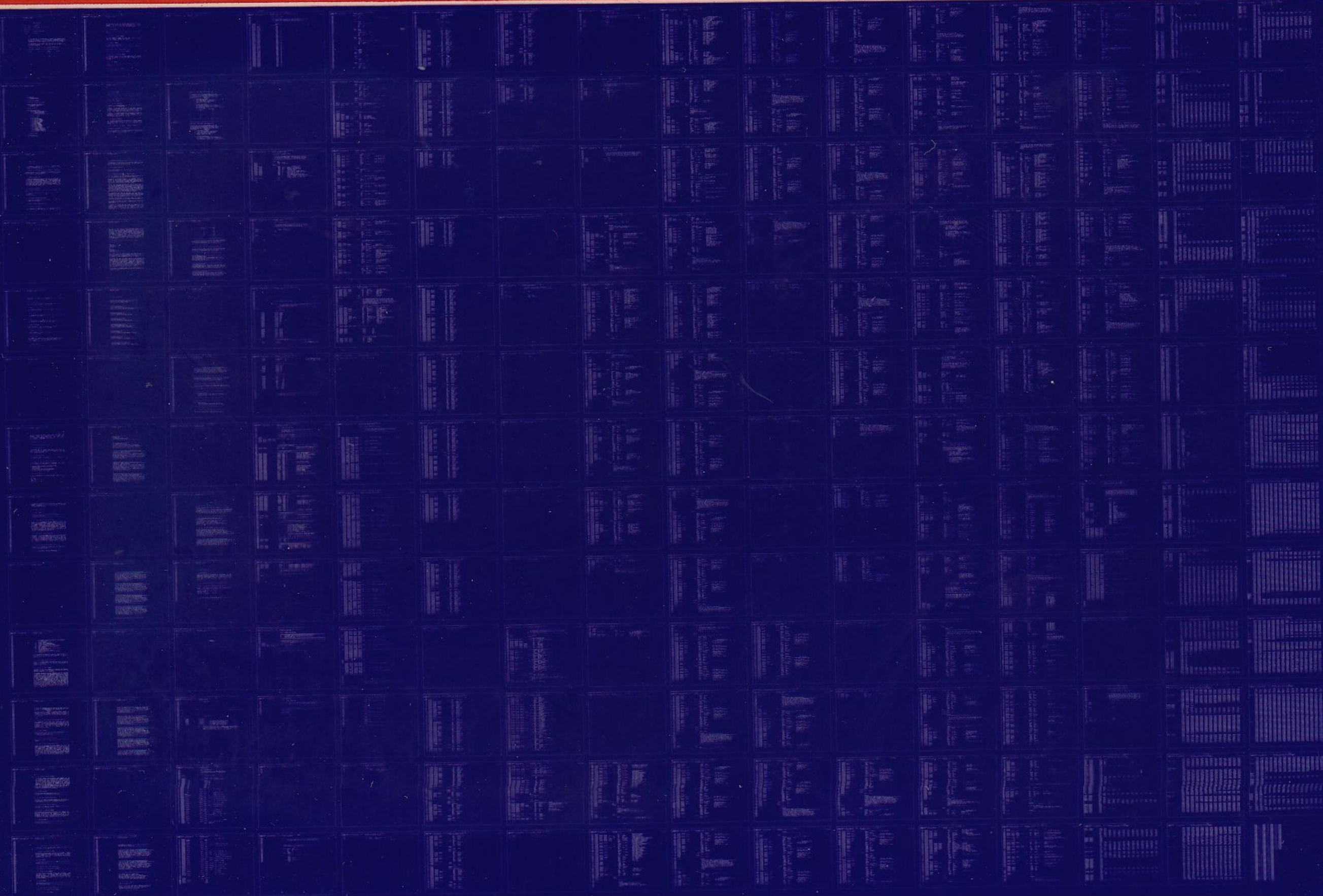


# M8207,MCPU

M8207 STATIC DIAG. #2  
CZDMQA0

AH-E229A-MC  
COPYRIGHT 1979  
FICHE 1 OF 2

SEP 1979  
**digital**  
MADE IN USA



# M8207,MCPU

M8207 STATIC DIAG. #2

CZDMQA0

AH-E229A-MC

COPYRIGHT 1979

FICHE 2 OF 2

SEP 1979

**digital**

MADE IN USA

3298  
3299  
3300  
3301  
3302  
3303  
3304  
3305  
3306  
3307  
3308  
3309  
3310  
3311  
3312  
3313  
3314  
3315  
3316  
3317  
3318  
3319  
3320  
3321  
3322  
3323  
3324  
3325  
3326  
3327  
3328  
3329  
3330  
3331  
3332

.REM @

IDENTIFICATION

PRODUCT CODE: AC-E228A-MC  
PRODUCT NAME: CZDMQA0 M8207 STATIC DIAG #2  
PRODUCT DATE: MAY, 1979  
MAINTAINER: DIAGNOSTICS MERRIMACK  
AUTHOR: ED BADGER

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

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

COPYRIGHT (C) 1979 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

TABLE OF CONTENTS

3334  
3335  
3336  
3337  
3338  
3339  
3340  
3341  
3342  
3343  
3344  
3345  
3346  
3347  
3348  
3349  
3350  
3351  
3352  
3353  
3354  
3355  
3356  
3357  
3358  
3359  
3360  
3361  
3362  
3363  
3364  
3365  
3366  
3367  
3368  
3369  
3370  
3371  
3372  
3373  
3374  
3375  
3376  
3377  
3378  
3379  
3380  
3381  
3382  
3383  
3384  
3385

- 1.0 INTRODUCTION
  - 1.1 PROGRAM ABSTRACT
  - 1.2 HARDWARE INTRODUCTION
- 2.0 HARDWARE REQUIREMENTS
- 3.0 PRELIMINARY PROGRAM REQUIREMENTS
- 4.0 GENERAL PROGRAM CONSIDERATIONS
  - 4.1 DIAGNOSTIC SUPERVISOR
  - 4.2 EXECUTION TIME
- 5.0 PROGRAM LOAD MEDIA
- 6.0 OPERATING INSTRUCTIONS
  - 6.1 LOADING AND STARTING PROCEDURES
    - 6.1.1 LOADING PROCEDURES
    - 6.1.2 STARTING PROCEDURES
    - 6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION
  - 6.2 INITIAL DIALOGUE
  - 6.3 PROGRAM OPTIONS
    - 6.3.1 START COMMAND
    - 6.3.2 RESTART COMMAND
    - 6.3.3 CONTINUE COMMAND
    - 6.3.4 PROCEED COMMAND
    - 6.3.5 ADD COMMAND
    - 6.3.6 DROP COMMAND
    - 6.3.7 PRINT COMMAND
    - 6.3.8 DISPLAY COMMAND
    - 6.3.9 FLAGS COMMAND
    - 6.3.10 ZFLAGS COMMAND
    - 6.3.11 CONTROL CHARACTERS
    - 6.3.12 HARDWARE PARAMETERS
    - 6.3.13 SOFTWARE PARAMETERS
    - 6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE
- 7.0 TEST DESCRIPTIONS
- 8.0 ERROR INFORMATION
  - 8.1 ERROR REPORTING

3387  
3388  
3389  
3390  
3391  
3392  
3393  
3394  
3395  
3396  
3397  
3398  
3399  
3400  
3401  
3402  
3403  
3404  
3405  
3406  
3407  
3408  
3409  
3410  
3411  
3412  
3413  
3414  
3415  
3416  
3417  
3418  
3419  
3420  
3421  
3422  
3423  
3424  
3425  
3426  
3427  
3428  
3429  
3430  
3431  
3432  
3433  
3434  
3435  
3436  
3437  
3438  
3439  
3440  
3441  
3442

## 1.0 INTRODUCTION

### 1.1 PROGRAM ABSTRACT

THIS DIAGNOSTIC WAS DESIGNED TO TEST OUT THE M8200, M8204, OR M8207 MICROPROCESSOR. IT IS THE SECOND OF TWO DIAGNOSTICS FOR THESE OPTIONS.

THE PROGRAM WAS IMPLEMENTED USING THE DIAGNOSTIC SUPERVISOR. THROUGH DIALOGUE WITH THE OPERATOR, THE PROGRAM WILL ALLOW MODIFICATION OF DEVICE PARAMETERS, SUCH AS UNIBUS ADDRESS, VECTOR ADDRESS, AND PROCESSOR TYPE.

### 1.2 HARDWARE INTRODUCTION

THE M820X MICROPROCESSOR USES AN EIGHT BIT DATA PATH WITH A SIXTEEN BIT INSTRUCTION MEMORY. THE INSTRUCTION MEMORY AND DATA MEMORY ARE TWO SEPARATE MEMORIES. THE MICROPROCESSOR IS DESIGNED FOR MOVING DATA AT HIGH RATES TO WORK AS A HIGH SPEED LINK BETWEEN PROCESSORS WHEN USED WITH A LINE UNIT. THE M8200 AND M8207 HAVE PROM INSTRUCTION MEMORIES. THE M8204 HAS WRITEABLE CONTROL STORE. THE MEMORY SIZES BETWEEN ALL THREE PROCESSORS VARY ALSO.

## 2.0 HARDWARE REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE M8207 LOGIC TESTS:

PDP-11/04,05,10,20,30,34,35,40,45,50,60, OR 70  
16K MEMORY  
CONSOLE TERMINAL

## 3.0 PRELIMINARY PROGRAM REQUIREMENTS

THE PROCESSOR AND MEMORY SHOULD BE THOROUGHLY TESTED PRIOR TO RUNNING THIS DIAGNOSTIC.

## 4.0 GENERAL PROGRAM CONSIDERATIONS

### 4.1 DIAGNOSTIC SUPERVISOR

THIS PROGRAM IS COMPATIBLE WITH THE STANDALONE DIAGNOSTIC SUPERVISOR, AND MUST BE LOADED TO BE CO-RESIDENT WITH THE

CZDMQA M8207 STATIC DIAG. #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 <sup>E 1</sup> PAGE 4-1  
PROGRAM DOCUMENT

SEQ 0004

3443

SUPERVISOR, OR BE PREVIOUSLY COMBINED WITH THE SUPERVISOR

3445  
3446  
3447  
3448  
3449  
3450  
3451  
3452  
3453  
3454  
3455  
3456  
3457  
3458  
3459  
3460  
3461  
3462  
3463  
3464  
3465  
3466  
3467  
3468  
3469  
3470  
3471  
3472  
3473  
3474  
3475  
3476  
3477  
3478  
3479  
3480  
3481  
3482  
3483  
3484  
3485  
3486  
3487  
3488  
3489  
3490  
3491  
3492  
3493  
3494  
3495  
3496  
3497  
3498  
3499  
3500

AND LOADED AS A SINGLE FILE. IN EITHER CASE, THE COMBINED PROGRAM WILL NOT EXCEED 16K OF MEMORY.

#### 4.2 EXECUTION TIME

THE TOTAL TIME REQUIRED TO RUN THE M8207 STATIC TESTS IS ABOUT 120 SECONDS PER PASS FOR EACH UNIT.

#### 4.3 XXDP+

THIS PROGRAM MAY BE LOADED UNDER XXDP+, AND MAY BE RUN IN DUMP MODE OR CHAIN MODE.

#### 4.4 ACT/SLIDE

THIS PROGRAM MAY BE LOADED UNDER ACT OR SLIDE AND MAY BE RUN IN DUMP MODE OR CHAIN MODE.

#### 4.5 APT

THIS PROGRAM MAY BE LOADED BY THE APT SYSTEM (INCLUDING APT-RD) AND RUN IN PROGRAM MODE OR SCRIPT MODE.

#### 4.6 MEMORY MANAGEMENT

MEMORY MANAGEMENT IS NOT UTILIZED IN THIS PROGRAM. IF IT IS INSTALLED, IT IS DISABLED BY THE PROGRAM.

#### 4.7 MEMORY PARITY OPTION

IF PARITY MEMORY IS INSTALLED, MEMORY PARITY TRAPS ARE DISABLED BY THE PROGRAM.

#### 4.8 ERROR LOGGING

THE NUMBER OF ERRORS WHICH HAVE OCCURRED ON EACH DEVICE UNDER TEST SINCE THE LAST START OR RESTART COMMAND IS KEPT IN AN ERROR LOG. THIS LOG MAY BE PRINTED BY USING THE 'PRINT' COMMAND (SEE SECTION 6.3.8).

#### 5.0 PROGRAM LOAD MEDIA

THIS PROGRAM CAN BE LOADED FROM PAPER TAPE USING THE

CZDMQA M8207 STATIC DIAG. #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 PAGE 5-1  
PROGRAM DOCUMENT

G 1

SEQ 0006

3501

ABSOLUTE LOADER OR FROM ACT, SLIDE, OR APT SYSTEMS, OR FROM



3503  
3504  
3505  
3506  
3507  
3508  
3509  
3510  
3511  
3512  
3513  
3514  
3515  
3516  
3517  
3518  
3519  
3520  
3521  
3522  
3523  
3524  
3525  
3526  
3527  
3528  
3529  
3530  
3531  
3532  
3533  
3534  
3535  
3536  
3537  
3538  
3539  
3540  
3541  
3542  
3543  
3544  
3545  
3546  
3547  
3548  
3549  
3550  
3551  
3552  
3553  
3554  
3555  
3556  
3557  
3558

ANY MEDIA SUPPORTED BY XXDP+. WHEN USING THE PAPER TAPE ABSOLUTE LOADER, THE PROGRAM SHOULD BE LOADED FIRST, FOLLOWED BY THE DIAGNOSTIC SUPERVISOR. WHEN USING XXDP+, THE DIAGNOSTIC SUPERVISOR SHOULD BE LOADED FIRST, FOLLOWED BY THE DIAGNOSTIC PROGRAM.

## 6.0 OPERATING INSTRUCTIONS

### 6.1 LOADING AND STARTING PROCEDURES

#### 6.1.1 LOADING PROCEDURES

THIS PROGRAM MAY BE LOADED FROM PAPER TAPE USING THE ABSOLUTE LOADER. IT MAY ALSO BE LOADED FROM ANY XXDP+ LOAD MEDIA. WHEN LOADED UNDER XXDP+, THE DIAGNOSTIC SUPERVISOR WILL BE LOADED AUTOMATICALLY.

#### 6.1.2 STARTING PROCEDURES

THE PROGRAM STARTS AT LOCATION 200. USE STANDARD DEC PROCEDURES TO START THE PROGRAM.

#### 6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

THE DIAGNOSTIC CAN BE EXECUTED STANDALONE UNDER XXDP+ WITHOUT READING THE REMAINDER OF THIS DOCUMENT, AS FOLLOWS:

- A) LOAD AND START DIAGNOSTIC USING RUN COMMAND
- B) RECEIVE DIAGNOSTIC SUPERVISOR PROMPT (DR>)
- C) ENTER STA<CR>
- D) ANSWER HARDWARE AND SOFTWARE QUESTIONS
- E) GET END OF PASS MESSAGES OR ERROR MESSAGES
- F) TO END EXECUTION, ENTER CONTROL/C

## 6.2 INITIAL DIALOGUE

AFTER THE PROGRAM AND THE SUPERVISOR ARE LOADED AND THE PROGRAM IS STARTED, THE FOLLOWING IDENTIFICATION IS TYPED:

DRS LOADED  
DIAG. RUN-TIME SERVICES  
CZDMQ-A-0  
M8207 DIAG. #2 OF 2  
UNIT IS M8200.4,7  
DR>

3560  
3561  
3562  
3563  
3564  
3565  
3566  
3567  
3568  
3569  
3570  
3571  
3572  
3573  
3574  
3575  
3576  
3577  
3578  
3579  
3580  
3581  
3582  
3583  
3584  
3585  
3586  
3587  
3588  
3589  
3590  
3591  
3592  
3593  
3594  
3595  
3596  
3597  
3598  
3599  
3600  
3601  
3602  
3603  
3604  
3605  
3606  
3607  
3608  
3609  
3610  
3611  
3612  
3613  
3614  
3615

THE OPERATOR THEN PROCEEDS BY TYPING ONE OR MORE OF THE COMMANDS DESCRIBED IN THE FOLLOWING SECTION 6.3. (FOR MORE DETAILED INFORMATION, REFER TO THE DIAGNOSTIC SUPERVISOR FUNCTIONAL SPECIFICATION).

### 6.3 PROGRAM OPTIONS

#### 6.3.1 START COMMAND

```
*****  
STA(RT)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:  
<FLAG-LIST>/EOP:<INCR>  
*****
```

##### 6.3.1.1 TESTS SWITCH (/TESTS:<TEST-LIST>)

<TEST-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5:8-10 ETC.) THAT SPECIFY THE TESTS TO BE EXECUTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER IN THE DIAGNOSTIC. THEY MAY BE SPECIFIED IN ANY ORDER. TESTS WILL BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS. ON THIS AND ALL SWITCHES, THE ANGLE BRACKETS <> ARE PUNCTUATION USED IN THE DEFINITION ONLY, AND ARE NOT TO BE TYPED BY THE OPERATOR. SEE EXAMPLE AT END OF 6.3.1.5.

##### 6.3.1.2 PASS SWITCH (/PASS:<PASS-CNT>)

<PASS-CNT> IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING EXECUTION. IN THIS CASE EXIT FROM THE PROGRAM IS ACCOMPLISHED EITHER BY TYPING A CONTROL/C OR BY OCCURANCE OF AN ERROR WITH THE HALT ON ERROR FLAG BEING SET. THE EXIT IS A RETURN TO COMMAND MODE. SEE EXAMPLE AT END OF 6.3.1.5.

##### 6.3.1.3 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS A SEQUENCE OF ELEMENTS OF THE FORM <FLAG>, <FLAG=1>, OR <FLAG=0>, SEPARATED BY COLONS, WHERE <FLAG> HAS ONE OF THE FOLLOWING VALUES:

HOE HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED

CZDMQA M8207 STATIC DIAG. #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 <sup>J 1</sup> PAGE 7-1  
PROGRAM DOCUMENT

SEQ 0009

3616

LOE LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP

3618  
3619  
3620  
3621  
3622  
3623  
3624  
3625  
3626  
3627  
3628  
3629  
3630  
3631  
3632  
3633  
3634  
3635  
3636  
3637  
3638  
3639  
3640  
3641  
3642  
3643  
3644  
3645  
3646  
3647  
3648  
3649  
3650  
3651  
3652  
3653  
3654  
3655  
3656  
3657  
3658  
3659  
3660  
3661  
3662  
3663  
3664  
3665  
3666  
3667  
3668  
3669  
3670  
3671  
3672

CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR

IER	INHIBIT ERROR REPORTING
IBE	INHIBIT BASIC ERROR REPORTS
IXE	INHIBIT EXTENDED ERROR REPORTS
PRI	DIRECT ALL MESSAGES TO A LINE PRINTER
PNT	PRINT NUMBER OF TEST BEING EXECUTED
BOE	BELL ON ERROR
UAM	RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS
ISR	INHIBIT STATISTICAL REPORTS
IDU	INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
LOT	LOOP ON TEST

THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0 ARE CLEARED. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS SWITCH IS NOT GIVEN ALL FLAGS ARE CLEARED. SEE EXAMPLE AT END OF 6.3.1.5.

#### 6.3.1.4 END OF PASS SWITCH (/EOP:<INCR>)

<INCR> IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF PASSES) IT IS DESIRED THAT THE END OF PASS MESSAGE BE PRINTED. THE DEFAULT IS AT THE END OF EVERY PASS. SEE EXAMPLE AT END OF 6.3.1.5.

#### 6.3.1.5 EFFECT OF START COMMAND

THE EFFECT OF THE START COMMAND IS TO INITIATE THE HARDWARE PARAMETER DIALOGUE, THE SOFTWARE PARAMETER DIALOGUE, AND THEN THE DIAGNOSTIC TESTS THEMSELVES.

THE HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE QUESTION '# UNITS?' TO WHICH THE OPERATOR REPLIES WITH A DECIMAL NUMBER N FROM 1 TO 16. THE TERM 'UNIT' REFERS TO THE DEVICE TO WHICH THIS SERIES OF DIAGNOSTICS IS DEDICATED. FOLLOWING THIS ARE THE QUESTIONS WHEREBY THE P-TABLES THEMSELVES WILL BE BUILT. EACH P-TABLE IS A CORE-RESIDENT TABLE CONTAINING ALL THE HARDWARE INFORMATION FOR ONE UNIT. THE OPERATOR MUST SUPPLY N (NUMBER OF UNITS) VALUES FOR EACH QUESTION. HE MAY DO THIS BY GIVING ONE ANSWER TO EACH QUESTION (IN WHICH CASE THE SERIES OF QUESTIONS WILL BE POSED N TIMES) OR BY GIVING N VALUES, SEPARATED BY COMMAS, TO EACH QUESTION (SERIES WILL BE POSED ONCE). EACH QUESTION IS FOLLOWED BY THE RESPONSE RADIX (D FOR DECIMAL, B FOR BINARY, O FOR OCTAL, L FOR YES/NO) IN PARENTHESES AND THE DEFAULT VALUE AFTER THE PARENTHESES.

3674  
3675  
3676  
3677  
3678  
3679  
3680  
3681  
3682  
3683  
3684  
3685  
3686  
3687  
3688  
3689  
3690  
3691  
3692  
3693  
3694  
3695  
3696  
3697  
3698  
3699  
3700  
3701  
3702  
3703  
3704  
3705  
3706  
3707  
3708  
3709  
3710  
3711  
3712  
3713  
3714  
3715  
3716  
3717  
3718  
3719  
3720  
3721  
3722  
3723  
3724  
3725  
3726  
3727  
3728

FOLLOWING THE HARDWARE QUESTIONS ARE THE SOFTWARE QUESTIONS TO BUILD THE SOFTWARE TABLES, WHICH DEFINE THE MODE (QUICK VERIFY ETC.) THAT THE DIAGNOSTIC WILL EXECUTE IN.

WHEN THE QUESTION '# UNITS?' IS ANSWERED, MEMORY STORAGE IS ALLOCATED FOR THE P-TABLES, AND IF THERE IS NOT ENOUGH TO ACCOMMODATE THEM THE MESSAGE 'TOO MANY UNITS' IS ISSUED. IN THIS CASE THE DIAGNOSTIC MUST BE EXECUTED MORE THAN ONCE TO TEST ALL UNITS.

EXAMPLE:

STA/TESTS:1:2-4:6:8-10/PASS:3/FLAGS:IER:HOE=1:UAM:LOE

THIS COMMAND WILL CAUSE THREE PASSES TO BE MADE, EACH PASS CONSISTING OF TESTS 1,2,3,4,6,8,9, AND 10 EXECUTED AGAINST ALL UNITS. THERE IS NO DIFFERENCE BETWEEN SAYING <FLAG> AND SAYING <FLAG=1>. THE NOTATION <FLAG=0> IS MEANINGFUL ONLY ON A COMMAND OTHER THAN START TO CLEAR A FLAG THAT WAS PREVIOUSLY SET. NOTE THAT ON ALL COMMANDS ONLY THE FIRST THREE LETTERS ARE SCANNED.

### 6.3.2 RESTART COMMAND

\*\*\*\*\*  
RES(TART)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:  
<FLAG-LIST>/UNITS:<UNIT-LIST>  
\*\*\*\*\*

#### 6.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

<TEST-LIST>, <PASS-CNT>, AND <FLAG-LIST> ARE AS IN THE START COMMAND.

#### 6.3.2.2 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (0,1 ETC.) OR RANGES OF DECIMAL NUMBERS (0-5, 8-10 ETC.) THAT SPECIFY THE UNITS TO BE TESTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS MAY RANGE FROM 0 THRU N-1 (N IS THE NUMBER OF UNITS SPECIFIED IN THE PREVIOUS START COMMAND). THE NUMBER INDICATES THE POSITION OF THE P-TABLE AS THE DATA WAS ENTERED DURING THE HARDWARE DIAGLOGUE. THE UNITS WHICH ARE SELECTED MUST NOT HAVE BEEN DROPPED BY THE DROP COMMAND. SEE THE DISCUSSION OF ADD AND DROP COMMANDS BELOW. DEFAULT IS TO TEST ALL UNITS WHICH HAVE NOT BEEN DROPPED BY A DROP COMMAND.

3730  
3731  
3732  
3733  
3734  
3735  
3736  
3737  
3738  
3739  
3740  
3741  
3742  
3743  
3744  
3745  
3746  
3747  
3748  
3749  
3750  
3751  
3752  
3753  
3754  
3755  
3756  
3757  
3758  
3759  
3760  
3761  
3762  
3763  
3764  
3765  
3766  
3767  
3768  
3769  
3770  
3771  
3772  
3773  
3774  
3775  
3776  
3777  
3778  
3779  
3780  
3781  
3782  
3783

### 6.3.2.3 EFFECT OF RESTART COMMAND

THE RESTART COMMAND DIFFERS FROM THE START COMMAND IN THAT THE P-TABLES FROM THE PREVIOUS START COMMAND (THERE MUST HAVE BEEN ONE) ARE USED, INSTEAD OF NEW ONES BEING BUILT. THE UNITS SWITCH GIVES THE ABILITY TO SELECT A SUBSET OF THESE. THE SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED (OPERATOR WILL BE ASKED). THE COMMAND CAN BE USED AFTER COMMAND MODE HAS BEEN REENTERED IN ANY OF THE THREE NORMAL WAYS: A) THE REQUESTED NUMBER OF PASSES HAVE BEEN MADE B) AN ERROR WAS ENCOUNTERED WITH THE HALT ON ERROR FLAG SET C) A CONTROL/C WAS ENTERED BY THE OPERATOR.

### 6.3.3 CONTINUE COMMAND

\*\*\*\*\*  
CON(TINUE)/PASS:<PASS-CNT/FLAGS:<FLAG-LIST>  
\*\*\*\*\*

#### 6.3.3.1 PASS SWITCH (/PASS:<PASS-CNT>)

<PASS-CNT> IS SAME AS IN START COMMAND, BUT THE DEFAULT IS THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART. IF NONE REMAINS, THE DEFAULT IS NON-ENDING EXECUTION.

#### 6.3.3.2 FLAG SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS SAME AS IN START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

#### 6.3.3.3 EFFECT OF CONTINUE COMMAND

CONTINUE MUST FOLLOW A START OR RESTART, AND COMMAND MODE MUST HAVE BEEN ENTERED DUE TO A HALT ON ERROR OR A CONTROL/C. THE EFFECT OF THE COMMAND IS TO GO TO THE BEGINNING OF THE TEST THAT WAS BEING EXECUTED WHEN THE HALT OR CONTROL/C TOOK PLACE. SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.

### 6.3.4 PROCEED COMMAND

\*\*\*\*\*  
PRO(CEED)/FLAGS:<FLAG-LIST>  
\*\*\*\*\*

3785  
3786  
3787  
3788  
3789  
3790  
3791  
3792  
3793  
3794  
3795  
3796  
3797  
3798  
3799  
3800  
3801  
3802  
3803  
3804  
3805  
3806  
3807  
3808  
3809  
3810  
3811  
3812  
3813  
3814  
3815  
3816  
3817  
3818  
3819  
3820  
3821  
3822  
3823  
3824  
3825  
3826  
3827  
3828  
3829  
3830  
3831  
3832  
3833  
3834  
3835  
3836  
3837  
3838  
3839

6.3.4.1 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS AS IN THE START COMMAND, BUT UNSPECIFIED  
FLAGS RETAIN THEIR CURRENT VALUE.

6.3.4.2 EFFECT OF PROCEED COMMAND

PROCEED MUST FOLLOW A START, RESTART, OR CONTINUE. COMMAND  
MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR. THE EFFECT  
OF THE COMMAND IS TO BEGIN EXECUTION AT THE LOCATION  
FOLLOWING THE ERROR CALL. NEITHER HARDWARE NOR SOFTWARE  
PARAMETERS MAY BE ALTERED.

6.3.5 ADD COMMAND

\*\*\*\*\*  
ADD/UNITS:<UNIT-LIST>  
\*\*\*\*\*

6.3.5.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.5.2 EFFECT OF ADD COMMAND

THE UNITS SPECIFIED ARE ADDED TO THE TEST SEQUENCE. EACH  
UNIT MUST HAVE A P-TABLE IN MEMORY DUE TO AN EARLIER  
HARDWARE DIALOGUE. THIS COMMAND MUST BE FOLLOWED BY A  
RESTART OR CONTINUE. THE UNITS SWITCH MUST BE SPECIFIED.  
THE ADD COMMAND IS MEANINGFUL ONLY FOR UNITS THAT WERE  
PREVIOUSLY DROPPED.

6.3.6 DROP COMMAND

\*\*\*\*\*  
DRO(P)/UNITS:<UNIT-LIST>  
\*\*\*\*\*

6.3.6.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.6.2 EFFECT OF DROP COMMAND

3841  
3842  
3843  
3844  
3845  
3846  
3847  
3848  
3849  
3850  
3851  
3852  
3853  
3854  
3855  
3856  
3857  
3858  
3859  
3860  
3861  
3862  
3863  
3864  
3865  
3866  
3867  
3868  
3869  
3870  
3871  
3872  
3873  
3874  
3875  
3876  
3877  
3878  
3879  
3880  
3881  
3882  
3883  
3884  
3885  
3886  
3887  
3888  
3889  
3890  
3891  
3892  
3893  
3894  
3895

THE UNITS SPECIFIED WILL BE DROPPED FROM TESTING. THE UNITS WILL BE RESELECTED ONLY BY THE EXECUTION OF AN ADD OR START COMMAND. THE UNITS SWITCH MUST BE ENTERED. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR A CONTINUE COMMAND.

### 6.3.7 PRINT COMMAND

\*\*\*\*\*  
PRI(NT)  
\*\*\*\*\*

#### 6.3.7.1 EFFECT OF PRINT COMMAND

THE TOTAL NUMBER OF ERRORS FOR EACH UNIT SINCE THE LAST START OR RESTART COMMAND ARE PRINTED. THE ISR (INHIBIT STATISTICAL REPORTING) FLAG IS CLEARED.

### 6.3.8 DISPLAY COMMAND

\*\*\*\*\*  
DIS(PLAY)/UNITS:<UNIT-LIST>  
\*\*\*\*\*

#### 6.3.8.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

#### 6.3.8.2 EFFECT OF DISPLAY COMMAND

THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED OUT IN THE FORMAT IN WHICH THEY WERE ENTERED. ANY UNITS THAT WERE DROPPED BY THE OPERATOR 'DROP' COMMAND ARE SO DESIGNATED.

### 6.3.9 FLAGS COMMAND

\*\*\*\*\*  
FLA(GS)  
\*\*\*\*\*

#### 6.3.9.1 EFFECT OF FLAGS COMMAND

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.



3897  
3898  
3899  
3900  
3901  
3902  
3903  
3904  
3905  
3906  
3907  
3908  
3909  
3910  
3911  
3912  
3913  
3914  
3915  
3916  
3917  
3918  
3919  
3920  
3921  
3922  
3923  
3924  
3925  
3926  
3927  
3928  
3929  
3930  
3931  
3932  
3933  
3934  
3935  
3936  
3937  
3938  
3939  
3940  
3941  
3942  
3943  
3944  
3945  
3946

### 6.3.10 ZFLAGS COMMAND

\*\*\*\*\*  
ZFL (AGS)  
\*\*\*\*\*

#### 6.3.10.1 EFFECT OF ZFLAGS COMMAND

ALL FLAGS ARE CLEARED.

#### 6.3.11 CONTROL CHARACTERS

A CONTROL C (C) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES A RETURN TO COMMAND MODE.

A CONTROL Z (Z) ENTERED DURING ONE OF THE THREE OPERATOR DIALOGUES- INITIAL DIALOGUE (SEE 6.2), HARDWARE DIALOGUE (SEE 6.3.1.5), OR SOFTWARE DIALOGUE (SEE 6.3.1.5) CAUSES THE DEFAULTS TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.

A CONTROL O (O) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES ALL TELETYPE OUTPUT TO BE SURPRESSED FOR THE REMAINDER OF THE DIAGNOSTIC OR UNTIL ANOTHER O IS TYPED, WHICH RESTORES NORMAL TELETYPE OUTPUT.

#### 6.3.12 HARDWARE PARAMETERS

THE FOLLOWING QUESTIONS WILL BE ASKED ON A START COMMAND. THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE.

1. WHICH MICRO-CPU? (0= M8200, 4= M8204, 7= M8207) (0) 7?

2. MICRO-CPU CSR ADDRESS: (0) 160170?

THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SELO) RESIDE ON THE UNIBUS. THE ALLOWABLE RANGE IS 160000-177776 (OCTAL), AND THE DEFAULT IS 160170.

3948  
3949  
3950  
3951  
3952  
3953  
3954  
3955  
3956  
3957  
3958  
3959  
3960  
3961  
3962  
3963  
3964  
3965  
3966  
3967  
3968  
3969  
3970  
3971  
3972  
3973  
3974  
3975  
3976  
3977  
3978  
3979  
3980  
3981  
3982  
3983  
3984  
3985  
3986  
3987  
3988  
3989  
3990  
3991  
3992  
3993  
3994  
3995  
3996  
3997  
3998  
3999  
4000  
4001  
4002  
4003

3. MICRO-PROCESSOR RUN SWITCH-TYPE 1 IF ON, IF OFF: (O) O?

THE RUN SWITCH IS E28, SWITCH 7 ON THE M8207. MORE TESTS CAN BE PERFORMED IF THE RUN SWITCH IS OFF. YOU MAY GENERATE AN ERROR IF YOU ANSWER THIS QUESTION WRONG.

### 6.3.13 SOFTWARE PARAMETERS

NO SOFTWARE PARAMETER QUESTIONS ARE ASKED BY PART 2 OF THE STATIC LOGIC TESTS.

### 6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

THE FULL CAPABILITY OF THE HARDWARE DIALOGUE IS REVEALED BY THE FOLLOWING DISCUSSION OF WHAT HAPPENS INTERNALLY.

AS SOON AS THE QUESTION '# UNITS?' IS ANSWERED (WITH THE NUMBER N, SAY) SPACE IN CORE IS ALLOCATED FOR N P-TABLES. ALL OF THE P-TABLES ARE OF THE SAME FORMAT, AND THERE IS A ONE-TO ONE CORRESPONDENCE BETWEEN THE HARDWARE PARAMETER QUESTIONS AND THE SLOTS IN THE P-TABLE FORMAT.

ON THE FIRST TRIP THRU THE QUESTIONS, ALL OF THE SLOTS IN ALL OF THE P-TABLES ARE FILLED. IF THE OPERATOR TYPES IN LESS THAN N EXPLICIT VALUES IN RESPONSE TO A PARTICULAR QUESTION, THESE VALUES ARE PLACED IN THE P-TABLES (ONE VALUE GOING INTO THE PROPER SLOT OF EACH P-TABLE BEGINNING WITH THE FIRST P-TABLE) UNTIL THE STRING OF VALUES IS EXHAUSTED. THE LAST VALUE IN THE STRING BECOMES THE NEW DEFAULT AND IS USED TO FILL THAT SLOT IN THE REMAINING P-TABLES.

ON SUBSEQUENT TRIPS THRU THE QUESTIONS, THE SAME PROCESS IS CARRIED OUT, EXCEPT THAT THE EARLIEST P-TABLE NOT TO HAVE RECEIVED AN EXPLICIT VALUE IN ANY OF ITS SLOTS NOW ASSUMES THE ROLE THAT TABLE NUMBER ONE PLAYED IN THE FIRST TRIP.

THE SERIES OF QUESTIONS IS REISSUED UNTIL AT LEAST ONE QUESTION HAS RECEIVED N EXPLICIT VALUES FROM THE OPERATOR.

IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING VALUES MAY BE USED TO INDICATE A REPETITION OF THE LAST NAMED VALUE.

A STRING OF VALUES MAY BE GIVEN AS A RANGE (6-10 FOR EXAMPLE). IF THE VALUES REPRESENT PURE NUMERICAL DATA, THIS SAMPLE RANGE TRANSLATES TO THE STRING 6,7,8,9,10 (AN INCREMENT OF 1). IF THE VALUES ARE ADDRESSES, THE SAMPLE RANGE TRANSLATES TO THE STRING 6,8,10 (AN INCREMENT OF 2).

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

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO CONSTRUCT A SET OF P-TABLES. ASSUME THAT WE HAVE 16 UNITS, AND THAT THERE ARE THREE HARDWARE PARAMETERS FOR EACH (THREE SLOTS IN THE P-TABLE, THREE HARDWARE QUESTIONS IN THE DIALOGUE). LET THE DESIRED VALUE FOR THE FIRST PARAMETER BE THE NUMBER 75 FOR ALL 16 TABLES. LET THE DESIRED VALUE FOR THE SECOND PARAMETER BE EQUAL TO THE UNIT NUMBER (0,1,2,...,15) EXCEPT FOR UNIT 12, WHICH SHOULD RECEIVE THE VALUE 11. LET THE DESIRED VALUE FOR THE THIRD PARAMETER BE THE NUMBER 76 FOR THE FIRST 7 UNITS AND THE NUMBER 77 FOR THE LAST 9 UNITS.

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

# UNITS (D) ? 16

UNIT 1  
<QUESTION 1> ? 75  
<QUESTION 2> ? 0-6  
<QUESTION 3> ? 76

UNIT 21  
<QUESTION 1> ?  
<QUESTION 2> ? 7-11,,13-15  
<QUESTION 3> ? 77

THE FIRST TIME THE SERIES IS ASKED, SLOT ONE RECEIVES A 75 IN ALL 16 TABLES. SLOT TWO RECEIVES THE VALUES 0,1,2,...,6 IN TABLES 0 THRU 6 AND A CONSTANT 6 IN TABLES 7 THRU 15. SLOT THREE RECEIVES A CONSTANT 76 IN ALL 16 TABLES.

THE SECOND TIME THRU THE SERIES, TABLES 16 THRU THE END ARE GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS PRINTED OUT FOR THE THE OPERATOR IN THE FORM 'UNIT XX' AT THE BEGINNING OF EACH SERIES). QUESTION 1 IS RESPONDED TO BY A <CR>, SO SLOT ONE STAYS AT CONSTANT 75 IN TABLES 7 THRU 15, SINCE NO NEW EXPLICIT VALUES ARE TYPED IN. SLOT TWO GETS THE VALUES 7,8,9,10,11 IN TABLES 7 THRU 11, AND GETS A 11 IN SLOT 12, AND GETS THE VALUES 13,14,15 IN TABLES 13 THRU 15. SLOT THREE GETS THE VALUE 77 IN TABLES 7 THRU 15.

THE DIALOGUE IS TERMINATED WHEN THE SOFTWARE RECOGNIZES THAT 16 EXPLICIT VALUES HAVE BEEN GIVEN FOR AT LEAST ONE QUESTION (NAMELY QUESTION 2).

4055  
4056  
4057  
4058  
4059  
4060  
4061  
4062  
4063  
4064  
4065  
4066  
4067  
4068  
4069  
4070  
4071  
4072  
4073  
4074  
4075  
4076  
4077  
4078  
4079  
4080  
4081  
4082  
4083  
4084  
4085  
4086  
4087  
4088  
4089  
4090  
4091  
4092  
4093  
4094  
4095  
4096  
4097  
4098  
4099  
4100  
4101  
4102  
4103  
4104  
4105  
4106  
4107  
4108  
4109  
4110

7.0 TEST DESCRIPTIONS

\*\*\*\*\* TEST 1 \*\*\*\*\*  
\*VERIFY THAT REFERENCING UNIBUS DEVICE REGISTERS  
\*DOES NOT CAUSE A TIME OUT TRAP  
\*\*\*\*\*

\*\*\*\*\* TEST 2 \*\*\*\*\*  
\*TEST OF BR RIGHT SHIFT  
\*VERIFY THAT A DEST OF BR RSH (011) OF A MICRO-INSTRUCTION  
\*SHIFTS THE RESULTING BR DATA RIGHT ONCE.  
\*\*\*\*\*

\*\*\*\*\* TEST 3 \*\*\*\*\*  
\*IOP CRAM WRITE/READ TEST  
\*FLOAT A 1 THROUGH EACH CRAM LOCATION  
\*\*\*\*\*

\*\*\*\*\* TEST 4 \*\*\*\*\*  
\*IOP CRAM WRITE/READ TEST  
\*FLOAT A 0 THROUGH EACH CRAM LOCATION  
\*\*\*\*\*

\*\*\*\*\* TEST 5 \*\*\*\*\*  
\*IOP CRAM DUAL ADDRESSING TEST  
\*WRITE EACH ADDRESS INTO ITSELF, READ EACH  
\*ADDRESS TO VERIFY CORRECT ADDRESSING  
\*\*\*\*\*

\*\*\*\*\* TEST 6 \*\*\*\*\*  
\*IOP MAIN MEMORY TEST  
\*FLOAT A 1 THROUGH ALL MAIN MEMORY LOCATIONS  
\*\*\*\*\*

\*\*\*\*\* TEST 7 \*\*\*\*\*  
\*IOP MAIN MEMORY TEST  
\*FLOAT A 0 THROUGH ALL MAIN MEMORY LOCATIONS  
\*\*\*\*\*

\*\*\*\*\* TEST 8 \*\*\*\*\*  
\*IOP MAIN MEMORY DUAL ADDRESSING TEST  
\*LOAD EACH MEMORY LOCATION WITH ITS OWN ADDRESS  
\*READ BACK EACH LOCATION TO VERIFY CORRECT ADDRESSING

CZDMQA MB207 STATIC DIAG. #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 <sup>G 2</sup> PAGE 16-1  
PROGRAM DOCUMENT

SEQ 0019

4111

\*\*\*\*\*

4113  
4114  
4115  
4116  
4117  
4118  
4119  
4120  
4121  
4122  
4123  
4124  
4125  
4126  
4127  
4128  
4129  
4130  
4131  
4132  
4133  
4134  
4135  
4136  
4137  
4138  
4139  
4140  
4141  
4142  
4143  
4144  
4145  
4146  
4147  
4148  
4149  
4150  
4151  
4152  
4153  
4154  
4155  
4156  
4157  
4158  
4159  
4160  
4161  
4162  
4163  
4164  
4165  
4166  
4167  
4168

\*\*\*\*\* TEST 9 \*\*\*\*\*  
\*IOP MAR TEST  
\*PERFORM DUAL ADDRESSING TEST  
\*USING MAR AUTO-INC FEATURE  
\*\*\*\*\*

\*\*\*\*\* TEST 10 \*\*\*\*\*  
\*IOP (CRAM) ODT BITS TEST  
\*LOAD MAR WITH A 0 INC MAR UNTIL IT OVERFLOWS  
\*VERIFY THAT IBUS\* 10 BITS IS SET ONLY WHEN MAR BIT 8 IS A ONE  
\*AND THAT IBUS\* 10 BIT6 IS SET ON MAR OVERFLOW  
\*\*\*\*\*

\*\*\*\*\* TEST 11 \*\*\*\*\*  
\*CRAM TEST OF JUMP(I) NEVER MICRO-PROCESSOR INSTRUCTION.  
\*PERFORM THE JUMP INSTRUCTION  
\*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION  
\*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE  
\*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT  
\*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT  
\*THE CRAM PC IS CORRECT. IF THE CRAM PC IN NOT RIGHT,  
\*THEN PORT4 CONTAINS A 37  
\*\*\*\*\*

\*\*\*\*\* TEST 12 \*\*\*\*\*  
\*CRAM TEST OF JUMP(I) ALWAYS MICRO-PROCESSOR INSTRUCTION.  
\*PERFORM THE JUMP INSTRUCTION  
\*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION  
\*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE  
\*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT  
\*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT  
\*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL  
\*THEN PORT4 WILL CONTAIN A 37  
\*\*\*\*\*

\*\*\*\*\* TEST 13 \*\*\*\*\*  
\*CRAM TEST OF JUMP(I) ON C BIT SET MICRO-PROCESSOR INSTRUCTION.  
\*SET THE C BIT, PERFORM THE JUMP INSTRUCTION.  
\*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION  
\*IN THE LOCATION IT IS AT. THIS INSTRUCITON LOADS THE  
\*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT  
\*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT  
\*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL  
\*THEN PORT4 WILL CONTAIN A 37.  
\*\*\*\*\*

4169

4171  
4172  
4173  
4174  
4175  
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  
4201  
4202  
4203  
4204  
4205  
4206  
4207  
4208  
4209  
4210  
4211  
4212  
4213  
4214  
4215  
4216  
4217  
4218  
4219  
4220  
4221  
4222  
4223  
4224  
4225  
4226

\*\*\*\*\* TEST 14 \*\*\*\*\*  
\*CRAM TEST OF JUMP(I) ON Z BIT SET MICRO-PROCESSOR INSTRUCTION.  
\*SET THE Z BIT, PERFORM THE JUMP INSTRUCTION.  
\*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION  
\*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE  
\*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT  
\*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT  
\*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL  
\*THEN PORT4 WILL CONTAIN A 37  
\*\*\*\*\*

\*\*\*\*\* TEST 15 \*\*\*\*\*  
\*CRAM TEST OF JUMP(I) ON BRO SET MICRO-PROCESSOR INSTRUCTION.  
\*SET THE BRO BIT, PERFORM THE JUMP INSTRUCTION.  
\*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION  
\*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE  
\*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT  
\*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT  
\*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL  
\*THEN THE PORT4 WILL CONTAIN A 37  
\*\*\*\*\*

\*\*\*\*\* TEST 16 \*\*\*\*\*  
\*CRAM TEST OF JUMP(I) ON BR1 SET MICRO-PROCESSOR INSTRUCTION.  
\*SET THE BR1 BIT, PERFORM THE JUMP INSTRUCTION.  
\*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION  
\*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE  
\*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT  
\*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT  
\*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL  
\*THEN PORT4 WILL CONTAIN A 37  
\*\*\*\*\*

\*\*\*\*\* TEST 17 \*\*\*\*\*  
\*CRAM TEST OF JUMP(I) ON BR4 SET MICRO-PROCESSOR INSTRUCTION.  
\*SET THE BR4 BIT, PERFORM THE JUMP INSTRUCTION.  
\*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION  
\*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE  
\*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT  
\*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT  
\*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL  
\*THEN PORT4 WILL CONTAIN A 37  
\*\*\*\*\*

\*\*\*\*\* TEST 18 \*\*\*\*\*  
\*CRAM TEST OF JUMP(I) ON BR7 SET MICRO-PROCESSOR INSTRUCTION.



CZDMA M8207 STATIC DIAG. #2  
CZDMA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 <sup>K 2</sup> PAGE 18-1  
PROGRAM DOCUMENT

SEQ 0023

4227

\*SET THE BR7 BIT, PERFORM THE JUMP INSTRUCTION

4229  
4230  
4231  
4232  
4233  
4234  
4235  
4236  
4237  
4238  
4239  
4240  
4241  
4242  
4243  
4244  
4245  
4246  
4247  
4248  
4249  
4250  
4251  
4252  
4253  
4254  
4255  
4256  
4257  
4258  
4259  
4260  
4261  
4262  
4263  
4264  
4265  
4266  
4267  
4268  
4269  
4270  
4271  
4272  
4273  
4274  
4275  
4276  
4277  
4278  
4279  
4280  
4281  
4282  
4283  
4284

\*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION  
\*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE  
\*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT  
\*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT  
\*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL  
\*THEN PORT4 WILL CONTAIN A 37  
\*\*\*\*\*

\*\*\*\*\* TEST 19 \*\*\*\*\*  
\*CRAM TEST OF JUMP(I) ON C BIT CLEAR MICRO-PROCESSOR INSTRUCTION.  
\*SET THE C BIT, PERFORM THE JUMP INSTRUCTION.  
\*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION  
\*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE  
\*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT  
\*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT  
\*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL  
\*THEN PORT4 WILL CONTAIN A 37  
\*\*\*\*\*

\*\*\*\*\* TEST 20 \*\*\*\*\*  
\*CRAM TEST OF JUMP(I) ON Z BIT CLEAR MICRO-PROCESSOR INSTRUCTION.  
\*CLEAR THE Z BIT, PERFORM THE JUMP INSTRUCTION.  
\*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION  
\*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE  
\*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT  
\*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT  
\*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL  
\*THEN PORT4 WILL CONTAIN A 37  
\*\*\*\*\*

\*\*\*\*\* TEST 21 \*\*\*\*\*  
\*CRAM TEST OF JUMP(I) ON BRO CLEAR MICRO-PROCESSOR INSTRUCTION.  
\*CLEAR THE BRO BIT, PERFORM THE JUMP INSTRUCTION.  
\*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION  
\*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE  
\*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT  
\*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT  
\*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL  
\*THEN PORT4 WILL CONTAIN A 37  
\*\*\*\*\*

\*\*\*\*\* TEST 22 \*\*\*\*\*  
\*CRAM TEST OF JUMP(I) ON BR1 CLEAR MICRO-PROCESSOR INSTRUCTION.  
\*CLEAR THE BR1 BIT, PERFORM THE JUMP INSTRUCTION.  
\*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION  
\*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE  
\*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT

CZDMQA M8207 STATIC DIAG. #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 M 2  
PROGRAM DOCUMENT PAGE 19-1

SEQ 0025

4285

\*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT

4287  
4288  
4289  
4290  
4291  
4292  
4293  
4294  
4295  
4296  
4297  
4298  
4299  
4300  
4301  
4302  
4303  
4304  
4305  
4306  
4307  
4308  
4309  
4310  
4311  
4312  
4313  
4314  
4315  
4316  
4317  
4318  
4319  
4320  
4321  
4322  
4323  
4324  
4325  
4326  
4327  
4328  
4329  
4330  
4331  
4332  
4333  
4334  
4335  
4336  
4337  
4338  
4339  
4340  
4341  
4342

\*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL  
\*THEN PORT4 WILL CONTAIN A 37

\*\*\*\*\*

\*\*\*\*\* TEST 23 \*\*\*\*\*

\*CRAM TEST OF JUMP(I) ON BR4 CLEAR MICRO-PROCESSOR INSTRUCTION.  
\*CLEAR THE BR4 BIT, PERFORM THE JUMP INSTRUCTION.  
\*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION  
\*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE  
\*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT  
\*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT  
\*THE CRAM PC IS CORRECT, IF THE CRAM PC IS NOT RIGHT  
\*THEN PORT4 CONTAINS A 37

\*\*\*\*\*

\*\*\*\*\* TEST 24 \*\*\*\*\*

\*CRAM TEST OF JUMP(I) ON BR7 CLEAR MICRO-PROCESSOR INSTRUCTION.  
\*CLEAR THE BR7 BIT, PERFORM THE JUMP INSTRUCTION.  
\*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION  
\*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE  
\*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT  
\*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT  
\*THE CRAM PC IS CORRECT, IF THE CRAM PC IS NOT RIGHT  
\*THEN PORT4 CONTAINS A 37

\*\*\*\*\*

\*\*\*\*\* TEST 25 \*\*\*\*\*

\*  
\*MAIN MEMORY PAGE DUAL ADDRESS TEST.  
\*IN THIS TEST WE WILL VERIFY THAT PAGES DO  
\*NOT DUAL ADDRESS. THIS TEST IS DIFFERENT FROM THE  
\*PREVIOUS DUAL ADDRESS TESTS IN THAT THE OTHER  
\*TEST REALLY DIDN'T CHECK PAGE DUAL ADDRESSING

\*\*\*\*\*

\*\*\*\*\* TEST 26 \*\*\*\*\*

\*  
\*JUMP FIELD,PAGE TEST  
\*  
\*IN THIS TEST WE WILL MAKE SURE A JUMP FIELD INSTRUCTION  
\*WORKS. TO DO THIS, WE'LL PUT THE DESIRED PAGE, FIELD  
\*INFORMATION IN IBUS\*<13> THEN ISSUE A JUMP FIELD  
\*THEN WE'LL READ PC REG. AND VERIFY.

\*\*\*\*\*

CZDMQA M8207 STATIC DIAG. #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 <sup>B 3</sup> PAGE 20-1  
PROGRAM DOCUMENT

SEQ 0027

4343

\*\*\*\*\* TEST 27 \*\*\*\*\*

4345  
4346  
4347  
4348  
4349  
4350  
4351  
4352  
4353  
4354  
4355  
4356  
4357  
4358  
4359  
4360  
4361  
4362  
4363  
4364  
4365  
4366  
4367  
4368  
4369  
4370  
4371  
4372  
4373  
4374  
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

\*  
\*JUMP TEST, JUMP ALWAYS, JUMP CHANGE FIELD  
\*  
\*IN THIS TEST, WE WILL CHECK THE ABILITY OF THE  
\*MICRO-PROCESSOR TO JUMP (BRANCH AND ALWAYS INSTRUCTION)  
\*TO LOCATIONS, FIELDS FROM OTHER LOCATIONS FIELDS.  
\*WE ALREADY KNOW THAT THE BRANCH INSTR WORKS FROM  
\*OTHER TEST. PROCEDURE:  
\* 1. START ADDR 0, FIELD 0  
\* 2. \*\*CALCULATE NEW ADDR, FIELD VIA INC,  
\* 3. CAUSE JUMP (BRANCH) TO NEW ADDRESS  
\* 4. READ PC FROM IBUS\*12 AND IBUS\*13  
\* 5. REPEAT STEP 2-4 256.TIMES  
\*  
\* TO CALCULATE NEW ADDRESS:  
\* 1. INC LOW BYTE OF ADDRESS FOR PC ADDRESS 0-7  
\* 2. INC LOW BYTE OF N ADDRESS FOR PC ADDRESS 8-11  
\* BITS REPRESENTED AS BITS 0-3. WHEN 0-3 OVERFLOWS,  
\* RESTARTS AT ZERO.  
\* NET RESULT IS JUMPS FROM:  
\* FIELD,PAGE LOC  
\* 0 0  
\* 1 1  
\* 2 2  
\* 3 3  
\* 10 7  
\* 11 11  
\* :TO :  
\* 17 377

\*\*\*\*\*

\*\*\*\*\* TEST 28 \*\*\*\*\*

\*  
\*JUMP TEST, JUMP ALWAYS, JUMP CHANGE FIELD  
\*  
\*IN THIS TEST, WE WILL CHECK THE ABILITY OF THE  
\*MICRO-PROCESSOR TO JUMP (BRANCH AND ALWAYS INSTRUCTION)  
\*TO LOCATIONS, FIELDS FROM OTHER LOCATIONS FIELDS.  
\*WE ALREADY KNOW THAT THE BRANCH INSTR WORKS FROM  
\*OTHER TESTS. PROCEDURE:  
\* 1. START ADDR 0, FIELD 0  
\* 2. \*\*CALCULATE NEW ADDR, FIELD VIA DEC,  
\* 3. CAUSE JUMP (BRANCH) TO NEW ADDRESS  
\* 4. READ PC FROM IBUS\*12 AND IBUS\*13  
\* 5. REPEAT STEP 2-4 256.TIMES  
\*  
\* TO CALCULATE NEW ADDRESS:  
\* 1. DEC LOW BYTE OF ADDRESS FOR PC ADDRESS 0-7  
\* 2. DEC LOW BYTE OF N ADDRESS FOR PC ADDRESS 8-11  
\* BITS REPRESENTED AS BITS 0-3. WHEN 0-3 OVERFLOWS,

CZDMQA M8207 STATIC DIAG. #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 <sup>D 3</sup> PAGE 21-1  
PROGRAM DOCUMENT

SEQ 0029

4401

\*                   RESTARTS AT ZERO

4403  
4404  
4405  
4406  
4407  
4408  
4409  
4410  
4411  
4412  
4413  
4414  
4415  
4416  
4417  
4418  
4419  
4420  
4421  
4422  
4423  
4424  
4425  
4426  
4427  
4428  
4429  
4430  
4431  
4432  
4433  
4434  
4435  
4436  
4437  
4438  
4439  
4440  
4441  
4442  
4443  
4444  
4445  
4446  
4447  
4448  
4449  
4450  
4451  
4452  
4453  
4454  
4455  
4456  
4457  
4458

```
*          NET RESULT IS JUMPS FROM:
*          FIELD,PAGE          LOC:
*              0              0
*              17             377
*              16             376
*              15             375
*              :TO            :
*              00             000
*****
```

\*\*\*\*\* TEST 29 \*\*\*\*\*

```
*
*IN THIS TEST WE'LL VERIFY THAT THE Z BIT CAN BE READ FROM
*IBUS* <13>. WE ALREADY KNOW THAT THE Z BIT WORKS PROPERLY,
*ALL WE WANT TO KNOW HERE IS THAT IT CAN BE READ.
*****
```

\*\*\*\*\* TEST 30 \*\*\*\*\*

```
*
*IN THIS TEST WE'LL VERIFY THAT THE C BIT CAN BE READ FROM
*IBUS* <13>. WE ALREADY KNOW THAT THE C BIT WORKS PROPERLY
*ALL WE WANT TO KNOW HERE IS THAT IT BE READ.
*****
```

\*\*\*\*\* TEST 31 \*\*\*\*\*

```
*TEST OF PROGRAM CLOCK BIT
*DO A MASTER CLEAR, VERIFY THAT PROGRAM CLOCK IS SET
*WRITE PROGRAM CLOCK BIT TO A ONE, VERIFY THAT IT CLEARS,
*AND THEN SETS SOME TIME LATER
*****
```

\*\*\*\*\* TEST 32 \*\*\*\*\*

```
*FORCE POWER FAIL TEST
*SET FORCE POWER FAIL BIT VERIFY THAT PROCESSOR TRAPS TO 24
*GOING DOWN AND COMING UP. VERIFY ALSO THAT BUS INIT WAS
*BLOCKED FROM GETTING TO THE M8200,4,7 DURING THE POWER FAIL
*****
```

\*\*\*\*\* TEST 33 \*\*\*\*\*

```
*MICRO-PROCESSOR NOISE TEST
*WRITE ALL ZERO'S THEN ALL ONE'S THEN A DATA PATTERN
*TO THE IBUS* AND IBUS REGISTERS AND TO THE SP AND MAIN MEM
*THEN GO BACK AND READ THE DATA PATTERNS TO VERIFY THAT
*READING AND WRITING OF OTHER LOCATIONS AND REGISTERS
```



CZDMQA M8207 STATIC DIAG. #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 F 3 PAGE 22-1  
PROGRAM DOCUMENT

SEQ 0031

4459

\*DID NOT CHANGE THE DATA.

