	IF ACTFLAG IS TRUE OR APTFLAG IS TRUE IF \$PASS NE #0 THEN \$RETURN END; OF IF ACTFLAG IF PMEMFLG IS TRUE THEN \$RETURN ;EXIT IF NOT MS11-M MOV #13,REALPAT ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY MOV #3,NOPAR ;INDICATE PARITY ACRION CALL SUPDO3 ;DO IT IN SUPERVISOR MODE RETURN

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 227 MT0013 SETUP CREATE DOUBLE BIT ERROR TEST

82	12 022350				MT0014: SUBTST < <mt0014 basic="" bit="" double="" error="" setup="" test="">></mt0014>
					*SUBTEST MT0014 SETUP BASIC DOUBLE BIT ERROR TEST
82 82	13 022350 14 022364				IF ACTFLAG IS TRUE OR APTFLAG IS TRUE IF \$PASS NE #0 THEN \$RETURN
	15 022374				END; OF IF ACTFLAG
82 82 82 82 82 82 82	17 022404	012737	000014	002274	MOV #14 REALPAT ; SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY CALL MAPKERNAL ; MAP KERNAL SPACE
82	19 022416				LET R1 := #100000 :SETUP TEST ADDRESS CALL GETCSR :GET CSR INFO FROM CONFIGURATION TABLE
82	19 022416 20 022422 21 022426 22 022432 23 022436	004737 004737 004737	041324 032264		CALL MTP014 ;DO BASIC DOUBLE BIT ERROR TEST
82 82	22 022432	000207	046774		CALL UNMAP CERNAL SPACE RETURN

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 229 MT0014 SETUP BASIC DOUBLE BIT ERROR TEST

8226 022440				MT0015: SUBTST < <mt0015 byte="" dbe="" inhibit="" of="" setup="" with="" write="">> :**SUBTEST MT0015 SETUP WRITE INHIBIT OF BYTE WITH DBE</mt0015>
8227 022440 8228 022454 8229 022464 8230 022464 8231 022474 8232 022502 8233 022510 8234 022514	012737 012737 004737 000207	000015 032510 027206	002274 002260	IF ACTFLAG IS TRUE OR APTFLAG IS TRUE IF \$PASS NE #0 THEN \$RETURN END ; OF IF ACTFLAG IF PMEMFLG IS TRUE THEN \$RETURN ; EXIT IF NOT MS11-M MOV #15, REALPAT ; SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY MOV #MTP015, SUPDOADD CALL SUPDO3 ; DO IT IN SUPERVISOR MODE RETURN MT0016: SUBTST < <mt0016 dbe="" inhibit="" of="" setup="" with="" word="" write="">></mt0016>
8235 022516				**SUBTEST MT0016 SETUP WRITE INHIBIT OF WORD WITH DBE
8236 022516 8237 022526 8238 022542 8239 022552 8240 022552 8241 022560 8242 022566 8243 022572	012737 012737 004737	000016 033254 027206	002274 002260	IF PMEMFLG IS TRUE THEN SRETURN ; EXIT IF NOT MSTT-M IF ACTFLAG IS TRUE OR APTFLAG IS TRUE IF SPASS NE #0 THEN SRETURN END ; OF IF ACTFLAG MOV #16, REALPAT ; SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY MOV #MTP016, SUPDOADD . CALL SUPDO3 ; DO IT IN SUPERVISOR MODE
8243 022572 8244 022574	000207			RETURN MT0017: SUBTST < <mt0017 &="" 0's="" 1's="" holding="" setup="">> :***********************************</mt0017>
8245 022574 8246 022602 8247 022610 8248 022614	012737 012737 004737 000207	000017 034036 027206	002274 002260	MOV #17, REALPAT ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY MOV #MTP017, SUPDOADD CALL SUPDO3 ;DO IT IN SUPERVISOR MODE RETURN

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113	26-APR-82 09:41	PAGE 231
---	-----------------	----------

8251	022616				MT0020: SUBTST < <mt0020 bit="" csr="" error="" on="" setup="" single="" syndromes="" to="">></mt0020>
					*SUBTEST MT0020 SETUP SYNDROMES TO CSR ON SINGLE BIT ERROR
8252	022616				IF ACTFLAG IS TRUE OR APTFLAG IS TRUE IF SPASS NE #0 THEN SRETURN
8253 8254 8255 8256 8257 8258 8259 8261 8261	022642	012777	000020	002274	END; OF IF ACTFLAG MOV #20, REALPAT ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
8256	022650	012737		002274	IF PMEMFLG IS FALSE THEN SRETURN ; EXIT IF NOT MS11-P
8257	022664	004737	046706		LET R1 := #100000 ;SETUP TEST ADDRESS
8259 8260	022670	004737 004737 004737 000207	041324 034114 046774		CALL MTPO20 ; DO SYNDROMES TO CSR ON SINGLE ERROR TEST
8261	022700	004737	046774		CALL UNMAP ; UNMAP KERNAL SPACE RETURN

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 232 MT0020 SETUP SYNDROMES TO CSR ON SINGLE BIT ERROR

8264	022706				MT0021:	******	******	SETUP MARCHING O'S & 1'S TEST>> RCHING O'S & 1'S TEST
8265 8266 8267 8268 8269 8270	022706 022714 022722 022726 022732 022734 022736 022742	012737 013702 004737 010203 000303 012701 010105	000021 002616 041224	002274		SET NOSO MOV MOV CALL MOV SWAB	#21,REALPAT BAKPAT,R2 BACKGND R2,R3 R3	SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
8271 8272 8273 8274 8275	022750	012704	160000 060000 000001	003752		MOV MOV CMP BEQ	#LAST+2,R1 R1,R5 #FIRST,R4 #1,PROTYP	:IS THIS AN 11/44? :BRANCH IF IT IS :IS THIS AN 11/24? :BRANCH IF SO
8276 8277 8278 8279	022766 022766 022770 022776	001441 022737 001407 022737 001003 012701	000003 000007 140000	003752		CMP BEQ CMP BNE MOV	#7,BANK 3\$ #140000 P1	
8280 8281 8282 8283 8284 8285 8286 8287 8288 8290 8291 8292 8293	023000 023004 023006 023014 023020 023026 023032 023034 023046 023054 023060 023060 023070 023070 023070 023100 023100	010105 012737 004737 012737	034430 027206 034460 027222	002260 002260	3\$:	MOV CALL MOV CALL	R1,R5 #MTPA21,SUPDOADD SUPDO3 #MTPB21,SUPDOADD SUPDO4 R4,R1	
8286 8288 8289 8290 8291	023032 023034 023042 023046 023054	010401 012737 004737 012737 004737 000434 022737 001003 012701	034514 027222 034550 027222	002260 002260		MOV MOV CALL MOV CALL BR	#MTPC21, SUPDOADE SUPDO4 #MTPD21, SUPDOADE SUPDO4 2\$	
8292 8293 8294 8295 8296	023062 023070 023072 023076 023100	022737 001003 012701 010105	000177 140000	002100	15:	CMP BNE MOV MOV BMOV	#177,BANK 4\$ #140000,R1 R1,R5 MTPA21 SUPD01	
8296 8297 8298 8299 8300	023106 023112 023120	004737	027030			BMOV CALL	SUPDO1 MTPB21 SUPDO2	
8298 8299 8300 8301 8302 8303 8304 8307 8308 8309	023124 023126 023134	010401 004737	027044			MOV BMOV CALL	R4,R1 MTPC21 SUPD02	
8306 8307 8308 8309	023140 023146 023152 023156	004737 005037 000207	027044 002434		2\$:	BMOV CALL CLR RETURN	MTPD21 SUPD02 NOSCOPE	

CZMSPAO MT0021	MS11-L/	M/P MEMO ARCHING	RY DIAG.	MACRO M	11113 26-APR-82 09:41 PAGE 233	021
8311	023160				MT0022: SUBTST < <mt0022 &="" diagonal="" refresh="" setup="" shifting="" test="">> :*SUBTEST MT0022 SETUP REFRESH & SHIFTING DIAGONAL TEST</mt0022>	
8312 8313 8314 8315 8316 8317	023160 023164 023170 023176 023204 023210	004737 012737 012737 004737 000207	026774 000022 034600 027206	002274 002260	CALL KAMITEST ON.ERROR THEN \$RETURN MOV #22,REALPAT ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLANCE OF THE SUPPOSE SUPPOSE ;DO IT IN SUPERVISOR MODE RETURN CALL SUPPOSE ;DO IT IN SUPERVISOR MODE	AY
8518	023212				MT0023: SUBTST < <mt0023 diagonal="" shifting="" test="">> ;***********************************</mt0023>	
8320 8321 8322 8323 8324 8325 8326 8327	023212 023216 023222 023230 023236 023244 023250 023254	004737 012737 012737 004737 005037 000207	026774 000023 034600 027206 002002	002274 002260	CALL KAMITEST ON.ERROR THEN \$RETURN MOV #23,REALPAT SETUP PATTERN NUMBER FOR TYPEOUT & DISPL MOV #MTPO22,SUPDOADD SET DIAGFLAG CALL SUPDO3 CLR DIAGFLAG RETURN CALL SUPDO3 CLR DIAGFLAG RETURN CHECK FOR KAMIKAZE MODE SET UN KAMIKAZE MODE RETURN SETUP PATTERN NUMBER FOR TYPEOUT & DISPL SETUP PATTERN NUMBER FOR TYPEOUT & DIS	AY

M10023	2HTL LTM	6 DIAGUNA	AL IEST				
8329	023256		4				ETUP FAST GALLOPING PATTERN TEST>> T GALLOPING PATTERN TEST
8330 8331	023256 023262 023266 023274	004737	026774		CALL ON.	RROR THEN SRETURN	CHECK FOR KAMIKAZE MODE : IF NOT IN KAMIKAZE MODE RETURN
8332 8333 8334 8335 8336 8337	023266 023274 023302 023306 023312 023314 023316 023320 023324 023330 023336 023340 023356 023360 023364 023366 023374 023376 023404 023412 023414	012737 013702 004737 010203 010304 000304	000024 002616 041224	002274	MOV MOV CALI MOV MOV SWAE	#24,REALPAT BAKPAT,R2 BACKGND R2,R3 R3,R4 B R4	CHECK FOR KAMIKAZE MODE ; IF NOT IN KAMIKAZE MODE RETURN ; SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
8339 8340 8341	023320 023324 023330	012701 012705 022737 001417 022737 001406 022737 001002 012705 104415 012737 000440 022737 001002	060000 157776 000001	003752	MOV MOV CMP BEQ CMP	WFIRST R1 WLAST R5 W1,PROTYP	
8342	023336	022737	000003	003752	CMP	#3.PROTYP	
8344 8345	023346	001406	000007	002100	BEQ	#7.BANK	
8346 8347	023356	001002 012705	137776		BNE	3\$ #137776,R5	
8348 8349	023364	104415	035314	002260	3\$: SAVI	MMTPB24, SUPDOADD	
8350 8351	023374	000440 022737	000177	002552	1\$: BR	#177,LASTBANK	
8352 8353	023404	001002 012705 104415	137776		BNE	4\$ #137776,R5	
8354 8355	023412	104415			45: SAVI	MTDA74	
8356 8357	023422			The state of	BMO'	MTPB24,SDPARO,8. MTPC24,KDPARO,8.	
8358 8359 8360 8361	023434 023446 023454 023462 023470 023476	012737 012737 012737 012737	172260 172260 172360 177660	002260 177676 172272 172374	MOV MOV MOV 2\$: CAL	#SDPARO,UDPAR7 #KDPARO,SDPAR5 #UDPARO,KDPAR6	SET UP PAR LINKS
8363	OLS VIO	004131	02.222				NT DATA
8365 8366 8366 8366 8366 8370	023502 023504 023506 023510 023514 0023520 023522	104416 000302 000303 004737 005037 000207	027222 002434		RES SWA SWA CAL CLR RET	REG B R2 B R3 L SUPDO4 NOSCOPE	
8371	023522	000207			MT0025: SUB	TST < <mt0025< td=""><td>ETUP INTERRUPT ENABLE TEST>></td></mt0025<>	ETUP INTERRUPT ENABLE TEST>>
					*SUBTEST	MT0025 SETUP IN	TERRUPT ENABLE TEST
837	023522				IF	ACTFLAG IS TRUE OR APT	FLAG IS TRUE
837	2 023522 3 023536 4 023546 5 023546				END	OF IF ACTFLAG	RETURN : EXIT IF NOT MS11-P
837	5 025556	012737	000025 035346 027206	002274 002260	MOV	#25 REALPAT	RETURN : EXIT IF NOT MS11-P :SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY :DO IT IN SUPERVISOR MODE
837 837	7 023564 8 023572 9 023576	012737 012737 004737 000207	027206	002200	CAL	L SUPDO3	;DO IT IN SUPERVISOR MODE

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 236 MT0025 SETUP INTERRUPT ENABLE TEST

8382	023600				MT0026:	SUBTST	< <mt0026< th=""><th>SETUP F</th><th>RANDOM DATA</th><th>TEST>></th><th>*******</th></mt0026<>	SETUP F	RANDOM DATA	TEST>>	*******
					*SUBTE	ST	MT0026 SETUP RA	ANDOM DA	TA TEST		************
8383 8384 8385 8386 8387	023600 023606 023612 023616 023622	012737 005037 013703 013702 010305	000026 002322 002570 002566	002274		MOV CLR MOV MOV MOV	#26,REALPAT PCBUMP SEEDLO,R3 SEEDHI,R2 R3,R5		Management of the control of the con	NOT ADD TO TE RANDOM NUM	
8388 8389 8390 8391 8392 8393	023600 023606 023612 023616 023622 023624 023626 023636 023636 023646 023656 023656 023672 023672 023672 023710 023710	012737 013703 013703 013702 010305 010204 012701 012701 012737 001437 0022737 001406 022737 001406 022737 0012737 012737 012737 012737 012737 012737 012737 012737 012737 012737 012737 012737 012737 012737 012737 012737 012737 012737 012737	060000 020000 000001 060003	003752 003752		MOV MOV CMP BEQ CMP BEQ	R2,R4 WFIRST,R1 WSIZE/2,R0 W1,PROTYP 1\$ W3,PROTYP 3\$:DO WE HAY :BRANCH IF :11/24? :BRANCH IF	/E AN 11/44? WE DO	
8395 8396 8397	023656 023664 023666 023672	022737 001002 012700 104415	000007 014000	002100	3\$:	CMP BNE MOV SAVREG	#7,BANK 3\$ #14000,R0				
8399 8400 8401 8402 8403	023674 023702 023710 023714 023720	012737 012737 004737 005037 012737	036020 036014 027206 036044 036034 036030	036120 002260 036120 002260	30.	MOV MOV CALL CLR MOV	#MTPA26+4,MTPD26 #MTPA26,SUPDOAD6 SUPDC3 RANODD #MTPB26+4,MTPD26 #MTPB26,SUPDOAD6		FOR ERROR	R REPORTING	
8404 8405 8406 8407 8408 8409	023726 023734 023736 023742	012737 104416 004737 000452 022737	036030 027206 000177	002260	15:	MOV RESREG CALL BR CMP BNE	WMTPB26, SUPDOADS SUPDO3 2\$ #177,BANK 4\$	D			
8410 8411 8412 8413	023754 023760 023762	012700 104415	014000		48:	MOV SAVREG BMOV BMOV	#14000,R0		;WRITE ROI :RANDOM SI	UTINE TO FAS	T MEMORY FAST MEMORY
8414	024002	012737		172376		MOV BMOV MOV	MTPA26 MTPC26,KDPAR0,8 #730,KDPAR7 MTPD26,SDPAR0,8 #KDPARO,UIPAR1		:WRITES '1	BR116" IN UBSUBPROGRAM	T MEMORY FAST MEMORY (BR SDPARO) TO FAST MEMORY
8417 8418 8419	024010 024030 024036 024042 024046 024054 024062 024064 024070 024074	012737 012737 004737 005037	172360 177644 027030 036044	177642 172274		MOV CALL CLR BMOV	#KDPARO, UIPAR1 #UIPAR2, SDPAR6 SUPDO1 RANODD MTPB26		;WRITE RAI ;FOR ERROI ;READ ROU	NDOM DATA R REPORTING TINE TO FAST AR LINK	MEMORY
8421 8422 8423	024054	012737 104416 004737	172360	177642		MOV RESREG	WKDPARO,UIPAR1		; SET UP P		
8424 8425 8426	024070 024074 024100	012737 104416 004737 010337 010237 000207	027030 002570 002566		2\$:	CALL MOV MOV RETURN	R3, SEEDLO R2, SEEDHI		UPDATE F	OR NEW RANDO	M NUMBERS

```
MT0027: SUBTST <<MT0027
                                                                                          UNIQUE BANK TEST>>
8429 024102
                                                ****************
                                                                     MT0027 UNIQUE BANK TEST
                                                : *SUBTEST
                                                           MAKE SURE THAT EACH BANK CAN HAVE UNIQUE DATA WRITE AND READ THE BANK NUMBER IN EACH BANK (EXCEPT WHERE THE PROGRAM IS)
                012737
104502
022737
001404
012737
000414
                                                                                                     SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
                                                                     #27, REALPAT
                           000027
                                     002274
                                                           MOV
                                                                                          :CLEAR CSRS
                                                           CLRCSR
      024112
024120
024122
024130
024132
024140
024146
                                                                                                     : IS THIS AN 11/44?
#1,PROTYP
                           000001
                                     003752
                                                           CMP
                                                                                                     BRANCH IF TRUE
                                                           BEQ
                                                                     #SUPDO3,LINK1
STAR27
                                                                                                     :SET UP LINK
                                                           MOV
                                     002516
                           027206
                                                                                                     BRANCH TO RUN
                                                           BR
                                                                     MTP034
                                                           BMOV
                                                WARN7:
                                                          MOV
                                                                     #UIPAR3, SUPDOADD
                                      002516
                                                                     #SUPDO1,LINK1
                                                                                                    :SET UP LINK
                                                           MOV
                                                           SET
                                                                NOF SMODE
                                                STAR27: FOR I := #1 TO #2
FOR BANK := #0 TO LASTBANK
                                                                CALL EXBANK
IF ACFLAG IS TRUE AND RRFLAG IS FALSE
                 004737
                           047020
                                                                                                     : INVALIDATE BACKGROUND PATTERN ON 'BANK'
                                                                   INVALIDATE
                 104511
                                                                   LET R2 := BANK
                 012700
010004
012701
010103
                                                                   MOV
                                                                                #FIRST,RO
                           060000
                                                                                RO,R4
#SIZE,R1
R1,R3
                                                                   MOV
                                                                   MOV
                           040000
                                                                   MOV
                                                                   IF I EQ #1
                 022737
001403
012737
004777
                                                                     CMP
                                                                                #1,PROTYP
                           000001 003752
                                                                     BEQ
          256
264
270
                                                                                #MTP034, SUPDOADD
                           036340
156226
                                                                     MOV
                                      002260
                                                                                aLINK1
                                                                2$: CALL
                                                                   END : OF
                 022737
001403
012737
004737
                           000001
                                                                                #1,PROTYP
                                      003752
                                                                      BEQ
                                                                      MOV
                                                                                #MTP034+6,SUPDOADD
                           036346
027206
                                      002260
                                                                                SUPDO3
                                                                3$: CALL
                                                           END OF IF
END OF FOR BANK
END OF FOR I
IF FS7FLAG IS TRUE
                                                                     NOF SMODE
                 005037
000207
                            002422
                                                              RETURN
                                                           END : OF IF FS7FLAG
FOR I := #1 TO #2
                                                              FOR BANK := LASTBANK DOWNTO #0
                                                                CALL EXBANK
                 004737
                           047020
                                                                 IF ACFLAG IS TRUE AND RRFLAG IS FALSE
LET R2 := BANK
                 005102
012700
010004
012701
010103
                                                                   COM
                                                                                #FIRST,RO
                            060000
                                                                   MOV
                                                                                RO R4
#SIZE R1
R1 R3
                                                                   MOV
                                                                   MOV
                            040000
                                                                   IF I EQ #1
                                                                                #1,PROTYP
                  022737
                            000001 003752
```

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 238-1

8483 024462 001403
8484 024464 012737 036340 002260
8485 024472 004777 156020

8486 024476
8487 024476
8488 024506 022737 000001 003752
8489 024514 001403
8490 024516 012737 036346 002260
8491 024524 004737 027206

8492 024530
8493 024530
8494 024530
8495 024544
8496 024560 005037 002422

8497 024564 0000207
```

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 240
MT0027 UNIQUE BANK TEST
                                                                                                   SETUP FLUSH OUT DBE'S TEST>>
                                                      MT0030: SUBTST <<MT0030
    8500 024566
                                                       ***************
                                                       : *SUBTEST
                                                                             MT0030 SETUP FLUSH OUT DBE'S TEST
    8501 024566
8502 024572
8503 024600
8504 024606
8505 024614
8506 024622
8507 024624
8508 024632
                                                                             PASFLG
                     005037
                                 002262
                                                                             FULLREL
#30, REALPAT
                                                                  SET
                                                                                                              SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY INDICATE COUNT PARITY ERRORS
                     012737
012737
022737
001007
                                           002274
002074
003752
                                                     MTA030: MOV
                                000030
000001
000001
                                                                             #1,NOPAR
                                                                  MOV
                                                                             #1,PROTYP
                                                                             MTP030
                     012737
000406
012737
012737
104470
                                 027030 002516
                                                                  MOV
                                                                             #SUPDO1,LINK1
    8509 024640
                                                                             #SUPDO3,LINK1
#MTP030,SUPDOADD
                                 027206 002516
036122 002260
                                                                  MOV
          024642
    8510
                                                                             NOFSMODE, NOSCOPE DISABLE ERROR CORRECTION
          024650
                                                                  MOV
    8511
                                                                  ECCDIS
                                                       15:
    8512
8513
          024660
024674
024700
024704
024712
024736
024736
024742
024742
024742
024764
024764
                                                                  FOR BANK := #0 TO LASTBANK
                                                                     CALL EXBANK
                     004737 047020
                                                                     IF MKFLAG IS TRUE
                                                                        IF ACFLAG IS TRUE AND RRFLAG IS FALSE
                                040000
060000
155554
                                                                                        #SIZE,R1
                     012701
012700
004777
                                                                                         #FIRST,RO
                                                                          MOV
                                                                                         aLINK1
                                                                           CALL
                                                                     END OF IF ACFLAG
                                                                  END ; OF FOR
                                                                  IF PASFLG IS FALSE
SET PASFLG
                                                                     CLRCSR
                                                                                                               :CLEAR CSRS
                      104502
004737 045172
                                                                     CALL RELOCATE
ON. ERROR
                                                                                                    ; TRAP ON DOUBLE BIT ERRORS (NORMAL)
                                                                        ECCINIT
           025002
                      104472
           025002
025004
025022
025022
025022
025030
025034
025040
025042
                                                                                        NOF SMODE, NOSCOPE, FULLREL
                                                                        CLEAR
                                                                        RETURN
                      000207
                                                                     END : OF ON . ERROR
                                 002304
047020
024600
                                                                     MOV NEWBANK, BANK
CALL EXBANK
CALL MTA030
                      013737
004737
004737
104472
004737
000207
                                           002100
                                                                                                   :TRAP ON DOUBLE BIT ERRORS (NORMAL)
                                                                     ECCINIT
    8536
8537
                                                                      CALL UNRELOCATE
                                 046056
                                                                     RETURN
     8538
                                                                  END : OF IF PASFLG
     8539
                                                                   ECCINIT : TRAP ON DOUBLE BIT ERRORS (NORMAL)
CLEAR NOFSMODE, NOSCOPE, FULLREL
            025050
                      104472
           025066
                      000207
                                                                   RETURN
```

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 242 MT0030 SETUP FLUSH OUT DBE'S TEST

8545	025070				MT0031:	SUBTST	< <mt0031 s<="" th=""><th>ETUP SOB</th><th>B-A-LONG T</th><th>EST>></th><th>******</th><th>*******</th><th>***</th></mt0031>	ETUP SOB	B-A-LONG T	EST>>	******	*******	***
					*SUBTES	T	MT0031 SETUP SOB	-A-LONG	TEST			***********	***
8546 8547	025070 025074	004737	026774			CALL ON FRROM	KAMITEST THEN SRETURN				AZE MODE R		
8546 8547 8548 8559 8551 8555 8553 8555 8555	025074 025100 025106 025114 025120 025134 025142 025154	012737 005037	000031 002074	002274		TESTAREA	NOSCOPE #31,REALPAT NOPAR BANK MTP031,FIRST,SOBL	FNGTH/2	SETUP PAT SETUP PAR MAP FIRST ENTER TES	TERN N ITY AC SO BL T MODE	UMBER FOR TION OCK MOVE W	TYPEOUT & DI	ISPLAY
	025154 025156 025163	104417 013702 010200	002556			MOV	SORK R2	:	ENTER KER				
8556 8557 8558 8559 8560 8561 8562 8563 8564 8565	025162 025164 025170 025174 025202 025210 025214 025216	010200 012701 012705 012737 012737 005737 001005 023737	100776 060056 060002 160000 002452	002260 002516		MOV MOV MOV TST BNE CMP	R2,R0 #100776,R1 #FIRST+SOBLENGTH, #FIRST+2,SUPDOADD #LAST+2,LINK1 NOSUPER 1\$	R5	; COMPLEMEN	T OF I	NSTRUCTION	N "SOB RO,DO"	r"
8563 8564 8565	025226	THE LATE	172252	172254		BEQ BR	SIPAR5,SIPAR6 28 38						
8566 8567 8568	025230	023737	177652	177654	1\$:	CMP BNE	UIPAR5,UIPAR6						
8568 8569 8570 8571	025230 025236 025240 025246 025252 025256	000407 023737 001003 012737 004737 005037 000207	140000 027222 002434	002516	2\$: 3\$:	MOV CALL CLR RETURN	#140000,LINK1 SUPDO4 NOSCOPE						

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 2	CZMSPAC	MS11-L/M/P	MEMORY DIAG.	MACRO M1113	26-APR-82 09:41	PAGE 24
--	---------	------------	--------------	-------------	-----------------	---------

MT0031	SETUP S	OB-A-LON	TEST							
8574	025260				MT0032: S	*****	MT0032 SETUP WRITE	***	RITE RECOVERY TEST>> ***********************************	
8575 8576	025260 025264	004737	026774		C	ALL N. ERROF	KAMITEST R THEN SRETURN		:CHECK FOR KAMIKAZE MODE :IF NOT IN KAMIKAZE MODE RETURN	
8578 8578 8579 8580	025264 025270 025276 025304 025310	012737 005037	000032 002074	002274	M	ET IOV LR IAP	NOSCOPE #32,REALPAT NOPAR BANK		SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY SETUP PARITY ACTION MAP FIRST SO THAT THE BLOCK MOVE WORKS OP CODE OF INSTRUCTION 'MOV R2,-(PC)'' OP CODE OF COMPLEMENT OF INSTRUCTION 'JMP USED FOR 1/2 BANK LOOP	
8581 8582 8583 8584 8585 8586 8586	025304 025310 025324 025330 025334 025340 025344 025360 025364 025366 025374 025376 025400 025410 025414 025420 025424	012700 012701 012702 010237 012703 012704 005037 005737 001005 023737	010247 177667 020000 002516 060000 160000 002520 002452			IOV IOV IOV IOV IOV ILR IST	NOPAR BANK #10247,R0 #177667,R1 #SIZE/2,R2 R2,LINK1 #FIRST,R3 #LAST+2,R4 LINK2 NOSUPER 1\$		OP CODE OF INSTRUCTION MOV R2,-(PC) OP CODE OF COMPLEMENT OF INSTRUCTION 'JMP USED FOR 1/2 BANK LOOP	(RO)**
8590 8591 8592	025366 025374 025376	023737 001405 000415 023737	172252		6	MP BEQ BR	SIPAR5, SIPAR6 2\$ 3\$			
8593 8594 8595 8596	025400 025406 025410 025414	023737 001011 012704 012702 010237 012737		177654	2\$:	MP BNE 10V 10V	UIPAR5,UIPAR6 3\$ #14000,R4 #14000,R2			
8597 8598 8599	025420 025424	010237 012737	002516	002520		10V 10V	R2,LINK1 #1,LINK2			
8600	025432					MOVE T	EST TO MEMORY UNDER	TEST	;ENTER TEST MODE	
8602 8603 8604	025440 025442 025444	010023 010144 077203			45:	10V 10V 808	RO,(R3)+ R1,-(R4) R2,4\$			
8606 8607 8608	025446 025454	022737 001003	000001	003752		MP BNE MOVE L	#1,PROTYP 5\$ AST PART OF TEST TO	FAST	CITY	
8609 8610	025456	104417			58:	BMOV KERNEL	MTP032		;ENTER KERNEL MODE	
8612 8613 8614 8615 8616	025466 025472 025476 025502 025510	012702 012700 012701 012737 005737	005141 025610 160000 060000 002520	002260		MOV MOV MOV TST REQ	#5141,R2 #10\$,R0 #LAST+2,R1 #FIRST,SUPDOADD LINK2 6\$		OP CODE OF INSTRUCTION "COM -(R1)" ADDRESS TO RETURN TO IN RO TOP OF BANK	
8618 8619 8620 8621 8622 8623	025516 025522 025526 025536 025536 025542	010023 010144 077203 022737 001003 104417 012702 012701 012737 005737 001402 012701 004737 012703 012703 012704 005737 001402 012703 012703 012703 012703 012703 012703 012703 012703	140000 027222 020000 000110 060000 002520		6\$:	BEQ MOV CALL MOV MOV TST	#140000,R1 SUPD04 #SIZE/2,R3 #110,R5 #FIRST,R4 LINK2			,
8624 8625 8626 8627	025550 025554 025554 025562	012703 022737 001406	014000 000001	003752	7\$:	BEQ MOV CMP BEQ	#14000_R3 #1_PROTYP 8\$			

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 244-1

86	28 02 329 02	5564 5572	012737 004737	036210 027222 027044	002260		MOV	#MTP032,SUPDOADD SUPD04
86	550 02 531 02	5600	000402	027044		8\$:	CALL	SUPD02

8632 025604 005037 002434 8633 025610 000207 8634 8635 CLR

NOSCOPE ; THIS RETURN ACTS AS A NORMAL RETURN FROM MT0032 ; ALSO A RETURN FROM THE 'CALL SUPDO4' ABOVE

#1,PROTYP

022737

026112

1											SEQ 022
	CZMSPAO MT0034	MS11-L/I	M/P MEMO ROR - BA	RY DIAG. CKGROUND	MACRO M PATTERN	1113 TEST	26-APR-82	09:41 PAGE	246-1		
	8689 8690 8691	026120 026122 026130	001406 012737 004737 000402 004737 052765	036340 027206	002260		BEQ MOV CALL	3\$ #MTP034,SUI SUPD03	PDOADD		
	8692 8693 8694 8695 8696	026120 026122 026130 026134 026136 026142 026150 026150	000402 004737 052765 000207	027030 020000	002652	3\$: 4\$:	BR CALL BIS END ; OF RETURN	4\$ SUPDO1 #BIT13,CON IF #BIT13	F1G+2(R5)	WRITE IT :VALIDATE IT	
	8697 8698	026152				****	SS: SUBTST	< <mt0035 MT0035 SE</mt0035 	*****	WORST CASE NOISE PARITY TEST>> ASE NOISE PARITY TEST	*****
	8699 8700 8701	026152 026160 026164	012737 013703 016301	000035 002102 002650	002274	;****	MOV MOV MOV SWAB	#35,REALPA BANKINDEX, CONFIG(R3) R1	T R3 ,R1	SET UP TEST NUMBER FOR DISPLAY	
	8699 8700 8701 8702 8703 8704 8705 8706 8707 8708 8709	026152 026160 026164 026170 026172 026176 026200 026204 026212 026214 026216	012737 013703 016301 000301 042701 006301 010137 023737	177760 002150 002150	002526		BIC ASL MOV CMP	#^C17,R1 R1 R1.CSRNO	SR		
	8707 8708 8709 8710	OFOFFE	023737 001001 000207 012702 004737 012737 004737	052524 041224 036364 027206		15:	RETURN MOV CALL	CSRNO, PGMC 1\$ #52524,R2 BACKGND #MTP035,SU SUPD03		; WRITE BACKROUND OF ALMOST ALT. 1	S AND O'S
	8712 8713 8714	026226 026234 026240 026250 026252			002260		COM	AG IS TRUE	THEN SRETUR	N ;WRITE COMPLEMENT PATTERN INTO MUT	
	8716 8717	026256	005102 004737 004737 000207	041224 027222			CALL CALL RETURN	BACKGND SUPDO4			

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 247 MT0035 SETUP WORST CASE NOISE PARITY TEST

8719 026264	MT0036: SUBTST < <mt0036 code="" correction="" setup="" test="">> :*SUBTEST MT0036 SETUP CORRECTION CODE TEST</mt0036>
8720 026264 8721 026274 012737 000036 002274 8722 026302 004737 041324 8723 026306 005037 002262 8724 026312 005000 8725 026314 012701 100000 8726 026320 004737 046706 8727 026324 004737 036526 8728 026330 004737 046774 8729 026334 000207	IF PMEMFLG IS FALSE THEN \$RETURN : IF NOT MS11-P THEN EXIT MOV #36, REALPAT : SET UP TEST NUMBER FOR TYPEOUT AND DISPLAY CALL GETCSR : GET CSR INFO FROM CONFIG TABLE CLR PASFLG : CLEAR LOOP COUNTER CLR RO : GET TEST DATA MOV #100000,R1 : GET FIRST ADDRESS IN BANK CALL MAPKERNAL : MAP KIPARS AND 6 TO BANK CALL MTP036 : EXECUTE TEST CALL UNMAP : REMAP KERNAL SPACE

CZMSPAO MT0036	MS11-L/M SETUP CO	M/P MEMO	RY DIAG.	MACRO M	11113 26-APR	-82 09:41	PAGE	248						SEG	0222
8731	026336				MT0037: SUB	*****	*****	*****	*****	*****	TEST>>	******	******	****	
8732 8733 8734 8735 8736 8737 8738 8739 8740 8741	026336 026346 026354 026360 026362 026366 026372 026376 026402	012737 012701 005000 004737 004737 004737 004737 000207	000037 100000 046706 041324 036752 046774	002274	MOV CLR CAL CAL CAL	PMEMFLG I #37. #100 RO L MAPK L GETO L MTPO	S FALSI REALPA 0000,R1 (ERNAL SR	E THEN		RETURN SETUP PA SET UP TO CLEAR DA MAP THIS	IF NOT A ATTERN AA TEST ADDR ATA TO BE S TEST TO INFO FROM CC DISABL ERNEL SPA	WRITTEN KERNEL S CONFIG T	********* FOR TYPEOU PACE ABLE	**** T AND	DISPLAY
8742	026404				MT0040: SUE	TST < <mi< td=""><td>0040</td><td>>:</td><td>></td><td>******</td><td>******</td><td>******</td><td>*******</td><td>****</td><td></td></mi<>	0040	>:	>	******	******	******	*******	****	
8744	026404	000207			*SUBTEST	MT00	040	*****	*****	******	*****	******	******	****	

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 249

8746 026406				MT0041: SUBTST < <mt0041 address="" bit="" csr="" double="" error="" on="" setup="" test="" to="">></mt0041>	
				*SUBTEST MT0041 SETUP ADDRESS TO CSR ON DOUBLE BIT ERROR TEST	
8747 026406				IF PMEMFLG IS FALSE THEN SRETURN : EXIT IF NOT MS11-P MOV #41, REALPAT ; SETUP PATTERN AND NUMBER FOR TYPEOUT AND DISPLAY	
8747 026406 8748 026416 8749 026424 8750 026430 8751 026436 8752 026442 8753 026446	012737	000041	002274	MOV #41.REALPAT ;SETUP PATTERN AND NUMBER FOR TYPEOUT AND DISPLAY CALL GETCSR ;GET CSR NUMBER AND ADDRESS FROM CONFIGURATION TABLE	-
8749 026424	004737	041324		CALL GETCSR ;GET CSR NUMBER AND ADDRESS FROM CONFIGURATION TABLE LET SUPDOADD := #MTP041 ;SET UP TEST ADDRESS LET R1 := #FIRST ;SET UP FIRST ADDRESS	
8751 026436				LET R1 := #FIRST ;SET UP FIRST ADDRESS	
8752 026442	004737	027206		CALL SUPDO3 ; EXECUTE ADDDRESS TO CSR TEST IN SUPVISIOR MODE	
8753 026446	000207			RETURN ;	

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 250 MT0041 SETUP ADDRESS TO CSR ON DOUBLE BIT ERROR TEST

8755 026450				MT0042: SUBTST < <mt0042 address="" csr="" extended="" setup="" test="" to="" unibus="">> :***********************************</mt0042>
8756 026450 8757 026460 8758 026466 8759 026472 8760 026476 8761 026502 8762 026506 8763 026512	012737 012701 004737 004737 004737 004737	000042 100000 046706 041324 037176 046774	002274	IF PMEMFLG IS FALSE THEN \$RETURN ; EXIT IF NOT MS11-P MOV #42, REALPAT ; SETUP PATTERN AND NUMBER FOR TYPEOUT AND DISPLAY MOV #100000, R1 ; SET UP TEST ADDRESS CALL MAPKERNAL ; MAP TO KERNEL SPACE CALL GETCSR ; SET UP CSRINFO FROM CONFIGURATION TABLE CALL MTP042 ; CHECK EXTENDED UNIBUS ADDRESS TO CSR CALL UNMAP ; REMAP KERNEL SPACE

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 251 MT0042 SETUP EXTENDED UNIBUS ADDRESS TO CSR TEST

M10042	SE LOD I	EXTENDED	ONTRO? Y	יישניים אעניי	O CSK 1E31
8765	026514				MT0043: SUBTST < <mt0043 byte="" clears="" sbe="" setup="" test="" write="">> :**SUBTEST MT0043 SETUP WRITE BYTE CLEARS SBE TEST :**SUBTEST MT0043 SETUP WRITE BYTE CLEARS SBE TEST</mt0043>
8766 8767 8768 8769 8770 8771	026536 026542 026546	004737 004737 004737		002274	IF PMEMFLG IS FALSE THEN \$RETURN ; EXIT IF NOT MS11-P MOV #43, REALPAT ; SET UP TEST NUMBER FOR TYPEOUT AND DISPLAY CALL MAPKERNAL ; MAP TO KERNEL SPACE LET R1 := #100000 ; SET UP TEST ADDRESS CALL MTP043 ; PERFORM WRITE BYTE TEST CALL UNMAP ; REMAP KERNEL SPACE RETURN
8773	026552	000207			MT0044: SUBTST < <mt0044 0's="" 1="" bits="" check="" setup="" shifting="" test="" the="" through="">> **SUBTEST MT0044 SETUP SHIFTING 1/0'S THROUGH THE CHECK BITS TEST **********************************</mt0044>
8774 8775 8776 8777 8778	026572 026576	012737 004737	046706	002274	IF PMEMFLG IS FALSE THEN SRETURN : EXIT IF NOT MS11-P MOV #44.REALPAT :SET UP TEST NUMBER FOR TYPEOUT AND DISPLAY CALL MAPKERNAL :MAP TO KERNEL SPACE LET R1 := #100000 :SET UP TEST ADDRESS CALL MTP044 :PERFORM SHIFTING 1/0'S THROUGH THE CHECK BITS
8779 8780 8781	026606 026612	004737 000207	046774		CALL MTP044 ;PERFORM SHIFTING 1/0'S THROUGH THE CHECK BITS CALL UNMAP ;REMAP KERNEL SPACE RETURN MT0045: SUBTST < <mt0045 bit="" csr="" double="" errur="" on="" setup="" syndromes="" to="">> **SUBTEST MT0045 SETUP SYNDROMES TO CSR ON DOUBLE BIT ERROR</mt0045>
8783 8784 8785	026636	012737 004737	046706	002274	IF PMEMFLG IS FALSE THEN SRETURN : EXIT IF NOT MS11-P MOV #45, REALPAT ;SET UP TEST NUMBER FOR TYPEOUT AND DISPLAY CALL MAPKERNAL ;MAP TO KERNEL SPACE LET R1 := #100000 ;SET UP TEST ADDRESS CALL MTP045 ;PERFORM SYNDROMES TO CSR ON DOUBLE BIT ERROR
8786 8787 8788 8789	026646	004737 000207	040142 046774		CALL UNMAP ; REMAP KERNEL SPACE RETURN MT0046: SUBTST < <mt0046 bit="" check="" disabled="" ecc="" errors="" setup="" single="" tet="" with="">> **SUBTEST MT0046 SETUP CHECK SINGLE BIT ERRORS WITH ECC DISABLED TET</mt0046>
8790 8791 8792	026654 026664 026672	012737 004737	000046	002274	IF PMEMFLG IS FALSE THEN \$RETURN : EXIT IF NOT MS11-P MOV
8793 8794 8795 8796 8797	026664 026672 026676 026702 026706 026712	004737 004737 000207	040330 046774		CALL MTP046 CALL UNMAP CALL UNMAP RETURN MT0047: SUBTST < <mt0047 csr="" dbe="" exsisting="" no="" on="" sbe="" setup="" test="" update="" with="">></mt0047>
8798 8799	026714	012737	000047	002274	**SUBTEST MT0047 SETUP NO CSR UPDATE ON SBE WITH EXSISTING DBE TEST IF PMEMFLG IS FALSE THEN SRETURN :EXIT IF NOT MS11-P MOV #47, REALPAT ;SET UP TEST NUMBER FOR TYPEOUT AND DISPLAY
8800 8801 8802 8803	026732 026736 026742 026746 026752	004737	046706		CALL MAPKERNAL :MAP TO KERNEL SPACE LET R1 := #100000 :SET UP TEST ADDRESS LET R2 := #120000 : SECOND TEST ADDRESS CALL MTP047 :PERFORM NO UPDATE TO CSR ON SBE WITH DBE CALL UNMAP :REMAP KERNEL SPACE
880	026756	000207	, 040114		RETURN

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 253 MT0047 SETUP NO CSR UPDATE ON SBE WITH EXSISTING DBE TEST

8808 026760	MT0999: SUBTST < <mt0999 null="" setup="" test="">> :***********************************</mt0999>
8809 026760 005037 002274 8810 026764 8811 026772 000207 8812 8813 026774	CLR REALPAT SET NULLFLAG RETURN
8812 8813 026774	KAMITEST:SUBTST < <check for="" kamikaze="" mode="">> :***********************************</check>
8814 026774 8815 027016 8816 027022 8817 027024 8818 027030	IF KAMIKAZE IS TRUE OR ACTFLAG IS TRUE OR APTFLAG IS TRUE \$RETURN NOERROR ;RUN THE TEST ELSE \$RETURN ERROR ;DON'T RUN THE TEST END ;OF IF KAMIKAZE

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 255 CHECK FOR KAMIKAZE MODE

8821	027030				SUPDO1:	****	*****	******	PATTERN PATTERN	*****	PERVISOR>	> ******	*****	******	****
					*SUBTE	*****	SUBR	*****	*****	****	****	*****	******	******	****
8822 8823 8824	027030 027044 027050 027060	004737	060450		SUPDO2:	MAP CALL PUSH	BANK GETDIS SLPERR	LPADR		;MAP	SUPERVISO	R SPACE	(TEST	AREA) TO	BANK
8822 8823 8824 8825 8826 8827 8828 8829 8831 8832 8833 8834 8835 8837 8838 8838	027070 027072 027072 027074	010037 012700 010120 010220 010320 010420 010520	002156 002160			PUSH MOV MOV MOV MOV MOV MOV MOV MOV	BANK GETDIS \$LPERR, RO, SUPDR #SUPDR1, R1, (R0) R2, (R0) R3, (R0) R4, (R0) R5, (R0) SP, (R0) SP, (R0) SUPDR6, -(R0), R1 -(R0), R1 -(R0), R2 -(R0), R3 -(R0), R3 -(R0), R4 -(R0), R4 -(R0), R5 -(R0), R6 -(R0), R6	RO							
8832 8833 8834 8835 8836 8837 8838	027100 027102 027104 027110 027116 027124 027130 027132	010620 013700 012737 013737 012700 014006 014005 014004 014003 014002 014001 014000	002156 027124 002606 002174	002606 002610	TAG4\$:	MOV MOV MOV	SUPURO, F #TAG4\$, \$LPADR, #SUPDR6; -(RO), SI -(RO), RI	LPADR LPERR 2,RO							
8840 8841 8842 8843	027136 027140 027142 027144 027146					MOV MOV MOV MOV MOV MOV SUPERVI	-(R0),R -(R0),R -(R0),R -(R0),R		:ENTER	SUPERV	ISOR MODE				
8840 8841 8842 8843 8844 8845 8846 8847 8848 8849 8850 8851 8852	027146 027154 027160 027162 027166 027170 027172 027174	012706 104424 004737 104423 104417 000004	000740 177640			MOV CACHOFF CALL CACHON KERNEL SCOPE	#SUPSTK FASTCIT	,SSP			FF TO THE U N MODE		TRUCTIO	ON PAR'S	
8850 8851 8852	027172 027174 027204	000004				POP RETURN	SLPADR,	SLPERR							

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 257 SUBR EXECUTE PATTERN IN SUPERVISOR

8855 027206 8856 027222 8857 027226 8858 027236 8859 027242 8860 027246 8861 027250 8862 027252 8863 027254 8864 027256 8865 027260 8866 027262 8867 027266 8868 027274 8869 027302 8870 027306 8871 027310 8872 027312 8873 027314 8874 027316 8875 027320 8876 027322 8877 027324 8878 027332 8879 027336 8880 027344 8881 027344 8882 027346 8883 027352 8887 027366 8886 027366 8888 027366	004737 010037 012700 010120 010220 010320 010420 010520 010620 013700 012737 012737 012700 014006 014005 014005 014001 014001 014000	060450 002156 002160 002156 027302 002606 002174	002606 002610	SUPDO4:	MOV MOV MOV MOV MOV MOV MOV MOV	BANK GETDIS \$LPERR,\$LPADR RO,SUPDRO #SUPDR1,RO R1,(RO)+ R2,(RO)+ R3,(RO)+ R4,(RO)+ R5,(RO)+ SUPDRO,RO #TBG4\$,\$LPADR \$LPADR,\$LPERR #SUPDR6+2,RO -(RO),R5 -(RO),R5 -(RO),R3 -(RO),R1 -(RO),R0	;MAP	SUPERVISOR	SPACE	(TEST	AREA)	TO 8	BANK
8876 027322 8877 027324 8878 027332 8879 027336 8880 027340 8881 027344 8882 027346 8883 027352 8884 027354 8885 027360 8886 027362 8887 027364 8888 027366 8889 027376	005737 001403 012706 000402 012706 104424 004777 104423 104417 000004	002452 000700 000740 152700		1\$: 2\$:	TESTARE/ TST BEQ MOV BR MOV CACHOFF CALL CACHON KERNEL SCOPE POP RETURN	NOSUPER 1\$ WUSESTK,USP 2\$ WSUPSTK,SSP aSUPDOADD \$LPADR,\$LPERR	;TUR	ER SUPERVISO					

MEMUKT	IESI PAI	IERN KUU	I THE 2											
8892 8893 8894 8895 8896 8897 8898 8900 8901				PATTER	N REGIST RO R1 R2	LUPY UF	NTIONS DRESS OF ADDRE	F PAT SSES IN (ON NECES NECES	****	(FIRST, LA TTERN (SI 525, ETC)	******* ST+2,ETC ZE)	******	******	**
8902	027400			: ******	******	< <mtp000< td=""><td>****</td><td>****</td><td>*****</td><td>A TEST>></td><td>*****</td><td>******</td><td>*******</td><td>**</td></mtp000<>	****	****	*****	A TEST>>	*****	******	*******	**
8903	027400	010220 077102		SUBTES	******	R2,(R0) R1,MTP00	*****			******	*****	******	*******	**
8909	027414	000240 012401 020102 001402 104430		2\$:	NOP MOV CMP BEQ PERRO2	(R4)+,R1 R1,R2 3\$		V1 V1 V1 V1	7644 7646 7650 7652 7654	EST>>				
8910 8911 8912	027420	000240 077306 000207		3\$:	NOP SOB RETURN	R3,2\$		V V1	77660 77662					
8913	027424	000201		MTP001:	SUBTST	< <mtp001< td=""><td>******</td><td></td><td></td><td>EST>></td><td>*****</td><td>******</td><td>*********</td><td>**</td></mtp001<>	******			EST>>	*****	******	*********	**
				:*SUBTES	ST	MTP001	ADDRESS	****	*****		******	*******	*********	**
8914 8915 8916 8917		010220 062702 077104 000240	000002	3\$:	MOV ADD SOB NOP	R2,(R0) #2,R2 R1,3\$; V1 ; V1 ; V1 ; V1	77640 77642 77646 77650 77652 77654 77656					
8918 8919 8920	027436	000240 012400 020005 001401		1\$:	MOV CMP BEQ	(R4)+,R(R0,R5 2\$)	V1 V1	77652 77654 77656					
8921 8921 8921	027446	104427 062705 077307 000207	000002	2\$:	PERRO1 ADD SOB	#2.R5 R3,1\$		V1	77662 77666 77672				N, READ UP)	
892 892	027456	000207		MTP002:	RETURN	< <mtp002< td=""><td>2</td><td>COM</td><td>PLEMEN</td><td>T ADDRESS</td><td>TEST (W</td><td>RITE DOWN</td><td>N. READ UP)</td><td>></td></mtp002<>	2	COM	PLEMEN	T ADDRESS	TEST (W	RITE DOWN	N. READ UP)	>
				*SUBTE	ST	MTP002	COMPLE	****	****		LLLLLLLL	IN, READ L		
892 892 892	6 027456 7 027460 8 027464	010540 062705 077104	000002	38:	MOV ADD SOB	R5,-(R0) #2,R5 R1,3\$: V1	77640 77642 77646					
8921 8929 893 893 893	027474 2 027476 3 027500	010540 062705 077104 000240 162702 012401 020102 001401 104430 077307 000207	000002	15:	NOP SUB MOV CMP BEQ	#2,R2 (R4)+,R' R1,R2 2\$	1	V1 V1 V1	77652 77656 77660 77662					
893 893 893	4 027502 5 027504 6 027506	104430 077307 000207		2\$:	PERRO2 SOB RETURN	R3,1\$		V1 V1 V1	77664 77666 77670					

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 261 MTP002 COMPLEMENT ADDRESS TEST (WRITE DOWN, READ UP)

			**								
8939	027510			******	*****	*****	*******	*****	*****	EST (WRITE):	*****
				:*SUBTES	ST	MTPA03 3 XOR 9	WORST CAS	E NOISE T	EST (WRI	TE)	
00/0				******		DRESS	*********	*****		*******	
8940					:R1 = A1	MALL LOOP CONSTAN	TV				
8942					:R3 = N	MALL LOOP CONSTANT	(LARGE LI	00P)			
8941 8942 8943					:R4 = G	DOD DATA					
8944					; R5 = M	DIUM LOOP CONSTA	ANT				
8945	027510	010421		15:	.ENABL MOV	LSB R4,(R1)+	:V177640				
8946 8947	027510 027512	010421		10.	MOV	R4,(R1)+	: V177642				
8948	027514	077203			MOV SOB COM	R2,1\$: V177644				
8949	027516	005104			COM	R4	:V177646				
8950	027520	000401		WARN2:	BIS 401	(PC)+,R4	:V177650	WARNING L	OCATION	IS MODIFIED	BEFORE LOADING
8952	027524	012702	000004	WALLIAG.	MOV	#4.R2	:V177654				
8953	027530	077511			MOV SOB COM	#4.R2 R5.1\$:V177660				
8954	027532	005104			COM	R4	:V177664				
8956	027536	000401		WARN3:	BIS 401	(PC)+,R4	:V177666	WARNING L	OCATION	IS MODIFIED	BEFORE LOADING
8957	027540	012705	000100	W/1/1/10	MOV	#64. R5 R3,1\$:V177670				
8958	027544	077317			SOB	R3,1\$; V177670 ; V177674 ; V177676				
8959	027546	000207			RETURN .DSABL	LSB	:4177070				
8961					. DONOL	L30					
8962	027550			MTPB03:	SUBTST	< <mtpb03< td=""><td>3 XOR 9 W</td><td>ORST CASE</td><td>NOISE 1</td><td>EST (READ)></td><td>></td></mtpb03<>	3 XOR 9 W	ORST CASE	NOISE 1	EST (READ)>	>
				******	*******	MTDDAZ Z VAD O	WORST CAS	E NOISE 1	ECT (DE	(0)	*******
				SUBTE	31 ****	MTPB03 3 XOR 9	*****	****	*****	******	********
8963				•	.ENABL	LSB					
8964		000137 077203	027610	1\$:	JMP	a#MTPC03	:V177640 :V177644 :V177646 :V177650 :V177652	GC	TO V172	2360	
8965	027554	077203			SOB	R2,1\$	· V177646				
8967	027556 027560	005104 052704			BIS	(PC)+,R4	¥177650				
8968	027562	000401		WARN4:	BIS 401		: V177652	WARNING L	OCATION	IS MODIFIED	BEFORE LOADING
8969	027564	012702	000004		MOV	#4,R2 R5,1\$:V177654				
8970 8971	02/5/0	005104			SOB	R5, 13	·V177662				
8972	027574	052704			BIS	R4 (PC)+,R4	:V177664				
8973	027576	077511 005104 052704 000401 012705 077317 000207	*****	WARN5:	401		v177660 v177662 v177664 v177666 v177670 v177674	WARNING L	OCATION	IS MODIFIED	BEFORE LOADING
8974	027600	072705	000100		VOM	#64. R5 R3,1\$: V177676				
8975 8976	027606	000207			SOB RETURN .DSABL	NJ, IV	¥177676				
8976 8977					.DSABL	LSB					

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 263 MTPB03 3 XOR 9 WORST CASE NOISE TEST (READ)

8980 027	610		MTPC03:	******	< <mtpc03 mtpc03="" test<="" th=""><th>TEST DATA SUBP</th><th>PROGRAM>></th><th>******</th></mtpc03>	TEST DATA SUBP	PROGRAM>>	******
8981 0276 8982 0276 8983 0276 8984 0276 8985 0276	612 001401 614 104431 616 005141		15:	CMP BEQ PERRO3 COM COM	R4,(R1)+ 1\$ -(R1) (R1)	; V172360 ; V172362 ; V172364 ; V172366 ; V172370		
8985 0276 8986 0276 8987 8988 0276		027626	MTPD03:	JMP SUBTST	awmTPD03	; V172372 TEST DATA SUBS	GO TO V172260 SUBPROGRAM>>	*****
			*SUBTE	ST	MTPD03 TEST	DATA SUBSUBPROGRA	M *******	*******
8989 027 8990 027 8991 027	626 020421 630 001401 632 104431			CMP BEQ PERRO3	R4,(R1)+	:V172260 :V172262 :V172264		
8992 027	634 005127 636 000000		1\$:	COM	(PC)+	V172266		
8994 027 8995 027		027554		BNE	MTPC03 a#MTPB03+4	:V172272 :V172274	GO TO V172360 GO TO V177644	

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 265

MIPUUS	IESI DA	IA 30030	רעאטטאאיז					
8998	027646			MTPA04:	****	< <mtpa04 mtpa04="" rotati<="" td=""><td>NG ZEROS TE</td><td>********</td></mtpa04>	NG ZEROS TE	********
8999 9000 9001	027646 027652 027654	012705 010504 000241 000137 016004	000010	1\$:	MOV MOV CLC	#8R5 R5,R4	:V177640 :V177644 :V177646	
9002 9003 9004	027646 027652 027654 027656 027662 027666 027670 027672 027674 027676	000137 016004 103402 020204 001401 104432 077115	027702 177776		JMP MOV BCS CMP REO	a/MTPB04 -2(R0),R4 2\$ R2,R4 3\$; V177640 : V177644 : V177646 : V177654 : V177660 : V177662 : V177664 : V177666 : V177670 : V177672	
9005 9006 9007 9008 9009 9010 9011	027674 027676 027700	104432 077115 000207		2\$: 3\$:	BEQ PERRO4 SOB RETURN	R1,1\$	v177666 v177670 v177672	
9011	027702			MTPB04:	SUBTST	< <mtpb04< td=""><td>SUBR F</td><td>ROTATING BIT>></td></mtpb04<>	SUBR F	ROTATING BIT>>
				*SUBTE	ST	MTPB04 SUBR	ROTATING	BIT
9012 9013 9014	027702 027704 027706 027710	106110 077502 106120		15:	ROLB SOB ROLB	(R0) R5.1\$ (R0)+	: V172360 : V172362 : V172364	
9015 9016 9017 9018	027712	106120 106110 077402 106120 000137	027662	2\$:	ROLB SOB ROLB JMP	R5.1\$ (R0)+ (R0) R4.2\$ (R0)+ a/MTPA04+14	; V172360 ; V172362 ; V172364 ; V172366 ; V172370 ; V172372 ; V172374	
9019	027722			MTP005:	SUBTST	< <mtp005 ***********************************<="" td=""><td>ROTATION</td><td>ONES TEST>></td></mtp005>	ROTATION	ONES TEST>>
9021	027722	012705	000010	15:	MOV	*****	;V177640	********
9022	027726	012705 010504 000261			MOV	#8.,R5 R5,R4	:V177644 :V177646	
9024 9025 9026 9027	027732 027736 027742 027744	000201 000137 016004 103002 020204 001401 104432 077115 000207	027702 177776		JMP MOV BCC CMP	a/MTPB04 -2(R0),R4 2\$ R2,R4 3\$:V177650 :V177654 :V177660 :V177662	IF THIS HAPPENS THE GOOD & BAD MATCH
9028 9029 9030 9031	027750 027750 027752 027754	104432 077115 000207		2\$: 3\$:	BEQ PERRO4 SOB RETURN	R1,1\$	V177666 V177670 V177672	

9034	027756				MTP006:	****	*****	INITIAL DATA TEST>> DATA TEST
9035 9036 9037 9038 9039 9040 9041 9042 9043 9046 9046 9046 9047 9048 9049 9050 9051 9052 9053 9054	027756 027764 027770 027776 030004 030010 030016	012737 005037 013771 013771 017102 023702 001401 104433	000001 002242 002240 002242 000000 002240	002240 000000 000002	18:	MOV	**********	T CHECKS THE DI/DO LINES BY A 1 THROUGH THE WORD.
9045 9046 9047 9048 9049	030020 030024 030030 030032	017102 023702 001401 104434	000002 002242		2\$:	MOV CMP BEQ PERR10	a2(R1),R2 DATBUF+2,R2 3\$:NOW READ SECOND WORD :BR IF OK :ERROR TRAP
9050 9051 9052 9053 9054	030034 030040 030042 030052	005737 100405 000746	002242		3\$:	TST BMI DLEFT BR	DATBUF+2 4\$ DATBUF 1\$;HAS LAST BIT BEEN TESTED ? ;MINUS MEANS BIT 31 ;NO, SHIFT TEST BIT LEFT ;GO WRITE NEW TEST DATA
9055 9056 9057 9058 9059 9060 9061 9062 9063	030054 030062 030070 030076	012737 012737 013771 013771 017102 023702 001401 104433	177776 177777 002240 002242 000000 002240	002240 002242 000000 000002	4\$: 5\$:	MOV MOV MOV MOV CMP BEQ PERRO7	#177776,DATBUF #-1, DATBUF+2 DATBUF,a(R1) DATBUF+2,a2(R1) a(R1),R2 DATBUF,R2 6\$; HAS LAST BIT BEEN TESTED ? ; MINUS MEANS BIT 31 ; NO, SHIFT TEST BIT LEFT ; GO WRITE NEW TEST DATA ; NOW GOING TO SHIFT A O IN DATA DIRECTION ; PUT A O IN BIT O ; AND 1'S IN ALL OTHERS ; WRITE THE DATA ; 2 WORDS WORTH ; NOW READ FIRST WORD ; BR IF OK
9066	030120 030124 030130 030132	017102 023702 001401 104434	000002 002242		6\$:	MOV CMP BEQ PERR10	a2(R1),R2 DATBUF+2,R2 7\$; NOW, READ SECOND WORD ; BR IF OK
9069 9070 9071 9072 9073 9074	030134 030140 030142	005737 100005 000746 000207	002242		7\$: 8\$:	TST BPL DLEFT BR RETURN	DATBUF+2 8\$ DATBUF 5\$;TESTED BIT 31 YET? ;BR IF YES, WE'RE DONE ;KEEP GOING

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 269 MTP006 INITIAL DATA TEST

MILLOOD	THAT I THE	שחות וני					. 18. 18. 18. 18. 18. 18. 18. 18. 18. 18	
9077	030156			MTP007:	*****	MTP007	ADDRESS BIT TEST>> ADDRESS BIT TEST	
9078 9079 9080 9081 9082		111100		,	MOVB	(KI), KU	THIS TEST CHECKS TO SEE THAT EACH ADDRESS BIT IN EACH 16K BANK CAN BE ASSERTED UNIQUELY. IT CHECKS FOR ADDRESS BITS THAT MAY BE STUCK HIGH, STUCK LOW OR STUCK TOGETHER.	
9083	030160	105700 001401 104435			TSTB BEQ PERR11	R0 1\$	READ AND COMPARE FOR ZEROS	
9087	030166	105111		1\$:	COMB	(R1) (R1),R0	COMPLEMENT THE BYTE	
9084 9085 9086 9087 9088 9089 9090 9091 9092 9093	030166 030170 030172 030174 030176	105700 001001 104436			TSTB BNE PERR12	R0 2\$	READ FOR NON ZEROS	
9093 9094 9095	030200 030202 030204 030206 030210	040201 006302 050201 011100 005700		2\$:	BIC ASL BIS MOV	R2,R1 R2 R2,R1 (R1),R0	:MASK OFF THE ASSERTED BIT :SHIFT R2 FOR NEXT BIT :SET THE NEW BIT INTO R1	
9094 9095 9096 9096 9098 9098	030212	005700 001401 104437			TST BEQ PERR13	R0 3\$	READ THE NEW ADDRESS READ FOR ZEROS	
9101	030216	005111 011100		3\$:	COM	(R1) (R1),R0	COMPL THE WORD	
9103 9103 9104 9105 9106	030224	005700 001031 104440			TST BNE PERR14	R0 4\$;READ IT AGAIN	
9107 9108 9109 9110 9111	030230 030234 0 030236 0 030242 1 030244 2 030246	022702 001407 022702 001356 006302 012701 000752 000207	100000 010000 160000	4\$:	CMP BEQ CMP BNE ASL MOV	#100000 5\$ #10000, 2\$ R2 #160000	CHECK FOR MSB IN 4K BANK :NOT LAST BIT, BRANCH	
911	030252	000752		5\$:	BR RETURN	2\$		

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 271 MTP007 ADDRESS BIT TEST

9117	030256				MTP010:	SUBTST	< <mtp010 MTP010 BYTE</mtp010 	BYTE ADD	RESSING TEST>>	******	******
9127 9128 9129 9130 9131 9132	030256 030260 030262 030266 030272 030300 030302 030304 030310 030312 030314	010402 010403 062702 012713 012763 105013 010401 020201 001420 020301 001007 111100	000004 177777 177777	000002	1\$: 2\$:	CMP	; TEST 3 THIS R4,R2 R4,R3 W4,R2 W-1,(R3) W-1,2(R3) (R3) R4,R1 R2,R1 6\$ R3,R1 4\$ (R1),R0 G IF YOU OPTO W0,R0 3\$	TEST CHECKS ADDRESSING ;R4 HAS ;PUT IT ;POINT R ;WRITE A ;THE 4 T ;CLEAR A ;INITIAL ;IF EQUA ;BR IF E ;IS THIS ;BR IF N MIZE CHANGE ;IT IS,	FOR PROPER WITH ECC DISABI LOWEST ADDRESS IN R3 ALSO 2 TO LAST BYTE LL ONES IN EST BYTES BYTE IZE R1 FOR EACH L, JUST READ LA QUAL THE BYTE OF ZI OT THE PCBUMP FOR COMPARE FOR ZEI	+1 +1 +1 A PASS AST BYTE EROS THIS ERROR INC	ASE OF TRAPS
9134 9135	030316 030322 030324	001401 104435				PERR11	3\$				
9136 9137 9138 9139	030326 030330	005201 000765 111100 122700			3\$:	INC BR	R1 2\$; NEXT BY ; RETURN	TE		
9139 9140 9141 9142	030326 030330 030332 030334 030340 030342	111100 122700 001401 104436	177777		45:	MOVB CMPB BEQ PERR12	(R1),R0 #-1,R0 5\$:ITS NOT	THE BYTE OF 0	'S, READ 1'S	
9143 9144 9145	030344	005201			5\$:	INC BR	R1 2\$		NEXT BYTE		
9146 9147 9148 9149 9150	030344 030346 030350 030354 030356 030360 030362	005201 000756 112713 005203 020302 001347 000207	177777		6\$:	MOVB INC CMP BNE RETURN	#-1,(R3) R3 R3,R2 1\$:RESTORE :INC TO :WAS THA :BR IF N	1'S TO BYTE JO NEXT BYTE IT JUST THE LAS	UST TESTED T ONE?	

MIPOIO	BALE WAL	WE221MG	1521							
9154	030364				MTP011:	T	*****	SINGLE BIT ERROR TEST	[>> +***********************************	
9155 9156 9157						(2)	READ BACK SBE UM	CORRECTED (WITH ECC I)ISABLE)	
9157 9158 9159						(3)	ENABLE ECC & REA	AD CORRECTED DATA		
9159 9160 9161						(4)	CHECK THAT THE	SBE FLAG WAS SET FROM	THE LAST READ	
9161 9162 9163 9164 9165 9166 9167						(5)	POSITIONS OF A I	TA CONSISTING OF 1 BI DOUBLE WORD N FOR 1 BIT CLEARED IN	N EACH OF 32 POSITIONS OF	
9168	070741	10/507				(6) CLR1CSR	TE (DIIN TEST AL	SBE IN EACH OF 32 BIT * 32 = 2048 TIMES)		
9171	030366	005737	014004			TST BEQ	PHEBE MTLA11	CLEAR 1 SELECTED CS	NDICATOR	
9173 9174 9175	030364 030366 030372 030374 030400 030406	104503 005737 001407 013702 013737 010237	172246 172252 172252	172246		MOV MOV MOV	SIPARS, AUSIPARS R2, AUSIPARS	SAME COMITMIS OF ST		
9176 9177 9178 9179	030412 030420	012737 005037	000001 002242	002240	MTLA11:	;BIG LO MOV CLR ;MEDIUM	#1,DATBUF DATBUF+2	; INITIAL DATA ;32 BITS WORTH		
9180 9181	030424 030432	012737 005037	000001 002252	002250	MTLB11:	MOV	#1,SBEMSK SBEMSK+2	:INITIAL ERROR MASK		
0403			002240 002242 002262	002244 002246	MTLC11:	MOV TSTB	DATBUF+2,TSTDAT PASFLG ; COMP D	+2: TO SAVE ORIG DATA ATA ON SECOND PASSONL	,	
9186 9187 9188 9189	030436 030444 030452 030456 030460 030464 030470 030474 030500 030506	013737 013737 105737 001404 005137 005137 013702 013703 012737 004737	002244 002246 002244 002246 002244 044366		45:	BEQ COM COM MOV MOV	TSTDAT TSTDAT+2 TSTDAT,R2 TSTDAT+2,R3 #TSTDAT,SOURCE	BR IF FIRST PASS SECOND PASS, COMP B	OTH WORDS	
9191 9192	030500 030506	012737	002244	002306		MOV	#TSTDAT, SOURCE CHKGEN	SET UP ADDRESS FOR GEN CHECKBITS ON TS	CHKGEN	
9192 9193 9194 9195						** CRE	ATE A SINGLE BIT	ERROR **		
9196 9197 9198 9198	030512 030516 030522	013701 074137 013701 074137 013701 013705	002250 002244 002252 002246 002406 002410		MP1 044	MOV XOR MOV XOR	SBEMSK,R1 R1,TSTDAT SBEMSK+2,R1 R1,TSTDAT+2 TESTADD,R1	. FIRST TEST ADDRESS		
9200 9201	030536	013701	002410		MTLD11:	MOV	IESTADD+2,KD	FIRST TEST ADDRESS	ECTEN CCD	
9202	030544	013711	002244			MOV COLCED	TSTDAT, (R1)	:DISABLE ECC ON 1 SE :WRITE FIRST 16 BITS :WRITE GENERATED CHE	LECTED CSR	
9200 9201 9202 9203 9204 9205 9206	030552	104475	002246			CB1CSR MOV	TSTDAT+2,(R5)	WRITE SECOND 16 BIT : CHECK BITS. WE NOW	CKBITS IN 1 SELECTED CSR S AND HAVE CHECKBITS	

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 20	6-APR-82 09:41	PAGE 273-1
--	----------------	------------

MIPUII	21MOLE	STI EKKO	K IE21				
9207 9208 9209 9210 9211 9212 9213 9214	030556 030560 030562 030566 030570 030574	104471 011100 020037 001403 010137 104455	002244 002032		ECCIDIS MOV CMP BEQ MOV PERR31	(R1),R0 R0,TSTDAT 63 R1,ADDRESS	GENERATED ON DATBUF AND DATA WITH ONE BIT IN ERROR (AS PER SBEMSK). DISABLE ECC ON 1 SELECTED CSR READ THE LOW WORD (UNCORRECTED) BR IF OK
9219 9220	030576 030600 030606 030606 030612	011500 020037 001403 010537 104455	002246 002032	6\$:	MOV CMP BEQ MOV PERR31	(R5) RO RO.TSTDAT+2 7\$ R5.ADDRESS	READ THE HIGH WORD (UNCORRECTED)
9221 9222 9223 9224 9225 9226 9227	030614 030622 030624 030644 030646 030646 030652	104426		7\$:	READC: IF #B ERR END; END; OF	IT4 OFF.IN CSR OR +45 OF IF #BIT4 IF KFLAG	OR #BIT15 OFF.IN CSR
9226 9227 9228 9229 9230 9231 9232 9234 9235 9236 9237	030646 030652 030654 030656 030660 030662 030664	005737 001001 104512 104503 011100 020002 001401 104456	014004	17\$:	TST BNE ERRGEN CLR1CSR MOV CMP BEQ PERR32	PHEBE 17\$ (R1),R0 R0,R2 8\$	CLEAR 1 SELECTED CSR SEE IF ITS BEEN CORRECTED IT SHOULD HAVE BEEN
9236 9237 9238 9239 9240 9241 9242 9243	030672 030674 030702 030706	104510 103411 010137 104460	002032	8\$:	TSTREAD BCS SET MOV PERR34 SET	9\$ HEADER R1,ADDRESS HEADER	:TEST LOC (R1) & TST FOR SBE (WITHOUT FETCHES) :BR IF IT IS SET :ENABLE PRINTING OF ERROR HEADER INFO :ENABLE PRINTING OF ERROR HEADER INFO
9244 9245 9246 9247	030716 030720 030722 030724	104503 011500 020003 001401 104456		9\$:	CLR1CSR MOV CMP BEQ PERR32	(R5),R0 R0,R3 10\$	CLEAR 1 SELECTED CSR SEE IF ITS BEEN CORRECTED BR IF OK
9259 9251 9252 9253 9253	030730 030732 030734 030742 030746	104510 103411 010137 104460	002032	10\$:	TSTREAD BCS SET MOV PERR34	11\$ HEADER R1,ADDRESS	:TEST LOC (R1) & TST FOR SBE (WITHOUT FETCHES) :BR IF YES :ENABLE PRINTING OF ERROR HEADER INFO
9248 9249 9250 9251 9253 9254 9255 9256 9257 9258 9260 9261	030730 030732 030734 030746 030750 030756 030760 030764 030766 030772 030774	104512 105737 100452 005737 100405	002262 002252	11\$:	SET ERRGEN TSTB BMI TST BMI	PASFLG 15\$ SBEMSK+2 12\$: TEST FOR LAST MASK BIT : MINUS MEANS BIT 31
9261 9262 9263	030774 031004 031006	000614		128:	DLEFT	SBEMSK MTLC11 1 SET.IN aSWR 1	

ı	CZMCDAO	mc11_1 /m/	D MEMOI	DV DIAG	MACRO M	1113	26-APR-82 09:41 PAGE 273-2	
ı	MTP011	SINGLE BI	T ERROR	R TEST	HACKO H		26-APR-82 09:41 PAGE 273-2	
	9264 9265 9266	031016 031024 031030 1	05737 00406	002242			IF QVFLAG IS TRUE THEN GOTO 13\$ TST DATBUF+2 ;LAST DATA BIT ? BMI 13\$;WHICH IS BIT 31 DLEFT DATBUF	
	9268 9269 9270	031042 0 031046 1 031052 0	00137 05737 01004	030424 002262		13\$:	JMP MTLB11 TSTB PASFLG ;FIRST OR SECOND PASS ? BNE 14\$:NON ZERO MEANS WE'RE DONE	
	9271 9272 9273	031064 1 031060 0 031064 0	00137 05737 01004 05237 00137 52737 05002 05003 05037 05037 12704 12737	002262 030412 000200	002262	14\$:	JMP MTLATT BIS #BIT7,PASFLG	
	9275 9276 9277	031074 0 031076 0 031102 0	05003 05037 05037	002244 002246			CLR R5 CLR TSTDAT CLR TSTDAT+2	
	9278 9279 9280 9281	031106 0 031112 0 031120 0	12704 12737 174437	002244 002246 000040 003740 002310	002310	15\$:	MOV #40,R4 MOV #3740,CHECK XOR R4,CHECK ASL R4	
	9282 9283 9284	031126 031132 031134 0	06304 32704 01002 00137	020000			BIT WBIT13,R4 BNE 16\$ JMP MTLD11	
	9285 9286 9287 9288	031016 031030 031030 031032 031042 031046 031052 031054 031064 031064 031072 031074 031076 031174 031126 031126 031126 031132 031134 031140 031140 031140 031152 031156 031156	04471 013701 013705	002406 002410		16\$:	MOV TESTADD,R1 MOV TESTADD+2,R5	
	9289 9290 9291	031152 031156 031160	04503				CLEAR (R1),(R5) CLR1CSR ;CLEAR 1 SELECTED CSR RETURN	

1111	,,,	21MOCE	DI I ENNO						
•	9294	031162				MTP012:	**************************************	MTP012 WRITE B	WRITE BYTE CLEARS SBE TEST>> ***********************************
	9295 9296					*****	******	SINGLE BIT ERR	OR TEST TO INSURE THAT A WRITE NGLE BIT ERRORS. CLEAR 1 SELECTED CSR INITIAL DATA 32 BITS WORTH INITIAL ERROR MASK 32 BITS WORTH SAVE ORIGINAL DATA +2:BOTH WORDS NEED ADDRESS FOR CHKGEN GENERATE CHECK BITS FIRST TEST ADDRESS PUT IT IN R1 ALSO DISABLE ECC ON 1 SELECTED CSR
	9297 9298	031162 031164 031172 031176 031204	104503 012737	000001	002240		CLR1CSR MOV	#1.DATBUF	CLEAR 1 SELECTED CSR INITIAL DATA
	9299 9300	031172	005037	000001 002242 000001	002250	1\$:	MOV	#1,SBEMSK	INITIAL ERROR MASK
	9302 9302	031204	005057	002252 002240 002242 002244	002244 002246 002306	2\$:	CLR MOV MOV	DATBUF, TSTDAT	SAVE ORIGINAL DATA
	9303	031224	012737	002244	002306		MOV	WISTDAT, SOURCE	:NEED ADDRESS FOR CHKGEN :GENERATE CHECK BITS
	9306 9307	031236	013701	044366 002250 002244 002252 002246 002406			MOV	SBEMSK,R1 R1,TSTDAT	
	9308 9309	031246 031252	013701 074137	002252			MOV	SBEMSX+2,R1 R1,TSTDAT+2	
	9310 9311 9312 9313	031256 031262	013704	002406			MOV MOV	TESTADD,R4 R4,R1	PUT IT IN R1 ALSO
	9312 9313	031224 031232 031236 031242 031246 031252 031256 031264 031264 031272 031274 031276	104503 012737 005037 012737 005037 013737 013737 013701 074137 013701 074137 013701 014471 013711 104475	002244			MOV	TSTDAT, (R1)	WRITE 16 BITS
	9314 9315	031274	060501 013711 104503	002246			CB1CSR ADD MOV	R5,R1 TSTDAT+2,(R1)	INDEX UP TO SECOND WORD
	9317 9318 9318			002240			CLR1CSR	10104112,4417	CLEAR 1 SELECTED CSR
	9319 9320	031304 031310	012702	002250			MOV SUB	#SBEMSK,R2 R5,R1 #-1,(R1)	ADDRESS OF ERROR MASK RETURN TO FIRST WORD
	9321 9322	031312 031316	112711	177777 002524		3\$:	MOVB	#-1 (R1) KFLAG	; WRITE A BYTE OF 1'S ; IS THIS MF11S-K
	6423	031304 031310 031312 031316 031322 031330 031332 031334	012702 160501 112711 005737 001403 132712	177777			BEQ BITB BEQ	4\$ #-1,(R2) 6\$	FIRST TEST ADDRESS PUT IT IN R1 ALSO DISABLE ECC ON 1 SELECTED CSR WRITE 16 BITS WRITE GENERATED CHECKBITS IN 1 SELECTED CSR INDEX UP TO SECOND WORD WRITE HIGH WORD+CHECKBITS CLEAR 1 SELECTED CSR IT'S DANGEROUS IF WE DON'T ADDRESS OF ERROR MASK RETURN TO FIRST WORD WRITE A BYTE OF 1'S IS THIS MF11S-K BRANCH IF NOT - IT'S MS11-M DID THIS BYTE HAVE THE BAD BIT IN IT? NO - BRANCH TEST LOC (R1) & TST FOR SBE (WITHOUT FETCHES) NO - SKIP
	9326	031332	001420 104510 103011			4\$:	TSTREAD	5\$	TEST LOC (R1) & TST FOR SBE (WITHOUT FETCHES)
	9328	031336		002032			SET	HEADEK	ENABLE PRINTING OF ERROR HEADER INFO
	9330 9331	031334 031336 031344 031350 031352	0101 <i>3</i> 7 104017	*******			ERROR SET	R1 ADDRESS +17 HEADER	:ENABLE PRINTING OF ERROR HEADER INFO
	9332 9333		111100 122700	42222		5\$:	MOVB	(R1) R0	-CHECK DATA
	9334	031360 031362 031365 031370	001414 104457	177777			CMPB BEQ PERR33	#-1,R0 7\$:CHECK DATA :BR IF OK
	9336 9337 9338 9339 9340 9341	031370				6\$:	TSTREAD		:TEST LOC (R1) & TST FOR SBE (WITHOUT FETCHES)
	9339	0313.2	104310						READ THE BYTE SBE ERROR BIT ONLY SET ? SHOULD BE SET, BR IF OK ENABLE PRINTING OF ERROR HEADER INFO
	9341	031374 031376 031404	103771				BCS	5\$ HEADER	SHOULD BE SET, BR IF OK SENABLE PRINTING OF ERROR HEADER INFO
	9342 9343 9344	031404 031410 031412	010137 104460	002032			MOV PERR34	R1,ADDRESS	
	9344 9345 9346 9347	051412	172713	177777		76.	SET	HEADER	:ENABLE PRINTING OF ERROR HEADER INFO :CHECK FOR LAST BYTE
	734/	031420	132712	177777		7\$:	BITB	#-1,(R2)	, CHECK TON EAST OTTE

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 275-1 MTP012 WRITE BYTE CLEARS SBE TEST

9348 031424 9349 031426	001012 005202 005201			BNE	8\$ R2 R1	:
9350 031430	005201	002/04		INC	R1 TESTADD,R4	:MOVE TO NEXT BYTE :FIRST TEST ADDRESS
9350 031430 9351 031432 9352 031436	032701	002406 000002		BIT	#2,R1 3\$	TEST FOR LOWER WORD
9353 031442 9354 031444	001723 062704 000720 005737	000002		BNE INC INC MOV BIT BEQ ADD BR	#2.R4	BR IF IT'S LOW 16 BITS :ADJUST POINTER FOR ERROR REPT.
9355 031450 9356 031452	000720	002252	8\$:	BR	3\$ SBEMSK+2	:LAST ERROR BIT ?
9357 031456	100405	002272		TST BMI DLEFT	9\$:LAST ERROR BIT ? :MINUS MEANS BIT 31
9358 031460 9359 031470 9360 031472	000647			BR	SBEMSK 2\$	EN COTO 100
9360 031472 9361 031502			9\$:	IF QVFL	1 SET. IN ASWR TH AG IS TRUE THEN	GOTO 10 \$
9361 031502 9362 031510 9363 031514	005737 100405	002242		TST BMI	DATBUF+2 10\$;LAST DATA BIT? ;MINUS = BIT 31
9363 031514 9364 031516 9365 031526	000623			DLEFT	DATBUF	
9366			100.	CLEAR	OUT ANY DBE'S OR	SBE'S :DISABLE ECC ON 1 SELECTED CSR
9367 031530 9368 031532 9369 031536	104471 013701	002406	10\$:	MOV	TESTADD,R1	, DISABLE ELL ON I SELECTED CON
9370 031540	005011 060501			CLR	(R1) R5,R1 (R1)	
9371 031542 9372 031544	005011 104503			CLR1CSR	(R1)	CLEAR 1 SELECTED CSR
9373 031546	000207			RETURN		

0774	071550				MTD013.	TOTALIO	CCMTP013	CREATE DOUBLE BIT ERROR TEST>>
4310	031550				*****	****	*****	· · · · · · · · · · · · · · · · · · ·
					:*SUBTE	ST	MTP013 CREATE D	OUBLE BIT ERROR TEST
9377					; *****	******	:DOUBLE BIT ERRO	R FORCE TO CHECK DOUBLE ERROR LOGIC ;CLEAR 1 SELECTED CSR
	031550	104503				CLR1CSR	MTFCTARR RT	
9379	031552	012701	002406		15:	MOV	#TESTADD,R1	:MAKE INITIAL DATA
9381	031562	005037	002242			CLR	DATBUF+2	;ALL ZEROS
9380 9381 9382 9383	031566	012737	000001	002250	2\$:	MOV	#1,SBEMSK	SECOND HOPD
9384	031600	012737	000001	002254	3\$:	CLR	#1.DBEMSK	:INITIAL DOUBLE ERROR MASK
9384 9385	031550 031556 031556 031562 031566 031574 031600 031612 031620 031626 031632 031634 031644 031644 031654	104503 012701 005037 005037 012737 005037 012737 005037 013737 013737	002406 002240 002242 000001 002252 000001 002256 002240 002242			CLR	DBEMSK+2	:MAKE INITIAL DATA :ALL ZEROS :INITIAL SINGLE ERROR MASK :SECOND WORD :INITIAL DOUBLE ERROR MASK :32 BITS HERE ALSO
9386	031612	013737	002240	002244	45:	MOV	DATBUF+2.TSTDAT	ž
9388 9389	031626	105737	002262	002240		1310	LYSLEG "40 COUR	PLEMENTING FIRST PASS
9389	031632	001404 005137 005137 104503 023737				COM	5S TSTDAT	; COMP FIRST WORD
9391	031640	005137	002244 002246			600	TCTDATAT	CECOND HODD
9392	031644	104503		002257	5\$:	CLR1CSR	CDEMCK VDEMCK	CLEAR 1 SELECTED CSR
9393	031654	001004	002250	002254		CMP BNE	6\$	IN BOTH MASKS
9395	031656	001004 023737	002252	002256		CMP	SBEMSK+2, DBEMSK	2: COULD BE EQUAL IN SECOND WORD
9396	031664 031666 031674 031700 031704 031710	001460 012737 004737 013702	002244	002306	6\$:	BEQ MOV	#TSTDAT SOURCE	CLEAR 1 SELECTED CSR CAN'T HAVE THE SAME ERROR BIT SET IN BOTH MASKS C: COULD BE EQUAL IN SECOND WORD GO MAKE THEM NOT EQUAL SOURCE ADDRESS FOR CHKGEN GO GENERATE CHECK BITS
9398	031674	004737	044366	002300		CALL	#TSTDAT, SOURCE CHKGEN SBEMSK, R2 R2, TSTDAT	GO GENERATE CHECK BITS
9399	031700	013702	002250 002244			MOV XOR	SBEMSK, RZ	
9401	031710	074237 013702	002252			MOV	SBÉMSK+2,R2 R2,TSTDAT+2	
9402	1151/14	074237	002246 002254			XOR MOV	RZ,TSTDAT+Z	
9404	031724	013702 074237	002244			XOR	DBEMSK, R2 R2, TSTDAT	
9405	031730	013702	002256 002246			MOV	DBÉMSK+2,R2 R2,TSTDAT+2	
9400	031720 031724 031730 031734 031740	074237 104471	002240		16\$:	ECC1DIS		:DISABLE ECC ON 1 SELECTED CSR :WRITE 16 BITS
9408	031742 031746	013731	002244			MOV	TSTDAT, a(R1)+	WRITE 16 BITS : WRITE GENERATED CHECKBITS IN 1 SELECTED CSR
9409	031746	013771	002246	000000		CB1CSR MOV	TSTDAT+2,a(R1)	WRITE HIGH WORD
9411	031756	104503				CLR1CSR		WRITE HIGH WORD CLEAR 1 SELECTED CSR ADJUST TEST ADDRESS
9412	031760	162701	000002			SUB	#2,R1 a(R1)	READ THE LOCATION
9414	031770	013771 104503 162701 005771 104501 103411	000000			WAS1DBE		READ THE LOCATION WAS THERE ANY DOUBLE BIT ERRORS ON 1 SELECTED CSR
9415	031772	103411				BCS	9\$ HEADER	:11 2MOULD BE 2E:
9417	032002	011137	002032			MOV	(R1),ADDRESS	
9418	031750 031756 031760 031764 031770 031772 031774 032002 032006 032010	104030				ERROR SET	(R1), ADDRESS +30 HEADER	
7417	032010					361	HEADER	

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 279 MTPO13 CREATE DOUBLE BIT ERROR TEST

9422 032016 9423 032020	104512 105737 100452 005737	002262		9\$:	ERRGEN TSTB PASFLG BMI 14\$
9422 032016 9423 032020 9424 032024 9425 032026 9426 032032 9427 032034 9428 032044 9429 032046 9430 032056 9431 032064 9432 032070	100452 005737 100405	002256		138:	TST DBEMSK+2 ; CHECK MASK FOR LAST BIT BMI 10\$; MINUS = BIT31 DLEFT DBEMSK
9428 032044 9429 032046 9430 032056	000662			10\$:	BR 4\$ IF #SW11 SET.IN ASWR THEN GOTO 11\$ IF QVFLAG IS TRUE THEN GOTO 11\$
9431 032064 9432 032070 9433 032072	005737 100405	002252			DLEFT SBEMSK
9433 032072 9434 032102 9435 032104 9436 032110 9437 032112	000636 105737 001003 105237	002262		11\$:	BR 3\$ TSTB PASFLG ;FIRST PASS BNE 12\$;NON ZERO MEANS WE'RE DONE INCB PASFLG ;FIRST PASS, NOT DONE ;CLEAR OUT ANY DBE'S OR SBE'S
9438		000200	002262	12\$:	CLEAR OUT ANY DBE'S OR SBE'S BR 18 :KFEP GOING BIS #BIT7, PASFLG :SET UP FOR CHECK BIT TEST
9440 032120 9441 032126 9442 032132 9443 032136	005037 005037 012737	002244 002246 000040 000100 003740	002250		CLR TSTDAT CLR TSTDAT+2 MOV #40.SBEMSK MOV #100.DBEMSK
9443 032136 9444 032144 9445 032152 9446 032160 9447 032164	000617 052737 005037 005037 012737 012737 012737 013702 074237 004237 006337 032737 001652 006337 032737 001006 013737 006337	003740 002250 002310	002254	14\$:	MOV #3740, CHECK MOV SBEMSK, R2 XOR R2, CHECK
9448 032170 9449 032174 9450 032200 9451 032204 9452 032212 9453 032214	074237 006337 032737	002250 002310 002254 002310 002254 020000	002254		XOR R2,CHECK ASL DBEMSK BIT #BIT13,DBEMSK
W434 U3///U	0016327 006337 032737	002250 004000	002250		BEQ 16\$ ASL SBEMSK BIT #BIT11,SBEMSK BNE 15\$
9455 032226 9456 032230 9457 032236 9458 032242	013737 006337 000743	002250 002254	002254		MOV SBEMSK, DBEMSK ASL DBEMSK BR 14\$
9459 032246 9460 032246 9461 032252	104471	002406		15\$:	#TESTADD,R1 CLEAR a(R1)+,a(R1) CLR1CSR ;DISABLE ECC ON 1 SELECTED CSR ;CLEAR 1 SELECTED CSR
9462 032260 9463 032262	104503 000207				CLR1CSR ;CLEAR 1 SELECTED CSR RETURN

