

# M8207, MCPU

M8207 STATIC DIAG. #1  
CZDMPA0

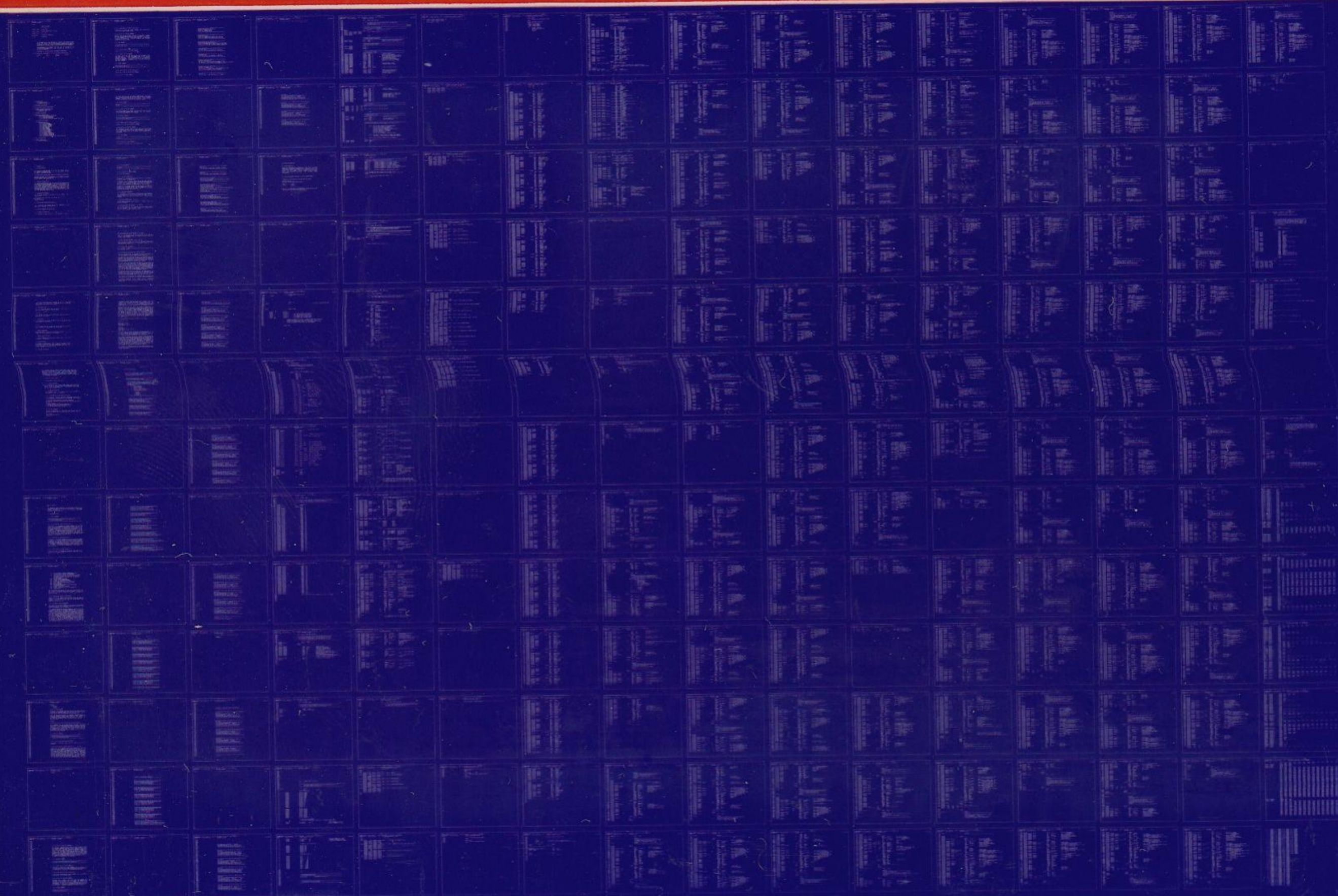
AH-E226A-MC

COPYRIGHT 1979  
FICHE 1 OF 2

SEP 1979

**digital**

MADE IN USA





# M8207,MCPU

M8207 STATIC DIAG. #1  
CZDMPA0

AH-E226A-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-E225A-MC  
PRODUCT NAME: CZDMPA0 M8207 STATIC DIAG #1  
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

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



1.0 INTRODUCTION

1.1 PROGRAM ABSTRACT

THIS DIAGNOSTIC WAS DESIGNED TO TEST OUT THE M8200, M8204,  
OR M8207 MICROPROCESSOR. IT IS THE FIRST 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 STATIC  
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  
PREVIOUS TO RUNNING THIS DIAGNOSTIC.

4.0 GENERAL PROGRAM CONSIDERATIONS

4.1 DIAGNOSTIC SUPERVISOR

THIS PROGRAM IS COMPATIBLE WITH THE STANDALONE DIAGNOSTIC

(ZDMPA M8207 STATIC DIAG. #1 MACY11 30A(1052) 17-JUL-79 14:39 E 1  
(ZDMPA.P11 17-JUL-79 14:33 PROGRAM DOCUMENT PAGE 4-1

SEQ 0004

3442

SUPERVISOR, AND MUST BE LOADED TO BE CO-RESIDENT WITH THE



3444  
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

SUPERVISOR, OR BE PREVIOUSLY COMBINED WITH THE SUPERVISOR 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 30 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

3500  
3501  
3502  
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

THIS PROGRAM CAN BE LOADED FROM PAPER TAPE USING THE ABSOLUTE LOADER OR FROM ACT, SLIDE, OR APT SYSTEMS, OR FROM 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 IDENTIFICATION AND 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  
CZDMP-A-0



CZDMPA M8207 STATIC DIAG. #1  
CZDMPA.P11 17-JUL-79 14:33  
3556

MACY11 30A(1052) 17-JUL-79 14:39 H 1  
PROGRAM DOCUMENT  
M8207 DIAG.#1 OF 2

SEQ 0007

UNIT IS M8200.4,7  
DR>

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:



3615  
3616  
3617  
3618  
3619  
3620  
3621  
3622  
3623  
3624  
3625  
3626  
3627  
3628  
3629  
3630  
3631  
3632  
3633  
3634  
3635  
3636  
3637  
3638  
3639  
3640  
3641  
3642  
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

HOE HALT ON ERROR, CAUSING COMMAND MODE TO BE  
ENTERED WHEN AN ERROR IS ENCOUNTERED  
LOE LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP  
CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK  
OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAIN-  
ING 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  
"N 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

CZDMPA M8207 STATIC DIAG. #1 MACY11 30A(1052) 17-JUL-79 14:39 K 1  
CZDMPA.P11 17-JUL-79 14:33 PROGRAM DOCUMENT PAGE 8-1

SEQ 0010

3671

OCTAL, L FOR YES/NO) IN PARENTHESES AND THE DEFAULT VALUE



AFTER THE PARENTHESES.

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 DIALOGUE. 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

CZDMPA M8207 STA-IC DIAG. #1 MACY11 30A(1052) 17-JUL-79 14:39 M 1  
CZDMPA.P11 17-JUL-79 14:33 PROGRAM DOCUMENT PAGE 9-1

SEQ 0012

3729

COMMAND.

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  
3784

#### 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>  
\*\*\*\*\*

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  
3840

#### 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



3842  
3843  
3844  
3845  
3846  
3847  
3848  
3849  
3850  
3851  
3852  
3853  
3854  
3855  
3856  
3857  
3858  
3859  
3860  
3861  
3862  
3863  
3864  
3865  
3866  
3867  
3868  
3869  
3870  
3871  
3872  
3873  
3874  
3875  
3876  
3877  
3878  
3879  
3880  
3881  
3882  
3883  
3884  
3885  
3886  
3887  
3888  
3889  
3890  
3891  
3892  
3893  
3894  
3895  
3896

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

\*\*\*\*\*  
PR(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.

3898  
3899  
3900  
3901  
3902  
3903  
3904  
3905  
3906  
3907  
3908  
3909  
3910  
3911  
3912  
3913  
3914  
3915  
3916  
3917  
3918  
3919  
3920  
3921  
3922  
3923  
3924  
3925  
3926  
3927  
3928  
3929  
3930  
3931  
3932  
3933  
3934  
3935  
3936  
3937  
3938  
3939  
3940  
3941  
3942  
3943  
3944  
3945  
3946  
3947  
3948  
3949  
3950  
3951

#### 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- INITAIL 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 4 QUESTION 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 CARRIGE RETURN RESPONSE.

##### 1. WHICH MICRO-PROCESSOR: (O) 7?

THE ALLOWABLE RESPONSES ARE 0 (M8200), 4 (M8204), AND THE DEFAULT 7 (M8207).

##### 2. MICRO-PROCESSOR CSR ADDRESS: (O) 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 VALUE IS 160170.

.

3. MICRO-PROCESSOR VECTOR ADDRESS: (0) 300?

THIS IS THE ADDRESS OF THE INPUT INTERRUPT VECTOR FOR THIS DEVICE. THE ALLOWABLE RANGE IS 000-674 (OCTAL), AND THE DEFAULT VALUE IS 300.

4. MICRO-PROCESSOR PRIORITY LEVEL: (0) 5?

THIS IS THE CPU PRIORITY AT WHICH THE INTERRUPT HANDLERS OF THE DEVICE WILL BE EXECUTED. THE ALLOWABLE RANGE IS 0-7, AND THE DEFAULT VALUE IS 5.

6.3.13 SOFTWARE PARAMETERS

NO SOFTWARE PARAMETER QUESTIONS ARE ASKED BY PART 1 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 1, 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.

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  
4054  
4055  
4056  
4057  
4058  
4059  
4060  
4061  
4062  
4063  
4064

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).

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 7 THRU THE END ARE GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS PRINTED OUT FOR 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).



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  
4111  
4112  
4113  
4114  
4115  
4116  
4117  
4118  
4119  
4120  
4121

## 7.0 TEST DESCRIPTION

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

\*\*\*\*\* TEST 2 \*\*\*\*\*  
\*VERIFY THAT RUN CAN BE CLEARED  
\*\*\*\*\*

\*\*\*\*\* TEST 3 \*\*\*\*\*  
\*UNIBUS REGISTER WORD DUAL ADDRESSING TEST  
\*LOAD ALL REGISTERS WITH INCREMENTING PATTERN  
\*READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING  
\*THE SEQUENCE:  
\* 1. CLEAR REGISTER  
\* 2. WRITE PATTERN  
\* 3. VERIFY PATTERN  
\* 4. DO ALL 4 REGISTERS  
\* 5. READ ALL BACK IF ERRORS,  
\* DUAL ADDRESS PROBLEM.  
\*  
\* 1 IN REG 0  
\* 2 IN REG 2  
\* 3 IN REG 4  
\* 4 IN REG 6  
\*\*\*\*\*

\*\*\*\*\* TEST 4 \*\*\*\*\*  
\*CONTROL STATUS REGISTER WRITE/READ TEST  
\*FLOAT A ONE THROUGH BSEL 0  
\*CLEAR BIT0, VERIFY BIT0 WAS CLEARED  
\*\*\*\*\*

\*\*\*\*\* TEST 5 \*\*\*\*\*  
\*CONTROL STATUS REGISTER WRITE/READ TEST  
\*SET BIT9, VERIFY BIT9 WAS SET  
\*CLEAR BIT9, VERIFY BIT9 WAS CLEARED  
\*\*\*\*\*

\*\*\*\*\* TEST 6 \*\*\*\*\*  
\*CONTROL STATUS REGISTER WRITE/READ TEST  
\*SET BIT11, VERIFY BIT11 WAS SET  
\*CLEAR BIT11, VERIFY BIT11 WAS CLEARED

CZDMPA M8207 STATIC DIAG. #  
CZDMPA.P11 17-JUL-79 14:33 MACY11 30A(1052) 17-JUL-79 14:39 H 2 PAGE 16-1  
PROGRAM DOCUMENT

SEQ 0020

4122

\*\*\*\*\*

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  
4169  
4170  
4171  
4172  
4173  
4174  
4175  
4176  
4177  
4178  
4179

\*\*\*\*\* TEST 7 \*\*\*\*\*  
\*CONTROL STATUS REGISTER WRITE/READ TEST  
\*SET BIT12, VERIFY BIT12 WAS SET  
\*CLEAR BIT 12, VERIFY BIT 12 WAS CLEARED  
\*\*\*\*\*

\*\*\*\*\* TEST 8 \*\*\*\*\*  
\*CONTROL OUT REGISTER WRITE/READ TEST  
\*FLOAT A ONE THROUGH SEL2  
\*\*\*\*\*

\*\*\*\*\* TEST 9 \*\*\*\*\*  
\*PORT4 REGISTER WRITE/READ TEST  
\*FLOAT A ONE THROUGH PORT4 REGISTER  
\*FLOAT A ZERO THROUGH PORT4 REGISTER  
\*\*\*\*\*

\*\*\*\*\* TEST 10 \*\*\*\*\*  
\*PORT6 REGISTER WRITE/READ TEST  
\*FLOAT A ONE THROUGH PORT6 REGISTER  
\*FLOAT A ZERO THROUGH PORT6 REGISTER  
\*\*\*\*\*

\*\*\*\*\* TEST 11 \*\*\*\*\*  
\*UNIBUS REGISTER BYTE DUAL ADDRESSING TEST  
\*LOAD ALL REGISTERS WITH INCREMENTING PATTERN  
\*READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING  
\*\*\*\*\*

\*\*\*\*\* TEST 12 \*\*\*\*\*  
\*MAINTENANCE INSTRUCTION REGISTER TEST  
\*VERIFY THAT THE MAINT IR CAN BE WRITTEN TO ALL ZEROS'  
\*AND ALL ONES'. VERIFY THAT IS IS CLEARED ON A BUS RESET.  
\*\*\*\*\*

\*\*\*\*\* TEST 13 \*\*\*\*\*  
\*MICRO PROCESSOR TEST  
\*LOAD KMP06 WITH A MICRO-PROCESSOR INSTRUCTION, CLOCK IT  
\*VERIFY INSTRUCTION EXECUTED PROPERLY  
\*INSTRUCTION SHOULD MOVE IBUS\*4 TO IBUS\*5, IBUS\*4 IS ALL 1'S  
\*AND IBUS\*5 IS ALL 0'S. RESULT SHOULD BE ALL 1'S IN SEL4  
\*\*\*\*\*

CZDMPA M8207 STATIC DIAG. #1 MACY11 30A(1052) 17-JUL-79 14:39 J 2  
CZDMPA.P11 17-JUL-79 14:33 PROGRAM DOCUMENT PAGE 17-1

SEQ 0022

4180

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  
4227  
4228  
4229  
4230  
4231  
4232  
4233  
4234  
4235  
4236  
4237

\*\*\*\*\* TEST 14 \*\*\*\*\*  
\*MICRO PROCESSOR IBUS\* REGISTER WRITE/READ TEST  
\*FLOAT A 1 THROUGH IBUS\* REGISTER 0  
\*FLOAT A 0 THROUGH IBUS\* REGISTER 0  
\*\*\*\*\*

\*\*\*\*\* TEST 15 \*\*\*\*\*  
\*MICRO PROCESSOR IBUS\* REGISTER WRITE/READ TEST  
\*FLOAT A 1 THROUGH IBUS\* REGISTER 2  
\*FLOAT A 0 THROUGH IBUS\* REGISTER 2  
\*\*\*\*\*

\*\*\*\*\* TEST 16 \*\*\*\*\*  
\*MICRO PROCESSOR IBUS\* REGISTER WRITE/READ TEST  
\*FLOAT A 1 THROUGH IBUS\* REGISTER 4  
\*FLOAT A 0 THROUGH IBUS\* REGISTER 4  
\*\*\*\*\*

\*\*\*\*\* TEST 17 \*\*\*\*\*  
\*MICRO PROCESSOR IBUS\* REGISTER WRITE/READ TEST  
\*FLOAT A 1 THROUGH IBUS\* REGISTER 5  
\*FLOAT A 0 THROUGH IBUS\* REGISTER 5  
\*\*\*\*\*

\*\*\*\*\* TEST 18 \*\*\*\*\*  
\*MICRO PROCESSOR IBUS\* REGISTER WRITE/READ TEST  
\*FLOAT A 1 THROUGH IBUS\* REGISTER 10  
\*FLOAT A 0 THROUGH IBUS\* REGISTER 10  
\*\*\*\*\*

\*\*\*\*\* TEST 19 \*\*\*\*\*  
\*MICRO PROCESSOR IBUS\* REGISTER WRITE/READ TEST  
\*FLOAT A 1 THROUGH IBUS\* REGISTER 11  
\*FLOAT A 0 THROUGH IBUS\* REGISTER 11  
\*\*\*\*\*

\*\*\*\*\* TEST 20 \*\*\*\*\*  
\*MICRO PROCESSOR IBUS REBISTER WRITE/READ TEST  
\*FLOAT A 1 THROUGH IBUS REGISTER 0  
\*FLOAT A 0 THROUGH IBUS REGISTER 0  
\*\*\*\*\*

\*\*\*\*\* TEST 21 \*\*\*\*\*  
\*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST



CZDMPA M8207 STATIC DIAG. #1 MACY11 30A(1052) 17-JUL-79 14:39 L 2  
CZDMPA.P11 17-JUL-79 14:33 PROGRAM DOCUMENT PAGE 18-1

SEQ 0024

4238

\*FLOAT A 1 THROUGH IBUS REGISTER 1

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  
4285  
4286  
4287  
4288  
4289  
4290  
4291  
4292  
4293  
4294  
4295

\*FLOAT A 0 THROUGH IBUS REGISTER 1

\*\*\*\*\*

\*\*\*\*\* TEST 22 \*\*\*\*\*

\*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST

\*FLOAT A 1 THROUGH IBUS REGISTER 2

\*FLOAT A 0 THROUGH IBUS REGISTER 2

\*\*\*\*\*

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

\*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST

\*FLOAT A 1 THROUGH IBUS REGISTER 3

\*FLOAT A 0 THROUGH IBUS REGISTER 3

\*\*\*\*\*

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

\*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST

\*FLOAT A 1 THROUGH IBUS REGISTER 4

\*FLOAT A 0 THROUGH IBUS REGISTER 4

\*\*\*\*\*

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

\*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST

\*FLOAT A 1 THROUGH IBUS REGISTER 5

\*FLOAT A 0 THROUGH IBUS REGISTER 5

\*\*\*\*\*

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

\*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST

\*FLOAT A 1 THROUGH IBUS REGISTER 6

\*FLOAT A 0 THROUGH IBUS REGISTER 6

\*\*\*\*\*

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

\*MICRO PROCESSOR IBUS\* REGISTER WRITE/READ TEST

\*FLOAT A 1 THROUGH IBUS\* REGISTER 7

\*FLOAT A 0 THROUGH IBUS\* REGISTER 7

\*\*\*\*\*

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

\*MICRO PROCESSOR IBUS DUAL ADDRESS TEST

\*WRITE ALL IBUS REGISTERS WITH INCREMENTING PATTERN

\*READ ALL IBUS REGISTERS TO VERIFY CORRECT ADDRESSING

\*\*\*\*\*

CZDMPA M8207 STATIC DIAG. #1 MACY11 30A(1052) 17-JUL-79 14:39 N 2  
CZDMPA.P11 17-JUL-79 14:33 PROGRAM DOCUMENT PAGE 19-1

S-L 0026

4296

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  
4343  
4344  
4345  
4346  
4347  
4348  
4349  
4350  
4351  
4352  
4353

\*\*\*\*\* TEST 29 \*\*\*\*\*  
\*MICRO PROCESSOR BR REGISTER TEST  
\*FLOAT A 1 THROUGH THE BR  
\*FLOAT A 0 THROUGH THE BR  
\*\*\*\*\*

\*\*\*\*\* TEST 30 \*\*\*\*\*  
\*SCRATCH PAD TEST  
\*FLOAT A 1 THROUGH EACH SCRATCH PAD LOCATION  
\*FLOAT A 0 THROUGH EACH SCRATCH PAD LOCATION  
\*\*\*\*\*

\*\*\*\*\* TEST 31 \*\*\*\*\*  
\*SCRATCH PAD DUAL ADDRESSING TEST  
\*WRITE AN INCREMENTING PATTERN IN ALL SP LOCATIONS  
\*READ ALL SP LOCATIONS TO VERIFY CORRECT ADDRESSING  
\*\*\*\*\*

\*\*\*\*\* TEST 32 \*\*\*\*\*  
\*INTERRUPT TEST  
\*TEST THAT DEVICE CAN INTERRUPT TO VECTOR A  
\*\*\*\*\*

\*\*\*\*\* TEST 33 \*\*\*\*\*  
\*INTERRUPT TEST  
\*TEST THAT DEVICE CAN INTERRUPT TO VECTOR B  
\*\*\*\*\*

\*\*\*\*\* TEST 34 \*\*\*\*\*  
\*PRIORITY INTERRUPT TEST  
\*SET PS TO ALL BR LEVELS EQUAL OR GREATER THAN  
\*THE M8200,4,7 LEVEL, VERIFY THAT M8200,4,7 DOES NOT INTERRUPT  
\*\*\*\*\*

\*\*\*\*\* TEST 35 \*\*\*\*\*  
\*PRIORITY INTERRUPT TESTS  
\*SET PS TO ALL BR LEVELS LESS THAN THE M8200,4,7 LEVEL  
\*VERIFY THAT ALL M8200,4,7 WILL INTERRUPT  
\*\*\*\*\*

\*\*\*\*\* TEST 36 \*\*\*\*\*  
\*NPR TEST  
\*TEST OF DAT0, 1 WORD FROM UPROC TO 11 MEMORY

CZDMPA M8207 STATIC DIAG. #1    MACY11 30A(1052) 17-JUL-79 14:39 <sup>C 3</sup> PAGE 20-1  
CZDMPA.P11    17-JUL-79 14:33    PROGRAM DOCUMENT

SEQ 0028

4354

\*\*\*\*\*



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  
4401  
4402  
4403  
4404  
4405  
4406  
4407  
4408  
4409  
4410  
4411

\*\*\*\*\* TEST 37 \*\*\*\*\*  
\*NPR TEST  
\*TEST OF DATI, 1 WORD FROM 11 MEMORY TO UPROC  
\*\*\*\*\*

\*\*\*\*\* TEST 38 \*\*\*\*\*  
\*NPR TEST  
\*TEST OF DATOB, 1 BYTE FROM UPROC TO 11 MEMORY  
\*\*\*\*\*

\*\*\*\*\* TEST 39 \*\*\*\*\*  
\*TEST OF EA BITS 16 AND 17  
\*DO A DATO TO AN ADDRESS USING OUT BA BITS 16 AND 17  
\*VERIFY CORRECT RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 40 \*\*\*\*\*  
\*TEST OF EA BITS 16 AND 17  
\*DO A DATI USING IN BA BITS 16 AND 17  
\*VERIFY CORRECT RESULTS  
\*IN ORDER TO DO THIS TEST, WE WILL READ THE DATA FROM THE  
\*CONSOL TTY CSR IF ONE EXISTS  
\*IF NO COSGL TTY CSR AT ADDRESS 177560, THIS TEST  
\*WILL BE SKIPPED  
\*\*\*\*\*

\*\*\*\*\* TEST 41 \*\*\*\*\*  
\*NPR NON-EXISTENT MEMORY TEST  
\*DO A DATO TO A NON-EXISTENT ADDRESS  
\*VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11  
\*\*\*\*\*

\*\*\*\*\* TEST 42 \*\*\*\*\*  
\*NPR NON-EXISTENT MEMORY TEST  
\*DO A DATI FROM A NON-EXISTENT ADDRESS  
\*VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11  
\*\*\*\*\*

\*\*\*\*\* TEST 43 \*\*\*\*\*  
\*NPR TEST  
\*USING DATO, NPR A BINARY COUNT (0-377)  
\*FROM MICRO-PROCESSOR TO ALL AVAILABLE MEMORY  
\*\*\*\*\*

CZDMPA M8207 STATIC DIAG. #1 MACY11 30A(1052) 17-JUL-79 14:39 E 3  
CZDMPA.P11 17-JUL-79 14:33 PROGRAM DOCUMENT PAGE 21-1

SEQ 0030

4412

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  
4459  
4460  
4461  
4462  
4463  
4464  
4465  
4466  
4467  
4468  
4469

\*\*\*\*\* TEST 44 \*\*\*\*\*  
\*ALU C BIT TEST  
\*TEST THAT AN ADD OF 377 AND 377 WILL SET THE C BIT  
\*\*\*\*\*

\*\*\*\*\* TEST 45 \*\*\*\*\*  
\*ALU TEST  
\*TEST OF ALU FUNCTION SEL B WITH C BIT CLEARED  
\*ALU FUNCTION (B) CODE=11  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 46 \*\*\*\*\*  
\*ALU TEST  
\*TEST OF ALU FUNCTION SEL A WITH C BIT CLEARED  
\*ALU FUNCTION (A) CODE=10  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 47 \*\*\*\*\*  
\*ALU TEST  
\*TEST OF ALU FUNCTION A OR NOTB WITH C BIT CLEARED  
\*ALU FUNCTION (A OR NOTB) CODE=12  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 48 \*\*\*\*\*  
\*ALU TEST  
\*TEST OF ALU FUNCTION A AND B WITH C BIT CLEARED  
\*ALU FUNCTION (A AND B) CODE=13  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 49 \*\*\*\*\*  
\*ALU TEST  
\*TEST OF ALU FUNCTION A OR B WITH C BIT CLEARED  
\*ALU FUNCTION (A OR B) CODE=14  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

CZDMPA M8207 STATIC DIAG. #1 MACY11 30A(1052) 17-JUL-79 14:39 G 3  
CZDMPA.P11 17-JUL-79 14:33 PROGRAM DOCUMENT PAGE 22-1

SFO 0032

4470

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  
4517  
4518  
4519  
4520  
4521  
4522  
4523  
4524  
4525  
4526  
4527

\*\*\*\*\* TEST 50 \*\*\*\*\*

\*ALU TEST  
\*TEST OF ALU FUNCTION A XOR B WITH C BIT  
\*ALU FUNCTION (A XOR B) CODE=15  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 51 \*\*\*\*\*

\*ALU TEST  
\*TEST OF ALU FUNCTION ADD WITH C BIT CLEARED  
\*ALU FUNCTION (A PLUS B) CODE=00  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 52 \*\*\*\*\*

\*ALU TEST  
\*TEST OF ALU FUNCTION 2A W.C WITH C BIT CLEARED  
\*ALU FUNCTION (A PLUS A PLUS C) CODE=6  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 53 \*\*\*\*\*

\*ALU TEST  
\*TEST OF ALU FUNCTION SUB WITH C BIT CLEARED  
\*ALU FUNCTION (A-B) CODE=16  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 54 \*\*\*\*\*

\*ALU TEST  
\*TEST OF ALU FUNCTION ADD W/C WITH C BIT CLEARED  
\*ALU FUNCTION (A PLUS B PLUS C) CODE=01  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 55 \*\*\*\*\*

\*ALU TEST  
\*TEST OF ALU FUNCTION SUB W/C WITH C BIT CLEARED  
\*ALU FUNCTION (A-B-C) CODE=2  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS

CZDMPA M8207 STATIC DIAG. #1 MACY11 30A(1052) 17-JUL-79 14:39 <sup>1</sup>3 PAGE 23-1  
CZDMPA.P11 17-JUL-79 14:33 PROGRAM DOCUMENT

SEQ 0034

4528

\*\*\*\*\*

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  
4572  
4573  
4574  
4575  
4576  
4577  
4578  
4579  
4580  
4581  
4582  
4583  
4584  
4585

\*\*\*\*\* TEST 56 \*\*\*\*\*  
\*ALU TEST  
\*TEST OF ALU FUNCTION INC A WITH C BIT CLEARED  
\*ALU FUNCTION (A PLUS 1)      CODE=3  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 57 \*\*\*\*\*  
\*ALU TEST  
\*TEST OF ALU FUNCTION 2A WITH C BIT CLEARED  
\*ALU FUNCTION (A PLUS A)      CODE=5  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 58 \*\*\*\*\*  
\*ALU TEST  
\*TEST OF ALU FUNCTION A PLUS C WITH C BIT CLEARED  
\*ALU FUNCTION (A PLUS C)      CODE=4  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 59 \*\*\*\*\*  
\*ALU TEST  
\*TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT CLEARED  
\*ALU FUNCTION (A-B-1)      CODE=17  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 60 \*\*\*\*\*  
\*ALU TEST  
\*TEST OF ALU FUNCTION DEC A WITH C BIT CLEARED  
\*ALU FUNCTION (A-1)      CODE=7  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 61 \*\*\*\*\*  
\*ALU TEST  
\*TEST OF ALU FUNCTION SEL B WITH C BIT SET  
\*ALU FUNCTION (B)      CODE=11



(ZDMPA M8207 STATIC DIAG. #1    MACY11 30A(1052) 17-JUL-79 14:39 K 3  
(ZDMPA.P11    17-JUL-79 14:33    PROGRAM DOCUMENT    PAGE 24-1

SEQ 0036

4586

\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA

4588  
4589  
4590  
4591  
4592  
4593  
4594  
4595  
4596  
4597  
4598  
4599  
4600  
4601  
4602  
4603  
4604  
4605  
4606  
4607  
4608  
4609  
4610  
4611  
4612  
4613  
4614  
4615  
4616  
4617  
4618  
4619  
4620  
4621  
4622  
4623  
4624  
4625  
4626  
4627  
4628  
4629  
4630  
4631  
4632  
4633  
4634  
4635  
4636  
4637  
4638  
4639  
4640  
4641  
4642  
4643

\*PERFORM THE FUNCTION, VERIFY THE RESULTS

\*\*\*\*\*

\*\*\*\*\* TEST 62 \*\*\*\*\*

\*ALU TEST

\*TEST OF ALU FUNCTION SEL A WITH C BIT SET

\*ALU FUNCTION (A) CODE=10

\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA

\*PERFORM THE FUNCTION, VERIFY THE RESULTS

\*\*\*\*\*

\*\*\*\*\* TEST 63 \*\*\*\*\*

\*ALU TEST

\*TEST OF ALU FUNCTION A OR NOTB WITH C BIT SET

\*ALU FUNCTION (A OR NOTB) CODE=12

\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA

\*PERFORM THE FUNCTION, VERIFY THE RESULTS

\*\*\*\*\*

\*\*\*\*\* TEST 64 \*\*\*\*\*

\*ALU TEST

\*TEST OF ALU FUNCTION A AND B WITH C BIT SET

\*ALU FUNCTION (A AND B) CODE=13

\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA

\*PERFORM THE FUNCTION, VERIFY THE RESULTS

\*\*\*\*\*

\*\*\*\*\* TEST 65 \*\*\*\*\*

\*ALU TEST

\*TEST OF ALU FUNCTION A OR B WITH C BIT SET

\*ALU FUNCTION (A OR B) CODE=14

\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA

\*PERFORM THE FUNCTION, VERIFY THE RESULTS

\*\*\*\*\*

\*\*\*\*\* TEST 66 \*\*\*\*\*

\*ALU TEST

\*TEST OF ALU FUNCTION A XOR B WITH C BIT SET

\*ALU FUNCTION (A XOR B) CODE=15

\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA

\*PERFORM THE FUNCTION, VERIFY THE RESULTS

\*\*\*\*\*

\*\*\*\*\* TEST 67 \*\*\*\*\*

\*ALU TEST

CZDMPA M8207 STATIC DIAG. #1  
CZDMPA.P11 17-JUL-79 14:33

MACY11 30A(1052) 17-JUL-79 14:39 PAGE 25-1  
PROGRAM DOCUMENT

M 3

SEQ 0038

4644

\*TEST OF ALU FUNCTION ADD WITH C BIT SET

4646  
4647  
4648  
4649  
4650  
4651  
4652  
4653  
4654  
4655  
4656  
4657  
4658  
4659  
4660  
4661  
4662  
4663  
4664  
4665  
4666  
4667  
4668  
4669  
4670  
4671  
4672  
4673  
4674  
4675  
4676  
4677  
4678  
4679  
4680  
4681  
4682  
4683  
4684  
4685  
4686  
4687  
4688  
4689  
4690  
4691  
4692  
4693  
4694  
4695  
4696  
4697  
4698  
4699  
4700  
4701

\*ALU FUNCTION (A PLUS B) CODE=00  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 68 \*\*\*\*\*  
\*ALU TEST  
\*TEST OF ALU FUNCTION 2A W/C WITH C BIT SET  
\*ALU FUNCTION (A PLUS A PLUS C) CODE=6  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 69 \*\*\*\*\*  
\*ALU TEST  
\*TEST OF ALU FUNCTION SUB WITH C BIT SET  
\*ALU FUNCTION (A-B) CODE=16  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 70 \*\*\*\*\*  
\*ALU TEST  
\*TEST OF ALU FUNCTION ADD W/C WITH C BIT SET  
\*ALU FUNCTION (A PLUS B PLUS C) CODE=01  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 71 \*\*\*\*\*  
\*ALU TEST  
\*TEST OF ALU FUNCTION SUB W/C WITH C BIT SET  
\*ALU FUNCTION (A-B-C) CODE=2  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 72 \*\*\*\*\*  
\*ALU TEST  
\*TEST OF ALU FUNCTION INC A WITH C BIT SET  
\*ALU FUNCTION (A PLUS 1) CODE=3  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

CZDMPA M8207 STATIC DIAG. #1  
CZDMPA.P11 17-JUL-79 14:33

MACY11 30A(1052) 17-JUL-79 14:39 <sup>B 4</sup> PAGE 26-1  
PROGRAM DOCUMENT

SEQ 0040

4702

\*\*\*\*\* TEST 73 \*\*\*\*\*

4704  
4705  
4706  
4707  
4708  
4709  
4710  
4711  
4712  
4713  
4714  
4715  
4716  
4717  
4718  
4719  
4720  
4721  
4722  
4723  
4724  
4725  
4726  
4727  
4728  
4729  
4730  
4731  
4732  
4733  
4734  
4735  
4736  
4737  
4738  
4739  
4740  
4741  
4742  
4743  
4744

\*ALU TEST  
\*TEST OF ALU FUNCTION 2A WITH C BIT SET  
\*ALU FUNCTION (A PLUS A) CODE=5  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 74 \*\*\*\*\*  
\*ALU TEST  
\*TEST OF ALU FUNCTION A PLUS C WITH C BIT SET  
\*ALU FUNCTION (A PLUS C) CODE=4  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 75 \*\*\*\*\*  
\*ALU TEST  
\*TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT SET  
\*ALU FUNCTION (A-B-1) CODE=17  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
\*\*\*\*\*

\*\*\*\*\* TEST 76 \*\*\*\*\*  
\*ALU TEST  
\*TEST OF ALU FUNCTION DEC A WITH C BIT SET  
\*ALU FUNCTION (A-1) CODE=7  
\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
\*PERFORM THE FUNCTION, VERIFY THE RESULT  
\*\*\*\*\*

4746  
4747  
4748  
4749  
4750  
4751  
4752  
4753  
4754  
4755  
4756  
4757  
4758  
4759  
4760  
4761  
4762  
4763  
4764  
4765  
4766  
4767  
4768  
4769  
4770  
4771  
4772  
4773  
4774  
4775  
4776  
4777  
4778  
4779  
4780  
4781  
4782  
4783

## 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 EXAMPLE PROVIDES A TYPICAL ERROR REPORT.

CZDMP DVC FTL ERR 00003 TST 029 SUB 000 PC:02<626

BR REGISTER DATA TEST  
UNIT=00; FAILING UNIT ADDRESS=160170

GOOD	BAD
177776	000011

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

Q



CZDMPA M8207 STATIC DIAG. #1 MACY11 30A(1052) 17-JUL-79 14:39 E 4  
CZDMPA.P11 17-JUL-79 14:33 PROGRAM DOCUMENT PAGE 29

SEQ 0043

4785  
4786  
4787  
4788  
4789

1

```
4791          .TITLE CZDMPAO M8207 STATIC DIAG #1
4799          .=2000
4800
4801
4802
4803
4804
4805
4806          .MCALI SVC
4807 002000    SVC                      ; INITIALIZE SUPERVISOR MACROS
4808
4809
4810
4811
4812
4813 002000    BGNMOD CZDMP
4814
4815
4816          000000 $LSTIN= 0
4817          000000 $LSTTAG 0
4818          000000 SVCINS= 0      ; LIST INSTRUCTIONS, SHIFTED RIGHT
4819          000000 SVCTST= 0     ; LIST TEST TAGS, SHIFTED RIGHT
4820          000000 SVCSUB= 0     ; LIST SUBTEST TAGS, SHIFTED RIGHT
4821          000000 SVCGBL= 0     ; LIST GLOBAL TAGS, SHIFTED RIGHT
4822          000000 SVCTAG= 0     ; LIST OTHER TAGS, SHIFTED RIGHT
4823
4824          ; CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
4825          ; TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS. CHANGE THE
4826          ; SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS. YOU MAY
4827          ; CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
4828
4829
4830          .ENABL AMA
```

```

4832      .SBTTL  PROGRAM HEADER
4833      :++
4834      : THE PROGRAM HEADER IS THE INTERFACE BETWEEN
4835      : THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
4836      :--
4837
4838      002000      POINTER BGNAU,BGNDU
4839
4847
4848      002000      HEADER  CZDMP,A,0,120,,0
      (4) 002000      L$NAME::      ;DIAGNOSTIC NAME
      (4) 002000      103      .ASCII /C/
      (4) 002001      132      .ASCII /Z/
      (4) 002002      104      .ASCII /D/
      (4) 002003      115      .ASCII /M/
      (4) 002004      120      .ASCII /P/
      (6) 002005      000      .BYTE 0
      (6) 002006      000      .BYTE 0
      (5) 002007      000      .BYTE 0
      (5) 002010      L$REV::      ;REVISION LEVEL
      (4) 002010      101      .ASCII /A/
      (5) 002011      L$DEPO::      ;0
      (4) 002011      060      .ASCII /O/
      (5) 002012      L$UNIT::      ;NUMBER OF UNITS
      (4) 002012      000000      .WORD 0
      (5) 002014      L$TIML::      ;LONGEST TEST TIME
      (4) 002014      000170      .WORD 120.
      (5) 002016      L$HPCP::      ;POINTER TO H.W. QUES.
      (4) 002016      034704      .WORD L$HARD
      (5) 002020      L$SPCP::      ;POINTER TO S.W. QUES.
      (4) 002020      000000      .WORD 0
      (5) 002022      L$HPTP::      ;PTR. TO DEF. H.W. PTABLE
      (4) 002022      002364      .WORD L$HW
      (5) 002024      L$SPTP::      ;PTR. TO S.W. PTABLE
      (4) 002024      000000      .WORD 0
      (5) 002026      L$LADP::      ;DIAG. END ADDRESS
      (4) 002026      040004      .WORD L$LAST
      (5) 002030      L$STA::      ;RESERVED FOR APT STATS
      (4) 002030      000000      .WORD 0
      (5) 002032      L$CO::      .WORD 0
      (4) 002032      000000      .WORD 0
      (5) 002034      L$DTYP::      ;DIAGNOSTIC TYPE
      (4) 002034      000000      .WORD 0
      (5) 002036      L$APT::      ;APT EXPANSION
      (4) 002036      000000      .WORD 0
      (5) 002040      L$DTP::      ;PTR. TO DISPATCH TABLE
      (4) 002040      002132      .WORD L$DISPATCH
      (5) 002042      L$EXP1::      ;EXPANSION WORDS
      (4) 002042      000000      .WORD 0
      (5) 002044      L$EXP2::      .WORD 0
      (4) 002044      000000      .WORD 0
      (5) 002046      L$EXP3::      .WORD 0
      (4) 002046      000000      .WORD 0
      (5) 002050      L$MREV::      ;SVC REV AND EDIT #
      (4) 002050      003      .BYTE C$REVISION
      (3) 002051      000      .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	003130		.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	011350		.WORD L\$AU	
(5)	002072		L\$DUT::		;PTR. TO DROP UNIT CODE
(4)	002072	011344		.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	002414		.WORD L\$DESC	
(5)	002100		L\$LOAD::		;GENERATE SPECIAL AUTOLOAD EMT
(4)	002100	104035		EMT E\$LOAD	
(5)	002102		L\$ETP::		;PTR. TO ERRTABL
(4)	002102	000000		.WORD 0	
(5)	002104		L\$ICP::		;PTR. TO INIT CODE
(4)	002104	010554		.WORD L\$INIT	
(5)	002106		L\$CCP::		;PTR. TO CLEAN-UP CODE
(4)	002106	011340		.WORD L\$CLEAN	
(5)	002110		L\$ACP::		;PTR. TO AUTO CODE
(4)	002110	011242		.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	
4849					
4850					
4856					
4857	002122			BGNPROT	
(3)	002122		L\$PROT::		
4858	002122	177777		.WORD -1	
4859	002124	177777		.WORD -1	
4860	002126	177777		.WORD -1	
4861	002130			ENDPROT	
4862					

4864  
4865  
4866  
4867  
4868  
4869  
4870  
4871

.SBTTL DISPATCH TABLE

:/ THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.  
:/ IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.

DISPATCH 76.

LSDISPATCH: .WORD 76  
.WORD T1  
.WORD T2  
.WORD T3  
.WORD T4  
.WORD T5  
.WORD T6  
.WORD T7  
.WORD T8  
.WORD T9  
.WORD T10  
.WORD T11  
.WORD T12  
.WORD T13  
.WORD T14  
.WORD T15  
.WORD T16  
.WORD T17  
.WORD T18  
.WORD T19  
.WORD T20  
.WORD T21  
.WORD T22  
.WORD T23  
.WORD T24  
.WORD T25  
.WORD T26  
.WORD T27  
.WORD T28  
.WORD T29  
.WORD T30  
.WORD T31  
.WORD T32  
.WORD T33  
.WORD T34  
.WORD T35  
.WORD T36  
.WORD T37  
.WORD T38  
.WORD T39  
.WORD T40  
.WORD T41  
.WORD T42  
.WORD T43  
.WORD T44  
.WORD T45  
.WORD T46

002130  
(4) 002130 000114  
(3) 002132  
(6) 002132 011352  
(6) 002134 011500  
(6) 002136 011544  
(6) 002140 011732  
(6) 002142 012076  
(6) 002144 012232  
(6) 002146 012362  
(6) 002150 012512  
(6) 002152 012654  
(6) 002154 013040  
(6) 002156 013224  
(6) 002160 013412  
(6) 002162 013562  
(6) 002164 013670  
(6) 002166 014120  
(6) 002170 014350  
(6) 002172 014600  
(6) 002174 015030  
(6) 002176 015314  
(6) 002200 015610  
(6) 002202 016040  
(6) 002204 016270  
(6) 002206 016520  
(6) 002210 016750  
(6) 002212 017200  
(6) 002214 017430  
(6) 002216 017660  
(6) 002220 020110  
(6) 002222 020406  
(6) 002224 020726  
(6) 002226 021272  
(6) 002230 021604  
(6) 002232 021746  
(6) 002234 022110  
(6) 002236 022264  
(6) 002240 022470  
(6) 002242 022634  
(6) 002244 023004  
(6) 002246 023150  
(6) 002250 023336  
(6) 002252 023552  
(6) 002254 023770  
(6) 002256 024122  
(6) 002260 024340  
(6) 002262 024500  
(6) 002264 024704

(6)	002266	025110	.WORD	T47
(6)	002270	025314	.WORD	T48
(6)	002272	025520	.WORD	T49
(6)	002274	025724	.WORD	T50
(6)	002276	026130	.WORD	T51
(6)	002300	026334	.WORD	T52
(6)	002302	026540	.WORD	T53
(6)	002304	026746	.WORD	T54
(6)	002306	027152	.WORD	T55
(6)	002310	027356	.WORD	T56
(6)	002312	027562	.WORD	T57
(6)	002314	027766	.WORD	T58
(6)	002316	030172	.WORD	T59
(6)	002320	030376	.WORD	T60
(6)	002322	030602	.WORD	T61
(6)	002324	031006	.WORD	T62
(6)	002326	031212	.WORD	T63
(6)	002330	031416	.WORD	T64
(6)	002332	031622	.WORD	T65
(6)	002334	032026	.WORD	T66
(6)	002336	032232	.WORD	T67
(6)	002340	032436	.WORD	T68
(6)	002342	032642	.WORD	T69
(6)	002344	033046	.WORD	T70
(6)	002346	033252	.WORD	T71
(6)	002350	033456	.WORD	T72
(6)	002352	033662	.WORD	T73
(6)	002354	034066	.WORD	T74
(6)	002356	034272	.WORD	T75
(6)	002360	034476	.WORD	T76

:LNT.ED DIFINED AT END OF PROGRAM TO BE LAST TEST NUMBER.

4872  
4873  
4880  
4881  
4882  
4883  
4884

4886  
4887  
4888  
4889  
4890  
4891  
4892  
4893  
4894  
4895  
4896  
4897  
4898  
4899  
4900  
4901  
4902  
4903  
4904  
4905  
4906  
4907  
4908  
4909  
4910  
4911  
4912  
4913  
4914

002362  
(3) 002362 000013  
(3) 002364  
(3) 002364  
002364 000007  
002366 160170  
002370 000300  
002372 005000  
002374 000003  
002376 000056  
002400 000000  
002402 000000  
002404 000000  
002406 000004  
002410 000000  
002412  
(3) 002412

.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.

BGNHW DFPTBL  
.WORD L10001-L\$HW/2  
L\$HW::  
DFPTBL::

.WORD 7 : MICRO CPU TYPE  
.WORD 160170 : M8200.4,7 CSR 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  
.WORD 0  
ENDHW  
L10001:

```
4916      .SBTTL  SOFTWARE P-TABLE
4917
4918      ://////////
4919      :// THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
4920      :// PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
4921      ://////////
4922
4923      002412      BGNSW  SFPTBL
      (3) 002412 000000      .WORD  L10002-L$SW/2
      (3) 002414
      (3) 002414      L$SW::
      SFPTBL::
4924
4925
4926      002414      ENDSW
      (3) 002414      L10002:
4927
4928
4929
4930
4931
4932
```



4934  
 4935  
 4936  
 4937  
 4938  
 4939  
 4940  
 4941  
 4942  
 4943  
 4944  
 4945  
 4946  
 4947  
 4948  
 4949  
 4950  
 4951  
 4952  
 4953

002414

.SBTTL GLOBAL EQUATES SECTION

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

EQUALS

: BIT DIFINITIONS

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)  
 (1)  
 (1)  
 (1) 100000  
 (1) 040000  
 (1) 020000  
 (1) 010000  
 (1) 004000  
 (1) 002000  
 (1) 001000  
 (1) 000400  
 (1) 000200  
 (1) 000100  
 (1) 000040  
 (1) 000020  
 (1) 000010  
 (1) 000004  
 (1) 000002  
 (1) 000001  
 (1)  
 (1) 001000  
 (1) 000400  
 (1) 000200  
 (1) 000100  
 (1) 000040  
 (1) 000020  
 (1) 000010  
 (1) 000004  
 (1) 000002  
 (1) 000001  
 (1)  
 (1)  
 (1)  
 (1) 000040  
 (1) 000037

```

(1)      000036      EF.CONTINUE== 30.      ; CONTINUE COMMAND WAS ISSUED
(1)      000035      EF.NEW== 29.           ; A NEW PASS HAS BEEN STARTED
(1)      000034      EF.PWR== 28.           ; A POWER-FAIL/POWER-UP OCCURRED
(1)      :
(1)      :
(1)      : PRIORITY LEVEL DEFINITIONS
(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      HGE== 100000
4954
4955
4956
4957
4958
4959      :*****
4960      :* INSTRUCTION DEFINITIONS
4961      :*****
4962      022626      POP2SP=22626           ; INCREMENT STACK TWICE
4963
4964
4965
4966      :*****
4967      :* PROGRAM EVENT FLAG DEFINITIONS
4968      :*****
4969
4970
4971
  
```

```

4973      .SBTTL  GLOBAL DATA SECTION
4974
4975      ;//////////
4976      ;/      THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
4977      ;/      IN MORE THAN ONE TEST.
4978      ;//////////
4979
4980      ;*****
4981      ;* STORAGE FOR DEVICE REGISTERS
4982      ;*****
4983      DESCRIPT      <M8207 DIAG. #1 OF 2>
      L$DESC::
      .ASCIZ  /M8207 DIAG. #1 OF 2/
4984
      .EVEN
4985      ;*****
4986      ;* PROGRAM CONTROL PARAMETERS
4987      ;*****
4988      002440  000000      NEXT:  .WORD  0      ;ADDRESS OF NEXT TEST TO BE EXECUTED
4989      002442  000000      LOCK:   .WORD  0      ;ADDRESS FOR LOCK CURRENT DATA
4990
4991      ;*****
4992      ;* BUFFERS FOR INPUT-OUTPUT
4993      ;*****
4994      002444  000000      TEMP:   0
4995      002506      .=. +40
4996      002506  000000      MDATA:  0
4997      002550      .=. +40
4998
4999      ;*****
5000      ;* MISCELLANEOUS STORAGE
5001      ;*****
5002      002550  000000      $TMP0:  .WORD  0      ;SCRATCH STORAGE
5003      002552  000000      LOGDEV: .WORD  0      ;LOGICAL DEVICE NUMBER
5004      002554  000000      PSTACK: .WORD  0      ;BASE LEVEL PROGRAM STACK POINTER
5005      002556  000000      SUBRPC:  .WORD  0      ;PC OF SUBR CALL FOR ERROR REPORTS
5006      002560  000000      ERRFLG: .WORD  0      ;SUBROUTINE ERROR FLAG
5007      002562  000000      RETADR:  .WORD  0      ;SUBR ERROR RETURN ADDRESS
5008      002564  000000      STRTSW:  .WORD  0      ;SWITCHES AT START OF PROGRAM
5009      002566  000000      STAT:    .WORD  0      ;M8200,4,7 STATUS WORD STORAGE
5010      002570  000000      CLKX:    .WORD  0
5011      002572  000000      MASKX:   .WORD  0
5012      002574  000000      SAVSP:   .WORD  0      ;STACK POINTER STORAGE
5013      002576  000000      SAVPC:   .WORD  0      ;PROGRAM COUNTER STORAGE
5014      002600  000000      ZERO:    .WORD  0
5015      002602  000001      ONE:     .WORD  1
5016      002604  037776      MEMLIM:  .WORD  MEMEND      ;HIGHEST LOCATION FOR NFR'S
5017      002606
5018      002606  000000      MEMSZ:
5019      002610  000001      KMACTV:  .BLKW  1      ;M8200,4,7 SELECTED ACTIVE
5020      002612  000001      KMNUM:   .BLKW  1      ;OCTAL NUMBER OF M8200,4,7
5021      002614  000001      SAVACT:  .BLKW  1      ;ORIGINAL ACTIVE DEVICES
5022      002616  000001      SAVNUM:  .BLKW  1      ;WORKABLE NUMBER
  
```

5023 002620 000000  
5024 002622 000000  
5025 002624 000000  
5026 002626 000000  
5027 002630 000000  
5028 002632 000000  
5029 002634 000000  
5030 002636 000000  
5031 002640 000000  
5032 002642 000000  
5033 002644 000000  
5034 002646 000000  
5035 002650 000000  
5036 002652 000000

FLAG: .WORD 0 ;SCRATCH STORAGE  
RUN: .WORD 0 ;POINTER TO RUNNING DEVICES  
MRO: .WORD 0  
WTYPE: .WORD 0  
TYPE: .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

5037  
5038  
5039  
5040  
5041 002654 000 377 000  
          002657 377 125 252  
          002662 125 252  
5042 002664 000 000 377  
          002667 377 125 125  
          002672 252 252

\*\*\*\*\*  
;\* DATA PATTERNS  
\*\*\*\*\*  
MEMDAT: .BYTE 0,-1,0,-1,125,252,125,252

SPDAT: .BYTE 0,0,-1,-1,125,125,252,252

.EVEN

5043  
5044  
5045  
5046  
5047  
5048 002674 000  
5049 002676 000  
5050 002676 000  
5051 002677 000

\*\*\*\*\*  
;\* 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

5052  
5053  
5054  
5055  
5056  
5057  
5058  
5059  
5060  
5061  
5062  
5063  
5064  
5065  
5066  
5067  
5068  
5069  
5070  
5071  
5072  
5073 002700 000000  
5074 002702 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  
\*\*\*\*\*  
STAT1: .WORD 0  
STAT2: .WORD 0

```

5075 002704 000000      STAT3: .WORD 0
5076
5077
5078 ;*****
5079 ;* POINTERS TO M8200,4,7 VECTORS AND REGISTERS
5080 ;*****
5080 002706 000000      KMRVEC: 0 ;POINTER TO M8200,4,7 RCV INTRPT VECTOR
5081 002710 000000      KMRLVL: 0 ;POINTER TO M8200,4,7 RCV INTRPT SERVICE PS
5082 002712 000000      KMTVEC: 0 ;POINTER TO M8200,4,7 TX INTRPT VECTOR
5083 002714 000000      KMTLVL: 0 ;POINTER TO M8200,4,7 TX INTRPT SERVICE PS
5084 002716 000000      KMCSR: 0 ;POINTER TO M8200,4,7 CONTROL STATUS REGISTER
5085 002720 000000      KMCSRH: 0 ;POINTER TO M8200,4,7 CONTROL STATUS REGISTER HIGH BYTE
5086 002722 000000      KMCTL: 0 ;POINTER TO M8200,4,7 CONTROL OUT REGISTER
5087 002724 000000      KMP04: 0
5088 002726 000000      KMP06: 0 ;POINTER TO M8200,4,7 PORT REGISTER - SEL6
5089
5090 ;:***** PRIMARY REG ADRS STORAGE FOR THIS UNIT *****
5091 ;THESE LOCATIONS WILL BE LOADED FOR THE CURRENT UNIT, IN INIT CODE
5092 002730      REGADR:
5093
5094 ;:***** STACK USED FOR SUBROUTINE LINKAGE *****
5095 002730 000100      .BLKW 100
5096 003130
5097
5098
5099
5100
5101
5102
5103

```