4461  
4462  
4463  
4464  
4465  
4466  
4467  
4468  
4469  
4470  
4471  
4472  
4473  
4474  
4475  
4476  
4477  
4478  
4479  
4480  
4481  
4482  
4483  
4484  
4485  
4486  
4487  
4488  
4489  
4490  
4491  
4492  
4493  
4494  
4495  
4496  
4497  
4498  
4499  
4500  
4501  
4502  
4503  
4504  
4505  
4506  
4507  
4508  
4509  
4510  
4511  
4512  
4513  
4514  
4515  
4516

\*\*\*\*\*

\*\*\*\*\* TEST 34 \*\*\*\*\*  
\*THIS TEST IS DESIGNED TO MAKE SURE THAT A NODST INSTRUCTION  
\*DOES NOT WRITE INTO PORT B OF THE MULTIPOINT RAM.  
\*TO DO THIS, WE'LL PUT A 125 INTO INDAT2, THEN WE'LL PUT A  
\*125 INTO BOTH SP1 AND BR. LAST WE'LL DO A NODST BR, SUBOC, SP1  
\*IF THERE IS A WRITE INTO PORTB, INADT2 WILL CONTAIN A 377  
\*\*\*\*\*

\*\*\*\*\* TEST 35 \*\*\*\*\*  
\*  
\*EXTENDED CRAM TEST FOR M8206. IN THIS TEST WE WILL LOAD DATA  
\*THROUGHOUT THE CRAM (TEST DATA IS JUST 4K OF DIAG. CODE) AND  
\*THEN READ IT BACK AND VERIFY THAT IT IS CORRECT  
\*\*\*\*\*

\*\*\*\*\* TEST 36 \*\*\*\*\*  
\*  
\*THIS TEST LOADS MICRO-CODE INTO A M8206 MCPU THEN EXECUTES IT.  
\*THE MICRO-CODE IS DESIGNED TO WRITE ALL ONES INTO THE SEL REGS.  
\*  
\*\*\*\*\*

\*\*\*\*\* TEST 37 \*\*\*\*\*  
\*  
\*NEGATIVE ADDRESS TEST.  
\* IN THIS TEST, WE'LL MAKE SURE THAT THE M8207  
\* DOES NOT RESPOND TO AN ADDRESS THAT ISN'T ASSIGNED  
\* TO IT  
\*  
\*\*\*\*\*

\*\*\*\*\* TEST 38 \*\*\*\*\*  
\*  
\*BYTE ADDRESSING TEST  
\* HERE, WE'RE GOING TO MAKE SURE THAT WE CAN  
\* WRITE INTO ONLY A HIGH OR LOW BYTE OF THE MCPU.  
\*  
\*\*\*\*\*

\*\*\*\*\* TEST 39 \*\*\*\*\*  
\*  
\*IN THIS TEST WE'RE GOING TO MAKE SURE THAT THE PC  
\*REG COUNTS UP PROPERLY. THE PC REG SHOULD INCREMENT

CZDMQA M8207 STATIC DIAG. #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 H 3  
PAGE 23-1  
PROGRAM DOCUMENT

SEQ 0033

4517

\*ONCE AFTER EACH INSTRUCTION.

4519  
4520  
4521  
4522  
4523  
4524  
4525  
4526  
4527  
4528  
4529  
4530  
4531  
4532  
4533  
4534  
4535  
4536  
4537  
4538  
4539  
4540  
4541  
4542  
4543  
4544  
4545  
4546  
4547  
4548  
4549  
4550  
4551  
4552  
4553  
4554  
4555  
4556  
4557  
4558  
4559  
4560  
4561  
4562  
4563  
4564  
4565  
4566  
4567  
4568  
4569  
4570  
4571

\*  
\*\*\*\*\*

\*\*\*\*\* TEST 40 \*\*\*\*\*

\*  
\*IN THIS TEST WE'LL MAKE SURE THAT 'BRANCH FIELD H' DOESN'T  
\*GET STUCK HIGH.  
\*FIRST WE'LL CLEAR THE PC HIGH REG. THEN WE'LL DO A BRANCH INSTR  
\*WITH BAB BITS 11+12 SET. IF PCR BITS 8+9 SET THEN WE'LL KNOW  
\*WE WERE SUCCESSFUL IF PCR BITS 8+9 FAIL TO SET, WE'LL KNOW  
\*THAT THE MAX SELECTED THE WRONG INPUT TO BE CLOCKED INTO THE PCR.  
\*\*\*\*\*

\*\*\*\*\* TEST 41 \*\*\*\*\*

\*  
\*IN THIS TEST WE'RE GOING TO MAKE SURE THAT ONLY SPO  
\*IS SELECTED FOR SOURCE WHEN THE DESTINATION  
\*IS THE OUTBUS  
\*FIRST WE'LL WRITE EACH SP ADDRS INTO ITSELF THEN WE'LL  
\*MOV SP TO OBUS4. THAT SHOULD SELECT  
\*SP ADDRESS 0. IF ANY OTHER DATA SHOWS UP, WE'LL  
\*BLAME IT ON THE SELECTION OF A DIFFERENT SCRATCH PAD.  
\*\*\*\*\*

\*\*\*\*\* TEST 42 \*\*\*\*\*

\*  
\*IN THIS TEST WE ARE GOING TO MAKE SURE THAT THE  
\*SIGNAL 'MOV INST H' (AND ITS ASSOC. TRIBS) DOESN'T GET  
\*STUCK HIGH. IN ORDER TO DO THIS WE'LL CLEAR THE PC HIGH REG  
\*PUT KNOWN DATA IN THE BREG AND SP1 THEN WE'LL BRANCH  
\*WITH CROM BITS 0-3 SET AS WELL AS CROM BIT 9 WITH CROM BITS 8 AND 11 CLEAR.  
\*IF 'MOV INST H' GETS STUCK HIGH, THE PC REG HIGH WILL GET LOADED  
\*WITH THE CONTENTS OF THE ALU  
\*\*\*\*\*

\*\*\*\*\* TEST 43 \*\*\*\*\*

\*TEST TAHT MASTER CLEAR, CLEARS BITS IN THE NPR CONTROL REGISTER AND  
\*MICROPROCESSOR MISCELLANEOUS REGISTER-FIRST WE'LL SET THE  
\*PRIORITY UP SO THAT WHEN WE SET THE BUS REQUEST BIT THAT IT WON'T BUG US  
\*THEN WE'LL SET ALL THE BITS IN BOTH REGS EXCEPT THE  
\*NPR REQUEST. WE'LL LOOK TO SEE THAT ALL GOT SET, NEXT  
\*WE'LL DO A MASTER CLEAR AND BE SURE THAT THEY ALL CLEAR.  
\*\*\*\*\*

4573  
4574  
4575  
4576  
4577  
4578  
4579  
4580  
4581  
4582  
4583  
4584  
4585  
4586  
4587  
4588  
4589  
4590  
4591  
4592  
4593  
4594  
4595  
4596  
4597  
4598  
4599  
4600  
4601  
4602  
4603  
4604  
4605  
4606  
4607  
4608  
4609  
4610  
4611

8.0 ERROR INFORMATION

8.1 ERROR REPORTING

ERRORS ARE REPORTED BY THE PROGRAM AS THEY OCCUR (IF NOT INHIBITED). THE REPORT CONFORMS TO THE DIAGNOSTIC SUPERVISOR ERROR REPORT FORMAT, AND CONSISTS OF A DESCRIPTION OF THE ERROR, THE TEST NUMBER, SUBTEST NUMBER, PC OF THE ERROR CALL, DEVICE ADDRESS, AND BASIC AND EXTENDED ERROR INFORMATION.

THE FOLLOWING EXAMPLES PROVIDE TYPICAL ERROR REPORTS:

CZDMQ DVC FTL ERR 00045 TST 027 SUB 000 PC:022572  
MASTER CLEAR FAILED TO CLEAR PC REG, CONTENTS=000624  
CZDMQ DVC FTL ERR 00015 TST 042 SUB 000 PC:027234  
UNIT=00, FAILING UNIT ADDRESS=160170  
JUMP TEST ERROR  
FROM ADDR TO ADDR BAD ADDR  
000402 000000 000114

FOR ALL OTHER ERRORS, THE REPORT MAY BE MORE EXTENSIVE AND REQUIRE ADDITIONAL DATA TO BE REPORTED.

@

4613  
4614  
4615  
4616  
4617

```
4619  
4627          002000          .TITLE CZDMQAO M8207 STATIC DIAG #2  
4628  
4629  
4630  
4631  
4632  
4633  
4634  
4635 002000          .MCALL  SVC  
4636  
4637  
4638  
4639  
4640  
4641 002000          BGNMOD  CZDMQ  
4642  
4643  
4644          000000          $LSTIN= 0  
4645          000000          $LSTTAG= 0  
4646          000000          SVCINS= 0          ; LIST INSTRUCTIONS, SHIFTED RIGHT  
4647          000000          SVCTST= 0         ; LIST TEST TAGS, SHIFTED RIGHT  
4648          000000          SVCSUB= 0         ; LIST SUBTEST TAGS, SHIFTED RIGHT  
4649          000000          SVCGBL= 0         ; LIST GLOBAL TAGS, SHIFTED RIGHT  
4650          000000          SVCTAG= 0         ; LIST OTHER TAGS, SHIFTED RIGHT  
4651  
4652          ;          CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH  
4653          ;          TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS. CHANGE THE  
4654          ;          SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS. YOU MAY  
4655          ;          CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.  
4656  
4657
```

4659  
 4660  
 4661  
 4662  
 4663  
 4664  
 4665 002000  
 4666  
 4674  
 4675 002000  
 (4) 002000  
 (4) 002000 103  
 (4) 002001 132  
 (4) 002002 104  
 (4) 002003 115  
 (4) 002004 121  
 (6) 002005 000  
 (6) 002006 000  
 (5) 002007 000  
 (5) 002010  
 (4) 002010 101  
 (5) 002011  
 (4) 002011 060  
 (5) 002012  
 (4) 002012 000000  
 (5) 002014  
 (4) 002014 000360  
 (5) 002016  
 (4) 002016 027232  
 (5) 002020  
 (4) 002020 000000  
 (5) 002022  
 (4) 002022 002262  
 (5) 002024  
 (4) 002024 000000  
 (5) 002026  
 (4) 002026 030032  
 (5) 002030  
 (4) 002030 000000  
 (5) 002032  
 (4) 002032 000000  
 (5) 002034  
 (4) 002034 000000  
 (5) 002036  
 (4) 002036 000000  
 (5) 002040  
 (4) 002040 002132  
 (5) 002042  
 (4) 002042 000000  
 (5) 002044  
 (4) 002044 000000  
 (5) 002046  
 (4) 002046 000000  
 (5) 002050  
 (4) 002050 003  
 (3) 002051 000

```

.SBTTL PROGRAM HEADER
:++
: THE PROGRAM HEADER IS THE INTERFACE BETWEEN
: THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
:--

          POINTER BGNAU,BGNDU

          HEADER CZDMQ,A,0,240,,0
L$NAME:: ;DIAGNOSTIC NAME
          .ASCII /C/
          .ASCII /Z/
          .ASCII /D/
          .ASCII /M/
          .ASCII /Q/
          .BYTE 0
          .BYTE 0
          .BYTE 0
L$REV:: ;REVISION LEVEL
          .ASCII /A/
L$DEPO:: ;0
          .ASCII /O/
L$UNIT:: ;NUMBER OF UNITS
          .WORD 0
L$TIML:: ;LONGEST TEST TIME
          .WORD 240.
L$HPCP:: ;POINTER TO H.W. QUES.
          .WORD L$HARD
L$SPCP:: ;POINTER TO S.W. QUES.
          .WORD 0
L$HPTP:: ;PTR. TO DEF. H.W. PTABLE
          .WORD L$HW
L$SPTP:: ;PTR. TO S.W. PTABLE
          .WORD 0
L$LADP:: ;DIAG. END ADDRESS
          .WORD L$LAST
L$STA:: ;RESERVED FOR APT STATS
          .WORD 0
L$CO:: ;
          .WORD 0
L$DTYP:: ;DIAGNOSTIC TYPE
          .WORD 0
L$APT:: ;APT EXPANSION
          .WORD 0
L$DTP:: ;PTR. TO DISPATCH TABLE
          .WORD L$DISPATCH
L$EXP1:: ;EXPANSION WORDS
          .WORD 0
L$EXP2:: ;
          .WORD 0
L$EXP3:: ;
          .WORD 0
L$MREV:: ;SVC REV AND EDIT #
          .BYTE C$REVISION
          .BYTE C$EDIT

```



(5) 002052		L\$EF::		:	DIAG. EVENT FLAGS
(4) 002052	000000		.WORD 0		
(5) 002054	000000		.WORD 0		
(5) 002056		L\$SPC::			
(4) 002056	000000		.WORD 0		
(5) 002060		L\$DEVP::		:	POINTER TO DEVICE TYPE LIST
(4) 002060	002730		.WORD L\$DVTYP		
(5) 002062		L\$REPP::		:	PTR. TO REPORT CODE
(4) 002062	000000		.WORD 0		
(5) 002064		L\$EXP4::			
(4) 002064	000000		.WORD 0		
(5) 002066		L\$EXP5::			
(4) 002066	000000		.WORD 0		
(5) 002070		L\$AUT::		:	PTR. TO ADD UNIT CODE
(4) 002070	012022		.WORD L\$AU		
(5) 002072		L\$DUT::		:	PTR. TO DROP UNIT CODE
(4) 002072	012016		.WORD L\$DU		
(5) 002074		L\$LUN::		:	LUN FOR EXERCISERS TO FILL
(4) 002074	000000		.WORD 0		
(5) 002076		L\$DESP::		:	POINTER TO DIAG. DESCRIPTION
(4) 002076	002312		.WORD L\$DESC		
(5) 002100		L\$LOAD::		:	GENERATE SPECIAL AUTOLOAD EMT
(4) 002100	104035		EMT E\$LOAD		
(5) 002102		L\$ETP::		:	POINTER TO ERRTBL
(4) 002102	000000		.WORD 0		
(5) 002104		L\$ICP::		:	PTR. TO INIT CODE
(4) 002104	011216		.WORD L\$INIT		
(5) 002106		L\$CCP::		:	PTR. TO CLEAN-UP CODE
(4) 002106	012012		.WORD L\$CLEAN		
(5) 002110		L\$ACP::		:	PTR. TO AUTO CODE
(4) 002110	011720		.WORD L\$AUTO		
(5) 002112		L\$PRT::		:	PTR. TO PROTECT TABLE
(4) 002112	002122		.WORD L\$PROT		
(5) 002114		L\$TEST::		:	TEST NUMBER
(4) 002114	000000		.WORD 0		
(5) 002116		L\$DLY::		:	DELAY COUNT
(4) 002116	000000		.WORD 0		
(5) 002120		L\$HIME::		:	PTR. TO HIGH MEM
(4) 002120	000000		.WORD 0		
4676					
4677					
4683	002122		BGNPROT		
(3)	002122	L\$PROT::			
4684	002122		.WORD -1		
4685	002124		.WORD -1		
4686	002126		.WORD -1		
4687	002130		ENDPROT		
4688					



CZDMOAO M8207 STATIC DIAG #2 MACY11 30A(1052) 17-JUL-79 09:23 <sup>C 4</sup> PAGE 26-5  
CZDMQA.P11 25-JUN-79 14:07 DISPATCH TABLE

SEQ 0041

4707  
4708  
4709

4711  
4712  
4713  
4714  
4715  
4716  
4717  
4718  
4719  
4720 002260  
(3) 002260 000013  
(3) 002262  
(3) 002262  
4721 002262 000007  
4722 002264 160170  
4723 002266 000300  
4724 002270 005000  
4725 002272 000003  
4726 002274 000056  
4727 002276 000000  
4728 002300 000000  
4729 002302 000000  
4730 002304 000004  
4731  
4732  
4733 002306 000000  
4734  
4735 002310  
(3) 002310

.SBTTL DEFAULT HARDWARE P-TABLE

```
////////////////////////////////////////  
:/ THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF  
:/ THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE  
:/ IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.  
////////////////////////////////////////
```

```
.ENABL AMA  
BGNHW DFPTBL  
.WORD L10001-L$HW/2  
L$HW::  
DFPTBL::  
.WORD 7 ;MICRO-CPU TYPE.  
.WORD 160170 ;M8200,4,7 CRS ADDRESS  
.WORD 300 ;M8200,4,7 VECTOR ADDRESS  
.WORD 5000 ;INTERRUPT PRIORITY LEVEL  
.WORD 3 ;LINE UNIT TYPE  
.WORD 56 ;SWITCH PACK #1 (DDCMP LINE #)  
.WORD 0 ;SWITCH PACK #2 (BM873 BOOT ADDRESS)  
.WORD 0 ;SWITCH PACK #3  
.WORD 0 ;TEST CONNECTOR INSTALLED FLAG  
.WORD 4 ;CONTAINS BAUD RATE 4=56K BAUD DEFAULT  
;0=2.4K , 1=4.8K , 2=9.6K , 3=19.2K , 4=56K  
;5=250K , 6=500K , 7=1 MEG BAUD  
;0=RUN SW OFF, 1=SW ON  
ENDHW  
L10001:
```

4737  
4738  
4739  
4740  
4741  
4742  
4743  
4744  
4745  
4746  
4747  
4748  
4749  
4750  
4751  
4752  
4753

```
.SBTTL SOFTWARE P-TABLE  
:////////////////////  
:/ THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM  
:/ PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.  
:////////////////////  
                BGNSW  SFPTBL  
                .WORD  L10002-L$SW/2  
L$SW::  
SFPTBL::  
  
                ENDSW  
L10002:
```

002310  
(3) 002310 000000  
(3) 002312  
(3) 002312  
  
002312  
(3) 002312

4755  
 4756  
 4757  
 4758  
 4759  
 4760  
 4761  
 4762  
 4763  
 4764  
 4765  
 4766  
 4767  
 4768  
 4769  
 4770  
 4771  
 4772  
 4773  
 4774  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)  
 (1)

002312

100000  
 040000  
 020000  
 010000  
 004000  
 002000  
 001000  
 000400  
 000200  
 000100  
 000040  
 000020  
 000010  
 000004  
 000002  
 000001  
 001000  
 000400  
 000200  
 000100  
 000040  
 000020  
 000010  
 000004  
 000002  
 000001  
 000040  
 000037

.SBTTL GLOBAL EQUATES SECTION

////  
 :// THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT  
 :// ARE USED IN MORE THAN ONE TEST.  
 ://  
 ////

EQUALS

:  
 : BIT DEFINITIONS

- BIT15== 100000
- BIT14== 40000
- BIT13== 20000
- BIT12== 10000
- BIT11== 4000
- BIT10== 2000
- BIT09== 1000
- BIT08== 400
- BIT07== 200
- BIT06== 100
- BIT05== 40
- BIT04== 20
- BIT03== 10
- BIT02== 4
- BIT01== 2
- BIT00== 1
- :  
 BIT9== BIT09
- BIT8== BIT08
- BIT7== BIT07
- BIT6== BIT06
- BIT5== BIT05
- BIT4== BIT04
- BIT3== BIT03
- BIT2== BIT02
- BIT1== BIT01
- BIT0== BIT00

:  
 : EVENT FLAG DEFINITIONS  
 : EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

- EF.START== 32.
- EF.RESTART== 31.

: START COMMAND WAS ISSUED  
 : RESTART COMMAND WAS ISSUED

```

(1)      000036      EF.CONTINUE== 30.
(1)      000035      EF.NEW==       29.
(1)      000034      EF.PWR==       28.
(1)
(1)
(1)
(1)      000340      PRI07== 340
(1)      000300      PRI06== 300
(1)      000240      PRI05== 240
(1)      000200      PRI04== 200
(1)      000140      PRI03== 140
(1)      000100      PRI02== 100
(1)      000040      PRI01==  40
(1)      000000      PRI00==  0
(1)
(1)      :OPERATOR FLAG BITS
(1)
(1)      000004      EVL==        4
(1)      000010      LOT==       10
(1)      000020      ADR==       20
(1)      000040      IDU==       40
(1)      000100      ISR==      100
(1)      000200      UAM==      200
(1)      000400      BOE==      400
(1)      001000      PNT==     1000
(1)      002000      PRI==     2000
(1)      004000      IXE==     4000
(1)      010000      IBE==    10000
(1)      020000      IER==    20000
(1)      040000      LOE==    40000
(1)      100000      HOE==   100000
  
```

```

; CONTINUE COMMAND WAS ISSUED
; A NEW PASS HAS BEEN STARTED
; A POWER-FAIL/POWER-UP OCCURRED
  
```

: PRIORITY LEVEL DEFINITIONS

: OPERATOR FLAG BITS

4775  
4776  
4777  
4778  
4779  
4780  
4781  
4782  
4783  
4784

```

:*****
: * PROGRAM EVENT FLAG DEFINITIONS
:*****
  
```

4786  
4787  
4788  
4789  
4790  
4791  
4792  
4793  
4794  
4795  
4796 002312  
(4) 002312  
(3) 002312 034115 030062 020067  
(3) 002320 044504 043501 020056  
(3) 002326 031043 047440 020106  
(3) 002334 000062  
(2)  
4797  
4798  
4799  
4800  
4801 002336 000000  
4802 002340 000000  
4803  
4804  
4805  
4806  
4807 002342 000000  
4808 002344 000000  
4809 002346 000000  
4810 002350 000000  
4811 002352 000000  
4812 002354 000000  
4813 002356 000000  
4814 002360 000000  
4815 002362 000000  
4816 002364 000000  
4817 002366 000000  
4818 002370 000000  
4819 002372 000001  
4820 002374 000000  
4821 002376 000001  
4822 002400 000001  
4823 002402 000001  
4824 002404 000001  
4825 002406 000000  
4826 002410 000000  
4827 002412 000000  
4828 002414 000000  
4829 002416 000 00  
4830 002420 000000  
4831 002422 000000  
4832 002424 000000  
4833 002426 000000  
4834 002430 000000  
4835 002432 000000

```
.SBTTL  GLOBAL DATA SECTION
://////
:/      THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
:/      IN MORE THAN ONE TEST.
://////
:*****
:* STORAGE FOR DEVICE REGISTERS
:*****
:      DESCRIPT      <M8207 DIAG. #2 OF 2>
L$DESC:
      .ASCIZ /M8207 DIAG. #2 OF 2/

      .EVEN

:*****
:* PROGRAM CONTROL PARAMETERS
:*****
NEXT:  .WORD  0           ;ADDRESS OF NEXT TEST TO BE EXECUTED
LOCK:  .WORD  0           ;ADDRESS FOR LOCK CURRENT DATA

:*****
:* MISCELLANEOUS STORAGE
:*****
LOGDEV: .WORD  0           ;LOGICAL DEVICE NUMBER
PSTACK: .WORD  0           ;BASE LEVEL PROGRAM STACK POINTER
SUBRPC: .WORD  0           ;PC OF SUBR CALL FOR ERROR REPORTS
ERRFLG: .WORD  0           ;SUBROUTINE ERROR FLAG
RETADR: .WORD  0           ;SUBR ERROR RETURN ADDRESS
STRTSW: .WORD  0           ;SWITCHES AT START OF PROGRAM
STAT:   .WORD  0           ;KM STATUS WORD STORAGE
CLKX:   .WORD  0
MASKX:  .WORD  0
SAVSP:  .WORD  0           ;STACK POINTER STORAGE
SAVPC:  .WORD  0           ;PROGRAM COUNTER STORAGE
ZERO:   .WORD  0
ONE:    .WORD  1
MEMLIM: .WORD  0           ;HIGHEST LOCATION FOR NPR'S
KMACTV: .BLKW  1           ;M8200,4,7 SELECTED ACTIVE
KMNUM:  .BLKW  1           ;OCTAL NUMBER OF M8200,4,7'S
SAVACT: .BLKW  1           ;ORIGINAL ACTIVE DEVICES
SAVNUM: .BLKW  1           ;WORKABLE NUMBER
FLAG:   .WORD  0           ;SCRATCH STORAGE
RUN:    .WORD  0           ;POINTER TO RUNNING DEVICES
FADR:   .WORD  0
WTYPE:  .WORD  0           ;M82XX NUMBER FOR TYPE OF MICO-CPU
$REG5:  .WORD  0           ;STORAGE USED FOR ERROR MSG DATA
$REG4:  .WORD  0
$REG3:  .WORD  0
$REG2:  .WORD  0
$REG1:  .WORD  0
$REG0:  .WORD  0
TYPE:   .WORD  0           ;=0 FOR DMP,=1 FOR M8206
```



4836 002434 000000  
4837 002436 003777  
4838 002440 000000  
4839 002442 000000  
4840 002444 000000  
4841 002446 000000  
4842 002450 000000  
4843 002452 000000  
4844 002454 000000  
4845 002456 000000  
4846 002460 000000  
4847 002462 000000  
4848 002464 000000  
4849 002466 000000  
4850 002470 000000  
4851 002472 000000

MRO: .WORD 0 ;MEMLOC USED INSTEAD OF RO.  
MEMSZ: .WORD 3777 ;INDICATES MEMORIE SIZE, LAST ADDR.  
TEMP: .WORD 0  
\$TEMPO: .WORD 0  
\$TMPO: .WORD 0  
\$GDADR: .WORD 0 ;CONTAINS ADDRESS OF 'GOOD' DATA  
\$BDADR: .WORD 0 ;CONTAINS ADDRESS OF 'BAD' DATA  
\$GDDAT: .WORD 0 ;CONTAINS 'GOOD' DATA  
\$BDDAT: .WORD 0 ;CONTAINS 'BAD' DATA  
 .WORD 0 ;RESERVED--NOT TO BE USED  
 .WORD 0  
FTIME: .WORD 0  
SAVE4: .WORD 0  
SAVE6: .WORD 0  
RUNB: .WORD 0 ;0= RUN OFF, 1= RUN SW ON  
RUNINH: .WORD 0 ;0=RUN SW OFF, 1=RUN SW ON

4852  
4853  
4854  
4855  
4856 002474 000  
4857 002476 000  
4858 002476 000  
4859 002477 000  
4860  
4861  
4862  
4863  
4864  
4865  
4866  
4867  
4868  
4869  
4870  
4871  
4872  
4873  
4874  
4875  
4876  
4877  
4878  
4879

\*\*\*\*\*  
;\* PROGRAM CONTROL FLAGS  
\*\*\*\*\*  
INIFLG: .BYTE 0 ;PROGRAM INITIALIZING FLAG  
 .EVEN  
LOKFLG: .BYTE 0 ;LOCK ON CURRENT TEST FLAG  
QV.FLG: .BYTE 0 ;QUICK VERIFY FLAG  
 .EVEN

4880  
4881 002500 000000  
4882 002502 000000  
4883 002504 000000  
4884  
4885  
4886  
4887  
4888 002506 000000  
4889 002510 000000  
4890 002512 000000  
4891 002514 000000

\*\*\*\*\*  
;\* DEFINITION OF M8200,4,7 STATUS WORDS - STAT1,STAT2,STAT3  
\*\*\*\*\*  
;\*  
;\* STAT1 - BITS 00-08 IS M8200,4,7 VECTOR ADDRESS  
;\* BIT15=1 LINE UNIT IS AN M8203  
;\* BIT14=0 NO TEST CONNECTOR(S) USED  
;\* BIT14=1 H-XXX TEST CONNECTOR WILL BE USED  
;\* BIT13=0 LINE UNIT IS AN M8201  
;\* BIT13=1 LINE UNIT IS AN M8202  
;\* BIT12=1 NO LINE UNIT  
;\* BITS 09-11 IS M8200,4,7 PRIORITY LEVEL  
;\*  
;\* STAT2 - LOW BYTE IS SWITCH PACK #1 (DDCMP LINE NUMBER)  
;\* HIGH BYTE IS SWITCH PACK #2 (BM873 BOOT ADDRESS)  
;\*  
;\* STAT3 - BIT0=1 DO FREE RUNNING TESTS ON M8200,4,7  
\*\*\*\*\*

\*\*\*\*\*  
;\* POINTERS TO M8200,4,7 VECTORS AND REGISTERS  
\*\*\*\*\*  
KMRVEC: 0 ;POINTER TO M8200,4,7 RCV INTRPT VECTOR  
KMRLVL: 0 ;POINTER TO M8200,4,7 RCV INTRPT SERVICE PS  
KMTVEC: 0 ;POINTER TO M8200,4,7 TX INTRPT VECTOR  
KMTLVL: 0 ;POINTER TO M8200,4,7 TX INTRPT SERVICE PS

4892	002516	000000	KMCSR:	0	:POINTER TO M8200,4,7 CONTROL STATUS REGISTER
4893	002520	000000	KMCSRH:	0	:POINTER TO M8200,4,7 CONTROL STATUS REGISTER HIGH BYTE
4894	002522	000000	KMCTL:	0	:POINTER TO M8200,4,7 CONTROL OUT REGISTER
4895	002524	000000	KMPO4:	0	:POINTER TO M8200,4,7 PORT REGISTER - SEL4
4896	002526	000000	KMPO6:	0	:POINTER TO M8200,4,7 PORT REGISTER - SEL6
4897					
4898					
4899			;:***** PRIMARY REG ADRS STORAGE FOR THIS UNIT *****		
4900	002530		;THESE LOCATIONS WILL BE LOADED FOR THE CURRENT UNIT, IN INIT CODE		
4901			REGADR:		
4902					
4903	002530	000100	;:***** STACK USED FOR SUBROUTINE LINKAGE *****		
4904	002730		.BLKW 100		
4905			SSTACK:		
4906					
4907					
4908					
4909					
4910					
4911					

4913  
4914  
4915  
4916  
4917  
4918  
4919  
4920  
4921  
4922  
4923  
4924  
(4)  
(3)  
(3)  
(2)  
4925  
4926  
4927  
4928  
4929  
4930  
4937  
4938  
4939  
4940  
4941

002730  
002730  
002730 034115 030062 026060  
002736 026064 000067

```
.SBTTL GLOBAL TEXT SECTION
:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
:% THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
:% MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
:% MORE THAN ONE TEST.
:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
:*****
:* NAMES OF DEVICES SUPPORTED BY PROGRAM
:*****
L$DVTYP: DEVTYP <M8200,4,7>
.ASCIZ /M8200,4,7/
.EVEN

:
: FORMAT STATEMENTS USED IN PRINT CALLS
:
```

4943  
4944  
4945  
4946  
4947  
4948  
4949  
4950  
4951  
4952  
4953  
4954  
4955  
4956  
4957

.SBTTL GLOBAL SUBROUTINES

:/ THE GLOBAL SUBROUTINES ARE CALLED BY MORE THAN ONE TEST  
:/

-----  
: MACRO'S NEEDED TO CALL SUBROUTINES  
-----

.MACRO POPSP2  
22626  
.ENDM

4959  
4960  
4961  
4962  
4963

4965  
4966  
4967  
4968  
4969  
4970  
4971  
4972  
4973  
4974  
4975  
4976  
4977  
4978  
4979  
4980  
4981  
4982  
4983  
4984  
4985  
4986  
4987  
4988  
4989  
4990

```
:/
: / THE GLOBAL SUBROUTINES ARE CALLED BY MORE THAN ONE TEST
: /
: /
```

```
-----
: MACRO'S NEEDED TO CALL SUBROUTINES
-----
```

```
.MACRO K4ONLY ?N2
      CMP MEMSZ,#2000
      BNE N2
      EXIT TST
      .ENDM
.MACRO ED$CALL XY
      .LIST
      :***** TEST 'XY' *****
      .NLIST
      .ENDM
      .MACRO BADHEAD
      .RADIX 10
      ED$CALL \T$TESTNUM+1
      .RADIX 8
      .ENDM
```

```
4992 .MACRO MYINT
4993 .LIST
4994 MOV KMCSR,R1 ;RECORD DEVICE ADDR.
4995 .NLIST
4996 .ENDM
4997
4998 .MACRO MACEX ?N2
4999 .LIST
5000 ;DO NOT DO TEST IF M8200
5001 .NLIST
5002 TST TYPE
5003 BNE N2
5004 EXIT TST
5005 N2:
5006 .ENDM
5007 .MACRO MACEX2 ?N2
5008 .LIST
5009 ;DO NOT DO TEST IF M8200
5010 .NLIST
5011 CMP WTYPE,#0
5012 BNE N2
5013 EXIT TST
5014 N2:
5015 .ENDM
5016 .MACRO K4ONLY ?N2
5017 .LIST
5018 ;DO NOT DO TEST IF M8200, OR M8204
5019 .NLIST
5020 CMP MEMSZ,#2000
5021 BNE N2
5022 EXIT TST
5023 N2:
5024 ;NOTE THIS TEST IS ONLY DESIGNED FOR 4K MODULE.
5025 .ENDM
5026
5027 .MACRO CLRMAR
5028 ROMCLK
5029 004000
5030 .ENDM
5031 .MACRO ROMCLK
5032 .LIST
5033 JSR R5,ROMCLK ;CLOCK INSTRUCTION
5034 .NLIST
5035 .ENDM
5036
5037 .MACRO SROMCLK
5038 .LIST
5039 JSR R5,SROMCLK
5040 .NLIST
5041 .ENDM
5042 .MACRO SKIP06 NNN
5043 .LIST
5044 ;GOTO 'NNN' IF M8206
5045 .NLIST
5046 CMP WTYPE,#6 ;SEE IF M8206
5047 BEQ NNN
```

```
5048  
5049 .ENDM  
5050 .MACRO SKIP07 NNN  
5051 .LIST  
5052 :GOTO 'NNN' IF M8207  
5053 .NLIST  
5054 CMP WTYPE,#7 ;SEE IF M8200,4,7  
5055 BEQ NNN  
5056 .ENDM  
5057 .MACRO SKIP04 NNN  
5058 .LIST  
5059 :GOTO 'NNN' IF M8204  
5060 .NLIST  
5061 CMP WTYPE,#4 ;SEE IF M8204  
5062 BEQ NNN  
5063 .ENDM  
5064 .MACRO MSTCLR  
5065 JSR R5, .MSTCLR ;CLEAR M8200,4,7  
5066 .ENDM  
5067 002742 .MSTCLR:  
5068 002742 112777 000100 177550 MOV B #BIT6,@KMCSRH ;SET INST.  
5069 002750 142777 000300 177542 BIC B #BIT6!BIT7,@KMCSRH  
5070 002756 000205 RTS R5  
5071  
5072 002760 000024 PATCH: .BLKW 20. ;PATCH AREA.  
5073  
5074  
5075  
5076 003030 ENDBUG:  
5077 : UNSAFE TO PATCH ANY OTHER AREA.  
5078 003030 .ROMCLK:  
5079 003030 000240 NOP  
5080 003032 000240 NOP  
5081 003034 152777 000002 177456 .REGT: BIS B #BIT1,@KMCSRH  
5082 003042 012577 177460 MOV (R5)+,@KMPO6  
5083 003046 152777 000003 177444 BIS B #BIT1!BIT0,@KMCSRH  
5084 003054 142777 000007 177436 BIC B #BIT2!BIT1!BIT0,@KMCSRH  
5085 003062 000205 RTS R5  
5086  
5087 003064 .SROMCLK:  
5088 003064 000240 NOP  
5089 003066 022737 000006 002414 CMP #6,WTYPE  
5090 003074 001357 BNE .REGT  
5091 003076 152777 000002 177414 BIS B #BIT1,@KMCSRH  
5092 003104 012577 177416 MOV (R5)+,@KMPO6  
5093 003110 000240 NOP  
5094 003112 000240 NOP  
5095 003114 142777 000007 177376 BIC B #7,@KMCSRH  
5096 003122 1$:  
5097 003122 152777 000001 177370 BIS B #BIT0,@KMCSRH ;STEP INSTR.  
5098 003130 142777 000007 177362 BIC B #BIT2!BIT1!BIT0,@KMCSRH  
5099 003136 000240 NOP  
5100 003140 000240 NOP  
5101 003142 152777 000002 177350 BIS B #2,@KMCSRH  
5102 003150 2$:  
5103 003150 000205 RTS R5
```



```
5104 003152
5105
5106 003152
(1) 003152 004537 003030
5107 003156 000400
5108 003160
(1) 003160 004537 003030
5109 003164 063220
5110 003166
(1) 003166 004537 003030
5111 003172 060400
5112 003174
(1) 003174 004537 003064
5113 003200 000000
5114 003202 000207
5115
5116 003204
5117
5118 003204
(1) 003204 004537 003030
5119 003210 000401
5120 003212 000207
5121
5122 003214
5123
5124
5125 003214
(1) 003214 004537 003030
5126 003220 000402
5127 003222 000207
5128
5129 003224
5130
5131
5132 003224
(1) 003224 004537 003030
5133 003230 000420
5134 003232 000207
5135
5136 003234
5137
5138
5139 003234
(1) 003234 004537 003030
5140 003240 000600
5141 003242 000207
5142
5143 003244
5144
5145
5146 003244
(1) 003244 004537 003030
5147 003250 000777
5148 003252
(1) 003252 004537 003030
5149 003256 063220
```

```
CLRALL:
;CLEAR C & Z BITS AND BR
ROMCLK
JSR R5,ROMCLK ;CLOCK INSTRUCTION
400 ;0 TO BR
ROMCLK
JSR R5,ROMCLK ;CLOCK INSTRUCTION
63220 ;SP(0) TO BR
ROMCLK
JSR R5,ROMCLK ;CLOCK INSTRUCTION
60400 ;BR,SP(0) + BR
SR0MCLK
JSR R5,SR0MCLK
0
RTS PC

SETBR0:
;SETS BR0 BIT
ROMCLK
JSR R5,ROMCLK ;CLOCK INSTRUCTION
401 ;1 TO BR
RTS PC

SETBR1:
;THIS SUBROUTINE SETS BR1 BIT
ROMCLK
JSR R5,ROMCLK ;NEXT WORD IS INSTRUCTION
000402 ;CLOCK INSTRUCTION
RTS PC ;BR_002

SETBR4:
;THIS SUBROUTINE SETS BR4 BIT
ROMCLK
JSR R5,ROMCLK ;NEXT WORD IS INSTRUCTION
420 ;CLOCK INSTRUCTION
RTS PC

SETBR7:
;THIS SUBROUTINE SETS BR7 BIT
ROMCLK
JSR R5,ROMCLK ;NEXT WORD IS INSTRUCTION
600 ;CLOCK INSTRUCTION
RTS PC

SETC:
;THIS SUBROUTINE SETS THE C BIT
ROMCLK
JSR R5,ROMCLK ;NEXT WORD IS INSTRUCTION
000777 ;CLOCK INSTRUCTION
ROMCLK ;BR 377
JSR R5,ROMCLK ;NEXT WORD IS INSTRUCTION
063220 ;CLOCK INSTRUCTION
;SP(0)_BR
```

```

5150 003260 ROMCLK ;NEXT WORD IS INSTRUCTION
(1) 003260 004537 003030 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
5151 003264 060400 060400 ;BR SP(0)+BR
5152 003266 SROMCLK ;NOW WE MUST CLOCK THE BITS INTO IBUS <13>
(1) 003266 004537 003064 JSR R5,.SROMCLK
5153 003272 000000 0
5154 003274 000207 RTS PC
5155
5156 003276 SETZ: ;THIS SUBROUTINE SETS THE Z BIT
5157
5158
5159 003276 ROMCLK ;NEXT WORD IS INSTRUCTION
(1) 003276 004537 003030 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
5160 003302 000777 000777 ;BR 377
5161 003304 SROMCLK ;NOW CLOCK THE BITS INTO IBUS<13>
(1) 003304 004537 003064 JSR R5,.SROMCLK
5162 003310 000777 0777
5163 003312 000207 RTS PC
5164
5165 003314 RAMDAT: ;THIS SUBROUTINE LOADS R4 WITH THE LOWEST
5166 ;8 BITS OF THE CRAM PC.
5167
5168
5169 003314 005004 CLR R4
5170 003316 017605 000000 MOV @ (SP),R5 ;GOOD DATA
5171 003322 062716 000002 ADD #2,(SP) ;ADJUST STACK
5172 003326 SKIP06 1$ ;IF M8206,WE'LL GET PC A DIFFERENT WAY.
(1) ;GOTO 1$ IF M8206
5173 003336 SKIP07 1$ ;IF M8200,4,7 WE'LL GET PC A DIFFERENT WAY.
(1) ;GOTO 1$ IF M8207
5174 003346 005011 CLR (R1) ;CLEAR BIT10
5175 003350 052711 000400 BIS #BIT8,(R1) ;CLOCK INSTRUCTION IN CRAM THAT
5176 ;JUMPED TO, IT LOADS BR WITH IT
5177 003354 005011 CLR (R1) ;CLR BIT8
5178 003356 ROMCLK ;NEXT WORD IS INSTRUCTION
(1) 003356 004537 003030 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
5179 003362 061225 061225 ;MOV BR TO PORT 5
5180 003364 116104 000005 MOVB 5(R1),R4 ;PUT 'FOUND' IN R4
5181 003370 000207 RTS PC ;RETURN
5182
5183 003372 1$: ROMCLK ;READ PC LOW REG DIRECTLY.
(1) 003372 004537 003030 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
5184 003376 121244 121244 ;IBUS*<12> TO PORT 4
5185 003400 116104 000004 MOVB 4(R1),R4 ;PUT INTO R4
5186 003404 000207 RTS PC ;EXIT
5187
5188 003406 WROM: ;THIS SUBROUTINE WRITES THE ROMMAP INTO THE CRAM
5189
5190
5191 ; BIT #BIT15,STAT1 ;BE SURE M8200,4,7 HAS CRAM
5192 ; BEQ 2$ ;SKIP IF NO CRAM
5193 003406 SKIP07 2$
(1) ;GOTO 2$ IF M8207
5194 003416 005000 CLR R0 ;R0=CRAM ADDRESS
5195 003420 012702 012024 MOV #ROMMAP,R2 ;R2 POINTS TO ROMMAP
5196 003424 012711 002000 1$: MOV #BIT10,(R1) ;SET ROMO

```

```
5197 003430 010061 000004      MOV      R0,4(R1)          ;LOAD CRAM ADDRESS
5198 003434 012261 000006      MOV      (R2)+,6(R1)      ;LOAD WORD TO BE WRITTEN
5199 003440 052711 020000      BIS      #BIT13,(R1)     ;WRITE IT!
5200 003444 005200                INC      R0                ;NEXT ADDRESS
5201 003446 023700 002436      CMP      MEMSZ,R0        ;DONE YET?
5202 003452 001364                BNE     1$                ;BR IF NO
5203 003454 005011                CLR     (R1)              ;CLEAR SELO
5204 003456 000207                2$:   RTS      PC          ;RETURN
5205
5206 003460                MEMSET:
5207                ;THIS SUBROUTINE LOADS CRAM WITH SPECIAL INSTRUCTIONS
5208                ;FOR THE CRAM JUMP TEST. ALL CRAM LOCATIONS ARE LOADED
5209                ;WITH INSTRUCTIONS THAT MOVE A 37 TO THE BR, EXCEPT THE
5210                ;FOLLOWING CRAM ADDRESSES: 0,1,4,7,525,1777. THESE LOCATIONS
5211                ;CONTAIN INSTRUCTIONS WHICH LOAD THE BR WITH THE LOWEST
5212                ;8 BITS OF THAT CRAM ADDRESS.
5213
5214 003460                SKIP07 3$          ;IF M8200,4,7 CAN'T WRITE CRAM!
(1)                ;GOTO 3$ IF M8207
5215 003470 005000                CLR     R0                ;R0 = CRAM ADDRESS
5216 003472 012711 002000                1$:   MOV     #BIT10,(R1)    ;SET ROMO
5217 003476 010061 000004                MOV     R0,4(R1)         ;LOAD CRAM ADDRESS
5218 003502 012761 000437 000006                MOV     #437,6(R1)      ;LOAD INSTRUCTION
5219 003510 052711 020000                BIS     #BIT13,(R1)     ;WRITE INSTRUCTION IN CRAM
5220 003514 005200                INC     R0                ;NEXT ADDRESS
5221 003516 023700 002436                CMP     MEMSZ,R0        ;DONE YET?
5222 003522 001363                BNE     1$                ;BR IF NO
5223 003524 005000                CLR     R0                ;INDEX REGISTER
5224 003526 012711 002000                2$:   MOV     #BIT10,(R1)    ;SET ROMO
5225 003532 016061 003566 000004                MOV     CRAMA(R0),4(R1) ;LOAD CRAM ADDRESS IN SEL4
5226 003540 016061 003602 000006                MOV     INSTU(R0),6(R1) ;LOAD INSTRUCTION TO BE WRITTEN
5227 003546 052711 020000                BIS     #BIT13,(R1)     ;WRITE CRAM!
5228 003552 005720                TST     (R0)+            ;NEXT
5229 003554 022700 000014                CMP     #14,R0          ;DONE YET?
5230 003560 001362                BNE     2$                ;BR IF NO
5231 003562 005011                CLR     (R1)              ;CLEAR ALL BITS
5232 003564 000207                3$:   RTS      PC          ;RETURN
5233
5234 003566 000000 000001 000004  CRAMA:  .WORD  0,1,4,7,1777,525
003574 000007 001777 000525
5235
5236 003602 000400                INSTU: 000400            ;BR_0
5237 003604 000401                000401            ;BR_1
5238 003606 000404                000404            ;BR_4
5239 003610 000407                000407            ;BR_7
5240 003612 000777                000777            ;BR_377
5241 003614 000525                000525            ;BR_125
5242
5243
5244                ;ROUTINE TO SAVE GENERAL REGISTERS FOR ERROR ROUTINE.
5245                ; CALL = JSR PC,SV05
5246 003616 010537 002416                SV05: MOV     R5,$REG5
5247 003622 010437 002420                MOV     R4,$REG4
5248 003626 010337 002422                MOV     R3,$REG3
5249 003632 010237 002424                MOV     R2,$REG2
5250 003636 010137 002426                MOV     R1,$REG1
```

5251	003642	013737	002434	002430	MOV	MRO,\$REGO
5252	003650	000207			RTS	PC
5253						
5254						

5256  
5257  
5258  
5259  
5260  
5261  
5262  
5263  
5264  
5265  
5266  
5267

.SBTTL GLOBAL ERROR REPORT SECTION

:/ THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES  
:/ THAT ARE USED IN MORE THAN ONE TEST.  
:////

003652	047045	047445	022466	TFM1:	.ASCIZ	/%N%06%S4%06%S4%04%N/	
003660	032123	047445	022466				
003666	032123	047445	022464				
003674	000116						
5268	003676	047045	047445	022463	TFM2:	.ASCIZ	/%N%03%S7%03%N/
	003704	033523	047445	022463			
	003712	000116					
5269	003714	047045	047445	022463	TFM3:	.ASCIZ	/%N%03%S10%03%S4%04%N/
	003722	030523	022460	031517			
	003730	051445	022464	032117			
	003736	047045	000				
5270	003741	045	022516	031517	TFM4:	.ASCIZ	/%N%03%S7%03%N/
	003746	051445	022467	031517			
	003754	047045	000				
5271	003757	045	022516	033117	TFM5:	.ASCIZ	/%N%06%S5%06%S3%06%N/
	003764	051445	022465	033117			
	003772	051445	022463	033117			
	004000	047045	000				
5272	004003	045	022516	051101	TFM36:	.ASCIZ	/%N%AREGISTER ADDRESS ERROR,ADDRESS = %06%A,UNIT = %02/
	004010	043505	051511	042524			
	004016	020122	042101	051104			
	004024	051505	020123	051105			
	004032	047522	026122	042101			
	004040	051104	051505	020123			
	004046	020075	047445	022466			
	004054	026101	047125	052111			
	004062	036440	022440	031117			
	004070	000					
5273	004071	045	022516	020101	TFM41:	.ASCIZ	/%N%A CSR HIGH BYTE GOT WRITTEN INTO ON A LOW BYTE XFER/
	004076	051503	020122	044510			
	004104	044107	041040	052131			
	004112	020105	047507	020124			
	004120	051127	052111	042524			
	004126	020116	047111	047524			
	004134	047440	020116	020101			
	004142	047514	020127	054502			
	004150	042524	054040	042506			
	004156	000122					
5274	004160	047045	040445	041440	TFM42:	.ASCIZ	/%N%A CSR LOW BYTE GOT WRITTEN INTO ON A HIGH BYTE XFER/
	004166	051123	046040	053517			
	004174	041040	052131	020105			
	004202	047507	020124	051127			
	004210	052111	042524	020116			
	004216	047111	047524	047440			
	004224	020116	020101	044510			

	004232	044107	041040	052131	
	004240	020105	043130	051105	
	004246	000			
5275	004247	045	022516	047101	TFM40: .ASCIZ /%N%ANEG ADDR TEST DUAL ADDR ERROR-BAD ADDR = %06/
	004254	043505	040440	042104	
	004262	020122	042524	052123	
	004270	042040	040525	020114	
	004276	042101	051104	042440	
	004304	051122	051117	041055	
	004312	042101	040440	042104	
	004320	020122	020075	047445	
	004326	000066			
5276	004330	040445	051440	051103	TFM43: .ASCIZ /%A SCRATCH PAD %03%A DUAL ADDRESS ERROR WITH SP%02/
	004336	052101	044103	050040	
	004344	042101	022440	031517	
	004352	040445	042040	040525	
	004360	020114	042101	051104	
	004366	051505	020123	051105	
	004374	047522	020122	044527	
	004402	044124	051440	022520	
	004410	031117	000		
5277	004413	045	022524	052101	TFM44: .ASCIZ /%T%ATHE MAR REG, CONTENTS= %06/
	004420	042510	046440	051101	
	004426	051040	043505	020054	
	004434	047503	052116	047105	
	004442	051524	020075	047445	
	004450	000066			
5278	004452	052045	040445	044124	TFM45: .ASCIZ /%T%ATHE PC REG, CONTENTS= %06/
	004460	020105	041520	051040	
	004466	043505	020054	047503	
	004474	052116	047105	051524	
	004502	020075	047445	000066	
5279	004510	047045	040445	047516	TFM45A: .ASCII /%N%ANOTE: THIS ERROR MAY BE FALSELY GENERATED IF THE/
	004516	042524	020072	044124	
	004524	051511	042440	051122	
	004532	051117	046440	054501	
	004540	041040	020105	040506	
	004546	051514	046105	020131	
	004554	042507	042516	040522	
	004562	042524	020104	043111	
	004570	052040	042510		
5280	004574	047045	040445	052522	.ASCIZ /%N%ARUN BIT (SW7 OF E28) IS ON/
	004602	020116	044502	020124	
	004610	051450	033527	047440	
	004616	020106	031105	024470	
	004624	044440	020123	047117	
	004632	000			
5281	004633	045	047101	051120	TFM46: .ASCIZ '%ANPR/MISC REGS DATA FAILURE, GOOD =%06%A, BAD =%06''
	004640	046457	051511	020103	
	004646	042522	051507	042040	
	004654	052101	020101	040506	
	004662	046111	051125	026105	
	004670	043440	047517	020104	
	004676	022475	033117	040445	
	004704	020054	040502	020104	
	004712	022475	033117	000	

5282	004717	045	050101	020103	TFM47:	.ASCIZ	'%APC INCR. INCORRECT: S/B= %06%A ; WAS = %06''
	004724	047111	051103	020056			
	004732	047111	047503	051122			
	004740	041505	035124	051440			
	004746	041057	020075	047445			
	004754	022466	020101	020073			
	004762	040527	020123	020075			
	004770	047445	000066				
5283	004774	040515	052123	051105	TMMC:	.ASCIZ	/MASTER CLEAR FAILED TO CLEAR /
	005002	041440	042514	051101			
	005010	043040	044501	042514			
	005016	020104	047524	041440			
	005024	042514	051101	000040			
5284	005032	047045	052045	047045	FM1:	.ASCIZ	/%N%T%N/
	005040	000					
5285							
5286							
5287							
5288	005041	000			EM0:	.ASCIZ	//
5289	005042	051103	046501	042040	EM1:	.ASCIZ	/CRAM DATA ERROR/
	005050	052101	020101	051105			
	005056	047522	000122				
5290	005062	051103	046501	042040	EM2:	.ASCIZ	/CRAM DUAL ADDRESSING ERROR/
	005070	040525	020114	042101			
	005076	051104	051505	044523			
	005104	043516	042440	051122			
	005112	051117	000				
5291	005115	112	046525	020120	EM3:	.ASCIZ	/JUMP ERROR/
	005122	051105	047522	000122			
5292	005130	051103	046501	045040	EM4:	.ASCIZ	/CRAM JUMP TEST FAULT/
	005136	046525	020120	042524			
	005144	052123	043040	052501			
	005152	052114	000				
5293	005155	111	050117	046440	EM5:	.ASCIZ	/IOP MAIN MEMORY TEST/
	005162	044501	020116	042515			
	005170	047515	054522	052040			
	005176	051505	000124				
5294	005202	047511	020120	040515	EM6:	.ASCIZ	/IOP MAR TEST/
	005210	020122	042524	052123			
	005216	000					
5295	005217	102	020122	044522	EM7:	.ASCIZ	/BR RIGHT SHIFT ERROR/
	005224	044107	020124	044123			
	005232	043111	020124	051105			
	005240	047522	000122				
5296	005244	040515	020122	052504	EM10:	.ASCIZ	/MAR DUAL ADDRESSING ERROR/
	005252	046101	040440	042104			
	005260	042522	051523	047111			
	005266	020107	051105	047522			
	005274	000122					
5297	005276	052512	050115	043040	EM11:	.ASCIZ	/JUMP FIELD ERROR/
	005304	042511	042114	042440			
	005312	051122	051117	000			
5298	005317	112	046525	020120	EM12:	.ASCIZ	/JUMP TEST ERROR/
	005324	042524	052123	042440			
	005332	051122	051117	000			
5299	005337	103	047117	044504	EM16:	.ASCIZ	/CONDITION CODE TESTING.Z & C/

5300	005344	044524	047117	041440	
	005352	042117	020105	042524	
	005360	052123	047111	026107	
	005366	020132	020046	000103	
5301	005374				EM35:
	005374	047506	041522	020105	EM17: .ASCIZ /FORCE POWER FAIL ERROR/
	005402	047520	042527	020122	
	005410	040506	046111	042440	
5302	005416	051122	051117	000	
	005423	111	052502	025123	EM27: .ASCIZ 'IBUS* WRITE/READ ERROR'
	005430	053440	044522	042524	
	005436	051057	040505	020104	
	005444	051105	047522	000122	
5303					
5304	005452	041111	051525	047457	EM29: .ASCIZ 'IBUS/OBUS WRITE/READ ERROR'
	005460	052502	020123	051127	
	005466	052111	027505	042522	
	005474	042101	042440	051122	
	005502	051117	000		
5305	005505	045	022516	025101	STM: .ASCIZ '%N%A*****'
	005512	025052	025052	025052	
	005520	025052	025052	025052	
	005526	025052	025052	025052	
	005534	025052	025052	025052	
	005542	025052	025052	025052	
	005550	025052	025052	025052	
	005556	025052	025052	025052	
	005564	025052	025052	025052	
	005572	025052	025052	025052	
	005600	000			
5306	005601	000			DH0: .ASCIZ //
5307					
5308					
5309	005602	054105	042520	052103	DH1: .ASCIZ /EXPECTED FOUND ADDRESS/
	005610	042105	020040	047506	
	005616	047125	020104	040440	
	005624	042104	042522	051523	
	005632	000			
5310	005633	105	050130	041505	DH2: .ASCIZ /EXPECTED FOUND/
	005640	042524	020104	043040	
	005646	052517	042116	000	
5311	005653	106	047522	020115	DH3: .ASCIZ /FROM ADDR TO ADDR BAD ADDR/
	005660	042101	051104	020040	
	005666	047524	040440	042104	
	005674	020122	041040	042101	
	005702	040440	042104	000122	
5312					
5313					
5314					.EVEN
5315					
5316					
5317					
5318					
5319					
5320					-----
5321					: MACRO'S NEEDED TO REPORT ERRORS
					-----



```
5322 .MACRO MDT1
5323 PRINTB #TFM1,$REG2,$REG4,$REG0
5324 .ENDM
5325
5326 .MACRO MDT2
5327 PRINTB #TFM1,$REG5,$REG4,$REG2
5328 .ENDM
5329
5330 .MACRO MDT3
5331 PRINTB #TFM2,$REG5,$REG4
5332 .ENDM
5333
5334 .MACRO MDT4
5335 PRINTB #TFM3,$REG5,$REG4,FLAG
5336 .ENDM
5337
5338 .MACRO MDT5
5339 PRINTB #TFM3,$REG5,$REG4,$REG2
5340 .ENDM
5341
5342 .MACRO MDT0
5343 .ENDM
5344 .MACRO MDT6
5345 PRINTB #TFM4,$REG2,$REG4
5346 .ENDM
5347
5348 .MACRO MDT7
5349 PRINTB #TFM4,$REG5,$REG4
```

CZDMQAO MB207 STATIC DIAG #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 M 5  
GLOBAL ERROR REPORT SECTION PAGE 29

SEQ 0064

5351  
5352  
5353  
5354

.ENDM  
.MACRO MDT8  
PRINTB #TFM5,FADR,\$REG5,\$REG4  
.ENDM

5356  
5357  
5358

.MACRO    \$MD    ERRNN    ERNB    ERHM    ERFM  
          BGNMSG    ERR'ERRNN

5360				PRINTB	#FM1,#EM'ERNB
5361				PRINTB	#FM1,#DH'ERHM
5362				MDT'ERFM	
5363				PRINTB	#STM
5364				ENDMSG	
5365				.ENDM	
5366				.MACRO	ERROR ECB
5367				JSR	PC,SV05
5368				ERRDF	'ECB',EMO,ERR'ECB'
5369				.ENDM	
5370					
5371					
5372					
5373					
5374	005710			\$MD	1,1,1,1
(4)	005710			ERR1::	
(9)	005710	012746	005042	MOV	#EM1,-(SP)
(8)	005714	012746	005032	MOV	#FM1,-(SP)
(7)	005720	012746	000002	MOV	#2,-(SP)
(4)	005724	010600		MOV	SP,R0
(5)	005726	104414		TRAP	C\$PNTB
(5)	005730	062706	000006	ADD	#6,SP
(9)	005734	012746	005602	MOV	#DH1,-(SP)
(8)	005740	012746	005032	MOV	#FM1,-(SP)
(7)	005744	012746	000002	MOV	#2,-(SP)
(4)	005750	010600		MOV	SP,R0
(5)	005752	104414		TRAP	C\$PNTB
(5)	005754	062706	000006	ADD	#6,SP
(12)	005760	013746	002430	MOV	\$REG0,-(SP)
(11)	005764	013746	002420	MOV	\$REG4,-(SP)
(10)	005770	013746	002424	MOV	\$REG2,-(SP)
(9)	005774	012746	003652	MOV	#TFM1,-(SP)
(8)	006000	012746	0000C4	MOV	#4,-(SP)
(5)	006004	010600		MOV	SP,R0
(6)	006006	104414		TRAP	C\$PNTB
(6)	006010	062706	000012	ADD	#12,SP
(8)	006014	012746	005505	MOV	#STM,-(SP)
(7)	006020	012746	000001	MOV	#1,-(SP)
(4)	006024	010600		MOV	SP,R0
(5)	006026	104414		TRAP	C\$PNTB
(5)	006030	062706	000004	ADD	#4,SP
(4)	006034			L10003:	
(4)	006034	104423		TRAP	C\$MSG
5375	006036			\$MD	2,2,1,1
(4)	006036			ERR2::	
(9)	006036	012746	005062	MOV	#EM2,-(SP)
(8)	006042	012746	005032	MOV	#FM1,-(SP)
(7)	006046	012746	000002	MOV	#2,-(SP)
(4)	006052	010600		MOV	SP,R0
(5)	006054	104414		TRAP	C\$PNTB
(5)	006056	062706	000006	ADD	#6,SP
(9)	006062	012746	005602	MOV	#DH1,-(SP)
(8)	006066	012746	005032	MOV	#FM1,-(SP)
(7)	006072	012746	000002	MOV	#2,-(SP)
(4)	006076	010600		MOV	SP,R0
(5)	006100	104414		TRAP	C\$PNTB