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 282 MTP013 CREATE DOUBLE BIT ERROR TEST

*SUBTEST MTP014 BASIC DOUBLE BIT ERROR TEST	***
9468 9469 9470 9471 9472 9472 THIS TEST CHECKS THAT A DOUBLE ERROR WILL BE DETECTED A BYTE WRITE WITH A DOUBLE ERROR ON A MS11-P WILL BE ABOTRED.	
9473 032264 104424	
9476 032300 LET ADDRESS := #FIRST ; SET ADDRESS FOR ERROR REPORT ; SET ADDRESS FOR ERROR REPORT ; ENABLE CHECK/SYNDROME BIT REGISTER	
9478 032310 9479 032316 104425 LOADCSR :WRITE DBE CHECK BITS TO CSR	
9481 032326 9482 032330 005711 9483 032332 USPBL ASSERTED????	
9484 032342 104055 ERROR +55 ; ERROR CALL ;; MISSED EXPECTED TRAP	
9485 032344 104426 9487 032346 042737 020000 002146 READCSR 9488 032354 104426 9488 032354 SET	CATOR XSISTS1
9490 032372 9491 032400 104065	
9492 032402 9493 032402 104473 9494 032404 005037 002264 END 9495 032410 ECC1INIT : ENABLE BUSPBL CLR PASSNO : CLEAR LOOP COUNTER	
9496 032410 104473	
9499 032422 LET (RT) :8= #3/7 ; WRITE BYTE SHOULD BE ABORTED	
9501 032430 9502 032440 9503 032446 9504 032452 9505 032460 9506 032462 9507 032462 9508 032464 9509 032474 005041 IF PARCNT NE #1 SET HEADER LET GOOD := #0 LET BAD := #377 BAD DATA ERROR +56 END INC R1 UNTIL PASSNO EQ #2 CLR -(R1) CLEAR LUT	
9506 032462 9507 032462 005201 END AND REPEAT ON HIGH BYTE	
9507 032462 005201 INC R1 ;AND REPEAT ON HIGH BYTE 9508 032464 UNTIL PASSNO EQ #2 :CLEAR LUT	
9500 032426	

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 283 MTP014 BASIC DOUBLE BIT ERROR TEST

MIFUIA	PY21C D	DOPLE DI	ENNON	1231				
9515	032510				******	******* ST	*****	WRITE INHIBIT OF BYTE WITH DBE>> ***********************************
					*****	*****	****	***********
9516 9517							CHECKS FOR WRIT	E INHIBIT DURING A WRITE BYTE. ORRECTED DATA. :INITIAL DATA :32 BITS WORTH :SINGLE ERROR MASK
9518	032510	005037	002240		1\$:	CLR	DATBUF	INITIAL DATA
9519	032514	005037	002242	002250	20.	CLR	MAIBUTTE	CINCLE EDDUD WACK
9520	032520	012/3/	000001	002250	2\$:	MOV	WI'SDELISK	STUDE EURON LINON
9521	032320	003037	000001	002254	3\$:	CLR MOV	#1 DREMSK	DOUBLE ERROR MASK
9525	032332	005037	002256			CLR	DREMSK+2	:
9524	032544	013737	002240	002244	45:	MOV	DATBUF, TSTDAT	PRESERVE ORIG DATA
9525	032510 032514 032526 032526 032532 032540 032544 032560 032560 032560 032616 032610 032616 032610 032616 032620 032632 032632 032632 032632 032636 032632 032636 032632 032636 032632 032636 032632 032636 032632 032636 032636 032636 032636 032636 032636 032636 032636 032636 032636 032636	005037 005037 005037 005037 012737 005037 013737 013737 105737 001404 005137 104503 023737 001004 023737 001474 012737 013701 074137 013701 074137 013701 074137 013701 074137 013701 074137 013701 074137 012701 104471	002240 002242 000001 002252 000001 002256 002240 002242	002246		MOV	DATBUF+2, TSTDAT	INITIAL DATA 32 BITS WORTH SINGLE ERROR MASK DOUBLE ERROR MASK PRESERVE ORIG DATA WHICH PASS ? FIRST PASS NO COMPLEMENTING
9526	032560	105737	002262			TSTB	PASFLG	WHICH PASS ?
9527	032564	001404				DEW	1-0	ALTIOI LUDO'S HO COLLICELLIANS
9528	032566	005137	002244 002246			COM	TSTDAT	SECOND DASS COMPLEMENT TSTDAT
9529	032372	10/503	002246		5\$:	CLR1CSR	ISIDAITE	CLEAR 1 SELECTED CSR
9530	032376	023737	002250	002254	,	CMP	SREMSK DREMSK	CHECK FOR SAME MASKS
9532	032606	001004	OULESO	002234		BNE	6\$	BR IF NOT EQUAL
9533	032610	023737	002252	002256		CMP	SBEMSK+2, DBEMSK	+2 ; SECOND WORD ALSO
9534	032616	001474				BEQ	115	;BR TO MAKE THEM NOT EQUAL
9535	032620	012737	002244 044366 002250 002244 002252 002246	002306	6\$:	MOV	#TSTDAT, SOURCE	ADDRESS FOR CHKGEN
9536	032626	004737	044366			CALL	CHKGEN D1	GU GENERATE CHECK BITS
9537	032632	013/01	002230			MOV	D1 TCTDAT	
9530	032636	013701	002253			MOV	SREMSK+2.R1	
9540	032646	074137	002246			XOR	R1.TSTDAT+2	
9541	032652	013701	002254			MOV	DBEMSK,R1	
9542	032656	074137	002244			XOR	R1,TSTDAT	
9543	032662	013701	002256			MOV	DBEMSK+2,R1	
9544	032666	074137	002244 002256 002246 002406		70.	XOR	RI, ISIDAI+2	TEST LOCATION
9545	032672	10/471	002406		7\$:	ECCIDIS	WIESIADO,KI	SECOND PASS, COMPLEMENT TSTDAT CLEAR 1 SELECTED CSR CHECK FOR SAME MASKS BR IF NOT EQUAL SECOND WORD ALSO BR TO MAKE THEM NOT EQUAL ADDRESS FOR CHKGEN GO GENERATE CHECK BITS TEST LOCATION DISABLE ECC ON 1 SELECTED CSR WRITE FIRST 16 BITS OM R2 WRITE GENERATED CHECKBITS IN 1 SELECTED CSR WRITE SECOND 16 BITS + CHECKBITS
9540	032070	013731	002244			MOV	TSTDAT a(R1)+	WRITE FIRST 16 BITS
9548	032100	013/31	002244			:LOAD C	SR WITH IMAGE FR	OM R2
9549	032704	104475				CB1CSR		; WRITE GENERATED CHECKBITS IN 1 SELECTED CSR ; WRITE SECOND 16 BITS + CHECKBITS ; CLEAR 1 SELECTED CSR
9550	032706	104475 013771	002246	000000		1101	Interest . Plantit.	WRITE SECOND 16 BITS + CHECKBITS
9551	032714	104503	******			CLR1CSR		CLEAR I SELECTED CSR
9552	032716	013702	002406			MOV	TESTADD,R2	DO DESIGNATES FIRST BY:
9223	032722	104503 013702 010203 062703 112722 012701 017100	200000			ADD	R2,R3 #3,R3	GET ADDRESS OF TEST LOC RE DESIGNATES FIRST BYTE RESIGNATES LAST BYTE
9555	032730	112722	000003 000360 002406 000000 002244		8\$:	MOVB	#360, (R2)+	TRY WRITING A BYTE
9556	032734	012701	002406			MOV	#TESTADD_R1	
9557	032740	017100	000000			MOV	a(R1),R0	
9558	032744	023700	002244			CMP	TSTDAT,RO	CHECK FOR UNCHANGED DATA
9559	032750	023700 001404 017137 104455		002072		BEQ	9\$;BR IF OK
9560	032752	01/15/	000000	002032		MOV DEDD 31	a(R1),ADDRESS	
9201	032700	104423				PERR31		
9502	032762	017100	000002		9\$:	MOV	a2(R1),R0	
9564	032766	023700	002246			CMP	TSTDAT+2.RO	READ SECOND WORD
9565	032772	017100 023700 001404 017137 104455				BEQ	10\$	BR IF UNCHANGED
9566	032774	017137	000002	002032		MOV	a2(R1), ADDRESS	
9564 9565 9566 9567 9568	032704 032706 032714 032716 032722 032724 032730 032734 032740 032740 032750 032750 032760 032760	104455				PERR31		
9568								

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 283-1 MTPO15 WRITE INHIBIT OF BYTE WITH DBE

111 013	music .	1011011 0						
9569	033004 033006 033010 033014 033016 033022 033024 033034	020203			10\$:	CMP	R2,R3	:TESTED LAST BYTE ?
9570	033006	020203 001350 105737 100452 005737				BNE	8\$;BR IF NO
9571	033010	105737	002262		115:	TSTB	PASFLG	
9572	033014	100452				BMI	15\$:BRANCH IF TESTING CHECK BITS :CHECKING FOR LAST ERROR BIT :BR IF DONE HERE
9573	033016	005737	002256			TST	DBEMSK+2	CHECKING FOR LAST ERROR BIT
9574	033022	100405				BMI	12\$	BR IF DONE HERE
9575	033024					DLEFT	DBEMSK	
9576	033034	000643				99	45	
9576 9577	033036	0000.5			12\$:	IF #SW11	SET.IN aSWR T	THEN GOTO 13\$ I GOTO 13\$:LAST SBE MASK :BR IF DONE WITH THIS PASS
9578	033046					IF QVFLA	AG IS TRUE THEN	GOTO 13\$
9579	033054	005737	002252			TST	SBEMSK+2	:LAST SBE MASK
9580	033060	005737 100405				BMI	13\$	BR IF DONE WITH THIS PASS
9581	033062					DLEFT	SBEMSK	
9582	033072	000617				BR	3\$	
9580 9581 9582 9583	033074	105737	002262		13\$:	TSTB	PASFLG ; TEST	PASS FLAG
9584	033100	001003				BNE	145	:NON ZERO MEANS WE'RE DONE
9585	033102	105237	002262			BNE	PASFLG ; NOT D	OONR
9586	033106	000600				BR	1\$	
9584 9585 9586 9587 9588 9589 9591 9592 9593 9594 9595	033062 033072 033074 033100 033106 033110 033116 033122 033126 033154 033154 033160 033164 033174 033174 033202 033204 033216 033220	000617 105737 001003 105237 000600 052737 005037 005037 012737 012737 012737 012737 012737 013702 074237 006337 032737 001633 006337 0013737 0013737	000200 002244 002246 000040 000100 003740 002250 002310 002254 002310 002254	002262	145:	BIS	#BIT7, PASFLG	
9588	033116	005037	002244			CLR	TSTDAT	
9589	033122	005037	002246			CLR		
9590	033126	012737	000040	002250 002254 002310		CLR	#40.SBEMSK #100.DBEMSK #3740.CHECK SBEMSK.R2 R2.CHECK DBEMSK.R2 R2.CHECK DBEMSK.R2	
9591	033134	012737	000100	002254		MOV	#100_DBEMSK	
9592	033142	012737	003740	002310	15\$:	MOV	#3740 CHECK	
9593	033150	013702	002250			MOV	SBEMSK R2	
9594	033154	074237	002310			XOR MOV XOR ASL BIT BEQ	R2.CHECK	
9595	033160	013702	002254			MOV	DBEMSK,R2	
9596	033164	074237	002310			XOR	R2, CHECK	
9596 9597	033170	006337	002254			ASL	DBEMSK	
9598	033174	032737	020000	002254		BIT	MBIT13, DBEMSK	
9599	033202	001633				BEQ	7\$	
9600	033204	006337	002250 004000			ASL	SBEMSK	
9601	033210	032737	004000	002250		BIT	#BIT11,SBEMSK	
9602	033216	001006				BNE	165	
9600 9601 9602 9603	033220	013737	002250 002254	002254		MOV	SBEMSK, DBEMSK	
9604	033226	006337	002254			ASL	DBEMSK	
9605	033232	000743				BR	15\$	
9606	033234	104471			16\$:	ECC1DIS		;DISABLE ECC ON 1 SELECTED CSR
9607	033236	012701	002406			MOV	#TESTADD,R1	; TEST LOCATION
9608	033242					CLEAR	#TESTADD,R1 a(R1)+,a(R1)	DISABLE ECC ON 1 SELECTED CSR TEST LOCATION TO ERASE ANY DBE'S FROM TESTING
9609	033226 033232 033234 033236 033242					RESTOR	E CSR	
9610	033250	104503				CLR1CSR		CLEAR 1 SELECTED CSR
9611	033250 033252	104503				RETURN		

MIP	013	MKTIE TI	MUTOT! O	DITE W	III DOL				
•	9614	033254				MTP016:	SUBTST	< <mtp016 1<="" mtp016="" th="" write=""><th>WRITE INVIBIT OF WORD WITH DBE>> ***********************************</th></mtp016>	WRITE INVIBIT OF WORD WITH DBE>> ***********************************
	9615					•	:DOUBLE	BIT ERROR WRITE	CANCEL WITH
	9615 9616						:WORD WE	RITE.	
	9617						: CHECKS	WRITE INHIBIT WI	ITH WORD WRITES TO
	9617 9618						: WORD WI	TH DOUBLE ERROR.	
	9619	033254	005037	002240		T12A:	CLR	DATBUF	;BACKGROUND FOR DOUBLE ERRORS
	9620	033260	005037	002242			CLR	DATBUF+2	; 2 WORDS WORTH
•	9621	033264	012737	000001	002250		MOV	#1,SBEMSK	SINGLE ERROR MASK
	9622	033272	005037	002252	000051	*430.	CLR	SBEMSK+2	DOUBLE EDDOD MACK
	9623	033276	012/3/	000001	002254	1128:	MUV	WI'NDEWOK	DOUBLE ERROR PMSK
	9024	022204	013737	002230	002244	18.	MOV	DATRUF TSTDAT	DATA FOR TEST
	9023 0434	022210	013737	002240	002244	10.	MOV	DATBUF+2 TSTDAT	+2:BOTH WORDS
	9627	033324	105737	002240 002242 000001 002252 000001 002256 002240 002242 002262	002240		TSTB	PASFLG : COMP DA	ATA ON SECOND PASS ONLY
	9628	033330	001404				BEQ	2\$	BR IF FIRST PASS
	9629	033332	005137	002244 002246 002250			COM	TSTDAT	COMP FIRST WORD
1	9630	033336	005137	002246			COM	TSTDAT+2	NOW SECOND WORD
	9631	033342	023737	002250	002254	52:	CMP	SREMOK , DREMOK	OD IE DIEEEDENT
	9032	022220	001004	002252	002254		CWD	CDEMCK+5 UBEWCK	42-LIPPER WORD TOO
	2022	033332	001502		002230		REO	8\$	BR TO MAKE THEM NOT EQUAL
	0633	033362	012737	002244	002306	35:	MOV	#TSTDAT SOURCE	NEED ADDR OF DATA FOR CHKGEN
	9636	033254 033264 033264 033276 0333276 0333310 0333310 03333310 03333310 03333310 03333310 03333310 03333310 03333310 0333410 0333410 0333410 0333410 0333410 0333410 0333410 0333410 0333410 0333410 0333410 0333410 0333410 0333410 0333410 0333410 0333410 0333410 0333410 0333410	005037 005037 012737 005037 012737 005037 013737 013737 001404 005137 005137 005137 001502 012737 001502 012737 013701 074137 013701 074137 013701 074137 013701 074137 012701	002244 044366 002250 002244 002252 002246 002254 002246 002266 002246	002500		CALL	CHKGEN	GO GENERATE CHECK BITS
	9637	033374	013701	002250			MOV	SBEMSK,R1	
	9638	033400	074137	002244			XOR	R1,TSTDAT	
	9639	033404	013701	002252			MOV	SBEMSK+2,RI	
	9640	033410	0/413/	002240			MOV	NDEMSK DI	
	9641	033414	074137	002234			XOB	R1 TSTDAT	
	2275	033750	013701	002256			MOV	DBEMSK+2.R1	
	9644	033430	074137	002246			XOR	R1.TSTDAT+2	
	9645	033434	012701	002406		45:	MOV	#TESTADD,R1	FIRST TEST ADRRESS
	9646	033440	104471				ECC1DIS		DISABLE ECC ON 1 SELECTED CSR
	9647 9648	033442	013731	002244			MOV	TSTDAT, a(R1)+	WALLE LINGS TO BITS
	9648	033446	104475	0022/4	000000		CB1CSR	TSTDAT+2,a(R1)	- UPITE SECOND 14 RITS + CHECKRITS
	9650	033430	105037	002246 002263 000002	000000		MOV	UPPFLG	SET FOR 2 LOOPS
	9651 9651	033770	162701	000002			SUB	#2.R1	POINT TO LOW WORD
	9652	033466	013731 104475 013771 105037 162701 104503 012771 012701 017100 023700 001404 017137 104455			5\$:	SUB CLR1CSR		:FIRST TEST ADRRESS :DISABLE ECC ON 1 SELECTED CSR :WRITE FIRST 16 BITS :WRITE GENERATED CHECKBITS IN 1 SELECTED CSR :WRITE SECOND 16 BITS + CHECKBITS :SET FOR 2 LOOPS :POINT TO LOW WORD :CLEAR 1 SELECTED CSR :TRY WRITING LOCATION
	9653	033470	012771	177400 002406 000000 002244	000000		MOV	#177400,a(R1) #TESTADD,R1	TRY WRITING LOCATION
	9654	033476	012701	002406			MOV	WIESINDO'NI	
	9655	033502	017100	000000			MOV	a(R1),R0	CHECK FOR ODICINAL DATA
	9656	033506	023700	002244			CMP	TSTDAT, RO	CHECK FOR ORIGINAL DATA SHOULD BE UNCHANGED
	2051	022215	001404	000000	002032		BEQ MOV	6\$ a(R1),ADDRESS	SHOOLD BE CHCHANGED
	2020	033314	104455	000000	002032		PERR31	OLK 17 , ADDRESS	
	9660						· Linia		
	9661	033524	062701	000002		6\$:	ADD	#2,R1	
	9662	033530	017100	000002 000000 002246			MOV	#2,R1 a(R1),R0 ISTDAT+2,R0	
	9663	033534	023700	002246			CMP	ISTDAT+2,RO	; THIS SHOULD BE UNCHANGED ALSO
	9658 9659 9660 9661 9662 9663 9664 9665	033524 033530 033534 033540 033542 033550	062701 017100 023700 001404 017137 104455	000000	002032		MOV _	75 a(R1),ADDRESS	
	7003	033342	104455	000000	002032		PERR31	OLK 17, ADDRESS	
	7000	033330	104477				. 5,11131		

CZMSPAO MS11-L/M/P MEMO MTP016 WRITE INHIBIT O	RY DIAG. MACRO M1113	26-APR-82 09:41 PAGE 287	
9669 033552 105737 9670 033556 001003 9671 033560 105237	002263 7\$:	TSTB UPPFLG BNE 8\$ INCB UPPFLG	WHICH LOOP ? SECOND, BR OUT FIRST, KEEP GOING
9669 033552 105737 9670 033556 001003 9671 033560 105237 9672 033564 000740 9673 033566 105737 9674 033572 100454 9675 033574 005737 9676 033600 100405	002262 8\$: 002256	BR 5\$ TSTB PASFLG BMI 12\$ TST DBEMSK+2	LAST BIT ? MINUS = BIT 31
9669 033552 105737 9670 033556 001003 9671 033560 105237 9672 033564 000740 9673 033566 105737 9674 033572 100454 9675 033574 005737 9676 033600 100405 9677 033602 9678 033612 000636 9679 033614 9680 033624 9681 033632 005737 9682 033636 100406 9683 033640 9684 033650 000137 9685 033654 105737 9686 033660 001004	9\$:	DLEFT DBEMSK	
9680 033624 9681 033632 005737 9682 033636 100406 9683 033640	002252	IF #SW11 SET. IN DSWR THEF IF QVFLAG IS TRUE THEN GO TST SBEMSK+2 BMI 10\$ DLEFT SBEMSK	LAST BIT IN THIS MASK ? BR IF LAST BIT
9684 033650 000137 9685 033654 105737 9686 033660 001004 9687 033662 105237	033276 002262 10\$:	BNE 11\$ INCB PASFLG : INDICAT	ASS ? BR IF SECOND E SECOND PASS COMING
9688 033666 000137 9689 033672 052737 9690 033700 005037 9691 033704 005037	033254 000200 002262 11\$: 002244 002246	JMP T12A BIS #BIT7,PASFLG CLR TSTDAT	
9684 033650 000137 9685 033654 105737 9686 033660 001004 9687 033662 105237 9688 033666 000137 9689 033672 052737 9690 033700 005037 9691 033704 005037 9692 033710 012737 9693 033716 012737 9694 033724 012737 9695 033732 013702 9696 033736 074237 9697 033742 013702 9698 033756 074237 9699 033756 032737 9701 033764 001623 9702 033766 006337 9703 033772 032737	002262 033254 000200 002262 11\$: 002244 002246 000040 002250 000100 002254 003740 002310 12\$: 002250 002310 002254 002310 002254 002310	MOV #100, DBEMSK	
9697 033742 013702 9698 033746 074237 9699 033752 006337 9700 033756 032737	002254 002310 002254 020000 002254	MOV DBÉMSK,R2 XOR R2,CHECK ASL DBÉMSK BIT #BIT13,DBEMSK	
9701 033764 001623 9702 033766 006337 9703 033772 032737 9704 034000 001006	002250 004000 002250	BEQ 4\$ ASL SBEMSK BIT WBIT11,SBEMSK BNE 13\$	
9703 033772 032737 9704 034000 001006 9705 034002 013737 9706 034010 006337 9707 034014 000743 9708 034016 104471 9709 034020 012701	002250 002254 002254 13\$:	MOV SBEMSK, DBEMSK ASL DBEMSK BR 12\$	DISABLE ECC ON 1 SELECTED CSR
1 9/10 034024 003031	002406 000000	MOV #TESTADD,R1 CLR a(R1)+ CLR a(R1)	:RESTORE TEST ADDRESS :CLEAR ANY DBE'S FROM TEST
9711 034026 005071 9712 034032 104503 9713 034034 000207		CLR1CSR RETURN	CLEAR 1 SELECTED MK11 CSR

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 289 MTP016 WRITE INHIBIT OF WORD WITH DBE

9716	034036			*****	SUBTST	< <mtp017< th=""><th>HOLDING 1'S & O'S TEST>></th></mtp017<>	HOLDING 1'S & O'S TEST>>
9725 9726 9727 9728 9729 9730 9731 9732	034036 034042 034044 034050 034054 034060 034062 034064 034066	012701 010104 012705 012700 010003 000303 110021 110321 020105 103774	060000 160000 000377	*SUBTE	*****	*****	1'S & O'S TEST THE MEMORY FOR THE CAPABILITY AND O'S BY WRITING A BACKGROUND EADING IT EN USING A BYTE AT A TIME REPEATED WITH A SWAPPED BACKGROUND PATTERN S BYTES & READS WORDS GET THE PATTERN INTO RO WRITE A BYTE WRITE THE MEMORY WITH THE BYTE STORED IN BAKPAT+1 COMPARE TEST LOC TO TOP + 2 BRANCH IF LOWER
9736	034070 034072	014102 020002		2\$:	MOV CMP	-(R1),R2 R0,R2	:TEST THE MEMORY TO SEE IF IT CONTAINS :THE WORD STORED IN BAKPAT
9739	034074 034076	001401 104446		\	PERR22		
9740 9741 9742	034100 034102 034104	020104 101372 000303 000300 001763		3\$:	CMP BHI SWAB	R1,R4 2\$ R3 R0 1\$:KEEP ON TESTING THE MEMORY UNTIL :R1 EQUALS THE LOWEST ADDRESS :CHANGE THE DATA PATTERN
9743 9744 9745	034106	000300 001763			SWAB BEQ	R0 1\$: IF THE DATA PATTERN DOES NOT HAVE LOW : BYTE =0 THEN FALL THRU
9746	034112	000207			RETURN		

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 290 MTP017 HOLDING 1'S & O'S TEST

9748

MTP017	HOLDING	1'5 & 0	'S TEST			
(Party	034114				MTP020: SUBTST < <mtp020 csf<="" syndromes="" th="" to=""><th>ES TO CSR ON SINGLE BIT ERROR TEST>> R ON SINGLE BIT ERROR TEST</th></mtp020>	ES TO CSR ON SINGLE BIT ERROR TEST>> R ON SINGLE BIT ERROR TEST
9751 9752 9753 9754					THIS TEST CHECKS TO SEE IF THE BIT IN THE CSR TO BE SET AND ALL 16 DATA BITS.	HE SINGLE BIT ERRORS CAUSE THE SBE CORRECT SYNDROME BITS ARE GENERATED FOR
9755 9756 9757 9758 9759	034114 034116 034120 034124	104424 005000 105037 104513	002262		CACHOFF CLR RO CLRB PASFLG CBREG REPEAT LET PASFLG : B= PASFLG + #1	:TURN OFF CACHE :CLEAR DATA :CLEAR PASFLG :ENABLE CHECK/SYNDROME BIT REGISTER
9760 9761 9762 9763 9764 9765	034114 034116 034120 034124 034126 034126 034132 034136 034142				LET R4 := #-1 LET BITNO := #0 IFB PASFLG EQ #1 LET R5 := #1	INCREMENT LOOP COUNTER INDEX TO SINGLE BIT ERROR TABLE CLEAR INNER LOOP COUNTER SELECT DATA TO BE CORRECTED BY PASSNO DATA=0;BIT TO BE CORRECTED IS A ONE
9766 9767 9768	034156 034160 034164				LET R5 := #177776	DATA=177776;BIT TO BE CORRECTED IS A ZERO
9769 9770 9771	034164 034170	005237	002320		INC BITNO LET R4 := R4 + #1 LET R2 :R= PTABLE (R4)	INCREMENT BIT POINTER POINT TO NEXT SET OF CHECK BITS GET NEXT SET OF CHECK BITS
9773 9774 9775	034176 034202 034206	072227 052702	000005 000004		ASH #5.R2 BIS #BIT2,R2 LET CSR := R2	SHIFT TO LINE UP IN CSR SENABLE DIAG MODE GET CHECK BITS TO BE WRITTEN
9776	034212	104425			LOADCSR LET (R1) := R0	LOAD CSR WITH DATA
9778 9779 9780 9781 9782 9783	034136 034142 034152 034156 034164 034164 034164 034164 034170 034170 034272 034214 034214 034216 034220 034224 034224	104503 005711 104426 042737	177757	002146	LET R4 := R4 + #1 LET R2 :B= PTABLE(R4) ASH #5,R2 BIS #BIT2,R2 LET CSR := R2 LOADCSR LET (R1) := R0 CLR1CSR TST (R1) READCSR BIC #^C20,CSR IF CSR NE #20 LET GOOD := #20	INCREMENT BIT POINTER POINT TO NEXT SET OF CHECK BITS GET NEXT SET OF CHECK BITS SHIFT TO LINE UP IN CSR ENABLE DIAG MODE GET CHECK BITS TO BE WRITTEN LOAD CSR WITH DATA WRITE DATA TO TEST ADDRESS CLEAR CSR CORRECT SBE READ CSR FOR CORRECT SGE BIT AND SYNDROMES CLEAR ALL BUT SBE INDICATOR WAS DATA CORRECTED???
9784 9785	034250 034256	104060			LET GOOD := #20 LET BAD := CSR ERROR +60	NO ERROR
9787 9788 9789	034260 034262 034264	104513 104426 042737	174033	002146	END CBREG READCSR BIC #^C3744, CSR	:ENABLE SYNDROME BIT REGISTER :GET SYNDROMES FROM CSR :MASK SYNDROME BITS :GET GOOD SYNDROMES
9786 9787 9788 9789 9790 9791 9792 9793 9794 9795 9796 9796 9797 9798 9798 9801 9802 9803	034242 034250 034256 034260 034262 034264 034272 034276 034306 034314 034326 034336 034336 034336 034350 034350	072327 052703	000005 000004		LET R3 :B= SBESYN(R4) ASH #5 R3 BIS #BIT2 R3 IF R3 NE CSR SET HEADER LET GOOD := R3 LET BAD := CSR	GET SYNDROMES FROM CSR MASK SYNDROME BITS GET GOOD SYNDROMES SHIFT INTO POSITION SET DIAG MODE IN DATA DO SYNDROME BITS AGREE
9797	034334	104042			ERROR +42	
9799	034336	005011			CLR (R1) IFB PASFLG EQ #1	CLEAR LUT SHIFT NEW DATA DEPENDING ON PASFLG SHIFT BITNO TO THE LEFT
9801	034350	006305			ASL R5	SHIFT BITNO TO THE LEFT
9803	034354	000261			ELSE	SET CARRY BIT AND

									G	4	
	CZMSPAO MTPO20	MS11-L/ SYNDROM	M/P MEMORY	V DIAG. MA	ACRO M	1113 26- ERROR TES	-APR-82 ST	09:41	PAGE 29	91-1	
	9804	034356	006105				ENI		R5		ROTATE LEFT
	9805 9806 9807	034360 034370 034370	005100				UNTILB COM F	BITNO	EQ #16.		UNTIL ALL BITS ARE DONE COMPLEMENT DATA AND REPEAT UNTIL 2 PASSES ARE COMPLETE!
	9809 9810 9811	034402 034404 034406	104503 104423 000207				CLR1CS/ CACHON RETURN	3			CLEAR CSR TURN CACHE
Ì	9812 9813						MS11-P	SINGLE	BIT ER	ROR SYNDROME	BIT TABLE
	9811 9812 9813 9814 9815	034410 034413 034416 034421 034424 034427	016 025 032 045 052 064	013 026 034 046 054	023 031 043 051 061	SBESYN:	.BYTE	16,13	,23,25,	26,31,32,34,	43,45,46,51,52,54,61,64
1	9816	034461	301								

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 292 MTPO20 SYNDROMES TO CSR ON SINGLE BIT ERROR TEST

0010	07//70			MTDA21.	CURTET	< <mtpa21< th=""><th></th><th>APCHING</th><th>1'S & O'S PATTERN TEST>></th></mtpa21<>		APCHING	1'S & O'S PATTERN TEST>>
9818	034430			*SUBTES	T	MTPA21	MARCHING	1'S & 0	S PATTERN TEST
9819 9820 9821 9822 9823	034430 034432 034434 034436	014100 020200 001401 104443		15:	;READ,BY MOV CMP BEQ PERR17	TESWAP-N -(R1),R0 R2,R0 2\$	ODIFY.REA 0; v177640 : v177642 : v177644 : v177646	D,DOWN	
9828 9829	034440 034442 034444 034446 034450	000311 011100 020300 001401 104444		2\$:	SWAB MOV CMP BEQ PERR20	(R1) (R1),R0 R3,R0 3\$; V177650 ; V177652 ; V177654 ; V177656 ; V177660		
9830 9831 9832	034452	020401 001365 000207		3\$:	CMP BNE RETURN		:V177662 :V177664 :V177666		DOME? NO - LOOP YES - RETURN
9833 9834 9835 9836 9837 9838 9839	034462	011100 020300 001401 104444		MTPB21: 1\$:	;READ,BY MOV CMP BEQ PERR20	YTESWAP-I (R1),R0 R3,R0 2\$	10DIFY,REA :V177640 :V177642 :V177644 :V177646	D.UP	
9838 9839 9840 9841 9842 9843 9845 9846 9847 9848	034470 034472 034474 034476 034500	000311 011100 020200 001401 104443		2\$:	SWAB MOV CMP BEQ PERR17	(R1) (R1),R0 R2,R0 3\$:V177650 :V177652 :V177654 :V177656 :V177660		
9847 9848 9849 9850 9851	034502 034506 034510 034512	062701 020501 001363 000207	000002	3\$:	ADD CMP BNE RETURN	#2,R1 R5,R1 1\$:V177662 :V177666 :V177670 :V177672		DONE? NO - LOOP YES - RETURN
9852 9853	034514	011100 020200 001401 104443		MTPC21: 1\$:	;READ,B' MOV CMP BEQ PERR17	YTESWAP-I (R1),R0 R2,R0 2\$	MODIFY,REA :V177640 :V177642 :V177644 :V177646	ID.UP	
9854 9855 9856 9857 9858 9859 9861 9862 9863 9864 9865 9866	034524 034526 034530 034532 034534	000311 011100 020300 001401 104444		2\$:	SWAB MOV CMP BEQ PERR20	(R1) (R1),R0 R3,R0 3\$:V177650 :V177652 :V177654 :V177656 :V177660		
9865 9865 9866 9867	034536 034542 034544 034546	062701 020501 001363 000207	000002	3\$:	ADD CMP BNE RETURN	#2.R1 R5.R1 1\$:V177662 :V177666 :V177670 :V177672		DONE? NO - LOOP YES - RETURN

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 294 MTPA21 MARCHING 1'S & O'S PATTERN TEST

9870 034550 9871 034550 9872 034552 9873 034554 9874 034556 9875	014100 020300 001401 104444	MTPD21: 1\$:	READ, BY MOV CMP BEQ PERR20	YTESWAP-1 -(R1),R(R3,R0 2\$	MODIFY, READ, DOWN 0: V177640 : V177642 : V177644 : V177646	
9875 9876 034560 9877 034562 9878 034564 9879 034566 9880 034570 9881	000311 011100 020200 001401 104443	2\$:	SWAB MOV CMP BEQ PERR17	(R1) (R1),R0 R2,R0 3\$:V177650 :V177652 :V177654 :V177656 :V177660	
9881 9882 034572 9883 034574 9884 034576 9885	020401 001365 000207	3\$:	CMP BNE RETURN	R4,R1 1\$:V177662 :V177664 :V177666	:DONE? :NO - LOOP :YES - RETURN

```
MTP022: SUBTST <<MTP022
                                                                                                                       REFRESH & SHIFTING DIAGONAL TEST>>
9888 034600
                                                                **********************
                                                                                           MTP022 REFRESH & SHIFTING DIAGONAL TEST
                                                                : *SUBTEST
                                                                *********
                                                                                           WE WRITE A DIAGONAL PATTERN IN MEMORY (WITH CACHE ON).

IF A REFRESH TEST WE DISTURB ALL ROWS FOR > 2 MS (WITH CACHE ON).

WE READ & CHECK FOR CORRECTNESS THE DIAGONAL PATTERN

(WITH CACHE OFF).
                                                                              (2)
                                                                             FOR EVEN := #1 TO #2 ; FOR DATA & COMPLEMENT DATA
9893

9894 034600

9895 034606

9896 034616

9897 034622

9898 034626

9899 034630

9900 034634

9901 034640

9902 034640

9903

9904

9905 034646

9907 034666

9909 034666

9910 034702

9911 034714

9912 034722

9913 034724

9914 034730

9915 034732

9916 034734

9917 034740

9918 034740

9919 034740

9919 034740

9919 034744

9920 034750

9921

9922

9923 034752

9924

9925 034764
                                                               KDIAG=8.
                      000010
                                                                                 IF EVEN EQ #1
LET R2 := ZEROS
LET R3 := ONES
                                                                                 ELSE
                                                                                    LET R2 := ONES
LET R3 := ZEROS
                                                                                 END : OF IF EVEN
FOR STRIPES := #0 TO #KDIAG-1
                                                                                                                                                   FOR THE NUMBER OF STRIPES
                                                                                     :WRITE LOOP
                                                                                                                                     :TURN CACHE ON
                                                                                     CACHON
                      104423
                                                                                     LET COUNT := STRIPES
                                                                                     LET R1 := #FIRST
                                                                                     WHILE RI LOS #LAST
                                                                                         IF COUNT LT #0 THEN LET COUNT := #KDIAG-1
IF #374 OFF.IN R1 THEN LET COUNT := COUNT - #1
                                                                                        IF COUNT NE #0

LET (R1) := R2

LET 2(R1) := R2

ELSE
                                                                                        LET (R1) := R3
LET 2(R1) := R3
END :OF IF COUNT
LET COUNT := COUNT - #1
                                                                                         LET R1 := R1 + #4
                                                                                     END OF WHILE
                                                                                     IF DIAGFLAG IS FALSE THEN SCALL REFRESH
                                                                                     READ LOOP
                                                                                      LET COUNT := STRIPES
                                                                                     LET R1 := #FIRST
                                                                                     CACHOFF
                                                                                                                                      :TURN CACHE OFF
                       104424
```

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 297
MTPO22 REFRESH & SHIFTING DIAGONAL TEST
                                                                                                        WHILE R1 LOS #LAST
     9929 035000
9930 035006
9931 035022
9932 035034
9933 035042
9934 035044
9935 035050
9936 035052
9937 035052
9938 035056
9939 035064
9941 035064
9941 035064
9942 035066
9943 035076
9944 035076
9945 035076
9946 035076
9947 035102
9948 035106
9949 035110
9950 035110
9951 035110
9951 035110
9952 035114
9953 035120
9954
9955
9956 035122
9959
9960 035154
                                                                                                             IF COUNT LT #0 THEN LET COUNT := #KDIAG-1
IF #374 OFF.IN R1 THEN LET COUNT := COUNT - #1
                                                                                                             IF COUNT NE #0
                                                                                                                LET RO := (R1)
IF R2 NE RO
PERR17
                                104443
                                                                                                                END : OF IF R2
LET RO := 2(R1)
IF R2 NE RO
PERR17
                                104443
                                                                                                                END : OF IF R2
                                                                                                            ELSE

LET RO := (R1)

IF R3 NE RO

PERR2O
                                104444
                                                                                                                END : OF IF R3
LET RO := 2(R1)
IF R3 NE RO
                                                                                                                     PERR20
                                104444
                                                                                                             END : OF IF R3
END : OF IF COUNT
LET COUNT := COUNT - #1
                                                                                                             LET COUNT := COUNT
LET R1 := R1 + #4
                                                                                                         END OF WHILE
                                                                                                END : OF FOR STRIPES
END : OF FOR EVEN
RETURN
                                000207
                                                                                 REFRESH:SUBTST <<SUBR REFRESH DELAY>>
                                                                                 ***********
                                                                                                                 SUBR
                                                                                                                                 REFRESH DELAY
                                                                                  : *SUBTEST
                                                                                  **********
                                                                                                 :DISTURB EACH ROW FOR > 3.2 MS
FOR RO := WFIRST TO WFIRST+374 BY #4
      9961
9962
9963
9964
9965
9966
9967
9968
9969
9971
9972
9973
9976
9976
9977
               035154
035160
035164
035176
035202
                                                                                                 CALL REFSUB
END : OF FOR RO
LET RO := #FIRST+BIT14
                                004737 035224
                                                                                                  WHILE RO LOS #LAST+BIT14+374
                                                                                                     CALL REFSUB
LET RO := RO + #4
                                 004737
                                                035224
                                                                                                 END OF WHILE
                                000207
012704
062700
005140
005120
005110
005110
077405
162700
000207
                                                                                                                 #640,R4
#2,R0
-(R0)
                                                 000640
                                                                                                                                                                  :TIME FOR A > 3.2 MS LOOP
                                                                                 REFSUB:
                                                                                                 MOV
                                                                                                  ADD
                                                                                 15:
                                                                                                  COM
COM
COM
COM
SOB
SUB
                                                                                                                 (R0)+
(R0)
(R0)
R4.1$
                                                 000002
                                                                                                  RETURN
```

9982	035254			**SUBTES	******* ST	MTPA24 FAS	FAST GALLOPIN T GALLOPING PATTER	N TEST
9983 9984 9985 9986 9987 9988 9989 9991 9991 9993 9994 9995 9998 9999 10000 10001 10003	035254 035256 035260			*****	*(1) *(2) *(3) *(4) *(5) *(6) *(7) *(8) *(10) *REGIST *RO *R1 *R2 *R3 *R5	TAL TEST (IN THIS TEST WISTORED AT LITEST BEGINS (LETS NAME LETS NAME TO SWAPS BYTES READS 'A', 'B' = 'B'+4 REPEATS STEEND OF THE REPEATS STAFTER EXECULAND STEPS 1 ERS ARE USED TEST DATA 'A' 'B' BAKPAT SWAPAT LAST	CLUDING SETUP) IS RITES THE MEMORY W DCATION BAKPAT AT LOWEST LOCATIO IT 'A') HE 1ST LOCATION IN FOR LOCATION 'A'. READS 'B' 00 (ADDS 64 DOUB PS 5 AND 6 UNTIL 'BANK A+2 EPS 3-8 UNTILL 'A' TING THE TEST DAT -9 ARE REPEATED AS FOLLOWS	AS FOLLOWS ITH A BACK GROUND PATTERN IN BEING TESTED I THE ROW/COLUMN UNDER TEST AS 'B'. ILE WORDS TO 'B') B' IS GREATER THAN THE REACHES THE END OF THE BANK 'A IS COMPLEMENTED
10005					;NOTE T	HE PATTERN S	TARTS AT MTPB24!!!	1111111111111
10011	033202	011100 020004 001401 104447		1\$:	;UIPAR' MOV CMP BEQ PERR23	S (R1),R0 R0,R4 2\$: V177640 : V177642 : V177644 : V177646	READ 'A' CHECK 'A' BR IF OK REPORT ERROR
10014 10015 10016	035264 035266 035270 035272	011200 020003 001401 104450		2\$:	MOV CMP BEQ PERR24	(R2),R0 R0,R3 3\$: V177650 : V177652 : V177654 : V177656	READ 'B' CHECK 'B' BR IF OK REPORT ERROR
10017 10018 10019 10020	035274 035300 035302 035304 035310	062702 020205 101764	000400	3\$:	ADD CMP BLOS	#400_R2 R2_R5 1\$; V177660 ; V177664 ; V177666	;BUMP 'B' ;AT END YET? ;BR IF NO
10022 10023	035304 035310	062701 000137	000002 035314		ADD	#2,R1 a#MTPB24	: V177670 : V177674	:BUMP 'A' :GOTO V177260

10026 035314	MTPB24: SUBTST		B>>
10027 10028 035314 010411 10029 035316 020105 10030 035320 001001 10031 035322 000207 10032 035324 000137 0353	SDPAR' MOV CMP BNE RETURN 30 1\$: JMP	S R4,(R1) ;V172260 R1,R5 ;V172262 1\$;V172264 ;V172266 a#MTPC24 ;V172270	;WRITE 'A' ;DONE? ;BR IF NO ;YES - RETURN ;GOTO V172360
10033 10034 035330	MTPC24: SUBTST	< <mtpc24 c<="" fast="" gallop="" part="" td=""><td>T C>></td></mtpc24>	T C>>
10035 10036 035330 010102 10037 035332 011100 10038 035334 020004 10039 035336 001401 10040 035340 104447 10041 035342 000137 0352	;*************************************	S R1,R2 (R1),R0 R0,R4 1\$ 200 201 201 201 201 201 201 201	RESET 'B' < 'A' READ 'A' CHECK 'A' BR IF OK REPORT ERROR GOTO V177660

10044 035346		**SUBTEST	< <mtr><<mtr>MTP025MTP025INTERRU</mtr></mtr>	INTERRUPT ENABLE TEST>> ***********************************
10045 035346 00503 10046 035352 00503 10047 035356 01273 10048 035364 00473 10049 035370 01273 10050 035376 01273 10051 035402 01273 10052 035410 00473 10053 035414 10443 10054 035416 00573 10055 035422 00573 10056 10057 035426 10450 10058 035430 00573 10059 035440 00046	71 000002	CALL MOV MOV CALL ECCTINI TST TST NONE - ENATSBE	TSTDAT+2 #TSTDAT, SOURCE CHKGEN #3, NOPAR #TÉSTADD, R1 #1\$, PARTHERE MTPA25 T a(R1) a2(R1) GOOD - ACCESS F a(R1)	;GENERATE CHECKBITS ON 0,,0 ;SETUP PARITY ACTION ;FIRST TEST ADDRESS ;SETUP TRAP DESTINATION ;WRITE DATA & CHECKBITS ;INITIALIZE 1 SELECTED MK11 CSR ;ACCESS LOCATIONS FOR DBE TRAPS OR SBE TRAPS ;DISABLE TRAPS ON SBE'S FROM 1 SELECTED CSR
10057 035426 10450 10058 035430 0057 10059 035434 0057 10060 035440 00040 10061 035442 10440 10062 035444 10063 035452 0052 10064 035456 0047 10065 035462 0127 10066 035470 0047 10067 035474 0050 10068 035500 0052 10069 035504 0047 10070 035510 0127	04 26	TST BR READCSR FATAL 2\$: INC CALL MOV CALL CLR INC CALL	27 TSTDAT MTPD25 #400, TSTDAT MTPD25 TSTDAT TSTDAT+2 MTPD25	; NONE - GOOD - SKIP ; CHECK FOR CORRECT ACTION ON SBE'S ; IN ALL 4 BYTES
10072 10073 035522 0050 10074 035526 0127 10075 035534 0047 10076 035540 0127 10077 035546 0047 10078 035552 0050 10079 035556 0127	37 002246 37 000003 002244 37 035634 37 001400 002244	MOV CALL CLR MOV CALL CLR MOV CALL MOV CALL CLR1CSR	MTPD25 TSTDAT+2 #3,TSTDAT MTPE25 #1400,TSTDAT MTPE25 TSTDAT #3,TSTDAT+2 MTPE25 #1400,TSTDAT+2 MTPE25 #1400,TSTDAT+2 MTPE25	CLEAR 1 SELECTED MK11 CSR
10080 035564 0047 10081 035570 0127 10082 035576 0047 10083 035602 1045 10084 035604 0050 10085 035610 0002 10086 10087 035612 0047 10088 035616 1044 10089 035620 0047 10090 035624 1045 10091 035626 0047 10092 035632 0002	37 035664 71 37 035706 607 37 035746	CLR RETURN MTPD25: CALL ECCIDIS CALL ENAISBE CALL RETURN	MTPB25	:INDICATE PARITY ACTION :WRITE DATA & CHECKBITS :DISABLE ECC ON 1 SELECTED CSR :CHECK FOR NO TRAPS :DISABLE TRAPS ON SBE'S FROM 1 SELECTED CSR :CHECK FOR EXPECTED TRAP

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 305 MTP025 INTERRUPT ENABLE TEST	
10095 035634 004737 035664 MTPE25: CALL MTPA25 ;WRITE 10096 035640 104471 ECCIDIS ;DISABI 10097 035642 004737 035706 CALL MTPB25 ;CHECK 10098	DATA & CHECKBITS LE ECC ON 1 SELECTED CSR FOR NO TRAPS
10100 035650 004737 035746 CALL MTPC25 CHECK 10101 035656 104507 ENAISBE CALL MTPC25 CHECK 10102 035656 004737 035746 CALL MTPC25 CHECK 10103 035662 000207 RETURN	ALIZE 1 SELECTED MK11 CSR FOR EXPECTED TRAP LE TRAPS ON SBE'S FROM 1 SELECTED CSR FOR EXPECTED TRAP
10107 035666 013771 002244 000000 MOV TSTDAT,@(R1) ;WRITE 10108 035674 104475 CB1CSR ;WRITE 10109 035676 013771 002246 000002 MOV TSTDAT+2,@2(R1) ;WRITE 10110 035704 000207	CKBITS LE ECC ON 1 SELECTED CSR FIRST 16 BITS GENERATED CHECKBITS IN 1 SELECTED CSR 2ND 16 BITS & CHECKBITS
10114 035714 005771 000000 TST a(R1) ;ACCES 10115 035720 005771 000002 TST a2(R1) 10116 035724 000207 RETURN ;NO TR	NDITION TRAP DESTINATION S LOCATIONS AP - GOOD - RETURN
10118 035726 104426 10119 035730 011137 002032 10120 035734 104024 10121 035736 10122 035744 000207 1\$: READCSR MOV (R1), ADDRESS ; SAVE ERROR +24 SET HEADER RETURN	VIRTUAL ADDRESS
10117 10118 035726 104426 10119 035730 011137 10120 035734 104024 10121 035736 10122 035744 000207 10123 10124 10125 035746 012737 035762 002300 MTPC25: MOV	TRAP DESTINATION S 1ST LOCATION AP - BAD NEWS - SKIP TRAP DESTINATION S 2ND LOCATION AP - BAD NEWS VIRTUAL ADDRESS

CZMSPAO MTPO25	MS11-L/	M/P MEMO PT ENABL	RY DIAG.	MACRO	M1113 2	6-APR-82	09:41	PAGE 3	307				254
The same of	036014					: SUBTST	< <mtp< td=""><td>A26</td><td>F</td><td>RANDOM DATA</td><td>*****</td><td>******</td><td>**********</td></mtp<>	A26	F	RANDOM DATA	*****	******	**********
10139 10140 10141 10142 10143	036014 036020 036022 036024 036026	000137 010221 010321 077005 000207	036064		18:	JMP MOV MOV SOB RETURN	2/MTP R2,(R R3,(R R0,1\$	C26		V177640 V177644 V177646 V177650 V177652	GOTO	v172360	
10145	036030				*SUBT	SUBTST	< <mtp MTPB2</mtp 	******	****	RANDOM DATA	(READ)>>	******	
10146 10147 10148 10149 10150		000137 020221 001401	036064		18:	.DSABL .ENABL JMP CMP BEQ	LSB BAMTP R2, (R 2\$	C26		v177640 v177644 v177646	GOTO	v172360	
10151 10152 10153 10154 10155	036030 036034 036036 036040 036042 036044 036046 036050 036052	104451 005127 000000 020321 001401 104451			2\$: RANODE	PERR25 COM): 0 CMP BEQ PERR25	(PC)+ R3,(R 3\$			V177640 V177646 V177650 V177652 V177654 V177656 V177660 V177662 V177664 V177670 V177672	FOR E	ERROR REPOR	RTING
10159 10160 10161	036062	005167 077015 000207	177764		3\$:	COM SOB RETURN .DSABL .ENABL	RANOD RO,1\$	D		V177664 V177670 V177672			
	036064				:****	TEST	*****	*****	****	SUBPROGRAM> ************************************	>	******	*******
10164 10165 10166 10167	036064 036070 036072 036074 036076 036102				****	CALLE	R MUST MOV MOV MOV	SFFD	DLO,R DHI,R R5 R4	3			
10168 10169 10170 10171	036064 036070 036072	073427 060305 005504 060204 062705 000240	000007			ASHC ADD ADC ADD ADD	MOV #7,R4 R3,R5 R4 R2,R4	5		: V172364			
10173 10174 10175	036076	062705 000240	001057			NOP	R2 R4 #1057			V172366 V172370 V172372 V172376		v172260	
10176	036104				MTPD2	6: SUBTST	******	*****	****	SUBSUBPROGRAM	*******	******	************
10177 10178 10179 10180 10181	7 036104 8 036106 9 036112 0 036114 1 036116	005504 062704 010503 010402 000137	047401		,,,,,,	ADC ADD MOV	R4 #4740 R5 .R3	01,R4		V172260 V172262 V172266 V172270 V172272			
10180	036114	010402 000137	036020			JMP	a#MTF	3 2 PA26+4		v172272	GOTO	V177644	

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 309 RANDOM NUMBER SUBSUBPROGRAM

```
MTP030: SUBTST <<MT0030
                                                                                                                         FLUSH OUT DBE'S>>
10184 036122
                                                                                             MT0030 FLUSH OUT DBE'S
                                                                  : *SUBTEST
                                                                   ********
                                                                                                                         ;V177640
;V177642
;V177644
;V177646
10185 036122
10186 036124
10187 036126
10188 036130
10189
                                                                                             (RO),R2
R2,(RO)+
R1,1$
                                                                  15:
                                                                               MOV
                                                                                SOB
                                                                               RETURN
                        000207
                                                                 MTP031: SUBTST <<MTP031
                                                                                                                         SOB-A-LONG TEST>>
10190 036132
                                                                  ***********
                                                                                             MTP031 SOB-A-LONG TEST
                                                                  : *SUBTEST
                                                                  ***********
10191
10192 036132
10193 036134
10194 036136
10195 036142
10196 036146
10197 036150
10198 036152
10199 036156
10200 036162
10201
10202 036164
10203 036164
10204 036170
10205 036172
10206 036176
10207 036200
10208
10209 036202
10211 036206
10212
10213
                                                                                .DSABL AMA
                                                                                                                         ;MOVE TERMINATOR
;SOB TILL RO UNDERFLOWS
;WRITE COMPLEMENT OF SOB
;READ & CHECK FOR NOT 'SOB RO, DOT'
                       000000
077001
005167
020167
                                                                                             RO.15
                                                                  15:
                                                                                COM
                                      177772
177766
                                                                                             R1,1$
                                                                                CMP
                       001403
104454
010167
005167
010200
                                                                                                                         :OK - SKIP
                                                                                BEQ
                                                                                PERR30
                                                                                             R1.18
18
R2.R0
                                      177756
177752
                                                                                MOV
                                                                                                                         : CORRECT SOB INSTRUCTION
                                                                  25:
                                                                                COM
                                                                                                                         REINITIALIZE SOB CONSTANT
                                                                                MOV
                                                                                             MOVE REGISTERS
R5.R3
(R5)+
                                                                                :UPDATE
                        010503
005725
010504
020537
001001
000207
                                                                                MOV
                                                                                TST
                                                                                                                         :BUMP (SAFELY) BY 2
                                                                                             R5,R4
R5,a/LINK1
3$
                                                                                MOV
                                                                                CMP
                                                                                                                          : DONE?
                                      002516
                                                                                                                          :NO - SKIP
                                                                                RETURN
                                                                                                                          :YES
                                                                                             -(R3),-(R4)
3$
1$
                        014344
001376
000752
                                                                  3$:
                                                                                MOV
                                                                                BNE
                                                                                BR
                         000056
                                                                  SOBLENGTH=.-MTP031
                                                                                .ENABL AMA
```

	CZMSPA0	MS11-L/M/P	MEMORY	DIAG.	MACRO	M1113	26-APR-82	09:41	PAGE	311	
--	---------	------------	--------	-------	-------	-------	-----------	-------	------	-----	--

10241 036210	MTP032: SUBTST < <mtp032 recovery="" test="" write="">> **SUBTEST MTP032 WRITE RECOVERY TEST</mtp032>
10242 10243 10244 10245	THE TEST ACTUALLY EXECUTED ALREADY IN THE MEMORY UNDER TEST. THIS CODE INSURES THAT IT CHANGED MEMORY TO HAVE 1/2 BANK OF #5141 WHICH IS A "COM -(R1)" INSTRUCTION AND 1/2 BANK OF #110 WHICH IS A "JMP (R0)" INSTRUCTION.
10246 10247 036210 012401 10248 036212 020102 10249 036214 001401	1\$: MOV (R4)+,R1 ;V177640 ;GET DATA FROM LOWER 1/2 BANK CMP R1,R2 ;V177642 ;IS IT #5141? BEQ 2\$;V177644 ;YES - SKIP PERRO2 ;V177646 ;NO - TAKE ERROR TRAP ;V177650 ;LOOP FOR 1/2 BANK MOV a#LINK1,R3 ;V177652 ;RESTORE LOOP SIZE 3\$: MOV (R4)+,R0 ;V177656 ;GET DATA FROM UPPER 1/2 BANK
10250 036216 104430 10251 036220 077305 10252 036222 013703 002516 10253 036226 012400	2\$: SOB R3.1\$:V177650 :LOOP FOR 1/2 BANK MOV @#LINK1.R3 :V177652 :RESTORE LOOP SIZE
10254 036230 020005 10255 036232 001401	MOV a/LINK1.R3 ;V177652 ;RESTORE LOOP SIZE 38: MOV (R4)+,RÓ ;V177656 ;GET DATA FROM UPPER 1/2 BANK CMP RO.R5 ;V177660 ;IS IT #110? BEQ 4\$;V177662 ;YES - SKIP PERRO1 ;V177664 ;NO- TAKE ERROR TRAP 48: SOB R3.3\$;V177666 ;LOOP FOR 1/2 BANK
10256 036234 104427 10257 036236 077305 10258 036240 000207	48: SOB R3.38 ;V177666 ;LOOP FOR 1/2 BANK

10261	036242			MTP033:	SUBTST	< <mtp033< td=""><td>BRANCH GOBBLE TEST>></td></mtp033<>	BRANCH GOBBLE TEST>>
a will train				*SUBTES		MTP033 BRANCH	GOBBLE TEST
10262 10263 10264 10265 10266 10267 10268 10269 10271	036242 036244 036246 036250 036252 036254 036256	000000 000000 000261 105511 100402 105212 000773		BGTEST: BRGOBB:	DSABL	(R1) 1\$ (R2) BRG088	:MOVE TERMINATOR :TEST WORD (TWO BYTES) :SET CARRY (TO BE ADDED TO 'BGTEST') :INCREMENT LOW BYTE OF 'BGTEST'' :BRANCH WHEN BIT? IS SET :INCREMENT HIGH BYTE OF 'BGTEST'' :LOOP 128 TIMES
10270	036260 036262 036264	102401 104461		15:	:NOW CHE BVS PERR35	CK FOR CORRECT	CONDITION CODES ;BR IF V-BIT SET (SHOULD BE) ;NO - REPORT ERROR AND ABORT TEST ;COND CODES NOT EQUAL TO 1010 ;CLEAR V-BIT
10275 10276 10277 10278 10279 10280 10281	036264 036266 036270 036272 036274 036276 036300 036302 036306 036310 036312 036314 036316	000242 105212 103402 102001 100401 104461		2\$:	CLV INCB BCS BVC BMI PERR35	(R2) 3\$ 3\$ 4\$	CLEAR V-BIT INCREMENT HIGH BYTE OF 'BGTEST' ONCE MORE BR IF C-BIT SET (SHOULD NOT BE) BR IF V-BIT CLEAR (SHOULD NOT BE) BR IF N-BIT SET (SHOULD BE) NO - REPORT ERROR AND ABORT TEST COND CODES NOT EQUAL TO 1010
10282 10283 10284 10285 10286 10287	036300 036302 036306 036310	010701 162701 010102 005202	000036	4\$: 5\$:	;UPDATE MOV SUB MOV INC	TEST POINTERS PC,R1 #5\$-BGTEST,R1 R1,R2 R2	
10288 10289 10290 10291 10292	036312 036314 036316	010503 005725 010504			;UPDATE MOV TST MOV	MOVE REGISTERS R5.R3 (R5)+ R5,R4	;BUMP (SAFELY) BY 2
10294 10295 10296 10297 10298	036324 036326	020537 001001 000207	002516		;DONE? CMP BNE RETURN	R5, a//LINK1 6\$;DONE? ;NO - SKIP ;YES - RETURN
10299 10300 10301 10302 10303 10304 10305	036330 036332	014344 001376 005011 000743 000076		6\$:	:MOVE CO MOV BNE CLR BR H=MTPO	DDE 1 LOCATION -(R3),-(R4) 6\$ (R1) BRGOBB 33 AMA	CLEAR TEST WORD 'BGTEST"; RUN MOVED CODE AGAIN

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 314 MTP033 BRANCH GOBBLE TEST

MILOSS	DIVALLI	GOODLE IESI					
10307	036340		MTP034	: SUBTST	< <mtp034< td=""><td>SOFT ERROR - BACKROUND PATTERN TES</td><td>ST>></td></mtp034<>	SOFT ERROR - BACKROUND PATTERN TES	ST>>
			*SUBT	EST	MTP034 SOFT	ERROR - BACKROUND PATTERN TEST	
10308 10309	036340 036342	010220 077102	18:	MOV SOB RETURN	R2,(R0)+ R1,MTP034	: V177640 : V177642 : V177644	
10310 10311 10312	036346 036350	012401 020102	2\$:	MOV CMP BEQ PERRO2	(R4)+,R1 R1,R2 3\$	v177646 v177650	
10313 10314	036352 036354	001402 104430		PERRO2	3\$: V177652 : V177654	
10315 10316 10317	036356 036360 036362	000240 077306 000207	3\$:	NOP SOB RETURN	R3,2\$; v177660 ; v177662	

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 315 MTP034 SOFT ERROR - BACKROUND PATTERN TEST

10319 036364	MTP035:SUBTST < <mtp035 case="" noise="" parity="" test="" worst="">> :**SUBTEST MTP035 WORST CASE NOISE PARITY TEST</mtp035>
	*SUBTEST MTP035 WORST CASE NOISE PARITY TEST
10320 036364 012737 000003 002074	MOV #3, NOPAR ; SET PARITY TRAPS TO RETURN TO 'PARTHERE'
10322 036372 10323 036376 012737 000005 002146 10324 036404 104425	FOR RO := #FIRST TO #LAST BY #4000 MOV #BIT2:BITO,CSR ;SET WRITE WRONG PARITY & PAR. TRAPS INTO CSR LOADCSR
10325 036406 012737 036442 002300	MOV #1\$,PARTHERE MOV (RO),(RO) ;WWP TEST LOCATION TST (RO)
10327 036416 005710 10328 036420 010037 002032 10329 036424 104050	MOV RO ADDRESS ERROR +50
10330 036426 004737 057476 10331 036432 032763 002000 002652 10332 036440 001002	CALL PERBNK BIT #BIT10, CONFIG+2(R3) BNE 2\$ 1\$: READCSR
10333 036442 104426 10334 036444 104512 10335	ERRGEN
10336 036446 104503 10337 036450 011010 10338 036452 012737 000001 002146	2\$: CLR1CSR MOV (RO),(RO) ;CLEAR WRONG PARITY IN MEMORY MOV #BITO,CSR
10339 036460 104425 10340 036462 012737 036474 002300 10341 036470 005710	LOADCSR MOV #3\$, PARTHERE TST (RO)
10342 036472 000405 10343 036474 010037 002032 10344 036500 104050	3\$: MOV RO ADDRESS ERROR +50
10345 036502 004737 057476 10346 036506 10347	CALL PERBNK 4\$: END; OF FOR
10347 10348 036520 005037 002074 10349 036524 000207	CLR NOPAR ; RESET PARITY TRAP ACTION

035 WOR	RST CA	SE NOIS	E PARITY T				9:41 PAGE 316		
0351 036	5526				*SUBTEST	*****	MTP036 CORREC	TION CODE	
0352 0353 0354						THIS T	EST CHECKS TO CORRECTED IND	SEE THAT	FROM A ZERO TO A ONE AND
0356 036 0357 036 0358 036	6526 6530 6534	104424 105037 104513	002262		Č	ACHOFF LRB PAS BREG	FLG		FROM A ZERO TO A ONE AND TURN OFF CACHE CLEAR PASFLG ENABLE CHECK/SYNDROME BIT REGISTER THEREMENT LOOP COUNTER
0356 036 0357 036 0358 036 0359 036 0360 036 0361 036 0362 036 0363 036 0364 036 0365 036 0366 036 0367 036 0371 036 0371 036 0372 036 0373 036 0374 036 0376 036 0377 036 0377 036 0378 0378 036 0379 036	6536 6536 6542 6546 6552 6562					LET RA	ASFLG :B= PASFL := #-1 TNO := #0 ASFLG EQ #1 R5 := #1	G + #1	INCREMENT LOOP COUNTER INDEX TO SINGLE BIT ERROR TABLE CLEAR INNER LOOP COUNTER SELECT DATA TO BE CORRECTED BY PASSNO DATA=0;BIT TO BE CORRECTED IS A ONE
0365 036 0366 036 0367 036 0368 036	6566 6570 6574 6574					END REPEAT	R5 := #177776		:DATA=177776;BIT TO BE CORRECTED IS A 2
0369 0370 030 0371 030	6574 6600	005237	002320			INC	BITNO R4 := R4 + #1	P4)	:INCREMENT BIT POINTER :POINT TO NEXT SET OF CHECK BITS :GET NEXT SET OF CHECK BITS
0373 030 0374 030 0375 030	6606 6612 6616	072227 052702	000005 000004			ASH BIS LET	#5,R2 #BIT2,R2 CSR := R2	N42	INCREMENT BIT POINTER POINT TO NEXT SET OF CHECK BITS GET NEXT SET OF CHECK BITS SHIFT TO LINE UP IN CSR ENABLE DIAG MODE GET CHECK BITS TO BE WRITTEN LOAD CSR WITH DATA WRITE DATA TO TEST ADDRESS CORRECT SBE WAS DATA CORRECTED??? MOV ERROR INFORMATION IN
0376 03 0377 03 0378 03	6622 6624 6626	104425				LET	(R1) := R0 (R1) (R1) NE R5 LET ADDRESS :=		; WRITE DATA TO TEST ADDRESS ; CORRECT SBE
0380 03 0381 03 0382 03	6634 6642 6646					1	LET ADDRESS := LET CHECK := R2 LET TSTDAT := R LET TSTDAT+2 :=	#60000 5	MOV ERROR INFORMATION IN
0383 03 0384 03	6656	104052				END	ERROR +52	(R1)	NO ERROR
0386 03 0387 03	6660 6662	005011				CLR	(R1) PASFLG EQ #1		CLEAR LUT
0388 03 0389 03	6674	006305				ELS	ASL R5		SHIFT BITNO TO THE LEFT
0391 03 0392 03	6700	000261 006105				FND	SEC ROL R5		ROTATE LEFT
0392 03 0393 03 0394 03 0395 03	6702 6712	005100				COM R	BITNO EQ #16.		COMPLEMENT DATA AND REPEAT CUNTIL 2 PASSES ARE COMPLETE!
0395 03 0396 03 0397 03 0398 03 0399 0400	6652 6656 6660 6660 6662 6672 6672 6670 6702 6702	104503 104423 000207				CLR1CSR CACHON RETURN	PASFLG EQ #2		CLEAR CSR TURN CACHE
0400					:		MS11-P SINGLE	BIT ERROR	CHECK BIT TABLE
10401	6732 6735 6740	002 031 026	007 032 020	037 025 057	PTABLE:	.BYTE	2,7,37,31,32,2	5,26,20,5	57,51,52,45,46,40,75,70

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 316-1 MTP036 CORRECTION CODE TEST

036743 051 052 045 036746 046 040 075 036751 070

.

```
CHECK ECC DISABLE TEST>>
                                                   MTP037: SUBTST <<MTP037
10404 036752
                                                   ********
                                                                        MTP037 CHECK ECC DISABLE TEST
                                                   *SUBTEST
                                                   ************
10405
10406
10407
10408
10409 036752
10410 036754
10411 036764
10412 036764
10413 036766
10414 036774
10415 036776
10416 037000
10417 037004
10418 037010
10419 037016
10420 037020
10421 037020
10422 037022
10405
                                                                THIS TEST CHECKS THAT ECC CAN BE DISABLED AND THAT NO CORRECTION TAKES PLACE WITH ECC DISABLED.
                                                                                              :TURN OFF CACHE
:GOOD DATA FOR ERROR PRINT OUT
:CLEAR CHECK BIT FIELD
                                                              CACHOFF
                  104424
                                                              LET GOOD := #0
LET CHECK := #0
                                                                                               ENABLE SYNDROME/CHECK BIT REGISTER
                                                              CB1CSR
                   104475
                                                              LET CHECK := #100
                                                                                               :SBE CHECK BITS
                                                                                              WRITE CHECK BITS TO CB REGISTER
                                                              CB1CSR
                   104475
                                                              LET (R1) := #0
                                                                                              WAS CORRECTION MADE????
                                                              IF (R1) NE #0
                                                                  LET BAD := (R1)
                                                                  LET ADDRESS := #60000;
ERROR +37
                   104037
                                                              END
                                                              CACHON
                                                                                              TURN ON CACHE
                                                              RETURN
```

10425 037024	MTP041: SUBTST < <mtp041 address="" bit="" csr="" double="" error="" on="" test="" to="">> :*SUBTEST MTP041 ADDRESS TO CSR ON DOUBLE BIT ERROR TEST</mtp041>
10426 10427 10428 10429	THIS TEST CHECKS TO SEE IF THE CORRECT ADDRESS APPEARS IN CSR BITS 5-11 ON A DOUBLE ERROR.
	LET R4 := BANK ASH #9. R4 BIC #^C7740.R4 LET R0 := #-40 LET R1 := #FIRST - #4000 ; GET LOW ADDRESS IN BANK LET PASFLG :B= #0 CLR CHECK CB1CSR REPEAT INCR PASFLG :INC LOOP COUNTER ; GET STARTING BANK NUMBER ; SHIFT INTO POSTION TO MATCH ADDRESS IN CSR ; CLEAR OFF EXTRANEOUS BITS ; INIT CSR ADDRESS TO 0 - 1K (BIT 5 = 1K ADD.) ; INIT CSR ADDRESS IN BANK ; INIT PASFLG ; CLEAR CHECK BIT FIELD TO BE LOADED ; ENABLE CHECK/SYNDROME BIT REGISTER ; ENABLE CHECK/SYNDROME BIT REGISTER
10435 037054 10436 037060 005037 002310 10437 037064 104475	CLR CHECK ; INIT PASFLG ; CLEAR CHECK BIT FIELD TO BE LOADED ; ENABLE CHECK/SYNDROME BIT REGISTER
10438 037066 10439 037066 105237 002268 10440 037072	INCB PASFLG :INC LOOP COUNTER LET RO := RO + #40 ;INC CSR ADDRESS TO BE EXPECTED LET R1 := R1 + #4000 ;INC CSR ADDRESS TO BE EXPECTED
10442 037102 10443 037110 104475 10444 037112	CB1CSR ; WRITE DOUBLE ERROR CHECK BITS CB1CSR ; WRITE DOUBLE ERROR CHECK BITS LET (R1) := #0 ; WRITE DATA AND D.F. CHK BITS AT A=0
10445 037114 104503 10446 037116 005711 10447 037120 104426	CLR1CSR TST (R1) READ ADDRESS TO GET DOUBLE ERROR READCSR LET R5 := CSR
10448 037122 10449 037126 042705 17003 10450 037132	BIC #^C7740.R5 LET R2 := R0 ;GET CORRECT ADDRESS ADD R4,R2 ;ADD STARTING BANK TO DOUBLE BIT ADDRESS
10451 037134 060402 10452 037136 000240 10453 037140	GET CORRECT ADDRESS ADD R4,R2 ADD STARTING BANK TO DOUBLE BIT ADDRESS DEBUG AIDE DEBUG AIDE DO ADDRESSES AGREE? LET BAD := R2
10455 037150 10456 037154 104455 10457 037156	LET GOOD := R5 PERR31 :NO ERROR END
10458 037156 10459 037160 104475 10460 037162	CB1CSR : ENABLE CHECK/SYNDROME BIT REGISTER UNTILB PASFLG EQ #16. ; DO 16K AT A TIME
10458 037156 10459 037160 104475 10460 037162 10461 037172 104503 10462 037174 000207 10463	CLR1CSR RETURN :

10466 037176				MTP042: SUBTST < <mtp042 :*subtest="" extended<="" mtp042="" th=""><th>EXTENDED ADDRESS TO CSR ON ERROR TEST>> A ADDRESS TO CSR ON ERROR TEST</th></mtp042>	EXTENDED ADDRESS TO CSR ON ERROR TEST>> A ADDRESS TO CSR ON ERROR TEST
10467 10468 10469 10470				THIS TESTS THE EXTENDED UNIBUS CSR BY CAUSING A SINGLE ERROR. FOR THE PROPER ADDRESS IN THE	S ADDRESS IN THE , ENABLING BIT # 14, THEN CHECKING CSR.
10471 10472 037176	104424	177607		CACHOFF	:TURN OFF CACHE MEMORY :GET BANK NUMBER TO FIGURE OUT EXTENDED ADDRESS
10475 037210	042704 072427 052704 062737	177607 000002 040000 000400	172352	BIC #^c170,R4 ASH #2,R4 BIS #BIT14,R4 ADD #400,KIPAR5 LET PASFLG :B= #0	TURN OFF CACHE MEMORY GET BANK NUMBER TO FIGURE OUT EXTENDED ADDRESS CLEAR OFF LOWER BITS SHIFT TO LINE UP WITH CSR SET EXTENDED ADDRESS BIT SET UP PAR TO POINT TO TOP OF A BANK INIT LOOP COUNTER RS GETS THE BANK NUMBER CLEAR ALL BUT THE LOWER BITS ROTATE INTO POSTION SET UP SBE INDICATOR; DATA TO BE EXPECTED ENABLE CHECK/SYNDROME BIT REGISTER INCR LOOP COUNTER
10480 037236 10481 037242 10482 037246 10483 037252	042705 072527 052705 104513	177770 000011 000020		BIC M^C7,R5 ASH M9.,R5 BIS MBIT4,R5 CBREG REPEAT INCB PASFLG LET CSR := #104 LOADCSR	CLEAR ALL BUT THE LOWER BITS ROTATE INTO POSTION SET UP SBE INDICATOR ;; DATA TO BE EXPECTED RENABLE CHECK/SYNDROME BIT REGISTER
10485 037254	105237	002262		INCB PASFLG	:INCR LOOP COUNTER :WRITE CHECK BITS TO CSR WITH DIAG MODE :LOAD CSR WITH DATA
10487 037266	104425			1 ET (D1) #0	· LIDY 7EDOS AT A=O AND SINGLE EPROP HITS
10490 037274 10491 037276 10492 037300 10493 037306 10494 037314	104503 005711 104426 042737	020000	002146	CLRICSR TST (R1) READCSR BIC #BIT13,CSR IF CSR NE R5 THEN LET BAD := CSR	CLEAR CSR READ A=0; DATA BIT O SHOULD BE CORRECTED TO A 1 READ CSR FOR DATA CLEAR POSSIBLE INHIBIT MODE IN DATA "CSR" HAS SINGLE ERROR BITS SET IN CSR?
10495 037322 10496 037326 10497 037330 10498 037330	104023			LET GOOD := R5 ERROR +23	
10497 037330 10498 037330				END LET CSR := #40000	WRITE EUB BIT TO CSR
10499 037336 10500 037340 10501 037342	104425 104426 042737	020000	002146	LOADCSR READCSR BIC #BIT13,CSR IF CSR NE R4 THEN LET BAD := CSR LET GOOD := R4 SET HEADER ERROR +23	READ FOR CORRECT EXTENDED UNIBUS ADDRESS CLEAR INHIBIT MODE POINTER IN DATA READ EUB ADDRESS
10506 037376	104023			END	
10502 037350 10503 037356 10504 037364 10505 037370 10506 037376 10507 037400 10508 037400 10509 037402 10510 037406 10511 037412 10512 037414 10513 037424 10514 037426 10515 037430 10516	062705 104513	000740		LET (R1) := #0 LET R1 := #137776 ADD #740,R5 CBREG UNTILB PASFLG EQ #2 CLR1CSR CACHON RETURN	CLEAR LUT SET UP NEW ADDRESS ADD TO GET NEW ADDRESS ENABLE CHECK/SYNDROME BIT REGISTER LOOP 2 TIMES CLEAR CSR TURN ON CACHE
10514 037426 10515 037430 10516 10517	104503 104423 000207			CACHON	TURN ON CACHE

10519	037432					TE CLEARS SINGLE BIT ERROR TEST>> TE CLEARS SINGLE BIT ERROR TEST
10520 10521 10522 10523						WRITE BYTE AND THAT THE CORRECT CHECK BITS WILL TE.
10528	037432 037434 037436 037442 037446	104424 104513 105037	002262		CACHOFF CBREG CLRB PASFLG LET R2 := R1 + #1 LET R4 := #1 REPEAT INCB PASFLG	:TURN OFF CACHE :ENABLE CHECK/SYNDROME BIT REGISTER :CLEAR LOOP COUNTER :R2 POINTS TO HIGH BYTE :INITIAL DATA = 1
10530 10531	037452	105237	002262			INCREMENT LOOP COUNTER WRITE CHECK BITS CORRESPONDING TO DATA OF O
10532 10533	037464	104425			LET CSR := #604 LOADCSR LET (R1) := R4 CLR1CSR LET (R2) :B= #377	WRITE CSR
10534 10535	037466 037470	104503			CLR1CSR	WRITE CSR TO NORMAL MODE
10536 10537 10538 10539 10540 10541 10543	037472	104426 042737	177757	002146	BIC #^C20,CSR IF CSR NE #20 LET GOOD := #20	WRITE CSR WRITE DATA OF 1 CREATING A SINGLE BIT ERROR WRITE CSR TO NORMAL MODE WRITE BYTE OF WORD READ CSR SEE IF SBE INDICATOR IS SET IS SBE SET?:???
10541	037524	104060			LET BAD := CSR ERROR +60	
10543 10544	037534 037534	104513 005711			END CBREG TST (R1)	WRITE CSR TO DIAG MODE
10545	037536	104426			READCSR	READ CSR
10547 10548 10549 10550	037476 037500 037506 037516 037524 037532 037534 037536 037540 037560 037560 037560 037574 037602	104426 042737	174037	002146		WRITE CSR TO DIAG MODE READ SAO FOR CORRECT CHECK BITS READ CSR MASK OUT CHECK BIT FIELD WERE CORRECT CHECK BITS GENERATED????
10551	037574	104061			LET BAD := CSR FRROR +61	
10553	037604				END	POINT TO HIGH BYTE AND REPEAT
10555	037606	005302			DEC R2 LET R4 := #400	BIT O OF HIGH BYTE
10553 10554 10555 10556 10557 10558	037604 037604 037606 037612 037622 037624	104423 000207			UNTILB PASFLG EQ #2 CACHON RETURN	TURN ON CACHE

	037626			*SUBTEST	MTP044 SHIFTING CHE	TING CHECK BITS THROUGH THE CSR TEST>> CK BITS THROUGH THE CSR TEST
10561 10562 10563 10564 10565				THIS TEST OF SHIFTING CORRECT PATE FIELD OF AL	HECKS THE ABILITY TO RE A ONE BIT THROUGH A FI TERNS. THE TEST IS THEN L ONES.	AD AND WRITE CHECKBITS INTO MEMORY ELD OF ZEROS. THE CSR IS READ FOR THE I REPEATED ON A ZERO BIT THROUGH A
10568 10569 10570 10571	037634 037640 037642	104424		CACHO LET R LET R CB1CS LET R	PASFLG :B= #0 15 := #174037 SR 12 := #46	:TURN OFF CACHE :INIT PASFLG :CHECK BIT MASK FOR CSR :ENABLE CHECK/SYNDROME BIT REGISTER :SET UP INITIAL CSR DATA
10574				LE LE	T PASFLG :B= PASFLG + A	CHK BITS = 1 DISABLE ECC:DIAG CHK SET INIT PASSNO(INNER LOOP COUNTER)
10577 10578 10579 10580 10581	037652 037656 037656 037662 037664 037670 037672 037674 037700 037702 037704 037706 037712	104425			LET PASSNO := PASSNO + LET R4 := R2 LET CSR := R2 LOADCSR LET (R1) := #0	#1 :INC LOOP COUNTER :COPY R2 TO R4 :GET CSR DATA TO BE WRITTEN :WRITE SBE CHECK BITS TO CSR :WRITE DATA AND CHECK BITS AT A=0 :COMPLEMENT MASK :SAVE R5 ON STACK :CREATE AN XOR FUNCTION
10583 10584 10585 10586 10587	037674 037676 037700 037702 037704	005105 010546 040416 040504 052604			LET PASSNO := PASSNO + LET R4 := R2 LET CSR := R2 LOADCSR LET (R1) := #0 COM R5 MOV R5,-(SP) BIC R4,(SP) BIC R5,R4 BIS (SP)+,R4 LET CSR := R4 LOADCSR READCSR LET R3 := CSR BIC #BIT13,R3 IF R3 NE R4 THEN LET ADDRESS := #FIR	SAVE R5 ON STACK CREATE AN XOR FUNCTION
10588 10589 10590 10591 10592	037706 037712 037714 037716 037722	104425 104426 042703	020000		LET CSR := R4 LOADCSR READCSR LET R3 := CSR BIC #BIT13,R3	LOAD CSR WITH COMPLEMENT CHECK BITS READ CSR FOR COMPLEMENT CHECK BITS COPY CSR DATA TO R3 CLEAR ANY POSSIBLE INHIBIT MODE POINTER READ CSR FOR PROPER CHECK BITS
10593 10594 10595 10596 10597	037726 037732 037740 037744 037750	404057			LET ADDRESS := #FIR LET GOOD := R4 LET BAD := R3 SET HEADER ERROR +53	ST ERROR CALL
10598 10599 10600 10601 10602	037760 037760 037760 037762 037764	104053 005105 005711 000240 104426 040537			END	ACMDI PMPAIT MACH
10603 10604 10605 10606 10607 10608	037716 037722 037726 037726 037732 037740 037750 037750 037760 037760 037764 037764 037776 037776 040000 040006 040012 040026 040034 040036	040537	002146		COM R5 TST (R1) NOP READCSR BIC R5,CSR LET R4 := R2 BIC R5,R4 IF R4 NE CSR LET GOOD := R4 LET BAD := CSR LET ADDRESS := #FIR SET HEADER ERROR +54 END	READ CSR FOR CORRECT CHECK BITS MASK OUT CHECK BIT FIELD GET CHECK BITS THAT WERE WRITTEN MASK OUT CHECK BIT FIELD ARE CHECK BITS THE SAME?
10610 10611 10612 10613	040020 040026 040034 040036	104054			LET ADDRESS := #FIR SET HEADER ERROR +54 END	ERROR CALL

CZMSPAO	MS11-L/M/P	MEMORY DI	AG. MACRO	M1113	26-APR-82 TEST	09:41	PAGE	323-1
MTP044	SHIFTING C	HECK BITS	THROUGH 1	HE CSR	TEST			

10614 040036 10615 040040	040502	BIC R5.R2 IFB PASFLG EQ #1
10616	006302	ASL R2 ELSE
10615 040040 10616 10617 040050 10618 040052 10619 040054 10620 040056 10621 040060 10622 040062 10623 040064 10624 040066 10625 040070 10626 040072 10627 040074 10628 040076 10629 040100	005105 010546 040216 040502 052602 006302 010546 040216 040502 052602	COM R5 MOV R5,-(SP) BIC R2,(SP) BIC R5,R2 BIS (SP)+,R2 ASL R2 MOV R5,-(SP) BIC R2,(SP) BIC R5,R2 BIS (SP)+,R2 COM R5
10628 040076 10629 040100 10630 040102 10631 040102 10632 040106 10633 040116 10634 040122 10635 040132 10636 040134 10637 040136	104503 005011 104423	END LET R2 := R2 + #6 UNTILB PASSNO EQ #6 LET R2 := #3706 UNTILB PASFLG EQ #2 CLR1CSR CLR (R1) CACHON
10638 040140	000207	RETURN

SHIFT CHECK BITS AND CREATE NEW DATA FOR CSR
SELECT FUNCTION
DO A FIELD OF ZEROS--->ONES
SHIFT CHECK BITS
DO A FIELD OF ONES --->ZEROS

TAKE OUT CHECK BIT FIELD

SHIFT CHECK BITS

COMPLEMENENT DATA PATTERN

;ADD 6 SO THAT WRITE ON CSR WILL ENABLE DIAG MODE ;DO ALL CHECK BITS ;REPEAT WITH FIELD OF ONES

TURN ON CACHE