```
5105      .SBTTL  GLOBAL TEXT SECTION
5106
5107      ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
5108      ;%      THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
5109      ;%      MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
5110      ;%      MORE THAN ONE TEST.
5111      ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
5112
5113      ;*****
5114      ;* NAMES OF DEVICES SUPPORTED BY PROGRAM
5115      ;*****
5116      003130      DEVTYP  <M8200,4,7>
5117      (4) 003130
5118      (3) 003130 034115 030062 026060
5119      (3) 003136 026064 000067
5120      (2)
5121      .EVEN
5122
5123      ;
5124      ; FORMAT STATEMENTS USED IN PRINT CALLS
5125      ;
5126
5127
5128
5129
5130
5131
5132
5133
```

5135  
 5136  
 5137  
 5138  
 5139  
 5140  
 5141  
 5142  
 5143  
 5144  
 5145  
 5146  
 5147  
 5148  
 5149  
 5150  
 5151  
 5152  
 5153  
 5154  
 5155  
 5156  
 5157  
 5158  
 5159  
 5160  
 5161  
 5162  
 5163  
 5164  
 5165  
 5166  
 5167  
 5168  
 5169  
 5170  
 5171  
 5172  
 5173  
 5174  
 5175  
 5176  
 5177  
 5178  
 5179  
 5180  
 5181  
 5182  
 5183  
 5184  
 5185  
 5186  
 5187  
 5188  
 5189  
 5190

```
.SBTTL GLOBAL SUBROUTINES

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

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

.MACRO ERROR,XYX,ZZ
MOV R4,$BDDAT
.IF B ZZ
MOV R2,$GDDAT
.ENDC
MOV MRO,$BDADR
ERRDF XYX',EM'XYX',ERR'XYX'
.ENDM
.MACRO RERROR XXX
MOV R4,$BDDAT
CLRB $BDDAT+1
CLRB $GDDAT+1
MOV R2,$GDADR
ERRDF XXX',EM'XXX',ERR'XXX'
.ENDM
.MACRO BERROR XXX
MOV R4,$BDDAT
MOV R5,$GDDAT
CLRB $BDDAT+1
CLRB $GDDAT+1
ERRDF XXX',EM'XXX',ERR'XXX'
.ENDM
.MACRO ED$CALL XY
.LIST
:***** TEST 'XY' *****
.NLIST
.ENDM
.MACRO BADHEAD
.RADIX 10
ED$CALL \T$TESTNUM+1
.RADIX 8
.ENDM
.MACRO K4ONLY ?N2
.LIST
:DON'T DO TEST IF M8200 OR M8204
.NLIST
CMP MEMSZ,#2000
BNE N2
EXIT TST
N2:
.ENDM
.MACRO MYINT
.LIST
MOV KMCSR,R1 ;GET DEVICE ADDRESS.
```

```

5191      .NLIST
5192      .ENDM
5193
5194      .MACRO ROMCLK
5195      .LIST
5196      JSR      R5,,ROMCLK      ;CLOCK INSTRUCTION
5197      .NLIST
5198      .ENDM
5199
5200      .MACRO MSTCLR
5201      .LIST
5202      JSR      R5,,MSTCLR      ;CLEAR M8200,4,7
5203      .NLIST
5204      .ENDM
5205
5206      003142      .MSTCLR:
5207      003142      112777      000100      177550      MOVB      #BIT6,@KMCSRH      ;SET INST.
5208      003150      142777      000300      177542      BICB      #BIT6!BIT7,@KMCSRH
5209      003156      000205      RTS      R5
5210
5211      ;
5212      003160      000024      .BLKW      20.      ;PATCH AREA.
5213
5214
5215
5216      003230      ENDBG:
5217      ;      UNSAFE TO PATCH ANY OTHER AREA.
5218
5219
5220
5221      003230      .ROMCLK:
5222      003230      152777      000002      177462      BISB      #BIT1,@KMCSRH
5223      003236      012577      177464      MOV      (R5)+,@KMP06
5224      003242      152777      000003      177450      BISB      #BIT1!BIT0,@KMCSRH
5225      003250      142777      000007      177442      BICB      #BIT2!BIT1!BIT0,@KMCSRH
5226      003256      000205      RTS      R5
5227
5228      003260      CLRALL:
5229      ;CLEAR C & Z BITS AND BR
5230      003260      ROMCLK
5231      (1) 003260      004537      003230      JSR      R5,,ROMCLK      ;CLOCK INSTRUCTION
5232      003264      000400      4J0      ;0 TO BR
5233      (1) 003266      004537      003230      ROMCLK
5234      003272      063220      JSR      R5,,ROMCLK      ;CLOCK INSTRUCTION
5235      003274      004537      003230      63220      ;SP(0) TO BR
5236      003300      060400      JSR      R5,,ROMCLK      ;CLOCK INSTRUCTION
5237      003302      000207      60400      ;BR,SP(0) + BR
5238      003304      RTS      PC
5239
5240      003304      SETBR0:
5241      (1) 003304      004537      003230      ;SETS BR0 BIT
5242      003310      000401      ROMCLK
5243      003312      000207      JSR      R5,,ROMCLK      ;CLOCK INSTRUCTION
5244      401      ;1 TO BR
5245      RTS      PC
  
```



```

5243
5244 003314      SETBR1:
5245              ;THIS SUBROUTINE SETS BR1 BIT
5246
5247 003314      ROMCLK      ;NEXT WORD IS INSTRUCTION
5248 (1) 003314 004537 003230      JSR      R5,,ROMCLK      ;CLOCK INSTRUCTION
5249 003320 000402      000402      ;BR_002
5250 003322 000207      RTS      PC
5251
5252 003324      SETBR4:
5253              ;THIS SUBROUTINE SETS BR4 BIT
5254
5255 003324      ROMCLK      ;NEXT WORD IS INSTRUCTION
5256 (1) 003324 004537 003230      JSR      R5,,ROMCLK      ;CLOCK INSTRUCTION
5257 003330 000402      402
5258 003332 000207      RTS      PC
5259
5260 003334      SETBR7:
5261              ;THIS SUBROUTINE SETS BR7 BIT
5262
5263 003334      ROMCLK      ;NEXT WORD IS INSTRUCTION
5264 (1) 003334 004537 003230      JSR      R5,,ROMCLK      ;CLOCK INSTRUCTION
5265 003340 000600      600
5266 003342 000207      RTS      PC
5267
5268 003344      SETZ:
5269              ;THIS SUBROUTINE SETS THE Z BIT
5270
5271 003344      ROMCLK      ;NEXT WORD IS INSTRUCTION
5272 (1) 003344 004537 003230      JSR      R5,,ROMCLK      ;CLOCK INSTRUCTION
5273 003350 000777      000777      ;BR_377
5274 003352 000207      RTS      PC
5275
5276 003354      RAMDAT:
5277              ;THIS SUBROUTINE LOADS R4 WITH THE LOWEST
5278              ;8 BITS OF THE CRAM PC.
5279
5280 003354 017605 000000      MOV      @ (SP),R5      ;GOOD DATA
5281 003360 062716 000002      ADD      #2,(SP)      ;ADJUST STACK
5282 003364 005011      CLR      (R1)      ;CLEAR BIT10
5283 003366 052711 000400      BIS      #BIT8,(R1)      ;CLOCK INSTRUCTION IN CRAM THAT
5284              ;JUMPED TO, IT LOADS BR WITH IT
5285
5286 003372 005011      CLR      (R1)      ;CLR BIT8
5287 003374      ROMCLK      ;NEXT WORD IS INSTRUCTION
5288 (1) 003374 004537 003230      JSR      R5,,ROMCLK      ;CLOCK INSTRUCTION
5289 003400 061225      061225      ;MOV BR TO PORT 5
5290 003402 116104 000005      MOVB     5(R1),R4      ;PUT 'FOUND' IN R4
5291 003406 000207      RTS      PC      ;RETURN
5292
5293
5294      MEMSET:
5295              ;THIS SUBROUTINE LOADS CRAM WITH SPECIAL INSTRUCTIONS
5296              ;FOR THE CRAM JUMP TEST. ALL CRAM LOCATIONS ARE LOADED
5297              ;WITH INSTRUCTIONS THAT MOVE A 37 TO THE BR, EXCEPT THE
5298              ;FOLLOWING CRAM ADDRESSES: 0,1,4,7,525,1777. THESE LOCATIONS

```

```

5294                                     ;CONTAIN INSTRUCTIONS WHICH LOAD THE BR WITH THE LOWEST
5295                                     ;8 BITS OF THAT CRAM ADDRESS.
5296
5297 003410 005000                                     CLR    R0                ;R0 = CRAM ADDRESS
5298 003412 012711 002000 1$:  MOV    #BIT10,(R1)          ;SET ROMO
5299 003416 010061 000004  MOV    R0,4(R1)              ;LOAD CRAM ADDRESS
5300 003422 012761 000437 000006  MOV    #437,6(R1)         ;LOAD INSTRUCTION
5301 003430 052711 020000  BIS    #BIT13,(R1)          ;WRITE INSTRUCTION IN CRAM
5302 003434 005200  INC    R0                ;NEXT ADDRESS
5303 003436 022700 002000  CMP    #2000,R0             ;DONE YET?
5304 003442 001363  BNE    1$                    ;BR IF NO
5305 003444 005000  CLR    R0                ;INDEX REGISTER
5306 003446 012711 002000 2$:  MOV    #BIT10,(R1)          ;SET ROMO
5307 003452 016061 003506 000004  MOV    CRAMA(R0),4(R1)    ;LOAD CRAM ADDRESS IN SEL4
5308 003460 016061 003522 000006  MOV    INSTU(R0),6(R1)    ;LOAD INSTRUCTION TO BE WRITTEN
5309 003466 052711 020000  BIS    #BIT13,(R1)          ;WRITE CRAM!
5310 003472 005720  TST    (R0)+                ;NEXT
5311 003474 022700 000014  CMP    #14,R0             ;DONE YET?
5312 003500 001362  BNE    2$                    ;BR IF NO
5313 003502 005011  CLR    (R1)                ;CLEAR ALL BITS
5314 003504 000207  RTS    PC                    ;RETURN
5315
5316 003506 000000 000001 000004 CRAMA: .WORD 0,1,4,7,1777,525
5317 003514 000007 001777 000525
5318 003522 000400  INSTU: 000400                ;BR_0
5319 003524 000401  000401                ;BR_1
5320 003526 000404  000404                ;BR_4
5321 003530 000407  000407                ;BR_7
5322 003532 000777  000777                ;BR_377
5323 003534 000525  000525                ;BR_125
5324
5325 003536  SETVEC:
5326                                     ;THIS SUBROUTINE LOADS THE VECTORS AND VECTOR LEVELS
5327
5328 003536 012577 177144  MOV    (R5)+,@KMRVEC      ;LOAD BASE VECTOR
5329 003542 012577 177144  MOV    (R5)+,@KMTVEC      ;LOAD VECTOR + 2
5330 003546 012577 177136  MOV    (R5)+,@KMRLVL      ;LOAD VECTOR + 4
5331 003552 012577 177136  MOV    (R5)+,@KMTLVL      ;LOAD VECTOR + 6
5332 003556 000205  RTS    R5                    ;RETURN
5333
5334 003560  NPRSET:
5335                                     ;THIS SUBROUTINE LOADS IBUS REGISTERS 0-7
5336                                     ;WITH NPR INFORMATION (INBA, OUTBA, OUT DATA)
5337
5338 003560 010246  MOV    R2,-(SP)                ;SAVE R2
5339 003562 005002  CLR    R2                    ;START AT IBUS REG 0
5340 003564 112561 000004 1$:  MOV    (R5)+,4(R1)          ;LOAD PORT4
5341 003570 042737 000017 003606  BIC    #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
5342 003576 050237 003606  BIS    R2,2$            ;ADD ADDRESS TO INSTRUCTION
5343 003602  ROMCLK
5344 003602 004537 003230  JSR    R5,ROMCLK          ;CLOCK INSTRUCTION
5345 003606 122100 2$:  122100                ;MOVE PORT4 TO IBUS REG
5346 003610 005202  INC    R2                    ;NEXT ADDRESS
5347 003612 022702 000010  CMP    #10,R2          ;ALL DONE?

```

```

5348 003616 001362      BNE      1$          ;BR IF NO
5349 003620 012602      MOV      (SP)+,R2      ;RESTORE R2
5350 003622 000205      RTS       R5          ;RETURN
5351
5352
5353 003624      MEMLD:      ;THIS SUBROUTINE LOADS THE FIRST 8 LOCATIONS OF MAIN
5354                      ;MEMORY WITH THIS DATA: 0,-1,,0,-1,125,252,125,252
5355
5356
5357 003624 013637 002550      MOV      @ (SP)+,$TMP0      ;PUT POINTER TO DATA IN R0
5358 003630 062746 000002      ADD      #2,-(SP)      ;ADJUST STACK
5359
5360 003634 013700 002550      MEMLD2: MOV      $TMP0,R0      ;GET ADDR.
5361 003640 012704 000010      MOV      #10,R4      ;DO 8 LOADS
5362 003644      ROMCLK
5363 (1) 003644 004537 003230      JSR      R5,.,ROMCLK      ;CLOCK INSTRUCTION
5364 003650 010000      010000      ;MAR < 0
5365 (1) 003652 004537 003230      ROMCLK      ;CLR      MAR HI
5366 003656 004000      JSR      R5,.,ROMCLK      ;CLOCK INSTRUCTION
5367 003660 112077 177040      1$:      MOV      (R0)+,@KMP04      ;LOAD PORT4
5368 (1) 003664 004537 003230      ROMCLK
5369 003670 136500      JSR      R5,.,ROMCLK      ;CLOCK INSTRUCTION
5370 003672 005304      136500      ;MOV DATA TO MEM, AUTO INC MAR
5371 003674 001371      DEC      R4      ;DECREMENT COUNT
5372      BNE      1$      ;BR IF NOT DONE
5373
5374 003676      ROMCLK      ;LOAD MEM ADDR. 0
5375 (1) 003676 004537 003230      JSR      R5,.,ROMCLK      ;CLOCK INSTRUCTION
5376 003702 010000      10000
5377 003704 012703 000010      MOV      #10,R3      ;CHECK 8. MEM LOCS.
5378 003710 013700 002550      MOV      $TMP0,R0
5379 (1) 003714 004537 003230      2$:      ROMCLK      ;READ FROM MEM,PUT INTO PORT 4
5380 003720 055224      JSR      R5,.,ROMCLK      ;CLOCK INSTRUCTION
5381 055224
5382
5383 003722 112037 002636      MOV      (R0)+,$GDDAT      ;EXPECTED.
5384 003726 117704 176772      MOV      @KMP04,R4      ;RECIEVED.
5385 003732 123704 002636      CMP      $GDDAT,R4      ;OK?
5386 003736 001414      BEQ      3$
5387 003740      ERROR      36
5388 (5) 003756 104455      TRAP      C$ERDF
5389 (6) 003760 000044      .WORD      36
5390 (6) 003762 005624      .WORD      EM36
5391 (6) 003764 010416      .WORD      ERR36
5392 003766 000402      BR      4$
5393 003770 005303      3$:      DEC      R3      ;CHECKED ALL?
5394 003772 001350      BNE      2$      ;NO-DO NEXT ONE.
5395 003774      4$:
5396 000207      RTS      PC      ;RETURN
5397
5398
5399 003776      SPLD:      ;THIS SUBROUTINE LOADS THE FIRST 8 SCRATCH PAD
5400                      ;LOCATIONS WITH: 0,0,-1,-1,125,125,252,252

```

Address	Hex	Dec	Label	Instruction	Comment
5397	003776	013600		MOV @ (SP)+, R0	; PUT POINTER TO DATA IN R5
5398	004000	062746	000002	ADD #2, -(SP)	; ADJUST STACK
5399	004004	005004		CLR R4	; START AT SP ADDRESS 0
5400	004006	112077	176712	MOVB (R0)+, @KMP04	; LOAD PORT4 WITH DATA
5401	004012	042737	000017	BIC #17, 2\$	; CLEAR ADDRESS FIELD OF INSTRUCTION
5402	004020	050437	004030	BIS R4, 2\$	; ADD ADDRESS TO INSTRUCTION
5403	004024			ROMCLK	
(1)	004024	004537	003230	JSR R5, .ROMCLK	; CLOCK INSTRUCTION
5404	004030	123100		123100	; MOVE DATA TO SP
5405	004032	005204		INC R4	; INCREMENT COUNT
5406	004034	022704	000010	CMP #10, R4	; DONE YET?
5407	004040	001362		BNE 1\$	; BR IF NO
5408	004042	000207		RTS PC	; RETURN
5409					
5410					
5411	004044			CLRC:	
5412					; THIS SUBROUTINE CLEARS THE MICRO PROCESSOR C BIT
5413					
5414	004044			ROMCLK	
(1)	004044	004537	003230	JSR R5, .ROMCLK	; CLOCK INSTRUCTION
5415	004050	010000		010000	; MAR_0
5416	004052			ROMCLK	
(1)	004052	004537	003230	JSR R5, .ROMCLK	; CLOCK INSTRUCTION
5417	004056	040400		040400! <0*20>	; CLEAR C BIT
5418	004060	000207		RTS PC	; RETURN
5419					
5420					
5421	004062			SETC:	
5422					; THIS SUBROUTINE SETS THE MICRO PROCESSOR C BIT
5423					
5424	004062			ROMCLK	
(1)	004062	004537	003230	JSR R5, .ROMCLK	; CLOCK INSTRUCTION
5425	004066	010003		010003	; MAR_3
5426	004070			ROMCLK	
(1)	004070	004537	003230	JSR R5, .ROMCLK	; CLOCK INSTRUCTION
5427	004074	040403		040403! <0*20>	; SET C BIT
5428	004076	000207		RTS PC	; RETURN
5429					
5430					
5431					
5432					

5434  
5435  
5436  
5437  
5438  
5439

.SBTTL GLOBAL ERROR REPORT SECTION

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

5441  
5442  
5443  
5444 004100 047045 052045 047045 FM1: .ASCIZ /%N%T%N/  
004106 000  
5445 004107 045 031517 051445 TFM1: .ASCIZ /%03%S5%03%S5%03%N2/  
004114 022465 031517 051445  
004122 022465 031517 047045  
004130 000062  
5446 004132 047445 022466 031123 TFM2: .ASCIZ /%06%S2%06%N2/  
004140 047445 022466 031116  
004146 000  
5447 004147 045 031517 051445 TFM5: .ASCIZ /%03%S5%03%N2/  
004154 022465 031517 047045  
004162 000062  
5448 004164 047045 047445 022463 TFM27: .ASCIZ /%N%03%S5%03%S7%03%N2/  
004172 032523 047445 022463  
004200 033523 047445 022463  
004206 031116 000  
5449 004211 045 022516 043101 TFM37: .ASCIZ /%N%AFAILING ADDRESS IS: %06/  
004216 044501 044514 043516  
004224 040440 042104 042522  
004232 051523 044440 035123  
004240 022440 033117 000  
5450  
5451

5453					
5454					
5455	004245	122	043505	051511	EM1: .ASCIIZ &REGISTER ADDRESS TEST&
	004252	042524	020122	042101	
	004260	051104	051505	020123	
	004266	042524	052123	000	
5456	004273	111	052502	025123	EM2: .ASCIIZ &IBUS* REGISTER DUAL ADDRESSING TEST&
	004300	051040	043505	051511	
	004306	042524	020122	052504	
	004314	046101	040440	042104	
	004322	042522	051523	047111	
	004330	020107	042524	052123	
	004336	000			
5457	004337	111	052502	020123	EM30: .ASCIIZ ''IBUS REGISTER DUAL ADDRESSING TEST''
	004344	042522	044507	052123	
	004352	051105	042040	040525	
	004360	020114	042101	051104	
	004366	051505	044523	043516	
	004374	052040	051505	000124	
5458	004402	051102	051040	043505	EM3: .ASCIIZ /BR REGISTER DATA TEST/
	004410	051511	042524	020122	
	004416	040504	040524	052040	
	004424	051505	000124		

CZDMPAO M8207 STATIC DIAG #1  
CZDMPA.P11 17-JUL-79 14:33

MACY11 30A(1052) 17-JUL-79 14:39 PAGE 32  
GLOBAL ERROR REPORT SECTION

SEQ 0066

5460	004430	041523	040522	041524	EM4:	.ASCIZ /SCRATCH PAD DATA TEST/
	004436	020110	040520	020104		
	004444	040504	040524	052040		
	004452	051505	000124			



CZDMPAO M8207 STATIC DIAG #1 MACY11 30A(1052) 17-JUL-79 14:39 C 6  
CZDMPA.P11 17-JUL-79 14:33 GLOBAL ERROR REPORT SECTION PAGE 33

SEQ 0067

5462 004456 041523 040522 041524 EM5: .ASCIZ /SCRATCH PAD DUAL ADDRESSING TEST/  
004464 020110 040520 020104  
004472 052504 046101 040440  
004500 042104 042522 051523  
004506 047111 020107 042524  
004514 052123 000

CZDMPAO M8207 STATIC DIAG #1  
CZDMPA.P11 17-JUL-79 14:33

MACY11 30A(1052) 17-JUL-79 14:39 PAGE 34  
GLOBAL ERROR REPORT SECTION

SEQ 0068

5464	004517	115	044501	020116	EM6:	.ASCIIZ /MAIN MEMORY DATA TEST/
	004524	042515	047515	054522		
	004532	042040	052101	020101		
	004540	042524	052123	000		
5465	004545	115	044501	020116	EM7:	.ASCIIZ /MAIN MEMORY DUAL ADDRESSING TEST/
	004552	042515	047515	054522		
	004560	042040	040525	020114		
	004566	042101	051104	051505		
	004574	044523	043516	052040		
	004602	051505	000124			

5467	004606	052501	047524	046440	EM10:	.ASCIZ	/AUTO MARINC FUNCTION TEST/
	004614	051101	047111	020103			
	004622	052506	041516	044524			
	004630	047117	052040	051505			
	004636	000124					
5468	004640	050116	020122	042524	EM11:	.ASCIZ	/NPR TEST/
	004646	052123	000				
5469	004651	115	046125	044524	EM12:	.ASCIZ	/MULTIPLE NPR TEST/
	004656	046120	020105	050116			
	004664	020122	042524	052123			
	004672	000					
5470	004673	116	047117	042440	EM13:	.ASCIZ	/NON EX MEM FAILED/
	004700	020130	042515	020115			
	004706	040506	046111	042105			
	004714	000					
5471	004715	120	047522	051107	EM14:	.ASCIZ	/PROGRAM CLOCK TEST/
	004722	046501	041440	047514			
	004730	045503	052040	051505			
	004736	000124					
5472	004740	046101	020125	052506	EM15:	.ASCIZ	/ALU FUNCTION WITH C BIT CLEAR TEST/
	004746	041516	044524	047117			
	004754	053440	052111	020110			
	004762	020103	044502	020124			
	004770	046103	040505	020122			
	004776	042524	052123	000			

5474	005003	120	053517	051105	EM16:	.ASCIZ /POWER FAIL: BUS INIT WAS NOT BLOCKED/
	005010	043040	044501	035114		
	005016	041040	051525	044440		
	005024	044516	020124	040527		
	005032	020123	047516	020124		
	005040	046102	041517	042513		
	005046	000104				
5475	005050				EM35:	
5476	005050	047506	041522	020105	EM17:	.ASCIZ /FORCE POWER FAIL ERROR/
	005056	047520	042527	020122		
	005064	040506	046111	042440		
	005072	051122	051117	000		
5477	005077	116	044517	042523	EM20:	.ASCIZ /NOISE TEST ON IBUS*,IBUS,SPAD,MEMORY/
	005104	052040	051505	020124		
	005112	047117	044440	052502		
	005120	025123	044454	052502		
	005126	026123	050123	042101		
	005134	046454	046505	051117		
	005142	000131				
5478	005144	046101	020125	020103	EM21:	.ASCIZ /ALU C BIT TEST FAILURE/
	005152	044502	020124	042524		
	005160	052123	043040	044501		
	005166	052514	042522	000		
5479	005173	124	046511	020105	EM22:	.ASCIZ /TIME OUT ERROR/
	005200	052517	020124	051105		
	005206	047522	000122			
5480	005212	046101	020125	052506	EM23:	.ASCIZ /ALU FUNCTION TEST WITH C BIT SET/
	005220	041516	044524	047117		
	005226	052040	051505	020124		
	005234	044527	044124	041440		
	005242	041040	052111	051440		
	005250	052105	000			
5481	005253	125	041520	051440	EM24:	.ASCIZ /UPC SEQUENCE ERROR/
	005260	050505	042525	041516		
	005266	020105	051105	047522		
	005274	000122				
5482	005276	050125	043040	044501	EM31:	.ASCIZ 'UP FAILED TO INTERRUPT'
	005304	042514	020104	047524		
	005312	044440	052116	051105		
	005320	052522	052120	000		
5483	005325	125	020120	047111	EM32:	.ASCIZ 'UP INTERRUPTED TO WRONG VECTOR'
	005332	042524	051122	050125		
	005340	042524	020104	047524		
	005346	053440	047522	043516		
	005354	053040	041505	047524		
	005362	000122				
5484	005364	047125	054105	042520	EM33:	.ASCIZ 'UNEXPECTED INTERRUPT FROM UP'
	005372	052103	042105	044440		
	005400	052116	051105	052522		
	005406	052120	043040	047522		
	005414	020115	050125	000		
5485	005421	101	052514	043040	EM34:	.ASCIZ 'ALU FLAG TEST'
	005426	040514	020107	042524		
	005434	052123	000			
5486	005437	110	046105	020114	EM25:	.ASCIZ /HELL RAISER TEST/
	005444	040522	051511	051105		

5487	005452	052040	051505	000124	
	005460	040515	047111	040524	EM26: .ASCIZ /MAINTANCE REGISTER ERROR/
	005466	041516	020105	042522	
	005474	044507	052123	051105	
	005502	042440	051122	051117	
	005510	000			
5488	005511	111	052502	025123	EM27: .ASCIZ 'IBUS* WRITE/READ ERROR'
	005516	053440	044522	042524	
	005524	051057	040505	020104	
	005532	051105	047522	000122	
5489	005540	047111	052123	052522	EM28: .ASCIZ /INSTRUCTION TEST FAILURE/
	005546	052103	047511	020116	
	005554	042524	052123	043040	
	005562	044501	052514	042522	
	005570	000			
5490	005571	111	052502	027523	EM29: .ASCIZ 'IBUS/OBUS WRITE/READ ERROR'
	005576	041117	051525	053440	
	005604	044522	042524	051057	
	005612	040505	020104	051105	
	005620	047522	000122		
5491					
5492	005624	047511	020120	040515	EM36: .ASCIZ 'IOP MAIN MEM. LOAD ERROR-RUN MCPU MEM. DIAG.'
	005632	047111	046440	046505	
	005640	020056	047514	042101	
	005646	042440	051122	051117	
	005654	051055	047125	046440	
	005662	050103	020125	042515	
	005670	027115	042040	040511	
	005676	027107	000		
5493	005701	000			EM37: .ASCIZ //
5494					
5495					

(ZDMPAO M8207 STATIC DIAG #1 MACY11 30A(1052) 17-JUL-79 14:39 H 6  
(ZDMPA.P11 17-JUL-79 14:33 PAGE 37  
GLOBAL ERROR REPORT SECTION

SEQ 0072

5497

CZDMPAO M8207 STATIC DIAG #1 MA Y11 30A(1052) 17-JUL-79 14:39 I 6  
CZDMPA.P11 17-JUL-79 14:33 GLOBAL ERROR REPORT SECTION PAGE 38

SEQ 0073

5499 005702 000

DH0: .ASCIZ //

5501	005703	107	047517	020104	DH1:	.ASCIZ	/GOOD	BAD	REGISTER/
	005710	020040	041040	042101					
	005716	020040	020040	051040					
	005724	043505	051511	042524					
	005732	000122							
5502	005734	047507	042117	020040	DH2:	.ASCIZ	/GOOD	BAD/	
	005742	020040	040502	000104					
5503	005750	047507	042117	020040	DH3:	.ASCIZ	/GOOD	BAD	ADDRESS/
	005756	020040	040502	020104					
	005764	020040	020040	042101					
	005772	051104	051505	000123					
5504	006000	047507	042117	020040	DH4:	.ASCIZ	/GOOD	BAD/	
	006006	020040	040502	000104					
5505	006014	042522	027107	042440	DH27:	.ASCIZ	/REG. EXPECTED FOUND/		
	006022	050130	041505	042524					
	006030	020104	047506	047125					
	006036	000104							

5506  
5507  
5508  
5509  
5510

.EVEN

1



CZDMPAO M8207 STATIC DIAG #1 MACY11 30A(1052) 17-JUL-79 14:39 K 6  
CZDMPA.P11 17-JUL-79 14:33 GLOBAL ERROR REPORT SECTION PAGE 40

SEQ 0075

5512  
5513  
5514

CZDMPAO M8207 STATIC DIAG #1 MACY11 30A(1052) 17-JUL-79 14:39 L 6  
CZDMPA.P11 17-JUL-79 14:33 PAGE 41  
GLOBAL ERROR REPORT SECTION

SEQ 0076

5516  
5517

-----  
; MACRO'S NEEDED TO REPORT ERRORS

5519  
5520  
5521  
5522  
5523  
5524  
5525  
5526  
5527

-----  
.MACRO MDT0  
.ENDM  
  
.MACRO MDT1  
PRINTB #TFM1,\$GDDAT,\$BDDAT,\$GDADR  
.ENDM  
  
.MACRO MDT2

5  
CZDMPAO M8207 STATIC DIAG #1 MACY11 30A(1052) 17-JUL-79 14:39 N 6  
CZDMPA.P11 17-JUL-79 14:33 PAGE 43  
GLOBAL ERROR REPORT SECTION

SEQ 0078

```
5529          PRINTB #TFM2,$GDDAT,$BDDAT
5530      .ENDM
5531
5532      .MACRO MDT5
5533          PRINTB #TFM5,$GDDAT,$BDDAT
5534      .ENDM
```

```
5536 .MACRO MDT27
5537 PRINTB #TFM27,MRO,$GDDAT,$BDDAT
5538 .ENDM
5539
5540 .MACRO $MD,ERNB,ERHM,ERFM
5541 .NLIST
5542 : ERNB = ERROR NUMBER
5543 : ERFM = FORMAT NUMBER
5544 : ERHM = HEADER NUMBER
5545 .LIST
5546 BGNMSG ERR'ERNB'
5547 PRINTB #FM1,#DH'ERHM'
5548 MDT'ERFM'
5549 ENDMSG
```

```

5551
5552
5553
5554
5555
5556 006040
(4) 006040
(9) 006040 012746 005734
(8) 006044 012746 004100
(7) 006050 012746 000002
(4) 006054 010600
(5) 006056 104414
(5) 006060 062706 000006
(11) 006064 013746 002640
(10) 006070 013746 002636
(9) 006074 012746 004132
(8) 006100 012746 000003
(5) 006104 010600
(6) 006106 104414
(6) 006110 062706 000010
(4) 006114
(4) 006114 104423
5557 006116
(4) 006116
(9) 006116 012746 005734
(8) 006122 012746 004100
(7) 006126 012746 000002
(4) 006132 010600
(5) 006134 104414
(5) 006136 062706 000006
(11) 006142 013746 002640
(10) 006146 013746 002636
(9) 006152 012746 004132
(8) 006156 012746 000003
(5) 006162 010600
(6) 006164 104414
(6) 006166 062706 000010
(4) 006172
(4) 006172 104423
5558 006174
(4) 006174
(9) 006174 012746 005734
(8) 006200 012746 004100
(7) 006204 012746 000002
(4) 006210 010600
(5) 006212 104414
(5) 006214 062706 000006
(11) 006220 013746 002640
(10) 006224 013746 002636
(9) 006230 012746 004132
(8) 006234 012746 000003
(5) 006240 010600
(6) 006242 104414
(6) 006244 062706 000010
(4) 006250
(4) 006250 104423

```

.ENDM

```

ERR1:: $MD 1.2.2
MOV #DH2,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #6,SP
MOV $BDDAT,-(SP)
MOV $GDDAT,-(SP)
MOV #TFM2,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #10,SP
L10003: TRAP C$MSG
$MD 2.2.2
ERR2:: MOV #DH2,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #6,SP
MOV $BDDAT,-(SP)
MOV $GDDAT,-(SP)
MOV #TFM2,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #10,SP
L10004: TRAP C$MSG
$MD 3.2.2
ERR3:: MOV #DH2,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #6,SP
MOV $BDDAT,-(SP)
MOV $GDDAT,-(SP)
MOV #TFM2,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #10,SP
L10005: TRAP C$MSG

```

5559 006252  
(4) 006252  
(9) 006252 012746 005703  
(8) 006256 012746 004100  
(7) 006262 012746 000002  
(4) 006266 010600  
(5) 006270 104414  
(5) 006272 062706 000006  
(12) 006276 013746 002632  
(11) 006302 013746 002640  
(10) 006306 013746 002636  
(9) 006312 012746 004107  
(8) 006316 012746 000004  
(5) 006322 010600  
(6) 006324 104414  
(6) 006326 062706 000012  
(4) 006332  
(4) 006332 104423  
5560 006334  
(4) 006334  
(9) 006334 012746 005703  
(8) 006340 012746 004100  
(7) 006344 012746 000002  
(4) 006350 010600  
(5) 006352 104414  
(5) 006354 062706 000006  
(12) 006360 013746 002632  
(11) 006364 013746 002640  
(10) 006370 013746 002636  
(9) 006374 012746 004107  
(8) 006400 012746 000004  
(5) 006404 010600  
(6) 006406 104414  
(6) 006410 062706 000012  
(4) 006414  
(4) 006414 104423  
5561 006416  
(4) 006416  
(9) 006416 012746 005750  
(8) 006422 012746 004100  
(7) 006426 012746 000002  
(4) 006432 010600  
(5) 006434 104414  
(5) 006436 062706 000006  
(12) 006442 013746 002632  
(11) 006446 013746 002640  
(10) 006452 013746 002636  
(9) 006456 012746 004107  
(8) 006462 012746 000004  
(5) 006466 010600  
(6) 006470 104414  
(6) 006472 062706 000012  
(4) 006476  
(4) 006476 104423  
5562 006500  
(4) 006500

ERR4:: \$MD 4,1,1  
MOV #DH1,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV \$GDADR,-(SP)  
MOV \$BDDAT,-(SP)  
MOV \$GDDAT,-(SP)  
MOV #TFM1,-(SP)  
MOV #4,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #12,SP  
L10006: TRAP C\$MSG  
\$MD 5,1,1  
ERR5:: MOV #DH1,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV \$GDADR,-(SP)  
MOV \$BDDAT,-(SP)  
MOV \$GDDAT,-(SP)  
MOV #TFM1,-(SP)  
MOV #4,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #12,SP  
L10007: TRAP C\$MSG  
\$MD 6,3,1  
ERR6:: MOV #DH3,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV \$GDADR,-(SP)  
MOV \$BDDAT,-(SP)  
MOV \$GDDAT,-(SP)  
MOV #TFM1,-(SP)  
MOV #4,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #12,SP  
L10010: TRAP C\$MSG  
\$MD 7,3,1  
ERR7::

(9)	006500	012746	005750	MOV	#DH3,-(SP)
(8)	006504	012746	004100	MOV	#FM1,-(SP)
(7)	006510	012746	000002	MOV	#2,-(SP)
(4)	006514	010600		MOV	SP,R0
(5)	006516	104414		TRAP	C\$PNTB
(5)	006520	062706	000006	ADD	#6,SP
(12)	006524	013746	002632	MOV	\$GDADR,-(SP)
(11)	006530	013746	002640	MOV	\$BDDAT,-(SP)
(10)	006534	013746	002636	MOV	\$GDDAT,-(SP)
(9)	006540	012746	004107	MOV	#TFM1,-(SP)
(8)	006544	012746	000004	MOV	#4,-(SP)
(5)	006550	010600		MOV	SP,R0
(6)	006552	104414		TRAP	C\$PNTB
(6)	006554	062706	000012	ADD	#12,SP
(4)	006560				
(4)	006560	104423		L10011:	TRAP
5563	006562				C\$MSG
(4)	006562			ERR10::	\$MD
(9)	006562	012746	005750		10,3,1
(8)	006566	012746	004100	MOV	#DH3,-(SP)
(7)	006572	012746	000002	MOV	#FM1,-(SP)
(4)	006576	010600		MOV	#2,-(SP)
(5)	006600	104414		MOV	SP,R0
(5)	006602	062706	000006	TRAP	C\$PNTB
(12)	006606	013746	002632	ADD	#6,SP
(11)	006612	013746	002640	MOV	\$GDADR,-(SP)
(10)	006616	013746	002636	MOV	\$BDDAT,-(SP)
(9)	006622	012746	004107	MOV	\$GDDAT,-(SP)
(8)	006626	012746	000004	MOV	#TFM1,-(SP)
(5)	006632	010600		MOV	#4,-(SP)
(6)	006634	104414		MOV	SP,R0
(6)	006636	062706	000012	TRAP	C\$PNTB
(4)	006642			ADD	#12,SP
(4)	006642	104423		L10012:	TRAP
5564	006644				C\$MSG
(4)	006644			ERR11::	\$MD
(9)	006644	012746	005734		11,2,2
(8)	006650	012746	004100	MOV	#DH2,-(SP)
(7)	006654	012746	000002	MOV	#FM1,-(SP)
(4)	006660	010600		MOV	#2,-(SP)
(5)	006662	104414		MOV	SP,R0
(5)	006664	062706	000006	TRAP	C\$PNTB
(11)	006670	013746	002640	ADD	#6,SP
(10)	006674	013746	002636	MOV	\$BDDAT,-(SP)
(9)	006700	012746	004132	MOV	\$GDDAT,-(SP)
(8)	006704	012746	000003	MOV	#TFM2,-(SP)
(5)	006710	010600		MOV	#3,-(SP)
(6)	006712	104414		MOV	SP,R0
(6)	006714	062706	000010	TRAP	C\$PNTB
(4)	006720			ADD	#10,SP
(4)	006720	104423		L10013:	TRAP
5565	006722				C\$MSG
(4)	006722			ERR12::	\$MD
(9)	006722	012746	005734		12,2,2
(8)	006726	012746	004100	MOV	#DH2,-(SP)
(7)	006732	012746	000002	MOV	#FM1,-(SP)
				MOV	#2,-(SP)



(4)	006736	010600		MOV	SP,R0
(5)	006740	104414		TRAP	C\$PNTB
(5)	006742	062706	000006	ADD	#6,SP
(11)	006746	013746	002640	MOV	\$BDDAT,-(SP)
(10)	006752	013746	002636	MOV	\$GDDAT,-(SP)
(9)	006756	012746	004132	MOV	#TFM2,-(SP)
(8)	006762	012746	000003	MOV	#3,-(SP)
(5)	006766	010600		MOV	SP,R0
(6)	006770	104414		TRAP	C\$PNTB
(6)	006772	062706	000010	ADD	#10,SP
(4)	006776				
(4)	006776	104423		L10014:	TRAP
5566	007000				C\$MSG
(4)	007000				\$MD
(9)	007000	012746	005702	ERR13::	13,0,0
(8)	007004	012746	004100		
(7)	007010	012746	000002	MOV	#DH0,-(SP)
(4)	007014	010600		MOV	#FM1,-(SP)
(5)	007016	104414		MOV	#2,-(SP)
(5)	007020	062706	000006	MOV	SP,R0
(4)	007024			TRAP	C\$PNTB
(4)	007024	104423		ADD	#6,SP
				L10015:	TRAP
					C\$MSG

CZDMPAO M8207 STATIC DIAG #1  
CZDMPA.P11 17-JUL-79 14:33

MACY11 30A(1052) 17-JUL-79 14:39 PAGE 46  
GLOBAL ERROR REPORT SECTION

G 7

SEQ 0084

5568 007026  
(4) 007026  
(9) 007026 012746 005734  
(8) 007032 012746 004100  
(7) 007036 012746 000002  
(4) 007042 010600  
(5) 007044 104414  
(5) 007046 062706 000006  
(11) 007052 013746 002640  
(10) 007056 013746 002636  
(9) 007062 012746 004132  
(8) 007066 012746 000003  
(5) 007072 010600  
(6) 007074 104414  
(6) 007076 062706 000010  
(4) 007102  
(4) 007102 104425

ERR14:: \$MD 14,2,2  
MOV #DH2,-(SP)  
MOV #FM1,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #6,SP  
MOV \$BDDAT,-(SP)  
MOV \$GDDAT,-(SP)  
MOV #TFM2,-(SP)  
MOV #3,-(SP)  
MOV SP,R0  
TRAP C\$PNTB  
ADD #10,SP  
L10016: TRAP C\$MSG

5570 007104  
 (4) 007104  
 (9) 007104 012746 006000  
 (8) 007110 012746 004100  
 (7) 007114 012746 000002  
 (4) 007120 010600  
 (5) 007122 104414  
 (5) 007124 062706 000006  
 (11) 007130 013746 002640  
 (10) 007134 013746 002636  
 (9) 007140 012746 004147  
 (8) 007144 012746 000003  
 (5) 007150 010600  
 (6) 007152 104414  
 (6) 007154 062706 000010  
 (4) 007160  
 (4) 007160 104423  
 5571 007162  
 (4) 007162  
 (9) 007162 012746 005702  
 (8) 007166 012746 004100  
 (7) 007172 012746 000002  
 (4) 007176 010600  
 (5) 007200 104414  
 (5) 007202 062706 000006  
 (4) 007206  
 (4) 007206 104423  
 5572 007210  
 (4) 007210  
 (9) 007210 012746 005702  
 (8) 007214 012746 004100  
 (7) 007220 012746 000002  
 (4) 007224 010600  
 (5) 007226 104414  
 (5) 007230 062706 000006  
 (4) 007234  
 (4) 007234 104423  
 5573 007236  
 (4) 007236  
 (9) 007236 012746 005734  
 (8) 007242 012746 004100  
 (7) 007246 012746 000002  
 (4) 007252 010600  
 (5) 007254 104414  
 (5) 007256 062706 000006  
 (11) 007262 013746 002640  
 (10) 007266 013746 002636  
 (9) 007272 012746 004132  
 (8) 007276 012746 000003  
 (5) 007302 010600  
 (6) 007304 104414  
 (6) 007306 062706 000010  
 (4) 007312  
 (4) 007312 104423  
 5574 007314  
 (4) 007314

ERR15:: SMD 15,4,5  
 MOV #DH4,-(SP)  
 MOV #FM1,-(SP)  
 MOV #2,-(SP)  
 MOV SP,R0  
 TRAP C\$PNTB  
 ADD #6,SP  
 MOV \$BDDAT,-(SP)  
 MOV \$GDDAT,-(SP)  
 MOV #TFM5,-(SP)  
 MOV #3,-(SP)  
 MOV SP,R0  
 TRAP C\$PNTB  
 ADD #10,SP  
 L10017: TRAP C\$MSG  
 SMD 16,0,0  
 ERR16:: MOV #DH0,-(SP)  
 MOV #FM1,-(SP)  
 MOV #2,-(SP)  
 MOV SP,R0  
 TRAP C\$PNTB  
 ADD #6,SP  
 L10020: TRAP C\$MSG  
 SMD 17,0,0  
 ERR17:: MOV #DH0,-(SP)  
 MOV #FM1,-(SP)  
 MOV #2,-(SP)  
 MOV SP,R0  
 TRAP C\$PNTB  
 ADD #6,SP  
 L10021: TRAP C\$MSG  
 SMD 20,2,2  
 ERR20:: MOV #DH2,-(SP)  
 MOV #FM1,-(SP)  
 MOV #2,-(SP)  
 MOV SP,R0  
 TRAP C\$PNTB  
 ADD #6,SP  
 MOV \$BDDAT,-(SP)  
 MOV \$GDDAT,-(SP)  
 MOV #TFM2,-(SP)  
 MOV #3,-(SP)  
 MOV SP,R0  
 TRAP C\$PNTB  
 ADD #10,SP  
 L10022: TRAP C\$MSG  
 SMD 21,0,0  
 ERR21::

(9)	007314	012746	005702	MOV	#DH0,-(SP)
(8)	007320	012746	004100	MOV	#FM1,-(SP)
(7)	007324	012746	000002	MOV	#2,-(SP)
(4)	007330	010600		MOV	SP,R0
(5)	007332	104414		TRAP	C\$PNTB
(5)	007334	062706	000006	ADD	#6,SP
(4)	007340			L10023:	
(4)	007340	104423		TRAP	C\$MSG
5575	007342			\$MD	22,0,0
(4)	007342			ERR22::	
(9)	007342	012746	005702	MOV	#DH0,-(SP)
(8)	007346	012746	004100	MOV	#FM1,-(SP)
(7)	007352	012746	000002	MOV	#2,-(SP)
(4)	007356	010600		MOV	SP,R0
(5)	007360	104414		TRAP	C\$PNTB
(5)	007362	062706	000006	ADD	#6,SP
(4)	007366			L10024:	
(4)	007366	104423		TRAP	C\$MSG
5576	007370			\$MD	23,4,5
(4)	007370			ERR23::	
(9)	007370	012746	006000	MOV	#DH4,-(SP)
(8)	007374	012746	004100	MOV	#FM1,-(SP)
(7)	007400	012746	000002	MOV	#2,-(SP)
(4)	007404	010600		MOV	SP,R0
(5)	007406	104414		TRAP	C\$PNTB
(5)	007410	062706	000006	ADD	#6,SP
(11)	007414	013746	002640	MOV	\$BDDAT,-(SP)
(10)	007420	013746	002636	MOV	\$GDDAT,-(SP)
(9)	007424	012746	004147	MOV	#TFM5,-(SP)
(8)	007430	012746	000003	MOV	#3,-(SP)
(5)	007434	010600		MOV	SP,R0
(6)	007436	104414		TRAP	C\$PNTB
(6)	007440	062706	000010	ADD	#10,SP
(4)	007444			L10025:	
(4)	007444	104423		TRAP	C\$MSG
5577	007446			\$MD	24,0,0
(4)	007446			ERR24::	
(9)	007446	012746	005702	MOV	#DH0,-(SP)
(8)	007452	012746	004100	MOV	#FM1,-(SP)
(7)	007456	012746	000002	MOV	#2,-(SP)
(4)	007462	010600		MOV	SP,R0
(5)	007464	104414		TRAP	C\$PNTB
(5)	007466	062706	000006	ADD	#6,SP
(4)	007472			L10026:	
(4)	007472	104423		TRAP	C\$MSG
5578	007474			\$MD	25,2,2
(4)	007474			ERR25::	
(9)	007474	012746	005734	MOV	#DH2,-(SP)
(8)	007500	012746	004100	MOV	#FM1,-(SP)
(7)	007504	012746	000002	MOV	#2,-(SP)
(4)	007510	010600		MOV	SP,R0
(5)	007512	104414		TRAP	C\$PNTB
(5)	007514	062706	000006	ADD	#6,SP
(11)	007520	013746	002640	MOV	\$BDDAT,-(SP)
(10)	007524	013746	002636	MOV	\$GDDAT,-(SP)
(9)	007530	012746	004132	MOV	#TFM2,-(SP)

(8)	007534	012746	000003		MOV	#3,-(SP)
(5)	007540	010600			MOV	SP,R0
(6)	007542	104414			TRAP	C\$PNTB
(6)	007544	062706	000010		ADD	#10,SP
(4)	007550			L10027:		
(4)	007550	104423			TRAP	C\$MSG
5579	007552				\$MD	26,2,2
(4)	007552			ERR26::		
(9)	007552	012746	005734		MOV	#DH2,-(SP)
(8)	007556	012746	004100		MOV	#FM1,-(SP)
(7)	007562	012746	000002		MOV	#2,-(SP)
(4)	007566	010600			MOV	SP,R0
(5)	007570	104414			TRAP	C\$PNTB
(5)	007572	062706	000006		ADD	#6,SP
(11)	007576	013746	002640		MOV	\$BDDAT,-(SP)
(10)	007602	013746	002636		MOV	\$GDDAT,-(SP)
(9)	007606	012746	004132		MOV	#TFM2,-(SP)
(8)	007612	012746	000003		MOV	#3,-(SP)
(5)	007616	010600			MOV	SP,R0
(6)	007620	104414			TRAP	C\$PNTB
(6)	007622	062706	000010		ADD	#10,SP
(4)	007626			L10030:		
(4)	007626	104423			TRAP	C\$MSG
5580	007630				\$MD	27,27,27
(4)	007630			ERR27::		
(9)	007630	012746	006014		MOV	#DH27,-(SP)
(8)	007634	012746	004100		MOV	#FM1,-(SP)
(7)	007640	012746	000002		MOV	#2,-(SP)
(4)	007644	010600			MOV	SP,R0
(5)	007646	104414			TRAP	C\$PNTB
(5)	007650	062706	000006		ADD	#6,SP
(12)	007654	013746	002640		MOV	\$BDDAT,-(SP)
(11)	007660	013746	002636		MOV	\$GDDAT,-(SP)
(10)	007664	013746	002624		MOV	MR0,-(SP)
(9)	007670	012746	004164		MOV	#TFM27,-(SP)
(8)	007674	012746	000004		MOV	#4,-(SP)
(5)	007700	010600			MOV	SP,R0
(6)	007702	104414			TRAP	C\$PNTB
(6)	007704	062706	000012		ADD	#12,SP
(4)	007710			L10031:		
(4)	007710	104423			TRAP	C\$MSG
5581	007712				\$MD	28,2,2
(4)	007712			ERR28::		
(9)	007712	012746	005734		MOV	#DH2,-(SP)
(8)	007716	012746	004100		MOV	#FM1,-(SP)
(7)	007722	012746	000002		MOV	#2,-(SP)
(4)	007726	010600			MOV	SP,R0
(5)	007730	104414			TRAP	C\$PNTB
(5)	007732	062706	000006		ADD	#6,SP
(11)	007736	013746	002640		MOV	\$BDDAT,-(SP)
(10)	007742	013746	002636		MOV	\$GDDAT,-(SP)
(9)	007746	012746	004132		MOV	#TFM2,-(SP)
(8)	007752	012746	000003		MOV	#3,-(SP)
(5)	007756	010600			MOV	SP,R0
(6)	007760	104414			TRAP	C\$PNTB
(6)	007762	062706	000010		ADD	#10,SP

(4) 007766  
 (4) 007766 104423  
 5582 007770  
 (4) 007770  
 (9) 007770 012746 006014  
 (8) 007774 012746 004100  
 (7) 010000 012746 000002  
 (4) 010004 010600  
 (5) 010006 104414  
 (5) 010010 062706 000006  
 (12) 010014 013746 002640  
 (11) 010020 013746 002636  
 (10) 010024 013746 002624  
 (9) 010030 012746 004164  
 (8) 010034 012746 000004  
 (5) 010040 010600  
 (6) 010042 104414  
 (6) 010044 062706 000012  
 (4) 010050  
 (4) 010050 104423  
 5583 010052  
 (4) 010052  
 (9) 010052 012746 005734  
 (8) 010056 012746 004100  
 (7) 010062 012746 000002  
 (4) 010066 010600  
 (5) 010070 104414  
 (5) 010072 062706 000006  
 (11) 010076 013746 002640  
 (10) 010102 013746 002636  
 (9) 010106 012746 004132  
 (8) 010112 012746 000003  
 (5) 010116 010600  
 (6) 010120 104414  
 (6) 010122 062706 000010  
 (4) 010126  
 (4) 010126 104423  
 5584 010130  
 (4) 010130  
 (9) 010130 012746 005702  
 (8) 010134 012746 004100  
 (7) 010140 012746 000002  
 (4) 010144 010600  
 (5) 010146 104414  
 (5) 010150 062706 000006  
 (4) 010154  
 (4) 010154 104423  
 5585 010156  
 (4) 010156  
 (9) 010156 012746 005702  
 (8) 010162 012746 004100  
 (7) 010166 012746 000002  
 (4) 010172 010600  
 (5) 010174 104414  
 (5) 010176 062706 000006  
 (4) 010202

L10032:  
 TRAP C\$MSG  
 SMD 29,27,27  
 ERR29::  
 MOV #DH27,-(SP)  
 MOV #FM1,-(SP)  
 MOV #2,-(SP)  
 MOV SP,R0  
 TRAP C\$PNTB  
 ADD #6,SP  
 MOV \$BDDAT,-(SP)  
 MOV \$GDDAT,-(SP)  
 MOV MR0,-(SP)  
 MOV #TFM27,-(SP)  
 MOV #4,-(SP)  
 MOV SP,R0  
 TRAP C\$PNTB  
 ADD #12,SP  
 L10033:  
 TRAP C\$MSG  
 SMD 30,2,2  
 ERR30::  
 MOV #DH2,-(SP)  
 MOV #FM1,-(SP)  
 MOV #2,-(SP)  
 MOV SP,R0  
 TRAP C\$PNTB  
 ADD #6,SP  
 MOV \$BDDAT,-(SP)  
 MOV \$GDDAT,-(SP)  
 MOV #TFM2,-(SP)  
 MOV #3,-(SP)  
 MOV SP,R0  
 TRAP C\$PNTB  
 ADD #10,SP  
 L10034:  
 TRAP C\$MSG  
 SMD 31,0,0  
 ERR31::  
 MOV #DH0,-(SP)  
 MOV #FM1,-(SP)  
 MOV #2,-(SP)  
 MOV SP,R0  
 TRAP C\$PNTB  
 ADD #6,SP  
 L10035:  
 TRAP C\$MSG  
 SMD 32,0,0  
 ERR32::  
 MOV #DH0,-(SP)  
 MOV #FM1,-(SP)  
 MOV #2,-(SP)  
 MOV SP,R0  
 TRAP C\$PNTB  
 ADD #6,SP  
 L10036:

(4)	010202	104423		TRAP	C\$MSG
5586	010204			\$MD	33,2,2
(4)	010204			ERR33::	
(9)	010204	012746	005734	MOV	#DH2,-(SP)
(8)	010210	012746	004100	MOV	#FM1,-(SP)
(7)	010214	012746	000002	MOV	#2,-(SP)
(4)	010220	010600		MOV	SP,R0
(5)	010222	104414		TRAP	C\$PNTB
(5)	010224	062706	000006	ADD	#6,SP
(11)	010230	013746	002640	MOV	\$BDDAT,-(SP)
(10)	010234	013746	002636	MOV	\$GDDAT,-(SP)
(9)	010240	012746	004132	MOV	#TFM2,-(SP)
(8)	010244	012746	000003	MOV	#3,-(SP)
(5)	010250	010600		MOV	SP,R0
(6)	010252	104414		TRAP	C\$PNTB
(6)	010254	062706	000010	ADD	#10,SP
(4)	010260			L10037:	
(4)	010260	104423		TRAP	C\$MSG
5587	010262			\$MD	34,2,2
(4)	010262			ERR34::	
(9)	010262	012746	005734	MOV	#DH2,-(SP)
(8)	010266	012746	004100	MOV	#FM1,-(SP)
(7)	010272	012746	000002	MOV	#2,-(SP)
(4)	010276	010600		MOV	SP,R0
(5)	010300	104414		TRAP	C\$PNTB
(5)	010302	062706	000006	ADD	#6,SP
(11)	010306	013746	002640	MOV	\$BDDAT,-(SP)
(10)	010312	013746	002636	MOV	\$GDDAT,-(SP)
(9)	010316	012746	004132	MOV	#TFM2,-(SP)
(8)	010322	012746	000003	MOV	#3,-(SP)
(5)	010326	010600		MOV	SP,R0
(6)	010330	104414		TRAP	C\$PNTB
(6)	010332	062706	000010	ADD	#10,SP
(4)	010336			L10040:	
(4)	010336	104423		TRAP	C\$MSG
5588	010340			\$MD	35,2,2
(4)	010340			ERR35::	
(9)	010340	012746	005734	MOV	#DH2,-(SP)
(8)	010344	012746	004100	MOV	#FM1,-(SP)
(7)	010350	012746	000002	MOV	#2,-(SP)
(4)	010354	010600		MOV	SP,R0
(5)	010356	104414		TRAP	C\$PNTB
(5)	010360	062706	000006	ADD	#6,SP
(11)	010364	013746	002640	MOV	\$BDDAT,-(SP)
(10)	010370	013746	002636	MOV	\$GDDAT,-(SP)
(9)	010374	012746	004132	MOV	#TFM2,-(SP)
(8)	010400	012746	000003	MOV	#3,-(SP)
(5)	010404	010600		MOV	SP,R0
(6)	010406	104414		TRAP	C\$PNTB
(6)	010410	062706	000010	ADD	#10,SP
(4)	010414			L10041:	
(4)	010414	104423		TRAP	C\$MSG
5589	010416			\$MD	36,2,2
(4)	010416			ERR36::	
(9)	010416	012746	005734	MOV	#DH2,-(SP)
(8)	010422	012746	004100	MOV	#FM1,-(SP)

(7)	010426	012746	000002	MOV	#2,-(SP)
(4)	010432	010600		MOV	SP,R0
(5)	010434	104414		TRAP	C\$PNTB
(5)	010436	062706	000006	ADD	#6,SP
(11)	010442	013746	002640	MOV	\$BDDAT,-(SP)
(10)	010446	013746	002636	MOV	\$GDDAT,-(SP)
(9)	010452	012746	004132	MOV	#TFM2,-(SF)
(8)	010456	012746	000003	MOV	#3,-(SP)
(5)	010462	010600		MOV	SP,R0
(6)	010464	104414		TRAP	C\$PNTB
(6)	010466	062706	000010	ADD	#10,SP
(4)	010472				
(4)	010472	104423		L10042:	TRAP C\$MSG
5590					
5591	010474			BGNMSG	ERR37
(3)	010474			ERR37::	
5592	010474			PRINTF	#FM1,#EM1
(8)	010474	012746	004245	MOV	#EM1,-(SP)
(7)	010500	012746	004100	MOV	#FM1,-(SP)
(6)	010504	012746	000002	MOV	#2,-(SP)
(3)	010510	010600		MOV	SP,R0
(4)	010512	104417		TRAP	C\$PNTF
(4)	010514	062706	000006	ADD	#6,SP
5593	010520			PRINTF	#TFM37,\$GDADR
(8)	010520	013746	002632	MOV	\$GDADR,-(SP)
(7)	010524	012746	004211	MOV	#TFM37,-(SP)
(6)	010530	012746	000002	MOV	#2,-(SP)
(3)	010534	010600		MOV	SP,R0
(4)	010536	104417		TRAP	C\$PNTF
(4)	010540	062706	000006	ADD	#6,SP
5594	010544			ENDMSG	
(3)	010544			L10043:	
(3)	010544	104423		TRAP	C\$MSG
5595					
5596					
5597					



5599  
5600  
5601  
5602  
5603  
5604  
5605  
5606  
5607  
5608  
5614  
5615  
5616  
5623  
5624  
5625  
5626  
5627  
5628  
5629

010546  
(3) 010546  
  
010546 000167  
(3) 010550 000000  
  
010552  
(3) 010552  
(3) 010552 104425

.SBTTL REPORT CODING SECTION

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

BGNRPT  
L\$RPT::

EXIT RPT  
WORD JSJMP  
.WORD L10044-2-.

ENDRPT  
L10044:  
TRAP (\$RPT