(5)	006102	062706	000006	ADD	#6,SP
(12)	006106	013746	002430	MOV	\$REG0,-(SP)
(11)	006112	013746	002420	MOV	\$REG4,-(SP)
(10)	006116	013746	002424	MOV	\$REG2,-(SP)
(9)	006122	012746	003652	MOV	#TFM1,-(SP)
(8)	006126	012746	000004	MOV	#4,-(SP)
(5)	006132	010600		MOV	SP,R0
(6)	006134	104414		TRAP	C\$PNTB
(6)	006136	062706	000012	ADD	#12,SP
(8)	006142	012746	005505	MOV	#STM,-(SP)
(7)	006146	012746	000001	MOV	#1,-(SP)
(4)	006152	010600		MOV	SP,R0
(5)	006154	104414		TRAP	C\$PNTB
(5)	006156	062706	000004	ADD	#4,SP
(4)	006162				
(4)	006162	104423		L10004:	TRAP
5376	006164			\$MD	C\$MSG
(4)	006164				3,1,1,2
(9)	006164	012746	005042	ERR3::	MOV
(8)	006170	012746	005032		#EM1,-(SP)
(7)	006174	012746	000002		MOV
(4)	006200	010600			#FM1,-(SP)
(5)	006202	104414			MOV
(5)	006204	062706	000006		#2,-(SP)
(9)	006210	012746	005602		MOV
(8)	006214	012746	005032		SP,R0
(7)	006220	012746	000002		TRAP
(4)	006224	010600			C\$PNTB
(5)	006226	104414			ADD
(5)	006230	062706	000006		#6,SP
(12)	006234	013746	002424		MOV
(11)	006240	013746	002420		\$REG2,-(SP)
(10)	006244	013746	002416		MOV
(9)	006250	012746	003652		\$REG4,-(SP)
(8)	006254	012746	000004		MOV
(5)	006260	010600			\$REG5,-(SP)
(6)	006262	104414			MOV
(6)	006264	062706	000012		#TFM1,-(SP)
(8)	006270	012746	005505		MOV
(7)	006274	012746	000001		#4,-(SP)
(4)	006300	010600			MOV
(5)	006302	104414			SP,R0
(5)	006304	062706	000004		TRAP
(4)	006310				C\$PNTB
(4)	006310	104423		L10005:	ADD
5377	006312				#4,SP
(4)	006312			ERR4::	TRAP
(9)	006312	012746	005115		C\$MSG
(8)	006316	012746	005032		\$MD
(7)	006322	012746	000002		4,3,2,3
(4)	006326	010600			MOV
(5)	006330	104414			#EM3,-(SP)
(5)	006332	062706	000006		MOV
(9)	006336	012746	005633		#FM1,-(SP)
(8)	006342	012746	005032		MOV
(7)	006346	012746	000002		#2,-(SP)

(4)	006352	010600		MOV	SP,R0
(5)	006354	104414		TRAP	C\$PNTB
(5)	006356	062706	000006	ADD	#6,SP
(11)	006362	013746	002420	MOV	\$REG4,-(SP)
(10)	006366	013746	002416	MOV	\$REG5,-(SP)
(9)	006372	012746	003676	MOV	#TFM2,-(SP)
(8)	006376	012746	000003	MOV	#3,-(SP)
(5)	006402	010600		MOV	SP,R0
(6)	006404	104414		TRAP	C\$PNTB
(6)	006406	062706	000010	ADD	#10,SP
(8)	006412	012746	005505	MOV	#STM,-(SP)
(7)	006416	012746	000001	MOV	#1,-(SP)
(4)	006422	010600		MOV	SP,R0
(5)	006424	104414		TRAP	C\$PNTB
(5)	006426	062706	000004	ADD	#4,SP
(4)	006432				
(4)	006432	104423		L10006:	TRAP C\$MSG

5379 006434  
(4) 006434  
(9) 006434 012746 005130  
(8) 006440 012746 005032  
(7) 006444 012746 000002  
(4) 006450 010600  
(5) 006452 104414  
(5) 006454 062706 000006  
(9) 006460 012746 005633  
(8) 006464 012746 005032  
(7) 006470 012746 000002  
(4) 006474 010600  
(5) 006476 104414  
(5) 006500 062706 000006  
(11) 006504 013746 002420  
(10) 006510 013746 002416  
(9) 006514 012746 003676  
(8) 006520 012746 000003  
(5) 006524 010600  
(6) 006526 104414  
(6) 006530 062706 000010  
(8) 006534 012746 005505  
(7) 006540 012746 000001  
(4) 006544 010600  
(5) 006546 104414  
(5) 006550 062706 000004  
(4) 006554  
(4) 006554 104423

ERR5:: \$MD 5,4,2,3  
MOV #EM4,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV #DH2,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV \$REG4,-(SP)  
MOV \$REG5,-(SP)  
MOV #TFM2,-(SP)  
MOV #3,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #10,SP  
MOV #STM,-(SP)  
MOV #1,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #4,SP  
L10007: TRAP C\$MSG

5381 006556  
(4) 006556  
(9) 006556 012746 005155  
(8) 006562 012746 005032  
(7) 006566 012746 000002  
(4) 006572 010600  
(5) 006574 104414  
(5) 006576 062706 000006  
(9) 006602 012746 005602  
(8) 006606 012746 005032  
(7) 006612 012746 000002  
(4) 006616 010600  
(5) 006620 104414  
(5) 006622 062706 000006  
(12) 006626 013746 002406  
(11) 006632 013746 002420  
(10) 006636 013746 002416  
(9) 006642 012746 003714  
(8) 006646 012746 000004  
(5) 006652 010600  
(6) 006654 104414  
(6) 006656 062706 000012  
(8) 006662 012746 005505  
(7) 006666 012746 000001  
(4) 006672 010600  
(5) 006674 104414  
(5) 006676 062706 000004  
(4) 006702  
(4) 006702 104423  
5382 006704  
(4) 006704  
(9) 006704 012746 005202  
(8) 006710 012746 005032  
(7) 006714 012746 000002  
(4) 006720 010600  
(5) 006722 104414  
(5) 006724 062706 000006  
(9) 006730 012746 005602  
(8) 006734 012746 005032  
(7) 006740 012746 000002  
(4) 006744 010600  
(5) 006746 104414  
(5) 006750 062706 000006  
(12) 006754 013746 002424  
(11) 006760 013746 002420  
(10) 006764 013746 002416  
(9) 006770 012746 003714  
(8) 006774 012746 000004  
(5) 007000 010600  
(6) 007002 104414  
(6) 007004 062706 000012  
(8) 007010 012746 005505  
(7) 007014 012746 000001  
(4) 007020 010600  
(5) 007022 104414  
(5) 007024 062706 000004

ERR6:: \$MD 6,5,1,4  
MOV #EM5,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV #DH1,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV FLAG,-(SP)  
MOV \$REG4,-(SP)  
MOV \$REG5,-(SP)  
MOV #TFM3,-(SP)  
MOV #4,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #12,SP  
MOV #STM,-(SP)  
MOV #1,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #4,SP  
L10010:  
TRAP C\$MSG  
ERR7:: \$MD 7,6,1,5  
MOV #EM6,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV #DH1,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV \$REG2,-(SP)  
MOV \$REG4,-(SP)  
MOV \$REG5,-(SP)  
MOV #TFM3,-(SP)  
MOV #4,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #12,SP  
MOV #STM,-(SP)  
MOV #1,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #4,SP



(4) 007030  
(4) 007030 104423  
5383 007032  
(4) 007032  
(9) 007032 012746 005217  
(8) 007036 012746 005032  
(7) 007042 012746 000002  
(4) 007046 010600  
(5) 007050 104414  
(5) 007052 062706 000006  
(9) 007056 012746 005633  
(8) 007062 012746 005032  
(7) 007066 012746 000002  
(4) 007072 010600  
(5) 007074 104414  
(5) 007076 062706 000006  
(11) 007102 013746 002420  
(10) 007106 013746 002416  
(9) 007112 012746 003676  
(8) 007116 012746 000003  
(5) 007122 010600  
(6) 007124 104414  
(6) 007126 062706 000010  
(8) 007132 012746 005505  
(7) 007136 012746 000001  
(4) 007142 010600  
(5) 007144 104414  
(5) 007146 062706 000004  
(4) 007152  
(4) 007152 104423  
5384 007154  
(4) 007154  
(9) 007154 012746 005244  
(8) 007160 012746 005032  
(7) 007164 012746 000002  
(4) 007170 010600  
(5) 007172 104414  
(5) 007174 062706 000006  
(9) 007200 012746 005633  
(8) 007204 012746 005032  
(7) 007210 012746 000002  
(4) 007214 010600  
(5) 007216 104414  
(5) 007220 062706 000006  
(11) 007224 013746 002420  
(10) 007230 013746 002424  
(9) 007234 012746 003741  
(8) 007240 012746 000003  
(5) 007244 010600  
(6) 007246 104414  
(6) 007250 062706 000010  
(8) 007254 012746 005505  
(7) 007260 012746 000001  
(4) 007264 010600  
(5) 007266 104414  
(5) 007270 062706 000004

L10011:  
TRAP C\$MSG  
\$MD 10,7,2,3  
ERR10::  
MOV #EM7,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV #DH2,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV \$REG4,-(SP)  
MOV \$REG5,-(SP)  
MOV #TFM2,-(SP)  
MOV #3,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #10,SP  
MOV #STM,-(SP)  
MOV #1,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #4,SP  
L10012:  
TRAP C\$MSG  
\$MD 11,10,2,6  
ERR11::  
MOV #EM10,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV #DH2,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV \$REG4,-(SP)  
MOV \$REG2,-(SP)  
MOV #TFM4,-(SP)  
MOV #3,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #10,SP  
MOV #STM,-(SP)  
MOV #1,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #4,SP

(4) 007274  
(4) 007274 104423  
5385 007276  
(4) 007276  
(9) 007276 012746 005217  
(8) 007302 012746 005032  
(7) 007306 012746 000002  
(4) 007312 010600  
(5) 007314 104414  
(5) 007316 062706 000006  
(9) 007322 012746 005633  
(8) 007326 012746 005032  
(7) 007332 012746 000002  
(4) 007336 010600  
(5) 007340 104414  
(5) 007342 062706 000006  
(11) 007346 013746 002420  
(10) 007352 013746 002416  
(9) 007356 012746 003741  
(8) 007362 012746 000003  
(5) 007366 010600  
(6) 007370 104414  
(6) 007372 062706 000010  
(8) 007376 012746 005505  
(7) 007402 012746 000001  
(4) 007406 010600  
(5) 007410 104414  
(5) 007412 062706 000004  
(4) 007416  
(4) 007416 104423  
5386 007420  
(4) 007420  
(9) 007420 012746 005244  
(8) 007424 012746 005032  
(7) 007430 012746 000002  
(4) 007434 010600  
(5) 007436 104414  
(5) 007440 062706 000006  
(9) 007444 012746 005633  
(8) 007450 012746 005032  
(7) 007454 012746 000002  
(4) 007460 010600  
(5) 007462 104414  
(5) 007464 062706 000006  
(11) 007470 013746 002420  
(10) 007474 013746 002416  
(9) 007500 012746 003676  
(8) 007504 012746 000003  
(5) 007510 010600  
(6) 007512 104414  
(6) 007514 062706 000010  
(8) 007520 012746 005505  
(7) 007524 012746 000001  
(4) 007530 010600  
(5) 007532 104414  
(5) 007534 062706 000004

L10013:  
TRAP C\$MSG  
\$MD 12,7,2,7  
ERR12::  
MOV #EM7,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV #DH2,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV \$REG4,-(SP)  
MOV \$REG5,-(SP)  
MOV #TFM4,-(SP)  
MOV #3,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #10,SP  
MOV #STM,-(SP)  
MOV #1,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #4,SP  
L10014:  
TRAP C\$MSG  
\$MD 13,10,2,3  
ERR13::  
MOV #EM10,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV #DH2,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV \$REG4,-(SP)  
MOV \$REG5,-(SP)  
MOV #TFM2,-(SP)  
MOV #3,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #10,SP  
MOV #STM,-(SP)  
MOV #1,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #4,SP

(4) 007540  
(4) 007540 104423  
5387 007542  
(4) 007542  
(9) 007542 012746 005276  
(8) 007546 012746 005032  
(7) 007552 012746 000002  
(4) 007556 010600  
(5) 007560 104414  
(5) 007562 062706 000006  
(9) 007566 012746 005633  
(8) 007572 012746 005032  
(7) 007576 012746 000002  
(4) 007602 010600  
(5) 007604 104414  
(5) 007606 062706 000006  
(11) 007612 013746 002420  
(10) 007616 013746 002424  
(9) 007622 012746 003741  
(8) 007626 012746 000003  
(5) 007632 010600  
(6) 007634 104414  
(6) 007636 062706 000010  
(8) 007642 012746 005505  
(7) 007646 012746 000001  
(4) 007652 010600  
(5) 007654 104414  
(5) 007656 062706 000004  
(4) 007662  
(4) 007662 104423

L10015:  
TRAP C\$MSG  
\$MD 14,11,2,6  
ERR14::  
MOV #EM11,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV #DH2,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV \$REG4,-(SP)  
MOV \$REG2,-(SP)  
MOV #TFM4,-(SP)  
MOV #3,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #10,SP  
MOV #STM,-(SP)  
MOV #1,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #4,SP  
L10016:  
TRAP C\$MSG

5389 007664  
(4) 007664  
(9) 007664 012746 005317  
(8) 007670 012746 005032  
(7) 007674 012746 000002  
(4) 007700 010600  
(5) 007702 104414  
(5) 007704 062706 000006  
(9) 007710 012746 005653  
(8) 007714 012746 005032  
(7) 007720 012746 000002  
(4) 007724 010600  
(5) 007726 104414  
(5) 007730 062706 000006  
(12) 007734 013746 002420  
(11) 007740 013746 002416  
(10) 007744 013746 002412  
(9) 007750 012746 003757  
(8) 007754 012746 000004  
(5) 007760 010600  
(6) 007762 104414  
(6) 007764 062706 000012  
(8) 007770 012746 005505  
(7) 007774 012746 000001  
(4) 010000 010600  
(5) 010002 104414  
(5) 010004 062706 000004  
(4) 010010  
(4) 010010 104423  
5390 010012  
(4) 010012  
(9) 010012 012746 005337  
(8) 010016 012746 005032  
(7) 010022 012746 000002  
(4) 010026 010600  
(5) 010030 104414  
(5) 010032 062706 000006  
(9) 010036 012746 005633  
(8) 010042 012746 005032  
(7) 010046 012746 000002  
(4) 010052 010600  
(5) 010054 104414  
(5) 010056 062706 000006  
(11) 010062 013746 002420  
(10) 010066 013746 002416  
(9) 010072 012746 003741  
(8) 010076 012746 000003  
(5) 010102 010600  
(6) 010104 104414  
(6) 010106 062706 000010  
(8) 010112 012746 005505  
(7) 010116 012746 000001  
(4) 010122 010600  
(5) 010124 104414  
(5) 010126 062706 000004  
(4) 010132

ERR15:: \$MD 15,12,3,8  
MOV #EM12,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV #DH3,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV \$REG4,-(SP)  
MOV \$REG5,-(SP)  
MOV FADR,-(SP)  
MOV #TFM5,-(SP)  
MOV #4,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #12,SP  
MOV #STM,-(SP)  
MOV #1,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #4,SP  
L10017: TRAP C\$MSG  
\$MD 16,16,2,7  
ERR16:: MOV #EM16,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV #DH2,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV \$REG4,-(SP)  
MOV \$REG5,-(SP)  
MOV #TFM4,-(SP)  
MOV #3,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #10,SP  
MOV #STM,-(SP)  
MOV #1,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #4,SP  
L10020:

CZDMQAO M8207 STATIC DIAG #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 <sup>K 6</sup> PAGE 34-1  
GLOBAL ERROR REPORT SECTION

SEQ 0075

(4) 010132 104423  
5391

TRAP C\$MSG

5393					
5394	010134			ERR17::	\$MD 17,17,0,0
(4)	010134				MOV #EM17,-(SP)
(9)	010134	012746	005374		MOV #FM1,-(SP)
(8)	010140	012746	005032		MOV #2,-(SP)
(7)	010144	012746	000002		MOV SP,R0
(4)	010150	010600			TRAP C\$PNTB
(5)	010152	104414			ADD #6,SP
(5)	010154	062706	000006		MOV #DH0,-(SP)
(9)	010160	012746	005601		MOV #FM1,-(SP)
(8)	010164	012746	005032		MOV #2,-(SP)
(7)	010170	012746	000002		MOV SP,R0
(4)	010174	010600			TRAP C\$PNTB
(5)	010176	104414			ADD #6,SP
(5)	010200	062706	000006		MOV #STM,-(SP)
(8)	010204	012746	005505		MOV #1,-(SP)
(7)	010210	012746	000001		MOV SP,R0
(4)	010214	010600			TRAP C\$PNTB
(5)	010216	104414			ADD #4,SP
(5)	010220	062706	000004	L10021:	TRAP C\$MSG
(4)	010224				\$MD 29,29,2,3
(4)	010224	104423		ERR29::	MOV #EM29,-(SP)
5395	010226				MOV #FM1,-(SP)
(4)	010226				MOV #2,-(SP)
(9)	010226	012746	005452		MOV SP,R0
(8)	010232	012746	005032		TRAP C\$PNTB
(7)	010236	012746	000002		ADD #6,SP
(4)	010242	010600			MOV #DH2,-(SP)
(5)	010244	104414			MOV #FM1,-(SP)
(5)	010246	062706	000006		MOV #2,-(SP)
(9)	010252	012746	005633		MOV SP,R0
(8)	010256	012746	005032		TRAP C\$PNTB
(7)	010262	012746	000002		ADD #6,SP
(4)	010266	010600			MOV #REG4,-(SP)
(5)	010270	104414			MOV #REG5,-(SP)
(5)	010272	062706	000006		MOV #TFM2,-(SP)
(11)	010276	013746	002420		MOV #3,-(SP)
(10)	010302	013746	002416		MOV SP,R0
(9)	010306	012746	003676		TRAP C\$PNTB
(8)	010312	012746	000003		ADD #10,SP
(5)	010316	010600			MOV #STM,-(SP)
(6)	010320	104414			MOV #1,-(SP)
(6)	010322	062706	000010		MOV SP,R0
(8)	010326	012746	005505		TRAP C\$PNTB
(7)	010332	012746	000001		ADD #4,SP
(4)	010336	010600		L10022:	TRAP C\$MSG
(5)	010340	104414			\$MD 35,35,2,3
(5)	010342	062706	000004	ERR35::	MOV #EM35,-(SP)
(4)	010346				MOV #FM1,-(SP)
(4)	010346	104423			MOV #2,-(SP)
5396	010350				MOV SP,R0
(4)	010350				TRAP C\$PNTB
(9)	010350	012746	005374		ADD #4,SP
(8)	010354	012746	005032		MOV #EM35,-(SP)
(7)	010360	012746	000002		MOV #FM1,-(SP)
(4)	010364	010600			MOV #2,-(SP)
					MOV SP,R0

(5)	010366	104414		TRAP	C\$PNTB
(5)	010370	062706	000006	ADD	#6,SP
(9)	010374	012746	005633	MOV	#DH2,-(SP)
(8)	010400	012746	005032	MOV	#FM1,-(SP)
(7)	010404	012746	000002	MOV	#2,-(SP)
(4)	010410	010600		MOV	SP,R0
(5)	010412	104414		TRAP	C\$PNTB
(5)	010414	062706	000006	ADD	#6,SP
(11)	010420	013746	002420	MOV	\$REG4,-(SP)
(10)	010424	013746	002416	MOV	\$REG5,-(SP)
(9)	010430	012746	003676	MOV	#TFM2,-(SP)
(8)	010434	012746	000003	MOV	#3,-(SP)
(5)	010440	010600		MOV	SP,R0
(6)	010442	104414		TRAP	C\$PNTB
(6)	010444	062706	000010	ADD	#10,SP
(8)	010450	012746	005505	MOV	#STM,-(SP)
(7)	010454	012746	000001	MOV	#1,-(SP)
(4)	010460	010600		MOV	SP,R0
(5)	010462	104414		TRAP	C\$PNTB
(5)	010464	062706	000004	ADD	#4,SP
(4)	010470				
(4)	010470	104423		L10023: TRAP	C\$MSG
5397					
5398	010472			BGNMSG	ERR36
(3)	010472			ERR36::	
5399	010472			PRINTB	#STM
(7)	010472	012746	005505	MOV	#STM,-(SP)
(6)	010476	012746	000001	MOV	#1,-(SP)
(3)	010502	010600		MOV	SP,R0
(4)	010504	104414		TRAP	C\$PNTB
(4)	010506	062706	000004	ADD	#4,SP
5400	010512			ENDMSG	
(3)	010512			L10024:	
(3)	010512	104423		TRAP	C\$MSG
5401					
5402	010514			BGNMSG	ERR40
(3)	010514			ERR40::	
5403	010514			PRINTF	#TFM40,R2
(8)	010514	010246		MOV	R2,-(SP)
(7)	010516	012746	004247	MOV	#TFM40,-(SP)
(6)	010522	012746	000002	MOV	#2,-(SP)
(3)	010526	010600		MOV	SP,R0
(4)	010530	104417		TRAP	C\$PNTF
(4)	010532	062706	000006	ADD	#6,SP
5404	010536			PRINTB	#STM
(7)	010536	012746	005505	MOV	#STM,-(SP)
(6)	010542	012746	000001	MOV	#1,-(SP)
(3)	010546	010600		MOV	SP,R0
(4)	010550	104414		TRAP	C\$PNTB
(4)	010552	062706	000004	ADD	#4,SP
5405	010556			ENDMSG	
(3)	010556			L10025:	
(3)	010556	104423		TRAP	C\$MSG
5406	010560			BGNMSG	ERR41
(3)	010560			ERR41::	
5407	010560			PRINTF	#TFM41

(7)	010560	012746	004071		MOV	#TFM41,-(SP)
(6)	010564	012746	000001		MOV	#1,-(SP)
(3)	010570	010600			MOV	SP,R0
(4)	010572	104417			TRAP	C\$PNTF
(4)	010574	062706	000004		ADD	#4,SP
5408	010600			PRINTB	#STM	
(7)	010600	012746	005505		MOV	#STM,-(SP)
(6)	010604	012746	000001		MOV	#1,-(SP)
(3)	010610	010600			MOV	SP,R0
(4)	010612	104414			TRAP	C\$PNTB
(4)	010614	062706	000004		ADD	#4,SP
5409	010620			ENDMSG		
(3)	010620			L10026:		
(3)	010620	104423			TRAP	C\$MSG
5410	010622			BGNMSG	ERR42	
(3)	010622			ERR42::		
5411	010622			PRINTF	#TFM42	
(7)	010622	012746	004160		MOV	#TFM42,-(SP)
(6)	010626	012746	000001		MOV	#1,-(SP)
(3)	010632	010600			MOV	SP,R0
(4)	010634	104417			TRAP	C\$PNTF
(4)	010636	062706	000004		ADD	#4,SP
5412	010642			PRINTB	#STM	
(7)	010642	012746	005505		MOV	#STM,-(SP)
(6)	010646	012746	000001		MOV	#1,-(SP)
(3)	010652	010600			MOV	SP,R0
(4)	010654	104414			TRAP	C\$PNTB
(4)	010656	062706	000004		ADD	#4,SP
5413	010662			ENDMSG		
(3)	010662			L10027:		
(3)	010662	104423			TRAP	C\$MSG
5414						
5415	010664			BGNMSG	ERR43	
(3)	010664			ERR43::		
5416	010664			PRINTF	#TFM43,R5,R4	
(9)	010664	010446			MOV	R4,-(SP)
(8)	010666	010546			MOV	R5,-(SP)
(7)	010670	012746	004330		MOV	#TFM43,-(SP)
(6)	010674	012746	000003		MOV	#3,-(SP)
(3)	010700	010600			MOV	SP,R0
(4)	010702	104417			TRAP	C\$PNTF
(4)	010704	062706	000010		ADD	#10,SP
5417	010710			PRINTB	#STM	
(7)	010710	012746	005505		MOV	#STM,-(SP)
(6)	010714	012746	000001		MOV	#1,-(SP)
(3)	010720	010600			MOV	SP,R0
(4)	010722	104414			TRAP	C\$PNTB
(4)	010724	062706	000004		ADD	#4,SP
5418	010730			ENDMSG		
(3)	010730			L10030:		
(3)	010730	104423			TRAP	C\$MSG
5419	010732			BGNMSG	ERR44	
(3)	010732			ERR44::		
5420	010732			PRINTF	#TFM44,#TMMC,R4	
(9)	010732	010446			MOV	R4,-(SP)
(8)	010734	012746	004774		MOV	#TMMC,-(SP)



(7)	010740	012746	004413	MOV	#TFM44,-(SP)
(6)	010744	012746	000003	MOV	#3,-(SP)
(3)	010750	010600		MOV	SP,R0
(4)	010752	104417		TRAP	C\$PNTF
(4)	010754	062706	000010	ADD	#10,SP
5421	010760			PRINTB	#STM
(7)	010760	012746	005505	MOV	#STM,-(SP)
(6)	010764	012746	000001	MOV	#1,-(SP)
(3)	010770	010600		MOV	SP,R0
(4)	010772	104414		TRAP	C\$PNTB
(4)	010774	062706	000004	ADD	#4,SP
5422	011000			ENDMSG	
(3)	011000			L10031:	
(3)	011000	104423		TRAP	C\$MESSG
5423	011002			BGNMSG	ERR45
(3)	011002			ERR45::	
5424	011002			PRINTF	#TFM45,#TMMC,R4
(9)	011002	010446		MOV	R4,-(SP)
(8)	011004	012746	004774	MOV	#TMMC,-(SP)
(7)	011010	012746	004452	MOV	#TFM45,-(SP)
(6)	011014	012746	000003	MOV	#3,-(SP)
(3)	011020	010600		MOV	SP,R0
(4)	011022	104417		TRAP	C\$PNTF
(4)	011024	062706	000010	ADD	#10,SP
5425	011030			PRINTB	#TFM45A
(7)	011030	012746	004510	MOV	#TFM45A,-(SP)
(6)	011034	012746	000001	MOV	#1,-(SP)
(3)	011040	010600		MOV	SP,R0
(4)	011042	104414		TRAP	C\$PNTB
(4)	011044	062706	000004	ADD	#4,SP
5426	011050			PRINTB	#STM
(7)	011050	012746	005505	MOV	#STM,-(SP)
(6)	011054	012746	000001	MOV	#1,-(SP)
(3)	011060	010600		MOV	SP,R0
(4)	011062	104414		TRAP	C\$PNTB
(4)	011064	062706	000004	ADD	#4,SP
5427	011070			ENDMSG	
(3)	011070			L10032:	
(3)	011070	104423		TRAP	C\$MESSG
5428	011072			BGNMSG	ERR46
(3)	011072			ERR46::	
5429	011072			PRINTF	#TFM46,\$GDDAT,R4
(9)	011072	010446		MOV	R4,-(SP)
(8)	011074	013746	002452	MOV	\$GDDAT,-(SP)
(7)	011100	012746	004633	MOV	#TFM46,-(SP)
(6)	011104	012746	000003	MOV	#3,-(SP)
(3)	011110	010600		MOV	SP,R0
(4)	011112	104417		TRAP	C\$PNTF
(4)	011114	062706	000010	ADD	#10,SP
5430	011120			PRINTB	#STM
(7)	011120	012746	005505	MOV	#STM,-(SP)
(6)	011124	012746	000001	MOV	#1,-(SP)
(3)	011130	010600		MOV	SP,R0
(4)	011132	104414		TRAP	C\$PNTB
(4)	011134	062706	000004	ADD	#4,SP
5431	011140			ENDMSG	

(3)	011140			L10033:	
(3)	011140	104423		TRAP	C\$MSG
5432					
5433	011142			BGNMSG	ERR47
(3)	011142			ERR47::	
5434	011142			PRINTF	#TFM47,R5,R4
(9)	011142	010446		MOV	R4,-(SP)
(8)	011144	010546		MOV	R5,-(SP)
(7)	011146	012746	004717	MOV	#TFM47,-(SP)
(6)	011152	012746	000003	MOV	#3,-(SP)
(3)	011156	010600		MOV	SP,R0
(4)	011160	104417		TRAP	C\$PNTF
(4)	011162	062706	000010	ADD	#10,SP
5435	011166			PRINTB	#STM
(7)	011166	012746	005505	MOV	#STM,-(SP)
(6)	011172	012746	000001	MOV	#1,-(SP)
(3)	011176	010600		MOV	SP,R0
(4)	011200	104414		TRAP	C\$PNTB
(4)	011202	062706	000004	ADD	#4,SP
5436	011206			ENDMSG	
(3)	011206			L10034:	
(3)	011206	104423		TRAP	C\$MSG
5437					

CZDMQAO M8207 STATIC DIAG #2 MACY11 30A(1052) 17-JUL-79 09:23 <sup>D 7</sup> PAGE 35-5  
CZDMQA.P11 25-JUN-79 14:07 REPORT CODING SECTION

SEQ 0081

5439  
5440  
5441

.SBTTL REPORT CODING SECTION

CZDMQAO M8207 STATIC DIAG #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 E 7  
REPORT CODING SECTION PAGE 36

SEQ 0082

5443

::+

5445  
5446  
5447  
5448  
5449 011210  
    (3) 011210  
5450  
5456  
5457 011210  
    (4) 011210 000167  
    (3) 011212 000000  
5458

: THE REPORT CODING SECTION CONTAINS THE  
: 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.  
:--

L\$RPT:: BGNRPT  
  
EXIT RPT  
.WORD J\$JMP  
.WORD L10035-2-

CZDMQAO M8207 STATIC DIAG #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 G 7  
REPORT CODING SECTION PAGE 38

SEQ 0084

5466  
5467 011214  
(3) 011214  
(3) 011214 104425  
5468

L10035: ENDRPT  
TRAP CSRPT

CZDMQAO M8207 STATIC DIAG #2    MACY11 30A(1052) 17-JUL-79 09:23 H 7  
CZDMA.P11    25-JUN-79 14:07    REPORT CODING SECTION    PAGE 39

SEQ 0085

5470  
5471

5473  
5474



CZDMQAO MB207 STATIC DIAG #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 J 7  
INITIALIZE SECTION PAGE 40-1

SEQ 0087

5476

.SBTTL INITIALIZE SECTION

5478  
5479  
5480  
5481  
5482  
5483  
5484  
(3)  
5485  
5486  
5487  
5488  
5489  
5490  
5491  
5492  
5493  
5494  
5495  
5496  
5497  
5498  
5499  
(3)  
(3)  
5500  
(2)  
5501  
5502  
(3)  
(3)  
5503  
(2)  
5504  
5505  
(3)  
(3)  
5506  
(2)  
5507  
5508  
(3)  
(3)  
5509  
(2)  
5510  
5511  
5512  
5513  
5514  
5515  
5516  
5517  
5518  
5519  
5520

011216  
011216  
  
011216 012705 002730  
  
011222 010637 002344  
011226 005737 002462  
011232 001011  
011234 013737 000004 002464  
011242 013737 000006 002466  
011250 012737 000001 002462  
011256 013737 002464 000004  
011264 013737 002466 000006  
  
011272  
(3) 011272 012700 000040  
(3) 011276 104447  
011300  
(2) 011300 103414  
  
011302  
(3) 011302 012700 000035  
(3) 011306 104447  
011310  
(2) 011310 103410  
  
011312  
(3) 011312 012700 000036  
(3) 011316 104447  
011320  
(2) 011320 103576  
  
011322  
(3) 011322 012700 000037  
(3) 011326 104447  
011330  
(2) 011330 103003  
  
011332  
011332 012737 177777 002342  
  
011340  
011340 005237 002342  
011344 023737 002342 002012

```
////////////////////////////////////  
// THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED  
// AT THE BEGINNING OF EACH PASS.  
////////////////////////////////////  
  
BGNINIT  
L$INIT::  
  
;INITIALIZE SUBROUTINE STACK  
MOV #SSTACK,R5  
;STORE BASE LEVEL PROGRAM STACK POINTER  
MOV SP,PSTACK  
TST FTIME  
BNE 1$  
MOV @#4,SAVE4  
MOV @#6,SAVE6  
MOV #1,FTIME  
1$: MOV SAVE4,@#4  
MOV SAVE6,@#6  
  
;SEE IF PROGRAM JUST STARTED, BR IF YES  
READEF #EF.START  
MOV #EF.START,R0  
TRAP C$REFG  
BCOMplete NEWST  
BCS NEWST  
;SEE IF THIS IS A NEW PASS, BR IF YES  
READEF #EF.NEW  
MOV #EF.NEW,R0  
TRAP C$REFG  
BCOMplete NEWST  
BCS NEWST  
;SEE IF PROGRAM WAS JUST CONTINUED  
READEF #EF.CONTINUE  
MOV #EF.CONTINUE,R0  
TRAP C$REFG  
BCOMplete ENDIT  
BCS ENDIT  
;SEE IF PROGRAM JUST RESTARTED, BR IF NOT  
READEF #EF.RESTART  
MOV #EF.RESTART,R0  
TRAP C$REFG  
BNCOMplete GETPRM  
BCC GETPRM  
  
NEWST:  
;RESET LOGICAL DEVICE TO -1  
MOV #-1,LOGDEV  
  
;GET UNIBUS ADRS, VECTOR, PRIORITY LEVEL, LINE UNIT, SWITCH  
; PACKS, TEST CONNECTOR INFO. FOR THIS M8200,4,7 (CURRENT LOGICAL  
; DEVICE).  
GETPRM:  
INC LOGDEV  
CMP LOGDEV,L$UNIT
```

5521	011352	002367			BGE	NEWST
5522	011354				GPHARD	LOGDEV,R1
(3)	011354	013700	002342		MOV	LOGDEV,R0
(3)	011360	104442			TRAP	C\$GPHRD
(3)	011362	010001			MOV	R0,R1
5523	011364				BNCOMPLETE	GETPRM
(2)	011364	103365			BCC	GETPRM
5524	011366	012137	002414		MOV	(R1)+,WTYPE
5525						;GET ADDRESS OF M8200,4,7
5526	011372	011137	002516		MOV	(R1),KMCSR
5527						;GET POINTER TO M8200,4,7 CSR HI BYTE
5528	011376	011137	002520		MOV	(R1),KMCSRH
5529	011402	005237	002520		INC	KMCSRH
5530						;GET POINTER TO M8200,4,7 CTL OUT REG
5531	011406	011137	002522		MOV	(R1),KMCTL
5532	011412	062737	000002	002522	ADD	#2,KMCTL
5533						;GET POINTER TO M8200,4,7 PORT REG - SEL 4
5534	011420	011137	002524		MOV	(R1),KMP04
5535	011424	062737	000004	002524	ADD	#4,KMP04
5536						;GET POINTER TO M8200,4,7 PORT REG - SEL 6
5537	011432	012137	002526		MOV	(R1)+,KMP06
5538	011436	062737	000006	002526	ADD	#6,KMP06
5539						;GET POINTER TO RCV VECTOR
5540	011444	011137	002506		MOV	(R1),KMRVEC
5541						;GET POINTER TO RCV PRIORITY LEVEL
5542	011450	011137	002510		MOV	(R1),KMRLVL
5543	011454	062737	000002	002510	ADD	#2,KMRLVL
5544						;GET POINTER TO TX VECTOR
5545	011462	011137	002512		MOV	(R1),KMTVEC
5546	011466	062737	000004	002512	ADD	#4,KMTVEC
5547						;GET POINTER TO TX PRIORITY LEVEL
5548	011474	011137	002514		MOV	(R1),KMTLVL
5549	011500	062737	000006	002514	ADD	#6,KMTLVL
5550						;PUT VECTOR INTO STAT1
5551	011506	016137	000020	002472	MOV	20(R1),RUNINH
5552	011514	012137	002500		MOV	(R1)+,STAT1
5553						;PUT PRIORITY INTO STAT1
5554	011520	052137	002500		BIS	(R1)+,STAT1
5555						;SEE IF NO LINE UNIT, SET BIT IF YES
5556	011524	005711			TST	(R1)
5557	011526	001004			BNE	50000\$
5558	011530	052737	010000	002500	BIS	#BIT12,STAT1
5559	011536	000416			BR	4\$
5560	011540					50000\$:
5561						;SEE IF M8201 LINE UNIT, SET BIT IF YES
5562	011540	021127	000001		CMP	(R1),#1
5563	011544	001001			BNE	50001\$
5564	011546	000412			BR	4\$
5565	011550					50001\$:
5566						;SEE IF M8202 LINE UNIT, SET BIT IF YES
5567	011550	021127	000002		CMP	(R1),#2
5568	011554	001004			BNE	50002\$
5569	011556	052737	020000	002500	BIS	#BIT13,STAT1
5570	011564	000403			BR	4\$
5571	011566					50002\$:
5572						;SET BIT FOR M8203 LINE UNIT

```

5573 011566 052737 100000 002500      BIS      #BIT15,STAT1
5574 011574
5575
5576 011574 056137 000006 002500      BIS      6(R1),STAT1
5577 011602 062701 000002      ADD      #2,R1
5578
5579 011606 012137 002502      ;SET SWITCH PACK #1 IN STAT2 LOW BYTE
5580
5581 011612 111137 002503      MOV      (R1)+,STAT2
5582
5583
5584
5585 011616 000240      ;SET SWITCH PACK #2 IN STAT2 HIGH BYTE
5586 011620 000240      MOV      (R1),STAT2+1
5587
5588 011622 012737 002000 002436      ;INCREMENT LOGICAL UNIT (DEVICE) NUMBER
5589 011630 005037 002432      :
5590 011634 123727 002414 000000      INC      LOGDEV
5591 011642 001425      NOP
5592 011644 123727 002414 000004      NOP
5593 011652 001004      MOV      #2000,MEMSZ
5594 011654 012737 000001 002432      CLR      TYPE
5595 011662 000415      CMPB    WTYPE,#0
5596 011664 012737 007777 002436      BEQ     ENDIT
5597 011672 123727 002414 000006      CMPB    WTYPE,#4      ;KMC?
5598 011700 001003      BNE     5$
5599 011702 012737 000001 002432      MOV      #1,TYPE
5600 011710 013737 002472 002470      BR      ENDIT
5601 011716
5602 011716
5603 011716      5$: MOV      #7777,MEMSZ
(3) 011716      CMPB    WTYPE,#6
(3) 011716 104411      BNE     7$
5604
5605      6$: MOV      #1,TYPE
5606 011720      BR      ENDIT
(3) 011720      ENDIT:
5607
5608 011720 013701 002516      L10036: ENDINIT
5609 011724 012705 000004      TRAP    C$INIT
5610 011730 012737 011762 000004      .EVEN
5611 011736 012737 000240 000006      L$AUTO: BGNAUTO
5612 011744 005711      ;DEVICE DOES NOT HAVE A 'READY'
5613 011746 000240      MOV      KMCSR,R1      ;R1 CONTAINS BASE M8200,4,7 ADDRESS
5614 011750 062701 000002      MOV      #4,R5      ;4 REGISTERS TO BE TESTED
5615 011754 005305      MOV      #2$,4      ;SET OUT TIMEOUT TRAP
5616 011756 001372      MOV      #240,6      ;LEVEL 7
5617 011760 000405      1$: TST      (R1)      ;REFERENCE DEVICE REGISTERS
5618
5619 011762 062706 000004      NOP
5620 011766      ADD      #2,R1      ;NEXT REGISTER
(3) 011766 013700 002342      DEC      R5      ;DEC REGISTER COUNT
(3) 011772 104451      BNE     1$      ;BR IF NOT LAST REGISTER
5621
5622 011774 013737 002464 000004      2$: ADD      #4,SP
5623 012002 013737 002466 000006      DODU    LOGDEV
      MOV      LOGDEV,R0
      TRAP    C$DODU
      3$: MOV      SAVE4,4
      MOV      SAVE6,6
  
```

CZDMQAO M8207 STATIC DIAG #2 MACY11 30A(1052) 17-JUL-79 09:23 N 7  
CZDMQA.P11 25-JUN-79 14:07 INITIALIZE SECTION PAGE 41-3

SEQ 0091

5624 012010  
(3) 012010  
(3) 012010 104461  
5625

L10037: ENDAUTO  
TRAP C\$AUTO

5627  
5628  
5629  
5630  
5631  
5632  
5633  
5634 012012  
    (3) 012012  
5635 012012  
    (3) 012012 104433  
5636  
5637 012014  
    (3) 012014  
    (3) 012014 104412  
5638  
5639  
5640  
5641  
5642

.SBTTL CLEANUP CODING SECTION

:///  
:// THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED  
:// AT THE END OF EACH PASS.  
:///

                  BGNCLN  
L\$CLEAN: :  
                  BRESET  
                  TRAP     C\$RESET

                  ENDCLN  
L10040: :  
                  TRAP     C\$CLEAN

5644  
5645  
5646  
5647  
5648  
5649  
5650  
5651 012016  
(3) 012016  
5652  
5653 012016 104433  
(3) 012016 104433  
5654 012020  
(3) 012020  
(3) 012020 104453  
5655  
5656  
5657  
5658  
5659

```
.SBTTL DROP UNIT SECTION  
://///  
:/ THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE  
:/ TO NO LONGER BE TESTED.  
://///  
BGNDU  
L$DU::  
:ISSUE UNIBUS RESET TO CLEAN UP  
BRESET  
TRAP C$RESET  
ENDDU  
L10041:  
TRAP C$DU
```

5661  
5662  
5663  
5664  
5665  
5666  
5667  
5668  
5669 012022  
(3) 012022  
5670 012022  
(3) 012022  
(3) 012022 104452  
5671  
5672  
5673  
5674  
5675  
5676

.SBTTL ADD UNIT SECTION

:/ THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE  
:/ TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING. IF  
:/ 'EF.AUNIT' IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.

L\$AU:: BGNAU  
L10042: ENDAU  
TRAP C\$AU



```
5678 .SBTTL HARDWARE TESTS
5679
5680
5681
5682 ;START OF CODE BLOCK WHICH IS USED AS DATA
5683 012024 ROMMAP:
5684
5685 012024 BADHEAD
(2) ;***** TEST 1 *****
5686 ;*VERIFY THAT REFERENCING UNIBUS DEVICE REGISTERS
5687 ;*DOES NOT CAUSE A TIME OUT TRAP
5688 012024 BADHEAD
(2) ;***** TEST 1 *****
5689
5690 012024 BGNTST
(3) 012024 T1::
5691 012024 013701 002516 MOV KMCSR,R1 ;R1 CONTAINS BASE M8200,4,7 ADDRESS
5692 012030 012705 000004 MOV #4,R5 ;4 REGISTERS TO BE TESTED
5693 012034 012737 012072 000004 MOV #2$,4 ;SET OUT TIMEOUT TRAP
5694 012042 012737 000240 000006 MOV #240,6 ;LEVEL 7
5695 012050 005711 1$: TST (R1) ;REFERENCE DEVICE REGISTERS
5696 012052 000240 NOP
5697 012054 ESCAPE TST
(3) 012054 104410 TRAP C$ESCAPE
(3) 012056 000054 .WORD L10043-.
5698 012060 062701 000002 ADD #2,R1 ;NEXT REGISTER
5699 012064 005305 DEC R5 ;DEC REGISTER COUNT
5700 012066 001370 BNE 1$ ;BR IF NOT LAST REGISTER
5701 012070 000410 BR 3$
5702
5703 012072 062706 000004 2$: ADD #4,SP
5704 012076 ERROR 36 ;TIME OUT ERROR
(5) 012102 104455 TRAP C$ERDF
(6) 012104 000044 .WORD 36
(6) 012106 005041 .WORD EMO
(6) 012110 010472 .WORD ERR36
5705
5706 012112 013737 002464 000004 3$: MOV SAVE4,4
5707 012120 013737 002466 000006 MOV SAVE6,6
5708 012126 ESCAPE TST
(3) 012126 104410 TRAP C$ESCAPE
(3) 012130 000002 .WORD L10043-.
5709
5710 012132 ENDTST
(3) 012132 L10043:
(3) 012132 104401 TRAP C$ETST
5711 .EVEN
5712
5713
5714 012134 BADHEAD
(2) ;***** TEST 2 *****
5715 ;*TEST OF BR RIGHT SHIFT
5716 ;*VERIFY THAT A DEST OF BR RSH (011) OF A MICRO-INSTRUCTION
5717 ;*SHIFTS THE RESULTING BR DATA RIGHT ONCE.
5718 012134 BADHEAD
(2) ;***** TEST 2 *****
```

```

5719
5720 012134          BGNTST
(3) 012134          T2::
5721
5722 012134          MSTCLR          ;R1 CONTAINS BASE M8200,4,7 ADDRESS
5723 012140 013701 002516  MOV      KMCSR,R1      ;MASTER CLEAR M8200,4,7
5724 012144 005011          CLR      (R1)          ;R1 = M8200,4,7 BASE ADDRESS
5725 012146 012705 052525  MOV      #52525,R5     ;CLEAR SELO
5726 012152 010561 000004  MOV      R5,4(R1)     ;START WITH 125
5727 012156          ROMCLK          ;PORT4 125
(1) 012156 004537 003030  JSR      R5,..ROMCLK  ;NEXT WORD IS INSTRUCTION
5728 012162 120500          120500      ;CLOCK INSTRUCTION
5729 012164          ROMCLK          ;PORT4 TO BR-REG
(1) 012164 004537 003030  JSR      R5,..ROMCLK  ;NEXT WORD IS INSTRUCTION
5730 012170 061620          061620      ;CLOCK INSTRUCTION
5731 012172          ROMCLK          ;BR RSH BR, SHIFT BR RIGHT
(1) 012172 004537 003030  JSR      R5,..ROMCLK  ;NEXT WORD IS INSTRUCTION
5732 012176 061225          061225      ;CLOCK INSTRUCTION
5733 012200 006005          ROR      R5           ;PORT5 BR
5734 012202 005004          CLR      R4           ;R5 = "EXPECTED"
5735 012204 116104 000005  MOVB    5(R1),R4      ;R4 = 'FOUND'
5736 012210 120504          CMPB    R5,R4        ;DID BR SHIFT RIGHT ONCE?
5737 012212 001410          BEQ     1$           ;BR IF YES
5738 012214          ERROR    12           ;BR RIGHT SHIFT ERROR
(5) 012220 104455          TRAP    C$ERDF
(6) 012222 000014          .WORD  12
(6) 012224 005041          .WORD  EMO
(6) 012226 007276          .WORD  ERR12
5739
5740 012230          ESCAPE TST          ;SHOULD BE 52
(3) 012230 104410          TRAP    C$ESCAPE
(3) 012232 000044          .WORD  L10044-
5741 012234          1$:
5742 012234          ROMCLK          ;NEXT WORD IS INSTRUCTION
(1) 012234 004537 003030  JSR      R5,..ROMCLK  ;CLOCK INSTRUCTION
5743 012240 061620          061620      ;BR RSH BR, SHFT BR RIGHT AGAIN
5744 012242          ROMCLK          ;NEXT WORD IS INSTRUCTION
(1) 012242 004537 003030  JSR      R5,..ROMCLK  ;CLOCK INSTRUCTION
5745 012246 061225          061225      ;PORT5 BR
5746 012250 006005          ROR      R5           ;R5 = "EXPECTED"
5747 012252 116104 000005  MOVB    5(R1),R4      ;R4 = 'FOUND'
5748 012256 120504          CMPB    R5,R4        ;DID BR SHIFT RIGHT?
5749 012260 001406          BEQ     2$           ;BR IF YES
5750 012262          ERROR    12           ;BR RIGHT SHIFT ERROR
(5) 012266 104455          TRAP    C$ERDF
(6) 012270 000014          .WORD  12
(6) 012272 005041          .WORD  EMO
(6) 012274 007276          .WORD  ERR12
5751
5752 012276          2$:
5753 012276          ENDTST
(3) 012276          L10044:
(3) 012276 104401          TRAP    C$ETST
5754
5755 012300          BADHEAD
(2)
;***** TEST 3 *****

```

```
5756                                     ;*IOP CRAM WRITE/READ TEST
5757                                     ;*FLOAT A 1 THROUGH EACH CRAM LOCATION,
5758 012300                               BADHEAD
(2)                                     ;***** TEST 3 *****
5759
5760 012300                               BGNTST
(3) 012300                               T3::
5761 012300                               MACEX
(1)                                     ;DO NOT DO TEST IF M8200
(4) 012306 104432                         TRAP C$EXIT
(4) 012310 000116                         .WORD L10045-.
5762 012312                               MYINT
(1) 012312 013701 002516                 MOV KMCSR,R1
5763                                     ;RECORD DEVICE ADDR.
5764 012316 005037 002434                 CLR MRO
5765 012322 012702 000001                 ADR4: MOV #1,R2
5766 012326                               ADR5:
5767 012326                               BGNSEG
(3) 012326 104404                         TRAP C$BSEG
5768 012330 012711 002000                 3$: MOV #BIT10,(R1)
5769 012334 013761 002434 000004         MOV MRO,4(R1)
5770 012342 010261 000006                 MOV R2,6(R1)
5771 012346 052711 020000                 MOV #BIT13,(R1)
5772 012352 016104 000006                 BIS #BIT13,(R1)
5773 012356 020204                         MOV 6(R1),R4
5774 012360 001410                         CMP R2,R4
5775 012362                               BEQ 4$
(5) 012366 104455                         ERROR 1
(6) 012370 000001                         TRAP C$ERDF
(6) 012372 005041                         .WORD 1
(6) 012374 005710                         .WORD EMO
5776 012376                               .WORD ERR1
(3) 012376 104410                         ESCAPE SEG
(3) 012400 000002                         TRAP C$ESCAPE
5777 012402                               .WORD 10000$-.
(3) 012402                               4$: ENDSEG
(3) 012402 104405                         10000$:
5778 012404 000241                         TRAP C$ESEG
5779 012406 006102                         CLC
5780 012410 001346                         ROL R2
5781 012412 005237 002434                 BNE ADR5
5782 012416 023737 002436 002434         INC MRO
5783 012424 001336                         CMP MEMSZ,MRO
5784 012426                               BNE ADR4
5785 012426                               5$:
(3) 012426                               ENDTST
(3) 012426 104401                         L10045:
5786                                     TRAP C$ETST
5787 012430                               BADHEAD
(2)                                     ;***** TEST 4 *****
5788                                     ;*IOP CRAM WRITE/READ TEST
5789                                     ;*FLOAT A 0 THROUGH EACH CRAM LOCATION
5790 012430                               BADHEAD
(2)                                     ;***** TEST 4 *****
5791
5792 012430                               BGNTST
```

```

(3) 012430
5793 012430
(1)
(4) 012436 104432
(4) 012440 000126
5794 012442
(1) 012442 013701 002516
5795 012446
5796 012452 005037 002434
5797 012456 012702 000001
5798 012462
5799 012462
(3) 012462 104404
5800 012464 005102
5801 012466 012711 002000
5802 012472 013761 002434 000004
5803 012500 010261 000006
5804 012504 052711 020000
5805 012510 016104 000006
5806 012514 020204
5807 012516 001410
5808 012520
(5) 012524 104455
(6) 012526 000001
(6) 012530 005041
(6) 012532 005710
5809 012534
(3) 012534 104410
(3) 012536 000002
5810 012540
(3) 012540
(3) 012540 104405
5811 012542 005102
5812 012544 000241
5813 012546 006102
5814 012550 001344
5815 012552 005237 002434
5816 012556 023737 002436 002434
5817 012564 001334
5818 012566
5819 012566
(3) 012566
(3) 012566 104401
5820
5821 012570
(2)
5822
5823
5824
5825 012570
(2)
5826
5827 012570
(3) 012570
5828 012570
(1)
  
```

```

T4::
MACEX
:DO NOT DO TEST IF M8200
TRAP C$EXIT
.WORD L10046-.
MYINT
MOV KMCSR,R1 ;RECORD DEVICE ADDR.
MSTCLR ;MASTER CLEAR M8200,4,7
CLR MRO ;MRO = CRAM ADDRESS
MOV #1,R2 ;R2 = WRITE DATA
ADR1:
ADR2:
BGNSEG
TRAP C$BSEG
COM R2 ;MAKE IT A FLOATING ZERO
MOV #BIT10,(R1) ;SET ROMO
MOV MRO,4(R1) ;WRITE ADDRESS TO SEL4
MOV R2,6(R1) ;LOAD SEL6 WITH WRITE DATA
BIS #BIT13,(R1) ;WRITE SEL6 INTO CRAM
MOV 6(R1),R4 ;READ CRAM INTO 'FOUND'
CMP R2,R4 ;IS DATA CORRECT?
BEQ 4$ ;BR IF OK
ERROR 1 ;ERROR
TRAP C$ERDF
.WORD 1
.WORD EMO
.WORD ERR1
ESCAPE SEG
TRAP C$ESCAPE
.WORD 10000$-.
4$:
10000$:
ENDSEG
TRAP C$ESEG
COM R2 ;BACK TO FLOATING ONE
CLC ;CLEAR CARRY
ROL R2 ;SHIFT WRITE DATA
BNE ADR2 ;BR IF NOT DONE THIS ADDRESS
INC MRO ;BUMP TO NEXT CRAM ADDRESS
CMP MEMSZ,MRO ;DONE YET?
BNE ADR1 ;BR IF NO
5$:
ENDTST
L10046:
TRAP C$ETST
BADHEAD
:***** TEST 5 *****
:*IOP CRAM DUAL ADDRESSING TEST
:*WRITE EACH ADDRESS INTO ITSELF, READ EACH
:*ADDRESS TO VERIFY CORRECT ADDRESSING
BADHEAD
:***** TEST 5 *****
BGNTST
T5::
MACEX
:DO NOT DO TEST IF M8200
  
```

```
(4) 012576 104432 TRAP C$EXIT
(4) 012600 000230 .WORD L10047-.
5829 012602 MYINT
(1) 012602 013701 002516 MOV KMCSR,R1 ;RECORD DEVICE ADDR.
5830 ;R1 CONTAINS BASE M8200,4,7 ADDRESS
5831 012606 MSTCLR ;MASTER CLEAR M8200,4,7
5832 012612 005037 002434 CLR MRO ;MRO =CRAM ADDRESS
5833 012616 BGNSEG
(3) 012616 104404 TRAP C$BSEG
5834 012620 013702 002434 1$: MOV MRO,R2 ;SAVE R2 FOR TYPEOUT
5835 012624 012711 002000 MOV #BIT10,(R1) ;SET ROMO
5836 012630 013761 002434 000004 MOV MRO,4(R1) ;WRITE ADDRESS TO SEL4
5837 012636 013761 002434 000006 MOV MRO,6(R1) ;LOAD SEL6 WITH WRITE DATA
5838 012644 052711 020000 BIS #BIT13,(R1) ;WRITE CRAM
5839 012650 SKIP06 15$ ;IF M8206,SKIP NEXT INSTR.
(1) ;GOTO 15$ IF M8206
5840 012660 005061 000006 CLR 6(R1) ;CLEAR SEL 6
5841 012664 15$:
5842 012664 016104 000006 MOV 6(R1),R4 ;SHOULD READ BACK OWN ADDRESS
5843 012670 023704 002434 CMP MRO,R4 ;IS DATA CORRECT?
5844 012674 001410 BEQ 2$ ;BR IF YES
5845 012676 ERROR 1 ;DATA ERROR
(5) 012702 104455 TRAP C$ERDF
(6) 012704 000001 .WORD 1
(6) 012706 005041 .WORD EMO
(6) 012710 005710 .WORD ERR1
5846 012712 ESCAPE SEG
(3) 012712 104410 TRAP C$ESCAPE
(3) 012714 000002 .WORD 10000$-.
5847 012716 2$:
(3) 012716 10000$: ENDSEG
(3) 012716 104405 TRAP C$ESEG
5848 012720 BGNSEG
(3) 012720 104404 TRAP C$BSEG
5849 012722 005237 002434 INC MRO ;BUMP TO NEXT ADDRESS
5850 012726 023737 002436 002434 CMP MEMSZ,MRO ;DONE WRITING YET?
5851 012734 001331 BNE 1$ ;BR IF NO
5852 012736 005037 002434 CLR MRO ;RESTART AT ADDRESS 0
5853 012742 013702 002434 3$: MOV MRO,R2 ;SAVE R2 FOR TYPEOUT
5854 012746 012711 002000 MOV #BIT10,(R1) ;SET ROMO
5855 012752 013761 002434 000004 MOV MRO,4(R1) ;SEL4 = CRAM ADDRESS
5856 012760 016104 000006 MOV 6(R1),R4 ;READ CRAM INTO 'FOUND'
5857 012764 023704 002434 CMP MRO,R4 ;IS DATA CORRECT?
5858 012770 001411 BEQ 4$ ;BR IF YES
5859 012772 ERROR 2 ;DUAL ADDRESSING ERROR
(5) 012776 104455 TRAP C$ERDF
(6) 013000 000002 .WORD 2
(6) 013002 005041 .WORD EMO
(6) 013004 006036 .WORD ERR2
5860 013006 ESCAPE SEG
(3) 013006 104410 TRAP C$ESCAPE
(3) 013010 000002 .WORD 10001$-.
5861 013012 10001$:
(3) 013012 104405 TRAP C$ESEG
5862 013014 4$: ;LOOP TO 3$ IF SW09=1
```

5863	013014	005237	002434		INC	MRO	:BUMP TO NEXT ADDRESS
5864	013020	023737	002436	002434	CMP	MEMSZ,MRO	:DONE WRITING YET?
5865	013026	001345			BNE	3\$	:BR IF NO
5866	013030						
5867	013030						
(3)	013030						
(3)	013030	104401			TRAP	C\$ETST	
5868							
5869							
5870	013032						
(2)							
5871							
5872							
5873	013032						
(2)							
5874							

5\$:  
ENDTST  
L10047:

BADHEAD  
:\*\*\*\*\* TEST 6 \*\*\*\*\*  
:\*IOP MAIN MEMORY TEST  
:\*FLOAT A 1 THROUGH ALL MAIN MEMORY LOCATIONS  
BADHEAD  
:\*\*\*\*\* TEST 6 \*\*\*\*\*