```
SYNDROMES TO CSR ON DOUBLE BIT ERROR TEST>>
                                                       MTP045: SUBTST <<MTP045
10640 040142
                                                       ************************
                                                                              MTP045 SYNDROMES TO CSR ON DOUBLE BIT ERROR TEST
                                                        *SUBTEST
                                                       ************************
10641
10642
10643
10644
10645
                                                                       THIS TEST CHECKS TO SEE IF THE DOUBLE BIT ERROR INDICATOR IS SET ON A DOUBLE BIT ERROR AND THE CORRECT SYNDROMES ARE LATCHED INTO THE
                                                                       CSR. THIS TEST IS THEN REPEATED WITH MULTIPLE ERROR CHECK/SYNDROME BITS
10645
10646 040142
10647 040144
10648 040146
10649 040152
10650 040160
10651 040166
10652 040166
10653 040172
10654 040174
10655 040176
                                                                   CACHOFF
                                                                                                      :TURN OFF CACHE
                                                                                                      : ENABLE CHECK/SYNDROME BIT REGISTER
                                                                   CBREG
                                                                   LET PASSNO := #0
                                                                                                      :CLEAR LOOP COUNTER
                                                                   LET GOOD := #3744
LET CSR := #3144
                                                                                                      GOOD DATA
                                                                                                      :DBE CHECK BITS FOR CSR
                                                                       INC PASSNO
                                002264
                                                                                                      WRITE DBE CHECK BITS TO CSR
WRITE ZEROS AND DBL ERROR CHK BITS A=0
                                                                       LOADCSR
                                                                       LET (R1) := #0
10655 040176
10656 040200
10657 040202
10658 040204
10659 040214
10660 040222
10661 040230
10662 040232
10663 040232
10664 040234
10665 040236
10666 040240
10667 040246
10669 040264
10670 040272
10671 040274
10672 040274
10673 040276
10675 040324
10676 040322
                                                                       CLR1CSR
                                                                                                      :CLEAR CSR OUT
                    104503
005711
                                                                                                      READ A=0 TO GET DOUBLE BIT ERROR WAS UNCORRECABLE ERROR BIT SET???
                                                                       TST (R1)
                                                                       READCSR
                                                                       IF WBIT15 OFF. IN CSR
                                                                            SET HEADER
                                                                            LET BAD := CSR
ERROR +63
                                                                                                      BIT NOT SET
                    104063
                    104513
104426
000240
042737
                                                                                                      : ENABLE SYNDROME BIT REGISTER
                                                                        CBREG
                                                                                                      READ CSR FOR CORRECT SYNDROME BITS
                                                                        READCSR
                                                                                                      : DEBUG AIDE
                                                                                                      :MASK SYNDROMES OUT
                                                                       BIC #*C3744.CSR
                               174033 002146
                                                                       IF CSR NE GOOD THEN
LET BAD := CSR
                                                                                                      CHECK IF DOUBLE ERROR BIT IS SET
                                                                                                      :BAD DATA
                                                                            SET HEADER
                                                                            ERROR +42
                    104042
                                                                       END
                                                                                                      :CLEAR LUT
                                                                        CLR (R1)
                    005011
                                                                   LET GOOD := #3604
LET CSR := #3004
UNTIL PASSNO EQ #2
CLR1CSR
                                                                                                      REPEAT WITH MULTIPLE ERROR SYNDROMES
                                                                                                      :MULTIPLE ERROR CHECK BITS
                    104503
104423
000207
                                                                    CACHON
                                                                    RETURN
```

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 325 MTP045 SYNDROMES TO CSR ON DOUBLE BIT ERROR TEST CHECK SINGLE BIT ERRORS WITH ECC DISABLED>> MTP046: SUBTST <<MTP046 10680 040330 ******************************* MTP046 CHECK SINGLE BIT ERRORS WITH ECC DISABLED *SUBTEST THIS TEST CHECKS TO SEE THAT FOR EACH BIT OF A DATA WORD THAT A SBE 10682 10683 IS TREATED LIKE A UNCORRECTABLE ERROR WITH ECC DISABLED AND TRAPS ARE DETECTED. 10684 10685 10686 040330 10687 040334 10688 040336 10689 040336 10690 040342 10691 040352 10693 040352 10694 040352 10694 040352 10695 040356 10696 040362 10697 040370 10698 040374 10699 040404 10700 040410 10701 040412 10702 040416 10685 CLR PASSNO CLEAR OUTER LOOP COUNTER 002264 005037 TURN OFF CACHE CACHOFF REPEAT LET PASSNO := PASSNO + #1 005000 105037 104513 CLEAR DATA CLR RO CLRB PASFLG 002262 ENABLE CHECK/SYNDROME BIT REGISTER CBREG REPEAT LET PASFLG :B= PASFLG + #1 INCREMENT LOOP COUNTER :INDEX TO SINGLE BIT ERROR TABLE LET R4 := #-1 LET NOPAR := #1 ENABLE PARITY ACTION LET BITNO := #0 CLEAR INNER LOOP COUNTER SELECT DATA TO BE CORRECTED BY PASSNO IFB PASFLG EQ #1 LET R5 := #1 :DATA=0:BIT TO BE CORRECTED IS A ONE 10701 040412 10702 040416 10703 040416 LET R5 := #177776 :DATA=177776;BIT TO BE CORRECTED IS A ZERO REPEAT LET PARCNT := #0 :CLEAR PARITY COUNTER 10704 040416
10705 040422
10706 040426
10707 040430
10708 040434
10709 040440
10710 040444
10711 040452
10712 040452
10713 040454
10714 040464
10715 040466
10716 040470
10717 040472
10718 040472
10718 040472
10719 040474
10720 040500
10721 040502
10722 040512
10723 040520
10724 040526
10725 040530
10726 040534
10729 040544 INCREMENT BIT POINTER
POINT TO NEXT SET OF CHECK BITS
GET NEXT SET OF CHECK BITS
SHIFT TO LINE UP IN CSR
ENABLE DIAG MODE
GET CHECK BITS TO BE WRITTEN INC BITNO LET R4 := R4 + #1 005237 002320 LET R2 :B= PTABLE(R4)
ASH #5.R2
BIS #BIT2!BIT1,R2 072227 000006 LET CSR := R2 :LOAD CSR WITH DATA LOADCSR 104425 WRITE DATA TO TEST ADDRESS LET (R1) := R0 IF PASSNO EQ #1 :FIRST PASS :ECC DISABLE, NO PBL ECC1DIS 104471 SECOND PASS SECC DISABLE, PBL ENABLED 104507 ENA1SBE 005711 004737 104426 CORRECT SBE TST (R1) CALL CHKTRP READCSR CHECK FOR CORRECT TRAP 040614 READ THE CSR FOR UNCORRECTABLE ERROR IF WBIT15 OFF. IN CSR :IS UNCORRECTABLE ERROR BIT SET???? LET BAD := CSR SET HEADER ERROR +45 104045

CLR1CSR

ELSE

END

CLR (R1)

SEC

IFB PASFLG EQ #1

104503

006305

000261 006105

CLEAR LUT SHIFT NEW DATA DEPENDING ON PASFLG SHIFT BITNO TO THE LEFT SET CARRY BIT AND..... ROTATE LEFT

	CZMSPAO	MS11-L/M/P	MEMORY D	IAG.	MACRO	M11	13	26-APR-82	09:41	PAGE	325-1
ı	IMTPO46	CHECK SING	LE BII EK	KUK2	MIIME	LL	D 1 21	ABLED			

10734 040554 10735 040564 10736 040566	005100	UNTIL BITNO EQ #16. COM RO UNTILB PASFLG EQ #2 UNTIL PASSNO EQ #2
10737 040576 10738 040606 10739 040610 10740 040612 10741	104503 104423 000207	CLR1CSR CACHON RETURN
10742 10743 040614 10744 040624 10745 040634 10746 040642 10747 040644	104057	CHKTRP: IF PASSNO EQ #1 IF PARCNT EQ #1 SET HEADER ERROR +57 END
10748 040644 10749 040646 10750 040656 10751 040664 10752 040666 10753 040666	104062	ELSE IF PARCNT NE #1 SET HEADER ERROR +62 END END
10754 040666	000207	RETURN

COMPLEMENT DATA AND REPEAT UNTIL 2 PASSES ARE COMPLETE!

TURN CACHE

:PASS 1 CHECK FOR NO TRAP

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 326 MTP046 CHECK SINGLE BIT ERRORS WITH ECC DISABLED

10756 040670			ST MTP047 NO CSR	NO CSR UPDATE ON SBE WITH EXSISTING DBE>> UPDATE ON SBE WITH EXSISTING DBE
10757 10758 10759 10760 10761			THIS TEST CHECKS TO WITH A SINGLE BIT EREXISTS.	SEE THAT THE CSR CONTENTS WILL NOT CHANGE RROR WHEN A DOUBLE BIT ERROR ALREADY
10762 0/0670	104424		CACHOFF	:TURN OFF CACHE :GET BANK NUMBER :SHIFT INTO PLACE
10762 040670 10763 040672 10764 040676 10765 040702 10766 040706 10767 040712 10768 040714 10769 040722 10770 040724	072427 000011 042704 170037 052704 100000 104513		CACHOFF LET R4 := BANK ASH #9. R4 BIC #^C7740,R4 BIS #BIT15,R4 CBREG LET CSR := #3144	SHIFT INTO PLACE MASK OUT UNWANTED BITS SET UP GOOD DATA ENABLE CHECK/SYNDROME BIT REGISTER CHECK BITS FOR DOUBLE BIT ERROR
10//0 060//24	104425		LET CSR := #3144 LOADCSR LET (R1) := #0	WRITE DBE CHECK BITS WRITE SBE CHECK BITS
10//1 040/20	104425		LOADCSR	WHITE SEE CHECK BITS
10773 040736 10774 040740 10775 040742 10776 040744	104503 005711 104426 042737 020000		LET CSR := #3144 LOADCSR LET (R1) := #0 LET CSR := #104 LOADCSR LET (R2) := #0 CLR1CSR TST (R1) READCSR BIC #BIT13,CSR IF CSR NE R4	WRITE SBE CHECK BITS AT ADDRESS + 4K CLEAR CSR READ DBE LOCATION READ FOR CSR DBE INDICATOR CLEAR INHIBIT MODE POINTER
10772 040734 10773 040736 10774 040740 10775 040742 10776 040744 10777 040746 10778 040754 10779 040762 10780 040770 10781 040774 10782 041002 10783 041004 10784 041004 10785 041010 10786 041012 10787 041014 10788 041022 10789 041030 10790 041036	042737 020000	002146	BIC #BIT13,CSR IF CSR NE R4 LET BAD := CSR LET GOOD := R4 SET HEADER ERROR +63	CLEAR INHIBIT MODE POINTER
10782 041002	104063		END #20 P/	CET DIT IN COOR DATA
10784 041004	052704 000020 005712		TST (R2)	SET BIT IN GOOD DATA READ SBE READ CSR FOR NO CHANGE CLEAR INHIBIT MODE POINTER
10786 041012 10787 041014 10788 041022 10789 041030 10790 041036 10791 041042	104426 042737 020000	002146	BIS #20.R4 TST (R2) READCSR BIC #BIT13.CSR IF CSR NE R4 LET BAD := CSR LET GOOD := R4 SET HEADER ERROR +51	CLEAR INHIBIT MODE POINTER
10792 041050 10793 041052	104051		ERROR +51	
10794 041052	104503 005011		END CLR1CSR CLR (R1) CLR (R2)	CLEAR 1 CSR
10796 041056 10797 041060 10798 041062	104503 005011 005012 104423 000207		CACHON RETURN	TURN ON CACHE

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 327 MISC SUBROUTINES

10800	.SBTTL MISC SUBROUTINES
10801 10802 041064	REGCOPY:SUBTST < <subr &="" copy="" r2="" r3,="" r4,r1="" r5="" ro="" to="">> :**SUBTEST SUBR COPY RO TO R4,R1 TO R3, & R2 TO R5</subr>
10803 041064 010004 10804 041066 010103 10805 041070 010205 10806 041072 000207 10807 10808 041074	MOV RO,R4 MOV R1,R3 MOV R2,R5 RETURN
10808 041074	FLIPWARN: SUBTST < <flip case="" constants="" in="" noise="" tests="" warning="" worst="">> :*SUBTEST FLIP WARNING CONSTANTS IN WORST CASE NOISE TESTS</flip>
10809 041074 10810 041076 005237 002602 10811 041102 042737 177774 002602 10812 041110 022737 000001 002602 10813 041116 001414 10814 041120 022737 000002 002602 10815 041126 001413 10816 041130 022737 000003 002602 10817 041136 001414 10818 041140 005000 10819 041142 013704 002600 10820 041146 000414 10821 041150 10822 041154 000411 10823 041156 012700 000401 10824 041162 013704 002600 10825 041166 000404 10826 041170 012700 000401 10827 041174 012704 000401 10828 041200 010037 0275322 10829 041204 010037 027536 10830 041210 010037 027562 10831 041214 010037 027576 10832 041220 10833 041222 000207	PUSH RO INC
10832 041220 10833 041222 000207	POP RO RETURN

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 328 FLIP WARNING CONSTANTS IN WORST CASE NOISE TESTS

10835 041224	BACKGND:SUBTST < <subr backgrou<="" th="" write=""><th>· 自由自由自由自由自由自由自由自由自由自由自由自由自由自由自由自由自由自由自由</th></subr>	· 自由自由自由自由自由自由自由自由自由自由自由自由自由自由自由自由自由自由自由
10836 10837 041224 104415 10838 041226 012700 060000 10839 041232 012701 040000 10840 041236 022737 000001 003752 10841 041244 001415 10842 041246 012737 000207 027404 10843 041254 012737 027400 002260 10844 041262 004737 027206 10845 041266 012737 027206 10845 041266 012737 000240 027404 10846 041274 104416 10847 041276 000207 10848 041300 10849 041306 012737 000207 177644 10850 041314 004737 027030 10851 041320 104416 10852 041322 000207	######################################	; WARNING PUTTING 'RETURN' AFTER WRITE ; RESTORE 'NOP' AFTER WRITE ; WARNING PUTTING 'RETURN' INSTRUCTION AFTER WRITE

CZMSPAO MS11-L/M/P MEMORY DIAG	MACRO M1113	26-APR-82 09:41	PAGE 329
--------------------------------	-------------	-----------------	----------

SUBR	WRITE BACKGROU	ND			
10854	041324		GETCSR: SUBTST	***********	NFORMATION FROM CONFIGURATION TABLE>> NFORMATION FROM CONFIGURATION TABLE
10855 10856 10857 10858 10859 10860 10861 10863 10864 10865	041324 013702 041330 016203 041334 000303 041336 006303 041340 042703 041344 010337 041350 000207	002102 002650 177741 002150	OUTPUTS OUTPUT MOV MOV SWAB ASL BIC MOV RETURN	: NONE : CSRNO = CSR NUME BANKINDEX,R2 CONFIG(R2),R3 R3 R3 R3 R3 R3,CSRNO	GET INDEX INTO CONFIG TABLE MOV IT INTO R3 CLEAR OFF SOME BITS SAVE CSR NUMBER

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 331 SUBR GET CSR INFORMATION FROM CONFIGURATION TABLE

10868 041352		IGURATION MAP>> IGURATION MAP IGURATION MAP
10869 041352 10870 041364 010637 041652 10871 041370 012737 041620 0000 10872 041376 012737 000340 0000 10873 041404 017700 141220 10874 041410 042737 000200 1777 10875 041416 052777 000100 1412	MOV astkb.RO	;SAVE LAST GOOD SP ;KILL ANY OLD INTERRUPT ;LOWER CPU PRIORITY TO 140 ;ENABLE KEYBOARD INTERRUPTS
10876 10877 041424 10878 041430 10879 041434 10880 041440 022737 000060 0025 10881 041446 002006	BGE NOOJ :IF FAT PAPER ON TERMINAL	GOTO 1\$
10883 041450 10884 041464 012700 000074 10885 041470 010004 10886 041472 10887 041476 10888 041502 004737 041654 10889 041506 022737 000060 002 10890 041514 002041	NOOJ: MOV #60RO MOV RO.R4 CLEAR R1.R3 TYPE MSG004	JOHPTO PCONFT
10888 041502 004737 041654 10889 041506 022737 000060 002 10890 041514 002041 10891 041516	52 CMP #60.LASTBANK BGE PCONF2 TYPE \$CRLE	;GO TYPE CONFIGURATION (1ST HALF) ;PRINT SPACE(S)
10892 041522 10893 041526 10894 041532 10895 041536 10896 041542	TYPE MSG017 TYPE MSG011 TYPE SCRLF TYPE MSG017 TYPE MSG012	;PRINT SPACE(S)
10900 041560 000417	TYPE MSG012 MOV #60.+2+2,R1 MOV R1,R3 CALL TCONFIG BR PCONF2	
10902 041562 012700 000170 10903 041566 010004 10904 041570 10905 041574 10906 041600 10907 041604 10908 041610 10909 041614 004737 041654 10910	PCONF1: MOV #120.,RO MOV RO,R4 CLEAR R1,R3 TYPE MSG014 TYPE MSG011 TYPE MSG004 TYPE MSG012 CALL TCONFIG	; SPACE
10910 10911 041620 013706 041652		RESTORE STACK
10911 041620 013706 041652 10912 041624 042777 000100 140 10913 041632 117700 140772 10914 041636 10915 041650 000207 10916 10917 041652 000000	PCONF2: MOV PCONFS,SP BIC MBIT6, a\$TKS MOVB A\$TKB,RO POP RO,TKVEC+2,TKVEC	READ CHAR TO KILL FLAG
10916 10917 041652 000000	PCONFS: 0	STACK SAVED HERE!

10920 041654					SUBTST	< <subr co<="" th="" type=""><th>ONF I GURATION>></th><th></th></subr>	ONF I GURATION>>	
				*SUBTE	ST	SUBR TYPE CO	ONFIGURATION	
10921 10922 10923 10924 10925 10926 10927 10928 10929 10930				CALL:	MOV MOV MOV MOV CALL RETURN	#N,R0 R0,R4 #K.R1 R1,R3 TCONFIG	:N=NUMBER OF CHARACTERS :BACKUP :INDEX CONSTANT :BACKUP :ACTUAL CALL :ONLY RETURN	
10928 10929 10930 10931 10932				;*****	; ***** ; ** ER/			
10933 041654	012737	000340	177776	TCONFIG	:MOV TYPE	#340.PSW MSG005	:DISABLE INTERUPTS	
10934 041662 10935 041666 10936 041674 10937 041676	032761 001403	000001	002650	1\$:	BIT BEQ TYPE	#BITO, CONFIG(R1 2\$ MSG013	:ERROR ON THIS BANK? :NO - SKIP :PRINT 'X'	
10938 041702 10939 041704	000402			2\$:	BR	3\$ MSG014	PRINT SPACE	
10940 041710 10941 041714 10942 041716 10943 041720	062701 077014 010400 010301	000004		2\$: 3\$:	ADD SOB MOV MOV	#4,R1 R0,1\$ R4,R0 R3,R1	BUMP POINTER : LOOP UNTIL DONE	

10946						******	****	
10947						** INTERLEA	VE **	
10949	041722					TYPE MSGO	07	EDDOD DESCRIPTS
10950	041726	012737	000340	177776	TCFIG1:	MOV #340	ENTRY POINT FROM),PSW ;DIS/ 12,CONFIG+2(R1)	ABLE INTERUPTS
10952	041734	032761	000340 010000	002652		BIT #BIT BNE 1\$	12,CONFIG+2(R1)	
10954	041744	032761	000002	002650		BIT #BIT	1, CONFIG(R1)	:IS THERE ANY MEMORY HERE? :BRANCH IF MEMORY PRESENT. :MOVE A BLANK IN TO BE PRINTED :BRANCH TO TYPE ROUTINE
10955	041752	112737	000040	074704		BNE 18\$ MOVB #	MSG015	MOVE A BLANK IN TO BE PRINTED
10957	041722 041726 041734 041742 041744 041752 041754 041754 041762 041764 041772 041774 042000 042006 042012 042016 042016 042020 042020 042024 042030 042030 042030 042050 042056	012737 032761 001014 032761 001004 112737 000424 112737 000420 016105 042705 000305 072527 022705 100002 062705 110537	000055	074704	18\$:	DV IOA	MSG015	BRANCH TO TYPE ROUTINE
10959	041772	000420		014104		BR 16\$		
10960 10961	041774	016105	002650 007777		1\$:	MOV CONF	FIG(R1),R5 170000,R5	GET CSR INTERLEAVE
10962	042004	000305	177774			SWAB R5 ASH #-4		
10964	042012	022705	000011			CMD #0	DS	
10965	042016	062705	000007			ADD #7	25	
10967	042024	062705	000060		2\$:	ADD #60	,R5 ISG015	:MAKE ASCII :PLUG INTO MEMORY
10969	042004 042006 042012 042020 042024 042030 042034 042040 042050 042054	110331	014104		16\$:	TYPE MSG	R5 R5 RSG015 015 #0 THEN \$RETURN	
10970	042040	062701 077054	000004			ADD #4.	CFIG1	BUMP POINTER
10972	042054	077054				SOB RO.	CFIG1	LOOP UNTIL DONE
10974	042060	010400 010301				MOV R4.1	RÍ	
10975 10976 10977						;******	*****	
10977 10978						** MEMORY	*****	
10979	0/20/2					ENABL LSB	000	
10980	042062	033761	002104	002650	TCF1G2:	BIT CPU	BIT, CONFIG(R1)	
10982	042074	001432	002652			BEQ 17\$ MOV CON	FIG+2(R1),R5	
10984	042102	033761 001432 016105 000305 042705 020527 003022				SWAB R5	7,R5	GET MEMORY TYPE CLEAR NON INTERESTING BITS
10986	042110	020527	177770 000003			CMP R5	v3	IS IT A LEGAL MEMORY TYPE
10987 10988	042062 042066 042074 042076 042102 042110 042114 042116 042124 042132 042140 042140	003022				CMP R5.7 BGT 17\$ IF #BITO SE	T.IN R5	IS IT A LEGAL MEMORY TYPE IF IF SO BRANCH!!!!!! IS IT AN ECC MEMORY???? IS IT A MS11-P OR A MS11-M??? IT IS A MS11-P
10989	042124	112777	000120	074704		IF #BIT1	SET.IN R5 #'P,MSG015	IS IT A MS11-P OR A MS11-M???
10991	042140	112737				ELSE		
10992	042142	112737	000115	074704		END	#'M,MSG015	IT IS A MS11-M
10981 10983 10984 10985 10986 10987 10988 10989 10991 10993 10994 10995	042150 042152	112737	000114	074704		ELSE	#"L,MSG015	IT IS A MS11-L
10996	042160		000114	014104		END		:
10996 10997 10998 10999	042160 042162 042170 042174	000403 112737	000040	074704	175:	BR 8\$ MOVB #	MSG015	
10999	042170				8\$:	TYPE MSG	015 #0 THEN SRETURN R1 TCFIG2	
	1100/1/64					TL MOTUD ME	MO INCH SKETUKN	

CZMSPAO SUBR	MS11-L/I	M/P MEMOI	RY DIAG.	MACRO N	11113 26	-APR-82	09:41	PAGE	335-1	
11003 11004 11005	042212 042214	010400 010301				MOV MOV .DSABL	R4,R0 R3,R1 LSB			
11010 11011 11012 11013 11014 11015 11016 11017 11018 11020 11021 11022 11023 11024 11025 11026	042214 042214 042216 042222 042230 042234 042246 042246 042256 042256 042256 042256 042272 042276 042276 042316 042316	112737 016105 032705 001414 042705 000305 022705 100002 062705 062705 110537	000040 002650 000002 170377 000011 000007 000060 074704	074704	TCFIG3: 10\$: 16\$:	****** TYPE MOVB MOV BIT BEQ BIC SWAB CMP BPL ADD ADD MOVB TYPE IF NOTA ADD SOB MOV MOV	MSG010 # MSG	SG015 S(R1) R5 00,R5 S G015 THE	R5	:MAKE ASCII :PLUG INTO MEMORY :BUMP POINTER
11029 11030 11031 11032 11033 11034 11035 11036 11037 11038 11039 11040 11041 11042 11043	042320 042324 042330 042332 042340 042342 042350 042352 042360 042364	105761 100004 112737 000407 032761 001406 112737 000402 062701 077026 010400 010301 000207	002650 000120 000100 000111	074704 002650 074704	11\$: 12\$: 13\$: 14\$: 15\$:	******* TYPE TSTB BPL MOVB BR BIT BEQ MOVB TYPE BR TYPE ADD SOB MOV MOV RETURN	MSG01: CONFI 128 #'P.M	*** 0	IG(R1)	;BANK PROTECTED? ;NO - SKIP ;PROTECTED REGION OF ECC? ;NO - SKIP ;PRINT SPACE ;BUMP POINTER ;LOOP UNTIL DONE

	11050 11051 11052 11053 11054				VECTOR	.SBTTL TO HERE ERRORS	TRAP PARITY ERROR FROM TRAPS TO 114 BUT COUNT IF NOPAR FL	********
	11055 11056					CODE	ACTION	
	11057 11058 11059 11060 11061 11062					1 2 3	PRINT UNEXPECTED PAR COUNT ERROR SET 'ABORT' / SETUP RETURN VIA 'PARTHERE	ITY TRAP "BADPC" / RETURN VIA PCBUMP
	11062 11063 042406 11064 042414	022737	000001	002074	PARITY:		#1.NOPAR	COUNTING PARITY ERRORS?
	11065 042416	001003	002070			BNE INC RTI	PARCNT	; NO - SKIP ; PARITY ERROR COUNTER + 1
Н	11066 042422 11067 042424 11068 042432 11069 042434	022737 001003 005237 000002 022737 001013	000002	002074	1\$:	CMP	#2.NOPAR	ACTION CODE = 2 ?
	11070 042442 11071 042446 11072 042452	004737 063716 042766 000002 022737 001003 013716	042614 002322 000004	000002		BNE SET CALL ADD BIC	ABORTFLAG BADSTACK PCBUMP, (SP) #BIT2,2(SP)	; YES ; FIND BAD SP.PC.PSW OFF STACK ; UPDATE RETURN PC ; SHOW FAILURE BY .NE.
	11073 042460 11074 042462 11075 042470	000002	000003	002074	2\$:	RTI	#3,NOPAR	ACTION CODE = 3 ?
	11076 042472	013716	002300			MOV	PARTHERE, (SP)	, NO - 3/1F
	11077 042476 11078 042500 11079 042504	000002	042614		3\$:	RTI CALL FATAL	BADSTACK 32	FIND BAD SP.PC.PSW OFF STACK

11082 11083 11084 11085 11086 11087 11088					VECTOR	CODE IN	(SOMETIMES) FROM TRAPS 1	NORY (HOLES) HANDLER O 4 AS FOLLOWS: I (NO NON-EXISTANT MEMORY) FLAG = 1. IF NON-EXIST MEM ERROR
11080	512	022737	000001	002076	NONEXIST			COUNTING NON-EXISTANT MEMORY ERRORS?
11090 042 11091 042 11092 042 11093 042 11094 042 11095 042	520 (522 (022737 001011 005237 022737	002066 000001			BNE INC	2\$ NEMCNT #1,NEMCNT	:NO - SKIP BUMP NON-EXISTANT MEMORY COUNTER
11093 042 11094 042	526	022737	000001	002066		CMP BNE	15	FIRST ERROR?
11095 042	536	001002	002032		10.	MOV	RO, ADDRESS	ASSUME RO CONTAINS THE ADDRESS ACCESSED
11096 042 11097 042 11098 042 11099 042 11100	544 550 554	000002 005237 012701 000002	002066 000001		1\$: 2\$:	INC MOV RTI	NEMCNT #1,R1	;BUMP NON-EXISTANT MEMORY COUNTER ;DUMMY UP R1 FOR A FORCED SOB EXIT
11101					;*****	******	***********	*************************
11102 11103 042 11104 042	556 562	004737	042614		TIMEOUT	SBTTL CALL FATAL	TRAP TIMEOUT (TRAP TO BADSTACK	;FIND BAD SP,PC,PSW OFF STACK
11105 11106					;*****	SBTTL	TRAP MEMORY MANAGEMEN	T (TRAP TO 250) HANDLER ;FIND BAD SP,PC,PSW OFF STACK
11107 042 11108 042	574	004/3/	042614		MMTRAP:	FATAL	BADSTACK 7	
111109 111110 042 111111 042	602	004737	042614		PDP1105	SBTTL CALL FATAL	TRAP RESERVED INSTRUCTOR	;FIND BAD SP,PC,PSW OFF STACK
11112 11118 11119 042	614				BADSTAC	K:SUBTST	< <find &="" bad="" pc,="" ps<="" sp,="" td=""><td>FROM STACK>></td></find>	FROM STACK>>
					*SUBTE		FIND BAD SP, PC, & PSW	FROM STACK
11120 042	2614	010637	002024		;*****	MOV	SP.BADSP	***********
11120 042 11121 042 11122 042	620	010637 062737 016637	002024 000002 000002	002024		MOV	#2.BADSP 2(SP).BADPC	
11123 046	2034	016637	000004	002030		MOV RETURN	4(SP) , BADPSW	
11124 042	2642	000207				KEIUKN		

11111	116111166				
11127 11128 11129 11130					SBTTL TRAP KERNEL TRAP HANDLER KERNEL IS A TRAP THAT COMES HERE
11132 11133	042644 042652	042766 000002	140000	000002	\$KERNEL: BIC #140000,2(SP) RTI
11134 11135 11136 11137	042654 042662	052737 000002	000001	177572	SBTTL TRAP ENERGIZE TRAP HANDLER SENERGIZE:BIS #BITO,MMRO RTI
11138 11139 11140 11141	042664 042672	042737 000002	000001	177572	SBTIL TRAP DEENERGIZE TRAP HANDLER SDEENERGIZE:BIC #BITO,MMRO RTI
11142 11143 11144 11145	042674 042700	005737 001406 013737 052737	002540	1777/4	SBTTL TRAP CACHON TRAP HANDLER SCACHN: TST CACHKN ; IS THERE A CACHE BEQ 1\$; NO - RETURN MOV CACHKN, CONTRL ; SETUP CACHE AS PER CONSTANT (USUALLY 1 = FULLY ON)
11147	042702 042710 042716	052737 000002	002540 000001	177746 177746	BIS #BITO, CONTRL ; DISABLE TRAPS (BUT NOT ABORTS) 1\$: RTI : ************************************
11150 11151 11152	042720 042724	005737 001403	002540		SBTTL TRAP CACHOFF TRAP HANDLER SCACHF: TST CACHKN ; IS THERE A CACHE? SO THE CACHKN ; IS THERE A CACHE?
11153 11154 1115	042726 042734	053737 000002	002544	177746	;DISABLE TRAPS (NOT ABORTS), FORCE MISSES, FLUSH, BYPASS BIS CACHKF, CONTRL 18: RTI

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 343 TRAP LOAD CSR TRAP HANDLER

	11158 11159 11160 11161 042736 11162 042742	013700	002150		\$LOADC:	PUSH RO, MOV CSR	CT CSR WITH DATA IN R'S ASSERT INHIBIT R1 NO.RO	CSR MODE POINTER WHEN LOADED ;SAVE REGISTERS ;CREATE CSR ADDRESS	
ı	11164 042754	005737	002526			TST PGM	S TRUE THEN GOTO 3\$:PROGRAM IN INTERLEAVED SPACE?	
	11164 042754 11165 042760 11166 042762 11167 042766 11168 042772	005737 100007 113701 042701 020137 001404 123737 001003 052737 013760	002527 177740 002150			BIC #^C	ICSR+1,R1 37,R1 CSRNO	;BRANCH IF NOT ;CHECK SECOND CSR ;CLEAR UNNECESSARY BITS ;IS THIS THE CURRENT CSR? ;BRANCH IF IT IS	
ı	11170 043000	123737	002526	002150	1\$:	CMPB PGM	ICSR, CSRNO	IS THIS THE CURRENT CSR?	
	11169 042776 11170 043000 11171 043006 11172 043010 11173 043016 11174 043024 11175 043030	001003 052737 013760 000002	020000 002146	002146 172100	2\$: 3\$:		T13,CSR R,CSRADD(RO) RO	SET THE INHIBIT MODE POINTER TO 1ST 16K LOAD THE CSR RESTORE REGISTERS	
	11176 11177 11178 11179 043032	000002			\$READC:	.SBTTL TRA	P READ CSR TRAP		
	11180 043034 11181 043040 11182 043046 11183 043050	013700 016037 000002	002150 172100	002146		MOV CSR	RNO,RO RADD(RO),CSR	;READ IT	

11185 11186 043052 11187 043060 11188 043064 11189 043070 11190 043072 11191 043076 11192 043100 11193 043104 11194 043110 11195 043114 11196 043116 11197 043124 11198 043126 11199 043132 11200 043140 11201 043142 11202 043146	012700 063700 005002 005737 100007 113703 042703 020337 001404 123737 001002 012702 0022737 001403 004737	172100 002150 002526 002527 000200 002150		\$TSTRD:	RPI	TRAP TEST RO,R2,R3 #C\$RADD,R0 C\$RNO,RÔ R2 PGMC\$R 1\$ PGMC\$R+1,R3 #BIT7,R3 R3,C\$RNO 2\$		CSR CAREFULLY	
11196 043116	123737	002526	002150	1\$:	CMPB	PGMCSR, CSRNO			
11197 043124 11198 043126 11199 043132	001002 012702 022737	020000 000001	003752	2\$: 3\$:	BNE MOV CMP BEQ	#BIT13,R2 #1_PROTYP	:IS THIS	S AN 11/44? IF IT IS	
11201 043142	004737	043230			CALL	TSTRD1	, Oranderi		
11203 043150 11204 043156		177640		45:	BMOV CALL : IF SING	TSTRD1 FASTCITY SLE BIT ERROR	ONLY - SET	CALL TO THE	USER INSTRUCTION PAR'S
11206 043162				5\$:	POP	R3,R2,R0	ND #RITIS	OFF. IN CSR	
11208 043210	052766	000001	000002		BIS	#BITO,2(SP)			
11206 043162 11207 043170 11208 043210 11209 043216 11210 043220 11211 043226 11212 043226	042766	000001	000002		BIC	#BITO,2(SP)			
11211 043226 11212 043226	000002				RTI ;UF	IF #BIT4			
11213 11214 043230 11215 043232	010210			TSTRD1:	MOV TESTAREA	R2,(R0)		:V177640 :V177642 ;ENT	ER SUPERVISOR MODE
11214 043230 11215 043232 11216 043240 11217 043242 11218 043250 11219 043254	105711 042737 011037 000207	140000 002146	177776		TSTB	(R1) #BIT15!BIT14, (R0),CSR	PSW	:V177640 :V177642 ;ENT :V177646 :V177650 :V177656 :V177662	

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 346 TRAP ECC DISABLE ALL CSR'S TRAP HANDLER

11222 11223 043256 01 11224 043264 00 11225 043270 00	2737 000002 4737 044002 0002	002146	ECCDIS:MOV WBIT1,CSR CALL CSROUT	ALL CSR'S TRAP HANDLER
11226 11227 043272 01 11228 043300 10		002146	ECCIDIS:MOV WBIT1,CSR LOADCSR	OF 1 SELECTED CSR TRAP HANDLER
11230 11231 043304 01	2737 000001 4737 044002 00002	002146	ECCINIT: MOV #BITO.CSR CALL CSROUT RTI	ALL CSR'S TRAP HANDLER
11234 11235 043320 01 11236 043326 10	2737 000001 4425 0002	002146	ECC1INIT:MOV #BITO,CSR LOADCSR RTI	1 SELECTED CSR TRAP HANDLER
11238 11239 043332 01 11240 043340 00 11241 043344 00	2737 000003 4737 044002 00002	002146	ENASBE:MOV #BITO!BIT1,CSR CALL CSROUT RTI	PARITY TRAPS ON ALL CSR'S
11242 11243 043346 01 11244 043354 10	2737 000003 04425 00002	002146	ENAISBE:MOV #BITO:BITI,CSR LOADCSR	PARITY TRAPS ON 1 SELECTED CSR
11247 043360 01 11248 043366 05 11249 043374 00	13737 002310 52737 000006 04737 044002	002146 002146	CALL CSROUT	BITS THRU ALL CSR'S TRAP HANDLER ;BITS 11-5 ;CHECK MODE
11251 11252 043402 01 11253 043410 05	13737 002310 52737 000006 04425 00002	002146 002146	SBITL TRAP WRITE CHECK CB1CSR:MOV CHECK,CSR BIS #BIT1!BIT2,CSR LOADCSR RTI	BITS THRU 1 SELECTED CSR TRAP HANDLER ;BITS 11-5 ;CHECK MODE

```
SWASSBE : PUSH R1 .R4
                                                                                                      WAS THERE A SBE ON ANY CSR TRAP HANDLER
                                                                                         R1,R4
                                                                                          TOTCSRS,R1
                       013701
                                                                                                                    :GET CSR'S BYTE
                                                                            MOV
                                    002222
                                                                            BEGIN LWSBE
                                                                               FOR CSRNO := #0 TO #36 BY #2
                                                                                   ASL
                                                                                                       R1
                       006301
                                                                                   ON. ERROR
                                                                                      READCSR
                       104426
11267 043446

11268 043456

11269 043462

11270 043464

11271 043464

11272 043464

11273 043470

11274 043506

11275 043506

11276 043510

11277 043514

11278 043516

11279 043524

11280 043526

11281 043534

11282 043534

11283

11284

11285 043536

11287 043546

11288 043554

11289 043556

11290 043564
                                                                                       IF #BIT4 SET.IN CSR
                                                                           LEAVE LWSBE
END ; OF IF #BIT4
END ; OF ON.ERROR
IF R1 EQ #0 THEN LEAVE LWSBE
END ; OF FOR CSRNO
END LWSBE
                                                                                                                    :SET C BIT FOR ERROR
                       006004
                                                                            ON. ERROR
                                                                                BIS #BITO,2(SP)
                       052766
                                    000001
                                                  000002
                                                                                BIC #BITO,2(SP)
                       042766
                                    000001
                                                  000002
                                                                            END OF ON ERROR
                        000002
                                                                                                       WAS THERE A SBE IN 1 SELECTED CSR TRAP HANDLER
                                                                             .SBTTL TRAP
                                                                              ON RETURN IF CARRY IS SET THERE WAS A SBE
                       104426
042766
032737
001403
052766
                                                               SWAS1SBE: READCSR
                                                  000002
                                                                                          #BITO.2(SP)
                                     000001
000020
                                                                            BIC
                                                                                                                    :CLR C BIT ON STACK
                                                                                          MBIT4, CSR
                                                                            BEQ
                                                                            BIS
RTI
                                                                                          #BITO,2(SP)
                                     000001
                                                  000002
                                                                                                                    :SET C BIT ON STACK
                                                               15:
```

```
SWASDBE : PUSH R1,R4
                                                                                                           WAS THERE A DBE ON ANY CSR TRAP HANDLER
11293
11294 043566
11295 043572
11296 043576
11297 043600
11298 043600
11299 043604
11300 043610
11302 043612
11303 043622
11304 043626
11305 043630
11306 043630
11307 043630
11308 043634
11309 043652
11310 043652
11311 043654
11312 043660
11313 043662
11314 043670
11315 043672
11316 043700
11317 043700
                                                                                              TOTCSRS,R1
                                                                                                                         :GET CSR'S BYTE
                        013701
                                      002222
                                                                                MOV
                                                                                BEGIN LWDBE
                                                                                   FOR CSRNO := #0 TO #36 BY #2
                         006301
                                                                                       ON. ERROR
                                                                                          READCSR
                         104426
                                                                                          IF #BIT15 SET.IN CSR
                                                                                SET R4

LEAVE LWDBE

END : OF IF #BIT4

END : OF ON. ERROR

IF R1 EQ #0 THEN LEAVE LWDBE

END : OF FOR CSRNO

END LWDBE
                                                                                ROR
                                                                                                                         :SET C BIT FOR ERROR
                                                                                             R4
R4,R1
                         006004
                                                                                POP
                                                                                ON. ERROR
                                                                                   BIS #BITO,2(SP)
                         052766
                                     000001
                                                    000002
                                                    000002
                                                                                   BIC #BITO,2(SP)
                         042766
                                      000001
                                                                                END OF ON ERROR
                         000002
                                                                                .SBTTL TRAP
                                                                                                           WAS THERE A DBE ON 1 SELECTED CSR TRAP HANDLER
                                                                                 ON RETURN IF CARRY IS SET THERE WAS A DBE
          043702
043704
043710
043712
043720
043722
043730
                         104426
005737
100004
052766
000002
042766
000002
                                                                  $WAS1DBE: READCSR
                                                                                                                          :DBE?
                                       002146
                                                                                 TST
                                                                                                                          :NO - SKIP
                                                                                BPL
                                                                                                                          SET C BIT ON STACK
                                                                                              #BITO,2(SP)
                                       000001
                                                     000002
                                                                                              #BITO,2(SP)
                                                                                                                          CLR C BIT ON STACK
                                       000001
                                                     000002 3$:
```

11329 11330 11331	043732 043736	004737 000002	044002		SCLRCSR: CLEAR CALL	TRAP CSR CSROUT	CLEAR ALL ECC CSR'S TRAP HANDLER
11332	043742	000002			SBITL	TRAP	CLEAR 1 SELECTED CSR TRAP HANDLER
11334	043744 043750 043752	104425			\$CLR1CSR:CLEAR LOADCSR	CSR	
11337	043136	000002			.SBTTL	TRAP	ECC DISABLE, CHECK MODE, & WRITE CHECKBITS IN ALL CSR'S TRAP HANDLER
11338 11339 11340	043754 043762 043766	052737 004737 000002	000006 044002	002146	CALL	COKOUT	ECC DISABLE, CHECK MODE, & WRITE CHECKBITS IN ALL CSR'S TRAP HANDLER ADY IN LOC "CSR"; ECC DISABLE & DIAG CHECK MODE
11342	043700	000002			SBTTL	TRAP	ECC DISABLE, CHECK MODE, & WRITE CHECKBITS IN 1 SELECTED CSR
11343 11344 11345 11346	043770 043776 044000	052737 104425 000002	000006	002146	\$CHK1DIS:BIS LOADCSF RTI	BITS ALRE #BIT1!8	ECC DISABLE, CHECK MODE, & WRITE CHECKBITS IN 1 SELECTED CSR EADY IN LOC "CSR" ECC DISABLE & DIAG CHECK MODE 112,CSR ;ECC DISABLE & DIAG CHECK MODE

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 354 TRAP ECC DISABLE, CHECK MODE, & WRITE CHECKBITS IN 1 SELECTED
                                        CSROUT: SUBTST <<SUBR WRITE IN ALL CSR'S>>
  11349 044002
                                        **************
                                                        SUBR WRITE IN ALL CSR'S
                                        : *SUBTEST
                                        *************************
                                                PUSH
                                                        TOTCSRS,R1
                                                MOV
                                                                        GET CSR'S BYTE
                013701 002222
                                                BEGIN LCSROUT
                                                 FOR CSRNO := #0 TO #36 BY #2
                                                                R1
                                                    ASL
                006301
                                                    ON. ERROR
                                                        LOADCSR
                104425
                                              END : OF FOR CSRNO
END LCSROUT
POP
                                                  END : OF ON. ERROR
IF R1 EQ #0 THEN LEAVE LCSROUT
  11360 044044
11361 044044
  11362 044046
11363
11364 044050
                                                RETURN
       044046
                000207
                                                        SUBTST <<TRAP INVALIDATE BACKGROUND PATTERN>>
                                        SINVALID:
                                        *************
                                                                INVALIDATE BACKGROUND PATTERN
                                        : *SUBTEST
                                        ***********************
  11365 044050
11366 044054
11367 044060
11368 044062
11369 044064
11370 044072
11371 044076
                                                        RO,R1
BANK,R1
                013701 002100
006301
006301
042761 020000
                                                MOV
                                                        R1
                                                ASL
                                                ASL
                                                        #BIT13, CONFIG+2(R1)
                        020000 002652
                                                BIC
                                                        R1,R0
                000002
```

THANKTON	TIE DACK	ONCOME 17					
044100				*****	*****	*****	NERATE AND TEST ERROR ADDRESS>> **********************************
044100 044110 044114 044120 044122	013703 005737 001003 013700	002102 002452 172246			PUSH MOV TST BNE MOV	RO,R1,R2,R3 BANKINDEX,R3 NOSUPER 6\$ SIPAR3,R0	GENERATE WHAT ERROR ADDR SHOULD BE
044126 044130 044134 044140 044144	000402 013700 072027 005737 001002	177646 177773 002130		6\$: 7\$:	MOV ASH TST BNE	UIPAR3,RO #-5,RO EUFLAG 1\$	
044146 044152 044154 044156 044160	042700 000301 006201 006201 006201			1\$:	SWAB ASR ASR ASR	R1 R1 R1 R1	GET CURRENT ADDRESS BITS 11 AND 12
001770	000100				ADD ;GET ER	R1,R0 ROR ADDRESS FROM CSF	ADD THEM TO THE ADJUSTED PAR VALUE
044174 044200 044204	072127 042701 005737	177773 177600 002450			ASH BIC TST	NO22811	:IS THIS AN 11/44 OR 11/24?
044210 044212 044216 044220		002130			TST BEQ PUSH	EUFLAG 2\$:IS IT EUB? :BRANCH IF NOT :SAVE GENERATED ERROR ADDRESS
044222 044226 044234 044240 044246	013702 052762 016200 042762 042700	002150 040000 172100 040000 177037	172100 172100		MOV BIC	WBIT14, CSRADD (R2) CSRADD (R2), R0 WBIT14, CSRADD (R2) W^C740, R0	:IS THIS AN 11/44 OR 11/24? :BRANCH IF NOT NECESSARY :IS IT EUB? :BRANCH IF NOT :SAVE GENERATED ERROR ADDRESS :GET CSR NUMBER :TURN ON EUB BIT CAREFULLY :GET CSR CONTENTS :TURN OFF EUB BIT CAREFULLY :CLEAR EVERYTHING BUT ERROR ADDR
044252 044254 044256	006300 006300 060001				ASL ASL ADD	RO RO RO, R1	SHIFT ADDR BITS 18-21 INTO POSITION ADD TO CURRENT ERROR ADDRESS
044260 044262 044264 044266	020001 001420 005737	002134		2\$:	CMP BEQ TST	RO,R1 5\$ INTFLAG	COMPARE REAL AND GENERATED ERR. ADDR. BRANCH IF THEY ARE THE SAME INTERLEAVED? NO - WE HAVE AN ERROR
044274 044300 044304	062700 005737 001002	000100 002136			ADD TST BNE	#100,R0 INT64K 4\$:64K INTERLEAVED MEMORY?
044306 044312 044314	062700 020001 001404	000100		48:	CMP	RO,R1 5\$	ADE HE SHIPPOSED TO SKIP EDDOD D O 2
044316 044322 044324 044326	001001 104462 010137	002064		5\$:	BNE PERR36 MOV	5\$ R1,ERRADD	:ARE WE SUPPOSED TO SKIP ERROR P.O.? :YES - SKIP ERROR PRINTOUT :ELSE PRINT ERROR ADDRESS ERROR :SAVE CSR'S ERROR ADDRESS :ENABLE THE ERROR PRINTOUT AGAIN :RESTORE REGISTERS
044332 044336 044346	005037	002064			CLR POP RTI	SKPERR R3,R2,R1,R0	RESTORE REGISTERS
	044100 044110 044114 044120 044126 044126 044126 044134 044146 044146 044152 044156 044156 044160 044160 044160 044174 044204 044210 044210 044210 044210 044210 044220 044220	044100 044110 044110 044120 044120 044120 044126 0044126 0044130 013700 044134 072027 044140 005737 044144 001002 044152 004301 044154 006201 044156 006201 044160 044160 044160 044160 044160 044160 044160 044170 044160 044170 0441	044100 044110 013703 002102 044114 005737 002452 044120 001003 044126 000402 044130 013700 177646 044134 072027 177773 044140 005737 002130 044144 001002 044146 042700 177600 044152 000301 044152 000301 044154 006201 044156 006201 044160 006201 044160 006201 044160 006201 044160 006201 044174 072127 177773 044200 042701 177600 044200 042701 177600 044210 001024 044210 001024	044100 044110 013703 002102 044114 005737 002452 044120 001003 044122 013700 172246 044130 013700 177646 044134 072027 177773 044140 005737 002130 044144 001002 044146 042700 177600 044152 000301 044154 006201 044156 006201 044160 006201 044160 006201 044160 006201 044160 006201 044160 006201 044174 072127 177773 044200 042701 177600 044204 005737 002450 044210 001024 044212 005737 002450 044216 001421 044220 044220 013702 002150 044220 013702 002150 044220 044220 013702 002150 044224 016200 172100 044234 016200 172100 044234 016200 172100 044234 016200 172100	044100 044110 044110 044114 005737 002452 044120 001003 044120 001003 044126 0044126 0044126 0044130 013700 0177646 044134 072027 077773 044140 005737 002130 044144 001002 044146 042700 044152 000301 044152 000301 044154 006201 044160 006201 044170 013701 002146 004160 006201 044170 013701 002146 004170 013701 002146 004170 013702 002150 0044200 044220 013702 002150 044226 044226 013702 002150 044226 044226 013702 002150 044226 044226 04262 04262 040000 172100 044240 044240 042762 040000 172100	\$ERRGEN: ;********** ;*SUBTEST ;********* ******** 044100 044110 013703 002102 044114 005737 002452 TST 044120 001003 044122 013700 0172246 0044126 0044126 0044130 013703 0172246 013700 0177646 05: MOV 044134 072027 075: ASH 044140 005737 002130 044144 001002 044144 001002 044146 042700 044152 00301 044152 00301 044152 00301 044160 044160 044160 044160 044160 044160 044174 072127 177773 044160 044160 044174 072127 177773 044160 044160 044174 072127 177773 044160 044160 044174 072127 177773 044160 044210 001024 044160 044210 001024 044220 044220 044220 044220 044224 013702 002150 044220 044224 016200 172100 046224 016200 172100 046224 016200 172100 046224 0172100 046224 0172100 046234 016200 172100 046240 0462762 040000 172100 046240 0462762 040000 172100 046240 0462762 040000 172100 046240 0462762 040000 172100 046240 0462762 040000 172100 046240 0462762 040000 172100 046280 046281 04	SERRGEN: SUBTST <trap gen="" td="" ="" <=""></trap>

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 355-1 TRAP GENERATE AND TEST ERROR ADDRESS

```
GENERATE CHECK BITS>>
                                                                  CHKGEN: SUBTST<<SUBR
11434 044366
                                                                   ***********
                                                                   : *SUBTEST
                                                                                                            GENERATE CHECK BITS
                                                                   *********
                                                                                 : CHECK BIT GENERATOR ROUTINE
11435
11436
11437
11438
11439
11440
11441
11442 044366
11443 044406
11445 044412
11446 044416
11447 044420
11448
11449 044422
11450 044424
11451 044426
11452 044430
11453 044434
11454 044436
11454 044436
11456 044446
11459 044450
11460 044452
11461 044456
11462 044472
                                                                                 : CALLING SEQUENCE IS:
                                                                                                                                        :SOURCE = ADDRESS OF DATA
                                                                                              MOV
                                                                                                            #WORD1, SOURCE
                                                                                               CALL
                                                                                                            CHKGEN
                                                                                 CHECK BITS RETURNED IN BITS 11-5 OF LOCATION CHECK
                                                                                              RO,R1,R2,R3,R4,R5
#77,R2 ; 1
#CHKTAB,R3 ; A
SOURCE,R5 ; (
(R5)+,R1 ; (
(R5),R0 ; (
                                                                                 PUSH
                        012702
012703
013705
012501
011500
                                                                                                                          DEFAULT CHECKBITS FOR DOUBLE WORD OF ZEROS
                                      000077
044474
002306
                                                                                 MOV
                                                                                                                          ADDRESS OF CHECKBIT TABLE
GET SOURCE ADDRESS
GET LSB'S
GET MSB'S
                                                                                 MOV
                                                                                MOV
                                                                                 MOV
                                                                                                                           EXTEND SIGN OF DOUBLE WORD TO R4
                        006704
142304
074402
073027
001372
                                                                   15:
                                                                                               R4
(R3)+,R4
                                                                                 BICB
                                                                                                                           COMPLEMENT MASKED BITS IN CHECKBITS DOUBLE PRECISION LEFT SHIFT RO., R1 :LOOP TILL ALL BITS ARE CHECKED
                                                                                              R4,R2
#1,R0
1$
                                                                                 XOR
                                       000001
                                                                                 ASHC
                                                                                 BNE
                        042702
000302
006202
006202
006202
010237
                                                                                                                           :POSITION CHECKBITS IN BITS 11-5
                                                                                               #^C177,R2
                                      177600
                                                                                              R2
R2
R2
R2
R2
R2, CHECK
R5, R4, R3, R2, R1, R0
                                                                                 SWAB
                                                                                 ASR
                                                                                 ASR
                                                                                 ASR
                                      002310
                                                                                 MOV
                                                                                 POP
                                                                                 RETURN
                         000207
```

	CZMSPA0	MS11-L/M/P	MEMORY DIAG.	MACRO M1113	26-APR-82 09:41	PAGE 359
1	LZUBK	GENERALE LI	MECK BILD			

CHKTAB: :BYTE #3 .BYTE *C076 .BYTE *C075	:BIT 31 :BIT 30 :BIT 29
BYTE *C174 BYTE *C073 BYTE *C172 BYTE *C171 BYTE *C070	:BIT 31 :BIT 30 :BIT 29 :BIT 28 :BIT 27 :BIT 26 :BIT 25 :BIT 24
BYTE *C067 BYTE *C166 BYTE *C165 BYTE *C064 BYTE *C163 BYTE *C062 BYTE *C061	:BIT 23 :BIT 22 :BIT 21 :BIT 20 :BIT 19 :BIT 18 :BIT 17 :BIT 16
BYTE #1 BYTE *C057 BYTE *C156 BYTE *C155 BYTE *C054 BYTE *C054 BYTE *C052 BYTE *C051	:BIT 15 :BIT 14 :BIT 13 :BIT 12 :BIT 11 :BIT 10 :BIT 9 :BIT 8
BYTE #0 BYTE *C037 BYTE *C136 BYTE *C135 BYTE *C133 BYTE *C133 BYTE *C032 BYTE *C031 BYTE *C130	:BIT 7 :BIT 6 :CIT 5 :BIT 4 :BIT 3 :BIT 2 :BIT 1
	BYTE

00011	OE									
11503 11504 11505 11506 11507 11508 11509	044534				****** *SUBTE THIS SI IN R3 THE 11 PDP-11	****** ST	SUBR MAPPER>> ************ SUBR MAPPER ************ E MAPS THE MEMORY EST PATTERN AREA 11/45-55; USER VI	******	SK WORDS = 1 BANK) SOR VIRTUAL (60000 - 157 0000 - 157777) FOR ALL 0	**************************************
11510					CALL	MOV CALL RETURN	BANKNO,R3 MAPPER		SET UP BANK ARGUEMENT ACTUAL CALL ONLY RETURN	
11512 11513 11514 11515 11516 11517 11518 11520 11521 11522 11523 11524 11525 11526 11527 11528 11529	044534 044546 044552 044556 044562 044570 044574 044600 044604 044612 044614 044614	012700 012701 012704 005737 001404 012701 012704 012702 012705 012021 010224 077503 012741	172340 172240 172200 002452 177640 177600 077406 000010		MAPPER: 4\$: 1\$:	SET SUPUSH MOV MOV TST BEQ MOV MOV MOV MOV MOV MOV SOB MOV	PERVISOR/USER UP RO,R1,R2,R4,R5 #KIPARO,R0 #SIPARO,R1 #SIPDRO,R4 NOSUPER 4\$ #UIPARO,R1 #UIPDRO,R4 #77406,R2 #8.R5 (RO)+,(R1)+ R2,(R4)+ R5,1\$ #177600,-(R1)	FOR 1 TO	FIRST AREA TO MAP TO FIRST ADDRESS REGISTER FIRST DESCRIPTOR REGIST CAN WE USE SUPERVISOR MYES, BRANCH FIRST ADDRESS REGISTER FIRST DESCRIPTOR REGIST CONSTANT FOR 4K PAGE, UCOUNTER PUT IN SUPERVISOR ADDREST OF TILL DONE CORRECT LAST FIELD FOR	ER IODE? IER IP, R/W ISS IPTOR PERIPHERALS PAGE
11530 11531 11532 11533 11534 11535 11536	044622 044626 044630 044634 044640 044644	022703 001516 072327 012701 005737 001402	000170 000011 172246 002452 177646 000004 000200 002236 000010		5\$: 2\$:	SET UP CMP BEQ ASH MOV TST BEQ MOV MOV ADD SOB TST BEQ	SUPERVISOR/USER #120.,R3 3\$ #9.,R3 #SIPAR3,R1 NOSUPER 5\$ #UIPAR3,R1 #4,R2 R3,(R1)+ #200,R3 R2,2\$	FOR TEST	AREA MAP NOTHING (1 TO 1)? YES - SKIP BANK 1 STARTS AT 100,00 FOR MEMORY MANAGEMENT = SETUP FOR AUTO INCREMEN DO WE HAVE SUPERVISOR M YES - BRANCH SETUP FOR AUTO INCREMEN COUNTER PLUG IN PAR INFO BUMP ADDRESS 4K LOOP TILL DONE	O LESS 6 LSB'S 1000 ITING HODE?
11538 11539 11540 11541 11543 11543 11544 11546 11548 11548 11551 11551 11553 11553	044646 044656 044660 044664 044666 044672 044674 044700 044702 044714 044716 044720 044724 044724 044724 044724	012701 012702 010321 062703 077204 005737 001442 162701 010102 062702 022737 001403 010200 010102 010001 012122 011112 013700 005737 001403	000004 000001 002102 002136	002236	10\$:	SUB MOV ADD CMP BEQ MOV MOV MOV MOV MOV TST BEQ	SPLTCSR 9\$ #10,R1 R1,R2 #4,R2 #1,SPLTCSR 10\$ R2,R0 R1,R2 R0,R1 (R1)+,(R2)+ (R1),(R2) BANKINDEX,R0 INT64K 11\$			

CZMSPAO MS11-L/M/F SUBR MAPPER	P MEMORY DIAG.	MACRO M1113	26-APR-82	09:41 PAGE 361	-1
11557 044742 01	12700 004000		MOV	#4000,R0	
11557 044742 01 11558 044746 00 11559 044750 01 11560 044754 00	00402	118.	BR MOV	12\$ #10000,R0	
11559 044750 01 11560 044754 00 11561 044760 00	12700 010000 05737 002452	11\$: 12\$:	TST	NOSUPER	
11561 044760 00	01403		BEQ	13\$ #UIPAR5,R1	
11561 044760 00 11562 044762 01 11563 044766 00 11564 044770 01	00402		BR	148	
11564 044770 01	12701 172252 60021	13\$: 14\$:	MOV	#SIPAR5,R1 R0,(R1)+	
11565 044774 06 11566 044776 06 11567 11568 11569 11570	60011	140.	ADD	RO.(R1)	AN EVETEM HE BONET HANT TO TEST THE
11567 11568			: IF WE	4K, WHERE THE UN	4K SYSTEM, WE DON'T WANT TO TEST THE IBUS DEVICE PAGE IS. INSTEAD, THE
11569			PROGR	AM WILL REMAP TH	IBUS DEVICE PAGE IS. INSTEAD, THE E LAST 4K TO 8-12K. ALSO, IF THERE 1/44, THE PROGRAM WILL REMAP THE LAST AME REASON.
11570			4K 10	8-12K FOR THE S	AME REASON.
11572 045000 07	22737 000007	002552 9\$:	BNE	#7,LASTBANK	
11574 045010 00	05737 002450		TST	NO22BIT	:11/44 OR 24?
11575 045014 00 11576 045016 00	01423 22737 000007	002100	BEQ CMP	#7,BANK	BRANCH IF SO BANK 7?
11577 045024 0	01017	002.00	BNE	3\$ 8\$:NO - BRANCH
11573 045000 06 11573 045006 06 11574 045010 06 11575 045014 06 11576 045016 06 11577 045024 06 11578 045026 06 11579 045030 06 11580 045036 06 11581 045040 06 11582 045044 06 11583 045046 06	22737 000007 01010 05737 002450 01423 22737 000007 01017 00404 22737 000177 01012 05737 002452 01404 13737 177652	002552 7\$:	BR CMP	#177, LASTBANK	
11580 045036 0	01012	8\$:	BNE	3\$ NOSUPER	
11582 045044 0	05737 002452 01404		BEQ	6\$	
11583 045046 01 11584 045054 00	13737 177652	177654	MOV BR	UIPAR5, UIPAR6	
11585 045056 0	00403 13737 172252	172254 6\$:	MOV	SIPARS, SIPAR6	
11586 045064 11587 045076 0	00207	3\$:	POP	R5,R4,R2,R1,R0	
11588	00201	Avm	.SBTTL	TRAP MAP KE	RNEL (ALMOST 1 TO 1) TRAP HANDLER
11590 045112 0	05000	\$KMA	P: PUSH CLR	RO,R1,R2,R3,R4	ACT ADEA TO MAD TO
11591 045114 0	12701 172340 12702 077406 12703 172300 12704 000010		MOV	#KIPARO,R1	FIRST ADDRESS
11593 045124 0	12703 172300		MOV	#KIPDRO,R3	:1ST PAGE DESCRIPTOR REGISTER
11594 045130 0	12704 000010	15:	MOV	#8R4 PO (R1)+	COUNTER PUT IN KERNEL ADDRESS
11596 045136 0	10223		MOV	R2,(R3)+	PUT IN KERNEL DISCRIPTOR
11597 045140 0	62700 000200		SOB	#200,R0 R4.1\$:LOOP TILL DONE
11580 045036 00 11581 045040 00 11582 045044 00 11583 045046 00 11584 045054 00 11585 045056 00 11586 045064 11587 045076 00 11588 11589 045112 00 11591 045114 00 11592 045120 00 11593 045124 00 11594 045130 00 11595 045134 00 11596 045136 00 11597 045140 00 11598 045144 00 11598 045146 00 11599 045146 00 11600 045152 00 11607 045156 11608 045170 00	05000 12701 172340 12702 077406 12703 172300 12704 000010 10021 10023 62700 000200 177405 12741 177600 12741 177400		MOV	#KIPARO,R1 #77406,R2 #KIPDRO,R3 #8.,R4 RO,(R1)+ R2.(R3)+ #200,RO R4,1\$ #177600,-(R1) #177400,-(R1) R4,R3,R2,R1,R0	:1ST AREA TO MAP TO :FIRST ADDRESS :CONSTANT FOR 4K PAGE,UP,R/W :1ST PAGE DESCRIPTOR REGISTER :COUNTER :PUT IN KERNEL ADDRESS :PUT IN KERNEL DISCRIPTOR :ADD ADDRESS CONSTANT FOR 4K CHANGE :LOOP TILL DONE :THE PERIPHERALS PAGE TO KIPAR7 :AND NEXT LOWER PAGE TO KIPAR6
11607 045156			MOV	R4,R3,R2,R1,R0)
11608 045170 0	00002		RTI		

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 363
              MAP KERNEL (ALMOST 1 TO 1) TRAP HANDLER
                                                                                RELOCATE:SUBTST <<RELOCATE PROGRAM>>
   11611 045172
                                                                                **************
                                                                                                                 RELOCATE PROGRAM
                                                                                 : *SUBTEST
                                                                                ****************
                                                                                               IF #SW12 SET.IN aswr THEN $RETURN ERROR
IF APTFLAG IS TRUE OR ACTFLAG IS TRUE
IF $PASS NE #0 THEN $RETURN ERROR
END; OF IF APTFLAG
BEGIN LOADERBANK
FOR BANK := #1 TO LASTBANK
CALL EXBANK
IF ACFLAG IS TRUE AND PFLAG IS FALSE AND BMFLAG IS FALSE
   11612
11613
11614
11615
              045172
045234
045234
045234
045234
045234
045236
045300
045310
045310
045314
045336
045336
045336
045336
045336
045336
045372
045372
045372
045372
045372
045406
045406
045436
   11616
11617
11618
11619
                                004737
                                               047020
                               013700
010037
013701
004737
004737
013701
                                                                                                            MOV
                                                                                                                                 BANK, RO
   11620
11621
11622
11623
11624
11625
11626
11627
11628
11630
11631
11633
11633
11635
11636
11637
11640
11641
11642
11643
11646
11647
11648
11649
11650
11651
                                                                                                                                 RO, LOADBANK
                                                                                                             MOV
                                                                                                                                 LOADHOME, R1
                                                                                                             MOV
                                                                                                                                 BANKMOV
                                                                                                             CALL
                                                                                                                                                                 :MAP NEW LOADER BANK IN KERNEL
                                                                                                             CALL
                                                                                                                                 NEWLOAD
                                                                                                             MOV
                                                                                                                                 BANKINDEX, R1
                                                                                                                                 #BIT15, CONFIG+2(R1)
#BIT13, CONFIG+2(R1)
                                                                                                                                                                                  :MARK LOADER
                                                                                                             BIS
                                                                                                                                                                                  :INVALIDATE BACKGROUND PATTERN
                                                                                                    LEAVE LOADERBANK
END : OF IF ACFLAG
END : OF FOR BANK
IF #SW13 OFF. IN aswr
                                                                                                         TYPE
                                                                                                                                                                 :RELOCATION NOT POSSIBLE
                                                                                                                                 MSG075
                                                                                                     END : OF IF #SW13
                                                                                                 END LOADERBANK
                                                                                                 BEGIN FINDBANK
                                                                                                                 LASTBANK,R2
R2
R2
                                013702
006302
006302
                                                002552
                                                                                                 ASL
                                                                                                    FOR R1 := #2+2 TO R2 BY #4
                                                                                                        IF #BIT7:BIT0 OFF.IN CONFIG(R1) :IF NO ERRORS & NOT PROGRAM SPACE
IF #BIT15 OFF.IN CONFIG+2(R1) :IF NOT LOADER BANK
IF CPUBIT SET.IN CONFIG(R1) :IF ACCESSABLE
IF #BIT8 OFF.IN CONFIG+2(R1) THEN LEAVE FINDBANK :IF PARITY
IF #BIT6 SET.IN CONFIG(R1) AND #BIT7 OFF.IN CONFIG(R1)
:IF 1ST PROTECTABLE ECC BANK
                                                                                                                         LEAVE FINDBANK
END : OF IF #BIT6
IF INHECC IS FALSE
               045466
045470
045476
045504
045510
045510
045510
045510
045520
045526
045534
045546
045552
045552
                                                                                                                             SET
                                                                                                                                         INHECC
                                                                                                    MOV R1, INHBANK
END; OF IF INHECC
END; OF IF CPUBIT
END; OF IF #BIT15
END; OF IF #BIT7
END; OF FOR
IF FULLREL IS FALSE
IF INHECC IS TRUE
                                010137 002534
    11656
11657
    11658
11659
11660
11661
11662
11663
11664
                                                 002534
002274
                                                                                                                            INHBANK, R1
                                                                                                             MOV
                                                                                                                                                                  : IS THIS PATTERN 30?
                                                                000030
                                                                                                                            REALPAT,#30
                                                                                                                                                                  :YES - SKIP MESSAGE
                                                                                                              BEQ
                                                                                                                            RELENT1
                                                                                                             BR
                                                                                                                            RELENT1
                                                                                                     END: OF IF INHECC
END: OF IF FULLREL
```

	CZMSPAO MS11-L/M/P	MEMORY	DIAG.	MACRO	M1113	26-APR-82	09:41	PAGE	363-1	
1	IRFLOCATE PROGRAM									

RELOCATE PRO	GRAM							
11665 0455	52 00503;	002532			CLR	INHECC TH DOUB	; MAKE SURE FLAG IS TURNED OFF!	
11665 0455 11666 0455 11667 0455 11668 0455 11669 0455 11670 0456 11671 0456 11672 0456 11673 0456 11674 0456 11677 0456 11677 0456 11681 0456 11682 0456 11683 0456 11684 0456 11687 0457 11689 0457 11690 0457 11691 0457 11691 0457 11692 0457 11693 0457 11694 0457 11696 0457 11697 0457 11697 0457 11698 0457 11698 0457 11699 0457 11699 0457 11699 0457 11699 0457	56 66 023727 74 001402 76 02	002274	000030		TVP	F MSG075	:IS THIS PATTERN 30? :YES - SKIP MESSAGE :RELOCATION NOT POSSIBLE	
11671 0456 11672 0456 11673 0456	02 06 06 12 042761	020000	002652	SKUB:	SRETUI END FINI CLEAR : BIC	OF IF #SW13 RN ERROR DBANK INHECC #BIT13.CONFIG+2(R1)	; IF WE RELOCATED PROPERLY, THIS SHOULD BE OFF	F!
11675 0456	042761 20 005000 22 071027	000004	002072		CLR	R0		
11677 0456 11678 0456 11679 0456 11680 0456	26 32 013737 40 004737		002530	RELOC1:	MOV CALL	BANK := RO PGMCSR,PGMCSR+2 USERMAP	SAVE CURRENT PGM. CSR MAP NEWBANK TO USER PAR ENTER USER MODE	
11681 0456 11682 0456 11683 0456 11684 0456	552 664 104417 666 022737 674 001021 676 042737 604 013700 610 006200	000001	003752		USER BMOV KERNEL CMP BNE BJC	0,100000,SIZE #1,PROTYP JMPRL1 #BIT5,MMR3 NEWBANK,R0	SAVE CURRENT PGM. CSR MAP NEWBANK TO USER PAR ENTER USER MODE MOVE PROGRAM ENTER KERNEL MODE IS THIS AN 11/44 ? JUMP IF NOT TURN OFF UNIBUS MAP	
11685 0456 11686 0457 11687 0457 11688 0457	04 013700 10 006200		172516 170200		MOV ASR ON.ERRO MOV	RU	FIGURE OFF ONIBOS FOR	
11690 0457 11691 0457 11692 0457	714 012737 722 010037 726 004737				END ; OF MOV CALL	ON.ERRÓR RO,MAPHO LOWMAP WBITS,MMR3 WBITO,MMRO NEWKERNEL	SETUP LOWER 16K IN UNIBUS MAP ENERGIZE UNIBUS MAP DEENERGIZE MEMORY MANAGEMENT	
11695 0457 11695 0457 11696 0457	732 032737 740 042737 746 004737 752 013700	000001 046640 002304	172516 177572	JMPRL1:	MOV	NE WBANK, KU	DEENERGIZE MEMORY MANAGEMENT	
11697 0457 11698 0457 11699 0457 11700 0457	756 006300 760 006300 762 016002 766 000302	002650			ASL ASL MOV SWAB	RO RO CONFIG(RO),R2 R2	;R0 <- R0 * 4	
11701 0457	70 042702 74 006302	177760			BIC ASL	R2 #^C17,R2 R2		
11703 0457 11704 0460 11705 0460	776 052737 004 010237 010 032760	000001 002526 010000			BIS MOV BIT	#BITO,MMRO R2,PGMCSR #BIT12,CONFIG+2(R0)	;ENERGIZE MEMORY MANAGEMENT ;PUT NEW PGM. CSR INTO PGMCSR ;IS THE NEW BANK INTERLEAVED? ;BRANCH IF NOT INTERLEAVED	
11700 0456 11701 0457 11702 0457 11703 0457 11704 0466 11705 0466 11707 0466 11709 0466 11710 0466 11711 0466 11711 0466 11712 0466	722 010037 726 004737 732 052737 740 042737 746 004737 752 013700 756 006300 760 006300 760 006302 770 042702 774 006302 776 052737 776 052737 776 010237 776 010237 776 042702 776 042702 776 052737	002650 007777 177775 100000 002526			MOV BIC ASH BIS	CONFIG(RO),R2 #^C170000,R2 #-3,R2 #BIT15,R2 R2,PGMCSR RLFLAG NOERROR		
11711 0466 11712 0466 11713 0466	040 050237 044 052	002526		1\$:	BIS SET SRETURN	R2,PGMCSR RLFLAG NOERROR		

RELOCATE PROGRAM		
11716 046056	UNRELOCATE:SUBTST < <unrelocate :************************************<="" td=""><td>PROGRAM>></td></unrelocate>	PROGRAM>>
11717 11718 046056 11719 046060 013701 002426 11720 046064 013700 002562 11721 046070 004737 046410 11722 046074 004737 046742 11723 046100 11724 046104 013737 002426 0 11725 046112 004737 047020 11726 046116 013701 002102 11727 046122 042761 100000 0 11728 046130 013737 002562 0 11729 046136 004737 047020 11730 046142 013701 002102	; RESTORE LOADERS PUSH RO MOV LOADBANK, R1 MOV LOADHOME, RO CALL BANKMOV CALL NEWLOAD PUSH BANK DO2100 MOV LOADBANK, BANK CALL EXBANK	;MAP NEW LOADER BANK IN KERNEL SPACE
11724 046104 013737 002426 0 11725 046112 004737 047020 11726 046116 013701 002102 11727 046122 042761 100000 0 11728 046130 013737 002562 0 11729 046136 004737 047020 11730 046142 013701 002102 11731 046146 042761 020000 0	002652 BIC #BIT15,CONFIG+2(R1) 002100 MOV LOADHOME,BANK CALL EXBANK MOV BANKINDEX,R1 BIC #BIT13,CONFIG+2(R1) POP BANK	:INVALIDATE BACKGROUND PATTERN
11733 046160 11734 11735	CLEAR INHECC RESTORE BANK 0 002652 BIC #BIT13, CONFIG+2 LET NEWBANK := #0	; MAKE SURE ECC TESTS ARE NOT INHIBITED! ; INVALIDATE BACKGROUND PATTERN
11738 046176 004737 046556 11739 046202 11740 046210 11741 046222 104417 11742 046224 042737 000001 1	CALL USERMAP USER BMOV 0,100000,SIZE KERNEL BIC #BITO,MMRO CALL NEWKERNEL	MAP NEWBANK TO USER PAR ENTER USER MODE MOVE PROGRAM ENTER KERNEL MODE DEENERGIZE MEMORY MANAGEMENT
11741 046222 104417 11742 046224 042737 000001 1 11743 046232 004737 046640 11744 046236 013737 002530 (1) 11745 046244 052737 000001 1 11746 046252 005037 002124 11747 046256 022737 000001 (1) 11748 046264 001014 11749 046266 042737 000040 1	002526 MOV PGMCSR+2,PGMCSR 177572 BIS #BITO,MMRO CLR RLFLAG 003752 CMP #1,PROTYP BNE 1\$	RESTORE PREVIOUS PGM. CSR ;ENERGIZE MEMORY MANAGEMENT ;IS THIS AN 11/44 ?
11751 046304 004737 046344	CLEAR MAPLO, MAPHO	; TURN OFF UNIBUS MAP ; SETUP LOWER 16K OF UNIBUS MAP ; ENERGIZE UNIBUS MAP E 2ND WORD OF CONFIG TO RO AR BACKGROUND VALID BIT REMENT TO NEXT BANK
11755 046326 062700 000004 11756 046332 020027 003620 11757 046336 003771 11758 046340 11759 046342 000207 11760	BLE 2\$;NO =	BRANCH
	LOWMAP: SUBTST < <setup 16k="" lower="" maplo,ro<="" mov="" of="" push="" ro,r1,r2="" setup="" subtest="" td="" un=""><td>*********</td></setup>	*********
11762 046344 11763 046352 012700 170200 11764 046356 012701 170204 11765 046362 012702 000003 11766 046366 012011	MOV #MAPLO,RO MOV #MAPL1,R1 MOV #3,R2 1\$: MOV (RO)+,(R1)	

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 365-1 SETUP LOWER 16K OF UNIBUS MAP

062721 020000 012021 077205

000207

#BIT13,(R1)+ (R0)+,(R1)+ R2,1\$ R2,R1,R0 ADD MOV SOB POP RETURN

11774 046410			BANKMOV: SUBTST	MOVE BANKS	***********	*******
11775 11776 11777 11778	10//15		:MOVE 3/ :CALLING :R0 = DE :R1 = SC	/4 OF A BANK G SEQUENCE ESTINATION BANK DURCE BANK		
11779 046410 11780 046412 11781 046416	004737	046556	CALL RESREG SAVREG	USERMAP		
11779 046410 11780 046412 11781 046416 11782 046420 11783 046422 11784 046426 11785 046432 11786 046436	104415 004737 104416 104415 072027 072127 012702 012703	000011 000011 177650 000200	SAVREG ASH ASH MOV MOV	#9R0 #9R1 #UIPAR4.R2 #200.R3		
11780 046412 11781 046416 11782 046420 11783 046422 11784 046426 11785 046432 11786 046436 11787 11788 046442 11789 046444 11790 046446 11791 046450 11792 11793 046452 11794 046454 11795 046456 11796 046460 11797 11798 046462 11799 046470 11800 046502 11801 11802 046504 11803 11804 046510 11805 046512 11806 046514 11807 046516	010122 060301 010122 960301		MOV ADD MOV ADD	R1,(R2)+ R3,R1 R1,(R2)+ R3,R1	:MAP 1ST HALF B	ANK
11792 11793 046452 11794 046454 11795 046456 11796 046460	010022 060300 010022 060300		MOV ADD MOV ADD	RO,(R2)+ R3,R0 RO,(R2)+ R3,R0		
11798 046462 11799 046470 11800 046502	104417		USER BMOV KERNEL	100000,140000,51	ZE/2 : MOV 1ST HALF E	ANK
11802 046504	012702	177650	MOV	#UIPAR4,R2		
11804 046510 11805 046512 11806 046514 11807 046516	010122 060301 010122 060301		MOV ADD MOV ADD	R1,(R2)+ R3,R1 R1,(R2)+ R3,R1	; MAP 2ND HALF E ; BUMP BY 4K	ANK
11808 11809 046520 11810 046522 11811 046524 11812 046526	010022 060300 010022 060300		MOV ADD MOV ADD	RO,(R2)+ R3,R0 RO,(R2)+ R3,R0		
11813 11814 046530 11815 046536 11816 046550 11817	104417		USER BMOV KERNEL	100000,140000,51	ZE/4 :MOV 3ND FOURTH	OF BANK
11818 046552 11819 046554	104416 000207		RESREG RETURN			

MORE BUILD			
11822 046556	USERM ; **** ; *SUB	TEST SUBR MAP US	ER TO NEW BANK>> ***********************************
11823 046556 012701 1 11824 046562 012702 1 11825 046566 012703 1 11826 046572 012704 1 11827 046576 012705 0 11828 046602 012221 11829 046604 011423 11830 046606 077503 11831 11832 046610 013700 0	177640 172340 177600 172300 000004 1\$:	MOV #UIPARO,R1 MOV #KIPARO,R2 MOV #UIPDRO,R3 MOV #KIPDRO,R4 MOV #4,R5 MOV (R2)+,(R1)+ MOV (R4),(R3)+ SOB R5,1\$	COPY KERNEL PAR'S & PDR'S (0-3)
11832 046610 013700 (11833 046614 072027 (002304 000011	MOV NEWBANK, RO ASH #9.,RO	BANK 1 STARTS AT 100,000 LESS 6 LSB'S
11836 046624 010021 11837 046626 062700 (11838 046632 011423 11839 046634 077505 11840 046636 000207	000004 000200	MOV #4,R5 MOV R0,(R1)+ ADD #200,R0 MOV (R4),(R3)+ SOB R5,2\$ RETURN	SETUP UIPAR(4-7) BUMP ADDRESS 4K SETUP UIPDR(4-7)
11841 11842 046640	NEWKE	RNEL:SUBTST < <subr< td=""><td>KERNEL PAR'S FOR NEW BANK</td></subr<>	KERNEL PAR'S FOR NEW BANK
11843 046640 11844 046646 012700 11845 046652 013701 11846 046656 072127 11847 11848 046662 012705 11849 046666 010120 11850 046670 062701	;**** 172340 002304 000011	PUSH RO.R1.R5 MOV #KIPARO.RO MOV NEWBANK.R1 ASH #9.,R1	BANK 1 STARTS AT 100,000 LESS 6 LSB'S
11852 046676 11853 046704 000207	000004 000200	MOV #4.R5 MOV R1.(R0)+ ADD #200.R1 SOB R5.1\$ POP R5.R1.R0 RETURN	;SETUP KIPAR(0-3)
11854 11855 046706	;****	******	ERNAL PARS 4 AND 5 TO A BANK>> ERNAL PARS 4 AND 5 TO A BANK
11856 11857 046706 013705 11858 046712 072527 11859 046716 013737 11860 046724 010537 11861 046730 062705 11862 046734 010537 11863 046740 000207 11864 11865 046742	002100 000011 172350 002266 172350 000200 172352	MOV BANK,R5 ASH #9.,R5 MOV KIPAR4,SAVPAR MOV R5,KIPAR4 ADD #200,R5 MOV R5,KIPAR5 RETURN	MOV BANK NUMBER TO R5 R5 ENTERS 100000 LESS SHIFT TO CREATE MAPPING SAVE OLD PAR GET NEW PAR'S
11864 11865 046742 11866	;***	**************	KERNEL PAR'S FOR NEW LOADER BANK>> KERNEL PAR'S FOR NEW LOADER BANK KERNEL PAR'S FOR NEW LOADER BANK INATION BANK

									K 8				SEQ 0307
	CZMSPAO MS1 SUBR SET	1-L/I	M/P MEMO ERNEL PA	RY DIAG.	MACRO M	1113 26. ER BANK	-APR-82	09:41 PAG	E 369-1				324 0307
the same and the s	11869 046 11870 046 11871 046 11872 046 11873 046 11874 046	742 746 752 756 760 764 766	012701 072027 010021 062700 010021	172350 000011 000200			PUSH MOV ASH MOV ADD MOV POP RETURN	RO,R1 #KIPAR4,R #9.,R0 RO,(R1)+ #200,R0 RO,(R1)+ R1,R0	1	;BANK 1 ;SETUP K	IPAR4	100000 LESS 6 LSB	'S (1000)
	11875 11876 046	774				UNMAP:	SUBTST ST	******	******	PAR'S 4 AND PAR'S 4 AND	******	*******	****
The second secon	11877 046 11878 047 11879 047 11880 047	774 2002 2010 2016	013737 062737 013737 000207	002266 000200 002266	172350 002266 172352	;*****	MOV ADD MOV RETURN	SAVPAR,KI #200,SAVP SAVPAR,KI	PAR4 AR PAR5	RESTORE ADD 200 RESTORE	KIPAR4 FOR NEXT KIPAR5	PAR	

```
EXBANK: SUBTST <<SUBR EXAMINE BANK>>
11883 047020
                                                                                   EXAMINE BANK
                                                                                    *SUBTEST
                                                                                    DOES THE FOLLOWING:

(1) SETS UP 'BANKINDEX' AND R1 BASED ON VALUE OF 'BANK'.

(2) SETS THE 'MKFLAG' IF THE BANK IS ECC.

(3) SETS THE 'KPFLAG' IF THE BANK IS THE PROTECTED REGION OF ECC MEMORY.

(4) SETS THE 'ACFLAG' IF THE BANK CAN BE ACCESSED BY THIS CPU.

(5) SETS THE 'PFLAG' IF THE BANK IS IN PROGRAM SPACE.

(6) SETS THE 'RRFLAG' IF RELOCATION IS REQUIRED TO TEST THIS BANK; HOWEVER, IT COMPLEMENTS THIS FLAG IF THE RELOCATION FLAG 'RLFLAG' IS SET (THIS IS NECESSARY FOR THE USE OF THE RECURSIVE 'MODE' SUBROUTINES). THE 'RRFLAG' IS AL WAYS SET TO DISABLE TESTING IF FIELD SERVICE MODE 'SELECTED BANKS'
 11884
11885
 11886
 11887
 11888
 11889
 11890
11891
                                                                                    NECESSARY FOR THE USE OF THE RECURSIVE 'MODE' SUBROUTINES). THE 'RRFLAG' IS ALWAYS SET TO DISABLE TESTING IF FIELD SERVICE MODE 'SELECTED BANKS' ARE BEING TESTED AND THIS BANK IS NOT SELECTED.

(7) SETS THE 'BMFLAG' IF THE BANK IS A BAD MEMORY: HOWEVER, IT COMPLEMENTS THIS FLAG IF THE 'WORST' FLAG IS NOT SET (THIS IS NECESSARY FOR THE USE OF THE RECURSIVE 'MODE' SUBROUTINES).

(8) SETS THE 'INTFLAG' IF THE BANK IS INTERLEAVED.

(9) SETS THE 'INT64K' FLAG IF THE BANK IS INTERLEAVED ON 64K WORD BOUNDS.