```

5631      .SBTTL  INITIALIZE SECTION
5632
5633      ;////////////////////////////////////
5634      ;// THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
5635      ;// AT THE BEGINNING OF EACH PASS.
5636      ;////////////////////////////////////
5637
5638      010554      BGNINIT
5639      (3) 010554      L$INIT::
5640
5641      010554      012705      003130      ;INITIALIZE SUBROUTINE STACK
5642      MOV          #SSTACK,R5
5643      ;STORE BASE LEVEL PROGRAM STACK POINTER
5644      MOV          SP,PSTACK
5645      TST          FTIME
5646      BNE          1$
5647      MOV          @#4,SAVE4
5648      MOV          @#6,SAVE6
5649      MOV          #1,FTIME
5650      1$: MOV          SAVE4,@#4
5651      MOV          SAVE6,@#6
5652      ;SEE IF PROGRAM JUST STARTED, BR IF YES
5653      READEFS      #EF.START
5654      MOV          #EF.START,R0
5655      TRAP          C$REFG
5656      BCOMPLETE      NEWST
5657      BCS          NEWST
5658      ;SEE IF THIS IS A NEW PASS, BR IF YES
5659      READEFS      #EF.NEW
5660      MOV          #EF.NEW,R0
5661      TRAP          C$REFG
5662      BCOMPLETE      NEWST
5663      BCS          NEWST
5664      ;SEE IF PROGRAM WAS JUST CONTINUED
5665      READEFS      #EF.CONTINUE
5666      MOV          #EF.CONTINUE,R0
5667      TRAP          C$REFG
5668      BCOMPLETE      ENDIT
5669      BCS          ENDIT
5670
5671      ;SEE IF PROGRAM JUST RESTARTED, BR IF NOT
5672      READEFS      #EF.RESTART
5673      MOV          #EF.RESTART,R0
5674      TRAP          C$REFG
5675      BNCOMPLETE      GETPRM
5676      BCC          GETPRM
5677      NEWST:
5678      ;RESET LOGICAL DEVICE TO -1
5679      MOV          #-1,LOGDEV
5680
5681      ;GET UNIBUS ADRS, VECTOR, PRIORITY LEVEL, LINE UNIT, SWITCH
5682      ; PACKS, TEST CONNECTOR INFO. FOR THIS M8200,4,7 (CURRENT LOGICAL
5683      ; DEVICE).
5684      GETPRM:
5685      INC          LOGDEV
5686      CMP          LOGDEV,L$UNIT
  
```

5674	010710	002367			BGE	NEWST
5675	010712				GPHARD	LOGDEV,R1
(3)	010712	013700	002552		MOV	LOGDEV,R0
(3)	010716	104442			TRAP	C\$GPHRD
(3)	010720	010001			MOV	R0,R1
5676	010722				BNCOMPLETE	GETPRM
(2)	010722	103365			BCC	GETPRM
5677					:GET ADDRESS OF	M8200,4,7
5678	010724	012137	002626		MOV	(R1)+,WTYPE
5679	010730	011137	002716		MOV	(R1),KMCSR
5680					:GET POINTER TO	M8200,4,7 CSR HI BYTE
5681	010734	011137	002720		MOV	(R1),KMCSRH
5682	010740	005237	002720		INC	KMCSRH
5683					:GET POINTER TO	M8200,4,7 CTL OUT REG
5684	010744	011137	002722		MOV	(R1),KMCTL
5685	010750	062737	000002	002722	ADD	#2,KMCTL
5686					:GET POINTER TO	M8200,4,7 PORT REG - SEL 4
5687	010756	011137	002724		MOV	(R1),KMPO4
5688	010762	062737	000004	002724	ADD	#4,KMPO4
5689					:GET POINTER TO	M8200,4,7 PORT REG - SEL 6
5690	010770	012137	002726		MOV	(R1)+,KMPO6
5691	010774	062737	000006	002726	ADD	#6,KMPO6
5692					:GET POINTER TO	RCV VECTOR
5693	011002	011137	002706		MOV	(R1),KMRVEC
5694					:GET POINTER TO	RCV PRIORITY LEVEL
5695	011006	011137	002710		MOV	(R1),KMRLVL
5696	011012	062737	000002	002710	ADD	#2,KMRLVL
5697					:GET POINTER TO	TX VECTOR
5698	011020	011137	002712		MOV	(R1),KMTVEC
5699	011024	062737	000004	002712	ADD	#4,KMTVEC
5700					:GET POINTER TO	TX PRIORITY LEVEL
5701	011032	011137	002714		MOV	(R1),KMTLVL
5702	011036	062737	000006	002714	ADD	#6,KMTLVL
5703					:PUT VECTOR INTO	STAT1
5704	011044	012137	002700		MOV	(R1)+,STAT1
5705					:PUT PRIORITY INTO	STAT1
5706	011050	052137	002700		BIS	(R1)+,STAT1
5707					:SEE IF NO LINE	UNIT, SET BIT IF YES
5708	011054	005711			TST	(R1)
5709	011056	001004			BNE	50000\$
5710	011060	052737	010000	002700	BIS	#BIT12,STAT1
5711	011066	000416			BR	4\$
5712	011070				50000\$:	
5713					:SEE IF M8201	LINE UNIT, SET BIT IF YES
5714	011070	021127	000001		CMP	(R1),#1
5715	011074	001001			BNE	50001\$
5716	011076	000412			BR	4\$
5717	011100				50001\$:	
5718					:SEE IF M8202	LINE UNIT, SET BIT IF YES
5719	011100	021127	000002		CMP	(R1),#2
5720	011104	001004			BNE	50002\$
5721	011106	052737	020000	002700	BIS	#BIT13,STAT1
5722	011114	000403			BR	4\$
5723	011116				50002\$:	
5724					:SET BIT FOR	M8203 LINE UNIT
5725	011116	052737	100000	002700	BIS	#BIT15,STAT1

```

5726 011124      4$:
5727      ;SET BIT IN STAT1 FOR TEST CONNECTOR
5728 011124 056137 000006 002700      BIS      6(R1),STAT1
5729 011132 062701 000002      ADD      #2,R1
5730      ;SET SWITCH PACK #1 IN STAT2 LOW BYTE
5731 011136 012137 002702      MOV      (R1)+,STAT2
5732      ;SET SWITCH PACK #2 IN STAT2 HIGH BYTE
5733 011142 111137 002703      MOVB     (R1),STAT2+1
5734
5735      ;INCREMENT LOGICAL UNIT (DEVICE) NUMBER
5736      :      INC      LOGDEV
5737 011146 000240      NOP
5738 011150 000240      NOP
5739
5740 011152 012737 002000 002606      MOV      #2000,MEMSZ
5741 011160 005037 002630      CLR      TYPE
5742 011164 123727 002626 000000      CMPB     WTYPE,#0
5743 011172 001422      BEQ      ENDIT
5744 011174 123727 002626 000004      CMPB     WTYPE,#4      ;KMC?
5745 011202 001004      BNE      5$
5746 011204 012737 000001 002630      MOV      #1,TYPE
5747 011212 000412      BR      ENDIT
5748 011214 012737 003777 002606 5$:      MOV      #3777,MEMSZ
5749 011222 123727 002626 000006      CMPB     WTYPE,#6
5750 011230 001003      BNE      ENDIT
5751 011232 012737 000001 002630      MOV      #1,TYPE
5752 011240      ENDIT:
5753 011240      ENDINIT
5754 (3) 011240      L10045:      TRAP      C$INIT
5755 (3) 011240 104411
5756 011242      .EVEN
5757 (3) 011242      L$AUTO:      BGNAUTO
5758 011242 013701 002716      ;DEVICE DOES NOT HAVE A 'READY'
5759 011246 012705 000004      MOV      KMC$R,R1      ;R1 CONTAINS BASE M8200,4,7 ADDRESS
5760 011252 012737 011304 000004      MOV      #4,R5      ;4 REGISTERS TO BE TESTED
5761 011260 012737 000340 000006      MOV      #2$,4      ;SET UP TIMEOUT TRAP
5762 011266 005711      MOV      #340,6      ;LEVEL 7
5763 011270 000240      1$:      TST      (R1)      ;REFERENCE DEVICE REGISTER
5764 011272 062701 000002      NOP
5765 011276 005305      ADD      #2,R1      ;NEXT REGISTER
5766 011300 001372      DEC      R5      ;DEC REGISTER COUNT
5767 011302 000407      BNE      1$      ;BR IF NOT LAST REGISTER
5768 011304 062706 000004      BR      3$
5769 011310 010137 002632      2$:      ADD      #4,SP
5770 011314      MOV      R1,$GDADR
5771 (3) 011314 013700 002552      DODU      LOGDEV
5772 (3) 011320 104451      MOV      LOGDEV,R0
5773 011322 013737 002650 000004      TRAP      C$DODU
5774 011330 013737 002652 000006      3$:      MOV      SAVE4,4
5775 011336      MOV      SAVE6,6
5776 (3) 011336      ENDAUTO
5777 (3) 011336 104461      L10046:      TRAP      C$AUTO
  
```

CZDMPAO M8207 STATIC DIAG #1 MACY11 30A(1052) 17-JUL-79 14:39 E 8  
CZDMPA.P11 17-JUL-79 14:33 INITIALIZE SECTION PAGE 47-10

SEQ 0095

5775

```
5777 .SBTTL CLEANUP CODING SECTION
5778
5779 :////////////////////
5780 :// THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
5781 :// AT THE END OF EACH PASS.
5782 :////////////////////
5783
5784 011340 BGNCLN
(3) 011340 L$CLEAN::
5785 011340 BRESET
(3) 011340 104433 TRAP C$RESET
5786
5787 011342 ENDCLN
(3) 011342 L10047:
(3) 011342 104412 TRAP C$CLEAN
5788
5789
5790
5791
5792
```

5794  
5795  
5796  
5797  
5798  
5799  
5800  
5801  
(3)  
5802  
5803  
(3)  
5804  
(3)  
(3)  
5805  
5806  
5807  
5808  
5809

011344  
011344  
011344  
011344 104433  
011346  
011346 104453

.SBITL 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  
L10050:  
TRAP C\$DU

5811  
5812  
5813  
5814  
5815  
5816  
5817  
5818  
5819  
(3)  
5820  
(3)  
(3)  
5821  
5822  
5823  
5824  
5825  
5826

011350  
011350  
011350  
011350  
011350 104452

.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  
ENDAU  
L10051: TRAP C\$AU



```

5828 .SBTTL HARDWARE TESTS
5829
5830
5831
5832
5833 011352 BADHEAD
(2) :***** TEST 1 *****
5834 :*VERIFY THAT REFERENCING UNIBUS DEVICE REGISTERS
5835 :*DOES NOT CAUSE A TIME OUT TRAP
5836 011352 BADHEAD
(2) :***** TEST 1 *****
5837
5838 011352 BGN'TST
(3) 011352 T1::
5839 011352 013701 002716 MOV KMCSR,R1 ;R1 CONTAINS BASE M8200,4,7 ADDRESS
5840 011356 012705 000004 MOV #4,R5 ;4 REGISTERS TO BE TESTED
5841 011362 012737 011420 000004 MOV #2$,4 ;SET UP TIMEOUT TRAP
5842 011370 012737 000340 000006 MOV #340,6 ;LEVEL 7
5843 011376 005711 1$: TST (R1) ;REFERENCE DEVICE REGISTER
5844 011400 000240 NOP
5845 011402 ESCAPE TST
(3) 011402 104410 TRAP C$ESCAPE
(3) 011404 000072 .WORD L10052-.
5846 011406 062701 000002 ADD #2,R1 ;NEXT REGISTER
5847 011412 005305 DEC R5 ;DEC REGISTER COUNT
5848 011414 001370 BNE 1$ ;BR IF NOT LAST REGISTER
5849 011416 000417 BR 3$
5850 011420 062706 000004 2$: ADD #4,SP
5851 011424 010137 002632 MOV R1,$GDADR
5852 011430 ERROR 37 ;TIME-OUT ERROR
(5) 011446 104455 TRAP C$ERDF
(6) 011450 000045 .WORD 37
(6) 011452 005701 .WORD EM37
(6) 011454 010474 .WORD ERR37
5853
5854 011456 013737 002650 000004 3$: MOV SAVE4,4
5855 011464 013737 002652 000006 MOV SAVE6,6
5856 011472 ESCAPE TST
(3) 011472 104410 TRAP C$ESCAPE
(3) 011474 000002 .WORD L10052-.
5857 011476 ENDTST
(3) 011476 L10052: TRAP C$ETST
(3) 011476 104401
5858
5859 011500 BADHEAD
(2) :***** TEST 2 *****
5860 :*VERIFY THAT RUN CAN BE CLEARED
5861 011500 BADHEAD
(2) :***** TEST 2 *****
5862
5863 011500 BGN'TST
(3) 011500 T2::
5864 011500 MVINT
(1) 011500 013701 002716 MOV KMCSR,R1 ;GET DEVICE ADDRESS.
5865 011504 005011 CLR (R1) ;CLEAR KMCSR
5866 011506 005002 CLR R2 ;CLEAR 'EXPECTED'

```

```

5867 011510 011104      MOV      (R1),R4      ;PUT KMCSR IN 'FOUND'
5868 011512 001413      BEQ      1$          ;BR IF CLEARED
5869 011514              ERROR    26          ;ERROR KMCSR NOT CLEARED
(5) 011532 104455      TRAP     C$ERDF
(6) 011534 000032      .WORD    26
(6) 011536 005460      .WORD    EM26
(6) 011540 007552      .WORD    ERR26
5870 011542              1$:
5871 011542              ENDTST
(3) 011542              L10053:
(3) 011542 104401      TRAP     C$ETST
5872
5873 011544      BADHEAD
(2)
5874      ;***** TEST 3 *****
5875      ;*UNIBUS REGISTER WORD DUAL ADDRESSING TEST
5876      ;*LOAD ALL REGISTERS WITH INCREMENTING PATTERN
5877      ;*READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING
5878      ;*THE SEQUENCE:
5879      ;*
5880      ;* 1. CLEAR REGISTER
5881      ;* 2. WRITE PATTERN
5882      ;* 3. VERIFY PATTERN
5883      ;* 4. DO ALL 4 REGISTERS
5884      ;* 5. READ ALL BACK IF ERRORS,
5885      ;*    DUAL ADDRESS PROBLEM.
5886      ;*
5887      ;* 1 IN REG 0
5888      ;* 2 IN REG 2
5889      ;* 3 IN REG 4
5890      ;* 4 IN REG 6
5891 011544      BADHEAD
(2)
5892      ;***** TEST 3 *****
5893
5894      BGNST
(3) 011544      T3::
5895      MYINT
(1) 011544 013701 002716      MOV      KMCSR,R1      ;GET DEVICE ADDRESS.
5896      ;R1 CONTAINS BASE M8200.4,7 ADDRESS
5897      MSTCLR
5898      JSR      R5,MSTCLR      ;MASTER CLEAR M8200.4,7
5899      MOV      #1,R2          ;CLEAR M8200.4,7
5900      BGNSEG
5901      TRAP     C$BSEG
5902      CLR      (R1)          ;START PATTERN AT 1
(5) 011612 104455      1$:
(6) 011614 000032      CLR      (R1)          ;CLEAR REGISTER
(6) 011616 005460      MOV      R2,(R1)      ;WRITE M8200.4,7 REGISTER WITH PATTERN
(6) 011620 007552      MOV      (R1),R4      ;READ M8200.4,7 REGISTER INTO 'FOUND'
5903 011622              CMP      R2,R4
(3) 011622 104410      BEQ      2$          ;IS DATA CORRECT
(3) 011624 000014      ERROR    26          ;BR IS YES
5904 011626 005721      TRAP     C$ERDF      ;DATA ERROR
(5) 011612 104455      .WORD    26
(6) 011614 000032      .WORD    EM26
(6) 011616 005460      .WORD    ERR26
5903 011622              2$:
(3) 011622 104410      ESCAPE   SEG
(3) 011624 000014      TRAP     C$ESCAPE
5904 011626 005721      .WORD    10000$-
TST      (R1)+          ;NEXT REGISTER

```

```

5905 011630 005202          INC      R2          ;INCREMENT DATA PATTERN
5906 011632 022702 000005    CMP      #5,R2      ;LAST REGISTER?
5907 011636 001351          BNE      1$          ;BR IF NO
5908 011640          ENDSEG
      (3) 011640          10000$:
      (3) 011640 104405    TRAP      C$ESEG
5909 011642 013701 002716    MOV      KMCSR,R1      ;BASE M8200,4,7 ADDRESS TO R1
5910 011646 012702 000001    MOV      #1,R2      ;RESTART PATTERN AT 1
5911 011652          BGNSEG
      (3) 011652 104404    TRAP      C$BSEG
5912 011654          3$:
5913 011654 011104    MOV      (R1),R4      ;READ COMM. MICR-PROCESSOR FAMILY REGISTER INTO 'FOUND'
5914 011656 020204    CMP      R2,R4      ;IS DATA CORRECT
5915 011660 001413    BEQ      4$          ;BR IF YES
5916 011662          ERROR      2          ;DUAL ADDRESSING ERROR
      (5) 011700 104455    TRAP      C$ERDF
      (6) 011702 000002    .WORD      2
      (6) 011704 004273    .WORD      EM2
      (6) 011706 006116    .WORD      ERR2
5917 011710          4$:
      (3) 011710 104410    ESCAPE      SEG
      (3) 011712 000014    TRAP      C$ESCAPE
5918 011714 005721    .WORD      10001$-
5919 011716 005202    TST      (R1)+      ;NEXT REGISTER
5920 011720 022702 000005    INC      R2          ;INCREMENT PATTERN
5921 011724 001353    CMP      #5,R2      ;LAST REGISTER?
5922 011726          BNE      3$          ;BR IF NO
      (3) 011726          10001$:
      (3) 011726 104405    TRAP      C$ESEG
5923 011730          ENDTST
      (3) 011730          L10054:
      (3) 011730 104401    TRAP      C$ETST
5924
5925 011732          BADHEAD
      (2)
5926          ;***** TEST 4 *****
5927          ;*CONTROL STATUS REGISTER WRITE/READ TEST
5928          ;*FLOAT A ONE THROUGH BSEL 0
5929 011732          ;*CLEAR BIT0, VERIFY BIT0 WAS CLEARED
      (2)          BADHEAD
5930          ;***** TEST 4 *****
5931 011732          BGNST
      (3) 011732          T4::
5932 011732          MSTCLR
      (1) 011732 004537 003142    JSR      R5,.MSTCLR      ;MASTER CLEAR M8200,4,7
5933 011736 005037 002624    CLR      MRO          ;CLEAR M8200,4,7
5934 011742 012702 000001    MOV      #BIT0,R2      ;INDICATE BSEL0
5935 011746          BGNSEG
      (3) 011746 104404    TRAP      C$BSEG
5936 011750 013701 002716    1$:
5937 011754 010237 002636    MOV      KMCSR,R1      ;PUT REGISTER ADDRESS IN R1
5938 011760 013711 002636    MOV      R2,$GDDAT
5939 011764 011104          MOV      $GDDAT,(R1)      ;WRITE BIT 0
5940 011766 023704 002636    MOV      (R1),R4      ;READ CONTROL STATUS REGISTER
5941 011772 001411    CMP      $GDDAT,R4      ;IS DATA CORRECT
5942 011774          BEQ      2$          ;BR IF YES
          ERROR      27,YES      ;DATA ERROR

```

(5)	012006	104455		TRAP	C\$ERDF	
(6)	012010	000033		.WORD	27	
(6)	012012	005511		.WORD	EM27	
(6)	012014	007630		.WORD	ERR27	
5943	012016		2\$:	ESCAPE	SEG	
(3)	012016	104410		TRAP	C\$ESCAPE	
(3)	012020	000052		.WORD	10000\$-	
5944	012022	040211	3\$:	BIC	R2,(R1)	;CLEAR BSELO
5945	012024	005037		CLR	\$GDDAT	;CLEAR 'EXPECTED'
5946	012030	011104		MOV	(R1),R4	;READ CONTROL STATUS REGISTER
5947	012032	001413		BEQ	4\$	;BR IF ZERO
5948	012034			ERROR	2	;DATA ERROR BSEL NOT CLEARED
(5)	012052	104455		TRAP	C\$ERDF	
(6)	012054	000002		.WORD	2	
(6)	012056	004273		.WORD	EM2	
(6)	012060	006116		.WORD	ERR2	
5949	012062		4\$:	ESCAPE	SEG	
(3)	012062	104410		TRAP	C\$ESCAPE	
(3)	012064	000006		.WORD	10000\$-	
5950	012066	106302		ASLB	R2	
5951	012070	001327		BNE	1\$	
5952	012072			ENDSEG		
(3)	012072		10000\$:			
(3)	012072	104405		TRAP	C\$ESEG	
5953	012074		ENDTST			
(3)	012074		L10055:			
(3)	012074	104401		TRAP	C\$ETST	
5954						
5955						

```

5957
5958
5959
5960
5961
5962 012076      BADHEAD
(2)              :***** TEST 5 *****
5963              :*CONTROL STATUS REGISTER WRITE/READ TEST
5964              :*SET BIT9, VERIFY BIT9 WAS SET
5965              :*CLEAR BIT9, VERIFY BIT9 WAS CLEARED
5966 012076      BADHEAD
(2)              :***** TEST 5 *****
5967
5968 012076      BGNST
(3) 012076      T5::
5969 012076      MSTCLR          ;MASTER CLEAR M8200,4,7
(1) 012076      JSR          R5,,MSTCLR          ;CLEAR M8200,4,7
5970 012102      BGNSEG
(3) 012102      TRAP          C$BSEG
5971 012104      MOV          KMCSR,R1          ;PUT REGISTER ADDRESS IN R1
5972 012110      MOV          #BIT9,R2          ;PUT DATA IN 'EXPECTED'
5973 012114      MOV          R2,(R1)          ;WRITE BIT 9
5974 012116      MOV          (R1),R4          ;READ CONTROL STATUS REGISTER
5975 012120      MOV          R2,R4          ;IS DATA CORRECT
5976 012122      CMP          R2,R4          ;BR IF YES
5977 012124      BEQ          2$          ;DATA ERROR
(5) 012142      104455
(6) 012144      000032
(6) 012146      005460
(6) 012150      007552
5978 012152      2$: ESCAPE SEG
(3) 012152      104410
(3) 012154      000002
5979 012156      TRAP          C$ERDF
(3) 012156      .WORD          26
(3) 012156      .WORD          EM26
(3) 012156      .WORD          ERR26
5980 012160      10000$: ENDSEG
(3) 012160      104405
5981 012162      TRAP          C$ESEG
(3) 012162      BGNSEG
5982 012166      005002      001000      3$: BIC          #BIT9,(R1)          ;CLEAR BIT 9
5983 012170      011104      CLR          R2          ;CLEAR 'EXPECTED'
5984 012172      001416      MOV          (R1),R4          ;READ CONTROL STATUS REGISTER
5985 012174      BEQ          4$          ;BR IF ZERO
(5) 012212      104455
(6) 012214      000032
(6) 012216      005460
(6) 012220      007552
5986 012222      10001$: TRAP          C$ERDF
(3) 012222      104410
(3) 012224      000002
5987 012226      .WORD          26
(3) 012226      .WORD          EM26
(3) 012226      .WORD          ERR26
5988 012230      ESCAPE SEG
5989 012230      TRAP          C$ESEG
(3) 012230      .WORD          10001$-
5988 012230      10001$: TRAP          C$ESEG
5989 012230      4$:
(3) 012230      ENDTST
L10056:

```

```

(3) 012230 104401 TRAP C$ETST
5990
5991 012232 BADHEAD
(2)
5992 :***** TEST 6 *****
5993 :*CONTROL STATUS REGISTER WRITE/READ TEST
5994 :*SET BIT11, VERIFY BIT11 WAS SET
5995 :*CLEAR BIT11, VERIFY BIT11 WAS CLEARED
5996 BADHEAD
(2)
5997 :***** TEST 6 *****
5998
(3) 012232 BGNST
5999 012232 T6::
(1) 012232 004537 003142 MSTCLR JSR R5, MSTCLR ;MASTER CLEAR M8200,4,7
5999 012236 BGNSEG ;CLEAR M8200,4,7
(3) 012236 104404 TRAP C$BSEG
6000 012240 013701 002716 1$: MOV KMCSR, R1 ;PUT REGISTER ADDRESS IN R1
6001 012244 012702 004000 MOV #BIT11, R2 ;PUT DATA IN 'EXPECTED'
6002 012250 010211 MOV R2, (R1) ;WRITE BIT 11
6003 012252 011104 MOV (R1), R4 ;READ CONTRCL STATUS REGISTER
6004 012254 020204 CMP R2, R4 ;IS DATA CORRECT
6005 012256 001413 BEQ 2$ ;BR IF YES
6006 012260 ERROR 26 ;DATA ERROR
(5) 012276 104455 TRAP C$ERDF
(6) 012300 000032 .WORD 26
(6) 012302 005460 .WORD EM26
(6) 012304 007552 .WORD ERR26
6007 012306 2$: ESCAPE SEG
(3) 012306 104410 TRAP C$ESCAPE
(3) 012310 000002 .WORD 10000$-
6008 012312 ENDSEG
(3) 012312 10000$: TRAP C$ESEG
6009 012314 BGNSEG
(3) 012314 104404 TRAP C$BSEG
6010 012316 042711 004000 3$: BIC #BIT11, (R1) ;CLEAR BIT 11
6011 012322 005002 CLR R2 ;CLEAR 'EXPECTED'
6012 012324 011104 MOV (R1), R4 ;READ CONTROL STATUS REGISTER
6013 012326 001414 BEQ 4$ ;BR IF ZERO
6014 012330 ERROR 26 ;DATA ERROR BIT11 NOT CLEARED
(5) 012346 104455 TRAP C$ERDF
(6) 012350 000032 .WORD 26
(6) 012352 005460 .WORD EM26
(6) 012354 007552 .WORD ERR26
6015 012356 ENDSEG
(3) 012356 10001$: TRAP C$ESEG
6016 012360 4$:
6017 012360 ENDTST
(3) 012360 L10057: TRAP C$ETST
(3) 012360 104401
6018 BADHEAD
6019 012362 :***** TEST 7 *****
(2) :*CONTROL STATUS REGISTER WRITE/READ TEST
6020 :*SET BIT12, VERIFY BIT12 WAS SET
6021

```

```

6022                                     ;*CLEAR BIT12, VERIFY BIT12 WAS CLEARED
6023 012362 BADHEAD
(2)                                     ;***** TEST 7 *****
6024
6025 012362 BGNTST
(3) 012362 T7::
- 0 012362 MSTCLR ;MASTER CLEAR M8200,4,7
(1) 012362 004137 003142 JSR R5,,MSTCLR ;CLEAR M8200,4,7
6027 012366 BGNSEG
(3) 012366 104404 TRAP C$BSEG
6028 012370 013701 002716 1$: MOV KMCSR,R1 ;PUT REGISTER ADDRESS IN R1
6029 012374 012702 010000 MOV #BIT12,R2 ;PUT DATA IN 'EXPECTED'
6030 012400 010211 MOV R2,(R1) ;WRITE BIT 12
6031 012402 011104 MOV (R1),R4 ;READ CONTROL STATUS REGISTER
6032 012404 020204 CMP R2,R4 ;IS DATA CORRECT
6033 012406 001413 BEQ 2$ ;BR IF YES
6034 012410 ERROR 26 ;DATA ERROR
(5) 012426 104455 TRAP C$ERDF
(6) 012430 000032 .WORD 26
(6) 012432 005460 .WORD EM26
(6) 012434 007552 .WORD ERR26
6035 012436 2$: ESCAPE SEG
(3) 012436 104410 TRAP C$ESCAPE
(3) 012440 000002 .WORD 10000$-
6036 012442 ENDSEG
(3) 012442 10000$:
(3) 012442 104405 TRAP C$ESEG
6037 012444 BGNSEG
(3) 012444 104404 TRAP C$BSEG
6038 012446 042711 010000 3$: BIC #BIT12,(R1) ;CLEAR BIT 12
6039 012452 005002 CLR R2 ;CLEAR 'EXPECTED'
6040 012454 011104 MOV (R1),R4 ;READ CONTROL STATUS REGISTER
6041 012456 001414 BEQ 4$ ;BR IF ZERO
6042 012460 ERROR 26 ;DATA ERROR BIT12 NOT CLEARED
(5) 012476 104455 TRAP C$ERDF
(6) 012500 000032 .WORD 26
(6) 012502 005460 .WORD EM26
(6) 012504 007552 .WORD ERR26
6043 012506 ENDSEG
(3) 012506 10001$:
(3) 012506 104405 TRAP C$ESEG
6044 012510 4$:
6045 012510 ENDTST
(3) 012510 L10060:
(3) 012510 104401 TRAP C$ETST
6046
6047 012512 BADHEAD
(2) ;***** TEST 8 *****
6048 ;*CONTROL OUT REGISTER WRITE/READ TEST
6049 ;*FLOAT A ONE THROUGH SEL2
6050 012512 BADHEAD
(2) ;***** TEST 8 *****
6051
6052 012512 BGNTST
(3) 012512 T8::
6053 012512 MSTCLR ;MASTER CLEAR M8200,4,7

```

(1)	012512	004537	003142		JSR	R5, .MSTCLR	;CLEAR M8200,4,7
6054	012516	012737	000002	002624	MOV	#2, MRO	
6055	012524	012702	000001		MOV	#1, R2	
6056	012530				BGNSEG		
(3)	012530	104404			TRAP	C\$BSEG	
6057							
6058	012532	013701	002722		1\$: MOV	KMCTL, R1	;PUT REGISTER ADDRESS IN R1
6059	012536	010237	002636		MOV	R2, \$GDDAT	;PUT DATA IN 'EXPECTED'
6060	012542	013711	002636		MOV	\$GDDAT, (R1)	;WRITE BIT 0
6061	012546	011104			MOV	(R1), R4	;READ CONTROL OUT REGISTER
6062	012550	023704	002636		CMP	\$GDDAT, R4	;IS DATA CORRECT
6063	012554	001411			BEQ	2\$	;BR IF YES
6064	012556				ERROR	27, YES	;DATA ERROR
(5)	012570	104455			TRAP	C\$ERDF	
(6)	012572	000033			.WORD	27	
(6)	012574	005511			.WORD	EM27	
(6)	012576	007630			.WORD	ERR27	
6065	012600				2\$: ESCAPE	SEG	
(3)	012600	104410			TRAP	C\$ESCAPE	
(3)	012602	000046			.WORD	10000\$-	
6066	012604	040211			3\$: BIC	R2, (R1)	;CLEAR BIT
6067	012606	005037	002636		CLR	\$GDDAT	;CLEAR 'EXPECTED'
6068	012612	011104			MOV	(R1), R4	;READ CONTROL OUT REGISTER
6069	012614	001411			BEQ	4\$	;BR IF ZERO
6070	012616				ERROR	27, YES	;DATA ERROR BIT0 NOT CLEARED
(5)	012630	104455			TRAP	C\$ERDF	
(6)	012632	000033			.WORD	27	
(6)	012634	005511			.WORD	EM27	
(6)	012636	007630			.WORD	ERR27	
6071	012640				4\$: ESCAPEE	SEG	
(3)	012640	104410			TRAP	C\$ESCAPE	
(3)	012642	000006			.WORD	10000\$-	
6072	012644	006302			ASL	R2	
6073	012646	001331			BNE	1\$	
6074	012650				ENDSEG		
(3)	012650				10000\$: TRAP	C\$ESEG	
(3)	012650	104405					
6075	012652				ENDTST		
(3)	012652				L10061: TRAP	C\$ETST	
(3)	012652	104401					
6076							
6077							
6078							
6079							
6080							
6081							
6082	012654				BADHEAD		
(2)					:***** TEST 9 *****		
6083					:*PORT4 REGISTER WRITE/READ TEST		
6084					:*FLOAT A ONE THROUGH PORT4 REGISTER		
6085					:*FLOAT A ZERO THROUGH PORT4 REGISTER		
6086	012654				BADHEAD		
(2)					:***** TEST 9 *****		
6087							
6088							
6089	012654				BGNTST		



```

(3) 012654
6090 012654 012737 000004 002624 T9:: MOV #4,MRO
6091 012662 MSTCLR ;MASTER CLEAR M8200,4,7
(1) 012662 004537 003142 JSR R5,.MSTCLR ;CLEAR M8200,4,7
6092 012666 013701 002724 MOV KMP04,R1 ;PUT REGISTER ADDRESS IN R1
6093 012672 012702 000001 MOV #1,R2 ;START WITH BIT0
6094 012676 BGNSEG
(3) 012676 104404 TRAP C$BSEG
6095 012700 64$: MOV R2,(R1) ;WRITE PORT4 REGISTER
6096 012700 010211 MOV (R1),R4 ;READ PORT4 REGISTER
6097 012702 011104 CMP R2,R4 ;COMPARE EXPECTED AND FOUND
6098 012704 020204 BEQ 65$ ;BR IF OK
6099 012706 001413 ERROR 27 ;WRITE/READ ERROR
6100 012710 TRAP C$ERDF
(5) 012726 104455 .WORD 27
(6) 012730 000033 .WORD EM27
(6) 012732 005511 .WORD ERR27
(6) 012734 007630 65$: ESCAPE SEG
6101 012736 TRAP C$ESCAPE
(3) 012736 104410 .WORD 10000$-
(3) 012740 000010 CLC ;CLEAR CARRY
6102 012742 000241 ROL R2 ;SHIFT TO NEXT BIT
6103 012744 006102 BNE 64$ ;BR IF NOT DONE YET?
6104 012746 001354 ENDSEG
6105 012750 10000$: TRAP C$ESEG
(3) 012750 104405 MOV #1,R2 ;START WITH BIT0
6106 012752 012702 000001 BGNSEG
6107 012756 104404 TRAP C$BSEG
6108 012760 66$: COM R2 ;CHANGE TO A FLOATING ZERO
6109 012760 005102 MOV R2,(R1) ;WRITE PORT4 REGISTER
6110 012762 010211 MOV (R1),R4
6111 012764 011104 CMP R2,R4 ;COMPARE EXPECTED AND FOUND
6112 012766 020204 BEQ 67$ ;BR IF OK
6113 012770 001413 ERROR 27 ;WRITE/READ ERROR
6114 012772 TRAP C$ERDF
(5) 013010 104455 .WORD 27
(6) 013012 000033 .WORD EM27
(6) 013014 005511 .WORD ERR27
(6) 013016 007630 67$: ESCAPE SEG
6115 013020 TRAP C$ESCAPE
(3) 013020 104410 .WORD 10001$-
(3) 013022 000012 COM R2 ;CHANGE BACK TO A FLOATING ONE
6116 013024 005102 CLC ;CLEAR CARRY
6117 013026 000241 ROL R2 ;SHIFT TO NEXT BIT
6118 013030 006102 BNE 66$ ;BR IF NOT DONE YET?
6119 013032 001352 ENDSEG
6120 013034 10001$: TRAP C$ESEG
(3) 013034 104405
6121 013036 ENDTST
(3) 013036 L10062: TRAP C$ETST
(3) 013036 104401
6122 013040
6123 013040 BADHEAD
  
```

```

(2)
6124
6125
6126
6127 013040
(2)
6128
6129 013040
(3) 013040
6130 013040 012737 000006 002624
6131 013046
(1) 013046 004537 003142
6132 013052 013701 002726
6133 013056 012702 000001
6134 013062
(3) 013062 104404
6135 013064
6136 013064 010211
6137 013066 011104
6138 013070 020204
6139 013072 001413
6140 013074
(5) 013112 104455
(6) 013114 000033
(6) 013116 005511
(6) 013120 007630
6141 013122
(3) 013122 104410
(3) 013124 000010
6142 013126 000241
6143 013130 006105
6144 013132 001354
6145 013134
(3) 013134
(3) 013134 104405
6146 013136 012702 000001
6147 013142
(3) 013142 104404
6148 013144
6149 013144 005102
6150
6151 013146 010211
6152 013150 011104
6153 013152 020204
6154 013154 001413
6155 013156
(5) 013174 104455
(6) 013176 000033
(6) 013200 005511
(6) 013202 007630
6156 013204
(3) 013204 104410
(3) 013206 000012
6157 013210 005102
6158 013212 000241
6159 013214 006102

***** TEST 10 *****
;PORT6 REGISTER WRITE/READ TEST
;FLOAT A ONE THROUGH PORT6 REGISTER
;FLOAT A ZERO THROUGH PORT6 REGISTER
BADHEAD
***** TEST 10 *****

BGNST
T10::
MOV #6,MRO
MSTCLR ;MASTER CLEAR M8200,4,7
JSR R5,.MSTCLR ;CLEAR M8200,4,7
MOV KMP06,R1 ;PUT REGISTER ADDRESS IN R1
MOV #1,R2 ;START WITH BIT0
BGNSEG
TRAP C$BSEG
64$:
MOV R2,(R1) ;WRITE PORT6 REGISTER
MOV (R1),R4 ;READ PORT6 REGISTER
CMP R2,R4 ;COMPARE EXPECTED AND FOUND
BEQ 65$ ;BR IF OK
ERROR 27 ;WRITE/READ ERROR
TRAP C$ERDF
.WORD 27
.WORD EM27
.WORD ERR27
65$:
ESCAPE SEG
TRAP C$ESCAPE
.WORD 10000$-.
CLC ;CLEAR CARRY
ROL R5 ;SHIFT TO NEXT BIT
BNE 64$ ;BR IF NOT DONE YET?
ENDSEG
10000$:
TRAP C$ESEG
MOV #1,R2 ;START WITH BIT0
BGNSEG
TRAP C$BSEG
66$:
COM R2 ;CHANGE TO A FLOATING ZERO
MOV R2,(R1) ;WRITE PORT6 REGISTER
MOV (R1),R4 ;READ PORT6 REGISTER
CMP R2,R4 ;COMPARE EXPECTED AND FOUND
BEQ 67$ ;BR IF OK
ERROR 27 ;WRITE/READ ERROR
TRAP C$ERDF
.WORD 27
.WORD EM27
.WORD ERR27
67$:
ESCAPE SEG
TRAP C$ESCAPE
.WORD 10001$-.
COM R2 ;CHANGE BACK TO A FLOATING ONE
CLC ;CLEAR CARRY
ROL R2 ;SHIFT TO NEXT BIT

```

```

6160 013216 001352      BNE      66$          ;BR IF NOT DONE YET?
6161 013220      ENDSEG
(3) 013220      10001$:
(3) 013220 104405      TRAP      C$ESEG
6162 013222      .ENDTST
(3) 013222      L10063:
(3) 013222 104401      TRAP      C$ETST
6163
6164 013224      BADHEAD
(2)
6165      ;***** TEST 11 *****
6166      ;*UNIBUS REGISTER BYTE DUAL ADDRESSING TEST
6167      ;*LOAD ALL REGISTERS WITH INCREMENTING PATTERN
6168 013224      ;*READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING
(2)      BADHEAD
6169      ;***** TEST 11 *****
6170 013224      BGNST
(3) 013224      T11::
6171 013224      MYINT
(1) 013224 013701 002716      MOV      KMCSR,R1          ;GET DEVICE ADDRESS.
6172 013230      MSTCLR          ;MASTER CLEAR M8200,4,7
(1) 013230 004537 003142      JSR      R5,,MSTCLR          ;CLEAR M8200,4,7
6173 013234 012702 000001      MOV      #1,R2          ;START PATTERN AT 1
6174 013240      BGNSEG
(3) 013240 104404      TRAP      C$BSEG
6175 013242 105011      1$:      CLRB      (R1)          ;CLEAR REGISTER
6176 013244 110211      MOV      R2,(R1)          ;WRITE M8200,4,7 REGISTER WITH PATTERN
6177 013246 111104      MOV      (R1),R4          ;READ M8200,4,7 REGISTER INTO 'FOUND'
6178 013250 120204      CMP      R2,R4          ;IS DATA CORRECT
6179 013252 001413      BEQ      2$          ;BR IF YES
6180 013254      ERROR      2          ;DATA ERROR
(5) 013272 104455      TRAP      C$ERDF
(6) 013274 000002      .WORD      2
(6) 013276 004273      .WORD      EM2
(6) 013300 006116      .WORD      ERR2
6181 013302      2$:      ESCAPE      SEG
(3) 013302 104410      TRAP      C$ESCAPE
(3) 013304 000024      .WORD      10000$-
6182 013306 105721      TSTB      (R1)+          ;NEXT REGISTER
6183 013310 005202      INC      R2          ;INCREMENT DATA PATTERN
6184 013312 022702 000011      CMP      #11,R2          ;LAST REGISTER?
6185 013316 001351      BNE      1$          ;BR IF NO
6186 013320 013701 002716      MOV      KMCSR,R1          ;BASE M8200,4,7 ADDRESS TO R1
6187 013324 012702 000001      MOV      #1,R2          ;RESTART PATTERN AT 1
6188 013330      ENDSEG
(3) 013330      10000$:
(3) 013330 104405      TRAP      C$ESEG
6189 013332      BGNSEG
(3) 013332 104404      TRAP      C$BSEG
6190 013334      3$:
6191 013334 111104      MOV      (R1),R4          ;READ COMM.MICRO-PROCESSOR FAMILY REGISTER INTO 'FOUND'
6192 013336 120204      CMP      R2,R4          ;IS DATA CORRECT
6193 013340 001413      BEQ      4$          ;BR IF YES
6194 013342      ERROR      2          ;DUAL ADDRESSING ERROR
(5) 013360 104455      TRAP      C$ERDF
(6) 013362 000002      .WORD      2

```

(6)	013364	004273				
(6)	013366	006116				
6195	013370			4\$:	.WORD EM2	
(3)	013370	104410			.WORD ERR2	
(3)	013372	000014			ESCAPE SEG	
6196	013374	105721			TRAP C\$ESCAPE	
6197	013376	005202			.WORD 10001\$-	
6198	013400	022702	000011		TSTB (R1)+	;NEXT REGISTER
6199	013404	001353			INC R2	;INCREMENT PATTERN
6200	013406				CMP #11,R2	;LAST REGISTER?
(3)	013406				BNE 3\$	;BR IF NO
(3)	013406	104405		10001\$:	ENDSEG	
6201	013410			ENDTST	TRAP C\$ESEG	
(3)	013410			L10064:		
(3)	013410	104401			TRAP C\$ETST	
6202						
6203	013412				BADHEAD	
(2)					***** TEST 12 *****	
6204					*MAINTENANCE INSTRUCTION REGISTER TEST	
6205					*VERIFY THAT THE MAINT IR CAN BE WRITTEN TO ALL ZEROS'	
6206					*AND ALL ONES'. VERIFY THAT IT IS CLEARED ON A BUS RESET.	
6207	013412				BADHEAD	
(2)					***** TEST 12 *****	
6208						
6209	013412			BGNTST		
(3)	013412			T12::		
6210						;R1 CONTAINS BASE M8200,4,7 ADDRESS
6211	013412				MSTCLR	;MASTER CLEAR M8200,4,7
(1)	013412	004537	003142		JSR R5,,MSTCLR	;CLEAR M8200,4,7
6212	013416				MYINT	
(1)	013416	013701	002716		MOV KMCSR,R1	;GET DEVICE ADDRESS.
6213	013422				BGNSEG	
(3)	013422	104404			TRAP C\$BSEG	
6214	013424	012711	003000		MOV #BIT9:BIT10,(R1)	;SEL6 IS NOW THE IR
6215	013430	005002			CLR R2	;PUT 'EXPECTED' IN \$GDDAT
6216	013432	010261	000006	1\$:	MOV R2,6(R1)	;CLEAR THE IR
6217	013436	016104	000006		MOV 6(R1),R4	;READ THE IR
6218	013442	020204			CMP R2,R4	;IS IT CLEARED?
6219	013444	001413			BEQ 2\$	;BR IF YES
6220	013446				ERROR 26	;ERROR IR IS NOT CLEAR
(5)	013464	104455			TRAP C\$ERDF	
(6)	013466	000032			.WORD 26	
(6)	013470	005460			.WORD EM26	
(6)	013472	007552			.WORD ERR26	
6221	013474			2\$:	ESCAPE SEG	
(3)	013474	104410			TRAP C\$ESCAPE	
(3)	013476	000002			.WORD 10000\$-	
6222	013500				ENDSEG	
(3)	013500			10000\$:		
(3)	013500	104405			TRAP C\$ESEG	
6223	013502	012702	177777		MOV #-1,R2	;PUT 'EXPECTED' IN \$GDDAT
6224	013506				BGNSEG	
(3)	013506	104404			TRAP C\$BSEG	
6225	013510	010261	000006	3\$:	MOV R2,6(R1)	;WRITE ALL ONES TO THE IR
6226	013514	016104	000006		MOV 6(R1),R4	;READ THE IR
6227	013520	020204			CMP R2,R4	;IS IT ALL ONES?

6228	013522	001413		BEQ	4\$		
6229	013524			ERROR	26		;BR IF YES
(5)	013542	104455		TRAP	C\$ERDF		;ERROR IR IS NOT - ALL ONES
(6)	013544	000032		.WORD	26		
(6)	013546	005460		.WORD	EM26		
(6)	013550	007552		.WORD	ERR26		
6230	013552		4\$:	ESCAPE	SEG		
(3)	013552	104410		TRAP	C\$ESCAPE		
(3)	013554	000002		.WORD	10001\$-		
6231	013556			ENDSEG			
(3)	013556		10001\$:				
(3)	013556	104405		TRAP	C\$ESEG		
6232	013560		ENDTST				
(3)	013560		L10065:				
(3)	013560	104401		TRAP	C\$ETST		
6233							
6234							