5876	013032					BGNTST					
(3)	013032					T6::					
5877	013032					MYINT					
(1)	013032	013701	002516			MOV	KMCSR,R1				
5878											
5879	013036					MSTCLR					
5880	013042	005037	002406			CLR	FLAG				
5881	013046	012737	000001	002434	1\$:	MOV	#1,MRO				
5882	013054	042737	003777	013110	65\$:	BIC	#3777,66\$				

;RECORD DEVICE ADDR.  
;R1 CONTAINS BASE M8200,4,7 ADDRESS  
;MASTER CLEAR M8200,4,7  
;START WITH ADDRESS 0  
;START WITH BIT 0  
;CLEAR ADDRESS FIELD OF INSTRUCTION

CZDMQAO M8207 STATIC DIAG #2 MACY11 30A(1052) 17-JUL-79 09:23 L 8  
CZDMQA.P11 25-JUN-79 14:07 HARDWARE TESTS PAGE 43

SEQ 0102

5884 013062 042737 000037 013116 BIC #37.68\$ ;CLEAR ADDRESS FIELD OF INSTRUCTION



```

5886 013070 153737 002406 013110      BISB  FLAG,66$      ;ADD ADDRESS TO INSTRUCTION7
5887 013076 153737 002407 013116      BISB  FLAG+1,68$    ;ADD ADDRESS TO INSTRUCTION
5888 013104      ROMCLK      ;NEXT WORD IS INSTRUCTION,
(1) 013104 004537 003030      JSR   R5,..ROMCLK  ;CLOCK INSTRUCTION
5889 013110 010000      66$: 010000      ROMCLK
5890 013112      JSR   R5,..ROMCLK  ;CLOCK INSTRUCTION
(1) 013112 004537 003030      JSR   R5,..ROMCLK  ;CLOCK INSTRUCTION
5891 013116 004000      68$: 004000      ;LOAD MAR HI
5892 013120 013761 002434 000004      MOV   MRO,4(R1)    ;WRITE PATTERN IN PORT4
5893 013126      ROMCLK      ;NEXT WORD IS INSTRUCTION,
(1) 013126 004537 003030      JSR   R5,..ROMCLK  ;CLOCK INSTRUCTION
5894 013132 122500      122500      ;MOVE PORT4 TO MEMORY
5895 013134      ROMCLK      ;NEXT WORD IS INSTRUCTION,
(1) 013134 004537 003030      JSR   R5,..ROMCLK  ;CLOCK INSTRUCTION
5896 013140 040620      040620      ;MOVE MEMORY TO BR
5897 013142      ROMCLK      ;NEXT WORD IS INSTRUCTION,
(1) 013142 004537 003030      JSR   R5,..ROMCLK  ;CLOCK INSTRUCTION
5898 013146 061225      61225      ;MOVE BR TO PORT5
5899 013150 013705 002434      MOV   MRO,R5       ;PUT 'EXPECTED' IN R5
5900 013154 116104 000005      MOVB  5(R1),R4     ;PUT 'FOUND' IN R4
5901 013160 120504      CMPB  R5,R4        ;DATA CORRECT?
5902 013162 001410      BEQ   67$          ;BR IF YES
5903 013164      ERROR 6           ;DATA ERROR
(5) 013170 104455      TRAP  C$ERDF
(6) 013172 000006      .WORD 6
(6) 013174 005041      .WORD EMO
(6) 013176 006556      .WORD ERR6
5904 013200      ESCAPE TST
(3) 013200 104410      TRAP  C$ESCAPE
(3) 013202 000030      .WORD L10050-.
5905 013204      67$:
5906 013204 000241      CLC
5907 013206 106137 002434      ROLB  MRO
5908 013212 001320      BNE   65$          ;SW09=1?
5909 013214      BREAK
(3) 013214 104422      TRAP  C$BRK        ;CLEAR CARRY
5910 013216 005237 002406      INC   FLAG         ;SHIFT BIT IN MRO
5911 013222 023737 002436 002406      CMP   MEMSZ,FLAG   ;DONE IF MRO=0
5912 013230 001306      BNE   1$           ;NEXT ADDRESS
5913 013232      2$:
5914 013232      ENDTST
(3) 013232      L10050:
(3) 013232 104401      TRAP  C$ETST
5915
5916 013234      BADHEAD
(2)
5917      ;***** TEST 7 *****
5918      ;*IOP MAIN MEMORY TEST
5919 013234      ;*FLOAT A 0 THROUGH ALL MAIN MEMORY LOCATIONS
(2)
5920      BADHEAD
5921      ;***** TEST 7 *****
(3) 013234      BGNTST
(3) 013234      T7::
5922 013234      MYINT
(1) 013234 013701 002516      MOV   KMCSR,R1
5923      ;RECORD DEVICE ADDR.
      ;R1 CONTAINS BASE M8200,4,7 ADDRESS
  
```

```

5924 013240          MSTCLR          ;MASTER CLEAR M8200,4,7
5925 013244 005037 002406          CLR          FLAG          ;START WITH ADDRESS 0
5926 013250 012737 000001 002434 1$: MOV          #1,MRO        ;START WITH BIT 0
5927 013256 005137 002434          COM          MRO          ;CHANGE TO FLOATING 0
5928 013262 042737 003777 013316 64$: BIC          #3777,66$    ;CLEAR ADDRESS FIELD OF INSTRUCTION
5929 013270 042737 000037 013324 65$: BIC          #37,68$     ;CLEAR ADDRESS FIELD OF INSTRUCTION
5930 013276 153737 002406 013316  BISB          FLAG,66$     ;ADD ADDRESS TO INSTRUCTION
5931 013304 153737 002407 013324  BISB          FLAG+1,68$   ;ADD ADDRESS TO INSTRUCTION
5932 013312          ROMCLK          ;NEXT WORD IS INSTRUCTION,
(1) 013312 004537 003030          JSR          R5,ROMCLK     ;CLOCK INSTRUCTION
5933 013316 010000          66$: 010000          ;LOAD MAR LO WITH ADDRESS IN FLAG
5934 013320          ROMCLK          ;NEXT WORD IS INSTRUCTION,
(1) 013320 004537 003030          JSR          R5,ROMCLK     ;CLOCK INSTRUCTION
5935 013324 004000          68$: 004000          ;LOAD MAR HI
5936 013326 013761 002434 000004  MOV          MRO,4(R1)     ;WRITE PATTERN IN PORT4
5937 013334          ROMCLK          ;NEXT WORD IS INSTRUCTION,
(1) 013334 004537 003030          JSR          R5,ROMCLK     ;CLOCK INSTRUCTION
5938 013340 122500          122500          ;MOVE PORT4 TO MEMORY
5939 013342          ROMCLK          ;NEXT WORD IS INSTRUCTION,
(1) 013342 004537 003030          JSR          R5,ROMCLK     ;CLOCK INSTRUCTION
5940 013346 040620          040620          ;MOVE MEMORY TO BR
5941 013350          ROMCLK          ;NEXT WORD IS INSTRUCTION,
(1) 013350 004537 003030          JSR          R5,ROMCLK     ;CLOCK INSTRUCTION
5942 013354 061225          61225           ;MOVE BR TO PORT5
5943 013356 013705 002434          MOV          MRO,R5        ;PUT 'EXPECTED' IN R5
5944 013362 116104 000005          MOVB         5(R1),R4      ;PUT 'FOUND' IN R4
5945 013366 120504          CMPB         R5,R4        ;DATA CORRECT?
5946 013370 001406          BEQ          67$         ;BR IF YES
5947 013372          ERROR          6           ;DATA ERROR
(5) 013376 104455          TRAP         C$ERDF
(6) 013400 000006          .WORD        6
(6) 013402 005041          .WORD        EMO
(6) 013404 006556          .WORD        ERR6
5948 013406          67$: ESCAPE TST
(3) 013406 104410          TRAP         C$ESCAPE
(3) 013410 000034          .WORD        L10051-
5949 013412 005137 002434          COM          MRO          ;CHANGE TO FLOATING 1
5950 013416 000241          CLC          ;CLEAR CARRY
5951 013420 106137 002434          ROLB         MRO          ;SHIFT BIT IN MRO
5952 013424 001314          BNE          64$         ;DONE IF MRO=0
5953 013426          BREAK
(3) 013426 104422          TRAP         C$BRK
5954 013430 005237 002406          INC          FLAG          ;NEXT ADDRESS
5955 013434 023737 002436 002406  CMP          MEMSZ,FLAG    ;LAST ADDRESS?
5956 013442 001302          BNE          1$         ;BR IF NO
5957 013444          2$:
5958 013444          ENDTST
(3) 013444          L10051:
(3) 013444 104401          TRAP         C$ETST
5959
5960 013446          BADHEAD
(2)
5961          ;***** TEST 8 *****
5962          ;*IOP MAIN MEMORY DUAL ADDRESSING TEST
5963          ;*LOAD EACH MEMORY LOCATION WITH ITS OWN ADDRESS
5964 013446          ;*READ BACK EACH LOCATION TO VERIFY CORRECT ADDRESSING
          BADHEAD
  
```

:\*\*\*\*\* TEST 8 \*\*\*\*\*

```

(2)
5965
5966 013446          BGNTST
(3) 013446          T8::
5967 013446          MYINT
(1) 013446 013701 002516  MOV      KMCSR,R1          ;RECORD DEVICE ADDR.
5968                                     ;R1 CONTAINS BASE M8200,4,7 ADDRESS
5969 013452          MSTCLR          ;MASTER CLEAR M8200,4,7
5970 013456 005037 002406  CLR      FLAG          ;START WITH ADDRESS 0
5971 013462 013702 002406  1$:  MOV     FLAG,R2        ;PUT DATA IN R2
5972 013466 042737 003777 013522  BIC     #3777,2$      ;CLEAR ADDRESS FIELD OF INSTRUCTION
5973 013474 042737 000037 013530  BIC     #37,7$       ;CLEAR ADDRESS FIELD OF INSTRUCTION
5974 013502 153737 002406 013522  BISB   FLAG,2$       ;ADD ADDRESS TO INSTRUCTION
5975 013510 153737 002407 013530  BISB   FLAG+1,7$    ;ADD ADDRESS TO INSTRUCTION
5976 013516          ROMCLK          ;NEXT WORD IS INSTRUCTION,
(1) 013516 004537 003030  JSR     R5,.,ROMCLK  ;CLOCK INSTRUCTION
5977 013522 010000          2$:  010000          ;LOAD MAR LO
5978 013524          ROMCLK          ;NEXT WORD IS INSTRUCTION,
(1) 013524 004537 003030  JSR     R5,.,ROMCLK  ;CLOCK INSTRUCTION
5979 013530 004000          7$:  004000          ;LOAD MAR HI
5980 013532 010261 000004  MOV     R2,4(R1)
5981 013536          ROMCLK          ;NEXT WORD IS INSTRUCTION,
(1) 013536 004537 003030  JSR     R5,.,ROMCLK  ;CLOCK INSTRUCTION
5982 013542 122500          ;MOVE PORT4 TO MEMORY
5983 013544          ROMCLK          ;NEXT WORD IS INSTRUCTION,
(1) 013544 004537 003030  JSR     R5,.,ROMCLK  ;CLOCK INSTRUCTION
5984 013550 040620          ;MOVE MEMORY TO THE BR
5985 013552          ROMCLK          ;NEXT WORD IS INSTRUCTION,
(1) 013552 004537 003030  JSR     R5,.,ROMCLK  ;CLOCK INSTRUCTION
5986 013556 061225          ;MOVE BR TO PORT5
5987 013560 010205          MOV     R2,R5          ;PUT 'EXPECTED' IN R5
5988 013562 116104 000005  MOVB   5(R1),R4      ;PUT 'FOUND' IN R4
5989 013566 120504          CMPB   R5,R4          ;DATA CORRECT?
5990 013570 001406          BEQ    3$             ;BR IF YES
5991 013572          ERROR          ;DATA ERROR
(5) 013576 104455          TRAP   C$ERDF
(6) 013600 000006          .WORD 6
(6) 013602 005041          .WORD EMO
(6) 013604 006556          .WORD ERR6
5992 013606          3$:  ESCAPE TST
(3) 013606 104410          TRAP   C$ESCAPE
(3) 013610 000156          .WORD L10052-.
5993 013612 005237 002406  INC     FLAG          ;NEXT ADDRESS
5994 013616 023737 002436 002406  CMP    MEMSZ,FLAG    ;LAST ADDRESS?
5995 013624 001316          BNE   1$             ;BR IF NO
5996 013626 012737 013640 002340  MOV    #4$,LOCK      ;NEW SCOPE 1
5997 013634 005037 002406  CLR    FLAG          ;RESTART AT ADDRESS 0
5998 013640 013702 002406  4$:  MOV     FLAG,R2        ;PUT DATA IN R2
5999 013644 042737 003777 013672  BIC    #3777,5$      ;CLEAR ADDRESS FIELD OF INSTRUCTION
6000 013652 042737 000037 013700  BIC    #37,8$       ;CLEAR ADDRESS FIELD OF INSTRUCTION
6001 013660 153737 002406 013672  BISB   FLAG,5$       ;ADD ADDRESS TO INSTRUCTION
6002 013666          ROMCLK          ;NEXT WORD IS INSTRUCTION, ROMCLK PC+5304
(1) 013666 004537 003030  JSR    R5,.,ROMCLK  ;CLOCK INSTRUCTION
6003 013672 010000          5$:  010000          ;LOAD THE MAR LO
6004 013674          ROMCLK          ;NEXT WORD IS INSTRUCTION,
(1) 013674 004537 003030  JSR    R5,.,ROMCLK  ;CLOCK INSTRUCTION

```

```
6005 013700 004000      8$: 004000      ;LOAD MAR HI
6006 013702      ROMCLK      ;NEXT WORD IS INSTRUCTION,
(1) 013702 004537 003030 JSR R5,,ROMCLK ;CLOCK INSTRUCTION
6007 013706 040620      040620      ;MOVE MEMORY TO THE BR
6008 013710      ROMCLK      ;NEXT WORD IS INSTRUCTION,
(1) 013710 004537 003030 JSR R5,,ROMCLK ;CLOCK INSTRUCTION
6009 013714 061225      61225      ;MOV BR TO PORT5
6010 013716 010205      MOV R2,R5      ;PUT 'EXPECTED' IN R5
6011 013720 116104 000005 MOVB 5(R1),R4  ;PUT 'FOUND' IN R4
6012 013724 120504      CMPB R5,R4     ;DATA CORRECT?
6013 013726 001406      BEQ 6$        ;BR IF YES
6014 013730      ERROR 6      ;ADDRESSING ERROR
(5) 013734 104455      TRAP C$ERDF
(6) 013736 000006      .WORD 6
(6) 013740 005041      .WORD EMO
(6) 013742 006556      .WORD ERR6
6015 013744      6$: ESCAPE TST
(3) 013744 104410      TRAP C$ESCAPE
(3) 013746 000020      .WORD L10052-.
6016 013750      BREAK
(3) 013750 104422      TRAP C$BRK
6017 013752 005237 002406 INC FLAG      ;NEXT ADDRESS
6018 013756 023737 002436 002406 CMP MEMSZ,FLAG ;IS IT THE LAST
6019 013764 001325      BNE 4$        ;BR IF NO
6020 013766
6021 013766      9$:
(3) 013766      ENDTST
(3) 013766 104401      L10052: TRAP C$ETST
6022
6023 013770      BADHEAD
(2)          ;***** TEST 9 *****
6024          ;*IOP MAR TEST
6025          ;*PERFORM DUAL ADDRESSING TEST
6026          ;*USING MAR AUTO-INC FEATURE
6027 013770      BADHEAD
(2)          ;***** TEST 9 *****
6028
6029 013770      BGNTST
(3) 013770      T9::
6030 013770      K4ONLY      ;FOR 4K CPUS ONLY.
(1)          ;DO NOT DO TEST IF M8200, OR M8204
(4) 014000 104432      TRAP C$EXIT
(4) 014002 000326      .WORD L10053-.
6031 014004      MYINT
(1) 014004 013701 002516 MOV KMCSR,R1 ;RECORD DEVICE ADDR.
6032          ;R1 CONTAINS BASE M8200,4,7 ADDRESS
6033 014010      MSTCLR      ;MASTER CLEAR M8200,4,7
6034 014014 005002      CLR R2        ;START WITH A ZERO
6035 014016 013703 002436 MOV MEMSZ,R3  ;GET MEMORY SIZE
6036 014022 005203      INC R3        ;STOP ADDR=MEMSZ+1
6037 014024      ROMCLK      ;NEXT WORD IS INSTRUCTION,
(1) 014024 004537 003030 JSR R5,,ROMCLK ;CLOCK INSTRUCTION
6038 014030 010000      010000      ;LOAD MAR WITH A ZERO
6039 014032      CLRMAR
(2) 014032 004537 003030 JSR R5,,ROMCLK ;CLOCK INSTRUCTION
6040 014040 010261 000004 1$: MOV R2,4(R1) ;WRITE DATA TO PORT4
```

6041	014044			ROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	014044	004537	003030	JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
6042	014050	136500		136500		:MEM PORT4, AUTO-INC MAR
6043	014052	005202		INC	R2	:INCREMENT DATA
6044	014054	020302		CMP	R3,R2	:DONE YET?
6045	014056	001370		BNE	1\$	:BR IF NO
6046	014060	005002		CLR	R2	:RESTART WITH A ZERO
6047	014062			ROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	014062	004537	003030	JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
6048	014066	010000		010000		:LOAD MAR WITH A ZERO
6049	014070			CLRMAR		
(2)	014070	004537	003030	JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
6050	014076					
6051	014076					
(1)	014076	004537	003064	SROMCLK		:NEXT WORD IS INSTRUCTION,
6052	014102	055224		JSR	R5,.SROMCLK	
6053	014104	010205		055224		:MOVE MEM TO PORT4
6054	014106	116104	000004	MOV	R2,R5	:PUT 'EXPECTED' IN R5
6055	014112	120504		MOVB	4(R1),R4	:PUT 'FOUND' IN R4
6056	014114	001406		CMPB	R5,R4	:DATA CORRECT?
6057	014116			BEQ	3\$	:BR IF YES
(5)	014122	104455		ERROR	11	:MAR ERROR
(6)	014124	000013		TRAP	C\$ERDF	
(6)	014126	005041		.WORD	11	
(6)	014130	007154		.WORD	EMO	
6058	014132			.WORD	ERR11	
(1)	014132	004537	003064	SROMCLK		
6059	014136	000000		JSR	R5,.SROMCLK	
6060	014140	005004		0		:DUMP NOP INSTR. TO CLK AUTO INC IN MAR.
6061	014142			CLR	R4	
(1)	014142	004537	003030	ROMCLK		:READ IBUS* <15> (MAR HIGH)
6062	014146	121325		JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
6063				121325		:MAR HIGH _POT 5
6064	014150					
(1)	014150	004537	003030	ROMCLK		:READ IBUS* <14> (MAR LOW)
6065	014154	121304		JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
6066	014156	016104	000004	121304		:ADD TO MAR HIGH.
6067	014162	042704	160000	MOV	4(R1),R4	
6068	014166	005202		BIC	#160000,R4	
6069	014170	020237	002436	INC	R2	
6070	014174	001002		CMP	R2,MEMSZ	
6071	014176	052702	010000	BNE	35\$	
6072	014202			BIS	#10000,R2	:IF AT HIGH LIMIT,ADD IN OVERFLOW BIT.
6073	014202	020204				
6074	014204	001406		CMP	R2,R4	:ADDR. OK?
6075	014206			BEQ	4\$	
(5)	014212	104455		ERROR	11	:ERROR MAR ADDR. BAD IN IBUS <14>AND <15>
(6)	014214	000013		TRAP	C\$ERDF	
(6)	014216	005041		.WORD	11	
(6)	014220	007154		.WORD	EMO	
6076				.WORD	ERR11	
6077						:EXPECTED (R4) IS COMBINATION OF
6078	014222					:IBUS* <14> AND <15>
6079	014222					
(3)	014222	104410		ESCAPE	TST	
(3)	014224	000104		TRAP	C\$ESCAPE	
				.WORD	L10053-	

```
6080 014226          BREAK
(3) 014226 104422    TRAP    C$BRK
6081 014230 032702 010000 BIT      #10000,R2      ;DONE YET?
6082 014234 001720    BEQ     2$
6083
6084                ;*
6085                ;*THIS SECTION OF CODE ADDED TO MAKE SURE
6086                ;*THAT MASTER CLEAR, CLEARS THE MAR
6087                ;*
6088 014236          SKIP06 40$
(1)                ;GOTO 40$ IF M8206
6089 014246 052711 040000 BIS      #40000,(R1)    ;SET MASTER CLEAR
6090 014252 005011    CLR      (R1)         ;CLEAR MASTER CLEAR
6091 014254          ROMCLK          ;WE MUST FIRST CLOCK
(1) 014254 004537 003030 JSR     R5,,ROMCLK    ;CLOCK INSTRUCTION
6092 014260 121325    121325          ;THE MAR LATCH REGS
6093 014262          ROMCLK          ;BEFORE WE CAN READ THEM
(1) 014262 004537 003030 JSR     R5,,ROMCLK    ;CLOCK INSTRUCTION
6094 014266 121304    121304          ;READ IBUS* <15> PUT IN PORT5
6095 014270          ROMCLK          ;CLOCK INSTRUCTION
(1) 014270 004537 003030 JSR     R5,,ROMCLK    ;MAR HIGH
6096 014274 121325    121325          ;READ IBUS* <14>, PUT IN PORT4
6097 014276          ROMCLK          ;CLOCK INSTRUCTION
(1) 014276 004537 003030 JSR     R5,,ROMCLK    ;MAR LOW
6098 014302 121304    121304          ;EXPECT MAR CLEAR
6099 014304 005002    CLR      R2         ;READ PORTS 4&5. THEY CONTAIN
6100 014306 016104 000004 MOV     4(R1),R4      ;THE CONTENTS OF THE MAR
6101
6102                ;MASTER CLEAR SHOULD HAVE
6103                ;CLEARED THE MAR
6104                ;BRANCH END TST IF CLEAR
6105 014312 001406    BEQ     40$
6106 014314          ERROR    44
(5) 014320 104455    TRAP    C$ERDF
(6) 014322 000054    .WORD  44
(6) 014324 005041    .WORD  EMO
(6) 014326 010732    .WORD  ERR44
6107 014330          40$:
6108 014330          ENDTST
(3) 014330          L10053:
(3) 014330 104401    TRAP    C$ETST
6109
6110 014332          BADHEAD
(2)                ;***** TEST 10 *****
6111                ;*IOP (CRAM) ODT BITS TEST
6112                ;*LOAD MAR WITH A 0 INC MAR UNTIL IT OVERFLOWS
6113                ;*VERIFY THAT IBUS* 10 BITS IS SET ONLY WHEN MAR BIT 8 IS A ONE
6114                ;*AND THAT IBUS* 10 BIT6 IS SET ON MAR OVERFLOW
6115 014332          BADHEAD
(2)                ;***** TEST 10 *****
6116
6117 014332          BGNTST
(3) 014332          T10::
6118 014332          MACEX
(1)                ;DO NOT DO TEST IF M8200
(4) 014340 104432    TRAP    C$EXIT
```

6119	(4)	014342	000234		.WORD	L10054-	
6120	(1)	014344	013701	002516	MYINT		
6121		014350			MOV	KMCSR,R1	:RECORD DEVICE ADDR.
6122		014354	005002		MSTCLR		:R1 CONTAINS BASE M8200,4,7 ADDRESS
6123		014356			CLR	R2	:MASTER CLEAR M8200,4,7
6124	(1)	014356	004537	003030	ROMCLK		:R2=SAME AS MAR CONTENTS
6125		014362	010000		JSR	R5,ROMCLK	:NEXT WORD IS INSTRUCTION,
6126		014364				010000	:CLOCK INSTRUCTION
6127	(1)	014364	004537	003030	ROMCLK		:MAR_0
6128		014370	121204		JSR	R5,ROMCLK	:NEXT WORD IS INSTRUCTION,
6129		014372	005005			121204	:CLOCK INSTRUCTION
6130		014374	032702	000400	CLR	R5	:PORT4=IBUS*10
6131		014400	001402		BIT	#BIT8,R2	:R5='EXPECTED'
6132		014402	012705	000040	BEQ	.+6	:IS BIT8 SET IN MAR?
6133		014406	016104	000004	MOV	#BIT5,R5	:BR IF NO
6134		014412	042704	177637	MOV	4(R1),R4	:IF YES THEN SET BITS
6135		014416	020504		BIC	#177637,R4	:R4='FOUND'
6136		014420	001410		CMP	R5,R4	:CLEAR UNWANTED BITS
6137	(5)	014422	104455		BEQ	15\$	:BITS 5&6 SHOULD BE CLEAR
6138	(6)	014426	000007		ERROR	7	:BR IF OK
6139	(6)	014430	000007		TRAP	C\$ERDF	:ERROR BITS 5&6 NOT CLEAR
6140	(6)	014432	005041		.WORD	7	
6141	(6)	014434	006704		.WORD	EMO	
6142		014436	104410		.WORD	ERR7	
6143	(3)	014440	000136		ESCAPE	TST	
6144	(3)	014442	000136		TRAP	C\$ESCAPE	
6145		014442			.WORD	L10054-	
6146	(1)	014442	004537	003030	ROMCLK		:NEXT WORD IS INSTRUCTION,
6147		014446	014000		JSR	R5,ROMCLK	:CLOCK INSTRUCTION
6148		014450	005202			014000	:INC MAR
6149		014452	022702	002000	INC	R2	:BUMP MEM ADDRESS
6150		014456	001342		CMP	#2000,R2	:OVERFLOWED YET?(OVFL PAGE BITS).
6151	(1)	014460	004537	003030	BNE	1\$	:BR IF NO
6152		014464	121204		ROMCLK		:NEXT WORD IS INSTRUCTION,
6153		014466	012705	000100	JSR	R5,ROMCLK	:CLOCK INSTRUCTION
6154		014472	016104	000004		121204	:PART4 IBUS* 10
6155		014476	042704	177627	MOV	#BIT6,R5	:R5='EXPECTED'
6156		014502	020504		MOV	4(R1),R4	:R4='FOUND'
6157		014504	001406		BIC	#177627,R4	:CLEAR UNWANTED BITS
6158		014506			CMP	R5,R4	:BIT6 SHOULD BE SET
6159	(5)	014512	104455		BEQ	17\$	:BR IF OK
6160	(6)	014514	000007		ERROR	7	:ERROR, BIT6 NOT SET
6161	(6)	014516	005041		TRAP	C\$ERDF	
6162	(6)	014520	006704		.WORD	7	
6163		014522			.WORD	EMO	
6164	(1)	014522	004537	003030	.WORD	ERR7	
6165		014526	010000		ROMCLK		:NEXT WORD IS INSTRUCTION,
6166		014530			JSR	R5,ROMCLK	:CLOCK INSTRUCTION
6167	(1)	014530	004537	003030		010000	:MAR_0
6168		014534	004000		ROMCLK		:NEXT WORD IS INSTRUCTION,
6169		014536			JSR	R5,ROMCLK	:CLOCK INSTRUCTION
6170						004000	:MAR HI_0
6171					ROMCLK		:NEXT WORD IS INSTRUCTION,

```
(1) 014536 004537 003030 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
6157 014542 121204 121204 CLR R5 ;PORT4 IBUS* 10
6158 014544 005005 MOV 4(R1),R4 ;R5='EXPECTED'
6159 014546 016104 000004 BIC #177637,R4 ;R4='FOUND'
6160 014552 042704 177637 CMP R5,R4 ;CLEAR UNWANTED BITS
6161 014556 020504 BEQ 2$ ;BITS 5&6 SHOULD BE CLEAR
6162 014560 001406 ERROR 7 ;BR IF OK
6163 014562 TRAP C$ERDF ;ERROR 5&6 NOT BOTH CLEAR
(5) 014566 104455 .WORD 7
(6) 014570 000007 .WORD EMO
(6) 014572 005041 .WORD ERR7
(6) 014574 006704
6164 014576
6165 014576 2$:
(3) 014576 ENDTST
(3) 014576 104401 L10054:
6166 TRAP C$ETST
6167 014600 BADHEAD
(2) ;***** TEST 11 *****
6168 ;*CRAM TEST OF JUMP(I) NEVER MICRO-PROCESSOR INSTRUCTION.
6169 ;*PERFORM THE JUMP INSTRUCTION
6170 ;*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
6171 ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
6172 ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
6173 ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
6174 ;*THE CRAM PC IS CORRECT. IF THE CRAM PC IS NOT RIGHT,
6175 ;*THEN PORT4 CONTAINS A 37
6176 014600 BADHEAD
(2) ;***** TEST 11 *****
6177
6178 014600 BGNTST
(3) 014600 T11::
6179 014600 SKIP04 10$
(1) ;GOTO 10$ IF M8204
6180 014610 EXIT TST ;CAN'T DO IF ROM,4K
(3) 014610 104432 TRAP C$EXIT
(3) 014612 000230 .WORD L10055-.
6181 014614 10$:
6182 014614 MYINT
(1) 014614 013701 002516 MOV KMCSR,R1 ;RECORD DEVICE ADDR.
6183 ;R1 CONTAINS BASE M8200,4,7 ADDRESS
6184 014620 MSTCLR ;MASTER CLEAR M8200,4,7
6185 014624 BGNSEG
(3) 014624 104404 TRAP C$BSEG
6186 014626 004737 003460 JSR PC,MEMSET ;SET MEM AND RAM
6187 014632 1$:
6188 014632 004737 003152 JSR PC,CLRALL ;CLEAR ALL CONDITIONS
6189 014636 004537 003064 SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 014636 004537 003064 JSR R5,.SROMCLK
6190 014642 100400 ;START AT ROM PC=0
6191 014644 SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 014644 004537 003064 JSR R5,.SROMCLK
6192 014650 114377 114377!<400*0> ;JUMP TO ROM PC OF 1777
6193 014652 004737 003314 JSR PC,RAMDAT ;R4=CRAM PC (LSB 8 BITS)
6194 014656 000001 ;EXPECTED DATA
6195 014660 120504 CMPB R5,R4 ;IS ROM PC CORRECT?
```



6196	014662	001406		BEQ	2\$		:BR IF NO
6197	014664			ERROR	5		:ERROR, CRAM PC IS WRONG
(5)	014670	104455		TRAP	C\$ERDF		
(6)	014672	000005		.WORD	5		
(6)	014674	005041		.WORD	EMO		
(6)	014676	006434		.WORD	ERR5		
6198	014700			ESCAPE	SEG		
(3)	014700	104410		TRAP	C\$ESCAPE		
(3)	014702	000002		.WORD	10000\$-		
6199	014704			ENDSEG			
(3)	014704		2\$:				
(3)	014704	104405		TRAP	C\$ESEG		
6200	014706			BGNSEG			
(3)	014706	104404		TRAP	C\$BSEG		
6201	014710	004737	003152	JSR	PC,CLRALL		:CLEAR ALL CONDITIONS
6202	014714			SROMCLK			:NEXT WORD IS INSTRUCTION,
(1)	014714	004537	003064	JSR	R5,..SROMCLK		
6203	014720	100403		100403			:START AT ROM PC=3
6204	014722			SROMCLK			:NEXT WORD IS INSTRUCTION,
(1)	014722	004537	003064	JSR	R5,..SROMCLK		
6205	014726	100000		100000!	<400*0>		:JUMP TO ROM PC OF 0
6206	014730	004737	003314	JSR	PC,RAMDAT		:R4=CRAM PC (LSB 8 BITS)
6207	014734	000004		4			:EXPECTED DATA
6208	014736	120504		CMPB	R5,R4		:IS ROM PC CORRECT?
6209	014740	001406		BEQ	4\$		:BR IF YES
6210	014742			ERROR	5		:ERROR, CROM PC IS WRONG
(5)	014746	104455		TRAP	C\$ERDF		
(6)	014750	000005		.WORD	5		
(6)	014752	005041		.WORD	EMO		
(6)	014754	006434		.WORD	ERR5		
6211	014756			ESCAPE	SEG		
(3)	014756	104410		TRAP	C\$ESCAPE		
(3)	014760	000002		.WORD	10001\$-		
6212	014762			ENDSEG			
(3)	014762		4\$:				
(3)	014762	104405		TRAP	C\$ESEG		
6213	014764			BGNSEG			
(3)	014764	104404		TRAP	C\$BSEG		
6214	014766	004737	003152	JSR	PC,CLRALL		:CLEAR ALL CONDITINS
6215	014772			SROMCLK			:NEXT WORD IS INSTRUCION,
(1)	014772	004537	003064	JSR	R5,..SROMCLK		
6216	014776	100406		100406			:START AT ROM PC=6
6217	015000			SROMCLK			:NEXT WORD IS INSTRUCTION,
(1)	015000	004537	003064	JSR	R5,..SROMCLK		
6218	015004	104125		104125!	<400*0>		:JUMP TO ROM PC OF 525
6219	015006	004737	003314	JSR	PC,RAMDAT		:R4=CRAM PC (LSB 8 BITS)
6220	015012	000007		7			:EXPECTED DATA
6221	015014	120504		CMPB	R5,R4		:IS ROM PC CORRECT?
6222	015016	001406		BEQ	6\$		:BR IF YES
6223	015020			ERROR	5		:ERROR, CRAM PC IS WRONG
(5)	015024	104455		TRAP	C\$ERDF		
(6)	015026	000005		.WORD	5		
(6)	015030	005041		.WORD	EMO		
(6)	015032	006434		.WORD	ERR5		
6224	015034			ESCAPE	SEG		
(3)	015034	104410		TRAP	C\$ESCAPE		
			6\$:				

```

(3) 015036 000002 .WORD 10002$-.
6225 015040 ENDSEG
(3) 015040 10002$:
(3) 015040 104405 TRAP C$ESEG
6226 015042 ENDTST
(3) 015042 L10055:
(3) 015042 104401 TRAP C$ETST
6227
6228 015044 BADHEAD
(2)
6229 :***** TEST 12 *****
6230 :*CRAM TEST OF JUMP(I) ALWAYS MICRO-PROCESSOR INSTRUCTION.
6231 :*PERFORM THE JUMP INSTRUCTION
6232 :*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
6233 :*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
6234 :*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
6235 :*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
6236 :*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
6237 015044 :*THEN PORT4 WILL CONTAIN A 37
(2) BADHEAD
6238 :***** TEST 12 *****
6239 015044 BGNTST
(3) 015044 T12::
6240 015044 MACEX2 ;DON'T DO If M8200
(1) ;DO NOT DO TEST IF M8200
(4) 015054 104432 TRAP C$EXIT
(4) 015056 000214 .WORD L10056-.
6241 015060 MYINT
(1) 015060 013701 002516 MOV KMCSR,R1 ;RECORD DEVICE ADDR.
6242 ;R1 CONTAINS BASE M8200,4,7 ADDRESS
6243 015064 MSTCLR ;MASTER CLEAR M8200,4,7
6244 015070 004737 003460 JSR PC,MEMSET ;SET MEM AND RAM
6245 015074 104404 1$: BGNSEG
(3) 015074 104404 TRAP C$BSEG
6246 015076 SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 015076 004537 003064 JSR R5, .SROMCLK
6247 015102 100400 ;START AT ROM PC=0
6248 015104 SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 015104 004537 003064 JSR R5, .SROMCLK
6249 015110 114777 114377! <400*1> ;JUMP TO ROM PC OF 1777
6250 015112 004737 003314 JSR PC,RAMDAT ;R4=CRAM PC (LSB 8 BITS)
6251 015116 000377 377 ;EXPECTED DATA
6252 015120 120504 CMPB R5,R4 ;IS ROM PC CORRECT?
6253 015122 001406 BEQ 2$ ;BR IF YES
6254 015124 104455 ERROR 5 ;ERROR, CRAM PC IS WRONG
(5) 015130 104455 TRAP C$ERDF
(6) 015132 000005 .WORD 5
(6) 015134 005041 .WORD EM0
(6) 015136 006434 .WORD ERR5
6255 015140 2$: ESCAPE SEG
(3) 015140 104410 TRAP C$ESCAPE
(3) 015142 000002 .WORD 10000$-.
6256 015144 ENDSEG
(3) 015144 104405 10000$:
6257 015146 TRAP C$ESEG
BGNSEG
  
```

```

(3) 015146 104404 TRAP C$BSEG
6258 015150 SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 015150 004537 003064 JSR R5, SROMCLK
6259 015154 100403 100403 ;START AT ROM PC=3
6260 015156 SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 015156 004537 003064 JSR R5, SROMCLK
6261 015162 100400 100000!<400*1> ;JUMP TO ROM PC OF 0
6262 015164 004737 003314 JSR PC, RAMDAT ;R4=CRAM PC (LSB 8 BITS)
6263 015170 000000 0 ;EXPECTED DATA
6264 015172 120504 CMPB R5, R4 ;IS ROM PC CORRECT?
6265 015174 001406 BEQ 4$ ;BR IF YES
6266 015176 ERROR 5 ;ERROR, CRAM PC IS WRONG
(5) 015202 104455 TRAP C$ERDF
(6) 015204 000005 .WORD 5
(6) 015206 005041 .WORD EMO
(6) 015210 006434 .WORD ERR5
6267 015212 4$: ESCAPE SEG
(3) 015212 104410 TRAP C$ESCAPE
(3) 015214 000002 .WORD 10001$-.
6268 015216 ENDSEG
(3) 015216 104405 10001$: TRAP C$ESEG
6269 015220 BGNSEG
(3) 015220 104404 TRAP C$BSEG
6270 015222 SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 015222 004537 003064 JSR R5, SROMCLK
6271 015226 100406 100406 ;START AT ROM PC=6
6272 015230 SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 015230 004537 003064 JSR R5, SROMCLK
6273 015234 104525 104125!<400*1> ;JUMP TO ROM PC OF 525
6274 015236 004737 003314 JSR PC, RAMDAT ;R4=CRAM PC (LSB 8 BITS)
6275 015242 000125 125 ;EXPECTED DATA
6276 015244 120504 CMPB R5, R4 ;IS ROM PC CORRECT?
6277 015246 001406 BEQ 6$ ;BR IF YES
6278 015250 ERROR 5 ;ERROR, CRAM PC IS WRONG
(5) 015254 104455 TRAP C$ERDF
(6) 015256 000005 .WORD 5
(6) 015260 005041 .WORD EMO
(6) 015262 006434 .WORD ERR5
6279 015264 6$: ESCAPE SEG
(3) 015264 104410 TRAP C$ESCAPE
(3) 015266 000002 .WORD 10002$-.
6280 015270 ENDSEG
(3) 015270 104405 10002$: TRAP C$ESEG
6281 015272 ENDTST
(3) 015272 L10056: TRAP C$ETST
(3) 015272 104401
6282 015274 BADHEAD
6283 (2) ;***** TEST 13 *****
6284 ;*CRAM TEST OF JUMP(I) ON C BIT SET MICRO-PROCESSOR INSTRUCTION.
6285 ;*SET THE C BIT, PERFORM THE JUMP INSTRUCTION.
6286 ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
6287 ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
6288 ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT!

```

```
6289                                     ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
6290                                     ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
6291                                     ;*THEN PORT4 WILL CONTAIN A 37
6292 015274                               BADHEAD
(2)                                       ;***** TEST 13 *****
6293
6294 015274                               BGNTST
(3) 015274                               T13::
6295 015274                               MACEX2
(1)                                       ;DO NOT DO TEST IF M8200
(4) 015304 104432                       ;DON'T DO IF M8200
(4) 015306 000230                       TRAP C$EXIT
6296 015310                               .WORD L10057-.
(1) 015310 013701 002516               MYINT
6297                                       MOV KMCSR,R1
6298 015314                               ;RECORD DEVICE ADDR.
6299 015320 004737 003460               ;R1 CONTAINS BASE M8200,4,7 ADDRESS
6300 015324                               ;MASTER CLEAR M8200,4,7
(3) 015324 104404                       ;SET MEM AND RAM
6301 015326 004737 003244               1$:
6302 015332                               MSTCLR
(1) 015332 004537 003064               JSR PC,MEMSET
6303 015336 100400                       BGNSEG
6304 015340                               TRAP C$BSEG
(1) 015340 004537 003064               JSR PC,SETC
6305 015344 115377                       ;SET THE C BIT'
6306 015346 004737 003314               SR0MCLK
6307 015352 000377                       ;NEXT WORD IS INSTRUCTION,
6308 015354 120504                       JSR R5,.SR0MCLK
6309 015356 001406                       100400
6310 015360                               ;START AT ROM PC=0
(5) 015364 104455                       SR0MCLK
(6) 015366 000005                       JSR R5,.SR0MCLK
(6) 015370 005041                       114377!<400*2>
(6) 015372 006434                       ;JUMP TO ROM PC OF 1777
6311 015374                               ;R4=CRAM PC (LSB 8 BITS)
6312 015374 104410                       ;EXPECTED DATA
(3) 015374 104410                       ;IS ROM PC CORRECT?
(3) 015376 000002                       ;BR IF YES
6313 015400                               ;ERROR, CRAM PC IS WRONG
(3) 015400 104405                       2$:
6314 015402                               ESCAPE SEG
(3) 015402 104404                       TRAP C$ESCAPE
6315 015404 004737 003244               .WORD 10000$-.
6316 015410                               ENDSEG
(1) 015410 004537 003064               10000$:
6317 015414 100403                       TRAP C$ESEG
6318 015416 004537 003064               BGNSEG
(1) 015416 004537 003064               TRAP C$BSEG
6319 015422 101000                       JSR PC,SETC
6320 015424 004737 003314               ;SET THE C BIT'
6321 015430 000000                       SR0MCLK
6322 015432 120504                       ;NEXT WORD IS INSTRUCTION,
6323 015434 001406                       JSR R5,.SR0MCLK
6324 015436                               100403
;START AT ROM PC=3
;NEXT WORD IS INSTRUCTION,
;JUMP TO ROM PC OF 0
;R4=CRAM PC (LSB 8 BITS)
;EXPECTED DATA
;IS ROM PC CORRECT?
;BR IF YES
;ERROR, CRAM PC IS WRONG
```

```
(5) 015442 104455 TRAP C$ERDF
(6) 015444 000005 .WORD 5
(6) 015446 005041 .WORD EMO
(6) 015450 006434 .WORD ERR5
6325 015452 4$: ;LOOP TO 3$ IF SW09=1
6326 015452 ESCAPE SEG
(3) 015452 104410 TRAP C$ESCAPE
(3) 015454 000002 .WORD 10001$-.
6327 015456 ENDSEG
(3) 015456 10001$:
(3) 015456 104405 TRAP C$ESEG
6328 015460 BGNSEG
(3) 015460 104404 TRAP C$BSEG
6329 015462 004737 003244 JSR PC,SETC ;SET THE C BIT'
6330 015466 SROMCLK JSR R5,.SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 015466 004537 003064 JSR R5,.SROMCLK
6331 015472 100406 100406 ;START AT ROM PC=6
6332 015474 SROMCLK JSR R5,.SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 015474 004537 003064 JSR R5,.SROMCLK
6333 015500 105125 104125!<400*2> ;JUMP TO ROM PC OF 525
6334 015502 004737 003314 JSR PC,RAMDAT ;R4=CRAM PC (LSB 8 BITS)
6335 015506 000125 125 ;EXPECTED DATA
6336 015510 120504 CMPB R5,R4 ;IS ROM PC CORRECT?
6337 015512 001406 BEQ 6$ ;BR IF YES
6338 015514 ERROR 5 ;ERROR, CRAM PC IS WRONG
(5) 015520 104455 TRAP C$ERDF
(6) 015522 000005 .WORD 5
(6) 015524 005041 .WORD EMO
(6) 015526 006434 .WORD ERR5
6339 015530 6$: ESCAPE SEG
(3) 015530 104410 TRAP C$ESCAPE
(3) 015532 000002 .WORD 10002$-.
6340 015534 ENDSEG
(3) 015534 10002$:
(3) 015534 104405 TRAP C$ESEG
6341 015536 ENDTST
(3) 015536 L10057: TRAP C$ETST
(3) 015536 104401
6342
6343 015540 BADHEAD
(2) ;***** TEST 14 *****
6344 ;*CRAM TEST OF JUMP(I) ON Z BIT SET MICRO-PROCESSOR INSTRUCTION.
6345 ;*SET THE Z BIT, PERFORM THE JUMP INSTRUCTION.
6346 ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
6347 ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
6348 ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
6349 ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
6350 ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
6351 ;*THEN PORT4 WILL CONTAIN A 37
6352 015540 BADHEAD
(2) ;***** TEST 14 *****
6353
6354 015540 BGNST
(3) 015540 T14::
6355 015540 MACEX2 ;DON'T DO IF M8200.
(1) ;DO NOT DO TEST IF M8200
```

(4)	015550	104432		TRAP	C\$EXIT	
(4)	015552	000230		.WORD	L10060-	
6356	015554			MYINT		
(1)	015554	013701	002516	MOV	KMCSR,R1	:RECORD DEVICE ADDR.
6357						:R1 CONTAINS BASE M8200,4,7 ADDRESS
6358	015560			MSTCLR		:MASTER CLEAR M8200,4,7
6359	015564	004737	003460	JSR	PC,MEMSET	:SET MEM AND RAM
6360	015570			BGNSEG		
(3)	015570	104404		TRAP	C\$BSEG	
6361	015572	004737	003276	JSR	PC,SETZ	:SET THE Z BIT'
6362	015576			SROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	015576	004537	003064	JSR	R5,.SROMCLK	
6363	015602	100400		100400		:START AT ROM PC=0
6364	015604			SROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	015604	004537	003064	JSR	R5,.SROMCLK	
6365	015610	115777		114377!<400*3>		:JUMP TO ROM PC OF 1777
6366	015612	004737	003314	JSR	PC,RAMDAT	:R4=CRAM PC (LSB 8 BITS)
6367	015616	000377		377		:EXPECTED DATA
6368	015620	120504		CMPB	R5,R4	:IS ROM PC CORRECT?
6369	015622	001406		BEQ	2\$	:BR IF YES
6370	015624			ERROR	5	:ERROR, CRAM PC IS WRONG
(5)	015630	104455		TRAP	C\$ERDF	
(6)	015632	000005		.WORD	5	
(6)	015634	005041		.WORD	EMO	
(6)	015636	006434		.WORD	ERR5	
6371	015640			ESCAPE	SEG	
(3)	015640	104410		TRAP	C\$ESCAPE	
(3)	015642	000002		.WORD	10000\$-	
6372	015644			ENDSEG		
(3)	015644					
(3)	015644	104405		TRAP	C\$ESEG	
6373	015646			BGNSEG		
(3)	015646	104404		TRAP	C\$BSEG	
6374	015650	004737	003276	JSR	PC,SETZ	:SET THE Z BIT'
6375	015654			SROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	015654	004537	003064	JSR	R5,.SROMCLK	
6376	015660	100403		100403		:START AT ROM PC=3
6377	015662			SROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	015662	004537	003064	JSR	R5,.SROMCLK	
6378	015666	101400		100000!<400*3>		:JUMP TO ROM PC OF 0
6379	015670	004737	003314	JSR	PC,RAMDAT	:R4=CRAM PC (LSB 8 BITS)
6380	015674	000000		0		:EXPECTED DATA
6381	015676	120504		CMPB	R5,R4	:IS ROM PC CORRECT?
6382	015700	001406		BEQ	4\$	:BR IF YES
6383	015702			ERROR	5	:ERROR, CRAM PC IS WRONG
(5)	015706	104455		TRAP	C\$ERDF	
(6)	015710	000005		.WORD	5	
(6)	015712	005041		.WORD	EMO	
(6)	015714	006434		.WORD	ERR5	
6384	015716			ESCAPE	SEG	
(3)	015716	104410		TRAP	C\$ESCAPE	
(3)	015720	000002		.WORD	10001\$-	
6385	015722			ENDSEG		
(3)	015722					
(3)	015722	104405		TRAP	C\$ESEG	
6386	015724			BGNSEG		

(3)	015724	104404		TRAP	C\$BSEG	
6387	015726	004737	003276	JSR	PC,SETZ	;SET THE Z BIT'
6388	015732			SROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	015732	004537	003064	JSR	R5, SROMCLK	
6389	015736	100406		100406		;START AT ROM PC=6
6390	015740			SROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	015740	004537	003064	JSR	R5, SROMCLK	
6391	015744	105525		104125!<400*3>		;JUMP TO ROM PC OF 525
6392	015746	004737	003314	JSR	PC,RAMDAT	;R4=CRAM PC (LSB 8 BITS)
6393	015752	000125		125		;EXPECTED DATA
6394	015754	120504		CMPB	R5,R4	;IS ROM PC CORRECT?
6395	015756	001406		BEQ	6\$	;BR IF YES
6396	015760			ERROR	5	;ERROR, CRAM PC IS WRONG
(5)	015764	104455		TRAP	C\$ERDF	
(6)	015766	000005		.WORD	5	
(6)	015770	005041		.WORD	EMO	
(6)	015772	006434		.WORD	ERR5	
6397	015774			6\$: ESCAPE	SEG	
(3)	015774	104410		TRAP	C\$ESCAPE	
(3)	015776	000002		.WORD	10002\$-	
6398	016000			ENDSEG		
(3)	016000			10002\$: TRAP	C\$ESEG	
(3)	016000	104405		ENDTST		
6399	016002			L10060: TRAP	C\$ETST	
(3)	016002	104401				
6400						
6401	016004			BADHEAD		
(2)				:***** TEST 15 *****		
6402				:*CRAM TEST OF JUMP(I) ON BRO SET MICRO-PROCESSOR INSTRUCTION.		
6403				:*SET THE BRO BIT, PERFORM THE JUMP INSTRUCTION.		
6404				:*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION		
6405				:*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE		
6406				:*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT		
6407				:*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT		
6408				:*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL		
6409				:*THEN PORT4 WILL CONTAIN A 37		
6410	016004			BADHEAD		
(2)				:***** TEST 15 *****		
6411						
6412	016004			BGNTST		
(3)	016004			T15::		
6413	016004			MACEX2		;DON'T DO IF M8200.
(1)				:DO NOT DO TEST IF M8200		
(4)	016014	104432		TRAP	C\$EXIT	
(4)	016016	000230		.WORD	L10061-	
6414	016020			MYINT		
(1)	016020	013701	002516	MOV	KMCSR,R1	;RECORD DEVICE ADDR.
6415						;R1 CONTAINS BASE M8200,4,7 ADDRESS
6416	016024			MSTCLR		;MASTER CLEAR M8200,4,7
6417	016030	004737	003460	JSR	PC,MEMSET	;SET MEM AND RAM
6418	016034			1\$:		
6419	016034			BGNSEG		
(3)	016034	104404		TRAP	C\$BSEG	
6420	016036	004737	003204	JSR	PC,SETBRO	;SET THE BRO BIT'
6421	016042			SROMCLK		;NEXT WORD IS INSTRUCTION,

(1)	016042	004537	003064	JSR	R5,,SROMCLK	
6422	016046	100400		100400		;START AT ROM PC=0
6423	016050			SROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	016050	004537	003064	JSR	R5,,SROMCLK	
6424	016054	116377		114377!<400*4>		;JUMP TO ROM PC OF 1777
6425	016056	004737	003314	JSR	PC,RAMDAT	;R4=CRAM PC (LSB 8 BITS)
6426	016062	000377		377		;EXPECTED DATA
6427	016064	120504		CMPB	R5,R4	;IS ROM PC CORRECT?
6428	016066	001406		BEQ	2\$	;BR IF YES
6429	016070			ERROR	5	;ERROR, CRAM PC IS WRONG
(5)	016074	104455		TRAP	C\$ERDF	
(6)	016076	000005		.WORD	5	
(6)	016100	005041		.WORD	EMO	
(6)	016102	006434		.WORD	ERR5	
6430	016104			ESCAPE	SEG	
(3)	016104	104410		TRAP	C\$ESCAPE	
(3)	016106	000002		.WORD	10000\$-	
6431	016110			ENDSEG		
(3)	016110					
(3)	016110	104405		TRAP	C\$ESEG	
6432	016112			BGNSEG		
(3)	016112	104404		TRAP	C\$BSEG	
6433	016114	004737	003204	JSR	PC,SETBRO	;SET THE BRO BIT'
6434	016120			SROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	016120	004537	003064	JSR	R5,,SROMCLK	
6435	016124	100403		100403		;START AT ROM PC=3
6436	016126			SROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	016126	004537	003064	JSR	R5,,SROMCLK	
6437	016132	102000		100000!<400*4>		;JUMP TO ROM PC OF 0
6438	016134	004737	003314	JSR	PC,RAMDAT	;R4=CRAM PC (LSB 8 BITS)
6439	016140	000000		0		;EXPECTED DATA
6440	016142	120504		CMPB	R5,R4	;IS ROM PC CORRECT?
6441	016144	001406		BEQ	4\$	;BR IF YES
6442	016146			ERROR	5	;ERROR, CRAM PC IS WRONG
(5)	016152	104455		TRAP	C\$ERDF	
(6)	016154	000005		.WORD	5	
(6)	016156	005041		.WORD	EMO	
(6)	016160	006434		.WORD	ERR5	
6443	016162			ESCAPE	SEG	
(3)	016162	104410		TRAP	C\$ESCAPE	
(3)	016164	000002		.WORD	10001\$-	
6444	016166			ENDSEG		
(3)	016166					
(3)	016166	104405		TRAP	C\$ESEG	
6445	016170			BGNSEG		
(3)	016170	104404		TRAP	C\$BSEG	
6446	016172	004737	003204	JSR	PC,SETBRO	;SET THE BRO BIT'
6447	016176			SROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	016176	004537	003064	JSR	R5,,SROMCLK	
6448	016202	100406		100406		;START AT ROM PC=6
6449	016204			SROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	016204	004537	003064	JSR	R5,,SROMCLK	
6450	016210	106125		104125!<400*4>		;JUMP TO ROM PC OF 525
6451	016212	004737	003314	JSR	PC,RAMDAT	;R4=CRAM PC (LSB 8 BITS)
6452	016216	000125		125		;EXPECTED DATA
6453	016220	120504		CMPB	R5,R4	;IS ROM PC CORRECT?



```
6454 016222 001406      BEQ      6$      ;BR IF YES
6455 016224              ERROR     5      ;ERROR, CRAM PC IS WRONG
(5) 016230 104455      TRAP     C$ERDF
(6) 016232 000005      .WORD   5
(6) 016234 005041      .WORD   EMO
(6) 016236 006434      .WORD   ERR5
6456 016240              6$:  ESCAPE  SEG
(3) 016240 104410      TRAP     C$ESCAPE
(3) 016242 000002      .WORD   10002$-.
6457 016244              ENDSEG
(3) 016244              10002$:
(3) 016244 104405      TRAP     C$ESEG
6458 016246              ENDTST
(3) 016246              L10061:
(3) 016246 104401      TRAP     C$ETST
6459
6460 016250              BADHEAD
(2)
6461                    ;***** TEST 16 *****
6462                    ;*CRAM TEST OF JUMP(I) ON BR1 SET MICRO-PROCESSOR INSTRUCTION.
6463                    ;*SET THE BR1 BIT, PERFORM THE JUMP INSTRUCTION.
6464                    ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
6465                    ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
6466                    ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
6467                    ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
6468                    ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
6469 016250              ;*THEN PORT4 WILL CONTAIN A 37
(2)                    BADHEAD
6470                    ;***** TEST 16 *****
6471 016250              BGNTST
(3) 016250              T16::
6472 016250              MACEX2      ;DON'T DO IF M8200.
(1)                    ;DO NOT DO TEST IF M8200
(4) 016260 104432      TRAP     C$EXIT
(4) 016262 000230      .WORD   L10062-.
6473 016264              MYINT
(1) 016264 013701 002516  MOV      KMCSR,R1      ;RECORD DEVICE ADDR.
6474                    ;R1 CONTAINS BASE M8200,4,7 ADDRESS
6475 016270              MSTCLR      ;MASTER CLEAR M8200,4,7
6476 016274 004737 003460  JSR      PC, MEMSET   ;SET MEM AND RAM
6477 016300              1$:
6478 016300              BGNSEG
(3) 016300 104404      TRAP     C$BSEG
6479 016302 004737 003214  JSR      PC, SETBR1   ;SET THE BR1 BIT*
6480 016306              SROMCLK   ;NEXT WORD IS INSTRUCTION,
(1) 016306 004537 003064  JSR      R5, .SROMCLK
6481 016312 100400              100400   ;START AT ROM PC=0
6482 016314              SROMCLK   ;NEXT WORD IS INSTRUCTION,
(1) 016314 004537 003064  JSR      R5, .SROMCLK
6483 016320 116777              114377!<400*5>   ;JUMP TO ROM PC OF 1777
6484 016322 004737 003314  JSR      PC, RAMDAT   ;R4=CRAM PC (LSB 8 BITS)
6485 016326 000377              377       ;EXPECTED DATA
6486 016330 120504              CMPB     R5,R4       ;IS ROM PC CORRECT?
6487 016332 001406      BEQ      2$         ;BR IF YES
6488 016334              ERROR     5      ;ERROR, CRAM PC IS WRONG
(5) 016340 104455      TRAP     C$ERDF
```

(6)	016342	000005		.WORD	5	
(6)	016344	005041		.WORD	EMO	
(6)	016346	006434		.WORD	ERR5	
6489	016350			ESCAPE	SEG	
(3)	016350	104410	2\$:	TRAP	C\$ESCAPE	
(3)	016352	000002		.WORD	10000\$-	
6490	016354			ENDSEG		
(3)	016354		10000\$:			
(3)	016354	104405		TRAP	C\$ESEG	
6491	016356			BGNSEG		
(3)	016356	104404		TRAP	C\$BSEG	
6492	016360	004737	003214	JSR	PC,SETBR1	
6493	016364			SROMCLK		
(1)	016364	004537	003064	JSR	R5,..SROMCLK	;SET THE BR1 BIT'
6494	016370	100403		100403		;NEXT WORD IS INSTRUCTION,
6495	016372			SROMCLK		
(1)	016372	004537	003064	JSR	R5,..SROMCLK	;START AT ROM PC=3
6496	016376	102400		100000!<400*5>		;NEXT WORD IS INSTRUCTION,
6497	016400	004737	003314	JSR	PC,RAMDAT	;JUMP TO ROM PC OF 0
6498	016404	000000		0		;R4=CRAM PC (LSB 8 BITS)
6499	016406	120504		CMPB	R5,R4	;EXPECTED DATA
6500	016410	001406		BEO	4\$	;IS ROM PC CORRECT?
6501	016412			ERROR	5	;BR IF YES
(5)	016416	104455		TRAP	C\$ERDF	;ERROR, CRAM PC IS WRONG
(6)	016420	000005		.WORD	5	
(6)	016422	005041		.WORD	EMO	
(6)	016424	006434		.WORD	ERR5	
6502	016426			ESCAPE	SEG	
(3)	016426	104410	4\$:	TRAP	C\$ESCAPE	
(3)	016430	000002		.WORD	10001\$-	
6503	016432			ENDSEG		
(3)	016432		10001\$:			
(3)	016432	104405		TRAP	C\$ESEG	
6504	016434			BGNSEG		
(3)	016434	104404		TRAP	C\$BSEG	
6505	016436	004737	003214	JSR	PC,SETBR1	
6506	016442			SROMCLK		
(1)	016442	004537	003064	JSR	R5,..SROMCLK	;SET THE BR1 BIT'
6507	016446	100406		100406		;NEXT WORD IS INSTRUCTION,
6508	016450			SROMCLK		
(1)	016450	004537	003064	JSR	R5,..SROMCLK	;START AT ROM PC=6
6509	016454	106525		104125!<400*5>		;NEXT WORD IS INSTRUCTION,
6510	016456	004737	003314	JSR	PC,RAMDAT	;JUMP TO ROM PC OF 525
6511	016462	000125		125		;R4=CRAM PC (LSB 8 BITS)
6512	016464	120504		CMPB	R5,R4	;EXPECTED DATA
6513	016466	001406		BEO	6\$	;IS ROM PC CORRECT?
6514	016470			ERROR	5	;BR IF YES
(5)	016474	104455		TRAP	C\$ERDF	;ERROR, CRAM PC IS WRONG
(6)	016476	000005		.WORD	5	
(6)	016500	005041		.WORD	EMO	
(6)	016502	006434		.WORD	ERR5	
6515	016504			ESCAPE	SEG	
(3)	016504	104410	6\$:	TRAP	C\$ESCAPE	
(3)	016506	000002		.WORD	10002\$-	
6516	016510			ENDSEG		
(3)	016510		10002\$:			

(3) 016510 104405  
 6517 016512  
 (3) 016512  
 (3) 016512 104401  
 6518  
 6519 016514  
 (2)  
 6520  
 6521  
 6522  
 6523  
 6524  
 6525  
 6526  
 6527  
 6528 016514  
 (2)  
 6529  
 6530 016514  
 (3) 016514  
 6531 016514  
 (1)  
 (4) 016524 104432  
 (4) 016526 000230  
 6532 016530  
 (1) 016530 013701 002516  
 6533 016534  
 6534 016540 004737 003460  
 6535 016544  
 6536 016544  
 (3) 016544 104404

TRAP C\$ESEG  
 ENDTST  
 L10062:  
 TRAP C\$ETST

BADHEAD

:\*\*\*\*\* TEST 17 \*\*\*\*\*  
 :\*CRAM TEST OF JUMP(J) ON BR4 SET MICRO-PROCESSOR INSTRUCTION.  
 :\*SET THE BR4 BIT, PERFORM THE JUMP INSTRUCTION.  
 :\*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION  
 :\*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE  
 :\*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT  
 :\*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT  
 :\*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL  
 :\*THEN PORT4 WILL CONTAIN A 37

BADHEAD

:\*\*\*\*\* TEST 17 \*\*\*\*\*

BGNTST  
 T17::

MACEX2 ;DON'T DO IF M8200.  
 :DO NOT DO TEST IF M8200  
 TRAP C\$EXIT  
 .WORD L10063-.