(10) SETS THE 'SKIPMK' FLAG IF THIS BANK IS INTERLEAVED, AND HAS ALREADY
 11892
 11893
 11894
11895
  11896
  1189
 11898
11899
 11900
                                                                                                BEEN TESTED.
 11901
 11902
                                                                                   (11) SETS THE 'PMEMFLG' IF THE ECC MEMORY UNDER TEST IS A MS11-P
11903

11904 047020

11905 047026

11906 047042

11907 047050

11908 047064

11909 047100

11910 047104

11911 047106

11912 047110

11913 047114

11914 047122

11915 047124

11916 047132

11917 047136

11918 047156

11920 047156

11921 047162

11921 047162

11922 047164

11923 047170

11924 047172

11925 047176

11926 047202

11927 047204

11928 047212

11930 047224

11931 047226

11931 047226

11932 047234

11933 047242

11933 047242
                                                                                                    CLEAR
                                                                                                                     MKFLAG, KPFLAG, PMEMFLAG
                                                                                                                      ACFLAG
                                                                                                    SET
                                                                                                                      PFLAG, RRFLAG, BMFLAG
                                                                                                    CLEAR
                                                                                                                      INTFLAG, INT64K, SKIPMK
                                                                                                    CLEAR
                               013701
                                                                                                    MOV
                                                                                                                      BANK, R1
                                                002100
                               006301
006301
010137
                                                                                                                      R1
                                                                                                    ASL
                                                                                                                                                       ;R1 <- R1 + 4
                                                                                                    ASL
                                                                                                                      R1, BANKINDEX
                                                                                                    MOV
                                                 002102
                                                000100 002650
                                                                                                                      #BIT6, CONFIG(R1)
                                                                                                                                                                         :PROTECTED REGION OF ECC MEMORY?
                                                                                                    BIT
                                                                                                                                                                         :NO - SKIP
                                                                                                                      15
                                                                                                    BEQ
                                                                                                    SET
                                                                                                                      KPFLAG
                                                                                                    MOV #BIT1.RO
IF RO SET.IN CPUBIT AND RO OFF.IN CONFIG(R1)
                                                                                   15:
                               012700
                                                000002
                                                                                                                     ACFLAG
IF RO
                                                                                                        CLR
                               005037
                                                002114
                                                                                                    END : OF
                               005737
001415
016102
                                                                                                                      ACFLAG
2$
                                                                                                                                                                         :ACTIVE MEMORY?
                                                 002114
                                                                                                     TST
                                                                                                                                                                         :BRANCH IF NOT
                                                                                                    BEQ
                                                                                                                      CONFIG+2(R1),R2
                                                                                                     MOV
                                                 002652
                               000302
042702
020227
003405
                                                                                                     SWAB
                                                                                                                                                                         :ISOLATE MEM TYPE BITS
:IS THIS AN ILLEGAL MEM TYPE?
:BRANCH IF NOT
                                                                                                                     #*C7,R2
R2,#3
2$
                                                 177770
                                                 000003
                                                                                                                                                                     SET BAD BANK FLAG
JUMP OVER REST OF FLAG TESTS
                                                                                                                      BMFLAG
                               000137
032761
001412
                                                 047436
000400
                                                                                                                      ENEXBK
                                                                                                                      #BIT8, CONFIG+2(R1)
                                                                  002652 2$:
                                                                                                                                                                          :15 THERE ECC THERE?
                                                                                                                                                                          :NO - SKIP
                                                                                                                                                                          YES - SET MKFLAG
                                                                                                                      MKFLAG
                                                                                                                      #BIT9, CONFIG+2(R1)
                                                 001000 002652
                                                                                                                                                                          :NO SKIP!!!
                                                                                                                                                                          SET MS11-P FLAG
                                                                                                                      PMEMFLG
                                                                                                                      #BIT7, CONFIG(R1)
                                                                                                                                                                         :BANK = PROGRAM SPACE?
                                                 000200 002650 3$:
```

CZMSPAO SUBR	MS11-L/I	M/P MEMO BANK	RY DIAG.	MACRO M	1113 26	-APR-82	09:41	PAGE	371-1	SEQ 03	0
11937 11938 11939 11940	047262 047276 047302 047304	005737 001402 005137 032761 001403	002124 002122 000001	002650	5\$: 6\$:	SET TST BEQ COM BIT	RLFLA 6\$ RRFLA	G	AG IG(R1)	:IS PROGRAM RELOCATED? :NO - SKIP :YES - COMPLEMENT RELOCATION REQUIRED FLAG :ERRORS PRESENT IN THIS BANK?	
11942 11943 11944 11945	047316 047320 047326 047332	001403 005737 001002 005137	002564	002030	8\$:	BIT BEQ SET TST BNE COM	8\$ BMFLA WORST	G		;NO - SKIP :IS THIS A WORST FIRST PASS? :YES - SKIP	
11947 11948 11949 11950	047340 047356 047364 047364	032761	010000	002652	9\$:	SET END ; OI BIT	RRFLA IF SE	LONLY 2.CON	AND #BIT14 FIG+2(R1)	:NO - COMPLEMENT BAD MEMORY FLAG 4 OFF.IN CONFIG+2(R1) :IS THIS BANK INTERLEAVED? :BRANCH IF IT IS NOT	
11951 11952 11953 11954 11955	047374 047402 047410 047412	032761 001403	004000	002652		BEQ SET BEQ SET BIT BEQ SET	10\$	1,CON	FIG+2(R1)	:IS THIS BANK INTERLEAVED WITH 64K BOARDS?	
11956 11957 11958 11959 11960	047420 047426 047430 047436 047444	032761 001403 000207	000040	002650	10\$: ENEXBK:	BIT BEQ SET POP RETURN	WBITS ENEXE SKIPM R2,R1	SK SK	IG(R1)	:SHOULD THIS BANK BE TESTED? :BRANCH IF IT SHOULD :RESTORE REGISTERS	

1

```
BOOT: SUBTST <<BOOTSTRAP ROUTINE>>
11993 047524
                                                                                               : *SUBTEST
                                                                                                                                     BOOTSTRAP ROUTINE
                                                                                               *******************
                                                                                                                 ;INITIALIZE ALL CSR'S
;UNRELOCATE IF NECESSARY
;FLUSH OUT ANY DBE'S
;TURN OFF MEMORY MANAGEMENT
;TURN OFF THE UNIBUS MAP
;BOOT RKO OR RK1
ECCINIT ;TRAP ON DOUSET4 #BOOT1 ;TRAP
11997
11998
11999
12000 047524
12001 047526
12002 047534
12003 047546
12004 047552
12005 047554
12006 047560
12007 047562
12008 047570
12009 047572
12010 047574
12011 047600
12012 047604
12013 047610
12014 047614
12015 047616
12016 047620
12017 047624
12018 047626
12019 047630
                                                                                                                  ECCINIT ;TRAP ON DOUBLE BIT ERRORS (NORMAL)
SET4 #BOOT1 ;TRAPS TO 4 GOTO BOOT1
IF RLFLAG IS TRUE THEN SCALL UNRELOCATE
CALL MT0030 ;FLUSH OUT DBE'S
DEENERGIZE ;TURN OFF MEMORY MANAGEMENT
TST NO22BIT ;IS THIS AN 11/44 OR 11/24?
                                  104472
                                  004737
104421
005737
                                                      024566
                                                       002450
                                  001003
042737
005001
000005
012700
                                                                                                                                     BOOT1
#BIT5,MMR3
                                                                                                                  BNE
                                                                                                                                                                                                                     :TURN OFF THE UNIBUS MAP
                                                       000040 172516
                                                                                                                  BIC
                                                                                              B00T1:
                                                                                                                  CLR
                                                                                                                  RESET
                                                      177406
000004
177400
000005
                                                                                                                                     #177406,R0
R1,4(R0)
#177400,(R0)
                                                                                                                  MOV
                                   012700
010160
012710
012740
105710
100376
062701
005710
100761
005007
                                                                                                                  MOV
                                                                                                                  MOV
                                                                                                                                      #5,-(RO)
                                                                                                                  MOV
                                                                                              2$:
                                                                                                                  TSTB
                                                                                                                  BPL
                                                                                                                                      #BIT13,R1
                                                       020000
                                                                                                                   ADD
                                                                                                                  TST
                                                                                                                                       (RO)
                                                                                                                                      15
                                                                                                                                      PC
```

```
EXIT: SUBTST <<HALT PROGRAM>>
12022 047632
                                                 ***********
                                                                     HALT PROGRAM
                                                 : *SUBTEST
 12023 047632
12024 047636
12025 047652
12026 047654
12027 047656
12028 047660
12029 047664
                                                                     SHUTUP
                  004737 047664
                                                          IF APTFLAG IS TRUE OR ACTFLAG IS TRUE
                                                 EXIT2:
                                                           BR
                  000777
                                                 SEXHALT: HALT
12028 047660
12029 047664
12030
12031 047664
                                                              JMP
                                                                     START
                            003654
                                                           END : OF IF APTFLAG
                                                 SHUTUP: SUBTST <<SHUTDOWN DIAGNOSTIC>>
                                                 ************************
                                                                     SHUTDOWN DIAGNOSTIC
                                                  *SUBTEST
                                                  *********
                                                           :INITIALIZE ALL CSR'S
:UNRELOCATE
:FLUSH OUT DBE'S
                                                            RESTORE LOADERS
TURN OFF MEMORY MANAGEMENT
                                                            UNMAP THE UNIBUS MAP
12037
12041 047664
12042 047666
12043 047700
12044 047706
12045 047712
12046 047712
12047 047716
12048 047722
12049 047726
12050 047730
12051 047734
12052 047736
12056 047744
12057
                                                            CCINIT :TRAP ON DOUBLE BIT ERRORS (NORMAL)

IF RLFLAG IS TRUE THEN SCALL UNRELOCATE
                                                            ECCINIT
                  104472
                                                           IF QUICK IS FALSE
CALL MT0030
END : OF IF QUICK
MOV #1,R0
                                                                                          :FLUSH OUT DBE'S
                  004737 024566
                  012700
013701
004737
104421
005737
                                                                                          :DESTINATION BANK
                                                            MOV
                            000001
                                                                      LOADHOME , R1
                            002562
                                                                                          :SOURCE BANK
                                                           MOV
                                                                      BANKMOV
                            046410
                                                            CALL
                                                                                          TURN OFF MEMORY MANAGEMENT
                                                            DEENERGIZE
                                                                                          DOES THIS PDP-11 HAVE 22-BIT ADDR?
                                                                      NO22BIT
                                                            TST
                             002450
                  001003
042737
000207
                                                                      WBIT5, MMR3
                                                                                                    :TURN OFF UNIBUS MAP
                             000040 172516
                                                           RETURN
                                                 15:
                                                 APTDOWN:SUBTST <<APT SHUTDOWN SEQUENCE>>
                                                                    #APTDOWN, FIRST+24
#340, FIRST+26
#0, FIRST+APTDOWN
                                                 ****************
                                                 : *SUBTEST
                                                  *******************
 12059 047746
12060 047762
12061 047770
12062 047776
12063 050004
12064 050012
                                                            MAP
                                                            TESTAREA
                  012737
012737
012737
104417
                                                           MOV
                                                           MOV
                                                            MOV
                                                            KERNEL
                                                 APTHLT: HALT
```

12068	050016			*SUBTES	*****	< <block move="" subroutine="">> BLOCK MOVE SUBROUTINE</block>
12069 12070 12071 12072 12073 12074					:BLOCK2	HAS 3 ARGUEMENTS HAS 2 ARGUEMENTS HAS 1 ARGUEMENTS
12075 12076 12077 12078	050016 050024 050030 050034	012702 012701 000413	177640 000020	BLOCK1:	ALL ARE ENABL PUSH MOV MOV BR	CALLED BY THE BMOV MACRO LSB RO.R1.R2 WFASTCITY.R2 W16R1 3\$
12079 12080 12081 12082 12083	050036 050044 050050	012701 000404	000020	BLOCK2:	PUSH MOV BR	RO_R1_R2 #16R1 2\$
12083	050052	012501		BLOCK3:	PUSH	RO,R1,R2 (R5)+,R1 (R5)+,R2
12084 12085 12086 12087 12088	050052 050060 050062 050064	012501 012502 012500		2\$: 3\$:	MOV MOV	(R5)+,R2 (R5)+,R0
12088 12089 12090 12091	050066 050070 050072 050100	012022 077102		1\$:	MOV SOB POP	(R0)+,(R2)+ R1,1\$ R2,R1,R0 R5
12092 12093	050100	000205			RTS .DSABL	R5 LSB

```
.SBTTL FIELD SERVICE MODE
12095
12096
12097 050102
                                                                                                          <<SUBR FIELD SERVICE COMMAND MODE>>
                                                                 FIELDSERVICE:SUBTST
                                                                 **************
                                                                                             SUBR
                                                                                                          FIELD SERVICE COMMAND MODE
                                                                 : *SUBTEST
                                                                  **************
12098 050102
12099 050104
12100
                                                                               SAVREG
                       104415
                                                                               TYPE MSG020
                                                                                                                        :FIELD SERVICE COMMAND MODE
12100
12101 050110
12102 050124
12103 050130
12104 050132
12105 050134
12106 050134
12106 050140
12108 050142
12109 050146
12110 050156
12111 050160
12112 050166
12113 050174
12115 050176
12116 050202
12117 050204
12118 050210
                                                                               IF RLFLAG IS TRUE OR NOFSMODE IS TRUE
TYPE MSG048 ;NOT AVAILABLE
RESREG
                                                                                                                        :NOT AVAILABLE NOW - TRY LATER!
                        104416 000207
                                                                                  RETURN
                                                                               END : OF IF RLFLAG
                        005737
                                                                               TST
                                                                                             CACHKN
                                     002540
                                                                               BEQ
                                                                                                                      SAVE CACHE STATUS
SAVE CSR & KAMIKAZE STATUS
TURN CACHE OFF
                                                                               PUSH
                                                                                             CONTRL
                                                                               PUSH
                                                                                             CSRNO, KAMIKAZE
                                                                 15:
                                                                               CACHOFF
                        104424
                                                                                             KAMIKAZE
                                                                               SET
                                                                               TYPE
                                                                 FS1:
                                                                                             MSG026
                                                                                                                         : COMMAND:
                                                                                                                         :READ A DECIMAL NUMBER
:COMMAND --> RO
                                                                               RDDEC
                         104414
                                                                               POP
                                                                                            RO.#18.
                                                                               CMP
                        020027
                                      000022
                                                                               BLOS
12117 050204
12118 050210
12119 050212
12120 050222
12121 050224
12122 050226
12123 050230
12124 050232
12125 050234
12126 050236
12127 050240
12128 050242
12129 050244
12130 050246
12131 050250
12132 050252
12133 050254
12134 050256
12139 050266
12141 050266
12142 050266
12143 050270
12144 050276
                                                                                             MSG021
                                                                                TYPE
                        000766
                                                                               BR
                                                                                             FS1
                                                                               CASE RO
FSCMDO
                                                                  15:
                        050300
050402
050512
050660
051134
051454
052372
052400
052672
053076
                                                                                                                         EXIT FIELD SERVICE COMMANDS
                                                                                                                         READ CSR
LOAD CSR
EXAMINE MEMORY
                                                                                   FSCMD1
                                                                                   FSCMD2
FSCMD3
                                                                                                                         MODIFY MEMORY
SELECT BANK & PATTERN
TYPE CONFIGURATION MAP
                                                                                   FSCMD4
FSCMD5
                                                                                   FSCMD6
FSCMD7
                                                                                                                         SOB-A-LONG TEST
                                                                                    FSCMD8
                                                                                                                         :REFRESH TEST
                                                                                   FSCMD9
                                                                                                                         SET FILL COUNT
                                                                                   FCMD10
                                                                                                                         ENTER KAMIKAZE MODE
                                                                                   FCMD11
                                                                                                                         EXIT KAMIKAZE MODE
TURN CACHE OFF
TURN CACHE ON
TEST ONLY SELECTED BANKS
RESUME TESTING ALL BANKS
ENABLE TRACE
                                                                                   FCMD12
                                                                                   FCMD13
                                                                                    FCMD14
          050260
050262
050264
050264
050270
050276
                                                                                   FCMD15
                                                                                   FCMD16
FCMD17
                                                                                                                         :DISABLE TRACE
                                                                                   FCMD18
                                                                                END OF CASE
                         000733
```

12147 050300			FSCMDO:	SUBTST	< <comman< th=""><th>ID O</th><th>EXIT>></th><th>*****</th></comman<>	ID O	EXIT>>	*****
			: *SUBTES	ST	COMMAND	0	EXIT	
12148 050300 12149 050304 12150 050310 12151 050316	062706	000002	;*****	TYPE ADD IF SKIP	MSG103 #2.SP KAMÍ IS 1	RUF	;LEAVING	FIELD SERVICE MODE
12150 050310 12151 050316 12152 050322 12153 050326	062706 005037	000002 002006		ADD	#2.SP SKIPKAMI	1		THROW AWAY OLD KAMIKAZE FLAG
12154 050330 12155 050334 12156 050334				POP	IF SKIPK CSRNO	AMI		RESTORE OLD KAMIKAZE FLAG
12157 050340 12158 050344 12159 050346	005737 001414	002540		TST BEQ IF CACHI	CACHKN RESO KN EQ CAC	CHKF		:IF CACHE IS OFF
12160 050356	062706	000002		ELSE	#2.SP			: IF CACHE IS OFF : THROW AWAY CACHE STATUS
12161 050362 12162 050364 12163 050370 12164 050372 12165 050376	005737 001402	002540		BEQ POP	CACHKN RESO CONTRL			RESTORE CACHE STATUS
12166 050376 12167 050400	104416 000207		RESO:	END ; OF RESREG RETURN	IF CACHI	KN		
12168 12169 050402			FSCMD1:	SUBTST	< <fs< td=""><td>COMMAND</td><td>1</td><td>READ CSR>></td></fs<>	COMMAND	1	READ CSR>>
			*SUBTE	******* ST ******	FS	COMMAND	1	READ CSR
12170 050402 12171 050406	004737 010637	053674 002302		MOV	WHICHCS! SP,FSST/ #RES1	R ACK	.TDADC 1	TO 4 GOTO RES1
12173 050412	104426			SET4 READCSR			, INAFS I	O 4 GOTO REST
12172 050412 12173 050420 12174 050422 12175 050430 12176 050432 12177 050454	104026			SET ERROR RES4	NOERROR +26		;USE ERR	ROR ROUTINE FOR PRINTOUT
12177 050454	000207		RES1:	RES4 RETURN TYPE	MSG025			SR DOES NOT EXIST
12178 050456 12179 050462	013706	002302	REST.	MOV RES4	FSSTACK	.SP		
12180 050466 12181 050510	000207			RES4 RETURN			;RESET 1	TRAPS TO 4 TO DEFAULT

1.2	COMPAND		KEND COK						
12184	050512			FSCMD2: SUBTST	< <fs< td=""><td>COMMAND</td><td>2</td><td>LOAD</td><td>CSR>></td></fs<>	COMMAND	2	LOAD	CSR>>
				*SUBTEST	FS	COMMAND	2	LOAD	CSR
12185 12186	050512 050516	004737 010637	053674 002302	CALL	WHICHCSF SP,FSST/ #RES2	ACK	:TRAPS 1	0 4 0	GOTO RES2
12188 12189	050530 050532 050536	104426		SET4 READCSR TYPE SET	PG027 NOERROR				
12191 12192 12193	050544 050546 050570	104026		ERROR RES4 TYPE	+26 MSG023		:USE ERF	ROR RO	OUTINE FOR PRINTOUT TO 4 TO DEFAULT ORD AL NUMBER DC 'CSR'
12194 12195	050574 050576	104413		RDOCT	CSR		READ AN	IN LO	AL NUMBER DC 'CSR'
12196 12197 12198	050602 050604 050606 050612	104425 104426		LOADCSR READCSR TYPE SET	MSG028 NOERROR				
12200 12201	050620	104026		ERROR RETURN	+26				NTOUT - NOT AN ERROR
12202 12203	050622 050624 050630		002302	RES2: TYPE	MSG025 FSSTACK	,SP			TO 4 TO DEFAULT
12204 12205	050634 050656	000207		RES4 RETURN			KESET	IRAF3	TO 4 TO DEPAULT

12208	050660				FSCMD3:	SUBTST	< <fs< th=""><th>COMMAND</th><th>3</th><th>EXAMINE</th><th>MEMORY>></th><th>*****</th><th>******</th><th>*****</th><th>***</th></fs<>	COMMAND	3	EXAMINE	MEMORY>>	*****	******	*****	***
					*SUBTE			COMMAND	******	EXAMINE	********	*****		*****	•••
12209 12210 12211 12212 12213	050660 050700 050706 050712 050716 050720 050726 050730 050732 050734 050742 050744	012737	000002	002074	15:	PUSH MOV TYPE TYPE RDOCT	BANK,NOP #2,NOPAR MSG029 MSG031	AR,PARTH	ERE,4 ;INDICA ;EXAMIN ;PHYSIC	TE PARIT IE MEMORY AL ADDRE	Y ACTION SS (0-1777 BER ONTO S ANK 0	5776)?	? \$H10CT		
12214	050720	104413 013737	065140	002100		MOV	SHIOCT,B	ANK	PUT MS	B'S IN B	ANK				
12216 12217 12218 12218	050720 050730 050732 050734	000241 006100 006137 000241 006000 023737 003357 062700 032700 001352 020027 101347 012737	002100			CLC ROL ROL CLC	RO BANK								
12218 12219 12220 12221 12222 12223	050742 050744 050752	006000 023737 003357	002100	002552		ROR CMP BGT	RO BANK, LAS		: CHECK : BRANCH	FOR BANK	TOO HIGH				
12223 12224 12225	050754 050760 050764	062700 032700	060000 000001			BIT	#FIRST,R)	: CHECK	FOR ODD	ADDRESS				
12225 12226	050764 050766	020027	157776			BNE CMP BHI	RO.#LAST		: CHECK	FOR ADDR	ESS OVER 1	16K			
12226 12227 12228 12229 12230 12231 12233 12234 12236 12237 12238 12239 12239	050766 050772 050774 051002 051010 051024 051032 051034	012737	051046	002300		MOV SET4 MAP TESTARE	#35,PART #45 BANK	HERE	: INCASE : TRAPS : ENTER	OF ABOR TO 4 GOT MAP SU TEST MOD	ADDRESS ADDRESS ESS OVER 1 16K TS 0 4\$ PERVISOR S	SPACE (TEST ARE	A) TO	BANK
12232 12233 12234	051032 051034 051036	011001 104417				MOV KERNEL TYPOCS	(RO),R1			KERNEL M					
12235	051036 051044	000410				BR	EXCMD3								
12237 12238 12238	051046 051052	000405			3\$:	TYPE BR	MSG032 EXCMD3			;PARITY	ABORT				
12240	051054	062706	000004		45:	ADD TYPE	M4,SP MSG033 EXCMD3			:FIX ST	ACK T TRAP				
12242	051054 051060 051064	000400				BR	EXCMD3			, 12/16/00					
12241 12242 12243 12244 12245	051070	104417			EXCMD3:	POP	4,PARTHE	ERE, NOPA	R, BANK	KERNEL M		MIII T			
12246	051110	000207				RES4 RETURN			;KESE I	IRAPS 10	4 TO DEF	AUL I			

CZMSPA0 FS	MS11-L/I	M/P MEMO	RY DIAG.	MACRO M MEMORY	1113 26-	-APR-82 (9:41 PAGE 388		
12250	051134				FSCMD4:	SUBTST	< <fs command<="" td=""><td>*****</td><td>MODIFY MEMORY>> MODIFY MEMORY</td></fs>	*****	MODIFY MEMORY>> MODIFY MEMORY
12251 12252 12253 12254	051134 051154 051162 051166	012737	000003	002074	18:	PUSH MOV TYPE TYPE	BANK, NOPAR, PART #3, NOPAR MSG036 MSG031	HERE,4	;INDICATE PARITY ACTION ;MODIFY MEMORY AL ADDRESS (0-17775776)?? CTAL NUMBER ONTO STACK & SHIOCT B'S IN BANK B'S IN RO
12255 12256 12257	051172 051174 051202	104413	065140	002100		MOV POP	SHIOCT, BANK	; PUT MSE ; PUT LSE	S'S IN RO
12258 12259 12260 12261	051162 051166 051172 051174 051202 051204 051206 051210 051214 051216	000241 006100 006137 000241 006000	002100			CLC ROL ROL CLC ROR	RO BANK RO		
12263 12264 12265	051210 051220 051230 051234 051242 051250 051256 051264 051300 051302 051310	062700	060000			IF BANK ADD IF #BIT(GT LASTBANK THE #FIRST,RO D SET.IN RO THEN	O 15	CHECK FOR BANK TOO HIGH CHECK FOR ODD ADDRESS CHECK FOR ADDRESS OVER 16K INCASE OF ABORTS TO 4 GOTO 4\$ MAP SUPERVISOR SPACE (TEST AREA) TO BANK
12267 12267 12268 12269	051250 051250 051256 051264	012737	051316	002300		MOV SET4 MAP	#3\$ PARTHERE #4\$ BANK	;TRAPS	INCASE OF ABORTS TO 4 GOTO 4\$;MAP SUPERVISOR SPACE (TEST AREA) TO BANK
12270 12271 12272 12273	051300 051302 051310	104511 011001				TESTARE	(R0) .R1	:ENTER	TEST MODE
12274 12275 12276	051312	104417 000410				BR	S HERE MEANS WE	;ENTER	
12277 12278 12279	051316 051322	000431	000001		3\$:	TYPE BR	MSG032 EXCMD4	.EIV ST	;PARITY ABORT ;EXIT
12281 12282 12283	051324 051330 051334	062706 000424	000004		48:	ADD TYPE BR	#4,SP MSG033 EXCMD4	;FIX ST.;TIMEOU	;EXII
12284 12285 12286 12287 12288 12289	051336 051342 051350 051354 051356	104413			5\$:	TYPE TYPOCS TYPE RDOCT POP	MSG037 R1 MSG039	ENTER	OLD DATA WAS PRINT IT INPUT NEW DATA READ ON OCTAL NUMBER ONTO THE STACK GET NEW NUMBER
	051336 051342 051350 051354 051356 051360 051366 051370 051372 051374 051400	010110 011001 104417				TESTAREMOV MOV KERNEL TYPE TYPOCS	R1,(R0) (R0),R1 MSG038 R1	;ENTER	OLD DATA WAS PRINT IT INPUT NEW DATA READ ON OCTAL NUMBER ONTO THE STACK GET NEW NUMBER TEST MODE PUT IT IN MEMORY READ IT AGAIN KERNEL MODE DATA IS NOW PRINT IT
12295 12296 12297 12298	051406 051410 051430 051452	104417			EXCMD4:	KERNEL POP RES4 RETURN	4, PARTHERE, NOP		TRAPS TO 4 TO DEFAULT
12299	051452	000207				RETURN			

ZMSPA0	MS11-L/I	M/P MEMO	RY DIAG.	MACRO M	1113 26	-APR-82	09:41 PAGE 390		SEC	Q 03
	051454					SUBTST	< <fs commani<="" td=""><td>5</td><td>SELECT BANK & PATTERN>> SELECT BANK & PATTERN</td><td></td></fs>	5	SELECT BANK & PATTERN>> SELECT BANK & PATTERN	
12303 12304 12305 12306 12307 12308 12309	051454 051504 051510 051514 051520 051522 051526	010637 104413	002302		15:	PUSH MOV TYPE TYPE RDOCT POP IF BANK	BANK, PATTERN, TO SP, FSSTACK MSG040 MSG030 BANK GT LASTBANK THE	ESTADD,PC	SUMP, TKVEC, TKVEC+2 ;SAVE LAST GOOD STACK POINTER ;SELECT BANK & PATTERN TEST ;BANK(0-177)? ;READ AN OCTAL NUMBER ONTO THE STACK ;PUT IT IN BANK ;CHECK FOR BANK TOO HIGH	
12311 12312 12313 12314 12315 12316 12317	051536 051542 051544 051546 051556 051562 051564	013701 006301 006301	002100			END . OF	BANK,R1 R1 R1 II OFF.IN CONFIC MSG041 1\$;BANK NOT ACCESSABLE	
12315 12316 12317 12318 12319 12320 12321 12322 12323	051564 051570 051572 051576	104413			2\$:	TYPE RDOCT POP IF PATT	MSG042 PATTERN ERN GT #47 THEN	GOTO 2 \$:PATTERN(0-45)? :READ AN OCTAL NUMBER ONTO THE STACK :PUT IT IN PATTERN :CHECK FOR PATTERN TO HIGH :PATTERN 0 DATA IS?	
12323 12324 12325 12326 12327 12328	051614	104413				TYPE RDOCT POP END ; OF	R2		:PATTERN O DATA IS? :READ AN OCTAL NUMBER ONTO THE STACK :PUT IT IN R2	
12329		104511 004737	047020			MAP INVALID CALL IF RRFL	FYRANK		:MAP SUPERVISOR SPACE (TEST AREA) TO BAN :SET NEW MARGINS	K
12334 12335 12336 12337	051654 051660 051664 051664		052274	002152		TYPE JMP (END ; OF TYPE MOV	AG IS TRUE MSG049 MD5C IF RRFLAG MSG046 CSRNO, SAVCSR	;TO ESC	;BANK REQUIRES RELOCATION APE TYPE ANY KEY! LD CSR NUMBER	
12339 12340 12341 12342 12343	051676 051702 051706 051712 051716	013702 072227 016203 072327 042703	002150 002100 000002 002650 177770 177760	000.130		MOV ASH MOV ASH BIC ASL	BANK,RZ #2,R2		TE INDEX INTO CONFIGURATION TABLE OW WORD OF CONFIGURATION TABLE FOR THIS B ON CSR CODE IN BITS 0-3 ALL BUT THE CSR CODE CSR NUMBER	ANK
12345 12346 12346 12347 12348 12350	051624 051640 051642 051646 051654 051660 051664 051670 051676 051702 051706 051712 051716 051724 051730 051730 051736 051736 051736 051750 051750	013737 013702 072227 016203 072327 042703 006303 010337 012737 012737 012737 012737 012737	002150 052274 000340 130660 000200 000100	000060 000062 177776 130642		MOV MOV MOV BIC BIS	#-10,R3 #-10,R3 R3 R3,CSRNO #CMD5C,TKVEC #340,TKVEC+2 a\$TKB,R0 #BIT7,PSW #BIT6,a\$TKS	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	;KILL ANY OLD INTERRUPT ;LOWER CPU PRIORITY TO 140 ;ENABLE KEYBOARD INTERRUPTS	
12352 12352 12353 12354 12355	051764 052000 052004	013701 006301	002100		CMD58:	SET MOV ASL	HEADER, MUT BANK, R1 R1			

12356	052006	006301				ASL	R1		
12356 12357 12358 12359 12360 12361 12362 12363 12364 12365 12366 12367 12370 12371 12373 12376 12376 12377 12378 12383 12383 12383 12383 12383 12386	052006 052010 052014 052020 052026 052034 052034 052030 052064 052072 052072 052072 052102 052112 052114 052120 052114 052120 052134 052146 052146 052146	006301 005037 005037 012737 012737	002236 002262 060000 060002			CLR	SPLTCSR PASFLG		
12359	052020	012737	060000	002406 002410		MOV	PASFLG #FIRST,TESTADD #FIRST+2,TESTADD #FIRST+2,TESTADD 12 SET.IN CONFIG+ SPLTCSR BANK #120000,TESTADD+ 1F #BIT12 SET.IN @SWR CSRNO		
12360	052026	012737	060002	002410		IF #BIT	12 SET.IN CONFIG+	2(R1)	
2362	052044	005237	002236			INC	SPLTCSR		
2363	052050	012737	120000	002410		MOV	#120000, TESTADD+	2	
2365	052072					END; OF	IF #BIT12		
2367	052102	104470				ECCDI	SEI.IN WOWN		;DISABLE ERROR CORRECTION
2368	052104					ELSE	CSRNO		
2370	052112	104502				CLRCSI	3		;CLEAR CSRS
2371	052114					POP OF	CSRNO		
2373	052120	012737	200000	002074		END ; OF MOV MOV MOV	#2.NOPAR		:PARITY ACTION :TRAPS ADD 2 TO PC
2374	052126	012737	000002 000002 002110	002322		MOV	#2.PCBUMP PATTERN,RO		TRAPS ADD 2 TO PC
2376	052140	012737 012737 013700 006300 004770				ASL	RO		
2377	052142	004770	052154 002074			CALL	af SPAT (RO) NOPAR		
2379	052152	005037 000712	0020.4			BR	CMD5B		:LOOP TILL KEYBOARD INTERR
2380	052154	020700			FSPAT:	MT0000 MT0001 MT0002 MT0003 MT0004 MT0005 MT0006 MT0007 MT00010	;<1 SEC	DATA PAT	TERN TEST
2382	052154 052156 052160 052162 052164 052166 052170	020700 020760 021100 021240 021472 021614 021750				MT0001	<pre></pre>	ADDRESS COMPLEM	TEST FNT ADDRESS TEST
2384	052162	021240				MT0003	: 1 SEC	3 XOR 9	WORST CASE NOISE TEST G ZEROS TEST G ONES TEST DATA TEST DRESSING TEST
2385	052164	021472				MT0004	: 1 SEC	ROTATING	G ZERUS TEST
12387	052170	021750				MT0006	<1 SEC	INITIAL	DATA TEST
12388	052172 052174	022004 022046				MT0007	:<1 SEC :<1 SEC :<1 SEC	ADDRESS RYTE ADI	DRESSING TEST
2390	052176	022102				MT0011	:<2 SEC	CREATE	SINGLE BIT ERROR TEST
12391	052200	022160				MT0012	SEC	CREATE I	DOURLE RIT FROM TEST
2393	052204	022350				MT0014	: 1 SEC	BASIC D	DUBLE BIT ERROR TEST
2394	052206	022440				MT0015	: 1 SEC	WRITE II	NHIBIT OF BYTE WITH DBE
2396	052212	022574				MT0017	SEC SEC	HOLDING	1'S & 0'S TEST
2397	052214	022616				MT0020	: 1 SEC	SYNDROM	ES TO CSR ON SBE TEST
2399	052220	023160				MT0022	10 SEC	REFRESH	& SHIFTING DIAGONAL TEST
2400	052222	023212				MT0023	:10 SEC	SHIFTIN	G DIAGONAL TEST
12402	052226	023522				MT0025	:<1 SEC	INTERRU	PT ENABLE TEST
12403	052230	023600				MT0026	:<1 SEC	RANDOM	DATA TEST
12390 12391 12392 12393 12394 12395 12396 12397 12400 12401 12403 12404 12406 12407 12408 12409 12410	052176 052200 052202 052204 052206 052210 052212 052214 052220 052222 052224 052224 052230 052232 052234 052236 052236 052236 052236 052236 052236 052236 052236	022102 022160 022264 022350 022440 022516 022574 022616 023160 023160 02312 023600 024102 024566 025070 025260 025612 026000 026152 026336				MT0011 MT0012 MT0013 MT0014 MT0015 MT0016 MT0020 MT0021 MT0023 MT0024 MT0025 MT0027 MT0030 MT0031 MT0033 MT0033 MT0033 MT0033 MT0033	<pre></pre>	FLUSH O	DRESSING TEST SINGLE BIT ERROR TEST YTE CLEARS SBE TEST DOUBLE BIT ERROR TEST DUBLE BIT ERROR TEST NHIBIT OF BYTE WITH DBE NHIBIT OF WORD WITH DBE 1'S & O'S TEST ES TO CSR ON SBE TEST G O'S & 1'S TEST & SHIFTING DIAGONAL TEST LLOPING PATTERN TEST PT ENABLE TEST DATA TEST BANK TEST UT DBE'S TEST ONG TEST FOODE TEST ASE NOISE PARITY TEST ION CODE TEST ABLE TEST
12406	052236	025070				MT0031	3 SEC	SOB-A-L	ONG TEST
12407 12408	052242	025612				MT0032	35 SEC	BRANCH	GOBBLE TEST
12409	052244	026000				MT0034	: 1 SEC	SOFT ER	ROR TEST
	1157766	ロノトエラブ				MI UUSS	: CI 25 C	MUK21 C	MOTOR LAWTH 1E91

١	CZMSPAO	MS11-L/M/P	MEMORY DIAG.	MACRO	M1113	26-APR-82	09:41	PAGE	390-2
---	---------	------------	--------------	-------	-------	-----------	-------	------	-------

1.2	COMPANIE	,	SEEEC	DUIN A L	A1161111			
12413 12414 12415 12416 12417 12418 12419 12420	052254 052256 052260 052262 052264 052266 052270 052272	026404 026406 026450 026514 026554 026614 026654				MT0040 MT0041 MT0042 MT0043 MT0044 MT0045 MT0046 MT0047	: 1 SEC : 1 SEC	NO WRITE ABORT WITH ECC DISABLED TEST ADDRESS TO CSR ON DOUBLE BIT ERROR TEST EXTENDED UNIBUS ADDRESS TEST WRITE BYTE CLEARS SBE TEST SHIFTING 1/0'S THROUGH THE CHECK BITS SYNDROMES TO CSR ON DBE TEST CHECK SINGLE BIT ERROR WITH ECC DISABLED TEST NO CSR UPDATE WITH SBE ON DBE TEST
12422	052274 052300	013706	002302	130320	CMD5C:	MOV BIC POP	FSSTACK, SP #BIT6, @STKS	RECOVER OLD STACK POINTER
12424 12425 12426	052306 052316 052322		130306			POP MOVB POP	TKVEC+2, TKVEC a\$TKB, RO PCBUMP, TESTADD	GET CHARACTER TO GET RID OF FLAG
12427 12428	052332 052342					POP	PATTERN, BANK BANK	REMAP OLD BANK
12429 12430 12431	052356 052362 052370	004737 013737 000207	047020 002152	002150		CALL MOV RETURN	EXBANK SAVCSR, CSRNO	:RESTORE CSRNO.

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 391 FS COMMAND 5 SELECT BANK & PATTERN

12433 052372

COMMAND 6 TYPE CONFIGURATION MAP>> FSCMD6: SUBTST <<FS *********** *SUBTEST TYPE CONFIGURATION MAP FS COMMAND 6 *************************

12434 052372 004737 041352 12435 052376 000207 12436

CALL PCONFIG

ZMSPAO S	MS11-L/COMMAND	M/P MEMO	RY DIAG.	MACRO M	1113 26- ION MAP	-APR-82 (9:41 PAGE 393		
	052400				FSCMD7:	*******	FS COMMAND	7	SOB-A-LONG TEST>> ***********************************
12440 12441 12442	052400 052424 052430	010637	002302			PUSH MOV TYPE	BANK, PATTERN, TKV SP, FSSTACK MSG055	EC,TKVEC	+2,NOPAR ;SAVE LAST GOOD STACK POINTER ;SOB-A-LONG TEST
12444	052434 052444 052446 052450	104470				ECCDIS	SET.IN aSWR		DISABLE ERROR CORRECTION
12447	052450	104502				CLRCS			CLEAR CSRS
12449	052452					END ; OF	MSG056		BELL = EACH PASS COMPLETE
7 7/ 5/1	052456 052462 052470 052476 052502 052510	012737	052606	000060		TYPE MOV	MSG046 #CMD7C,TKVEC		TO ESCAPE TYPE ANY KEY!
12453	052476	017700	052606 000340 130126	000062		MOV	aSTKB,RO		KILL ANY OLD INTERRUPT
12455 12456 12457	052502 052510	012737 012737 017700 042737 052777	000200	177776 130110		BIC	#CMD7C,TKVEC #340,TKVEC+2 a\$TKB,R0 #BIT7,PSW #BIT6,a\$TKS		:KILL ANY OLD INTERRUPT :LOWER CPU PRIORITY TO 140 :ENABLE KEYBOARD INTERRUPTS
12458 12459 12460	052516					SET	HEADER, MUT		
12461	052536	004737	047020		CMD7B:	CALL	K := #0 TO LASTBA		C FALCE
12463 12464	052542 052556 052560	104511				INV	FLAG IS TRUE AND	KKFLAG I	5 PALSE
12465 12466	052560 052564	104511 004737	025070			CAL	L MT0031		
12467	052564 052600 052604					END : OF TYPE GOTO	OF IF ACFLAG FOR BANK \$BELL CMD7B		RING BELL
12470 12471	052606	013706	002302		CMD7C:	MOV	FSSTACK, SP		RECOVER OLD STACK POINTER
12472 12473 12474	052612 052620 052624	042777	000100 130004	130006		BIC MOVB POP	#BIT6, astks astkb, ro Nopar, tkvec+2, th	CVEC,PATT	READ CHAR TO KILL FLAG
12475 12476 12477	052650	004737	047020			MAP CALL RETURN	BANK EXBANK		MAP SUPERVISOR SPACE (TEST AREA) TO BANK

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 395
                                       SOB-A-LONG TEST
                                                                                                                                    ERROR SUMMARY>>
                                                                                                          COMMAND 8
                                                                  FSCMD8: SUBTST <<FS
   12480 052672
                                                                  ***************
                                                                                                         COMMAND 8
                                                                                                                                    ERROR SUMMARY
                                                                  : *SUBTEST
                                                                  ************
                                                                                            RO,R2,R3,BANK
SPASS,TEMP
                                                                               PUSH
   12481 052672
12482 052704
12483 052712
12484 052716
12485 052724
12486 052730
12487 052736
12489 052750
12490 052754
12491 052760
12492 052764
12493 052770
12494 052776
12494 052776
12496 053016
12497 053016
12498 053016
12498 053016
12499 053026
12500 053032
12501 053032
12501 053036
12502 053042
12503 053046
12504 053062
12506 053062
12506 053062
12507 053074
                          013737
005337
                                                    002430
                                                                                             TEMP
                                                                                TYPDEC
                                                                               TYPE
                                                                                             MSG125
                                                                                                                                    :PASSES COMPLETED
                                                                               TYPDEC
                                                                                            SERTTL
                                                                                             MSG079
                                                                                                                                    :ERROR(S) DETECTED
                                                                                TYPE
                                                                               IF SERTTL NE #0
CLR SUCCESS
                                                                                  CLR
                          005037
                                       002330
                                                                                  FOR BANK := #0 TO LASTBANK
MOV BANK, R3
MUL #4, R3
                          013703
070327
                                       002100
                                                                                      IFB CONFIG+2(R3) NE #0
IF SUCCESS IS FALSE
                                                                                             TYPE
                                                                                                          MSG076
                                                                                                                                    BANK ERRORS
                                                                                             SET SUCCESS
                                                                                         END OF IF SUCCESS
                                                                                                         BANK,3
CONFIG+2(R3),R0
#^C377,R0
                          116300
042700
                                                                                         MOVB
                                                                                         BIC
                                                                                                          RO
SCRLF
                                                                                         TYPDEC
                                                                                         TYPE
                                                                               END : OF IFB CONFIG(R3)
END : OF FOR BANK
END : OF IF SERTTL
POP BANK, R3, R2, R0
                                                                               RETURN
                          000207
```

CZMSPAO MS11-	L/M/P MEMOR	Y DIAG. MACRO	M1113	26-APR-82 09:41	PAGE 397
---------------	-------------	---------------	-------	-----------------	----------

12	CUMMAND	0	EKKUK 3	UPPART				
12510	053076				FSCMD9:		FS COMMAND 9	REFRESH TEST>> REFRESH TEST
12511 12512 12513	053076 053122 053126	010637	002302			PUSH MOV TYPE	BANK, PATTERN, TKVEC, TKVI SP, FSSTACK MSG073	C+2.NOPAR ;SAVE LAST GOOD STACK POINTER ;REFRESH TEST
12515 12516 12517	053132 053142 053144 053146 053150 053150	104470				ELSE ELSE		;DISABLE ERROR CORRECTION
12518 12519 12520	053146 053150 053150	104502				CLRCSF END ; OF TYPE	IF MSG056	;CLEAR CSRS ;BELL = EACH PASS COMPLETE
12523 12524 12524	053154 053160 053166 053174 053200 053206 053214	012737 012737 017700 042737 052777	053304 000340 127430 000200 000100	000060 000062 177776 127412		TYPE MOV MOV BIC BIS	MSG046 #CMD9C.TKVEC #340.TKVEC+2 a\$TKB.RO #BIT7.PSW #BIT6.a\$TKS	;TO ESCAPE TYPE ANY KEY! ;KILL ANY OLD INTERRUPT ;LOWER CPU PRIORITY TO 140 ;ENABLE KEYBOARD INTERRUPTS
12528 12529	053214					SET	HEADER, MUT	
12532 12533 12534 12535 12536 12537 12538	053254 053254 053256 053262 053262 053262 053276 053302	004737 104511 004737	047020 023160		CMD9B:	IF ACI	C:= #0 TO LASTBANK EXBANK FLAG IS TRUE AND RRFLAG ALIDATE L MT0022 DF IF ACFLAG FOR BANK \$BELL CMD98	;RING BELL
12541 12542 12543 12544 12545	053304 053310 053316 053322 053346 053362 053366	013706 042777 117700 004737 000207	002302 000100 127306 047020	127310	CMD9C:	MOV BIC MOVB POP MAP CALL RETURN	FSSTACK, SP #BIT6, 3\$TKS 3\$TKB, RO NOPAR, TKVEC+2, TKVEC, PA BANK EXBANK	;RECOVER OLD STACK POINTER ;READ CHAR TO KILL FLAG TTERN, BANK ;MAP SUPERVISOR SPACE (TEST AREA) TO BANK
12548	053300	000207				KETOKA		

CZMSPA0 FS	MS11-L/COMMAND	M/P MEMO	RY DIAG.	MACRO M11	113 26-APR-82	09:41 PAG	SE 399						SEQ 032
	053370				CMD10: SUBTST	< <fs (<="" th=""><th>COMMAND</th><th>*****</th><th>****</th><th>L COUNT></th><th>******</th><th>******</th><th>****</th></fs>	COMMAND	*****	****	L COUNT>	******	******	****
12552 12553 12554 12555 12556 12557 12558 12559	053370 053372 053376 053400 053402 053406 053412 053414	104413 042700 110037 000207	177760 002353		PUSH TYPE RDOCT POP BIC MOVB POP RETURN	RO MSG085 RO #^C17,RO RO,\$FILLS	s		;FILL (OUNT (OCT	AL)?		
12560 12561	053416				FCMD11: SUBTST	*****	COMMAND COMMAND	*****	****	AMIKAZE AMIKAZE	*****	******	****
12562 12563 12564	053416 053422 053436 053440	000207			TYPE SET RETURN	MSG101 KAMIKAZE	,SKIPKAP	ENTERIN	G KAMI	AZE MODE			
12565	053440				FCMD12: SUBTST	*****	COMMAND COMMAND	*****	*****	MIKAZE M MIKAZE M	*****	********	****
12567 12568 12569 12570	053440 053444 053450 053456	005037 000207	002004		TYPE CLR SET RETURN	MSG102 KAMIKAZE SKIPKAMI		;LEAVING	KAMIK	AZE MODE			
12571 12572	053460				FCMD13: SUBTST	******	COMMAND	*****	*****	ACHE OFF	******	********	****
12573 12574 12575 12576 12576	053460 053464 053466 053474 053500 053502	104424 013737 005037 000207	002540 002540		TYPE CACHOF MOV CLR RETURN	CACHKN, C	ACHKN+2		: TURN	IS OFF CACHE OFF OLD CACHE CACHE OFF	ON STATE		
12579	053502				FCMD14: SUBTST	******	COMMAND	******	*****	ACHE ON>	******	*******	****
12580 12582 12583 12583 12584	053502 053506 053514 053516	013737 104423 000207	002542	002540	TYPE MOV CACHON RETURN	MSG107 CACHKN+2	, CACHKN		:CACHE :RESTO :TURN	IS ON (E RE OLD CA CACHE ON	CHE ON ST	ING ACTUAL P	ATTERNS)

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 400 FS COMMAND 14 TURN CACHE ON
```

12597 12598 053520 FCI	MD15: SUBTST < <fs 15="" banks="" command="" only="" selected="" test="">></fs>
1 1	MD15: SUBTST < <fs 15="" banks="" command="" only="" selected="" test="">> SUBTEST FS COMMAND 15 TEST ONLY SELECTED BANKS</fs>
**	
12599 053520 12600 053524 004737 053614	TYPE MSG105 ;ENTER BANKS IN OCTAL - USE NUMBER OUTSIDE RANGE TO TERMINAT CALL CMD16A ;ERASE OLD SELECTIONS
12599 053520 12600 053524 004737 053614 12601 053530 12602 053530	REGIN CMD16LOOP
1202 053530	REPEAT TYPE MSG030 ;BANK(0-177)?
12604 053534 104413	TYPE MSG030 ;BANK (0-177)? RDOCT ;READ AN OCTAL NUMBER ONTO THE STACK POP R1 ;PUT IT IN R1
12605 053536	POP R1 ;PUT IT IN R1
12606 053540	IF R1 GT #177 OR R1 LT #0
1 12607 053552	LEAVE CMD16LOOP END : OF IF R1 ASL R1 ASL R1 ASL R1 :R1 <- R1 * 4
12608 053554 12609 053554 006301	ASI DI
12610 053556 006301	ASL R1 :R1 <- R1 * 4
12609 053554 006301 12610 053556 006301 12611 053560 052761 040000 002652 12612 053566 12613 053570 12614 053570 12615 053574	BIS #BIT14.CONFIG+2(R1) END : OF REPEAT END CMD16LOOP
12612 053566	END OF REPEAT
12613 053570	TYPE MSG110 ;ONLY SELECTED BANKS WILL BE TESTED
12614 053570	TYPE MSG110 ;ONLY SELECTED BANKS WILL BE TESTED SET SELONLY
12615 053574 12616 053602 000207	RETURN
12617	
12618 053604 FC	MD16: SUBTST < <fs 16="" all="" banks="" command="" resume="" testing="">></fs>
1.5	SUBTEST FS COMMAND 16 RESUME TESTING ALL BANKS
	SUBTEST FS COMMAND 16 RESUME TESTING ALL BANKS
12619 053604	TYPE MSG111 ;ALL BANKS WILL BE TESTED
12620 053610 005037 002000	CLR SELONLY
12619 053604 12620 053610 005037 002000 12621 12622	
12622	ENTRY POINT FROM CMD15
12623 053614 013702 002552 CM	D16A: MOV LASTBANK, R2
12624 053620 006302 12625 053622 006302	ASL R2 ASL R2
12626 053624	FOR R1 := #0 TO R2 BY #4
12623 053614 013702 002552 CM 12624 053620 006302 12625 053622 006302 12626 053624 12627 053626 042761 040000 002652	FOR R1 := #0 TO R2 BY #4 BIC #BIT14, CONFIG+2(R1) END : OF FOR R1
1 12628 033634	END ; OF FOR R1
12629 053644 000207	RETURN

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 402 FS COMMAND 16 RESUME TESTING ALL BANKS FCMD17: SUBTST <<FS COMMAND 17 ENABLE TRACE>> 12632 053646 *SUBTEST FS COMMAND 17 ENABLE TRACE TYPE MSG127 MOV #-1,TRACE 12633 053646 12634 053652 012737 177777 006204 12635 053660 000207 RETURN

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 404 FS COMMAND 17 ENABLE TRACE

12638 053662

FCMD18: SUBTST <<FS COMMAND 18 DISABLE TRACE>>

*SUBTEST FS COMMAND 18 DISABLE TRACE

12639 053662 12640 053666 005037 006204 12641 053672 000207 TYPE MSG128 CLR TRACE RETURN

12644	053674			******	*****	****	NE CORRECT CSR>>	*******
				*SUBTE	ST ******	*****	NE CORRECT CSR	*************
12645 12646	053674 053700	013700 022700 001003 005037 000207	002222 100000		MOV	TOTCSRS,RO	GET CSR'S FLAG CSR 0? NO - SKIP YES - SET IT UP	
12647	053704 053706 053712	001003 005037	002150		BNE	1\$ CSRNO	:NO - SKIP :YES - SET IT UP	
12648 12649 12650		000207			RETURN			
12651 12652	053714 053720 053722 053724	104412		1\$:	TYPE	MSG022	:WHICH CSR(O-F) :GET CHARACTER	
12653 12654	053722 053724	011000			POP MOV	RO (RO),RO	PUT IN RO	
12655 12656	053726 053732	101370	000106		BHI	RO.#106	CHECK LIMIT	ES IT RIGHT
12657 12658	053734	011000 020027 101370 022700 103002 162700 162700	000101		BHI CMP BHIS	#'A,R0 2\$ #7,R0		
12659	053742	162700	000007 000060	2\$:	SUB	#60,R0 R0		
12661 12662	053752 053754 053760	006300 010037 000207	002150		ASL MOV RETURN	RO.CSRNO		
12663	001550	000501			146 1 0144			

```
SPER25: LET ADDRESS := R1 - #2
IF ABORTFLAG IS FALSE
                                                                            ERROR DATA (SUPERVISOR) SETUP STUFF
        053762
053774
054002
054010
054016
054020
054020
054026
                                                                      TESTAREA
                                                                                                      ENTER TEST MODE
                                                                      LET BAD := -2(R1)
                                                                      KERNEL
                                                                                                      ENTER KERNEL MODE
                    104417
                                                                   END : OF IF ABORTFLAG
IF 177654 EQ #0
                                                                     LET GOOD := R2
                                                                   ELSE
                                                                     LET GOOD := R3
                                                                   END ; OF IF
13224 054040
13225
13226 054044
        054040
                                                                   JMP
                                                                              PERRAW
                    000137 057242
                                                       PERRA3: SUBTST <<DATA WAS 3 WORDS>>
                                                       ************
                                                       : *SUBTEST
                                                                              DATA WAS 3 WORDS
                                                       *******************
                                                                   IF BADPC EQ #0 THEN SCALL BADSTACK
13227 054044
13228 054056
13229 054060
13230 054064
13231 054066
13232 054074
13233 054076
13234 054100
13235 054106
13237 054110
13238 054114
13239 054120
13240 054124
13241 054130
13242 054136
13243 054146
13244 054146
13245 054150
13246 054154
13247 054160
13248 054164
13249 054164
13250 054202
13253 054204
13254 054206
13255 054206
                                                                   PUSH
                                                                   CLR
                                                                                                      :MAKE SURE CSR BIT HOLDER IS CLEAR
                     005037
                                                                              CSR
                                002146
                                                                                                      ;DISABLE ECC & WRITE CHECKBITS FROM 1 SELECTED CSR
                                                                   CHK1DIS
                     104505
                                                                   TESTAREA
                    005711
104417
104426
013700
                                                                                                      READ LOCATION TO READ CHECKBITS INTO CSR
                                                                   TST
                                                                               (R1)
                                                                   KERNEL
                                                                                                      :GET CSR CONTENTS
                                                                   READCSR
                                                                              CSR,RO
                                                                                                      SAVE CSR CONTENTS IN RO
                                                                   MOV
                                002146
                                                                                                      RETURN CSR TO NORMAL MODE :MOVE CHECK BITS TO BOTTOM OF WORD
                    104503
072027
042700
                                                                   CLR1CSR
                                                                                                     CLEAR OFF EXTRANEOUS GARBAGE
SAVE VIRTUAL ADDRESS FOR PRINTOUT
FIRST TEST WORD WRITTEN SHOULD ALWAYS BE ZERO
ENTER TEST MODE
                                                                               #-5.RO
#^C177.RO
                                                                   ASH
                                177600
                                                                   LET ADDRESS := R1
                                                                   CLR
                                                                               GOOD
                     005037
                                002042
                                                                   TESTAREA
                    011137
011437
104417
110037
105037
004737
104033
                                                                                                      GET BAD DATA FROM MUT - FIRST WORD
                                                                               (R1),BAD
                                                                   MOV
                                                                               (R4),BAD2
                                                                                                       AND SECOND WORD
                                                                   MOV
                                                                                                       ENTER KERNEL MODE
                                                                   KERNEL
                                002054
002055
057476
                                                                               RO,BAD3
BAD3+1
                                                                                                      MOVE BAD CHECKBITS FOR PRINTOUT
                                                                   MOVB
                                                                                                      CLEAR OFF THE OTHER UNUSED BITS
                                                                   CLRB
                                                                              PERBNK
+33
                                                                                                      MARK BANK AS BAD IN CONFIG TABLE
                                                                    CALL
                                                                    ERROR
                                                                   POP
                                                                                                      :RESTORE RO
                                                                   IF #SWO SET.IN aSWR
                                                                                                      TRAP ON SINGLE BIT ERRORS
                                                                      ENASBE
                     104506
                                                                      ECCINIT
                                                                                                      :TRAP ON UNCORRECTABLE ERRORS
                     104472
                                                                    END; OF IF #SWO
                     000002
```

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 423 DATA WAS 3 WORDS
                                          SPER30: LET GOOD := R1
LET ADDRESS := (SP) - 16
IF ABORTFLAG IS FALSE
                                                                            :ENTER TEST MODE
                                                     TESTAREA
                                                     LET BAD := @ADDRESS
                                                                                     :ENTER KERNEL MODE
                                                     KERNEL
                 104417
                                                   END : OF IF ABORTFLAG
  13265 054252
13266
13267 054256
                 000137 057242
                                           GETDATA: SUBTST <<GET DATA FROM ABORTED AREA IF POSSIBLE>>
                                           *************
                                           : *SUBTEST
                                                            GET DATA FROM ABORTED AREA IF POSSIBLE
                                           *************************
                                                           RO.4.114
SP.GETDA1
#1$.4
#1$,114
                                                   PUSH
                 010637
012737
012737
013700
                                                   MOV
                                  000004
                                                   MOV
                                                   MOV
                                                            ADDRESS, RO
                                                   MOV
                                                   TESTAREA
                 011037
104417
005037
013706
                         002050
                                                            (RO),BAD
                                                   MOV
                                                   KERNEL
                                                           ABORTFLAG
GETDA1,SP
114,4,RO
                                                   CLR
                                                                                     RESTORE KNOWN GOOD STACK POINTER
                                           15:
                                                   MOV
                                                   POP
                                                   RETURN
                 000207
                                           GETDA1: 0
```

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 425 POWER FAIL AUTO RESTART

SBTTL POWER FAIL AUTO RESTART SBTTL ROUTINE POWER DOWN AND UP 13285 13286 13287 054356 SPWRDN:	***
13295 13296 054356 005737 002540 13297 054362 001403 13298 054364 13298 054364 ; SAVE CACHE STATUS TST CACHKN BEQ 5\$ PUSH CONTRL	
13296 054356 005737 002540 13297 054362 001403 13298 054364 13299 054370 104423 13300 054372 012737 055330 000024 5\$: MOV #\$ILLUP.PWRVEC ;:SET FOR FAST UP 13301 054400 012737 000340 000026 13302 054406 13303	
13304 054426 012700 177700 MOV #177700,R0 13305 054432 012701 000021 MOV #17.,R1 13306 054436 1\$: PUSH -(R0)	
13308 13309 054442 005737 002452 13310 054446 001013 13311 054450 012700 172300 13312 054454 012701 000020 13313 054460 13314 054462 077102 13315 054464 13315 054464	
13309 054442 005737 002452 TST NOSUPER 13310 054446 001013	

MODITIVE	LOWEL D	OWIT AIT	0,			
13325	05/53/					SAVE USER & SUPERVISOR STACK POINTERS USER
13326 13327 13328 13329	054524 054532 054534 054536 054546 054546 054556 054566 054566 054566 054572 054576 054576 054604 054604 054604 054604 054660 0546604 054660 054660 054660 054660	010600 104417				MOV USP,RO KERNEL PUSH RO ;ENTER KERNEL MODE
13330 13331	054540	005737	002452			TST NOSUPER BNE 7\$
13332	054546	010400				SUPERVISOR ;ENTER SUPERVISOR MODE MOV SSP,RO
13334	054556 054560	010600 104417				KERNEL ;ENTER KERNEL MODE
13336	054562	013701	002222		7\$:	SAVE ECC REGISTERS MOV TOTCSRS.R1 ;GET CSR'S BEGIN LCSRSAVE
13330	054566					FOR CSRNO := #0 TO #36 BY #2
13340	054572	006301				ASL R1
13341	054574	104426				ON. ERROR READCSR
13343	054600	104420				DIICH CCD
13344	054604					END ; OF ON.ERROR
13345	054610					END : OF ON.ERROR IF R1 EQ #0 THEN LEAVE LCSRSAVE END : OF FOR CSRNO END LCSRSAVE ; SAVE MMR0,1,2,3 PUSH MMR0,MMR1,MMR2
13347	054626					END LCSRSAVE
13348	054626					PUSH MMRO MMR1 MMR2
13350	054642	005737	002452			131 NUSUPER
13351	054646	001002				BNE 8\$ PUSH MMR3
13352	024620					SAVE KERNEL PAR'S MOV #172400,RO
13354	054654	012700	172400		8\$:	MOV #172400,R0
13355	054660	012701	000020		45:	MOV #16R1 PUSH -(RQ)
13357	054666	077102				SOR R1.45
13358	05/470	022777	000001	003752		SAVE UNIBUS MAP REGISTERS CMP #1, PROTYP ; IS THIS AN 11/44?
13360	054676	022737	000001	003/32		BNE 9\$;BRANCH IF NOT
13361	054700					PUSH MAPHO, MAPLO
13362	054710				9\$:	SAVE POSSIBLE SOFTWARE SWITCH REGISTER
13364	034110					SAVE STACK POINTER
13365	054714	010637	055334			MOV SP. SSAVR6 ;; SAVE SP ; NOW SET UP REAL VECTOR
13367	054720	012737	054732	000024		MOV #SPWRUP, PWRVEC ;; SET UP VECTOR
13368 13369	054676 054700 054710 054714 054720 054726 054730	000000 000776			\$DOWN:	HALT BR SDOWN ;;HANG UP

4.2220		
13372 13373	POWER	UP ROUTINE
1 1777/ NE/ 773	900024 SPWRUP:	MOV #\$ILLUP, PWRVEC ;; SET FOR FAST DOWN ;RESTORE STACK POINTER MOV \$SAVR6, SP ;; GET SP ;; GET SP
13378 054732 012737 055330 13379 13380 054740 013706 055334 13381 054744 005037 055334 13382 054750 005237 055334 13383 054754 001375 13384 13385 054756 13386 054762 022737 000001	15:	MOV \$SAVR6, SP :: GET SP CLR \$SAVR6 :: WAIT LOOP FOR THE TTY INC \$SAVR6 :: WAIT FOR THE INC BNE 1\$:: OF A WORD : RESTORE POSSIBLE SOFTWARE SWITCH REGISTER
13385 054756		POP aswR
13386 13387 054762 022737 000001 13388 054770 001006	003752	RESTORE UNIBUS MAP CMP #1.PROTYP ; IS THIS AN 11/44? BNE 10\$
13388 054770 001006 13389 054772 13390 055002 004737 046344 13391		POP MAPLO, MAPHO CALL LOWMAP ;SETUP LOWER 16K OF UNIBUS MAP
13393 055012 012702 172300 13394 055016 012701 000020	10\$:	CALL LOWMAP :SETUP LOWER 16K OF UNIBUS MAP :RESTORE KERNEL PAR'S & PDR'S MOV #172340,RO MOV #KIPDRO,R2 MOV #16.,R1
13395 055022 13396 055024 012722 077406 13397 055030 077104 13398	6\$:	POP (R0)+ MOV #77406,(R2)+ SOB R1,6\$;RESTORE MMR3,2,1,0
13398 13399 055032 005737 002452 13400 055036 001002 13401 055040 13402 055044 13403	11\$:	TST NOSUPER BNE 11\$ POP MMR3 POP MMR2,MMR1,MMR0
13403 13404 055060 013701 002222 13405 055064 042701 177400 13406 055070		RESTORE ECC REGISTERS MOV TOTCSRS,R1 ;GET CSR'S BIC #177400,R1 REGIN L CSRRESTORE
13407 055070 13408 055076 006201 13409 055100		BEĞIN LCSRRESTORE FOR CSRNO := #36 DOWNTO #0 BY #2 ASR R1 ON.ERROR
13410 055102 13411 055106 104425 13412 055110 13413 055110 13414 055114 13415 055132 13416 13417 055132 012700 172300		POP CSR LOADCSR END : OF ON. ERROR IF R1 EQ #0 THEN LEAVE LCSRRESTORE END : OF FOR CSRNO END LCSRRESTORE
13416 13417 055132 012700 172300 13418 055136 012701 177600 13419 055142 012702 172200 13420 055146 012703 000040 13421 055152 011021 13422 055154 012022 13423 055156 077303	3\$:	LOADCSR END ; OF ON ERROR IF R1 EQ #0 THEN LEAVE LCSRRESTORE END ; OF FOR CSRNO END LCSRRESTORE ; COPY KERNEL MAP TO USER & SUPERVISOR MOV #KIPDRO.RO MOV #UIPDRO.R1 MOV #SIPDRO.R2 MOV #32R3 MOV (R0).(R1)+ MOV (R0)+.(R2)+ SOB R3.3\$

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 429 ROUTINE POWER DOWN AND UP

	13425 13426 055160 13427 055164	005737 001006	002452			RESTORE SUPERVISOR & USER STACK POINTERS TST NOSUPER BNE 13\$
	13426 055160 13427 055164 13428 055166 13429 055170 13430 055176 13431 055200 13432 055202 13433 055204 13434 055212 13435 055214	010006 104417			13\$:	POP RO SUPERVISOR MOV RO,SSP KERNEL POP RO ;ENTER SUPERVISOR MODE ;ENTER KERNEL MODE
	1 364 373	010006 104417				USER MOV RO,USP
	13437 055216 13438 055222 13439 055226 13440 055230	012700 012701 077102	172240 000020		7\$:	MOV #172240,R0 MOV #16.,R1 POP (R0)+ SOB R1,7\$
	13441 13442 055232 13443 055236 13444 055242 13445 055244 13446	012700 012701 077102	177636 000021		8\$:	RESTORE SUPERVISOR PAR'S MOV #172240,RO MOV #16.,R1 POP (R0)+ SOB R1,7\$;RESTORE USER PAR'S & PDR7 MOV #177636,RO MOV #17.,R1 POP (R0)+ SOB R1,8\$:RESTORE POSSIBLE SOFTWARE DISPLAY REGISTER
	13446 13447 055246 13448 055254	013777 013737	002010 002010	125350 000174		MOV \$PATMAR, aDISPLAY MOV \$PATMAR, DISPREG POP CSPNO R5 R4 R3 R2 R1 R0
3	13450 055302 13451 055310 13452 13453 055314	012737 005737 001402	054356	000024		MOV #\$PWRDN, PWRVEC ; SET UP THE POWER DOWN VECTOR TYPE MSG051 ; REPORT THE POWER FAILURE :RESTORE CACHE STATUS TOT CACHEN EEQ 9\$
	13450 055302 13451 055310 13452 13453 055314 13454 055320 13455 055322 13456 055326 13457 055330 13458 055332 13459 055334	000002 000000 000776 000000			9\$: \$ILLUP: \$SAVR6:	FOP CONTRL RTI HALT BR \$ILLUP :: THE POWER UP SEQUENCE WAS STARTED :: BEFORE THE POWER DOWN WAS COMPLETE :: PUT THE SP HERE

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 430 ROUTINE POWER DOWN AND UP

١	13472 055336	WOOPS: SUBTST < <power fail="" relocated="" while="">></power>
ı		*SUBTEST POWER FAIL WHILE RELOCATED
ı	13473 055336	PUSH BANK
ı	13474 055342 005037 002100	CI D DANK
ı	13475 055362	MAP BANK ; MAP SUPERVISOR SPACE (TEST AREA) TO BANK SUPERVISOR SUPERVISOR MODE
۱	13477 055370 013737 060024 055734 13478 055376 013737 060026 055736	MOV FIRST+PWRVEC_WOOPSAV
ı	13478 055376 013737 060026 055736	DMOV ETDETALIONDI ID UNODERVAG LIONDEND-LIONDI ID/2412
ı	13480 055416 012737 055522 060024 13481 055424 012737 000340 060026	MOV WWOOPUP, FIRST+PWRVEC
ı	13481 055424 012737 000340 060026	MOV #340,FIRST+PWRVEC+2 BMOV WOOPUP,FIRST+WOOPUP,WOOPEND-WOOPUP/2
ı	13483 055444 012700 172340	MOV #KIPARO,RO
ı	13483 055444 012700 172340 13484 055450 012701 135704 13485 055454 012702 000010	MOV #FIRST+WOOPEND,R1
ı	13486 055460 012021	15: MOV #8. R2 15: MOV (R0)+, (R1)+
ı	13487 055462 077202	SOB R2.1\$ TST NOSUPER
ı	13488 055464 005737 002452	RNF 2S
ı	13490 055472 013721 172516	MOV MMR3, (R1)+
ı	13490 055472 013721 172516 13491 055476 013721 177576 13492 055502 013721 177574 13493 055506 013721 177572	2\$: MOV MMR2,(R1)+ MOV MMR1,(R1)+
ı	13493 055506 013721 177572	MOV MMRO, (R1)+
١	13493 055506 013721 177572 13494 055512 104417	KERNEL ; ENTER KERNEL MODE
ı	13495 055514 13496 055520 000207	POP BANK RETURN

13499 055522	WOOPUP: SUBTST < <power 0="" bank="" from="" relocation="" to="" up="">> :***********************************</power>
13500 055522 012700 055704 13501 055526 012701 172340 13502 055532 012703 172300 13503 055536 012702 000010	MOV #WOOPEND.RO MOV #KIPARO.R1 MOV #KIPDRO.R3 MOV #8R2
13504 055542 012021 13505 055544 012723 077406	1\$: MOV (R0)+,(R1)+ MOV #77406,(R3)+
13504 055542 012021 13505 055544 012723 077406 13506 055550 077204 13507 055552 005737 002452 13508 055556 001002	SOB R2,1\$ TST NOSUPER BNE 3\$
13509 055560 012037 172516 13510 055564 012037 177576 13511 055570 012037 177574 13512 055574 012037 177572	MOV (RO)+,MMR3 3\$: MOV (RO)+,MMR2
13513 055600 013706 055334 13514 055604 13515 055610 005037 002100 13516 055614 13517 055630	PUSH BANK CLR BANK MAP BANK SUPERVISOR SPACE (TEST AREA) TO BANK SUPERVISOR S
13517 055630 13518 055636 013737 055734 060026 13519 055644 013737 055736 060026 13520 13521	SUPERVISOR ;ENTER SUPERVISOR MODE MOV WOOPSAV,FIRST+PWRVEC MOV WOOPSAV+2,FIRST+PWRVEC+2 ;SIMULATE THE FOLLOWING BLOCK MOV BUT WITH NO STACK ACCESSES PMOV HOOPSAV+4 FIRST+HOOPIN HOOPEND-WOOPIN/2+12
13522 055652 012700 055740 13523 055656 012701 000105 13524 055662 012702 135522 13525 055666 012022 13526 055670 077102	MOV (RO)+, MMRO MOV \$SAVR6, SP PUSH BANK CLR BANK MAP BANK SUPERVISOR MOV WOOPSAV, FIRST+PWRVEC MOV WOOPSAV+2, FIRST+PWRVEC+2 ; SIMULATE THE FOLLOWING BLOCK MOV BUT WITH NO STACK ACCESSES ; BMOV WOOPSAV+4, FIRST+WOOPUP, WOOPEND-WOOPUP/2+12. MOV #WOOPSAV+4, RO MOV #WOOPSAV+4, RO MOV #WOOPEND-WOOPUP/2+12., R1 MOV #FIRST+WOOPUP, R2 2\$: MOV (RO)+, (R2)+ SOB R1, 2\$
13527 13528 055672 104417 13529 055674 13530 055700 000137 054732	POP BANK
13530 055700 000137 054732 13531 055704 000014 13534 055734 000107	JMP SPWRUP WOOPEND:.REPT 12. WOOPSAV:.REPT WOOPEND-WOOPUP/2+12.+2

```
.SBTTL IO SUBROUTINES
                                                                                 .SBITL ROUTINE TYPE
                                                                  **ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A O BYTE.

**THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.

**NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CH
                                                                                               SHULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
SFILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
SFILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                                                   : *NOTE2:
                                                                    *NOTE3:
                                                                    *CALL:
                                                                   *1) USING A TRAP INSTRUCTION
* TYPE MESADR
                                                                                                                            :: MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                                                   : *OR
                                                                                 TYPE
                                                                                 MESADR
                                                                   : *
                                                                  :*
                       105737
100407
010046
017600
                                                                   STYPE:
                                                                                 TSTB
                                                                                               STPFLG
                                                                                                                             :: IS THERE A TERMINAL?
        056152
056156
056160
056166
056170
056172
056174
056176
056202
056204
056210
056212
056224
056226
056230
                                      002354
                                                                                                                             BR IF NO
                                                                                                6$
                                                                                 BMI
                                                                                               RO,-(SP)
a2(SP),RO
                                                                                                                             :: SAVE RO
                                                                   15:
                                                                                 MOV
                                                                                                                            GET ADDRESS OF ASCIZ STRING
PUSH CHARACTER TO BE TYPED ONTO STACK
BR IF IT ISN'T THE TERMINATOR
THE TERMINATOR POP IT OFF THE STACK
                                                                                 MOV
                                      000002
                       017600
112046
001005
005726
012600
062716
000002
122716
001002
112716
122716
001006
005726
3562
3563
3564
3565
3566
3566
                                                                                                (RO)+,-(SP)
                                                                   45:
                                                                                 MOVB
                                                                                 BNE
                                                                                                (SP)+
                                                                                  TST
                                                                                                                             :: RESTORE RO
                                                                                                (SP)+,R0
                                                                                 MOV
                                                                   5$:
                                                                                                #2, (SP)
                                                                                                                             ::ADJUST RETURN PC
                                                                   65:
                                                                                  ADD
                                      000002
                                                                                                                              :RETURN
                                                                                 RTI
                                                                                 CMPB
BNE
MOVB
CMPB
                                                                                                WHT, (SP)
                                                                                                                             BRANCH IF NOT <HT>
                                      000011
                                                                   75:
                                                                                                115
                                                                                                     (SP)
                                                                                                                             REPLACE TAB WITH SPACE
                                      000040
                                                                                                                             :: BRANCH IF NOT < CRLF>
                                                                                                #CRLF, (SP)
                                                                   115:
13572
13573
13574
13575
                                                                                 BNE
                                                                                                                             ::POP <CR><LF> EQUIV
::TYPE A CR AND LF
                                                                                                (SP)+
                                                                                  TYPE
                        002644
105037
000753
004737
123726
001346
013746
                                                                                  SCRLF
                                                                                                                             :: CLEAR CHARACTER COUNT
                                      056464
                                                                                  CLRB
                                                                                                SCHARCNT
 3576
                                                                                                                            GET NEXT CHARACTER
GO TYPE THIS CHARACTER
IS IT TIME FOR FILLER CHARS.?
IF NO GO GET NEXT CHAR.
GET # OF FILLER CHARS. NEEDED
                                                                                  BR
                                                                                  CALL
                                      056300
002636
                                                                                                SFILLC, (SP)+
                                                                                  BNE
                                                                                                $NULL,-(SP)
                                      002352
                                                                                  MOV
                                                                                                                             :: AND THE NULL CHAR.
                        105366
002770
004737
105337
000770
000000
                                                                                                1(SP)
                                                                                                                             ::DOES A NULL NEED TO BE TYPED?
                                                                   105:
                                       000001
                                                                                                                                BR IF NO--GO POP THE NULL OFF OF STACK
                                                                                  BLT
                                       056300
056464
                                                                                                STYPEC
                                                                                                                                GO TYPE A NULL
                                                                                  CALL
                                                                                                                             :: DO NOT COUNT AS A COUNT
                                                                                  DECB
                                                                                                SCHARCNT
                                                                                                                             ::L00P
                                                                                  BR
                                                                                                10$
                                                                                 .WORD 0
                                                                   XOCHAR:
STYPEC:
                                                                                                4(SP),R1
                                       000004
002540
                        116601
005737
001402
                                                                                  MOVB
                                                                                                CACHKN
                                                                                  TST
                                                                                  BEQ
                                                                                  PUSH
                                                                                                 CONTRL
                                                                                  PUSH
                                                                   2$:
                        104424
                                                                                  CACHOFF
                                                                                                                             :TURN CACHE OFF
```

CZMSPAO MS11-L/M/P MEMORY DI	G. MACRO M1113	26-APR-82 09:41	PAGE 434-1
------------------------------	----------------	-----------------	------------

KOUTTNE								
13620	056324 056330 056332 056336 056342 056352 056360 056366 056370 056374	105777 100375	124302		3\$:	TSTB	astps 3s	:: WAIT UNTIL PRINTER IS READY
13621 13623 13624 13625 13626 13627 13630 13631 13631 13633 13634 13635 13636 13642 13643	056332	005037	056276 124264			CLR	XOCHAR astks	:: CHECK FOR XOFF
13624	056342	100032				BPL	NC	::SKIP IF NO CHARACTER
13625	056344	117737	124260 177600 056276	056276 056276 000023		MOVB BIC	#*C177,XOCHAR XOCHAR,#023	STRIP OFF ASCII
13627	056360	023727	056276	000023		CMP	XOCHAR #023	:: WAS IT A CONTROL S?
13628	056366	001020		***************************************		BNE	NC astks_	; ; BRANCH IF NOT
13629	056370	105777	124232		CONTS3:	TSTB	CONTES	;;WAIT FOR CHARACTER
13630	056376	042737 023727 001020 105777 100375 117737	124226	056276		BPL MOVB	CONTS3 a\$TKB,XOCHAR #^C177,XOCHAR	::GET CHARACTER
13632	056404	042737	177600	056276		BIC	#^C177,XOCHAR	::GET CHARACTER ::STRIP OFF ASCII :: IF IT IS A ~Q
13633	056404 056412 056422 056424 056426 056430 056430 056442 056444 056450 056452					IF XOCH	AR EQ #21	;; IF IT IS A ~Q
13634	056422	000402				BR	NC	
13636	056426	000760				BR	CONTS3	
13637	056430		404000			END ; OF MOVB	IF XOCHAR	CAN CHAR TO BE TYPEN INTO DATA DEG
13638	056430	110177	124200 000015	000002	NC:	CMPB	R1, astpb #CR, 2(SP)	:: IS CHARACTER A CARRIAGE RETURN?
13643	056442	001003	000013	000002		BNE	15	;;BRANCH IF NO
13644 13645	056444	105037	056464			CLRB	SCHARCHT	::YESCLEAR CHARACTER COUNT
13645	056450	122766 001003 105037 000406 122766 001402 105227 000000	000012	000002	15:	BR CMPB	SCHARCNT STYPEX #LF,2(SP) STYPEX	::LOAD CHAR TO BE TYPED INTO DATA REG. ::IS CHARACTER A CARRIAGE RETURN? ::BRANCH IF NO ::YESCLEAR CHARACTER COUNT ::EXIT ::IS CHARACTER A LINE FEED? ::BRANCH IF YES ::COUNT THE CHARACTER ::CHARACTER COUNT STORAGE
13646 13647	056460	001402	000012	000002	10.	BEQ	STYPEX	::BRANCH IF YES
13648 13649	056462 056464	105227				INCB	(PC)+	COUNT THE CHARACTER
13649	056464	000000			SCHARCN STYPEX:	POP	O RO	;; CHARACTER COUNT STURAGE
13651	056466 056470	005737	002540		SITPEA:	TST	CACHKN	:IS THERE A CACHE?
13652	056474	001402	002210			BEQ	2\$	BRANCH IF NOT
13653	056476				20.	POP	CONTRL	POP CACHE STATUS
13655	056474 056476 056502 056504	000207			2\$:	POP RETURN	R1	
13656	056506	000207			SUPLIMI	T:;!!!!!	!!!!!!!THIS IS	THE LIMIT ON SUPERVISOR MAPPED TO MUT SPACE

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 449 ERROR DATA SETUP

14334	.SBTTL	ERROR DATA SETUP
14335 14336	USE THIS	IF THIS CONDITION DISCRIBES THE ERROR
14337 14338 14339 14340 14341	PERRO1	TRAP BAD DATA IN RO UNLESS ABORTED THEN BAD DATA IS POINTED TO BY -(R4) GOOD DATA IN R5
14339 14340 14341 14342 14343 14344 14345 14346 14347 14348 14351 14352 14353 14354 14355 14355 14356 14357 14368 14367 14368 14369 14370 14371	PERRO2	TRAP BAD DATA IN R1 UNLESS ABORTED THEN BAD DATA IS POINTED TO BY -(R4) GOOD DATA IN R2
14347 14348 14349 14350	PERRO3	TRAP BAD DATA IS POINTED TO BY -(R1) GOOD DATA IN R4
14351 14352 14353 14354 14355	PERRO4	TRAP BAD DATA IN R4 UNLESS ABORTED THEN BAD DATA IS POINTED TO BY -2(RO) GOOD DATA IN R2
14356 14357 14358 14359 14360	PERRO5	JSR PC BAD DATA IS POINTED TO BY -(RO) GOOD DATA IN R2 RETURN AFTER SETTING UP GOOD, BAD, ADDRESS
14361 14362 14363 14364 14365	PERRO6	JSR PC BAD DATA IS POINTED TO BY -(RO) GOOD DATA IS ZERO RETURN AFTER SETTING UP GOOD, BAD, ADDRESS
14366 14367 14368 14369 14370	PERR07	TRAP BAD DATA IN R2 UNLESS ABORTED THEN BAD DATA IS POINTED TO BY (R1) GOOD DATA IN DATBUF
14372 14373 14374	PERR10	TRAP BAD DATA IN R2 UNLESS ABORTED THEN BAD DATA IS POINTED TO BY 2(R1) GOOD DATA IN DATBUF+2
14375 14376 14377 14378 14379 14380	PERR11	TRAP BYTE TEST BAD DATA IN RIGHT BYTE OF RO UNLESS ABORTED THEN BAD DATA IS POINTED TO BY (R1) GOOD DATA IS A ZERO BYTE
14379 14380 14381 14382 14383 14384 14385 14386 14387 14388 14389	PERR12	TRAP BYTE TEST BAD DATA IN RIGHT BYTE OF RO UNLESS ABORTED THEN BAD DATA IS POINTED TO BY (R1) GOOD DATA IS A BYTE OF ONES
14387		GOOD DATA IS A BYTE OF ONES
14389 14390	PERR13	TRAP BAD DATA IN RO UNLESS ABORTED
	a Yiki in landarin	

CZMSPAO MS11-L/M/P	MEMORY	DIAG.	MACRO M1113	26-APR-82 09:41	PAGE 449-1
--------------------	--------	-------	-------------	-----------------	------------

ERROR DATA SETUP		
14391 14392 14393		GOOD DATA IS ZERO TO BY (R1)
14394 14395 14396 14397 14398 14399 14400	PERR14	TRAP BAD DATA IN RO UNLESS ABORTED THEN BAD DATA IS POINTED TO BY (R1) GOOD DATA IS ONES
14398 14399 14400 14401 14402	PERR15	TRAP BAD DATA IN RO UNLESS ABORTED THEN BAD DATA IS POINTED TO BY (R1) GOOD DATA IN TSTDAT
14403 14404 14405 14406 14407 14408	PERR16	TRAP BAD DATA IN RO UNLESS ABORTED THEN BAD DATA IS POINTED TO BY (R1) GOOD DATA IN TSTDAT+2
14408 14409 14410 14411 14412	PERR17	TRAP BAD DATA IN RO UNLESS ABORTED THEN BAD DATA IS POINTED TO BY (R1) GOOD DATA IN R2
14413 14414 14415 14416 14417	PERR20	TRAP BAD DATA IN RO UNLESS ABORTED THEN BAD DATA IS POINTED TO BY (R1) GOOD DATA IN R3
14418	PERR21	TRAP 7 BIT BYTE TEST BAD DATA IN RIGHT BYTE OF RO UNLESS ABORTED THEN BAD DATA IS POINTED TO BY (R1) GOOD DATA IS A 7 BIT BYTE ON ONES
14420 14421 14422 14423 14424 14425 14426 14427	PERR22	TRAP BAD DATA IN R2 UNLESS ABORTED THEN BAD DATA IS POINTED TO BY (R1) GOOD DATA IN RO
14429 14430 14431 14432 14433	PERR23	TRAP BAD DATA IN RO UNLESS ABORTED THEN BAD DATA IS POINTED TO BY (R1) GOOD DATA IN R4
14434 14435 14436 14437 14438	PERR24	TRAP BAD DATA IN RO UNLESS ABORTED THEN BAD DATA IS POINTED TO BY (R2) GOUD DATA IN R3
14427 14428 14429 14430 14431 14432 14433 14434 14435 14436 14437 14438 14439 14440 14441 14442	PERR25	TRAP BAD DATA POINTED TO BY -(R1) GOOD DATA IN R2 UNLESS LOC V177654 IS SET THEN GOOD DATA IS IN R3
14444 14445 14446 14447	PERR26	TRAP BAD DATA IS DOUBLE WORD POINTED TO BY R1 AND IN LOW 7 BITS OF RO GOOD DATA IS 000000,,100000,,100

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 449-2 ERROR DATA SETUP

14448 14449 14450 14451	PERR27	TRAP BAD DATA IS DOUBLE WORD POINTED TO BY R1 AND IN LOW 7 BITS OF RO GOOD DATA IS 000000,,000000,,077
14450 14451 14452 14453 14454 14455	PERR30	TRAP BAD DATA IS POINTED TO BY -16(SP) GOOD DATA IS IN R1
14456 14457 14458	PERR31	TRAP SPECIAL ECC FAILURE HANDLER
14459 14460 14461	PERR32	TRAP SPECIAL ECC FAILURE HANDLER
14462 14463 14464 14465	PERR33	TRAP SPECIAL ECC FAILURE HANDLER
14466 14467	PERR34	TRAP SPECIAL ECC FAILURE HANDLER
14468 14469 14470	PERR35	TRAP SPECIAL BRANCH GOBBLE FAILURE HANDLER
14471 14472 14473 14474 14475	CALLING BEQ PERRXX	2\$:NO - ERROR, BRANCH FOR CARD :TRAP TO ERROR ROUTINE
14475	;2\$: NEXT	INSTRUCTION ; CONTINUE TESTING

CZMSPAO MS11-L/M/P MEMORY DIAG. ERROR DATA SETUP	MACRO M1113	3 26-APR-82	09:41 PAGE	151
14478 056506 010437 002032 14479 056512 162737 000002 14480 056520 010037 002050 14481 056524 010537 002042 14482 056530 000137 057242	002032 SPE	RO1: MOV SUB MOV MOV JMP PER	R4.ADDRESS #2.ADDRESS R0.BAD R5.GOOD RRAW	
14484 056534 010437 002032 14485 056540 162737 000002 14486 056546 010137 002050 14487 056552 010237 002042 14488 056556 000137 057242	002032 SPE	ERO2: MOV SUB MOV MOV JMP PEI	R4,ADDRESS #2,ADDRESS R1,BAD R2,GOOD RRAW	
14490 056562 010137 002032 14491 056566 162737 000002 14492 056574 010437 002042 14493 056600 016137 177776 14494 056606 000137 057242	002032 002050	ER03: MOV SUB MOV MOV JMP PEI	R1,ADDRESS #2,ADDRESS R4,GOOD -2(R1),BAD RRAW	
14495 14496 056612 010037 002032 14497 056616 162737 000002 14498 056624 010437 002050 14499 056630 010237 002042 14500 056634 000137 057242	002032 SPI	ER04: MOV SUB MOV MOV JMP	RO,ADDRESS #2,ADDRESS R4,BAD R2,GOOD PERRAW	
14501 14502 056640 010237 002042 14503 056644 014037 002050 14504 056650 010037 002032 14505 056654 062700 000002 14506 056660 004737 042614 14507 056664 000207		RRO5: MOV RA05: MOV MOV ADD CALL RETURN	R2,GOOD -(R0),BAD R0,ADDRESS #2,R0 BADSTACK	;RESTORE RO
14508 14509 056666 005037 002042 14510 056672 000764	PE	RR06: CLR BR	GOOD PERAOS	
14511 14512 056674 010137 002032 14513 056700 010237 002050 14514 056704 013737 002240 14515 056712 000137 057242 14516 14517 056716 14518 056730	002042	ER07: MOV MOV MOV JMP	R1,ADDRESS R2,BAD DATBUF,GOOD PERRAW	
1 4/F40 0F/77/	\$P	LET BA	DRESS := R1 + D := R2 OD := DATBUF+ PERRAW	
14519 056734 14520 056742 000137 057242 14521 14522 056746 14523 056752 14524 056756 14525 056762 000137 057314 14526 14527 056766 14528 056772 14529 056776 14530 057004 000137 057314	\$P	LET BA	DRESS := R1 D := R0 OD := #0 PERRAB	
14527 056766 14528 056772 14529 056776 14530 057004 000137 057314	\$P	ER12: LET AD LET BA LET GO JMP	DRESS := R1 D := R0 OD := #377 PERRAB	

CZMSPAO MS11-L/M/P MEMORY DIAG	MACRO M1113	26-APR-82 09:41	PAGE 453
--------------------------------	-------------	-----------------	----------

14533 057010 14534 057014			SPER13:	LET	ADDRESS := BAD := RO GOOD := #0	R1
14535 057020 14536 057024	000137	057242		JMP	GOOD := #0 PERRAW	
14533 057010 14534 057014 14535 057020 14536 057024 14537 14538 057030 14539 057034 14540 057040 14541 057046 14542 14543 057052 14544 057056 14545 057062 14546 057070 14547	000137	057242	SPER14:	LET LET JMP	ADDRESS := BAD := RO GOOD := ONE PERRAW	
14543 057052 14544 057056 14545 057062 14546 057070	000137	057242	SPER15:	LET LET JMP	ADDRESS := BAD := RO GOOD := TST PERRAW	
14548 057074 14549 057100 14550 057104 14551 057112	000453		SPER16:	LET LET LET BR	ADDRESS := BAD := RO GOOD := TST PERRAW	
14549 057100 14550 057104 14551 057112 14552 14553 057114 14554 057120 14555 057124 14556 057130 14557 14558 057132 14559 057136 14560 057142 14561 057146 14562 14563 057150	000444		\$PER17:	LET LET LET BR	ADDRESS := BAD := RO GOOD := R2 PERRAW	R1
14558 057132 14559 057136 14560 057142 14561 057146	000435		\$PER20:	LET LET LET BR		R1
14564 057154 14565 057160 14566 057166	000477		\$PER21:	LET LET LET BR	ADDRESS := BAD := RO GOOD := #17 PERRA7	
14567 14568 057170 14569 057174 14570 057200 14571 057204	000416		\$PER22:	LET	ADDRESS := BAD := R2 GOOD := R0 PERRAW	R1
14573 057206 14574 057212 14575 057216 14576 057222	000407		\$PER23:	LET LET LET BR	ADDRESS := BAD := RO GOOD := R4 PERRAW	R1
14570 057200 14571 057204 14572 14573 057206 14574 057212 14575 057216 14576 057222 14577 14578 057224 14579 057230 14580 057234 14581 057240	000400		SPER24:	LET LET LET BR	ADDRESS := BAD := RO GOOD := R3 PERRAW	R2

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 454 ERROR DATA SETUP
```

14583 057242			PERRAW: SUBTST < <data a="" was="" word="">></data>
			*SUBTEST DATA WAS A WORD
			- 我我我我我我我我我我我我我我我我我我我我我我我我我我我我我我我我我我我我
14584 057242 14585 057246 14586 057260 14587 057272 14588 057276 14589 057304 14590 057306 14591 057310 14592 057312 14593 057312	004737	057476	CALL PERBNK
14585 057246			IF ABORTFLAG IS TRUE THEN SCALL GETDATA IF BADPC EQ #0 THEN SCALL BADSTACK CALL PERXOR
14586 057260	00/777	057/52	THE BADPLEY WU THEN STALL BADSTACK
14587 057274	004/3/	057452	IF ARORTELAG IS FALSE
14589 057304	104011		IF ABORTFLAG IS FALSE ERROR +11
14590 057306	104011		ELSE
14591 057310	104012		ERROR +12
14592 057312			END ; OF IF ABORTFLAG
14593 057312	000002		RTI
14595 057314			PERRAB: SUBTST < <data a="" byte="" was="">></data>
14373 03/314			**************************************
			*SUBTEST DATA WAS A BYTE

14596 057314	004737	057476	CALL PERBNK
14597 057320 14598 057332 14599 057344 14600 057350			IF ABORTFLAG IS TRUE THEN SCALL GETDATA IF BADPC EQ #0 THEN SCALL BADSTACK CALL PERXOR
14598 057332	00/777	057/52	IL RADAL ER MO INEM STAFF BADSTACK
14599 057344	004/3/	057452	IF ARORTELAG IS FALSE
14601 057356	104014		IF ABORTFLAG IS FALSE ERROR +14
14602 057360	101011		ELSE
14602 057360 14603 057362	104015		ERROR +15
14604 057364			END ; OF IF ABORTFLAG
14605 057364	000002		RTI

CZMSPAO MS11-L/M/P MEMORY DIAG.	MACRO M1113	26-APR-82 09:41	PAGE 456
---------------------------------	-------------	-----------------	----------

14608	057366				:*SUBTEST	< <data 7="" a="" bit="" byte="" was="">> DATA WAS A 7 BIT BYTE</data>
14611 14612 14613		004737 004737 104022 000002	057452 057476		IF BADPO CALL CALL ERROR RTI	C EQ #0 THEN \$CALL BADSTACK PERXOR PERBNK +22
14614 14615 14616 14617	057414 057422 057430	000137	054044		SPER26: LET GOOD LET GOOD JMP PERI	D2 := #100000 D3 := #100 RA3
14618 14619 14620 14621	057434 057440 057446 057452	005037 000137	002044 054044		SPER27: CLR LET GOOD JMP PERI	G00D2 D3 := #077 RA3
14623	057452				PERXOR: SUBTST	< <determine &="" bad="" good="" of="" xor="">> DETERMINE XOR OF GOOD & BAD</determine>
14625 14626 14627 14628	057454 057460 057466 057472	013700 013737 074037 000207	002042 002050 002056	002056	PUSH MOV MOV XOR POP RETURN	RO GOOD,RO BAD,BADXOR RO,BADXOR RO

14632 057476				*SUBTES	
14633 14634 057476 14635 057502 14636 057506 14637 057510 14638 057512 14639 057520 14640 057524 14641 057526 14642 057532 14643 057540 14644 057542 14645 057550 14646 057554	013701 006301 006301 052761 105261 001002 105361 126137 101403	002100 000001 002652 002652 002652	002650 002550	125:	; WHILE WE'RE HERE LET'S MARK THE BAD BANK IN THE CONFIGURATION TABLE PUSH RO,R1 MOV BANK,R1 ASL R1 ASL R1 BIS #BITO,CONFIG(R1) INCB CONFIG+2(R1) ; BUMP BANK COUNTER BNE 12\$; NO OVERFLOW - SKIP DECB CONFIG+2(R1) ; SET BACK TO 255. CMPB CONFIG+2(R1),ERRMAX ; IS IT PAST MAX? BLOS 11\$; NO - SKIP SET TOOMANY ; YES POP R1,R0 RETURN
14647 14648 057556 14649 057562 14650 057572 14651 057600 14652 057602 14653 057610 14654 057610 14655 057614 14656 057622 14657	010037 013737 013737 004737 000207	002050 002244 002246 057452	002042 002042	PERECC:	
14658 057624 14658 057634 14660 057636 14661 057636 14662 057650 14663 057654 14664 057664 14665 057666 14667 057676 14668 057700 14670 057710 14671 057712 14672 057712	104023 004737 104037 104043 104044 000002	057556		\$PER31:	IF REALPAT EQ #41 ERROR +23 END IF BADPC EQ #0 THEN \$CALL BADSTACK CALL PERECC IF REALPAT EQ #11 ERROR +37 END :OF IF REALPAT IF REALPAT EQ #15 ERROR +43 END :OF IF REALPAT IF REALPAT EQ #16 ERROR +44 END :OF IF REALPAT SET HEADER RTI

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 460 LOG ERROR ON BAD BANK

14676 057722 14677 057734 14678 057740 14679 057744 14680 057750 14681 057756 14682 057760 14683 057766	010137 010037 010237 104040 000002	002032 002050 002042		\$PER32:	IF BADPO MOV MOV SET ERROR SET RTI	R1,ADDRESS R0.BAD R2,GOOD HEADER +40 HEADER	BADSTACK	
14678 057740 14679 057744 14680 057750 14681 057756 14682 057760 14683 057766 14684 14685 057770 14686 060002 14687 060006 14688 060012 14689 060016 14690 060024 14691 060030 14692 060036 14693 060040 14694 060046 14695 14696 060050 14697 060062 14698 060072 14699 060074 14700 060076 14701 060100	010137 010037 105037 012737 004737 104041 000002	002032 002050 002051 000377 057452	002042	\$PER33:	IF BADP MOV CLRB MOV CALL SET ERROR SET RTI	C EQ #0 THEN \$CALL R1,ADDRESS R0,BAD BAD+1 #377,GOOD PERXOR HEADER +41 HEADER	BADSTACK	
14696 060050 14697 060062 14698 060072 14699 060074 14700 060076 14701 060100 14702 060100	104016 104001 000002			\$PER34:	ERROR END ; OF RTI	IF #BIT15!BIT4	EXPECTED SBE SO DE	BE MUST HAVE GOTTEN SET
14704 14705 060102 14706 060106 14707 060112 14708 060120 14709 060126 14710 060130 14711 060134	004737 004737 013737 012737 104047 062706 000207	057476 042614 002030 000012 000004	002050 002042	\$PER35:	DURING CALL CALL MOV MOV ERROR ADD RETURN	BRANCH GOBBLE THE PERBNK BADSTACK BADPSW.BAD #12,GOOD +47 #4,SP	:FIX STACK FROM TRA	
14712 14713 060136 14714 060142 14715 060146 14716 060154 14717 060156 14718 060164	010037 010137 104023 000002	002042 002050		\$PER36:	MOV MOV SET ERROR SET RTI	RO.GOOD R1.BAD HEADER +23 HEADER		

```
.SBTTL ROUTINE SCOPE HANDLER
                                                                                      ***************
                                                                                     **THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT **AND LOAD THE DISPLAY DATA INTO THE DISPLAY REGISTER **THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE: **SW14=1 LOOP ON TEST
                                                                                     : *SW9=1 LOOP ON ERROR
14728

14729

14730 060166

14731 060172

14732 060174

14733 060200

14734 060204

14736 060206

14737 060212

14738 060214

14739 060220

14740 060220

14741 060224

14742 060224

14744 060244

14744 060244

14745 060246

14755 060264

14756 060270
                                                                                     : *CALL
                                                                                                                                          ::SCOPE=IOT
                                                                                                       SCOPE
                                                                                                                        SDEVCT
                                                                                                                                                            :TELL APT WE ARE ALIVE
                              005237 065650
                                                                                     SSCOPE: INC
                                                                                                       IF RESULT IS LT
                                                                                                           CLR
INCB
                                                                                                                         $DEVCT
                               005037
105237
                                                                                                                         SUNIT
                                                                                                      END OF IF RESULT
                                                                                                                                                             :: TEST FOR CHANGE IN SOFT-SWR
                               104410
005737
001402
004737
                                                 006204
                                                                                                                         NOTRCE
                                                                                                       BEQ
                                                                                                                                                             :TRACE
                                                                                                                         CONTT
                                                                                                       CALL
                                                 064144
                                                                                     NOTRCE:
                                                                                                                        CPERRF
SKJ
BRANCH IF NOT
BRANCH IF NOT
GET CONTENTS OF ERROR REGISTER
IS THE POWER FAIL MONITOR BIT SET?
SKJ
H177
REPORT IF NOT
REPORT IF SO
                               005737
001410
013737
032737
001401
104177
                                                 061322
                                                                                                       TST
                                                                                                                                                                                                                                                       :R-C
                                                                                                       BEQ
                                                                                                                                                                                                                                                       :R-C
                                                                  061320
061320
                                                                                                       MOV
                                                 177766
                                                 000001
                                                                                                       BIT
                                                                                                       BEQ
                                                                                                       ERROR
                                                                                                       IF STOPOK IS TRUE AND #SW8 SET.IN OSWR
                                                                                     SKJ:
                                                                                                           CLR STOPOK
                               005037
000137
14756 060270
14756 060274
14758 060274
14758 060274
14759 060302
14760 060304
14761 060304
                                                                                                       JMP EXIT
END : OF IF STOPOK
IF NOSCOPE IS TRUE
                               000002
                                                                                                           RTI
                                                                                     END : UF IF NOSCOPE

18: IF #SW14 SET. IN ASWR THEN GOTO SOVER
: #####START OF CODE FOR THE XOR TESTER####
                                                                                                                       CODE FOR THE XOR TESTER#####

;:IF RUNNING ON THE 'XOR' TESTER CHANGE
;:THIS INSTRUCTION TO A 'NOP' (NOP=240)

ERRVEC.-(SP) ;:SAVE THE CONTENTS OF THE ERROR VECTOR

#1$,ERRVEC ;:SET FOR TIMEOUT

177060 ;:TIME OUT ON XOR?