```

6236
6237 013562          BADHEAD
(2)                ;***** TEST 13 *****
6238                ;*MICRO PROCESSOR TEST
6239                ;*LOAD KMP06 WITH A MICRO-PROCESSOR INSTRUCTION, CLOCK IT
6240                ;*VERIFY INSTRUCTION EXECUTED PROPERLY
6241                ;*INSTRUCTION SHOULD MOVE IBUS*4 TO IBUS*5, IBUS*4 IS ALL 1'S
6242                ;*AND IBUS*5 IS ALL 0'S. RESULT SHOULD BE ALL 1'S IN SEL4
6243 013562          BADHEAD
(2)                ;***** TEST 13 *****
6244
6245 013562          BGNTST
(3) 013562          T13::
6246 013562          MYINT
(1) 013562 013701 002716  MOV KMCSR,R1          ;GET DEVICE ADDRESS.
6247 013562          MSTCLR
(1) 013562 004537 003142  JSR R5,MSTCLR          ;CLEAR M8200,4,7
6248 013572 012761 000377 000004  MOV #377,4(R1)      ;PORT4 HI BYTE=1'S
6249 013600 012711 001000          MOV #BIT9,(R1)        ;SET ROMI
6250 013604 012761 121105 000006  MOV #121105,6(R1)    ;INSTR TO PORT 6.
6251 013612 052711 001400          BIS #BIT8!BIT9,(R1)   ;CLK INSTR.
6252 013616 000240          NOP
6253 013620 012702 177777          MOV #-1,R2          ;EXPECT ALL ONES.
6254 013624 116104 000004          MOV 4(R1),R4          ;READ FOUND.
6255 013630 020204          CMP R2,R4          ;DATA CORRECT?
6256 013632 001413          BEQ 1$
6257 013634          ERROR 28
(5) 013652 104455          TRAP C$ERDF
(6) 013654 000034          .WORD 28
(6) 013656 005540          .WORD EM28
(6) 013660 007712          .WORD ERR28
6258
6259 013662          1$:  ESCAPE TST
(3) 013662 104410          TRAP C$ESCAPE
(3) 013664 000002          .WORD L10066-.
6260
6261 013666          ENDTST
(3) 013666          L10066:
(3) 013666 104401          TRAP C$ETST
6262
6263 013670          BADHEAD
(2)                ;***** TEST 14 *****
6264                ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
6265                ;*FLOAT A 1 THROUGH IBUS* REGISTER 0
6266                ;*FLOAT A 0 THROUGH IBUS* REGISTER 0
6267 013670          BADHEAD
(2)                ;***** TEST 14 *****
6268
6269 013670          BGNTST
(3) 013670          T14::
6270 013670          MSTCLR
(1) 013670 004537 003142          JSR R5,MSTCLR          ;MASTER CLEAR M8200,4,7
6271 013674 012737 000000 002624  MOV #0,MRO          ;CLEAR M8200,4,7
6272 013702 012705 000001          MOV #1,R5          ;SAVE REGISTER ADDRESS FOR TYPEOUT
6273                ;STAR1 WITH BIT 0
6274 013706          MYINT

```

(1)	013706	013701	002716	MOV	KMCSR,R1	;GET DEVICE ADDRESS.
6275	013712			BGNSEG		
(3)	013712	104404		TRAP	C\$BSEG	
6276	013714			64\$:		
6277	013714	010561	000004	MOV	R5,4(R1)	;PUT PATTERN INTO PORT4
6278	013720			ROMCLK		;NEXT WORD IS INSTRUCTION, BBN
(1)	013720	004537	003230	JSR	R5,..ROMCLK	;CLOCK INSTRUCTION
6279	013724	121100		121100		;MOV DATA TO IBUS* REGISTER 0
6280	013726			ROMCLK		;NEXT WORD IS INSTRUCTION, BBN
(1)	013726	004537	003230	JSR	R5,..ROMCLK	;CLOCK INSTRUCTION
6281	013732	121005		121005		;READ FROM IBUS* REGISTER 0
6282	013734	116104	000005	MOVB	5(R1),R4	;PUT 'FOUND' INTO R4
6283	013740	120504		CMPB	R5,R4	;DATA CORRECT?
6284	013742	001414		BEQ	65\$	;BR IF YES
6285	013744			BERROR	27	;ERROR
(5)	013764	104455		TRAP	C\$ERDF	
(6)	013766	000033		.WORD	27	
(6)	013770	005511		.WORD	EM27	
(6)	013772	007630		.WORD	ERR27	
6286	013774			65\$:	ESCAPE	SEG
(3)	013774	104410		TRAP	C\$ESCAPE	
(3)	013776	000010		.WORD	10000\$-	
6287	014000	000241		CLC		;CLEAR CARRY
6288	014002	106105		ROLB	R5	;SHIFT BIT IN R5
6289	014004	001343		BNE	64\$	;IF R2=0 THEN DONE
6290	014006			ENDSEG		
(3)	014006			10000\$:		
(3)	014006	104405		TRAP	C\$ESEG	
6291	014010	012705	000001	MOV	#1,R5	;START WITH BIT 0
6292				;69\$:	COM	R5
6293	014014			BGNSEG		;CHANGE TO FLOATING ZERO
(3)	014014	104404		TRAP	C\$BSEG	
6294	014016			67\$:		
6295	014016	005105		COM	R5	
6296	014020	010561	000004	MOV	R5,4(R1)	;PUT PATTERN INTO PORT4
6297	014024			ROMCLK		;NEXT WORD IS INSTRUCTION, BBN
(1)	014024	004537	003230	JSR	R5,..ROMCLK	;CLOCK INSTRUCTION
6298	014030	121100		121100		;MOV DATA TO IBUS* REGISTER 0
6299	014032			ROMCLK		;NEXT WORD IS INSTRUCTION, BBN
(1)	014032	004537	003230	JSR	R5,..ROMCLK	;CLOCK INSTRUCTION
6300	014036	121005		121005		;READ FROM IBUS* REGISTER 0
6301	014040	116104	000005	MOVB	5(R1),R4	;PUT 'FOUND' INTO R4
6302	014044	120504		CMPB	R5,R4	;DATA CORRECT?
6303	014046	001414		BEQ	68\$	;BR IF YES
6304	014050			BERROR	27	;ERROR
(5)	014070	104455		TRAP	C\$ERDF	
(6)	014072	000033		.WORD	27	
(6)	014074	005511		.WORD	EM27	
(6)	014076	007630		.WORD	ERR27	
6305	014100			68\$:	ESCAPE	SEG
(3)	014100	104410		TRAP	C\$ESCAPE	
(3)	014102	000012		.WORD	10001\$-	
6306	014104	005105		COM	R5	;CHANGE TO FLOATING 1
6307	014106	000241		CLC		;CLEAR CARRY
6308	014110	106105		ROLB	R5	;SHIFT BIT IN R5
6309	014112	001341		BNE	67\$	;IF R2=0 THEN DONE

6310	014114				ENDSEG	
(3)	014114				10001\$:	
(3)	014114	104405			TRAP	C\$ESEG
6311	014116				ENDTST	
(3)	014116				L10067:	
(3)	014116	104401			TRAP	C\$ETST
6312						
6313	014120				BADHEAD	
(2)					:***** TEST 15 *****	
6314					:*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST	
6315					:*FLOAT A 1 THOUGH IBUS* REGISTER 2	
6316					:*FLOAT A 0 THROUGH IBUS* REGISTER 2	
6317	014120				BADHEAD	
(2)					:***** TEST 15 *****	
6318						
6319	014120				BGNTST	
(3)	014120				T15::	
6320	014120				MSTCLR	:MASTER CLEAR M8200,4,7
(1)	014120	004537	003142		JSR R5, MSTCLR	:CLEAR M8200,4,7
6321	014124	012737	000002	002624	MOV #2, MRO	:SAVE REGISTER ADDRESS FOR TYPEOUT
6322	014132	012705	000001		MOV #1, R5	:START WITH BIT 0
6323	014136				MYINT	
(1)	014136	013701	002716		MOV KMCSR, R1	:GET DEVICE ADDRESS.
6324	014142				BGNSEG	
(3)	014142	104404			TRAP	C\$BSEG
6325	014144				64\$:	
6326	014144	010561	000004		MOV R5, 4(R1)	:PUT PATTERN INTO PORT4
6327	014150				ROMCLK	:NEXT WORD IS INSTRUCTION, BBN
(1)	014150	004537	003230		JSR R5, ROMCLK	:CLOCK INSTRUCTION
6328	014154	121102			121100.2	:MOV DATA TO IBUS* REGISTER 0
6329	014156				ROMCLK	:NEXT WORD IS INSTRUCTION, BBN
(1)	014156	004537	003230		JSR R5, ROMCLK	:CLOCK INSTRUCTION
6330	014162	121045			121005!<2*20>	:READ FROM IBUS* REGISTER 2
6331	014164	116104	000005		MOVB 5(R1), R4	:PUT 'FOUND' INTO R4
6332	014170	120504			CMPB R5, R4	:DATA CORRECT?
6333	014172	001414			BEQ 65\$	:BR IF YES
6334	014174				BERROR 27	:ERROR
(5)	014214	104455			TRAP C\$ERDF	
(6)	014216	000033			.WORD 27	
(6)	014220	005511			.WORD EM27	
(6)	014222	007630			.WORD ERR27	
6335	014224				65\$:	
(3)	014224	104410			ESCAPE SEG	
(3)	014226	000010			TRAP C\$ESCAPE	
6336	014230	000241			.WORD 10000\$-	
6337	014232	106105			CLC	:CLEAR CARRY
6338	014234	001343			ROLB R5	:SHIFT BIT IN R2
6339	014236				BNE 64\$	:IF R2=0 THEN DONE
(3)	014236				ENDSEG	
(3)	014236	104405			10000\$:	
6340	014240	012705	000001		TRAP	C\$ESEG
6341					MOV #1, R5	:START WITH BIT 0
6342	014244				69\$:	
(3)	014244	104404			COM R5	:CHANGE TO FLOATING ZERO
6343	014246				BGNSEG	
6344	014246	005105			TRAP	C\$BSEG
					67\$:	
					COM R5	



6345	014250	010561	000004		MOV R5,4(R1)	;PUT PATTERN INTO PORT4
6346	014254				ROMCLK	;NEXT WORD IS INSTRUCTION, BBN
(1)	014254	004537	003230		JSR R5,ROMCLK	;CLOCK INSTRUCTION
6347	014260	121102			121100!2	;MOV DATA TO IBUS* REGISTER 2
6348	014262				ROMCLK	;NEXT WORD IS INSTRUCTION, BBN
(1)	014262	004537	003230		JSR R5,ROMCLK	;CLOCK INSTRUCTION
6349	014266	121045			121005!<2*20>	;READ FROM IBUS* REGISTER 2
6350	014270	116104	000005		MOVB 5(R1),R4	;PUT 'FOUND' INTO R4
6351	014274	120504			CMPB R5,R4	;DATA CORRECT?
6352	014276	001414			BEQ 68\$	;BR IF YES
6353	014300				BERROR 27	;ERROR
(5)	014320	104455			TRAP C\$ERDF	
(6)	014322	000033			.WORD 27	
(6)	014324	005511			.WORD EM27	
(6)	014326	007630			.WORD ERR27	
6354	014330			68\$:	ESCAPE SEG	
(3)	014330	104410			TRAP C\$ESCAPE	
(3)	014332	000012			.WORD 10001\$-	
6355	014334	005105			COM R5	;CHANGE TO FLOATING 1
6356	014336	000241			CLC	;CLEAR CARRY
6357	014340	106105			ROLB R5	;SHIFT BIT IN R2
6358	014342	001341			BNE 67\$	;IF R2=0 THEN DONE
6359	014344				ENDSEG	
(3)	014344			10001\$:		
(3)	014344	104405			TRAP C\$ESEG	
6360	014346			ENDTST		
(3)	014346			L10070:		
(3)	014346	104401			TRAP C\$ETST	
6361						
6362	014350				BADHEAD	
(2)					;***** TEST 16 *****	
6363					;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST	
6364					;*FLOAT A 1 THROUGH IBUS* REGISTER 4	
6365					;*FLOAT A 0 THROUGH IBUS* REGISTER 4	
6366	014350				BADHEAD	
(2)					;***** TEST 16 *****	
6367						
6368	014350			BGNTST		
(3)	014350			T16::		
6369	014350				MSTCLR	;MASTER CLEAR M8200.4,7
(1)	014350	004537	003142		JSR R5,MSTCLR	;CLEAR M8200.4,7
6370	014354	012737	000004	002624	MOV #4,MRO	;SAVE REGISTER ADDRESS FOR TYPEOUT
6371	014362	012705	000001		MOV #1,R5	;START WITH BIT 0
6372	014366				MYINT	
(1)	014366	013701	002716		MOV KMCSR,R1	;GET DEVICE ADDRESS.
6373	014372				BGNSEG	
(3)	014372	104404			TRAP C\$BSEG	
6374	014374			64\$:		
6375	014374	010561	000004		MOV R5,4(R1)	;PUT PATTERN INTO PORT4
6376	014400				ROMCLK	;NEXT WORD IS INSTRUCTION, BBN
(1)	014400	004537	003230		JSR R5,ROMCLK	;CLOCK INSTRUCTION
6377	014404	121104			121100!4	;MOV DATA TO IBUS* REGISTER 4
6378	014406				ROMCLK	;NEXT WORD IS INSTRUCTION, BBN
(1)	014406	004537	003230		JSR R5,ROMCLK	;CLOCK INSTRUCTION
6379	014412	121105			121005!<4*20>	;READ FROM IBUS* REGISTER 4
6380	014414	116104	000005		MOVB 5(R1),R4	;PUT 'FOUND' INTO R4

6381	014420	120504		CMPB	R5,R4	:DATA CORRECT?
6382	014422	001414		BEQ	65\$	:BR IF YES
6383	014424			BERROR	27	:ERROR
(5)	014444	104455		TRAP	C\$ERDF	
(6)	014446	000033		.WORD	27	
(6)	014450	005511		.WORD	EM27	
(6)	014452	007630		.WORD	ERR27	
6384	014454		65\$:	ESCAPE	SEG	
(3)	014454	104410		TRAP	C\$ESCAPE	
(3)	014456	000010		.WORD	10000\$-	
6385	014460	000241		CLC		:CLEAR CARRY
6386	014462	106105		ROLB	R5	:SHIFT BIT IN R2
6387	014464	001343		BNE	64\$	:IF R2=0 THEN DONE
6388	014466			ENDSEG		
(3)	014466		10000\$:			
(3)	014466	104405		TRAP	C\$ESEG	
6389	014470	012705	000001	MOV	#1,R5	:START WITH BIT 0
6390				COM	R5	:CHANGE TO FLOATING ZERO
6391	014474			BGNSEG		
(3)	014474	104404		TRAP	C\$BSEG	
6392	014476		67\$:			
6393	014476	005105		COM	R5	
6394	014500	010561	000004	MOV	R5,4(R1)	:PUT PATTERN INTO PORT4
6395	014504			ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	014504	004537	003230	JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
6396	014510	121104		121100!4		:MOV DATA TO IBUS* REGISTER 4
6397	014512			ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	014512	004537	003230	JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
6398	014516	121105		121005!<4*20>		:READ FROM IBUS* REGISTER 4
6399	014520	116104	000005	MOVB	5(R1),R4	:PUT 'FOUND' INTO R4
6400	014524	120504		CMPB	R5,R4	:DATA CORRECT?
6401	014526	001414		BEQ	68\$	:BR IF YES
6402	014530			BERROR	27	:ERROR
(5)	014550	104455		TRAP	C\$ERDF	
(6)	014552	000033		.WORD	27	
(6)	014554	005511		.WORD	EM27	
(6)	014556	007630		.WORD	ERR27	
6403	014560		68\$:	ESCAPE	SEG	
(3)	014560	104410		TRAP	C\$ESCAPE	
(3)	014562	000012		.WORD	10001\$-	
6404	014564	005105		COM	R5	:CHANGE TO FLOATING 1
6405	014566	000241		CLC		:CLEAR CARRY
6406	014570	106105		ROLB	R5	:SHIFT BIT IN R2
6407	014572	001341		BNE	67\$	:IF R2=0 THEN DONE
6408	014574			ENDSEG		
(3)	014574		10001\$:			
(3)	014574	104405		TRAP	C\$ESEG	
6409	014576		ENDTST			
(3)	014576		L10071:			
(3)	014576	104401		TRAP	C\$ETST	
6410						
6411	014600			BADHEAD		
(2)				:***** TEST 17 *****		
6412				:*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST		
6413				:*FLOAT A 1 THROUGH IBUS* REGISTER 5		
6414				:*FLOAT A 0 THROUGH IBUS* REGISTER 5		

```

6415 014600 BADHEAD
(2) ;***** TEST 17 *****
6416
6417 014600 BGNTST
(3) 014600 T17::
6418 014600 MSTCLR ;MASTER CLEAR M8200,4,7
(1) 014600 004537 003142 JSR R5,,MSTCLR ;CLEAR M8200,4,7
6419 014604 012737 000005 002624 MOV #5,MRO ;SAVE REGISTER ADDRESS FOR TYPEOUT
6420 014612 012705 000001 MOV #1,R5 ;START WITH BIT 0
6421 014616 MYINT
(1) 014616 013701 002716 MOV KMCSR,R1 ;GET DEVICE ADDRESS.
6422 014622 BGNSEG
(3) 014622 104404 TRAP C$BSEG
6423 014624
6424 014624 010561 000004 64$: MOV R5,4(R1) ;PUT PATTERN INTO PORT4
6425 014630 ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
(1) 014630 004537 003230 JSR R5,,ROMCLK ;CLOCK INSTRUCTION
6426 014634 121105 121100!5 ;MOV DATA TO IBUS* REGISTER 5
6427 014636 ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
(1) 014636 004537 003230 JSR R5,,ROMCLK ;CLOCK INSTRUCTION
6428 014642 121125 121005!<5*20> ;READ FROM IBUS* REGISTER 5
6429 014644 116104 000005 MOVB 5(R1),R4 ;PUT 'FOUND' INTO R4
6430 014650 120504 CMPB R5,R4 ;DATA CORRECT?
6431 014652 001414 BEQ 65$ ;BR IF YES
6432 014654 BERROR 27 ;ERROR
(5) 014674 104455 TRAP C$ERDF
(6) 014676 000033 .WORD 27
(6) 014700 005511 .WORD EM27
(6) 014702 007630 .WORD ERR27
6433 014704 65$: ESCAPE SEG
(3) 014704 104410 TRAP C$ESCAPE
(3) 014706 000010 .WORD 10000$-.
6434 014710 000441 CLC ;CLEAR CARRY
6435 014712 106105 ROLB R5 ;SHIFT BIT IN R5
6436 014714 001343 BNE 64$ ;IF R5=0 THEN DONE
6437 014716 ENDSeg
(3) 014716 10000$:
(3) 014716 104405 TRAP C$ESEG
6438 014720 012705 000001 MOV #1,R5 ;START WITH BIT 0
6439 ;69$: COM R5 ;CHANGE TO FLOATING ZERO
6440 014724 BGNSEG
(3) 014724 104404 TRAP C$BSEG
6441 014726
6442 014726 005105 67$: COM R5
6443 014730 010561 000004 MOV R5,4(R1) ;PUT PATTERN INTO PORT4
6444 014734 ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
(1) 014734 004537 003230 JSR R5,,ROMCLK ;CLOCK INSTRUCTION
6445 014740 121105 121100!5 ;MOV DATA TO IBUS* REGISTER 5
6446 014742 ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
(1) 014742 004537 003230 JSR R5,,ROMCLK ;CLOCK INSTRUCTION
6447 014746 121125 121005!<5*20> ;READ FROM IBUS* REGISTER 5
6448 014750 116104 000005 MOVB 5(R1),R4 ;PUT 'FOUND' INTO R4
6449 014754 120504 CMPB R5,R4 ;DATA CORRECT?
6450 014756 001414 BEQ 68$ ;BR IF YES
6451 014760 BERROR 27 ;ERROR
(5) 015000 104455 TRAP C$ERDF

```

```

(6) 015002 000033      .WORD 27
(6) 015004 005511      .WORD EM27
(6) 015006 007630      .WORD ERR27
6452 015010      68$:  ESCAPE SEG
(3) 015010 104410      TRAP C$ESCAPE
(3) 015012 000012      .WORD 10001$-.
6453 015014 005105      COM R5      ;CHANGE TO FLOATING 1
6454 015016 000241      CLC      ;CLEAR CARRY
6455 015020 106105      ROLB R5      ;SHIFT BIT IN R5
6456 015022 001341      BNE 67$      ;IF R5=0 THEN DONE
6457 015024      ENDSEG
(3) 015024      10001$:
(3) 015024 104405      TRAP C$ESEG
6458 015026      ENDTST
(3) 015026      L10072:
(3) 015026 104401      TRAP C$ETST
6459
6460 015030      BADHEAD
(2)      ;***** TEST 18 *****
6461      ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
6462      ;*FLOAT A 1 THROUGH IBUS* REGISTER 10
6463      ;*FLOAT A 0 THROUGH IBUS* REGISTER 10
6464 015030      BADHEAD
(2)      ;***** TEST 18 *****
6465
6466 015030      BGNTST
(3) 015030      T18::
6467 015030      MSTCLR      ;MASTER CLEAR M8200.4,7
(1) 015030 004537 003142      JSR R5,.MSTCLR      ;CLEAR M8200.4,7
6468 015034 012737 000010 002624      MOV #10,MRO      ;SAVE REGISTER ADDRESS FOR TYPEOUT
6469 015042 012705 000001      MOV #1,R5      ;START WITH BIT 0
6470 015046      MYINT
(1) 015046 013701 002716      MOV KMCSR,R1      ;GET DEVICE ADDRESS.
6471 015052      BGNSEG
(3) 015052 104404      TRAP C$BSEG
6472 015054      64$:
6473 015054 010561 000004      MOV R5,4(R1)      ;PUT PATTERN INTO PORT4
6474 015060 042761 000141 000004      BIC #141,4(R1)      ;CLEAR UNWANTED BITS
6475 015066      ROMCLK      ;NEXT WORD IS INSTRUCTION, BBN
(1) 015066 004537 003230      JSR R5,.ROMCLK      ;CLOCK INSTRUCTION
6476 015072 121110      121100!10      ;MOV DATA TO IBUS* REGISTER 10
6477 015074      ROMCLK      ;NEXT WORD IS INSTRUCTION, BBN
(1) 015074 004537 003230      JSR R5,.ROMCLK      ;CLOCK INSTRUCTION
6478 015100 121205      121005!<10*20>      ;READ FROM IBUS* REGISTER 10
6479 015102 010502      MOV R5,R2
6480 015104 042705 000141      BIC #141,R5      ;CLEAR UNWANTED BITS
6481 015110 116104 000005      MOVB 5(R1),R4      ;PUT 'FOUND' INTO R4
6482 015114 120504      CMPB R5,R4      ;DATA CORRECT?
6483 015116 001414      BEQ 65$      ;BR IF YES
6484 015120      BEPROR 27      ;ERROR
(5) 015140 104455      TRAP C$ERDF
(6) 015142 000033      .WORD 27
(6) 015144 005511      .WORD EM27
(6) 015146 007630      .WORD ERR27
6485 015150      65$:  ESCAPE SEG
(3) 015150 104410      TRAP C$ESCAPE

```

```

(3) 015152 000012 .WORD 10000$-.
6486 015154 010205 MOV R2,R5
6487 015156 000241 CLC ;CLEAR CARRY
6488 015160 106105 ROLB R5 ;SHIFT BIT IN R5
6489 015162 001334 BNE 64$ ;IF R5=0 THEN DONE
6490 015164 ENDSEG
(3) 015164 10000$:
(3) 015164 104405 TRAP C$ESEG
6491 015166 012705 000001 MOV #1,R5 ;START WITH BIT 0
6492 ;69$: COM R5 ;CHANGE TO FLOATING ZERO
6493 015172 104404 BGNSEG
(3) 015172 104404 TRAP C$BSEG
6494 015174 005105
6495 015174 010561 000004 COM R5
6496 015176 042761 000141 000004 MOV R5,4(R1) ;PUT PATTERN INTO PORT4
6497 015202 004537 003230 BIC #141,4(R1) ;CLEAR UNWANTED BITS
6498 015210 121110 JSR R5,.ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
(1) 015210 004537 003230 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
6499 015214 121100!10 ROMCLK ;MOV DATA TO IBUS* REGISTER 10
6500 015216 004537 003230 JSR R5,.ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
(1) 015216 121205 121005!<10*20> ;CLOCK INSTRUCTION
6501 015222 010502 MOV R5,R2 ;READ FROM IBUS* REGISTER 10
6502 015224 042705 000141 BIC #141,R5
6503 015226 116104 000005 MOVB 5(R1),R4 ;CLEAR UNWANTED BITS
6504 015232 120504 CMPB R5,R4 ;PUT 'FOUND' INTO R4
6505 015240 001414 BEQ 68$ ;DATA CORRECT?
6506 015242 BERRR 27 ;BR IF YES
6507 015262 104455 TRAP C$ERDF ;ERROR
(5) 015264 000033 .WORD 27
(6) 015266 005511 .WORD EM27
(6) 015270 007630 .WORD ERR27
6508 015272 104410 68$: ESCAPE SEG
(3) 015272 000014 TRAP C$ESCAPE
(3) 015274 010205 .WORD 10001$-.
6509 015300 005105 MOV R2,R5
6510 015302 000241 COM R5 ;CHANGE TO FLOATING 1
6511 015304 106105 CLC ;CLEAR CARRY
6512 015306 001332 ROLB R5 ;SHIFT BIT IN R5
6513 015310 10001$: BNE 67$ ;IF R5=0 THEN DONE
(3) 015310 ENDSEG
(3) 015310 104405 TRAP C$ESEG
6515 015312 104401 TRAP C$ETST
(3) 015312 104401
6516 015314 BADHEAD
6517 (2) ;***** TEST 19 *****
6518 ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
6519 ;*FLOAT A 1 THROUGH IBUS* REGISTER 11
6520 ;*FLOAT A 0 THROUGH IBUS* REGISTER 11
6521 015314 BADHEAD
(2) ;***** TEST 19 *****
6522
6523 015314 BGNSTST

```

(3)	015314				T19::	MSTCLR		;MASTER CLEAR M8200.4,7
6524	015314					JSR	R5, MSTCLR	;CLEAR M8200.4,7
(1)	015314	004537	003142			MOV	#11, MRO	;SAVE REGISTER ADDRESS FOR TYPEOUT
6525	015320	012737	000011	002624		MOV	#1, R5	;START WITH BIT 0
6526	015326	012705	000001			MYINT		
6527	015332					MOV	KMCSR, R1	;GET DEVICE ADDRESS.
(1)	015332	013701	002716			BGNSEG		
6528	015336					TRAP	C\$BSEG	
(3)	015336	104404						
6529	015340				64\$:	MOV	R5, 4(R1)	;PUT PATTERN INTO PORT4
6530	015340	010561	000004			BIC	#262, 4(R1)	;CLEAR UNWANTED BITS
6531	015344	042761	000262	000004		ROMCLK		;NEXT WORD IS INSTRUCTION, BBN
6532	015352					JSR	R5, ROMCLK	;CLOCK INSTRUCTION
(1)	015352	004537	003230			121100!	11	;MOV DATA TO IBUS* REGISTER 11
6533	015356	121111				ROMCLK		;NEXT WORD IS INSTRUCTION, BBN
6534	015360					JSR	R5, ROMCLK	;CLOCK INSTRUCTION
(1)	015360	004537	003230			121005!	<11*20>	;READ FROM IBUS* REGISTER 11
6535	015364	121225				MOV	R5, R2	
6536	015366	010502				BIC	#262, R5	;CLEAR UNWANTED BITS
6537	015370	042705	000262			MOVB	5(R1), R4	;PUT 'FOUND' INTO R4
6538	015374	116104	000005			BIC	#20, R4	
6539	015400	042704	000020			CMPB	R5, R4	;DATA CORRECT?
6540	015404	120504				BEQ	65\$	;BR IF YES
6541	015406	001414				BERROR	27	;ERROR
6542	015410					TRAP	C\$ERDF	
(5)	015430	104455				.WORD	27	
(6)	015432	000033				.WORD	EM27	
(6)	015434	005511				.WORD	ERR27	
(6)	015436	007630						

6544	015440			65\$:	ESCAPE	SEG	
(3)	015440	104410			TRAP	C\$ESCAPE	
(3)	015442	000012			.WORD	10000\$-	
6545	015444	010205			MOV	R2,R5	
6546	015446	000241			CLC		;CLEAR CARRY
6547	015450	106105			ROLB	R5	;SHIFT BIT IN R5
6548	015452	001332			BNE	64\$	;IF R5=0 THEN DONE
6549	015454				ENDSEG		
(3)	015454			10000\$:			
(3)	015454	104405			TRAP	C\$ESEG	
6550	015456	012705	000001		MOV	#1,R5	;START WITH BIT 0
6551				;69\$:	COM	R5	;CHANGE TO FLOATING ZERO
6552	015462				BGNSEG		
(3)	015462	104404			TRAP	C\$BSEG	
6553	015464			67\$:			
6554	015464	005105			COM	R5	
6555	015466	010561	000004		MOV	R5,4(R1)	;PUT PATTERN INTO PORT4
6556	015472	042761	000262	000004	BIC	#262,4(R1)	;CLEAR UNWANTED BITS
6557	015500				ROMCLK		;NEXT WORD IS INSTRUCTION, BBN
(1)	015500	004537	003230		JSR	R5,..ROMCLK	;CLOCK INSTRUCTION
6558	015504	121111			121100!11		;MOV DATA TO IBUS* REGISTER 11
6559	015506				ROMCLK		;NEXT WORD IS INSTRUCTION, BBN
(1)	015506	004537	003230		JSR	R5,..ROMCLK	;CLOCK INSTRUCTION
6560	015512	121225			121005!<11*20>		;READ FROM IBUS* REGISTER 11

6562	015514	010502			MOV	R5,R2	
6563	015516	042705	000262		BIC	#262,R5	;CLEAR UNWANTED BITS
6564	015522	052705	000020		BIS	#20,R5	;ADD THESE BITS
6565	015526	116104	000005		MOVB	5(R1),R4	;PUT 'FOUND' INTO R4
6566	015532	120504			CMPB	R5,R4	;DATA CORRECT?
6567	015534	001414			BEQ	68\$	;BR IF YES
6568	015536				BERROR	27	;ERROR
(5)	015556	104455			TRAP	C\$ERDF	
(6)	015560	000033			.WORD	27	
(6)	015562	005511			.WORD	EM27	
(6)	015564	007630			.WORD	ERR27	
6569	015566			68\$:	ESCAPE	SEG	
(3)	015566	104410			TRAP	C\$ESCAPE	
(3)	015570	000014			.WORD	10001\$-	
6570	015572	010205			MOV	R2,R5	
6571	015574	005105			COM	R5	;CHANGE TO FLOATING 1
6572	015576	000241			CLC		;CLEAR CARRY
6573	015600	106105			ROLB	R5	;SHIFT BIT IN R5
6574	015602	001330			BNE	67\$	;IF R5=0 THEN DONE
6575	015604				ENDSEG		
(3)	015604			10001\$:			
(3)	015604	104405			TRAP	C\$ESEG	
6576	015606			END1ST			
(3)	015606			L10074:			
(3)	015606	104401			TRAP	C\$ETST	
6577							
6578	015610				BADHEAD		
(2)					;***** TEST 20 *****		
6579					;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST		
6580					;*FLOAT A 1 THROUGH IBUS REGISTER 0		
6581					;*FLOAT A 0 THROUGH IBUS REGISTER 0		
6582	015610				BADHEAD		
(2)					;***** TEST 20 *****		
6583							
6584	015610			BGNTST			
(3)	015610			T20::			
6585	015610				MSTCLR		;MASTER CLEAR M8200,4,7
(1)	015610	004537	003142		JSR	R5,.,MSTCLR	;CLEAR M8200,4,7
6586	015614	012737	000000	002624	MOV	#0,MRO	;SAVE REGISTER ADDRESS FOR TYPEOUT
6587	015622	012705	000001		MOV	#1,R5	;START WITH BIT 0
6588	015626				MYINT		
(1)	015626	013701	002716		MOV	KMCSR,R1	;GET DEVICE ADDRESS.
6589	015632				BGNSEG		
(3)	015632	104404			TRAP	C\$BSEG	
6590	015634			64\$:			
6591	015634	010561	000004		MOV	R5,4(R1)	;PUT PATTERN INTO PORT4
6592	015640				ROMCLK		;NEXT WORD IS INSTRUCTION, BBN
(1)	015640	004537	003230		JSR	R5,.,ROMCLK	;CLOCK INSTRUCTION
6593	015644	122100			122100		;MOV DATA TO IBUS* REGISTER 0
6594	015646				ROMCLK		;NEXT WORD IS INSTRUCTION, BBN
(1)	015646	004537	003230		JSR	R5,.,ROMCLK	;CLOCK INSTRUCTION
6595	015652	021005			21005		;READ FROM IBUS* REGISTER 0
6596	015654	116104	000005		MOVB	5(R1),R4	;PUT 'FOUND' INTO R4
6597	015660	120504			CMPB	R5,R4	;DATA CORRECT?
6598	015662	001414			BEQ	65\$	;BR IF YES
6599	015664				BERROR	29	;ERROR



(5)	015704	104455		TRAP	C\$ERDF	
(6)	015706	000035		.WORD	29	
(6)	015710	005571		.WORD	EM29	
(6)	015712	007770		.WORD	ERR29	
6600	015714		65\$:	ESCAPE	SEG	
(3)	015714	104410		TRAP	C\$ESCAPE	
(3)	015716	000010		.WORD	10000\$-	
6601	015720	000241		CLC		:CLEAR CARRY
6602	015722	106105		ROLB	R5	:SHIFT BIT IN R5
6603	015724	001343		BNE	64\$	:IF R5=0 THEN DONE
6604	015726			ENDSEG		
(3)	015726		10000\$:			
(3)	015726	104405		TRAP	C\$ESEG	
6605	015730	012705	000001	MOV	#1,R5	:START WITH BIT 0
6606			66\$:	COM	R5	:CHANGE TO FLOATING ZERO
6607	015734			BGNSEG		
(3)	015734	104404		TRAP	C\$BSEG	
6608	015736		67\$:			
6609	015736	005105		COM	R5	
6610	015740	010561	000004	MOV	R5,4(R1)	:PUT PATTERN INTO PORT4
6611	015744			ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	015744	004537	003230	JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
6612	015750	122100				:MOV DATA TO IBUS* REGISTER 0
6613	015752			ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	015752	004537	003230	JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
6614	015756	021005				:READ FROM IBUS* REGISTER 0
6615	015760	116104	000005	MOVB	5(R1),R4	:PUT 'FOUND' INTO R4
6616	015764	120504		CMPB	R5,R4	:DATA CORRECT?
6617	015766	001414		BEQ	68\$	:BR IF YES
6618	015770			BERROR	29	:ERROR
(5)	016010	104455		TRAP	C\$ERDF	
(6)	016012	000035		.WORD	29	
(6)	016014	005571		.WORD	EM29	
(6)	016016	007770		.WORD	ERR29	
6619	016020		68\$:	ESCAPE	SEG	
(3)	016020	104410		TRAP	C\$ESCAPE	
(3)	016022	000012		.WORD	10001\$-	
6620	016024	005105		COM	R5	:CHANGE TO FLOATING 1
6621	016026	000241		CLC		:CLEAR CARRY
6622	016030	106105		ROLB	R5	:SHIFT BIT IN P5
6623	016032	001341		BNE	67\$	:IF R5=0 THEN DONE
6624	016034			ENDSEG		
(3)	016034		10001\$:			
(3)	016034	104405		TRAP	C\$ESEG	
6625	016036		ENDTST			
(3)	016036		L10075:			
(3)	016036	104401		TRAP	C\$ETST	
6626						
6627	016040			BADHEAD		
(2)				:***** TEST 21 *****		
6628				:*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST		
6629				:*FLOAT A 1 THROUGH IBUS REGISTER 1		
6630				:*FLOAT A 0 THROUGH IBUS REGISTER 1		
6631	016040			BADHEAD		
(2)				:***** TEST 21 *****		
6632						

```
6633 016040          BGNTST
(3) 016040          T21::
6634 016040          MSTCLR          ;MASTER CLEAR M8200,4,7
(1) 016040 004537 003142          JSR R5,,MSTCLR          ;CLEAR M8200,4,7
6635 016044 012737 000001 002624 MOV #1,MRO          ;SAVE REGISTER ADDRESS FOR TYPEOUT
6636 016052 012705 000001          MOV #1,R5          ;START WITH BIT 0
6637 016056          MYINT
(1) 016056 013701 002716          MOV KMCSR,R1          ;GET DEVICE ADDRESS.
6638 016062          BGNSEG
(3) 016062 104404          TRAP C$BSEG
6639 016064          64$:
6640 016064 010561 000004          MOV R5,4(R1)          ;PUT PATTERN INTO PORT4
6641 016070          ROMCLK          ;NEXT WORD IS INSTRUCTION, BBN
(1) 016070 004537 003230          JSR R5,,ROMCLK          ;CLOCK INSTRUCTION
6642 016074 122101          122100!1          ;MOV DATA TO IBUS* REGISTER 1
6643 016076          ROMCLK          ;NEXT WORD IS INSTRUCTION, BBN
(1) 016076 004537 003230          JSR R5,,ROMCLK          ;CLOCK INSTRUCTION
6644 016102 021025          21005!<1*20>          ;READ FROM IBUS* REGISTER 1
6645 016104 116104 000005          MOVB 5(R1),R4          ;PUT 'FOUND' INTO R4
6646 016110 120504          CMPB R5,R4          ;DATA CORRECT?
6647 016112 001414          BEQ 65$          ;BR IF YES
6648 016114          BERROR 29          ;ERROR
(5) 016134 104455          TRAP C$ERDF
(6) 016136 000035          .WORD 29
(6) 016140 005571          .WORD EM29
(6) 016142 007770          .WORD ERR29
6649 016144          65$:
(3) 016144 104410          ESCAPE SEG
(3) 016146 000010          TRAP C$ESCAPE
6650 016150 000241          .WORD 10000$-
6651 016152 106105          CLC          ;CLEAR CARRY
6652 016154 001343          ROLB R5          ;SHIFT BIT IN R5
6653 016156          BNE 64$          ;IF R5=0 THEN DONE
(3) 016156          ENDSEG
(3) 016156 104405          10000$:
6654 016160 012705 000001          TRAP C$ESEG
6655          MOV #1,R5          ;START WITH BIT 0
6656 016164          ;69$:
(3) 016164 104404          COM R5          ;CHANGE TO FLOATING ZERO
6657 016166          BGNSEG
6658 016166 005105          TRAP C$BSEG
6659 016170 010561 000004          COM R5
6660 016174          67$:
(1) 016174 004537 003230          MOV R5,4(R1)          ;PUT PATTERN INTO PORT4
6661 016200 122101          ROMCLK          ;NEXT WORD IS INSTRUCTION, BBN
6662 016202          JSR R5,,ROMCLK          ;CLOCK INSTRUCTION
(1) 016202 004537 003230          122100!1          ;MOV DATA TO IBUS* REGISTER 1
6663 016206 021025          ROMCLK          ;NEXT WORD IS INSTRUCTION, BBN
6664 016210 116104 000005          JSR R5,,ROMCLK          ;CLOCK INSTRUCTION
6665 016214 120504          21005!<1*20>          ;READ FROM IBUS* REGISTER 1
6666 016216 001414          MOVB 5(R1),R4          ;PUT 'FOUND' INTO R4
6667 016220          CMPB R5,R4          ;DATA CORRECT?
(5) 016240 104455          BEQ 68$          ;BR IF YES
(6) 016242 000035          BERROR 29          ;ERROR
(6) 016244 005571          TRAP C$ERDF
(6) 016246 007770          .WORD 29
          .WORD EM29
          .WORD ERR29
```

6668	016250				68\$:	ESCAPE	SEG	
(3)	016250	104410				TRAP	C\$ESCAPE	
(3)	016252	000012				.WORD	10001\$-	
6669	016254	005105				COM	R5	;CHANGE TO FLOATING 1
6670	016256	000241				CLC		;CLEAR CARRY
6671	016260	106105				ROLB	R5	;SHIFT BIT IN R5
6672	016262	001341				BNE	67\$	;IF R5=0 THEN DONE
6673	016264					ENDSEG		
(3)	016264				10001\$:			
(3)	016264	104405				TRAP	C\$ESEG	
6674	016266				ENDTST			
(3)	016266				L10076:			
(3)	016266	104401				TRAP	C\$ETST	
6675								
6676	016270					BADHEAD		
(2)						:***** TEST 22 *****		
6677						:*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST		
6678						:*FLOAT A 1 THROUGH IBUS REGISTER 2		
6679						:*FLOAT A 0 THROUGH IBUS REGISTER 2		
6680	016270					BADHEAD		
(2)						:***** TEST 22 *****		
6681								
6682	016270				BGNTST			
(3)	016270				T22::			
6683	016270					MSTCLR		;MASTER CLEAR M8200,4,7
(1)	016270	004537	003142			JSR	R5, MSTCLR	;CLEAR M8200,4,7
6684	016274	012737	000002	002624		MOV	#2, MRO	;SAVE REGISTER ADDRESS FOR TYPEOUT
6685	016302	012705	000001			MOV	#1, R5	;START WITH BIT 0
6686	016306					MYINT		
(1)	016306	013701	002716			MOV	KMCSR, R1	;GET DEVICE ADDRESS.
6687	016312					BGNSEG		
(3)	016312	104404				.RAP	C\$BSEG	
6688	016314				64\$:			
6689	016314	010561	000004			MOV	R5, 4(R1)	;PUT PATTERN INTO PORT4
6690	016320					ROMCLK		;NEXT WORD IS INSTRUCTION, BBN
(1)	016320	004537	003230			JSR	R5, ROMCLK	;CLOCK INSTRUCTION
6691	016324	122102				122100!2		;MOV DATA TO IBUS* REGISTER 2
6692	016326					ROMCLK		;NEXT WORD IS INSTRUCTION, BBN
(1)	016326	004537	003230			JSR	R5, ROMCLK	;CLOCK INSTRUCTION
6693	016332	021045				21005!<2*20>		;READ FROM IBUS* REGISTER 2
6694	016334	116104	000005			MOVB	5(R1), R4	;PUT 'FOUND' INTO R4
6695	016340	120504				CMPB	R5, R4	;DATA CORRECT?
6696	016342	001414				BEQ	65\$	;BR IF YES
6697	016344					BERROR	29	;ERROR
(5)	016364	104455				TRAP	C\$ERDF	
(6)	016366	000035				.WORD	29	
(6)	016370	005571				.WORD	EM29	
(6)	016372	007770				.WORD	ERR29	
6698	016374				65\$:	ESCAPE	SEG	
(3)	016374	104410				TRAP	C\$ESCAPE	
(3)	016376	000010				.WORD	10000\$-	
6699	016400	000241				CLC		;CLEAR CARRY
6700	016402	106105				ROLB	R5	;SHIFT BIT IN R5
6701	016404	001343				BNE	64\$	;IF R5=0 THEN DONE
6702	016406					ENDSEG		
(3)	016406				10000\$:			

(3)	016406	104405			TRAP	C\$ESEG	
6703	016410	012705	000001		MOV	#1,R5	;START WITH BIT 0
6704				;69\$:	COM	R5	;CHANGE TO FLOATING ZERO
6705	016414				BGNSEG		
(3)	016414	104404			TRAP	C\$BSEG	
6706	016416			67\$:			
6707	016416	005105			COM	R5	
6708	016420	010561	000004		MOV	R5,4(R1)	;PUT PATTERN INTO PORT4
6709	016424				ROMCLK		;NEXT WORD IS INSTRUCTION, BBN
(1)	016424	004537	003230		JSR	R5,.ROMCLK	;CLOCK INSTRUCTION
6710	016430	122102			122100!2		;MOV DATA TO IBUS* REGISTER 2
6711	016432				ROMCLK		;NEXT WORD IS INSTRUCTION, BBN
(1)	016432	004537	003230		JSR	R5,.ROMCLK	;CLOCK INSTRUCTION
6712	016436	021045			21005!<2*20>		;READ FROM IBUS* REGISTER 2
6713	016440	116104	000005		MOVB	5(R1),R4	;PUT 'FOUND' INTO R4
6714	016444	120504			CMPB	R5,R4	;DATA CORRECT?
6715	016446	001414			BEQ	68\$	;BR IF YES
6716	016450				BERROR	29	;ERROR
(5)	016470	104455			TRAP	C\$ERDF	
(6)	016472	000035			.WORD	29	
(6)	016474	005571			.WORD	EM29	
(6)	016476	007770			.WORD	ERR29	
6717	016500			68\$:	ESCAPE	SEG	
(3)	016500	104410			TRAP	C\$ESCAPE	
(3)	016502	000012			.WORD	10001\$-	
6718	016504	005105			COM	R5	;CHANGE TO FLOATING 1
6719	016506	000241			CLC		;CLEAR CARRY
6720	016510	106105			RULB	R5	;SHIFT BIT IN R5
6721	016512	001341			BNE	67\$	;IF R5=0 THEN DONF
6722	016514				ENDSEG		
(3)	016514			10001\$:			
(3)	016514	104405			TRAP	C\$ESEG	
6723	016516			ENDTST			
(3)	016516			L10077:			
(3)	016516	104401			TRAP	C\$ETST	
6724							
6725	016520				BADHEAD		
(2)					;***** TEST 23 *****		
6726					;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST		
6727					;*FLOAT A 1 THROUGH IBUS REGISTER 3		
6728					;*FLOAT A 0 THROUGH IBUS REGISTER 3		
6729	016520				BADHEAD		
(2)					;***** TEST 23 *****		
6730							
6731	016520			BGNTST			
(3)	016520			T23::			
6732	016520				MSTCLR		;MASTER CLEAR M8200,4,7
(1)	016520	004537	003142		JSR	R5,.MSTCLR	;CLEAR M8200,4,7
6733	016524	012737	000003	002624	MOV	#3,MRO	;SAVE REGISTER ADDRESS FOR TYPEOUT
6734	016532	012705	000001		MOV	#1,R5	;START WITH BIT 0
6735	016536				MYINT		
(1)	016536	013701	002716		MOV	KMCSR,R1	;GET DEVICE ADDRESS.
6736	016542				BGNSEG		
(3)	016542	104404			TRAP	C\$BSEG	
6737	016544			64\$:			
6738	016544	010561	000004		MOV	R5,4(R1)	;PUT PATTERN INTO PORT4

6739	016550			ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	016550	004537	003230	JSR	R5, ROMCLK	:CLOCK INSTRUCTION
6740	016554	122103		122100!3		:MOV DATA TO IBUS* REGISTER 3
6741	016556			ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	016556	004537	003230	JSR	R5, ROMCLK	:CLOCK INSTRUCTION
6742	016562	021065		21005!<3*20>		:READ FROM IBUS* REGISTER 3
6743	016564	116104	000005	MOVB	5(R1), R4	:PUT 'FOUND' INTO R4
6744	016570	120504		CMPB	R5, R4	:DATA CORRECT?
6745	016572	001414		BEQ	65\$	:BR IF YES
6746	016574			BERROR	29	:ERROR
(5)	016614	104455		TRAP	C\$ERDF	
(6)	016616	000035		.WORD	29	
(6)	016620	005571		.WORD	EM29	
(6)	016622	007770		.WORD	ERR29	
6747	016624			65\$: ESCAPE	SEG	
(3)	016624	104410		TRAP	C\$ESCAPE	
(3)	016626	0C0010		.WORD	10000\$-	
6748	016630	000241		CLC		:CLEAR CARRY
6749	016632	106105		ROLB	R5	:SHIFT BIT IN R5
6750	016634	001343		BNE	64\$	:IF R5=0 THEN DONE
6751	016636			ENDSEG		
(3)	016636			10000\$: TRAP	C\$ESEG	
(3)	016636	104405		MOV	#1, R5	:START WITH BIT 0
6752	016640	012705	000001	69\$: COM	R5	:CHANGE TO FLOATING ZERO
6753				BGNSEG		
6754	016644			67\$: TRAP	C\$BSEG	
(3)	016644	104404				
6755	016646			COM	R5	
6756	016646	005105		MOV	R5, 4(R1)	:PUT PATTERN INTO PORT4
6757	016650	010561	000004	ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
6758	016654			JSR	R5, ROMCLK	:CLOCK INSTRUCTION
(1)	016654	004537	003230	122100!3		:MOV DATA TO IBUS* REGISTER 3
6759	016660	122103		ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	016662	004537	003230	JSR	R5, ROMCLK	:CLOCK INSTRUCTION
6761	016666	021065		21005!<3*20>		:READ FROM IBUS* REGISTER 3
6762	016670	116104	000005	MOVB	5(R1), R4	:PUT 'FOUND' INTO R4
6763	016674	120504		CMPB	R5, R4	:DATA CORRECT?
6764	016676	001414		BEQ	68\$	:BR IF YES
6765	016700			BERROR	29	:ERROR
(5)	016720	104455		TRAP	C\$ERDF	
(6)	016722	000035		.WORD	29	
(6)	016724	005571		.WORD	EM29	
(6)	016726	007770		.WORD	ERR29	
6766	016730			68\$: ESCAPE	SEG	
(3)	016730	104410		TRAP	C\$ESCAPE	
(3)	016732	000012		.WORD	10001\$-	
6767	016734	005105		COM	R5	:CHANGE TO FLOATING 1
6768	016736	000241		CLC		:CLEAR CARRY
6769	016740	106105		ROLB	R5	:SHIFT BIT IN R5
6770	016742	001341		BNE	67\$	:IF R5=0 THEN DONE
6771	016744			ENDSEG		
(3)	016744			10001\$: TRAP	C\$ESEG	
(3)	016744	104405				
6772	016746			ENDTST		
(3)	016746			L10100:		

(3)	016746	104401			TRAP	C\$ETST	
6773							
6774	016750				BADHEAD		
(2)					:***** TEST 24 *****		
6775					:*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST		
6776					:*FLOAT A 1 THROUGH IBUS REGISTER 4		
6777					:*FLOAT A 0 THROUGH IBUS REGISTER 4		
6778	016750				BADHEAD		
(2)					:***** TEST 24 *****		
6779							
6780	016750				BGNTST		
(3)	016750				T24::		
6781	016750				MSTCLR		
(1)	016750	004537	003142		JSR	R5, MSTCLR	:MASTER CLEAR M8200,4,7
6782	016754	012737	000004	002624	MOV	#4, MRO	:CLEAR M8200,4,7
6783	016762	012705	000001		MOV	#1, R5	:SAVE REGISTER ADDRESS FOR TYPEOUT
6784	016766				MYINT		:START WITH BIT 0
(1)	016766	013701	002716		MOV	KMCSR, R1	:GET DEVICE ADDRESS.
6785	016772				BGNSEG		
(3)	016772	104404			TRAP	C\$BSEG	
6786	016774						
6787	016774	010561	000004		MOV	R5, 4(R1)	:PUT PATTERN INTO PORT4
6788	017000				ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	017000	004537	003230		JSR	R5, ROMCLK	:CLOCK INSTRUCTION
6789	017004	122104			122100!4		:MOV DATA TO IBUS* REGISTER 4
6790	017006				ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	017006	004537	003230		JSR	R5, ROMCLK	:CLOCK INSTRUCTION
6791	017012	021105			21005!<4*20>		:READ FROM IBUS* REGISTER 4
6792	017014	116104	000005		MOVB	5(R1), R4	:PUT 'FOUND' INTO R4
6793	017020	120504			CMPS	R5, R4	:DATA CORRECT?
6794	017022	001414			BEQ	65\$	:BR IF YES
6795	017024				BERROR	29	:ERROR
(5)	017044	104455			TRAP	C\$ERDF	
(6)	017046	000035			.WORD	29	
(6)	017050	005571			.WORD	EM29	
(6)	017052	007770			.WORD	ERR29	
6796	017054				ESCAPE	SEG	
(3)	017054	104410			TRAP	C\$ESCAPE	
(3)	017056	000010			.WORD	10000\$-	
6797	017060	000241			CLC		:CLEAR CARRY
6798	017062	106105			ROLB	R5	:SHIFT BIT IN R5
6799	017064	001343			BNE	64\$	:IF R5=0 THEN DONE
6800	017066				ENDSEG		
(3)	017066						
(3)	017066	104405			10000\$:		
6801	017070	012705	000001		TRAP	C\$ESEG	
6802					MOV	#1, R5	:START WITH BIT 0
6803	017074				COM	R5	:CHANGE TO FLOATING ZERO
(3)	017074	104404			BGNSEG		
6804	017076				TRAP	C\$BSEG	
6805	017076	005105					
6806	017100	010561	000004		COM	R5	
6807	017104				MOV	R5, 4(R1)	:PUT PATTERN INTO PORT4
(1)	017104	004537	003230		ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
6808	017110	122104			JSR	R5, ROMCLK	:CLOCK INSTRUCTION
6809	017112				122100!4		:MOV DATA TO IBUS* REGISTER 4
					ROMCLK		:NEXT WORD IS INSTRUCTION, BBN

```
(1) 017112 004537 003230 JSR R5,ROMCLK ;CLOCK INSTRUCTION
6810 017116 021105 21005!<4*20> ;READ FROM IBUS* REGISTER 4
6811 017120 116104 000005 MOVB 5(R1),R4 ;PUT 'FOUND' INTO R4
6812 017124 120504 CMPB R5,R4 ;DATA CORRECT?
6813 017126 001414 BEQ 68$ ;BR IF YES
6814 017130 BERROR 29 ;ERROR
(5) 017150 104455 TRAP C$ERDF
(6) 017152 000035 .WORD 29
(6) 017154 005571 .WORD EM29
(6) 017156 007770 .WORD ERR29
6815 017160 68$: ESCAPE SEG
(3) 017160 104410 TRAP C$ESCAPE
(3) 017162 000012 .WORD 10001$-.
6816 017164 005105 COM R5 ;CHANGE TO FLOATING 1
6817 017166 000241 CLC ;CLEAR CARRY
6818 017170 106105 ROLB R5 ;SHIFT BIT IN R5
6819 017172 001341 BNE 67$ ;IF R5=0 THEN DONE
6820 017174 ENDSEG
(3) 017174 10001$: TRAP C$ESEG
(3) 017174 104405 ENDTST
(3) 017176 L10101: TRAP C$ETST
(3) 017176 104401
6822 017200 BADHEAD
6823 017200 (2) ;***** TEST 25 *****
6824 017200 ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
6825 017200 ;*FLOAT A 1 THROUGH IBUS REGISTER 5
6826 017200 ;*FLOAT A 0 THROUGH IBUS REGISTER 5
6827 017200 BADHEAD
6828 017200 (2) ;***** TEST 25 *****
6829 017200 BGNTST
(3) 017200 T25::
6830 017200 MSTCLR ;MASTER CLEAR M8200,4,7
(1) 017200 004537 003142 JSR R5,MSTCLR ;CLEAR M8200,4,7
6831 017204 012737 000005 002624 MOV #5,MRO ;SAVE REGISTER ADDRESS FOR TYPEOUT
6832 017212 012705 000001 MOV #1,R5 ;START WITH BIT 0
6833 017216 MYINT
(1) 017216 013701 002716 MOV KMCSR,R1 ;GET DEVICE ADDRESS.
6834 017222 BGNSFG
(3) 017222 104404 TRAP C$BSEG
6835 017224 64$: MOV R5,4(R1) ;PUT PATTERN INTO PORT4
6836 017224 010561 000004 ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
6837 017230 (1) 017230 004537 003230 JSR R5,ROMCLK ;CLOCK INSTRUCTION
6838 017234 122105 122100!5 ;MOV DATA TO IBUS* REGISTER 5
6839 017236 ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
(1) 017236 004537 003230 JSR R5,ROMCLK ;CLOCK INSTRUCTION
6840 017242 021125 21005!<5*20> ;READ FROM IBUS* REGISTER 5
6841 017244 116104 000005 MOVB 5(R1),R4 ;PUT 'FOUND' INTO R4
6842 017250 120504 CMPB R5,R4 ;DATA CORRECT?
6843 017252 001414 BEQ 65$ ;BR IF YES
6844 017254 BERROR 29 ;ERROR
(5) 017274 104455 TRAP C$ERDF
(6) 017276 000035 .WORD 29
```

```
(6) 017300 005571
(6) 017302 007770
6845 017304 104410
(3) 017304 104410
(3) 017306 000010
6846 017310 000241
6847 017312 106105
6848 017314 001343
6849 017316
(3) 017316
(3) 017316 104405
6850 017320 012705 000001
6851
6852 017324
(3) 017324 104404
6853 017326
6854 017326 005105
6855 017330 010561 000004
6856 017334
(1) 017334 004537 003230
6857 017340 122105
6858 017342
(1) 017342 004537 003230
6859 017346 021125
6860 017350 116104 000005
6861 017354 120504
6862 017356 001414
6863 017360
(5) 017400 104455
(6) 017402 000035
(6) 017404 005571
(6) 017406 007770
6864 017410
(3) 017410 104410
(3) 017412 000012
6865 017414 005105
6866 017416 000241
6867 017420 106105
6868 017422 001341
6869 017424
(3) 017424
(3) 017424 104405
6870 017426
(3) 017426
(3) 017426 104401
6871
6872 017430
(2)
6873
6874
6875
6876 017430
(2)
6877
6878 017430
(3) 017430
```

```
65$: .WORD EM29
      .WORD ERR29
      ESCAPE SEG
      TRAP C$ESCAPE
      .WORD 10000$-.
      CLC ;CLEAR CARRY
      ROLB R5 ;SHIFT BIT IN R5
      BNE 64$ ;IF R5=0 THEN DONE
      ENDSEG

10000$: TRAP C$ESEG
        MOV #1,R5 ;START WITH BIT 0
        COM R5 ;CHANGE TO FLOATING ZERO
        BGNSEG
        TRAP C$BSEG