MYINT  
 MOV KMCSR,R1 ;RECORD DEVICE ADDR.  
 MSTCLR ;MASTER CLEAR M8200,4,7  
 JSR PC, MEMSET ;SET MEM AND RAM

1\$:

BGNSEG  
 TRAP C\$BSEG

CZDMQAO M8207 STATIC DIAG #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 F 10 PAGE 45  
HARDWARE TESTS

SEQ 0122

6538 016546 004737 003224  
6539 016552  
(1) 016552 004537 003064  
6540 016556 100400  
6541 016560  
(1) 016560 004537 003064

JSR PC,SETBR4  
SROMCLK  
JSR R5,.SROMCLK  
100400  
SROMCLK  
JSR R5,.SROMCLK

;SET THE BR4 BIT'  
;NEXT WORD IS INSTRUCTION,  
;START AT ROM PC=0  
;NEXT WORD IS INSTRUCTION,

CZDMQAO M8207 STATIC DIAG #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 G 10  
HARDWARE TESTS PAGE 46

SEQ 0123

6543 016564 117377  
6544 016566 004737 003314

114377!<400\*6>  
JSR PC, RAMDAT

:JUMP TO ROM PC OF 1777  
:R4=CRAM PC (LSB 8 BITS)

CZDMQAO M8207 STATIC DIAG #2 MACY11 30A(1052) 17-JUL-79 09:23 H 10  
CZDMQA.P11 25-JUN-79 14:07 HARDWARE TESTS PAGE 47

SEQ 0124

6546 016572 000377  
6547 016574 120504

377  
CMPB R5,R4

:EXPECTED DATA  
:IS ROM PC CORRECT?

CZDMQAO MB207 STATIC DIAG #2 MACY11 30A(1052) 17-JUL-79 09:23 I 10 PAGE 48  
CZDMQA.P11 25-JUN-79 14:07 HARDWARE TESTS

SEQ 0125

6549 016576 001406

BEQ 28

:BR IF YES

CZDMQAO M8207 STATIC DIAG #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 J 10  
HARDWARE TESTS PAGE 49

SEQ 0126

6551 016600  
(5) 016604 104455  
(6) 016606 000005  
(6) 016610 005041  
(6) 016612 006434

ERROR 5  
TRAP C\$ERDF  
.WORD 5  
.WORD EMO  
.WORD ERR5

;ERROR, CRAM PC IS WRONG



```

6553 016614
(3) 016614 104410
(3) 016616 000002
6554 016620
(3) 016620
(3) 016620 104405
6555 016622
(3) 016622 104404
6556 016624 004737 003224
6557 016630
(1) 016630 004537 003064
6558 016634 100403
6559 016636
(1) 016636 004537 003064
6560 016642 103000
6561 016644 004737 003314
6562 016650 000000
6563 016652 120504
6564 016654 001406
6565 016656
(5) 016662 104455
(6) 016664 000005
(6) 016666 005041
(6) 016670 006434
6566 016672
(3) 016672 104410
(3) 016674 000002
6567 016676
(3) 016676
(3) 016676 104405
6568 016700
(3) 016700 104404
6569 016702 004737 003224
6570 016706
(1) 016706 004537 003064
6571 016712 100406
6572 016714
(1) 016714 004537 003064
6573 016720 107125
6574 016722 004737 003314
6575 016726 000125
6576 016730 120504
6577 016732 001406
6578 016734
(5) 016740 104455
(6) 016742 000005
(6) 016744 005041
(6) 016746 006434
6579 016750
(3) 016750 104410
(3) 016752 000002
6580 016754
(3) 016754
(3) 016754 104405
6581 016756
(3) 016756
  
```

```

2$:      ESCAPE SEG
        TRAP  C$ESCAPE
        .WORD 10000$-.
        ENDSEG
10000$:  TRAP  C$ESEG
        BGNSEG
        TRAP  C$BSEG
        JSR   PC,SETBR4
        SROMCLK
        JSR   R5,..SROMCLK
        100403
        SROMCLK
        JSR   R5,..SROMCLK
        100000!<400*6>
        JSR   PC,RAMDAT
        0
        CMPB  R5,R4
        BEQ   4$
        ERROR 5
        TRAP  C$ERDF
        .WORD 5
        .WORD EMO
        .WORD ERR5
4$:      ESCAPE SEG
        TRAP  C$ESCAPE
        .WORD 10001$-.
        ENDSEG
10001$:  TRAP  C$ESEG
        BGNSEG
        TRAP  C$BSEG
        JSR   PC,SETBR4
        SROMCLK
        JSR   R5,..SROMCLK
        100406
        SROMCLK
        JSR   R5,..SROMCLK
        104125!<400*6>
        JSR   PC,RAMDAT
        125
        CMPB  R5,R4
        BEQ   6$
        ERROR 5
        TRAP  C$ERDF
        .WORD 5
        .WORD EMO
        .WORD ERR5
6$:      ESCAPE SEG
        TRAP  C$ESCAPE
        .WORD 10002$-.
        ENDSEG
10002$:  TRAP  C$ESEG
ENDTST
L10063:
  
```

```

;SET THE BR4 BIT'
;NEXT WORD IS INSTRUCTION,

;START AT ROM PC=3
;NEXT WORD IS INSTRUCTION,

;JUMP TO ROM PC OF 0
;R4=CRAM PC (LSB 8 BITS)
;EXPECTED DATA
;IS ROM PC CORRECT?
;BR IF YES
;ERROR, CRAM PC IS WRONG
  
```

```

;SET THE BR4 BIT'
;NEXT WORD IS INSTRUCTION,

;START AT ROM PC=6
;NEXT WORD IS INSTRUCTION,

;JUMP TO ROM PC OF 525
;R4=CRAM PC (LSB 8 BITS)
;EXPECTED DATA
;IS ROM PC CORRECT?
;BR IF YES
;ERROR, CRAM PC IS WRONG
  
```

```
(3) 016756 104401 TRAP C$ETST
6582
6583 016760 BADHEAD
(2) :***** TEST 18 *****
6584 :*CRAM TEST OF JUMP(I) ON BR7 SET MICRO-PROCESSOR INSTRUCTION.
6585 :*SET THE BR7 BIT, PERFORM THE JUMP INSTRUCTION.
6586 :*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
6587 :*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
6588 :*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
6589 :*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
6590 :*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
6591 :*THEN PORT4 WILL CONTAIN A 37
6592 016760 BADHEAD
(2) :***** TEST 18 *****
6593
6594 016760 BGNTST
(3) 016760 T18::
6595 016760 MACEX2 ;DON'T DO IF M8200.
(1) ;DO NOT DO TEST IF M8200
(4) 016770 104432 TRAP C$EXIT
(4) 016772 000230 .WORD L10064-.
6596 016774 MYINT
(1) 016774 013701 002516 MOV KMCSR,R1 ;RECORD DEVICE ADDR.
6597 ;R1 CONTAINS BASE M8200,4,7 ADDRESS
6598 017000 MSTCLR ;MASTER CLEAR M8200,4,7
6599 017004 004737 003460 JSR PC,MEMSET ;SET MEM AND RAM
6600 017010 1$: BGNSEG
(3) 017010 104404 TRAP C$BSEG
6601 017012 004737 003234 JSR PC,SETBR7 ;SET THE BR7 BIT'
6602 017016 SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 017016 004537 003064 JSR R5, SROMCLK
6603 017022 100400 100400 ;START AT ROM PC=0
6604 017024 SROMCLK ;NEXT WORD IS INSTRUCCION,
(1) 017024 004537 003064 JSR R5, SROMCLK
6605 017030 117777 114377! <400*7> ;JUMP TO ROM PC OF 1777
6606 017032 004737 003314 JSR PC,RAMDAT ;R4=CRAM PC (LSB 8 BITS)
6607 017036 000377 377 ;EXPECTED DATA
6608 017040 120504 CMPB R5,R4 ;IS ROM PC CORRECT?
6609 017042 001406 BEQ 2$ ;BR IF YES
6610 017044 ERROR 5 ;ERROR, CRAM PC IS WRONG
(5) 017050 104455 TRAP C$ERDF
(6) 017052 000005 .WORD 5
(6) 017054 005041 .WORD EMO
(6) 017056 006434 .WORD ERR5
6611 017060 2$: ESCAPE SEG
(3) 017060 104410 TRAP C$ESCAPE
(3) 017062 000002 .WORD 10000$-.
6612 017064 ENDSEG
(3) 017064 10000$: TRAP C$ESEG
6613 017066 BGNSEG
(3) 017066 104404 TRAP C$BSEG
6614 017070 004737 003234 JSR PC,SETBR7 ;SET THE BR7 BIT'
6615 017074 SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 017074 004537 003064 JSR R5, SROMCLK
6616 017100 100403 100403 ;START AT ROM PC=3
```

6617	017102			SROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	017102	004537	003064	JSR R5, SROMCLK		
6618	017106	103400		100000!<400*7>		;JUMP TO ROM PC OF 0
6619	017110	004737	003314	JSR PC, RAMDAT		;R4=CRAM PC (LSB 8 BITS)
6620	017114	000000		0		;EXPECTED DATA
6621	017116	120504		CMPB R5, R4		;IS ROM PC CORRECT?
6622	017120	001406		BEQ 4\$		;BR IF YES
6623	017122			ERROR 5		;ERROR, CRAM PC IS WRONG
(5)	017126	104455		TRAP C\$ERDF		
(6)	017130	000005		.WORD 5		
(6)	017132	005041		.WORD EMO		
(6)	017134	006434		.WORD ERR5		
6624	017136			4\$: ESCAPE SEG		
(3)	017136	104410		TRAP C\$ESCAPE		
(3)	017140	000002		.WORD 10001\$-		
6625	017142			ENDSEG		
(3)	017142			10001\$: TRAP C\$ESEG		
(3)	017142	104405		BGNSEG		
6626	017144			TRAP C\$BSEG		
(3)	017144	104404		JSR PC, SETBR7		;SET THE BR7 BIT'
6627	017146	004737	003234	SROMCLK		;NEXT WORD IS INSTRUCTION,
6628	017152			JSR R5, SROMCLK		
(1)	017152	004537	003064	100406		;START AT ROM PC=6
6629	017156	100406		SROMCLK		;NEXT WORD IS INSTRUCTION,
6630	017160			JSR R5, SROMCLK		
(1)	017160	004537	003064	104125!<400*7>		;JUMP TO ROM PC OF 525
6631	017164	107525		JSR PC, RAMDAT		;R4=CRAM PC (LSB 8 BITS)
6632	017166	004737	003314	125		;EXPECTED DATA
6633	017172	000125		CMPB R5, R4		;IS ROM PC CORRECT?
6634	017174	120504		BEQ 6\$		;BR IF YES
6635	017176	001406		ERROR 5		;ERROR, CRAM PC IS WRONG
6636	017200			TRAP C\$ERDF		
(5)	017204	104455		.WORD 5		
(6)	017206	000005		.WORD EMO		
(6)	017210	005041		.WORD ERR5		
(6)	017212	006434		6\$: ESCAPE SEG		
6637	017214			TRAP C\$ESCAPE		
(3)	017214	104410		.WORD 10002\$-		
(3)	017216	000002		ENDSEG		
6638	017220			10002\$: TRAP C\$ESEG		
(3)	017220	104405				
(3)	017220			ENDTST		
6639	017222			L10064: TRAP C\$ETST		
(3)	017222					
(3)	017222	104401				
6640						
6641	017224			BADHEAD		
(2)				:***** TEST 19 *****		
6642				:*CRAM TEST OF JUMP(I) ON C BIT CLEAR MICRO-PROCESSOR INSTRUCTION.		
6643				:*SET THE C BIT, PERFORM THE JUMP INSTRUCTION.		
6644				:*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION		
6645				:*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE		
6646				:*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT		
6647				:*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT		
6648				:*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL		
6649				:*THEN PORT4 WILL CONTAIN A 37		

```

6650 017224          BADHEAD
      (2)             ;***** TEST 19 *****
6651
6652 017224          BGNTST
      (3) 017224      T19::
6653 017224          MACEX2          ;DON'T DO IF M8200.
      (1)             ;DO NOT DO TEST IF M8200
      (4) 017234 104432 TRAP C$EXIT
      (4) 017236 000244 .WORD L10065-.
6654 017240          MYINT
      (1) 017240 013701 002516 MOV KMCSR,R1          ;RECORD DEVICE ADDR.
6655 017244          MSTCLR          ;MASTER CLEAR M8200,4,7
6656 017250 004737 003460 JSR PC,MEMSET      ;SET MEM AND RAM
6657 017254          1$: BGNSEG
      (3) 017254 104404 TRAP C$BSEG
6658 017256 004737 003244 JSR PC,SETC
6659 017262 004737 003152 JSR PC,CLRALL
6660 017266          SRMCLK          ;NEXT WORD IS INSTRUCTION,
      (1) 017266 004537 003064 JSR R5,SRMCLK
      (1) 017272 100400 100400 ;START AT ROM PC=0
6662 017274          SRMCLK          ;NEXT WORD IS INSTRUCTION,
      (1) 017274 004537 003064 JSR R5,SRMCLK
6663 017300 115377 114377!<400*2> ;JUMP TO ROM PC OF 1777
6664 017302 004737 003314 JSR PC,RAMDAT      ;R4=CRAM PC (LSB 8 BITS)
6665 017306 000001 1 ;EXPECTED DATA
6666 017310 120504 CMPB R5,R4          ;IS ROM PC CORRECT?
6667 017312 001406 BEQ 2$          ;BR IF YES
6668 017314          ERROR 5          ;ERROR, CRAM PC IS WRONG
      (5) 017320 104455 TRAP C$ERDF
      (6) 017322 000005 .WORD 5
      (6) 017324 005041 .WORD EMO
      (6) 017326 006434 .WORD ERR5
6669 017330          2$: ESCAPE SEG
      (3) 017330 104410 TRAP C$ESCAPE
      (3) 017332 000002 .WORD 10000$-.
6670 017334          ENDSEG
      (3) 017334 104405 10000$: TRAP C$ESEG
6671 017336          BGNSEG
      (3) 017336 104404 TRAP C$BSEG
6672 017340          SKIP06 6$
      (1)             ;GOTO 6$ IF M8206
6673 017350 004737 003152 JSR PC,CLRALL      ;CLEAR ALL CONDITIONS
6674 017354          SRMCLK          ;NEXT WORD OF INSTRUCTION
      (1) 017354 004537 003064 JSR R5,SRMCLK
6675 017360 100403 100403 ;START AT ROM PC=3
6676 017362          SRMCLK          ;NEXT WORD OF INSTRUCTION
      (1) 017362 004537 003064 JSR R5,SRMCLK
6677 017366 101000 100000!<400*2> ;JUMP TO ROM PC OF 0
6678 017370 004737 003314 JSR PC,RAMDAT      ;R4=CRAM PC(LSB 8 BITS)
6679 017374 000004 4 ;EXPECTED DATA
6680 017376 120504 CMPB R5,R4          ;IS ROM PC CORRECT?
6681 017400 001406 BEQ 4$          ;BR IF YES
6682 017402          ERROR 5          ;ERROR, CRAM PC IS WRONG
      (5) 017406 104455 TRAP C$ERDF
      (6) 017410 000005 .WORD 5
  
```

```
(6) 017412 005041 .WORD EMO
(6) 017414 006434 .WORD ERR5
6683 017416 104410 4$: ESCAPE SEG
(3) 017416 104410 TRAP C$ESCAPE
(3) 017420 000002 .WORD 10001$-.
6684 017422 ENDSEG
(3) 017422 10001$: TRAP C$ESEG
(3) 017422 104405 BGNSEG
6685 017424 TRAP C$BSEG
(3) 017424 104404 TRAP C$BSEG
6686 017426 004737 003152 JSR PC,CLRALL ;CLEAR ALL CONDITIONS
6687 017432 SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 017432 004537 003064 JSR R5, .SROMCLK
6688 017436 100406 100406 ;START AT ROM PC=6
6689 017440 SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 017440 004537 003064 JSR R5, .SROMCLK
6690 017444 105125 104125!<400*2> ;JUMP TO ROM PC OF 525
6691 017446 004737 003314 JSR PC, RAMDAT ;R4=CRAM PC (LSB 8 BITS)
6692 017452 000007 7 ;EXPECTED DATA
6693 017454 120504 CMPB R5,R4 ;IS ROM PC CORRECT?
6694 017456 001406 BEQ 6$ ;BR IF YES
6695 017460 ERROR 5 ;ERROR, CRAM PC IS WRONG
(5) 017464 104455 TRAP C$ERDF
(6) 017466 000005 .WORD 5
(6) 017470 005041 .WORD EMO
(6) 017472 006434 .WORD ERR5
6696 017474 104410 6$: ESCAPE SEG
(3) 017474 104410 TRAP C$ESCAPE
(3) 017476 000002 .WORD 10002$-.
6697 017500 ENDSEG
(3) 017500 10002$: TRAP C$ESEG
(3) 017500 104405
6698 017502 ENDTST
(3) 017502 L10065: TRAP C$ETST
(3) 017502 104401
6699
6700 017504 BADHEAD
(2) ;***** TEST 20 *****
6701 ;*CRAM TEST OF JUMP(I) ON Z BIT CLEAR MICRO-PROCESSOR INSTRUCTION.
6702 ;*CLEAR THE Z BIT, PERFORM THE JUMP INSTRUCTION.
6703 ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
6704 ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
6705 ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
6706 ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
6707 ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
6708 ;*THEN PORT4 WILL CONTAIN A 37
6709 017504 BADHEAD
(2) ;***** TEST 20 *****
6710
6711 017504 BGNTST
(3) 017504 T20::
6712 017504 MACEX2 ;DON'T DO IF M8200.
(1) ;DO NOT DO TEST IF M8200
(4) 017514 104432 TRAP C$EXIT
(4) 017516 000244 .WORD L10066-.
6713 017520 MYINT
```

(1)	017520	013701	002516	MOV	KMCSR,R1				
6714	017524			MSTCLR					:RECORD DEVICE ADDR.
6715	017530	004737	003460	JSR	PC,MEMSET				:MASTER CLEAR M8200,4,7
6716	017534			1\$:	BGNSEG				:SET MEM AND RAM
(3)	017534	104404		TRAP	C\$BSEG				
6717	017536	004737	003276	JSR	PC,SETZ				
6718	017542			MSTCLR					
6719	017546			SROMCLK					:NEXT WORD IS INSTRUCTION,
(1)	017546	004537	003064	JSR	R5,..SROMCLK				
6720	017552	100400		100400					:START AT ROM PC=0
6721	017554			SROMCLK					:NEXT WORD IS INSTRUCCION,
(1)	017554	004537	003064	JSR	R5,..SROMCLK				
6722	017560	115777		114377!<400*3>					:JUMP TO ROM PC OF 1777
6723	017562	004737	003314	JSR	PC,RAMDAT				:R4=CRAM PC (LSB 8 BITS)
6724	017566	000001		1					:EXPECTED DATA
6725	017570	120504		CMPB	R5,R4				:IS ROM PC CORRECT?
6726	017572	001406		BEQ	2\$				:BR IF YES
6727	017574			ERROR	5				:ERROR, CRAM PC IS WRONG
(5)	017600	104455		TRAP	C\$ERDF				
(6)	017602	000005		.WORD	5				
(6)	017604	005041		.WORD	EMO				
(6)	017606	006434		.WORD	ERR5				
6728	017610			2\$:	ESCAPE SEG				
(3)	017610	104410		TRAP	C\$ESCAPE				
(3)	017612	000002		.WORD	10000\$-				
6729	017614			ENDSEG					
(3)	017614			10000\$:					
(3)	017614	104405		TRAP	C\$ESEG				
6730	017616			BGNSEG					
(3)	017616	104404		TRAP	C\$BSEG				
6731	017620			SKIP06	6\$				
(1)				:GOTO 6\$ IF M8206					
6732	017630	004737	003152	JSR	PC,CLRALL				:CLEAR ALL CONDITIONS
6733	017634			SROMCLK					:NEXT WORD IS INSTRUCTION,
(1)	017634	004537	003064	JSR	R5,..SROMCLK				
6734	017640	100403		100403					:START AT ROM PC=3
6735	017642			SROMCLK					:NEXT WORD IS INSTRUCTION,
(1)	017642	004537	003064	JSR	R5,..SROMCLK				
6736	017646	101400		100000!<400*3>					:JUMP TO ROM PC OF 0
6737	017650	004737	003314	JSR	PC,RAMDAT				:R4=CRAM PC (LSB 8 BITS)
6738	017654	000004		4					:EXPECTED DATA
6739	017656	120504		CMPB	R5,R4				:IS ROM PC CORRECT?
6740	017660	001406		BEQ	4\$				:BR IF YES
6741	017662			ERROR	5				:ERROR, CRAM PC IS WRONG
(5)	017666	104455		TRAP	C\$ERDF				
(6)	017670	000005		.WORD	5				
(6)	017672	005041		.WORD	EMO				
(6)	017674	006434		.WORD	ERR5				
6742	017676			4\$:	ESCAPE SEG				
(3)	017676	104410		TRAP	C\$ESCAPE				
(3)	017700	000002		.WORD	10001\$-				
6743	017702			ENDSEG					
(3)	017702			10001\$:					
(3)	017702	104405		TRAP	C\$ESEG				
6744	017704			BGNSEG					
(3)	017704	104404		TRAP	C\$BSEG				

6745	017706	004737	003152	JSR	PC,CLRALL	;CLEAR ALL CONDITIONS
6746	017712			SROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	017712	004537	003064	JSR	R5,..SROMCLK	
6747	017716	100406		100406		;START AT ROM PC=6
6748	017720			SROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	017720	004537	003064	JSR	R5,..SROMCLK	
6749	017724	105525		104125!<400*3>		;JUMP TO ROM PC OF 525
6750	017726	004737	003314	JSR	PC,RAMDAT	;R4=CRAM PC (LSB 8 BITS)
6751	017732	000007		7		;EXPECTED DATA
6752	017734	120504		CMPB	R5,R4	;IS ROM PC CORRECT?
6753	017736	001406		BEQ	6\$	;BR IF YES
6754	017740			ERROR	5	;ERROR, CRAM PC IS WRONG
(5)	017744	104455		TRAP	C\$ERDF	
(6)	017746	000005		.WORD	5	
(6)	017750	005041		.WORD	EMO	
(6)	017752	006434		.WORD	ERR5	
6755	017754			6\$:	ESCAPE SEG	
(3)	017754	104410		TRAP	C\$ESCAPE	
(3)	017756	000002		.WORD	10002\$-	
6756	017760			ENDSEG		
(3)	017760			10002\$:		
(3)	017760	104405		TRAP	C\$ESEG	
6757	017762			ENDTST		
(3)	017762			L10066:		
(3)	017762	104401		TRAP	C\$ETST	
6758						
6759	017764			BADHEAD		
(2)				:***** TEST 21 *****		
6760				:*CRAM TEST OF JUMP(I) ON BRO CLEAR MICRO-PROCESSOR INSTRUCTION.		
6761				:*CLEAR THE BRO BIT, PERFORM THE JUMP INSTRUCTION.		
6762				:*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION		
6763				:*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE		
6764				:*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT		
6765				:*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT		
6766				:*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL		
6767				:*THEN PORT4 WILL CONTAIN A 37		
6768	017764			BADHEAD		
(2)				:***** TEST 21 *****		
6769						
6770	017764			BGNTST		
(3)	017764			T21::		
6771	017764			MACEX2		;DON'T DO IF M8200.
(1)				;DO NOT DO TEST IF M8200		
(4)	017774	104432		TRAP	C\$EXIT	
(4)	017776	000240		.WORD	L10067-	
6772	020000			MYINT		
(1)	020000	013701	002516	MOV	KMCSR,R1	;RECORD DEVICE ADDR.
6773	020004			MSTCLR		;MASTER CLEAR M8200,4,7
6774	020010	004737	003460	JSR	PC,MEMSET	;SET MEM AND RAM
6775	020014			1\$:	BGNSEG	
(3)	020014	104404		TRAP	C\$BSEG	
6776	020016	004737	003152	JSR	PC,CLRALL	;CLEAR ALL CONDITIONS
6777	020022			SROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	020022	004537	003064	JSR	R5,..SROMCLK	
6778	020026	100400		100400		;START AT ROM PC=0
6779	020030			SROMCLK		;NEXT WORD IS INSTRUCTION,

(1)	020030	004537	003064	JSR	R5, .SROMCLK	
6780	020034	116377		114377!	<400*4>	: JUMP TO ROM PC OF 1777
6781	020036	004737	003314	JSR	PC, RAMDAT	: R4=CRAM PC (LSB 8 BITS)
6782	020042	000001		1		: EXPECTED DATA
6783	020044	120504		CMPB	R5, R4	: IS ROM PC CORRECT?
6784	020046	001406		BEQ	2\$	: BR IF YES
6785	020050			ERROR	5	: ERROR, CRAM PC IS WRONG
(5)	020054	104455		TRAP	C\$ERDF	
(6)	020056	000005		.WORD	5	
(6)	020060	005041		.WORD	EMO	
(6)	020062	006434		.WORD	ERR5	
6786	020064			2\$:	ESCAPE SEG	
(3)	020064	104410		TRAP	C\$ESCAPE	
(3)	020066	000002		.WORD	10000\$-	
6787	020070			ENDSEG		
(3)	020070			10000\$:		
(3)	020070	104405		TRAP	C\$ESEG	
6788	020072			BGNSEG		
(3)	020072	104404		TRAP	C\$BSEG	
6789	020074			SKIP06	6\$	
(1)				:GOTO 6\$	IF M8206	
6790	020104	004737	003152	JSR	PC, CLRALL	: CLEAR ALL CONDITIONS
6791	020110			SROMCLK		: NEXT WORD IS INSTRUCTION,
(1)	020110	004537	003064	JSR	R5, .SROMCLK	
6792	020114	100403		100403		: START AT ROM PC=3
6793	020116			SROMCLK		: NEXT WORD IS INSTRUCTION,
(1)	020116	004537	003064	JSR	R5, .SROMCLK	
6794	020122	102000		100000!	<400*4>	: JUMP TO ROM PC OF 0
6795	020124	004737	003314	JSR	PC, RAMDAT	: R4=CRAM PC (LSB 8 BITS)
6796	020130	000004		4		: EXPECTED DATA
6797	020132	120504		CMPB	R5, R4	: IS ROM PC CORRECT?
6798	020134	001406		BEQ	4\$	: BR IF YES
6799	020136			ERROR	5	: ERROR, CRAM PC IS WRONG
(5)	020142	104455		TRAP	C\$ERDF	
(6)	020144	000005		.WORD	5	
(6)	020146	005041		.WORD	EMO	
(6)	020150	006434		.WORD	ERR5	
6800	020152			4\$:	ESCAPE SEG	
(3)	020152	104410		TRAP	C\$ESCAPE	
(3)	020154	000002		.WORD	10001\$-	
6801	020156			ENDSEG		
(3)	020156			10001\$:		
(3)	020156	104405		TRAP	C\$ESEG	
6802	020160			BGNSEG		
(3)	020160	104404		TRAP	C\$BSEG	
6803	020162	004737	003152	JSR	PC, CLRALL	: CLEAR ALL CONDITIONS
6804	020166			SROMCLK		: NEXT WORD IS INSTRUCTION,
(1)	020166	004537	003064	JSR	R5, .SROMCLK	
6805	020172	100406		100406		: START AT ROM PC=6
6806	020174			SROMCLK		: NEXT WORD IS INSTRUCTION,
(1)	020174	004537	003064	JSR	R5, .SROMCLK	
6807	020200	106125		104125!	<400*4>	: JUMP TO ROM PC OF 525
6808	020202	004737	003314	JSR	PC, RAMDAT	: R4=CRAM PC (LSB 8 BITS)
6809	020206	000007		7		: EXPECTED DATA
6810	020210	120504		CMPB	R5, R4	: IS ROM PC CORRECT?
6811	020212	001406		BEQ	6\$	: BR IF YES



```

6812 020214          ERROR 5          ;ERROR, CRAM PC IS WRONG
(5) 020220 104455   TRAP C$ERDF
(6) 020222 000005   .WORD 5
(6) 020224 005041   .WORD EMO
(6) 020226 006434   .WORD ERR5
6813 020230          6$: ESCAPE SEG
(3) 020230 104410   TRAP C$ESCAPE
(3) 020232 000002   .WORD 10002$-.
6814 020234          10002$: ENDSEG
(3) 020234          TRAP C$ESEG
(3) 020234 104405
6815 020236          ENDTST
(3) 020236          L10067: TRAP C$ETST
(3) 020236 104401
6816
6817 020240          BADHEAD
(2)                  ;***** TEST 22 *****
6818                  ;*CRAM TEST OF JUMP(I) ON BR1 CLEAR MICRO-PROCESSOR INSTRUCTION.
6819                  ;*CLEAR THE BR1 BIT, PERFORM THE JUMP INSTRUCTION.
6820                  ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
6821                  ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
6822                  ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
6823                  ;*THE BR DATA IS MOVFD TO PORT4. IF THIS DATA IS CORRECT
6824                  ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
6825                  ;*THEN PORT4 WILL CONTAIN A 37
6826 020240          BADHEAD
(2)                  ;***** TEST 22 *****
6827
6828 020240          BGNTST
(3) 020240          T22::
6829 020240          MACEX2          ;DON'T DO IF M8200.
(1)                  ;DO NOT DO TEST IF M8200
(4) 020250 104432   TRAP C$EXIT
(4) 020252 000240   .WORD L10070-.
6830 020254          MYINT
(1) 020254 013701 002516 MOV KMCSR,R1          ;RECORD DEVICE ADDR.
6831 020260          MSTCLR          ;MASTER CLEAR M8200,4,7
6832 020264 004737 003460 JSR PC,MEMSET        ;SET MEM AND RAM
6833 020270          1$: BGNSEG
(3) 020270 104404   TRAP C$BSEG
6834 020272 004737 003152 JSR PC,CLRALL        ;CLEAR ALL CONDITIONS
6835 020276          SROMCLK        ;NEXT WORD IS INSTRUCTION,
(1) 020276 004537 003064 JSR R5,.SROMCLK
6836 020302 100400   100400          ;START AT ROM PC=0
6837 020304          SROMCLK        ;NEXT WORD IS INSTRUCTION,
(1) 020304 004537 003064 JSR R5,.SROMCLK
6838 020310 116777   114377!<400*5>    ;JUMP TO ROM PC OF 1777
6839 020312 004737 003314 JSR PC,RAMDAT        ;R4=CRAM PC (LSB 8 BITS)
6840 020316 000001   1          ;EXPECTED DATA
6841 020320 120504   CMPB R5,R4          ;IS ROM PC CORRECT?
6842 020322 001406   BEQ 2$            ;BR IF YES
6843 020324          ERROR 5          ;ERROR, CRAM PC IS WRONG
(5) 020330 104455   TRAP C$ERDF
(6) 020332 000005   .WORD 5
(6) 020334 005041   .WORD EMO
(6) 020336 006434   .WORD ERR5
  
```

6844	020340			2\$:	ESCAPE SEG		
(3)	020340	104410			TRAP	C\$ESCAPE	
(3)	020342	000002			.WORD	10000\$-	
6845	020344				ENDSEG		
(3)	020344			10000\$:			
(3)	020344	104405			TRAP	C\$ESEG	
6846	020346				BGNSEG		
(3)	020346	104404			TRAP	C\$BSEG	
6847	020350				SKIP06	6\$	
(1)					:GOTO 6\$ IF M8206		
6848	020360	004737	003152		JSR	PC,CLRALL	:CLEAR ALL CONDITIONS
6849	020364				SROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	020364	004537	003064		JSR	R5, .SROMCLK	
6850	020370	100403			100403		:START AT ROM PC=3
6851	020372				SROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	020372	004537	003064		JSR	R5, .SROMCLK	
6852	020376	102400			100000!<400*5>		:JUMP TO ROM PC OF 0
6853	020400	004737	003314		JSR	PC, RAMDAT	:R4=CRAM PC (LSB 8 BITS)
6854	020404	000004			4		:EXPECTED DATA
6855	020406	120504			CMPB	R5, R4	:IS ROM PC CORRECT?
6856	020410	001406			BEQ	4\$	:BR IF YES
6857	020412				ERROR	5	:ERROR, CRAM PC IS WRONG
(5)	020416	104455			TRAP	C\$ERDF	
(6)	020420	000005			.WORD	5	
(6)	020422	005041			.WORD	EMO	
(6)	020424	006434			.WORD	ERR5	
6858	020426			4\$:	ESCAPE SEG		
(3)	020426	104410			TRAP	C\$ESCAPE	
(3)	020430	000002			.WORD	10001\$-	
6859	020432				ENDSEG		
(3)	020432			10001\$:			
(3)	020432	104405			TRAP	C\$ESEG	
6860	020434				BGNSEG		
(3)	020434	104404			TRAP	C\$BSEG	
6861	020436	004737	003152		JSR	PC, CLRALL	:CLEAR ALL CONDITIONS
6862	020442				SROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	020442	004537	003064		JSR	R5, .SROMCLK	
6863	020446	100406			100406		:START AT ROM PC=6
6864	020450				SROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	020450	004537	003064		JSR	R5, .SROMCLK	
6865	020454	106525			104125!<400*5>		:JUMP TO ROM PC OF 525
6866	020456	004737	003314		JSR	PC, RAMDAT	:R4=CRAM PC (LSB 8 BITS)
6867	020462	000007			7		:EXPECTED DATA
6868	020464	120504			CMPB	R5, R4	:IS ROM PC CORRECT?
6869	020466	001406			BEQ	6\$	:BR IF YES
6870	020470				ERROR	5	:ERROR, CRAM PC IS WRONG
(5)	020474	104455			TRAP	C\$ERDF	
(6)	020476	000005			.WORD	5	
(6)	020500	005041			.WORD	EMO	
(6)	020502	006434			.WORD	ERR5	
6871	020504			6\$:	ESCAPE SEG		
(3)	020504	104410			TRAP	C\$ESCAPE	
(3)	020506	000002			.WORD	10002\$-	
6872	020510				ENDSEG		
(3)	020510			10002\$:			
(3)	020510	104405			TRAP	C\$ESEG	

6873 020512  
(3) 020512  
(3) 020512 104401  
6874  
6875 020514  
(2)  
6876  
6877  
6878  
6879  
6880  
6881  
6882 020514 020514  
6883  
6884  
6885 020516  
(2)  
6886

ENDTST  
L10070:  
TRAP C\$ETST

BADHEAD

\*\*\*\*\* TEST 23 \*\*\*\*\*  
;\*CRAM TEST OF JUMP(I) ON BR4 CLEAR MICRO-PROCESSOR INSTRUCTION.  
;\*CLEAR THE BR4 BIT, PERFORM THE JUMP INSTRUCTION.  
;\*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION  
;\*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE  
;\*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT  
;\*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT

;\*THE CRAM PC IS CORRECT, IF THE CRAM PC IS NOT RIGHT.  
;\*THEN PORT4 CONTAINS A 37

BADHEAD

\*\*\*\*\* TEST 23 \*\*\*\*\*

6888	020516			BGNTST		
(3)	020516			T23::		
6889	020516				MACEX2	:DON'T DO IF M8200.
(1)					:DO NOT DO TEST IF M8200	
(4)	020526	104432			TRAP C\$EXIT	
(4)	020530	000240			.WORD L10071-	
6890	020532				MYINT	
(1)	020532	013701	002516		MOV KMCSR,R1	:RECORD DEVICE ADDR.
6891	020536				MSTCLR	:MASTER CLEAR M8200,4,7
6892	020542	004737	003460		JSR PC, MEMSET	:SET MEM AND RAM
6893	020546			18:	BGNSEG	
(3)	020546	104404			TRAP C\$BSEG	
6894	020550	004737	003152		JSR PC, CLRALL	:CLEAR ALL CONDITIONS
6895	020554				SROMCLK	:NEXT WORD IS INSTRUCTION,
(1)	020554	004537	003064		JSR R5, .SROMCLK	
6896	020560	100400			100400	:START AT ROM PC=0
6897	020562				SROMCLK	:NEXT WORD IS INSTRUCTION,
(1)	020562	004537	003064		JSR R5, .SROMCLK	
6898	020566	117377			114377! <400*6>	:JUMP TO ROM PC OF 1777
6899	020570	004737	003314		JSR PC, RAMDAT	:R4=CRAM PC (LSB 8 BITS)

6901	020574	000001		1					
6902	020576	120504		CMPB	R5,R4			:EXPECTED DATA	
6903	020600	001406		BEQ	2\$			:IS ROM PC CORRECT?	
6904	020602			ERROR	5			:BR IF YES	
(5)	020606	104455		TRAP	C\$ERDF			:ERROR, CRAM PC IS WRONG	
(6)	020610	000005		.WORD	5				
(6)	020612	005041		.WORD	EMO				
(6)	020614	006434		.WORD	ERR5				
6905	020616			ESCAPE	SEG				
(3)	020616	104410		TRAP	C\$ESCAPE				
(3)	020620	000002		.WORD	10000\$-				
6906	020622			ENDSEG					
(3)	020622		2\$:						
(3)	020622	104405		TRAP	C\$ESEG				
6907	020624			BGNSEG					
(3)	020624	104404		TRAP	C\$BSEG				
6908	020626			SKIP06	6\$				
(1)				:GOTO 6\$	IF M8206				
6909	020636	004737	003152	JSR	PC,CLRALL			:CLEAR ALL CONDITIONS	
6910	020642			SROMCLK				:NEXT WORD IS INSTRUCTION,	
(1)	020642	004537	003064	JSR	R5,.SROMCLK				
6911	020646	100403		100403				:START AT ROM PC=3	
6912	020650			SROMCLK				:NEXT WORD IS INSTRUCTION,	
(1)	020650	004537	003064	JSR	R5,.SROMCLK				
6913	020654	103000		100000!	<400*6>	:JUMP TO ROM PC OF 0			
6914	020656	004737	003314	JSR	PC,RAMDAT			:R4=CRAM PC (LSB 8 BITS)	
6915	020662	000004		4				:EXPECTED DATA	

CZDMQAO M8207 STATIC DIAG #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 PAGE 53  
HARDWARE TESTS

K 11

SEQ 0140

6917 020664 120504  
6918 020666 001406

CMPB R5,R4  
BEQ 4\$

:IS ROM PC CORRECT?  
:BSR IF YES

CZDMQAO M8207 STATIC DIAG #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 L 11 PAGE 54  
HARDWARE TESTS

SEQ 0141

6920 020670  
(5) 020674 104455  
(6) 020676 000005  
(6) 020700 005041  
(6) 020702 006434

ERROR 5  
TRAP C\$ERDF  
.WORD 5  
.WORD EMO  
.WORD ERR5

;ERROR, CRAM PC IS WRONG

6922 020704  
 (3) 020704 104410  
 (3) 020706 000002  
 6923 020710  
 (3) 020710  
 (3) 020710 104405  
 6924 020712  
 (3) 020712 104404  
 6925 020714 004737 003152  
 6926 020720  
 (1) 020720 004537 003064  
 6927 020724 100406  
 6928 020726  
 (1) 020726 004537 003064  
 6929 020732 107125  
 6930 020734 004737 003314  
 6931 020740 000007  
 6932 020742 120504  
 6933 020744 001406  
 6934 020746  
 (5) 020752 104455  
 (6) 020754 000005  
 (6) 020756 005041  
 (6) 020760 006434  
 6935 020762  
 (3) 020762 104410  
 (3) 020764 000002  
 6936 020766  
 (3) 020766  
 (3) 020766 104405  
 6937 020770  
 (3) 020770  
 (3) 020770 104401  
 6938  
 6939 020772  
 (2)  
 6940  
 6941  
 6942  
 6943  
 6944  
 6945  
 6946  
 6947  
 6948 020772  
 (2)  
 6949  
 6950 020772  
 (3) 020772  
 6951 020772  
 (1)  
 (4) 021002 104432  
 (4) 021004 000240  
 6952 021006  
 (1) 021006 013701 002516  
 6953 021012

```

4$:   ESCAPE SEG
      TRAP   C$ESCAPE
      .WORD  10001$-.
      ENDSEG

10001$:
      TRAP   C$ESEG
      BGNSEG
      TRAP   C$BSEG
      JSR    PC,CLRALL      ;CLEAR ALL CONDITIONS
      SROMCLK
      JSR    R5,.SROMCLK   ;NEXT WORD IS INSTRUCTION,
      100406                ;START AT ROM PC=6
      SROMCLK                ;NEXT WORD IS INSTRUCTION,
      JSR    R5,.SROMCLK
      104125!<400*6> ;JUMP TO ROM PC OF 525
      JSR    PC,RAMDAT     ;R4=CRAM PC (LSB 8 BITS)
      7                    ;EXPECTED DATA
      CMPB   R5,R4         ;IS ROM PC CORRECT?
      BEQ    6$            ;BR IF YES
      ERROR  5             ;ERROR, CRAM PC IS WRONG
      TRAP   C$ERDF
      .WORD  5
      .WORD  EMO
      .WORD  ERR5

6$:   ESCAPE SEG
      TRAP   C$ESCAPE
      .WORD  10002$-.
      ENDSEG

10002$:
      TRAP   C$ESEG

ENDTST
L10071:
      TRAP   C$ETST

BADHEAD
:***** TEST 24 *****
:*CRAM TEST OF JUMP(I) ON BR7 CLEAR MICRO-PROCESSOR INSTRUCTION.
:*CLEAR THE BR7 BIT, PERFORM THE JUMP INSTRUCTION.
:*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
:*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
:*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
:*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
:*THE CRAM PC IS CORRECT, IF THE CRAM PC IS NOT RIGHT.
:*THEN PORT4 CONTAINS A 37
BADHEAD
:***** TEST 24 *****

BGNTST
T24::
MACEX2                ;DON'T DO IF M8200.
;DO NOT DO TEST IF M8200
TRAP   C$EXIT
.WORD  L10072-.
MYINT
MOV    KMCSR,R1       ;RECORD DEVICE ADDR.
MSTCLR                ;MASTER CLEAR M8200,4,7

```



```

6954 021016 004737 003460      JSR    PC, MEMSET      ;SET MEM AND RAM
6955 021022      1$:    BGNSEG
      (3) 021022 104404      TRAP   C$BSEG
6956 021024 004737 003152      JSR    PC, CLRALL     ;CLEAR ALL CONDITIONS
6957 021030      SROMCLK      ;NEXT WORD IS INSTRUCTION,
      (1) 021030 004537 003064      JSR    R5, .SROMCLK
6958 021034 100400      100400      ;START AT ROM PC=0
6959 021036      SROMCLK      ;NEXT WORD IS INSTRUCTION,
      (1) 021036 004537 003064      JSR    R5, .SROMCLK
6960 021042 117777      114377! <400*7>      ;JUMP TO ROM PC OF 1777
6961 021044 004737 003314      JSR    PC, RAMDAT     ;R4=CRAM PC (LSB 8 BITS)
6962 021050 000001      1           ;EXPECTED DATA
6963 021052 120504      CMPB   R5, R4        ;IS ROM PC CORRECT?
6964 021054 001406      BEQ    2$           ;BR IF YES
6965 021056      ERROR    5         ;ERROR, CRAM PC IS WRONG
      (5) 021062 104455      TRAP   C$ERDF
      (6) 021064 000005      .WORD  5
      (6) 021066 005041      .WORD  EMO
      (6) 021070 006434      .WORD  ERR5
6966 021072      2$:    ESCAPE SEG
      (3) 021072 104410      TRAP   C$ESCAPE
      (3) 021074 000002      .WORD  10000$-.
6967 021076      10000$:  ENDSEG
      (3) 021076 104405      TRAP   C$ESEG
6968 021100      BGNSEG
      (3) 021100 104404      TRAP   C$BSEG
6969 021102      SKIP06 6$
      (1)           ;GOTO 6$ IF M8206
6970 021112 004737 003152      JSR    PC, CLRALL     ;CLEAR ALL CONDITIONS
6971 021116      SROMCLK      ;NEXT WORD IS INSTRUCTION,
      (1) 021116 004537 003064      JSR    R5, .SROMCLK
6972 021122 100403      100403      ;START AT ROM PC=3
6973 021124      SROMCLK      ;NEXT WORD IS INSTRUCTION,
      (1) 021124 004537 003064      JSR    R5, .SROMCLK
6974 021130 103400      100000! <400*7>      ;JUMP TO ROM PC OF 0
6975 021132 004737 003314      JSR    PC, RAMDAT     ;R4=CRAM PC (LCB 8 BITS)
6976 021136 000004      4           ;EXPECTED DATA
6977 021140 120504      CMPB   R5, R4        ;IS ROM PC CORRECT?
6978 021142 001406      BEQ    4$           ;BR IF YES
6979 021144      ERROR    5         ;ERROR, CRAM PC IS WRONG
      (5) 021150 104455      TRAP   C$ERDF
      (6) 021152 000005      .WORD  5
      (6) 021154 005041      .WORD  EMO
      (6) 021156 006434      .WORD  ERR5
6980 021160      4$:    ESCAPE SEG
      (3) 021160 104410      TRAP   C$ESCAPE
      (3) 021162 000002      .WORD  10001$-.
6981 021164      10001$:  ENDSEG
      (3) 021164 104405      TRAP   C$ESEG
6982 021166      BGNSEG
      (3) 021166 104404      TRAP   C$BSEG
6983 021170 004737 003152      JSR    PC, CLRALL     ;CLEAR ALL CONDITIONS
6984 021174      SROMCLK      ;NEXT WORD IS INSTRUCTION,
      (1) 021174 004537 003064      JSR    R5, .SROMCLK
  
```

```
6985 021200 100406          100406          ;START AT ROM PC=6
6986 021202          SROMCLK          ;NEXT WORD IS INSTRUCTION,
(1) 021202 004537 003064    JSR R5,SROMCLK
6987 021206 107525          104125!<400*7> ;JUMP TO ROM PC OF 525
6988 021210 004737 003314    JSR PC,RAMDAT    ;R4=CRAM PC (LSB 8 BITS)
6989 021214 000007          7              ;EXPECTED DATA
6990 021216 120504          CMPB R5,R4      ;IS ROM PC CORRECT?
6991 021220 001406          BEQ 6$         ;BR IF YES
6992 021222          ERROR 5          ;ERROR, CRAM PC IS WRONG
(5) 021226 104455          TRAP C$ERDF
(6) 021230 000005          .WORD 5
(6) 021232 005041          .WORD EMO
(6) 021234 006434          .WORD ERR5
6993 021236          6$: ESCAPE SEG
(3) 021236 104410          TRAP C$ESCAPE
(3) 021240 000002          .WORD 10002$-.
6994 021242          ENDSEG
(3) 021242          10002$:
(3) 021242 104405          TRAP C$ESEG
6995 021244          ENDTST
(3) 021244          L10072:
(3) 021244 104401          TRAP C$ETST
6996
6997 021246          BADHEAD
(2)          ;***** TEST 25 *****
6998          ;*
6999          ;*MAIN MEMORY PAGE DUAL ADDRESS TEST.
7000          ;*IN THIS TEST WE WILL VERIFY THAT PAGES DO
7001          ;*NOT DUAL ADDRESS. THIS TEST IS DIFFERENT FROM THE
7002          ;*PREVIOUS DUAL ADDRESS TESTS IN THAT THE OTHER
7003          ;*TEST REALLY DIDN'T CHECK PAGE DUAL ADDRESSING
7004 021246          BADHEAD
(2)          ;***** TEST 25 *****
7005
7006 021246          BGNTST
(3) 021246          T25::
7007 021246          K4ONLY          ;FOR 4K CPUS ONLY.
(1)          ;DO NOT DO TEST IF M8200, OR M8204
(4) 021256 104432          TRAP C$EXIT
(4) 021260 000156          .WORD L10073-.
7008 021262          MYINT
(1) 021262 013701 002516    MOV KMCSR,R1    ;RECORD DEVICE ADDR.
7009 021266          MSTCLR
7010 021272 005002          CLR R2          ;R2 WILL BE PAGE #
7011 021274 042737 000037 021320 1$: BIC #37,2$     ;CLEAR UNUSED BITS
7012 021302 050237 021320    BIS R2,2$      ;ADD CURRENT PAGE MARKER.
7013 021306          ROMCLK          ;SET ADDR D
(1) 021306 004537 003030    JSR R5,ROMCLK  ;CLOCK INSTRUCTION
7014 021312 010000          10000
7015 021314          ROMCLK          ;OF PAGE X
(1) 021314 004537 003030    JSR R5,ROMCLK  ;CLOCK INSTRUCTION
7016 021320 004000          2$: 4000      ;THIS LOCATION MODIFIED BY LOST
7017          ;FEW INSTRUCTIONS
7018 021322 010261 000004    MOV R2,4(R1)   ;PUT PAGE # INTO PART 4
7019 021326          ROMCLK          ;CLOCK PART 4 INTO MEMORY
(1) 021326 004537 003030    JSR R5,ROMCLK  ;CLOCK INSTRUCTION
```

```

7020 021332 122500          122500          :WHOSE PAGE # IS IN R2
7021 021334 005202          INC R2          :UPDATE PAGE #
7022 021336 032702 000020  BIT #20,R2     :DONE ALL PAGES?
7023 021342 001754          BEQ 1$         :NO-DC NEXT ONE
7024
7025
7026
7027
7028
7029
7030 021344 005002          CLR R2         :R2 STILL HAS PAGE NUMBER
7031
7032 021346 042737 000037 021364 3$: BIC #37,4$
7033 021354 050237 021364  BIC R2,4$
7034 021360          ROMCLK          :LOAD PAGE NUMBER
(1) 021360 004537 003030  JSR R5,ROMCLK :CLOCK INSTRUCTION
7035 021364 004000          4$: 4000
7036 021366          ROMCLK          :MOVE MEM TO PART 4
(1) 021366 004537 003030  JSR R5,ROMCLK :CLOCK INSTRUCTION
7037 021372 041224          041224
7038 021374 116104 000004  MOVB 4(R1),R4 :'FOUND'
7039 021400 110205          MOVB R2,R5     :'EXPECTED'
7040 021402 120504          CMPB R5,R4     :ADDRESS PROBLEM?
7041 021404 001406          BEQ 5$
7042
7043 021406          ERROR 13          :PAGE ADDRESSING ERROR IN MAIN
(5) 021412 104455          TRAP C$ERDF
(6) 021414 000015          .WORD 13
(6) 021416 005041          .WORD EMO
(6) 021420 007420          .WORD ERR13
7044
7045
7046
7047
7048 021422          5$: ESCAPE TST
(3) 021422 104410          TRAP C$ESCAPE
(3) 021424 000012          .WORD L10073-.
7049 021426 005202          INC R2          :UPDATE PAGE ADDRESS
7050 021430 032702 000020  BIT #20,R2     :ALL DONE?
7051 021434 001744          BEQ 3$         :NO-CHECK NEXT PAGE.
7052
7053 021436          ENDTST:
(3) 021436          L10073:
(3) 021436 104401          TRAP C$ETST
7054
7055
7056 021440          BADHEAD
(2)
7057
7058
7059
7060
7061
7062
7063
7064 021440          :***** TEST 26 *****
          :*
          :*JUMP FIELD,PAGE TEST
          :*
          :*IN THIS TEST WILL MAKE SURE A JUMP FIELD INSTRUCTION
          :*WORKS. TO DO THIS, WE'LL PUT THE DESIRED PAGE,FIELD
          :*INFORMATION IN IBUS*<13> THEN ISSUE A JUMP FIELD
          :*THEN WE'LL READ PC REG. AND VERIFY.
          BADHEAD
  
```

\*\*\*\*\* TEST 26 \*\*\*\*\*

```
(2)
7065
7066 021440          BGNTST
(3) 021440          T26::
7067 021440          K4ONLY                ;FOR 4K CPUS ONLY
(1) ;DO NOT DO TEST IF M8200, OR M8204
(4) 021450 104432   TRAP C$EXIT
(4) 021452 000132   .WORD L10074-.
7068 021454          MYINT
(1) 021454 013701 002516 MOV KMCSR,R1                ;RECORD DEVICE ADDR.
7069 021460          MSTCLR
7070
7071 021464 005002   CLR R2                ;R2 TO CONTAIN FIELD #
7072
7073 021466 042737 000017 021504 1$: BIC #17,2$           ;CLEAR ANY JUNK
7074 021474 050237 021504          BIS R2,2$           ;SET FIELD # INTO INSTR.
7075
7076 021500          ROMCLK                ;CLOCK FIELD BITS INTO BREG.
(1) 021500 004537 003030          JSR R5,ROMCLK        ;CLOCK INSTRUCTION
7077 021504 000400          000400                ;CONTAINS FIELD,PAGE BITS
7078 021506          ROMCLK                ;XFERR BREG INTO IBUS*<13>
(1) 021506 004537 003030          JSR R5,ROMCLK        ;CLOCK INSTRUCTION
7079 021512 061233          061233                ;GET INSTRUCTION CLOCKED.
7080 021514          SROMCLK               ;BAS FORM FOR JUM FIELD INSTR.
(1) 021514 004537 003064          JSR R5,SROMCLK
7081 021520 100000          100000
7082
7083
7084 021522 142761 000002 000001 BICB #BIT1,1(R1)      ;CLEAR ROMI
7085 021530          ROMCLK                ;CLOCK NEXT INSTR.
(1) 021530 004537 003030          JSR R5,ROMCLK        ;CLOCK INSTRUCTION
7086 021534 121264          121264                ;MOVE IBUS*TO PORT 4
7087 021536 116104 000004          MOVB 4(R1),R4        ;GET IT.
7088 021542 042704 177760          BIC #^C<17>,R4
7089 021546 120402          CMPB R4,R2          ;FIELD OK?
7090 021550 001407          BEQ 3$            ;IF OK GO AHEAD
7091 021552 010205          MOV R2,R5
7092 021554          ERROR 12              ;CHANGE FIELD INSTRUCTION
(5) 021560 104455          TRAP C$ERDF
(6) 021562 000014          .WORD 12
(6) 021564 005041          .WORD EMO
(6) 021566 007276          .WORD ERR12
7093
7094
7095
7096 021570          ESCAPE TST
(3) 021570 104410          TRAP C$ESCAPE
(3) 021572 000012          .WORD L10074-.
7097
7098
7099 021574 005202          INC R2
7100 021576 032702 000020          BIT #20,R2
7101 021602 001731          BEQ 1$
7102
7103 021604          ENDTST
(3) 021604          L10074:
```

(3) 021604 104401  
7104  
7105 021606  
(2)  
7106  
7107  
7108  
7109  
7110  
7111  
7112  
7113  
7114  
7115  
7116  
7117  
7118  
7119  
7120  
7121  
7122  
7123  
7124  
7125  
7126  
7127  
7128  
7129  
7130  
7131  
7132  
7133  
7134  
7135  
7136 021606  
(2)  
7137  
7138 021606  
(3) 021606  
7139 021606  
(1) 021606 013701 002516  
7140 021612  
(1)  
(4) 021622 104432  
(4) 021624 000336  
7141 021626  
7142  
7143 021632 012737 000000 002406  
7144  
7145  
7146  
7147 021640 012702 000000  
7148  
7149  
7150 021644 012737 000000 002412  
7151

```
TRAP C$ETST  
BADHEAD  
:***** TEST 27 *****  
:  
:*JUMP TEST, JUMP ALWAYS, JUMP CHANGE FIELD  
:  
:*IN THIS TEST, WE WILL CHECK THE ABILITY OF THE  
:*MICRO PROCESSOR TO JUMP (BRANCH & ALWAYS INSTRUCTION)  
:*TO LOCATIONS, FIELDS FROM OTHER LOCATIONS FIELDS.  
:*WE ALREADY KNOW THAT THE BRANCH INSTR WORKS FROM  
:*OTHER TEST. PROCEDURE:  
:* 1. START ADDR 0, FIELD 0  
:* 2. **CALCULATE NEW ADDR, FIELD VIA INC,  
:* 3. CAUSE JUMP (BRANCH) TO NEW ADDRESS  
:* 4. READ PC FROM IBUS*12 AND IBUS*13  
:* 5. REPEAT STEP 2-4 256.TIMES  
:  
:* TO CALCULATE NEW ADDRESS:  
:* 1. INC LOW BYTE OF ADDRESS FOR PC ADDRESS 0-7  
:* 2. INC LOW BYTE OF NADDRESS FOR PC ADDRESS 8-11  
:* BITS REPRESENTED AS BITS 0-3. WHEN 0-3 OVERFLOWS,  
:* RESTARTS AT ZERO.  
:* NET RESULT IS JUMPS FROM:  
:* FIELD,PAGE LOC  
:* 0 0  
:* 1 1  
:* 2 2  
:* 3 3  
:* 10 7  
:* 11 11  
:* :TO :  
:* 17 377  
:  
BADHEAD  
:***** TEST 27 *****  
:  
BGNTST  
T27::  
MYINT  
MOV KMCSR,R1 ;RECORD DEVICE ADDR.  
K4ONLY ;4K CPUS ONLY.  
:DO NOT DO TEST IF M8200, OR M8204  
TRAP C$EXIT  
.WORD L10075-  
MSTCLR  
MOV #0, FLAG ;FLAG TO REPRESENT  
;FIELD,PAGE  
;TO VARIE STARTING PAGE,FIELD,  
;CHANGE #0 PORTION OF INSTR.  
MOV #0, R2 ;R2 TO CONTAIN JUMPED  
;TO CHANGE STARTING IMM ADDR.,  
;VARIE #0 PORTIONS OF INSTR.  
MOV #0, FADR ;ADDRESS  
;LOOP HERE
```

```

7152 021652
7153 021652 042737 000017 021712 1$: BIC #17,2$ ;CLEAR JUNK FROM FIELD
7154 ;PORTION OF CHANGE FIELD INSTR
7155 021660 013700 002406 MOV FLAG,R0 ;INORDER TO INC, DEC FIELD,PAGE
7156 021664 042700 177760 BIC #^C<17>,R0
7157 021670 050037 021712 BIS R0,2$ ;NOW POSITION IN INSTR.
7158 021674 042737 077777 021726 BIC #077777,3$ ;NOW FOR IMMED. BR INSTR.
7159 021702 050237 021726 BIS R2,3$ ;NOW ADD IMMEDIATE ADDR
7160
7161
7162
7163 021706 ROMCLK
(1) 021706 004537 003030 JSR R5,..ROMCLK ;CLOCK INSTRUCTION
7164 021712 000400 2$: 000400 ;MOVE PAGE,FIELD # TO BREG.
7165 021714 ROMCLK
(1) 021714 004537 003030 JSR R5,..ROMCLK ;CLOCK INSTRUCTION
7166 021720 061233 61233 ;MOV BREG TO PC HIGH REG.
7167 021722 SROMCLK
(1) 021722 004537 003064 JSR R5,..SROMCLK
7168 021726 100000 3$: 100000 ;NOW CLOCK IT IN BY JMP FIELD INSTR.
7169
7170 021730 ROMCLK
(1) 021730 004537 003030 JSR R5,..ROMCLK ;READ PC REG HI
7171 021734 121265 121265 ;CLOCK INSTRUCTION
7172 021736 ROMCLK
(1) 021736 004537 003030 JSR R5,..ROMCLK ;READ PC REG LOW
7173 021742 121244 121244 ;CLOCK INSTRUCTION
7174
7175 021744 016104 000004 MOV 4(R1),R4 ;READ PC REG (NOW IN SEL 4)
7176 021750 042704 170000 BIC #170000,R4 ;STRIP FOR ONLY PAGE,FIELD BITS.
7177
7178 021754 013705 021712 MOV 2$,R5 ;NOW FROM ADDR WE WANTED TO
7179 021760 000305 SWAB R5 ;JUMP TO
7180 021762 042705 170377 BIC #170377,R5 ;CLEAR JUNK
7181 021766 050205 BIS R2,R5 ;ADD IMMED ADDR
7182 021770 SKIP06 5$
(1) ;GOTO 5$ IF M8206
7183 022000 105205 INCB R5 5$: ;UPDATE ADDR. EXPECTED SENCE THE READ
7184 022002 ;OF THE IBUS <13> INC THE PC.
7185
7186
7187 022002 020504 CMP R5,R4 ;JUMP GO OK?
7188 022004 001406 BEQ 4$ ;YEA, CONTINUES
7189 022006 ERROR 15 ;FAILED TO JUMP PROPERLY.
(5) 022012 104455 TRAP C$ERDF
(6) 022014 000017 .WORD 15
(6) 022016 005041 .WORD EMO
(6) 022020 007664 .WORD ERR15
7190
7191
7192
7193
7194
7195
7196
7197

```

```

;'FROM ADDR' REPRESENTS
;THE ADDRESS WE STARTED AT
;'TO ADDR' REPRESENTS WHERE
;WE EXPECTED TO JUMP TO,
;'BAD ADDR' REPRESENTS WHERE
;WE WENT TO.