(SP)+,ERRVEC ;:RESTORE THE ERROR VECTOR

$VLAD ;:GO TO THE NEXT TEST

#4,SP ;:FIX STACK FROM TRAP

#1,PROTYP ;IS THIS AN 11/44?

6$ ;BRANCH IF NOT

CPUERR ;RESET CPU ERROR REGISTER

(SP)+,ERRVEC ;:RESTORE THE ERROR VECTOR

4$ ;:LOOP ON THE PRESENT TEST
                                                                                     SXTSTR: BR
             060314
                               000425
14764
14765
14766
14767
14768
14769
14770
14771
14772
14773
14774
14775
14776
14777
14778
14778
                               013746
012737
005737
012637
000430
062706
022737
001002
005037
012637
000407
                                                 000004
060342
177060
             060316
                                                                                                       MOV
                                                                   000004
                                                                                                       MOV
             060330
                                                                                                        TST
             060334
060340
060346
060346
060354
060356
060362
060366
                                                  000004
                                                                                                       MOV
                                                                                                       BR
                                                                                                        ADD
                                                  000004
                                                                                                        CMP
BNE
                                                  000001
                                                                   003752
                                                                                                        CLR
                                                 177766 000004
                                                                                                        MOV
                                                                                     6$:
                                                                                                                         4$ CODE FOR THE XOR TESTERMANN
             060370
060374
060376
060404
060406
060414
060416
060422
060426
                               105737
001412
032777
001404
013737
000410
105037
011637
                                                                                                                         SERFLG
SSVLAD
                                                                                                                                                             ;; HAS AN ERROR OCCURRED?
                                                                                                        TSTB
                                                 002012
                                                                                                                                                              ::BR IF NO
                                                                                                        BEQ
                                                                                                                                                              ::LOOP ON ERROR?
                                                                                                                          #SW9, aswR
                                                                                                        BIT
                                                  001000 122216
                                                                                                                                                             .. BR IF NO
                                                                                                        BEQ
                                                                                                                          SLPERR, SLPADR
                                                                                                                                                              ::SET LOOP ADDRESS TO LAST SCOPE
                                                                                                        MOV
                                                  002610
                                                                   002606
  14781
                                                                                                        BR
                                                                                                                          SOVER
                                                                                                                          SERFLG
(SP), SLPADR
(SP), SLPERR
                                                                                                                                                             ::ZERO THE ERROR FLAG
::SAVE SCOPE LOOP ADDRESS
::SAVE ERROR LOOP ADDRESS
                                                                                     $SVLAD: MOV
```

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 462-1 ROUTINE SCOPE HANDLER

SOVER: CALL MOV RTI

SESCAPE GETDIS SLPADR, (SP)

:: CLEAR THE ESCAPE FROM ERROR ADDRESS

::FUDGE RETURN ADDRESS ::FIXES PS

CZMSPAO MS11-L/M/P MEMORY DIAG	MACRO M1113	26-APR-82 09:41	PAGE 463
--------------------------------	-------------	-----------------	----------

KOOLINE	SCOPE H	ANDLEK						
14791	060450				GETDIS:	SUBTST	< <subr display="">></subr>	
					*SUBTE	ST	SUBR DISPLAY	*****
14792 14793	060450 060456	113737 113737	002100 002274	002011 002010		MOVB MOVB PUSH	BANK, \$BANK REALPAT, \$PATMAR RO	
14795	060464	005737 001403 052737	002124			TST BEQ BIS	RLFLAG	;ARE WE RELOCATED?
14797	060472	052737	100000	002010	15:	BIS	WBIT15, \$PATMAR	:NO - SKIP :YES - SET MSB
14802 14803	060502 060510 060516	013777 013737	002010 002010	122114 000174		MOV MOV POP	SPATMAR, adisplay SPATMAR, dispred RO	SOFTWARE DISPLAY REGISTER
14804	060520	000207				RETURN		

ROUTINE	ERROR HA	WDLER							
14808 14809 14810 14811 14812 14813 14814 14815 14816 14817					, -CALL	ROUTINE OF THE ERROR OF TO SERIULITCH OP TO		ANDLER ***********************************	******
14819 14820					**			-ENI AND M-ERROR TIEN MONDER	
14821 14822	060522	105037	061316		SERROR:	.ENABL	TREAVE	;R-C	
14823 14824	060526 060534	104410				IF NOER	ROR IS FALSE	:: TEST FOR CHANGE IN SOFT-SWR	
14825 14826	060536 060536	105237	002012		BACK: 1\$:	INCB		:: SET THE ERROR FLAG	
14827 14828	060542 060544	105237 001775 004737 013737	060450 002010			CALL	1\$ GETDIS	SET THE ERROR FLAG DON'T LET THE FLAG GO TO ZERO SETUP DISPLAY STUFF	
14821 14822 14823 14824 14825 14826 14827 14829 14830 14831 14832	060522 060526 060534 060536 060536 060542 060544 060550 060556	013737 032777 001404	002010 002000	065644 122036		MOV	SPAIMAR, SIESIN	POR API	
14831 14832	060564 060566	001404				TYPE	\$BELL	::BELL ON ERROR? ::NO - SKIP ::RING BELL :CONTROL Z	
14834	060576	005237	002614		2\$:	INC	MSG014 SERTTL	CONTROL Z	
14835 14836	060602	012737	077777	002614		IF RE MOV	2\$ \$BELL MSG014 \$ERTTL SULT IS MI #77777,\$ERTTL		
14836 14837 14838 14839	060612 060612		00001/					::GET ADDRESS OF ERROR INSTRUCTION	
14839 14840 14841	060612 060616	011637 162737 010637 016637 117737	002016 000002 002022 000002 121154	002016		SUB	IF NOERROR (SP), ERRPC #2, ERRPC SP, ERRSP 2(SP), ERRPSW	THE ADDRESS OF ERROR INSTRUCTION	
14842	060630	010637	000005	002026 002013		MOV	2(SP) ERRPSW BERRPC, \$1TEMB	STRIP AND SAVE THE ERROR ITEM CODE	
14843	060636					MOVB CMPB	BENKLC' 91 I FUD		:R-C
14846	060652	122737 001431 105737 001024 005737 001423 013737 032737 001414 042737 112737 112737 000402 105037	061316	002013		BEQ TSTB	#177,\$ITEMB 1001\$ IRSAVE	:IS THIS THE POWER FAIL CALL? :BRANCH IF SO :2ND ERROR CALL? :BRANCH IF SO :IS THERE A CPU ERROR REGISTER? :BRANCH IF NOT :SAVE CONTENTS :POWER MONITOR BIT SET? :BRANCH IF NOT :CLEAR THE BIT :MAKE IBSAVE NON-ZERO FOR DUAL CALL :SET \$ITEMB TO POWER FAIL POINTER	;R-C
14848	060660	001024	061310			BNE	1000\$	BRANCH IF SO	R-C
14850	060666	001423	177766	061320		BEQ MOV	IBSAVE 1000\$ CPERRF 1001\$ 177766, CPSAVE WBITO, CPSAVE 1001\$ WBITO, 177766 WSITEMB, IBSAVE #177, SITEMB 1001\$ IBSAVE	BRANCH IF NOT	R-C R-C R-C R-C R-C
14852	060676	032737	000001	061320 061320		BIT	#BITO, CPSAVE	POWER MONITOR BIT SET?	R-C
14854	060706	042737	000001 002013 000177	177766 061316 002013		BIC	WBITO 177766	CLEAR THE BIT MAKE IBSAVE NON-ZERO FOR DUAL CALL	R-C
14856	060722	112737	000177	002013		MOVB BR	#177,\$ITEMB	SET SITEMB TO POWER FAIL POINTER	R-C R-C
14858	060732	105037	061316		1000\$: 1001\$:	CLRB	IBSAVE		; R-C
14860	060636 060644 060652 060654 060660 060662 060666 060670 060676 060704 060714 060732 060736 060736 060736 060736 060736					IF NOER	ROR IS FALSE DPC NE #0 BADPC,ERRPC B #2,ERRPC BADSP,ERRSP		
14862	060752	013737 162737 013737	002020 000002 002024	002016 002016 002022		MOV	BADPC ERRPC		
14864	060766	013737	002024	002022		MOV	BADSP, ERRSP		

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 465-1 ROUTINE ERROR HANDLER
   14865 060774
14866 061002
14867 061006
14868 061006
14869 061014
14870 061020
14871 061030
14872 061032
14873 061032
14874 061050
14875 061052
14876 061052
14877 061052
14878 061056
14879 061064
14880 061070
14881 061072
                                                                                                        MOV BADPSW, ERRPSW
CLR BADPC
                                               002030
                                                               002026
                               013737
005037
                                                                                                   END : IF
MOV ERRPC, SFATAL : FOR APT
CALL PERBNK :: LOG ERROR ON BAN
IF #SW13 SET. IN aSWR
BR 3$
END : OF IF #SW13
IF #SW5 SET. IN aSWR AND TOOMANY IS TRUE
                                                                065642
                                013737
004737
                                                                                                                                               :: LOG ERROR ON BANK
                                000420
                                                                                                        GOTO 3$
                                                                                                    END : OF IF #SW5
                                                                                                END OF IF NOERROR
                                                                                                                                                :: GO TO USER ERROR ROUTINE
                                                                                                CALL
                                004737 061324
                                                                                                                                                SHOULD WE RETURN TO XXDP MONITOR ???
                                                                                                IF MONFLG IS TRUE
MOV SAVMON, SP
                                013706
000207
                                                002270
                                                                                                                                                :: GO TO MONITOR
                                                                                                      RTS PC
                                                                                                END
```

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 466 ROUTINE ERROR HANDLER

14883 14884 14885 14886 14887	061072 061100 061104 061106 061110	005777 100002 000000 104410	121516		3\$: \$HALT: 7\$:	IF NOERROR IS FALSE TST aswr ::HALT ON ERROR BPL 7\$::SKIP IF CONTINUE HALT ::HALT ON ERROR! CKSWR ::TEST FOR CHANGE IN SOFT-SWR IF NOSCOPE IS FALSE AND #SW9 SET.IN aswr MOV \$LPERR,(SP) ::FUDGE RETURN FOR LOOPING END :OF IF NOSCOPE TST \$ESCAPE ::CHECK FOR AN ESCAPE ADDRESS BEQ 9\$::BR IF NONE MOV \$ESCAPE,(SP) ::FUDGE RETURN ADDRESS FOR ESCAPE TE DETELAG IS FALSE
14889	061130	013716	002610			MOV \$LPERR, (SP) ;; FUDGE RETURN FOR LOOPING
14890 14891 14892	061134	005737 001402 013716	002356			TST SESCAPE :: CHECK FOR AN ESCAPE ADDRESS REQ 98 :: BR IF NONE
14893	061142	013716	002356			MOV SESCAPE (SP) ;; FUDGE RETURN ADDRESS FOR ESCAPE
14894 14895	061146 061154	022737	000001	003752	95:	CMP #1, PROTYP ; IS THIS AN 11/44?
14896 14897	061162 061164	022737 001002 005037	177766			CLR CPUERR
14898	061170		000001	065640		115: IF ACTFLAG IS TRUE OR APTFLAG IS TRUE OR FATALS IS TRUE MOV #1.SMSGTY :FOR APT
14900 14901 14902	061220 061224 061224	012737 000137	000001 047632	005040		IMP EXIT END : OF IF ACTFLAG IF XXPPCHAIN IS TRUE AND SERTTL HI #20 FOR THE PROPERTY OF THE
14903 14904 14905 14906 14907 14908	061242 061246 061252 061256 061262 061262	013700 005037 000137	000042 000042 015212			HALT CKSWR IF NOSCOPE IS FALSE AND #SW9 SET.IN ASWR MOV \$LPERR.(SP) END :OF IF NOSCOPE TST \$ESCAPE ::CHECK FOR AN ESCAPE ADDRESS BEQ 9\$ MOV \$ESCAPE.(SP) IF DETFLAG IS FALSE CMP #1, PROTYP BNE 11\$ CLR CPUERR 11\$: IF ACTFLAG IS TRUE OR APTFLAG IS TRUE OR FATAL\$ IS TRUE MOV #1, \$MSGTY JMP EXIT END :OF IF ACTFLAG IF XXDPCHAIN IS TRUE AND \$ERTTL HI #20 TYPE MSGO66 MOV 42, RO CLR 42 JMP \$ZAP42 END :OF IF XXDPCHAIN END :OF IF DETFLAG ELSE SET HEADER END :OF IF NOERROR
14884 14885 14886 14887 14888 14889 14891 14893 14894 14895 14896 14897 14903 14904 14905 14908 14908 14908 14918 14918 14918 14918 14918 14918 14918 14918	061072 061100 061106 061110 061112 061130 061134 061134 061134 061134 061146 061146 061162 061224 061224 061224 061224 061224 061224 061224 061224 061224 061232 061262 061262 061262 061262 061306 061310 061316 061316	105737 001402 000137 000002 000000 000000 000000	061316 060536		10\$:	END ; OF IF XXDPCHAIN END ; OF IF DETFLAG ELSE SET HEADER END ; OF IF NOERROR CLEAR TOOMANY, NOERROR TSTB IBSAVE ; POWER FAIL ERROR CALL? ; R-C BEQ 213\$ JMP BACK ; RETURN ; R-C RTI ; RETURN ; R-C WORD 0 ; R-C WORD 0 ; R-C JSABL LSB

14923 14924	.SBTTL	ROUTINE ERROR	MESSAGE TYPEOUT
14925 14926 14927 14928	*THIS ROUTINE *ERROR IS TO E *AND REPORTS	USES THE "ITEM BE REPORTED. IT THE APPROPRIATE	CONTROL BYTE' (\$ITEMB) TO DETERMINE WHICH THEN OBTAINS, FROM THE 'ERROR TABLE' (\$ERRTB), INFORMATION CONCERNING THE ERROR.
14929 14930 061324 104415 14931 061326 14932 061332 005000 14933 061334 153700 002013 14934 061340 001004	SERRTYP: SAVREG TYPE CLR BISB BNE	SCRLF RO	:: 'CARRIAGE RETURN' & 'LINE FEED'' :: PICKUP THE ITEM INDEX :: IF ITEM NUMBER IS ZERO, JUST
14930 061324 104415 14931 061326 14932 061332 005000 14933 061334 153700 002013 14934 061340 001004 14935 14936 061342 14937 061350 000511 14938 061352 122700 000177 14939 061350 001003 14940 061360 012700 061634 14941 061364 000406 14942 061366 005300 14943 061370 006300 14944 061372 006300 14945 061374 006300 14946 061376 062700 066262 14947 061402 012037 061440 14948 061406 001417 14949 061410 005737 002424 14950 061414 001003 14951 061416 005737 002576 14952 061422 100011 14953 061424 005737 002062 14954 061430 001402 14955 061432 14956 061436 14957 061440 000000 14958 061442	1\$: TYPOCT BR CMPB BNE MOV	#177,R0 100\$ #PFECWS,R0	;; IF ITEM NUMBER IS ZERO, JUST ;; TYPE THE PC OF THE ERROR ADDRESS> ;; GET OUT ; POWER MONITOR CALL? ; BRANCH IF NOT ; R-C ; MOV ADDRESS OF PFE BIT ERROR TO RO ; R-C
14941 061364 000406 14942 061366 005300 14943 061370 006300 14944 061372 006300 14945 061374 006300 14946 061376 062700 066262 14947 061402 012037 061440	100\$: BR DEC ASL ASL ASL	RO RO	:: ADJUST THE INDEX SO THAT IT WILL ;R-C :: WORK FOR THE ERROR TABLE
14946 061376 062700 066262 14947 061402 012037 061440 14948 061406 001417 14949 061410 005737 002424 14950 061414 001003	110\$: MOV BEQ TST BNE	#\$ERRTB,R0 (R0)+,3\$ 4\$ NOERROR 12\$;; FORM TABLE POINTER ;; PICKUP 'ERROR MESSAGE' POINTER ;; SKIP TYPEOUT IF NO POINTER ;IS THIS REALLY AN ERROR? ;YES - SKIP ;TYPE HEADER?
14950 061414 001003 14951 061416 005737 002576 14952 061422 100011 14953 061424 005737 002062 14954 061430 001402 14955 061432	12\$: TST BPL TST BEQ TYPE 2\$: TYPE	HEADER 4\$ FATAL\$ 2\$ MSG067	WAS IT A FATAL ERROR?
4/050 0/4/// 043077 0/4/73	2\$: TYPE 3\$: .WORD TYPE 4\$: MOV BEQ TST	0 \$CRLF (R0)+,5\$ 6\$ NOERROR	* · DITEID DATA MPADER PUINTER
14960 061452 001412 14961 061454 005737 002424 14962 061460 001003 14963 061462 005737 002576 14964 061466 100004 14965 061470	BNE TST BPL 13\$: TYPE	13\$ HEADER 6\$	SKIP TYPEOUT IF O IS THIS REALLY AN ERROR? YES - SKIP TYPE HEADER? NO - SKIP TYPE THE 'DATA HEADER' TYPE THE 'DATA HEADER'
14960 061452 001412 14961 061454 005737 002424 14962 061460 001003 14963 061462 005737 002576 14964 061466 100004 14965 061470 14966 061472 000000 14967 061474 14968 061500 012001 14969 061502 001427 14970 061504 012002	5\$: .WORD TYPE 6\$: MOV BEQ MOV	0 \$CRLF (R0)+,R1 10\$ (R0)+,R2	NO - SKIP TYPE THE 'DATA HEADER' 'DATA HEADER' POINTER GOES HERE 'CARRIAGE RETURN' & 'LINE FEED' PICKUP 'DATA TABLE' POINTER BR IF NO DATA TO BE TYPED PICKUP 'DATA FORMAT' POINTER

ZMSPAO DUT1NE	MS11-L/ ERROR M	M/P MEMOR	RY DIAG.	MACRO I	M1113 26	-APR-82	09:41	PAGE	470		
14973 14974 14975 14976 14977 14978 14979 14980 14981 14982 14983 14984	061506 061510 061512 061516 061520 061522 061524 061526 061530 061532 061534 061536 061540 061550 061552 061554	112203 006303 004773 000412 061744 061754 061764 062034 062074 062106 062120 062164 062172 062252 062701 005711 001403	061520		7\$: 8\$:	MOVB ASL CALL BR TAG70\$ TAG71\$ TAG72\$ TAG73\$ TAG75\$ TAG76\$ TAG77\$ TAG77\$	(R2)+, R3 a8\$(R3 9\$;MAKE IT A WORD	ADDRESS
14980 14981 14982 14983 14984 14985 14986 14991 14993 14994 14995 14996	061542 061544 061550 061552 061554 061560	062252 062701 005711 001403	000002		9\$:	TAG/98 ADD TST BEQ TYPE BR	#2,R1 (R1) 10\$ MSG018 7\$	3		:UPDATE DATA TA ::IS THERE ANOT ::BR IF NO :TYPE 2 SPACES ::LOOP	BLE POINTER HER NUMBER?
14996 14997 14998 14999 15000 15001	061562 061566	005737 001402 005237 104416	002106		10\$:	TST BEQ	MUT 11\$:15 THERE A MEM	ORY UNDER TEST
14999 15000	061570 061574	104416	002576		115:	INC RESREG	HEADER		D AND	YES - BUMP HEA	
15001 15002 15003 15004 15005	061562 061566 061570 061574 061576 061622 061626 061632	004737	062274			IF #SW7 CALL END : OF TYPE RETURN .EVEN	DETAIL IF #SI MSG10	47	K AND	CONTROL Z	E AND NOERROR IS FALSE
15006		061644 061740	061700	061730	PFECWS	.EVEN	PFECE	M,PFE	CDH,P	FECDT, PFECDF	; R-C ; R-C
15008	061634 061642 061644 061647 061652 061663 061663 061666 061671 061674	120 105 115 111 122 111	117 122 117 124 040 124 117 104	127 040 116 117 102 040 125 040	PFECEM	: .ASCIZ	'POWE	R MON	ITOR	BIT FOUND SET"	;R-C
15009	061700 061703 061706 061711 061714 061717 061722 061725	106 116 123 000 124 124 040 122 120 040 125 122	105 116 040 122 103 103 105 000	123 117 105 040 040 120	,	: .ASCIZ	"TEST	NO E	RR PC	CPUERR''	;R-C
15010 15011			002016	061320	PFECDT	.EVEN	STEST	N,ERR	PC.CP	SAVE,0	R-C
15012	061730 061736 061740 061743	000000	000	000	PFECDF	: .BYTE	0.0.0	.0			;R-C
15013	001743	000									

ZMSPAO MS11 OUTINE ERRO	-L/M/P MEMO R MESSAGE T	RY DIAG. MAC	RO M1113 26	-APR-82	09:41	PAGE 472	
			******	******			**************
15016 15017			*** OC	TAL ***			
15018			*****	TYPOCT	2/01)	******	;;TYPE AN OCTAL NUMBER
15020 0617	52 000207		IAG/US:	RETURN	WINI		,, TIPE AN OCIAL NORDER
15021	JE 000E01						
15022			· +++ DE	CIMAL ++	*		*********
15024			*****	*****	*****	******	;;TYPE A DECIMAL NUMBER
15025 0617	54		TAG71\$:	TYPDEC	a(R1)		;;TYPE A DECIMAL NUMBER
15018 15019 0617 15020 0617 15021 15022 15023 15024 15025 0617 15026 0617 15027 15030 0617 15031 0617 15032 0617 15033 0617 15033 0617 15034 0620 15036 0620 15036 0620 15039 0620 15039 0620 15040 0620 15041 15042 15043 15044 15045 0620 15047 0620	62 000207			RETURN			
15028			;*****	******	*****	******	*********
15029			;*** IN	TERLEAVE	***	******	*********
15031 0617	64		TAG72\$:	PUSH	R1,R5		
15032 0617	70 013701 74 070127	002100 000004		MOV	BANK,R	1	
15034 0620	00	000004		MUL	NOTAB		:INDICATE NO TABLE TO BE PRINTED - NO
15035 0620	06			TYPE	MSG014 TCFIG1		
15036 0620	12 004737 16 005037	041726 002366		CALL	NOTAB		
15038 0620	22	002300		POP	R5,R1		
15039 0620	26			TYPE	MSG014		;1 SPACE
15040 0620	32 000207			RETURN			
15042			;*****	*****	*****	******	******
15043			*** CS	****	******	******	*********
15045 0620	34		TAG73\$:	PUSH	R1,R5		
15046 0620	40 013701 44 070127	002100 000004		MOV	BANK,R #4,R1 NOTAB	11	
15048 0620	50			SET	NOTAB		
15049 0620	56 004737 62 005037	042222 002366		CALL CLR POP	TCF IGS	•	
15050 0620	62 005037	002300		POP	NOTAB R5,R1		
15052 0620	72 000207			RETURN			
15053				*******	******	******	************
15055			: *** PA	TTERN **	**		
15056			******	******	******	*******	***********************
15058 0621	04 000207		IAG/43:	RETURN	KEALPA	II, CITE	(0-77)>,2,2
15047 0620 15048 0620 15049 0620 15050 0620 15051 0620 15053 15054 15055 15056 15056 15057 0620 15061 15061 15062 15063 0621 15064 0621							
15060			*** BA	NK +++	******	******	
15062			******	******	******	******	**********
15063 0621	06		TAG75\$:	TYPOCS	BANK, <	CTYPE (0-	-167)>,3
15004 002	16 000207			RETURN			

```
************
15066
15067
15068
15069
                                               *** MTYPE ***
                                                                                      **********
                                               *************
15068
15069 062120
15070 062124
15071 062130
15072 062134
15073 062142
15074 062146
15075 062152
15076 062156
15077 062162
15078
15079
15080
                                               TAG76$: PUSH
                                                                  R1.R5
                                                                  BANK,R1
                          002100
                                                        MOV
                 070127
                                                        MUL
                                                                  NOTAB
                                                        SET
                                                                  MSG019
                                                        TYPE
                 004737
005037
                          042066
                                                                  TCF1G2
                                                        CALL
                                                        CLR
                                                                  NOTAB
                                                                  R5,R1
                                                        RETURN
                 000207
                                               *** UNKNOWN DATA ***
15081
15082 062164
15083 062170
                                               *****
                                               TAG77$: TYPE
                                                                  MSG061
                                                        RETURN .
                000207
15084
15085
15086
15087
                                               *************************
                                               :*** PHYSICAL ADDRESS ***
15087
15088 062172
15089 062200
15090 062206
15091 062214
15092 062220
15093 062222
15094 062230
15095 062234
15096 062240
15097 062244
15098 062250
15099
15100
                013737
162737
013737
006237
103003
052737
012746
004737
062706
                           002032
060000
002100
002040
                                               TAG78$: MOV
                                                                  ADDRESS, PHYADD
                                                                  #FIRST, PHYADD
                                                        SUB
                                                        MOV
                                                                  BANK, PHYADD+2
                                                                  PHYADD+2
                                                        ASR
                                                        BCC
                           100000
002036
065520
000002
                                                                  #BIT15, PHYADD
                                    002036
                                                        BIS
                                                                                                POINTER TO DOUBLE WORD ON STACK
                                                                  #PHYADD, - (SP)
                                               15:
                                                        MOV
                                                                                                CALL DOUBLE PRECISION CONVERSION ROUTINE
                                                                  $DB20
                                                        CALL
                                                                  #2,SP
$0CT8
                                                                                                :FIX STACK
                                                         ADD
                                                         TYPE
                                                         RETURN
                 000207
                                               15101
                                               :*** OCTAL BYTE ***
15102
15103 062252
15104 062256
15105 062266
15106 062272
                                                TYPE MSG018 ;2 SPACES
TYPOCS a(R1), <TYPE BYTE>,3,2
TYPE MSG014 ;SPACE
                                               TAG79$: TYPE
                                                         RETURN
                 000207
```

15150 062274	DETAIL: SUBTST < <subr detailed="" error="" report="">> :*SUBTEST SUBR DETAILED ERROR REPORT</subr>
15151 062274 005237 002216 15152 062300 022737 000003 002216 15153 062306 101473 15154 062310 022737 000002 002216 15155 062316 001435 15156 062320 15157 062330 15158 062336 005037 002106 15159 062342 010037 002176	BLOS 45 CMP #2,DETFLAG BEQ 25 PUSH HEADER,MUT
15151 062274 005237 002216 15152 062300 022737 000003 002216 15153 062306 101473 15154 062310 022737 000002 002216 15155 062316 001435 15156 062320 15157 062330 15158 062336 005037 002106 15159 062342 010037 002176 15160 062346 012700 002200 15161 062352 010120 15162 062354 010220 15163 062356 010320 15164 062360 010420 15165 062364 013720 002022 15167 062370 013720 002026 15168 062374 013700 002176	SET HEADER CLR MUT MOV RO,DETRO MOV #DETR1,RO MOV R1,(RO)+ MOV R2,(RO)+ MOV R3,(RO)+
15164 062360 010420 15165 062362 010520 15166 062364 013720 002022 15167 062370 013720 002026 15168 062374 013700 002176 15169 062400	MOV R4,(R0)+ MOV R5,(R0)+ MOV ERRSP,(R0)+ MOV ERRPSW,(R0)+ MOV DETRO,RO SET NOERROR
15170 062406 104013 15171 062410 000423 15172 062412 15173 062422 15174 062430 005037 002106	ERROR +13 BR 1\$ 2\$: PUSH HEADER, MUT SET HEADER CLR MUT
15170 062406 104013 15171 062410 000423 15172 062412 15173 062422 15174 062430 005037 002106 15175 062434 15176 062442 104031 15177 062444 022737 000001 003752 15178 062452 001002 15179 062454 005037 177766 15180 062460	BNE 1\$ CLR CPUERR 1\$: POP MUT_HEADER
15181 15182 062470 004737 062274 15183 062474 000207	WARNING RECURSIVE CALL DETAIL RETURN

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 478
SUBR DETAILED ERROR REPORT
                                                                                   SIMULATE CONTROL "T"
   15186
15187 062476 004737 064144
15188
15189
                                                                                                                                        :DISPLAY 'DISPLAY' INFO
                                                                45:
                                                                                  TYPE CONTENTS OF ALL CSR'S
CSR,CSRNO,R1
TYPE MSG058
TYPE $CRLF
   15189
15190 062502
15191 062514
15192 062520
15193 062524
15194 062530
15195 062530
15196 062534
15197 062536
15198 062540
15199 062542
15200 062550
15201 062554
15202 062554
15203 062560
15204 062576
15205 062576
                                                                                                 TOTCSRS,R1
                                                                                   MOV
                           013701 002222
                                                                                   BEGIN DUMPCSRLOOP
                                                                                      FOR CSRNO := #0 TO #36 BY #2
                                                                                                              R1
                                                                                          ASL
                           006301
                                                                                          ON. ERROR
                                                                                             READCSR
                           104426
                                                                                             TYPOCT
                                                                                                               MSG018
                                                                                                                                           :2 SPACES
                                                                                             TYPE
                                                                                          END : OF ON ERROR
IF R1 EQ #0 THEN LEAVE DUMPCSRLOOP
                                                                                   END OF FOR CSRNO
                                                                                   POP R1, CSRNO, CSR
                                                                                   PUSH RO.R
   15207
15208 062610
15209 062614
15210 062620
15211 062624
15212 062630
15213 062632
15214 062636
15215 062642
15216 062646
15217 062652
15218
15219 062666
15221 062670
                                                                                                 RO.R1
MSG088
                                                                                   TYPE
                                                                                                                             :KERNEL STACK
                                                                                                 KSTACK,R1
#2,R1
R0 := SP TO R1 BY #2
                           013701
162701
                                                                                   MOV
                                                                                   SUB
                                                                                   FOR
                                                                                     TYPE SCRLF
                                                                                      TYPOCT
                                                                                                               RO
                                                                                                               MSG018
                                                                                                                                          :2 SPACES
                                                                                      TYPE
                                                                                      TYPOCT
                                                                                                               (RO)
                                                                                   END : OF FOR RO : SET PREVIOUS MODE TO SUPERVISOR TST NOSUPER
                           005737
001036
042737
052737
             062662
062670
062676
062704
062706
062712
062716
062724
062730
062734
062740
062760
062760
                                         002452
                                                                                   BNE
                                                                                                 DET1
                                                                                                 WBIT13!BIT12,PSW
                                          030000
                                                        177776
                                          010000
                                                                                                 SSP
R1,R0
                            006506
                                                                                   MFPI
                                                                                   POP
                                                                                    TYPE
                                                                                                 MSG089
                                                                                                                            :SUPERVISOR STACK
                                                                                   IF RO LT #SUPSTK
FOR RO := RO TO #SUPSTK-2 BY #2
                                                                                                               SCRLF
                                                                                          TYPE
                                                                                          TYPOCT
                                                                                                               RO
                                                                                                               MSG018
(R0)
                                                                                                                                      :2 SPACES
                                                                                           TYPOCT
    15231
15232
15233
15234
15235
15236
15237
15238
15239
15241
                                                                                       END OF FOR RO
                                                                                   TYPE MSG091 ; IS

END : OF IF RO

: SET PREVIOUS MODE TO USER

BIS MBIT13!BIT12,PSW
                                                                                                                             :IS EMPTY
              062764
062772
062774
062776
063002
063010
                            052737
006506
                                          030000 177776 DET1:
                                                                                                 USP
RO
                                                                                    MFPI
                                                                                    POP
                                                                                                MSG090
                                                                                    TYPE
                                                                                                                             :USER STACK
                                                                                    IF RO LT MUSESTK
                                                                                       FOR RO := RO TO #USESTK-2 BY #2
```

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 478-1

15243 063010
15244 063014
15245 063020
15246 063024
15247 063030
15248 063042
15249 063044
15250 063050
15251 063050
15252 063054 005037 002216
15253 063060
15254 063062 000207

RETURN

TYPE $CRLF
(RO)
ELSE
TYPE MSG091 ;IS EMPTY
END ;OF IF RO
TYPE $CRLF
CLR DETFLAG
POP RO
RETURN
```

```
.SBTTL ROUTINE BINARY TO OCTAL (ASCII) AND TYPE
                                          **********
                                         *THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                         *OCTAL (ASCII) NUMBER AND TYPE IT.
                                         **STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                         : *CALL:
                                                                                  :: NUMBER TO BE TYPED
                                                             NUM, -(SP)
                                                                                  :: CALL FOR TYPEOUT
                                                   TYPOS
                                                                                  :: N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
:: M=1 OR 0
                                                   BYTE.
                                                                                            ::1=TYPE LEADING ZEROS
::0=SUPPRESS LEADING ZEROS
                                          *STYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                          *STYPOS OR STYPOC
                                          *CALL:
                                                                                  :: NUMBER TO BE TYPED
                                                             NUM, -(SP)
                                          *
                                                                                  :: CALL FOR TYPEOUT
                                                   TYPON
                                         **STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                         : *CALL:
                                                   MOV
                                                                                  :: NUMBER TO BE TYPED
                                                              NUM, -(SP)
                                                   TYPOC
                                                                                  :: CALL FOR TYPEOUT
          017646
116637
112637
062716
000406
112737
112737
                                                              a(SP),-(SP)
1(SP),$0fILL
                                                                                  ::PICKUP THE MODE
                                         $TYPOS: MOV
                                                                                  :: LOAD ZERO FILL SWITCH
063070
063076
063102
063106
063110
                              063307
                    000001
063311
                                                   MOVB
                                                                                  :: NUMBER OF DIGITS TO TYPE
                                                              (SP)+, SOMODE+1
                                                   MOVB
                                                                                  :: ADJUST RETURN ADDRESS
                     000002
                                                              #2,(SP)
                                                   ADD
                                                              STYPON
                                                                                  ::SET THE ZERO FILL SWITCH ::SET FOR SIX(6) DIGITS
                              063307
063311
063306
                    000001
000006
000005
                                                              #1,50FILL
                                         STYPOC:
                                                   MOVB
                                                              #6,$0MODE+1
                                                    MOVB
                                                              #5,$0CNT
                                                                                  SET THE ITERATION COUNT
                                                   MOVB
          112737 010346
                                                             R3,-(SP)
R4,-(SP)
R5,-(SP)
$0MODE+1,R4
                                                                                  SAVE R3
                                                    MOV
                                                                                  SAVE R4
          010446
010546
                                                    MOV
                                                                                  SAVE R5
                                                    MOV
                                                                                   GET THE NUMBER OF DIGITS TO TYPE
                    063311
063140
063146
063152
063156
063162
063166
063172
063174
063176
063202
063202
                                                   NEG
                                                              #6,R4
R4,$0MODE
$0FILL,R4
                                                                                  ::SUBTRACT IT FOR MAX. ALLOWED ::SAVE IT FOR USE
                                                    ADD
                    063310
063307
000012
                                                    MOVB
                                                                                  GET THE ZERO FILL SWITCH
                                                    MOVB
                                                              12(SP),R5
R3
R5
R5
R5
R5
R5
R5
          016605
                                                    MOV
                                                                                  CLEAR THE OUTPUT WORD
                                                   CLR
ROL
BR
POL
                                                                                   :: ROTATE MSB INTO
                                         15:
                                                                                   :: GO DO MSB
                                                                                   :: FORM THIS DIGIT
                                         25:
                                                    ROL
                                                    ROL
                                                              R5,R3
                                                    MOV
                                                                                  :: GET LSB OF THIS DIGIT :: TYPE THIS DIGIT?
                                         35:
                                                    ROL
                                                              SOMODE
                                                    DECB
                     063310
                                                   BPL
BIC
BNE
TST
                                                                                  ::BR IF NO
                                                                                  GET RID OF JUNK
                                                              #177770,R3
                     177770
                                                              4$
R4
5$
R4
                                                                                   SUPPRESS THIS 0?
                                                                                   BR IF YES ... DON'T SUPPRESS ANYMORE O'S
                                                    BEQ
                                         45:
```

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1	113 26-APR-82	09:41 F	PAGE 481-1
--	---------------	---------	------------

MODITIME	DIMMI	IO OCIAL	1496111	MIND III				
15349 15350 15351	063230 063234 063240	052703 052703 110337	000060 000040 063304		5\$:	BIS BIS MOVB TYPE	#'0,R3 #',R3 R3,8\$::MAKE THIS DIGIT ASCII ::MAKE ASCII IF NOT ALREADY ::SAVE FOR TYPING ::GO TYPE THIS DIGIT
15353 15354 15355	063254 063254 063256	105337 003347 002402	063306		6\$:	DECB BGT BLT INC	8\$ \$0CNT 2\$ 7\$ R4	::COUNT BY 1 ::BR IF MORE TO DO ::BR IF DONE ::INSURE LAST DIGIT ISN'T A BLANK
15357 15358 15359 15360	063262 063264 063266 063270	000744 012605 012604 012603			7\$:	BR MOV MOV	2\$ (SP)+,R5 (SP)+,R4 (SP)+,R3	:: GO DO THE LAST DIGIT :: RESTORE RS :: RESTORE R4 :: RESTORE R3
15361 15362 15363 15364	063272 063300 063302 063304	016666 012616 000002 000	000002	000004	8\$:	MOV MOV RTI	2(SP),4(SP) (SP)+,(SP)	::SET THE STACK FOR RETURNING ::RETURN ::STORAGE FOR ASCII DIGIT
15365 15366 15367 15368	063305 063306 063307 063310	000 000 000 000			SOCNT: SOFILL: SOMODE:	BYTE BYTE BYTE BYTE WORD	0	TERMINATOR FOR TYPE ROUTINE COCTAL DIGIT COUNTER ZERO FILL SWITCH NUMBER OF DIGITS TO TYPE

KOOLINE	COMACKI	DIMANI	IO DECIPO	AL AND I	11.6			
15370 15371 15372 15373 15374 15375 15376					****** *THIS I *SIGNEI *NUMBEI *BEFORI *REPLAI *CALL:	ROUTINE D DECIMAL R IS POS E THE FIL CED WITH		BINARY TO DECIMAL AND TYPE ***********************************
15378					*	MOV	NUM,-(SP)	::PUT THE BINARY NUMBER ON THE STACK ::GO TO THE ROUTINE
15380 15381 15382 15383	063312 063324 063330 063334 063336 063340 063346 063350	012746 016605 100004 005405 112766	020200		\$TYPDS:	PUSH MOV MOV BPL	RO,R1,R2,R3,R5 #20200,-(SP) 20(SP),R5 1\$ R5	::SET BLANK SWITCH AND SIGN ::GET THE INPUT NUMBER ::BR IF INPUT IS POS. ::MAKE THE BINARY NUMBER POS. ::MAKE THE ASCII NUMBER NEG. ::ZERO THE CONSTANTS INDEX ::SETUP THE OUTPUT POINTER ::SET THE FIRST CHARACTER TO A BLANK ::CLEAR THE BCD NUMBER ::GET THE CONSTANT ::FORM THIS BCD DIGIT ::BR IF DONE ::INCREASE THE BCD DIGIT BY 1
15384 15385	063336	112766	000055	000001	10.	MOVB	#'-,1(SP) RO	:: MAKE THE ASCII NUMBER NEG.
15386 15387 15388	063350	005000 012703 112723 005002 016001 160105	063526 000040		1\$:	CLR MOV MOVB	#\$DBLK,R3 #',(R3)+ R2	SETUP THE OUTPUT POINTER
15389 15390	063360	005002	063516		2\$:	CLR	R2 \$DTBL(R0),R1	;; CLEAR THE BCD NUMBER ;; GET THE CONSTANT
15391 15392 15393	063360 063362 063366 063370 063372 063374	005202	003310		3\$:	SUB BLT INC BR	R1,R5 4\$ R2 3\$	FORM THIS BCD DIGIT BR IF DONE INCREASE THE BCD DIGIT BY 1
15381 15382 15383 15384 15385 15386 15387 15390 15391 15392 15393 15394 15397 15397	063376 063400 063402 063404 063406 063410	000774 060105 005702 001002 105716			48:	ADD TST BNE TSTB	R1,R5 R2 5\$ (SP) 7\$::ADD BACK THE CONSTANT ::CHECK IF BCD DIGIT=0 ::FALL THROUGH IF 0 ::STILL DOING LEADING 0'S?
15399 15400	063410	100407			5\$:	BMI ASLB BCC	(SP)	::MSD?
15401 15402 15403 15404 15405	063410 063412 063414 063422 063426 063432 063434 063446 063446 063450 063454 063456 063464	106316 103003 116663 052702 052702 110223	000001 000060 000040	177777	6\$: 7\$:	MOVB BIS BIS MOVB TST	1(SP),-1(R3) #'0,R2 #',R2 R2,(R3)+ (R0)+	::ADD BACK THE CONSTANT ::CHECK IF BCD DIGIT=0 ::FALL THROUGH IF 0 ::STILL DOING LEADING 0'S? ::BR IF YES ::MSD? ::BR IF NO ::YESSET THE SIGN ::MAKE THE BCD DIGIT ASCII ::MAKE IT A SPACE IF NOT ALREADY A DIGIT ::PUT THIS CHARACTER IN THE OUTPUT BUFFER ::JUST INCREMENTING
15407 15408 15409 15410	063436 063442 063444 063446	110223 005720 020027 002746 003002 010502	000010			CMP BLT BGT MOV BR	R0.#10 2\$ 8\$ R5.R2 6\$::PUT THIS CHARACTER IN THE OUTPUT BUFFER ::JUST INCREMENTING ::CHECK THE TABLE INDEX ::GO DO THE NEXT DIGIT ::GO TO EXIT ::GO TO EXIT ::GET THE LSD ::GO CHANGE TO ASCII ::WAS THE LSD THE FIRST NON-ZERO? ::BR IF NO ::YESSET THE SIGN FOR TYPING ::SET THE TERMINATOR
15412	063452	105726			8\$:	TSTB	(SP)+ 9\$:: WAS THE LSD THE FIRST NON-ZERO?
15414 15415 15416	063456 063464	000764 105726 100003 116663 105013	177777	177776	9\$:	MOVB CLRB POP	-1(SP),-2(R3) (R3) R5,R3,R2,R1,R0	::YESSET THE SIGN FOR TYPING ::SET THE TERMINATOR
15417 15418	063500 063504	016666	000002	000004		MOV	\$DBLK 2(SP),4(SP)	;; NOW TYPE THE NUMBER ;; ADJUST THE STACK
15429 15421 15422 15423	063464 063500 063504 063512 063514 063516 063520 063522 063524 063524	016666 012616 000002 023420 001750 000144 000012 000000			\$DTBL:	MOV RTI 10000. 1000.	(SP)+,(SP)	;;RETURN TO USER
15425	063526 063534	000000	000000	000000	\$DBLK:	10. .WORD	0.0.0.0	

Settle S	MODITHE	III LINE	01						
15428 15430 15431 15432 15433 15434 063536 05737 056276 15440 063542 001406 15442 063542 001406 15442 063542 001406 15442 0635540 005037 056276 15443 0635540 001577 17042 15445 063564 105777 17042 15445 063564 105777 17042 15445 063566 105777 17042 15445 063566 105777 17042 15445 063566 105777 17042 15445 063566 105777 17042 15445 063566 105777 17042 15445 063566 105777 17042 15445 063566 105777 17042 15445 063566 105777 17042 15445 063566 105777 17042 15445 1063664 1001002 15450 1063664 1001002 1785	15427						.SBTTL	ROUTINE TTY IN	PUT
15420 1543	15428					;*****		*********	
15435 15436 063536 105737 056276	15429					: *SOFTWA	ARE SWIT	CH REGISTER CHA	NGE ROUTINE.
15435 15436 063536 105737 056276	15430					; *ROUTIN	NE IS EN	TERED FROM THE	TRAP HANDLER, AND WILL
15435 15436 063536 105737 056276	15431					; *SERVI	CE THE T	EST FOR CHANGE	IN 201 MAKE 2MILLY KERTZIEK IKAL CATT
15434 063536 15440 063536 005737 056276 BEQ NOCH	15432					: *WHEN	PERALIN	G IN TIT FLAG M	IUDE.
15447 063566 117766 117036 MOVB BIC M°C177 (SP) 15448 063572 042716 177600 BIC M°C177 (SP) 15449 063576 022716 000006 CONTS1: CMP M6 (SP) 15450 063602 001002 CALL FIELDSERVICE 15451 063604 004737 050102 CALL FIELDSERVICE 15452 063610 022716 000024 1\$: CMP M24 (SP) 15453 063614 001002 BNE 16\$ NO SKIP 15453 063616 004737 064144 CONTT : YES - CALL CONTROL T? 15455 063620 022716 000003 16\$: CMP M3 (SP) 15456 063620 022716 000003 16\$: CMP M3 (SP) 15457 063630 022716 000003 16\$: CMP M23 (SP) 15458 063634 001002 BNE 17\$: YES EXIT *****NOTE***** STACK IS SCREWED UP! 15459 063636 004737 064220 CALL CONTS : YES EXIT *****NOTE***** STACK IS SCREWED UP! 15461 063646 001005 BNE 17\$: NO - SKIP 15461 063646 001005 TYPE SCNTLK : YES - CALL CONTROL R? 15462 063650 13706 002144 MOV CTLKVEC, SP RETURN TO PATTERN EXEC ROUTINE 15463 063664 0022737 000176 002622 6\$: CMP MSUREG.SUR :: IS IT EOFT-SWR SELECTEP?	15433	047574				CLACHD.	.EMABLE	F20	
15447 063566 117766 117036 MOVB BIC M°C177 (SP) 15448 063572 042716 177600 BIC M°C177 (SP) 15449 063576 022716 000006 CONTS1: CMP M6 (SP) 15450 063602 001002 CALL FIELDSERVICE 15451 063604 004737 050102 CALL FIELDSERVICE 15452 063610 022716 000024 1\$: CMP M24 (SP) 15453 063614 001002 BNE 16\$ NO SKIP 15453 063616 004737 064144 CONTT : YES - CALL CONTROL T? 15455 063620 022716 000003 16\$: CMP M3 (SP) 15456 063620 022716 000003 16\$: CMP M3 (SP) 15457 063630 022716 000003 16\$: CMP M23 (SP) 15458 063634 001002 BNE 17\$: YES EXIT *****NOTE***** STACK IS SCREWED UP! 15459 063636 004737 064220 CALL CONTS : YES EXIT *****NOTE***** STACK IS SCREWED UP! 15461 063646 001005 BNE 17\$: NO - SKIP 15461 063646 001005 TYPE SCNTLK : YES - CALL CONTROL R? 15462 063650 13706 002144 MOV CTLKVEC, SP RETURN TO PATTERN EXEC ROUTINE 15463 063664 0022737 000176 002622 6\$: CMP MSUREG.SUR :: IS IT EOFT-SWR SELECTEP?	15434	063536	005737	056276		SCKSWK:	TST	YOCHAR	::SOMETHING THERE?
15447 063566 117766 117036 MOVB BIC M°C177 (SP) 15448 063572 042716 177600 BIC M°C177 (SP) 15449 063576 022716 000006 CONTS1: CMP M6 (SP) 15450 063602 001002 CALL FIELDSERVICE 15451 063604 004737 050102 CALL FIELDSERVICE 15452 063610 022716 000024 1\$: CMP M24 (SP) 15453 063614 001002 BNE 16\$ NO SKIP 15453 063616 004737 064144 CONTT : YES - CALL CONTROL T? 15455 063620 022716 000003 16\$: CMP M3 (SP) 15456 063620 022716 000003 16\$: CMP M3 (SP) 15457 063630 022716 000003 16\$: CMP M23 (SP) 15458 063634 001002 BNE 17\$: YES EXIT *****NOTE***** STACK IS SCREWED UP! 15459 063636 004737 064220 CALL CONTS : YES EXIT *****NOTE***** STACK IS SCREWED UP! 15461 063646 001005 BNE 17\$: NO - SKIP 15461 063646 001005 TYPE SCNTLK : YES - CALL CONTROL R? 15462 063650 13706 002144 MOV CTLKVEC, SP RETURN TO PATTERN EXEC ROUTINE 15463 063664 0022737 000176 002622 6\$: CMP MSUREG.SUR :: IS IT EOFT-SWR SELECTEP?	15441	0633330	001406	070270			REO	NOCH	:: GO ON IF NOT
15447 063566 117766 117036 MOVB BIC M°C177 (SP) 15448 063572 042716 177600 BIC M°C177 (SP) 15449 063576 022716 000006 CONTS1: CMP M6 (SP) 15450 063602 001002 CALL FIELDSERVICE 15451 063604 004737 050102 CALL FIELDSERVICE 15452 063610 022716 000024 1\$: CMP M24 (SP) 15453 063614 001002 BNE 16\$ NO SKIP 15453 063616 004737 064144 CONTT : YES - CALL CONTROL T? 15455 063620 022716 000003 16\$: CMP M3 (SP) 15456 063620 022716 000003 16\$: CMP M3 (SP) 15457 063630 022716 000003 16\$: CMP M23 (SP) 15458 063634 001002 BNE 17\$: YES EXIT *****NOTE***** STACK IS SCREWED UP! 15459 063636 004737 064220 CALL CONTS : YES EXIT *****NOTE***** STACK IS SCREWED UP! 15461 063646 001005 BNE 17\$: NO - SKIP 15461 063646 001005 TYPE SCNTLK : YES - CALL CONTROL R? 15462 063650 13706 002144 MOV CTLKVEC, SP RETURN TO PATTERN EXEC ROUTINE 15463 063664 0022737 000176 002622 6\$: CMP MSUREG.SUR :: IS IT EOFT-SWR SELECTEP?	15442	063544	013746	056276				XOCHAR,-(SP)	:: USE IT
15447 063566 117766 117036 MOVB BIC M°C177 (SP) 15448 063572 042716 177600 BIC M°C177 (SP) 15449 063576 022716 000006 CONTS1: CMP M6 (SP) 15450 063602 001002 CALL FIELDSERVICE 15451 063604 004737 050102 CALL FIELDSERVICE 15452 063610 022716 000024 1\$: CMP M24 (SP) 15453 063614 001002 BNE 16\$ NO SKIP 15453 063616 004737 064144 CONTT : YES - CALL CONTROL T? 15455 063620 022716 000003 16\$: CMP M3 (SP) 15456 063620 022716 000003 16\$: CMP M3 (SP) 15457 063630 022716 000003 16\$: CMP M23 (SP) 15458 063634 001002 BNE 17\$: YES EXIT *****NOTE***** STACK IS SCREWED UP! 15459 063636 004737 064220 CALL CONTS : YES EXIT *****NOTE***** STACK IS SCREWED UP! 15461 063646 001005 BNE 17\$: NO - SKIP 15461 063646 001005 TYPE SCNTLK : YES - CALL CONTROL R? 15462 063650 13706 002144 MOV CTLKVEC, SP RETURN TO PATTERN EXEC ROUTINE 15463 063664 0022737 000176 002622 6\$: CMP MSUREG.SUR :: IS IT EOFT-SWR SELECTEP?	15443	063550	005037	056276			CLR	XOCHAR	
15447 063566 117766 117036 MOVB BIC M°C177 (SP) 15448 063572 042716 177600 BIC M°C177 (SP) 15449 063576 022716 000006 CONTS1: CMP M6 (SP) 15450 063602 001002 CALL FIELDSERVICE 15451 063604 004737 050102 CALL FIELDSERVICE 15452 063610 022716 000024 1\$: CMP M24 (SP) 15453 063614 001002 BNE 16\$ NO SKIP 15453 063616 004737 064144 CONTT : YES - CALL CONTROL T? 15455 063620 022716 000003 16\$: CMP M3 (SP) 15456 063620 022716 000003 16\$: CMP M3 (SP) 15457 063630 022716 000003 16\$: CMP M23 (SP) 15458 063634 001002 BNE 17\$: YES EXIT *****NOTE***** STACK IS SCREWED UP! 15459 063636 004737 064220 CALL CONTS : YES EXIT *****NOTE***** STACK IS SCREWED UP! 15461 063646 001005 BNE 17\$: NO - SKIP 15461 063646 001005 TYPE SCNTLK : YES - CALL CONTROL R? 15462 063650 13706 002144 MOV CTLKVEC, SP RETURN TO PATTERN EXEC ROUTINE 15463 063664 0022737 000176 002622 6\$: CMP MSUREG.SUR :: IS IT EOFT-SWR SELECTEP?	15444	063554	000137	063576			JMP	CONTS1	
15447 063566 117766 117036 MOVB BIC M°C177 (SP) 15448 063572 042716 177600 BIC M°C177 (SP) 15449 063576 022716 000006 CONTS1: CMP M6 (SP) 15450 063602 001002 CALL FIELDSERVICE 15451 063604 004737 050102 CALL FIELDSERVICE 15452 063610 022716 000024 1\$: CMP M24 (SP) 15453 063614 001002 BNE 16\$ NO SKIP 15453 063616 004737 064144 CONTT : YES - CALL CONTROL T? 15455 063620 022716 000003 16\$: CMP M3 (SP) 15456 063620 022716 000003 16\$: CMP M3 (SP) 15457 063630 022716 000003 16\$: CMP M23 (SP) 15458 063634 001002 BNE 17\$: YES EXIT *****NOTE***** STACK IS SCREWED UP! 15459 063636 004737 064220 CALL CONTS : YES EXIT *****NOTE***** STACK IS SCREWED UP! 15461 063646 001005 BNE 17\$: NO - SKIP 15461 063646 001005 TYPE SCNTLK : YES - CALL CONTROL R? 15462 063650 13706 002144 MOV CTLKVEC, SP RETURN TO PATTERN EXEC ROUTINE 15463 063664 0022737 000176 002622 6\$: CMP MSUREG.SUR :: IS IT EOFT-SWR SELECTEP?	15445	063560	105777	117042		NOCH:		STKS	;; CHAR THERE?
15449 063576 022716 000006 CONTS1: CMP	15446	063564	100130	44707/				125	;; IF NU, DUN'T WALL AROUND
15449 063576 022716 000006 CONTS1: CMP	15447	063266	11//46	117036				#4C177 (SP)	STOID-OFF THE ASCII
15451 063604 004737 050102	15448	063377	022716	000006		CONTS1.		#6 (SP)	IS IT CONTROL F?
15451 063604 004737 050102	15450	063376	001002	000000		CONTST.	RNF	15	NO SKIP
15462 063650 15463 063654 013706 002144 MOV CTLKVEC,SP ;RESET KSP TO AFTER PATTERN EXEC ROUTINE 15464 063660 000207 RETURN ;RETURN TO PATTERN EXEC ROUTINE 15465 063662 022737 000176 002622 6\$: CMP #SWREG.SWR ::IS THE SOFT-SWR SELECTED?	15751	063602	004737	050102				FIELDSERVICE	
15462 063650 15463 063654 013706 002144 MOV CTLKVEC,SP ;RESET KSP TO AFTER PATTERN EXEC ROUTINE 15464 063660 000207 RETURN ;RETURN TO PATTERN EXEC ROUTINE 15465 063662 022737 000176 002622 6\$: CMP #SWREG.SWR ::IS THE SOFT-SWR SELECTED?	15452	063610	022716	000024		15:		#24_(SP)	; IS IT CONTROL T?
15462 063650 15463 063654 013706 002144 MOV CTLKVEC,SP ;RESET KSP TO AFTER PATTERN EXEC ROUTINE 15464 063660 000207 RETURN ;RETURN TO PATTERN EXEC ROUTINE 15465 063662 022737 000176 002622 6\$: CMP #SWREG.SWR ::IS THE SOFT-SWR SELECTED?	15453	063614	001002					165	:NO - SKIP
15462 063650 15463 063654 013706 002144 MOV CTLKVEC,SP ;RESET KSP TO AFTER PATTERN EXEC ROUTINE 15464 063660 000207 RETURN ;RETURN TO PATTERN EXEC ROUTINE 15465 063662 022737 000176 002622 6\$: CMP #SWREG.SWR ::IS THE SOFT-SWR SELECTED?	15454	063616	004737	064144			CALL	CONTT	; YES - CALL CONTROL T ROUTINE
15462 063650 15463 063654 013706 002144 MOV CTLKVEC,SP ;RESET KSP TO AFTER PATTERN EXEC ROUTINE 15464 063660 000207 RETURN ;RETURN TO PATTERN EXEC ROUTINE 15465 063662 022737 000176 002622 6\$: CMP #SWREG.SWR ::IS THE SOFT-SWR SELECTED?	15455	063622	022716	000003		165:		#5,(SP)	:15 IT CONTROL C?
15462 063650 15463 063654 013706 002144 MOV CTLKVEC,SP ;RESET KSP TO AFTER PATTERN EXEC ROUTINE 15464 063660 000207 RETURN ;RETURN TO PATTERN EXEC ROUTINE 15465 063662 022737 000176 002622 6\$: CMP #SWREG.SWR ::IS THE SOFT-SWR SELECTED?	15456	063626	001454	000027		20.		#27 (CD)	TES ENT CONTROL CO
15462 063650 15463 063654 013706 002144 MOV CTLKVEC,SP ;RESET KSP TO AFTER PATTERN EXEC ROUTINE 15464 063660 000207 RETURN ;RETURN TO PATTERN EXEC ROUTINE 15465 063662 022737 000176 002622 6\$: CMP #SWREG.SWR ::IS THE SOFT-SWR SELECTED?	15457	063630	001003	000023		29:	PNE	178	·NO - SKIP
15462 063650 15463 063654 013706 002144 MOV CTLKVEC,SP ;RESET KSP TO AFTER PATTERN EXEC ROUTINE 15464 063660 000207 RETURN ;RETURN TO PATTERN EXEC ROUTINE 15465 063662 022737 000176 002622 6\$: CMP #SWREG.SWR ::IS THE SOFT-SWR SELECTED?	15450	053634 A5A5A0	001002	064220				CONTS	:YES - CALL CONTROL S ROUTINE
15462 063650 15463 063654 013706 002144 MOV CTLKVEC,SP ;RESET KSP TO AFTER PATTERN EXEC ROUTINE 15464 063660 000207 RETURN ;RETURN TO PATTERN EXEC ROUTINE 15465 063662 022737 000176 002622 6\$: CMP #SWREG.SWR ::IS THE SOFT-SWR SELECTED?	15460	063642	022716	000013		175:		#13.(SP)	:IS IT CONTROL K?
15462 063650 15463 063654 013706 002144 MOV CTLKVEC,SP ;RESET KSP TO AFTER PATTERN EXEC ROUTINE 15464 063660 000207 RETURN ;RETURN TO PATTERN EXEC ROUTINE 15465 063662 022737 000176 002622 6\$: CMP #SWREG.SWR ::IS THE SOFT-SWR SELECTED?	15461	063646	001005	000013			BNE	6\$;NO - SKIP
15463 063664 013706 002144 MOV CTLKVEC,SP RETURN (15464 063660 000207) 15466 063662 022737 000176 002622 6\$: CMP RETURN (1547 063672 022716 000007) 15466 063670 001067 002070 CMP RETURN (1547 063704 001061) 15470 063704 001061	15462	063650					TYPE	SCNTLK	TYPE A AK
15464	15463	063654	013706	002144				CTLKVEC, SP	RESET KSP TO AFTER PATTERN EXEC ROUTINE
15465 063662 022/37 0001067 00007	15464	063660	000207	00017/	002/22	10.	RETURN	MCLIDEC CLID	RETURN TO PATTERN EXEL ROUTINE
15467 063672 022716 000007	12462	063662	022/3/	000176	002022	09:	DNE	WOMKED'OMK	· · RPANCH IF NO.
15468 063760 0050737 0050737 002060 TST \$AUTO ;ARE WE RUNNING IN AUTO-MODE? 15470 063704 001061 TYPE \$CKEND ;BRANCH IF YES 15473 063716 TYPE \$MSWR ;:FYPE CURRENT CONTENTS 15473 063716 TYPE \$MSWR ;:FYPE CURRENT CONTENTS 15473 063730 005046 TYPE \$MSWR ;:FROMPT FOR NEW SWR 15473 063730 005046 TYPE \$MNEW ;PROMPT FOR NEW SWR 15475 063732 005046 CLR -(SP) ;CLEAR COUNTER 15476 063732 005046 TSTSTB \$STKS ;CHAR THERE? 15478 063740 100375 BPL \$STKS ;CHAR THERE? 15479 063740 100375 BPL \$STKS ;CHAR THERE? 15480 063746 042716 177600 BIC MCC177 (SP) ;PICK UP CHAR THERE SWR 15480 063746 042716 177600 BIC MCC177 (SP) ;PICK UP CHAR THERE SWR 15480 063746 042716 177600 BIC MCC177 (SP) ;PICK UP CHAR THERE SWR 15483 063760 TST CMP (SP), M3 ;IS IT A CONTROL-C? 15483 063760 TST CMP (SP), M3 ;IS IT A CONTROL-C (CC) 15484 063764 063770 000137 047524 JMP BOOT ;CONTROL-C RESTART TYPE \$CNTLU ;YES, ECHO CONTROL-U (CU)	15460	063670	022716	000007				47 (SP)	:: IS IT A CONTROL G?
15469 063700 005737 002060 15470 063704 001061 15471 063706 15472 063712 15473 063716 15474 063724 15475 063730 005046 15476 063732 005046 15477 063734 105777 11666 15477 063734 105777 11666 15478 063740 100375 15478 063740 100375 15480 063740 005046 15480 063750 005046 15480 063750 005046 15480 063760 005046 15480 063770 00505 78: CMP (SP).#35 ;:IS IT A CONTROL-C (*C) 15480 063760 00505 78: CMP (SP).#25 ;:IS IT A CONTROL-C (*C) 15480 063770 00505 78: CMP (SP).#25 ;:IS IT A CONTROL-C (*C) 15480 064002	15468	063676	001064	000001				CKEND	::NO. RETURN TO USER
15470 063704 001061	15469	063700	005737	002060			TST	SAUTO	; ARE WE RUNNING IN AUTO-MODE?
15471 063706 15473 063716 15473 063716 15474 063724 15475 063730 15476 063732 15476 063732 15477 063732 15477 063732 15477 063734 105777 116666 15478 063740 15479 063740 15479 063740 15479 063740 15479 063740 15479 063740 15480 063746 15481 063752 15481 063752 15482 063760 15483 063760 15484 063764 15484 063764 15485 063770 15486 063774 15486 063774 15486 063774 15487 064000 15488 064002 15488 064002 15488 064002 15488 064002	15470	063704	001061				BNE	CKEND	BRANCH IF YES
15472 063712 15473 063716 15474 063724 15475 063730 15476 063732 15476 063732 105046 15477 063734 105777 116666 15478 063740 100375 15479 063742 117746 116662 15479 063742 117746 116662 15480 063746 042716 177600 15481 063750 021627 000003 15482 063756 001006 15483 063760 15484 063764 062706 000006 15484 063764 062706 000006 15485 063770 000137 047524 15485 063770 001005 15486 063774 021627 000025 15488 064002 SGTSWR: TYPE SMSWR 17PE SMNEW 1PROMPT FOR NEW SWR 1PROMPT FOR NEW SWR 1CLEAR COUNTER 1PROMPT FOR NEW SWR 1PROMP	15471	063706					TYPE	SCNTLG	;; ECHO THE CONTROL-G (~G)
15475 063724 15476 063730 005046 15476 063732 005046 15477 063734 105777 116666 15478 063740 100375 15479 063740 100375 15479 063742 117746 116662 15480 063746 042716 177600 15481 063752 021627 000003 15482 063756 001006 15483 063760 001006 15484 063764 062706 000006 15484 063764 062706 000006 15485 063770 000137 047524 15486 063774 021627 000025 15488 064002 15488 064002 15488 064002 15488 064002 15488 064002 15488 064002 15488 064002	15472	063712				SGTSWR:	TYPE	SMSWR	;; ITPE CURRENT CUNTENTS
15475 063730 005046 15476 063732 005046 15477 063734 105777 116666 15478 063740 100375 15478 063740 100375 15479 063742 117746 116662 15480 063746 042716 177600 15481 063752 021627 000003 15482 063756 001006 15483 063760 01006 15484 063764 062706 000006 15484 063764 062706 000006 15485 063770 000137 047524 15486 063774 021627 000025 15487 064000 001005 15488 064002 15488 064002 15488 064002 15488 064002 15488 064002 15488 064002 3\$: CLEAR COUNTER : CLEAR THERE? : IF NOT TRY AGAIN :	15473	063/16					TYPE	@2MK	DDUMDI EUD MEH CHD
15476 063732 005046 15477 063734 105777 116666 15478 063740 100375 15478 063740 100375 15479 063742 117746 116662 15480 063746 042716 177600 15481 063752 021627 000003 15482 063756 001006 15483 063760 15483 063760 15484 063764 062706 00006 15484 063764 062706 00006 15485 063770 000137 047524 15486 063774 021627 000025 15487 064000 001005 15488 064002 15488 064002 15488 064002 15488 064002	15/75	063724	005046			76.	CIP	-(92)-	: CLEAR COUNTER
15477 063734 105777 116666 4\$: TSTB a\$TK\$;: CHAR THERE? 15478 063740 100375 BPL 4\$;: IF NOT TRY AGAIN 15479 063742 117746 116662 MOVB a\$TKB,-(SP) ;: PICK UP CHAR 15480 063746 042716 177600 BIC M^C177, (SP) ;: MAKE IT 7-BIT ASCII 15481 063752 021627 000003 CMP (SP), M3 ;: IS IT A CONTROL-C? 15482 063756 001006 BNE 7\$;: BRANCH IF NOT 15483 063760 062706 000006 ADD M6, SP ;: CLEAN UP STACK 15485 063770 000137 047524 JMP BOOT ;: CONTROL-C RESTART 15486 063774 021627 000025 7\$: CMP (SP), M25 ;: IS IT A CONTROL-U? 15487 064000 001005 BNE 9\$;: BRANCH IF NOT 15488 064002 TYPE \$CNTLU ;: YES, ECHO CONTROL-U (^U)	15476	063732	005046			34.	CLR	-(SP)	:: THE NEW SWR
15478 063740 100375 15479 063742 117746 116662 MOVB @\$TKB,-(SP) ;:PICK UP CHAR 15480 063746 042716 177600 BIC M^C177,(SP) ;:MAKE IT 7-BIT ASCII 15481 063752 021627 000003 CMP (SP), M3 ;:IS IT A CONTROL-C? 15482 063756 001006 BNE 7\$;:BRANCH IF NOT 15484 063764 062706 000006 ADD M6, SP ;:CLEAN UP STACK 15485 063770 000137 047524 JMP BOOT ;:CONTROL-C RESTART 15486 063774 021627 000025 7\$: CMP (SP), M25 ;:IS IT A CONTROL-U? 15487 064000 001005 BNE 9\$;:BRANCH IF NOT 15488 064002 TYPE \$CNTLU ;:YES, ECHO CONTROL-U (^U)	15477	063734	105777	116666		45:	TSTB	astks	:: CHAR THERE?
15479 063742 117746 116662 MOVB @\$TKB,-(SP) ;:PICK UP CHAR 15480 063746 042716 177600 BIC M^C177,(SP) ;:MAKE IT 7-BIT ASCII 15481 063752 021627 000003 CMP (SP), M3 ;:IS IT A CONTROL-C? 15482 063756 001006 BNE 7\$;:BRANCH IF NOT 15484 063764 062706 000006 ADD M6, SP ;:CLEAN UP STACK 15485 063770 000137 047524 JMP BOOT ;:CONTROL-C RESTART 15486 063774 021627 000025 7\$: CMP (SP), M25 ;:IS IT A CONTROL-U? 15487 064000 001005 BNE 9\$;:BRANCH IF NOT 15488 064002 TYPE \$CNTLU ::YES, ECHO CONTROL-U (^U)	15478	063740	100375				BPL	4\$;; IF NOT TRY AGAIN
15480 063746 042716 177600 15481 063752 021627 000003 15482 063756 001006 15483 063760 15484 063764 062706 000006 15484 063770 000137 047524 15486 063774 021627 000025 15487 064000 001005 15488 064002 15488 064002 15480 064000	15479	063742	117746	116662			MOVB	astkB(SP)	;;PICK UP CHAR
15481 063752 021627 000003 15482 063756 001006 15483 063760 15484 063764 062706 000006 15485 063770 000137 047524 15486 063774 021627 000025 15487 064000 001005 15488 064002 TYPE \$CNTLU SP),#3 ;:IS IT A CONTROL-C? ;:BRANCH IF NOT ;:CLEÁN UP STACK ;:CONTROL-C RESTART ;:IS IT A CONTROL-U? ;:IS IT A CONTROL-U? ;:BRANCH IF NOT	15480	063746	042716	177600			BIC	#*C1// (SP)	;;MAKE IT /-BIT ASCII
15482 063760 15484 063764 062706 000006 15485 063770 000137 047524 15486 063774 021627 000025 15487 064000 001005 15488 064002 5\$: TYPE \$CNTLC ;:YES, ECHO CONTROL-C (^C) ;:CLEAN UP STACK ;:CONTROL-C RESTART ;:CONTROL-C RESTART ;:BRANCH IF NOT	15481	063752	021627	000003			CMP	(37),#3	PRANCH IS NOT
15484 063764 062706 000006 15485 063770 000137 047524 15486 063774 021627 000025 15487 064000 001005 15488 064002 TYPE \$CNTLU ; YES, ECHO CONTROL-U (*U)	12482	063736	001008			58.	TYPE	SCHILC	·· VES ECHO CONTROL -C (^C)
15485 063770 000137 047524 JMP BOOT ;:CONTROL-C RESTART 15486 063774 021627 000025 7\$: CMP (SP),#25 ;:IS IT A CONTROL-U? 15487 064000 001005 BNE 9\$;:BRANCH IF NOT 15488 064002 TYPE \$CNTLU ;:YES, ECHO CONTROL-U (^U)	15/8/	063760	062706	000006		30.		#6.SP	::CLEAN UP STACK
15486 063774 021627 000025 7\$: CMP (SP),#25 ;:IS IT A CONTROL-U? 15487 064000 001005 BNE 9\$;:BRANCH IF NOT 15488 064002 TYPE \$CNTLU ;:YES, ECHO CONTROL-U (^U)	15485	063770	000137	047524			JMP	BOOT	:: CONTROL-C RESTART
15487 064000 001005 BNE 9\$;;BRANCH IF NOT 15488 064002 TYPE \$CNTLU ;;YES, ECHO CONTROL-U (^U)	15486	063774	021627	000025		75:	CMP	(SP),#25	::IS IT A CONTROL-U?
15488 064002 TYPE \$CNTLU ;;YES, ECHO CONTROL-U (~U)	15487	064000	001005				BNE	9\$; BRANCH IF NOT
	15488	064002					TYPE	SCNTLU	:: YES, ECHO CONTROL-U (-U)

CZMSPAO	MS11-L/M/P	MEMORY	DIAG.	MACRO	M1113	26-APR-82	09:41	PAGE	483-	1
---------	------------	--------	-------	-------	-------	-----------	-------	------	------	---

		******				#4 CD	ICHORE PREVIOUS TAIRUT
15489 064006	062706 000746	000006		8\$:	ADD BR	#6.SP	:: IFT'S TRY IT AGAIN
15490 064012 15491 064014 15492 064020 15493 064022 15494 064026 15495 064030 15496 064036 15497 064042	021627	000015		9\$:	CMP		::IGNORE PREVIOUS INPUT ::LET'S TRY IT AGAIN ::IS IT A <cr>? ::BRANCH IF NO ::YES, IS IT THE FIRST CHAR? ::BRANCH IF YES ::SAVE NEW SWR ::CLEAR UP STACK ::ECHO <cr> AND <lf> ::RETURN ::FIX STACK ::RETURN ::ECHO CHAR ::CHAR < 0? ::BRANCH IF YES ::STRIP-OFF ASCII ::IS THIS THE FIRST CHAR ::BRANCH IF YES ::NO, SHIFT PRESENT</lf></cr></cr>
15492 064020	021627 001016 005766				BNE	(SP) ,#15 13\$;;BRANCH IF NO
15493 064022	005766	000004			121	4(SP) 10\$:: YES, 15 II IME FIRST CHAR!
15492 064020 15493 064022 15494 064026 15495 064030	016677	000002	116564		TST BEQ MOV	2(SP), aSWR	SAVE NEW SWR
15496 064036	001403 016677 062706	000006	110301	10\$:	ADD TYPE	#6.SP \$CRLF	::CLEAR UP STACK
15496 064036 15497 064042 15498 064046				120.	TYPE	SCRLF	;;ECHO <cr> AND <lf></lf></cr>
12478 004040	000002 062706	000002		12\$: CKEND:	ADD	#2,SP	FIX STACK
15499 064050 15500 064054	000002				RTI		RETURN
15501 064056	004737	056300		13\$:	CALL	STYPEC	::ECHO CHAR
15502 064062	021627	000060			CMP	(SP),#60 15\$:: BRANCH IF YES
15499 064050 15500 064054 15501 064056 15502 064062 15503 064066 15504 064070 15505 064074 15506 064076 15507 064102 15508 064106 15509 064110 15510 064112 15511 064114 15512 064116 15513 064122 15514 064126 15515 064130	002706 000002 004737 021627 002420 021627 003015 042726 005766	000067			BLT	(SP) .#67 15\$;; CHAR > 7?
15505 064074	003015				BGT BIC TST BEQ ASL	15\$;;BRANCH IF YES
15506 064076	042726	000002			TST	#60,(SP)+ 2(SP)	:: IS THIS THE FIRST CHAR
15507 064102 15508 064106 15509 064110 15510 064112	001403	000002			BEQ	14\$;;BRANCH IF YES
15509 064110	001403 006316				ASL	(SP)	::NO, SHIFT PRESENT
15510 064112	006316 006316				ASL	(SP)	ROOM FOR NEW ONE.
15512 064116	005266	000002		145:	ASL	2(SP)	KEEP COUNT OF CHAR
15511 064114 15512 064116 15513 064122 15514 064126 15515 064130	005266 056616	000002 177776			BIS	-2(SP),(SP)	: BRANCH IF TES : NO, SHIFT PRESENT : CHAR OVER TO MAKE : ROOM FOR NEW ONE. : KEEP COUNT OF CHAR : SET IN NEW CHAR : GET THE NEXT ONE : TYPE ? <cr><lf> : SIMULATE CONTROL-U : CONTROL K ASCII STRING</lf></cr>
15514 064126	000702			15\$:	BR TYPE	4\$ \$QUES	;;GET THE NEXT UNE
15516 064134	000724			130.	BR	8\$::SIMULATE CONTROL-U
15517 064136	136 012	113	015	SCNTLK:	.ASCIZ	/*K/<15><12>	CONTROL K ASCII STRING
16518 064141	012	000			EVEN		
15518 15519					.EVEN	LSB	