67$: COM R5
      MOV R5,4(R1) ;PUT PATTERN INTO PORT4
      ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
      JSR R5,ROMCLK ;CLOCK INSTRUCTION
      122100!5 ;MOV DATA TO IBUS* REGISTER 5
      ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
      JSR R5,ROMCLK ;CLOCK INSTRUCTION
      21005!<5*20> ;READ FROM IBUS* REGISTER 5
      MOVB 5(R1),R4 ;PUT 'FOUND' INTO R4
      CMPB R5,R4 ;DATA CORRECT?
      BEQ 68$ ;BR IF YES
      BERROR 29 ;ERROR
      TRAP C$ERDF
      .WORD 29
      .WORD EM29
      .WORD ERR29
      ESCAPE SEG
      TRAP C$ESCAPE
      .WORD 10001$-.
      COM R5 ;CHANGE TO FLOATING 1
      CLC ;CLEAR CARRY
      ROLB R5 ;SHIFT BIT IN R5
      BNE 67$ ;IF R5=0 THEN DONE
      ENDSEG

10001$: TRAP C$ESEG

ENDTST
L10102: TRAP C$ETST

BADHEAD
;***** TEST 26 *****
;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
;*FLOAT A 1 THROUGH IBUS REGISTER 6
;*FLOAT A 0 THROUGH IBUS REGISTER 6
BADHEAD
;***** TEST 26 *****

BGNTEST
T26::
```



6879	017430				MSTCLR		;MASTER CLEAR M8200,4,7
(1)	017430	004537	003142		JSR	R5, MSTCLR	;CLEAR M8200,4,7
6880	017434	012737	000006	002624	MOV	#6, MRO	;SAVE REGISTER ADDRESS FOR TYPEOUT
6881	017442	012705	000001		MOV	#1, R5	;START WITH BIT 0
6882	017446				MYINT		
(1)	017446	013701	002716		MOV	KMCSR, R1	;GET DEVICE ADDRESS.
6883	017452				BGNSEG		
(3)	017452	104404			TRAP	C\$BSEG	
6884	017454			64\$:			
6885	017454	010561	000004		MOV	R5, 4(R1)	;PUT PATTERN INTO PORT4
6886	017460				ROMCLK		;NEXT WORD IS INSTRUCTION, BBA
(1)	017460	004537	003230		JSR	R5, ROMCLK	;CLOCK INSTRUCTION
6887	017464	122106			122100!6		;MOV DATA TO IBUS* REGISTER 6
6888	017466				ROMCLK		;NEXT WORD IS INSTRUCTION, BBN
(1)	017466	004537	003230		JSR	R5, ROMCLK	;CLOCK INSTRUCTION
6889	017472	021145			21005!<6*20>		;READ FROM IBUS* REGISTER 6
6890	017474	116104	000005		MOVB	5(R1), R4	;PUT 'FOUND' INTO R4
6891	017500	120504			CMPB	R5, R4	;DATA CORRECT?
6892	017502	001414			BEQ	65\$	;BR IF YES
6893	017504				BERROR	29	;ERROR
(5)	017524	104455			TRAP	C\$ERDF	
(6)	017526	000035			.WORD	29	
(6)	017530	005571			.WORD	EM29	
(6)	017532	007770			.WORD	ERR29	
6894	017534			65\$:	ESCAPE	SEG	
(3)	017534	104410			TRAP	C\$ESCAPE	
(3)	017536	000010			.WORD	10000\$-	
6895	017540	000241			CLC		;CLEAR CARRY
6896	017542	106105			ROLB	R5	;SHIFT BIT IN R5
6897	017544	001343			BNE	64\$	;IF R5=0 THEN DONE
6898	017546				ENDSEG		
(3)	017546			10000\$:			
(3)	017546	104405			TRAP	C\$ESEG	
6899	017550	012705	000001		MOV	#1, R5	;START WITH BIT 0
6900				;69\$:	COM	R5	;CHANGE TO FLOATING ZERO
6901	017554				BGNSEG		
(3)	017554	104404			TRAP	C\$BSEG	
6902	017556			67\$:			
6903	017556	005105			COM	R5	
6904	017560	010561	000004		MOV	R5, 4(R1)	;PUT PATTERN INTO PORT4
6905	017564				ROMCLK		;NEXT WORD IS INSTRUCTION, BBN
(1)	017564	004537	003230		JSR	R5, ROMCLK	;CLOCK INSTRUCTION
6906	017570	122106			122100!6		;MOV DATA TO IBUS* REGISTER 6
6907	017572				ROMCLK		;NEXT WORD IS INSTRUCTION, BBN
(1)	017572	004537	003230		JSR	R5, ROMCLK	;CLOCK INSTRUCTION
6908	017576	021145			21005!<6*20>		;READ FROM IBUS* REGISTER 6
6909	017600	116104	000005		MOVB	5(R1), R4	;PUT 'FOUND' INTO R4
6910	017604	120504			CMPB	R5, R4	;DATA CORRECT?
6911	017606	001414			BEQ	68\$	;BR IF YES
6912	017610				BERROR	29	;ERROR
(5)	017630	104455			TRAP	C\$ERDF	
(6)	017632	000035			.WORD	29	
(6)	017634	005571			.WORD	EM29	
(6)	017636	007770			.WORD	ERR29	
6913	017640			68\$:	ESCAPE	SEG	
(3)	017640	104410			TRAP	C\$ESCAPE	

```

(3) 017642 000012      .WORD 10001$-.
6914 017644 005105      COM R5      ;CHANGE TO FLOATING 1
6915 017646 000241      CLC          ;CLEAR CARRY
6916 017650 106105      ROLB R5      ;SHIFT BIT IN R5
6917 017652 001341      BNE 67$      ;IF R5=0 THEN DONE
6918 017654      ENDSEG
(3) 017654      10001$:
(3) 017654 104405      TRAP C$ESEG
6919 017656      ENDTST
(3) 017656      L10103:
(3) 017656 104401      TRAP C$ETST
6920
6921 017660      BADHEAD
(2)      ;***** TEST 27 *****
6922      ;*MICRO PROCEOR IBUS* REGISTER WRITE/READ TEST
6923      ;*FLOAT A 1 THOUGH IBUS* REGISTER 7
6924      ;*FLOAT A 0 THROUGH IBUS* REGISTER 7
6925 017660      BADHEAD
(2)      ;***** TEST 27 *****
6926
6927 017660      BGNST
(3) 017660      T27::
6928 017660      MSTCLR      ;MASTER CLEAR M8200.4,7
(1) 017660 004537 003142      JSR R5,.MSTCLR      ;CLEAR M8200.4,7
6929 017664 012737 000007 002624      MOV #7,MRO      ;SAVE REGISTER ADDRESS FOR TYPEOUT
6930 017672 012705 000001      MOV #1,R5      ;START WITH BIT 0
6931 017676      MYINT
(1) 017676 013701 002716      MOV KMCSR,R1      ;GET DEVICE ADDRESS.
6932 017702      BGNSEG
(3) 017702 104404      TRAP C$BSEG
6933 017704      64$:
6934 017704 010561 000004      MOV R5,4(R1)      ;PUT PATTERN INTO PORT4
6935 017710      ROMCLK      ;NEXT WORD IS INSTRUCTION, BBN
(1) 017710 004537 003230      JSR R5,.ROMCLK      ;CLOCK INSTRUCTION
6936 017714 122107      122100!7      ;MOV DATA TO IBUS* REGISTER 7
6937 017716      ROMCLK      ;NEXT WORD IS INSTRUCTION, BBN
(1) 017716 004537 003230      JSR R5,.ROMCLK      ;CLOCK INSTRUCTION
6938 017722 021165      21005!<7*20>      ;READ FROM IBUS* REGISTER 7
6939 017724 116104 000005      MOVB 5(R1),R4      ;PUT 'FOUND' INTO R4
6940 017730 120504      CMPB R5,R4      ;DATA CORRECT?
6941 017732 001414      BEQ 65$      ;BR IF YES
6942 017734      BERROR 29      ;ERROR
(5) 017754 104455      TRAP C$ERDF
(6) 017756 000035      .WORD 29
(6) 017760 005571      .WORD EM29
(6) 017762 007770      .WORD ERR29
6943 017764      65$:
(3) 017764 104410      ESCAPE SEG
(3) 017766 000010      TRAP C$ESCAPE
6944 017770 000241      .WORD 10000$-.
6945 017772 106105      CLC          ;CLEAR CARRY
6946 017774 001343      ROLB R5      ;SHIFT BIT IN R5
6947 017776      BNE 64$      ;IF R5=0 THEN DONE
(3) 017776      10000$:
(3) 017776 104405      TRAP C$ESEG
6948 020000 012705 000001      MOV #1,R5      ;START WITH BIT 0
  
```

```

6949      ;69$: COM R5 ;CHANGE TO FLOATING ZERO
6950      BGNSEG
(3) 020004 104404 TRAP C$BSEG
6951      67$: COM R5
6952      MOV R5,4(R1) ;PUT PATTERN INTO PORT4
6953      ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
(1) 020014 004537 003230 JSR R5,ROMCLK ;CLOCK INSTRUCTION
6954      122100!7 ;MOV DATA TO IBUS* REGISTER 7
6955      ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
(1) 020022 004537 003230 JSR R5,ROMCLK ;CLOCK INSTRUCTION
6956      21005!<7*20> ;READ FROM IBUS* REGISTER 7
6957      MOV 5(R1),R4 ;PUT 'FOUND' INTO R4
6958      CMPB R5,R4 ;DATA CORRECT?
6959      BEQ 68$ ;BR IF YES
6960      BERROR 29 ;ERROR
(5) 020060 104455 TRAP C$FRDF
(6) 020062 000035 .WORD 29
(6) 020064 005571 .WORD EM29
(6) 020066 007770 .WORD ERR29
6962      68$: ESCAPE SEG
(3) 020070 104410 TRAP C$ESCAPE
(3) 020072 000012 .WORD 10001$-
6963      COM R5 ;CHANGE TO FLOATING 1
6964      CLC ;CLEAR CARRY
6965      ROLB R5 ;SHIFT BIT IN R5
6966      BNE 67$ ;IF R5=0 THEN DONE
6967      ENDSEG
(3) 020104 104405 10001$: TRAP C$ESEG
(3) 020106 104401 ENDTST
(3) 020106 104401 L10104: TRAP C$ETST
6969      BADHEAD
6970      (2) ;***** TEST 28 *****
6971      ;*MICRO PROCESSOR IBUS DUAL ADDRESS TEST
6972      ;*WRITE ALL IBUS REGISTERS WITH INCREMENTING PATTERN
6973      ;*READ ALL IBUS REGISTERS TO VERIFY CORRECT ADDRESSING
6974      BADHEAD
6975      (2) ;***** TEST 28 *****
6976      BGNST
(3) 020110 T28::
6977      (1) 020110 004537 003142 MSTCLR ;MASTER CLEAR M8200,4,7
6978      020114 012705 000001 JSR R5,MSTCLR ;CLEAR M8200,4,7
6979      020120 005002 MOV #1,R5 ;START WITH A ONE
6980      (1) 020122 013701 002716 CLR R2 ;R2 CONTAINS ADDRESS OF REGISTER
6981      020126 104404 MYINT ;GET DEVICE ADDRESS.
(3) 020126 104404 TRAP C$BSEG
6982      020130 010203 1$: MOV R2,R3 ;R3=REGISTER ADDRESS
6983      020132 010561 000004 MOV R5,4(R1) ;WRITE DATA TO PORT4
6984      020136 042737 000017 BIC #17,5$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
6985      020144 050337 020154 BIS R3,5$ ;ADD ADDRESS TO INSTRUCTION
  
```

6986	020150			ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	020150	004537	003230	JSR	R5,ROMCLK	:CLOCK INSTRUCTION
6987	020154	122100		5\$: 122100		:MOVE DATA TO IBUS REGISTER
6988	020156	006303		ASL	R3	:SHIFT ADDRESS
6989	020160	006303		ASL	R3	:4 TIMES TO GET
6990	020162	006303		ASL	R3	:IT TO BITS 4-7
6991	020164	006303		ASL	R3	:OF NEXT INSTRUCTION
6992	020166	042737	000360 020204	BIC	#360,6\$	:CLEAR ADDRESS FIELD
6993	020174	050337	020204	BIS	R3,6\$	:ADD ADDRESS TO INSTRUCTION
6994	020200			ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	020200	004537	003230	JSR	R5,ROMCLK	:CLOCK INSTRUCTION
6995	020204	021005		6\$: 21005		:READ FROM IBUS REGISTER
6996	020206	116104	000005	MOVB	5(R1),R4	:PUT 'FOUND' IN R4
6997	020212	120504		CMPB	R5,R4	:IS DATA CORRECT?
6998	020214	001414		BEQ	2\$	:BR IF YES
6999	020216			BERROR	29	:DATA ERROR
(5)	020236	104455		TRAP	C\$ERDF	
(6)	020240	000035		.WORD	29	
(6)	020242	005571		.WORD	EM29	
(6)	020244	007770		.WORD	ERR29	
7000	020246			2\$: ESCAPE	SEG	
(3)	020246	104410		TRAP	C\$ESCAPE	
(3)	020250	000014		.WORD	10000\$-	
7001	020252	005205		INC	R5	:INCREMENT PATTERN
7002	020254	005202		INC	R2	:INCREMENT REGISTER ADDRESS
7003	020256	022702	000010	CMP	#7+1,R2	:LAST ADDRESS DONE?
7004	020262	001322		BNE	1\$	:BR IF NO
7005	020264			ENDSEG		
(3)	020264			10000\$: TRAP	C\$ESEG	
(3)	020264	104405		MOV	#1,R5	:RESTART PATTERN TO 1
7006	020266	012705	000001	CLR	R2	:RESTART AT ADDRESS 0
7007	020272	005002		BGNSEG		
7008	020274			TRAP	C\$BSEG	
(3)	020274	104404		CLR	R3	:RESTART AT ADDRESS 0
7009	020276	005003		BIC	#360,7\$	:CLEAR ADDRESS FIELD OF INSTRUCTION
7010	020300	042737	000360 020316	BIS	R3,7\$	:ADD ADDRESS TO INSTRUCTION
7011	020306	050337	020316	ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
7012	020312			JSR	R5,ROMCLK	:CLOCK INSTRUCTION
(1)	020312	004537	003230	7\$: 21005		:READ FROM IBUS REGISTER
7013	020316	021005		MOVB	5(R1),R4	:PUT 'FOUND' IN \$GDDAT
7014	020320	116104	000005	CMPB	R5,R4	:DATA CORRECT?
7015	020324	120504		BEQ	4\$	:BR IF YES
7016	020326	001414		BERROR	30	:DUAL ADDRESSING ERROR
7017	020330			TRAP	C\$ERDF	
(5)	020350	104455		.WORD	30	
(6)	020352	000036		.WORD	EM30	
(6)	020354	004337		.WORD	ERR30	
(6)	020356	010052		4\$: ESCAPE	SEG	
7018	020360			TRAP	C\$ESCAPE	
(3)	020360	104410		.WORD	10001\$-	
(3)	020362	000020		INC	R5	:INCREMENT PATTERN
7019	020364	005205		INC	R2	:NEXT ADDRESS
7020	020366	005202		ADD	#20,R3	:ADD 1 TO ADDRESS IN R3(SHIFTED 4 TIMES)
7021	020370	062703	000020	CMP	#7+1,R2	:LAST ADDRESS DONE?
7022	020374	022702	000010	BNE	3\$	:BR IF NO
7023	020400	001337				

7024	020402			ENDSEG	
(3)	020402			10001\$:	
(3)	020402	104405		TRAP	C\$ESEG
7025	020404			ENDTST	
(3)	020404			L10105:	
(3)	020404	104401		TRAP	C\$ETST
7026					
7027	020406			BADHEAD	
(2)				:***** TEST 29 *****	
7028				:*MICRO PROCESSOR BR REGISTER TEST	
7029				:*FLOAT A 1 THOUGH THE BR	
7030				:*FLOAT A 0 THOUGH THE BR	
7031	020406			BADHEAD	
(2)				:***** TEST 29 *****	
7032					
7033	020406			BGNTST	
(3)	020406			T29::	
7034					
7035	020406	004537	003142	MSTCLR	:R1 CONTAINS BASE M8200,4,7 ADDRESS
(1)	020406	012702	000001	JSR	:MASTER CLEAR COMM. MICRO-PROCESSOR FAMILY
7036	020412			MOV	:CLEAR M8200,4,7
7037	020416			MYINT	:START PATTERN WITH BIT0
(1)	020416	013701	002716	MOV	
7038	020422			KMCSR,R1	:GET DEVICE ADDRESS.
(3)	020422	104404		BGNSEG	
7039	020424			TRAP	C\$BSEG
7040	020424	010261	000004	64\$:	
7041	020430			MOV	R2,4(R1) :WRITE PATTERN IN PORT4
(1)	020430	004537	003230	ROMCLK	:NEXT WORD IS INSTRUCTION, BBN
7042	020434	120500		JSR	R5,.ROMCLK :CLOCK INSTRUCTION
7043	020436			120500	:MOVE DATA TO THE BR REGISTER
(1)	020436	004537	003230	ROMCLK	:NEXT WORD IS INSTRUCTION, BBN
7044	020442	061225		JSR	R5,.ROMCLK :CLOCK INSTRUCTION
7045	020444	116104	000005	061225	:MOVE BR TO PORT 5
7046	020450	120204		MOV	5(R1),R4 :PUT 'FOUND' IN R4
7047	020452	001414		CMPS	R2,R4 :IS DATA CORRECT?
7048	020454			BEQ	65\$ :BR IF YES
(5)	020474	104455		BERROR	3 :DATA ERROR
(6)	020476	000003		TRAP	C\$ERDF
(6)	020500	004402		.WORD	3
(6)	020502	006174		.WORD	EM3
7049	020504			.WORD	ERR3
(3)	020504	104410		65\$:	
(3)	020506	000010		ESCAPE	SEG
7050	020510	000241		TRAP	C\$ESCAPE
7051	020512	106102		.WORD	10000\$-
7052	020514	001343		CLC	
7053	020516			ROLB	R2 :CLEAR CARRY
(3)	020516			BNE	64\$ :SHIFT BIT IN R2
(3)	020516	104405		ENDSEG	:DONE IF R2=0
7054	020520	012702	000001	10000\$:	
7055	020524			TRAP	C\$ESEG
7056	020524			MOV	#1,R2 :START PATTERN WITH BIT0
(3)	020524	104404		69\$:	
7057	020526			BGNSEG	
7058	020526	005102		TRAP	C\$BSEG
				67\$:	
				COM	R2

7059	020530	010261	000004	MOV	R2,4(R1)	;WRITE PATTERN IN PORT4
7060	020534			ROMCLK		;NEXT WORD IS INSTRUCTION, BBN
(1)	020534	004537	003230	JSR	R5,ROMCLK	;CLOCK INSTRUCTION
7061	020540	120500		120500		;MOVE DATA TO THE BR REGISTER
7062	020542			ROMCLK		;NEXT WORD IS INSTRUCTION, BBN
(1)	020542	004537	003230	JSR	R5,ROMCLK	;CLOCK INSTRUCTION
7063	020546	061225		061225		;MOVE BR TO PORT 5
7064	020550	116104	000005	MOVB	5(R1),R4	;PUT 'FOUND' IN \$GDDAT
7065	020554	010205		MOV	R2,R5	
7066	020556	120204		CMPB	R2,R4	;DATA CORRECT?
7067	020560	001414		BEQ	68\$	;BR IF YES
7068	020562			BERROR	3	;DATA ERROR
(5)	020602	104455		TRAP	C\$ERDF	
(6)	020604	000003		.WORD	3	
(6)	020606	004402		.WORD	EM3	
(6)	020610	006174		.WORD	ERR3	
7069	020612			ESCAPE	SEG	
(3)	020612	104410		TRAP	C\$ESCAPE	
(3)	020614	000106		.WORD	10001\$-	
7070	020616	052711	040000	BIS	#40000,(R1)	;SET MASTER CLEAR
7071	020622	105761	000001	TSTB	1(R1)	
7072	020626	001427		BEQ	70\$	
7073	020630	100426		BMI	70\$	
7074	020632	042711	040000	BIC	#40000,(R1)	
7075						
7076						;TO RUN THIS SECTION OF CODE YOU MUST TURN SW7 OF SWITCH PACK #E28
7077						;OFF SO THAT M8207 NOT SELFSTARTING.
7078						
7079	020636			ROMCLK		;PUT BR IN PORT5
(1)	020636	004537	003230	JSR	R5,ROMCLK	;CLOCK INSTRUCTION
7080	020642	061225		061225		
7081	020644	116104	000005	MOVB	5(R1),R4	;READ IT
7082	020650	001416		BEQ	70\$	;IF ZERO, OK
7083	020652	005005		CLR	R5	
7084	020654			BERROR	3	;MASTER CLEAR
(5)	020674	104455		TRAP	C\$ERDF	
(6)	020676	000003		.WORD	3	
(6)	020700	004402		.WORD	EM3	
(6)	020702	006174		.WORD	ERR3	
7085	020704			CKLOOP		
(3)	020704	104406		TRAP	C\$CLP1	
7086						
7087	020706	105061	000001	CLRB	1(R1)	;FAILED TO CLEAR
7088	020712	005102		COM	R2	;BRG
7089	020714	000241		CLC		;CHANGE BACK TO A ONE
7090	020716	106102		ROLB	R2	;CLEAR CARRY
7091	020720	001302		BNE	67\$	;SHIFT BIT IN R5
7092	020722			ENDSEG		;DONE IF R5=0
(3)	020722					
(3)	020722	104405		TRAP	C\$ESEG	
7093	020724					
(3)	020724			ENDTST		
(3)	020724	104404		TRAP	C\$ETST	
7094						
7095	020726			BADHEAD		
(2)						;***** TEST 30 *****

```

7096                                     ;*SCRATCH PAD TEST
7097                                     ;*FLOAT A 1 THOUGH EACH SCRATCH PAD LOCATION
7098                                     ;*FLOAT A 0 THOUGH EACH SCRATCH PAD LOCATION
7099 020726 BADHEAD
      (2)                                ;***** TEST 30 *****
7100
7101 020726 BGNTST
      (3) 020726 T30::
7102 020726 MYINT
      (1) 020726 013701 002716 MOV KMCSR,R1 ;GET DEVICE ADDRESS.
7103 020732 MSTCLR ;MASTER CLEAR M8200,4,7
      (1) 020732 004537 003142 JSR R5,.,MSTCLR ;CLEAR M8200,4,7
7104 020736 005002 CLR R2 ;START AT ADDRESS ZERO
7105 020740 012705 000001 MOV #1,R5 ;START WITH BIT0
7106 020744 BGNSUB
      (3) 020744 T30.1:
      (3) 020744 104402 TRAP C$BSUB
7107 020746 1$:
      (3) 020746 104404 BGNSEG
7108 020750 042737 000017 020772 64$: BIC #17,65$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
7109 020756 050237 020772 BIS R2,65$ ;ADD ADDRESS TO INSTRUCTION
7110 020762 010561 000004 MOV R5,4(R1) ;WRITE PATTERN IN PORT4
7111 020766 ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
      (1) 020766 004537 003230 JSR R5,.,ROMCLK ;CLOCK INSTRUCTION
7112 020772 123100 65$: 123100 ;WRITE SCRATCH PAD(ADDRESS IN R2)
7113 020774 042737 000017 021012 BIC #17,66$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
7114 021002 050237 021012 BIS R2,66$ ;ADD ADDRESS TO INSTRUCTION
7115 021006 ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
      (1) 021006 004537 003230 JSR R5,.,ROMCLK ;CLOCK INSTRUCTION
7116 021012 040600 66$: 040600 ;MOVE SP TO BR
7117 021014 ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
      (1) 021014 004537 003230 JSR R5,.,ROMCLK ;CLOCK INSTRUCTION
7118 021020 061225 ;MOVE BR TO PORT5
7119 021022 010537 002636 MOV R5,$GDDAT ;PUT 'EXPECTED' IN $GDDAT
7120 021026 116104 000005 MOVB 5(R1),R4 ;PUT 'FOUND' IN R4
7121 021032 123704 002636 CMPB $GDDAT,R4 ;DATA CORRECT
7122 021036 001414 BEQ 67$ ;BR IF YES
7123 021040 RERROR 4 ;DATA ERROR
      (5) 021060 104455 TRAP C$ERDF
      (6) 021062 000004 .WORD 4
      (6) 021064 004430 .WORD EM4
      (6) 021066 006252 .WORD ERR4
7124 021070 67$: ESCAPE SEG
      (3) 021070 104410 TRAP C$ESCAPE
      (3) 021072 000010 .WORD 10000$-
7125 021074 000241 CLC ;CLEAR CARRY
7126 021076 106105 ROLB R5 ;SHIFT BIT IN R5
7127 021100 001323 BNE 64$ ;DONE IF R5=0
7128 021102 ENDSEG
      (3) 021102 10000$:
      (3) 021102 104405 TRAP C$ESEG
7129 021104 012705 000001 MOV #1,R5 ;START WITH BIT0
7130 021110 BGNSEG
      (3) 021110 104404 TRAP C$BSEG
7131
7132 021112 005105 73$: COM R5 ;CHANGE TO FLOATING ZERO
  
```

7133	021114	042737	000017	021136	69\$:	BIC	#17,70\$	:CLEAR ADDRESS FIELD OF INSTRUCTION
7134	021122	050237	021136			BIS	R2,70\$	:ADD ADDRESS TO INSTRUCTION
7135	021126	010561	000004			MOV	R5,4(R1)	:WRITE PATTERN IN PORT4
7136	021132					ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	021132	004537	003230			JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
7137	021136	123100		021156	70\$:	123100		:WRITE SCRATCH PAD(ADDRESS IN R2)
7138	021140	042737	000017			BIC	#17,71\$	:CLEAR ADDRESS FIELD OF INSTRUCTION
7139	021146	050237	021156			BIS	R2,71\$	:ADD ADDRESS TO INSTRUCTION
7140	021152					ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	021152	004537	003230			JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
7141	021156	040600			71\$:	040600		:MOVE SP TO BR
7142	021160					ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	021160	004537	003230			JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
7143	021164	061225				061225		:MOVE BR TO PORT5
7144	021166	010537	002636			MOV	R5,\$GDDAT	:PUT 'EXPECTED' IN \$GDDAT
7145	021172	116104	000005			MOVB	5(R1),R4	:PUT 'FOUND' IN \$GDDAT
7146	021176	123704	002636			CMPS	\$GDDAT,R4	:DATA CORRECT?
7147	021202	001414				BEG	72\$	:BR IF YES
7148	021204					RERROR	4	:DATA ERROR
(5)	021224	104455				TRAP	C\$ERDF	
(6)	021226	000004				.WORD	4	
(6)	021230	004430				.WORD	EM4	
(6)	021232	006252				.WORD	ERR4	
7149	021234				72\$:	ESCAPE	TST	
(3)	021234	104410				TRAP	C\$ESCAPE	
(3)	021236	000032				.WORD	L10107-	
7150	021240	005105				COM	R5	:CHANGE BACK TO A ONE
7151	021242	000241				CLC		:CLEAR CARRY
7152	021244	106105				ROLB	R5	:SHIFT BIT IN R5
7153	021246	001321				BNE	73\$	:DONE IF R5=0
7154	021250				ENDSEG			
(3)	021250				10001\$:			
(3)	021250	104405				TRAP	C\$ESEG	
7155	021252	012705	000001			MOV	#1,R5	:RESTART AT BIT 0
7156	021256	005202				INC	R2	:NEXT SP ADDRESS
7157	021260	022702	000020			CMP	#20,R2	:LAST ADDRESS?
7158	021264	001230				BNE	1\$	:BR IF NO
7159	021266					ENDSUB		
(3)	021266				-10110:			
(3)	021266	104403				TRAP	C\$ESUB	
7160	021270				ENDTST			
(3)	021270				L10107:			
(3)	021270	104401				TRAP	C\$ETST	
7161	021272					BADHEAD		
(2)						:***** TEST 31 *****		
7163						:*SCRATCH PAD DUAL ADDRESSING TEST		
7164						:*WRITE AN INCREMENTING PATTERN IN ALL SP LOCATIONS		
7165						:*READ ALL SP LOCATIONS TO VERIFY CORRECT ADDRESSING		
7166	021272					BADHEAD		
(2)						:***** TEST 31 *****		
7167								
7168	021272				BGNTST			
(3)	021272				T31::			
7169	021272					MSTCLR		:MASTER CLEAR M8200,4,7
(1)	021272	004537	003142			JSR	R5,.MSTCLR	:CLEAR M8200,4,7



7170	021276	012705	000001		MOV	#1,R5	;START WITH A 1
7171	021302	005003			CLR	R3	;ADDRESS 0
7172	021304				MYINT		
(1)	021304	013701	002716		MOV	KMCSR,R1	;GET DEVICE ADDRESS.
7173	021310				BGNSEG		
(3)	021310	104404			TRAP	C\$BSEG	
7174	021312	010302			MOV	R3,R2	;MOVE ADDRESS TO R2
7175	021314	042737	000017	021336	BIC	#17,2\$	;CLEAR ADDRESS FIELD
7176	021322	050237	021336		BIS	R2,2\$	;ADD ADDRESS TO INSTRUCTION
7177	021326	010561	000004		MOV	R5,4(R1)	;WRITE PATTERN IN PORT4
7178	021332				ROMCLK		;NEXT WORD IS INSTRUCTION, BBN
(1)	021332	004537	003230		JSR	R5,.ROMCLK	;CLOCK INSTRUCTION
7179	021336	123100			123100		;WRITE SP(ADDRESS IN R2)
7180	021340	042737	000017	021356	BIC	#17,3\$	;CLEAR ADDRESS FIELD OF INSTRUCTION

CZDMPAO M8207 STATIC DIAG #1  
CZDMPA.P11 17-JUL-79 14:33

MACY11 30A(1052) 17-JUL-79 14:39  
HARDWARE TESTS

K 11  
PAGE 52

SEQ 0140

7182 021346 050237 021356  
7183 021352  
(1) 021352 004537 003230

BIS R2,3\$  
ROMCLK  
JSR R5,ROMCLK

;ADD ADDRESS TO INSTRUCTION  
;NEXT WORD IS INSTRUCTION, BBN  
;CLOCK INSTRUCTION

Address	Hex	Dec	Label	Instruction	Comment
7185	021356	060600	3\$:	60600	;MOVE SP TO BR
7186	021360			ROMCLK	;NEXT WORD IS INSTRUCTION, BBN
(1)	021360	004537	003230	JSR R5,ROMCLK	;CLOCK INSTRUCTION
7187	021364	061225		61225	;MOVE BR TO PORT5
7188	021366	010537	002636	MOV R5,\$GDDAT	;PUT 'EXPECTED' IN \$GDDAT
7189	021372	116104	000005	MOVB 5(R1),R4	;PUT 'FOUND' IN R4
7190	021376	123704	002636	CMPS \$GDDAT,R4	;DATA CORRECT
7191	021402	001414		BEQ 4\$	;BR IF YES
7192	021404			RERROR 4	;DATA ERROR
(5)	021424	104455		TRAP C\$ERDF	
(6)	021426	000004		.WORD 4	
(6)	021430	004430		.WORD EM4	
(6)	021432	006252		.WORD ERR4	
7193	021434		4\$:	ESCAPE SEG	
(3)	021434	104410		TRAP C\$ESCAPE	
(3)	021436	000014		.WORD 10000\$-	
7194	021440	005205		INC R5	;INCREMENT PATTERN
7195	021442	005203		INC R3	;NEXT ADDRESS
7196	021444	022703	000020	CMP #20,R3	;LAST ADDRESS DONE?
7197	021450	001320		BNE 1\$	;BR IF NO
7198	021452			ENDSEG	
(3)	021452		10000\$:		
(3)	021452	104405		TRAP C\$ESEG	
7199	021454	012705	000001	MOV #1,R5	;RESTART PATTERN AT 1
7200	021460	005003		CLR R3	;RESTART AT ADDRESS ZERO
7201	021462			BGNSEG	
(3)	021462	104404		TRAP C\$BSEG	
7202	021464	010302		MOV R3,R2	;PUT ADDRESS IN R2
7203	021466	042737	000017	BIC #17,6\$	;CLEAR ADDRESS FIELD OF INSTRUCTION
7204	021474	050237	021504	BIS R2,6\$	;ADD ADDRESS TO INSTRUCTION
7205	021500			ROMCLK	;NEXT WORD IS INSTRUCTION, BBN
(1)	021500	004537	003230	JSR R5,ROMCLK	;CLOCK INSTRUCTION
7206	021504	060600		60600	;MOV SP TO BR
7207	021506			ROMCLK	;NEXT WORD IS INSTRUCTION, BBN
(1)	021506	004537	003230	JSR R5,ROMCLK	;CLOCK INSTRUCTION
7208	021512	061225		61225	;MOV BR TO PORT5
7209	021514	010537	002636	MOV R5,\$GDDAT	;PUT 'EXPECTED' IN \$GDDAT
7210	021520	116104	000005	MOVB 5(R1),R4	;PUT 'FOUND' IN \$GDDAT
7211	021524	123704	002636	CMPS \$GDDAT,R4	;DATA CORRECT?
7212	021530	001414		BEQ 7\$	;BR IF YES
7213	021532			RERROR 5	;SP ADDRESSING ERROR
(5)	021552	104455		TRAP C\$ERDF	
(6)	021554	000005		.WORD 5	
(6)	021556	004456		.WORD EM5	
(6)	021560	006334		.WORD ERR5	
7214	021562		7\$:	ESCAPE SEG	
(3)	021562	104410		TRAP C\$ESCAPE	
(3)	021564	000014		.WORD 10001\$-	
7215	021566	005205		INC R5	;INCREMENT PATTERN
7216	021570	005203		INC R3	;NEXT ADDRESS
7217	021572	022703	000020	CMP #20,R3	;LAST ADDRESS DONE?
7218	021576	001332		BNE 5\$	;BR IF NO
7219	021600			ENDSEG	
(3)	021600		10001\$:		
(3)	021600	104405		TRAP C\$ESEG	
7220	021602		ENDTST		

```

(3) 021602
(3) 021602 104401
7221
7222 021604
(2)
7223
7224
7225 021604
(2)
7226
7227 021604
(3) 021604
7228 021604
(1) 021604 013701 002716
7229 021610
(3) 021610 104433
7230 021612 005011
7231 021614 004537 003536
7232 021620 021740
7233 021622 021712
7234 021624 000340 000340
7235 021630
(3) 021630 012700 000340
(3) 021634 104441
7236 021636 012761 000200 000004
7237 021644
(1) 021644 004537 003230
7238 021650 121111
7239 021652
(3) 021652 012700 000000
(3) 021656 104441
7240 021660 000240
7241 021662
(5) 021700 104455
(6) 021702 000037
(6) 021704 005276
(6) 021706 010130
7242 021710 000415
7243 021712
(5) 021730 104455
(6) 021732 000040
(6) 021734 005325
(6) 021736 010156
7244 021740 062706 000004
7245 021744
7246 021744
(3) 021744
(3) 021744 104401
7247
7248 021746
(2)
7249
7250
7251 021746
(2)
7252
  
```

```

L10111:
  TRAP    C$ETST

  BADHEAD
  ;***** TEST 32 *****
  ;*INTERRUPT TEST
  ;*TEST THAT DEVICE CAN INTERRUPT TO VECTOR A
  BADHEAD
  ;***** TEST 32 *****

  BGNTST
  T32::

  MYINT
  MOV     KMCSR,R1      ;GET DEVICE ADDRESS.
  BRESET      ;BUS RESET
  TRAP     C$RESET
  CLR      (R1)        ;CLEAR RUN
  JSR      R5,SETVEC    ;SET UP VECTORS
  3$      ;XX0
  2$      ;XX4
  .WORD    340,340      ;LEVEL 7
  SETPRI   #PRI07       ;PS = LEVEL 7
  MOV      #PRI07,R0
  TRAP     C$SPRI
  MOV      #200,4(R1)   ;WRITE PORT4
  ROMCLK   R5,.ROMCLK   ;NEXT WORD IS INSTRUCTION, BBN
  JSR      121111       ;CLOCK INSTRUCTION
  SETPRI   #PRI00       ;SET BR RQ IN IBUS* REG 11
  MOV      #PRI00,R0    ;ALLOW INTERRUPT
  TRAP     C$SPRI
  NOP
  ERROR    31           ;NO INTERRUPT
  TRAP     C$ERDF
  .WORD    31
  .WORD    EM31
  .WORD    ERR31
  BR       4$
  2$:      ERROR    32           ;WRONG VECTOR
  TRAP     C$ERDF
  .WORD    32
  .WORD    EM32
  .WORD    ERR32
  3$:      ADD      #4,SP      ;RESET STACK
  4$:
  ENDTST
  L10112:
  TRAP     C$ETST

  BADHEAD
  ;***** TEST 33 *****
  ;*INTERRUPT TEST
  ;*TEST THAT DEVICE CAN INTERRUPT TO VECTOR B
  BADHEAD
  ;***** TEST 33 *****
  
```

```

7253 021746          BGNTST
      (3) 021746      T33::
7254 021746          MYINT
      (1) 021746 013701 002716      MOV      KMCSR,R1          ;GET DEVICE ADDRESS.
7255 021752          MSTCLR          ;MASTER CLEAR M8200,4,7
      (1) 021752 004537 003142      JSR      R5,,MSTCLR          ;CLEAR M8200,4,7
7256 021756 004537 003536      JSR      R5,SETVEC          ;SET UP VECTORS
7257 021762 022054          2$      ;XX0
7258 021764 022102          3$      ;XX4
7259 021766 000340 000340          .WORD 340,340          ;LEVEL 7
7260 021772          1$:      SETPRI #PRI07          ;PS = LEVEL 7
      (3) 021772 012700 000340      MOV      #PRI07,R0
      (3) 021776 104441          TRAP      C$SPRI
7261 022000 012761 000300 000004      MOV      #300,4(R1)          ;WRITE PORT4
7262 022006          ROMCLK          ;NEXT WORD IS INSTRUCTION, BBN
      (1) 022006 004537 003230      JSR      R5,,ROMCLK          ;CLOCK INSTRUCTION
7263 022012 121111          121111          ;SET BR RQ IN IBUS* REG 11
7264 022014          SETPRI #PRI00          ;ALLOW INTERRUPT
      (3) 022014 012700 000000      MOV      #PRI00,R0
      (3) 022020 104441          TRAP      C$SPRI
7265 022022 000240          NOP
7266 022024          ERROR 31          ;NO INTERRUPT
      (5) 022042 104455          TRAP      C$ERDF
      (6) 022044 000037          .WORD 31
      (6) 022046 005276          .WORD EM31
      (6) 022050 010130          .WORD ERR31
7267 022052 000415          BR      4$
7268 022054          2$:      ERROR 32          ;WRONG VECTOR
      (5) 022072 104455          TRAP      C$ERDF
      (6) 022074 000040          .WORD 32
      (6) 022076 005325          .WORD EM32
      (6) 022100 010156          .WORD ERR32
7269 022102 062706 000004          3$:      ADD      #4,SP          ;RESET STACK
7270 022106          4$:
7271 022106          ENDTST
      (3) 022106          L10113:
      (3) 022106 104401          TRAP      C$ETST
7272 022110          BADHEAD
7273 022110          ;***** TEST 34 *****
      (2) 022110          ;*PRIORITY INTERRUPT TEST
7274 022110          ;*SET PS TO ALL BR LEVELS EQUAL OR GREATER THAN
7275 022110          ;*THE M8200,4,7 LEVEL. VERIFY THAT M8200,4,7 DOES NOT INTERRUPT
7276 022110          BADHEAD
7277 022110          ;***** TEST 34 *****
      (2) 022110
7278 022110
7279 022110          BGNTST
      (3) 022110      T34::
7280 022110          MYINT
      (1) 022110 013701 002716      MOV      KMCSR,R1          ;GET DEVICE ADDRESS.
7281 022114          MSTCLR          ;MASTER CLEAR M8200,4,7
      (1) 022114 004537 003142      JSR      R5,,MSTCLR          ;CLEAR M8200,4,7
7282 022120 012704 000340      MOV      #340,R4          ;PUT LEVEL 7 IN R2
7283 022124          SETPRI R4          ;SET PRIORITY TO 7
      (3) 022124 010400          MOV      R4,R0
      (3) 022126 104441          TRAP      C$SPRI
  
```

7284	022130	013705	002700		MOV	STAT1,R5	:GET BR LEVEL OF M8200,4,7
7285	022134	006205			ASR	R5	:SHIFT R5 4 TIMES
7286	022136	006205			ASR	R5	:TO GET PROPER LEVEL
7287	022140	006205			ASR	R5	
7288	022142	006205			ASR	R5	
7289	022144	042705	177437		BIC	#177437,R5	:CLEAR UNWANTED BITS
7290	022150	010537	002636		MOV	R5,\$GDDAT	
7291	022154	004537	003536		JSR	R5,SETVEC	:SET UP VECTORS
7292	022160	022224			2\$		:A VECTOR
7293	022162	022224			2\$		:B VECTOR
7294	022164	000340	000340		.WORD	340,340	:PRIORITY 7
7295	022170	012761	C00200	000004	4\$: MOV	#200,4(R1)	:LOAD PORT4
7296	022176				ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	022176	004537	003230		JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
7297	022202	121111			121111		:SET BR REQUEST
7298	022204			5\$:	SETPRI	R4	:PUT LEVEL IN R2 IN PS
(3)	022204	010400			MOV	R4,R0	
(3)	022206	104441			TRAP	C\$SPRI	
7299	022210	000240			NOP		
7300	022212	020504			CMP	R5,R4	:IS PRESENT PS LEVEL = TO M8200,4,7 LEVEL
7301	022214	001420			BEQ	1\$	:BR IF YES
7302	022216	162704	000040		SUB	#40,R4	:NO GET NEXT LOWER LEVEL IN R2
7303	022222	000770			BR	5\$	:AND CONTINUE WITH TEST
7304	022224			2\$:	BRESET		
(3)	022224	104433			TRAP	C\$RESET	
7305	022226				ERROR	33	:ERROR UNEXPECTED INTERRUPT
(5)	022244	104455			TRAP	C\$ERDF	
(6)	022246	000041			.WORD	33	
(6)	022250	005364			.WORD	EM33	
(6)	022252	010204			.WORD	ERR33	
7306	022254	000002			RTI		
7307	022256			1\$:	MSTCLR		
(1)	022256	004537	003142		JSR	R5,.MSTCLR	:CLEAR M8200,4,7
7308	022262				ENDTST		
(3)	022262			L10114:			
(3)	022262	104401		.	TRAP	C\$ETST	
7309							
7310	022264				BADHEAD		
(2)					:***** TEST 35 *****		
7311					:*PRIORITY INTERRUPT TESTS		
7312					:*SET PS TO ALL BR LEVELS LESS THAN THE M8200,4,7 LEVEL		
7313					:*VERIFY THAT M8200,4,7 WILL INTERRUPT		
7314	022264				BADHEAD		
(2)					:***** TEST 35 *****		
7315							
7316	022264			BGNTST			
(3)	022264			T35::			
7317	022264				MYINT		
(1)	022264	013701	002716		MOV	KMCSR,R1	:GET DEVICE ADDRESS.
7318	022270				MSTCLR		:MASTER CLEAR M8200,4,7
(1)	022270	004537	003142		JSR	R5,.MSTCLR	:CLEAR M8200,4,7
7319	022274	012704	000340		MOV	#340,R4	:PUT LEVEL 7 IN R2
7320	022300				SETPRI	R4	:SET PRIORITY TO 7
(3)	022300	010400			MOV	R4,R0	
(3)	022302	104441			TRAP	C\$SPRI	
7321	022304	013705	002700		MOV	STAT1,R5	:GET BR LEVEL OF M8200,4,7

7322	022310	006205			ASR	R5		;SHIFT R5 4 TIMES
7323	022312	006205			ASR	R5		;TO GET PROPER LEVEL
7324	022314	006205			ASR	R5		
7325	022316	006205			ASR	R5		
7326	022320	042705	177437		BIC	#177437,R5		;CLEAR UNWANTED BITS
7327	022324	010502			MOV	R5,R2		;PUT M8200,4,7 LEVEL IN R2
7328	022326	162702	000040		SUB	#40,R2		;GET NEXT LOWER LEVEL IN R2
7329	022332	004537	003536		JSR	R5,SETVEC		;SET UP VECTORS
7330	022336	022420			2\$			;A VECTOR
7331	022340	022426			3\$			;B VECTOR
7332	022342	000340	000340		.WORD	340,340		;PRIORITY 7
7333	022346	012761	000200	000004	4\$: MOV	#200,4(R1)		;LOAD PORT4
7334	022354				ROMCLK			;NEXT WORD IS INSTRUCTION, BBN
(1)	022354	004537	003230		JSR	R5,.ROMCLK		;CLOCK INSTRUCTION
7335	022360	121111			121111			;SET BR REQUEST
7336	022362			5\$:	SETPRI	R2		;PUT LEVEL IN R2 IN PS
(3)	022362	010200			MOV	R2,R0		
(3)	022364	104441			TRAP	C\$SPRI		
7337	022366	000240			NOP			
7338	022370				ERROR	31		;ERROR, NO INTERRUPT
(5)	022406	104455			TRAP	C\$ERDF		
(6)	022410	000037			.WORD	31		
(6)	022412	005276			.WORD	EM31		
(6)	022414	010130			.WORD	ERR31		
7339	022416	000421		6\$:	BR	1\$		
7340	022420	012716	022416	2\$:	MOV	#6\$, (SP)		;SET UP FOR RTI
7341	022424	000002			RTI			
7342	022426			3\$:	ERROR	32		;ERROR, WRONG VECTOR
(5)	022444	104455			TRAP	C\$ERDF		
(6)	022446	000040			.WORD	32		
(6)	022450	005325			.WORD	EM32		
(6)	022452	010156			.WORD	ERR32		
7343	022454	012716	022462		MOV	#1\$, (SP)		;SET UP FOR RTI
7344	022460	000002			RTI			
7345	022462			1\$:	MSTCLR			
(1)	022462	004537	003142		JSR	R5,.MSTCLR		;CLEAR M8200,4,7
7346	022466				ENDTST			
(3)	022466			L10115:				
(3)	022466	104401			TRAP	C\$ETST		
7347								
7348	022470				BADHEAD			
(2)					;***** TEST 36 *****			
7349					;*NPR TEST			
7350					;*TEST OF DATO, 1 WORD FROM UPROC TO 11 MEMORY			
7351	022470				BADHEAD			
(2)					;***** TEST 36 *****			
7352								
7353	022470			BGNTST				
(3)	022470			T36::				
7354	022470				BRESET			;BUS RESET
(3)	022470	104433			TRAP	C\$RESET		
7355								
7356	022472				MYINT			
(1)	022472	013701	002716		MOV	KMCSR,R1		;GET DEVICE ADDRESS.
7357	022476	005011			CLR	(R1)		;CLEAR RUN
7358	022500	005061	000004		CLR	4(R1)		;CLR PORT4

7359	022504	004537	003560		JSR	R5,NPRSET	:SET UP IBUS REG 0-7
7360	022510	000000			0		:IN DATA
7361	022512	177777			-1		:OUT DATA
7362	022514	022630			3\$		:IN BA
7363	022516	022626			2\$		:OUT BA
7364	022520	005037	022626		CLR	2\$	:CLEAR 2\$
7365	022524	005061	000004		CLR	4(R1)	:CLEAR PORT 4
7366	022530				ROMCLK		:NOW MOVE TO IBUS* <11>
(1)	022530	004537	003230		JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
7367	022534	121111			121111		
7368	022536	012761	000021	000004	MOV	#21,4(R1)	:WRITE PORT4
7369	022544				ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	022544	004537	003230		JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
7370	022550	121110			121110		:SET NPR BITS IN IBUS* REG 10
7371	022552	000240			NOP		
7372	022554	012737	177777	002636	MOV	#-1,\$GDDAT	:PUT 'EXPECTED' IN \$GDDAT
7373	022562	013704	022626		MOV	2\$,R4	:PUT 'FOUND' IN R4
7374	022566	023704	002636		CMP	\$GDDAT,R4	:DATA CORRECT?
7375	022572	001413			BEQ	4\$	:BR IF YES
7376	022574				ERROR	11,YES	:ERROR NPR FAILED
(5)	022606	104455			TRAP	C\$ERDF	
(6)	022610	000013			.WORD	11	
(6)	022612	004640			.WORD	EM11	
(6)	022614	006644			.WORD	ERR11	
7377	022616				ESCAPE	TST	
(3)	022616	104410			TRAP	C\$ESCAPE	
(3)	022620	000012			.WORD	L10116-	
7378	022622				EXIT	TST	
(3)	022622	104432			TRAP	C\$EXIT	
(3)	022624	000006			.WORD	L10116-	
7379	022626	000000			0		:OUT BA
7380	022630	000000			0		:IN BA
7381	022632				ENDTST		
(3)	022632				L10116:		
(3)	022632	104401			TRAP	C\$ETST	
7382							
7383	022634				BADHEAD		
(2)					:***** TEST 37 *****		
7384					:*NPR TEST		
7385					:*TEST OF DATI, 1 WORD FROM 11 MEMORY TO UPROC		
7386	022634				BADHEAD		
(2)					:***** TEST 37 *****		
7387							
7388	022634				BGNTST		
(3)	022634				T37::		
7389	022634				MYINT		
(1)	022634	013701	002716		MOV	KMCSR,R1	:GET DEVICE ADDRESS.
7390	022640				MSTCLR		:MASTER CLEAR M8200,4,7
(1)	022640	004537	003142		JSR	R5,.MSTCLR	:CLEAR M8200,4,7
7391	022644	005061	000004		CLR	4(R1)	:CLR PORT4
7392	022650	004537	003560		JSR	R5,NPRSET	:SET UP IBUS REG 0-7
7393	022654	000000			0		:IN DATA
7394	022656	177777			-1		:OUT DATA
7395	022660	023000			3\$		:IN BA
7396	022662	022776			2\$		:OUT BA
7397	022664	012737	177777	023000	MOV	#-1,3\$	:PUT DATA IN 3\$



```

7398 022672 012761 000001 000004      MOV    #1,4(R1)      ;WRITE PORT4
7399 022700      ROMCLK      ;NEXT WORD IS INSTRUCTION, BBN
(1) 022700 004537 003230      JSR    R5,ROMCLK      ;CLOCK INSTRUCTION
7400 022704 121110      121110      ;SET NPR BITS IN IBUS* REG 11
7401 022706 000240      NOP
7402 022710 012737 177777 002636      MOV    #-1,$GDDAT      ;PUT 'EXPECTED' IN $GDDAT
7403 022716      ROMCLK      ;NEXT WORD IS INSTRUCTION, BBN
(1) 022716 004537 003230      JSR    R5,ROMCLK      ;CLOCK INSTRUCTION
7404 022722 021004      021004      ;MOVE IN DATA LOW BYTE TO PORT4
7405 022724      ROMCLK      ;NEXT WORD IS INSTRUCTION, BBN
(1) 022724 004537 003230      JSR    R5,ROMCLK      ;CLOCK INSTRUCTION
7406 022730 021025      021025      ;MOVE IN DATA HIGH BYTE TO PORT5
7407 022732 016104 000004      MOV    4(R1),R4      ;PUT 'FOUND' IN R4
7408 022736 023704 002636      CMP    $GDDAT,R4      ;DATA CORRECT?
7409 022742 001413      BEQ    4$
7410 022744      ERROR      11,YES      ;BR IF YES
(5) 022756 104455      TRAP    C$ERDF      ;ERROR NPR FAILED
(6) 022760 000013      .WORD    11
(6) 022762 004640      .WORD    EM11
(6) 022764 006644      .WORD    ERR11
7411 022766      ESCAPE      TST
(3) 022766 104410      TRAP    C$ESCAPE
(3) 022770 000012      .WORD    L10117-
7412 022772      4$:      EXIT      TST
(3) 022772 104432      TRAP    C$EXIT
(3) 022774 000006      .WORD    L10117-
7413 022776 000000      2$:      0
7414 023000 000000      3$:      0      ;OUT BA
7415 023002      ENDTST      ;IN BA
(3) 023002      L10117:
(3) 023002 104401      TRAP    C$ETST
7416
7417 023004      BADHEAD
(2)
7418      ;***** TEST 38 *****
7419      ;*NPR TEST
7420 023004      ;*TEST OF DATOB, 1 BYTE FROM UPROC TO 11 MEMORY
(2)      BADHEAD
7421      ;***** TEST 38 *****
7422 023004      BGNTST
(3) 023004      T38::
7423 023004      MYINT
(1) 023004 013701 002716      MOV    KMCSR,R1      ;GET DEVICE ADDRESS.
7424 023010      MSTCLR      ;MASTER CLEAR M8200,4,7
(1) 023010 004537 003142      JSR    R5,MSTCLR      ;CLEAR M8200,4,7
7425 023014 005061 000004      CLR    4(R1)      ;CLR PORT4
7426 023020 004537 003560      JSR    R5,NPRSET      ;SET UP IBUS REG 0-7
7427 023024 000000      0      ;IN DATA
7428 023026 177777      -1      ;OUT DATA
7429 023030 023144      3$      ;IN BA
7430 023032 023143      2$+1      ;OUT BA
7431 023034 005037 023142      CLR    2$      ;CLEAR 2$
7432 023040 005061 000004      CLR    4(R1)      ;CLEAR PORT 4
7433 023044      ROMCLK      ;NOW MOVE IT TO IBUS*<11>
(1) 023044 004537 003230      JSR    R5,ROMCLK      ;CLOCK INSTRUCTION
7434 023050 121111      121111

```