```

.REM %

7198  
 7199  
 7200  
 7201  
 7202  
 7203  
 7204  
 7205  
 7206  
 7207  
 7208  
 7209  
 7210  
 7211  
 7212  
 7213  
 7214  
 7215  
 7216  
 7217  
 7218  
 7219  
 7220  
 (3)  
 (3)  
 7221  
 7222  
 7223  
 7224  
 7225  
 7226  
 7227  
 7228  
 7229  
 7230  
 (1)  
 7231  
 7232  
 7233  
 7234  
 7235  
 7236  
 7237  
 7238  
 7239  
 7240  
 7241  
 7242  
 (1)  
 7243  
 7244  
 (1)  
 7245  
 7246  
 (1)  
 7247

022022 104410  
 022022 000136  
 022026 010437 002412  
 022032 005237 002406  
 022036 105202  
 022040 001304  
 022042  
 022052 005737 002470  
 022056 001041  
 022060 005737 002472  
 022064 001036  
 022066 052711 040000  
 022072 105761 000001  
 022076 042711 040000  
 022102 004537 003030  
 022106 121265  
 022110 004537 003030  
 022114 121244  
 022116 004537 003030  
 022122 121265

%  
 4\$:

15	12	11	10	9	8	7	0
:UNUSED :		FIELD		PAGE		IMMED. ADDR	
:		BITS		BITS		:	

:THIS IS A PICTURE OF THE P.C. REG.  
 BITS 0-7 ARE IN IBUS\*<12>  
 BITS 8-11 ARE IN IBUS\*<13>  
 THEY GOT CLOCK IN THERE VIA JUMPS TAKEN  
 THE FIELD BITS  
 ARE IN BIT POSITION 0,1 OF THE INSTRUCTION AT 2\$.

3\$ WAS THE JUMP ALWAYS INSTRUCTION. THE IMMED. ADDR.  
 WAS IN 0-7 OF THE JUMP INSTR. THE PAGE BITS,  
 PC REG BITS 8,9, WERE IN BITS 11,12 OF THE INSTR.  
 JUMP INSTRUCTIONS HAVE BEEN CHECKED OUT  
 BEFORE, SO THE IMPORTANT THING TO REMEMBER TO  
 WATCH IS THE 'FROM ADDR','TO ADDR'

```

ESCAPE TST
TRAP C$ESCAPE
.WORD L10075-.
MOV R4,FADR
INC FLAG           ;UPDATE PAGE,FIELD
INCB R2           ;UPDATE IMMED. ADDR
BNE 1$           ;LOOP IF NOT DONE.

;*
;*CHECK HERE TO SEE IF MASTER CLEAR CLEARS P.C. REG
;*

SKIP06 40$
;GOTO 40$ IF M8206
TST RUNB
BNE 40$
TST RUNINH
BNE 40$
BIS #40000,(R1) ;SET MASTER CLEAR
TSTB 1(R1)
BIC #40000,(R1)

;TO RUN THIS SECTION OF CODE YOU MUST TURN SW7 OF SWITCH PACK #E28
;OFF SO THAT M8207 NOT SELFSTARTING.

ROMCLK
JSR R5,..ROMCLK ;WE MUST FIRST CLOCK
121265           ;CLOCK INSTRUCTION
ROMCLK          ;THE PC LATCH REGS
JSR R5,..ROMCLK ;BEFORE WE CAN READ THEM
121244           ;CLOCK INSTRUCTION
ROMCLK          ;REG PC REG HI, PUT IN PORTS
JSR R5,..ROMCLK ;CLOCK INSTRUCTION
121265
  
```

```

7248
7249 022124          ROMCLK          :REG PC REG LOW, PUT IN PORT4
(1) 022124 004537 003030 JSR      R5,.ROMCLK      :CLOCK INSTRUCTION
7250 022130 121244 CLR      R5              :EXPECT ZERO
7251 022132 005005 MOV      4(R1),R4        :READ PC REG FROM PORT 4&5
7252 022134 016104 BIC      #170003,R4
7253 022140 042704 BEQ      40$
7254 022144 001406
7255
7256
7257
7258
7259
7260
7261 022146          ERROR 45          :MASTER CLEAR FAILED TO CLEAR
(5) 022152 104455 TRAP    C$ERDF
(6) 022154 000055 .WORD  45
(6) 022156 005041 .WORD  EMO
(6) 022160 011002 .WORD  ERR45
7262
7263 022162          40$:
7264 022162          ENDTST
(3) 022162          L10075:
(3) 022162 104401 TRAP    C$ETST
7265 022164          BADHEAD
(2)
7266
7267
7268
7269
7270
7271
7272
7273
7274
7275
7276
7277
7278
7279
7280
7281
7282
7283
7284
7285
7286
7287
7288
  
```

```

:REG PC REG LOW, PUT IN PORT4
:CLOCK INSTRUCTION
:EXPECT ZERO
:READ PC REG FROM PORT 4&5
:IF CLEARED, EXIT
:NOTE WE ALSO CLEARED BIT 1 OF THE
:PC REG, BECAUSE AFTER THE MASTER
:CLEAR, WE DID TWO INSTRUCTIONS TO
:READ IT, THUS CAUSING THE PC REG
:TO GET BUMPED.
:MASTER CLEAR FAILED TO CLEAR
:PC REG
:***** TEST 28 *****
:*
:*JUMP TEST, JUMP ALWAYS, JUMP CHANGE FIELD
:*
:*IN THIS TEST, WE WILL CHECK THE ABILITY OF THE
:*MICRO PROCESSOR TO JUMP (BRANCH & ALWAYS INSTRUCTION)
:*TO LOCATIONS, FIELDS FROM OTHER LOCATIONS FIELDS.
:*WE ALREADY KNOW THAT THE BRANCH INSTR WORKS FROM
:*OTHER TEST.
PROCEDURE:
1. START ADDR 0, FIELD 0
2. **CALCULATE NEW ADDR, FIELD VIA DEC,
3. CAUSE JUMP (BRANCH) TO NEW ADDRESS
4. READ PC FROM IBUS*12 AND IBUS*13
5. REPEAT STEP 2-4 256.TIMES
TO CALCULATE NEW ADDRESS:
1. DEC LOW BYTE OF ADDRESS FOR PC ADDRESS 0-7
2. DEC LOW BYTE OF NADDRESS FOR PC ADDRESS 8-11
BITS REPRESENTED AS BITS 0-3. WHEN 0-3 OVERFLOWS,
RESTARTS AT ZERO.
NET RESULT IS JUMPS FROM:
FIELD,PAGE LOC
0 0
17 377
  
```



```

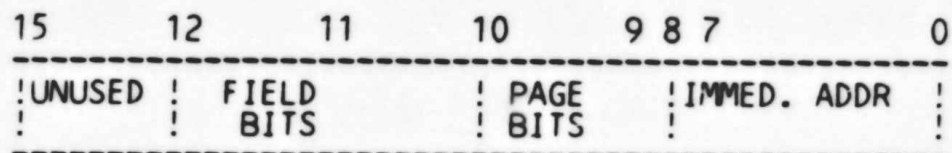
7290          : *          16          376
7291          : *          15          375
7292          : *          :TO
7293          : *          00          000
7294          : *
7295 022164    BADHEAD
(2)           ;***** TEST 28 *****
7296
7297 022164    BGNTST
(3) 022164    T28::
7298 022164    MYINT
(1) 022164 013701 002516  MOV KMCSR,R1 ;RECORD DEVICE ADDR.
7299 022170    K4ONLY ;4K CPUS ONLY.
(1)           ;DO NOT DO TEST IF M8200, OR M8204
(4) 022200 104432  TRAP C$EXIT
(4) 022202 000216  .WORD L10076-.
7300 022204    MSTCLR
7301
7302 022210 012737 000000 002406  MOV #0, FLAG ;FLAG TO REPRESENT
7303          ;FIELD,PAGE
7304          ;TO VARIE STARTING PAGE,FIELD,
7305          ;CHANGE #0 PORTION OF INSTR.
7306 022216 012702 000000  MOV #0, R2 ;R2 TO CONTAIN JUMPED
7307          ;TO CHANGE STARTING IMM ADDR.,
7308          ;VARIE #0 PORTIONS OF INSTR.
7309 022222 012737 000000 002412  MOV #0, FADR ;ADDRESS
7310          ;LOOP HERE
7311 022230
7312 022230 042737 000017 022270 1$: BIC #17,2$ ;CLEAR JUNK FROM FIELD
7313          ;PORTION OF CHANGE FIELD INSTR
7314 022236 013700 002406  MOV FLAG,R0 ;INORDER TO INC, DEC FIELD,PAGE
7315 022242 042700 177760  BIC #^C<17>,R0
7316 022246 050037 022270  BIS R0,2$ ;NOW POSITION IN INSTR.
7317 022252 042737 077777 022304  BIC #077777,3$ ;NOW FOR IMMED. BR INSTR.
7318 022260 050237 022304  BIS R2,3$ ;NOW ADD IMMEDIATE ADDR
7319
7320
7321
7322 022264
(1) 022264 004537 003030  ROMCLK
7323 022270 000400 2$: JSR R5,..ROMCLK ;CLOCK INSTRUCTION
7324 022272 000400 ;MOVE PAGE,FIELD # TO BREG.
(1) 022272 004537 003030  ROMCLK
7325 022276 061233 JSR R5,..ROMCLK ;CLOCK INSTRUCTION
7326 022300 61233 ;MOV BREG TO PC HIGH REG.
(1) 022300 004537 003064  SROMCLK
7327 022304 100000 3$: JSR R5,..SROMCLK
7328          ;NOW CLOCK IT IN BY JMP FIELD INSTR.
7329 022306
(1) 022306 004537 003030  ROMCLK ;READ PC REG HI
7330 022312 121265 JSR R5,..ROMCLK ;CLOCK INSTRUCTION
7331 022314 121265
(1) 022314 004537 003030  ROMCLK ;READ PC REG LOW
7332 022320 121244 JSR R5,..ROMCLK ;CLOCK INSTRUCTION
7333
7334 022322 016104 000004  MOV 4(R1),R4 ;READ PC REG (NOW IN SEL 4)
  
```

7335 022326 042704 170000  
 7336  
 7337 022332 013705 022270  
 7338 022336 000305  
 7339 022340 042705 170377  
 7340 022344 050205  
 7341 022346  
 (1)  
 7342 022356 105205  
 7343 022360  
 7344  
 7345  
 7346 022360 020504  
 7347 022362 001406  
 7348 022364  
 (5) 022370 104455  
 (6) 022372 000017  
 (6) 022374 005041  
 (6) 022376 007664

BIC #170000,R4 ;STRIP FOR ONLY PAGE, FIELD BITS.  
 MOV 2\$,R5 ;NOW FROM ADDR WE WANTED TO  
 SWAB R5 ;JUMP TO  
 BIC #170377,R5 ;CLEAR JUNK  
 BIS R2,R5 ;ADD IMMED ADDR  
 SKIP06 5\$  
 :GOTO 5\$ IF M8206  
 INCB R5 ;UPDATE ADDR. EXPECTED SENCE THE READ  
 5\$: ;OF THE IBUS <13> INC THE PC.  
 CMP R5,R4 ;JUMP GO OK?  
 BEQ 4\$ ;YEA, CONTINUES  
 ERROR 15 ;FAILED TO JUMP PROPERLY.  
 TRAP C\$ERDF  
 .WORD 15  
 .WORD EMO  
 .WORD ERR15

;'FROM ADDR' REPRESENTS  
 ;THE ADDRESS WE STARTED AT  
 ;'TO ADDR' REPRESENTS WHERE  
 ;WE EXPECTED TO JUMP TO,  
 ;'BAD ADDR' REPRESENTS WHERE  
 ;WE WENT TO.

.REM %



:THIS IS A PICTURE OF THE P.C. REG.  
 BITS 0-7 ARE IN IBUS\* <12>  
 BITS 8-11 ARE IN IBUS\* <13>  
 THEY GOT CLOCK IN THERE VIA JUMPS TAKEN  
 THE FIELD BITS  
 ARE IN BIT POSITION 0,1 OF THE INSTRUCTION AT 2\$.

3\$ WAS THE JUMP ALWAYS INSTRUCTION. THE IMMED. ADDR.  
 WAS IN 0-7 OF THE JUMP INSTR. THE PAGE BITS,  
 PC REG BITS 8,9, WERE IN BITS 11,12 OF THE INSTR.  
 JUMP INSTRUCTIONS HAVE BEEN CHECKED OUT  
 BEFORE, SO THE IMPORTANT THING TO REMEMBER TO  
 WATCH IS THE 'FROM ADDR', 'TO ADDR'

%

7379 022400  
 (3) 022400 104410  
 (3) 022402 000016  
 7380 022404 010437 002412  
 7381 022410 005337 002406  
 7382 022414 105302  
 7383 022416 001304

4\$:

ESCAPE TST  
 TRAP C\$ESCAPE  
 .WORD L10076-  
 MOV R4,FADR  
 DEC FLAG ;UPDATE PAGE, FIELD  
 DECB R2 ;UPDATE IMMED. ADDR  
 BNE 1\$ ;LOOP IF NOT DONE.

```

7384
7385
7386 022420          ENDTST
(3) 022420          L10076:
(3) 022420 104401   TRAP   C$ETST
7387 022422          BADHEAD
(2)                   :***** TEST 29 *****
7388                   :*
7389                   :* IN THIS TEST WE'LL VERIFY THAT THE Z BIT CAN BE READ FROM
7390                   :* IBUS*<13>. WE ALLREADY KNOW THAT THE Z BIT WORKS PROPERLY,
7391                   :* ALL WE WANT TO KNOW HERE IS THAT IT CAN BE READ.
7392                   :*
7393 022422          BADHEAD
(2)                   :***** TEST 29 *****
7394
7395 022422          BGNTST
(3) 022422          T29::
7396 022422          K4ONLY                               :M8206 &M8207 ONLY!
(1)                   :DO NOT DO TEST IF M8200, OR M8204
(4) 022432 104432   TRAP   C$EXIT
(4) 022434 000200   .WORD  L10077-.
7397 022436          MSTCLR
7398 022442          MYINT
(1) 022442 013701 002516  MOV   KMCSR,R1           :RECORD DEVICE ADDR.
7399 022446 004737 003152  JSR   PC,CLRALL        :CLR CONDITION CODES.
7400 022452          ROMCLK                               :NOW READ IBUS*<15>PUT IN PORT 4
(1) 022452 004537 003030  JSR   R5,ROMCLK        :CLOCK INSTRUCTION
7401 022456 121264          121264
7402 022460 116104 000004  MOVB  4(R1),R4         :READ IT FROM PORT 4
7403 022464 042704 177477  BIC   #177477,R4      :STRIP ANY JUNK,C&Z BITS 6,7
7404 022470 012705 000000  MOV   #0,R5           :EXPECT IT CLEAR
7405 022474 120405          CMPB  R4,R5           :OK?
7406 022476 001410          BEQ   1$
7407 022500          ERROR 16                               :FAILURE OF Z&C TO BE CLEAR.
(5) 022504 104455          TRAP  C$ERDF
(6) 022506 000020          .WORD 16
(6) 022510 005041          .WORD EMO
(6) 022512 010012          .WORD ERR16
7408
7409 022514          ESCAPE TST
(3) 022514 104410          TRAP  C$ESCAPE
(3) 022516 000116          .WORD L10077-.
7410 022520 004737 003276  JSR   PC,SETZ         :SET Z BIT.
7411 022524          ROMCLK                               :NOW GO BACK AND CHECK Z BIT SET.
(1) 022524 004537 003030  JSR   R5,ROMCLK        :CLOCK INSTRUCTION
7412 022530 121264          121264
7413
7414 022532 016104 000004  MOV   4(R1),R4         :GET INFO.
7415 022536 042704 177477  BIC   #^C<300>,R4     :STRIP FOR C&Z BITS.
7416 022542 012705 000200  MOV   #200,R5         :EXPECT ONLY Z BIT SET.
7417 022546 120405          CMPB  R4,R5           :SET OK?
7418 022550 001410          BEQ   2$
7419 022552          ERROR 16                               :Z BIT FAILED TO SET PROPERLY.
(5) 022556 104455          TRAP  C$ERDF
(6) 022560 000020          .WORD 16
(6) 022562 005041          .WORD EMO
  
```

```
(6) 022564 010012 .WORD ERR16
7420
7421 022566 ESCAPE TST
(3) 022566 104410 TRAP C$ESCAPE
(3) 022570 000044 .WORD L10077-.
7422 022572 004737 003152 2$: JSR PC,CLRALL ;NOW TRY TO CLEAR Z BIT.
7423 022576 ROMCLK
(1) 022576 004537 003030 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
7424 022602 121264 121264
7425 022604 016104 000004 MOV 4(R1),R4
7426 022610 042704 177477 BIC #*C<300>,R4 ;STRIP FOR C&Z BITS
7427 022614 001407 BEQ 3$ ;IF ZERO,WE'RE OK
7428 022616 005005 CLR R5 ;ELSE REPORT ERROR
7429 022620 ERROR 16 ;Z BIT FAILED TO CLEAR PROPERLY.
(5) 022624 104455 TRAP C$ERDF
(6) 022626 000020 .WORD 16
(6) 022630 005041 .WORD EMO
(6) 022632 010012 .WORD ERR16
7430 022634 3$:
7431 022634 L10077: ENDTST
(3) 022634 (3) 022634 104401 TRAP C$ETST
7432 ;FINDFAST
7433 022636 BADHEAD
(2) ;***** TEST 30 *****
7434 ;*
7435 ;* IN THIS TEST WE'LL VERIFY THAT THE C BIT CAN BE READ FROM
7436 ;* IBUS*<13>. WE ALLREADY KNOW THAT THE C BIT WORKS PROPERLY,
7437 ;* ALL WE WANT TO KNOW HERE IS THAT IT CAN BE READ.
7438 ;*
7439 022636 BADHEAD
(2) ;***** TEST 30 *****
7440
7441 022636 BGNTST
(3) 022636 T30::
7442 022636 K4ONLY ;M8206 &M8207 ONLY!
(1) ;DO NOT DO TEST IF M8200, OR M8204
(4) 022646 104432 TRAP C$EXIT
(4) 022650 000200 .WORD L10100-.
7443 022652 MSTCLR
7444 022656 MYINT
(1) 022656 013701 002516 MOV KMCSR,R1 ;RECORD DEVICE ADDR.
7445 022662 004737 003152 JSR PC,CLRALL ;CLR CONDITION CODES.
7446 022666 ROMCLK ;NOW READ IBUS*<13>PUT IN PORT 4
(1) 022666 004537 003030 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
7447 022672 121264 121264
7448 022674 116104 000004 MOV 4(R1),R4 ;READ IT FROM PORT 4
7449 022700 042704 177477 BIC #177477,R4 ;STRIP ANY JUNK,C&Z BITS 6,7
7450 022704 012705 000000 MOV #0,R5 ;EXPECT IT CLEAR
7451 022710 120405 CMPB R4,R5 ;OK?
7452 022712 001410 BEQ 1$
7453 022714 ERROR 16 ;FAILURE OF Z&C TO BE CLEAR.
(5) 022720 104455 TRAP C$ERDF
(6) 022722 000020 .WORD 16
(6) 022724 005041 .WORD EMO
(6) 022726 010012 .WORD ERR16
```

```

7454
7455 022730          ESCAPE TST
      (3) 022730 104410 TRAP C$ESCAPE
      (3) 022732 000116 .WORD L10100-.
7456 022734 004737 003244 1$: JSR PC,SETC          ;SET C BIT.
7457 022740          ROMCLK          ;NOW GO BACK AND CHECK C BIT SET.
      (1) 022740 004537 003030 JSR R5,ROMCLK      ;CLOCK INSTRUCTION
7458 022744 121264          MOV 4(R1),R4          ;GET INFO.
7459 022746 016104 000004 BIC #^C<300>,R4    ;STRIP FOR C&Z BITS.
7460 022752 042704 177477 BIC #100,R5        ;EXPECT ONLY C BIT SET.
7461 022756 012705 000100 MOV R4,R5          ;SET OK?
7462 022762 120405          CMPB R4,R5
7463 022764 001410          BEQ 2$
7464 022766          ERROR 16          ;C BIT FAILED TO SET PROPERLY.
      (5) 022772 104455 TRAP C$ERDF
      (6) 022774 000020 .WORD 16
      (6) 022776 005041 .WORD EMO
      (6) 023000 010012 .WORD ERR16

7465
7466 023002          ESCAPE TST
      (3) 023002 104410 TRAP C$ESCAPE
      (3) 023004 000044 .WORD L10100-.
7467 023006 004737 003152 2$: JSR PC,CLRALL          ;NOW TRY TO CLEAR C BIT.
7468 023012          ROMCLK          ;CLOCK INSTRUCTION
      (1) 023012 004537 003030 JSR R5,ROMCLK
7469 023016 121264          MOV 4(R1),R4
7470 023020 016104 000004 BIC #^C<300>,R4    ;STRIP FOR C&Z BITS
7471 023024 042704 177477 BIC #100,R5        ;IF ZERO,WE'RE OK
7472 023030 001407          BEQ 3$          ;ELSE REPORT ERROR
7473 023032 005005          CLR R5          ;C BIT FAILED TO CLEAR PROPERLY.
7474 023034          ERROR 16
      (5) 023040 104455 TRAP C$ERDF
      (6) 023042 000020 .WORD 16
      (6) 023044 005041 .WORD EMO
      (6) 023046 010012 .WORD ERR16

7475 023050          3$:
7476 023050          ENDTST
      (3) 023050          L10100:
      (3) 023050 104401 TRAP C$SETST
7477 023052          BADHEAD
      (2)          ;***** TEST 31 *****
7478          ;*TEST OF PROGRAM CLOCK BIT
7479          ;*DO A MASTER CLEAR, VERIFY THAT PROGRAM CLOCK IS SET
7480          ;*WRITE PROGRAM CLOCK BIT TO A ONE, VERIFY THAT IT CLEARS,
7481          ;*AND THEN SETS SOME TIME LATER
7482 023052          BADHEAD
      (2)          ;***** TEST 31 *****
7483
7484 023052          BGNTST
      (3) 023052          T31::
7485 023052 013701 002516 MYINT
      (1) 023052          MOV KMCSR,R1          ;RECORD DEVICE ADDR.
7486 023056          MSTCLR          ;MASTER CLEAR M8200,4,7
7487 023062 005037 002440 CLR TEMP          ;PREPARE FOR
7488 023066 005037 002444 CLR $TMP0         ;DELAY
7489 023072 012702 000011 MOV #11,R2        ;SAVE FOR TYPEOUT
  
```

```
7490 023076 ROMCLK ;NEXT WORD IS INSTRUCTION,  
(1) 023076 004537 003030 JSR R5,,ROMCLK ;CLOCK INSTRUCTION  
7491 023102 121224 121224 ;PORT 4 LU11  
7492 023104 016104 000004 MOV 4(R1),R4 ;PUT 'FOUND' IN R4  
7493 023110 042704 000357 BIC #357,R4 ;CLEAR UNWANTED BITS  
7494 023114 012737 000020 002452 MOV #20,$GDDAT ;PUT 'EXPECTED' IN $GDDAT  
7495 023122 123704 002452 CMPB $GDDAT,R4 ;IS PGM CLOCK SET?  
7496 023126 001410 BEQ 1$  
7497 023130 013702 002452 MOV $GDDAT,R2  
7498 023134 ERROR 14 ;ERROR, PGM CLOCK IS NOT SET  
(5) 023140 104455 TRAP C$ERDF  
(6) 023142 000016 .WORD 14  
(6) 023144 005041 .WORD EMO  
(6) 023146 007542 .WORD ERR14  
7499 023150 012761 000020 000004 1$: MOV #20,4(R1) ;LOAD PORT 4  
7500 023156 152761 000002 000001 BISB #BIT1,1(R1) ;SET ROMI  
7501 023164 012761 121111 000006 MOV #121111,6(R1) ;SEL6 INSTRUCTION  
7502 023172 152761 000003 000001 BISB #BIT1!BIT0,1(R1) ;SET CLOCK BIT  
7503 023200 012761 121224 000006 MOV #121224,6(R1) ;LOAD NEXT INSTRUCTION  
7504 023206 152761 000003 000001 BISB #BIT1!BIT0,1(R1) ;READ CLOCK BIT  
7505 023214 142761 030001 000001 BICB #BIT!BIT0,1(R1) ;CLEAR MAINT BITS  
7506 023222 016104 000004 MOV 4(R1),R4 ;PUT 'FOUND' IN R4  
7507 023226 005037 002452 CLR $GDDAT ;PUT 'EXPECTED' IN $GDDAT  
7508 023232 123704 002452 CMPB $GDDAT,R4 ;IS PGM CLOCK CLEAR?  
7509 023236 001410 BEQ 2$  
7510 023240 013702 002452 MOV $GDDAT,R2  
7511 023244 ERROR 14 ;ERROR, PGM CLOCK IS NOT CLEAR  
(5) 023250 104455 TRAP C$ERDF  
(6) 023252 000016 .WORD 14  
(6) 023254 005041 .WORD EMO  
(6) 023256 007542 .WORD ERR14  
7512 023260 2$:  
7513 023260 ROMCLK ;NEXT WORD IS INSTRUCTION,  
(1) 023260 004537 003030 JSR R5,,ROMCLK ;CLOCK INSTRUCTION  
7514 023264 121224 121224 ;PORT4 LU11  
7515 023266 122761 000020 000004 CMPB #20,4(R1) ;IS PGM CLOCK SET?  
7516 023274 001422 BEQ 3$ ;BR IF YES  
7517 023276 005237 002440 INC TEMP ;INCREMENT DELAY  
7518 023302 005537 002444 ADC $TMP0 ;INCREMENT DELAY  
7519 023306 022737 000006 002444 CMP #6,$TMP0 ;IS DELAY DONE  
7520 023314 001361 BNE 2$ ;BR IF NO  
7521 023316 012702 000006 MOV #6,R2  
7522 023322 013704 002444 MOV $TMP0,R4  
7523 023326 ERROR 14 ;ERROR PGM CLOCK NOT SET  
(5) 023332 104455 TRAP C$ERDF  
(6) 023334 000016 .WORD 14  
(6) 023336 005041 .WORD EMO  
(6) 023340 007542 .WORD ERR14  
7524 023342 3$:  
7525  
7526 023342 ENDTST  
(3) 023342 L10101:  
(3) 023342 104401 TRAP C$ETST  
7527  
7528 023344 BADHEAD  
(2) ;***** TEST 32 *****
```



```
7573 023564 013706 002344      MOV    PSTACK,SP      ;RESTORE STACK
7574 023570 032711 004000      4$:  BIT    #BIT11,(R1) ;BIT11 STILL SET?
7575 023574 001016                BNE    7$
7576 023576 005737 002470      TST    RUNB
7577 023602 001013                BNE    7$
7578 023604 011104                MOV    (R1),R4
7579 023606 012705 004000      MOV    #BIT11,R5
7580 023612                ERROR  35              ;OAC FAILED
      (5) 023616 104455          TRAP  C$ERDF
      (6) 023620 000043          .WORD 35
      (6) 023622 005041          .WORD EMO
      (6) 023624 010350          .WORD ERR35
7581                                ;TO PREVENT
7582                                ;INIT FROM
7583                                ;CLEARING CSR
7584 023626                EXIT   TST
      (3) 023626 104432          TRAP  C$EXIT
      (3) 023630 000104          .WORD L10102-
7585 023632 012711 003000      7$:  MOV    #BIT9!BIT10,(R1) ;SEL6 = MAINT IR
7586 023636 012705 121111      MOV    #121111,R5      ;R5 = EXPECTED
7587 023642 016104 000006      MOV    6(R1),R4        ;R4 = FOUND
7588 023646 020504                CMP    R5,R4           ;MAINT IR SHOULD = 12111
7589 023650 001431                BEQ    6$              ;BR IF OK
7590 023652                MSTCLR
7591 023656                ERROR  35              ;IF = 0 THEN BUS INIT WAS
      (5) 023662 104455          TRAP  C$ERDF
      (6) 023664 000043          .WORD 35
      (6) 023666 005041          .WORD EMO
      (6) 023670 010350          .WORD ERR35
7592                                ;NOT BLOCKED FROM CLEARING
7593                                ;THE M8200,4,7
7594
7595 023672 000000                11$: .WORD 0           ;TEMP COUNT FOR STALL ON POWER UP.
7596
7597 023674 052711 040000      10$: BIS    #BIT14,(R1) ;CLR THE THING SO IT CAN'T ASSIRT AC LOW
7598                                ;AGAIN!
7599                                MSTCLR
7600 023704                ERROR  17              ;ERROR GLIP GAVE US SECOUND UNEXPECTED
      (5) 023710 104455          TRAP  C$ERDF
      (6) 023712 000021          .WORD 17
      (6) 023714 005041          .WORD EMO
      (6) 023716 010134          .WORD ERR17
7601                                ;ASSERTION OF AC LOW ON UNIBUS.
7602                                ;FATEL TYPE OF ERROR.
7603 023720 062706 000004      ADD    #4,SP           ;RESTORE STACK.
7604 023724 012637 000024      MOV    (SP)+,@#24
7605 023730                MSTCLR
7606 023734                6$:
7607 023734                ENDTST
      (3) 023734                L10102:
      (3) 023734 104401          TRAP  C$ETST
7608
7609 023736                BADHEAD
      (2)
7610                ;***** TEST 33 *****
7611                ;*MICRO-PROCESSOR NOISE TEST
                ;*WRITE ALL ZERO'S THEN ALL ONE'S THEN A DATA PATTTERN
```



```

7612                                     ;*TO THE IBUS* AND IBUS REGISTERS AND TO THE SP AND MAIN MEM
7613                                     ;*THEN GO BACK AND READ THE DATA PATERNS TO VERIFY THAT
7614                                     ;*READING AND WRITING OF OTHER LOCATIONS AND REGISTERS
7615                                     ;*DID NOT CHANGE THE DATA.
7616 023736                               BADHEAD
(2)                                       ;***** TEST 33 *****
7617
7618 023736                               BGNSTST
(3) 023736                               T33::
7619 023736                               MYINT
(1) 023736 013701 002516                 MOV      KMCSR,R1          ;RECORD DEVICE ADDR.
7620 023742                               MSTCLR                     ;MASTER CLEAR M8200,4,7
7621 023746 005002                               CLR      R2                ;R2 IS INDEX REGISTER
7622 023750 042737 000017 023776 1$:      BIC      #17,2$           ;CLEAR ADDRESS FIELD
7623 023756 156237 024772 023776         BISB     30$(R2),2$       ;ADD IBUS* REG ADDRESS TO INSTRUCTION
7624 023764 116261 025000 000004         MOVB     31$(R2),4(R1)   ;LOAD PORT4
7625 023772                               ROMCLK                     ;NEXT WORD IS INSTRUCTION,
(1) 023772 004537 003030                 JSR      R5,ROMCLK      ;CLOCK INSTRUCTION
7626 023776 121100                               121100                    ;WRITE IBUS* REGISTER
7627 024000 005202                               INC      R2                ;INC INDEX REGISTER
7628 024002 022702 000005                 CMP      #5,R2           ;DONE YET?
7629 024006 001360                               BNE     1$                ;BR IF NO
7630 024010 005002                               CLR      R2                ;R2 IS IBUS REGISTER ADDRESS
7631 024012 042737 000017 024060 3$:      BIC      #17,4$           ;CLEAR ADDRESS FIELD OF INSTRUCTIONS
7632 024020 042737 000017 024074         BIC      #17,5$
7633 024026 042737 000017 024106         BIC      #17,6$
7634 024034 050237 024060                               BIS      R2,4$           ;ADD IBUS REG ADDRESS TO INSTRUCTION
7635 024040 050237 024074                               BIS      R2,5$
7636 024044 050237 024106                               BIS      R2,6$
7637 024050 105061 000004                 CLRB     4(R1)           ;CLEAR PORT4
7638 024054                               ROMCLK                     ;NEXT WORD IS INSTRUCTION,
(1) 024054 004537 003030                 JSR      R5,ROMCLK      ;CLOCK INSTRUCTION
7639 024060 122100                               122100                    ;WRITE 0 TO IBUS REG
7640 024062 112761 000377 000004         MOVB     #377,4(R1)     ;LOAD PORT4
7641 024070                               ROMCLK                     ;NEXT WORD IS INSTRUCTION,
(1) 024070 004537 003030                 JSR      R5,ROMCLK      ;CLOCK INSTRUCTION
7642 024074 122100                               122100                    ;WRITE ALL ONES TO IBUS REG
7643 024076 110261 000004         MOVB     R2,4(R1)       ;LOAD PORT4
7644 024102                               ROMCLK                     ;NEXT WORD IS INSTRUCTION,
(1) 024102 004537 003030                 JSR      R5,ROMCLK      ;CLOCK INSTRUCTION
7645 024106 122100                               122100                    ;WRITE ITS OWN ADDRESS TO IBUS REG
7646 024110 005202                               INC      R2                ;NEXT ADDRESS
7647 024112 022702 000010                 CMP      #10,R2         ;DONE YET?
7648 024116 001335                               BNE     3$                ;BR IF NO
7649 024120 005002                               CLR      R2                ;START AT SP ADDRESS 0
7650 024122 042737 000017 024170 7$:      BIC      #17,8$           ;CLEAR ADDRESS FIELD
7651 024130 042737 000017 024204         BIC      #17,9$
7652 024136 042737 000017 024216         BIC      #17,10$
7653 024144 050237 024170                               BIS      R2,8$           ;ADD ADDRESS TO INSTRUCTION
7654 024150 050237 024204                               BIS      R2,9$
7655 024154 050237 024216                               BIS      R2,10$
7656 024160 105061 000004                 CLRB     4(R1)           ;CLEAR PORT4
7657 024164                               ROMCLK                     ;NEXT WORD IS INSTRUCTION,
(1) 024164 004537 003030                 JSR      R5,ROMCLK      ;CLOCK INSTRUCTION
7658 024170 123100                               123100                    ;WRITE ZERO TO SP
7659 024172 112761 000377 000004         MOVB     #377,4(R1)     ;LOAD PORT4

```

```
7660 024200          ROMCLK           ;NEXT WORD IS INSTRUCTION,  
(1) 024200 004537 003030       JSR      R5,ROMCLK       ;CLOCK INSTRUCTION  
7661 024204 123100          9$: 123100           ;WRITE ALL ONES TO SP  
7662 024206 110261 000004       MOV      R2,4(R1)        ;LOAD PORT4  
7663 024212          ROMCLK           ;NEXT WORD IS INSTRUCTION,  
(1) 024212 004537 003030       JSR      R5,ROMCLK       ;CLOCK INSTRUCTION  
7664 024216 123100          10$: 123100          ;WRITE SP ADDRESS TO ITSELF  
7665 024220 005202          INC      R2               ;NEXT SP ADDRESS  
7666 024222 022702 000020       CMP      #20,R2          ;DONE YET?  
7667 024226 001335          BNE     7$                ;BR IF NO  
7668 024230 005002          CLR     R2                ;R2 = ,AOM ,E , ADDRESS  
7669 024232          ROMCLK           ;NEXT WORD IS INSTRUCTION,  
(1) 024232 004537 003030       JSR      R5,ROMCLK       ;CLOCK INSTRUCTION  
7670 024236 010000          010000          ;MAR _ 0  
7671 024240          ROMCLK           ;CLOCK INSTRUCTION  
(1) 024240 004537 003030       JSR      R5,ROMCLK  
7672 024244 004000          4000  
7673 024246 105061 000004       11$: CLR      4(R1)          ;CLEAR PORT4  
7674 024252          ROMCLK           ;NEXT WORD IS INSTRUCTION,  
(1) 024252 004537 003030       JSR      R5,ROMCLK       ;CLOCK INSTRUCTION  
7675 024256 122500          122500          ;WRITE ZEROS TO MEM  
7676 024260 112761 000377 000004 MOV      #377,4(R1)      ;LOAD PORT4  
7677 024266          ROMCLK           ;NEXT WORD IS INSTRUCTION,  
(1) 024266 004537 003030       JSR      R5,ROMCLK       ;CLOCK INSTRUCTION  
7678 024272 122500          122500          ;WRITE ONES TO MEM  
7679 024274 110261 000004       MOV      R2,4(R1)        ;LOAD PORT4  
7680 024300          ROMCLK           ;NEXT WORD IS INSTRUCTION,  
(1) 024300 004537 003030       JSR      R5,ROMCLK       ;CLOCK INSTRUCTION  
7681 024304 136500          136500          ;WRITE TO MEM IT OWN ADDRESS  
7682 024306 005202          INC     R2                ;NEXT MEM ADDRESS  
7683 024310 022702 001000       CMP      #1000,R2        ;DONE YET?  
7684 024314 001354          BNE     11$              ;BR IF NO  
7685  
7686  
7687                   ;NOW GO BACK AND READ EVERYTHIN  
7688 024316          ROMCLK           ;NEXT WORD IS INSTRUCTION,  
(1) 024316 004537 003030       JSR      R5,ROMCLK       ;CLOCK INSTRUCTION  
7689 024322 010000          010000          ;MAR 0  
7690 024324          ROMCLK           ;NEXT WORD IS INSTRUCTION,  
(1) 024324 004537 003030       JSR      R5,ROMCLK       ;CLOCK INSTRUCTION  
7691 024330 004000          4000             ;MAR HI _ 0 (M8200,4,7 ONLY)  
7692                   ;WOULD BE CRAM CODE  
7693 024332 005737 002432       TST     TYPE  
7694 024336 001452          BEQ     40$  
7695 024340 005005          CLR     R5                ;R5 IS INDEX REGISTER  
7696 024342 042737 000360 024404 12$: BIC     #360,13$         ;CLEAR ADDRESS FIELD  
7697 024350 116502 024772          MOV      30$(R5),R2      ;R2 = IBUS* ADDRESS  
7698 024354 010203          MOV     R2,R3            ;PUT IBUS* ADDRESS IN R3  
7699 024356 006303          ASL     R3                ;SHIFT ADDRESS TO BITS 4-7  
7700 024360 006303          ASL     R3  
7701 024362 006303          ASL     R3  
7702 024364 006303          ASL     R3  
7703 024366 050337 024404          BIS     R3,13$           ;ADD ADDRESS TO INSTRUCTION  
7704 024372 116537 025000 002452 MOV      31$(R5),$GDDAT  ;$GDDAT = 'EXPECTED'  
7705 024400          ROMCLK           ;NEXT WORD IS INSTRUCTION,  
(1) 024400 004537 003030       JSR      R5,ROMCLK       ;CLOCK INSTRUCTION
```

```

7706 024404 121004          13$: 121004          :PORT4 - IBUS* REGISTER
7707 024406 016104 000004      MOV      4(R1),R4      :R4 = 'FOUND'
7708 024412 123704 002452      CMPB    $GDDAT,R4    :IBUS* CONTENTS OK?
7709 024416 001416          BEQ     20$          :BR IF YES
7710 024420 010237 002434      MOV     R2,MRO
7711 024424 105037 002453      CLRB   $GDDAT+1
7712 024430 013705 002452      MOV    $GDDAT,R5
7713 024434          ERROR  29          :IBUS* DATA ERROR
(5) 024440 104455          TRAP   C$ERDF
(6) 024442 000035          .WORD  29
(6) 024444 005041          .WORD  EMO
(6) 024446 010226          .WORD  ERR29
7714 024450          ESCAPE TST
(3) 024450 104410          TRAP   C$ESCAPE
(3) 024452 000334          .WORD  L10103-
7715 024454 005205          20$: INC      R5          :INC COUNTER
7716 024456 022705 000005      CMP    #5,R5        :DONE YET?
7717 024462 001327          BNE    12$          :BR IF NO
7718
7719 024464          40$:
7720          :END CRAM,GENERAL TESTS
7721
7722 024464 005002          CLR     R2          :R2 = IBUS REG ADDRESS
7723 024466 042737 000360 024522 14$: BIC    #360,15$     :CLEAR ADDRESS FIELD OF INSTRUCTION
7724 024474 010203          MOV    R2,R3        :R3 = IBUS ADDRESS
7725 024476 006303          ASL   R3            :SHIFT ADDRESS TO BITS 4-7
7726 024500 006303          ASL   R3
7727 024502 006303          ASL   R3
7728 024504 006303          ASL   R3
7729 024506 050337 024522      BIS    R3,15$       :ADD ADDRESS TO INSTRUCTION
7730 024512 010237 002452      MOV    R2,$GDDAT   :$GDDAT = 'EXPECTED'
7731 024516          ROMCLK
(1) 024516 004537 003030      JSR   R5, .ROMCLK  :NEXT WORD IS INSTRUCTION,
7732 024522 021004          15$: 021004          :CLOCK INSTRUCTION
7733 024524 016104 000004      MOV    4(R1),R4     :PORT4 - IBUS REG
7734 024530 123704 002452      CMPB  $GDDAT,R4    :IBUS = 'FOUND'
7735 024534 001410          BEQ   21$          :IBUS CONTENTS OK?
7736 024536 013705 002452      MOV    $GDDAT,R5   :BR IF YES
7737 024542          ERROR  29          :IBUS DATA ERROR
(5) 024546 104455          TRAP   C$ERDF
(6) 024550 000035          .WORD  29
(6) 024552 005041          .WORD  EMO
(6) 024554 010226          .WORD  ERR29
7738 024556 005202          21$: INC     R2          :NEXT IBUS REGISTER
7739 024560 022702 000010      CMP   #10,R2       :DONE YET?
7740 024564 001340          BNE   14$          :BR IF NO
7741 024566 005002          CLR   R2           :R2 = SP ADDRESS
7742 024570 042737 000017 024606 16$: BIC   #17,17$     :CLEAR ADDRESS FIELD OF INSTRUCTION
7743 024576 050237 024606      BIS   R2,17$       :ADD ADDRESS TO INSTRUCTION
7744 024602          ROMCLK
(1) 024602 004537 003030      JSR   R5, .ROMCLK  :NEXT WORD IS INSTRUCTION,
7745 024606 040600          17$: 040600          :CLOCK INSTRUCTION
7746 024610 010237 002452      MOV    R2,$GDDAT   :BR - SP
7747 024614          ROMCLK
(1) 024614 004537 003030      JSR   R5, .ROMCLK  :$GDDAT = 'EXPECTED'
7748 024620 061224          061224          :NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
:PORT4 - BR
  
```

7749	024622	016104	000004		MOV	4(R1),R4		;R4 = 'FOUND'
7750	024626	123704	002452		CMPB	\$GDDAT,R4		;SP CONTENTS OK?
7751	024632	001412			BEQ	22\$		;BR IF YES
7752	024634	013705	002452		MOV	\$GDDAT,R5		
7753	024640				ERROR	7		;SP DATA ERROR
(5)	024644	104455			TRAP	C\$ERDF		
(6)	024646	000007			.WORD	7		
(6)	024650	005041			.WORD	EMO		
(6)	024652	006704			.WORD	ERR7		
7754	024654				ESCAPE	TST		
(3)	024654	104410			TRAP	C\$ESCAPE		
(3)	024656	000130			.WORD	L10103-		
7755	024660	005202		22\$:	INC	R2		;NEXT SP LOCATION
7756	024662	022702	000020		CMP	#20,R2		;DONE YET?
7757	024666	001340			BNE	16\$		;BR IF NO
7758	024670	005002			CLR	R2		;R2 = MEMORY ADDRESS
7759	024672				ROMCLK			;NEXT WORD IS INSTRUCTION,
(1)	024672	004537	003030		JSR	R5,.ROMCLK		;CLOCK INSTRUCTION
7760	024676	010000				010000		;MAR = 0
7761	024700				ROMCLK			;NEXT WORD IS INSTRUCTION,
(1)	024700	004537	003030		JSR	R5,.ROMCLK		;CLOCK INSTRUCTION
7762	024704	004000				4000		;MAR HI = 0 (M8200,4,7 OR FAMILY ONLY)
7763	024706	010237	002452	18\$:	MOV	R2,\$GDDAT		;R4 = 'FOUND'
7764	024712				ROMCLK			;NEXT WORD IS INSTRUCTION,
(1)	024712	004537	003030		JSR	R5,.ROMCLK		;CLOCK INSTRUCTION
7765	024716	055224				055224		;PORT4 = MAIN MEM
7766	024720	016104	000004		MOV	4(R1),R4		;R4 = 'FOUND'
7767	024724	123704	002452		CMPB	\$GDDAT,R4		;MAIN MEM CONTENTS OK?
7768	024730	001412			BEQ	23\$		;BR IF YES
7769	024732	013705	002452		MOV	\$GDDAT,R5		
7770	024736				ERROR	6		;MAIN MEM DATA ERROR
(5)	024742	104455			TRAP	C\$ERDF		
(6)	024744	000006			.WORD	6		
(6)	024746	005041			.WORD	EMO		
(6)	024750	006556			.WORD	ERR6		
7771	024752				ESCAPE	TST		
(3)	024752	104410			TRAP	C\$ESCAPE		
(3)	024754	000032			.WORD	L10103-		
7772	024756	005202		23\$:	INC	R2		;NEXT MEM ADDRESS
7773	024760	022702	001000		CMP	#1000,R2		;DONE YET?
7774	024764	001350			BNE	18\$		;BR IF NO
7775	024766				EXIT	TST		
(3)	024766	104432			TRAP	C\$EXIT		
(3)	024770	000016			.WORD	L10103-		
7776	024772	000	002	003	.BYTE	0,2,3,5,10		
	024775	005	010					
7777								
7778		025000						
7779	025000	001	003	004	.EVEN			
	025000	006	010		.BYTE	1,3,4,6,10		

```

7781
7782          025006          .EVEN
7783
7784 025006          ENDTST
(3) 025006          L10103:
(3) 025006 104401          TRAP    C$ETST
7785
7786 025010          BADHEAD
(2)
7787          :***** TEST 34 *****
7788          :* THIS TEST IS DESIGNED TO MAKE SURE THAT A NODST INSTRUCTION
7789          :* DOES NOT WRITE INTO PORT B OF THE MULTI PORT RAM.
7790          :* TO DO THIS, WE'LL PUT A 125 INTO INDAT2, THEN WE'LL PUT A
7791          :* 125 INTO BOTH SP1 AND BR. LAST WE'LL DO A NODST BR, SUBOC, SP1
7792          :* IF THERE IS A WRITE INTO PORTB, INDAT2 WILL CONTAIN A 377.
7792 025010          BADHEAD
(2)
7793          :***** TEST 34 *****
7794 025010          BGNTST
(3) 025010          T34::
7795 025010          MYINT
(1) 025010 013701 002516          MOV    KMCSR,R1          ;RECORD DEVICE ADDR.
7796 025014          ROMCLK
(1) 025014 004537 003030          JSR    R5,..ROMCLK      ;CLOCK INSTRUCTION
7797 025020 000525          00525          ;PUT A 125 INTO BRG.
7798 025022          ROMCLK
(1) 025022 004537 003030          JSR    R5,..ROMCLK      ;CLOCK INSTRUCTION
7799 025026 062221          062221          ;NOW INTO OI DAT2
7800 025030          ROMCLK
(1) 025030 004537 003030          JSR    R5,..ROMCLK      ;CLOCK INSTRUCTION
7801 025034 063221          63221          ;NOW INTO SP1
7802 025036          ROMCLK
(1) 025036 004537 003030          JSR    R5,..ROMCLK      ;CLOCK INSTRUCTION
7803 025042 060361          060361          ;NOW THE 'NODST BR, SUBOC, SP1'
7804          ;THE NODST SHOULD NOT MODIFY INDAT2!
7805
7806 025044          ROMCLK
(1) 025044 004537 003030          JSR    R5,..ROMCLK      ;CLOCK INSTRUCTION
7807 025050 020420          020420          ;PUT CONTENT OF INDAT2 IN BRG.
7808
7809 025052          ROMCLK
(1) 025052 004537 003030          JSR    R5,..ROMCLK      ;CLOCK INSTRUCTION
7810 025056 061220          061220          ;PUT BRG INTO BSELO
7811
7812 025060 111104          MOVB   (R1),R4          ;SEE WHAT CAME BACK.
7813 025062 012705 000125          MOV    #125,R5          ;SHOULD BE 125 IF 377 CAME BACK.
7814          ;YOU CAN BET THAT THE 'NODST' WROTE
7815          ;INTO THE MULTI PORT RAM! WATCH SIGNAL
7816          ; 'D1 WRITE OUT L'
7817
7818 025066 020405          CMP    R4,R5          ;NOW LOOK.
7819 025070 001406          BEQ   10$
7820
7821 025072          ERROR  7
(5) 025076 104455          TRAP  C$ERDF
(6) 025100 000007          .WORD 7
(6) 025102 005041          .WORD EMO
    
```

```

(6) 025104 006704 .WORD ERR7
7822
7823 025106 10$:
7824 025106 L10104: ENDTST
(3) 025106 TRAP C$ETST
(3) 025106 104401
7825
7826 025110 BADHEAD
(2) :***** TEST 35 *****
7827 :*
7828 :* EXTENDED CRAM TEST FOR M8206. IN THIS TEST WE WILL LOAD DATA
7829 :* THROUGHOUT THE CRAM (TEST DATA IS JUST 4K OF DIAG. CODE) AND
7830 :* THEN READ IT BACK AND VERIFY THAT IT IS CORRECT.
7831 025110 BADHEAD
(2) :***** TEST 35 *****
7832
7833 025110 T35:: BGNTST
(3) 025110
7834 025110 SKIP06 10$ ;DO TEST ONLY IF IT IS A M8206
(1) :GOTO 10$ IF M8206 ;OTHERWISE,SKIP TEST.
7835 025120 EXIT TST
(3) 025120 104432 TRAP C$EXIT
(3) 025122 000132 .WORD L10105-.
7836
7837 025124 10$: MYINT
(1) 025124 013701 002516 MOV KMCSR,R1 ;RECORD DEVICE ADDR.
7838 MOV #ROMMAP,R2 ;GET ADDR. OF LIST.
7839 025130 012702 012024 MOV #2000,(R1) ;SET TO WRITE DATA.
7840 025134 012711 002000 CLR R3 ;CRAM ADDR ZERO.
7841 025140 005003
7842 025142 010361 000004 15$: MOV R3,4(R1) ;SET ADDR.
7843 025146 012261 000006 MOV (R2)+,6(R1) ;WRITE DATA.
7844 025152 020337 002436 CMP R3,MEMSZ ;DONE WHOLE CRAM?
7845 025156 001402 BEQ 20$ ;YES,EXIT THIS LOOP.
7846 025160 005203 INC R3 ;NO,UPDAT ADDR.
7847 025162 000767 BR 15$
7848 025164 005003 20$: CLR R3 ;NOW WE WILL READ BACK,STARTING AT
7849 025166 012705 012024 MOV #ROMMAP,R5 ;CRAM ADDR. ZERO.
7850 025172 010361 000004 30$: MOV R3,4(R1) ;GET ADDR. LIST OF DATA
7851 025176 011502 MOV (R5),R2 ;SET ADDR.
7852 025200 016104 000006 MOV 6(R1),R4 ;PUT EXPECTED INTO R2
7853 025204 020204 CMP R2,R4 ;READ ACCUAL
7854 025206 001411 BEQ 40$ ;EQUAL?
7855 025210 010300 MOV R3,R0 ;YES,CONTINUE.
7856
7857 025212 ERROR 1 ;ERROR CRAM DATA TEST,DATA
(5) 025216 104455 TRAP C$ERDF
(6) 025220 000001 .WORD 1
(6) 025222 005041 .WORD EMO
(6) 025224 005710 .WORD ERR1
    
```



```

7907 025330 061220           61220           :MOVE BRG,BSELO
7908 025332 061222           61222           :MOVE BRG,BSEL2
7909 025334 061223           61223           :MOVE BRG,BSEL3
7910 025336 061224           61224           :MOVE BRG,BSEL4
7911 025340 061225           61225           :MOVE BRG,BSEL5
7912 025342 061226           61226           :MOVE BRG,,BSEL6
7913 025344 061227           61227           :MOVE BRG,BSEL7
7914 025346 123000           123000          :MOVE BSELO,SPO
7915 025350 104022           104022          :BRANCH BACK ONE LOCATION.
7916 025352 177777           177777
7917
7918 025354 012711 040000     MOV #040000,(R1)  :INITIALIZE MCPU
7919 025360 012711 100000     MOV #100000,(R1)  :START CPU.
7920
7921 025364 012700 000062     MOV #50.,R0       :THE CYCLE TIME ON THE M8206 IS
7922                                     :200NS. WE ARE ASKING THE MCPU TO
7923                                     :DO 8 INSTRUCTIONS. WE'LL DELAY
7924                                     :100 PDP11 INSTRUCTIONS
7925                                     :THIS REALLY SHOULD BE PLENTY OF TIME.
7926
7927 025370 005300           20$: DEC R0
7928 025372 001376           BNE 20$
7929
7930 025374 005005           CLR R5
7931 025376 012705 000377     MOV #377,R5       :JUST FOR TYPEOUT.
7932 025402 111104           MOV (R1),R4       :EXPECT 377
7933 025404 120405           MOV B(R4),R5      :READ MCPU
7934 025406 001410           CMPB R4,R5        :SEE IF OK.
7935
7936 025410           ERROR 29           :ERROR! MCPU WAS TO WRITE ALL
(5) 025414 104455           TRAP C$ERDF
(6) 025416 000035           .WORD 29
(6) 025420 005041           .WORD EMO
(6) 025422 010226           .WORD ERR29
7937
7938 025424           ESCAPE TST
(3) 025424 104410           TRAP C$ESCAPE
(3) 025426 000304           .WORD L10106-.
7939
7940 025430 012705 177777     30$: MOV #177777,R5   :EXPECT ALL ONES
7941 025434 016104 000002     MOV 2(R1),R4      :RECIEVED
7942 025440 020405           CMP R4,R5         :RECIEVE OK?
7943 025442 001410           BEQ 40$
7944
7945 025444           ERROR 29           :ERROR! MCPU WAS TO WRITE ALL ONES
(5) 025450 104455           TRAP C$ERDF
(6) 025452 000035           .WORD 29
(6) 025454 005041           .WORD EMO
(6) 025456 010226           .WORD ERR29
7946
7947                                     :INTO BSEL 2&3
7948 025460           ESCAPE TST
(3) 025460 104410           TRAP C$ESCAPE
(3) 025462 000250           .WORD L10106-.
7949
7950 025464 016104 000004     40$: MOV 4(R1),R4   :READ BSEL 4&5