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 485
ROUTINE TTY INPUT
                                                    CONTT: SUBTST <<CONTROL T>>
  15522 064144
                                                    *****************
                                                                         CONTROL T
                                                    : *SUBTEST
                                                    **********
  15523 064144
15524 064146
15534 064152
15535 064160
15536 064164
15537 064164
15538 064170
15539 064200
15540 064204
15544 064214
15545 064216
                                                              PUSH RO
TYPE $CRLF
IF RLFLAG IS TRUE
TYPE MSGO
END : OF IF RLFLAG
TYPE MSGO93
                                                                                   MSG092
                                                                                                        :RELOCATED
                                                                                                        :BANK=
:TYPE 3 DIGITS
  15538 064170
15539 064200
15540 064204
15544 064214
15545 064216
15546
15547 064220
                                                                 TYPOCS
                                                                                   BANK,,3
                                                                 TYPE MSG095
                                                                                                         :PAT=
                                                                                                         :TYPE 2 DIGITS
                                                                                   REALPAT .. 2
                                                               POP
                                                               RETURN
                    000207
                                                    CONTS: SUBTST <<CONTROL S & CONTROL Q>>
                                                    : *SUBTEST
                                                                         CONTROL S & CONTROL Q
                                                    *********************
                                                                                                            *********
                                                                                              GET RID OF RETURN ADDRESS FROM STACK
                     105777
100375
117716
042716
                                                    CONTS2: TSTB
                                                                         asTKS
                                                                                              :WAIT FOR CHARACTER
                               116400
                                                                         CONTS2
                                                              BPL
                                                                         a$TKB, (SP)
                               116374
177600
                                                                                              REPLACE OVER OLD CHARACTER ON STACK
                                                                                              STRIP ALL BUT ASCII
                                                               BIC
                                                               IF (SP) EQ #21
                                                               JMP
                                                                         CONTS1
                     000137
                               063576
                                                              BR CONTS2
END ; OF IF (SP)
                     000762
```

15559 15560					*THIS	ROUTINE	WILL INPUT A SIN	GLE CHARACTER FROM THE TTY
15559 15560 15561 15562 15563 15564 15565					*CALL:	RDCHR RETURN	HERE	::INPUT A SINGLE CHARACTER FROM THE TTY ::CHARACTER IS ON THE STACK ::WITH PARITY BIT STRIPPED OFF
15566 15567	064256 064260	011646	000004	000002	SRDCHR:	MOV	(SP),-(SP) 4(SP),2(SP)	:: PUSH DOWN THE PC
15569 15570	064266	016666	000004 116334	000002	15:	TSTB	astks	;;WAIT FOR
15571 15572 15573	064274 064302 064310	117766 042766 026627	116330 177600 000004	000004 000004 000023		BPL BIC CMP	astkB,4(SP) #^C<177>,4(SP) 4(SP),#23	::READ THE TTY ::GET RID OF JUNK IF ANY ::IS IT A CONTROL-S?
15574	064316	001013	116302		2\$:	BNE	3\$ astks	;;BRANCH IF NO ;;WAIT FOR A CHARACTER
15576	064324	100375				BPL	2\$ astka(SP)	:: LOOP UNTIL ITS THERE
15578 15579 15580	064266 064272 064274 064302 064316 064316 064324 064326 064332 064336 064344 064366 064366 064364 064366 064374	105777 100375 117766 042766 026627 001013 105777 100375 117746 042716 022627 061366 000750 026627 001744 026627 002407 026627 003003	116276 177600 000021			BIC CMP BNE	a\$TKB,-(SP) #^C177,(SP) (SP)+,#21 2\$ 1\$::PUSH DOWN THE PC ::SAVE THE PS ::WAIT FOR ::A CHARACTER ::READ THE TTY ::GET RID OF JUNK IF ANY ::IS IT A CONTROL-S? ::BRANCH IF NO ::WAIT FOR A CHARACTER ::LOOP UNTIL ITS THERE ::GET CHARACTER ::MAKE IT 7-BIT ASCII ::IS IT A CONTROL-Q? ::IF NOT DISCARD IT ::YES, RESUME ::IS IT A RANDOM CONTROL-Q? :R-C ::BRANCH BACK IF SO :R-C ::BRANCH IF YES ::IS IT A SPECIAL CHAR? ::BRANCH IF YES ::MAKE IT UPPER CASE ::GO BACK TO USER
15581 15582	064344	000750	000004	000021	3\$:	BR CMP	4(SP),#21	:: IS IT A RANDOM CONTROL-Q? :R-C
15580 15581 15582 15583 15584 15585	064354	001744	000004	000140		BEQ	1\$ 4(SP),#140	;:BRANCH BACK IF SO ;R-C ;:IS IT UPPER CASE?
12280	064364	002407	000004	000175		BLT	4(SP) ,#175	:: IS IT A SPECIAL CHAR?
15587 15588	064374 064376	042766	000040	000004		BGT BIC RTI	4\$ #40,4(SP)	; BRANCH IF YES ; MAKE IT UPPER CASE
15589 15590	064404	000002			45:	****	********	;;GO BACK TO USER
15591 15592					;*THIS		MILL INLAI V 214	ING FRUM INE III
15593 15594						RDLIN	HERE	INPUT A STRING FROM THE TTY ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK TERMINATOR WILL BE A BYTE OF ALL O'S SAVE R3 CLEAR THE RUBOUT KEY GET ADDRESS BUFFER FULL? BR IF YES GO READ ONE CHARACTER FROM THE TTY GET CHARACTER IS IT A CONTROL-C? BRANCH IF NO TYPE A CONTROL-C (^C) CLEAN RUBOUT KEY OFF OF THE STACK RESTORE R3 IS THERE A HALT FLAG SET IN THE SWR? BRANCH IF NOT TO BOOT ROUTINE
15596 15597	064406	010346			\$RDLIN:	CLR	R3,-(SP) -(SP)	CLEAR THE RUBOUT KEY
15598 15599	064412	005046 012703 022703 101477	064704 064730		1\$: 2\$:	MOV	#STTYIN,R3 #STTYIN+20.,R3	::GET ADDRESS ::BUFFER FULL?
15600 15601	064422	101477				BLOS RDCHR	8\$:: BR IF YES :: GO READ ONE CHARACTER FROM THE TTY
15602 15603	064426	104411 112613 122713 001016	000003			MOVB	(SP)+,(R3) #3,(R3)	;;GET CHARACTER ;;IS IT A CONTROL-C?
15604 15605	064434					BNE	3\$ \$CNTLC	::BRANCH IF NO ::TYPE A CONTROL-C (^C)
15606 15607	064442	005726				TST	(SP)+ (SP)+,R3	:: CLEAN RUBOUT KEY OFF OF THE STACK :: RESTORE R3
15608	064406 064412 064416 064416 064422 064426 064436 064436 064436 064446 064446 064466 064466 064472 064476	032777	000400	116146		BIT	#BIT8, aSWR	:: IS THERE A HALT FLAG SET IN THE SWR? :: BRANCH IF NOT TO BOOT ROUTINE
15610	064456	005037	002414			CLR	STOPOK EXIT	GET READY TO HALT PROGRAM GO HALT PROGRAM GOTO CONTROL-C RESTART
15612	064466	000137	002414 047632 047524 000177		11\$: 3\$:	JMP CMPB	BOOT #177, (R3)	::GOTO CONTROL-C RESTART
15614 15615	064476	005726 012603 032777 001404 005037 000137 000137 122713 001022 005716	000111			BNE	5\$ (SP)	:: IS IT A RUBOUT :: BR IF NO :: IS THIS THE FIRST RUBOUT?

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 486-1 CONTROL S & CONTROL Q

10 mg 30 mg	064502					BNE	45	::BR IF NO
15617	064502 064512 064516 064516 064522 064524 064530 064536 064536 064546 064546 064550 064556 064562 064572 064576 064576 064670 064604 064604 064604 064610 064614 064622 064626	001007 112737	000134	064702		MOVB	10\$	FOR IF NO
15619	064516	012716	177777			MOV	#-1,(SP)	SET THE RUBOUT KEY
15620 15621	064522	012716 005303 020327 103434 1 11337	064704		48:	DEC	R3 #STTYIN	::SET THE RUBOUT KEY ::BACKUP BY ONE ::STACK EMPTY? ::BR IF YES
15622	064530	103434	654702			BLO MOVB TYPE	8\$ (R3) .10\$:: SETUP TO TYPEOUT THE DELETED CHAR.
15624	064536	000725				TYPE	10\$	GO TYPE
15626	064544	000725 005716 001406 112737			55:	BR TST	8\$ (R3),10\$ 10\$ 2\$ (SP) 6\$ #'10\$::GO TYPE ::GO READ ANOTHER CHAR. ::RUBOUT KEY SET? ::BR IF NO ::TYPE A BACK SLASH
15627	064546	001406	000134	064702		MOVB TYPE	65 #'_10\$::TYPE A BACK SLASH
15629	064556	005016				TYPE	10\$ (SP)	CLEAR THE RURGUT KEY
15631	064564	005016	000025		6\$:	CMPB	#25, (R3)	:: IS CHARACTER A LTRL U?
15632 15633	064570	001003				BNE	7S SCNTLU	TYPE A CONTROL 'U''
15634	064576	000705	000022		75:	BR CMPB	1\$ #22,(R3)	::CLEAR THE RUBOUT KEY :IS CHARACTER A CTRL U? :BR IF NO :TYPE A CONTROL 'U' :GO START OVER :IS CHARACTER A "AR"? :BRANCH IF NO :CLEAR THE CHARACTER :TYPE A "CR" & "LF" :TYPE THE INPUT STRING :GO PICKUP ANOTHER CHACTER :TYPE A "?" :CLEAR THE BUFFER AND LOOP
15636	064604	001011	000022			BNE	9\$::BRANCH IF NO
15637 15638	064610	105013				CLRB	9\$ (R3) \$CRLF \$TTYIN 2\$	TYPE A "CR" & "LF"
15639	064614	000676				TYPE	STTYIN	:: TYPE THE INPUT STRING :: GO PICKUP ANOTHER CHACTER
15641	064622	000070			85:	TYPE	SQUES	TYPE A '?'
15643	064630	000671 111337	064702		9\$:	MOV8 TYPE	(R3),10\$::TYPE A ?? ::CLEAR THE BUFFER AND LOOP ::ECHO THE CHARACTER
15644 15645	064614 064620 064626 064630 064634 064644 064646 064656 064656 064662 064662 064664	122723	000015			CMPB	10\$ #15,(R3)+	:: CHECK FOR RETURN
15646	064644	122723 001264 105063	177777			CMPB BNE CLRB TYPE	2\$ -1(R3)	::LOOP IF NOT RETURN
15648	064652	005724				TYPE	\$LF (SP)+ (SP)+,R3 (SP),-(SP) 4(SP),2(SP) #\$TTYIN,4(SP)	TYPE A LINE FEED
15650	064660	012603				MOV MOV	(SP)+,R3	RESTORE R3
15651	064662	011646	000004	000002 000004		MOV	(SP),-(SP) 4(SP),2(SP)	:: ADJUST THE STACK AND PUT ADDRESS OF THE :: FIRST ASCII CHARACTER ON IT
15653	064672	012766	064704	000004		MOV	#STTYIN,4(SP)	;;RETURN
15655	064702	000002 000 000			10\$:	.BYTE	0	::STORAGE FOR ASCII CHAR. TO TYPE
15656 15657	064703 064704	000024			STTYIN:	.BYTE	Ŏ 20.	:: TERMINATOR :: RESERVE SIZE BYTES FOR TTY INPUT :: CONTROL 'C'
15660	064730	136	103	015	SCNTLC:	.ASCIZ	/*C/<15><12>	;; CONTROL "C"
15661	064735	136	103 000 125 000 107	015	SCNTLU:	.ASCIZ	/^U/<15><12>	;;CONTROL 'U''
15662	064742	012 136 012	107	015	SCNTLG:	.ASCIZ	/*G/<15><12>	;;CONTROL 'G'
15663	064745 064747 064752	015 127 075	012 122	123 040	\$MSWR:	.ASCIZ	<15><12>/SWR =	/
15664	064755 064760 064763 064766	075 040 105 075	000 012 122 040 040 127 040	000 116 040 000	SMNEW:	.ASCIZ	/ NEW = /	
15665	004700	0,7	040	000		.EVEN		

15667 15668 15669 15670 15671 15672 15673 15674 15675				****** *THIS I *CHANGI *THE II *OCTAL *FOLLO *THEN I	ROUTINE E IT TO NPUT CHA DIGITS.	WILL READ AN OCT BINARY. RACTERS WILL BE IF AN ILLEGAL (CHECKED TO INSURED THEY ARE LEGAL CHARACTER IS READ A "?" WILL BE TYPED W-LINE FEED. THE COMPLETE NUMBER MUST STERMINATED BY TYPING A CARRIAGE RETURN. :: READ AN OCTAL NUMBER
15677 15678	011646 016666			*	RDOCT RETURN MOV	HERE (SP),-(SP)	TERMINATED BY TYPING A CARRIAGE RETURN. ;:READ AN OCTAL NUMBER ;:LOW ORDER BITS ARE ON TOP OF THE STACK ;:HIGH ORDER BITS ARE IN SHIOCT ;:PROVIDE SPACE FOR THE ;:INPUT NUMBER ;:READ AN ASCIZ LINE
15680 064774 15681 065002		000004	000002		MOV PUSH RDLIN	4(SP),2(SP) RO,R1,R2	;;INPUT NUMBER
15679 064772 15680 064774 15681 065002 15682 065010 15683 065012 15684 065014 15685 065020 15686 065022 15687 065024 15688 065026 15689 065030 15690 065034 15691 065036 15692 065042 15693 065044 15694 065046 15695 065050 15696 065052 15697 065054	104412 012600 010037 005001	065120		1\$:	RDLIN MOV CLR CLR	(SP)+,R0 R0,5\$ R1	GET ADDRESS OF 1ST CHARACTER GRAND SAVE IT
15687 065024 15688 065026 15689 065030	005001 005002 112046 001420 122716 003026 122716 002423 006301	000060		2\$:	MOVB BEQ CMPB	(RO)+,-(SP) 3\$ #'0,(SP) 4\$ #'7,(SP)	::PICKUP THIS CHARACTER ::IF ZERO GET OUT ::MAKE SURE THIS CHARACTER ::IS AN OCTAL DIGIT
15690 065034 15691 065036	003026 122716	000067			BGT CMPB	W.('(2b)	;; IS AN OCTAL DIGIT
15692 065042 15693 065044	002423				ASL	45 R1	::*2
15694 065046 15695 065050	006102 006301 006102 006301 006102 042716				ROL ASL	R2 R1	::*4
15696 065052 15697 065054 15698 065056	006301				ROL ASL ROL	R2 R1 R2	::*8
15699 065060 15700 065064	042716 062601 000756	177770			BIC ADD BR	#^C7,(SP) (SP)+,R1	::STRIP THE ASCII JUNK ::ADD IN THIS DIGIT ::LOOP ::CLEAN TERMINATOR FROM STACK
15702 065070 15703 065072	005726 010166 010237	000012 065140		3\$:	TST MOV MOV POP	2\$ (SP)+ R1.12(SP) R2,\$HIOCT R2,R1,R0	CLEAN TERMINATOR FROM STACK
15704 065076 15705 065102 15706 065110 15707 065112 15708 065114 15709 065116 15710 065120 15711 065122 15712 065126 15713 065132 15714 065136 15715 065140	000002 005726 105010			45:	RTI TST CLRB TYPE	(SP)+ (R0)	;;RETURN ;;CLEAN PARTIAL FROM STACK ;;SET A TERMINATOR ;;TYPE UP THRU THE BAD CHAR.
15708 065114 15709 065116 15710 065120 15711 065122 15712 065126 15713 065132 15714 065136 15715 065140	000000			5\$:	.WORD TYPE TYPE	0 MSG062 MSG063	INPUT MUST BE A
15713 065132 15714 065136	000724				TYPE	MSG064	; NUMBER ; TRY AGAIN
15715 065140 15716 15717	000000			\$HIOCT:		0	HIGH ORDER BITS GO HERE DECIMAL NUMBER FROM THE TTY
15718 15719 15720 15721 15722 15723				**CHANG **ARE R	EAD A "	BINARY. IF TOO I ""FOLLOWED BY A NUMBER MUST BE	IMAL (ASCII) NUMBER FROM THE TTY AND MANY CHARACTERS OR ANY ILLEGAL CHARACTERS CARRIAGE RETURN-LINE FEED WILL BE TYPED. RETYPED. THE INPUT IS TERMINATED BY THE N. THE RANGE OF THE INPUT NUMBER IS

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 487-1 ROUTINE READ A DECIMAL NUMBER FROM THE TTY

15724					:*POSIT		7 TO NEGATIVE 3	
15725 15726 15727 15728	065142 065144 065152 065160 065162 065164 065170 065174 065202 065204 065204 065214 065216 065214 065224 065230 065232 065234 065234 065242				*	RDDEC RETURN	HERE	;; READ A DECIMAL NUMBER ;; NUMBER IS ON TOP OF THE STACK
15730 15731 15732	065142 065144 065153	011646 016666	000004	000002	\$RDDEC:	MOV MOV PUSH	(SP),-(SP) 4(SP),2(SP) RO,R1,R2	;:PROVIDE SPACE FOR ;:THE INPUT NUMBER
15733	065160 065162	104412			1\$:	RDLIN		;;READ AN ASCIZ LINE ;;ADDRESS OF 1ST CHAR.
15735 15736	065164 065170	010037 005046	065310			MOV	(SP)+,R0 R0,6\$ -(SP)	;:SAVE INCASE OF BAD INPUT ;:CLEAR DATA WORD
15737 15738	065172	122710	000055			CLR CMPB	R2 #'-,(R0)	::SIGN SET PUSITIVE ::SEE IF A MINUS SIGN WAS TYPED ::BP IF NO MINUS SIGN
15740 15741	065202 065204	112002			2\$:	BNE MOVB MOVB	(RO)+,R2 (RO)+,R1	:: SAVE FOR LATER USE :: PICKUP THIS CHARACTER
15742 15743	065206 065210	104412 012600 010037 005046 005002 122710 001001 112002 112001 001424 122701 003032 122701 002427 032716	000060			BEQ CMPB	2\$ (R0)+,R2 (R0)+,R1 3\$ #'0,R1	::READ AN ASCIZ LINE ::ADDRESS OF 1ST CHAR. ::SAVE INCASE OF BAD INPUT ::CLEAR DATA WORD ::SIGN SET POSITIVE ::SEE IF A MINUS SIGN WAS TYPED ::BR IF NO MINUS SIGN ::SAVE FOR LATER USE ::PICKUP THIS CHARACTER ::GET OUT IF ZERO ::MAKE SURE THIS CHARACTER ::IS A DIGIT BETWEEN 0 & 9
15744 15745 15746	065214 065216 065222	003032 122701 002427	000071			BGT CMPB BLT	5\$ #'9,R1 5\$	
15747 15748	065224 065230	032716 001024 006316	170000			BIT	#^(7777,(SP) 5\$ (SP)	::DON'T LET NUMBER GET TO BIG ::BR IF NUMBER WOULD OVERFLOW
15749 15750	065232 065234	006316 011646 006316				ASL MOV ASL	(SP) (SP),-(SP) (SP)	SAVE FOR LATER
15752 15753	065240 065242	006316 062616				ASL	(SP) (SP)+,(SP)	**************************************
15754 15755	065244 065246	102416	000060			BVS SUB	5\$ #'0.R1	:: *10. ::OVERFLOW ISN'T ALLOWED ::STRIP AWAY THE ASCII JUNK ::ADD IN THIS DIGIT ::OVERFLOW ISN'T ALLOWED ::LOOP ::CHECK IF NUMBER IS NEG
15757 15758	065252 065254 065256 065260 065262	162701 060116 102412 000752				ADD BVS BR	R1,(SP) 5\$ 2\$::OVERFLOW ISN'T ALLOWED
15759 15760	065260 065262	005/02			3\$:	TST BEQ	2\$ R2 4\$	EDR IF NO
15761 15762 15763	065264 065266 065272	005416 012666	000012		4\$:	NEG MOV POP	(SP) (SP)+,12(SP) R2,R1,R0	::YESNEGATE THE NUMBER ::SAVE THE RESULT
15764 15765	065300	000002				RTI		;;RETURN
15766 15767	065302	005726 105010			5\$:	CLRB	(SP)+ (R0)	;; CLEAN PARTIAL NUMBER FROM STACK ;; SET A TERMINATOR ; TYPE THE INDUIT UP TO BAD CHAP
15769 15770 15771	065262 065264 065266 065272 065300 065304 065306 065310 065312 065316 065322 065326	000000			6\$:	TYPE .WORD TYPE TYPE	0 MSG062 MSG065	::CLEAN PARTIAL NUMBER FROM STACK ::SET A TERMINATOR ::TYPE THE INPUT UP TO BAD CHAR. ::POINTER GOES HERE :INPUT MUSST BE A :DECIMAL :NUMBER
15772 15773	065322	000714				BR	MSG064 1\$;;TRY AGAIN

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 488
ROUTINE SAVE AND RESTORE RO-R5
                                              .SBTTL ROUTINE SAVE AND RESTORE RO-R5
 15775
15776
15777
                                                               *********
                                       : *SAVE RO-R5
  15778
                                       : *CALL:
                                              SAVREG
                                       *UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
                                       : *TOP---(+16)
                                       : + +2---(+18)
: + +4---R5
                                       : * +6---R4
                                       : * +8---R3
                                       : *+12---R1
                                       : *+14---RO
  15790
```

15790 15791 15792 065330 15793 065330 15794 065344 15795 065350 15796 065354 15797 065360 **\$SAVREG:** RO,R1,R2,R3,R4,R5 22(SP),-(SP) 22(SP),-(SP) 22(SP),-(SP) 22(SP),-(SP) PUSH ::SAVE PS OF MAIN FLOW ::SAVE PC OF MAIN FLOW ::SAVE PS OF CALL ::SAVE PC OF CALL 016646 016646 016646 016646 000002 000022 000022 000022 MOV MOV MOV 15798 065364 15799 15800 :*RESTORE RO-R5 15800 15801 15802 15803 065366 15804 065366 15805 065372 15806 065376 15807 065402 15808 065406 15809 065422 :*CALL: . * RESREG SRESREG: (SP)+,22(SP) ;:RESTORE PC OF CALL (SP)+,22(SP) ;:RESTORE PS OF CALL (SP)+,22(SP) ;:RESTORE PC OF MAIN FLOW (SP)+,22(SP) ;:RESTORE PS OF MAIN FLOW R5,R4,R3,R2,R1,R0 012666 012666 012666 012666 000022 000022 000022

MOV MOV MOV POP

000002

15811		.SBTTL	ROUTINE RANDOM	NUMBER GENERATOR
15813 15814 15815	******* *THIS R *WITH A	OUTINE I	IS A DOUBLE PRECOPE OF 0 TO 2**(+33)	ISION PSEUDO RANDOM NUMBER GENERATOR
15811 15812 15813 15814 15815 15816 15817 15818 15819 15820 15821 15822 065424 15823 065432 013700 002570	*CALL:	CALL RETURN	\$RAND	::CALL THE ROUTINE ::RETURN HERE THE RANDOM ::NUMBER WILL BE IN ::SHINUM, \$LONUM
15822 065424 15823 065432 013700 002570 15824 065436 013701 002566 15825 065442 012702 000007 15826 065446 006300	1\$:	PUSH MOV MOV ASL ROL	RO,R1,R2 SEEDLO,RO SEEDHI,R1 #7,R2 RO	SET RO WITH LOW SET R1 WITH HIGH SET SHIFT COUNT SHIFT RO LEFT AND R1 AND
15823 065432 013700 002570 15824 065436 013701 002566 15825 065442 012702 000007 15826 065446 006300 15827 065450 006101 15828 065452 077203 15829 065454 063700 002570 15830 065460 005501 15831 065462 063701 002566 15832 065466 062700 001057 15833 065472 005501 15834 065474 062701 047401 15835 065500 010037 002570 15836 065504 010137 002566		SOB ADD ADC ADD ADD ADC ADD MOV	R2,15 SEEDLO,RO R1 SEEDHI,R1 #1057,R0 R1 #47401.R1	:ADD NUMBER TO MAKE X 129 ::PROPOGATE CARRY :ADD NUMBER TO MAKE X 129 ::ADD LOW CONSTANT ::PROPOGATE CARRY ::ADD HIGH CONSTANT ::SAVE RO
15835 065500 010037 002570 15836 065504 010137 002566 15837 065510 15838 065516 000207		MOV POP RETURN	RO, SEEDLO R1, SEEDHI R2, R1, RO	SAVE R1

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 491 ROUTINE DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT

15841 15842 15843 15844	**************************************	******	LENGTH BINARY TO OCTAL ASCII CONVERT 2-BIT UNSIGNED BINARY NUMBER TO AN
15843 15844 15845 15846 15847 15848	*CALL * MOV * CALL * RETURN	#PNTR,-(SP) \$DB20	;; THE ADDRESS UP THE FIRST ASCIZ CHAR. IS ON THE STACK
15850 15851 065520 104415 15852 065522 016601 000002 15853 065526 012705 065637 15854 065532 012704 000014 15855 065536 012703 177770 15856 065542 012100 15857 065544 012101 15858 065546 005002 15859 065550 110245 15860 065552 010002 15861 065554 005304 15862 065556 003007 15863 065560 001405 15864 065562 005205 15865 065564 010566 000002 15866 065570 104416 15867 065572 000207 15868 065574 006203 15869 065576 006001 15871 065602 006001 15872 065604 006000 15873 065606 006001 15874 065610 006000 15875 065612 040302 15876 065614 062702 000060 15877 065620 000753	\$DB20: SAVREG MOV MOV MOV MOV MOV CLR 1\$: MOVB MOV DEC BGT BE9	2(SP),R1 #\$0CTVL+13.,R5 #12.,R4 #^C7,R3 (R1)+,R0 (R1)+,R1 R2 R2,-(R5) R0,R2 R4 3\$ 2\$ R5,2(SP)	: SAVE ALL REGISTERS : PICKUP THE POINTER TO LOW WORD : POINTER TO DATA TABLE : DO ELEVEN CHARACTERS : MASK : LOWER WORD : HIGH WORD : TERMINATOR : PUT CHARACTER IN DATA TABLE : GET THIS DIGIT : COUNT THIS CHARACTER : BR IF NOT THE LAST DIGIT : BR IF IT IS THE LAST DIGIT : ALL DIGITS DONE-ADJUST POINTER FOR FIRST : ASCIZ CHAR. & PUT IT ON THE STACK : RESTORE ALL REGISTERS : RETURN TO USER : POSITION THE MASK FOR THE LAST DIGIT : POSITION THE BINARY NUMBER FOR THE NEXT OCTAL DIGIT
15864 065562 005205 15865 065564 010566 000002 15866 065570 104416 15867 065572 000207 15868 065574 006203 15869 065576 006001 15870 065600 006000 15871 065602 006001 15872 065604 006000 15873 065606 006001 15874 065610 006000	INC MOV RESREG RETURN 2\$: ASR 3\$: ROR ROR ROR ROR ROR ROR ROR	R5 R5,2(SP) R3 R1 R0 R1 R0 R1 R0	
15875 065612 040302 15876 065614 062702 000060 15877 065620 000753 15878 065622 000016 15881 065626	BIC ADD BR SOCTVL: REPT SOCT8=SOCTVL+4	R3.R2 #'0.R2 1\$ 14.	;; MASK OUT ALL JUNK ;: MAKE THIS CHAR. ASCII :: GO PUT IT IN THE DATA TABLE ;: RESERVE DATA TABLE ; POINTER TO 11 DIGIT NUMBER

```
.SBTTL TABLES
15883
15884
15885
                                                                                                                 .SBTTL APT MAILBOX-ETABLE
15885
15886 065640
15887 065640
15888 065642
15889 065644
15890 065650
15892 065652
15893 065654
15894 065656
15895 065660
15896 065660
15897
15898
15899
15900 065661
                                                                                                                                                      ::MESSAGE TYPE CODE
::FATAL ERROR NUMBER (ERROR PC)
::TEST PATTERN NUMBER
::PASS COUNT
::DEVICE COUNT
::I/O UNIT NUMBER
                                                                                             SMSGTY: . WORD
                                  000000
                                 SFATAL: . WORD
                                                                                             STESTN: . WORD
                                                                                              SPASS: . WORD
                                                                                              SDEVCT: . WORD
                                                                                             SUNIT: . WORD
                                                                                                                                                        MESSAGE ADDRESS
                                                                                             $MSGAD: .WORD
$MSGLG: .WORD
                                                                                                                                                        ; APT ENVIRONMENT TABLE
                                                                                              SETABLE:
                                                                                                               BYTE O : ENVIRONMENT BYTE : SET TO A 1 FOR APT AUTO MODE IF BIT #7 IS SET IN SENVM THE TABLE BELOW (BEGINNING AT SMAMS1 AND ENDING AT SMADR4) MUST BE FILLED IN TO INDICATE THE PROPER AMOUNT OF EACH TYPE OF MEMORY.
                                                                                             SENV:
                                                                                              :NOTE:
                                                                                                                                                        ;ENVIRONMENT MODE
;BIT7(200)=USE APT SIZE INFO ;BIT5(40)=NO CONSOLE
                                                                                             SENVM: .BYTE
 15900 065661
15901
15902 065662
15903 065664
15904 065666
15905
15906
                                                                                                                                                       ;BIT7(200)=USE APT SIZE INFO ;BIT5(40)=NO CONSOLE

;;APT SWITCH REGISTER

;USED TO LIMIT THE NUMBER OF PASSES

;;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

··HIGH ADDRESS M.S. BYTE :DEFAULT = 64K
                                                                                             $SWREG: .WORD
$USWR: .WORD
                                                                                                                                     101
                                  000101
                                                                                                                                     00
                                                                                              SCPUOP: . WORD
                                  000000
 15907
 15908
 15909
 15910
                                                                                                                                                        ;;HIGH ADDRESS.M.S. BYTE ;DEFAULT = 64K
;;MEM. TYPE,BLK#1
 15911 065670
15912 065671
15913
                                                                                              $MAMS1: .BYTE
                                                                                              SMTYP1: .BYTE
                                                                                                                                                         MEM. TYPE BYTE -- (HIGH BYTE)
                                                                                                                                                                           900 NSEC CORE=001
300 NSEC BIPOLAR=002
PARITY MOS=003
  15914
  15915
  15916
                                                                                                                                                                            ERROR CORRECTING MOS=004
15917
15918 065672 177776
15919
15920 065674 000
15921 065675 000
15922 065676 000000
15923 065700 000
15924 065701 000
15925 065702 000000
15926 065704 000
15927 065705 000
15928 065706 000000
15929 065710 000000
15930 065712 000000
15931 065714 000000
15932 065716 000000
15933 15934 065720 000000
                                                                                                                                     177776 ;;HIGH ADDRESS,BLK#1
                                                                                              $MADR1: .WORD
                                                                                                                                                       MEM.LAST ADDR.=3 BYTES, THIS WORD AND LOW OF 'TYPE" ABOVE :: HIGH ADDRESS, M.S. BYTE :: MEM.TYPE, BLK#2 :: MEM.LAST ADDRESS, BLK#2
                                                                                              $MAMS2: .BYTE
                                                                                             SMAMS2: .BYTE
SMADR2: .WORD
SMAMS3: .BYTE
SMTYP3: .BYTE
SMADR3: .WORD
                                                                                                                                                        : MEM.LAST ADDRESS, BLK#2
: HIGH ADDRESS, M.S.BYTE
: MEM.TYPE, BLK#3
: MEM.LAST ADDRESS, BLK#3
: HIGH ADDRESS, M.S.BYTE
: MEM.TYPE, BLK#4
: MEM.LAST ADDRESS, BLK#4
: INTERRUPT VECTOR#1, BUS PRIORITY#1
: INTERRUPT VECTOR#2BUS PRIORITY#2
: PASE ADDRESS OF FOULTPMENT LINDER TE
                                                                                              SMAMS4: .BYTE
SMTYP4: .BYTE
                                                                                               SMADR4: . WORD
                                                                                                                 . WORD
                                                                                               SVECT1:
                                                                                               SVECT2: . WORD
                                                                                                                  . WORD
                                                                                                                                                          BASE ADDRESS OF EQUIPMENT UNDER TEST
                                                                                               SBASE:
                                                                                                                                                         :: DEVICE MAP
                                                                                               SDEVM:
                                   000000
               065720
065722
                                                                                                                   . WORD
                                                                                              SCDW1:
                                                                                              SCDW2:
                                                                                                                  . WORD
```

15937 15938 15939 15940 15941 15942	THE FOLLOWING LOCATIONS SPECIFY WHICH PATTERNS ARE TO BE RUN FOR PARTICULAR MEMORIES REFERENCE THE TABLE LISTED BELOW TO RELATE BITS TO PATTERNS. BITO SET WILL RUN THE FIRST ENTRY IN THE TABLE, BITO SET IN THE SECOND WORD WILL RUN THE 17TH ENTRY IN THE TABLE	
15944 15945 15946 065724 177777 15947 065726 177777 15948 065730 177777 15949 065732 177777 15950 065734 177777 15951 065736 177777	SDDWO: .WORD 177777 ;ECC CSR TESTS 177777 TABLE = MKCSRT: SDDW1: .WORD 177777 ;ECC CSR TESTS 177777 TABLE = MKCSRT: SDDW2: .WORD 177777 ;ECC PATTERNS 103777 TABLE = MKPAT: SDDW3: .WORD 177777 ;ECC PATTERNS 177777 TABLE = MKPAT: SDDW4: .WORD 177777 ;PARITY PATTERNS 003777 TABLE = MJPAT: SDDW5: .WORD 177777 ;PARITY PATTERNS 177774 TABLE = MJPAT: SDDW5: .WORD 177777 ;PARITY PATTERNS 177774 TABLE = MJPAT:	
15955 065740 15956 15957 15958 15959 065740 15960 065740 000000 15961 065742 065640 15962 065744 000043 15963 065746 001274 15964 065750 000000 15965 065752 000040	SETEND: ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC ;INTERFACE SPEC. SAPTHD: SHIBTS: .WORD 0 ;:TWO HIGH BITS OF 18 BIT MAILBOX ADDR. SMBADR: .WORD SMAIL ;:ADDRESS OF APT MAILBOX (BITS 0-15) STSTM: .WORD 35. ;:RUN TIM OF LONGEST TEST SPASTM: .WORD 700. ;:RUN TIME IN SECS. OF 1ST PASS ON 128K (QUICK VERIFY) SUNITM: .WORD 0. ;:EXTRA RUN TIME OF A PASS FOR EACH ADDITIONAL 128K (QV)	

11001211	111111 06							
15967 15968 15969 15970 15971 15972 15973					;***** ;*THIS I ;*AND U ;*OF THI ;*GO TO		POUTINE TRAP DE WILL PICKUP THE DINDEX THROUGH 1 ED ROUTINE. THEN DUTINE.	LOWER BYTE OF THE "TRAP" INSTRUCTION THE TRAP TABLE FOR THE STARTING ADDRESS USING THE ADDRESS OBTAINED IT WILL
15975 15976 15977 15978	065754 065756 065762 065764 065766 065770	010046 016600 005740 111000 006300 016000 000200	000002		STRAP:	MOV TST MOVB ASL	RO,-(SP) 2(SP),RO -(RO) (RO),RO RO \$TRPAD(RO),RO	::SAVE RO ::GET TRAP ADDRESS ::BACKUP BY 2 ::GET RIGHT BYTE OF TRAP ::POSITION FOR INDEXING ::INDEX TO TABLE ::GO TO ROUTINE
15980 15981 15982 15983 15984 15985	065774	000200	066016		::THIS	MOV RTS	TO HANDLE THE "GE	
15985 15986 15987 15988 15989	065776 066000 066006	011646 016666 000002	000004	000002	\$TRAP2:	MOV MOV RTI	(SP),-(SP) 4(SP),2(SP)	::MOVE THE PC DOWN ::MOVE THE PSW DOWN ::RESTORE THE PSW
15989 15990 15991	066010 066014	000000			SNOTRAP SHALT2:	TYPE HALT	MSG006	:UNDEFINED TRAP INSTRUCTION

```
15994
15995
15996
15997
                                                                                                                                                                                    .SBTTL TRAP TABLE
                                                                                                                                                   **THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED **BY THE 'TRAP' INSTRUCTION.
                                                                                                                                                                                    ROUTINE
  16000
                    066016
066020
066022
066024
066026
066030
066032
                                                    065776
056152
063110
063064
066010
063312
                                                                                                                                                   STRPAD: .WORD STRAP2 ; CALL=
                                                                                                                                                                                  $TYPE ; CALL=TYPEIT TRAP+1(104401) TTY TYPEOUT ROUTINE

$TYPOC ; CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)

$TYPOS ; CALL=TYPOS TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)

$NOTRAP; $TYPON ; CALL=TYPON TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)

$TYPOS ; CALL=TYPOS TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)

$NOTRAP; $TYPBN ; CALL=TYPBN TRAP+6(104406) TYPE BINARY (ASCII) NUMBER
  16001
16002
16003
16004
16005
16006
                                                       066010
16008
16009
16010
                                                                                                                                                                                                                                                                                  TRAP+7(104407) GET SOFT-SWR SETTING
                                                                                                                                                                                    SGTSWR ; CALL=GTSWR
                      066034
066036
                                                                                                                                                                                                                                                                                  TRAP+10(104410) TEST FOR CHANGE IN SOFT-SWR
                                                                                                                                                                                    SCKSWR : CALL=CKSWR
16011 066036
16011
16012 066040
16013 066042
16014 066044
16015 066046
                                                                                                                                                                                                                                                                                  TRAP+11(104411) TTY TYPEIN CHARACTER ROUTINE
TRAP+12(104412) TTY TYPEIN STRING ROUTINE
TRAP+13(104413) READ AN OCTAL NUMBER FROM TTY
                                                      064256
054406
064772
065142
                                                                                                                                                                                    SRDCHR
                                                                                                                                                                                                                   : CALL=RDCHR
                                                                                                                                                                                     SRDLIN
                                                                                                                                                                                                                   : CALL=RDLIN
                                                                                                                                                                                                                  ; CALL=RDOCT
; CALL=RDDEC
                                                                                                                                                                                      SRDOCT
                                                                                                                                                                                                                                                                                  TRAP+14(104414) READ A DECIMAL NUMBER FROM TTY
                                                                                                                                                                                     SRDDEC
 16016
                                                                                                                                                                                                                                                                                 TRAP+15(104415) SAVE RO-R5 ROUTINE
TRAP+16(104406) RESTORE RO-R5 ROUTINE
                                                                                                                                                                                    $SAVREG ; CALL=SAVREG
$RESREG ; CALL=RESREG
  16017 066050
16018 066052
  16019
16019
16020 066054 042644
16021 066056 042654
16022 066060 042664
16023
16024 066062 045100
16025
16026 066064 042674
16027 066066 042720
                                                                                                                                                                                   SKERNEL ; CALL=KERNEL TRAP+17(104417) ENTER KERNEL MODE

SENERGIZE; CALL=ENERGIZETRAP+20(104420) TURN ON MEMORY MANAGEMENT & TRAPS

SDEENERGI; CALL=DEENERGITRAP+21(104421) TURN OFF MEMORY MENAGEMENT & TRAPS
                                                                                                                                                                                                                                                                                   TRAP+22(104422) MAP KERNEL 1 TO 1
                                                                                                                                                                                     SKMAP
                                                                                                                                                                                                                        : CALL=KMAP
                                                                                                                                                                                                                                                                                  TRAP+23(104423) TURN CACHE ON TRAP+24(104424) TURN CACHE OFF
                                                                                                                                                                                      SCACHN ; CALL=CACHON
                                                                                                                                                                                     SCACHE
                                                                                                                                                                                                                    : CALL=CACHOFF
  16028
16029 066070
16030 066072
                                                                                                                                                                                                                                                                                  TRAP+25(104425) LOAD CORRECT CSR
TRAP+26(104426) READ CORRECT CSR
                                                                                                                                                                                    $LOADC
$READC
                                                   042736
043032
                                                                                                                                                                                                                    ; CALL=LOADCSR
  16030 066072
16031
16032 066074
16033 066076
16034 066100
16035 066102
16036 066104
16037 066106
16038 066110
16039 066112
16040 066114
16041 066116
16042 066120
16043 066122
16044 066124
16045 066126
16046 066130
16047 066132
16048 066134
16049 066136
16049 066136
                                                                                                                                                                                                                    : CALL=READCSR
                                                                                                                                                                                                                                                                                TRAP+27(104427) PROGRAM DETECTED ERROR TRAP+30(104430) PROGRAM DETECTED ERROR TRAP+31(104431) PROGRAM DETECTED ERROR TRAP+32(104432) PROGRAM DETECTED ERROR TRAP+33(104433) PROGRAM DETECTED ERROR TRAP+34(104434) PROGRAM DETECTED ERROR TRAP+35(104435) PROGRAM DETECTED ERROR TRAP+36(104436) PROGRAM DETECTED ERROR TRAP+37(104437) PROGRAM DETECTED ERROR TRAP+40(104440) PROGRAM DETECTED ERROR TRAP+41(104441) PROGRAM DETECTED ERROR TRAP+42(104442) PROGRAM DETECTED ERROR TRAP+43(104443) PROGRAM DETECTED ERROR TRAP+44(104444) PROGRAM DETECTED ERROR TRAP+45(104444) PROGRAM DETECTED ERROR TRAP+46(104446) PROGRAM DETECTED ERROR TRAP+47(104447) PROGRAM DETECTED ERROR TRAP+47(104447) PROGRAM DETECTED ERROR TRAP+50(104450) PROGRAM DETECTED ERROR TRAP+51(104451) PROGRAM DETECTED ERROR TRAP+51(104451)
                                                       056506
056534
056562
056612
056674
056716
056746
056766
                                                                                                                                                                                                                     : CALL=PERRO1
                                                                                                                                                                                      SPERO2
SPERO3
                                                                                                                                                                                                                     : CALL=PERRO2
                                                                                                                                                                                                                     : CALL=PERRO3
                                                                                                                                                                                                                   ;CALL=PERRO3;CALL=PERRO7;CALL=PERR10;CALL=PERR11;CALL=PERR12;CALL=PERR13
                                                                                                                                                                                      SPER04
SPER07
SPER10
                                                                                                                                                                                      SPER11
SPER12
SPER13
                                                                                                                                                                                                                    ; CALL=PERR14
                                                         057030
                                                                                                                                                                                       SPER14
                                                                                                                                                                                                                    : CALL=PERR15
                                                         057052
                                                                                                                                                                                        SPER15
                                                                                                                                                                                      SPER16
SPER17
                                                                                                                                                                                                                    :CALL=PERR16
:CALL=PERR17
                                                         057074
                                                        057114
057132
057150
                                                                                                                                                                                     SPER20
SPER21
SPER22
SPER23
SPER24
SPER25
                                                                                                                                                                                                                     : CALL=PERR20
                                                                                                                                                                                                                   :CALL=PERR21
:CALL=PERR22
:CALL=PERR23
:CALL=PERR24
:CALL=PERR25
                                                        057170
057206
057224
053762
```

```
PROGRAM DETECTED ERROR PROGRAM DETECTED ERROR
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              TRAP+52(104452)
TRAP+53(104453)
TRAP+54(104454)
TRAP+55(104455)
TRAP+56(104456)
TRAP+57(104457)
TRAP+60(104460)
TRAP+61(104461)
TRAP+63(104462)
TRAP+63(104463)
TRAP+64(104464)
TRAP+65(104465)
TRAP+66(104466)
TRAP+66(104467)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  SPER26
SPER27
SPER30
SPER31
                                                                                                                                                                          057414
057434
054210
057624
057722
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        : CALL=PERR26
: CALL=PERR27
16051 066142
16052 066144
16053 066146
16054 066150
16055 066152
16056 066154
16057 066156
16059 066162
16060 066164
16061 066166
16062 066170
16063 066172
16064 066174
16065
16066 066174
16067 066200
16068 066202
16069 066204
16071 066210
16072 066212
16073 066212
16073 066212
16074 066216
16075 066220
16076 066222
16077 066224
16078 066232
16080 066232
16081 066234
16080 066232
16081 066234
16082 066236
16083 066240
16084 066242
16085 066244
16086 066246
16087 066256
16089 066256
16089 066256
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          : CALL=PERR30
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            CALL=PERR30

CALL=PERR32

CALL=PERR33

CALL=PERR34

CALL=PERR35

CALL=PERR36

CALL=PERR37
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     $PER32
$PER33
                                                                                                                                                                          057722
057770
060050
060102
060136
066010
066010
066010
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    SPER34
SPER35
SPER36
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      SNOTRAP
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  SNOTRAP : CALL=PERR40
SNOTRAP : CALL=PERR41
SNOTRAP : CALL=PERR42
SNOTRAP : CALL=PERR43
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      SECCDIS; CALL=ECCDIS
SECCIDIS; CALL=ECCIDIS
SECCIDIS; CALL=ECCIDIS
SECCIDIS; CALL=ECCIDIS
SECCINIT; CALL=ECCIDIS
SECCINIT; CALL=ECCINIT
SECCINIT; CALL=CBCSR
SECINIT; CALL=CBCSR
SECION SECCINITION
SECURITION
SECURI
                                                                                                                                                                              043272
043304
043320
043360
043360
043402
043536
043566
043702
043732
043744
043754
043754
043754
043336
043052
044050
044100
044050
044100
066010
066010
066010
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    STSTRD ; CALL=TSTREAD
SINVALID; CALL=INVALID
SERRGEN ; CALL=ERRGEN
SCBREG ; CALL=CBREG
SNOTRAP
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           SNOTRAP
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          SNOTRAP
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          SNOTRAP
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          SNOTRAP
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          SNOTRAP
```

SEQ 0381

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 498 TRAP TABLE

16094 177776

ST = 177776

STATUS REGISTER

```
.SBTTL TABLE ERROR POINTER
  16097
 16098
16099
16100
                                                                                                                                            **THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.

**THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN

**LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.

**NOTE1: IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS (ERRPC).

**NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
  16101
  16102
  16103
 16104
16105
16106
16107
                                                                                                                                                                                                                                    ::POINTS TO THE ERROR MESSAGE
::POINTS TO THE DATA HEADER
::POINTS TO THE DATA
::POINTS TO THE DATA
                                                                                                                                            *
                                                                                                                                                                          DH
                                                                                                                                           .
                                                                                                                                                                          DT
                                                                                                                                                                          DF
  16108
16109
                                                                                                                                             :*
 16110
16111
16111 066262
16112 066262
16113 066264
16114 066266
16115 066270
16116
                                                                                                                                           SERRIB: ;ERROR 1
EM24
DH13
DT13
                                                    070750
073105
067306
067667
                                                                                                                                                                          DF11
                                                                                                                                                                           ERROR
                                                                                                                                                                                                      2
 16116
16117 066272
16118 066274
16119 066276
16120 066300
16121
16122 066302
16123 066304
16124 066306
16125 066310
16126
16127 066312
16128 066314
16129 066316
16130 066320
                                                    067735
072414
067132
067545
                                                                                                                                                                           DH1
                                                                                                                                                                           DT1
                                                                                                                                                                           DF2
                                                                                                                                                                         ERROR 3
                                                    067773
072474
067150
067662
                                                                                                                                                                           DH3
DT3
                                                                                                                                                                           DF9
                                                                                                                                                                          :ERROR 4
EM4
DH3
                      066312
066314
066316
066320
                                                    070025
072474
067160
067662
                                                                                                                                                                           DT4
DF9
 16130 066320

16131

16132 066322

16133 066324

16134 066326

16135 066330

16136

16137 066332

16138 066334

16139 066336

16140 066340

16141

16142 066342

16143 066344

16144 066346

16145 066350

16146

16147 066352

16148 066354

16149 066356

16149 066356
                                                                                                                                                                         :ERROR 5
EM5
DH5
DT5
DF2
                                                    070073
072530
067170
067545
                                                                                                                                                                         ERROR 6
EM6
DH5
DT5
DF2
                                                    070150
072530
067170
067545
                                                                                                                                                                            :ERROR 7
                                                    070175
072530
067170
067545
                                                                                                                                                                           EM7
                                                                                                                                                                           DH5
DT5
DF2
                                                                                                                                                                           ERROR
EM53
DH25
DT25
DF2
                                                                                                                                                                                                      10
                                                     072161
073627
067464
067545
```

ZMSPAO ABLE	MS11-L/ ERROR P	M/P MEMORY OINTER	DIAG.	MACRO	M1113	26-APR-82	09:41	PAGE	502
16153 16154 16155 16156 16157	066362 066364 066366 066370	070235 072654 067222 067571				ERROR EM11 DH7 DT7 DF3			
16158 16159 16160 16161 16162	066362 066366 066370 066372 066374 066376 066402 066404 066406 066410 066412 066414 066416 066420 066420 066424 066426 066430 066432 066434 066436 066436	070235 072654 067222 067604				ERROR EM11 DH7 DT7 DF4			
16163 16164 16165 16166 16167	066402 066404 066406 066410	070257 072764 067252 067545				ERROR EM12 DH10 DT10 DF2			
16168 16169 16170 16171 16172	066412 066414 066416 066420	070235 072654 067222 067617				ERROR EM11 DH7 DT7 DF5			
16173 16174 16175 16176	066422 066424 066426 066430	070235 072654 067222 067632				ERROR EM11 DH7 DT7 DF6	15		
16178 16179 16180 16181 16182	066432 066434 066436 066440	070303 073105 067306 067667				ERROR EM13 DH13 DT13 DF11	16		
16183 16184 16185 16186	066442 066444 066446	070335				ERROR EM14 DH13 DT13 DF11	17		
16188 16189 16190 16191	066452 066454 066456 066460	070401 073105 067306 067667				ERROR EM15 DH13 DT13 DF11	20		
16193 16194 16195 16196	066446 066450 066452 066454 066456 066460 066464 066466 066470	072210 073665 067476 067545				;ERROR EM55 DH26 DT26 DF2			
16198	066472	070447 072654 067222 067617				ERROR EM17 DH7 DT7 DF5			
16200 16201 16202 16203 16204 16205 16206	066502 066504 066506 066510	072030 073501 067422 067700				: ERROR EM50 DH23 DT23 DF13	23		

CZMSPAO M TABLE E	S11-L/M/P MEMORY	DIAG. MA	CRO M1113	26-APR-82	09:41	PAGE 504
16210 16211 0 16212 0 16213 0 16214 0	66512 070507 66514 073105 66516 067306 66520 067667			ERROR EM19 DH13 DT13 DF11	24	
16211 0 16212 0 16213 0 16214 0 16215 16216 0 16217 0 16218 0 16219 0	66522 070561 66524 073105 66526 067306 66530 067667			;ERROR EM20 DH13 DT13 DF11 ;ERROR	25	
16221 0 16222 0 16223 0 16224 0	66532 000000 66534 073100 66536 067302 66540 067545			0 DH12 DT12 DF2		;NO MESSAGE
16225 16226 0 16227 0 16228 0	66542 070640 66544 073062 66546 067274 66550 067545			ERROR EM21 DH11 DT11 DF2	27	
16230 16231 0 16232 0 16233 0	066552 070674 066554 073105 066556 067306			;ERROR EM22 DH13 DT13	30	
16235 16236 0 16237 0	066560 067667 066562 000000 066564 073202 066566 067330			DF11 :ERROR O DH14 DT14	31	;NO MESSAGE
16242 0 16243 0	066570 067545 066572 070721 066574 072530 066576 067170			DF2 ;ERROR EM23 DH5 DT5	32	
16244 0 16245 16246 0 16247 0 16248 0	066602 071027 066604 073261 066606 067346			DF2 ;ERROR EM25 DH15 DT16	33	
16250 16251 16252 16253	066610 067645 066612 071054 066614 073400 066616 067376 066620 067571			DF7 :ERROR EM26 DH16 DT17 DF3	34	

										-
CZMSPA0 TABLE	MS11-L/ ERROR P	M/P MEMORY POINTER	DIAG.	MACRO	M1113	26-APR-82	09:41	PAGE	506	
16257 16258 16259 16260	066622 066624 066626	072134 073627 067464				;ERROR EM52 DH25 DT25	35			
16261 16262 16263	066630	067545				DF2 ;ERROR EM27	36			

;ERROR EM35 DH7 DT7 DF3 071622 072654 067222 067571 :ERROR 40 EM29 DH7

071215 072654 067222 067571 DT7 DF3

071277 072654 067222 067617

DF3 :ERROR 41 EM30 DH7 DT7 DF5 :ERROR 42 EM60 DH20 DH23 DF13 :ERROR 43 EM32 DH7 DT7 DF3 :ERROR 44 EM33 DH7 DT7 072331 073422 067422 067700

071407 072654 067222 067571

071514 072654 067222 067571 DT7 DF3

DF3
PF3
072064 073560 067444 067710

071707 072607 067206 067545

CZMSPAO TABLE	MS11-L/I	M/P MEMORY DINTER	DIAG.	MACRO	M1113	26-APR-82	09:41	PAGE	508
16309 16310 16311 16312 16313	066742 066744 066746 066750 066752 066754 066756 066764 066766 066766 066770 066772 066774 066776 067000 067002 067004 067010 067012 067014 067016 067012 067020	071756 072451 067404 067545				;ERROR EM40 DH2 DT20 DF2 ;ERROR	50		
16314 16315 16316 16317 16318	066752 066754 066756 066760	072231 073703 067504 067544				EM56 DH27 DT27 DF1			
16329 16321 16322 16323	066762 066764 066766 066770	072373 073560 067444 067710				ERROR EM61 DH24 DT24 DF14	51		
16324 16325 16326 16327 16328	066772 066774 066776 067000	071215 073105 067306 067667				ERROR EM29 DH13 DT13 DF11			
16329 16330 16331 16332 16333	067002 067004 067006 067010	071027 073757 067524 067726				ERROR EM25 DH30 DT30 DF16	53		
16334 16335 16336 16337 16338	067012 067014 067016 067020	072263 073757 067524 067726				:ERROR EM57 DH30 DT30 DF16			
16339 16340 16341 16342 16343	067022 067024 067026 067030	070561 073560 067444 067710				ERROR EM20 DH24 DT24 DF14			
16344 16345 16346 16347 16348	067032 067034 067036 067040	070561 073757 067524 067726				ERROR EM20 DH30 DT30 DF16			
16349 16350 16351 16352 16353	067042 067044 067046 067050	070507 073560 067444 067710				;ERROR EM19 DH24 DT24 DF14	57		
16354 16355 16356 16357 16358	067026 067030 067032 067034 067036 067040 067042 067044 067046 067050 067052 067054 067056 067060 067060 067060	070303 073422 067422 067700				;ERROR EM13 DH20 DT23 DF13	60		
16359 16360 16361 16362	067062 067064 067066 067070	071277 073422 067422 067700				ERROR EM30 DH20 DT23 DF13	61		
16364 16365	067072	070561				ERROR EM20	62		

ı										M 14	
	CZMSPA0 TABLE	MS11-L	/M/P MEMORY POINTER	DIAG.	MACRO	M1113	26-APR-82	09:41	PAGE	508-1	
	16366 16367 16368	067074 067076 067100	073560 067444 067710				DH24 DT24 DF14 ;ERROR	63			
	16369 16370 16371 16372 16373	067102 067104 067106 067110	070674 073560 067444 067710				EM22 DH24 DT24 DF14	03			
	16374 16375 16376 16377	067112 067114 067116	100300 072530 067170				ERROR EM62 DH5 DT5	64			
	16378 16379 16380 16381 16382	067120 067122 067124 067126	070674 073422				DF2 :ERROR EM22 DH20 DT23	65			
	16383	067130					DF13				

RROR D	MS11-L/	(DT)	RY DIAG.	MACRO M	11113	20-APK-02	09:41 PAGE 509
16385 16386	067132	002016	002032	002042	DT1:	.SBTTL .WORD	ERROR DATA TAGS (DT) ERRPC, ADDRESS, GOOD, BAD, 0
16387 16388	067150	002016 002050 002016 002016	000000	002070	DT2: DT3:	.WORD	ERRPC, O ERRPC, PADDRESS, PARCNT, O
16389	067156 067160	000000 002016	002032	002066	DT4:	.WORD	ERRPC, ADDRESS, NEMCNT, 0
	067166 067170 067176	000000 002016 177576	177572 172516	177574 177766	DT5:	.WORD	ERRPC, MMRO, MMR1, MMR2, MMR3, CPUERR, 0
16391	067204 067206 067214	000000	002416 002376	002374	DT6:	.WORD	ERRPC, APTPAR, LSIZE, APTECC, MSIZE, 0
16392	067214 067222 067230	000000 002016 002420 002016 002174 002056	002376 002174 002042	000000 002032 002050	DT7:	.WORD	ERRPC, DUMMY, ADDRESS, DUMMY, GOOD, BAD, BADXOR
16393	067240	002036 002174 002174	002174	002174		. WORD	DUMMY, DUMMY, DUMMY, O
16394	067214 067222 067230 067236 067240 067246 067252 067260 067266 067274 067302 067306	002174 002176 002204 002212 002016	000000 002200 002206 002214	002202 002210 000000	DT10	. WORD	DETRO, DETR1, DETR2, DETR3, DETR4, DETR5, DETSP, DETPSW, O
16395	067274	002016	002146	000000	DITT	. WORD	ERRPC.CSR.O
	UU1 2 19	002146 002016 002174	000000 002174 002244	002032 002246	DT12 DT13	. WORD	CSR.O ERRPC, DUMMY, ADDRESS, DUMMY, TSTDAT, TSTDAT+2, CHECK, CSR, C
16398	067322 067330 067336 067344	002310 177746 177576	002146 177572 172516	000000 177574 177766	DT14	. WORD	CONTRL,MMRO,MMR1,MMR2,MMR3,CPUERR,O
16399	067346	000000 002016 002042	002174	002174	DT16	. WORD	ERRPC, DUMMY, DUMMY, GOOD, GOOD2, GOOD3
16400	067354 067362	002050	002052	002046		.WORD	BAD, BAD2, BAD3, DUMMY, DUMMY, O
	067370 067376 067404	002174 002016 002016	002174 002174 002042	000000 000000 002050	DT17 DT20		ERRPC, DUMMY, O ERRPC, GOOD, BAD, O
16403 16404	067412 067414 067422 067430 067436	002016 002016 002050	002174 002174 002174 002174 002174 002174	000000 002042 002174 000000 002146 002174	DT22	. WORD	ERRPC, DUMMY, GOOD, BAD, DUMMY, DUMMY, DUMMY, DUMMY, O
16405	067444	002174 002016 002174	002174 002174 002174	000000 002146 002174	DT24	: .WORD	ERRPC, DUMMY, CSR, DUMMY, DUMMY, DUMMY, DUMMY, O
16406	067460 067464 067472	002016	000000	002146	DT25	: .WORD	ERRPC,GOOD,CSR,CSRNO,O
16407 16408	067476 067504 067512	000000 002016 002016 002050 002174 002174 002174 002150 002016 00216 002174 002174	000000 002050 002174 002174	000000 002032 002174	DT26 DT27	. WORD	ERRPC, BAD, O ERRPC, DUMMY, ADDRESS, DUMMY,
16409	067520 067524 067532 067540	002174 002016 002042 002174	000000 002174 002050 000000	002174 002146	DT30	: .WORD	ERRPC, DUMMY, DUMMY, GOOD, BAD, CSR, DUMMY, O

CZMSPAO MS11-L/M/P ERROR DATA FORMATS		DIAG.	MACRO M	1113	26-APR-82	09:41	PAGE 511
16412 16413 067544 16414 067545 067550 067553 067556 067561	000 000 000 000 000 000 000	000 000 000 000 000	000 000 000 000 000 000	DF1: DF2:	.SBTTL .BYTE .BYTE	0	DATA FORMATS (DF) ,0,0,0,0,0,0,0,0,0,0,0,0,0,0
16415 067567 067571 067574 067577	010	000 005 000 003	000 000 006	DF3:	.BYTE	0.5.0	.8.,0,0,0,3,6,2,4
16416 067602 067604 067607 067612	002 000 010 010	004 005 000 003	000 010 006	DF4:	.BYTE	0.5.0	,8.,0,8.,8.,3,6,2,4
16417 067617 067622 067625	002 000 010 011	004 005 011 003 004	000 011 006	DF5:	.BYTE	0,5,0	.8.,9.,9.,9.,3.6,2,4
16418 067632 067635 067640	002 000 010 010	005 011 003	000 010 006	DF6:	.BYTE	0,5,0	,8.,9.,8.,8.,3,6,2,4
16419 067645 067650 067653 067656	002 000 000 000 002	004 005 000 000 004	010 011 011	DF7:	.BYTE	0.5.8	0.0.90.0.92.4
16420 067660 16421 067662	000	005 001 001	001	DF8: DF9:	BYTE.	0.5	.1.1
16422 067667 067672	001 000 010	005	000	DF11	: .BYTE	0,5,0	.8.,0,0,0,0,0
16423 067700 067703 067706 16424 067710	000 000 000 002	000 005 003 004 005 006	000 000 006	DF13	: .BYTE	0.5.0	.0.3.6.2.4
06//13	000	005 006	000	DF14	: .BYTE	0.5.0	.3.6.2.4
16425 067716 067717 067722	004 000 010	005 003	000 006	DF15	: .BYTE	0,5,0	.8.,3,6,4
067722 067725 067726 067731 067734	004 000 000 004	005 000	01 <u>0</u> 003	DF16	: .BYTE	0,5,8	0.0.3.4

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 513 ERROR MESSAGES (EM)

16429				.SBTTL	ERROR MESSAGES (EM)
16435 067735 16436 067773 16437 070025	103 120	101	116 EM2: 122 EM3: 116 EM4: 114 EM5:	.ASCIZ	/CAN'T SET 22 BIT MODE IN MMR3/
16436 067773 16437 070025	116	101	122 EM3: 116 EM4:	ASC17	/PARITY ERROP(S) IN BANK 0/
16438 070073	111	114	114 EM5:	ASCIZ	/NON-EXISTANT MEMORY (HOLES) IN BANK 0/ /ILLEGAL OR RESERVED INSTRUCTION (TRAP TO 10)/ /UNEXPECTED TRAP TO 4/
16438 070073 16439 070150	125	116	105 EM6: 115 EM7:	.ASCIZ	/UNEXPECTED TRAP TO 4/
16440 070175	125 115	105	115 EM7:	.ASCIZ	/MEMORY MANAGEMENT (TRAP TO 250)/
16441 070235	115	105	115 EM11:	ASCIZ	/MEMORY DATA ERROR/
16442 070257 16443 070303	104 115	105 111	124 EM12: 123 EM13: 111 EM14:	ASCIZ	MISSING EXPECTED SRE FLAG/
16444 070335	127	122	111 EM14:	ASCIZ	/WRITE BYTE FAILED TO CLEAR SBE FLAG/
16445 070401	127 106 115	101	111 EM15:	.ASCIZ	/FAILED TO GET INTERRUPT WITH DBE FLAG/
16446 070447	115	105	115 EM17:	.ASCIZ	MEMORY DATA ERROR IN CHECK BITS/
16447 070507	123	102	105 EM19:	ASCIZ	SPE-DE CAUSED PARTIT TRAP WHEN INTIDITED
16448 070561 16449 070640	153	102 102	105 EM20: 105 EM21:	ASCIZ	/SBE-DBE ON MASTER TEST WORD/
16450 070674	123 123 123 115	111	123 EM22: 105 EM23:	.ASCIZ	/MISSING EXPECTED DBE/
16451 070721	125	116	105 EM23:	.ASCIZ	/UNEXPECTED PARITY TRAP/
16452 070750	125 122 103 101	105	105 EM19: 105 EM20: 105 EM21: 123 EM22: 105 EM23: 103 EM24: 105 EM25:	.ASCIZ	/RECEIVED DBE FLAG WHEN EXPECTING UNLY SBE FLAG/
16453 071027 16454 071054	103	110 104	105 EM25:	ASC17	ADDRESS PARITY ERROR DID NOT CAUSE ABORT/
16455 071125	105	103	104 EM26: 103 EM27:	ASCIZ	/ECC INHIBIT MODE POINTER FAILURE - DID NOT PROTECT BANK/
16456 071215	103	117	122 EM29:	.ASCIZ	/CORRECTION FAILURE WITH ECC ENABLED ON FORCED SBE/
16457 071277 16458 071363	103 127 106 115	122	111 EM30:	.ASCII	/UNEXPECTED TRAP TO 4/ /MEMORY MANAGEMENT (TRAP TO 250)/ /MEMORY DATA ERROR/ /DETAILED ERROR DUMP/ /MISSING EXPECTED SBE FLAG/ /MENTE BYTE FAILED TO CLEAR SBE FLAG/ /MEMORY DATA ERROR IN CHECK BITS/ /SBE-DBE CAUSED PARITY TRAP WHEN INHIBITED/ /SBE-DBE DID NOT CAUSE PARITY TRAP WHEN ENABLED/ /SBE-DBE ON MASTER TEST WORD/ /MISSING EXPECTED DBE/ /UNEXPECTED PARITY TRAP/ /RECEIVED DBE FLAG WHEN EXPECTING ONLY SBE FLAG/ /CHECK BIT DATA ERROR/ /ADDRESS PARITY ERROR DID NOT CAUSE ABORT/ /ECC INHIBIT MODE POINTER FAILURE - DID NOT PROTECT BANK/ /CORRECTION FAILURE WITH ECC ENABLED ON FORCED SBE/ /WRITE BYTE WITH ECC ENABLED TAILED TO CLEAR DATA AT/ <crlf> /FORCED SBE LOCATION/ /MOVB #360, (R2)+ WITH ECC ENABLED CHANGED DATA AT FORCED DBE LOCATION/</crlf>
16458 071363	106		122 EM32.	ASCIZ	MOVE #360 (P2) + WITH ECC ENABLED CHANGED DATA AT FORCED DRE LOCATION/
16459 071407 16460 071514	115	117	126 EM32: 126 EM33:	ASCIZ	/MOV #177400 (R1) WITH ECC ENABLED CHANGED DATA AT FORCED DBE LOCATION/
16461 071622	125	116	105 EM35:	.ASCIZ	/UNEXPECTED CORRECTION WITH ECC DISABLE ON FORCED SBE/
16462 071707	125 101 102 102 106 102 102 111	120	124 EM36:	.ASCIZ	/MOVB #360 (R2) + WITH ECC ENABLED CHANGED DATA AT FORCED DBE LOCATION/ /MOV #177400 (R1) WITH ECC ENABLED CHANGED DATA AT FORCED DBE LOCATION/ /UNEXPECTED CORRECTION WITH ECC DISABLE ON FORCED SBE/ /APT SIZE DISAGREES WITH PROGRAM SIZING/
16463 071756 16464 072030	102	122	101 EM40:	.ASCIZ	/BRANCH GOBBLE FAILED CONDITION CODES TEST/
16464 0/2030	102	101 114	104 EM50: 101 EM51:	ASCIZ	/FLAGS NOT SET ON FORCED UNCORRECTED SBE/
16465 072064 16466 072134	102	111	124 EM52:	ASCIZ	/BIT SET ERROR IN CSR/
16667 072161	102	111	124 EM53:	.ASCIZ	/BIT CLEAR ERROR IN CSR/
16468 072210	111	114	114 EM55:	.ASCIZ	/BANCH GOBBLE FAILED CONDITION CODES TEST/ /BAD ERROR ADDRESS GENERATED/ /FLAGS NOT SET ON FORCED UNCORRECTED SBE/ /BIT SET ERROR IN CSR/ /BIT CLEAR ERROR IN CSR/ /ILLEGAL CSR TYPE/ /PAD PARITY TRAP GENERATED/
16469 072251	102	101	122 126 EM32: 126 EM33: 105 EM35: 124 EM36: 101 EM40: 104 EM50: 101 EM51: 124 EM52: 124 EM53: 114 EM55: 104 EM56: 117 EM57:	ASCIZ ASCIZ	/BAD PARITY TRAP GENERATED/ /WRONG CHECK BIT READ BACK FROM MEMORY/ /WRONG SYNDROME BITS READ INTO CSR/
16471 0722331	127	125	117 EM60:	.ASCIZ	/WRONG SYNDROME BITS READ INTO CSR/
16468 072210 16469 072231 16470 072263 16471 072331 16472 072373	102 127 127 103	122 122 123	122 EM61:	.ASCIZ	/CSR UPDATE ERROR/

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 515	CZMSPAO MS11-L/M/P	MEMORY DIAG	. MACRO M1113	26-APR-82 09:41	PAGE 515
--	--------------------	-------------	---------------	-----------------	----------

16475 16476 072414 16477 072451 16478 072474 16479 072530 16480 072607 16481 072654 16482 072720 16483 072764 16484 073062 16485 073100 16486 073105 16487 073162 16488 073202	040 040 040 040 040 040 040 040 040	040 040 040 040 040 040 040 103 040 103	120 DH1: 120 DH2: 120 DH3: 120 DH5: 120 DH6: 120 DH7: 040 122 DH10: 120 DH11: 123 DH12: 120 DH13:	.SBTTL ERROR DATA HEADERS (DH) .ASCIZ / PC DEV ADD GOOD BAD/ .ASCIZ / PC GD-CC BD-CC/ .ASCIZ / PC 1ST ADD # OF ERRORS/ .ASCIZ / PC MMRO MMR1 MMR2 MMR3 CPUERR/ .ASCIZ / PC APTPAR LSIZE APTECC MSIZE/ .ASCII / PC BANK VADD PADD GOOD/ .ASCIZ / BAD XOR CSR MTYP INT PAT/	
16482 072720 16483 072764	040	040	122 DH10:	ASCIZ / RO R1 R2 R3 R4 R5 SF	PSW/
16484 073062	040	040	120 DH11: 123 DH12:	.ASCIZ / PC CSR/	
16486 073100	040	040	120 DH13:	.ASCII / PC BANK VADD PADD WROTE1 WROTE2/	
16487 073162	040	103		ASCIZ / CHKBITS CSR/	Fig. 1. Law
16488 073202		11/	116 DH14:	.ASCIZ /CONTRL MMRO MMR1 MMR2 MMR3 CPUERR/ .ASCII / PC BANK PADD GD-WD1 GD-WD2 GD-CHK/	
16489 073261	040	102	120 DH15:	ASCIZ / BAD-WD1 BAD-WD2 BAD-CHK INT PAT/	
16491 073400	040	040	120 DH16:	.ASCIZ / PC BANK/	
16492 073415	040	040	120 DH19:	.ASCIZ / PC/	
16492 073415 16493 073422	040	040	120 DH20:	.ASCIZ / PC BANK GD-CSR (CSR) CSR MTYP INT PAT/	
16494 073501 16495 073560	040	040	120 DH23:	.ASCIZ / PC BANK GD-ERR BAD-ERR CSR MTYP INT PAT/ .ASCIZ / PC BANK (CSR) CSR MTYP INT PAT/	
16490 073336 16491 073400 16492 073415 16493 073422 16494 073501 16495 073560 16496 073627 16497 073665	040	040	120 DH24: 120 DH25:	.ASCIZ / PC BANK (CSR) CSR MTYP INT PAT/ .ASCIZ / PC GD-DAT (CSR) CSRNO/	
16497 073665	040	040	120 DH26:	ASCIZ / PC BADCODE/	
10498 0/3/03	040 040 040 040 040 040 040 040	040 040 040 040 040 040	120 DH27:	.ASCIZ / PC BANK VADD PADD CSR MTYP PAT/	
16499 073757	040	040	120 DH30:	.ASCIZ / PC BANK PADD WROTE READ CSR PAT/	

```
MSG001: ASCIZ
MSG002: ASCIZ
MSG003: ASCII
                                                                                                                                                                                                                                MESSAGES
16502
16503 074036
16504 074120
16505 074175
16506 074237
16507 074302
16508 074343
16509 074410
16510 074422
16511 074457
16512 074471
16513 074503
16514 074515
16515 074603
16516 074700
16517 074702
16518 074704
16519 074706
16520 074720
16521 074731
16522 074734
16523 074740
16524 074761
16525 075016
16526 075016
16527 075033
16528 075030
16529 075073
16530 075115
16531 075144
16532 075170
16533 075214
16534 075236
16535 075257
16536 075302
16537 075332
16538 075302
16539 075404
16540 075456
16540 075456
16541 075456
16542 075571
16543 075547
16546 075571
16546 075571
16547 075605
16557 076067
16558 076012
16557 076067
16558 076012
16557 076067
                                                                                                                                                                                                                                                                                                                                                               MEMORY CONFIGURATION MAP/
16K WORD BANKS/
2 3/
                                                                                             <CRLF>/
                                                                                                                                        040
040
040
040
062
116
116
122
040
066
                                                                                                                                                                                                                  <CRLF>/
                                                                                                                                                                                                .ASCIZ
                                                                                                                                                           MSG004: ASCII
ASCII
MSG005: ASCII
MSG006: ASCII
MSG007: ASCII
MSG009: ASCII
MSG009: ASCII
MSG010: ASCII
MSG011: ASCII
MSG012: ASCII
MSG013: ASCII
MSG014: ASCII
MSG015: BYTE
MSG016: ASCII
MSG017: ASCII
MSG018: ASCII
MSG019: ASCII
MSG019: ASCII
MSG019: ASCII
                                                                                                                                                             MSG004: .ASCI
                                                                                                                                                                                                                            .ASCII
                                                                                                                                                                                               ASCII
ASCII
ASCII
ASCII
ASCII
ASCII
ASCII
                                                                                                                                                                                                .ASCII
                                                                                                                                                                                                .ASCII
                                                                                                                                                                                                .ASCII
                                                                                                                                                                                               ASCII
ASCII
ASCII
ASCII
BYTE
                                                                                                                                                              MSG022:
MSG025:
MSG026:
MSG027:
MSG028:
MSG029:
MSG030:
MSG031:
MSG033:
MSG033:
                                                                                                                                                                                                ASCIZ
ASCIZ
ASCIZ
                                                                                                                                                                                                 ASCIZ
ASCIZ
ASCIZ
ASCIZ
ASCIZ
ASCIZ
                                                                                                                                            101
111
131
125
000
                                                                                                                                                                                                  .ASCI
                                                                                                                                                                 MSGA34:
                                                                                                                                                                                                  .ASCI
                                                                                                                                                                 MSGB34:
MSG035:
                                                                                                                                                                                                  .ASCIZ
```

```
CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 517-1
 MESSAGES
                                                                                                                                                                                                                                                                                                                                                                                                        <CRLF>/MODIFY MEMORY/
<CRLF>/OLD DATA WAS /
<CRLF>/DATA IS NOW /
<CRLF>/INPUT NEW DATA? /
<CRLF>/SELECT BANK & TEST/
<CRLF>/BANK NOT ACCESSABLE/
<CRLF>/TEST (0-47)? /
<CRLF>/TEST 0 DATA IS? /
<CRLF>/TO ESCAPE TYPE ANY KEY/<CRLF><12><12>
<CRLF>/TEST COMPLETE/
/ NOT AVAILABLE NOW - TRY LATER!/
<CRLF>/BANK REQUIRES RELOCATION/
                                                                                                                                                                                                                                                                                      MSG036: ASCIZ
MSG037: ASCIZ
MSG038: ASCIZ
MSG039: ASCIZ
MSG040: ASCIZ
MSG041: ASCIZ
MSG042: ASCIZ
MSG043: ASCIZ
MSG046: ASCIZ
MSG047: ASCIZ
MSG049: ASCIZ
MSG049: ASCIZ
               16559 076120
16560 076137
16561 076156
16562 076174
16563 076216
16564 076242
16565 076267
16566 076305
16567 076327
16568 076362
16569 076401
16570 076441
                                                                                                                                     117
104
111
123
102
124
124
124
124
116
                                                                                                                                                                                                                                                      101
                                                                                                                                                                                                                                                      116
105
101
105
105
117
                                                                                                                                                                                                                                                        105
                                                                                                                                                                                                                                                                                   MSG048: ASCIZ
MSG049: ASCIZ
EVEN
MSG055: ASCIZ
MSG056: ASCIZ
MSG056: ASCIZ
MSG058: ASCIZ
MSG061: ASCIZ
MSG062: ASCIZ
MSG063: ASCIZ
MSG064: ASCIZ
MSG065: ASCIZ
MSG066: ASCIZ
MSG066: ASCIZ
MSG070: ASCIZ
MSG070: ASCIZ
MSG070: ASCIZ
MSG077: ASCIZ
MSG077: ASCIZ
MSG079: ASCIZ
MSG079: ASCIZ
MSG079: ASCIZ
MSG079: ASCIZ
MSG088: ASCIZ
MSG090: ASCIZ
MSG090: ASCIZ
MSG090: ASCIZ
MSG091: ASCIZ
MSG091: ASCIZ
MSG091: ASCIZ
MSG092: ASCIZ
MSG093: ASCIZ
MSG093: ASCIZ
MSG095: ASCIZ
MSG101: ASCIZ
MSG102: ASCIZ
MSG103: ASCIZ
MSG104: BYTE
MSG105: ASCIZ
                                                                                                                                                                                                                                                      101
             16571
16572 076474
16573 076513
16574 076534
16575 076567
16576 076611
16577 076620
16578 076640
16579 076651
16580 076661
16581 076673
16582 076742
16583 076751
16584 077002
16585 077020
16586 077052
16587 077073
16589 077133
16589 077133
16590 077160
16591 077176
16592 077220
16593 077234
16594 077246
16595 077262
16596 077270
16597 077300
16598 077330
16599 077357
16600 077401
16601 077403
16602 077456
16603 077474
16604 077551
16605 077615
16606 077647
16607 077664
16608 0777730
16611 077737
16612 077730
16613 100000
16614 100014
16615 100015
                                                                                                                                                                                                                                                    127
117
105
040
077
120
117
115
                                                                                                                                                                                                                                                                                                                                                                                                           120
200
200
200
077
111
                                                                                                                                                                      /N OCTAL /
/NUMBER/<CRLF>
                                                                                                                       / DECIMAL /

<CRLF>/ERRORS > 20 - ABORTING FOR XXDP CHAIN/
                                                                                                                                                                                                                                                      CRLF>/ERRORS > 20 - ABORTING FOR XXDP CHAIN/
/FATAL /
/K WORDS OF MEMORY TOTAL/<CRLF>
<CRLF>/REFRESH TEST/
<CRLF>/RELOCATION NOT POSSIBLE/<32>
<CRLF>/REDOCATION NOT POSSIBLE/<32>
<CRLF>/ BANK ERRORS/<CRLF>
<CRLF>/END PASS #/
/ ERROR(S) DETECTED/<CRLF>
<CRLF>/FILL COUNT (OCTAL)? /
<CRLF>/KERNEL STACK/
<CRLF>/USER STACK/
/ IS EMPTY/
/RELOCATED /
/BANK=/
/ TEST=/
<CRLF>/LEAVING KAMIKAZE MODE/
<CRLF>/LEAVING FS MODE/<CRLF>
32.0

<CRLF>/ENTER BANKS - USE NUMBER 200 TO TERMINATE/
<CRLF>/CACHE IS ON (EXCEPT DURING ACTUAL PATTERNS)/
<CRLF>/CACHE IS ON (EXCEPT DURING ACTUAL PATTERNS)/
<CRLF>/CACHE IS ON (EXCEPT DURING ACTUAL PATTERNS)/
<CRLF>/ONLY SELECTED BANKS WILL BE TESTED/
/K OF MS11-M/<CRLF>
/K OF MS11-M/<CRLF>
/K OF MS11-M/<CRLF>
/K OF MS11-P/<CRLF>
/CCRLF>/ NO/
/ CACHE AVAILABLE/
/ CACHE BYPASSED/

CONTROLS TOO MANY BANKS/
                                                                                                                                                                                                                                                                                                                                                                                                               /FATAL /
                                                                                                                                                                         105
101
040
105
114
                                                                                                                                                                         114
000
105
103
103
                                                                                                                                                                                                                                                                                   MSG103: ASCIZ
MSG104: BYTE
MSG105: ASCIZ
MSG106: ASCIZ
MSG107: ASCIZ
MSG110: ASCIZ
MSG111: ASCIZ
MSG112: ASCIZ
MSG113: ASCIZ
MSG114: ASCIZ
MSG114: ASCIZ
MSG117: ASCIZ
MSG119: ASCIZ
MSG120: ASCIZ
MSG120: ASCIZ
MSG121: ASCIZ
MSG122: ASCIZ
MSG122: ASCIZ
MSG122: ASCIZ
                                                                                                                                                                                                                                                        116
101
101
                                                                                                                                                                                                                                                                                                                                                                                                                / CONTROLS TOO MANY BANKS/
```

								(15		
CZMSPAO MESSAGES	MS11-L/M/	P MEMORY	DIAG.	MACRO M	11113 2	6-APR-82	09:41	PAGE S	517-2		
16616 16617 16618 16619 16620 16621 16622 16623 16630 16631 16632 16636	100046 100070 100135 100154 100174 100226 100275 100300 100354 100354	040 200 200 200 200 200 200 120	120 120 124 124 200 040 200 122	101 122 122 122 040 103 000 117	MSG125 MSG126 MSG127 MSG128 MSG008 MSG000 MSG129 EM62:	: .ASCIZ : .ASCIZ : .ASCIZ : .ASCIZ	CRLF CRLF CRLF CRLF CRLF CRLF	>/TRACI >/TRACI >>CRLF: >'' CZM: >>CRLF:	RAM CSF E ENABL E DISAE >/ SPA MS NOT SUF	R COULD NO LED/ BLED/ S11-L/M/P PPORTED BY	CSR MAP/ <crlf> MEMORY DIAGNOSTIC' THIS DIAGNOSTIC/</crlf>

CZMSPAO MS11-L/M/P	MEMORY DIAG.	MACRO M1113	26-APR-82 09:41	PAGE 517-3
--------------------	--------------	-------------	-----------------	------------

ABORTF	002142	в0	004614	B63	044014	CPUERR=	177766	DH14	073202
ACFLAG	002114 002344	B1 B10	005550 013462	B64 B65	044014 045234 045242	CR =	000015 000200	DH15 DH16	073202 073261 073400
ADDRES ANA2	002032 010002	B100 B101	055070 055076	B66 B67 B7	045242 045372 045406	CSR CSRADD=	002146 172100	DH19	073415 072451
APTDOW	047746	B102 B103	062530 062534	B7 B70	013420 052536	CSRCAS CSRFBA	020174 002230	DH20 DH23 DH24 DH25 DH26 DH27	073422 073501
APTECC APTFLA	002420 002346	B104	062632	B71	052760 053234	CSRF IR CSRHOL	002224 002522	DH24	073560 073627
APTHAN APTHLT	015316 050014	B105 B106	062724 063010	B72 B73	053530	CSRINC	002324 002456	DH26	073665 073703
APTPAR APTSIZ	002416 002440	B11 B12	014214 014302	B74 B75	053530 053626	CSRINF	002234 002226	DH3 DH30	072474
BACKGN	060536 041224	B13 B14	014302 014344 014570	B76 B77	054566 054572	CSRLAS	002232	DH5	072530
BADPC	002050 002020	B15 B16	015362 017000	CACHKR	002544 002540	CSRL00 CSRMAP	002326 005774	DH6 DH7	072607
BADPSW	002030 002024	B17 B2	017250 006010	CACHOF= CACHON=	104423	CSRNO CSROUT	002150 044002	DIAGFL	002002
BADSTA	042614 002056	B20 B21	017256 017274 017336	CACHVE= CBCSR =	104474	CSRSTU= CSR1S	000021 002316	DISPRE= DISPTB	000174
BAD2 BAD3	002052 002054	B22 B23	017410	CBITS CBREG =	002312 1045 <u>13</u>	CTEST	006656 002144	DOBACK DONE DSWR =	015352 006642
BAFPAF	015452 015560	B24 B25	024170 024174	CB1CSR= CHECK	002310	DATARG= DATBUF	177754 002240	DT1	177570 067132
BAKPAT BANK	002616 002100	B26 B27	024374 024402	CHKDIS= CHKGEN	044366	DBEMSK DDISP =	002240 002254 177570	DT10 DT11	067252 067274
BANKIN BANKMO	002102 046410	B21 B22 B23 B24 B25 B26 B27 B30 B31 B32 B33 B33 B35 B37	006060 024700	CHKTAB	044474 040614	DEENER= DETAIL	104421 062274	DT12 DT13	067302 067306
BANKOK BAWPAF	047446 015666	B31 B32	032410 034126	CHK1DI=	104505 064050	DETFLA	002216 002214	DT14 DT16	067306 067330 067346 067376
BAWPAR BGTEST	016016 036244	B33 B34	034164 034606	CKSWR = CLRCSR=		DETRO DETRI	002176 002200	DT17	06/144
BITNO =	002320	B35 B36	034644 034660	CLREX	007752 007642	DETR2 DETR3	002202 002204	DT2 DT20 DT22 DT23 DT24 DT25	067404 067414
BIT1 = BIT10 =	000002	B37 B4	035000 006366	CLR1CS= CMD16A		DETR4 DETR5	002206 002210	DT23 DT24	067422
BIT11 =	221222	B40	035160	CMD 141 -	000073	DETSP	002212 062764	DT25	067464
BIT13 =	020000	B42	036376	CMD5C CMD7B	052274 052532	DET1 DF1 DF11	067544 067667	DT27	067504 067150
BIT14 = BIT15 = BIT2 =	100000	B44 B45	036574	CMD7C	052606 053230	DF13 DF14	067700 067710	DT30	067524 067160
BIT3 =	000010	846 847	037254	CMD TOL = CMD 5B CMD 7B CMD 7C CMD 9B CMD 9C CONF GE CONF I E CONF I E CONT F L CONT R L =	053304	DF11 DF13 DF14 DF15 DF16	067717 067726	DT5	067476 067504 067504 067524 067160 067170 067206 067222 002174 000102 104470
BIT5 =	000040	B5	006422	CONFIE	003654	DF2 DF3	067545 067571	DT7 DUMMY	067222
BIT7 =	000200	B51	037656	CONTEL	002220	DF4 DF5	067604 067617	DUMPCS= FCCDIS=	000102
BIT9 =	001000	B53	040336	CONTS	064220	DF6 DF7	067632	ECCINI=	104472 005714 104471
BIT12 = BIT13 = BIT14 = BIT15 = BIT2 = BIT3 = BIT4 = BIT5 = BIT6 = BIT6 = BIT7 = BIT8 = BIT9 = BLOCK1 BLOCK2 BLOCK3 BMFLAG	= 020000 = 040000 = 100000 = 000004 = 000020 = 000020 = 000400 = 001000 = 001000 050016 050036 050052 047524 047570 036246 002370	B41 B42 B43 B445 B467 B50 B512 B518 B518 B518 B518 B518 B61 B61 B62	035202 036376 036536 036574 037066 037254 037452 006422 037646 037656 040166 040336 040352 040416 043440 012574 043600 043604 044010	CONTS CONTS1 CONTS2 CONTS3 CONTT COUNT CPERRF CPSAVE CPUBIT	052000 052274 052532 052606 053230 053304 002444 002650 002220 177746 064220 063576 064222 056370 064144 002364 061322 061320 002104	DF8 DF9	062764 067544 067667 067700 067710 067717 067726 067545 067571 067604 067617 067632 067645 067660 072764 073100 073100	DT26 DT27 DT3 DT30 DT4 DT5 DT6 DT7 DUMPCS= ECCINI= ECCIVP ECCIDI= ECCIIN=	104471
BMFLAG	002126	B57	043440	CONTT	064144	DH1 DH10	072414	FMTVFC=	000030
BOOT BOOT1 BRGOBB BSIZE	047570	B60	043600	CPERRE	061322	DH11	073062 073100	EM11 EM12 EM13 EM14	104473 000030 070235 070257 070303 070335
BRITE	002370	B62	044010	CPUBIT	002104	DH12 DH13	073105	EM14	070335

CZMSPAO MS11-L/M/P MEMORY	DIAG.	MACRO M1113	26-APR-82 09:41	PAGE 517-4
---------------------------	-------	-------------	-----------------	------------

10000									
EM15 EM17 EM19 EM20 EM20 EM21 EM22 EM23 EM24 EM25 EM26 EM27 EM26 EM27 EM30 EM32 EM35 EM36 EM36 EM40 EM51 EM52 EM53 EM56 EM57 EM60 EM51 EM60 EM61 EM62 EM7 EM60 EM61 EM62 EM7 EM60 EM61 EM62 EM7 EM60 EM61 EM62 EM7 EM60 EM61 EM60 EM61 EM61 EM60 EM61 EM61 EM60 EM61 EM61 EM60 EM61 EM60 EM61 EM61 EM60 EM60 EM60 EM61 EM60 EM60 EM60 EM60 EM60 EM60 EM60 EM60	= 104507 100354 = 104420 047436 002454 = 104512 002550 = 104000 002016 002026	EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE	055132 062576 062576 062662 062756 063042 014274 014372 014372 014710 015450 017604 017564 017514 017514 017514 017514 017514 017514 017516 024756 024756 034370 035152 035152 035152 035176 035176 035176 035176 035724 035722 036724 037622 036724 037622 036724 037622 036724 037622 036724 037622 040116 040322 040366 040376	E667 E77 E77 E77 E77 E77 E77 E77 E77 E77	053470 053416 053440 053502 053502 053604 053662 050102 000066 007174 060000 002602 041074 015062 050402 050402 050402 050512 050402 051134 051454 052472 05	IMPTES INCBNK INCPAT IN	002004 026774 000010 172360 172374 172376 104417 002000 002524 172350 172352 172352 172354 172350 172352 002112 002372 002560 157776 002552 002554 002014 000607 000106 000601 000450 0000623 000062 0000100 000010	LOOP LOWMAP LSIZE LWSBE L100 L100 L100 L100 L100 L100 L100 L110 L1	22222

CZMSPAO MS11-L/M/P	MEMORY DIAG. MACRO M1113	26-APR-82 09:41 PAGE	J 15 517-5	
154 016740 155 017020 156 017056 157 017066 16 005502 160 017514 161 017476 162 017546 163 020016 164 020126 165 020264 166 020312 167 020316 17 005614 170 020320 171 020364 172 020420 173 020424 174 020426 175 020574 176 021250 177 022116 120 005642 120 005642 120 002126 1201 022126 1202 022136 1203 022174 1204 022204 1205 022204 1206 022214 1207 022244 1208 022204 1210 022250 1211 022300 1211 022300 1212 022310 1212 022310 1213 022310 1214 022320 1215 022364 1216 022374 121 022464 1227 022552 123 022464 1217 022374 1218 022374 1219 022464 1219 022454 1210 022502 1211 022300 1212 022310 1212 022310 1213 022310 1214 022320 1215 022364 1216 022374 1217 022374 1220 022464 1221 022454 1222 022464	L237 023266 L241 005762 L240 023536 L241 023546 L242 023546 L243 023556 L244 024322 L245 024270 L246 024322 L247 024366 L250 024530 L251 024530 L251 024530 L252 024530 L253 024742 L253 024742 L254 024742 L255 025050 L266 025022 L257 025100 L261 025622 L263 026112 L263 026150 L264 026250 L265 026274 L266 026346 L270 026460 L271 026524 L272 026564 L273 026624 L274 026664 L275 026764 L275 026764 L276 027016 L277 027024 L276 027016 L277 027024 L276 027016 L277 027024 L276 027016 L271 036460 L271 026524 L272 026564 L273 026624 L274 026664 L275 026724 L275 026724 L276 027016 L300 027030 L301 030646 L300 034660 L311 034160 L312 034630 L321 034630 L321 034650 L322 034702	L324 L325 C34732 L326 C34740 L327 C34764 L333 C06142 L330 C35122 L331 C35034 L333 C350366 L334 C35052 L335 C35066 L336 C35110 L337 C35076 L340 C35110 L341 C351 C36570 L341 C342 C36570 L342 C36570 L343 C36660 L344 C36660 L345 C36676 L346 C36702 L351 C352 C37156 L344 C366702 L351 C352 C37156 L354 C36702 L355 C37400 L355 C37760 L355 C3760 L360 C36702 C357 C370 C4360 C36702 C37760 C4365 C36702 C4377 C406250 C400 C40054 C4365 C40054 C4365 C40054 C4377 C4006250 C40102 C400 C40054 C4377 C4006362 C400 C4006666 C411 C410 C406666 C411 C410 C41052 C417 C417 C42150	L420 042160 L421 042204 L422 042300 L423 043220 L424 043220 L425 043466 L427 043520 L430 043536 L431 043630 L432 043630 L433 043630 L434 04570 L435 044020 L436 045230 L447 045231 L440 045231 L440 045231 L440 045531 L441 045531 L445 045531 L445 045531 L445 045531 L445 045531 L446 045531 L451 045553 L452 045553 L453 04560 L454 04572 L456 04736 L457 04754 L460 04765 L461 04765 L462 04766 L463 04770 L460 04765 L464 04771 L460 04765 L464 04771 L460 04765 L464 04771 L460 04765 L464 04771 L460 04765 L461 04765 L462 04766 L463 04770 L464 04771 L465 05013 L477 05033 L471 05033	L502

CZMSPAO MS11-L/M/F	MEMORY DIAG.	MACRO M1113	26-APR-82 09:41	PAGE 517-6
--------------------	--------------	-------------	-----------------	------------

L\$70 061264 MITORAP 042570 MISGOS6 076367 MIPBOS 027550 MITORO1 02201 L\$71 061134 MITORAP 042570 MISGOS8 076367 MIPBOS 027550 MITORO1 02201 L\$72 061264 MITORAP 042570 MISGOS8 076367 MIPBOS 027550 MITORO1 02201 L\$72 061262 MISGOS8 076361 076611 MIPBOS 02772 MITORO1 02201 L\$73 061262 MISEDH 002572 MISGOS6 076620 MIPBOS 035314 MITORO1 02211 L\$73 061262 MISEDH 002572 MISGOS6 076650 MIPBOS 035314 MITORO1 02211 L\$73 061264 MISEDH 002574 MISGOS6 076660 MIPBOS 0353706 MITORO1 02251 L\$74 061224 MISEDH 002574 MISGOS6 076661 MIPBOS 0353706 MITORO1 02251 L\$75 061265 MISGOS 076661 MITOROS 0353706 MITORO1 02251 L\$75 061265 MISGOS 076661 MIPBOS 0353706 MITORO1 02251 L\$75 061265 MISGOS 076661 MIPBOS 035706 MITORO1 02251 L\$76 061272 MISGOS 076067 MISGOS 076673 MIPCOS 036030 MITORO1 02251 L\$76 061272 MISGOS 076067 MISGOS 076673 MIPCOS 035514 MITORO1 02251 L\$00 0011570 MISGOS 076067 MISGOS 076751 MIPCOS 035514 MITORO1 02251 L\$00 0011570 MISGOS 074020 MISGOS 076751 MIPCOS 0353746 MITORO1 02251 L\$00 0052764 MISGOS 074120 MISGOS 077020 MIPC25 035746 MITORO2 0227 L\$00 062764 MISGOS 074125 MISGOS 077020 MIPC25 035746 MITORO2 0227 L\$00 062764 MISGOS 074175 MISGOS 077020 MIPC25 035064 MITORO2 02351 L\$00 062764 MISGOS 074175 MISGOS 077020 MIPC25 035064 MITORO2 02351 L\$00 063044 MISGOS 074410 MISGOS 077103 MIPDOS 036561 MITORO2 02351 L\$00 063044 MISGOS 074410 MISGOS 077103 MIPDOS 036561 MITORO2 02351 L\$00 063044 MISGOS 074410 MISGOS 077103 MIPDOS 036561 MITORO2 02351 L\$00 064164 MISGOS 074410 MISGOS 077103 MIPDOS 036561 MITORO2 02351 L\$00 064164 MISGOS 074410 MISGOS 077103 MIPDOS 037726 MIPDOS 037726 MITOROS 02351 L\$00 064256 MISGOS 074410 MISGOS 077103 MIPDOS 037726 MITOROS 02740 MITOROS 02756	MTPA26 036014 MT0007 022004 MTPB03 027550 MT0010 022046 MTPB04 027702 MT0011 022102 MTPB21 034460 MT0012 022160 MTPB24 035314 MT0013 022264 MTPB25 035706 MT0014 022350 MTPB26 036030 MT0015 022440 MTPC03 027610 MT0016 022516 MTPC21 034514 MT0017 022574 MTPC24 035330 MT0020 022616 MTPC25 035746 MT0021 022706 MTPC25 035746 MT0021 022706 MTPD03 027626 MT0023 023212 MTPD26 036064 MT0025 023522 MTPD26 036104 MT0026 023600 MTPD25 035612 MT0027 024102 MTPD00 027400 MT0030 024566 MTPD00 027424 MT0031 025070 MTPD00 027426 MT0032 025260 MTPD00 027722 MT0033 025612 MTPO06 027756 MT0034 026000 MTPD007 030156 MT0034 026000 MTPD007 030156 MT0034 026000 MTD007 026152
MAPHO	MTP020 034114 MT0046 026654 MTP022 034600 MT0047 026714 MTP025 035346 MT0999 026760 MTP030 036122 MT1 017026 MTP031 036132 MT2 017032 MTP032 036210 MUT 002106 MTP033 036242 NC 056430 MTP034 036340 NEMCNT 002066