```

7435 023052 012761 000221 000004      MOV      #221,4(R1)      ;WRITE PORT4
7436 023060      ROMCLK      ;NEXT WORD IS INSTRUCTION, BBN
(1) 023060 004537 003230      JSR      R5,ROMCLK      ;CLOCK INSTRUCTION
7437 023064 121110      121110      ;SET NPR BITS IN IBUS* REG 11
7438 023066 000240      NOP
7439 023070 012737 177400 002636      MOV      #177400,$GDDAT  ;PUT 'EXPECTED' IN $GDDAT
7440 023076 013704 023142      MOV      2$,R4      ;PUT 'FOUND' IN R4
7441 023102 023704 002636      CMP      $GDDAT,R4      ;DATA CORRECT?
7442 023106 001413      BEQ      4$      ;BR IF YES
7443 023110      ERROR      11,YES      ;ERROR NPR FAILED
(5) 023122 104455      TRAP      C$ERDF
(6) 023124 000013      .WORD      11
(6) 023126 004640      .WORD      EM11
(6) 023130 006644      .WORD      ERR11
7444 023132      ESCAPE      TST
(3) 023132 104410      TRAP      C$ESCAPE
(3) 023134 000012      .WORD      L10120-.
7445 023136      4$:      EXIT      TST
(3) 023136 104432      TRAP      C$EXIT
(3) 023140 000006      .WORD      L10120-.
7446 023142 000000      2$:      0      ;OUT BA
7447 023144 000000      3$:      0      ;IN BA
7448 023146      ENDTST
(3) 023146 104401      L10120:
7449      TRAP      C$ETST
7450 023150      BADHEAD
(2)      ;***** TEST 39 *****
7451      ;*TEST OF EA BITS 16 AND 17
7452      ;*DO A DATO TO AN ADDRESS USING OUT BA BITS 16 AND 17
7453      ;*VERIFY CORRECT RESULTS
7454 023150      BADHEAD
(2)      ;***** TEST 39 *****
7455
7456 023150      BGNTST
(3) 023150      T39::
7457 023150      MSTCLR      ;MASTER CLEAR M8200,4,7
(1) 023150 004537 003142      JSR      R5,MSTCLR      ;CLEAR M8200,4,7
7458 023154      MYINT
(1) 023154 013701 002716      MOV      KMCSR,R1      ;GET DEVICE ADDRESS.
7459 023160 013737 002726 023206      MOV      KMP06,1$      ;USE SEL4 FOR ADDRESS
7460 023166 013737 002726 023204      MOV      KMP06,2$      ;USE SEL4 FOR ADDRESS
7461 023174 004537 003560      JSR      R5,NPRSET      ;LOAD BA AND DATA
7462 023200 000000      0      ;IN DATA
7463 023202 125252      125252      ;OUT DATA
7464 023204 000000      2$:      0      ;IN BA
7465 023206 000000      1$:      0      ;OUT BA
7466 023210 012761 000014 000004      MOV      #14,4(R1)      ;LOAD SEL 4 WITH OUT BA16 AND 17
7467 023216      ROMCLK      ;NEXT WORD IS INSTRUCTION, BBN
(1) 023216 004537 003230      JSR      R5,ROMCLK      ;CLOCK INSTRUCTION
7468 023222 121111      121111      ;SET OUTBA 16 AND 17
7469 023224 012761 000021 000004      MOV      #21,4(R1)      ;LOAD SEL4
7470 023232 012711 003000      MOV      #BIT9:BIT10,(R1)
7471 023236 012761 121110 000006      MOV      #121110,6(R1)      ;PUT INSTRUCTION IN SEL6
7472 023244 052711 000400      BIS      #BIT8,(R1)      ;CLOCK IT.
7473 023250 000240      NOP      ;WAIT FOR NPR

```

```

7474 023252 012737 121110 002636      MOV      #121110,$GDDAT ;PUT 'EXPECTED' IN $GDDAT
7475 023260 000240                      NOP
7476 023262 000240                      NOP
7477                                     ;OK,LISTEN UP.EXPLANATION TIME.
7478                                     ;
7479                                     ;ON THE NPR OUT,THE DATA ENDED UP
7480                                     ;IN THE IBUS(NOT IBUS*) SENCE SEL A
7481                                     ;WAS ONLY SELECTED IN THE NPR CYCLE.
7482                                     ;THAT IS,WE DIDN'T REALLY DO AN NPR TO
7483                                     ;PORT 6,THE NPR OUT REALLY ENDED UP IN
7484                                     ;OUT DATA LOW,AND OUT DATA HIGH
7485                                     ;(IBUS <2> AND IBUS <3>).
7486
7487                                     ;WHAT WE'RE DOING NEXT IS READING IBUS 2&3
7488                                     ;TO SEE IF THE DATA GOT XFERRED CORRECTLY.
7489 023264                                     ROMCLK
7490 (1) 023264 004537 003230      JSR      R5,ROMCLK ;CLOCK INSTRUCTION
7491 023270 021044      021044      ;READ IBUS <2> PUT IN PORT 4
7492 023272                                     ROMCLK
7493 (1) 023272 004537 003230      JSR      R5,ROMCLK ;CLOCK INSTRUCTION
7494 023276 021065      021065      ;READ IBUS <3> PUT IN PORT 5
7495 023300 016104 000004      MOV      4(R1),R4 ;PUT 'FOUND' IN R4
7496 023304 023704 002636      CMP      $GDDAT,R4 ;CORRECT RESULTS?
7497 023310 001411      BEQ      3$ ;BR IF YES
7498 023312                                     ERROR 11,YES ;ERROR BA 16 AND 17 FAILED
7499 (5) 023324 104455      TRAP      C$ERDF
7500 (6) 023326 000013      .WORD     11
7501 (6) 023330 004640      .WORD     EM11
7502 (6) 023332 006644      .WORD     ERR11
7503
7504 3$:
7505 ENDTST
7506 L10121:
7507 TRAP      C$ETST
7508
7509 BADHEAD
7510 ***** TEST 40 *****
7511 ;*TEST OF EA BITS 16 AND 17
7512 ;*DO A DATI USING IN BA BITS 16 AND 17
7513 ;*VERIFY CORRECT RESULTS
7514 ;*IN ORDER TO DO THIS TEST, WE WILL READ THE DATA FROM THE
7515 ;*CONSOL TTY CSR IF ONE EXSITS
7516 ;*IF NO CONSOL TTY CSR AT ADDRESS 177560, THIS TEST
7517 ;*WILL BE SKIPPED
7518 BADHEAD
7519 ***** TEST 40 *****
7520
7521 BGNTST
7522 T40::
7523 MYINT
7524 MOV      KMCSR,R1 ;GET DEVICE ADDRESS.
7525 MSTCLR   ;MASTER CLEAR M8200,4,7
7526 JSR      R5,MSTCLR ;CLEAR M8200,4,7
7527 MOV      #TOUTT,4 ;SET UP FOR TRAP IN CASE IF NO
7528 MOV      #340,6 ;TTY AT ADDRESS 177560
7529 TST      177560 ;ADDRESS THE TTY-TRAPS HERE IF NONE.
7530 MOV      #177560,1$ ;JSE SEL4 FOR ADDRESS
7531
7532 STOP:
7533
7534
7535
7536
7537
7538
7539
7540
7541
7542
7543
7544
7545
7546
7547
7548
7549
7550
7551
7552
7553
7554
7555
7556
7557
7558
7559
7560
7561
7562
7563
7564
7565
7566
7567
7568
7569
7570
7571
7572
7573
7574
7575
7576
7577
7578
7579
7580
7581
7582
7583
7584
7585
7586
7587
7588
7589
7590
7591
7592
7593
7594
7595
7596
7597
7598
7599
7600
7601
7602
7603
7604
7605
7606
7607
7608
7609
7610
7611
7612
7613
7614
7615
7616
7617
7618
7619
7620
7621
7622
7623
7624
7625
7626
7627
7628
7629
7630
7631
7632
7633
7634
7635
7636
7637
7638
7639
7640
7641
7642
7643
7644
7645
7646
7647
7648
7649
7650
7651
7652
7653
7654
7655
7656
7657
7658
7659
7660
7661
7662
7663
7664
7665
7666
7667
7668
7669
7670
7671
7672
7673
7674
7675
7676
7677
7678
7679
7680
7681
7682
7683
7684
7685
7686
7687
7688
7689
7690
7691
7692
7693
7694
7695
7696
7697
7698
7699
7700
7701
7702
7703
7704
7705
7706
7707
7708
7709
7710
7711
7712
7713
7714
7715
7716
7717
7718
7719
7720
7721
7722
7723
7724
7725
7726
7727
7728
7729
7730
7731
7732
7733
7734
7735
7736
7737
7738
7739
7740
7741
7742
7743
7744
7745
7746
7747
7748
7749
7750
7751
7752
7753
7754
7755
7756
7757
7758
7759
7760
7761
7762
7763
7764
7765
7766
7767
7768
7769
7770
7771
7772
7773
7774
7775
7776
7777
7778
7779
7780
7781
7782
7783
7784
7785
7786
7787
7788
7789
7790
7791
7792
7793
7794
7795
7796
7797
7798
7799
7800
7801
7802
7803
7804
7805
7806
7807
7808
7809
7810
7811
7812
7813
7814
7815
7816
7817
7818
7819
7820
7821
7822
7823
7824
7825
7826
7827
7828
7829
7830
7831
7832
7833
7834
7835
7836
7837
7838
7839
7840
7841
7842
7843
7844
7845
7846
7847
7848
7849
7850
7851
7852
7853
7854
7855
7856
7857
7858
7859
7860
7861
7862
7863
7864
7865
7866
7867
7868
7869
7870
7871
7872
7873
7874
7875
7876
7877
7878
7879
7880
7881
7882
7883
7884
7885
7886
7887
7888
7889
7890
7891
7892
7893
7894
7895
7896
7897
7898
7899
7900
7901
7902
7903
7904
7905
7906
7907
7908
7909
7910
7911
7912
7913
7914
7915
7916
7917
7918
7919
7920
7921
7922
7923
7924
7925
7926
7927
7928
7929
7930
7931
7932
7933
7934
7935
7936
7937
7938
7939
7940
7941
7942
7943
7944
7945
7946
7947
7948
7949
7950
7951
7952
7953
7954
7955
7956
7957
7958
7959
7960
7961
7962
7963
7964
7965
7966
7967
7968
7969
7970
7971
7972
7973
7974
7975
7976
7977
7978
7979
7980
7981
7982
7983
7984
7985
7986
7987
7988
7989
7990
7991
7992
7993
7994
7995
7996
7997
7998
7999
8000

```

7517	023374	012737	177560	023412	MOV	#177560,2\$	:USE SEL4 FOR ADDRESS	
7518	023402	004537	003560		JSR	R5,NPRSET	:LOAD BA AND DATA	
7519	023406	000000			0		:IN DATA	
7520	023410	125252			125252		:OUT DATA	
7521	023412	000000		2\$:	0		:IN BA	
7522	023414	000000		1\$:	0		:OUT BA	
7523	023416	012761	000015	000004	MOV	#15,4(R1)		
7524	023424	012711	003000		MOV	#BIT9,BIT10,(R1)	:SET CROMI AND CROMO.!	
7525	023430	012761	121110	000006	MOV	#121110,6(R1)	:PUT INSTR INTO SEL6 NW*	
7526	023436	052711	000400		BIS	#BIT8,(R1)	:CLOCK IT!	
7527	023442	000240			NOP		:WAIT FOR NPR	
7528	023444				ROMCLK		:NEXT WORD IS INSTRUCTION,	BBN
(1)	023444	004537	003230		JSR	R5,.ROMCLK	:CLOCK INSTRUCTION	
7529	023450	021004			021004		:MOVE OUT DATA LB TO SEL4	
7530	023452				ROMCLK		:NEXT WORD IS INSTRUCTION,	BBN
(1)	023452	004537	003230		JSR	R5,.ROMCLK	:CLOCK INSTRUCTION	
7531	023456	021025			021025		:MOVE OUT DATA HB TO SEL5	
7532	023460	016104	000004		MOV	4(R1),R4	:PUT 'F.UND' IN R4	
7533	023464	013737	177560	002636	MOV	177560,\$GDDAT		
7534	023472	042737	000200	002636	BIC	#200,\$GDDAT		
7535	023500	023704	002636		COMP	\$GDDAT,R4	:CORRECT RESULTS?	
7536	023504	001413			BEQ	TOUTP	:BR IF YES	
7537	023506				ERROR	11,YES	:ERROR BA 16 AND 17 FAILED	
(5)	023520	104455			TRAP	C\$ERDF		
(6)	023522	000013			.WORD	11		
(6)	023524	004640			.WORD	EM11		
(6)	023526	006644			.WORD	ERR11		
7538	023530			3\$:				
7539	023530	062706	000004		TOUTT:	ADD	#4,SP	:UPDATE STACK POITNER
7540	023534	013737	002652	000006	TOUTP:	MOV	SAVE6,6	:RESTORE TRAP VECTOR
7541	023542	013737	002650	000004	MOV	SAVE4,4		
7542	023550				ENDTST			
(3)	023550				L10'22:			
(3)	023550	104401			TRAP	C\$ETST		
7543								
7544	023552				BADHEAD			
(2)							:***** TEST 41 *****	

```
7546 ;*NPR NON-EXISTENT MEMORY TEST
7547 ;*DO A DATO TO A NON-EXISTENT ADDRESS
7548 ;*VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11
7549 023552 BADHEAD
(2) ;***** TEST 41 *****
7550
7551 023552 BGNTST
(3) 023552 T41::
7552 023552 MYINT
(1) 023552 013701 002716 MOV KMCSR,R1 ;GET DEVICE ADDRESS.
7553 023556 MSTCLR ;MASTER CLEAR M8200,4,7
(1) 023556 004537 003142 JSR R5,MSTCLR ;CLEAR M8200,4,7
7554 023562 004537 003560 JSR R5,NPRSET ;LOAD IBUS REGISTERS 0-7
7555 023566 000000 0 ;IN DATA
7556 023570 000000 0 ;OUT DATA
7557 023572 177320 ;IN BA
7558 023574 177320 ;IN BA
7559 023576 012761 000014 000004 MOV #14,4(R1) ;SET OUT BA BITS 16+17 IN PORT4
```

7561	023604				ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	023604	004537	003230		JSR	R5,ROMCLK	:CLOCK INSTRUCTION
7562	023610	121111			121111		:SET OUTBA 16 AND 17
7563	023612	012761	000021	000004	MOV	#21,4(R1)	:SET NPR REQUEST BITS IN PORT4
7564	023620				ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	023620	004537	003230		JSR	R5,ROMCLK	:CLOCK INSTRUCTION
7565	023624	121110			121110		:MOV IBUS* 4 TO IBUS* 10
7566	023626	000240			NOP		
7567	023630				ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	023630	004537	003230		JSR	R5,ROMCLK	:CLOCK INSTRUCTION
7568	023634	121225			121225		:MOV IBUS*11 TO IBUS*5
7569	023636	012737	000001	002636	MOV	#1,\$GDDAT	:PUT 'EXPECTED' IN \$GDDAT
7570	023644	116104	000005		MOVB	5(R1),R4	:PUT 'FOUND' IN R4
7571	023650	042704	177776		BIC	#177776,R4	:CLEAR UNWANTED BITS
7572	023654	023704	002636		CMP	\$GDDAT,R4	:DATA CORRECT?
7573	023660	001411			BEQ	1\$	:BR IF YES
7574	023662				ERROR	13,YES	:ERROR NON-EXISTENT MEM BIT FAILED TO SET
(5)	023674	104455			TRAP	C\$ERDF	
(6)	023676	000015			.WORD	13	
(6)	023700	004673			.WORD	EM13	
(6)	023702	007000			.WORD	ERR13	
7575	023704				1\$:		
7576	023704	152761	000100	000001	BISB	#100,1(R1)	:SET MASTER CLEAR
7577	023712	142761	000100	000001	BICB	#100,1(R1)	:CLEAR MASTER
7578	023720				ROMCLK		:MOV IBUS*11 TO
(1)	023720	004537	003230		JSR	R5,ROMCLK	:CLOCK INSTRUCTION
7579	023724	121225			121225		:PORT5
7580	023726	005037	002636		CLR	\$GDDAT	:EXPECT CLEAR
7581	023732	116104	000005		MOVB	5(R1),R4	:GET NPR REG
7582	023736	042704	177776		BIC	#177776,R4	:CLEAR JUNK
7583	023742	001411			BEQ	2\$	:EXIT IF CLEAR
7584	023744				ERROR	13,YES	:NON-EXISTANT MEM
(5)	023756	104455			TRAP	C\$ERDF	
(6)	023760	000015			.WORD	13	
(6)	023762	004673			.WORD	EM13	
(6)	023764	007000			.WORD	ERR13	
7585							:BIT FAILED TO CLEAR
7586	023766				2\$:		
7587	023766				ENDTST		
(3)	023766				L10123:		
(3)	023766	104401			TRAP	C\$ETST	
7588							
7589	023770				BADHEAD		
(2)					:***** TEST 42 *****		
7590					:*NPR NON-EXISTENT MEMORY TEST		
7591					:*DO A DATI FROM A NON-EXISTENT ADDRESS		
7592					:*VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11		
7593	023770				BADHEAD		
(2)					:***** TEST 42 *****		
7594							
7595	023770				BGNTST		
(3)	023770				T42::		
7596	023770				MYINT		
(1)	023770	013701	002716		MOV	KMCSR,R1	:GET DEVICE ADDRESS.
7597	023774				MSTCLR		:MASTER CLEAR M8200,4,7
(1)	023774	004537	003142		JSR	R5,MSTCLR	:CLEAR M8200,4,7

7598	024000	004537	003560		JSR	R5,NPRSET	:LOAD IBUS REGISTERS 0-7
7599	024004	000000			0		:IN DATA
7600	024006	000000			0		:OUT DATA
7601	024010	177320			177320		:IN BA
7602	024012	177320			177320		:OUT BA
7603	024014	005061	000004		CLR	4(R1)	
7604	024020				ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	024020	004537	003230		JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
7605	024024	121111			121111		:CLEAR NON-EXISTENT BIT
7606	024026	012761	000015	000004	MOV	#15,4(R1)	:SET NPR REQUEST BITS IN PORT4
7607	024034				ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	024034	004537	003230		JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
7608	024040	121110			121110		:MOV IBUS* 4 TO IBUS* 10
7609	024042	000240			NOP		
7610	024044				ROMCLK		:NEXT WORD IS INSTRUCTION, BBN
(1)	024044	004537	003230		JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
7611	024050	121225			121225		:MOV IBUS*11 TO IBUS*5
7612	024052	012737	000001	002636	MOV	#1,\$GDDAT	:PUT 'EXPECTED' IN \$GDDAT
7613	024060	116104	000005		MOV/B	5(R1),R4	:PUT 'FOUND' IN R4
7614	024064	042704	177776		BIC	#177776,R4	:CLEAR UNWANTED BITS
7615	024070	023704	002636		CMP	\$GDDAT,R4	:DATA CORRECT?
7616	024074	001411			BEQ	1\$	:BR IF YES
7617	024076				ERROR	13,YES	:ERROR NON-EXISTENT MEM BIT FAILED TO SET
(5)	024110	104455			TRAP	C\$ERDF	
(6)	024112	000015			.WORD	13	
(6)	024114	004673			.WORD	EM13	
(6)	024116	007000			.WORD	ERR13	
7618	024120						
7619	024120						
(3)	024120						
(3)	024120	104401			TRAP	C\$ETST	
7620							
7621	024122				BADHEAD		
(2)					*****	TEST 43	*****
7622					:*NPR TEST		
7623					:*USING DATO, NPR A BINARY COUNT (0-377)		
7624					:*FROM MICRO-PROCESSOR TO ALL AVAILABLE MEMORY		
7625	024122				BADHEAD		
(2)					:*****	TEST 43	*****
7626							
7627	024122						
(3)	024122						
7628	024122				MYINT		
(1)	024122	013701	002716		MOV	KMCSR,R1	:GET DEVICE ADDRESS.
7629	024126				MSTCLR		:MASTER CLEAR M8200,4,7
(1)	024126	004537	003142		JSR	R5,.MSTCLR	:CLEAR M8200,4,7
7630	024132	005037	024334		CLR	5\$	:START FLAG AT 0
7631	024136	005005			CLR	R5	:DATA
7632	024140	012702	035450		MOV	#CORMAX,R2	:ADDRESS
7633	024144						
7634	024144	010537	024174		MOV	R5,2\$	:LOAD DATA
7635	024150	010237	024200		MOV	R2,4\$	:LOAD BA
7636	024154	032702	000001		BIT	#BIT0,R2	:IS BA ODD?
7637	024160	001402			BEQ	+6	:BR IF NO
7638	024162	000337	024174		SWAB	2\$	:IF ODD PUT DATA IN HI-BYTE
7639	024166	004537	003560		JSR	R5,NPRSET	:LOAD NPR REGISTERS

```

7640 024172 000000
7641 024174 000000      2$: 0      ;IN DATA
7642 024176 000000      0      ;OUT DATA
7643 024200 000000      4$: 0      ;IN BA
7644 024202 105012      0      ;OUT BA
7645 024204 012761 000221 000004 CLR B (R2) ;CLEAR MEMORY LOCATION
7646 024212 004537 003230 MOV #221,4(R1) ;LOAD PORT4
(1) 024212 121110 JSR R5,.ROMCLK ;NEXT WORD IS INSTRUCTION, BBN
7647 024216 000240 121110 ;CLOCK INSTRUCTION
7648 024220 010537 002636 NOP ;DO THE NPR
7649 024222 111204 MOV R5,$GDDAT ;PUT 'EXPECTED' IN $GDDAT
7650 024226 123704 002636 MOV B (R2),R4 ;PUT 'FOUND' IN R4
7651 024230 001411 CMPB $GDDA1,R4 ;IS DATA CORRECT?
7652 024234 104455 BEQ 3$ ;BR IF YES
7653 024236 000013 ERROR 11,YES ;ERROR, DATA INCORRECT
(5) 024250 004640 TRAP C$ERDF
(6) 024252 006644 .WORD 11
(6) 024254 006644 .WORD EM11
(6) 024256 006644 .WORD ERR11
7654 024260 104410 3$: ESCAPE TST
(3) 024260 000054 TRAP C$ESCAPE
(3) 024262 005205 .WORD L10125-.
7655 024264 042705 177400 INC R5 ;NEXT CHARACTER
7656 024266 005737 024334 BIC #177400,R5 ;USE ONLY LOW BYTE
7657 024272 001402 TST 5$ ;HAS MAX MEMORY BEEN REACHED YET?
7658 024276 005705 BEQ 6$ ;BR IF NO
7659 024300 001412 TST R5 ;DONE PATTERN?
7660 024302 005202 BEQ 7$ ;BR IF YES
7661 024304 023702 002604 6$: INC B2 ;INC BA
7662 024306 001314 CMP MEMLIM,R2 ;REACHED MEMORY LIMIT YET?
7663 024312 012702 035450 BNE 1$ ;BR IF NOT
7664 024314 012737 177777 024334 MOV #CORMAX,R2 ;RESTART BA AT FIRST ADDRESS
7665 024320 000706 MOV #-1,5$ ;SET FLAG TO END TEST AT END OF DATA PATTERN
7666 024326 000706 BR 1$ ;CONTINUE
7667 024330
7668 024330 EXIT TST
(3) 024330 104432 TRAP C$EXIT
(3) 024332 000004 .WORD L10125-.
7669 024334 000000 5$: 0 ;THIS LOCATION IS A FLAG, IT STARTS AT 0,
7670 ;AND IS SET TO -1 WHEN LAST MEMORY ADDRESS
7671 ;IS USED, TEST IS THEN ENDED WHEN PATTERN IS FINISHED
7672 024336
(3) 024336
(3) 024336 104401 ENDTST
7673 L10125: TRAP C$ETST
7674 ;$MEM1
7675 ;$MEM0
7676 ;$MEM2 1K
7677 ;$MEM3 1K
7678 024340
(2) BADHEAD
7679 ;***** TEST 44 *****
7680 ;*ALU C BIT TEST
7681 024340 ;*TEST THAT AN ADD OF 377 AND 377 WILL SET THE C BIT
(2) BADHEAD
7682 ;***** TEST 44 *****

```



```
7683 024340          BGNTST
(3) 024340          T44::
7684 024340          MYINT
(1) 024340 013701 002716  MOV KMCSR,R1      ;GET DEVICE ADDRESS.
7685 024344          MSTCLR      ;MASTER CLEAR M8200,4,7
(1) 024344 004537 003142  JSR R5,,MSTCLR      ;CLEAR M8200,4,7
7686 024350 004737 003624  JSR PC,MEMLD      ;LOAD MAINMEM DATA
7687 024354 024470          TDATA      ;POINTER TO DATA
7688 024356 004737 003776  JSR PC,SPLD      ;LOAD SP DATA
7689 024362 024470          TDATA      ;POINTER TO DATA
7690 024364          BGNSEG
(3) 024364 104404          TRAP C$BSEG
7691 024366          1$:
7692 024366          ROMCLK      ;NEXT WORD IS INSTRUCTION, BBN
(1) 024366 004537 003230  JSR R5,,ROMCLK      ;CLOCK INSTRUCTION
7693 024372 010000          010000      ;MAR 0
7694 024374          ROMCLK      ;NEXT WORD IS INSTRUCTION, BBN
(1) 024374 004537 003230  JSR R5,,ROMCLK      ;CLOCK INSTRUCTION
7695 024400 054400          054400!<0*20> ;ADD 377 AND 377, TO SET C BIT
7696 024402          ROMCLK      ;NEXT WORD IS INSTRUCTION, BBN
(1) 024402 004537 003230  JSR R5,,ROMCLK      ;CLOCK INSTRUCTION
7697 024406 040421          040401!<1*20> ;ADD 0 AND 0 AND THE C BIT
7698 024410          ROMCLK      ;NEXT WORD IS INSTRUCTION, BBN
(1) 024410 004537 003230  JSR R5,,ROMCLK      ;CLOCK INSTRUCTION
7699 024414 061224          61224      ;PUT RESULTS IN PORT4
7700 024416 012737 000001 002636  MOV #1,$GDDAT      ;PUT 'EXPECTED' IN $GDDAT
7701 024424 016104 000004          MOV 4(R1),R4      ;PUT 'FOUND' IN R4
7702 024430 123704 002636          CMPB $GDDAT,R4      ;DATA CORRECT?
7703 024434 001411          BEQ 2$      ;BR IF YES
7704 024436          ERROR 34,YES      ;ERROR C BIT NOT SET
(5) 024450 104455          TRAP C$ERDF
(6) 024452 000042          .WORD 34
(6) 024454 005421          .WORD EM34
(6) 024456 010262          .WORD ERR34
7705 024460          2$:
(3) 024460 104410          ESCAPE SEG
(3) 024462 000002          TRAP C$ESCAPE
7706 024464          .WORD 10000$-
(3) 024464          10000$:
(3) 024464 104405          TRAP C$ESEG
7707 024466          ENDTST
(3) 024466          L10126:
(3) 024466 104401          TRAP C$ETST
7708 024470 377 000 000 000  TDATA: .BYTE -1,0,0,0,0,0,0,0
      024473 000 000 000
      024476 000 000 000

7709
7710          .EVEN
7711
7712 024500          BADHEAD
(2)          ;***** TEST 45 *****
7713          ;*ALU TEST
7714          ;*TEST OF ALU FUNCTION SEL B WITH C BIT CLEARED
7715          ;*ALU FUNCTION (B) CODE=11
7716          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7717          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
```

```

7718 024500                                BADHEAD
(2)                                         ;***** TEST 45 *****
7719                                         ;
7720 024500                                BGNTST
(3) 024500                                T45::
7721 024500                                MYINT
(1) 024500 013701 002716                    MOV    KMCSR,R1          ;GET DEVICE ADDRESS.
7722 024504 004537 003142                    MSTCLR          ;MASTER CLEAR M8200,4,7
(1) 024504 004537 003142                    JSR     R5,.,MSTCLR      ;CLEAR M8200,4,7
7723 024510 005005                            CLR     R5              ;MEM + SP ADDRESS
7724 024512 012702 024672                    MOV    #5$,R2          ;POINTER TO CORRECT DATA
7725 024516 004737 003624                    JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
7726 024522 002654                            MEMDAT          ;POINTER TO DATA
7727 024524 004737 003776                    JSR     PC,SPLD        ;LOAD 8 WORDS OF SP
7728 024530 002664                            SPDAT          ;POINTER TO DATA
7729 024532                                BGNSEG
(3) 024532 104404                            TRAP    C$BSEG
7730 024534 004737 004044                    JSR     PC,CLRC        ;CLEAR C BIT!
7731 024540 042737 000017 024556 1$:        BIC    #17,2$      ;CLEAR ADDRESS FIELD OF INSTRUCTION
7732 024546 050537 024556                    BIS     R5,2$        ;ADD ADDRESS TO INSTRUCTION
7733 024552                                ROMCLK          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 024552 004537 003230                    JSR     R5,.,ROMCLK    ;CLOCK INSTRUCTION
7734 024556 010000                                010000          ;LOAD MAR
7735 024560 042737 000017 024576 2$:        BIC    #17,3$      ;CLEAR ADDRESS OF INSTRUCTION
7736 024566 050537 024576                    BIS     R5,3$        ;ADD ADDRESS TO INSTRUCTION
7737 024572                                ROMCLK          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 024572 004537 003230                    JSR     R5,.,ROMCLK    ;CLOCK INSTRUCTION
7738 024576 040620 040400!<11*20> 3$:        ROMCLK          ;BR SEL B
7739 024600                                JSR     R5,.,ROMCLK    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 024600 004537 003230                                ;CLOCK INSTRUCTION
7740 024604 061224                                61224          ;MOVE BR TO PORT4
7741 024606 111237 002636                    MOVB    (R2), $GDDAT    ;PUT 'EXPECTED' IN $GDDAT
7742 024612 116104 000004                    MOVB    4(R1), R4      ;PUT 'FOUND' IN R4
7743 024616 123704 002636                    CMPB    $GDDAT,R4     ;DATA CORRECT?
7744 024622 001411                                BEQ     4$           ;BR IF YES
7745 024624                                ERROR    15,YES       ;ALU ERROR
(5) 024636 104455                            TRAP    C$ERDF
(6) 024640 000017                            .WORD   15
(6) 024642 004740                            .WORD   EM15
(6) 024644 007104                            .WORD   ERR15
7746 024646                                4$:    ESCAPE    SEG
(3) 024646 104410                            TRAP    C$ESCAPE
(3) 024650 000014                            .WORD   10000$-
7747 024652 005202                            INC     R2            ;NEXT DATA
7748 024654 005205                            INC     R5            ;NEXT ADDRESS
7749 024656 022705 000010                    CMP     #10,R5        ;DONE YET?
7750 024662 001324                            BNE     1$           ;BR IF NO
7751 024664                                ENDSEG
(3) 024664                                10000$:
(3) 024664 104405                            TRAP    C$ESEG
7752 024666                                EXIT    TST
(3) 024666 104432                            TRAP    C$EXIT
(3) 024670 000012                            .WORD   L10127-
7753 024672 000 377 000 5$:                .BYTE   0,-1,0,-1,125,252,125,252
      024675 377 125 252
      024700 125 252
  
```

```

7754
7755
7756 024702      .EVEN
(3) 024702      ENDTST
(3) 024702 104401 L10127:
7757                      TRAP      C$ETST
7758 024704      BADHEAD
(2)                      :***** TEST 46 *****
7759                      :*ALU TEST
7760                      :*TEST OF ALU FUNCTION SEL A WITH C BIT CLEARED
7761                      :*ALU FUNCTION (A)      CODE=10
7762                      :*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7763                      :*PERFORM THE FUNCTION, VERIFY THE RESULTS
7764 024704      BADHEAD
(2)                      :***** TEST 46 *****
7765
7766 024704      BGNTST
(3) 024704      T46::
7767 024704      MYINT
(1) 024704 013701 002716  MOV      KMCSR,R1      ;GET DEVICE ADDRESS.
7768 024710      MSTCLR      ;MASTER CLEAR M8200,4,7
(1) 024710 004537 003142  JSR      R5,.,MSTCLR      ;CLEAR M8200,4,7
7769 024714 005005      CLR      R5      ;MEM + SP ADDRESS
7770 024716 012702 025076  MOV      #5$,R2      ;POINTER TO CORRECT DATA
7771 024722 004737 003624  JSR      PC,MEMLD      ;LOAD 8 WORDS OF MAIN MEMORY
7772 024726 002654      MEMDAT      ;POINTER TO DATA
7773 024730 004737 003776  JSR      PC,SPLD      ;LOAD 8 WORDS OF SP
7774 024734 002664      SPDAT      ;POINTER TO DATA
7775 024736
(3) 024736 104404      TRAP      C$BSEG
7776 024740 004737 004044  JSR      PC,CLRC      ;CLEAR C BIT!
7777 024744 042737 000017 024762  BIC      #17,2$      ;CLEAR ADDRESS FIELD OF INSTRUCTION
7778 024752 050537 024762  BIS      R5,2$      ;ADD ADDRESS TO INSTRUCTION
7779 024756      ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
(1) 024756 004537 003230  JSR      R5,.,ROMCLK      ;CLOCK INSTRUCTION
7780 024762 010000      010000      ;LOAD MAR
7781 024764 042737 000017 025002  BIC      #17,3$      ;CLEAR ADDRESS OF INSTRUCTION
7782 024772 050537 025002  BIS      R5,3$      ;ADD ADDRESS TO INSTRUCTION
7783 024776      ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
(1) 024776 004537 003230  JSR      R5,.,ROMCLK      ;CLOCK INSTRUCTION
7784 025002 040600      040400!<10*20>  ;BR      SEL A
7785 025004      ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
(1) 025004 004537 003230  JSR      R5,.,ROMCLK      ;CLOCK INSTRUCTION
7786 025010 061224      61224      ;MOVE BR TO PORT4
7787 025012 111237 002636  MOVB      (R2), $GDDAT      ;PUT 'EXPECTED' IN $GDDAT
7788 025016 116104 000004  MOVB      4(R1), R4      ;PUT 'FOUND' IN R4
7789 025022 123704 002636  CMPB      $GDDAT, R4      ;DATA CORRECT?
7790 025026 001411      BEQ      4$      ;BR IF YES
7791 025030      ERROR      15, YES      ;ALU ERROR
(5) 025042 104455      TRAP      C$ERDF
(6) 025044 000017      .WORD      15
(6) 025046 004740      .WORD      EM15
(6) 025050 007104      .WORD      ERR15
7792 025052      ' $:      ESCAPE      SEG
(3) 025052 104410      TRAP      C$ESCAPE
(3) 025054 000014      .WORD      10000$-.
```

7793	025056	005202				INC	R2	:NEXT DATA
7794	025060	005205				INC	R5	:NEXT DATA
7795	025062	022705	000010			CMP	#10,R5	:DONE YET?
7796	025066	001324				BNE	1\$	:BR IF NO
7797	025070					ENDSEG		
(3)	025070			10000\$:				
(3)	025070	104405				TRAP	C\$ESEG	
7798	025072					EXIT	TST	
(3)	025072	104432				TRAP	C\$EXIT	
(3)	025074	000012				.WORD	L10130-	
7799	025076	000	000	377	5\$:	.BYTE	0,0,-1,-1,125,125,252,252	
	025101	377	125	125				
	025104	252	252					
7800								
7801								
7802	025106					.EVEN		
(3)	025106					ENDTST		
(3)	025106	104401				L10130:		
7803						TRAP	C\$ETST	
7804	025110					BADHEAD		
(2)						:***** TEST 47 *****		
7805						:*ALU TEST		
7806						:*TEST OF ALU FUNCTION A OR NOTB WITH C BIT CLEARED		
7807						:*ALU FUNCTION (A OR NOTB) CODE=12		
7808						:*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA		
7809						:*PERFORM THE FUNCTION, VERIFY THE RESULTS		
7810	025110					BADHEAD		
(2)						:***** TEST 47 *****		
7811								
7812	025110					BGNTST		
(3)	025110					T47::		
7813	025110					MYINT		
(1)	025110	013701	002716			MOV	KMCSR,R1	:GET DEVICE ADDRESS.
7814	025114					MSTCLR		:MASTER CLEAR M8200,4,7
(1)	025114	004537	003142			JSR	R5,.MSTCLR	:CLEAR M8200,4,7
7815	025120	005005				CLR	R5	:MEM + SP ADDRESS
7816	025122	012702	025302			MOV	#5\$,R2	:POINTER TO CORRECT DATA
7817	025126	004737	003624			JSR	PC,MEMLD	:LOAD 8 WORDS OF MAIN MEMORY
7818	025132	002654				MEMDAT		:POINTER TO DATA
7819	025134	004737	003776			JSR	PC,SPLD	:LOAD 8 WORDS OF SP
7820	025140	002664				SPDAT		:POINTER TO DATA
7821	025142					BGNSEG		
(3)	025142	104404				TRAP	C\$BSEG	
7822	025144	004737	004044			JSR	PC,CLRC	:CLEAR C BIT!
7823	025150	042737	000017	025166	1\$:	BIC	#17,2\$	:CLEAR ADDRESS FIELD OF INSTRUCTION
7824	025156	050537	025166			BIS	R5,2\$	:ADD ADDRESS TO INSTRUCTION
7825	025162					ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	025162	004537	003230			JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
7826	025166	010000				010000		:LOAD MAR
7827	025170	042737	000017	025206	2\$:	BIC	#17,3\$	:CLEAR ADDRESS OF INSTRUCTION
7828	025176	050537	025206			BIS	R5,3\$	:ADD ADDRESS TO INSTRUCTION
7829	025202					ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	025202	004537	003230			JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
7830	025206	040640				040400!<12*20>		:BR A OR NOTB
7831	025210					ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	025210	004537	003230			JSR	R5,.ROMCLK	:CLOCK INSTRUCTION

7832	025214	061224			61224				
7833	025216	111237	002636		MOVB	(R2), \$GDDAT			;MOVE BR TO PORT4
7834	025222	116104	000004		MOVB	4(R1), R4			;PUT 'EXPECTED' IN \$GDDAT
7835	025226	123704	002636		CMPB	\$GDDAT, R4			;PUT 'FOUND' IN R4
7836	025232	001411			BEQ	4\$			;DATA CORRECT?
7837	025234				ERROR	15, YES			;BR IF YES
(5)	025246	104455			TRAP	C\$ERDF			;ALU ERROR
(6)	025250	000017			.WORD	15			
(6)	025252	004740			.WORD	EM15			
(6)	025254	007104			.WORD	ERR15			
7838	025256				4\$: ESCAPE	SEG			
(3)	025256	104410			TRAP	C\$ESCAPE			
(3)	025260	000014			.WORD	10000\$-			
7839	025262	005202			INC	R2			;NEXT DATA
7840	025264	005205			INC	R5			;NEXT DATA
7841	025266	022705	000010		CMP	#10, R5			;DONE YET?
7842	025272	001324			BNE	1\$			;BR IF NO
7843	025274				ENDSEG				
(3)	025274				10000\$: TRAP	C\$ESEG			
(3)	025274	104405			EXIT	TST			
7844	025276				TRAP	C\$EXIT			
(3)	025276	104432			.WORD	L10131-			
(3)	025300	000012			5\$: .BYTE	-1, 0, -1, -1, -1, 125, 252, -1			
7845	025302	377	000	377					
	025305	377	377	125					
	025310	252	377						
7846									
7847									
7848	025312				.EVEN				
(3)	025312				ENDTST				
(3)	025312	104401			L10131: TRAP	C\$ETST			
7849									
7850	025314				BADHEAD				
(2)					:***** TEST 48 *****				
7851					:*ALU TEST				
7852					:*TEST OF ALU FUNCTION A AND B WITH C BIT CLEARED				
7853					:*ALU FUNCTION (A AND B) CODE=13				
7854					:*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA				
7855					:*PERFORM THE FUNCTION, VERIFY THE RESULTS				
7856	025314				BADHEAD				
(2)					:***** TEST 48 *****				
7857									
7858	025314				BGNTST				
(3)	025314				T48::				
7859	025314				MYINT				
(1)	025314	013701	002716		MOV	KMCSR, R1			;GET DEVICE ADDRESS.
7860	025320				MSTCLR				;MASIER CLEAR M8200, 4, 7
(1)	025320	004537	003142		JSR	R5, .MSTCLR			;CLEAR M8200, 4, 7
7861	025324	005005			CLR	R5			;MEM + SP ADDRESS
7862	025326	012702	025506		MOV	#5\$, R2			;POINTER TO CORRECT DATA
7863	025332	004737	003624		JSR	PC, MEMLD			;LOAD 8 WORDS OF MAIN MEMORY
7864	025336	002654			MEMDAT				;POINTER TO DATA
7865	025340	004737	003776		JSR	PC, SPLD			;LOAD 8 WORDS OF SP
7866	025344	002664			SPDAT				;POINTER TO DATA
7867	025346				BGNSEG				
(3)	025346	104404			TRAP	C\$BSEG			

```

7868 025350 004737 004044 1$: JSR PC,CLRC ;CLEAR C BIT!
7869 025354 042737 000017 025372 BIC #17,2$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
7870 025362 050537 025372 BIS R5,2$ ;ADD ADDRESS TO INSTRUCTION
7871 025366 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 025366 004537 003230 JSR R5,ROMCLK ;CLOCK INSTRUCTION
7872 025372 010000 2$: 010000 ;LOAD MAR
7873 025374 042737 000017 025412 BIC #17,3$ ;CLEAR ADDRESS OF INSTRUCTION
7874 025402 050537 025412 BIS R5,3$ ;ADD ADDRESS TO INSTRUCTION
7875 025406 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 025406 004537 003230 JSR R5,ROMCLK ;CLOCK INSTRUCTION
7876 025412 040660 3$: 040400,<13*20> ;BR A AND B
7877 025414 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 025414 004537 003230 JSR R5,ROMCLK ;CLOCK INSTRUCTION
7878 025420 061224 61224 ;MOVE BR TO PORT4
7879 025422 111237 002636 MOVB (R2),SGDDAT ;PUT 'EXPECTED' IN SGDDAT
7880 025426 116104 000004 MOVB 4(R1),R4 ;PUT 'FOUND' IN R4
7881 025432 123704 002636 CMPB SGDDAT,R4 ;DATA CORRECT?
7882 025436 001411 BEQ 4$ ;BR IF YES
7883 025440 ERROR 15,YES ;ALU ERROR
(5) 025452 104455 TRAP C$ERDF
(6) 025454 000017 .WORD 15
(6) 025456 004740 .WORD EM15
(6) 025460 007104 .WORD ERR15
7884 025462 4$: ESCAPE SEG
(3) 025462 104410 TRAP C$ESCAPE
(3) 025464 000014 .WORD 10000$-
7885 025466 005202 INC R2 ;NEXT DATA
7886 025470 005205 INC R5 ;NEXT DATA
7887 025472 022705 000010 CMP #10,R5 ;DONE YET?
7888 025476 001324 BNE 1$ ;BR IF NO
7889 025500 ENDSEG
(3) 025500 10000$: TRAP C$ESEG
7890 025502 EXIT TST
(3) 025502 104432 TRAP C$EXIT
(3) 025504 000012 .WORD L10132-
7891 025506 000 000 5$: .BYTE 0,0,0,-1,125,0,0,252
025511 377 125
025514 000 252

7892
7893 .EVEN
7894 025516 ENDTST
(3) 025516 L10132:
(3) 025516 104401 TRAP C$ETST
7895
7896 025520 BADHEAD
(2) ;***** TEST 49 *****
7897 ;*ALU TEST
7898 ;*TEST OF ALU FUNCTION A OR B WITH C BIT CLEARED
7899 ;*ALU FUNCTION (A OR B) CODE=14
7900 ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7901 ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
7902 025520 BADHEAD
(2) ;***** TEST 49 *****
7903
7904 025520 BGNTEST
  
```

```
(3) 025520
7905 025520
(1) 025520 013701 002716
7906 025524
(1) 025524 004537 003142
7907 025530 005005
7908 025532 012702 025712
7909 025536 004737 003624
7910 025542 002654
7911 025544 004737 003776
7912 025550 002664
7913 025552
(3) 025552 104404
7914 025554 004737 004044
7915 025560 042737 000017 025576
7916 025566 050537 025576
7917 025572
(1) 025572 004537 003230
7918 025576 010000
7919 025600 042737 000017 025616
7920 025606 050537 025616
7921 025612
(1) 025612 004537 003230
7922 025616 040700
7923 025620
(1) 025620 004537 003230
7924 025624 061224
7925 025626 111237 002636
7926 025632 116104 000004
7927 025636 123704 002636
7928 025642 001411
7929 025644
(5) 025656 104455
(6) 025660 000017
(6) 025662 004740
(6) 025664 007104
7930 025666
(3) 025666 104410
(3) 025670 000014
7931 025672 005202
7932 025674 005205
7933 025676 022705 000010
7934 025702 001324
7935 025704
(3) 025704
(3) 025704 104405
7936 025706
(3) 025706 104432
(3) 025710 000012
7937 025712 000 377 377
025715 377 125 377
025720 377 252
7938
7939
7940 025722
(3) 025722
```

T49::

```
MYINT
MOV KMCSR,R1 ;GET DEVICE ADDRESS.
MSTCLR ;MASTER CLEAR M8200,4,7
JSR R5,.,MSTCLR ;CLEAR M8200,4,7
CLR R5 ;MEM + SP ADDRESS
MOV #5$,R2 ;POINTER TO CORRECT DATA
JSR PC,MEMLD ;LOAD 8 WORDS OF MAIN MEMORY
MEMDAT ;POINTER TO DATA
JSR PC,SPLD ;LOAD 8 WORDS OF SP
SPDAT ;POINTER TO DATA
BGNSEG
TRAP C$BSEG
JSR PC,CLRC ;CLEAR C BIT!
BIC #17,2$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
BIS R5,2$ ;ADD ADDRESS TO INSTRUCTION
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
JSR R5,.,ROMCLK ;CLOCK INSTRUCTION
010000 ;LOAD MAR
BIC #17,3$ ;CLEAR ADDRESS OF INSTRUCTION
BIS R5,3$ ;ADD ADDRESS TO INSTRUCTION
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
JSR R5,.,ROMCLK ;CLOCK INSTRUCTION
040400!<14*20> ;BR A OR B
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
JSR R5,.,ROMCLK ;CLOCK INSTRUCTION
61224 ;MOVE BR TO PORT4
MOVB (R2), $GDDAT ;PUT 'EXPECTED' IN $GDDAT
MOVB 4(R1), R4 ;PUT 'FOUND' IN R4
CMPB $GDDAT, R4 ;DATA CORRECT?
BEQ 4$ ;BR IF YES
ERROR 15, YES ;ALU ERROR
TRAP C$ERDF
.WORD 15
.WORD EM15
.WORD ERR15
4$: ESCAPE SEG
TRAP C$ESCAPE
.WORD 10000$-.
INC R2 ;NEXT DATA
INC R5 ;NEXT DATA
CMP #10, R5 ;DONE YET?
BNE 1$ ;BR IF NO
ENDSEG
10000$:
TRAP C$ESEG
EXIT TST
TRAP C$EXIT
.WORD L10133-.
5$: .BYTE 0,-1,-1,-1,125,-1,-1,252
```

.EVEN  
ENDTST  
L10133:

(3)	025722	104401			TRAP	C\$ETST	
7941							
7942	025724				BADHEAD		
(2)					:***** TEST 50 *****		
7943					:*ALU TEST		
7944					:*TEST OF ALU FUNCTION A XOR B WITH C BIT CLEARED		
7945					:*ALU FUNCTION (A XOR B) CODE=15		
7946					:*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA		
7947					:*PERFORM THE FUNCTION, VERIFY THE RESULTS		
7948	025724				BADHEAD		
(2)					:***** TEST 50 *****		
7949							
7950	025724				BGNTST		
(3)	025724				T50::		
7951	025724				MYINT		
(1)	025724	013701	002716		MOV	KMCSR,R1	:GET DEVICE ADDRESS.
7952	025730				MSTCLR		:MASTER CLEAR M8200,4,7
(1)	025730	004537	003142		JSR	R5,,MSTCLR	:CLEAR M8200,4,7
7953	025734	005005			CLR	R5	:MEM + SP ADDRESS
7954	025736	012702	026116		MOV	#5\$,R2	:POINTER TO CORRECT DATA
7955	025742	004737	003624		JSR	PC,MEMLD	:LOAD 8 WORDS OF MAIN MEMORY
7956	025746	002654			MEMDAT		:POINTER TO DATA
7957	025750	004737	003776		JSR	PC,SPLD	:LOAD 8 WORDS OF SP
7958	025754	002664			SPDAT		:POINTER TO DATA
7959	025756				BGNSEG		
(3)	025756	104404			TRAP	C\$BSEG	
7960	025760	004737	004044		JSR	PC,CLRC	:CLEAR C BIT!
7961	025764	042737	000017	026002	BIC	#17,2\$	:CLEAR ADDRESS FIELD OF INSTRUCTION
7962	025772	050537	026002		BIS	R5,2\$	:ADD ADDRESS TO INSTRUCTION
7963	025776				ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
(1)	025776	004537	003230		JSR	R5,,ROMCLK	:CLOCK INSTRUCTION
7964	026002	010000			010000		:LOAD MAR
7965	026004	042737	000017	026022	BIC	#17,3\$	:CLEAR ADDRESS OF INSTRUCTION
7966	026012	050537	026022		BIS	R5,3\$	:ADD ADDRESS TO INSTRUCTION
7967	026016				ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
(1)	026016	004537	003230		JSR	R5,,ROMCLK	:CLOCK INSTRUCTION
7968	026022	040720			040400!<15*20>		:BR A XOR B
7969	026024				ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	026024	004537	003230		JSR	R5,,ROMCLK	:CLOCK INSTRUCTION
7970	026030	061224			61224		:MOVE BR TO PORT4
7971	026032	111237	002636		MOVB	(R2), \$GDDAT	:PUT 'EXPECTED' IN \$GDDAT
7972	026036	116104	000004		MOVB	4(R1), R4	:PUT 'FOUND' IN R4
7973	026042	123704	002636		CMPS	\$GDDAT, R4	:DATA CORRECT?
7974	026046	001411			BEQ	4\$	:BR IF YES
7975	026050				ERROR	15, YES	:ALU ERROR
(5)	026062	104455			TRAP	C\$ERDF	
(6)	026064	000017			.WORD	15	
(6)	026066	004740			.WORD	EM15	
(6)	026070	007104			.WORD	ERR15	
7976	026072				ESCAPE	SEG	
(3)	026072	104410			TRAP	C\$ESCAPE	
(3)	026074	000014			.WORD	10000\$-	
7977	026076	005202			INC	R2	:NEXT DATA
7978	026100	005205			INC	R5	:NEXT DATA
7979	026102	022705	000010		CMP	#10, R5	:DONE YET?
7980	026106	001324			BNE	1\$	:BR IF NO



```

7981 026110          ENDSEG
(3) 026110          10000$:
(3) 026110 104405    TRAP  C$ESEG
7982 026112          EXIT  TST
(3) 026112 104432    TRAP  C$EXIT
(3) 026114 000012    .WORD L10134-
7983 026116          .BYTE 0,-1,-1,0,0,-1,-1,0
      026121          000 377 377 5$:
      026124          377 000 377