```



7951	025470	020405			CMP	R4,R5		:READ OK?
7952	025472	001410			BEQ	50\$		
7953								
7954	025474				ERROR	29		:ERROR! FAILED TO WRITE BSEL \$85
(5)	025500	104455			TRAP	C\$ERDF		
(6)	025502	000035			.WORD	29		
(6)	025504	005041			.WORD	EMO		
(6)	025506	010226			.WORD	ERR29		
7955								: TO ALL ONES.
7956	025510				ESCAPE	TST		
(3)	025510	104410			TRAP	C\$ESCAPE		
(3)	025512	000220			.WORD	L10106-		
7957								
7958	025514	016104	000006	50\$:	MOV	6(R1),R4		:READ BSEL 6&7
7959	025520	020405			CMP	R4,R5		:READ OK?
7960	025522	001410			BEQ	60\$		
7961								
7962	025524				ERROR	29		:ERROR! FAILED TO WRITE BSEL 6&7
(5)	025530	104455			TRAP	C\$ERDF		
(6)	025532	000035			.WORD	29		
(6)	025534	005041			.WORD	EMO		
(6)	025536	010226			.WORD	ERR29		
7963								: TO ALL ONES.
7964	025540				ESCAPE	TST		
(3)	025540	104410			TRAP	C\$ESCAPE		
(3)	025542	000170			.WORD	L10106-		
7965	025544	105011		60\$:	CLRB	(R1)		:SIGNAL MCPU TO WRITE ALL ZEROS.
7966	025546	005005			CLR	R5		:EXPECT TO READ ALL ZEROS.
7967								
7968	025550	005004			CLR	R4		
7969	025552	111104			MOVB	(R1),R4		:READ BSELO
7970	025554	001410			BEQ	70\$		:EXPECT ZERO.
7971								
7972	025556				ERROR	29		:MCPU FAILED TO CLEAR BSELO
(5)	025562	104455			TRAP	C\$ERDF		
(6)	025564	000035			.WORD	29		
(6)	025566	005041			.WORD	EMO		
(6)	025570	010226			.WORD	ERR29		
7973								
7974	025572				ESCAPE	TST		
(3)	025572	104410			TRAP	C\$ESCAPE		
(3)	025574	000136			.WORD	L10106-		
7975	025576	016104	000002	70\$:	MOV	2(R1),R4		:READ BSEL 2&3
7976	025602	001410			BEQ	80\$		:IF ZERO,OK
7977								
7978	025604				ERROR	29		:MCPU FAILED TO CLEAR BSEL 2&3
(5)	025610	104455			TRAP	C\$ERDF		
(6)	025612	000035			.WORD	29		
(6)	025614	005041			.WORD	EMO		
(6)	025616	010226			.WORD	ERR29		
7979	025620				ESCAPE	TST		
(3)	025620	104410			TRAP	C\$ESCAPE		
(3)	025622	000110			.WORD	L10106-		
7980	025624							
7981	025624	016104	000004	80\$:	MOV	4(R1),R4		:READ BSEL 4&5
7982	025630	001410			BEQ	90\$		

```
7983
7984 025632          ERROR 29          ;MCPU FAILED TO CLEAR BSEL 4&5
(5) 025636 104455  TRAP  C$ERDF
(6) 025640 000035  .WORD 29
(6) 025642 005041  .WORD EMO
(6) 025644 010226  .WORD ERR29
7985 025646          ESCAPE TST
(3) 025646 104410  TRAP  C$ESCAPE
(3) 025650 000062  .WORD L10106-.
7986 025652
7987 025652 016104 000006 90$: MOV 6(R1),R4          ;READ BSEL 6&7
7988 025656 001406      BEQ 95$
7989
7990 025660          ERROR 29          ;MCPU FAILED TO CLEAR BSEL 6&7
(5) 025664 104455  TRAP  C$ERDF
(6) 025666 000035  .WORD 29
(6) 025670 005041  .WORD EMO
(6) 025672 010226  .WORD ERR29
7991
7992 025674          95$: EXIT TST
(3) 025674 104432  TRAP  C$EXIT
(3) 025676 000034  .WORD L10106-.
7993
7994
7995
7996
7997
7998
7999 025700 012711 002000  LOADER: MOV #2000,(R1)
8000
8001 025704 005000      CLR R0
8002
8003 025706 010061 000004 10$: MOV R0,4(R1)          ;SET ADDR.
8004 025712 005200      INC R0
8005 025714 011561 000006  MOV (R5),6(R1)          ;WRITE MICRO CODE.
8006 025720 022527 177777  CMP (R5)+,#177777      ;SEE IF TERM.
8007 025724 001370      BNE 10$
8008 025726 005011      CLR (R1)
8009 025730 000205      RTS R5
8010
8011 025732          L10106: ENDTST
(3) 025732          TRAP  C$ETST
(3) 025732 104401
8012
8013 025734          BADHEAD
(2)
8014          ;***** TEST 37 *****
8015          ;*
8016          ;*NEGATIVE ADDRESS TEST.
8017          ;*   IN THIS TEST, WE'LL MAKE SURE THAT THE M8207
8018          ;*   DOES NOT RESPOND TO AN ADDRESS THAT ISN'T ASSIGNED
8019          ;*   TO IT
8020 025734          BADHEAD
(2)          ;***** TEST 37 *****
8021
8022 025734          BGNTST
```

```

(3) 025734          T37::
8023 025734          MYINT
(1) 025734 013701 002516      MOV      KMCSR,R1          ;RECORD DEVICE ADDR.
8024
8025 025740 012711 000641      MOV      #641,(R1)        ;PUT A DEFINITE PATTERN IN MCPU.
8026 025744 012737 026022 000004      MOV      #20$,@#4        ;SET UP FOR TRAPS FROM NON-EX.
8027 025752 005037 000006      CLR      @#6
8028 025756 012702 160000      MOV      #160000,R2      ;GET STARTING ADDRESS.
8029
8030 025762 022712 000641      10$:    CMP      #641,(R2)   ;SEE IF CONTENTS OF THE ADDRESS
8031                                     ;POINTED TO BY R2 EQUALS THE CONTENTS
8032                                     ;OF THE MCPU CSR
8033 025766 001420          BEQ      40$
8034
8035 025770 062702 000002      15$:    ADD      #2,R2        ;UPDATE ADDRESS.
8036 025774 020227 177776      CMP      R2,#177776      ;DONE?
8037 026000 001370          BNE     10$              ;NO-LOOP
8038
8039 026002 013737 002464 000004 17$:    MOV      SAVE4,@#4        ;RESTORE TRAP CATCHER
8040 026010 013737 002466 000006      MOV      SAVE6,@#6        ;FROM VALUES SAVED BY INIT SECTION
8041 026016          EXIT      TST
(3) 026016 104432          TRAP    C$EXIT          ;EXIT, ALL DONE
(3) 026020 000052          .WORD  L10107-.
8042
8043 026022 062706 000004      20$:    ADD      #4,SP          ;SAVE FROM TRAP
8044 026026 000760          BR      15$              ;LOOP
8045
8046 026030          40$:    ;*OH NO, WE MAY HAVE A DUAL ADDRESS PROBLEM!
8047
8048 026030 012711 000174      MOV      #174,(R1)        ;WRITE NEW PATTERN IN MCPU CSR
8049 026034 022712 000174      CMP      #174,(R2)        ;DID NEW PATTERN SHOW UP IN ADDR?
8050 026040 001403          BEQ      60$
8051
8052 026042 012711 000641      50$:    MOV      #641,(R1)        ;PUT OLD PATTERN BACK IN MCPU CSR.
8053 026046 000750          BR      15$              ;LOOP
8054
8055 026050 020102          60$:    CMP      R1,R2          ;IS THIS THE MCPU ADDRESS?
8056 026052 001773          BEQ      50$              ;YES-NO ERROR
8057
8058 026054          ERROR  40              ;DUAL ADDRESS ERROR
(5) 026060 104455          TRAP    C$ERDF
(6) 026062 000050          .WORD  40
(6) 026064 005041          .WORD  EMO
(6) 026066 010514          .WORD  ERR40
8059 026070 000744          BR      17$
8060
8061 026072          L10107:  ENDTST
(3) 026072          TRAP    C$ETST
(3) 026072 104401
8062
8063 026074          BADHEAD
(2)
8064          ;***** TEST 38 *****
8065          ;*
8066          ;*BYTE ADDRESSING TEST
8067          ;*
           ;*   HERE, WE'RE GOING TO MAKE SURE THAT WE CAN
           ;*   WRITE INTO ONLY A HIGH OR LOW BYTE OF THE MCPU.

```

```

8068
8069 026074          : *
                        : BADHEAD
                        : ***** TEST 38 *****
8070
8071 026074          : BGNTST
      (3) 026074          T38::
8072 026074          : MYINT
      (1) 026074 013701 002516 : MOV KMCSR,R1          : RECORD DEVICE ADDR.
8073 026100 005061 000002 : CLR 2(R1)            : CLEAR CSR
8074 026104 112761 177777 000002 : MOVB #-1,2(R1)      : WRITE ALL ONES INTO LOW BYTE
8075                                     : OF CSR
8076 026112 032761 177400 000002 : BIT #177400,2(R1)   : SEE IF HIGH BYTE GOT WRITTEN
8077 026120 001410
8078
8079 026122          : ERROR 41             : HIGH BYTE GOT WRITTEN INTO ON A LOW BYTE
      (5) 026126 104455 : TRAP C$ERDF
      (6) 026130 000051 : .WORD 41
      (6) 026132 005041 : .WORD EMO
      (6) 026134 010560 : .WORD ERR41
8080 026136          : ESCAPE TST          : OPERATION
      (3) 026136 104410 : TRAP C$ESCAPE
      (3) 026140 000040 : .WORD L10110-.
8081
8082 026142 005061 000002 10$: CLR 2(R1)
8083 026146 112761 177777 000003 : MOVB #-1,3(R1)      : WRITE INTO HIGH BYTE
8084 026154 032761 000377 000002 : BIT #377,2(R1)     : SEE IF LOW BYTE GOT WRITTEN
8085 026162 001406
8086
8087 026164          : ERROR 42             : LOW BYTE GOT WRITTEN INTO ON A
      (5) 026170 104455 : TRAP C$ERDF
      (6) 026172 000052 : .WORD 42
      (6) 026174 005041 : .WORD EMO
      (6) 026176 010622 : .WORD ERR42
8088                                     : HIGH BYTE OPERATION.
8089
8090 026200          : 20$:
8091 026200          : L10110:
      (3) 026200 104401 : TRAP C$ETST
8092
8093 026202          : BADHEAD
      (2)                                     : ***** TEST 39 *****
8094                                     : *
8095                                     : *IN THIS TEST WE'RE GOING TO MAKE SURE THAT THE PC
8096                                     : *REG COUNTS UP PROPERLY. THE PC REG SHOULD INCREMENT
8097                                     : *ONCE AFTER EACH INSTRUCTION.
8098                                     : *
8099 026202          : BADHEAD
      (2)                                     : ***** TEST 39 *****
8100
8101 026202          : BGNTST
      (3) 026202          T39::
8102 026202          : SKIP07 10$          : ONLY DO IF M8207
      (1)                                     : GOTO 10$ IF M8207
8103 026212 104432 : EXIT TST
      (3) 026212          : TRAP C$EXIT
  
```

```
(3) 026214 000122          .WORD  L10111-.
8104
8105 026216          10$:  MYINT
(1) 026216 013701 002516    MOV    KMCSR,R1          ;RECORD DEVICE ADDR.
8106 026222          MSTCLR
8107 026226          ROMCLK
(1) 026226 004537 003030    JSR    R5,.ROMCLK      ;CLOCK INSTRUCTION
8108 026232 000400      400
8109 026234          ROMCLK
(1) 026234 004537 003030    JSR    R5,.ROMCLK      ;CLOCK INSTRUCTION
8110 026240 061233      61233
8111 026242          SROMCLK
(1) 026242 004537 003064    JSR    R5,.SROMCLK     ;CLOCK INSTRUCTION
8112 026246 100000      100000
8113 026250 012705 000001    MOV    #1,R5           ;START AT ZERO
8114
8115 026254          20$:  ROMCLK          ;READ PC HIGH REG.
(1) 026254 004537 003030    JSR    R5,.ROMCLK      ;CLOCK INSTRUCTION
8116 026260 121265      121265
8117
8118 026262          ROMCLK          ;READ PC LOW REG.
(1) 026262 004537 003030    JSR    R5,.ROMCLK      ;CLOCK INSTRUCTION
8119 026266 121244      121244
8120 026270 016104 000004    MOV    4(R1),R4        ;GET WHOLE PICTURE
8121 026274 042704 170000    BIC    #170000,R4
8122 026300 020405      R4,R5           ;INCREMENT OK?
8123 026302 001410      BEQ    30$
8124
8125 026304          ERROR  47           ;PC FAILED TO INCREMENT PROPERLY
(5) 026310 104455      TRAP   C$ERDF
(6) 026312 000057      .WORD  47
(6) 026314 005041      .WORD  EMO
(6) 026316 011142      .WORD  ERR47
8126
8127
8128 026320          ESCAPE  TST          ;SHOULD INCREMENT BY ONE
(3) 026320 104410      TRAP   C$ESCAPE        ;FOR EACH INSTRUCTION.
(3) 026322 000014      .WORD  L10111-.
8129
8130 026324 062705 000002    30$:  ADD    #2,R5          ;UPDATE EXPECTED ADDRESS BY 2.
8131 026330 020527 000777    CMP    R5,#777
8132 026334 001347      BNE    20$
8133
8134 026336          L10111:  ENDTST
(3) 026336          TRAP   C$ETST
(3) 026336 104401
8135
8136 026340          BADHEAD
(2)
8137          ;***** TEST 40 *****
8138          ;*
8139          ;*IN THIS TEST WE'LL MAKE SURE THAT 'BRANCH FIELD H' DOESN'T
8140          ;*GET SUCH HIGH.
8141          ;*FIRST WE'LL CLEAR THE PC HIGH REG. THEN WE'LL DO A BRANCH INSTR
8142          ;*WITH BAB BITS 11&12 SET. IF PCR BITS 8&9 SET THEN WE'LL KNOW
8143          ;*WE WERE SUCCESSFUL IF PCR BITS 8&9 FAIL TO SET, WE'LL KNOW
          ;*THAT THE MUX SELECTED THE WRONG INPUT TO BE CLOCKED INTO THE PCR.
```

```
8144
8145 026340          :*
(2)                :BADHEAD
8146                :***** TEST 40 *****
8147 026340          BGNTST
(3) 026340          T40::
8148 026340          SKIP07 10$          ;ONLY DO IF M8207
(1)                :GOTO 10$ IF M8207
8149 026350          EXIT TST
(3) 026350 104432   TRAP C$EXIT
(3) 026352 000062   .WORD L10112-.
8150
8151 026354          10$: MYINT          ;INITIALIZE PARAMETERS
(1) 026354 013701 002516 MOV KMCSR,R1          ;RECORD DEVICE ADDR.
8152 026360          MSTCLR          ;CLEAR DEVICE.
8153
8154 026364          ROMCLK          ;DO A 'BRANCH ALWAYS' WITH
(1) 026364 004537 003030 JSR R5,..ROMCLK          ;CLOCK INSTRUCTION
8155 026370 114400          ;BAB BITS 11&12 SET THIS SHOULD CLOCK
8156                ;THESE BITS INTO BITS 8&9 OF THE PCR.
8157 026372          ROMCLK          ;NOW READ THE PCR HIGH
(1) 026372 004537 003030 JSR R5,..ROMCLK          ;CLOCK INSTRUCTION
8158 026376 121265          ;AND PUT INTO PORT5.
8159                ;REG. BR NO CLK OF BAB BITS
8160 026400 116105 000005 MOVB 5(R1),R5          ;READ THE PCR.
8161 026404 112704 000003 MOVB #3,R4          ;EXPECT BITS 8,9 TO BE SET.
8162 026410 042705 000374 BIC #374,R5          ;STRIP ANY JUNK
8163 026414 020405          CMP R4,R5          ;OK?
8164 026416 001406          BEQ 20$
8165
8166 026420          ERROR 15          ;'BRANCH FIELD H' STUCK HIGH OR
(5) 026424 104455   TRAP C$ERDF
(6) 026426 000017   .WORD 15
(6) 026430 005041   .WORD EMO
(6) 026432 007664   .WORD ERR15
8167
8168
8169 026434          20$:
8170 026434          L10112:
(3) 026434 104401   TRAP C$ETST
(3) 026434
8171
8172 026436          BADHEAD
(2)                :***** TEST 41 *****
8173                :*
8174                ;*IN THIS TEST WE'RE GOING TO MAKE SURE THAT ONLY SPO
8175                ;*IS SELECTED FOR SOURCE WHEN THE DESTINATION
8176                ;*IS THE OUTBUS
8177                ;*FIRST WE'LL WRITE EACH SP ADDRS INTO ITSELF THEN WE'LL
8178                ;*MOV SP TO OBUS4. THAT SHOULD SELECT
8179                ;*SP ADDRESS 0. IF ANY OTHER DATA SHOWS UP, WE'LL
8180                ;*BLAME IT ON THE SELECTION OF A DIFFERENT SCRATCH PAD.
8181 026436          BADHEAD
(2)                :***** TEST 41 *****
8182
8183 026436          BGNTST
```

```

(3) 026436          T41::
8184 026436          MYINT
(1) 026436 013701 002516  MOV   KMCSR,R1          ;RECORD DEVICE ADDR.
8185 026442 005005          CLR   R5                ;START WITH ADDR-ZERO
8186
8187 026444 042737 000017 026466 10$:  BIC   #17,20$          ;STRIP SP ADDR FIELD FROM INSTR
8188 026452 010561 000004          MOV   R5,4(R1)          ;PUT SP ADDR INTO PORT4.
8189 026456 050537 026466          BIS   R5,20$          ;ADD SP ADDR TO INSTR.
8190 026462          ROMCLK
(1) 026462 004537 003030          JSR   R5,.,ROMCLK      ;CLOCK INSTRUCTION
8191 026466 123100          20$: 123100          ;WRITE TO SP
8192 026470 005205          INC   R5                ;UPDATE ADDRESS
8193 026472 120527 000020          CMPB R5,#20           ;IF NOT THROUGH, REPEAT.
8194 026475 001362          BNE   10$
8195
8196 026500          ROMCLK
(1) 026500 004537 003030          JSR   R5,.,ROMCLK      ;NOW MOV SPO TO OBUS* PORT4
8197 026504 061204          061204          ;CLOCK INSTRUCTION
8198 026506 116104 000004          MOVB  4(R1),R4          ;
8199 026512 001410          BEQ   30$              ;READ PORT4 IT S/B ZERO
8200 026514 012705 000000          MOV   #0,R5
8201 026520          ERROR  43
(5) 026524 104455          TRAP  C$ERDF           ;SPO NOT SELECTED FOR SOURCE-SEE
(6) 026526 000053          .WORD 43
(6) 026530 005041          .WORD EMO
(6) 026532 010664          .WORD ERR43
8202
8203
8204 026534          30$:  ENDTST
(3) 026534          L10113: TRAP  C$ETST
(3) 026534 104401
8205
8206 026536          BADHEAD
(2)
8207          ;***** TEST 42 *****
8208          ;*
8209          ;*IN THIS TEST WE ARE GOING TO MAKE SURE THAT THE
8210          ;*SIGNAL 'MOV INST H' (AND ITS ASSOC. TRIBS) DOESN'T GET
8211          ;*STUCK HIGH. IN ORDER TO DO THIS WE'LL CLEAR THE PC HIGH REG
8212          ;*PUT KNOWN DATA IN THE BREG AND SP1 THEN WE'LL A BRANCH
8213          ;*WITH CROM BITS 0-3 SET AS WELL AS CROM BIT 9 WITH CROM BITS 8 AND 11 CLEAR.
8214          ;*IF 'MOV INST H' GETS STUCK HIGH, THE PC REG HIGH WILL GET LOADED
8215 026536          ;*WITH THE CONTENTS OF THE ALU
(2)          BADHEAD
8216          ;***** TEST 42 *****
8217 026536          BGNTST
(3) 026536          T42::
8218 026536          SKIP07 10$          ;ONLY DO IF M8207
(1)          ;GOTO 10$ IF M8207
8219 026546          EXIT TST          ;ELSE EXIT
(3) 026546 104432          TRAP  C$EXIT
(3) 026550 000110          .WORD L10114-.
8220
8221 026552          10$:  MYINT
(1) 026552 013701 002516  MOV   KMCSR,R1          ;DO INITIAL TEST SET-UP.
8222 026556          MSTCLR          ;RECORD DEVICE ADDR.
;DO A MASTER CLEAR.
  
```

```

8223 026562 005737 002470      TST      RUNB
8224 026566 001034              BNE      20$
8225
8226                               ;TO RUN THIS SECTION OF CODE YOU MUST TURN SW7 OF SWITCH PACK #E28
8227                               ;OFF SO THAT M8207 NOT SELFSTARTING.
8228
8229 026570 012761 000002 000004  MOV      #2,4(R1)      ;PUT A 2 INTO SP1
8230 026576                      ROMCLK                     ;PORT4 TO SCRATCH PAD 1
      (1) 026576 004537 003030      JSR      R5,..ROMCLK   ;CLOCK INSTRUCTION
8231 026602 123101                      123101
8232 026604 012761 000004 000004  MOV      #4,4(R1)
8233 026612                      ROMCLK
      (1) 026612 004537 003030      JSR      R5,..ROMCLK   ;CLOCK INSTRUCTION
8234 026616 123100                      123100
8235 026620                      ROMCLK
      (1) 026620 004537 003030      JSR      R5,..ROMCLK   ;NOW DO A BRANCH ON C-BIT SET
8236 026624 141201                      141201                    ;CLOCK INSTRUCTION
8237                               ;BASED ON SP CONTENTS
8238                               ;OK-WHAT WE ARE REALLY
8239                               ;INTERESTED IN IS SEEING IF THE
8240                               ;PC HIGH REG GETS LOADED WITH
8241                               ;THE CONTENTS OF THE ALU (2)
8242                               ;IF THIS OCCURS, WE CAN PROBABLY
8243                               ;SAY THAT 'MOV INSTR' REMAINED
8244 026626                      ROMCLK
      (1) 026626 004537 003030      JSR      R5,..ROMCLK   ;HIGH.
8245 026632 121265                      121265                    ;READ PC HIGH, PUT INTO PORT5
8246 026634 116104 000005      MOV      5(R1),R4      ;CLOCK INSTRUCTION
8247 026640 001407                      BEQ      20$            ;READ PC REG HIGH FROM PORT
8248 026642 005005                      CLR      R5            ;SHOULD BE CLEAR
8249
8250 026644                      ERROR    15
      (5) 026650 104455      TRAP    C$ERDF         ;ERROR-PC REG HIGH S/B CLEAR-SEE HEADER
      (6) 026652 000017      .WORD  15
      (6) 026654 005041      .WORD  EMO
      (6) 026656 007664      .WORD  ERR15
8251                               ;DISCUSSION.
8252
8253 026660                      20$:
8254 026660                      L10114:
      (3) 026660 104401      TRAP    C$ETST
8255
8256 026662                      BADHEAD
      (2)
8257                               ;***** TEST 43 *****
8258                               ;*TEST THAT MASTER CLEAR, CLEARS BITS IN THE NPR CONTROL REGISTER AND
8259                               ;*MICROPROCESSOR MISCELLANEOUS REGISTER-FIRST WE'LL SET THE
8260                               ;*PRIORITY UP SO THAT WHEN WE SET THE BUS REQUEST BIT THAT IT WON'T BUG US
8261                               ;*THEN WE'LL SET ALL THE BITS IN BOTH REGS EXCEPT THE
8262                               ;*NPR REQUEST. WE'LL LOOK TO SEE THAT ALL GOT SET, NEXT
8263                               ;*WE'LL DO A MASTER CLEAR AND BE SURE THAT THEY ALL
8264 026662                      ;*CLEAR.
      (2)
8265                               BADHEAD
8266 026662                      ;***** TEST 43 *****
                               BGNTST
  
```



```

(3) 026662          T43::
8267 026662          MYINT
(1) 026662 013701 002516  MOV      KMCSR,R1          ;RECORD DEVICE ADDR.
8268 026666          MSTCLR
8269 026672          SETPRI  #PRI07          ;DON'T ALLOW INTERRUPTS.
(3) 026672 012700 000340  MOV      #PRI07,R0
(3) 026676 104441          TRAP     C$SPRI
8270 026700 012761 177777 000004  MOV      #-1,4(R1)        ;DATA TO BE SET
8271 026706 042761 000002 000004  BIC      #2,4(R1)        ;DON'T SET AC LOW!
8272 026714          ROMCLK
(1) 026714 004537 003030  JSR      R5,ROMCLK        ;CLOCK INSTRUCTION
8273 026720 121111          JSR      121111            ;PUT INTO MISC REG.
8274 026722 042761 000400 000004  BIC      #400,4(R1)      ;DON'T SET NPR BIT
8275 026730          ROMCLK
(1) 026730 004537 003030  JSR      R5,ROMCLK        ;CLOCK INSTRUCTION
8276 026734 121130          JSR      121130            ;PUT INTO NPR REG
8277 026736          ROMCLK
(1) 026736 004537 003030  JSR      R5,ROMCLK        ;CLOCK INSTRUCTION
8278 026742 121225          JSR      121225            ;MOV MISC REG (11) TO PORT5
8279
8280 026744          ROMCLK
(1) 026744 004537 003030  JSR      R5,ROMCLK        ;CLOCK INSTRUCTION
8281 026750 121204          JSR      121204            ;MOVE NPR REG (10) TO PORT4
8282 026752 012737 146636 002452  MOV      #146636,$GDDAT  ;EXPECT ALL TO SET
8283 026760 016104 000004          MOV      4(R1),R4        ;READ WHAT HAPPEN
8284 026764 042704 030000          BIC      #030000,R4
8285 026770 023704 002452          CMP      $GDDAT,R4
8286 026774 001410          BEQ     10$              ;DID ALL BITS GET SET?
8287 026776          BRESET  10$              ;YES CONTINUE.
(3) 026776 104433          TRAP     C$RESET
8288 027000          ERROR   46              ;SO SORT OF PROBLEM SETTING BITS
(5) 027004 104455          TRAP     C$ERDF
(6) 027006 000056          .WORD   46
(6) 027010 005041          .WORD   EMO
(6) 027012 011072          .WORD   ERR46
8289
8290 027014          CKLOOP
(3) 027014 104406          TRAP     C$CLP1
8291
8292 027016 152761 000100 000001 10$:  BISB    #100,1(R1)        ;SET MASTER CLEAR
8293 027024 142761 000300 000001  BICB    #300,1(R1)        ;CLEAR MASTER CLEAR
8294
8295 027032          ROMCLK
(1) 027032 004537 003030  JSR      R5,ROMCLK        ;CLOCK INSTRUCTION
8296 027036 121225          JSR      121225            ;MOV MISC REG (11) TO PORT5
8297
8298 027040          ROMCLK
(1) 027040 004537 003030  JSR      R5,ROMCLK        ;CLOCK INSTRUCTION
8299 027044 121204          JSR      121204            ;MOV NPR REG (10) TO PORT4
8300 027046 016104 000004          MOV      4(R1),R4        ;READ RESULTS
8301 027052 005037 002452          CLR     $GDDAT          ;EXPECT ZERO
8302 027056 042704 010000          BIC      #010000,R4
8303 027062 001407          BEQ     20$              ;STRIP PROG CLK BIT
8304
8305 027064          ERROR   46              ;IF ALL ZERO, EVERYTHING COOL.
(5) 027070 104455          TRAP     C$ERDF
  
```

(6)	027072	000056		.WORD	46	
(6)	027074	005041		.WORD	EMO	
(6)	027076	011072		.WORD	ERR46	
8306						;SOME BITS IN THE NPR AND/OR MISC REGS.
8307	027100			CKLOOP		
(3)	027100	104406		TRAP	C\$CLP1	
8308						
8309	027102					
8310	027102	012761	000014 000004	20\$:	MOV	#14,4(R1)
8311	027110			ROMCLK		;NOW WE ARE GOING TO TRY TO
(1)	027110	004537	003030	JSR	R5,ROMCLK	;SET THE EXT BITS (16&17) IN THE NPR REG.
8312	027114	121110				;CLOCK INSTRUCTION
8313	027116			ROMCLK		;IF MASTER CLEAR FAILED TO CLEAR ITSELF
(1)	027116	004537	003030	JSR	R5,ROMCLK	;THEN WE WILL BE UNABLE TO SET
8314	027122	121205				;CLOCK INSTRUCTION
8315	027124	116104	000005	MOV	5(R1),R4	;THESE BITS
8316	027130	012737	000014 002452	MOV	#14,\$GDDAT	;READ REG
8317	027136	023704	002452	CMP	\$GDDAT,R4	;STORE GOOD
8318	027142	001407		BEQ	30\$	;DID BITS SET?
8319						;YES-CONTINUE
8320	027144			ERROR	46	;MASTER CLEAR FAILED TO CLEAR
(5)	027150	104455		TRAP	C\$ERDF	
(6)	027152	000056		.WORD	46	
(6)	027154	005041		.WORD	EMO	
(6)	027156	011072		.WORD	ERR46	
8321						;ITSELF, THUS PROHIBITING US FROM
8322						;FURTHER SETTING BITS IN THE NPR REG.
8323	027160			CKLOOP		
(3)	027160	104406		TRAP	C\$CLP1	
8324						
8325	027162			30\$:	BRESET	
(3)	027162	104433		TRAP	C\$RESET	;NOW WE'LL SEE IF A BUS RESET CLEARS
8326						
8327	027164	005737	002470	TST	RUNB	;THESE BITS.
8328	027170	001016		BNE	40\$	;CAN'T DO THIS
8329	027172			ROMCLK		;TEST IF RUN SW SET.
(1)	027172	004537	003030	JSR	R5,ROMCLK	;CLOCK INSTRUCTION
8330	027176	121204				;READ MISC REG
8331	027200	116104	000004	MOV	4(R1),R4	
8332	027204	001410		BEQ	40\$	;IF ZERO-END TST
8333						
8334	027206	005037	002452	CLR	\$GDDAT	;S/B ZERO
8335						
8336	027212			ERROR	46	;BUS RESET FAILED TO CLEAR NPR REG
(5)	027216	104455		TRAP	C\$ERDF	
(6)	027220	000056		.WORD	46	
(6)	027222	005041		.WORD	EMO	
(6)	027224	011072		.WORD	ERR46	
8337						;MASTER CLEAR WAS ABLE TO LOOK TO THE
8338						;CIRCUITRY THAT CONVERTS BUS INIT
8339						;TO 'CLEAR'
8340						
8341	027226			40\$:	ENDTST	
8342	027226			L10115:		
(3)	027226					
(3)	027226	104401		TRAP	C\$ETST	

8345  
8346  
8347  
8348  
8349  
8350  
8351  
8352  
8353  
8354  
8355  
8356  
8357  
8358 027230  
(3) 027230 000016  
(3) 027232  
8359  
8360 027232  
(4) 027232 000032  
(4) 027234 027266  
(4) 027236 000007  
(4) 027240 000000  
(4) 027242 000007  
8361 027244  
(4) 027244 001031  
(4) 027246 027340  
(4) 027250 160000  
(4) 027252 177776  
8362  
8363  
8364  
8365  
8366  
8367  
8368 027254  
(4) 027254 012032  
(4) 027256 027736  
(4) 027260 000007  
(4) 027262 000000  
(4) 027264 000001  
8369 027266  
(2)  
(3) 027266  
8370  
8371 027266 044127 041511 020110  
027274 044515 051103 026517  
027302 050103 037525 024040  
027310 036460 034115 030062  
027316 026060 036464 034115  
027324 030062 026064 036467  
027332 034115 030062 000067  
8372 027340 044515 051103 026517  
027346 050103 020125 041440  
027354 051123 040440 042104  
027362 042522 051523 035040  
027370 000040

.SBTTL HARDWARE PARAMETER CODING SECTION

```

:////////////////////
:/ THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
:/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
:/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
:/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
:/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
:/ WITH THE OPERATOR.
:////////////////////

```

```

BGNHRD
.WORD L10116-L$HARD/2
L$HARD::

GPRMD WPM,0,0,7,0,7,YES
.WORD T$CODE
.WORD WPM
.WORD 7
.WORD T$LLOLIM
.WORD T$HILIM
GPRMA ADDRES,2,0,160000,177776,YES
.WORD T$CODE
.WORD ADDRES
.WORD T$LLOLIM
.WORD T$HILIM
: GPRMA VECTOR,4,0,0,674,YES
: GPRMD PRIRTY,6,0,7000,4,7,YES
: GPRMD LNUNIT,10,0,3,0,3,YES
: GPRMD SWPAC1,12,0,377,0,377,YES
: GPRMD SWPAC2,14,0,377,0,377,YES
: GPRMD LOOPBK,16,0,40000,0,1,YES
GPRMD ISRUN,24,0,7,0,1,YES
.WORD T$CODE
.WORD ISRUN
.WORD 7
.WORD T$LLOLIM
.WORD T$HILIM
ENDHRD
.EVEN

```

L10116:

WPM: .ASCIZ 'WHICH MICRO-CPU? (0=M8200,4=M8204,7=M8207)'

ADDRES: .ASCIZ /MICRO-CPU CSR ADDRESS : /

8373 027372 044515 051103 026517 VECTOR: .ASCIZ /MICRO-CPU VECTOR ADDRESS : /  
027400 050103 020125 042526  
027406 052103 051117 040440  
027414 042104 042522 051523  
027422 035040 000040  
8374 027426 044515 051103 026517 PRIRTY: .ASCIZ /MICRO-CPU PRIORITY LEVEL : /  
027434 050103 020125 051120  
027442 047511 044522 054524  
027450 046040 053105 046105  
027456 035040 000040  
8375 027462 044127 041511 020110 LNUNIT: .ASCIZ /WHICH LINE UNIT (0-3)? 0=NONE,1=M8201,2=M8202,3=M8203 : /  
027470 044514 042516 052440  
027476 044516 020124 030050  
027504 031455 037451 030040  
027512 047075 047117 026105  
027520 036461 034115 030062  
027526 026061 036462 034115  
027534 030062 026062 036463  
027542 034115 030062 020063  
027550 020072 000  
8376 027553 123 044527 041524 SWPAC1: .ASCIZ /SWITCH PACK #1 (DDCMP LINE #) : /  
027560 020110 040520 045503  
027566 021440 020061 042050  
027574 041504 050115 046040  
027602 047111 020105 024443  
027610 035040 000040  
8377 027614 053523 052111 044103 SWPAC2: .ASCIZ /SWITCH PACK #2 (BM873 BOOT ADR) : /  
027622 050040 041501 020113  
027630 031043 024040 046502  
027636 033470 020063 047502  
027644 052117 040440 051104  
027652 020051 020072 000  
8378 027657 127 046111 020114 LOOPBK: .ASCIZ /WILL TEST CONNECTOR(S) BE USED ? 0=NO,1=YES : /  
027664 042524 052123 041440  
027672 047117 042516 052103  
027700 051117 051450 020051  
027706 042502 052440 042523  
027714 020104 020077 036460  
027722 047516 030454 054475  
027730 051505 035040 000040  
8379 027736 044515 051103 026517 ISRUN: .ASCIZ 'MICRO-PROCESSOR RUN SWITCH - TYPE 1 IF ON, 0 IF OFF :'  
027744 051120 041517 051505  
027752 047523 020122 052522  
027760 020116 053523 052111  
027766 044103 026440 052040  
027774 050131 020105 020061  
030002 043111 047440 026116  
030010 030040 044440 020106  
030016 043117 020106 000072

8380  
8381  
8382  
8383  
8384  
8385  
8386

.EVEN

CZDMQAO M8207 STATIC DIAG #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 PAGE 58-2  
HARDWARE PARAMETER CODING SECTION

K 14

SEQ 0179

8387

8389  
8390  
8391  
8392  
8393  
8394  
8395  
8396  
8397  
8398  
8399  
8400  
8401 030024  
(3) 030024 000000  
(3) 030026  
8402  
8403  
8404 030026  
(2)  
(3) 030026  
8405  
8406  
8407  
8408  
8409  
8410  
8411  
8412  
8413 030026  
(2)  
(4) 030026 000000  
(4) 030030 000000  
(3) 030032  
8414  
8415 000001

```
.SBTTL SOFTWARE PARAMETER CODING SECTION

://////
:/ THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
:/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
:/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
:/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
:/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
:/ WITH THE OPERATOR.
://////

                BGNSFT
                .WORD L10117-L$SOFT/2
L$SOFT::

                ENDSFT
                .EVEN
L10117:

                .EVEN

                LASTAD
                .EVEN
                .WORD 0
                .WORD 0
L$LAST::

                .END
```

















CZDMQAO M8207 STATIC DIAG #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 PAGE 59-7  
CROSS REFERENCE TABLE -- USER SYMBOLS

L\$DVTY	002730	G	4675	4924#
L\$EF	002052	G	4675#	
L\$ETP	002102	G	4675#	
L\$EXP1	002042	G	4675#	
L\$EXP2	002044	G	4675#	
L\$EXP3	002046	G	4675#	
L\$EXP4	002064	G	4675#	
L\$EXP5	002066	G	4675#	
L\$HARD	027232	G	4675	8358#
L\$HIME	002120	G	4675#	
L\$HPCP	002016	G	4675#	
L\$HPTP	002022	G	4675#	
L\$HW	002262	G	4675	4720#
L\$ICP	002104	G	4675#	
L\$INIT	011216	G	4675	5484#
L\$LADP	002026	G	4675#	
L\$LAST	030032	G	4675	8413#
L\$LOAD	002100	G	4675#	
L\$LUN	002074	G	4675#	
L\$MREV	002050	G	4675#	
L\$NAME	002000	G	4675#	
L\$PROT	002122	G	4675	4683#
L\$PRT	002112	G	4675#	
L\$REPP	002062	G	4675#	
L\$REV	002010	G	4675#	
L\$RPT	011210	G	5449#	
L\$SOFT	030026	G	8401#	
L\$SPC	002056	G	4675#	
L\$SPCP	002020	G	4675#	
L\$SPTP	002024	G	4675#	
L\$STA	002030	G	4675#	
L\$SW	002312	G	4744#	
L\$TEST	002114	G	4675#	
L\$TIML	002014	G	4675#	
L\$UNIT	002012	G	4675#	5520
L10001	002310		4720	4735#
L10002	002312		4744	4747#
L10003	006034		5374#	
L10004	006162		5375#	
L10005	006310		5376#	
L10006	006432		5377#	
L10007	006554		5379#	
L10010	006702		5381#	
L10011	007030		5382#	
L10012	007152		5383#	
L10013	007274		5384#	
L10014	007416		5385#	
L10015	007540		5386#	
L10016	007662		5387#	
L10017	010010		5389#	
L10020	010132		5390#	
L10021	010224		5394#	
L10022	010346		5395#	
L10023	010470		5396#	
L10024	010512		5400#	
L10025	010556		5405#	







SETBR7	003234	5136#	6601	6614	6627														
SETC	003244	5143#	6301	6315	6329	6658	7456												
SETZ	003276	5156#	6361	6374	6387	6717	7410												
SFPTBL	002312	4744#																	
SSTACK	002730	4904#	5487																
STAT	002356	4813#																	
STAT1	002500	4881#	5552*	5554*	5558*	5569*	5573*	5576*											
STAT2	002502	4882#	5579*	5581*															
STAT3	002504	4883#																	
STM	005505	5305#	5374	5375	5376	5377	5379	5381	5382	5383	5384	5385	5386	5387					
		5389	5390	5394	5395	5396	5399	5404	5408	5412	5417	5421	5426	5430					
		5435																	
STRTSW	002354	4812#																	
SUBRPC	002346	4809#																	
SVCGBL	= 000000	4635#	4641	4649#	4675	4683	4697	4720	4744	4796	4924	5374	5375	5376	5377	5379			
		5377	5379	5381	5382	5383	5384	5385	5386	5387	5389	5390	5394	5395	5396	5399			
		5396	5398	5402	5406	5410	5415	5419	5423	5428	5433	5449	5484	5606					
		5634	5651	5669	8358	8401	8413#												
SVCINS	= 000000	4635#	4646#	4675	4697	4720	4744	4796	4924	5374	5375	5376	5377	5379					
		5381	5382	5383	5384	5385	5386	5387	5389	5390	5394	5395	5396	5399					
		5400	5403	5404	5405	5407	5408	5409	5411	5412	5413	5416	5417	5418					
		5420	5421	5422	5424	5425	5426	5427	5429	5430	5431	5434	5435	5436					
		5457	5467	5499	5500	5502	5503	5505	5506	5508	5509	5522	5523	5603					
		5620	5624	5635	5637	5653	5654	5670	5697	5704	5708	5710	5738	5740					
		5750	5753	5761	5767	5775	5776	5777	5785	5793	5799	5808	5809	5810					
		5819	5828	5833	5845	5846	5847	5848	5859	5860	5861	5867	5903	5904					
		5909	5914	5947	5948	5953	5958	5992	5992	6014	6015	6016	6021	6030					
		6057	6075	6079	6080	6106	6108	6118	6136	6137	6015	6016	6021	6030					
		6185	6197	6198	6199	6200	6210	6211	6212	6213	6223	6224	6225	6226					
		6240	6245	6254	6255	6256	6257	6266	6267	6268	6269	6278	6279	6280					
		6281	6295	6300	6310	6312	6313	6314	6324	6326	6327	6328	6338	6339					
		6340	6341	6355	6360	6370	6371	6372	6373	6383	6384	6385	6386	6396					
		6397	6398	6399	6413	6419	6429	6430	6431	6432	6442	6443	6444	6445					
		6455	6456	6457	6458	6472	6478	6488	6489	6490	6491	6501	6502	6503					
		6504	6514	6515	6516	6517	6531	6536	6551	6553	6554	6555	6565	6566					
		6567	6568	6578	6579	6580	6581	6595	6600	6610	6611	6612	6613	6623					
		6624	6625	6626	6636	6637	6638	6639	6653	6657	6668	6669	6670	6671					
		6682	6683	6684	6685	6695	6696	6697	6698	6712	6716	6727	6728	6729					
		6730	6741	6742	6743	6744	6754	6755	6756	6757	6771	6775	6785	6786					
		6787	6788	6799	6800	6801	6802	6812	6813	6814	6815	6829	6833	6843					
		6844	6845	6846	6857	6858	6859	6860	6870	6871	6872	6873	6889	6893					
		6904	6905	6906	6907	6920	6922	6923	6924	6934	6935	6936	6937	6951					
		6955	6965	6966	6967	6968	6979	6980	6981	6982	6992	6993	6994	6995					
		7007	7043	7048	7053	7067	7092	7096	7103	7140	7189	7220	7261	7264					
		7299	7348	7379	7386	7396	7407	7409	7419	7421	7429	7431	7442	7453					
		7455	7464	7466	7474	7476	7498	7511	7523	7526	7538	7553	7571	7580					
		7584	7591	7600	7607	7713	7714	7737	7753	7754	7770	7771	7775	7784					
		7821	7824	7835	7863	7866	7870	7876	7889	7936	7938	7945	7948	7954					
		7956	7962	7964	7972	7974	7978	7979	7984	7985	7990	7992	8011	8041					
		8058	8061	8079	8080	8087	8091	8103	8125	8128	8134	8149	8166	8170					
		8201	8204	8219	8250	8254	8269	8287	8288	8290	8305	8307	8320	8323					
		8325	8336	8342	8358	8360	8361	8368	8369	8401	8404	8413							
SVC SUB	= 000000	4635#	4648#																
SVC TAG	= 000000	4635#	4650#	4735	4747	5374	5375	5376	5377	5379	5381	5382	5383	5384					
		5385	5386	5387	5389	5390	5394	5395	5396	5400	5405	5409	5413	5418					
		5422	5427	5431	5436	5467	5603	5624	5637	5654	5670	5710	5753	5777					

CZDMQAO M8207 STATIC DIAG #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 PAGE 59-11  
K 15  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0192

	5785	5810	5819	5847	5861	5867	5914	5958	6021	6108	6165	6199	6212
	6225	6226	6256	6268	6280	6281	6313	6327	6340	6341	6372	6385	6398
	6399	6431	6444	6457	6458	6490	6503	6516	6517	6554	6567	6580	6581
	6612	6625	6638	6639	6670	6684	6697	6698	6729	6743	6756	6757	6787
	6801	6814	6815	6845	6859	6872	6873	6906	6923	6936	6937	6967	6981
	6994	6995	7053	7103	7264	7386	7431	7476	7526	7607	7784	7824	7876
	8011	8061	8091	8134	8170	8204	8254	8342	8369	8404			
SVCTST= 000000	4635#	4647#	5690	5720	5760	5792	5827	5876	5921	5966	6029	6117	6178
	6239	6294	6354	6412	6471	6530	6594	6652	6711	6770	6828	6888	6950
	7006	7066	7138	7297	7395	7441	7484	7537	7618	7794	7833	7886	8022
SV05 003616	8071	8101	8147	8183	8217	8266							
	5246#	5704	5738	5750	5775	5808	5845	5859	5903	5947	5991	6014	6057
	6075	6106	6136	6151	6163	6197	6210	6223	6254	6266	6278	6310	6324
	6338	6370	6383	6396	6429	6442	6455	6488	6501	6514	6551	6565	6578
	6610	6623	6636	6668	6682	6695	6727	6741	6754	6785	6799	6812	6843
	6857	6870	6904	6920	6934	6965	6979	6992	7043	7092	7189	7261	7348
	7407	7419	7429	7453	7464	7474	7498	7511	7523	7553	7571	7580	7591
	7600	7713	7737	7753	7770	7821	7863	7936	7945	7954	7962	7972	7978
	7984	7990	8058	8079	8087	8125	8166	8201	8250	8288	8305	8320	8336
SWPAC1 027553	8376#												
SWPAC2 027614	8377#												
S\$LSYM= 010000	4635#	4735#	4747#	5374#	5375#	5376#	5377#	537#	5381#	5382#	5383#	5384#	5385#
	5386#	5387#	5389#	5390#	5394#	5395#	5396#	540J#	5405#	5409#	5413#	5418#	5422#
	5427#	5431#	5436#	5467#	5603#	5624#	5637#	5654#	5670#	5710#	5753#	5767#	5785#
	5799#	5819#	5833#	5848#	5867#	5914#	5958#	6021#	6108#	6165#	6185#	6200#	6213#
	6226#	6245#	6257#	6269#	6281#	6300#	6314#	6328#	6341#	6360#	6373#	6386#	6399#
	6419#	6432#	6445#	6458#	6478#	6491#	6504#	6517#	6536#	6555#	6568#	6581#	6600#
	6613#	6626#	6639#	6657#	6671#	6685#	6698#	6716#	6730#	6744#	6757#	6775#	6788#
	6802#	6815#	6833#	6846#	6860#	6873#	6893#	6907#	6924#	6937#	6955#	6968#	6982#
	6995#	7053#	7103#	7264#	7386#	7431#	7476#	7526#	7607#	7784#	7824#	7876#	8011#
	8061#	8091#	8134#	8170#	8204#	8254#	8342#	8369#	8404#				
TEMP 002440	4838#	7487*	7517*	7542*	7550*								
TFM1 003652	5267#	5374	5375	5376									
TFM2 003676	5268#	5377	5379	5383	5386	5395	5396						
TFM3 003714	5269#	5381	5382										
TFM36 004003	5272#												
TFM4 003741	5270#	5384	5385	5387	5390								
TFM40 004247	5275#	5403											
TFM41 004071	5273#	5407											
TFM42 004160	5274#	5411											
TFM43 004330	5276#	5416											
TFM44 004413	5277#	5420											
TFM45 004452	5278#	5424											
TFM45A 004510	5279#	5425											
TFM46 004633	5281#	5429											
TFM47 004717	5282#	5434											
TFM5 003757	5271#	5389											
TMMC 004774	5283#	5420	5424										
TYPE 002432	4835#	5589*	5594*	5599*	5761	5793	5828	6118	7693				
T\$ARGC= 000001	4675#	5374#	5375#	5376#	5377#	5379#	5381#	5382#	5383#	5384#	5385#	5386#	5387#
	5389#	5390#	5394#	5395#	5396#	5399#	5403#	5404#	5407#	5408#	5411#	5412#	5416#
	5417#	5420#	5421#	5424#	5425#	5426#	5429#	5430#	5434#	5435#			
T\$CODE= 012032	8360#	8361#	8368#										
T\$ERRN= 000056	4635#	5704#	5738#	5750#	5775#	5808#	5845#	5859#	5903#	5947#	5991#	6014#	6057#
	6075#	6106#	6136#	6151#	6163#	6197#	6210#	6223#	6254#	6266#	6278#	6310#	6324#
	6338#	6370#	6383#	6396#	6429#	6442#	6455#	6488#	6501#	6514#	6551#	6565#	6578#

T\$EXCP= 000000  
T\$FLAG= 000040

T\$GMAN= 000000  
T\$HILI= 000001  
T\$LAST= 000001  
T\$LOLI= 000000  
T\$LSYM= 010000

T\$LTNO= 000053  
T\$NEST= 000000

T\$NSO = 000000  
T\$NS1 = 000005

6610#	6623#	6636#	6668#	6682#	6695#	6727#	6741#	6754#	6785#	6799#	6812#	6843#
6857#	6870#	6904#	6920#	6934#	6965#	6979#	6992#	7043#	7092#	7189#	7261#	7348#
7407#	7419#	7429#	7453#	7464#	7474#	7498#	7511#	7523#	7553#	7571#	7580#	7591#
7600#	7713#	7737#	7753#	7770#	7821#	7863#	7936#	7945#	7954#	7962#	7972#	7978#
7984#	7990#	8058#	8079#	8087#	8125#	8166#	8201#	8250#	8288#	8305#	8320#	8336#
8360#	8361#	8368#										
5457#	5697#	5708#	5740#	5761#	5776#	5793#	5809#	5828#	5846#	5860#	5904#	5948#
5992#	6015#	6030#	6079#	6118#	6137#	6180#	6198#	6211#	6224#	6240#	6255#	6267#
6279#	6295#	6312#	6326#	6339#	6355#	6371#	6384#	6397#	6413#	6430#	6443#	6456#
6472#	6489#	6502#	6515#	6531#	6553#	6566#	6579#	6595#	6611#	6624#	6637#	6653#
6669#	6683#	6696#	6712#	6728#	6742#	6755#	6771#	6786#	6800#	6813#	6829#	6844#
6858#	6871#	6889#	6905#	6922#	6935#	6951#	6966#	6980#	6993#	7007#	7048#	7067#
7096#	7140#	7220#	7299#	7379#	7396#	7409#	7421#	7442#	7455#	7466#	7584#	7714#
7754#	7771#	7775#	7835#	7866#	7870#	7889#	7938#	7948#	7956#	7964#	7974#	7979#
7985#	7992#	8041#	8080#	8103#	8128#	8149#	8219#					
4635#												
8360#	8361#	8368#										
4635#	8413#											
8360#	8361#	8368#										
4635#	4735	4747	5374	5375	5376	5377	5379	5381	5382	5383	5384	5385
5386	5387	5389	5390	5394	5395	5396	5400	5405	5409	5413	5418	5422
5427	5431	5436	5467	5603	5624	5637	5654	5670	5710	5753	5785	5819
5867	5914	5958	6021	6108	6165	6226	6281	6341	6399	6458	6517	6581
6639	6698	6757	6815	6873	6937	6995	7053	7103	7264	7386	7431	7476
7526	7607	7784	7824	7876	8011	8061	8091	8134	8170	8204	8254	8342
8369	8404											
8413#												
4635#	4641#	4683#	4687#	4720#	4735#	4744#	4747#	5374#	5375#	5376#	5377#	5379#
5381#	5382#	5383#	5384#	5385#	5386#	5387#	5389#	5390#	5394#	5395#	5396#	5398#
5400#	5402#	5405#	5406#	5409#	5410#	5413#	5415#	5418#	5419#	5422#	5423#	5427#
5428#	5431#	5433#	5436#	5449#	5467#	5484#	5603#	5606#	5624#	5634#	5637#	5651#
5654#	5669#	5670#	5690#	5710#	5720#	5753#	5760#	5767#	5777#	5785#	5792#	5799#
5810#	5819#	5827#	5833#	5847#	5848#	5861#	5867#	5876#	5914#	5921#	5958#	5966#
6021#	6029#	6108#	6117#	6165#	6178#	6185#	6199#	6200#	6212#	6213#	6225#	6226#
6239#	6245#	6256#	6257#	6268#	6269#	6280#	6281#	6294#	6300#	6313#	6314#	6327#
6328#	6340#	6341#	6354#	6360#	6372#	6373#	6385#	6386#	6398#	6399#	6412#	6419#
6431#	6432#	6444#	6445#	6457#	6458#	6471#	6478#	6490#	6491#	6503#	6504#	6516#
6517#	6530#	6536#	6554#	6555#	6567#	6568#	6580#	6581#	6594#	6600#	6612#	6613#
6625#	6626#	6638#	6639#	6652#	6657#	6670#	6671#	6684#	6685#	6697#	6698#	6711#
6716#	6729#	6730#	6743#	6744#	6756#	6757#	6770#	6775#	6787#	6788#	6801#	6802#
6814#	6815#	6828#	6833#	6845#	6846#	6859#	6860#	6872#	6873#	6888#	6893#	6906#
6907#	6923#	6924#	6936#	6937#	6950#	6955#	6967#	6968#	6981#	6982#	6994#	6995#
7006#	7053#	7066#	7103#	7138#	7264#	7297#	7386#	7395#	7431#	7441#	7476#	7484#
7526#	7537#	7607#	7618#	7784#	7794#	7824#	7833#	7876#	7886#	8011#	8022#	8061#
8071#	8091#	8101#	8134#	8147#	8170#	8183#	8204#	8217#	8254#	8266#	8342#	8358#
8369#	8401#	8404#										
4641#												
4683#	4687	4720#	4735	4744#	4747	5374#	5375#	5376#	5377#	5379#	5381#	5382#
5383#	5384#	5385#	5386#	5387#	5389#	5390#	5394#	5395#	5396#	5398#	5400	5402#
5405	5406#	5409	5410#	5413	5415#	5418	5419#	5422	5423#	5427	5428#	5431
5433#	5436	5449#	5467	5484#	5603	5606#	5624	5634#	5637	5651#	5654	5669#
5670	5690#	5710	5720#	5753	5760#	5785	5792#	5819	5827#	5867	5876#	5914
5921#	5958	5966#	6021	6029#	6108	6117#	6165	6178#	6226	6239#	6281	6294#
6341	6354#	6399	6412#	6458	6471#	6517	6530#	6581	6594#	6639	6652#	6698
6711#	6757	6770#	6815	6828#	6873	6888#	6937	6950#	6995	7006#	7053	7066#
7103	7138#	7264	7297#	7386	7395#	7431	7441#	7476	7484#	7526	7537#	7607

CZDMQA0 M8207 STATIC DIAG #2 MACY11 30A(1052) 17-JUL-79 09:23 PAGE 59-13  
CZDMQA.P11 25-JUN-79 14:07 CROSS REFERENCE TABLE -- USER SYMBOLS

M 15

SEQ 0194

T\$NS2 = 000003	7618#	7784	7794#	7824	7833#	7876	7886#	8011	8022#	8061	8071#	8091	8101#
	8134	8147#	8170	8183#	8204	8217#	8254	8266#	8342	8358#	8369	8401#	8404
	5767#	5777	5799#	5810	5833#	5847	5848#	5861	6185#	6199	6200#	6212	6213#
	6225	6245#	6256	6257#	6268	6269#	6280	6300#	6313	6314#	6327	6328#	6340
	6360#	6372	6373#	6385	6386#	6398	6419#	6431	6432#	6444	6445#	6457	6478#
	6490	6491#	6503	6504#	6516	6536#	6554	6555#	6567	6568#	6580	6600#	6612
	6613#	6625	6626#	6638	6657#	6670	6671#	6684	6685#	6697	6716#	6729	6730#
	6743	6744#	6756	6775#	6787	6788#	6801	6802#	6814	6833#	6845	6846#	6859
	6860#	6872	6893#	6906	6907#	6923	6924#	6936	6955#	6967	6968#	6981	6982#
	6994												
T\$PTNU= 000000	4635#												
T\$SAVL= 177777	4635#												
T\$SEGL= 177777	4635#	5767#	5776	5777#	5799#	5809	5810#	5833#	5846	5847#	5848#	5860	5861#
	6185#	6198	6199#	6200#	6211	6212#	6213#	6224	6225#	6245#	6255	6256#	6257#
	6267	6268#	6269#	6279	6280#	6300#	6312	6313#	6314#	6326	6327#	6328#	6339
	6340#	6360#	6371	6372#	6373#	6384	6385#	6386#	6397	6398#	6419#	6430	6431#
	6432#	6443	6444#	6445#	6456	6457#	6478#	6489	6490#	6491#	6502	6503#	6504#
	6515	6516#	6536#	6553	6554#	6555#	6566	6567#	6568#	6579	6580#	6600#	6611
	6612#	6613#	6624	6625#	6626#	6637	6638#	6657#	6669	6670#	6671#	6683	6684#
	6685#	6696	6697#	6716#	6728	6729#	6730#	6742	6743#	6744#	6755	6756#	6775#
	6786	6787#	6788#	6800	6801#	6802#	6813	6814#	6833#	6844	6845#	6846#	6858
	6859#	6860#	6871	6872#	6893#	6905	6906#	6907#	6922	6923#	6924#	6935	6936#
	6955#	6966	6967#	6968#	6980	6981#	6982#	6993	6994#				
T\$SEKO= 010002	5767#	5776	5777#	5799#	5809	5810#	5833#	5846	5847#	5848#	5860	5861	6185#
	6198	6199#	6200#	6211	6212#	6213#	6224	6225#	6245#	6255	6256#	6257#	6267
	6268	6269#	6279	6280#	6300#	6312	6313#	6314#	6326	6327#	6328#	6339	6340
	6360#	6371	6372#	6373#	6384	6385#	6386#	6397	6398#	6419#	6430	6431#	6432#
	6443	6444	6445#	6456	6457#	6478#	6489	6490#	6491#	6502	6503#	6504#	6515
	6516	6536#	6553	6554#	6555#	6566	6567#	6568#	6579	6580#	6600#	6611	6612
	6613#	6624	6625#	6626#	6637	6638#	6657#	6669	6670#	6671#	6683	6684#	6685#
	6696	6697#	6716#	6728	6729#	6730#	6742	6743#	6744#	6755	6756#	6775#	6786
	6787	6788#	6800	6801#	6802#	6813	6814#	6833#	6844	6845#	6846#	6858	6859
	6860#	6871	6872#	6893#	6905	6906#	6907#	6922	6923#	6924#	6935	6936	6955#
	6966	6967	6968#	6980	6981#	6982#	6993	6994#					
T\$SUBN= 000000	4635#	5690#	5720#	5760#	5792#	5827#	5876#	5921#	5966#	6029#	6117#	6178#	6239#
	6294#	6354#	6412#	6471#	6530#	6594#	6652#	6711#	6770#	6828#	6888#	6950#	7006#
	7066#	7138#	7297#	7395#	7441#	7484#	7537#	7618#	7794#	7833#	7886#	8022#	8071#
	8101#	8147#	8183#	8217#	8266#								
T\$TAGL= 177777	4635#												
T\$TAGN= 010120	4635#	4683#	4720#	4744#	5374#	5375#	5376#	5377#	5379#	5381#	5382#	5383#	5384#
	5385#	5386#	5387#	5389#	5390#	5394#	5395#	5396#	5398#	5402#	5406#	5410#	5415#
	5419#	5423#	5428#	5433#	5449#	5484#	5606#	5634#	5651#	5669#	5690#	5720#	5760#
	5792#	5827#	5876#	5921#	5966#	6029#	6117#	6178#	6239#	6294#	6354#	6412#	6471#
	6530#	6594#	6652#	6711#	6770#	6828#	6888#	6950#	7006#	7066#	7138#	7297#	7395#
	7441#	7484#	7537#	7618#	7794#	7833#	7886#	8022#	8071#	8101#	8147#	8183#	8217#
	8266#	8358#	8401#										
T\$TEMP= 000005	4687#	4697#	4735#	4747#	5374#	5375#	5376#	5377#	5379#	5381#	5382#	5383#	5384#
	5385#	5386#	5387#	5389#	5390#	5394#	5395#	5396#	5400#	5405#	5409#	5413#	5418#
	5422#	5427#	5431#	5436#	5457#	5467#	5603#	5624#	5637#	5654#	5670#	5697#	5708#
	5710#	5740#	5753#	5761#	5776#	5777#	5785#	5793#	5809#	5810#	5819#	5828#	5846#
	5847#	5860#	5861#	5867#	5904#	5914#	5948#	5958#	5992#	6015#	6021#	6030#	6079#
	6108#	6118#	6137#	6165#	6180#	6198#	6199#	6211#	6212#	6224#	6225#	6226#	6240#
	6255#	6256#	6267#	6268#	6279#	6280#	6281#	6295#	6312#	6313#	6326#	6327#	6339#
	6340#	6341#	6355#	6371#	6372#	6384#	6385#	6397#	6398#	6399#	6413#	6430#	6431#
	6443#	6444#	6456#	6457#	6458#	6472#	6489#	6490#	6502#	6503#	6515#	6516#	6517#
	6531#	6553#	6554#	6566#	6567#	6579#	6580#	6581#	6595#	6611#	6612#	6624#	6625#