CZMSPAO MS11-L/M/P	MEMORY DIAG.	MACRO M1113	26-APR-82 09:41	PAGE 517-7
--------------------	--------------	-------------	-----------------	------------

PERR31= 104455 SBEMSK 002250 SUPDR3 002164 TSTDAT 002244 WURST 002564	NXTCSR 005642 OLDCAC 002276 OLDCSR 002154 ONES 002600 PADDRE 002034 PAFBAF 016146 PAFBAW 016276 PARBAW 016600 PARCNT 002070 PARITY 042406 PARTHE 002300 PARVEC= 000114 PASFLG 002262 PASSNO 002264 PATERR 002072 PATPLU 004606 PATTER 002110 PCBUMP 002322 PCONFI 041352 PCONFS 041652 PCONFS 041652 PCONFS 041652 PCONF1 041562 PCONF2 041620 PDP11 042602 PD1 054476 PERRAW 057476 PERRAW 057476 PERRAW 057242 PERRAS 104431 PERRAY 057366 PERROS 104431 PERROS 104431 PERROS 104431 PERROS 104437 PERROS 1044437 PERROS 1044440 PERROS 10444440	PERR32= 104456 PERR33= 104467 PERR35= 104461 PERR36= 104463 PERR37= 104463 PERR40= 104464 PERR41= 104465 PERR42= 104466 PERR43= 104467 PERR43= 104467 PERCDF 061740 PFECDF 061740 PFECDH 061700 PFECDT 061730 PFECDM 061644 PFECWS 061634 PFLAG 002120 PGMCSR 002526 PHEBE 014004 PHYADD 002036 PMEMFL 0021400 PROTYP 003752 PSIZE 002400 PSW = 177776 PTABLE 036732 PWRVEC= 000024 QUICK 002432 QVFLAG 002342 RANODD 036044 RDCHR = 104411 RDDEC = 104414 RDLIN = 104412 READON 002404 REALPA 002274 REFRES 035154 READON 002404 REALPA 002274 REFRES 035154 REFSUB 035224 REGCOP 041064 RELENT 045612 RELOCA 045172 RESO 050376	SBENT 020146 SBESYN 034410 SBETES 017670 SCOPE = 000004 SDPAR0 = 172260 SDPAR5 = 172272 SDPAR6 = 172274 SDPAR7 = 172276 SEEDHI 002566 SEEDHI 002570 SELONL 002000 SETPAT 047500 SHADL1 012452 SHADL1 012452 SHARD = 172240 SIPAR3 = 172246 SIPAR6 = 172254 SIPAR6 = 002006 SKIPKA 002006 SKIPKA 002006 SKIPKA 002064 SKIPKA 002064 SKIPKA 002064 SKIPKA 002064 SKIPKA 002556 SOBLEN = 000556 SOBLEN = 000556 SOBLEN = 000306 SIPART 000300 START 000300 START 000300 START 000300 START 000310 START 000310 START 000300 START 000310 START 000310 START 000310 START 000310 START 000300 START 000310 START 000310 START 000310 START 000310 START 000310 START 000300 START 000300 START 000310 START 000300	SUPDR4 SUPDR5 SUPDR6 SUPDR6 SUPDR6 SUPDR6 SUPSTK= 000740 SWAPAT O02620 SWAPAT SWAPAT O02622 SWREG = 000176 SW0 = 0000001 SW1 = 004000 SW11 = 004000 SW12 = 010000 SW13 = 020000 SW15 = 100000 SW15 = 100000 SW2 = 000040 SW3 = 000010 SW4 = 000020 SW4 = 000020 SW5 = 000040 SW5 = 000040 SW6 = 000100 SW7 = 000200 SW7 = 000200 SW8 = 000100 SW7 = 000200 SW6 = 000040 SW7 = 000200 SW6 = 000040 SW7 = 000200 SW7 = 000200 SW6 = 000040 SW7 = 000200 SW7 = 000200 SW7 = 000040 SW7	TSTRD1 TSTREA= 104510 TST1 005522 TST2 0011404 TST3 011570 TST4 012556 TST5 014742 TST6 TST5 014742 TYPDS = 104405 TYPEIT= 104401 TYPOC = 104402 TYPS0 = 000000 TYPS1 = 000000 TYPS1 = 000000 TYPS3 = 000000 TYPS5 = 000000 TYPS5 = 000000 TYPS6 = 177660 UDPAR0= 177660 UDPAR0= 177640 UIPAR1= 177644 UIPAR2= 177644 UIPAR3= 177646 UIPAR4= 177650 UIPAR6= 176
	PERR30= 104454 PERR31= 104455	SAVREG= 104415 SBEMSK 002250	SUPDR3 002164	TSTBAN 012310	WORST 002564

CZMSPAO MS11-L/M/P SYMBOL TABLE	MEMORY DIAG. MACRO M11	13 26-APR-82 09:41 PAGE 5	517-8	
XOCHAR	\$DOAGN 015242 \$DOWN 054726 \$DTBL 063516 \$ECCDI 043256 \$ECCIN 043304 \$ECCID 043272 \$ECCII 043320 \$ENASB 043332 \$ENASB 043332 \$ENASB 043332 \$ENASB 043332 \$ENERG 042654 \$ENV 065660 \$ENVM 065661 \$EOP 015066 \$ERFLG 002012 \$ERRGE 044100 \$ERROR 060522 \$ERRTB 066262 \$ERRTU 061324 \$ERTTL 002614 \$ESCAP 002356 \$ETABL 065660 \$ETABL 065642 \$FILLS 002353 \$FS = 000001 \$FATAL 065642 \$FILLS 002353 \$FS = 000000 \$GTSWR 063712 \$HALT 061106 \$HALT2 066014 \$HIBTS 065740 \$HIDCT 065140 \$ILUP 055330 \$INVAL 044050 \$ITEMB 002013 \$IS = 000001 \$KERNE 042644 \$KMAP 045100 \$KS = 000102 \$L = 000107	\$LF \$LDADC \$LPADR \$LPERR \$LPERR \$02610 \$LPERR \$000000 \$MADR1 \$MADR2 \$MADR3 \$MADR3 \$MADR3 \$MAMS1 \$MAMS1 \$MAMS2 \$MAMS2 \$MAMS3 \$05700 \$MAMS4 \$MBADR \$05704 \$MBADR \$05704 \$MBADR \$05704 \$MSGLG \$05654 \$MSGLG \$MSGLG \$MSGTY \$MSGLG \$MSGTY \$MSGLG \$056675 \$MSTYP1 \$MTYP1 \$MTYP2 \$MSTYP1 \$MTYP2 \$SMTYP3 \$065701 \$MTYP4 \$065705 \$MTYP4 \$065705 \$MTYP4 \$065705 \$NOTRA \$06010 \$NULL \$00001 \$OCTVL \$065622 \$OCTVL \$065626 \$OMODE \$063310 \$OVER \$065646 \$OS = 000000 \$PASS \$PASTM \$065746 \$PATMA \$00001 \$PER01 \$0566612 \$PER03 \$0566612 \$PER07 \$056674	\$PER10 056716 \$PER11 056746 \$PER12 056766 \$PER13 057010 \$PER14 057030 \$PER15 057052 \$PER16 057074 \$PER17 057114 \$PER20 057132 \$PER21 057150 \$PER22 057170 \$PER22 057170 \$PER22 057170 \$PER23 057206 \$PER24 057224 \$PER25 053762 \$PER26 057414 \$PER20 054210 \$PER30 054210 \$PER30 054210 \$PER31 057624 \$PER30 054210 \$PER31 057624 \$PER32 057722 \$PER33 057770 \$PER34 060050 \$PER35 060102 \$PER35 060102 \$PER35 060102 \$PER36 060136 \$PWRUP 054732 \$PER37 060422 \$PER38 065330 \$PER38 065330 \$PER38 065330 \$PER38 060102 \$PER39 060422	\$SWREG \$T = 000610 \$TESTN 065644 \$TKB 002630 \$TKS 002626 \$TN = 000007 \$TPB 002634 \$TPFLG 002354 \$TPS 002632 \$TRAP 065754 \$TRAP2 065776 \$TRAPD 066016 \$TSTM 065744 \$TSTRD 043052 \$TTYIN 064704 \$TYPDS 063312 \$TYPE 056152 \$TYPEC 056300 \$TYPEX 056466 \$TYPOC 063110 \$TYPON 063124 \$TYPON 063124 \$TYPON 063124 \$TYPON 063124 \$TYPON 063124 \$TYPON 065652 \$UNIT 065652 \$UNIT 065750 \$USWR 065664 \$VECT1 065710 \$VECT2 065712 \$WASDB 043566 \$WASSB 043422 \$WASDB 043702 \$WASDB 043566 \$WASSB 043422 \$WASDB 043566 \$WASSB 043536 \$XTSTR 060314 \$Y\$ = 000000 \$ZAP42 015212 \$Z\$ = 000000
FLAPSED TIME: 00:	ED: 26094 WORDS (102 21558 WORDS (82 PAGES) :29:08 /-SP/CR=CZMSPA/ML,CZMSPA			

										SEQ (140
CZMSPA SYMBOL	CREATED BY CROSS REFERENCE	MACRO ON 2	6-APR-82 AT	09:54 F	AGE 1						
SYMBOL	VALUE	REFERENCES	*337-11069		423-13260	*423-13276	454-14585	454-14588	454-14597	454-14600	
ABORTF	002142 002114	#139-5600 #139-5589 201-7497 *371-11906	170-6863 203-7535	421-13214 172-6903 205-7580 371-11920	182-7045 207-7618 393-12463	191-7343 211-7712 397-12533 217-7908	193-7359 238-8445	195-7389 238-8474	197-7419 240-8517	199-7456 363-11619	
ACTFLA	002344	#139-5665	*153-5981 229-8227	167-6724 229-8237	173-6939 231-8252 *273-9213	234-8372	217-7923 253-8814 *273-9240	226-8178 363-11613	226-8187 377-12024	226-8202 466-14898	
ADDRES	002032	227-8213 #139-5565 *277-9417 *315-10343 *423-13259	*160-6196 *282-9476 *316-10380	*167-6718 *283-9560 *317-10418 423-13262	*283-9566 *323-10594 423-13272	*273-9219 *285-9658 *323-10610 *451-14478	*285-9665 *339-11095 *451-14479	*273-9253 *305-10119 *421-13213 *451-14484	*275-9329 *305-10131 *421-13213 *451-14485	*275-9343 *315-10328 *421-13239 *451-14490	
		*451-14491 *453-14533 *453-14578 509-16408	*451-14496 *453-14538 458-14649	*451-14497 *453-14543 *460-14677	*451-14504 *453-14548 *460-14686	*451-14512 *453-14553 473-15088	*451-14517 *453-14558 509-16386	*451-14517 *453-14563 509-16389	*451-14522 *453-14568 509-16392	*451-14527 *453-14573 509-16397	
ANA2 APTDOW	010002 047746	#166-6449 153-5975	#377-12058	377-12061	*377-12063 186-7237	509-16391					
APTECC	002420 002346	#139-5689 #139-5666 229-8237	*186-7232 *153-5974 231-8252	186-7237 186-7208 234-8372	189-7325 253-8814	226-8178 363-11613	226-8187 377-12024	226-8202 466-14898	227-8213	229-8227	
APTHAN	015316	#189-7327	231-8252 189-7333								
APTHLT APTPAR	050014 002416	229-8237 #189-7327 #377-12065 #139-5688	*186-7229 *153-5971	186-7237	509-16391						
APTSIZ BACK	002440 060536	#465-14825	466-14915	186-7208			0// 07/0	0// 07/0	#700 40075		
BACKGN BAD	060536 041224 002050	222-8115	222-8133	224-8161 *160-6244	232-8268 *282-9490	234-8335 *282-9504 *323-10596	246-8710 *291-9784	246-8715 *291-9796	#328-10835 *317-10417	*319-10454	
	***************************************	#139-5571 +321-10494 +326-10779	*321-10503	*322-10541	*322-10551	*323-10596 *423-13262	*323-10609 *423-13274	*324-10660 *451-14480	*324-10668 *451-14486	*325-10722 *451-14493	
		*451-14498	*451-14503	*451-14513	*421-13242 *451-14518 *453-14564	*423-13262 *451-14523 *453-14569	*423-13274 *451-14528 *453-14574	*453-14534 *453-14579	*453-14539 456-14626	*453-14544	
		*453-14549 *460-14678	*460-14687	*460-14688	*460-14707	*460-14714	509-16386	509-16392	509-16400	509-16402	
BADPC	002020	509-16404 #139-5560	509-16407 *339-11122	509-16409 421-13227 465-14862	454-14586	454-14598	456-14609	458-14661	460-14676	460-14685	
BADPSW	002030	#139-5564	*339-11123	460-14707	465-14865						
BADSP BADSTA BADXOR		#139-5560 460-14696 #139-5562 337-11070 454-14598 #139-5572 #139-5573 187-7249 #141-5742 #139-5583 *170-6869 *182-7071 *201-7492 213-7842 238-8465 242-8551 354-11366	\$509-16407 *339-11122 465-14861 *339-11123 *339-11120 337-11078 456-14609 *456-14626 *421-13243 *421-13243 *421-13243 *193-7355 #195-7385 *147-5871 *169-6739 *170-6873 182-7071 *203-7530 232-8278 *238-8472 244-8580 361-11576	421-13227 465-14862 460-14707 *339-11121 339-11103 458-14661 *456-14627 509-16400 *421-13246 193-7380 195-7410 232-8267 *169-6751 170-6875 *191-7341 *205-7575 232-8292 238-8475 246-8644 *363-11617	*465-14866 465-14865 465-14864 339-11107 460-14676 509-16392	339-11110 460-14685	#339-11119 460-14696	421-13227 460-14706	451-14506	454-14586	
BAD2 BAD3	002056 002052 002054 015452 015560 002616 002100	#139-5572 #139-5573	*421-13243 *421-13245	509-16400 *421-13246	509-16400						
BAFPAF BAFPAR	015452 015560	187-7248	#195-7355 #195-7385	195-7410							
BAKPAT BANK	002616 002100	#141-5742 #139-5583	*147-5871 *169-6739	232-8267 *169-6751	234-8334 169-6752 *172-6899 *191-7349 *207-7613 234-8345 *238-8494 255-8822 363-11620	169-6754	169-6761	*169-6780	*170-6860	*170-6866	
		*170-6869 *182-7071	*170-6873 182-7071	170-6875 *191-7341	*172-6899 *191-7349	*173-6936 191-7349	173-6936 *193-7356	*173-6938 *195-7386	174-6948 *197-7416	*170-6866 *182-7041 *199-7453 213-7817 *238-8465 *240-8533 326-10763 *365-11728	
		*201-7492	*203-7530	*205-7575	*207-7613	211-7681	211-7708	211-7745	*211-7761	213-7817	
		238-8465	*238-8472	238-8475	*238-8494	238-8494	*240-8514	*240-8523 331-10473	240-8523	*240-8533	
		354-11366	5 361-11576	*363-11617	363-11620	169-6754 *173-6936 191-7349 211-7681 236-8395 238-8494 257-8855 *363-11630	169-6761 173-6936 *193-7356 211-7708 236-8408 *240-8514 319-10430 363-11630	*169-6780 *173-6938 *195-7386 211-7745 *238-8443 *240-8523 321-10473 365-11723	*170-6860 174-6948 *197-7416 *211-7761 238-8447 240-8523 321-10479 *365-11724	*365-11728	

CZMSPA SYMBOL	CREATED BY	MACRO ON 26-APR-82 AT	09:54 PAGE 2 CREF	324 040
SYMBOL	VALUE	REFERENCES *365-11732 369-11857 386-12230 *386-12245 *390-12308 390-12309 393-12440 *393-12461 395-12498 *395-12504 397-12545 430-13473 458-14635 463-14792	*430-13474 430-13475 *430-13495 432-13514 *432-13515 432-13510 *	386-12221 390-12303 390-12428 395-12491 397-12544 432-13529
BANKIN BANKMO BANKOK BAWPAF BAWPAR	002102 046410 047446 015666 016016	#139-5584 *169-6757 246-8700 329-10859 189-7324 363-11623 201-7495 203-7533 187-7250 #197-7415 187-7251 #199-7452	169-6813 170-6862 172-6902 182-7043 211-7687 226-8192 355-11375 361-11554 363-11625 365-11726 365-11730 *371-11912 365-11721 #367-11774 377-12048 205-7578 207-7616 #373-11963 197-7439 197-7447 199-7476 199-7484	246-8674
BGTEST BITNO	036244 002320	#313-10264 313-10285 #139-5655 *291-9763	*291-9770 291-9806 *316-10362 *316-10370 316-10393 *325-10697 *	325-10705
BITO	= 000001	166-6521 166-6541 315-10323 315-10338 346-11231 346-11239	346-11239 346-11243 348-11278 348-11280 348-11286 348-11289 350-11325 363-11641 363-11694 363-11703 365-11742 365-11745	162-6311 169-6839 344-11210 350-11313 371-11941
BIT1	= 000002	386-12224 388-12263 #111-4738 151-5944 163-6363 163-6393 325-10709 335-10954 346-11253 352-11339	158-6133 158-6136 159-6160 159-6166 162-6298 162-6311 166-6521 166-6541 166-6548 166-6654 182-7084 182-7107	162-6319 182-7119 346-11248
BIT10 BIT11 BIT12	= 002000 = 004000 = 010000	#111-4729 315-1033 #111-4728 158-6128 #111-4727 155-6038 335-10952 363-1170	166-6581 182-7060 279-9454 283-9601 287-9703 371-11953 155-6039 155-6045 166-6577 182-7060 182-7090 182-7121 371-11950 390-12361 478-15221 478-15222 478-15237	226-8193
BIT13	= 020000	#111-4726 155-6038 246-8694 273-9282 326-10777 326-1078	155-6045 157-6095 157-6096 174-6950 174-6966 189-7286 279-9451 282-9487 283-9598 287-9700 321-10492 321-10501	246-8680 323-10592 365-11736
BIT14	= 040000	#111-4725 160-6236 355-11401 355-1140	375-12016 478-15221 478-15237 166-6629 166-6631 255-8844 297-9965 297-9966 321-10476 363-11680 365-11739 367-11798 367-11814 371-11947 400-12611 429-13429 429-13433 430-13476 432-13517	
BIT15	= 100000	#111-4724 157-6115 273-9224 324-1065 363-11680 363-1168 429-13433 460-1469 #111-4737 154-6018 291-9792 315-1032 355-11429 #111-4736 154-6018 #111-4735 137-5526 213-7836 273-9224 #111-4734 153-5967 365-11752 371-1195 #111-4733 172-6919 393-12456 393-1247 #111-4732 153-5970	163-6405 166-6594 166-6620 215-7811 215-7829 215-7832	213-7836 363-11642 426-13326
BIT2	= 000004	#111-4737 154-6018 291-9792 315-1032	316-10374 325-10709 337-11072 346-11248 346-11253 352-11339	291-9774 352-11344
BIT3 BIT4	= 000010 = 000020	#111-4736 154-6018 #111-4735 137-5526 213-7836 273-9224	154-6019 154-6022 157-6093 160-6211 160-6221 160-6251 145-5828 145-5829 157-6093 166-6594 166-6620 189-7320 321-10482 344-11207 348-11267 348-11287 460-14697 166-6514 166-6585 166-6615 189-7320 363-11685 363-11693 375-12007 377-12052 335-11037 363-11645 371-11913 390-12350 279-12527 397-12542 154-5989 154-5990 170-6871 273-9273 279-9440 283-9587	160-6269 213-7811
BIT5	= 000040	#111-4734 153-5967	321-10482 344-11207 348-11267 348-11287 460-14697 166-6514 166-6585 166-6615 189-7320 363-11685 363-11693	365-11749
BIT6	= 000100	213-7836 273-9224 #111-4734 153-5967 365-11752 371-1195 #111-4733 172-6919 393-12456 393-1247 #111-4732 153-5970	166-6514 166-6585 166-6615 189-7320 363-11685 363-11693 375-12007 377-12052 331-10875 331-10912 335-11037 363-11645 371-11913 390-12350 2 397-12527 397-12542 154-5989 154-5990 170-6871 273-9273 279-9440 283-9587	390-12423
BIT7	= 000200	#111-4732 153-5970	154-5989 154-5990 170-6871 273-9273 279-9440 283-9587	287-9689

										SEA (7403
CZMSPA	CREATED BY		-APR-82 AT	09:54 P	PAGE 3						
SYMBOL	VALUE	REFERENCES 331-10874	344-11193		363-11645 371-11929	371-11935	390-12349	393-12455	397-12526		
BIT8 BIT9	= 000400 = 001000	#111-4731 #111-4730	184-7157	174-6991	184-7158	486-15608 371-11932	****	222 0427	222 04/7	272 0204	
BLOCK1	050016	169-6748 232-8299 246-8678 #379-12080	220-8022 232-8303 328-10848	220-8044	220-8070	222-8094 236-8412	222-8097 236-8420	222 - 8123 238 - 8438	222 - 8143 240 - 8507	232-8296 244-8609	
BLOCK	050036	#379-12080			#379-12075	27/_9754	274_9757	234-8413	236-8415	242-8553	
BLOCK3		222-8098 246-8647 #139-5594	222-8099 363-11681 197-7421	222-8124 365-11740 199-7458	222-8144 367-11799 203-7537	234-8356 367-11815 207-7620	234-8357 #379-12084 363-11619	236-8413 430-13479 *371-11907	430-13482 +371-11927	*371-11943	
BOOT		*371-11946 #375-11993	483-15485		203-1731	201 1020	303 11017	-3.1.1176.			
BOOT1 BRGOBE	047524 047570 036246	375-12001 #313-10265	375-12006 313-10269	486-15612 #375-12008 313-10303 *184-7172							
BSIZE	002370	#139-5678 #141-5719	*184-/1/1 *145-5821	341-11154	*184-7173 382-12159	*184-7174	184-7175	*** *****	7/4 44464	700 12104	
CACHK	002540	#141-5718 382-12157 429-13453	*145-5823 382-12159 434-13591	154-6026 382-12162 434-13651	*154-6026 399-12575	*154-6027 *399-12575	341-11144 *399-12576	341-11146 399-12581	341-11151 *399-12581	380-12106 425-13296	
CACHOR	= 104424	#110-4615 172-6897 317-10409	154-6025	162-6279	166-6510 255-8846	167-6708 257-8883	169-6770 282-9473	169-6787 291-9756	169-6804 296-9927	169-6830 316-10356	
		317-10409 434-13595	211-7724 321-10472	322-10525	323-10567	324-10646	325-10687	326-10762	380-12110	399-12574	
CACHO	= 104423	#110-4614	147-5890 173-6944	163-6413 211-7736 317-10421	166-6686 213-7847	167-6692 213-7855	167-6711 255-8848	169-6778 257-8885	169-6791 282-9512	169-6812 291-9810	
		169-6835 296-9905 399-12582	316-10397 425-13299 111-4755	317-10421	321-10514	322-10557	323-10637	324-10677	325-10739	326-10797	
CACHVE	= 104474	#111-4/54			144-4555						
CBITS	002312 = 104513	#139-5652 #110-4673 322-10526	*166-6550 174-6977 322-10544	*166-6553 174-6992 324-10647	166-6555 282-9477 324-10663	291-9759 325-10692	291-9787	316-10358	321-10483	321-10511	
CB1CS	R = 104475	#110-4659 317-10414	166-6531 319-10437	273-9204	275-9314	277-9409	326-10767 283-9549	285-9648	305-10108	317-10412	
CHECK	002310	*283-9596	*166-6481 *287-9694	*273-9279	*273-9280 *287-9698	*279-9445 *316-10381	*279-9447 *317-10411	*279-9449 *317-10413	*283-9592 *319-10436	*283-9594 *319-10442	
CHKDI	s = 104 <u>5</u> 04	346-11247	346-11252								
CHKGE	3 044474	273-9192 357-11444 325-10719	#359-11465	277-9398	283-9536	285-9636	303-10048	#357-11434			
CHKTRI CHK1D		#110-4667	174-5955	174-6967	174-6982	174-6997	421-13230				
CKEND CKSWR CLRCS	= 104410	#110-4598	462-14735	465-14824 163-6406	174-6982 #483-15499 466-14887 #217-7911	238-8433	240-8526	390-12370	#393-12447	#397-12518	
CLREX	007752	164-6426 162-6333	164-6440 163-6414	#164-6444 #164-6426				474 4007	47/ 7004	47/ 7045	
CLR1C	s = 104503	#110-4665 213-7803	166-6530 213-7827	166-6539 213-7851	166-6676 273-9170	174-6959 273-9231	273-9244	273-9290	275-9297	174-7015 275-9317 283-9610	
		325-10/19 #110-4667 483-15466 #110-4598 #110-4664 164-6426 162-6333 #110-4665 213-7803 275-9372 285-9652 321-10489	275-9305 #359-11465 #325-10743 174-5955 483-15468 462-14735 162-6328 164-6440 163-6414 166-6530 213-7827 277-9378 287-9712 321-10513	174-6967 483-15470 465-14824 163-6406 #164-6444 #164-6426 166-6539 213-7851 277-9392 291-9778 322-10535	166-6676 273-9170 277-9411 291-9809 323-10635	174-6959 273-9231 279-9462 303-10083 324-10655	174-6972 273-9244 282-9510 315-10336 324-10676	174-6987 273-9290 283-9530 316-10396 325-10726	174-7001 275-9297 283-9551 319-10445 325-10738	319-10461 326-10774	
		321-10489	321-10313	322-10333	323-10033	324-10033	324-100/0	363-10120	363 10130	320 10174	

			. 00.51	DACE /					3EW 040
	CREATED BY	MACRO ON 26-APR-82 AT	09:54	CREF					
	VALUE	REFERENCES							
CMD5B CMD5C CMD7B CMD7C	053614 052000 052274 052532 052606 053230	/nn_124nn #/nn_12427	#390-1242	2					
CMD9C	053304	397-12523 #397-12541	*166-6646	*169-6821	*169-6840	*182-7067	*182-7146		
CONFIE	003654 002650	*166-6585 166-6586 166-6604 *166-6604 *167-6720 169-6762 173-6924 174-6976	*154-5989 *166-6586 166-6613 *169-6820 174-6991	*154-5990 166-6587 *166-6613 *169-6822 *174-7018	166-6514 *166-6587 166-6614 *169-6839 *174-7019	*166-6570 *166-6600 *166-6614 *169-6843 182-7046	*166-6574	444 4407	*166-6581 *166-6603 *166-6645 *172-6919 *182-7061 184-7178 329-10860 335-11037 363-11645 371-11913
		371-11917 371-1192	371-1192	371-11932	3/1-11933	371-11941	3/1-1174/	3/1-11730	371-11953 *458-14638
		*458-14639 *458-1464	458-1464	2					*205-7586
CONTFL	002220	*207-7626 209-7662	*211-7759	-193-1400					
CONTRL	= 177746	956 1911136 SRU91/1U	154-6005				154-6022 *434-13653		*541-11147
CONTS CONTS1 CONTS2 CONTS3 CONTT COUNT	064220 063576 064222 056370 064144 002364	483-15459 #485-1554 483-15444 #483-1544 #485-15549 485-1555 #434-13629 434-1363 462-14738 478-1518 #139-5676 *296-9906	7 9 485-1555 0 485-1555 0 434-1363 7 483-1545	4 6 6 4 #485-15522	*296-9910	296-9911	*296-9918	*296-9925	297-9930
CPSAVE	061320	*145-5857 *145-5859 *462-14742 462-1474 #139-5585 *151-5944	3 *465-1474 167-6720	0 465-14849 1 465-14852 169-6822	#466-14919 #466-14918 169-6843	470-15011 184-7156	184-7178	335-10981	
CPUERR	= 177766	#111-4766 *145-5860 *384-12204 *386-1224	*151-5956 6 *388-1229	*164-6446 8 *462-14773	*167-6721 *466-14897	*169-6781 *476-15179	*382-12176 509-16390	*382-12180 509-16398	*384-12192
CR CRLF	= 000015 = 000200	#111-4694 434-1364 #111-4695 434-1357 517-16511 517-1651 517-16528 517-1652 517-16537 517-1653 517-16547 517-1654 517-16567 517-1656 517-16583 517-1658 517-16591 517-1659 517-16604 517-1660	513-1645 517-1651 517-1653 517-1653 517-1654 517-1656 7517-1656 517-1656 517-1660 517-1660				517-16507 517-16525 517-16534 517-16543 517-16553 517-16564 517-16588 517-16601	517-16509 517-16526 517-16535 517-16545 517-16565 517-16565 517-16602 517-16613 517-16622	517-16510 517-16527 517-16536 517-16546 517-16555 517-16566 517-16581 517-16590
	CONTFL CONTSL CONTS CONTS CONTS CONTS CONTT COUNT CPERRECPSAVE CPUBIT CPUERR CR	SYMBOL CROSS REFERENCE SYMBOL VALUE CMD16A	SYMBOL CROSS REFERENCE SYMBOL VALUE CMD16A 053614 400-12600 #400-12622	SYMBOL CROSS REFERENCE SYMBOL VALUE CMD16A CMD16A CMD5B CS2000 CMD5C CMD5C CMD5C CMD5C CMD5C CMD5C CMD5C CMD7C CMD5C CMD7C CMD6C CMD7C CMC7C CM	SYMBOL CROSS REFERENCE S26-10794 421-13236 400-12600 4400-12623 421-13236 6400-12623 6400-12635	SYMBOL CROSS REFERENCE SYMBOL VALUE REFERENCES 326-10794 400-12600 #400-12623 CMD58 052000 #390-12335 390-12379 CMD58 052274 390-12335 390-12346 CMD7B 052532 #393-12461 393-12469 CMD7B 052532 #393-12461 393-12469 CMD7B 052530 #397-12531 397-12531 397-12531 CMD98 053230 #397-12523 #397-12541 CMD96 053304 397-12523 #397-12541 CONFIE 003654 #143-5782 151-5940 CONFIE 003654 #143-5782 151-5940 CONFIE 003654 #143-5782 151-5940 **166-6585 166-6585 166-6586 *166-6587 *166-6587 *166-6587 *166-6587 *166-6604 *167-6720 169-6820 *169-6822 *169-6829 *173-6924 174-6976 *174-6991 *174-7018 *174-7019 182-7121 189-7285 211-7688 211-7699 235-10954 335-10960 333-10952 335-10954 335-10960 335-10981 *354-11369 *363-11626 *363-11627 363-11641 363-11642 *363-11697 363-11707 *365-11	SYMBOL CROSS REFERENCE SYMBOL VALUE REFERENCES 326-10794 421-13236 CMD16A 053614 CMD5B 052000 CMD5C 052274 390-12353 390-12353 CMD7C 052606 393-12451 393-12469 CMD7C 052606 393-12452 M393-12539 CMD9C 053304 397-12523 M397-12539 CMD9C 053304 397-12523 M397-12539 CMD9C 053304 CONFIE 003654 CM143-5785 151-5940 166-6585 166-6586 166-6587 173-6924 174-6976 182-7107 182-7119 182-7119 182-7119 182-7119 182-7119 182-7121 184-7156 182-7107 182-7119	SYMBOL CROSS REFERENCES 326-10794 421-13236	SYMBOL CROSS REFERENCE SYMBOL VALUE REFERENCES 326-10794 400-12623 400-12600 #400-12623 100-12500 #390-12375 390-12375 390-12375 390-12375 390-12461 100-100-12523 390-12375 390-12462 100-100-12523 390-12463 100-12623 390-12464 100-12623 390-12464 100-12623 390-12464 100-12623 390-12464 100-12623 390-12464 100-12623 390-12464 100-12623 390-12464 100-12623 390-12464 100-12623 390-12464 100-12623 390-12464 100-12623 390-12464 100-12623 390-12464 100-12623 100-12634 100-12634 100-12634 100-12635 100-12636 100-12636 100-12637 100-12637 100-12637 100-12638 100-1263

CZMS		MACRO ON 26-APR-82 A	09:54	PAGE 5 CREF					SEQ 040
SYMB		REFERENCES #139-5602 *160-6209 *174-6978 *174-6980 *282-9487 282-9488 291-9796 *315-1032 *321-10498 *321-1050 322-10548 322-1055 324-10658 324-1066 *326-10768 *326-1077 343-11173 *343-1118 *346-11239 *346-1124 350-11321 *352-1133 *421-13229 421-1323 509-16396 509-1639 #113-4971 157-6085	*160-6219 *174-6993 282-9490 3 *315-10338 1 321-10502 1 *323-10580 0 *324-10666 1 *326-10777	*160-6245 *174-6995 *291-9775 *316-10375 321-10503 *323-10588	324-10668 326-10779 *344-11218 *346-11252 *352-11344	166-6536 213-7836 291-9782 *321-10486 *322-10538 *323-10604 *324-10674 *326-10787 *346-11223 *346-11253 355-11392 478-15190	*326-10788 *346-11227 348-11267	*174-6950 273-9224 *291-9789 321-10493 322-10541 323-10609 325-10721 326-10789 *346-11231 348-11287 *355-11429 *478-15205	*324-1030 325-10722 *343-11172 *346-11235 350-11302 *384-12195
CSRA	ADD = 172100	509-16396 509-1639 #113-4971 157-6085 *163-6385 163-6398 *343-11173 343-1118 211-7733 #215-7861	7 304-10403	*162-6303 *166-6526	\$09-16409 *162-6305 *166-6624 355-11402	*162-6308 166-6626	162-6310 *166-6629	*163-6353 166-6630	163-6368 *166-6631
CSRC CSRF CSRF CSRH	BA 002230 FIR 002224 HOL 002522	*163-6385 163-6398 *343-11173 343-1118 211-7733 #215-7861 #139-5630 *211-7682 #139-5628 *211-7707 #139-5708 *211-7692 #139-5657 #139-5704 *157-6092 159-6159 159-6160	211-7713	*211-7752 211-7715 211-7704	*211-7749	211-7749			
CSRI		#139-5037 #139-5704 *157-6092 159-6159 159-6160 *160-6269 162-6298 166-6571 166-6652	159-6166 162-6311 166-6654	157-6101 160-6197 162-6319 182-7038	157-6102 *160-6211 163-6363 *182-7049	*158-6126 *160-6221 163-6393 182-7077	*158-6133 160-6235 166-6521 182-7079	*158-6136 *160-6251 166-6541	159-6158 160-6261 166-6548
CSRI CSRL CSRL CSRL	AS 002226 BA 002232 00 002326	*160-6269 162-6298 166-6571 166-6652 #139-5632 *211-7684 #139-5629 *211-7713 #139-5631 *211-7683 #139-5658 *211-7686 157-6117 #159-6140	*211-7696 *211-7697	211-7751 211-7740 211-7749 211-7755					
CSRN	OUT 044002	#139-5603 *160-6195 *166-6509 166-6623 246-8706 *329-1086	*162-6286 *172-6915 4 343-11162	2 343-11168 3 *350-11298 5 390-12338 9 *426-13346 8 478-15203 9 346-11249	*350-11308 *390-12345 426-13346 *478-15205 352-11331	350-1130X	*390-12371 *428-13414	*163-6390 *211-7705 344-11194 *354-11359 *390-12430 428-13414	*166-6502 *246-8705 344-11196 354-11359 *406-12648 *429-13449
CSR1	1S 002316 ST 006656	#139-5654 *160-6202 157-6118 #162-6279 #139-5601 *211-7731	*160-6203 *211-7732	160-6213 *217-7920	160-6217 *217-7921	*219-7972	*219-7973	483-15463	
DATA	ARG = 177754 BUF 002240	*348-11263 *348-1127 355-11400 380-1210 *406-12662 425-1330 478-15190 *478-1519 346-11224 346-1123 #139-5654 *160-6202 157-6118 #162-6279 #139-5631 *211-7731 #111-4763 154-6009 #139-5634 *267-9037 *267-9053 *267-9056 *267-9072 *273-9177 *275-9299 275-9302 277-9387 *283-9518 #139-5637 *277-9384 #139-5637 *277-9384 #139-5637 *277-9384 #139-5637 *277-9384 #139-5637 *277-9384	*267-9038 *267-9057 *273-9178 275-9303 *283-9519		267-9040 267-9059 273-9184 *275-9364 283-9525	267-9042 267-9061 273-9265 *275-9364 *285-9619	267-9047 267-9066 *273-9267 *277-9380 *285-9620	267-9051 267-9070 *273-9267 *277-9381 285-9625	*267-9053 *267-9072 *275-9298 277-9386 285-9626
DBEA	MSK 002254	451-14514 451-1451 #139-5637 *277-9384 *279-9427 *279-9444 283-9531 283-9533 *283-9597 283-9598	*277-9385 279-9448 283-9541 *283-9603	277-9393 *279-9450 283-9543 *283-9604	277-9395 279-9451 283-9573 *285-9623	277-9403 *279-9456 *283-9575 *285-9624	277-9405 *279-9457 *283-9575 285-9631	279-9425 *283-9522 *283-9591 285-9633	*279-9427 *283-9523 283-9595 285-9641

ZMSPA	CREATED BY CROSS REFERENCE		26-APR-82 AT	09:54	PAGE 6 CREF					
YMBOL	VALUE	REFERENCE 285-9643 +287-9706 #111-4688	287-9675	*287-9677	*287-9677	*287-9693	287-9697	*287-9699	287-9700	*287-9705
ISP	= 177570 = 104421	#111-4688 #110-4611	141-5747	151-5951 375-12004 476-15182 470-15001	377-12049					
TFLA	062274 002216	#139-5624	466-14894	470-15001	*476-15151	476-15152	476-15154	*478-15252		
ISP ENER TAIL TFLA TPSW TRO TR1 TR2 TR3 TR4 TR5 TSP T1	062274 002216 002214 002176 002200 002202 002204 002206 002210 002212 062764 067544 067667 067710 067717 067726 067545	#110-461 470-1500 #139-562 #139-561 #139-561 #139-561 #139-562 #139-562 #139-562 #139-562 #139-562 #139-562 #139-562 #139-562 #139-562 #139-562 #139-562 #139-562 #139-562 #139-562 #139-562 #139-562 #139-562 #139-562	141-5747 155-6033 12 #476-15150 466-14894 509-16394 6 *476-15160 8 509-16394 509-16394 509-16394 509-16394 509-16394 2 509-16394 2 509-16394 2 509-16394 2 509-16394 509-16394 509-16394 509-16394 509-16394 509-16394 509-16394	476-15168 509-16394	509-16394					
	002206 002210 002212 062764 067544	#139-562 #139-562 #139-562 478-152 508-163	509-16394 509-16394 509-16394 0 #478-15237 18 #511-16413	500 4/407	502 1/102	FO. 1/31/	50/_1/210	50/-1427/	509_14729	#511-16422
F11 F13 F14 F15	067667 067700 067710	500-161 502-162(506-163(15 502-16182 07 506-16286 01 508-16323	502-16187 508-16358 508-16343	502-16192 508-16363 508-16353	504-16214 508-16383 508-16368	#511-16423 508-16373	504-16234 #511-16424	300-10320	W)11-10422
F16 F2	067571	508-163 500-161 504-162 502-161	533 508-16338 20 500-16135 39 504-16244 57 504-16254	508-16348 500-16140 506-16261 506-16271	#511-16426 500-16145 506-16306 506-16276	500-16150 508-16313 506-16291	502-16167 508-16378 506-16296	502-16197 #511-16414 #511-16415	504-16224	504-16229
F4 F5 F6	067604 067617 067632 067645 067660 067662 072414	502-161 502-161 502-161 504-162	62 #511-16416 72 502-16202 77 #511-16418 49 #511-16419	506-16281	#511-16417					
F7 F8 F9 H1 H10 H11	067660 067662 072414 072764 073062 073100	506-162 500-161 500-161 502-161 504-162	25 33 508-16338 20 500-16135 39 504-16244 57 504-16254 62 #511-16416 72 502-16202 77 #511-16418 49 #511-16420 25 500-16130 18 #515-16483 27 #515-16483 27 #515-16485 13 502-16180	#511-16421						
H12 H13 H14 H15 H16 H19	073105	504-162 500-161 504-162 504-162 504-163 508-163 506-162 502-162 500-161	22 #515-16485 13 502-16180 37 #515-16488 47 #515-16489 52 506-16264	502-16185	502-16190	504-16212	504-16217	504-16232	508-16326	#515-16486
H16 H19	073202 073261 073400 073415	504-162 #515-164	52 506-16264 92	#515-16491						
120	072451 073422 073501	506-162 502-163	84 508-16356 05 #515-16494	508-16361	508-16381	#515-16493				
12 12 123 124 125 126 127	073560 073627 073665	506-162 500-161 502-161	99 508-16321 48 506-16259 95 #515-16497	508-16341 #515-16496	508-16351	508-16366	508-16371	#515-16495		
H30 H5	073703 072474 073757 072530 072607 072654	500-161 508-163 500-161	92 11 #515-16477 84 508-16356 05 #515-16494 99 508-16321 48 506-16259 95 #515-16498 23 500-16128 23 500-16128 31 508-16336 33 500-16138 04 #515-16480 55 502-16160	#515-16478 508-16346 500-16143	#515-16499 504-16242	508-16376	#515-16479			
0H6 0H7	072607 072654	502-161	55 502-16160	502-16170	502-16175	502-16200	506-16269	506-16274	506-16279	506-16289

CZMSPA	CREATED BY	MACRO ON 26-A	PR-82 AT	09:54 P	AGE 7					SEQ 040
SYMBOL	CROSS REFERENCE	DEFEDENCES		C	REF					
SYMBOL DIAGFL DISPLA DISPRE DISPTB DOBACK DONE DSWR	VALUE 002002 002624 = 000174 014770	REFERENCES #139-5550 *2 #141-5747 *1 #113-4967 1 187-7246 #1 187-7257 1 160-6261 #1 #111-4687 1 500-16119 #5	233-8324 151-5951 151-5958	*233-8326 *151-5958 *429-13448	296-9923 429-13447 *463-14803	463-14802				
DOBACK	015352	187-7257	87-7268	#191-7339						
DT10	015352 006642 = 177570 067132 067252 067274	504-16228 #5	233-8324 151-5951 151-5958 187-7248 187-7268 160-6271 141-5746 1609-16386 1609-16396 1609-16396	151-5950						
DT12 DT13 DT14	067306 067330	500-16114 5 +145-5824 +1	502-16181 145-5836	502-16186 *145-5842	502-16191 504-16238	504-16213 #509-16398	504-16218	504-16233	508-16327	#509-16397
DT16	067346 067376	504-16253 5 504-16253 5	509-16399 506-16265	#509-16401						
DT2 DT20	067144 067404	504-16248 #5 504-16253 5 #509-16387 508-16312 #5 #509-16403	509-16402							
DT2 DT20 DT22 DT23 DT24 DT25 DT26 DT27	067132 067252 067274 067302 067306 067330 067346 067376 067144 067404 067414 067422 067444 06746 067504 067504 067504 067150 067524	#509-16403 502-16206 506-16300 500-16149 502-16196	506-16285 508-16322 506-16260 509-16407 509-16408	508-16357 508-16342 #509-16406	508-16362 508-16352	508-16382 508-16367	#509-16404 508-16372	#509-16405		
DT30	067504 067150 067524	508-16317 #5 500-16124 #5 508-16332 5 500-16129 #5	509-16408 509-16388 508-16337		#509-16409					
DT4 DT5	067170	*145-5835 *1	145-5841	500-16134	500-16139	500-16144	504-16243	508-16377	#509-16390	
DT6 DT7	067206 067222	502-16156	502-16161	502-16171	502-16176	502-16201	506-16270	506-16275	506-16280	506-16290
DUMMY	002174	506-16305 #5 502-16156 506-16295 #5 #139-5614 509-16399 509-16404 509-16408	509-16388 508-16337 509-16389 145-5841 509-16391 509-16392 509-16392 509-16399 509-16404	509-16392 509-16400 509-16405 509-16408	509-16393 509-16400 509-16405 509-16409	509-16409	509-16393 509-16403 509-16409 509-16409	509-16393 509-16404 509-16405	509-16397 509-16404 509-16408	509-16397 509-16404 509-16408
ECCINI	= 104470 = 104472	#110-4654 #110-4656 375-12000	217-7909 167-6693 377-12041	240-8512 #167-6727 #421-13253	390-12367 #173-6942	393-12445 211-7758	397-12516 #217-7926	240-8529	240-8536	240-8540
ECC1DI	= 104471	#110-4654 #110-4654 #110-4656 375-12000 157-6097 #110-4655 277-9407 325-10714 #110-4657 #111-4749 *502-16164 *502-16164 *502-16179 502-16189 #	217-7909 167-6693 377-12041 158-6125 174-7011 279-9459	213-7796 283-9546	213-7819 283-9606	273-9202 285-9646	273-9209 287-9708	273-9286 303-10088	275-9312 305-10096	275-9367 305-10106
ECC1IN EMTVEC	- 000020	#110-4657 #111-4749	213-7846	213-7854	282-9493	282-9496	303-10053	305-10099		
EM11 EM12	070235 070257	502-16154	502-16159	502-16169	502-16174	#513-16441				
EM13 EM14 EM15	070235 070257 070303 070335 070401 070447 070507 067735 070561	502-16179 502-16184 # 502-16189 #	213-7846 147-5876 502-16159 513-16442 508-16355 513-16444 513-16445	#513-16443						
EM17	070447 070507	504-16211	508-16350	#513-16447						
EM2 EM20	067735 070561	500-16117 #	513-16435 508-16340	508-16345	508-16365	#513-16448				

CZMSPA SYMBOL	CREATED BY	MACRO ON 26-APR-82 AT	09:54	PAGE 8 CREF					
SYMBOL	VALUE 070640 070674 070721	REFERENCES 504-16226 #513-16449 504-16231 508-16370	508-16380	#513-16450					
EM23 EM24 EM25	070750	504-16241 #513-16451 500-16112 #513-16452 504-16246 508-16330 504-16251 #513-16454	508-16380 #513-16453 #513-16456						
EM20 EM27 EM29	071054 071125 071215	506-16263 #513-16455 506-16273 508-16325 500-16122 #513-16436	#513-16456						
EM30 EM32 EM33	071277 071407 071514	506-16278 508-16360 506-16288 #513-16459 506-16293 #513-16460	#513 - 16457						
EM21 EM22 EM23 EM24 EM25 EM26 EM27 EM29 EM30 EM32 EM33 EM35 EM36 EM40	071622 071707 070025	506-16268 #513-1646 506-16303 #513-1646							
EM40 EM5 EM50	071756 070073 072030	508-16310 #513-16463 500-16132 #513-16438 502-16204 #513-16466							
EMSO EMSO EMS1 EMS2 EMS3 EMS5 EMS6	067773 071277 071407 071514 071622 071707 070025 071756 070073 072030 072064 072161 072210 072231 072263 070150	506-16258 #513-16466 500-16147 #513-16466 502-16194 #513-16466							
PMO	072231 072263 070150	508-16315 #513-16469 508-16335 #513-16479 500-16137 #513-16439							
EM60 EM61 EM62 EM7	072331 072373 100300 070175	506-16303 #513-16463 500-16127 #513-16463 508-16310 #513-16463 500-16132 #513-16463 500-16132 #513-16463 506-16298 #513-16463 500-16147 #513-16463 500-16147 #513-16463 508-16315 #513-16463 508-16335 #513-16473 508-16320 #513-16473	2						
ENASBE ENA1SB END	= 104506 = 104507 100354	#110-4668 167-6725 #110-4669 303-1005 #517-16631	173-6940 7 303-10090		421-13251 #325-10716				
ENERGI ENEXBK ERRADD ERRGEN ERRMAX	= 104420 047436 002454 = 104512		189-7321 1 371-11957 213-7839 213-7838	7 #371-11959 *355-11422 273-9230	273-9256	279-9422	315-10334		
ERRMAX ERROR	= 104000	#111-4682 154-5994 160-6268 167-6715 282-9491 282-9505	154-6013 167-6719 291-9785	154-6016 174-7020 291-9797	157-6103 186-7238 303-10062	160-6210 273-9225 305-10120	160-6220 275-9330 305-10132 323-10598 337-11079 #454-14591	160-6247 277-9418 315-10329 323-10612 339-11104	160-6249 282-9484 315-10344
		316-10384 317-1041 324-10670 325-1072 339-11111 382-1217	154-6013 167-6719 291-9785 9 321-10496 4 325-10746 5 384-12191 9 458-14664 6 462-14745 9 *465-14846	321-10506 325-10751 384-12200	322-10542 326-10782 421-13248	160-6210 273-9225 305-10120 322-10552 326-10792 454-14589 460-14681	323-10598 337-11079 #454-14591 460-14692	323-10612 339-11104 454-14601 #4 460-14698 #4	324-10661 339-11108 454-14603 460-14700
ERRPC	002016	371-11928 371-1195 #139-5703 213-7814 #110-4672 213-7813 #141-5721 458-1464 #111-4682 154-5994 160-6268 167-6715 282-9491 282-9505 316-10384 317-1041 324-10670 325-1072 339-11111 382-1217 456-14612 458-1465 460-14709 460-1471 #139-5559 *465-1483 509-16386 509-1638 509-16409 #139-5561 *465-1484 #111-4742 *147-5886	9 438-14664 6 462-14745 9 *465-14840 7 509-16388	707-10307	157-6103 186-7238 303-10062 322-10542 326-10782 421-13248 458-14670 476-15176 *465-14862 509-16390	*465-14863 509-16391	465-14868 509-16392 509-16406	468-14936 509-16395	470-15011 509-16397
ERRPSW	002026	509-16399 509-1640 509-16409 #139-5563 *465-1484	1 509-16402 2 *465-14865 1 *465-14866			509-16405	509-16406	509-16407	509-16408
ERRSP	= 000004	#139-5561 *465-1484 #111-4742 *147-5886	1 *465-14864 *147-5887	476-15166 462-14765	*462-14766	*462-14768	*462-14774		

CZMSPA CREATED BY MACRO ON 26-APR-82 AT 09:54 PAGE 9		
SYMBOL CROSS REFERENCE CREF		
SYMBOL VALUE EUFLAG 002130 #139-5595 222-8076 355-11382 355-11397 EVEN 002360 #139-5674 *296-9894 296-9895 *297-9957 297-9957 EXBANK 047020 170-6861 172-6901 182-7042 191-7342 193-7358 195-7388 197-7418	100-7455 201-7494	
EUFLAG 002130 #139-5595 222-8076 355-11382 355-11397 297-9957 297-	199-7455 201-7494 363-11618 365-11725 397-12546	
EXCMD3 051066 386-12235 386-12238 386-12242 #386-12244 EXCMD4 051406 388-12278 388-12282 #388-12296		
EXIT 047632 #377-12022 462-14756 466-14900 486-15611 EXIT2 047636 #377-12024		
E1 005666 #157-6111 E2 006054 #159-6155		
E31 032474 #282-9508 E32 034402 #291-9808		
E33 034370 #291-9806 E4 006564 #160-6260		
EXCMD3 051066 386-12235 386-12242 #386-12244 EXCMD4 051406 388-12282 #388-12284 #388-12282 #388-12282 #388-12282 #388-12282 #388-12282 #388-12282 #388-12282 #388-12282 #388-12282 #388-12282 #388-12282 #388-12282 #388-12282 #388-12282 #388-12282 #388-12282 #388-12284 #388-12282 #388-1		
E46 037424 #321-10512 E47 037622 #322-10556		
E5 006554 #160-6259 E50 040132 #323-10634		
SYMBOL VALUE EUFLAG 002360 EVEN 002360 EVEN 002360 EXBANK 047020 170-6861 203-7532 365-11729 365-7577 207-7615 238-8444 238-8473 240-8534 EXCMD3 051066 388-12278 388-12285 388-12285 388-12285 388-12285 388-12286 EXCMD4 051406 EXIT 047636 EXIT 047636 EXIT 047636 EXIT 005666 W157-6111 E2 006054 W159-6155 E31 032474 W282-9508 E33 034370 W291-9808 E33 034402 W291-9808 E44 036564 W159-6155 E45 036712 W316-10395 E44 036712 W316-10395 E44 036712 W316-10395 E44 036712 W316-10395 E45 037772 W319-10460 E46 037424 W321-10512 E47 037622 W322-10556 E5 006554 W160-6259 E50 040132 W323-10634 E51 04016 W323-10634 E52 040322 W324-1056 E53 040606 W325-10736 E54 040576 W325-10736 E55 040577 E56 040576 W325-10736 E57 040576 W325-10736 E77 1488 W325-10736 E77 1507 E77 17		
E53 040606 #325-10737 E54 040576 #325-10736 E55 040564 #325-10734		
FASTCI = 177640 #111-4796 169-6776 169-6810 255-8847 344-11204 379-12076 #139-5576 *167-6715 *167-6719 *303-10062 *337-11079 *339-11104 *339-11108 468-14953 #700-12551	*339-11111 466-14898	
FCMD10 053370 380-12130 #399-12551 FCMD11 053416 380-12131 #399-12561		
FCMD11 053416 380-12131 #399-12561 FCMD12 053440 380-12132 #399-12566 FCMD13 053460 380-12133 #399-12572 FCMD14 053502 380-12134 #399-12579 FCMD15 053520 380-12139 #400-12598 FCMD16 053604 380-12140 #400-12618 FCMD17 053646 380-12141 #402-12632 FCMD18 053662 380-12142 #404-12638		
FCMD12 053440 380-12132 #379-12500 780-12132 #399-12572 780-12133 #399-12572 780-12134 #399-12579 780-12134 #399-1		
FCMD17 053646 380-12141 #402-12632 FCMD18 053662 380-12142 #404-12638		
FIELDS 050102 #380-12097 483-15451 FINT 007174 162-6332 #163-6339 FINT 060000 #113-6974 169-6765 169-6708 169-6828 172-6900 211-7682 219-7965	219-7966 220-8014	
FIRST = 060000 #113-4974 169-6765 169-6798 169-6828 172-6900 211-7682 219-7965 220-8027 220-8051 222-8080 222-8116 222-8134 224-8162 232-8273 238-8448 238-8477 240-8519 242-8553 242-8558 242-8559 244-8585	234-8339 236-8389 244-8615 244-8622	
FIRST = 060000 #113-4974 169-6765 169-6798 169-6828 172-6900 211-7682 219-7965 220-8027 220-8051 222-8080 222-8116 222-8134 224-8162 232-8273 238-8448 238-8477 240-8519 242-8553 242-8558 242-8559 244-8585 246-8647 246-8650 246-8651 246-8652 246-8653 246-8670 249-8751 296-9907 296-9926 297-9962 297-9964 297-9965 315-10322 319-10434 328-10838 *377-12061 *377-12062 *377-12063 386-12223 388-12264 390-12359 430-13484 *432-13518	219-7966 220-8014 234-8339 236-8389 244-8615 244-8622 282-9476 289-9723 323-10594 323-10610 390-12360 430-13477 *432-13519 432-13524	
FCMD13	*432-13519 432-13524	
FCMD12 053440 380-12132 #399-12566 FCMD13 053460 380-12133 #399-12572 FCMD14 053502 380-12134 #399-12579 FCMD15 053520 380-12139 #400-12598 FCMD16 053604 380-12140 #400-12618 FCMD17 053646 380-12141 #402-12632 FCMD18 053662 380-12142 #404-12638 FIELDS 050102 #380-12097 483-15451 FIRST = 060000 #113-4974 169-6765 169-6798 169-6828 172-6900 211-7682 219-7965 220-8027 220-8051 222-8080 222-8116 222-8134 224-8162 232-8273 238-8448 238-8477 240-8519 242-8553 242-8558 242-8559 244-8585 246-8647 246-8650 246-8651 246-8652 246-8653 246-8670 249-8751 296-9907 296-9926 297-9962 297-9964 297-9965 315-10322 319-10434 430-13478 430-13479 4430-13480 4430-13481 430-13482 430-13484 4432-13518 FLIPLO 002602 #141-5734 *147-5865 222-8106 *327-10810 *327-10811 327-10812 327-10814	327-10816	

CZMCDA	CREATER DV	MACRO ON 26-APR-82 A	T 09.54	PAGE 10					356 041
SYMBOL	CREATED BY CROSS REFERENCE		07.54	CREF					
SYMBOL	VALUE 015062	REFERENCES #187-7273							
FLUSH FSCMDO	015062 050300 050402 050512 050660 051134 051454 052372 052400 052672 053076	#380-12120 #382-1214							
FSCMD1 FSCMD2 FSCMD3 FSCMD4 FSCMD5	050402	380-12121 #382-1216 380-12122 #384-1218							
FSCMD3	050660	380-12123 #386-1220	8						
FSCMD4	051134	380-12124 #388-1225	0						
FSCMD5	051454	380-12125 #390-1230 380-12126 #391-1243	3						
FSCMD7	052400	380-12127 #393-1243	9						
FSCMD6 FSCMD7 FSCMD8 FSCMD9	052672	380-12126 #391-1243 380-12127 #393-1243 380-12128 #395-1248 380-12129 #397-1251	0						
FSINFL	002430	#139-5696 *189-7284							
FSPAT	052154	390-12377 #390-1238	792-12170	*384-12186	384-12203	+390-12304	390-12422	*303-12441	393-12471
FSSTAC	002302	*397-12512 397-1254	1 302-12179	-304-12100	304-12203	-370 12304	370 12422	-5/5 12441	373 12111
FS1	050166	#380-12112 380-1211	8 380-12144						
FS7FLA	002442 002536	#139-5696 *189-7284 390-12377 #390-1238 #139-5648 *382-1217 *397-12512 397-1254 #380-12112 380-1211 #139-5698 238-8467 #139-5713 *240-8502	*240-8530	*240-8541	363-11657				
FULLRE GBLENG GETCSR	= 000076	246-8647 246-8650 227-8220 231-8259 #423-13267 454-1458 *423-13269 423-1327 255-8823 257-8856 #139-5568 *160-6208	#313-10304		2/0-97/0	250-8760	#329-10854		
GETCSR	041324 054256	#423-13267 454-1458	247-8722 5 454-14597 7 #423-13280	248-8737	249-8749	250-8760	#327-10034		
GETDA1	054354	*423-13269 423-1327	7 #423-13280		//5 1/000				
GETDIS	060450 002042	#139-5568 *160-6208	462-14787 +160-6218	*160-6266	465-14828 *282-9480	282-9488	*282-9503	*291-9783	*291-9795
0000	002042	*31/-10410 *319-1043	5 *321-10495	*160-6266 *321-10504	*322-10540	*322-10550	*323-10595	*323-10608	*324-10649
		324-10667 *324-1067 *451-14487 *451-1449	3 +326-10780 2 +451-14499	*326-10790 *451-14502	*421-13220 *451-14509	*322-10550 *421-13222 *451-14514 *453-14560	*421-13240 *451-14519	*423-13258 *451-14524	*451-14481 *451-14529
		*453-14535 *453-1454	0 +453-14545	*453-14550	*453-14555	*453-14560	*453-14565	*453-14570	*455-14575
		*453-14580 456-1462 509-16392 509-1639			*460-14679 509-16406	*400-14089	*460-14708	*460-14713	509-16386
G0002	002044	#139-5569 *456-1461	5 *456-14619	509-16399		207 10407			
GOOD3	002046	#139-5570 *456-1461	6 *456-14620	509-16399					
GTSWR HEADER	= 104407 002576	#110-4597 155-6053 #141-5732 *147-5866	*187-7263	*187-7265	*191-7344	*191-7347	*209-7656	*209-7671	*211-7727
I III	00257	*273-9239 *273-924	*187-7263 *273-9252 *282-9502 9 *324-10669	*187-7265 *273-9255 *291-9794 *325-10723	*191-7344 *275-9328 *305-10121 *325-10745 *460-14680	*191-7347 *275-9331 *305-10133 *325-10750 *460-14682	*209-7656 *275-9342 *321-10505 *326-10781 *460-14691	*209-7671 *275-9345 *322-10549 *326-10791	*211-7727 *277-9416 *323-10597
		*273-9239 *273-924 *277-9419 *282-948 *323-10611 *324-106 *393-12459 *397-125	9 *324-10669	*325-10723	*325-10745	*325-10750	*326-10781	+326-10791	*390-12353
		*393-12459 *397-125	(Y *4)5-140)	*458-14672	*460-14680	*460-14682	*460-14691	*460-14693 476-15172	*400-14/13
		*273-9239 *273-9242 *277-9419 *282-9489 *323-10611 *324-1069 *393-12459 *397-1252 *460-14717 *466-1499 *476-15180 205-7573 207-7611 #111-4692 434-1356 #139-5700 *149-5923		468-14963	*470-14999	476-15156	*476-15157	4/0-131/2	4/0-151/5
HIPAT	047500	205-7573 207-7611 #111-4692 434-1356 #139-5700 *149-5923 166-6606 166-6616 *184-7186 184-7206	#373-11982						
HT	= 000011 002446	#111-4692 454-1536 #139-5700 ±149-593	*149-5929	149-5929	*166-6517	*166-6519	166-6567	166-6575	166-6579
1.	002440	166-6606 166-6610 *184-7186 184-7200	166-6656	166-6661	166-6663	166-6667	166-6567 *184-7169	166-6575 *184-7175	184-7186
		*184-7186 184-720(*238-8466 238-846	166-6656 *211-7726 *238-8471	166-6661 211-7729 238-8481 \$465-14858	*166-6517 166-6663 *211-7735 238-8487 466-14913 159-6155 160-6259 282-9508 #291-9808	166-6667 211-7735 *238-8495 #466-14917 159-6155	*238-8442 238-8495	238-8452	238-8458
IBSAVE	061316	*238-8466 238-8466 *465-14822 465-1486	* 465-14853	*465-14858	466-14913	#466-14917	450 4455	****	450 /43/
IIII	= 061316 = 177777	157-6111 157-611 159-6176 159-617	157-6111	160-6259	160-6250	160-6259	159-6155 #160-6259	#159-6155 160-6260	159 - 6176 160 - 6260
		160-6260 #160-626	282-9508	282-9508	282-9508	160-6259 #282-9508 316-10393	#160-6259 291-9806 316-10393	291-9806	291-9806
		#291-9806 291-9808	3 291-9808 3 316-10395	#157-6111 160-6259 282-9508 291-9808 #316-10395	#291-9808 319-10460	316-10393 319-10460	316-10393 319-10460	316-10393 #319-10460	160-6260 291-9806 #316-10393 321-10512
		316-10395 316-1039	310-1039	#310-10393	317-10400	317-10400	317-10400	317 10400	361 10316

CZMSPA	CREATED BY	MACRO ON 26	-APR-82 AT	09:54 P	AGE 11					SEQ 0	411
SYMBOL	CROSS REFERENCE	REFERENCES 321-10512 323-10632	321-10512	#321-10512		322-10556	322-10556	#322-10556	323-10632	323-10632	
ILLCSR	014124	#324-10675 325-10737	321-10512 #323-10632 325-10734 325-10737 182-7098 173-6929 195-7401	323-10634 325-10734 325-10737 #182-7138	322-10556 323-10634 325-10734 #325-10737	322-10556 323-10634 #325-10734	#323-10634 325-10736	324-10675 325-10736	324-10675 325-10736	324-10675 #325-10736	
IMPTES INCBNK INCPAT INCRPT	013064 047510 047464 047464	182-7081 172-6921 193-7371 193-7367 201-7507	173-6929 195-7401 197-7429 203-7547 *363-11651	#174-6947 197-7433 #373-11974 #373-11973	199-7470	201-7504	203-7544	205-7587	207-7627	#373-11986	
INHBAN INHECC INTFLA INT64K	002534 002532 002134 002136	201-7507 #139-5712 #139-5711 #139-5597 #139-5598	*363-11651 211-7680 170-6863 170-6864	363-11659 343-11163 172-6906 355-11414	363-11649 173-6923 361-11555	*363-11650 182-7053 *371-11908	363-11658 211-7693 *371-11955	*363-11665 355-11411	*371-11908		
INVALI	= 104511 = 000020	#110-4671 #111-4747 363-11684	211-7709 *147-5874 #363-11694	217-7918 *147-5875	219-7970	238-8446	388-12270	390-12331	393-12464	397-12534	
JMPRL1 KAMIKA KAMITE	045740 002004 026774	#139-5551 233-8312	253-8814 233-8320	380-12109 234-8330	*380-12111 242-8546 297-9956	*382-12154 244-8575	*399-12563 246-8639	*399-12568 #253-8813			
KDIAG KDPARO	= 000010 = 172360	#296-9893 #111-4887 236-8413	296-9909 222-8098 236-8416	297-9930 222-8100 236-8421	222-8124	222-8125	222-8144	222-8145	234-8357	234-8360	
KDPAR6 KDPAR7 KERNEL	= 172374 = 172376 = 104417	#111-4893 #111-4894 #110-4608 244-8610 386-12233 423-13275	236-8416 *222-8101 *222-8126 169-6777 246-8648 386-12244 426-13328	*234-8361 *222-8146 169-6790 255-8849 388-12274 426-13334	*236-8414 169-6811 257-8886 388-12292 429-13431	169-6834 363-11682 388-12296 429-13435	174-7024 365-11741 421-13217 430-13494	213-7845 367-11800 421-13233 432-13528	213-7853 367-11816 421-13244	242-8554 377-12064 423-13263	
KERSTK KFLAG KIPARO KIPAR4	= 002000 002524 = 172340 = 172350	#111-4679 #139-5709 #111-4877 #111-4881 164-6439	273-9222 361-11515 *162-6281 *166-6497 369-11868	275-9322 361-11591 *163-6342 *166-6505	369-11824 *163-6357 166-6591	369-11844 *163-6407 166-6639	430-13483 *163-6411 *166-6678	432-13501 *163-6415 166-6679	*164-6429 166-6681	*164-6438 369-11859	
KIPAR5	= 172352	*369-11860 #111-4882 *369-11862	*166-6498 *369-11879	*166-6506	166-6642	*166-6659	*166-6665	*166-6671	*166-6679	*321-10477	
KIPAR6 KIPDRO KMAP KPFLAG	= 104422 002112	#111-4882 *369-11862 #111-4883 #111-4857 #110-4612 #139-5588 #139-5679 #141-5725 #141-5725 #141-5722 *182-7113 361-11572 395-12504 #141-5723 #139-5558 157-6111 322-10556 160-6259	361-11593 155-6058 *371-11905	369-11826 *371-11915	425-13317	428-13393	428-13417	432-13502			
KSIZE KSTACK LAST		#141-5725 #113-4975 289-9725	144-5794 211-7683 296-9908	166-6507 220-8049 297-9929 *145-5840 191-7349 363-11630 400-12623 *166-6495	189-7317 232-3271 297-9966 166-6490 234-8351 363-11637	478-15210 234-8340 315-10346 166-6492 238-8465 373-11989	242-8560 386-12226	244-8586 388-12266 170-6875	244-8614	246-8654	
LASTBA	002552	*182-7113 361-11572 395-12504	184-7152 361-11579 397-12537	191-7349 363-11630 400-12623	234-8351 363-11637	238-8465 373-11989	242-8560 386-12226 169-6752 238-8472 386-12221	240-8523 388-12263	173-6936 331-10880 390-12309	182-7071 331-10889 393-12467	
LASTBL	002554 002014	#141-5723 #139-5558	*166-6488 *189-7290	*166-6495	166-6681	202_0509	291-9808	316-10395	319-10460	321-10512	
LBLS0	= 000404 = 000403	322-10556 160-6259	323-10634 291-9806	159-6176 324-10675 316-10393	160-6260 325-10737 323-10632	282 - 9508 325 - 10736		310-10373	317-10400	361-10312	

CZMSPA		CREATED BY	MACRO	ON	26-APR-82 AT	09:54	PAGE 12					254 0	/41/
SYMBOL	CRO	DSS REFERENCE	REFER	-			CREF						
LBLS2	=	000402	325-	1073	54								
LF LINK1	=	000012 002516	#111- #139- *240-	·5706 ·8508	*169-6745 *240-8510	*169-6749 240-8520	169-6764 *242-8560	169-6803 *242-8568	*238-8436 *244-8584	*238-8440 *244-8597	238-8456 *246-8654	238-8485 *246-8662	
LINK2		002520	#139-		*169-6746	*169-6750	169-6789	*244-8587	*244-8598	244-8616	244-8623		
LKS LOADBA LOADCS		177546 002426 104425	#111- #139- #110- 323- 346-	-5692	*363-11621 282-9479	365-11719 291-9776 324-10653 352-11345	365-11724 315-10324 325-10711 354-11356	315-10339 326-10769 355-11430	316-10376 326-10772 384-12196	321-10487 346-11228 428-13411	321-10499 346-11236	322-10533 346-11244	
LOADHO LOOP		002562 014742	#141-	-5726	189-7322	363-11622	365-11720	365-11728	377-12047				
LOWMAP LSIZE LST\$\$	=	046344 002374	363- #139- 157-	-1169	2 365-11751 1 *184-7151 1 157-6111	#365-11761 *184-7164 157-6114 160-6260 316-10395	428-13390 184-7188 157-6114 282-9508 316-10395	184-7190 159-6155 282-9508 319-10460	186-7237 159-6155 291-9806 319-10460	509-16391 159-6176 291-9806 321-10512	159-6176 291-9808 321-10512	160-6259 291-9808 322-10556	
			322-	-1053 -1073	316-10393 6 323-10632 6 325-10736	323-10632 325-10737	323-10634 325-10737	323-10634	324-10675	324-10675	325-10734	325-10734	
MAINT MAPHO MAPKER	=	177750 170202 046706	#1113-	-4/01	+363-11691	*365-11750 247-8726		*428-13389 250-8759	251-8768	251-8776	251-8784	251-8792	
MAPLO	=		#113-	-4901	*363-11689	*365-11750	365-11763	426-13361	*428-13389				
MAPPER	=	170204 044534	257· 397·	-6059 -8855 -1254	9 169-6761 5 #361-11514 45 430-13475	173-6937 377-12059	174-6948 386-12230	211-7708 388-12269	242-8551 390-12330	244-8580 390-12363	246-8644 390-12428	255-8822 393-12475	
MASK MBERR MEMDON		002314 014002 015010	#139- *182- 187-	-5653 -7076 -7247	3 5 *182-7092 7 #187-7257	182-7096	*182~7100	*182-7105	#182-7115				
MFPT	=	000007 020620	#111- 149-	-4090 -5914	145-5845 4 219-7969 9 #219-7962	219-7969	219-7974	#219-7982					
MJTEST MKCNT MKCONT		020514 017666 017046	149- 209- *211- 209-	-7669 -7706 -7664 -5900	6 *211-7751	211-7751	#211-7763						
MKCSRT MKFLAG		020204 002116	#139-	-559(172-6904	182-7050	209-7659	240-8516	*371-11905	*371-11931	373-11969		
MKLOOP MKPAT		002116 017230 020434	#211-	-7704 -5907	7 717-7017	217-7917	217-7922	#217-7934					
MKTEST MMRO	=	020274 177572	149- 191- #111- 430- #111- #111-	-734! -477(-134	5 209-7667 0 *341-11136 93 *432-13512	#217-7907 *341-11140 509-16390		*363-11703	*365-11742	*365-11745	426-13349	*428-13402	
MMR1 MMR2 MMR3	=	177574 177576 172516	#111- #111- #365-	-477 -477 -477	209-7667 0 *341-11136 93 *432-13512 1 426-13349 2 426-13349 3 145-5826 52 *375-12007 8 #339-11107 7 *147-5888 4 *144-5792	*341-11140 509-16390 *428-13402 *428-13402 *145-5827 *377-12052	430-13492 430-13491 *145-5828	*432-13511 *432-13510 145-5829 *428-13401	509-16390 509-16390 *189-7320 430-13490	*363-11685	*363-11693 509-16390	*365-11749 509-16398	
MMTRAP		042570	147	-588	8 #339-11107	*3//-12032	420-13332	4420-13401	430-13490	4436-13309	707-10370	JU7-10J70	
MMVEC MONFLG MPT		000250 002272	#1111- #139- 187-	-475 -564 -726	7 *147-5888 4 *144-5792 9 311-10216	*147-5889 *154-5993 339-11113	*154-6012 361-11601	*154-6015 377-12038	465-14878 377-12053	380-12135	399-12585	406-12664	

						,				SEQ 0413
CZMSPA	CREATED BY CROSS REFERENCE	MACRO ON	26-APR-82 AT	09:54	PAGE 13					i i
SYMBOL	VALUE	DEFEDENCE	S 428-13375			434-13639	435-13658	462-14746	463-14799	470-14987
	002572	425-1328 473-1510 #141-5730 #141-5731 *182-7144	8 428-13375 7 479-15256	430-13462 483-15435	434-13596 485-15525	434-13639 485-15541	435-13658 493-15952			
MSEEDH	002572 002574	#141-5731	*147-5867 *147-5868	147-5869 147-5870						
MSGA12	100014 076031	*182-7144	#517-16614 #517-16556							
MSGB34	076067	211-7746 154-5996	#517-16557							
MSG001	074036	331-1087	7 #517-16503							
MSG002 MSG003	076067 100226 074036 074120 074175	331-1087 331-1087	9 #517-16505							
MSG004	074302 074410	331-1088	331-10707	W311 10301						
MSG006	074422 074457	333-1093 494-1599 335-1094	0 #517-16510							
MSG008	100174	150-6142	#517-16620	ME17_1/E12						
MSEEDL MSGA12 MSGA34 MSGB34 MSG000 MSG001 MSG003 MSG004 MSG005 MSG006 MSG007 MSG008 MSG009 MSG010	074471 074503 074515	335-1103	94 #517-16510 90 #517-16511 9 #517-16620 335-10980 32 #517-16513 331-10906 331-10908 37 #517-16516	717-10512						
MSG011 MSG012	074515 074603	331-1089 331-1089	331-10906 331-10908	#517-16514 #517-16515						
MSG013 MSG014	074603 074700 074702	159-6156 335-1103 331-1089 331-1089 333-1093 159-6153 #517-1651	37 #517-16516 159-6173	331-10905	333-10939	335-11042	465-14833	472-15035	472-15039	473-15105
		#517-1651 *159-6151	7 159-6152							*335-10958
MSG015	074704	*335-1096	8 335-10969	*159-6161 *335-10990	*335-10992	*159-6167 *335-10995 #517-16518	*335-10998	335-10999	*335-10956 *335-11011	*335-11021
MSG016	074706	335-1102 159-6143	< <<	#51/-1651U	333-11040	#317-10316				
MSG017 MSG018	074720	331-1089 470-1499	92 331-10895 94 473-15103	#517-16520 478-15200	478-15215	478-15230	478-15245	#517-16521		
MSG019 MSG020	074734	473-1507	73 #517-16522							
MSG021	074761	380-1209 380-1211	17 #517-16524 51 #517-16545							
MSG022 MSG023	075547 075571	406-1265 384-1219	3 #517-16546	WF47 4/F/7						
MSG025 MSG026	075605 075631	382-1217 380-1211	12 #517-16548	#31/-1034/						
MSG027	075643 075660	384-1218 384-1219	89 #517-16549 98 #517-16550							
MSG029	075571 075605 075631 075643 075660 075674	386-1221	11 #517-16551	#517-16552						
MSG029 MSG030 MSG031 MSG032	075733 075773	386-1221	388-12254	#517-16553						
1 MSG033	0/6012	386-1224	388-12281	#517-16555						
MSG035 MSG036	076115 076120	189-7293 388-1225	53 #517-16559							
MSG037 MSG038	076137	406-1263 384-1217 380-1211 384-1218 384-1219 386-1221 386-1223 386-1223 386-1223 386-1223 388-1223 388-1223 388-1223	84 #517-16560 93 #517-16561							
MSG039 MSG040	076174									
MSG041	076242	390-1230 390-1231	15 #517-16564							
MSG042 MSG043	076305	390-1231 390-1232	19 #517-16565 24 #517-16566							

	MSPA	CREATED BY	MACRO ON 2	6-APR-82 AT	09:54 P	AGE 14			
SY	MBOL MBOL GO46	CROSS REFERENCE VALUE 076327	REFERENCES 390-12337 #517-16568	393-12451					
MS	G047	076362	#517-16568 380-12102	#517-16569	37, 12722	#311 10301			
MS	G048 G049	076441	390-12334	#517-16570					
MS	G051 G055	076513	429-13451 393-12442	#517-16572 #517-16573	ME17 1/57/				
MS	G056 G058	076534 076567	478-15191	#517-16575	#517-16574				
MS	G061 G062	076611 076620	393-12449 478-15191 473-15082 487-15711 487-15712	#517-16576 487-15770	#517-16577				
MS	G063 G064	076327 076362 076401 076441 076474 076513 076534 076567 076611 076620 076640 076651	487-15712 487-15713 487-15771	487-15772	#517-16579				
MS	G065 G066	076661 076673	466-14903	#517-16581					
I MS	G067 G070	076742 076751	184-7201	#517-16582 #517-16583					
MS	G073	077002 077020	397-12513 363-11632	#517-16584 363-11669 #517-16586	#517-16585				
MS	G076 G077	076673 076742 076751 077002 077020 077052 077073	395-12495 189-7293	#517-16586 #517-16587					
MS	G079 G085	0//133	395-12487 399-12553	#517-16588 #517-16589					
1745	G088 G089	077160 077176	478-15209 478-15225	#517-16587 #517-16588 #517-16589 #517-16590 #517-16591					
MS	SG090 SG091	077220 077234	478-15240 478-15234	#517-16592 478-15249	#517-16593				
MS	SG092 SG093	077246 077262	485-15535 485-15537	#517-16594 #517-16595					
M:	SG095 SG101	077270 077300	485-15539 399-12562	#517-16596 #517-16597					
MS	sG102 sG103	077220 077234 077246 077262 077270 077300 077357	399-12567 382-12148	#517-16592 478-15249 #517-16594 #517-16595 #517-16596 #517-16598 #517-16599					
M:	SG104 SG105	077401	400-12599	#517-16601					
M:	SG106 SG107	077456 077474	399-12573 399-12580	#517-16603					
M:	SG110 SG111	077551 077615 077647	400-12614 400-12619	#517-16605					
M:	SG112 SG113	077664	184-7191 184-7195	#517-16606 #517-16607					
M:	SG114 SG117	077701 077716	184-7199 154-6010	#517-16608 #517-16609					
I M	SG119 SG120	077737	154-6029 154-6030 154-6024	#517-16610 #517-16611 #517-16612					
I M	SG121 SG122 SG125	077760 100000 100046	182-7145 395-12485	#517-16613 #517-16616					
M	SG126 SG127	100046 100070 100135	163-6416 402-12633	#517-16618 #517-16618					
M	SG128 SG129	100154	404-12639 159-6177	#517-16619 #517-16622					
M	SIZE	002376	#139-5681	*184-7151	*184-7161	184-7192	184-7194	186-7237	509-16391

) 1					
	The same of					00 51	DAGE 45					SEQ 04	15
	CZMSPA	CREATED BY	MACRO (ON 26-A	PR-82 AT	09:54	PAGE 15 CREF						
	SYMBOL SYMBOL MTA030 MTEST MTLA11 MTLB11 MTLD11 MTPA03 MTPA04 MTPA24 MTPA26 MTPB03 MTPB24 MTPB26 MTPB21 MTPB26 MTPC26 MTPC26 MTPC21 MTPC26 MTPC26 MTPC26 MTPC26 MTPD03 MTPD03 MTPD00 MTPO00 MTPO00 MTPO01 MTPO01 MTPO07 MTPO07 MTPO07	CROSS REFERENCE	DECEDER	ICEC			LKET						
	MTADEOL	034600	#240-85	503 2	40-8535								
	MTEST	016752	193-7	365 1	95-7395	197-7427	199-7464	201-7501	203-7541	205-7584	207-7624	#209-7655	
	MTLA11	030412	273-91	172 #2	40-8535 95-7395 73-9177	197-7427 273-9272							
	MTLB11	030424	#273-91	180 2	73-9268								
	MTLC11	030436	#273-91	183 2	73-9262								
	MTLD11	030532	#273-92	200 2	73-9284	#2/1 9070							
	MTPA03	027510	222-80	130 2	22-8094	222-81/1	#245-8008	265-9018					
П	MIPAU4	02/040	232-8	282 2	32-820A	#202-0818	W20J-0770	203-7010					
	MTDA24	035254	234-83	355 #2	99-9982	301-10041							
	MTPA25	035664	303-10	0052 "3	03-10087	305-10095	#305-10106						
	MTPA26	036014	236-83	399 2	36-8400	236-8412	#307-10138	307-10181					
	MTPB03	027550	222-80	091 2	22-8097	#261-8962	263-8995	245 0000	#2/5 0011	2/5 002/			
	MTPB04	027702	222-8	124 *2	22-8139	*222-8141	222-8144	265-9002	#265-9011	265-9024			
	MTPB21	034460	232-8	204	32-8299	292-9833	#301-10026						
	MTDD25	033314	303-10	0080	05-10097	#305-10113	W301-10020						
	MTPR26	VALUE 024600 016752 030412 030436 030532 027510 027646 034430 035254 035664 036014 027550 027702 034460 035314 035706 036030 027610 034514	236-8	403 2	73-9268 73-9262 73-9262 73-9284 22-8094 22-8123 32-8296 33-10087 22-8097 22-8139 23-8299 34-8356 35-10097 36-8404 32-8303 301-10032 305-10100 363-8986 33-8306 33-8306 33-10077 22-8022 22-8039 22-8039 23-8039 23-8039 23-8039 23-8039 23-8039 23-8039 23-8039 23-8039 23-8039 23-8039 23-8039	236-8420	#265-8998 #305-10106 #307-10138 263-8995 222-8144 #301-10026 #307-10145 263-8994 #305-10125 #307-10163 #307-10176 303-10082 259-8904 #265-9020						
	MTPC03	027610	222-80	098 2	61-8964	#263-8980	263-8994						
	MTPC21	034514	232-8	287 2	32-8303	#292-9852							
	MTPC24	035330	234-8	357 3	01-10032	#301-10034	#70F 1013E						
	MTPC25	035746	303-10	0091 3	107-10100	307-10102	#303-10123						
	MIPUZO	030004	222-8	100	042-80137	#263-8988	#307-10103						
	MTPD21	034550	232-8	289	32-8306	#294-9870							
	MTPD25	035612	303-1	0064 3	303-10066	303-10069	303-10071 #307-10176 303-10082 259-8904	#303-10087					
1	MTPD26	036104	*236-8	399 +2	236-8403	236-8415	#307-10176						
	MTPE25	035634	303-1	0075	503-10077	303-10080	303-10082	#305-10095	720 100/7	+720 100/F	328-10848		
	MTP000	027400	220-8	019	220-8022	#259-8902	259-8904	*328-10842	328-10843	*328-10845	320-10040		
	MTP001	027424	220-8	041	220-8044	#259-8913							
	MTPOOS	036064 027626 034550 035612 036104 035634 027400 027424 027456 027722 027756 030156	222-8	138	22-8139	222-8143	#265-9020						
-	MTP006	027756	224-8	155 #2	67-9034								
	MTP007	030156	224-8	165 #2	269-9077								
	MTP010	030256	224-8	172 #2	271-9117								
	MTP011	030364	226-8	183 #	273-9153								
	MTP012	031162	220-8	207 #3	277-0374								
1	MTP013 MTP014	031550 032264	227-8	221 #3	282-9467								
	MTP015	032510	229-8	232 #2	283-9515								
	MTP016	022327	229-8	241 #2	285-9614								
	MTP017	034036	229-8	246 #3	289-9716								
	MTP020	034114	231-8	260 #	291-9750	#204 DOOD							
	MTP022	034600	233-8	377 4	303-1004	#296-9888							
	MTP020 MTP022 MTP025 MTP030	036122	240-8	507	269-9077 271-9117 273-9153 275-9294 277-9376 282-9467 283-9515 285-9614 289-9716 291-9750 233-8323 303-10044 240-8511 309-10190 244-8628 313-10261 238-8455	#309-1018	4						
	MTP031	036132	242-8	553 #	309-10190	309-1021	2						
	MTP031 MTP032	036210	244-8	609	244-8628	#309-1018 309-1021 #311-1024 313-1030 238-8461							
	MTP033	036242	246-8	647 #	313-10261	313-1030	270 0101	370 0/00	2/4 9/79	2/4-9/9/	2/4-9400	#31/-10707	
	MTP034	034036 034114 034600 035346 036122 036132 036210 036242 036340	193-9: 1974	458	238-8455	258-8461	238-8484	238-8490	246-8678	246-8684	246-8690	#314-10307	
	MTDATE		3/4-1	711 #	315-10319								
	MTP035	036364	240-0	711	313-10319								

1							1				SEQ O	416
1	CZMSPA	CREATED BY	MACRO ON 2	26-APR-82 AT	09:54	PAGE 16					354 0	710
1	SYMBOL	CREATED BY				CREF						
	MTP036	VALUE 036526 036752 037024 037176 037432 037626 040142	REFERENCES 247-8727 248-8738	#316-10351 #317-10404								
	MTP037	036752	248-8738 249-8750	#317-10404 #319-10425								
	MTP041	037024	250-8761	#321-10466								
	MTP043	037432	250-8761 251-8770 251-8778	#322-10519 #323-10560								
	MTP045	040142	251-8786 251-8794	#324-10640								
	MTP046	040330	251-8794	#325-10680 #326-10756								
1	MTST3	012422	251-8803 169-6745	169-6746	169-6748	169-6774	169-6808	#169-6846	169-6847			
	MT0000	020700	#220-8012 217-7938	390-12381 219-7987	#220-8025	390-12382						
	MT0002	021100	217-7939	390-12381 219-7987 219-7988 #222-8075	#220-8047	Z00-12Z8Z						
	MT0003	040670 012422 020700 020760 021100 021240 021472 021614 021750 022004 022046 022102 022160 022350 022440 022516 022574 022576 023160 023212 023256	#220-8012 217-7938 217-7939 219-7989 217-7940		#224-8158 #224-8151 #224-8151 #224-8158	390-12385						
	MT0005	021614	21/-/941	219-7991	#222-8129	390-12385 390-12386 390-12387 390-12388						
1	MT0006 MT0007	021750	215-7867 217-7937	219-7986	#224-8158	390-12388						
	MT0010	022046	215-7884	219-7991 219-7984 219-7986 #224-8168 #226-8177	390-1238	9						
1	MT0012	022102	215-7891	#220-8180	390-1238 390-1239 390-1239	1						
1	MT0013	022264	215-7834 215-7890 215-7891 215-7892 215-7873 215-7893 215-7894 217-7936	#226-8201 #227-8212	390-1239	3						
	MT0013	022440	215-7893	#229-8226 #229-8235	390-1239	4						
	MT0016	022516	215-7894	219-7985	#229-R244	390-12396						
	MT0020	022616	212-1010	#231-8251 219-7992 219-7994	390-1239 390-1239 #229-8244 390-1239 #232-8264 #233-8311 390-1240 #234-8329 390-1240	7 700 12700						
1	MT0021	022706 023160	217-7942 217-7943	219-7992	#232-8204	390-12398 390-12399	397-12535					
1	MT0023	023212	219-7995	#233-8319 219-7997	390-1240	700-13/01						
1	MT0024 MT0025	023236	217-7945 215-7889	#234-8371	390-1240	390-12401						
	MT0026	023522 023600	215-7889 217-7944	#234-8371 219-7996	#236-8382	390-12403						
	MT0027	024566	187-7274	#240-8500	375-1200	3 377-12044	390-12405 393-12465					
1	MT0031	025070	217-7946	219-7998	#242-8545	377-12044 390-12406 390-12407 390-12408 219-7969	393-12465					
	MT0032	025612	217-7948	219-8000	#246-8638	390-12408				700 40400		
	MT0034	026000	217-7917	217-7935	217-7949	219-7969	219-7983	219-8001	#246-8668	390-12409		
	MT0036	026264	215-7875	#247-8719	390-1240 375-1200 #242-8545 #244-8574 #246-8638 217-7949 390-1241 390-1241	1						
1	MT0037	026336	215-7877	#248-8731 390-12413	390-1241							
	MT0041	026406	215-7878	#249-8746	390-1241	4						
	SYMBOL SYMBOL MTPO36 MTPO36 MTPO41 MTPO43 MTPO43 MTPO45 MTPO46 MTPO46 MTPO47 MTO000 MTO003 MTO004 MTO001 MTO011 MTO012 MTO013 MTO014 MTO015 MTO021 MTO021 MTO021 MTO023 MTO024 MTO023 MTO034 MTO037 MTO037 MTO037 MTO037 MTO037 MTO047 MTO047 MTO047 MTO047 MTO047 MTO047 MTO047 MTO047 MTO047 MTO047 MTO047 MTO047 MTO047 MTO047 MTO047 MTO047 MTO047 MTO047 MTO047	024102 024566 025070 025260 025612 026000 026152 026264 026336 026404 026406 026450 026514 026554 026614 026654	187-7264 187-7274 217-7946 217-7947 217-7948 217-7917 219-7993 215-7877 #248-8743 215-7878 215-7878 215-7880 215-7881 215-7881 215-7881 215-7882 149-5926 217-7953	#251-8765	390-1241	6						
	MT0044	026554	215-7872	#251-8773	390-1241	7						
	MT0045	026654	215-7881	#251-8789	390-1241	9						
	MT0047	026714	215-7882	#251-8797	390-1242	215-7900	215-7901	215-7902	217-7917	217-7951	217-7952	
	HIUYYY	020700	217-7953	219-7996 #238-8429 #240-8500 219-7998 219-7999 219-8000 217-7935 #246-8698 #247-8719 #248-8731 390-12413 #250-8755 #251-8765 #251-8765 #251-8773 #251-8789 #251-8789 #251-8797 215-7898 217-7954	390-1241 390-1241 390-1241 390-1241 390-1241 390-1242 215-7899 217-7955	215-7900 217-7956	215-7901 217-7957	215-7902 217-7958	217-7917 217-7959	217 - 7951 219 - 7969	217-7952 219-8003	