7984
7985
7986 026126          .EVEN
(3) 026126          ENDIST
(3) 026126 104401    L10134:
7987          TRAP  C$ETST
7988 026130          BADHEAD
(2)          :***** TEST 51 *****
7989          :*ALU TEST
7990          :*TEST OF ALU FUNCTION ADD WITH C BIT CLEARED
7991          :*ALU FUNCTION (A PLUS B) CODE=00
7992          :*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
7993          :*PERFORM THE FUNCTION, VERIFY THE RESULTS
7994 026130          BADHEAD
(2)          :***** TEST 51 *****
7995
7996 026130          BGNTST
(3) 026130          T51::
7997 026130          MYINT
(1) 026130 013701 002716  MOV  KMCSR,R1      ;GET DEVICE ADDRESS.
7998 026134          MSTCLR      ;MASTER CLEAR M8200,4,7
(1) 026134 004537 003142  JSR   R5,.,MSTCLR  ;CLEAR M8200,4,7
7999 026140 005005          CLR   R5        ;MEM + SP ADDRESS
8000 026142 012702 026322  MOV  #5$,R2     ;POINTER TO CORRECT DATA
8001 026146 004737 003624  JSR   PC,MEMLD   ;LOAD 8 WORDS OF MAIN MEMORY
8002 026152 002654          MEMDAT      ;POINTER TO DATA
8003 026154 004737 003776  JSR   PC,SPLD    ;LOAD 8 WORDS OF SP
8004 026160 002664          SPDAT       ;POINTER TO DATA
8005 026162          BGNSEG
(3) 026162 104404          TRAP  C$BSEG
8006 026164 004737 004044  JSR   PC,CLRC    ;CLEAR C BIT!
8007 026170 042737 000017  BIC   #17,2$    ;CLEAR ADDRESS FIELD OF INSTRUCTION
8008 026176 050537 026206  BIS   R5,2$     ;ADD ADDRESS TO INSTRUCTION
8009 026202          ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 026202 004537 003230  JSR   R5,.,ROMCLK ;CLOCK INSTRUCTION
8010 026206 010000          010000      ;LOAD MAR
8011 026210 042737 000017  BIC   #17,3$    ;CLEAR ADDRESS OF INSTRUCTION
8012 026216 050537 026226  BIS   R5,3$     ;ADD ADDRESS TO INSTRUCTION
8013 026222          ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 026222 004537 003230  JSR   R5,.,ROMCLK ;CLOCK INSTRUCTION
8014 026226 040400 040400!<00*20>      ;BR ADD
8015 026230          ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 026230 004537 003230  JSR   R5,.,ROMCLK ;CLOCK INSTRUCTION
8016 026234 061224          61224        ;MOVE BR TO PORT4
8017 026236 111237 002636  MOVB  (R2), $GDDAT ;PUT 'EXPECTED' IN $GDDAT
8018 026242 116104 000004  MOVB  4(R1), R4  ;PUT 'FOUND' IN R4
8019 026246 123704 002636  CMPB  $GDDAT, R4 ;DATA CORRECT?
  
```

PC	Instruction	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418
----	-------------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------



(1)	026544	004537	003142		JSR	R5, .MSTCLR	; CLEAR M8200, 4, 7
8091	026550	005005			CLR	R5	; MEM + SP ADDRESS
8092	026552	012702	026734		MOV	#5\$, R2	; POINTER TO CORRECT DATA
8093	026556	004737	003624		JSR	PC, MEMLD	; LOAD 8 WORDS OF MAIN MEMORY
8094	026562	002654			MEMDAT		; POINTER TO DATA
8095	026564	004737	003776		JSR	PC, SPLD	; LOAD 8 WORDS OF SP
8096	026570	002664			SPDAT		; POINTER TO DATA
8097	026572				BGNSEG		
(3)	026572	104404			TRAP	C\$BSEG	
8098	026574	004737	004044		JSR	PC, CLRC	; CLEAR C BIT!
8099	026600	042737	000017	026616	BIC	#17, 2\$	; CLEAR ADDRESS FIELD OF INSTRUCTION
8100	026606	050537	026616		BIS	R5, 2\$	; ADD ADDRESS TO INSTRUCTION
8101	026612				ROMCLK		; NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	026612	004537	003230		JSR	R5, .ROMCLK	; CLOCK INSTRUCTION
8102	026616	010000				010000	; LOAD MAR
8103	026620	042737	000017	026636	BIC	#17, 3\$	; CLEAR ADDRESS OF INSTRUCTION
8104	026626	050537	026636		BIS	R5, 3\$	; ADD ADDRESS TO INSTRUCTION
8105	026632				ROMCLK		; NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	026632	004537	003230		JSR	R5, .ROMCLK	; CLOCK INSTRUCTION
8106	026636	040740				040400! <16*20>	; BR - SUB
8107	026640				ROMCLK		
(1)	026640	004537	003230		JSR	R5, .ROMCLK	; CLOCK INSTRUCTION
8108	026644	061224				61224	; MOVE BR TO PORT4
8109	026646	111237	002636		MOVB	(R2), \$GDDAT	; PUT 'EXPECTED' IN \$GDDAT
8110	026652	116104	000004		MOVB	4(R1), R4	; PUT 'FOUND' IN R4
8111	026656	123737	002636	002636	CMPS	\$GDDAT, \$GDDAT	; DATA CORRECT?
8112	026664	001411			BEQ	4\$	; BR IF YES
8113	026666				ERROR	15, YES	; ALU ERROR
(5)	026700	104455			TRAP	C\$ERDF	
(6)	026702	000017			.WORD	15	
(6)	026704	004740			.WORD	EM15	
(6)	026706	007104			.WORD	ERR15	
8114	026710				ESCAPE	SEG	
(3)	026710	104410			TRAP	C\$ESCAPE	
(3)	026712	000014			.WORD	10000\$-	
8115	026714	005202			INC	R2	; NEXT DATA
8116	026716	005205			INC	R5	; NEXT ADDRESS
8117	026720	022705	000010		CMP	#10, R5	; DONE YET?
8118	026724	001323			BNE	1\$	; BR IF NO
8119	026726				ENDSEG		
(3)	026726					10000\$:	
(3)	026726	104405			TRAP	C\$ESEG	
8120	026730				EXIT	TST	
(3)	026730	104432			TRAP	C\$EXIT	
(3)	026732	000012			.WORD	L10137-	
8121	026734	000	001	377	.BYTE	0, 1, -1, 0, 0, 253, 125, 0	
	026737	000	000	253			
	026742	125	000				
8122							
8123							
8124							
8125	026744				.EVEN		
(3)	026744				ENDTST		
(3)	026744	104401			L10137:		
8126					TRAP	C\$ETST	
8127							

8128	026746				BADHEAD	
(2)					:***** TEST 54 *****	
8129					:*ALU TEST	
8130					:*TEST OF ALU FUNCTION ADD W/C WITH C BIT CLEARED	
8131					:*ALU FUNCTION (A PLUS B PLUS C) CODE 01	
8132					:*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA	
8133					:*PERFORM THE FUNCTION, VERIFY THE RESULTS	
8134	026746				BADHEAD	
(2)					:***** TEST 54 *****	
8135						
8136	026746				BGNTST	
(3)	026746				T54::	
8137	026746				MYINT	
(1)	026746	013701	002716		MOV KMCSR,R1	:GET DEVICE ADDRESS.
8138	026752				MSTCLR	:MASTER CLEAR M8200,4,7
(1)	026752	004537	003142		JSR R5,.MSTCLR	:CLEAR M8200,4,7
8139	026756	005005			CLR R5	:MEM + SP ADDRESS
8140	026760	012702	027140		MOV #5\$,R2	:POINTER TO CORRECT DATA
8141	026764	004737	003624		JSR PC,MEMLD	:LOAD 8 WORDS OF MAIN MEMORY
8142	026770	002654			MEMDAT	:POINTER TO DATA
8143	026772	004737	003776		JSR PC,SPLD	:LOAD 8 WORDS OF SP
8144	026776	002664			SPDAT	:POINTER TO DATA
8145	027000				BGNSEG	
(3)	027000	104404			TRAP C\$BSEG	
8146	027002	004737	004044		JSR PC,CLRC	:CLEAR C BIT!
8147	027006	042737	000017	027024	BIC #17,2\$	:CLEAR ADDRESS FIELD OF INSTRUCTION
8148	027014	050537	027024		BIS R5,2\$	:ADD ADDRESS TO INSTRUCTION
8149	027020				ROMCLK	:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	027020	004537	003230		JSR R5,.ROMCLK	:CLOCK INSTRUCTION
8150	027024	010000			010000	:LOAD MAR
8151	027026	042737	000017	027044	BIC #17,3\$	:CLEAR ADDRESS OF INSTRUCTION
8152	027034	050537	027044		BIS R5,3\$	:ADD ADDRESS TO INSTRUCTION
8153	027040				ROMCLK	:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	027040	004537	003230		JSR R5,.ROMCLK	:CLOCK INSTRUCTION
8154	027044	040420			040400!<01*20>	:BR ADD W/C
8155	027046				ROMCLK	:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	027046	004537	003230		JSR R5,.ROMCLK	:CLOCK INSTRUCTION
8156	027052	061224			61224	:MOVE BR TO PORT4
8157	027054	111237	002636		MOVB (R2), \$GDDAT	:PUT 'EXPECTED' IN \$GDDAT
8158	027060	116104	000004		MOVB 4(R1), R4	:PUT 'FOUND' IN R4
8159	027064	123704	002636		CMPB \$GDDAT, R4	:DATA CORRECT?
8160	027070	001411			BEQ 4\$	:BR IS YES
8161	027072				ERROR 15, YES	:ALU ERROR
(5)	027104	104455			TRAP C\$ERDF	
(6)	027106	000017			.WORD 15	
(6)	027110	004740			.WORD EM15	
(6)	027112	007104			.WORD ERR15	
8162	027114				ESCAPE SEG	
(3)	027114	104410			TRAP C\$ESCAPE	
(3)	027116	000014			.WORD 10000\$-	
8163	027120	005202			INC R2	:NEXT DATA
8164	027122	005205			INC R5	:NEXT ADDRESS
8165	027124	022705	000010		CMP #10, R5	:DONE YET?
8166	027130	001324			BNE 1\$	:BR IF NO
8167	027132				ENDSEG	
(3)	027132				10000\$:	

(3) 027132 104405 TRAP C\$ESEG  
8168 027134 EXIT TST  
(3) 027134 104432 TRAP C\$EXIT  
(3) 027136 000012 .WORD L10140-  
8169 027140 000 377 377 5\$: .BYTE 0,-1,-1,376,252,-1,-1,124  
027143 376 252 377  
027146 377 124

8170  
8171 .EVEN  
8172 027150 ENDTST  
(3) 027150 L10140:  
(3) 027150 104401 TRAP C\$ETST  
8173  
8174

8175 027152 BADHEAD  
(2) :\*\*\*\*\* TEST 55 \*\*\*\*\*  
8176 :\*ALU TEST  
8177 :\*TEST OF ALU FUNCTION SUB W/C WITH C BIT CLEARED  
8178 :\*ALU FUNCTION (A-B-C) CODE=2  
8179 :\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
8180 :\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
8181 027152 BADHEAD  
(2) :\*\*\*\*\* TEST 55 \*\*\*\*\*

8182  
8183 027152 BGNTST  
(3) 027152 T55::  
8184 027152 MYINT  
(1) 027152 013701 002716 MOV KMCSR,R1 ;GET DEVICE ADDRESS.

8186	027156					MSTCLR		;MASTER CLEAR M8200,4,7
(1)	027156	004537	003142			JSR	R5, MSTCLR	;CLEAR M8200,4,7
8187	027162	005005				CLR	R5	;MEM + SP ADDRESS
8188	027164	012702	027344			MOV	#5\$, R2	;POINTER TO CORRECT DATA
8189	027170	004737	003624			JSR	PC, MEMLD	;LOAD 8 WORDS OF MAIN MEMORY
8190	027174	002654				MEMDAT		;POINTER TO DATA
8191	027176	004737	003776			JSR	PC, SPLD	;LOAD 8 WORDS OF SP
8192	027202	002664				SPDAT		;POINTER TO DATA
8193	027204					BGNSEG		
(3)	027204	104404				TRAP	C\$BSEG	
8194	027206	004737	004044	027230	1\$:	JSR	PC, CLRC	;CLEAR C BIT!
8195	027212	042737	000017			BIC	#17, 2\$	;CLEAR ADDRESS FIELD OF INSTRUCTION
8196	027220	050537	027230			BIS	R5, 2\$	;ADD ADDRESS TO INSTRUCTION
8197	027224					ROMCLK		;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	027224	004537	003230			JSR	R5, ROMCLK	;CLOCK INSTRUCTION
8198	027230	010000			2\$:	010000		;LOAD MAR
8199	027232	042737	000017	027250		BIC	#17, 3\$	;CLEAR ADDRESS OF INSTRUCTION
8200	027240	050537	027250			BIS	R5, 3\$	;ADD ADDRESS TO INSTRUCTION
8201	027244					ROMCLK		;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	027244	004537	003230			JSR	R5, ROMCLK	;CLOCK INSTRUCTION
8202	027250	040440			3\$:	040400! <2*20>		;BR - SUB W/C
8203	027252					ROMCLK		
(1)	027252	004537	003230			JSR	R5, ROMCLK	;CLOCK INSTRUCTION
8204	027256	061224				61224		;MOVE BR TO PORT4
8205	027260	111237	002636			MOVB	(R2), \$GDDAT	;PUT 'EXPECTED' IN \$GDDAT
8206	027264	116104	000004			MOVB	4(R1), R4	;PUT 'FOUND' IN R4
8207	027270	123704	002636			CMPB	\$GDDAT, R4	;DATA CORRECT?
8208	027274	001411				BEQ	4\$	;BR IF YES
8209	027276					ERROR	15, YES	;ALU ERROR
(5)	027310	104455				TRAP	C\$ERDF	
(6)	027312	000017				.WORD	15	
(6)	027314	004740				.WORD	EM15	
(6)	027316	007104				.WORD	ERR15	
8210	027320				4\$:	F\$CAPE	SEG	
(3)	027320	104410				TRAP	C\$ESCAPE	
(3)	027322	000014				.WORD	10000\$-	
8211	027324	005202				INC	R2	;NEXT DATA
8212	027326	005205				INC	R5	;NEXT ADDRESS
8213	027330	022705	000010			CMP	#10, R5	;DONE YET?
8214	027334	001324				BNE	1\$	;BR IF NO
8215	027336					ENDSEG		
(3)	027336				10000\$:			
(3)	027336	104405				TRAP	C\$ESEG	
8216	027340					EXIT	TST	
(3)	027340	104432				TRAP	C\$EXIT	
(3)	027342	000012				.WORD	L10141-	
8217	027344	377	000	376	5\$:	.BYTE	-1, 0, 376, -1, -1, 252, 124, -1	
	027347	377	377	252				
	027352	124	377					
8218								
8219								
8220								
8221	027354				.EVEN			
(3)	027354				ENDTST			
(3)	027354	104401			L10141:			
8222						TRAP	C\$ETST	

```

MVI NT      KMCSR,R1      ;GET DEVICE ADDRESS.
MOV          ;MASTER CLEAR M8200,4,7
MSTCLR      R5,,MSTCLR    ;CLEAR M8200,4,7
JSR          #5$,R2       ;POINTER TO CORRECT DATA
MOV         R;
CLR         P;,,MEMLD     ;LOAD 8 WORDS OF MAIN MEMRY
JSR         PC,SPLD       ;POINTER TO DATA
MEMDAT      ;LOAD 8 WORDS OF SP
SPDAT       ;POINTER TO DATA
BGNSEG
TRAP        C$BSEG
JSR         PC,CLRC       ;CLEAR C BIT!
BIC         #17,2$        ;CLEAR ADDRESS FIELD OF INSTRUCTION
BIS         R5,2$         ;ADD ADDRESS TO INSTRUCTION
ROMCLK      R5,,ROMCLK    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
JSR         ;CLOCK INSTRUCTION
010000      ;LOAD MAR
BIC         #17,3$        ;CLEAR ADDRESS OF INSTRUCTION
BIS         R5,3$         ;ADD ADDRESS TO INSTRUCTION
ROMCLK      R5,,ROMCLK    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
JSR         ;CLOCK INSTRUCTION
040400!<3*20>
ROMCLK      R5,,ROMCLK    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
JSR         ;CLOCK INSTRUCTION
61224      ;MOVE BR TO PORT4
MOV B       (R2), $GDDAT   ;PUT 'EXPECTED' IN $GDDAT
MOV B       4(R1), R4      ;PUT 'FOUND' IN R4
CMP B       $GDDAT, R4     ;DATA CORRECT?
BEQ         4$            ;BR IF YES
ERROR       15, YES       ;ALU ERROR
TRAP        C$ERDF
.WORD       15
.WORD       EM15
.WORD       ERR15
ESCAPE     SEG
TRAP        C$ESCAPE
.WORD       10000$-
INC         R2             ;NEXT DATA
INC         R5
CMP         #10, R5        ;DONE YET?
BNE         1$            ;BR IF NO
ENDSEG

```



(3)	027542				10000\$:		
(3)	027542	104405				TRAP	C\$ESEG
8264	027544					EXIT	TST
(3)	027544	104432				TRAP	C\$EXIT
(3)	027546	000012				.WORD	L10142-
8265	027550	001	001	000	5\$:	.BYTE	1,1,0,0,126,126,253,253
	027553	000	126	126			
	027556	253	253				
8266							
8267							
8268	027560					.EVEN	
(3)	027560					ENDTST	
(3)	027560	104401				L10142:	
8269						TRAP	C\$ETST
8270							
8271	027562						
(2)						BADHEAD	
8272						:***** TEST 57 *****	
8273						:*ALU TEST	
8274						:*TEST OF ALU FUNCTION 2A WITH C BIT CLEARED	
8275						:*ALU FUNCTION (A PLUS A) CODE=5	
8276						:*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA	
8277	027562					:*PERFORM THE FUNCTION, VERIFY THE RESULTS	
(2)						BADHEAD	
8278						:***** TEST 57 *****	
8279	027562						
(3)	027562					BGNTST	
8280	027562					T57::	
(1)	027562	013701	002716			MYINT	
8281	027566					MOV	KMCSR,R1
(1)	027566	004537	003142			MSTCLR	
8282	027572	005005				JSR	R5,,MSTCLR
8283	027574	012702	027754			CLR	R5
8284	027600	004737	003624			MOV	#5\$,R2
8285	027604	002654				JSR	PC,MEMLD
8286	027606	004737	003776			MEMDAT	
8287	027612	002664				JSR	PC,SPLD
8288	027614					SPDAT	
(3)	027614	104404				BGNSEG	
8289	027616	004737	004044			TRAP	C\$BSEG
8290	027622	042737	000017	027640	1\$:	JSR	PC,CLRC
8291	027630	050537	027640			BIC	#17,2\$
8292	027634					BIS	R5,2\$
(1)	027634	004537	003230			ROMCLK	
8293	027640	010000				JSR	R5,,ROMCLK
8294	027642	042737	000017	027660	2\$:	010000	
8295	027650	050537	027660			BIC	#17,3\$
8296	027654					BIS	R5,3\$
(1)	027654	004537	003230			ROMCLK	
8297	027660	040520				JSR	R5,,ROMCLK
8298	027662					040400!<5*20>	
(1)	027662	004537	003230			ROMCLK	
8299	027666	061224				JSR	R5,,ROMCLK
8300	027670	111237	002636			61224	
8301	027674	116104	000004			MOVB	(R2),%GDDAT
8302	027700	123704	002636			MOVB	4(R1),R4
						CMPB	%GDDAT,R4

```

;GET DEVICE ADDRESS.
;MASTER CLEAR DMC11
;CLEAR M8200,4,7
;MEM * SP ADDRESS
;POINTER TO CORRECT DATA
;LOAD 8 WORDS OF MAIN MEMORY
;POINTER TO DATA
;LOAD 8 WORDS OF SP
;POINTER TO DATA

;CLEAR C BIT!
;CLEAR ADDRESS FIELD OF INSTRUCTION
;ADD ADDRESS TO INSTRUCTION
;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
;CLOCK INSTRUCTION
;LOAD MAR
;CLEAR ADDRESS OF INSTRUCTION
;ADD ADDRESS TO INSTRUCTION
;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
;CLOCK INSTRUCTION
;BR 2A
;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
;CLOCK INSTRUCTION
;MOVE BR TO PORT4
;PUT 'EXPECTED' IN %GDDAT
;PUT 'FOUND' IN R4
;DATA CORRECT?

```

```

8303 027704 001411      BEQ      4$              ;BR IF YES
8304 027706      ERROR    15,YES          ;ALU ERROR
      (5) 027720 104455      TRAP    C$ERDF
      (6) 027722 000017      .WORD   15
      (6) 027724 004740      .WORD   EM15
      (6) 027726 007104      .WORD   ERR15
8305 027730      4$:      ESCAPE  SEG
      (3) 027730 104410      TRAP    C$ESCAPE
      (3) 027732 000014      .WORD   10000$-.
8306 027734 005202      INC      R2              ;NEXT DATA
8307 027736 005205      INC      R5              ;NEXT ADDRESS
8308 027740 022705 000010  CMP      #10,R5      ;DONE YET?
8309 027744 001324      BNE      1$              ;BR IF NO
8310 027746      ENDSEG
      (3) 027746
      (3) 027746 104405      10000$:
8311 027750      TRAP    C$ESEG
      (3) 027750 104432      EXIT    TST
      (3) 027752 000012      TRAP    C$EXIT
8312 027754      000      000      376  5$:      .WORD   L10143-.
      027757      376      252      252      .BYTE   0,0,376,376,252,252,124,124
      027762      124      124

8313
8314
8315 027764      .EVEN
      (3) 027764      ENDTST
      (3) 027764 104401      L10143:
8316      TRAP    C$ETST
8317
8318 027766      BADHEAD
      (2)
8319      ;***** TEST 58 *****
8320      ;*ALU TEST
8321      ;*TEST OF ALU FUNCTION A PLUS C WITH C BIT CLEARED
8322      ;*ALU FUNCTION (A PLUS C)      CODE=4
8323      ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
8324 027766      ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
      (2)      BADHEAD
8325      ;***** TEST 58 *****
8326 027766      BGNTST
      (3) 027766      T58::
8327 027766      MYINT
      (1) 027766 013701 002716  MOV     KMCSR,R1      ;GET DEVICE ADDRESS.
8328 027772      MSTCLR    ;MASTER CLEAR M8200,4,7
      (1) 027772 004537 003142  JSR     R5,.MSTCLR    ;CLEAR M8200,4,7
8329 027776 005005      CLR      R5              ;MEM + SP ADDRESS
8330 030000 012702 030160  MOV      #5$,R2      ;POINTER TO CORRECT DATA
8331 030004 004737 003624  JSR     PC,MEMLD      ;LOAD 8 WORDS OF MAIN MEMORY
8332 030010 002654      MEMDAT    ;POINTER TO DATA
8333 030012 004737 003776  JSR     PC,SPLD      ;LOAD 8 WORDS OF SP
8334 030016 002664      SPDAT     ;POINTER TO DATA
8335 030020
      (3) 030020 104404      BGNSEG
8336 030022 004737 004044      TRAP    C$BSEG
8337 030026 042737 000017 030044  1$:      JSR     PC,CLRC      ;CLEAR C BIT!
8338 030034 050537 030044      BIC     #17,2$      ;CLEAR ADDRESS FIELD OF INSTRUCTION
      BIS     R5,2$      ;ADD ADDRESS TO INSTRUCTION
  
```

8339	030040					ROMCLK		;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	030040	004537	003230			JSR	R5, ROMCLK	;CLOCK INSTRUCTION
8340	030044	010000			2\$:	010000		;LOAD MAR
8341	030046	042737	000017	030064		BIC	#17, 3\$	;CLEAR ADDRESS OF INSTRUCTION
8342	030054	050537	030064			BIS	R5, 3\$	;ADD ADDRESS TO INSTRUCTION
8343	030060					ROMCLK		;NEXT WORD IS INSTRUCTION, ROMCLK PC=304
(1)	030060	004537	003230			JSR	R5, ROMCLK	;CLOCK INSTRUCTION
8344	030064	040500			3\$:	040400!<4*20>		;BR A PLUS C
8345	030066					ROMCLK		;NEXT WORD IS INSTRUCTION, ROMCLK PC=5305
(1)	030066	004537	003230			JSR	R5, ROMCLK	;CLOCK INSTRUCTION
8346	030072	061224				61224		;MOVE BR TO PORT4
8347	030074	111237	002636			MOVW	(R2), \$GDDAT	;PUT 'EXPECTED' IN \$GDDAT
8348	030100	116104	000004			MOVW	4(R1), R4	;PUT 'FOUND' IN R4
8349	030104	123704	002636			CMPS	\$GDDAT, R4	;DATA CORRECT?
8350	030110	001411				BEQ	4\$	;BR IS YES
8351	030112					ERROR	15, YES	;ALU ERROR
(5)	030124	104455				TRAP	C\$ERDF	
(6)	030126	000017				.WORD	15	
(6)	030130	004740				.WORD	EM15	
(6)	030132	007104				.WORD	ERR15	
8352	030134				4\$:	ESCAPE	SEG	
(3)	030134	104410				TRAP	C\$ESCAPE	
(3)	030136	000014				.WORD	10000\$-	
8353	030140	005202				INC	R2	;NEXT DATA
8354	030142	005205				INC	R5	;NEXT ADDRESS
8355	030144	022705	000010			CMP	#10, R5	;DONE YET?
8356	030150	001324				BNE	1\$	;BR IF NO
8357	030152					ENDSEG		
(3)	030152				10000\$:			
(3)	030152	104405				TRAP	C\$ESEG	
8358	030154					EXIT	TST	
(3)	030154	104432				TRAP	C\$EXIT	
(3)	030156	000012				.WORD	L10144-	
8359	030160	000	000	377	5\$:	.BYTE	0,0,-1,-1,125,125,252,252	
	030163	377	125	125				
	030166	252	252					
8360								
8361								
8362	030170					.EVEN		
(3)	030170					ENDTST		
(3)	030170	104401				L10144:		
8363						TRAP	C\$ETST	
8364								
8365	030172					BADHEAD		
(2)						;***** TEST 59 *****		
8366						;*ALU TEST		
8367						;*TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT CLEARED		
8368						;*ALU FUNCTION (A-B-1) CODE=17		
8369						;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA		
8370						;*PERFORM THE FUNCTION, VERIFY THE RESULTS		
8371	030172					BADHEAD		
(2)						;***** TEST 59 *****		
8372								
8373	030172					BGNTST		
(3)	030172					T59::		
8374	030172					MYINT		

(1)	030172	013701	002716			MOV	KMCSR,R1	:GET DEVICE ADDRESS.
8375	030176					MSTCLR		:MASTER CLEAR M8200,4,7
(1)	030176	004537	003142			JSR	R5,,MSTCLR	:CLEAR M8200,4,7
8376	030202	005005				CLR	R5	:MEM + SP ADDRESS
8377	030204	012702	030364			MOV	#5\$,R2	:POINTER TO CORRECT DATA
8378	030210	004737	003624			JSR	PC,MEMLD	:LOAD 8 WORDS OF MAIN MEMORY
8379	030214	002654				MEMDAT		:POINTER TO DATA
8380	030216	004737	003776			JSR	PC,SPLD	:LOAD 8 WORDS OF SP
8381	030222	002664				SPDAT		:POINTER TO DATA
8382	030224					BGNSEG		
(3)	030224	104404				TRAP	C\$BSEG	
8383	030226	004737	004044			JSR	PC,CLRC	:CLEAR C BIT!
8384	030232	042737	000017	030250	1\$:	BIC	#17,2\$	:CLEAR ADDRESS FIELD OF INSTRUCTION
8385	030240	050537	030250			BIS	R5,2\$	:ADD ADDRESS TO INSTRUCTION
8386	030244					ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	030244	004537	003230			JSR	R5,,ROMCLK	:CLOCK INSTRUCTION
8387	030250	010000				010000		:LOAD MAR
8388	030252	042737	000017	030270	2\$:	BIC	#17,3\$	:CLEAR ADDRESS OF INSTRUCTION
8389	030260	050537	030270			BIS	R5,3\$	:ADD ADDRESS TO INSTRUCTION
8390	030264					ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	030264	004537	003230			JSR	R5,,ROMCLK	:CLOCK INSTRUCTION
8391	030270	040760			3\$:	040400!<17*20>		:BR 2'S COMP SUB
8392	030272					ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	030272	004537	003230			JSR	R5,,ROMCLK	:CLOCK INSTRUCTION
8393	030276	061224				61224		:MOVE BR TO PORT4
8394	030300	111237	002636			MOVB	(R2),\$GDDAT	:PUT 'EXPECTED' IN \$GDDAT
8395	030304	116104	000004			MOVB	4(R1),R4	:PUT 'FOUND' IN R4
8396	030310	123704	002636			CMPS	\$GDDAT,R4	:DATA CORRECT?
8397	030314	001411				BEQ	4\$	:BR IS YES
8398	030316					ERROR	15,YES	:ALU ERROR
(5)	030330	104455				TRAP	C\$ERDF	
(6)	030332	000017				.WORD	15	
(6)	030334	004740				.WORD	EM15	
(6)	030336	007104				.WORD	ERR15	
8399	030340				4\$:	ESCAPE	SEG	
(3)	030340	104410				TRAP	C\$ESCAPE	
(3)	030342	000014				.WORD	10000\$-	
8400	030344	005202				INC	R2	:NEXT DATA
8401	030346	005205				INC	R5	:NEXT ADDRESS
8402	030350	022705	000010			CMP	#10,R5	:DONE YET?
8403	030354	001324				BNE	1\$	:BR IF NO
8404	030356					ENDSEG		
(3)	030356				10000\$:			
(3)	030356	104405				TRAP	C\$ESEG	
8405	030360					EXIT	TST	
(3)	030360	104432				TRAP	C\$EXIT	
(3)	030362	000012				.WORD	L10145-	
8406	030364	377	000	376	5\$:	.BYTE	-1,0,376,-1,-1,252,124,-1	
	030367	377	377	252				
	030372	124	377					
8407								
8408								
8409	030374				.EVEN			
(3)	030374				ENDTST			
(3)	030374	104401			L10145:			
8410					TRAP	C\$ETST		

8411  
8412 030376

(2)

8413

8414

8415

8416

8417

8418

8419 030376

(2)

8420

8421 030376

(3)

8422 030376

(1)

8423 030402 013701 002716

(1) 030402 004537 003142

8424 030406 005005

8425 030410 012702 030570

8426 030414 004737 003624

8427 030420 002654

8428 030422 004737 003776

8429 030426 002664

8430 030430

(3) 030430 104404

8431 030432 004737 004044

8432 030436 042737 000017 030454

8433 030444 050537 030454

8434 030450

(1) 030450 004537 003230

8435 030454 010000

8436 030456 042737 000017 030474

8437 030464 050537 030474

8438 030470

(1) 030470 004537 003230

8439 030474 040560

8440 030476

(1) 030476 004537 003230

8441 030502 061224

8442 030504 111237 002636

8443 030510 116104 000004

8444 030514 123704 002636

8445 030520 001411

8446 030522

(5) 030534 104455

(6) 030536 000017

(6) 030540 004740

(6) 030542 007104

8447 030544

(3) 030544 104410

(3) 030546 000014

8448 030550 005202

8449 030552 005205

8450 030554 022705 000010

8451 030560 001324

## BADHEAD

:\*\*\*\*\* TEST 60 \*\*\*\*\*

: \*ALU TEST

: \*TEST OF ALU FUNCTION DEC A WITH C BIT CLEARED

: \*ALU FUNCTION (A-1) CODE=7

: \*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA

: \*PERFORM THE FUNCTION, VERIFY THE RESULTS

## BADHEAD

:\*\*\*\*\* TEST 60 \*\*\*\*\*

BGNTEST  
T60::

MYINT

MOV KMCSR,R1

;GET DEVICE ADDRESS.

MSTCLR

;MASTER CLEAR CMC11

JSR R5,,MSTCLR

;CLEAR M8200,4,7

CLR R5

;MEM + SP ADDRESS

MOV #5\$,R2

;POINTER TO CORRECT DATA

JSR PC,MEMLD

;LOAD 8 WORDS OF MAIN MEMMOR

MEMDAT

;POINTER TO DATA

JSR PC,SPLD

;LOAD 8 WORDS OF SP

SPDAT

;POINTER TO DATA

BGNSEG

TRAP C\$BSEG

JSR PC,CLRC

;CLEAR C BIT!

BIC #17,2\$

;CLEAR ADDRESS FIELD OF INSTRUCTION

BIS R5,2\$

;ADD ADDRESS TO INSTRUCTION

ROMCLK

;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304

JSR R5,,ROMCLK

;CLOCK INSTRUCTION

010000

;LOAD MAR

BIC #17,3\$

;CLEAR ADDRESS OF INSTRUCTION

BIS R5,3\$

;ADD ADDRESS TO INSTRUCTION

ROMCLK

;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304

JSR R5,,ROMCLK

;CLOCK INSTRUCTION

040400!&lt;7\*20&gt;

;BR DEC A

ROMCLK

;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304

JSR R5,,ROMCLK

;CLOCK INSTRUCTION

61224

;MOVE BR TO PORT4

MOVB (R2),\$GDDAT

;PUT 'EXPECTED' IN \$GDDAT

MOVB 4(R1),R4

;PUT 'FOUND' IN R4

CMPB \$GDDAT,R4

;DATA CORRECT?

REQ 4\$

;BR IF YES

ERROR 15,YES

;ALU ERROR

TRAP C\$ERDF

.WORD 15

.WORD EM15

.WORD ERR15

.WORD ESCAPE

TRAP C\$ESCAPE

.WORD 10000\$-

INC R2

;NEXT DATA

INC R5

;NEXT ADDRESS

(MP #10,R5

;DONE YET?

BNE 1\$

;BR IF NO

8452 030562  
 (3) 030562  
 (3) 030562 104405  
 8453 030564  
 (3) 030564 104432  
 (3) 030566 000012  
 8454 030570 377 377 376 5\$:  
 030573 376 124 124  
 030576 251 251

10000\$: ENDSEG

TRAP C\$ESEG  
 EXIT TST  
 TRAP C\$EXIT  
 .WORD L10146-  
 .BYTE -1,-1,376,376,124,124,251,251

8455  
 8456  
 8457 030600  
 (3) 030600  
 (3) 030600 104401

.EVEN  
 ENDTST  
 L10146:

TRAP C\$ETST

8458  
 8459  
 8460 030602  
 (2)

BADHEAD

:\*\*\*\*\* TEST 61 \*\*\*\*\*  
 :\*ALU TEST  
 :\*TEST OF ALU FUNCTION SEL B WITH C BIT SET  
 :\*ALU FUNCTION (B) CODE=11  
 :\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
 :\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
 BADHEAD  
 :\*\*\*\*\* TEST 61 \*\*\*\*\*

8461  
 8462  
 8463  
 8464  
 8465  
 8466 030602  
 (2)  
 8467  
 8468 030602  
 (3) 030602  
 8469 030602  
 (1) 030602 013701 002716

BGNTST  
 T61::

MYINT  
 MOV KMCSR,R1 ;GET DEVICE ADDRESS.  
 MSTCLR ;MASTER CLEAR M8200,4,7  
 JSR R5,.MSTCLR ;CLEAR M8200,4,7  
 CLR R5 ;MEM + SP ADDRESS  
 MOV #5\$,R2 ;POINTER TO CORRECT DATA  
 JSR PC,MEMLD ;LOAD 8 WORDS OF MAIN MEMORY  
 MEMDAT ;POINTER TO DATA  
 JSR PC,SPLD ;LOAD 8 WORDS OF SP  
 SPDAT ;POINTER TO DATA  
 BGNSEG

8470 030606  
 (1) 030606 004537 003142  
 8471 030612 005005  
 8472 030614 012702 030774  
 8473 030620 004737 003624  
 8474 030624 002654  
 8475 030626 004737 003776  
 8476 030632 002664  
 8477 030634

TRAP

(3) 030634 104404  
 8478 030636 004737 004062  
 8479 030642 042737 000017 030660  
 8480 030650 050537 030660  
 8481 030654

1\$:

JSR PC,SETC ;SET C BIT!  
 BIC #17,2\$ ;CLEAR ADDRESS FIELD OF INSTRUCTION  
 BIS R5,2\$ ;ADD ADDRESS TO INSTRUCTION  
 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304  
 JSR R5,.ROMCLK ;CLOCK INSTRUCTION

(1) 030654 004537 003230  
 8482 030660 010000  
 8483 030662 042737 000017 030700  
 8484 030670 050537 030700  
 8485 030674

2\$:

010000 ;LOAD MAR  
 BIC #17,3\$ ;CLEAR ADDRESS OF INSTRUCTION  
 BIS R5,3\$ ;ADD ADDRESS TO INSTRUCTION  
 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304  
 JSR R5,.ROMCLK ;CLOCK INSTRUCTION

(1) 030674 004537 003230  
 8486 030700 040620  
 8487 030702  
 (1) 030702 004537 003230  
 8488 030706 061224  
 8489 030710 111237 002636  
 8490 030714 116104 000004

3\$:

040400.<11\*20> ;BR SEL B  
 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304  
 JSR R5,.ROMCLK ;CLOCK INSTRUCTION  
 61224 ;MOVE BR TO PORT4  
 MOVB (R2), \$GDDAT ;PUT 'EXPECTED' IN \$GDDAT  
 MOVB 4(R1), R4 ;PUT 'FOUND' IN R4

PC	Instruction	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418
----	-------------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

[illegible]



```

8552      .EVEN
8553 031210  ENDTST
      (3) 031210  L10150:
      (3) 031210 104401  TRAP  C$ETST
8554
8555
8556 031212  BADHEAD
      (2)
8557      ;***** TEST 63 *****
8558      ;*ALU TEST
8559      ;*TEST OF ALU FUNCTION A OR NOTB WITH C BIT SET
8560      ;*ALU FUNCTION (A OR NOTB)      CODE=12
8561      ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
8562 031212  ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
      (2)  BADHEAD
8563      ;***** TEST 63 *****
8564 031212  BGNST
      (3)  T63::
8565 031212  MYINT
      (1) 031212 013701 002716  MOV  KMCSR,R1      ;GET DEVICE ADDRESS.
8566 031216 004537 003142  MSTCLR      ;MASTER CLEAR M8200,4,7
      (1) 031216 004537 003142  JSR  R5,.,MSTCLR      ;CLEAR M8200,4,7
8567 031222 005005  CLR  R5      ;MEM + SP ADDRESS
8568 031224 012702 031404  MOV  #5$,R2      ;POINTER TO CORRECT DATA
8569 031230 004737 003624  JSR  PC,MEMLD      ;LOAD 8 WORDS OF MAIN MEMORY
8570 031234 002654  MEMDAT      ;POINTER TO DATA
8571 031236 004737 003776  JSR  PC,SPLD      ;LOAD 8 WORDS OF SP
8572 031242 002664  SPDAT      ;POINTER TO DATA
8573 031244  BGNSEG
      (3) 031244 104404  TRAP  C$BSEG
8574 031246 004737 004062  JSR  PC,SETC      ;SET C BIT!
8575 031252 042737 000017 031270  BIC  #17,2$      ;CLEAR ADDRESS FIELD OF INSTRUCTION
8576 031260 050537 031270  BIS  R5,2$      ;ADD ADDRESS TO INSTRUCTION
8577 031264  ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
      (1) 031264 004537 003230  JSR  R5,.,ROMCLK      ;CLOCK INSTRUCTION
8578 031270 010000 010000  ;LOAD MAR
8579 031272 042737 000017 031310  BIC  #17,3$      ;CLEAR ADDRESS OF INSTRUCTION
8580 031300 050537 031310  BIS  R5,3$      ;ADD ADDRESS TO INSTRUCTION
8581 031304  ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
      (1) 031304 004537 003230  JSR  R5,.,ROMCLK      ;CLOCK INSTRUCTION
8582 031310 040640 040400!<12*20>  ;BR  A OR NOTB
8583 031312  ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
      (1) 031312 004537 003230  JSR  R5,.,ROMCLK      ;CLOCK INSTRUCTION
8584 031316 061224 61224  ;MOVE BR TO PORT4
8585 031320 111237 002636  MOVB  (R2), $GDDAT      ;PUT 'EXPECTED' IN $GDDAT
8586 031324 116104 000004  MOVB  4(R1), R4      ;PUT 'FOUND' IN R4
8587 031330 123704 002636  CMPB  $GDDAT, R4      ;DATA CORRECT?
8588 031334 001411  BEQ  4$      ;BR IF YES
8589 031336  ERROR 15, YES      ;ALU ERROR
      (5) 031350 104455  TRAP  C$ERDF
      (6) 031352 000017  .WORD 15
      (6) 031354 004740  .WORD EM15
      (6) 031356 007104  .WORD ERR15
8590 031360 4$:  ESCAPE SEG
      (3) 031360 104410  TRAP  C$ESCAPE
      (3) 031362 000014  .WORD 10000$-

```

8591	031364	005202				INC	R2		:NEXT DATA
8592	031366	005205				INC	R5		:NEXT ADDRESS
8593	031370	022705	000010			CMP	#10,R5		:DONE YET?
8594	031374	001324				BNE	1\$		:BR IF NO
8595	031376					ENDSEG			
(3)	031376			10000\$:					
(3)	031376	104405				TRAP	C\$ESEG		
8596	031400					EXIT	TST		
(3)	031400	104432				TRAP	C\$EXIT		
(3)	031402	000012				.WORD	L10151-		
8597	031404	377	000	377	5\$:	.BYTE	-1,0,-1,-1,-1,125,252,-1		
	031407	377	377	125					
	031412	252	377						
8598									
8599									
8600	031414					.EVEN			
(3)	031414					ENDTST			
(3)	031414	104401				L1C151:			
8601						TRAP	C\$ETST		
8602									
8603	031416					BADHEAD			
(2)						:***** TEST 64 *****			
8604						:*ALU TEST			
8605						:*TEST OF ALU FUNCTION A AND B WITH C BIT SET			
8606						:ALU FUNCTION (A AND B) CODE=13			
8607						:*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA			
8608						:*PERFORM THE FUNCTION, VERIFY THE RESULTS			
8609	031416					BADHEAD			
(2)						:***** TEST 64 *****			
8610									
8611	031416					BGNTST			
(3)	031416					T64::			
8612	031416					MYINT			
(1)	031416	013701	002716			MOV	KMCSR,R1		:GET DEVICE ADDRESS.
8613	031422					MSTCLR			:MASTER CLEAR M8200,4,7
(1)	031422	004537	003142			JSR	R5,.MSTCLR		:CLEAR M8200,4,7
8614	031426	005005				CLR	R5		:MEM + SP ADDRESS
8615	031430	012702	031610			MOV	#5\$,R2		:POINTER TO CORRECT ADDRESS
8616	031434	004737	003624			JSR	PC,MEMLD		:LOAD 8 WORDS OF MAIN MEMORY
8617	031440	002654				MEMDAT			:POINTER TO DATA
8618	031442	004737	003776			JSR	PC,SPLD		:LOAD 8 WORDS OF SP
8619	031446	002664				SPDAT			:POINTER TO DATA
8620	031450					BGNSEG			
(3)	031450	104404				TRAP	C\$BSEG		
8621	031452	004737	004062			JSR	PC,SETC		:SET C BIT!
8622	031456	042737	000017	031474	1\$:	BIC	#17,2\$		:CLEAR ADDRESS FIELD OF INSTRUCTION
8623	031464	050537	031474			BIS	R5,2\$		:ADD ADDRESS TO INSTRUCTION
8624	031470					ROMCLK			:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	031470	004537	003230			JSR	R5,.ROMCLK		:CLOCK INSTRUCTION
8625	031474	010000				010000			:LOAD MAR
8626	031476	042737	000017	031514	2\$:	BIC	#17,3\$		:CLEAR ADDRESS OF INSTRUCTION
8627	031504	050537	031514			BIS	R5,3\$		:ADD ADDRESS TO INSTRUCTION
8628	031510					ROMCLK			:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	031510	004537	003230			JSR	R5,.ROMCLK		:CLOCK INSTRUCTION
8629	031514	040660			3\$:	040400.<13*20>			:BR A AND B
8630	031516					ROMCLK			:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304

(1)	031516	004537	003230			JSR	R5,ROMCLK		;CLOCK INSTRUCTION
8631	031522	061224				61224			;MOVE BR TO PORT4
8632	031524	111237	002636			MOVB	(R2),SGDDAT		;PUT 'EXPECTED' IN SGDDAT
8633	031530	116104	000004			MOVB	4(R1),R4		;PUT 'FOUND' IN R4
8634	031534	123704	002636			CMPB	SGDDAT,R4		;DATA CORRECT?
8635	031540	001411				BEQ	4\$		;BR IF YES
8636	031542					ERROR	23,YES		;ALU ERROR
(5)	031554	104455				TRAP	C\$ERDF		
(6)	031556	000027				.WORD	23		
(6)	031560	005212				.WORD	EM23		
(6)	031562	007370				.WORD	ERR23		
8637	031564				4\$:	ESCAPE	SEG		
(3)	031564	104410				TRAP	C\$ESCAPE		
(3)	031566	000014				.WORD	10000\$-		
8638	031570	005202				INC	R2		;NEXT DATA
8639	031572	005205				INC	R5		;NEXT ADDRESS
8640	031574	022705	000010			CMP	#10,R5		;DONE YET?
8641	031600	001324				BNE	1\$		;BR IF NO
8642	031602					ENDSEG			
(3)	031602				10000\$:				
(3)	031602	104405				TRAP	C\$ESEG		
8643	031604					EXIT	TST		
(3)	031604	104432				TRAP	C\$EXIT		
(3)	031606	000012				.WORD	L10152-		
8644	031610	000	000	000	5\$:	.BYTE	0,0,0,-1,125,0,0,252		
	031613	377	125	000					
	031616	000	252						
8645									
8646									
8647	031620				.EVEN				
(3)	031620				ENDTST				
(3)	031620	104401			L10152:	TRAP	C\$ETST		
8648									
8649									
8650	031622								
(2)									
8651									
8652									
8653									
8654									
8655									
8656	031622								
(2)									
8657									
8658	031622								
(3)	031622				BGNTST				
8659	031622				T65::				
(1)	031622	013701	002716			MYINT			
8660	031626					MOV	KMCSR,R1		;GET DEVICE ADDRESS.
(1)	031626	004537	003142			MSTCLR			;MASTER C'EAR M8200,4,7
8661	031632	005005				JSR	R5,MSTCLR		;CLEAR M8200,4,7
8662	031634	012702	032014			CLR	R5		;MEM + SP ADDRESS
8663	031640	004737	003624			MOV	#5\$,R2		;POINTER TO CORRECT DATA
8664	031644	002654				JSR	PC,MEMLD		;LOAD 8 WORDS OF MAIN MEMORY
8665	031646	004737	003776			MEMDAT			;POINTER TO DATA
8666	031652	002664				JSR	PC,SPLD		;LOAD 8 WORDS OF SP
						SPDAT			;POINTER TO DATA

```

8667 031654 BGNSEG
(3) 031654 104404 TRAP C$BSEG
8668 031656 004737 004062 031700 1$: JSR PC,SETC ;SET C BIT!
8669 031662 042737 000017 BIC #17,2$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
8670 031670 050537 031700 BIS R5,2$ ;ADD ADDRESS TO INSTRUCTION
8671 031674 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 031674 004537 003230 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
8672 031700 010000 031720 2$: 010000 ;LOAD MAR
8673 031702 042737 000017 BIC #17,3$ ;CLEAR ADDRESS OF INSTRUCTION
8674 031710 050537 031720 BIS R5,3$ ;ADD ADDRESS TO INSTRUCTION
8675 031714 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 031714 004537 003230 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
8676 031720 040700 3$: 040400!<14*20> ;BR A OR B
8677 031722 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 031722 004537 003230 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
8678 031726 061224 61224 ;MOVE BR TO PORT4
8679 031730 111237 002636 MOVB (R2), $GDDAT ;PUT 'EXPECTED' IN R4
8680 031734 116104 000004 MOVB 4(R1), R4 ;PUT 'FOUND' IN R4
8681 031740 123704 002636 CMPB $GDDAT, R4 ;DATA CORRECT?
8682 031744 001411 4$ BEQ 4$ ;BR IF YES
8683 031746 ERROR 23, YES ;ALU ERROR
(5) 031760 104455 TRAP C$ERDF
(6) 031762 000027 .WORD 23
(6) 031764 005212 .WORD EM23
(6) 031766 007370 .WORD ERR23
8684 031770 4$: ESCAPE SEG
(3) 031770 104410 TRAP C$ESCAPE
(3) 031772 000014 .WORD 10000$-.
8685 031774 005202 INC R2 ;NEXT DATA
8686 031776 005205 INC R5 ;NEXT ADDRESS
8687 032000 022705 000010 CMP #10, R5 ;DONE YET?
8688 032004 001324 BNE 1$ ;BR IF NO
8689 032006 10000$: ENDSEG
(3) 032006 377 377 5$: TRAP C$ESEG
(3) 032006 104405 EXIT TST
8690 032010 104432 TRAP C$EXIT
(3) 032012 000012 .WORD L10153-.
8691 032014 000 377 .BYTE 0,-1,-1,-1,125,-1,-1,252
032017 377
032022 377 252

8692
8693 .EVEN
8694 032024 ENDTST
(3) 032024 L10153:
(3) 032024 104401 TRAP C$ETST
8695
8696
8697 032026 BADHEAD
(2)
8698 :***** TEST 66 *****
8699 :*ALU TEST
8700 :*TEST OF ALU FUNCTION A XOR B WITH C BIT SET
8701 :*ALU FUNCTION (A XOR B) CODE=15
8702 :*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
8703 :*PERFORM THE FUNCTION, VERIFY THE RESULTS

```

```
8704 032026          BADHEAD
(2)                  ;***** TEST 66 *****
8705 032026          BGNTST
(3) 032026          T66::
8706 032026          MYINT
(1) 032026 013701 002716 MOV KMCSR,R1 ;GET DEVICE ADDRESS.
8707 032032          MSTCLR ;MASTER CLEAR M8200,4,7
(1) 032032 004537 003142 JSR R5,.MSTCLR ;CLEAR M8200,4,7
8708 032036 005005 CLR R5 ;MEM + SP ADDRESS
8709 032040 012702 032220 MOV #5$,R2 ;POINTER TO CORRECT DATA
8710 032044 004737 003624 JSR PC,MEMLD ;LOAD 8 WORDS OF MAIN MEMORY
8711 032050 002654 MEMDAT ;POINTER TO DATA
8712 032052 004737 003776 JSR PC,SPLD ;LOAD 8 WORDS OF SP
8713 032056 0026.4 SPDAT ;POINTER TO DATA
8714 032060          BGNSEG
(3) 032060 104404 TRAP C$BSEG
8715 032062 004737 004062 1$: JSR PC,SETC ;SET C BIT!
8716 032066 042737 000017 032104 BIC #17,2$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
8717 032074 050537 032104 BIS R5,2$ ;ADD ADDRESS TO INSTRUCTION
8718 032100          ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 032100 004537 003230 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
8719 032104 010000 2$: 010000 ;LOAD MAR
8720 032106 042737 000017 032124 BIC #17,3$ ;CLEAR ADDRESS OF INSTRUCTION
8721 032114 050537 032124 BIS R5,3$ ;ADD ADDRESS TO INSTRUCTION
8722 032120          ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 032120 004537 003230 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
8723 032124 040720 3$: 040400!<15*20> ;BR A XCR B
8724 032126          ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 032126 004537 003230 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
8725 032132 061224 61224 ;MOVE BR TO PORT4
8726 032134 111237 002636 MOVB (R2), $GDDAT ;PUT 'EXPECTED' IN $GDDAT
8727 032140 116104 000004 MOVB 4(R1), R4 ;PUT 'FOUND' IN R4
8728 032144 123704 002636 CMPB $GDDAT, R4 ;DATA CORRECT?
8729 032150 001411 BFQ 4$ ;BR IF YES
8730 032152          ERROR 23, YES ;ALU ERROR
(5) 032164 104455 TRAP C$ERDF
(6) 032166 000027 .WORD 23
(6) 032170 005212 .WORD EM23
(6) 032172 007370 .WORD ERR23
8731 032174          ESCAPE 4$:
(3) 032174 104410 TRAP C$ESCAPE
(3) 032176 000014 .WORD 10000$-.
8732 032200 005202 INC R2 ;NEXT DATA
8733 032202 005205 INC R5 ;NEXT ADDRESS
8734 032204 022705 000010 CMP #10, R5 ;DONE YET?
8735 032210 001324 BNE 1$ ;BR IF NO
8736 032212          ENDSEG
(3) 032212          10000$:
(3) 032212 104405 TRAP C$ESEG
8737 032214          EXIT TST
(3) 032214 104432 TRAP C$EXIT
(3) 032216 000012 .WORD L10154-.
8738 032220 000 377 377 5$: .BYTE 0,-1,-1,0,0,-1,-1,0
032223 000 000 377
032226 377 000
```

8739