6637#	6638#	6639#	6653#	6669#	6670#	6683#	6684#	6696#	6697#	6698#	6712#	6728#
6729#	6742#	6743#	6755#	6756#	6757#	6771#	6786#	6787#	6800#	6801#	6813#	6814#
6815#	6829#	6844#	6845#	6858#	6859#	6871#	6872#	6873#	6889#	6905#	6906#	6922#
6923#	6935#	6936#	6937#	6951#	6966#	6967#	6980#	6981#	6993#	6994#	6995#	7007#
7048#	7053#	7067#	7096#	7103#	7140#	7220#	7264#	7299#	7379#	7386#	7396#	7409#
7421#	7431#	7442#	7455#	7466#	7476#	7526#	7584#	7607#	7714#	7754#	7771#	7775#
7784#	7824#	7835#	7866#	7870#	7876#	7889#	7938#	7948#	7956#	7964#	7974#	7979#
7985#	7992#	8011#	8041#	8061#	8080#	8091#	8103#	8128#	8134#	8149#	8170#	8204#
8219#	8254#	8342#	8360#	8361#	8368#	8369#	8404#					
4635#	5685	5688	5690#	5714	5718	5720#	5755	5758	5760#	5787	5790	5792#
5821	5825	5827#	5870	5873	5876#	5916	5919	5921#	5960	5964	5966#	6023
6027	6029#	6110	6115	6117#	6167	6176	6178#	6228	6237	6239#	6283	6292
6294#	6343	6352	6354#	6401	6410	6412#	6460	6469	6471#	6519	6528	6530#
6583	6592	6594#	6641	6650	6652#	6700	6709	6711#	6759	6768	6770#	6817
6826	6828#	6875	6885	6888#	6939	6948	6950#	6997	7004	7006#	7056	7064
7066#	7105	7136	7138#	7265	7295	7297#	7387	7393	7395#	7433	7439	7441#
7477	7482	7484#	7528	7535	7537#	7609	7616	7618#	7786	7792	7794#	7826
7831	7833#	7879	7884	7886#	8013	8020	8022#	8063	8069	8071#	8093	8099
8101#	8136	8145	8147#	8172	8181	8183#	8206	8215	8217#	8256	8264	8266#
8413												
4635#	5374	5375	5376	5377	5379	5381	5382	5383	5384	5385	5386	5387
5389	5390	5394	5395	5396	5399	5400	5403	5404	5405	5407	5408	5409
5411	5412	5413	5416	5417	5418	5420	5421	5422	5424	5425	5426	5427
5429	5430	5431	5434	5435	5436	5467	5499	5502	5505	5508	5522	5603
5620	5624	5635	5637	5653	5654	5670	5697	5704	5708	5710	5738	5740
5750	5753	5761	5767	5775	5776	5777	5785	5793	5799	5808	5809	5810
5819	5828	5833	5845	5846	5847	5848	5859	5860	5861	5867	5903	5904
5909	5914	5947	5948	5953	5958	5991	5992	6014	6015	6016	6021	6030
6057	6075	6079	6080	6106	6108	6118	6136	6137	6151	6163	6165	6180
6185	6197	6198	6199	6200	6210	6211	6212	6213	6223	6224	6225	6226
6240	6245	6254	6255	6256	6257	6266	6267	6268	6269	6278	6279	6280
6281	6295	6300	6310	6312	6313	6314	6324	6326	6327	6328	6338	6339
6340	6341	6355	6360	6370	6371	6372	6373	6383	6384	6385	6386	6396
6397	6398	6399	6413	6419	6429	6430	6431	6432	6442	6443	6444	6445
6455	6456	6457	6458	6472	6478	6488	6489	6490	6491	6501	6502	6503
6504	6514	6515	6516	6517	6531	6536	6551	6553	6554	6555	6565	6566
6567	6568	6578	6579	6580	6581	6595	6600	6610	6611	6612	6613	6623
6624	6625	6626	6636	6637	6638	6639	6653	6657	6668	6669	6670	6671
6682	6683	6684	6685	6695	6696	6697	6698	6712	6716	6727	6728	6729
6730	6741	6742	6743	6744	6754	6755	6756	6757	6771	6775	6785	6786
6787	6788	6799	6800	6801	6802	6812	6813	6814	6815	6829	6833	6843
6844	6845	6846	6857	6858	6859	6860	6870	6871	6872	6873	6889	6893
6904	6905	6906	6907	6920	6922	6923	6924	6934	6935	6936	6937	6951
6955	6965	6966	6967	6968	6979	6980	6981	6982	6992	6993	6994	6995
7007	7043	7048	7053	7067	7092	7096	7103	7140	7189	7220	7261	7264
7299	7348	7379	7386	7396	7407	7409	7419	7421	7429	7431	7442	7453
7455	7464	7466	7474	7476	7498	7511	7523	7526	7538	7553	7571	7580
7584	7591	7600	7607	7713	7714	7737	7753	7754	7770	7771	7775	7784
7821	7824	7835	7863	7866	7870	7876	7889	7936	7938	7945	7948	7954
7956	7962	7964	7972	7974	7978	7979	7984	7985	7990	7992	8011	8041
8058	8061	8079	8080	8087	8091	8103	8125	8128	8134	8149	8166	8170
8201	8204	8219	8250	8254	8269	8287	8288	8290	8305	8307	8320	8323
8325	8336	8342										
4635#	5690#	5720#	5760#	5792#	5827#	5876#	5921#	5966#	6029#	6117#	6178#	6239#
6294#	6354#	6412#	6471#	6530#	6594#	6652#	6711#	6770#	6828#	6888#	6950#	7006#
7066#	7138#	7297#	7395#	7441#	7484#	7537#	7618#	7794#	7833#	7886#	8022#	8071#

T\$TEST= 000053

T\$TSTM= 177777

T\$TSTS= 000001



T26	021440 G	4697	7066#											
T27	021606 G	4697	7138#											
T28	022164 G	4697	7297#											
T29	022422 G	4697	7395#											
T3	012300 G	4697	5760#											
T30	022636 G	4697	7441#											
T31	023052 G	4697	7484#											
T32	023344 G	4697	7537#											
T33	023736 G	4697	7618#											
T34	025010 G	4697	7794#											
T35	025110 G	4697	7833#											
T36	025256 G	4697	7886#											
T37	025734 G	4697	8022#											
T38	026074 G	4697	8071#											
T39	026202 G	4697	8101#											
T4	012430 G	4697	5792#											
T40	026340 G	4697	8147#											
T41	026436 G	4697	8183#											
T42	026536 G	4697	8217#											
T43	026662 G	4697	8266#											
T5	012570 G	4697	5827#											
T6	013032 G	4697	5876#											
T7	013234 G	4697	5921#											
T8	013446 G	4697	5966#											
T9	013770 G	4697	6029#											
UAM =	000200 G	4774#												
VECTOR	027372	8373#												
WPM	027266	8360	8371#											
WROM	003406	5188#												
WTYPE	002414	4828#	5089	5172	5173	5193	5214	5524*	5590	5592	5597	5839	6088	6179
		6240	6295	6355	6413	6472	6531	6595	6653	6672	6712	6731	6771	6789
		6829	6847	6889	6908	6951	6969	7182	7230	7341	7834	7888	8102	8148
		8218												
X\$ALWA=	000000	4635#												
X\$FALS=	000040	4635#												
X\$OFFS=	000400	4635#												
X\$TRUE=	000020	4635#												
ZERO	002370	4818#												
\$BDADR	002450	4842#												
\$BDDAT	002454	4844#												
\$GDADR	002446	4841#												
\$GDDAT	002452	4843#	5429	7494*	7495	7497	7507*	7508	7510	7704*	7708	7711*	7712	7730*
		7734	7736	7746*	7750	7752	7763*	7767	7769	8282*	8285	8301*	8316*	8317
		8334*												
\$LSTIN=	000000	4644#												
\$LSTTA=	000000	4645#												
\$REG0	002430	4834#	5251*	5374	5375									
\$REG1	002426	4833#	5250*											
\$REG2	002424	4832#	5249*	5374	5375	5376	5382	5384	5387					
\$REG3	002422	4831#	5248*											
\$REG4	002420	4830#	5247*	5374	5375	5376	5377	5379	5381	5382	5383	5384	5385	5386
		5387	5389	5390	5395	5396								
\$REG5	002416	4829#	5246*	5376	5377	5379	5381	5382	5383	5385	5386	5389	5390	5395
		5396												
\$TEMPO	002442	4839#												
\$TMPO	002444	4840#	7488*	7518*	7519	7522	7543*	7569	7572					





CZDMQAO M8207 STATIC DIAG #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 PAGE 60  
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0199

BADHEA	4986#	5685	5688	5714	5718	5755	5758	5787	5790	5821	5825	5870	5873	5916	5919
	5960	5964	6023	6027	6110	6115	6167	6176	6228	6237	6283	6292	6343	6352	6401
	6410	6460	6469	6519	6528	6583	6592	6641	6650	6700	6709	6759	6768	6817	6826
	6875	6885	6939	6948	6997	7004	7056	7064	7105	7136	7265	7295	7387	7393	7433
	7439	7477	7482	7528	7535	7609	7616	7786	7792	7826	7831	7879	7884	8013	8020
	8063	8069	8093	8099	8136	8145	8172	8181	8206	8215	8256	8264			
BCOMPL	102#	4635#	5500	5503	5506										
BERROR	110#	4635#													
BGNAU	118#	4635#	5669												
BGNAUT	132#	4635#	5606												
BGNCLN	146#	4635#	5634												
BGNDU	159#	4635#	5651												
BGNHRD	172#	4635#	8358												
BGNHW	187#	4635#	4720												
BGNINI	203#	4635#	5484												
BGNMOD	215#	4635#	4641												
BGNMSG	232#	4635#	5374	5375	5376	5377	5379	5381	5382	5383	5384	5385	5386	5387	5389
	5390	5394	5395	5396	5398	5402	5406	5410	5415	5419	5423	5428	5433		
BGNPRO	245#	4635#	4683												
BGNPTA	259#	4635#													
BGNRPT	297#	4635#	5449												
BGNSEG	311#	4635#	5767	5799	5833	5848	6185	6200	6213	6245	6257	6269	6300	6314	6328
	6360	6373	6386	6419	6432	6445	6478	6491	6504	6536	6555	6563	6600	6613	6626
	6657	6671	6685	6716	6730	6744	6775	6788	6802	6833	6846	6860	6893	6907	6924
	6955	6968	6982												
BGNSET	325#	4635#													
BGNSFT	352#	4635#	8401												
BGNSRV	367#	4635#													
BGNSUB	380#	4635#													
BGNSW	408#	4635#	4744												
BGNTST	423#	4635#	5690	5720	5760	5792	5827	5876	5921	5966	6029	6117	6178	6239	6294
	6354	6412	6471	6530	6594	6652	6711	6770	6828	6888	6950	7006	7066	7138	7297
	7395	7441	7484	7537	7618	7794	7833	7886	8022	8071	8101	8147	8183	8217	8266
BINCOMP	457#	4635#	5509	5523											
BNERRO	465#	4635#													
BREAK	473#	4635#	5909	5953	6016	6080									
BRESET	481#	4635#	5635	5653	7538	8287	8325								
CKLOOP	491#	4635#	8290	8307	8323										
CLOCK	503#	4635#													
CLOSE	514#	4635#													
CLRMAR	5027#	6039	6049												
CLRVEC	528#	4635#													
COMMEN	537#	4635#													
DELAY	568#	4635#													
DESCRI	558#	4635#	4796												
DEVTYP	588#	4635#	4924												
DISPAT	597#	4635#	4697												
DISPLA	615#	4635#													
DOCLN	635#	4635#													
DODU	643#	4635#	5620												
DORPT	651#	4635#													
ED\$CAL	4981#	5685	5688	5714	5718	5755	5758	5787	5790	5821	5825	5870	5873	5916	5919
	5960	5964	6023	6027	6110	6115	6167	6176	6228	6237	6283	6292	6343	6352	6401
	6410	6460	6469	6519	6528	6583	6592	6641	6650	6700	6709	6759	6768	6817	6826
	6875	6885	6939	6948	6997	7004	7056	7064	7105	7136	7265	7295	7387	7393	7433
	7439	7477	7482	7528	7535	7609	7616	7786	7792	7826	7831	7879	7884	8013	8020

CZDMQAO M8207 STATIC DIAG #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 PAGE 60-1  
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0200

	8063	8069	8093	8099	8136	8145	8172	8181	8206	8215	8256	8264			
ENDAU	659#	4635#	5670												
ENDAUT	675#	4635#	5624												
ENDCLN	692#	4635#	5637												
ENDCOM	708#	4635#													
ENDDU	730#	4635#	5654												
ENDHRD	749#	4635#	8369												
ENDHW	765#	4635#	4735												
ENDINI	779#	4635#	5603												
ENDMOD	796#	4635#													
ENDMSG	813#	4635#	5374	5375	5376	5377	5379	5381	5382	5383	5384	5385	5386	5387	5389
	5390	5394	5395	5396	5400	5405	5409	5413	5418	5422	5427	5431	5436		
ENDPRO	830#	4635#	4687												
ENDPTA	844#	4635#													
ENDRPT	857#	4635#	5467												
ENDSEG	876#	4635#	5777	5810	5847	5861	6199	6212	6225	6256	6268	6280	6313	6327	6340
	6372	6385	6398	6431	6444	6457	6490	6503	6516	6554	6567	6580	6612	6625	6638
	6670	6684	6697	6729	6743	6756	6787	6801	6814	6845	6859	6872	6906	6923	6936
	6967	6981	6994												
ENDSET	895#	4635#													
ENDSFT	915#	4635#	8404												
ENDSRV	932#	4635#													
ENDSUB	952#	4635#													
ENDSW	974#	4635#	4747												
ENDTST	988#	4635#	5710	5753	5785	5819	5867	5914	5958	6021	6108	6165	6226	6281	6341
	6399	6458	6517	6581	6639	6698	6757	6815	6873	6937	6995	7053	7103	7264	7386
	7431	7476	7526	7607	7784	7824	7876	8011	8061	8091	8134	8170	8204	8254	8342
EQUALS	1009#	4635#	4774												
ERRDF	1087#	4635#	5704	5738	5750	5775	5808	5845	5859	5903	5947	5991	6014	6057	6075
	6106	6136	6151	6163	6197	6210	6223	6254	6266	6278	6310	6324	6338	6370	6383
	6396	6429	6442	6455	6488	6501	6514	6551	6565	6578	6610	6623	6636	6668	6682
	6695	6727	6741	6754	6785	6799	6812	6843	6857	6870	6904	6920	6934	6965	6979
	6992	7043	7092	7189	7261	7348	7407	7419	7429	7453	7464	7474	7498	7511	7523
	7553	7571	7580	7591	7600	7713	7737	7753	7770	7821	7863	7936	7945	7954	7962
	7972	7978	7984	7990	8058	8079	8087	8125	8166	8201	8250	8288	8305	8320	8336
ERRHRD	1099#	4635#													
ERROR	1109#	4635#	5367#	5704	5738	5750	5775	5808	5845	5859	5903	5947	5991	6014	6057
	6075	6106	6136	6151	6163	6197	6210	6223	6254	6266	6278	6310	6324	6338	6370
	6383	6396	6429	6442	6455	6488	6501	6514	6551	6565	6578	6610	6623	6636	6668
	6682	6695	6727	6741	6754	6785	6799	6812	6843	6857	6870	6904	6920	6934	6965
	6979	6992	7043	7092	7189	7261	7348	7407	7419	7429	7453	7464	7474	7498	7511
	7523	7553	7571	7580	7591	7600	7713	7737	7753	7770	7821	7863	7936	7945	7954
	7962	7972	7978	7984	7990	8058	8079	8087	8125	8166	8201	8250	8288	8305	8320
	8336														
ERRSF	1118#	4635#													
ERRSOF	1130#	4635#													
ERRTBL	1140#	4635#													
ESCAPE	1156#	4635#	5697	5708	5740	5776	5809	5846	5860	5904	5948	5992	6015	6079	6137
	6198	6211	6224	6255	6267	6279	6312	6326	6339	6371	6384	6397	6430	6443	6456
	6489	6502	6515	6553	6566	6579	6611	6624	6637	6669	6683	6696	6728	6742	6755
	6786	6800	6813	6844	6858	6871	6905	6922	6935	6966	6980	6993	7048	7096	7220
	7379	7409	7421	7455	7466	7714	7754	7771	7866	7938	7948	7956	7964	7974	7979
	7985	8080	8128												
EXIT	1186#	4635#	5457	5761	5793	5828	6030	6118	6180	6240	6295	6355	6413	6472	6531
	6595	6653	6712	6771	6829	6889	6951	7007	7067	7140	7299	7396	7442	7584	7775
	7835	7870	7889	7992	8041	8103	8149	8219							



CZDMQAO M8207 STATIC DIAG #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 PAGE 60-3  
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0202

M\$ENDE	3145#	4635#	4735#	4747#	5374#	5375#	5376#	5377#	5379#	5381#	5382#	5383#	5384#	5385#	5386#
	5387#	5389#	5390#	5394#	5395#	5396#	5400#	5405#	5409#	5413#	5418#	5422#	5427#	5431#	5436#
	5467#	5603#	5624#	5637#	5654#	5670#	5710#	5753#	5777#	5785#	5810#	5819#	5847#	5861#	5867#
	5914#	5958#	6021#	6108#	6165#	6199#	6212#	6225#	6226#	6256#	6268#	6280#	6281#	6313#	6327#
	6340#	6341#	6372#	6385#	6398#	6399#	6431#	6444#	6457#	6458#	6490#	6503#	6516#	6517#	6554#
	6567#	6580#	6581#	6612#	6625#	6638#	6639#	6670#	6684#	6697#	6698#	6729#	6743#	6756#	6757#
	6787#	6801#	6814#	6815#	6845#	6859#	6872#	6873#	6906#	6923#	6936#	6937#	6967#	6981#	6994#
	6995#	7053#	7103#	7264#	7386#	7431#	7476#	7526#	7607#	7784#	7824#	7876#	8011#	8061#	8091#
	8134#	8170#	8204#	8254#	8342#	8369#	8404#								
M\$ERRI	2365#	4635#	5704#	5738#	5750#	5775#	5808#	5845#	5859#	5903#	5947#	5991#	6014#	6057#	6075#
	6106#	6136#	6151#	6163#	6197#	6210#	6223#	6254#	6266#	6278#	6310#	6324#	6338#	6370#	6383#
	6396#	6429#	6442#	6455#	6488#	6501#	6514#	6551#	6565#	6578#	6610#	6623#	6636#	6668#	6682#
	6695#	6727#	6741#	6754#	6785#	6799#	6812#	6843#	6857#	6870#	6904#	6920#	6934#	6965#	6979#
	6992#	7043#	7092#	7189#	7261#	7348#	7407#	7419#	7429#	7453#	7464#	7474#	7498#	7511#	7523#
	7553#	7571#	7580#	7591#	7600#	7713#	7737#	7753#	7770#	7821#	7863#	7936#	7945#	7954#	7962#
	7972#	7978#	7984#	7990#	8058#	8079#	8087#	8125#	8166#	8201#	8250#	8288#	8305#	8320#	8336#
M\$ESCA	2921#	4635#	5697#	5708#	5740#	5776#	5809#	5846#	5860#	5904#	5948#	5992#	6015#	6079#	6137#
	6198#	6211#	6224#	6255#	6267#	6279#	6312#	6326#	6339#	6371#	6384#	6397#	6430#	6443#	6456#
	6489#	6502#	6515#	6553#	6566#	6579#	6611#	6624#	6637#	6669#	6683#	6696#	6728#	6742#	6755#
	6786#	6800#	6813#	6844#	6858#	6871#	6905#	6922#	6935#	6966#	6980#	6993#	7048#	7096#	7220#
	7379#	7409#	7421#	7455#	7466#	7714#	7754#	7771#	7866#	7938#	7948#	7956#	7964#	7974#	7979#
	7985#	8080#	8128#												
M\$ESCS	2932#	4635#	5697#	5708#	5740#	5776#	5809#	5846#	5860#	5904#	5948#	5992#	6015#	6079#	6137#
	6198#	6211#	6224#	6255#	6267#	6279#	6312#	6326#	6339#	6371#	6384#	6397#	6430#	6443#	6456#
	6489#	6502#	6515#	6553#	6566#	6579#	6611#	6624#	6637#	6669#	6683#	6696#	6728#	6742#	6755#
	6786#	6800#	6813#	6844#	6858#	6871#	6905#	6922#	6935#	6966#	6980#	6993#	7048#	7096#	7220#
	7379#	7409#	7421#	7455#	7466#	7714#	7754#	7771#	7866#	7938#	7948#	7956#	7964#	7974#	7979#
	7985#	8080#	8128#												
M\$EXCP	3186#	4635#	8360#	8361#	8368#										
M\$EXIT	2943#	4635#	5457#	5761#	5793#	5828#	6030#	6118#	6180#	6240#	6295#	6355#	6413#	6472#	6531#
	6595#	6653#	6712#	6771#	6829#	6889#	6951#	7007#	7067#	7140#	7299#	7396#	7442#	7584#	7775#
	7835#	7870#	7889#	7992#	8041#	8103#	8149#	8219#							
M\$EXSE	2965#	4635#	5457#	5761#	5793#	5828#	6030#	6118#	6180#	6240#	6295#	6355#	6413#	6472#	6531#
	6595#	6653#	6712#	6771#	6829#	6889#	6951#	7007#	7067#	7140#	7299#	7396#	7442#	7584#	7775#
	7835#	7870#	7889#	7992#	8041#	8103#	8149#	8219#							
M\$EXTJ	2954#	4635#	5457#	5761#	5793#	5828#	6030#	6118#	6180#	6240#	6295#	6355#	6413#	6472#	6531#
	6595#	6653#	6712#	6771#	6829#	6889#	6951#	7007#	7067#	7140#	7299#	7396#	7442#	7584#	7775#
	7835#	7870#	7889#	7992#	8041#	8103#	8149#	8219#							
M\$GEN	3087#	4635#	4641#	4675#	4683#	4697#	4720#	4735#	4744#	4747#	4790#	4924#	5374#	5375#	5376#
	5377#	5379#	5381#	5382#	5383#	5384#	5385#	5386#	5387#	5389#	5390#	5394#	5395#	5396#	5398#
	5400#	5402#	5405#	5406#	5409#	5410#	5413#	5415#	5418#	5419#	5422#	5423#	5427#	5428#	5431#
	5433#	5436#	5449#	5467#	5484#	5603#	5606#	5624#	5634#	5637#	5651#	5654#	5669#	5670#	5690#
	5710#	5720#	5753#	5760#	5777#	5785#	5792#	5810#	5819#	5827#	5847#	5861#	5867#	5876#	5914#
	5921#	5958#	5966#	6021#	6029#	6108#	6117#	6165#	6178#	6199#	6212#	6225#	6226#	6239#	6256#
	6268#	6280#	6281#	6294#	6313#	6327#	6340#	6341#	6354#	6372#	6385#	6398#	6399#	6412#	6431#
	6444#	6457#	6458#	6471#	6490#	6503#	6516#	6517#	6530#	6554#	6567#	6580#	6581#	6594#	6612#
	6625#	6638#	6639#	6652#	6670#	6684#	6697#	6698#	6711#	6729#	6743#	6756#	6757#	6770#	6787#
	6801#	6814#	6815#	6828#	6845#	6859#	6872#	6873#	6888#	6906#	6923#	6936#	6937#	6950#	6967#
	6981#	6994#	6995#	7006#	7053#	7066#	7103#	7138#	7264#	7297#	7386#	7395#	7431#	7441#	7476#
	7484#	7526#	7537#	7607#	7618#	7784#	7794#	7824#	7833#	7876#	7886#	8011#	8022#	8061#	8071#
	8091#	8101#	8134#	8147#	8170#	8183#	8204#	8217#	8254#	8266#	8342#	8358#	8369#	8401#	8404#
	8413#														
M\$GENB	2764#	4635#													
M\$GETS	3079#	4635#	4687#	4735#	4747#	5374#	5375#	5376#	5377#	5379#	5381#	5382#	5383#	5384#	5385#
	5386#	5387#	5389#	5390#	5394#	5395#	5396#	5400#	5405#	5409#	5413#	5418#	5422#	5427#	5431#
	5436#	5467#	5603#	5624#	5637#	5654#	5670#	5710#	5753#	5776#	5777#	5785#	5809#	5810#	5819#

	5846#	5847#	5860#	5861#	5867#	5914#	5958#	6021#	6108#	6165#	6198#	6199#	6211#	6212#	6224#
	6225#	6226#	6255#	6256#	6267#	6268#	6279#	6280#	6281#	6312#	6313#	6326#	6327#	6339#	6340#
	6341#	6371#	6372#	6384#	6385#	6397#	6398#	6399#	6430#	6431#	6443#	6444#	6456#	6457#	6458#
	6489#	6490#	6502#	6503#	6515#	6516#	6517#	6553#	6554#	6566#	6567#	6579#	6580#	6581#	6611#
	6612#	6624#	6625#	6637#	6638#	6639#	6669#	6670#	6683#	6684#	6696#	6697#	6698#	6728#	6729#
	6742#	6743#	6755#	6756#	6757#	6786#	6787#	6800#	6801#	6813#	6814#	6815#	6844#	6845#	6858#
	6859#	6871#	6872#	6873#	6905#	6906#	6922#	6923#	6935#	6936#	6937#	6966#	6967#	6980#	6981#
	6993#	6994#	6995#	7053#	7103#	7264#	7386#	7431#	7476#	7526#	7607#	7784#	7824#	7876#	8011#
	8061#	8091#	8134#	8170#	8204#	8254#	8342#	8369#	8404#						
M\$GETT	2634#	4635#	5457#	5697#	5708#	5740#	5761#	5776#	5793#	5809#	5828#	5846#	5860#	5904#	5948#
	5992#	6015#	6030#	6079#	6118#	6137#	6180#	6198#	6211#	6224#	6240#	6255#	6267#	6279#	6295#
	6312#	6326#	6339#	6355#	6371#	6384#	6397#	6413#	6430#	6443#	6456#	6472#	6489#	6502#	6515#
	6531#	6553#	6566#	6579#	6595#	6611#	6624#	6637#	6653#	6669#	6683#	6696#	6712#	6728#	6742#
	6755#	6771#	6786#	6800#	6813#	6829#	6844#	6858#	6871#	6889#	6905#	6922#	6935#	6951#	6966#
	6980#	6993#	7007#	7048#	7067#	7096#	7140#	7220#	7299#	7379#	7396#	7409#	7421#	7442#	7455#
	7466#	7584#	7714#	7754#	7771#	7775#	7835#	7866#	7870#	7889#	7938#	7948#	7956#	7964#	7974#
	7979#	7985#	7992#	8041#	8080#	8103#	8128#	8149#	8219#						
M\$GNGB	2689#	4635#	4641#	4675#	4683#	4697#	4720#	4744#	4796#	4924#	5374#	5375#	5376#	5377#	5379#
	5381#	5382#	5383#	5384#	5385#	5386#	5387#	5389#	5390#	5394#	5395#	5396#	5398#	5402#	5406#
	5410#	5415#	5419#	5423#	5428#	5433#	5449#	5484#	5606#	5634#	5651#	5669#	8358#	8401#	8413#
M\$GNIN	3101#	4635#	4675#	4697#	4720#	4744#	4796#	4924#	5374#	5375#	5376#	5377#	5379#	5381#	5382#
	5383#	5384#	5385#	5386#	5387#	5389#	5390#	5394#	5395#	5396#	5399#	5400#	5403#	5404#	5405#
	5407#	5408#	5409#	5411#	5412#	5413#	5416#	5417#	5418#	5420#	5421#	5422#	5424#	5425#	5426#
	5427#	5429#	5430#	5431#	5434#	5435#	5436#	5457#	5467#	5499#	5500#	5502#	5503#	5505#	5506#
	5508#	5509#	5522#	5523#	5603#	5620#	5624#	5635#	5637#	5653#	5654#	5670#	5697#	5704#	5708#
	5710#	5738#	5740#	5750#	5753#	5761#	5767#	5775#	5776#	5777#	5785#	5793#	5799#	5808#	5809#
	5810#	5819#	5828#	5833#	5845#	5846#	5847#	5848#	5859#	5860#	5861#	5867#	5903#	5904#	5909#
	5914#	5947#	5948#	5953#	5958#	5991#	5992#	6014#	6015#	6016#	6021#	6030#	6057#	6075#	6079#
	6080#	6106#	6108#	6118#	6136#	6137#	6151#	6163#	6165#	6180#	6185#	6197#	6198#	6199#	6200#
	6210#	6211#	6212#	6213#	6223#	6224#	6225#	6226#	6240#	6245#	6254#	6255#	6256#	6257#	6266#
	6267#	6268#	6269#	6278#	6279#	6280#	6281#	6295#	6300#	6310#	6312#	6313#	6314#	6324#	6326#
	6327#	6328#	6338#	6339#	6340#	6341#	6355#	6360#	6370#	6371#	6372#	6373#	6383#	6384#	6385#
	6386#	6396#	6397#	6398#	6399#	6413#	6419#	6429#	6430#	6431#	6432#	6442#	6443#	6444#	6445#
	6455#	6456#	6457#	6458#	6472#	6478#	6488#	6489#	6490#	6491#	6501#	6502#	6503#	6504#	6514#
	6515#	6516#	6517#	6531#	6536#	6551#	6553#	6554#	6555#	6565#	6566#	6567#	6568#	6578#	6579#
	6580#	6581#	6595#	6600#	6610#	6611#	6612#	6613#	6623#	6624#	6625#	6626#	6636#	6637#	6638#
	6639#	6653#	6657#	6668#	6669#	6670#	6671#	6682#	6683#	6684#	6685#	6695#	6696#	6697#	6698#
	6712#	6716#	6727#	6728#	6729#	6730#	6741#	6742#	6743#	6744#	6754#	6755#	6756#	6757#	6771#
	6775#	6785#	6786#	6787#	6788#	6799#	6800#	6801#	6802#	6812#	6813#	6814#	6815#	6829#	6833#
	6843#	6844#	6845#	6846#	6857#	6858#	6859#	6860#	6870#	6871#	6872#	6873#	6889#	6893#	6904#
	6905#	6906#	6907#	6920#	6922#	6923#	6924#	6934#	6935#	6936#	6937#	6951#	6955#	6965#	6966#
	6967#	6968#	6979#	6980#	6981#	6982#	6992#	6993#	6994#	6995#	7007#	7043#	7048#	7053#	7067#
	7092#	7096#	7103#	7140#	7189#	7220#	7261#	7264#	7299#	7348#	7379#	7386#	7396#	7407#	7409#
	7419#	7421#	7429#	7431#	7442#	7453#	7455#	7464#	7466#	7474#	7476#	7498#	7511#	7523#	7526#
	7538#	7553#	7571#	7580#	7584#	7591#	7600#	7607#	7713#	7714#	7737#	7753#	7754#	7770#	7771#
	7775#	7784#	7821#	7824#	7835#	7863#	7866#	7870#	7876#	7889#	7936#	7938#	7945#	7948#	7954#
	7956#	7962#	7964#	7972#	7974#	7978#	7979#	7984#	7985#	7990#	7992#	8011#	8041#	8058#	8061#
	8079#	8080#	8087#	8091#	8103#	8125#	8128#	8134#	8149#	8166#	8170#	8201#	8204#	8219#	8250#
	8254#	8269#	8287#	8288#	8290#	8305#	8307#	8320#	8323#	8325#	8336#	8342#	8358#	8360#	8361#
	8368#	8369#	8401#	8404#	8413#										
M\$GNLS	2717#	4635#	5777#	5810#	5847#	5861#	6199#	6212#	6225#	6256#	6268#	6280#	6313#	6327#	6340#
	6372#	6385#	6398#	6431#	6444#	6457#	6490#	6503#	6516#	6554#	6567#	6580#	6612#	6625#	6638#
	6670#	6684#	6697#	6729#	6743#	6756#	6787#	6801#	6814#	6845#	6859#	6872#	6906#	6923#	6936#
	6967#	6981#	6994#												
M\$GNSU	2679#	4635#													
M\$GNLS	2659#	4635#	4735#	4747#	5374#	5375#	5376#	5377#	5379#	5381#	5382#	5383#	5384#	5385#	5386#

	5387#	5389#	5390#	5394#	5395#	5396#	5400#	5405#	5409#	5413#	5418#	5422#	5427#	5431#	5436#
	5467#	5603#	5624#	5637#	5654#	5670#	5710#	5753#	5785#	5819#	5867#	5914#	5958#	6021#	6108#
	6165#	6226#	6281#	6341#	6399#	6458#	6517#	6581#	6639#	6698#	6757#	6815#	6873#	6937#	6995#
	7053#	7103#	7264#	7386#	7431#	7476#	7526#	7607#	7784#	7824#	7876#	8011#	8061#	8091#	8134#
MSGNTE	8170#	8204#	8254#	8342#	8369#	8404#									
	2669#	4635#	5690#	5720#	5760#	5792#	5827#	5876#	5921#	5966#	6029#	6117#	6178#	6239#	6294#
	6354#	6412#	6471#	6530#	6594#	6652#	6711#	6770#	6828#	6888#	6950#	7006#	7066#	7138#	7297#
	7395#	7441#	7484#	7537#	7618#	7794#	7833#	7886#	8022#	8071#	8101#	8147#	8183#	8217#	8266#
M\$HAPT	2477#	4635#	4675#												
M\$HNAP	2565#	4635#	4675#												
M\$INCR	3054#	4635#	4641#	4683#	4720#	4744#	5374#	5375#	5376#	5377#	5379#	5381#	5382#	5383#	5384#
	5385#	5386#	5387#	5389#	5390#	5394#	5395#	5396#	5398#	5399#	5400#	5402#	5403#	5404#	5405#
	5406#	5407#	5408#	5409#	5410#	5411#	5412#	5413#	5415#	5416#	5417#	5418#	5419#	5420#	5421#
	5422#	5423#	5424#	5425#	5426#	5427#	5428#	5429#	5430#	5431#	5433#	5434#	5435#	5436#	5449#
	5467#	5484#	5499#	5502#	5505#	5508#	5522#	5603#	5606#	5620#	5624#	5634#	5635#	5637#	5651#
	5653#	5654#	5669#	5670#	5690#	5697#	5704#	5708#	5710#	5720#	5738#	5740#	5750#	5753#	5760#
	5761#	5767#	5775#	5776#	5777#	5785#	5792#	5793#	5799#	5808#	5809#	5810#	5819#	5827#	5828#
	5833#	5845#	5846#	5847#	5848#	5859#	5860#	5861#	5867#	5876#	5903#	5904#	5909#	5914#	5921#
	5947#	5948#	5953#	5958#	5966#	5991#	5992#	6014#	6015#	6016#	6021#	6029#	6030#	6057#	6075#
	6079#	6080#	6106#	6108#	6117#	6118#	6136#	6137#	6151#	6163#	6165#	6178#	6180#	6185#	6197#
	6198#	6199#	6200#	6210#	6211#	6212#	6213#	6223#	6224#	6225#	6226#	6239#	6240#	6245#	6254#
	6255#	6256#	6257#	6266#	6267#	6268#	6269#	6278#	6279#	6280#	6281#	6294#	6295#	6300#	6310#
	6312#	6313#	6314#	6324#	6326#	6327#	6328#	6338#	6339#	6340#	6341#	6354#	6355#	6360#	6370#
	6371#	6372#	6373#	6383#	6384#	6385#	6386#	6396#	6397#	6398#	6399#	6412#	6413#	6419#	6429#
	6430#	6431#	6432#	6442#	6443#	6444#	6445#	6455#	6456#	6457#	6458#	6471#	6472#	6478#	6488#
	6489#	6490#	6491#	6501#	6502#	6503#	6504#	6514#	6515#	6516#	6517#	6530#	6531#	6536#	6551#
	6553#	6554#	6555#	6565#	6566#	6567#	6568#	6578#	6579#	6580#	6581#	6594#	6595#	6600#	6610#
	6611#	6612#	6613#	6623#	6624#	6625#	6626#	6636#	6637#	6638#	6639#	6652#	6653#	6657#	6668#
	6669#	6670#	6671#	6682#	6683#	6684#	6685#	6695#	6696#	6697#	6698#	6711#	6712#	6716#	6727#
	6728#	6729#	6730#	6741#	6742#	6743#	6744#	6754#	6755#	6756#	6757#	6770#	6771#	6775#	6785#
	6786#	6787#	6788#	6799#	6800#	6801#	6802#	6812#	6813#	6814#	6815#	6828#	6829#	6833#	6843#
	6844#	6845#	6846#	6857#	6858#	6859#	6860#	6870#	6871#	6872#	6873#	6888#	6889#	6893#	6904#
	6905#	6906#	6907#	6920#	6922#	6923#	6924#	6934#	6935#	6936#	6937#	6950#	6951#	6955#	6965#
	6966#	6967#	6968#	6979#	6980#	6981#	6982#	6992#	6993#	6994#	6995#	7006#	7007#	7043#	7048#
	7053#	7066#	7067#	7092#	7096#	7103#	7138#	7140#	7189#	7220#	7261#	7264#	7297#	7299#	7348#
	7379#	7386#	7395#	7396#	7407#	7409#	7419#	7421#	7429#	7431#	7441#	7442#	7453#	7455#	7464#
	7466#	7474#	7476#	7484#	7498#	7511#	7523#	7526#	7537#	7538#	7553#	7571#	7580#	7584#	7591#
	7600#	7607#	7618#	7713#	7714#	7737#	7753#	7754#	7770#	7771#	7775#	7784#	7794#	7821#	7824#
	7833#	7835#	7863#	7866#	7870#	7876#	7886#	7889#	7936#	7938#	7945#	7948#	7954#	7956#	7962#
	7964#	7972#	7974#	7978#	7979#	7984#	7985#	7990#	7992#	8011#	8022#	8041#	8058#	8061#	8071#
	8079#	8080#	8087#	8091#	8101#	8103#	8125#	8128#	8134#	8147#	8149#	8166#	8170#	8183#	8201#
	8204#	8217#	8219#	8250#	8254#	8266#	8269#	8287#	8288#	8290#	8305#	8307#	8320#	8323#	8325#
	8336#	8342#	8358#	8401#											
M\$IOSE	2431#	4635#													
M\$LDRO	2771#	4635#	5499#	5502#	5505#	5508#	5522#	5620#	8269#						
M\$MASK	2390#	4635#													
M\$MCHI	87#	4635#													
M\$MCLO	2327#	4635#													
M\$MSK1	2402#	4635#													
M\$POP	2646#	4635#	4687#	4735#	4747#	5374#	5375#	5376#	5377#	5379#	5381#	5382#	5383#	5384#	5385#
	5386#	5387#	5389#	5390#	5394#	5395#	5396#	5400#	5405#	5409#	5413#	5418#	5422#	5427#	5431#
	5436#	5467#	5603#	5624#	5637#	5654#	5670#	5710#	5753#	5777#	5785#	5810#	5819#	5847#	5861#
	5867#	5914#	5958#	6021#	6108#	6165#	6199#	6212#	6225#	6226#	6256#	6268#	6280#	6281#	6313#
	6327#	6340#	6341#	6372#	6385#	6398#	6399#	6431#	6444#	6457#	6458#	6490#	6503#	6516#	6517#
	6554#	6567#	6580#	6581#	6612#	6625#	6638#	6670#	6684#	6697#	6698#	6729#	6743#	6756#	6757#
	6757#	6787#	6801#	6814#	6815#	6845#	6859#	6872#	6873#	6906#	6923#	6936#	6937#	6967#	6981#

M\$PRIN	6994#	6995#	7053#	7103#	7264#	7386#	7431#	7476#	7526#	7607#	7784#	7824#	7876#	8011#	8061#
M\$PUSH	8091#	8134#	8170#	8204#	8254#	8342#	8369#	8404#							
	2349#	4635#	5374#	5375#	5376#	5377#	5379#	5381#	5382#	5383#	5384#	5385#	5386#	5387#	5389#
	5390#	5394#	5395#	5396#	5399#	5403#	5404#	5407#	5408#	5411#	5412#	5416#	5417#	5420#	5421#
M\$PUT	5424#	5425#	5426#	5429#	5430#	5434#	5435#								
	2337#	4635#	4641#	4683#	4720#	4744#	5374#	5375#	5376#	5377#	5379#	5381#	5382#	5383#	5384#
	5385#	5386#	5387#	5389#	5390#	5394#	5395#	5396#	5398#	5402#	5406#	5410#	5415#	5419#	5423#
	5428#	5433#	5449#	5484#	5606#	5634#	5651#	5669#	5690#	5720#	5760#	5767#	5792#	5799#	5827#
	5833#	5848#	5876#	5921#	5966#	6029#	6117#	6178#	6185#	6200#	6213#	6239#	6245#	6257#	6269#
	6294#	6300#	6314#	6328#	6354#	6360#	6373#	6386#	6412#	6419#	6432#	6445#	6471#	6478#	6491#
	6504#	6530#	6536#	6555#	6568#	6594#	6600#	6613#	6626#	6652#	6657#	6671#	6685#	6711#	6716#
	6730#	6744#	6770#	6775#	6788#	6802#	6828#	6833#	6846#	6860#	6888#	6893#	6907#	6924#	6950#
	6955#	6968#	6982#	7006#	7066#	7138#	7297#	7395#	7441#	7484#	7537#	7618#	7794#	7833#	7886#
	8022#	8071#	8101#	8147#	8183#	8217#	8266#	8358#	8401#						
M\$PUT1	2819#	4635#	5374#	5375#	5376#	5377#	5379#	5381#	5382#	5383#	5384#	5385#	5386#	5387#	5389#
	5390#	5394#	5395#	5396#	5399#	5403#	5404#	5407#	5408#	5411#	5412#	5416#	5417#	5420#	5421#
	5424#	5425#	5426#	5429#	5430#	5434#	5435#								
	2842#	4635#	5374#	5375#	5376#	5377#	5379#	5381#	5382#	5383#	5384#	5385#	5386#	5387#	5389#
	5390#	5394#	5395#	5396#	5399#	5403#	5404#	5407#	5408#	5411#	5412#	5416#	5417#	5420#	5421#
	5424#	5425#	5426#	5429#	5430#	5434#	5435#								
M\$RADI	3151#	4635#	8360#	8361#	8368#										
M\$RBRO	2787#	4635#													
M\$RNRO	2802#	4635#	5522#												
M\$SETS	3071#	4635#	4641#	4683#	4720#	4744#	5374#	5375#	5376#	5377#	5379#	5381#	5382#	5383#	5384#
	5385#	5386#	5387#	5389#	5390#	5394#	5395#	5396#	5398#	5402#	5406#	5410#	5415#	5419#	5423#
	5428#	5433#	5449#	5484#	5606#	5634#	5651#	5669#	5690#	5720#	5760#	5767#	5792#	5799#	5827#
	5833#	5848#	5876#	5921#	5966#	6029#	6117#	6178#	6185#	6200#	6213#	6239#	6245#	6257#	6269#
	6294#	6300#	6314#	6328#	6354#	6360#	6373#	6386#	6412#	6419#	6432#	6445#	6471#	6478#	6491#
	6504#	6530#	6536#	6555#	6568#	6594#	6600#	6613#	6626#	6652#	6657#	6671#	6685#	6711#	6716#
	6730#	6744#	6770#	6775#	6788#	6802#	6828#	6833#	6846#	6860#	6888#	6893#	6907#	6924#	6950#
	6955#	6968#	6982#	7006#	7066#	7138#	7297#	7395#	7441#	7484#	7537#	7618#	7794#	7833#	7886#
	8022#	8071#	8101#	8147#	8183#	8217#	8266#	8358#	8401#						
M\$STAR	2468#	4635#													
M\$SVC	2746#	4635#	5374#	5375#	5376#	5377#	5379#	5381#	5382#	5383#	5384#	5385#	5386#	5387#	5389#
	5390#	5394#	5395#	5396#	5399#	5400#	5403#	5404#	5405#	5407#	5408#	5409#	5411#	5412#	5413#
	5416#	5417#	5418#	5420#	5421#	5422#	5424#	5425#	5426#	5427#	5429#	5430#	5431#	5434#	5435#
	5436#	5457#	5467#	5499#	5502#	5505#	5508#	5522#	5603#	5620#	5624#	5635#	5637#	5653#	5654#
	5670#	5697#	5704	5708#	5710#	5738	5740#	5750	5753#	5761#	5767#	5775	5776#	5777#	5785#
	5793#	5799#	5808	5809#	5810#	5819#	5828#	5833#	5845	5846#	5847#	5848#	5859	5860#	5861#
	5867#	5903	5904#	5909#	5914#	5947	5948#	5953#	5958#	5991	5992#	6014	6015#	6016#	6021#
	6030#	6057	6075	6079#	6080#	6106	6108#	6118#	6136	6137#	6151	6163	6165#	6180#	6185#
	6197	6198#	6199#	6200#	6210	6211#	6212#	6213#	6223	6224#	6225#	6226#	6240#	6245#	6254
	6255#	6256#	6257#	6266	6267#	6268#	6269#	6278	6279#	6280#	6281#	6295#	6300#	6310	6312#
	6313#	6314#	6324	6326#	6327#	6328#	6338	6339#	6340#	6341#	6355#	6360#	6370	6371#	6372#
	6373#	6383	6384#	6385#	6386#	6396	6397#	6398#	6399#	6413#	6419#	6429	6430#	6431#	6432#
	6442	6443#	6444#	6445#	6455	6456#	6457#	6458#	6472#	6478#	6488	6489#	6490#	6491#	6501
	6502#	6503#	6504#	6514	6515#	6516#	6517#	6531#	6536#	6551	6553#	6554#	6555#	6565	6566#
	6567#	6568#	6578	6579#	6580#	6581#	6595#	6600#	6610	6611#	6612#	6613#	6623	6624#	6625#
	6626#	6636	6637#	6638#	6639#	6653#	6657#	6668	6669#	6670#	6671#	6682	6683#	6684#	6685#
	6695	6696#	6697#	6698#	6712#	6716#	6727	6728#	6729#	6730#	6741	6742#	6743#	6744#	6754
	6755#	6756#	6757#	6771#	6775#	6785	6786#	6787#	6788#	6799	6800#	6801#	6802#	6812	6813#
	6814#	6815#	6829#	6833#	6843	6844#	6845#	6846#	6857	6858#	6859#	6860#	6870	6871#	6872#
	6873#	6889#	6893#	6904	6905#	6906#	6907#	6920	6922#	6923#	6924#	6934	6935#	6936#	6937#
	6951#	6955#	6965	6966#	6967#	6968#	6979	6980#	6981#	6982#	6992	6993#	6994#	6995#	7007#
	7043	7048#	7053#	7067#	7092	7096#	7103#	7140#	7189	7220#	7261	7264#	7299#	7348	7379#
	7386#	7396#	7407	7409#	7419	7421#	7429	7431#	7442#	7453	7455#	7464	7466#	7474	7476#

M\$TLAB	7498#	7511#	7523#	7526#	7538#	7553#	7571#	7580#	7584#	7591#	7600#	7607#	7713#	7714#	7737#
	7753#	7754#	7770#	7771#	7775#	7784#	7821#	7824#	7835#	7863#	7866#	7870#	7876#	7889#	7936#
	7938#	7945#	7948#	7954#	7956#	7962#	7964#	7972#	7974#	7978#	7979#	7984#	7985#	7990#	7992#
	8011#	8041#	8058#	8061#	8079#	8080#	8087#	8091#	8103#	8125#	8128#	8134#	8149#	8166#	8170#
	8201#	8204#	8219#	8250#	8254#	8269#	8287#	8288#	8290#	8305#	8307#	8320#	8323#	8325#	8336#
	8342#														
	2739#	4635#	5374#	5375#	5376#	5377#	5379#	5381#	5382#	5383#	5384#	5385#	5386#	5387#	5389#
	5390#	5394#	5395#	5396#	5399#	5400#	5403#	5404#	5405#	5407#	5408#	5409#	5411#	5412#	5413#
	5416#	5417#	5418#	5420#	5421#	5422#	5424#	5425#	5426#	5427#	5429#	5430#	5431#	5434#	5435#
	5436#	5467#	5499#	5502#	5505#	5508#	5522#	5603#	5620#	5624#	5635#	5637#	5653#	5654#	5670#
	5697#	5704#	5708#	5710#	5738#	5740#	5750#	5753#	5761#	5767#	5775#	5776#	5777#	5785#	5793#
	5799#	5808#	5809#	5810#	5819#	5828#	5833#	5845#	5846#	5847#	5848#	5859#	5860#	5861#	5867#
	5903#	5904#	5909#	5914#	5947#	5948#	5953#	5958#	5991#	5992#	6014#	6015#	6016#	6021#	6030#
	6057#	6075#	6079#	6080#	6106#	6108#	6118#	6136#	6137#	6151#	6163#	6165#	6180#	6185#	6197#
	6198#	6199#	6200#	6210#	6211#	6212#	6213#	6223#	6224#	6225#	6226#	6240#	6245#	6254#	6255#
	6256#	6257#	6266#	6267#	6268#	6269#	6278#	6279#	6280#	6281#	6295#	6300#	6310#	6312#	6313#
	6314#	6324#	6326#	6327#	6328#	6338#	6339#	6340#	6341#	6355#	6360#	6370#	6371#	6372#	6373#
	6383#	6384#	6385#	6386#	6396#	6397#	6398#	6399#	6413#	6419#	6429#	6430#	6431#	6432#	6442#
	6443#	6444#	6445#	6455#	6456#	6457#	6458#	6472#	6478#	6488#	6489#	6490#	6491#	6501#	6502#
	6503#	6504#	6514#	6515#	6516#	6517#	6531#	6536#	6551#	6553#	6554#	6555#	6565#	6566#	6567#
	6568#	6578#	6579#	6580#	6581#	6595#	6600#	6610#	6611#	6612#	6613#	6623#	6624#	6625#	6626#
	6636#	6637#	6638#	6639#	6653#	6657#	6668#	6669#	6670#	6671#	6682#	6683#	6684#	6685#	6695#
	6696#	6697#	6698#	6712#	6716#	6727#	6728#	6729#	6730#	6741#	6742#	6743#	6744#	6754#	6755#
	6756#	6757#	6771#	6775#	6785#	6786#	6787#	6788#	6799#	6800#	6801#	6802#	6812#	6813#	6814#
	6815#	6829#	6833#	6843#	6844#	6845#	6846#	6857#	6858#	6859#	6860#	6870#	6871#	6872#	6873#
	6889#	6893#	6904#	6905#	6906#	6907#	6920#	6922#	6923#	6924#	6934#	6935#	6936#	6937#	6951#
	6955#	6965#	6966#	6967#	6968#	6979#	6980#	6981#	6982#	6992#	6993#	6994#	6995#	7007#	7043#
	7048#	7053#	7067#	7092#	7096#	7103#	7140#	7189#	7220#	7261#	7264#	7299#	7348#	7379#	7386#
	7396#	7407#	7409#	7419#	7421#	7429#	7431#	7442#	7453#	7455#	7464#	7466#	7474#	7476#	7498#
	7511#	7523#	7526#	7538#	7553#	7571#	7580#	7584#	7591#	7600#	7607#	7713#	7714#	7737#	7753#
	7754#	7770#	7771#	7775#	7784#	7821#	7824#	7835#	7863#	7866#	7870#	7876#	7889#	7936#	7938#
	7945#	7948#	7954#	7956#	7962#	7964#	7972#	7974#	7978#	7979#	7984#	7985#	7990#	7992#	8011#
	8041#	8058#	8061#	8079#	8080#	8087#	8091#	8103#	8125#	8128#	8134#	8149#	8166#	8170#	8201#
	8204#	8219#	8250#	8254#	8269#	8287#	8288#	8290#	8305#	8307#	8320#	8323#	8325#	8336#	8342#
M\$STSL	2728#	4635#	5374#	5375#	5376#	5377#	5379#	5381#	5382#	5383#	5384#	5385#	5386#	5387#	5389#
	5390#	5394#	5395#	5396#	5399#	5400#	5403#	5404#	5405#	5407#	5408#	5409#	5411#	5412#	5413#
	5416#	5417#	5418#	5420#	5421#	5422#	5424#	5425#	5426#	5427#	5429#	5430#	5431#	5434#	5435#
	5436#	5467#	5499#	5502#	5505#	5508#	5522#	5603#	5620#	5624#	5635#	5637#	5653#	5654#	5670#
	5697#	5704#	5708#	5710#	5738#	5740#	5750#	5753#	5761#	5767#	5775#	5776#	5777#	5785#	5793#
	5799#	5808#	5809#	5810#	5819#	5828#	5833#	5845#	5846#	5847#	5848#	5859#	5860#	5861#	5867#
	5903#	5904#	5909#	5914#	5947#	5948#	5953#	5958#	5991#	5992#	6014#	6015#	6016#	6021#	6030#
	6057#	6075#	6079#	6080#	6106#	6108#	6118#	6136#	6137#	6151#	6163#	6165#	6180#	6185#	6197#
	6198#	6199#	6200#	6210#	6211#	6212#	6213#	6223#	6224#	6225#	6226#	6240#	6245#	6254#	6255#
	6256#	6257#	6266#	6267#	6268#	6269#	6278#	6279#	6280#	6281#	6295#	6300#	6310#	6312#	6313#
	6314#	6324#	6326#	6327#	6328#	6338#	6339#	6340#	6341#	6355#	6360#	6370#	6371#	6372#	6373#
	6383#	6384#	6385#	6386#	6396#	6397#	6398#	6399#	6413#	6419#	6429#	6430#	6431#	6432#	6442#
	6443#	6444#	6445#	6455#	6456#	6457#	6458#	6472#	6478#	6488#	6489#	6490#	6491#	6501#	6502#
	6503#	6504#	6514#	6515#	6516#	6517#	6531#	6536#	6551#	6553#	6554#	6555#	6565#	6566#	6567#
	6568#	6578#	6579#	6580#	6581#	6595#	6600#	6610#	6611#	6612#	6613#	6623#	6624#	6625#	6626#
	6636#	6637#	6638#	6639#	6653#	6657#	6668#	6669#	6670#	6671#	6682#	6683#	6684#	6685#	6695#
	6696#	6697#	6698#	6712#	6716#	6727#	6728#	6729#	6730#	6741#	6742#	6743#	6744#	6754#	6755#
	6756#	6757#	6771#	6775#	6785#	6786#	6787#	6788#	6799#	6800#	6801#	6802#	6812#	6813#	6814#
	6815#	6829#	6833#	6843#	6844#	6845#	6846#	6857#	6858#	6859#	6860#	6870#	6871#	6872#	6873#
	6889#	6893#	6904#	6905#	6906#	6907#	6920#	6922#	6923#	6924#	6934#	6935#	6936#	6937#	6951#
	6955#	6965#	6966#	6967#	6968#	6979#	6980#	6981#	6982#	6992#	6993#	6994#	6995#	7007#	7043#
	7048#	7053#	7067#	7092#	7096#	7103#	7140#	7189#	7220#	7261#	7264#	7299#	7348#	7379#	7386#





CZDMQAO M8207 STATIC DIAG #2  
CZDMQA.P11 25-JUN-79 14:07

MACY11 30A(1052) 17-JUL-79 09:23 PAGE 60-9  
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0208

XFERT	2319#	4635#																
SMD	5357#	5374	5375	5376	5377	5379	5381	5382	5383	5384	5385	5386	5387	5389	5390			
	5394	5395	5396															

. ABS. 030032 000

ERRORS DETECTED: 0

SAIL: CZDMQA, CZDMQA/CRF/NL: TOC= CZDMP.MLB, CZDMQA.P11  
RUN-TIME: 163 188 17 SECONDS  
RUN-TIME RATIO: 923/370=2.4  
CORE USED: 19K (37 PAGES)