										SEQ 0	141
CZMSPA	CREATED BY CROSS REFERENCE		26-APR-82 AT		PAGE 17 CREF						
SYMBOL	VALUE	REFERENCES	219-8005	219-8006	219-8007	#253-8808					
MT1 MT2 MUT	017026 017032 002106	209-7660 209-7668 #139-5586 *397-12529	#209-7670 *187-7263	*187-7266	*191-7344	*191-7346	*209-7657	*209-7670	*390-12353	*393-12459	
NC	056430	434-13024	4 434-13060	476-15156 434-13634	*476-15158 #434-13638 *169-6760	476-15172 169-6784	169-6796	169-6818	169-6842	*339-11092	
NEMCNT	002066	#139-5578 339-11093	*167-6703 *339-11097	167-6716 509-16389 *363-11677			*365-11737		369-11845	-557 11072	
NEWBAN NEWKER NEWLOA	002304 046640 046742	#139-5649 363-11695 363-11624	240-8533 5 365-11743 4 365-11722	#369-11842 #369-11865	303-11000	303-11070	-505 11151	507 11052	307 11012		
NOCH	063560 002424	483-15441 #139-5691	1 #483-15445 *382-12174		*384-12199	465-14823	465-14860	466-14883	*466-14912	468-14949	
NOFSMO	002422	468-14961 #139-5690	*238-8441 *167-6704	*238-8468	*476-15175 *238-8496 *169-6827	*240-8513 *169-6836	*240-8530 339-11090	*240-8541	380-12101		
NONEM NONEXI NOOJ	002076 042512 041464	#139-5582 167-6705 331-1088	169-6742	*169-6741 #339-11090		*109~0030	337-11070				
NOPAR	002074	#139-5581 *217-7913	*144-5804 *217-7928	*144-5810 *219-7963	*157-6086 *219-7975	*164-6428 *226-8208 *315-10348	*167-6702 *240-8504	*169-6740 *242-8550	*169-6782 *244-8579	*211-7725 *246-8643	
		*282-9475 386-1220 *393-1247	* 303-10049	*303-10084	388-12251	*315-10348 *388-12252	*325-10696 *388-12297	337-11063 *390-12373	337-11067 *390-12378	337-11074 393-12440	
NORES NOSCOP	003664 002434	144-5/90	#144-5/92	*397-12544 *232-8308	*234-8332	*234-8369	*240-8513	*240-8530	*240-8541	*242-8548	
NOSUPE	002452	#139-5695 *242-8570 #139-5702	*244-8577 *145-5834	*232-8308 *244-8632 *145-5853	*246-8641 155-6034	*246-8665 220-8029 361-11560	*240-8513 462-14758 220-8053 361-11581	466-14888 242-8561 425-13309	244-8588	246-8655	
		#242-8370 #139-5702 257-8878 428-1339	*145-5834 355-11376 9 429-13426 335-10970	361-11518	432-13507	478-15219					
NOTAB	002366	*473-1507	5		335-11023	*472-15034	*4/2-1505/	*4/2-13048	*472-15050	*4/3-130/2	
NOTRCE NO22BI	060220 002450	462-1473 #139-5701 377-1205	7 #462-14739 *145-5837	*145-5839	154-5992	166-6486	189-7318	355-11395	361-11574	375-12005	
NULLFL	002340 005642	#139-5701 377-1205 #139-5663 157-6087 #139-5646 #139-5605 #141-5733 327-1081	*253-8810 #157-6105								
OLDCAC	002276 002154	#139-5646 #139-5605	*172-6898	172-6916 174-6968	*172-6917 174-6970	213-7822	213-7824	222-8114	296-9897	296-9899	
ONES PADDRE	002600	327-1081 #139-5566	9 327-10824	453-14540			213-7024	222-0114	270-7071	270 7077	
PAFBAF PAFBAW	016146 016276 016450	187-7252 187-7253	#201-7489 #203-7527	201-7514 203-7554 205-7597 207-7637 167-6713	201-7522 203-7559 205-7605 207-7642 *169-6759 *325-10704	203-7567					
PARBAF PARBAW	016600	187-7254 187-7255	#205-7572 #207-7610	205-7597	205-7605	207-7650 169-6794	169-6816	140_4878	*282-9474	282-9483	
PARCNT	002070	*282-9498 147-5882	282-9501 #337-11063	*282-9511		325-10744	325-10749				
PARTHE	002300	#139-5566 187-7252 187-7253 187-7254 187-7255 #139-5579 *282-9498 147-5882 #139-5647 *386-1222 #111-4755	*303-10051 8 *386-12245	*305-10113 388-12251 *147-5883 *160-6226	*305-10125 *388-12267	*305-10128 *388-12297	*315-10325	*315-10340	337-11076	386-12209	
PARVEC PASFLG	= 000114 002262	#111-4755 #139-5639	*172-6898 169-6802 9 327-10824 509-16388 #201-7489 #203-7527 #205-7572 #207-7610 *167-6701 282-9501 #337-11063 *303-10051 *386-12245 *147-5882 *160-6223	*147-5883 *160-6226	160-6228	160-6235	160-6246	160-6253	160-6260	*209-7658	
The second secon											

										SEQ O	418
CZMSPA	CREATED BY		6-APR-82 AT	09:54 F	PAGE 18						
SYMBOL	VALUE	REFERENCES *211-7728 *273-9273 *283-9585 291-9764 *319-10439 *323-10573	*240-8501 277-9388 *283-9587 291-9800	240-8524 279-9423 285-9627 291-9808 *321-10478	*240-8525 279-9435 287-9673	*247-8723 *279-9437 287-9685 *316-10360	273-9185 *279-9440 *287-9687 316-10363 *322-10527	273-9257 283-9526 *287-9689 316-10387	273-9269 283-9571 *291-9758 316-10395 322-10556 325-10736 *324-10648	*273-927 283-958 *291-9761 *319-10435	
PASSNO	002264	*319-10439 *323-10573 #139-5641 324-10675	319-10460 323-10615 *282-9494 *325-10686	323-10634 *282-9497 *325-10689	*316-10357 *321-10485 *325-10691 282-9508 325-10713	*325-10576 *323-10576 325-10737	*322-10527 325-10698 *323-10578 325-10743	*322-10531 325-10728 323-10632	322-10556 325-10736 *324-10648	*291-9761 *319-10435 *323-10568 *390-12358 *324-10652	
PATERR PATPLU PATTER	002072 004606 002110	*323-10573 #139-5641 324-10675 #139-5580 149-5902 #139-5587 *207-7630 390-12323 #139-5656 337-11071 184-7203 *331-10870	*169-6758 149-5905 *191-7340 217-7915	169-6792 149-5909 *193-7363 219-7967	169-6814 149-5912 *195-7397 *373-11977	*169-6852 149-5916 *197-7425 373-11978	149-5919 *199-7466 *373-11983	#149-5922 *201-7490 390-12303 *397-12544	*203-7528 *390-12321	*205-7590 390-12322	
PCBUMP	002322	#139-5656 337-11071	217-7915 390-12375 *217-7914 390-12303	*390-12427 *219-7964 *390-12374	393-12440 *222-8078 *390-12426	*393-12474 *222-8113	397-12511 *222-8131	*224-8153	*224-8170	*236-8384	
PCONFS PCONF1 PCONF2 PDP110	041352 041652 041562 041620 042602	*331-10870 331-10883 331-10871 147-5884 425-13310	390-12303 #331-10868 331-10911 #331-10902 331-10890 #339-11110	331-10900	#331-10911						
PD1 PERA05 PERBNK	054476 056644 057476	#/ 51-1/ 503	315-10330		421-13247	454-14584	454-14596	456-14611	#458-14632	460-14705	
PERECC PERRAB PERRAW PERRA3	057556 057314 057242 054044	211-7747 465-14869 #458-14648 451-14525 421-13224 453-14541 #421-13226	458-14662 451-14530 423-13265 453-14546 456-14617 #456-14608	#454-14595 451-14482 453-14551 456-14621	451-14488 453-14556	451-14494 453-14561	451-14500 453-14571	451-14515 453-14576	451-14520 453-14581	453-14536 #454-14583	
PERRA7 PERRO1 PERRO2 PERRO3	057366 = 104427 = 104430 = 104431 = 104432 056640 056666	#110-4620	259-8921	311-10256 259-8934 263-8991 265-9029	311-10250	314-10314					
PERRO4 PERRO5 PERRO6 PERRO7 PERR10 PERR11 PERR12 PERR13 PERR14 PERR15	= 104433 = 104434 = 104435 = 104436 = 104437 = 104440	#110-4621 #110-4622 #110-4623 #451-14509 #451-14509 #110-4624 #110-4625 #110-4627 #110-4630 #110-4631 #110-4633 #110-4633 #110-4634 #110-4635 #110-4637 #110-4637 #110-4638	267-9044 267-9049 269-9085 269-9091 269-9099 269-9105	267-9063 267-9068 271-9135 271-9142							
PERR16 PERR17 PERR20	= 104441 = 104442 = 104443 = 104444 = 104445	#110-4631 #110-4632 #110-4633	292-9823 292-9829	292-9845 292-9839	292-9856 292-9862	294-9880 294-9874	297-9935 297-9944	297-9939 297-9948			
PERR21 PERR22 PERR23 PERR24	= 104446 = 104447 = 104450	#110-4635 #110-4636 #110-4637	289-9738 299-10011 299-10016 307-10151	301-10040							
PERR25 PERR26	= 104451 = 104452	#110-4638 #110-4639	307-10151	307-10156							

CZMSPA	CREATED BY		-APR-82 AT	09:54 P	AGE 19					SEQ 041
SYMBOL SYMBOL PERR27 PERR30	VALUE = 104453	REFERENCES #110-4640 #110-4641			NET					
PERR30 PERR31 PERR32 PERR33	= 104454 = 104455 = 104456	#110-4642	309-10197 273-9214 273-9235	273-9220 273-9248	283-9561	283-9567	285-9659	285-9666	319-10456	
PERR35 PERR36 PERR37 PERR40	= 104457 = 104460 = 104461 = 104462 = 104463 = 104464	#110-4644 #110-4645 #110-4646 #110-4647 #110-4648 #110-4649	309-10197 273-9214 273-9235 275-9336 273-9241 313-10273 355-11421	273-9254 313-10280	275-9344					
PERR41 PERR42 PERR43 PERXOR PFECDF PFECDH	= 104465 = 104466 = 104467 057452 061740 061700	#110-4650 #110-4651 #110-4652 454-14587	454-14599 #470-15012 #470-15009 #470-15011 #470-15008	456-14610	#456-14623	458-14654	460-14690			
PFECDT PFECWS PFECWS PFLAG PGMCSR	061730 061644 061634 002120 002526	470-15007 470-15007 468-14940 #139-5591 #139-5710	#470-15011 #470-15008 #470-15007 172-6918 *162-6280 343-11166 365-11744 *182-7118 273-9228 *473-15088	363-11619 *162-6326 343-11170	*371-11907 162-6331 344-11190	*371-11937 *163-6374 344-11192	*163-6404 344-11196	*163-6405 363-11678	*163-6417 *363-11678	246-8706 *363-11704
PHEBE	014004	*363-11711 #182-7116	365-11744 *182-7118	*365-11744 *182-7123	182-7126	182-7128	182-7130	182-7132	*182-7134	*182-7136
PHYADD PMEMFL	002036 002140	273-9171 #139-5567 #139-5599 247-8720	273-9228 *473-15088 226-8181 248-8732 *371-11934	*473-15089 226-8190 249-8747	*473-15090 226-8205 250-8756	*473-15091 227-8216 251-8766	*473-15093 229-8230 251-8774	473-15094 229-8236 251-8782	231-8256 251-8790	234-8375 251-8798
PROTYP	003752	*371-11905 #145-5815 169-6781 232-8276 238-8488 363-11683	*145-5850 169-6806 234-8341 240-8505 365-11747 462-14771	145-5851 220-8017 234-8343 244-8606 382-12176	145-5860 220-8039 236-8391 244-8626 382-12180 476-15177	151-5956 220-8065 236-8393 246-8676 384-12192	164-6446 222-8085 238-8434 246-8682 384-12204	167-6721 222-8118 238-8453 246-8688 386-12246	169-6743 222-8136 238-8459 328-10840 388-12298	169-6772 232-8274 238-8482 344-11199 426-13359
PSIZE PSW	= 002400 = 177776	+213-7794	462-14771 *184-7151 *155-6038 *242-8552 *344-11217 *390-12349 *426-13332	*184-7159 *155-6039 *244-8600	476-15177 184-7196 *155-6045 *246-8646 *365-11739 *397-12526 *429-13433 325-10707 *153-5975 *430-13481	+169-6771	186-7237 *169-6788 *257-8877 *367-11814 *421-13231	*169-6805	*169-6831 *333-10933	*174-6951 *335-10951 *388-12271 *423-13273 *478-15237
PTABLE PWRVEC	036732 = 000024	291-9772 #111-4748	316-10372 *147-5880 430-13478	#316-10402 *147-5881 *430-13480	325-10707 *153-5975 *430-13481	*425-13300 *432-13518	*425-13301 *432-13519		*428-13378	
QUICK QVFLAG	002432 002342	430-13477 #139-5694 #139-5664 283-9578 *236-8402 #110-4600	*153-5974 *153-5974 287-9680	377-12043 *153-5979	189-7294	*189-7296	246-8713	273-9264	275-9361	279-9430
RANODD RDCHR	036044 = 104411	*236-8402 #110-4600	*236-8419 486-15601	#307-10153	*307-10157					
RDDEC RDL IN RDOCT	= 104414 = 104412 = 104413	#110-4603 #110-4601 #110-4602	*236-8419 486-15601 380-12113 406-12652 384-12194	487-15682 386-12213	487-15733 388-12255	388-12287	390-12307	390-12320	390-12325	399-12554

3

									SEQ 0420
CZMSPA SYMBOL SYMBOL	CREATED BY CROSS REFERENCE VALUE	REFERENCES	AT 09:54	PAGE 20 CREF					
READCS	= 104426	400-12604 #110-4618 166-65 315-10333 319-10 324-10664 325-10 384-12188 384-12	447 321-10491	282-9486 321-10500 326-10786 426-13342	291-9780 322-10537 348-11266 478-15198	291-9788 322-10546 348-11285	303-10061 323-10590 350-11301	305-10118 323-10603 350-11320	305-10130 324-10657 382-12173
REALPA	002404 002274	#139-5684 #139-5645 *220-80 *224-8169 *226-81 *232-8266 *233-83 *244-8578 *246-86 *251-8775 *251-87	*220-8026 82 *226-8191 14 *233-8322 42 *246-8669	*220-8048 *226-8206 *234-8333 *246-8699 *251-8799	*222-8077 *227-8217 *234-8376 *247-8721 *253-8809 485-15540	*222-8112 *229-8231 *236-8383 *248-8733 363-11660	*222-8130 *229-8240 *238-8432 *249-8748 363-11667	*224-8152 *229-8245 *240-8503 *250-8757 458-14658	*224-8159 *231-8255 *242-8549 *251-8767 458-14663
REFRES REFSUB REGCOP RELENT RELOCA	035154 035224 041064 045612 045172	458-14666 458-14 296-9923 #297-99 297-9963 297-99 220-8016 220-80 363-11661 363-11 193-7377 195-74	960 967 #297-9971 938 #327-10802 9662 #363-11674		201-7519	203-7564	205-7602	207-7647	240-8527
RELOC1 RESREG	045626 = 104416	#363-11677 #110-4606 222-80 367-11818 380-12	990 222-8104 2103 382-12166	234-8365 470-15000	236-8405 491-15866	236-8422	328-10845	328-10851	367-11781
RESTAR RESVEC RESO RES1 RES2	= 002612 = 000010 050376 050456 050624	*137-5538 *137-5 #111-4743 *147-58 382-12158 382-1 382-12172 #382-1 384-12187 #384-1 #139-5593 193-7	140 #141-5738 184 *147-5885	151-5937					
REFLAG	002124	#139-5593 193-73 *363-11712 *365-1 #139-5592 191-7 207-7622 211-7 390-12333 393-1	1746 371-11938 343 193-7361	195-7391 238-8474	199-7478 377-12042 197-7423 240-8517	201-7516 380-12101 199-7460 *371-11907	203-7561 425-13315 201-7499 *371-11937	205-7599 463-14795 203-7539 *371-11940	207-7644 485-15534 205-7582 *371-11948
RWCSR SAVCSR SAVMON SAVPAR SAVREG	006206 002152 002270 002266 = 104415	157-6100 #160-6	2338 390-12430 293 465-14879)	369-11879 234-8354	236-8398	236-8411	328-10837	367-11779
SBEMSK	002250	#139-5604 *390-12 #139-5643 *144-5 #139-5642 *369-1 #110-4605 222-8 367-11782 380-12 #139-5636 *273-9 *275-9301 275-9 277-9393 277-9 *279-9453 279-9 *283-9579 *283-9 *285-9622 285-9 *285-9622 285-9 *287-9695 *287-9 213-7799 213-7 213-7842 #213-7 213-7842 #213-7 #111-4683 157-6 #111-4852 *222-8 #111-4853 *236-8	1859 369-11877 1867 222-8095 1808 468-14930 180 *273-9181 306 275-9308 395 277-9399 454 279-9456 581 *283-9581 531 285-9633 702 287-9703 851 213-7806	234-8348 491-15851 273-9196 275-9319 277-9401 *283-9520 *283-9590 285-9637 287-9705 213-7808	273-9198 275-9356 279-9431 *283-9521 283-9593 285-9639	273-9259 *275-9358 *279-9433 283-9531 *283-9600 287-9681	*273-9261 *275-9358 *279-9433 283-9533 283-9601 *287-9683	*273-9261 *277-9382 *279-9443 283-9537 283-9603 *287-9683	*275-9300 *277-9383 279-9446 283-9539 *285-9621 *287-9692
SBENT	020146	287-9695 *287-9 213-7799 213-7 213-7842 #213-7	702 287-9703 801 213-7806 851	287-9705 213-7808	213-7817	213-7823	213-7825	213-7831	213-7833
SBESYN SBETES SCOPE SDPARO SDPARS SDPAR6	017670 = 000004 = 172260 = 172272	291-9790 #291-9 211-7722 #213-7 #111-4683 157-6 #111-4847 222-8 #111-4852 *222-8 #111-4853 *236-8	815 766 063 167-6694 099 222-8101 103 *234-8360 417	169-6731 234-8356	172-6879 234-8358	187-7242 234-8359	187-7259 236-8415	255-8850	257-8887

1

1											SEQ 0421
	CZMSPA SYMBOL	CREATED BY CROSS REFERENCE		6-APR-82 AT	09:54	PAGE 21 CREF					
	SYMBOL SDPAR7 SEEDHI SEEDLO SELONL SETPAT	VALUE = 172276 002566 002570 002000 047500	REFERENCES #111-4854 #141-5728 #141-5729 #139-5549 195-7393	*222-8102 *147-5869 *147-5870 187-7262 199-7462	236-8386 236-8385 211-7679 #373-11981	*236-8425 *236-8424 371-11947	489-15824 489-15823 *400-12615		*489-15836 *489-15835		
	SHADL1 SHUTUP SIPARO SIPAR3 SIPAR5	012452 047664 = 172240 = 172246 = 172252	#170-6861 189-7305 #111-4837 #111-4840 #111-4842 361-11585	377-12023 361-11516 273-9173	220-8055	355-11378 242-8563	361-11534 244-8590	246-8657	273-9174	*273-9175	361-11564
	SIPAR6 SIPDRO SIZE	= 172254 = 172200 = 040000	#111-4843 #111-4817 #113-4976 236-8390 355-11740	361-11517 167-6707	220-8055 425-13319 169-6767 238-8479 367-11815	242-8563 425 13419 169-6800 240-8518	244-8590 220-8015 244-8583	246-8657 220-8028 244-8620	*361-11585 220-8050 246-8671	222-8117 328-10839	222-8135 363-11681
	SKIPKA SKIPMK SKJ SKPERR SKUB	002006 002336 060246 002064 045602	#139-5552 #139-5662 462-14741 #139-5577	462-14744 *213-7812	207-1003	*371-11908	*399-12569 *371-11958 *355-11423				
	SKUJ SOBK SOBLEN SOFTPA SOURCE	014006 002556 = 000056 002604 002306	363-11668 182-7114 #141-5724 242-8553 #141-5735 #139-5650	#182-7117 242-8555 242-8558 246-8672 *273-9191	#309-10212 *275-9304		*283-9535	*285-9 <u>635</u>	*303- <u>1004</u> 7	357-11445	
	SPLTCS SSP ST	002236 = x 000006 = 177776	#139-5633 361-11542 #111-4701 #498-16094	*172-6908 2 361-11547 *155-6042	*173-6930 *390-12357 *255-8845	*277-9397 *211-7685 *390-12362 *257-8882	*211-7695 426-13333 141-5725	211-7717	*211-7750	*211-7753	*211-7760
	STACK START START1 START2 START3	= 002000 003654 000300 000310 000200	#111-4678 137-5539 137-5535 137-5536 #137-5535	111-4679 137-5541 #137-5538 #137-5540 517-16636 #238-8442 *174-7027	#144-5788	377-12028	141-5/25	133 3710	107 1301		
	STAR27 STOPOK STRIPE SUBAAA SUBAAB	024162 002414 002362 004644 004774	#139-5687 #139-5675 149-5920 #153-5963	#151-5933	462-14754 296-9906	*462-14755 296-9925	*486-15610 *297-9956	297-9956			
The second secon	SUBAAI SUBAAR SUBAAS SUCCES SUPDOA	012446 014166 013372 011400 002330 002260	137-5535 137-5536 #137-5535 238-8437 #139-5687 #139-5675 149-5920 #153-5963 169-6783 182-7137 173-6945 166-6508 #139-5659 #139-5659 #139-5638 *224-8165 *232-8284 *236-8404 *244-8628	#170-6856 #184-7150 #174-7027 #167-6692 *211-7711 *220-8019 *224-8172 *232-8287 *238-8439 *246-8651 220-8045	*211-7737 *220-8041 *226-8183	211-7743 *220-8067 *226-8198 *233-8315 *238-8461	*395-12489 *222-8088 *226-8207 *233-8323 *238-8484 *246-8690	395-12494 *222-8091 *229-8232	*395-12496 *222-8120 *229-8241	*222-8138 *229-8246	*224-8155 *232-8282
	SUPD01	027030	*232-8284 *236-8404 *244-8628 220-8023 246-8693	*232-8287 *238-8439 *246-8651 220-8045 #255-8821	*220-8041 *226-8183 *232-8289 *238-8455 *246-8679 220-8071 328-1085	*240-8084	*233-8323 *238-8484 *246-8690 232-8297	395-12494 *222-8091 *229-8232 *234-8349 *238-8490 *246-8711 236-8418	*395-12496 *222-8120 *229-8241 *234-8358 *240-8511 *249-8750 236-8423	*222-8138 *229-8246 *234-8377 *242-8559 257-8884 238-8440	*224-8155 *232-8282 *236-8400 *244-8615 *328-10843 240-8508

i

						1				SEQ 0422
CZMSPA SYMBOL	CREATED BY CROSS REFERENCE		S-APR-82 AT	09:54	PAGE 22 CREF					354 0422
SYMBOL SYMBOL SUPDO2 SUPDO3	VALUE 027044 027206	222-8105 220-8020 226-8209 236-8406	222-8127 220-8042 229-8233 238-8436 328-10844 222-8121 244-8629 *255-8825 255-8826	222-8147 220-8068 229-8242 238-8462	232-8300 222-8089 229-8247 238-8491	232-8304 224-8156 232-8283 240-8510	232-8307 224-8166 233-8316 246-8685	244-8631 224-8173 233-8325 246-8691	#255-8823 226-8184 234-8378 246-8712	226-8199 236-8401 249-8752
SUPD04	027222	#257-8855 222-8092 244-8619	222-8121 244-8629	222-8140	232-8285 246-8716	232-8288	232-8290	234-8362	234-8368	242-8569
SUPDRO SUPDR1 SUPDR2 SUPDR3 SUPDR4 SUPDR5 SUPDR6	002156 002160 002162 002164 002166 002170	REFERENCES 222-8105 220-8020 226-8209 236-8406 #257-8855 222-8092 244-8619 #139-5607 #139-5608 #139-5610 #139-5611 #139-5613 #434-13656 #111-4680	*255-8825 255-8826	222-8140 246-8664 255-8833 257-8859	*257-8858	#257-8856 257-8866				
SUPLIM	002172 056506 = 000740	#139-5613 #434-13656 #111-4680	255-8836 517-16631 155-6041 *147-5872	257-8869 517-16632 255-8845	257-8882	478-15226	478-15232			
SWAPAT	002620 002622	#111-4680 #141-5743 #141-5746 187-7244 331-10883 428-13385 470-15001	*151-5950 189-7294 363-11612 462-14754 483-15465	151-5952 217-7908 363-11631 462-14761 483-15473	*151-5957 217-7923 363-11666 462-14779 483-15495	*153-5976 273-9263 390-12366 465-14830 486-15608	155-6052 275-9360 393-12444 465-14870	167-6724 279-9429 397-12515 465-14873	173-6939 283-9577 421-13250 466-14884	184-7202 287-9679 426-13363 466-14888
SWREG SWO	= 000176 = 000001 = 000003	#113-4968 #111-4721 #111-4720	151-5957 167-6724	155-6052 173-6939	483-15465 217-7908	217-7923	390-12366	393-12444	397-12515	421-13250
SW1 SW10 SW11 SW12 SW13 SW14 SW15	= 000002 = 002000 = 004000 = 010000 = 020000 = 040000	#111-4711 #111-4710 #111-4709 #111-4708 #111-4706	465-14830 189-7294 363-11612 363-11631 462-14761	273-9263 363-11666	275 - 9360 465 - 14870	279-9429	283-9577	287-9679		
SW2 SW3 SW4 SW5 SW6 SW7 SW8 SW9 SYSSIZ TAG2\$ TAG2\$ TAG3\$ TAG70\$ TAG70\$	= 000004 = 000010 = 000020 = 000100 = 000200 = 000400 = 001000 003754 012042 012076 027124 061744	#111-4719 #111-4718 #111-4717 #111-4716 #111-4715 #111-4714 #111-4713 #111-4712 145-5814 169-6753 169-6779 255-8834 470-14977 470-14978 470-14980	331-10883 465-14873 184-7202 470-15001 462-14779 #145-5816 #169-6780 #169-6784 #255-8836 #472-15025 #472-15031 #472-15045 #472-15063 #473-15069	466-14888						
TAG72S TAG73S TAG74S TAG76S TAG77S TAG78S	061764 062034 062074 062106 062120 062164	470-14979 470-14980 470-14981 470-14982 470-14983 470-14984 470-14985	#472-15031 #472-15045 #472-15057 #472-15063 #473-15069 #473-15082 #473-15088							

CZMCDA	CDEATED DV	MACDO ON 36	-APR-82 AT	00.54	PAGE 23					35.4	0423
CZMSPA SYMBOL	CREATED BY CROSS REFERENCE		-AFR-02 AT	07.34	CREF						
SYMBOL TAG79\$ TAG9\$ TBG4\$	VALUE 062252 011670 027302	169-6747	#473-15103 #169-6751 #257-8869	169-6786	169-6823	169-6845					
TCFIG1 TCFIG2 TCFIG3 TCONFI	027302 041726 042066 042222 041654	#335-11011 331-10888	#257-8869 335-10972 335-11002 335-11025 331-10899	472-15036 473-15074 472-15049 331-10909	#333-10933	.104 7200	104 7224	±194_7227	*395-12482	+305-12493	
TEMP	002430	395-12484	*166-6538	166-6594	166-6596	*186-7209	186-7226	*186-7227			
TESTAD	002406	213-7793 275-9310 285-9645	162-6282 *211-7715 *219-7965 275-9351 285-9654	*162-6283 211-7716 *219-7966 275-9368 287-9709	*162-6284 *211-7716 224-8154 277-9379 303-10050	166-6482 *211-7719 224-8171 279-9460 390-12303	*166-6512 *211-7721 273-9200 283-9545 *390-12359	*166-6513 *211-7740 273-9201 283-9552 *390-12360	*166-6658 211-7740 273-9287 283-9556 *390-12364	*166-6669 213-7792 273-9288 283-9607 *390-12426	
TESTMO	002546	#141-5720 242-8552 421-13215	*145-5833 244-8600 421-13231	*145-5854 246-8646 421-13241	169-6771 257-8877 423-13261	169-6788 344-11215 423-13273	169-6805 377-12060	169-6831 386-12231	174-6951 388-12271	213-7794 388-12289	,
TIMEOU	002334 042556	#139-5661 145-5860	147-5886	151-5956	164-6446	167-6721	169-6781	#339-11103	382-12176	382-12180)
TKVEC	= 000060	384-12192 #111-4751 *390-12346 *393-12474	384-12204 331-10869 *390-12347 397-12511	*390-12424 397-12511	*39/-12023	*331-10872 393-12440 *397-12524	393-12440 *397-12544	*331-10914 *393-12452 *397-12544	390-12303 *393-12453	390-12303 *393-12474	
TMFLAG TOOMAN	002132 002402	#139-5683	*201-7510 *458-14644	*203-7550 465-14873	*466-14912	*207-7633	373-11967	7/0 440/0	350 44305	75/ 1175	
TOTCSR	002222	#139-5626 406-12645	*157-6116 426-13337	162-6287 428-13404	163-6345 478-15193	163-6377	166-6483	348-11260	350-11295		
TRACE TRAPVE	006204 = 000034	#159-6179 #111-4750	*164-6427 *147-5878	*164-6433 *147-5879	164-6435	*164-6441	*164-6445	*402-12634	*404-12640	462-14736	•
TSTBAN TSTDAT	012310 002244	*273-9199 275-0304	#169-6826 *273-9183 273-9203 *275-9307	*273-9184 273-9205 *275-9309	*273-9187 273-9211 275-9313	*273-9188 273-9217 275-9316	273-9189 *273-9276 *277-9386	273-9190 *273-9277 *277-9387	273-9191 *275-9302 *277-9390	*273-9197 *275-9303 *277-9391	
		277-9397 *283-9524 283-9547	*277-9400 *283-9525 283-9550	*277-9402 *283-9528 283-9558	*277-9404 *283-9529 283-9564	*277-9406 283-9535 *283-9588 *285-9642 *303-10046	*283-9538 *283-9589	*283-9540 *285-9625	*279-9441 *283-9542 *285-9626 285-9649 *303-10065	*279-9442 *283-9544 *285-9629	
		277-9397 *283-9524 283-9547 *285-9630 285-9663 *303-10068 305-10109 344-11201	*277-9400 *283-9525 283-9550 285-9635 *287-9690 *303-10070 *316-10382 344-11203 213-7810	*277-9402 *283-9528 283-9558 *285-9638 *287-9691 *303-10073 *316-10383	*277-9404 *283-9529 283-9564 *285-9640 *303-10045 *303-10074 453-14545	*285-9642 *303-10046 *303-10076 453-14550	*303-100/8	277-9410 *283-9540 *285-9625 285-9647 *303-10063 *303-10079 458-14652	285-9649 *303-10065 *303-10081 509-16397	*279-9442 *283-9544 *285-9629 285-9656 *303-10067 305-10107 509-16397	7
TSTRD1 TSTREA TST1	043230 = 104510 005522	344-11201 #110-4670 #157-6063	344-11203 213-7810	#344-11217 213-7835	273-9237	273-9250	275-9326	275-9338			
TST1 TST2 TST3 TST4 TST5	011404 011570 012556 014742	#110-4670 #157-6063 #167-6694 #169-6731 #172-6879 #187-7242 #187-7259 #110-4594 472-15025 #110-4590									
TST6 TYPDS	015014	#187-7259 #110-4594	184-7190	184-7194	184-7198	184-7200	189-7298	395-12484	395-12486	395-12501	1
TYPEIT	= 104401	472-15025 #110-4590	154-5996	154-6010	154-6024	154-6029	154-6030	159-6142	159-6143	159-6152	

CZMSPA	CREATED BY	MACRO ON 26	-APR-82 AT	09:54 P	AGE 24					3EW 042
SYMBOL	CROSS REFERENCE VALUE	REFERENCES 159-6153 184-7195 331-10879 331-10906 335-10999 380-12102 386-12211 388-12286 393-12442 397-12520 400-12603 465-14833 470-15004 478-15192 478-15530	159-6156 184-7199 331-10887 331-10907 335-11010 380-12112 386-12212 388-12293 393-12449 397-12522 400-12614 466-14903 472-15035 478-15200 478-15243 483-15474	159-6172 184-7201 331-10891 331-10908 335-11022 380-12117 386-12237 390-12305 393-12451 397-12538 400-12619 468-14931 472-15039 478-15245 483-15483 486-15618	159-6173 189-7293 331-10892 333-10934 335-11032 382-12148 386-12241 390-12306 393-12468 399-12553 402-12633 468-14955 478-15213 478-15213 478-15249 483-15488 486-15624 487-15711	159-6177 189-7295 331-10893 333-10937 335-11040 382-12178 388-12253 390-12315 395-12485 399-12562 404-12639 468-14956 478-15215 478-15215 478-15251 483-15497 486-15629 487-15712	163-6416 211-7744 331-10894 333-10939 335-11042 384-12189 388-12254 390-12319 395-12487 399-12567 406-12651 468-14958 473-15097 478-15225 481-15352 481-15352 483-15515 486-15633 487-15713	182-7145 211-7746 331-10895 335-10949 363-11632 384-12193 388-12277 390-12324 395-12495 399-12573 429-13451 468-14965 473-15103 478-15228 482-15417 485-15524 486-15638 487-15768	184-7187 331-10877 331-10896 335-10969 363-11669 384-12198 388-12281 390-12334 395-12502 399-12580 434-13574 468-14967 473-15105 478-15230 483-15462 485-15535 486-15639 487-15770	184-7191 331-10878 331-10905 335-10980 380-12099 384-12202 388-12284 390-12337 397-12513 400-12599 465-14832 470-14994 478-15191 478-15234 483-15471 485-15537 486-15641 487-15771
TYPOC	= 104402	486-15644 487-15772 #110-4591 478-15246 #110-4592 485-15538	486-15648 494-15990 468-14936 483-15473	487-15709 472-15019		478-15214	478-15216	478-15229	478-15231	478-15244
TYPOS	= 104403	478-15246 #110-4592	483-15473 211-7745	386-12234	388-12285	388-12294	395-12498	472-15057	472-15063	473-15104
TYPS0	= 000000	13/-0111	124-0122	159-6176	160-6260_	282-9508	291-9808	316-10395	319-10460	321-10512
TYPS1 TYPS2 T12A T12B UDPARO	= 000000 = 000000 033254 033276 = 177660	322-10556 160-6259 325-10734 #285-9619 #285-9623 #111-4807	323-10634 291-9806 287-9688 287-9684 234-8361	324-10675 316-10393	325-10737 323-10632	325-10736				
UDPAR/ UIPARO	= 177676 = 177640	111-4814	*234-8359 #111-4797	160-6192	*160-6227 *236-8421	*160-6234	160-6259	*160-6272	361-11520	369-11823
UIPAR1 UIPAR2 UIPAR3 UIPAR4	= 177642 = 177644 = 177646 = 177650	#111-4798 #111-4799 #111-4800 #111-4801	*222-8100 169-6749 169-6750 367-11785	*236-8416 222-8102 238-8439 367-11802	*236-8421 236-8417 246-8679	*328-10849 355-11380	361-11537			
UIPAR5	= 177652	#111-4802 361-11583	220-8034	220-8058	*222-8125	*222-8145	242-8566	244-8593	246-8660	361-11562
UIPAR6 UIPDRO	= 177654 = 177600	#111-4803	220-8034 361-11521	220-8058 369-11825	222 -8 126 425 - 13318	222-8146	242-8566	244-8593	246-8660	*361-11583
UNI TOP UNMAP	002412 046774	#139-5686 227-8222 251-8804 193-7381 #365-11716	*184-7179 231-8261 #369-11876 195-7411	*184-7182	*184-7183 248-8739	222-8146 428-13418 *184-7184 250-8762	*184-7185 251-8771	184-7186 251-8779	184-7186 251-8787	251-8795
UNRELO	046056	193-7381	195-7411	197-7448	199-7485	201-7523	203-7568	205-7606	207-7651	240-8537
UPPFLG USERMA USESTK USP WARN1 WARN2 WARN3	002263 046556 = 000700 = \$000006 011756 027522 027536	#139-5640 363-11679 #111-4681 #111-4702 #169-6764 #261-8951 #261-8956	375-12002 *285-9650 365-11738 155-6047 *155-6048 *327-10828 *327-10829	*257-8880 *257-8880	*287-9671 #369-11822 478-15241 426-13327	478-15247 *429-13434	478-15238			
WARNS	021730	WE01-0730	361 10067							

CZMCDA	CREATER DV	MACRO	ON 2	6-APR-82 AT	00.54	PAGE 25					SEQ 04
CZMSPA SYMBOL SYMBOL	CREATED BY	DEFEN			V/	CREF					
JARN4 JARN5 JARN6	VALUE 027562 027576 041306 041246 041300	#261- #261- #328-	8968 8973 10849	*327-10830 *327-10831							
IARNA IARNA	064140	REFER #261- #328- #328- #238- #110-	10842 10841 8439	#328-10848							
ASSBE ASSBE ASSBE	= 104500 = 104476 = 104501	#110-	4660	174-6961	174-6974	174-6989	174-7004	277-9414			
HICHC OOPEN	053674 055704	382- 430-	12170	384-12185 430-13482	#406-12644 430-13484	432-13500	432-13523	#432-13531	432-13534		
DOPSA OOPUP	= 104501 = 104477 053674 055704 055734 055522	*430- 430-	4661 12170 13479 13315 13477 13479	*430-13478 430-13479	#406-12644 430-13484 430-13479 430-13480	432-13518 430-13482	432-13519 430-13482	432-13522 430-13482	#432-13534 #432-13499	432-13523	432-13524
ORST OCHAR	002564 056276	#141- #434-	13334 5727 13588 15442	*147-5864 *434-13622 *483-15443	*197-7436 *434-13625	*199-7473 *434-13626	*203-7556 434-13627	*207-7639 *434-13631	371-11944 *434-13632	434-13633	483-15440
XDPCH EROS	002350 002332	#139- #139-	5667 5660	*153-5983 145-5824 #493-15959	466-14902 296-9896	296-9900					
APTHD AUTO BANK	065740 002070 002011	#139- #139-	5575 5555	*153-5974 *463-14792	*155-59/9	155-6051	483-15469				
EROS APTHD AUTO BANK BASE BELL CACHF CACHN CBCSR CBTCS CDW1 CDW2	002350 002332 065740 002010 002011 065714 002637 042720 042674 043360 044350 043402 065720 065722	#139- #139- #139- #139- #141- #341- #341- #346- #355- #492-	-15931 -5753 -11151 -11144 -11247 -11427 -11252 -15934 -15935	496-16085		465-14832					
CHARCE CHKDI CHKID CHKID CHKID CHKID CHKID CHKID CHTAG	056464 043754 043770 063536 043732 043744 002000	#492- #434- #352- #483- #352- #352- #139-	13576 -11337 -11344 -15434 -15434 -1548 -5714 -1548 -15462 -1548 -15904 -5755 -15192 -15387 -15095 -15947	434-13586 496-16078 496-16079 496-16010 496-16076 496-16077 144-5795	*434-13644	#434-13649					
CNTLC CNTLG CNTLK CNTLU	064730 064742 064136 064735 065666 002644	483- 483- 483- 483-	15483 15471 15462 15488	486-15605 #486-15662 #483-15517 486-15633	#486-15660 #486-15661						
CRLF	002644	1141-	5755 15102	184-7187	331-10891 478-15228 #482-15425		795-12502 478-15251	434-13575 483-15497	468-14931 485-15524	468-14958 486-15638	468-14967
DBLK DB20 DDW0 DDW1	063526 065520 065724 065726	482- 473- 149- #493-	15387 15095 5898 15947	184-7187 478-15213 482-15417 4491-15851 #493-15946	#482-15425	110 12613	1.0 12.21				

```
PAGE 26
CREF
CZMSPA
                   CREATED BY MACRO ON 26-APR-82 AT 09:54
SYMBOL CROSS REFERENCE
             VALUE
065730
065732
065734
065736
042664
065650
065716
015346
015242
054726
043256
043256
043304
043272
043320
043332
043332
SYMBOL
SDDW2
SDDW3
SDDW4
$DDW5
SDEENE
SDEVCT
SDEVM
SDOAGA
$DOAGN
$DOWN
$DTBL
SECCDI
SECCIN
SECC1D
SECC11
SENASB
SENA1S
SENDAD
SENERG
SENV
                 065661
015066
SENVM
$EOP
             015066
002012
044100
060522
066262
061324
002614
002356
065660
065740
047656
= 000001
SERFLG
SERRGE
SERROR
SERRTB
SERRTY
                                                                                  SERTTL
SESCAP
SETABL
SETEND
SEXHAL
SES
                                          *465-14868 #492-15888
#141-5752 434-13579
#139-5670 *399-12557
#483-15472 496-16009
#466-14886
#494-15991
#493-15960
                 065642
002636
002353
SFATAL
$FILLC
$FILLS
SGTSWR
                  063712
                  061106
SHALT
                 066014
065740
065140
055330
SHALT2
SHIBTS
                                                            388-12256 *487-15704 #487-15715

428-13378 #429-13457 429-13458

496-16083

*465-14843 465-14845 465-14855

496-16020

496-16020

496-16029

255-8824 *255-8834 255-8835

*462-14784 462-14788

255-8824 *255-8835 *255-8851
                                          #493-15960

386-12214

425-13300

#354-11364

#139-5557

#341-11132

#361-11589

#141-5756

#343-11161

#141-5736

*462-14781

#141-5737

466-14889
SHIOCT
$ILLUP
                 044050
002013
042644
045100
002645
042736
002606
SINVAL
SITEMB
                                                                                   465-14845 465-14855 *465-14856 468-14933
SKERNE
SKMAP
$LF
SLOADC
                                                                                $LPADR
                                                                                                                                            257-8857 *257-8867
                                                                                                                                                                                   257-8868 *257-8888
                  002610
$LPERR
                                                                                                                                                                                   462-14781 +462-14785
```

CZMSPA	CREATED BY	MACRO	ON 2	6-APR-82 AT	00.54	PAGE 27					SEQ 04	2
SYMBOL	CROSS REFERENCE			0-AFR-02 AT	07.34	CREF						
SYMBOL \$L\$	= 000000	#157-6 #316-1	1111	#159-6155 #319-10460	#159-6176 #321-10512	#160-6259 #322-10556	#160-6260 #323-10632	#282 - 9508 #323 - 10634	#291 - 9806 #324 - 10675	#291-9808 #325-10734	#316-10393 #325-10736	
\$MADR1 \$MADR2 \$MADR3 \$MADR4	065672 065676 065702 065706	#325-1 #492-1 #492-1 #492-1	5918 15922 15925									
\$MAIL \$MAMS1 \$MAMS2 \$MAMS3	065640 065670 065674 065700	#492-1 #492-1 #492-1 #493-1 #493-1	5886 210 15920 15923	493-15961 #492-15911	493-15965							
SMAMS4 SMBADR SMNEW SMSGAD	065704 065742 064760 065654	#493-1 483-1 #492-1	5926 15961 15474 15893	#486-15664								
SMSGLG SMSGTY SMSWR SMTYP1	065656 065640 064747 065671	*466-1	4899	#492-15887								
SMTYP2 SMTYP3 SMTYP4 SNOTRA	065675 065701 065705 066010	#492-1 #492-1 #492-1 #494-1	5921 15924 15927 15990	496-16005	496-16007	496-16060	496-16061	496-16062	496-16063	496-16064	496-16086	
\$NULL \$NWTST	= 002352	496-1 #139-5 #157-6 #172-6 *481-1	669	496-16005 496-16088 434-13581 157-6063 172-6879 *481-15353 #491-15878	496-16089 #157-6063 #172-6879	#167-6074 #187-7242	496-16091 167-6694 187-7242	#167-6694 #187-7242	#169-6731 #187-7259	169-6731 187 - 7259	#169-6731 #187-7259	
SOCNT SOCTVL SOCT8 SOMODE	063306 065622 = 065626 063310	*481-1 491-1 473-1 *481-1	5853 15097	#491-15XX1			*481-15342	#481-15368				
SOVER SPASS	060436 065646	462-1 *153-5	4761	*481-15323 462-14782 *189-7291 227-8214	481-15328 #462-14787 *189-7292 229-8228	189-7294 229-8238	189-7298 231-8253	189-7326 234-8373	*189-7332 363-11614	226-8179 395-12482	226-8188 #492-15890	
SPASTM SPATMA SPER01 SPER02 SPER03 SPER04 SPER10 SPER11 SPER12 SPER13 SPER14 SPER15 SPER16	065746 002010 056506 056534 056562 056612 056674 056746 056766 057010 057030 057052	226-8 #493-1 #139-5 #451-1 #451-1 #451-1 #451-1 #453-1 #453-1 #453-1 #453-1	1554 14478 14484 14496 14512 14517 14522 14533 14538 14548	429-13447 496-16032 496-16033 496-16035 496-16036 496-16037 496-16038 496-16040 496-16041 496-16042 496-16043		*463-14793	*463-14797	463-14802	463-14803	465-14829		
SPER17 SPER20 SPER21 SPER22	057114 057132 057150 057170	#453-1 #453-1 #453-1	4553 4558 4563 4568	496-16044 496-16045 496-16046 496-16047								

CZMSP		MACRO ON 26-APR-82 A	T 09:54	PAGE 28 CREF					SEQ 042
SYMBO SPER2 SPER2 SPER2 SPER3 SPER3 SPER3 SPER3 SPER3 SPER3 SPER3 SPER3 SPER3 SPER3 SPER3	VALUE 057206 4 057224 5 053762 6 057414 7 057434 0 054210 1 057624 2 057722 3 057770 4 060050 5 060102 6 060136 N 054356	REFERENCES #453-14573 496-1604 #453-14578 496-1605 #421-13213 496-1605 #456-14615 496-1605 #456-14619 496-1605 #458-14658 496-1605 #460-14676 496-1605 #460-14685 496-1605	89012345						
\$RAND \$RDCH \$RDDE \$RDDI \$READ \$READ \$RESR \$SAVR \$SCOP \$STN \$SVLA \$SWR \$SWR \$SWR \$TEST \$TKB	R 064256 C 065142 N 064406 T 064772 C 043032 E 065366 E 065330 6 055334 E 060166 = 000001 D 060422 = 163000 065662	#460-14705	5 3 4 60 8 7 80 *428-13381 60 *169-6731 78 *462-14784 167-6694	#172-6879 169-6731 390-12348	#429-13459 #187-7242 172-6879 390-12425 485-15551	#187-7259 187-7242 393-12454 486-15571	187-7259 393-12473 486-15577	397-12525	397-12543
STKS STN	002626	#141-5749 331-1087 434-13625 434-1363 #141-5748 331-1087 434-13623 434-1363 #108-4555 157-6063 #169-6731 172-687	331-10912 9 483-15445 157-6063 172-6879	390-12350 483-15477 #157-6063 #172-6879	390-12423 485-15549 167-6694 187-7242	393-12456 486-15569 167-6694 187-7242	393-12473 486-15577 393-12472 486-15575 #167-6694 #187-7242	397-12527 169-6731 187-7259	397-12542 169-6731 187-7259
STPB STPFL STPS STRAF STRAF STRAF	002632 065754 02 065776	#141-5748	434-13558 20 75 01						
STSTA STSTA STTY! STYPE	D 043052 N 064704 OS 063312	#344-11186 496-1608 486-15598 486-1559 #482-15380 496-1608	99 486-15621 06		486-15653	#486-15657			
STYPE STYPE STYPE STYPE	C 056300 X 056466 C 063110	434-13578 434-1358 434-13645 434-1368 434-13645 434-1368 4481-15322 496-1608 481-15321 #481-1538	95 #434-13589 #7 #434-13650 93	483-15501					

						CEO O	1/20
CZMSPA SYMBOL	CREATED BY CROSS REFERENCE	MACRO ON 26-APR-82 AT	09:54 PAGE 29			SEQ 0	1429
SYMBOL STYPOS ST2	VALUE 063064 = 000053	REFERENCES #481-15317 496-16004 157-6111 159-6155 316-10395 319-10460 325-10737	159-6176 160-6259	160-6260 282 6 323-10632 323	2-9508 291-9806 3-10634 324-10675	291-9808 316-10393 325-10734 325-10736	
SUNIT SUNITM	065652 065750	*189-7330 *462-14733 #493-15964	#492-15892				
SUSWR SVECT1	065664	189-7326 #492-15903					
SVECT2 SWASDB SWASSB SWASSD SWASSD SWASSS	065664 065710 065712 043566 043422 043702 043536 060314	#492-15929 #492-15930 #350-11294 496-16074 #348-11259 496-16072 #350-11320 496-16075 #348-11285 496-16073 #462-14763 #189-7301 466-14906					
\$ZAP42 \$\$T	015212 = 000404	#157-6111 #157-6111 #160-6260 #282-9508 #316-10395 #316-10395 #323-10632 #323-10634	#159-6155 #159-6155 #282-9508 #291-9806 #319-10460 #319-1046 #323-10634 #324-1067	#291 - 9806 #291 0 #321 - 10512 #321	1-9808 #291-9808 1-10512 #322-10556	#160-6259 #160-6260 #316-10393 #316-10393 #322-10556 #323-10632 #325-10736 #325-10736	
\$0FILL	063307	#325-10737 #325-10737 *481-15318 *481-15322	481-15332 #481-1536	7			

MACRO CROSS RE MACRO NAME	REFERENCES	RO ON 26-AI	PR-82 AT 09	:54 PAG	E 30					SEQ 0430
AND ANDB BEGIN	#108-4547 #108-4545	#182-7052	#209-7661 #478-15194	#211-7710	#348-11262	#350-11297	#354-11352	#363-11616	#363-11636	#400-12601
BMOV	#426-13338 #126-5283 222-8124 236-8412 328-10848	169-6748 222-8143 236-8413 344-11203	220-8022 222-8144 236-8415 363-11601	220-8044 232-8296 236-8420 365-11740	220-8070 232-8299 238-8438 367-11799	222-8094 232-8303 240-8507 367-11815	222-8097 232-8306 242-8553 430-13479	222-8098 234-8355 244-8609 430-13482	222-8099 234-8356 246-8647	222-8123 234-8357 246-8678
CASE	#108-4542 #283-9608 #371-11907	#215-7862 #144-5792 #327-10821 #371-11908	#380-12119 #184-7151	#213-7797		#213-7852 #352-11334	#240-8530 #363-11673	#240-8541 #365-11733	#273-9289 #365-11750	#279-9461 #371-11905
CLEARB	#108-4542	267-9053 287-9677	267-9072 287-9683	273-9261	273-9267	275-9358	275-9364	279-9427	279-9433	283-9575
DOWNTO	283-9581 #108-4545 #108-4544 160-6255 184-7163 296-9914 335-10994 421-13221	153-5977 167-6726 186-7217 297-9941 344-11209 421-13252	153-5982 170-6867 217-7910 316-10365 348-11279 434-13635	154-5995 170-6872 217-7925 316-10389 350-11314 454-14590	158-6135 172-6909 226-8195 323-10618 377-12026 454-14602	159-6162 173-6941 246-8686 325-10700 382-12153 458-14651	159-6169 174-6979 253-8816 325-10715 382-12161 460-14699	160-6199 174-6994 291-9766 325-10730 390-12368 466-14909	160-6230 182-7064 291-9802 325-10748 393-12446 478-15233	160-6248 184-7160 296-9898 335-10991 397-12517 478-15248
END	421-13221 485-15555 #108-4545 154-5997 159-6165 160-6258 173-6932 174-7007 184-7166 289-7666 213-7843 #227-8215 238-8470 240-8539 291-9798 297-9953 319-10457 325-10702 335-10996 350-11309 #363-11648 363-11690 390-12327 395-12505 421-13254 454-14604 465-14837 466-14908 478-15250 #115-4987	186-7239 211 - 7735	149-5929 155-6054 159-6171 167-6728 173-6934 #182-7063 184-7176 186-7240 #211-7739 217-7912 #229-8239 238-8492 253-8818 296-9901 297-9957 321-10507 325-10725 #348-11270 354-11357 363-11653 371-11949 390-12365 426-13344 458-14660 465-14867 470-15003 485-15557 167-6719	151-5942 155-6055 160-6201 169-6841 173-6935 #182-7066 184-7180 187-7267 211-7740 217-7919 #231-8254 238-8493 273-9226 296-9917 297-9964 322-10543 348-11271 354-11359 363-11654 377-12029 390-12372 397-12537 426-13346 458-14665 465-14872 478-15201	151-5959 157-6098 160-6212 169-6844 173-6936 182-7068 184-7181 189-7297 211-7741 217-7927 #234-8374 238-8494 273-9227 296-9920 297-9969 322-10553 325-10747 348-11273 354-11360 363-11655 377-12045 426-13347 458-14668 465-14875 478-15203	153-5969 157-6104 160-6222 170-6870 173-6943 182-7069 184-7204 189-7334 211-7742 219-7971 238-8457 238-8457 238-8457 238-8457 238-8457 238-10599 325-10752 348-11274 #363-11656 380-12105 393-12466 400-12612 428-13412 458-14671 465-14876 478-15204	153-5972 157-6110 160-6232 170-6874 174-6981 182-7070 186-7219 189-7335 211-7748 #226-8180 238-8463 240-8521 282-9492 297-9940 316-10367 323-10613 325-10753 348-11281 #363-11663 380-12143 393-12467 400-12613 428-13414 460-14701 465-14881 478-15217	153-5984 158-6134 160-6237 172-6911 174-6996 182-7071 186-7230 191-7348 211-7749 #226-8189 238-8464 240-8522 282-9506 297-9945 316-10385 323-10630 326-10783 #350-11305 363-11664 382-12155 395-12497 400-12628 428-13415 462-14734 466-14890 478-15232	153-5985 158-6137 160-6250 172-6920 174-7005 184-7162 186-7233 191-7349 211-7751 226-8197 238-8465 240-8523 291-9768 297-9949 316-10392 324-10662 326-10793 350-11306 363-11633 363-11670 382-12165 395-12503 421-13218 434-13637 462-14757 466-14901 478-15235	153-5986 159-6164 160-6252 173-6931 174-7006 184-7165 186-7235 209-7665 213-7818 #226-8204 238-8466 240-8532 291-9786 297-9950 317-10420 324-10671 335-10993 350-11308 363-11635 363-11672 390-12317 395-12504 421-13223 454-14592 462-14760 466-14907 479-15247
FATAL	#115-4987	167-6715	167-6719	303-10062						

0034M VG 035									SEQ 0431
TED BY MACRO	ON 26-API	R-82 AT 09:	54 PAGE CREF	31					
#108-4545 211-7707 297-9962 400-12626	149-5923 211-7715 315-10322 426-13339	172-6899 211-7726 #348-11263 #428-13407	182-7041 238-8442 #350-11298 #478-15195	184-7155 238-8443 #354-11353 478-15212	184-7170 238-8471 #363-11617 478-15227	184-7177 238-8472 363-11640 478-15242	186-7211 240-8514 393-12461	191-7341 296-9894 395-12490	211-7706 296-9902 397-12531
157-5101 160-6228 170-6863 174-6976 184-7186 #209-7662 217-7908 226-8190 229-8230 238-8445 246-8680 251-8798 282-9483 #296-9909 297-9947 324-10658 335-10970 350-11307 363-11644 377-12042 # 390-12322 397-12533 434-13633 458-14661 462-14758	158-6126 160-6235 170-6864 174-6991 184-7202 209-7663 217-7917 226-8193 229-8236 238-8452 246-8713 253-8814 282-9488 296-9910 316-10379 324-10667 335-10988 354-11358 353-11645 377-12043 363-11645 377-12043 390-12323 400-12606 454-14663	#229-8237 238-8458 247-8720 273-9222 282-9501 #296-9911 317-10416 325-10713 335-10989 363-11649 380-12101 390-12333 421-13214 #454-14586 458-14666 465-14823	229-8238 238-8467 248-8732 273-9224 283-9577 296-9923 319-10453 325-10721 335-11000 #363-11613 363-11657 382-12150 390-12361 421-13219 454-14588 458-14669 465-14835	465-14874 153-5980 159-6159 160-6253 172-6905 #182-7053 186-7237 211-7712 222-8076 226-8205 231-8252 238-8474 249-8747 273-9263 283-9578 #297-9930 321-10493 325-10743 335-11023 363-11614 363-11658 382-12159 390-12366 421-13227 454-14597 460-14676 465-14860 470-15001	460-14685 465-14861 478-15202	154-5992 159-6166 160-6265 172-6916 #184-7156 189-7294 213-7811 226-8179 227-8214 231-8256 238-8487 251-8766 275-9360 287-9680 #297-9932 322-10539 #325-10749 344-11207 363-11631 371-11917 388-12265 393-12463 423-13260 454-14600 460-14696 465-14870 478-15226	388-12266 395-12488 425-13315 456-14609 #460-14697 465-14873 478-15241	465-14878 485-15534	466-14883 485-15553
#108-4547 316-10387	153-5967 323-10615	153-5970 325-10698	325-10728	395-12493	180-7228	180-7231	291-9/04	291-9800	316-10363
#160-6200 #160-6223 #160-6263 #231-8258 #251-8802 #282-9504 #291-9783 #296-9907 #297-9937 #316-10364 #317-10410 #319-10440 #321-10509	1160-6202 1160-6227 1160-6266 1238-8447 1282-9474 1291-9761 1291-9784 1296-9912 1297-9942 1317-10411 1319-10441 1321-10479	#160-6204 #160-6229 #160-6267 #238-8475 #282-9475 #291-9762 #291-9790 #296-9913 #297-9946 #316-10371 #317-10413 #319-10442 #321-10486	#211-7738 #158-6127 #160-6205 #160-6231 #160-6271 #249-8750 #282-9476 #291-9763 #291-9763 #291-9795 #296-9915 #297-9951 #316-10372 #317-10415 #319-10444 #321-10488	#348-11269 #158-6130 #160-6238 #160-6238 #184-7159 #249-8751 #282-9478 #291-9765 #291-9766 #296-9916 #297-9952 #316-10375 #317-10417 #319-10448 #321-10494	#350-11304 #158-6131 #160-6209 #160-6239 #184-7161 #251-8769 #282-9480 #291-9767 #296-9896 #296-9918 #297-9965 #316-10377 #317-10418 #319-10450 #321-10495	#160-6193 #160-6213 #160-6241 #184-7164 #251-8777 #282-9481 #291-9771 #296-9897 #296-9919 #297-9968 #316-10380 #319-10430 #319-10454 #321-10498	#160-6195 #160-6214 #160-6244 #184-7179 #251-8785 #282-9490 #291-9772 #296-9899 #296-9925 #316-10360 #316-10381 #319-10453 #319-10455 #321-10503	#160-6196 #160-6218 #160-6245 #211-7729 #251-8793 #282-9499 #291-9775 #296-9900 #296-9926 #316-10361 #316-10382 #319-10458 #321-10504	#321-10508 #322-10551
	REFERENCES #108-4545 211-7707 297-9962 400-12626 #108-4546 #108-4544 157-6101 160-6228 170-6863 174-6976 184-7186 #209-7662 217-7908 226-8190 229-8230 238-8445 246-8680 251-8798 282-9483 #296-9909 297-9947 324-10658 335-10970 350-11307 363-11644 377-12042 390-12322 397-12533 434-13633 458-14661 462-14758 466-14888 #108-4546	REFERENCES #108-4545 211-7707 297-9962 400-12626 #426-13339 #108-4546 #108-4546 #108-4546 #108-4546 #108-6228 160-6228 160-6235 170-6863 170-6863 170-6863 170-6864 174-6976 184-7186 184-7202 #209-7662 217-7908 227-8230 229-8230 229-8236 238-8445 246-8680 229-8230 229-8236 238-8445 246-8680 229-8230 229-8236 238-8445 246-8680 229-8230 239-8230 239-8236 238-8445 246-8680 231-8798 249-9909 297-9947 335-10970 335-10970 335-10978 335-10970 335-10988 #296-9909 #296-9910 335-10677 335-10677 3363-11644 363-11645 377-12042 #377-12043 3390-12322 390-12323 397-12533 400-12606 434-13633 454-14585 458-14661 458-14663 462-14758 462-14761 466-14888 466-14894 #108-4546	REFERENCES #108-4545 211-7707 297-9962 400-12626 #426-13339 #428-13407 #108-4546 #108-4546 #108-4546 #108-4546 #108-4546 #108-4546 #108-6268 #160-6228 #160-6238 #170-6863 #180-7202 #180-7045 #180-7045 #180-909 #190-9910 #190-9910 #190-9911 #190-9	REFERENCES #108-4545 149-5923 172-6899 182-7041 211-7707 211-7715 211-7715 211-7726 238-8442 297-9962 400-12626 #426-13339 #428-13407 #478-15198 #108-4546 390-12316 393-12469 397-12539 #108-4544 151-5937 151-5952 #153-5978 157-6101 158-6128 160-6228 160-6235 160-6243 160-6246 170-6863 170-6863 170-6864 172-6903 172-6904 174-6976 174-6991 182-7045 184-7186 184-7202 186-7208 186-7208 186-7215 #209-7662 209-7663 211-7679 211-7680 217-7908 217-7917 217-7923 229-8230 229-8230 229-8236 229-8236 229-8236 229-8236 229-8236 229-8236 229-8236 229-8236 229-8236 229-8236 229-8236 229-8236 229-8237 229-8238 229-8236 229-8236 229-8236 229-8237 229-8237 229-8238 229-8236 229-8237 229-8238 238-8452 238-8458 238-8467 246-8680 246-8713 247-8720 248-8732 251-8798 251-8798 251-8798 235-10721 335-10970 335-10988 335-10989 335-11000 335-10988 335-10989 335-11001 3363-11644 363-11645 363-11649 363-11649 363-11649 363-11649 363-11649 363-11649 363-11649 363-11649 363-11649 363-11649 363-11649 363-11649 363-11649 434-13633 454-14588 458-14666 458-14689 466-14898 466-14902 #108-4546 #	### REFERENCES #*1008-4545* 149-5923* 172-6899* 182-7041* 184-7155* 211-7707* 211-7707* 211-7715* 211-7726* 238-8442* 238-8443* 2400-12626* #426-13339* #428-13407* #478-15195* 478-15212* #108-4546* 390-12316* 393-12469* 397-12539* 465-14874* #108-4546* 151-5937* 157-6101* 158-6128* 160-628* 160-628* 160-628* 160-628* 174-6976* 174-6976* 174-6976* 174-6976* 174-6976* 174-6976* 174-6976* 174-6976* 174-6976* 174-6976* 174-6976* 174-6976* 174-6976* 174-6976* 174-6976* 174-6976* 174-6978* 182-7045* 182-7045* 182-7050* 184-7186* 184-7202* 186-7208* 186-7215* 186-7237* #209-7662* 209-7663* 2211-7679* #211-7680* 211-7712* 217-7908* 217-7917* 217-7908* 217-7917* 217-7923* 219-7969* 226-8190* 226-8193* 226-8202* 226-8203* 226-8203* 229-8230* 229-8236* 229-8236* 238-8452* 238-8458* 238-8467* 238-8474* 246-8680* 246-8713* 247-8720* 248-8732* 249-8747* 251-8788* 253-8814* 273-9222* 273-9224* 273-923* 329-1333* 330-12363* 330-12363* 330-12363* 330-12363* 330-12363* 330-12363* 330-12364* 3355-10970* 3355-10988* 3355-10970* 3355-10988* 3355-10970* 3355-10988* 3355-10970* 3355-10448* 3355-10713* 3355-10721* 3355-10448* 3355-10721* 3355-10721* 3355-10448* 3363-11644* 363-11645* 363-11649* 363-11644* 363-11645* 363-11649* 363-11644* 363-11645* 363-11649* 363-11644* 363-11645* 363-11644* 363-11645* 363-11646* 466-14888* 466-14888* 466-14898* 466-14898* 466-14898* 466-14898* 466-14898* 466-14898* 466-14898* 466-14898* 466-14898* 466-14898* 466-14898* 466-14898* 466-14898* 466-14898* 466-14898* 466-14898* 466-14898* 466-14898* 466-14898* 466-14998* 470-15008* 325-10073* 325-10073* 325-10698* 325-10738* 325-10728* 329-12333* 329-12333* 3390-12361* 3390-12366* 3390-12368* 3390-12368* 3390-12369* 3390-12369* 3390-12369* 3390-12369* 3390-12369* 3390-12369* 3390-12369* 3390-12369* 3390-12369* 3390-12369* 3390-12369* 3390-12369* 3390-12369* 3390-12369*	## 108-4545 ## 108-4545 ## 108-4545 ## 108-4545 ## 108-4545 ## 108-4545 ## 108-4545 ## 108-4546 ##	18	## 108-6545 149-5923 172-6899 182-7041 184-7155 184-7170 184-7177 186-7211 184-7170 181-7170	**************************************

		DO DN 24-45	D 92 AT 00	.S/ DAC	. 72					SEQ 043
MACRO CROSS	REFERENCE		PR-82 AT 09:	CREI						
MACRO NAME	REFERENCES #323-10588	#323-10591	#323-10594	#323-10595	#323-10596 #324-10654 #325-10699 #326-10770 #421-13216 #451-14522 #453-14539 #453-14555	#323-10605	#323-10608	#323-10609	#323-10610	#323-10631
	#323-10633	#324-10648 #325-10695	#324-10649	#324-10650	#324-10654	#324-10660 #325-10701	#324-10668 #325-10704	#324-10673	#324-10674	#325-10689 #325-10710
	#325-10694 #325-10712	#325-10722	#326-10763	#326-10768	#326-10770	#326-10771	#326-10773	#326-10779	#326-10780	#326-10789
	#326-10790 #423-13262	#363-11677 #451-14517	#365-11737	#421-13213	#421-13216 #451-14522	#421-13220	#451-14524	#451-14527	#451-14528	#451-14529
	#453-14533	#453-14534	#453-14535	#453-14538	#453-14539	#453-14540	#453-14543	#453-14544	#453-14545	#453-14548
	#453-14549	9 #453-14550 5 #453-14568	#453-14569	#453-14570	#453-14573	#453-14574	#453-14575	#453-14578	#453-14579	#453-14580
***	#456-14615		#456-14620 #169-6761		#174-6948		#242-8551	#244-8580	#246-8644	#255-8822
MAP	#128-5346 #257-8855	#377-12059	#386-12230	#388-12269	#390-12330	#390-12363	#390-12428	#393-12475	#397-12545	#430-13475
NEWTST	#432-13516 #117-5037	#157-6063	#167-6694	#169-6731	#172-6879	#187-7242	#187-7259			
ON.ERR	#108-4547	157-6108 233-8313	174-7008	193-7378	195-7408	197-7445 242-8547	199-7482 244-8576	201-7520 246-8640	203-7565 348-11265	205-7603 348-11277
	207-7648 350-1130	350-11312	233-8321 354-11355	234-8331 363-11688	426-13341	428-13409	478-15197	240-0040	340-11203	340 11211
ON.NOE OR	#108-4547 #108-4544	149-5925	174-6965	174-6975	174-6990	211-7723				
ORB	#108-4547	***** ****	****	****	#144-4477	#140-4741	#169-6837	#173-6937	#174-6948	#174-7021
POP	#108-4546 #174-7023	#155-6059 #182-7104	#160-6272 #189-7306	#166-6540 #211-7708	#166-6637 #211-7734	#169-6761 #211-7761	#213-7848	#213-7856	#242-8551	#244-8580
	#246-8644 #348-1127	#255-8822 6 #350-11311	#255-8851 #354-11361	#257-8855 #354-11370	#257-8888 #355-11408	#327-10832 #355-11424	#331-10914 #357-11461	#343-11174	#343-11182 #361-11607	#344-11206 #365-11732
	#365-11758	8 #365-11770	#369-11852	#369-11873	#371-11959	#377-12059	#379-12091	#380-12114	#382-12154	#382-12156
	#382-12164 #390-1232	4 #384-12195 1 #390-12326	#390-12330	#390-12363	#386-12245 #390-12371	#390-12424	#390-12426	#390-12427	#388-12297 #390-12428	#393-12474
	#393-1247	5 #395-12506	#397-12544 #428-13395	#397-12545 #428-13401	#399-12555 #428-13402	#399-12558	#400-12605	#406-12653 #429-13432	#421-13249	
	#428-1338 #429-1344	9 #429-13455	#430-13475	#430-13495	#432-13516	#432-13529	#434-13650	#434-13653	#434-13654	#456-1462
	#458-1464	4 4/05-155//	M/ DE_1EE/ D	#/ D7_15705	#487-15763	#476-15180 #488-15808	#478-15205 #489-15837	#478-15224		
PUSH	4100-1514	155-6041	155-6047	155-6059	160-6192	4// / 570	9// //75	169-6761	169-6826 213-7791	173-6937 242-8551
	174-6948 244-8580 344-1118	246-8644	255-8822	255-8824	257-8855	257-8857	327-10809	331-10869	343-11161	343-1117
	344-1118	6 348-11259 8 365-11723	350-11294 365-11762	354-11350 369-11843	354-11365 369-11867	355-11374 371-11904	355-11399 377-12059	379-11442 379-12075	361-11514 379-12080	379-1208
	380-1210	155-6041 174-6952 246-8644 6 348-11259 8 365-11723 8 380-12109 8 393-12440 2 425-13306 3 430-13473 472-15031 1 487-15732	155-6047 174-6954 255-8822 350-11294 365-11762 386-12209 393-12475 425-13313 430-13475 472-15045 488-15793	155-6059 182-7072 255-8824 354-11350 369-11843 386-12230 395-12481 426-13329 432-13514 473-15069	189-7304 257-8855 354-11365 369-11867 388-12251 397-12511 426-13335 432-13516 476-15156	#211-7681 257-8857 355-11374 371-11904 388-12269 397-12545 426-13343 434-13589 476-15172	166-6625 211-7708 327-10809 355-11399 377-12059 390-12303 399-12552 426-13349 434-13593 478-15190	169-6761 211-7730 331-10869 357-11442 379-12075 390-12330 #421-13228 426-13352 434-13594 478-15208	213-7791 343-11161 361-11514 379-12080 390-12363 423-13268 426-13356 456-14624 482-15380	242-8551 343-11179 361-11589 379-12084 #390-12369 425-13298 426-13361 458-14634 485-15523
	425-1330	2 425-13306	425-13313	426-13329	426-13335	426-13343	426-13349	426-13352	426-13356	426-13361
	426-1336	3 430-13473 4 472-15031	430-13475	432-13514 473-15069	432-13516	434-13589	434-13593 478-15190	434-13394	482-15380	485-1552
	487-1568	1 487-15732	488-15793	489-15822						
RCC RCS	#108-4543		~							
REPEAT	#108-4545	#157-6088	#159-6145	#159-6157	#160-6224	#160-6233 #323-10577	#282-9495 #324-10651	#291-9760 #325-10688	#291-9769 #325-10693	#316-10359 #325-1070
	#400-1260	2 #317-10436	#361-10404	#322-10330	#JEJ-107/E	#J6J-10J//	#364-10031			
REQ RES4	#108-4543 #133-5458	6 348-11259 8 365-11723 8 380-12109 8 393-12440 2 425-13306 3 430-13473 4 472-15031 1 487-15732 #157-6088 8 #319-10438 2 #145-5860 6 #388-12298	#151-5956	#164-6446	#167-6721	#169-6781	#382-12176	#382-12180	#384-12192	#384-12204
	#386-1224	6 #388-12298								
RGE	#105-4343									

IACRO CROSS	REFERENCE		PR-82 AT 09	:54 PAGE	33					SEQ 0433
ACRO NAME	REFERENCE #108-4543 #108-4543 #108-4543 #108-4543 #108-4543 #108-4543 #108-4543 #108-4543 #108-4543 #108-4543 #154-5993 #187-7263 #213-7813 #244-8577 #275-9345 #348-1126	3								
HIS LE LO	#108-4543 #108-4543 #108-4543									
LOS	#108-4543 #108-4543									
MI	#108-4543 #108-4543	157-6114								
PL ET	#108-4542 #154-5993	#137-5540	#147-5864 #154-6015	#147-5866	#153-5968 #166-6646	#153-5971 #169-6821	#153-5974 #169-6840	#153-5979 #174-7027	#153-5981 #182-7067	#153-5983 #182-7146
	#187-7263 #213-7812	#154-6012 #187-7265 #213-7837	#191-7344 #232-8265	#191-7347 #233-8324	#209-7656 #234-8332	#209-7657 #238-8441	#209-7671 #240-8502	#211-7727 #240-8513	#211-7737 #240-8525	#211-7759 #242-8548
	#244-8577 #275-9345	#246-8641 #277-9416	#253-8810 #277-9419	#273-9239 #282-9489	#273-9242 #282-9502	#273-9252 #291-9794	#305~10121 #325~10750	#305-10133 #326-10781	#321-10505 #326-10791	#322-10549 #337-11069
	#348-1126	58 #350-11303 53 #371-11948	#363-11650 #371-11952	#147-5866 #166-6602 #191-7347 #233-8324 #273-9239 #282-9489 #324-10669 #363-11712 #371-11955 #399-12563	#371-11906 #371-11958	#371-11915 #380-12111	#371-11927 #382-12174	#371-11931 #384-12190	#371-11934 #384-12199	#153-5983 #182-7146 #211-7759 #242-8548 #275-9342 #322-10549 #337-11069 #371-11937 #390-12353 #460-14680
	#371-1194 #393-1245 #460-1468	7 #323-10611 8 #350-11303 3 #371-11948 59 #395-12496 32 #460-14691	M400-14073	#399-12563 #460-14715	#399-12569 #460-14717	#400-12615 #466-14910	#458-14644 #472-15034	#458-14655 #472-15048	#458-14672 #473-15072	#460-14680 #476-15157
TB	#476-1516 #108-4542 #133-5446	9 #4/0-131/3	#4/6-151/5		4/5 50//	1/5 5954	151-50/0	154-6004	154-6006	154-6008
T4	157-6087 382-1217	163-6339 72 384-12187	145-5818 163-6355 386-12229	145-5825 163-6375 388-12268	145-5844 163-6408	145-5856 164-6426	151-5949 166-6480	167-6705	169-6742	375-12001
MACIT	#108-4541 #119-5102 153-596 166-6449	108-4556 144-5788 154-5988 167-6723 199-7452				147-5873 155-6057	149-5893	149-5922	151-5933	151-5945
	153-596. 166-644	154-5988 167-6723	144-5800 154-6000 170-6856 201-7489	155-6032 182-7036	155-6050 184-7150	155-6057 186-7207	158-6120 187-7273	159-6140 191-7339	160-6181 193-7355	164-6419 195-7385
	217-790	219-7962	220-8012	220-8025	220-8047	222-8075	222-8111	222-8129	151-5933 160-6181 193-7355 213-7766 224-8151 231-8251 244-8574 251-8939 273-9153 297-9960 309-10190 323-10560 339-11119 367-11774 373-11986 386-12208 399-12572 432-13499	215-7861 224-8158 232-8264
	233-831	233-8319	234-8329	234-8371 248-8731	236-8382 248-8743	238-8429 249-8746	240-8500 250-8755	242-8545 251-8765	244-8574 251-8773	232-8264 246-8638 251-8781
	251-878 263-898	251-8797 263-8988	253-8808 265-8998	253-8813 265-9011	255-8821 265-9020	259-8902 267-9034	259-8913 269-9077	259-8925 271-9117	261-8939 273-9153	261-8962 275-9294
	277-9370 301-100	282-9467 26 301-10034	283-9515 303-10044	285-9614 307-10138	289-9716 307-10145	291-9750 307-10163	307-10176 321-10466	309-10184 322-10519	309-10190 323-10560	311-10241 324-10640
	325-1066 354-113	80 326-10756 64 355-11373	327-10802 355-11427	327-10808 357-11434	328-10835 361-11503	329-10854 363-11611	331-10868 365-11716	333-10920 365-11761	339-11119 367-11774	354-11349 369-11822
	369-118 377-120	42 369-11855 22 377-12031	369-11865 377-12058	369-11876 379-12068	371-11883 380-12097	373-11963 382-12147	373-11974 382-12169	373-11982 384-12184	373-11986 386-12208	375-11993 388-12250
	390-123 400-125	219-7962 226-8177 233-8319 246-8698 251-8797 263-8988 282-9467 301-10034 314-10307 80 326-10756 355-11373 42 369-11855 22 377-12031 391-12433 98 400-12618 95 456-14608 255-8844 8 #169-6771 7 #344-11215	393-12439 402-12632	145-5813 155-6032 182-7036 203-7527 220-8025 226-8201 234-8371 248-8731 253-8813 265-9011 285-9614 307-10138 316-10351 327-10808 357-11434 369-11876 379-12068 395-12480 404-12638 429-13429 #169-6805	147-5863 155-6050 184-7150 205-7572 220-8047 227-8212 236-8382 248-8743 255-8821 265-9020 289-9716 307-10145 317-10404 328-10835 361-11503 371-11883 380-12097 397-12510 406-12644 463-14791 430-13476 #169-6831 #388-12271	186-7207 207-7610 222-8075 229-8226 238-8429 249-8746 259-8902 267-9034 291-9750 307-10163 319-10425 329-10854 363-11611 373-11963 382-12147 399-12551 421-13226 476-15150 432-13517 #174-6951 #388-12289	158-6120 187-7273 209-7655 222-8111 229-8235 240-8500 250-8755 259-8913 269-9077 292-9818 307-10176 321-10466 331-10868 365-11716 373-11974 382-12169 399-12561 423-13267 485-15522	191-7339 211-7675 222-8129 229-8244 242-8545 251-8765 259-8925 271-9117 296-9888 309-10184 322-10519 333-10920 365-11761 373-11982 384-12184 399-12566 430-13472 485-15547	432-13499	246-8638 251-8781 261-8962 275-9294 299-9982 311-10241 324-10640 354-11349 369-11822 375-11993 388-12250 399-12579 454-14583
UPERV ESTAR	#130-537	95 456-14608 5 255-8844 8 #149-4771	426-13332	429-13429 #169-6805	430-13476	432-13517				#246-8646
ESTAR	224-816 233-831 246-866 251-878 263-898 277-937 301-100 313-102 325-106 354-113 369-118 377-120 390-123 400-125 454-145 #131-541 #257-887 #423-132 #108-454	7 #344-11215 73	201-7489 220-8012 226-8186 234-8329 247-8719 253-8808 265-8998 283-9515 303-10044 315-10319 327-10802 355-11427 369-11865 377-12058 393-12439 402-12632 456-14623 426-13332 #169-6788	#386-12231	#388-12271	#388-12289	#421-13215	#242-8552 #421-13231	#421-13241	#423-13261
THEN	#108-454	4								

MACRO CROSS RI	EATED BY MACR	0 ON 26-AF	PR-82 AT 09	:54 PAGE	34					SEQ 0434
MACRO NAME THRU TO TYPDEC TYPE	REFERENCES #108-4545 #108-4545 #125-5247 #115-5001 159-6156 184-7201 331-10892 333-10937 335-11042 384-12193	#184-7190 #154-5996 159-6172 189-7293 331-10893 333-10939 363-11632 384-12198	#184-7194 154-6010 159-6173 189-7295 331-10894 335-10949 363-11669 384-12202 388-12286 393-12449 397-12538 402-12633 468-14956 473-15097	#184-7198 154-6024 159-6177 211-7744 331-10895 335-10969 380-12099 386-12211 388-12293 393-12451 399-12553	#184-7200 154-6029 163-6416 211-7746 331-10896 335-10980 380-12102 386-12212 390-12305	#189-7298 154-6030 182-7145 331-10877 331-10905 335-10999 380-12112 386-12237 390-12306 395-12485	#395-12484 159-6142 #184-7187 331-10878 331-10906 335-11010 380-12117 386-12241 390-12315	#395-12486 159-6143 184-7191 331-10879 331-10907 335-11022 382-12148 388-12253 390-12319 395-12495 399-12580 465-14832 470-15004	#395-12501 159-6152 184-7195 331-10887 331-10908 335-11032 382-12178 388-12254 390-12324 395-12502 400-12599	#472-15025 159-6153 184-7199 331-10891 333-10934 335-11040 384-12189 388-12277 390-12334
TYPOCS	388-12281 390-12337 397-12520 400-12614 468-14931 473-15073 478-15215 481-15352 485-15524 486-15639 487-15771	388-12284 393-12442 397-12522 400-12619 468-14955 473-15082 478-15225 482-15417 485-15535 486-15641 487-15772	393-12449 397-12538 402-12633 468-14956 473-15097 #478-15228 483-15462 485-15537 486-15644 494-15990 #386-12234	483-15471 485-15539 486-15648	393-12468 399-12562 406-12651 468-14965 473-15105 #478-15234 483-15472 486-15605 487-15709	399-12483 399-12567 429-13451 468-14967 478-15191 478-15240 483-15474 486-15618 487-15711	395-12487 399-12573 434-13574 470-14994 478-15192 #478-15243 483-15483 486-15624 487-15712	478-15245 483-15488 486-15629 487-15713	#478-15249 483-15497 486-15633 487-15768	472-15039 #478-15213 478-15251 483-15515
	#485-15540	/49_1/074								
TYPOCT	483-15473	468-14936				160-6260	282-9508	291-9806	316-10393	
UNTILB USER WHILE WHILEB	#123-5194 #485-15540 #121-5148 483-15473 #108-4545 325-10734 #108-4547 #130-5396 #108-4544 #108-4548	290-9908	297-9929	297-9966		322-10556 426-13326	323-10632 429-13433	323-10634	325-10736	
SCALL SRETUR SSUBTS	#108-4548 #108-4548 #119-5109 153-5963 166-6449 197-7415 217-7907 224-8168 233-8311 246-8668 251-8789 263-8980 277-9376 301-10026 313-10261 325-10680 354-11364 369-11842 377-12022 390-12302 400-12598	#213-7849 144-5788 154-5988 167-6723 199-7452 219-7962 226-8177 233-8319 246-8698 251-8797 263-8988 282-9467 301-10034 314-10307 326-10756 355-11373 369-11855 377-12031 391-12433 400-12618	#213-7857 144-5800 154-6000 170-6856 201-7489 220-8012 226-8186 234-8329 247-8719 253-8808 265-8998 283-9515 303-10044 315-10319 327-10802 355-11427 369-11865 377-12058 393-12439 402-12632	#253-8815 145-5813 155-6032 182-7036 203-7527 220-8025 226-8201 234-8371 248-8731 253-8813 265-9011 285-9614 307-10138 316-10351 327-10808 357-11434 369-11876 379-12068 395-12480 404-12638	#253-8817 147-5863 155-6050 184-7150 205-7572 220-8047 227-8212 236-8382 248-8743 255-8821 265-9020 289-9716 307-10145 317-10404 328-10835 361-11503 371-11883 380-12097 397-12510 406-12644	#363-11634 147-5873 155-6057 186-7207 207-7610 222-8075 229-8226 238-8429 249-8746 259-8902 267-9034 291-9750 307-10163 319-10425 329-10854 363-11611 373-11963 382-12147 399-12551 421-13226	#363-11671 149-5893 158-6120 187-7273 209-7655 222-8111 229-8235 240-8500 250-8755 259-8913 269-9077 292-9818 307-10176 321-10466 331-10868 365-11716 373-11974 382-12169 399-12561 423-13267	#363-11713 149-5922 159-6140 191-7339 211-7675 222-8129 229-8244 242-8545 251-8765 259-8925 271-9117 296-9888 309-10184 322-10519 333-10920 365-11761 373-11982 384-12184 399-12566 430-13472	151-5933 160-6181 193-7355 213-7766 224-8151 231-8251 244-8574 251-8773 261-8939 273-9153 297-9960 309-10190 323-10560 339-1119 367-11774 373-11986 386-12208 399-12572 432-13499	151-5945 164-6419 195-7385 215-7861 224-8158 232-8264 246-8638 251-8781 261-8962 275-9294 299-9982 311-10241 324-10640 354-11349 369-11822 375-11993 388-12250 399-12579 454-14583

CZMSPA C	REATED BY MACE	O ON 26-AF	PR-82 AT 09:	54 PAGE	35					SEQ 0435
MACRO CROSS	REFERENCE	10 ON 20 A	N 02 N 07	CRE						
MACRO NAME	REFERENCES 454-14595	456-14608	456-14623	458-14632	463-14791	476-15150	485-15522	485-15547		
\$\$END_	#108-4546	456-14608 #517-16630				#187-7242	#187-7259			
S\$NEWT .ARITH	#119-5069 #108-4547	#157-6063	#167-6694	#169-6731	#172-6879	#107-7242	#101-1239			
.EMIT	#108-4547 #108-4541	M157 /111	#1E0_41EE	#150-4174	#140-4350	#160-6260	#282-9508	#291-9806	#291-9808	#316-10393
.EMITL	#108-4541 #316-10395	#157-6111 #319-10460	#159-6155 #321-10512	#159-6176 #322-10556	#160-6259 #323-10632	#323-10634	#324-10675	#325-10734	#325-10736	#325-10737
-EMITN	#108-4541 #108-4541	157-6114								
.EMITR .GENBR	#108-4542									
.GOTO .IFARI	#108-4544 #108-4544									
.IFOPR	#108-4542									
.IS .LEAVE	#108-4542 #108-4544									
.OPADD	#108-4542									
.OPSUB	#108-4542 #108-4544									
.SIMPL	#108-4547 #108-4541									
BRAN POP	#108-4541	#157-6111	#159-6155	#159-6176	#160-6259	#160-6260	#282-9508	#291-9806	#291-9808	#316-10393
PUSH	#316-10395 #108-4541	#319-10460	#321-10512	#322-10556	#323-10632	#323-10634			#323-10/36	#325-10737
TAG	#108-4541	157-6111	159-6155	159-6176	160-6259	160-6260	282-9508 324-10675	291-9806 325-10734	291-9808	316-10393 325-10737
	316-10395	319-10460	321-10512	322-10556	323-10632	323-10634	324-100/3	323-10/34	325-10736	323-10/3/