```

8740
8741 032230      .EVEN
      (3) 032230      ENDTST
      (3) 032230 104401 L10154:
8742
8743
8744 032232      TRAP    C$ETST
      (2)
8745
8746
8747
8748
8749
8750 032232      BADHEAD
      (2)
8751
8752 032232      ;***** TEST 67 *****
      (3) 032232      ;*ALU TEST
8753 032232      ;*TEST OF ALU FUNCTION ADD WITH C BIT SET
      (1) 032232 013701 002716 ;*ALU FUNCTION (A PLUS B)      CODE=00
8754 032236      ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
      (1) 032236 004537 003142 ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
8755 032242 005005      -BADHEAD
8756 032244 012702 032424      ;***** TEST 67 *****
8757 032250 004737 003624      BGNTST
8758 032254 002654      T67::
8759 032256 004737 003776      MYINT
8760 032262 002664      MOV    KMCSR,R1      ;GET DEVICE ADDRESS.
8761 032264 104404      MSTCLR      ;MASTER CLEAR M8200,4,7
      (3) 032264      JSR    R5,.MSTCLR      ;CLEAR M8200,4,7
8762 032266 004737 004062      CLR    R5      ;MEM + SP ADDRESS
8763 032272 042737 000017 032310 1$: JSR    #5$,R2      ;POINTER TO CORRECT DATA
8764 032300 050537 032310      MOV    PC,MEMLD      ;LOAD 8 WORDS OF MAIN MEMORY
8765 032304      JSR    PC,SPLD      ;POINTER TO DATA
      (1) 032304 004537 003230      SPDAT      ;LOAD 8 WORDS OF SP
8766 032310 010000      BGNSEG      ;OINTER TO DATA
8767 032312 042737 000017 032330 2$: TRAP    C$BSEG
8768 032320 050537 032330      JSR    PC,SETC      ;SET C BIT!
      (1) 032324 004537 003230      BIC    #17,2$      ;CLEAR ADDRESS FIELD OF INSTRUCTION
8770 032330 040400!<00*20> 3$: BIS    R5,2$      ;ADD ADDRESS TO INSTRUCTION
8771 032332 004537 003230      ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
      (1) 032332 004537 003230      JSR    R5,.ROMCLK      ;CLOCK INSTRUCTION
8772 032336 061224      ;LOAD MAR
      (1) 032336 004537 003230      BIC    #17,3$      ;CLEAR ADDRESS OF INSTRUCTION
8773 032340 111237 002636      BIS    R5,3$      ;ADD ADDRESS TO INSTRUCTION
8774 032344 116104 000004      ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
      (1) 032344 004537 003230      JSR    R5,.ROMCLK      ;CLOCK INSTRUCTION
8775 032350 123704 002636      ;BR    ADD
      (1) 032350 004537 003230      ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
8776 032354 001411      JSR    R5,.ROMCLK      ;CLOCK INSTRUCTION
8777 032356      61224      ;MOVE BR TO PORT4
      (1) 032356 004537 003230      MOV    (R2),%GDDAT      ;PUT 'EXPECTED' IN %GDDAT
8778 032400      MOV    -4(R1),R4      ;PUT 'FOUND' IN R4
      (3) 032400 104410      CMP    %GDDAT,R4      ;DATA CORRECT?
      (3) 032402 000014      BEQ    4$      ;BR IF YES
      (5) 032370 104455      ERROR    23,YES      ;ALU ERROR
      (6) 032372 000027      TRAP    C$ERDF
      (6) 032374 005212      .WORD    23
      (6) 032376 007370      .WORD    EM23
      (6) 032376 007370      .WORD    ERR23
      (6) 032376 007370      .WORD    ESCAPE
8778 032400      4$: ESCAPE SEG
      (3) 032400 104410      TRAP    C$ESCAPE
      (3) 032402 000014      .WORD    10000$-
  
```

8779	032404	005202				INC	R2		;NEXT DATA
8780	032406	005205				INC	R5		;NEXT ADDRESS
8781	032410	022705	000010			CMP	#10,R5		;DONE YET?
8782	032414	001324				BNE	1\$		;BR IF NO
8783	032416					ENDSEG			
(3)	032416			10000\$:					
(3)	032416	104405				TRAP	C\$ESEG		
8784	032420					EXIT	TST		
(3)	032420	104432				TRAP	C\$EXIT		
(3)	032422	000012				.WORD	L10155-		
8785	032424	000	377	377	5\$:	.BYTE	0,-1,-1,376,252,-1,-1,124		
	032427	376	252	377					
	032432	377	124						
8786									
8787									
8788	032434					.EVEN			
(3)	032434					ENDTST			
(3)	032434	104401				L10155:			
8789						TRAP	C\$ETST		
8790									
8791	032436								
(2)						BADHEAD			
8792						;***** TEST 68 *****			
8793						;*ALU TEST			
8794						;*TEST OF ALU FUNCTION 2A W/C WITH C BIT SET			
8795						;*ALU FUNCTION (A PLUS A PLUS C) CODE=6			
8796						;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA			
8797	032436					;*PERFORM THE FUNCTION, VERIFY THE RESULTS			
(2)						BADHEAD			
8798						;***** TEST 68 *****			
8799	032436								
(3)	032436					BGNTST			
8800	032436					T68::			
(1)	032436	013701	002716			MYINT			
8801	032442					MOV	KMCSR,R1		;GET DEVICE ADDRESS.
(1)	032442	004537	003142			MSTCLR			;MASTER CLEAR M8200,4,7
8802	032446	005005				JSR	R5,.MSTCLR		;CLEAR M8200,4,7
8803	032450	012702	032630			CLR	R5		;MEM + SP ADDRESS
8804	032454	004737	003624			MOV	#5\$,R2		;POINTER TO CORRECT DATA
8805	032460	002654				JSR	PC,MEMLD		;LOAD 8 WORDS OF MAIN MEMORY
8806	032462	004737	003776			MEMDAT			;POINTER TO DATA
8807	032466	002664				JSR	PC,SPLD		;LOAD 8 WORDS OF SP
8808	032470					SPDAT			;POINTER TO DATA
(3)	032470	104404				BGNSEG			
8809	032472	004737	004062			TRAP	C\$BSEG		
8810	032476	042737	000017	032514	1\$:	JSR	PC,SETC		;SET C BIT!
8811	032504	050537	032514			BIC	#17,2\$		;CLEAR ADDRESS FIELD OF INSTRUCTION
8812	032510					BIS	R5,2\$		;ADD ADDRESS TO INSTRUCTION
(1)	032510	004537	003230			ROMCLK			;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
8813	032514	010000				JSR	R5,.ROMCLK		;CLOCK INSTRUCTION
8814	032516	042737	000017	032534	2\$:	010000			;LOAD MAR
8815	032524	050537	032534			BIC	#17,3\$		;CLEAR ADDRESS OF INSTRUCTION
8816	032530					BIS	R5,3\$		;ADD ADDRESS TO INSTRUCTION
(1)	032530	004537	003230			ROMCLK			;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
8817	032534	040540				JSR	R5,.ROMCLK		;CLOCK INSTRUCTION
8818	032536				3\$:	040400.<6*26>			;BR 2A W/C
						ROMCLK			;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304

(1)	032536	004537	003230			JSR	R5,ROMCLK		;CLOCK INSTRUCTION
8819	032542	061224				61224			;MOVE BR TO PORT4
8820	032544	111237	002636			MOVB	(R2),SGDDA		;PUT 'WXPECTED' IN SGDDAT
8821	032550	116104	000004			MOVB	4(R1),R4		;PUT 'FOUND' IN R4
8822	032554	123704	002636			CMPS	SGDDAT,R4		;DATA CORRECT?
8823	032560	001411				BEQ	4\$		;BR IF YES
8824	032562					ERROR	23,YES		;ALU ERROR
(5)	032574	104455				TRAP	C\$ERDF		
(6)	032576	000027				.WORD	23		
(6)	032600	005212				.WORD	EM23		
(6)	032602	007370				.WORD	ERR23		
8825	032604					ESCAPE	SEG		
(3)	032604	104410				TRAP	C\$ESCAPE		
(3)	032606	000014				.WORD	10000\$-		
8826	032610	005202				INC	R2		;NEXT DATA
8827	032612	005205				INC	R5		;NEXT ADDRESS
8828	032614	022705	000010			CMP	#10,R5		;DONE YET?
8829	032620	001324				BNE	1\$		;BR IF NO
8830	032622					ENDSEG			
(3)	032622								
(3)	032622	104405				TRAP	C\$ESEG		
8831	032624					EXIT	TST		
(3)	032624	104432				TRAP	C\$EXIT		
(3)	032626	000012				.WORD	L10156-		
8832	032630	001	001	377	5\$:	.BYTE	1,-1,-1,253,253,125,125		
	032633	377	253	253					
	032636	125	125						
8833									
8834									
8835	032640								
(3)	032640								
(3)	032640	104401				TRAP	C\$ETST		
8836									
8837									
8838	032642								
(2)									
8839									
8840									
8841									
8842									
8843									
8844	032642								
(2)									
8845									
8846	032642								
(3)	032642								
8847	032642								
(1)	032642	013701	002716			MYINT			
8848	032646					MOV	KMCSR,R1		;GET DEVICE ADDRESS.
(1)	032646	004537	003142			MSTCLR			;MASTER CLEAR M8200,4,7
8849	032652	005005				JSR	R5,MSTCLR		;CLEAR M8200,4,7
8850	032654	012702	033034			CLR	R5		;MEM + SP ADDRESS
8851	032660	004737	003624			MOV #5\$,R2			;POINTER TO CORRECT DATA
8852	032664	002654				JSR	PC,MEMLD		;LOAD 8 WORDS OF MAIN MEMORY
8853	032666	004737	003776			MEMDAT			;POINTER TO DATA
8854	032672	002664				JSR	PC,SPLD		;LOAD 8 WORDS OF SP
						SPDAT			;POINTER TO DATA

.EVEN  
ENDTST  
L10156:

BGNTST  
T69::

BADHEAD  
\*\*\*\*\* TEST 69 \*\*\*\*\*  
;\*ALU TEST  
;\*TEST OF ALU FUNCTION SUB WITH C BIT SET  
;\*ALU FUNCTION (A-B) CODE=16  
;\*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA  
;\*PERFORM THE FUNCTION, VERIFY THE RESULTS  
BADHEAD  
\*\*\*\*\* TEST 69 \*\*\*\*\*



```

8855 032674      BGNSEG
      (3) 032674 104404
8856 032676 004737 004062
8857 032702 042737 000017 032720 1$: JSR PC,SETC ;SET C BIT!
8858 032710 050537 032720 BIC #17,2$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
      (1) 032714 004537 003230 BIS R5,2$ ;ADD ADDRESS TO INSTRUCTION
8859 032714 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
      (1) 032714 004537 003230 JSR R5,ROMCLK ;CLOCK INSTRUCTION
8860 032720 010000 032740 2$: JSR 010000 ;LOAD MAR
8861 032722 042737 000017 BIC #17,3$ ;CLEAR ADDRESS OF INSTRUCTION
8862 032730 050537 032740 BIS R5,3$ ;ADD ADDRESS TO INSTRUCTION
8863 032734 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
      (1) 032734 004537 003230 JSR R5,ROMCLK ;CLOCK INSTRUCTION
8864 032740 040740 3$: JSR 040400!<16*20> ;BR SUB
8865 032742 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
      (1) 032742 004537 003230 JSR R5,ROMCLK ;CLOCK INSTRUCTION
8866 032746 061224 ;MOVE BR TO PORT4
8867 032750 111237 002636 MOVB (R2),SGDDAT ;PUT 'EXPECTED' IN SGDDAT
8868 032754 116104 000004 MOVB 4(R1),R4 ;PUT 'FOUND' IN R4
8869 032760 123704 002636 CMPB SGDDAT,R4 ;DATA CORRECT?
8870 032764 001411 BEQ 4$ ;BR IF YES
8871 032766 ERROR 23,YES ;ALU ERROR
      (5) 033000 104455 TRAP C$ERDF
      (6) 033002 000027 .WORD 23
      (6) 033004 005212 .WORD EM23
      (6) 033006 007370 .WORD ERR23
8872 033010 4$: ESCAPE SEG
      (3) 033010 104410 TRAP C$ESCAPE
      (3) 033012 000014 .WORD 10000$-
8873 033014 005202 INC R2 ;NEXT DATA
8874 033016 005205 INC R5 ;NEXT ADDRESS
8875 033020 022705 000010 CMP #10,R5 ;DONE YET?
8876 033024 001324 BNE 1$ ;BR IF NO
8877 033026 ENDSEG
      (3) 033026 10000$: TRAP C$ESEG
8878 033030 EXIT TST
      (3) 033030 104432 TRAP C$EXIT
      (3) 033032 000012 .WORD L10157-
8879 033034 000 001 377 5$: .BYTE 0,1,-1,0,0,253,125,0
      033037 000 000 253
      033042 125 000

8880
8881
8882 033044 .EVEN
      (3) 033044 ENDTST
      (3) 033044 104401 L10157: TRAP C$ETST
8883
8884
8885 033046 BADHEAD
      (2)
8886 ;***** TEST 70 *****
8887 ;*ALU TEST
8888 ;*TEST OF ALU FUNCTION ADD W/C WITH C BIT SET
8889 ;*ALU FUNCTION (A PLUS B PLUS C) CODE=01
8890 ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
8891 033046 ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
      BADHEAD

```

```

(2)
8892
8893 033046 BGNTST
(3) 033046 T70::
8894 033046 MYINT
(1) 033046 013701 002716 MOV KMCSR,R1 ;GET DEVICE ADDRESS.
8895 033052 MSTCLR ;MASTER CLEAR M8200,4,7
(1) 033052 004537 003142 JSR R5,,MSTCLR ;CLEAR M8200,4,7
8896 033056 005005 CLR R5 ;MEM +SP ADDRESS
8897 033060 012702 033240 MOV #5$,R2 ;POINTER TO CORRECT DATA
8898 033064 004737 003624 JSR PC,MEMLD ;LOAD 8 WORDS OF MAIN MEMORY
8899 033070 002654 MEMDAT ;POINTER TO DATA
8900 033072 004737 003776 JSR PC,SPLD ;LOAD 8 WORDS OF SP
8901 033076 002664 SPDAT ;POINTER TO DATA
8902 033100 BGNSEG
(3) 033100 104404 TRAP C$BSEG
8903 033102 004737 004062 1$: JSR PC,SETC ;SET C BIT.
8904 033106 042737 000017 033124 BIC #17,2$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
8905 033114 050537 033124 BIS R5,2$ ;ADD ADDRESS TO INSTRUCTION
8906 033120 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 033120 004537 003230 JSR R5,,ROMCLK ;CLOCK INSTRUCTION
8907 033124 010000 2$: 010000 ;LOAD MAR
8908 033126 042737 000017 033144 BIC #17,3$ ;CLEAR ADDRESS OF INSTRUCTION
8909 033134 050537 033144 BIS R5,3$ ;ADD ADDRESS TO INSTRUCTION
8910 033140 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 033140 004537 003230 JSR R5,,ROMCLK ;CLOCK INSTRUCTION
8911 033144 040420 3$: 040400!<01*20> ;BR - ADD W/C
8912 033146 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 033146 004537 003230 JSR R5,,ROMCLK ;CLOCK INSTRUCTION
8913 033152 061224 61224 ;MOVE BR TO PORT4
8914 033154 111237 002636 MOVB (R2),$GDDAT ;PUT 'EXPECTED' IN $GDDAT
8915 033160 116104 000004 MOVB 4(R1),R4 ;PUT 'FOUND' IN R4
8916 033164 123704 002636 CMPB $GDDAT,R4 ;DATA CORRECT?
8917 033170 001411 BEQ 4$ ;BR IF YES
8918 033172 ERROR 23,YES ;ALU ERROR
(5) 033204 104455 TRAP C$ERDF
(6) 033206 000027 .WORD 23
(6) 033210 005212 .WORD EM23
(6) 033212 007370 .WORD ERR23
8919 033214 4$: ESCAPE SEG
(3) 033214 104410 TRAP C$ESCAPE
(3) 033216 000014 .WORD 10000$-
8920 033220 005202 INC R2 ;NEXT DATA
8921 033222 005205 INC R5 ;NEXT ADDRESS
8922 033224 022705 000010 CMP #10,R5 ;DONE YET?
8923 033230 001324 BNE 1$ ;BR IF NO
8924 033232 10000$: ENDSEG
(3) 033232 TRAP C$ESEG
(3) 033232 104405 EXIT TST
(3) 033234 104432 TRAP C$EXIT
(3) 033236 000012 .WORD L10160-
8926 033240 001 000 000 5$: .BYTE 1,0,0,-1,253,0,0,125
033243 377 253 000
033246 000 125
8927

```

```

8928          .EVEN
8929 033250      ENDTST
      (3) 033250      L10160:
      (3) 033250 104401      TRAP    C$ETST

8930
8931
8932 033252      BADHEAD
      (2)
8933          ;***** TEST 71 *****
8934          ;*ALU TEST
8935          ;*TEST OF ALU FUNCTION SUB W/C WITH C BIT SET
8936          ;*ALU FUNCTION (A-B-C) CODE=2
8937          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
8938 033252      ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
      (2)      BADHEAD
8939          ;***** TEST 71 *****
8940
8941 033252      BGNTST
      (3) 033252      T71::
8942 033252      MYINT
      (1) 033252 013701 002716      MOV    KMCSR,R1      ;GET DEVICE ADDRESS.
8943 033256      MSTCLR      ;MASTER CLEAR M8200,4,7
      (1) 033256 004537 003142      JSR    R5,.,MSTCLR      ;CLEAR M8200,4,7
8944 033262 005005      CLR    R5      ;MEM + SP ADDRESS
8945 033264 012702 033444      MOV    #5$,R2      ;POINTER TO CORRECT DATA
8946 033270 004737 003624      JSR    PC,MEMLD      ;LOAD 8 WORDS OF MAIN MEMORY
8947 033274 002654      MEMDAT      ;POINTER TO DATA
8948 033276 004737 003776      JSR    PC,SPLD.      ;LOAD 8 WORDS OF SP
8949 033302 002664      SPDAT      ;POINTER TO DATA
8950 033304
      (3) 033304 104404      BGNSEG
8951 033306 004737 004062      TRAP    C$BSEG
8952 033312 042737 000017 033330 1$:      JSR    PC,SETC      ;SET C BIT!
8953 033320 050537 033330      BIC    #17,2$      ;CLEAR ADDRESS FIELD OF INSTRUCTION
8954 033324      BIS    R5,2$      ;ADD ADDRESS TO INSTRUCTION
      (1) 033324 004537 003230      ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
8955 033330 010000      JSR    R5,.,ROMCLK      ;CLOCK INSTRUCTION
8956 033332 042737 000017 033350 2$:      010000      ;LOAD MAR
8957 033340 050537 033350      BIC    #17,3$      ;CLEAR ADDRESS OF INSTRUCTION
8958 033344      BIS    R5,3$      ;ADD ADDRESS TO INSTRUCTION
      (1) 033344 004537 003230      ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=55304
8959 033350 040440      JSR    R5,.,ROMCLK      ;CLOCK INSTRUCTION
8960 033352      040400!<2*20>      ;BR SUB W/C
      (1) 033352 004537 003230      ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
8961 033356 061224      JSR    R5,.,ROMCLK      ;CLOCK INSTRUCTION
8962 033360 111237 002636      MOV    61224      ;MOVE BR TO PORT4
8963 033364 116104 000004      MOV    (R2), $GDDAT      ;PUT 'EXPECTED' IN $GDDAT
8964 033370 123704 002636      MOV    4(R1), R4      ;PUT 'FOUND' IN R.
8965 033374 001411      CMP    $GDDAT, R4      ;DATA CORRECT?
8966 033376      BEQ    4$      ;BR IF YES
      (5) 033410 104455      ERROR 23,YES      ;ALU ERROR
      (6) 033412 000027      TRAP    C$ERDF
      (6) 033414 005212      .WORD 23
      (6) 033416 007370      .WORD EM23
8967 033420      .WORD ERR23
      (3) 033420 104410      .WORD ESCAPE SEG
      4$:      TRAP    C$ESCAPE
  
```

```

(3) 033422 000014 .WORD 10000$-
8968 033424 005202 INC R2 ;NEXT DATA
8969 033426 005205 INC R5 ;NEXT ADDRESS
8970 033430 022705 000010 CMP #10,R5 ;DONE YET?
8971 033434 001324 BNE 1$ ;BR IF NO
8972 033436 ENDSEG
(3) 033436 10000$:
(3) 033436 104405 TRAP C$ESEG
8973 033440 EXIT TST
(3) 033440 104432 TRAP C$EXIT
(3) 033442 000012 .WORD L10161-
8974 033444 000 001 377 5$: .BYTE 0,1,-1,0,0,253,125,0
033447 000 000 253
033452 125 000

8975
8976 .EVEN
8977 033454 ENDTST L10161:
(3) 033454 104401 TRAP C$ETST
8978
8979
8980 033456 BADHEAD
(2) ;***** TEST 72 *****
8981 ;*ALU TEST
8982 ;*TEST OF ALU FUNCTION INC A WITH C BIT SET
8983 ;*ALU FUNCTION (A PLUS 1) CODE=3
8984 ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
8985 ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
8986 033456 BADHEAD
(2) ;***** TEST 72 *****
8987
8988 033456 BGNSTST
(3) 033456 T72::
8989 033456 MYINT
(1) 033456 013701 002716 MOV KMCSR,R1 ;GET DEVICE ADDRESS.
8990 033462 MSTCLR ;MASTER CLEAR M8200,4,7
(1) 033462 004537 003142 JSR R5,.MSTCLR ;CLEAR M8200,4,7
8991 033466 005005 CLR R5 ;MEM + SP ADDRESS
8992 033470 012702 033650 MOV #5$,R2 ;POINTER TO CORRECT DATA
8993 033474 004737 003624 JSR PC,MEMLD ;LOAD 8 WORDS OF MAIN MEMORY
8994 033500 002654 MEMDAT ;POINTER TO DATA
8995 033502 004737 003776 JSR PC,SPLD ;LOAD 8 WORDS OF SP
8996 033506 002664 SPDAT ;POINTER TO DATA
8997 033510 BGNSEG
(3) 033510 104404 TRAP C$BSEG
8998 033512 004737 004062 1$: JSR PC,SETC ;SET C BIT!
8999 033516 042737 000017 033534 BIC #17,2$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
9000 033524 050537 033534 BIS R5,2$ ;ADD ADDRESS TO INSTRUCTION
9001 033530 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 033530 004537 003230 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
9002 033534 010000 2$: 010000 ;LOAD MAR
9003 033536 042737 000017 033554 BIC #17,3$ ;CLEAR ADDRESS OF INSTRUCTION
9004 033544 050537 033554 BIS R5,3$ ;ADD ADDRESS TO INSTRUCTION
9005 033550 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 033550 004537 003230 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
9006 033554 040460 3$: 040400.<3*20> ;BR _ INC A

```

Address	Hex	Dec	Label	Instruction	Comment
9007	033556			ROMCLK	;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
(1)	033556	004537	003230	JSR R5,ROMCLK	;CLOCK INSTRUCTION
9008	033562	061224		61224	;MOVE BR TO PORT4
9009	033564	111237	002636	MOVB (R2),SGDDAT	;PUT 'EXPECTED' IN SGDDAT
9010	033570	116104	000004	MOVB 4(R1),R4	;PUT 'FOUND IN R4
9011	033574	123704	002636	CMPB SGDDAT,R4	;DATA CORRECT?
9012	033600	001411		BEQ 4\$	;BR IF YES
9013	033602			ERROR 23,YES	;ALU ERROR
(5)	033614	104455		TRAP C\$ERDF	
(6)	033616	000027		.WORD 23	
(6)	033620	005212		.WORD EM23	
(6)	033622	007370		.WORD ERR23	
9014	033624		4\$:	ESCAPE SEG	
(3)	033624	104410		TRAP C\$ESCAPE	
(3)	033626	000014		.WORD 10000\$-	
9015	033630	005202		INC R2	;NEXT DATA
9016	033632	005205		INC R5	;NEXT ADDRESS
9017	033634	022705	000010	CMP #10,R5	;DONE YET?
9018	033640	001324		BNE 1\$	;BR IF NO
9019	033642			ENDSEG	
(3)	033642		10000\$:		
(3)	033642	104405		TRAP C\$ESEG	
9020	033644			EXIT TST	
(3)	033644	104432		TRAP C\$EXIT	
(3)	033646	000012		.WORD L10162-	
9021	033650	001	*001	.BYTE 1,1,0,0,126,126,253,253	
	033653	000	126		
	033656	253	253		
9022					
9023					
9024	033660			.EVEN	
(3)	033660			ENDTST	
(3)	033660	104401		L10162: TRAP C\$ETST	
9025					
9026					
9027	033662			BADHEAD	
(2)				***** TEST 73 *****	
9028				*ALU TEST	
9029				*TEST OF ALU FUNCTION 2A WITH C BIT SET	
9030				*ALU FUNCTION (A PLUS A) CODE-5	
9031				*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA	
9032				*PERFORM THE FUNCTION, VERIFY THE RESULTS	
9033	033662			BADHEAD	
(2)				***** TEST 73 *****	
9034					
9035	033662			BGNTST	
(3)	033662			T73::	
9036	033662			MYINT	
(1)	033662	013701	002716	MOV KMCSR,R1	;GET DEVICE ADDRESS.
9037	033666			MSTCLR	;MASTER CLEAR M8200,4,7
(1)	033666	004537	003142	JSR R5,MSTCLR	;CLEAR M8200,4,7
9038	033672	005005		CLR R5	;MEM + SP ADDRESS
9039	033674	012702	034054	MOV #5\$,R2	;POINTER TO CORRECT DATA
9040	033700	004737	003624	JSR PC,MEMLD	;LOAD 8 WORDS OF MAIN MEMORY
9041	033704	002654		MEMDAT	;POINTER TO DATA
9042	033706	004737	003776	JSR PC,SPLD	;LOAD 8 WORDS OF SP

9043	033712	002664			SPDAT				
9044	033714				BGNSEG				: POINTER TO DATA
(3)	033714	104404			TRAP	C\$BSEG			
9045	033716	004737	004062		JSR	PC,SETC			: SET C BIT!
9046	033722	042737	000017	033740	BIC	#17,2\$			: CLEAR ADDRESS FIELD OF INSTRUCTION
9047	033730	050537	033740		BIS	R5,2\$			: ADD ADDRESS TO INSTRUCTION
9048	033734				ROMCLK				: NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	033734	004537	003230		JSR	R5,,ROMCLK			: CLOCK INSTRUCTION
9049	033740	010000			010000				: LOAD MAR
9050	033742	042737	000017	033760	BIC	#17,3\$			: CLEAR ADDRESS OF INSTRUCTION
9051	033750	050537	033760		BIS	R5,3\$			: ADD ADDRESS TO INSTRUCTION
9052	033754				ROMCLK				: NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	033754	004537	003230		JSR	R5,,ROMCLK			: CLOCK INSTRUCTION
9053	033760	040520			040400!	<5*20>			: BR 2A
9054	033762				ROMCLK				: NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	033762	004537	003230		JSR	R5,,ROMCLK			: CLOCK INSTRUCTION
9055	033766	061224			61224				: MOVE BR TO PORT4
9056	033770	111237	002636		MOVB	(R2),SGDDAT			: PUT 'EXPECTED' IN SGDDAT
9057	033774	116104	000004		MOVB	4(R1),R4			: PUT 'FOUND IN R4
9058	034000	123704	002636		CMPB	SGDDAT,R4			: DATA CORRECT?
9059	034004	001411			BEQ	4\$			: BR IF YES
9060	034006				ERROR	23,YES			: ALU ERROR
(5)	034020	104455			TRAP	C\$ERDF			
(6)	034022	000027			.WORD	23			
(6)	034024	005212			.WORD	EM23			
(6)	034026	007370			.WORD	ERR23			
9061	034030				ESCAPE	SEG			
(3)	034030	104410			TRAP	C\$ESCAPE			
(3)	034032	000014			.WORD	10000\$-			
9062	034034	005202			INC	R2			: NEXT DATA
9063	034036	005205			INC	R5			: NEXT ADDRESS
9064	034040	022705	000010		CMP	#10,R5			: DONE YET?
9065	034044	001324			BNE	1\$			: BR IF NO
9066	034046				ENDSEG				
(3)	034046				10000\$:				
(3)	034046	104405			TRAP	C\$ESEG			
9067	034050				EXIT	TST			
(3)	034050	104432			TRAP	C\$EXIT			
(3)	034052	000012			.WORD	L10163-			
9068	034054	000	000	376	.BYTE	0,0,376,376,252,252,124,124			
	034057	376	252	252					
	034062	124	124						
9069									
9070					.EVEN				
9071	034064				ENDTST				
(3)	034064				L10163:				
(3)	034064	104401			TRAP	C\$ETST			
9072									
9073									
9074	034066				BADHEAD				
(2)					:***** TEST 74 *****				
9075					:*ALU TEST				
9076					:*TEST OF ALU FUNCTION A PLUS C WITH C BIT SET				
9077					:*ALU FUNCTION (A PLUS C) CODE=4				
9078					:*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA				
9079					:*PERFORM THE FUNCTION, VERIFY THE RESULTS				

```

9080 034066          BADHEAD
(2)                :***** TEST 74 *****
9081
9082 034066          BGNTST
(3) 034066          T74::
9083 034066          MYINT
(1) 034066 013701 002716 MOV KMCSR,R1      ;GET DEVICE ADDRESS.
9084 034072          MSTCLR      ;MASTER CLEAR M8200,4,7
(1) 034072 004537 003142 JSR R5,,MSTCLR    ;CLEAR M8200,4,7
9085 034076 005005          CLR R5      ;MEM + SP ADDRESS
9086 034100 012702 034260 MOV #5$,R2     ;POINTER TO CORRECT DATA
9087 034104 004737 003624 JSR PC,MEMLD  ;LOAD 8 WORDS OF MAIN MEMORY
9088 034110 002654          MEMDAT ;POINTER TO DATA
9089 034112 004737 003776 JSR PC,SPLD   ;LOAD 8 WORDS OF SP
9090 034116 002664          SPDAT  ;POINTER TO DATA
9091 034120          BGNSEG
(3) 034120 104404          TRAP C$BSEG
9092 034122 004737 004062 JSR PC,SETC   ;SET C BIT!
9093 034126 042737 000017 034144 BIC #17,2$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
9094 034134 050537 034144 BIS R5,2$      ;ADD ADDRESS TO INSTRUCTION
9095 034140          ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 034140 004537 003230 JSR R5,,ROMCLK    ;CLOCK INSTRUCTION
9096 034144 010000          010000 ;LOAD MAR
9097 034146 042737 000017 034164 BIC #17,3$ ;CLEAR ADDRESS OF INSTRUCTION
9098 034154 050537 034164 BIS R5,3$      ;ADD ADDRESS TO INSTRUCTION
9099 034160          ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 034160 004537 003230 JSR R5,,ROMCLK    ;CLOCK INSTRUCTION
9100 034164 040500          3$: 040400!<4*20> ;BR A PLUS C
9101 034166          ROMCLK      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1) 034166 004537 003230 JSR R5,,ROMCLK    ;CLOCK INSTRUCTION
9102 034172 061224          61224 ;MOVE BR TO PORT4
9103 034174 111237 002636 MOVB (R2),$GDDAT ;PUT 'EXPECTED' IN $GDDAT
9104 034200 116104 000004 MOVB 4(R1),R4 ;PUT 'FOUND IN R4
9105 034204 123704 002636 CMPB $GDDAT,R4 ;DATA CORRECT?
9106 034210 001411          BEQ 4$      ;BR IF YES
9107 034212          ERROR 23,YES      ;ALU ERROR
(5) 034224 104455          TRAP C$ERDF
(6) 034226 000027          .WORD 23
(6) 034230 005212          .WORD EM23
(6) 034232 007370          .WORD ERR23
9108 034234          4$: ESCAPE SEG
(3) 034234 104410          TRAP C$ESCAPE
(3) 034236 000014          .WORD 10000$-
9109 034240 005202          INC R2      ;NEXT DATA
9110 034242 005205          INC R5      ;NEXT ADDRESS
9111 034244 022705 000010 CMP #10,R5   ;DONE YET?
9112 034250 001324          BNE 1$      ;BR IF NO
9113 034252          10000$: ENDSEG
(3) 034252          TRAP $ESEG
(3) 034252 104405          EXIT TST
9114 034254          TRAP C$EXIT
(3) 034256 000012          .WORD L10164-
9115 034260 001 001 000 5$: .BYTE 1,1,0,0,126,126,253,253
034263 000 126 126
034266 253 253

```

```
9116
9117
9118 034270      .EVEN
(3) 034270      ENDTST
(3) 034270 104401 L10164:
9119                                TRAP    C$ETST
9120 034272      BADHEAD
(2)                                ;***** TEST 75 *****
9121                                ;*ALU TEST
9122                                ;*TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT SET
9123                                ;*ALU FUNCTION (A-B-1) CODE=17
9124                                ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
9125                                ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
9126 034272      BADHEAD
(2)                                ;***** TEST 75 *****
9127
9128 034272      BGNTST
(3) 034272      175::
```



9130	034272				MYINT		
(1)	034272	013701	002716		MOV	KMCSR,R1	:GET DEVICE ADDRESS.
9131	034276				MSTCLR		:MASTER CLEAR M8200,4,7
(1)	034276	004537	003142		JSR	R5, .MSTCLR	:CLEAR M8200,4,7
9132	034302	005005			CLR	R5	:MEM + SP ADDRESS
9133	034304	012702	034464		MOV	#5\$,R2	:POINTER TO CORRECT DATA
9134	034310	004737	003624		JSR	PC, MEMLD	:LOAD 8 WORDS OF MAIN MEMORY
9135	034314	002654			MEMDAT		:POINTER TO DATA
9136	034316	004737	003776		JSR	PC, SPLD	:LOAD 8 WORDS OF SP
9137	034322	002664			SPDAT		:POINTER TO DATA
9138	034324				BGNSEG		
(3)	034324	104404			TRAP	C\$BSEG	
9139	034326	004737	004062		JSR	PC, SETC	:SET C BIT!
9140	034332	042737	000017	034350	BIC	#17, 2\$	:CLEAR ADDRESS FIELD OF INSTRUCTION
9141	034340	050537	034350		BIS	R5, 2\$	:ADD ADDRESS TO INSTRUCTION
9142	034344				ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	034344	004537	003230		JSR	R5, .ROMCLK	:CLOCK INSTRUCTION
9143	034350	010000					:LOAD MAR
9144	034352	042737	000017	034370	BIC	#17, 3\$	:CLEAR ADDRESS OF INSTRUCTION
9145	034360	050537	034370		BIS	R5, 3\$	:ADD ADDRESS TO INSTRUCTION
9146	034364				ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	034364	004537	003230		JSR	R5, .ROMCLK	:CLOCK INSTRUCTION
9147	034370	040760					:BR 2'S COMP SUB
9148	034372				ROMCLK	040400! <17*20>	:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
(1)	034372	004537	003230		JSR	R5, .ROMCLK	:CLOCK INSTRUCTION
9149	034376	061224					:MOVE BR TO PORT4
9150	034400	111237	002636		MOVB	(R2), \$GDDAT	:PUT 'EXPECTED' IN \$GDDAT
9151	034404	116104	000004		MOVB	4(R1), R4	:PUT 'FOUND IN R4
9152	034410	123704	002636		CMPB	\$GDDAT, R4	:DATA CORRECT?
9153	034414	001411			BEQ	4\$	:BR IF YES
9154	034416				ERROR	23, YES	:ALU ERROR
(5)	034430	104455			TRAP	C\$ERDF	
(6)	034432	000027			.WORD	23	
(6)	034434	005212			.WORD	EM23	
(6)	034436	007370			.WORD	ERR23	
9155	034440				ESCAPE	SEG	
(3)	034440	104410			TRAP	C\$ESCAPE	
(3)	034442	000014			.WORD	10000\$-	
9156	034444	005202			INC	R2	:NEXT DATA
9157	034446	005205			INC	R5	:NEXT ADDRESS
9158	034450	022705	000010		CMP	#10, R5	:DONE YET?
9159	034454	001324			BNE	1\$	:BR IF NO
9160	034456				ENDSEG		
(3)	034456						
(3)	034456	104405			TRAP	C\$FSEG	
9161	034460				EXIT	TST	
(3)	034460	104432			TRAP	C\$EXIT	
(3)	034462	000012			.WORD	L10165-	
9162	034464	377	000	376	.BYTE	-1, 0, 376, -1, -1, 252, 124, -1	
	034467	377	377	252			
	034472	124	377				
9163							
9164					.EVEN		
9165	034474				ENDTST		
(3)	034474				L10165:		
(3)	034474	104401			TRAP	C\$ETST	

9166									
9167									
9168	034476								
(2)									
9169									
9170									
9171									
9172									
9173									
9174	034476								
(2)									
9175									
9176	034476								
(3)	034476								
9177	034476								
(1)	034476	013701	002716						
9178	034502								
(1)	034502	004537	003142						
9179	034506	005005							
9180	034510	012702	034670						
9181	034514	004737	003624						
9182	034520	002654							
9183	034522	004737	003776						
9184	034526	002664							
9185	034530								
(3)	034530	104404							
9186	034532	004737	004062						
9187	034536	042737	000017	034554	1\$:				
9188	034544	050537	034554						
9189	034550								
(1)	034550	004537	003230						
9190	034554	010000			2\$:				
9191	034556	042737	000017	034574					
9192	034564	050537	034574						
9193	034570								
(1)	034570	004537	003230						
9194	034574	040560			3\$:				
9195	034576								
(1)	034576	004537	003230						
9196	034602	061224							
9197	034604	111237	002636						
9198	034610	116104	000004						
9199	034614	123704	002636						
9200	034620	001411							
9201	034622								
(5)	034634	104455							
(6)	034636	000027							
(6)	034640	005212							
(6)	034642	007370							
9202	034644				4\$:				
(3)	034644	104410							
(3)	034646	000014							
9203	034650	005202							
9204	034652	005205							
9205	034654	022705	000010						
9206	034660	001324							

```

BADHEAD
***** TEST 76 *****
;*ALU TEST
;*TEST OF ALU FUNCTION DEC A WITH C BIT SET
;*ALU FUNCTION (A-1) CODE=7
;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
;*PERFORM THE FUNCTION, VERIFY THE RESULTS
BADHEAD
***** TEST 76 *****

BGNTST
T76::

MYINT
MOV KMCSR,R1 ;GET DEVICE ADDRESS.
MSTCLR R5 ;MASTER CLEAR M8200,4,7
JSR R5,MSTCLR ;CLEAR M8200,4,7
CLR R5 ;MEM + SP ADDRESS
MOV #5$,R2 ;POINTER TO CORRECT DATA
JSR PC,MEMLD ;LOAD 8 WORDS OF MAIN MEMORY
MEMDAT ;POINTER TO DATA
JSR PC,SPLD ;LOAD 8 WORDS OF SP
SPDAT ;POINTER TO DATA
BGNSEG
TRAP C$BSEG
JSR PC,SETC ;SET C BIT!
BIC #17,2$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
BIS R5,2$ ;ADD ADDRESS TO INSTRUCTION
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
JSR R5,ROMCLK ;CLOCK INSTRUCTION
010000 ;LOAD MAR
BIC #17,3$ ;CLEAR ADDRESS OF INSTRUCTION
BIS R5,3$ ;ADD ADDRESS TO INSTRUCTION
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
JSR R5,ROMCLK ;CLOCK INSTRUCTION
040400!<7*20> ;BR DEC A
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
JSR R5,ROMCLK ;CLOCK INSTRUCTION
61224 ;MOVE BR TO PORT4
MOVB (R2),SGDDAT ;PUT 'EXPECTED' IN SGDDAT
MOVB 4(R1),R4 ;PUT 'FOUND IN R4
CMPB SGDDAT,R4 ;DATA CORRECT?
BEQ 4$ ;BR IF YES
ERROR 23,YES ;ALU ERROR
TRAP C$ERDF
WORD 23
WORD EM23
WORD ERR23
4$: ESCAPE SEG
TRAP C$ESCAPE
WORD 10000$-
INC R2 ;NEXT DATA
INC R5 ;NEXT ADDRESS
CMP #10,R5 ;DONE YET?
BNE 1$ ;BR IF NO
  
```

```
9207 034662          ENDSEG
(3) 034662          10000$:
(3) 034662 104405    TRAP  C$ESEG
9208 034664          EXIT  TST
(3) 034664 104432    TRAP  C$EXIT
(3) 034666 000012    .WORD L10166-
9209 034670          377   377   376 5$: .BYTE -1,-1,376,376,124,124,251,251
      034673          376   124   124
      034676          251   251

9210
9211
9212 034700          .EVEN
(3) 034700          ENDTST
(3) 034700 104401    L10166:
9213          TRAP  C$ETST
9214
```

CZDMPAO M8207 STATIC DIAG #1 MACY11 30A(1052) 17-JUL-79 14:39 D 16  
CZDMPA.P11 17-JUL-79 14:33 HARDWARE TESTS PAGE 59

SEQ 0198

9216  
9217  
9218  
9219  
9220  
9221

9223

.SBITL HARDWARE PARAMETER CODING SECTION

9224

9225

9226

9227

9228

9229

9230

9231

9232

9233

9234

9235

9236

034702

RGNHRD

(3)

034702

000022

.WORD L10167-L\$HARD/2

(3)

034704

L\$HARD::

9237

9238

034704

GPRMD

WMP,0,0,7,0,7,YES

(4)

034704

000032

.WORD

T\$CODE

(4)

034706

034750

.WORD

WMP

(4)

034710

000007

.WORD

7

(4)

034712

000000

.WORD

T\$LOLIM

(4)

034714

000007

.WORD

T\$HILIM

9239

034716

GPRMA

ADDRES,2,0,160000,177776,YES

(4)

034716

001031

.WORD

T\$CODE

(4)

034720

035027

.WORD

ADDRES

(4)

034722

160000

.WORD

T\$LOLIM

(4)

034724

177776

.WORD

T\$HILIM

9240

034726

GPRMA

VECTOR,4,0,0,674,YES

(4)

034726

002031

.WORD

T\$CODE

(4)

034730

035066

.WORD

VECTOR

(4)

034732

000000

.WORD

T\$LOLIM

(4)

034734

000674

.WORD

T\$HILIM

9241

034736

GPRMD

PRIITY,6,0,7000,4,7,YES

(4)

034736

003032

.WORD

T\$CODE

(4)

034740

035130

.WORD

PRIITY

(4)

034742

007000

.WORD

7000

(4)

034744

000004

.WORD

T\$LOLIM

(4)

034746

000007

.WORD

T\$HILIM

9242

GPRMD

LNUNIT,10,0,3,0,3,YES

9243

GPRMD

SWPAC1,12,0,377,0,377,YES

9244

GPRMD

SWPAC2,14,0,377,0,377,YES

9245

GPRMD

LOOPBK,16,0,40000,0,1,YES

9246

9247

034750

ENDHRD

(2)

.EVEN

(3)

034750

L10167:

9248

9249

034750

044127

041511

020110

WMP:

.ASCIZ

'WHICH MICRO-PROCESSOR:'

031756

044515

051103

026517

034764

051120

041517

051505

034772

047523

035122

000

9250

034777

060

046475

031070

.ASCIZ

'0-M8200,4=M8204,7-M8207'

035004

030060

032054

046475

035012

031070

032060

033454

035020

046475

031070

033460

9251	035026	000			
	035027	115	041511	047522	ADDRES: .ASCIZ /MICRO-PROCESSOR CSR ADDRESS : /
	035034	050055	047522	042503	
	035042	051523	051117	041440	
	035050	051123	040440	042104	
	035056	042522	051523	035040	
	035064	000040			
9252	035066	044515	051103	026517	VECTOR: .ASCIZ /MICRO-PROCESSOR VECTOR ADDRESS : /
	035074	051120	041517	051505	
	035102	047523	020122	042526	
	035110	052103	051117	040440	
	035116	042104	042522	051523	
	035124	035040	000040		
9253	035130	044515	051103	026517	PRIPTY: .ASCIZ /MICRO-PROCESSOR PRIORITY LEVEL : /
	035136	051120	041517	051505	
	035144	047523	020122	051120	
	035152	047511	044522	054524	
	035160	046040	053105	046105	
	035166	035040	000040		
9254	035172	044127	041511	020110	LNUNIT: .ASCIZ /WHICH LINE UNIT (0-3)? 0=NONE,1=M8201,2=M8202,3=M8203 : /
	035200	044514	042516	052440	
	035206	044516	020124	030050	
	035214	031455	037451	030040	
	035222	047075	047117	026105	
	035230	036461	034115	030062	
	035236	026061	036462	034115	
	035244	030062	026062	036463	
	035252	034115	030062	020063	
	035260	020072	000		
9255	035263	123	044527	041524	SWPAC1: .ASCIZ /SWITCH PACK #1 (DDCMP LINE #) : /
	035270	020110	040520	045503	
	035276	021440	020061	042050	
	035304	041504	050115	046040	
	035312	047111	020105	024443	
	035320	035040	000040		
9256	035324	053523	052111	044103	SWPAC2: .ASCIZ /SWITCH PACK #2 (BM873 BOOT ADR) : /
	035332	050040	041501	020113	
	035340	031043	024040	046502	
	035346	033470	020063	047502	
	035354	052117	040440	051104	
	035362	020051	020072	000	
9257	035367	127	046111	020114	LOOPBK: .ASCIZ /WILL TEST CONNECTOR(S) BE USED ? 0=NO,1=YES : /
	035374	042524	052123	041440	
	035402	047117	042516	052103	
	035410	051117	051450	020051	
	035416	042502	052440	042513	
	035424	020104	020077	036460	
	035432	047516	030454	054475	
	035440	051505	035040	000040	

9258  
 9259  
 9260  
 9261  
 9262  
 9263  
 9264

.EVEN

CZDMPAO M8207 STATIC DIAG #1 MACY11 30A(1052) 17-JUL-79 14:39 G 16  
CZDMPA.P11 17-JUL-79 14:33 PAGE 59-3  
HARDWARE PARAMETER CODING SECTION

SEQ 0201

9265

```
9267 .SBTTL SOFTWARE PARAMETER CODING SECTION
9268
9269
9270 ;////////////////////
9271 ;/ THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
9272 ;/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
9273 ;/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
9274 ;/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
9275 ;/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
9276 ;/ WITH THE OPERATOR.
9277 ;////////////////////
9278
9279 035446 BGNSFT
(3) 035446 000000 .WORD L10170-L$SOFT/2
(3) 035450 L$SOFT::
9280
9281
9282 035450 ENDSFT
(2) .EVEN
(3) 035450 L10170:
9283 .EVEN
9284
9285
9286
9287
9288
9289
9290 035450 ENDMOD
9291
9292
9293 035450 CORMAX:
9294
9295 035450 000000 .WORD 0 ;START OF NPR AREA (TEST 55)
9296 037776 037776 .-37776
9297 037776 000000 MEMEND: .WORD 0 ;END OF NPR AREA
9298 040000 LASTAD
(2) .EVEN
(4) 040000 000000 .WORD 0
(4) 040002 000000 .WORD 0
(3) 040004 L$LAST::
9299 000114 .LTN.ED=TESTNUM
9300
9301 ; W A R N I N G < < < <
9302
9303 ;AREA BETWEEN CORMAX AND MEMEND USED BY TESTS IN DIAGNOSTIC.
9304 ; NO PATCHS OR DATA MY BE STORED IN THIS AREA.
9305 ;A SMALL PATCH AREA IS PROVIDED NEAR AREA 'DEBUG' FOR YOUR USE.
9306 ;ALSO THE AREA ABOVE ADDRESS 077776 MAY BE USED.
9307
9308 ;ANYONE FOOLISH ENOUGH TO IGNOR THIS WARNING WILL BE DESTROYED.
9309
9310
9311
9312
9313 000001 .END
```



[illegible]

C\$DRPT= 000024	4807#												
C\$DU = 000053	4807#	5804											
C\$EDIT= 000000	4807#	4848											
C\$ERDF= 000055	4807#	5383	5852	5869	5902	5916	5942	5948	5977	5985	6006	6014	6034
	6042	6064	6070	6100	6114	6140	6155	6180	6194	6220	6229	6257	6285
	6304	6334	6353	6383	6402	6432	6451	6484	6507	6542	6568	6599	6618
	6648	6667	6697	6716	6746	6765	6795	6814	6844	6863	6893	6912	6942
	6961	6999	7017	7048	7068	7084	7123	7148	7192	7213	7241	7243	7266
	7268	7305	7338	7342	7376	7410	7443	7496	7537	7574	7584	7617	7653
	7704	7745	7791	7837	7883	7929	7975	8021	8067	8113	8161	8209	8257
	8304	8351	8398	8446	8493	8542	8589	8636	8683	8730	8777	8824	8871
	8918	8966	9013	9060	9107	9154	9201						
C\$ERHR= 000056	4807#												
C\$ERRO= 000060	4807#												
C\$ERSF= 000054	4807#												
C\$ERSO= 000057	4807#												
C\$ESCA= 000010	4807#	5845	5856	5903	5917	5943	5949	5978	5986	6007	6035	6065	6071
	6101	6115	6141	6156	6181	6195	6221	6230	6259	6286	6305	6335	6354
	6384	6403	6433	6452	6485	6508	6544	6569	6600	6619	6649	6668	6698
	6717	6747	6766	6796	6815	6845	6864	6894	6913	6943	6962	7000	7018
	7049	7069	7124	7149	7193	7214	7377	7411	7444	7654	7705	7746	7792
	7838	7884	7930	7976	8022	8068	8114	8162	8210	8258	8305	8352	8399
	8447	8494	8543	8590	8637	8684	8731	8778	8825	8872	8919	8967	9014
	9061	9108	9155	9202									
C\$ESEG= 000005	4807#	5908	5922	5952	5979	5987	6008	6015	6036	6043	6074	6105	6120
	6145	6161	6188	6200	6222	6231	6290	6310	6339	6359	6388	6408	6437
	6457	6490	6514	6549	6575	6604	6624	6653	6673	6702	6722	6751	6771
	6800	6820	6849	6869	6898	6918	6947	6967	7005	7024	7053	7092	7128
	7154	7198	7219	7706	7751	7797	7843	7889	7935	7981	8027	8073	8119
	8167	8215	8263	8310	8357	8404	8452	8499	8548	8595	8642	8689	8736
	8783	8830	8877	8924	8972	9019	9066	9113	9160	9207			
C\$ESUB= 000003	4807#	7159											
C\$ETST= 000001	4807#	5857	5871	5923	5953	5989	6017	6045	6075	6121	6162	6201	6232
	6261	6311	6360	6409	6458	6515	6576	6625	6674	6723	6772	6821	6870
	6919	6968	7025	7093	7160	7220	7246	7271	7308	7346	7381	7415	7448
	7498	7542	7587	7619	7672	7707	7756	7802	7848	7894	7940	7986	8032
	8078	8125	8172	8221	8268	8315	8362	8409	8457	8504	8553	8600	8647
	8694	8741	8788	8835	8882	8929	8977	9024	9071	9118	9165	9212	
C\$EXIT= 000032	4807#	7378	7412	7445	7668	7752	7798	7844	7890	7936	7982	8028	8074
	8120	8168	8216	8264	8311	8358	8405	8453	8500	8549	8596	8643	8690
	8737	8784	8831	8878	8925	8973	9020	9067	9114	9161	9208		
C\$GETB= 000026	4807#												
C\$GETW= 000027	4807#												
C\$GMAN= 000043	4807#												
C\$GPHR= 000042	4807#	5675											
C\$GPLO= 000030	4807#												
C\$GPRI= 000040	4807#												
C\$INIT= 000011	4807#	5753											
C\$INLP= 000020	4807#												
C\$MANI= 000050	4807#												
C\$MEM = 000031	4807#												
C\$MSG - 000023	4807#	5556	5557	5558	5559	5560	5561	5562	5563	5564	5565	5566	5568
	5570	5571	5572	5573	5574	5575	5576	5577	5578	5579	5580	5581	5582
	5583	5584	5585	5586	5587	5588	5589	5594					
C\$OPEN= 000034	4807#												
C\$PNTB= 000014	4807#	5556	5557	5558	5559	5560	5561	5562	5563	5564	5565	5566	5568

K 16

PAGE 60-2

CROSS REFERENCE TABLE -- USER SYMBOLS

	5570	5571	5572	5573	5574	5575	5576	5577	5578	5579	5580	5581	5582
C\$PNTF= 000017	5583	5584	5585	5586	5587	5588	5589						
C\$PNTS= 000016	4807#	5592	5593										
C\$PNTX= 000015	4807#												
C\$QIO = 000377	4807#												
C\$RDBU= 000007	4807#												
C\$REFG= 000047	4807#	5652	5655	5658	5662								
C\$RESE= 000033	4807#	5785	5803	7229	7304	7354							
C\$REVI= 000003	4807#	4848											
C\$RFLA= 000021	4807#												
C\$RPT = 000025	4807#	5624											
C\$SEFG= 000046	4807#												
C\$SPRI= 000041	4807#	7235	7239	7260	7264	7283	7298	7320	7336				
C\$SEVC= 000037	4807#												
C\$TPRI= 000013	4807#												
DFPTBL 002364 G	4894#												
DH0 005702	5499#	5566	5571	5572	5574	5575	5577	5584	5585				
DH1 005703	5501#	5559	5560										
DH2 005734	5502#	5556	5557	5558	5564	5565	5568	5573	5578	5579	5581	5583	5586
	5587	5588	5589										
DH27 006014	5505#	5580	5582										
DH3 005750	5503#	5561	5562	5563									
DH4 006000	5504#	5570	5576										
DIAGMC= 000000	4807												
EF.CON= 000036 G	4953#	5658											
EF.NEW= 000035 G	4953#	5655											
EF.PWR= 000034 G	4953#												
EF.RES= 000037 G	4953#	5662											
EF.STA= 000040 G	4953#	5652											
EM1 004245	5455#	5592											
EM10 004606	5467#												
EM11 004640	5468#	7376	7410	7443	7496	7537	7653						
EM12 004651	5469#												
EM13 004673	5470#	7574	7584	7617									
EM14 004715	5471#												
EM15 004740	5472#	7745	7791	7837	7883	7929	7975	8021	8067	8113	8161	8209	8257
	8304	8351	8398	8446	8589								
EM16 005003	5474#												
EM17 005050	5476#												
EM2 004273	5456#	5916	5948	6180	6194								
EM20 005077	5477#												
EM21 005144	5478#												
EM22 005173	5479#												
EM23 005212	5480#	8493	8542	8636	8683	8730	8777	8824	8871	8918	8966	9013	9060
	9107	9154	9201										
EM24 005253	5481#												
EM25 005437	5486#												
EM26 005460	5487#	5869	5902	5977	5985	6006	6014	6034	6042	6220	6229		
EM27 005511	5488#	5942	6064	6070	6100	6114	6140	6155	6285	6304	6334	6353	6383
	6402	6432	6451	6484	6507	6542	6568						
EM28 005540	5489#	6257											
EM29 005571	5490#	6599	6618	6648	6667	6697	6716	6746	6765	6795	6814	6844	6863
	6893	6912	6942	6961	6999								
EM3 004402	5458#	7048	7068	7084									
EM30 004337	5457#	7017											

EM31	005276	5482#	7241	7266	7338														
EM32	005325	5483#	7243	7268	7342														
EM33	005364	5484#	7305																
EM34	005421	5485#	7704																
EM35	005050	5475#																	
EM36	005624	5383	5492#																
EM37	005701	5493#	5852																
EM4	004430	5460#	7123	7148	7192														
EM5	004456	5462#	7213																
EM6	004517	5464#																	
EM7	004545	5465#																	
ENDBUG	003230	5216#																	
ENDIT	011240	5659	5743	5747	5750	5752#													
ERRFLG	002560	5006#																	
ERR1	006040 G	5556#																	
ERR10	006562 G	5563#																	
ERR11	006644 G	5564#	7376	7410	7443	7496	7537	7653											
ERR12	006722 G	5565#																	
ERR13	007000 G	5566#	7574	7584	7617														
ERR14	007026 G	5568#																	
ERR15	007104 G	5570#	7745	7791	7837	7883	7929	7975	8021	8067	8113	8161	8209	8257					
		8304	8351	8398	8446	8589													
ERR16	007162 G	5571#																	
ERR17	007210 G	5572#																	
ERR2	006116 G	5557#	5916	5948	6180	6194													
ERR20	007236 G	5573#																	
ERR21	007314 G	5574#																	
ERR22	007342 G	5575#																	
ERR23	007370 G	5576#	8493	8542	8636	8683	8730	8777	8824	8871	8918	8966	9013	9060					
		9107	9154	9201															
ERR24	007446 G	5577#																	
ERR25	007474 G	5578#																	
ERR26	007552 G	5579#	5869	5902	5977	5985	6006	6014	6034	6042	6220	6229							
ERR27	007630 G	5580#	5942	6064	6070	6100	6114	6140	6155	6285	6304	6334	6353	6383					
		6402	6432	6451	6484	6507	6542	6568											
ERR28	007712 G	5581#	6257																
ERR29	007770 G	5582#	6599	6618	6648	6667	6697	6716	6746	6765	6795	6814	6844	6863					
		6893	6912	6942	6961	6999													
ERR3	006174 G	5558#	7048	7068	7084														
ERR30	010052 G	5583#	7017																
ERR31	010130 G	5584#	7241	7266	7338														
ERR32	010156 G	5585#	7243	7268	7342														
ERR33	010204 G	5586#	7305																
ERR34	010262 G	5587#	7704																
ERR35	010340 G	5588#																	
ERR36	010416 G	5383	5589#																
ERR37	010474 G	5591#	5852																
ERR4	006252 G	5559#	7123	7148	7192														
ERR5	006334 G	5560#	7213																
ERR6	006416 G	5561#																	
ERR7	006500 G	5562#																	
EVL	= 000004 G	4953#																	
E\$END	= 002100	4807#																	
E\$LOAD	= 000035	4807#	4848																
FLAG	002620	5023#																	
FM1	004100	5444#	5556	5557	5558	5559	5560	5561	5562	5563	5564	5565	5566	5568					

	5570	5571	5572	5573	5574	5575	5576	5577	5578	5579	5580	5581	5582
	5583	5584	5585	5586	5587	5588	5589	5592					
FTIME 002646	5034#	5644	5648*										
F\$AU = 000015	4807#	5819	5820										
F\$AUTO= 000020	4807#	5756	5774										
F\$BGN = 000040	4807#	4813	4857	5556	5557	5558	5559	5560	5561	5562	5563	5564	5565
	5566	5568	5570	5571	5572	5573	5574	5575	5576	5577	5578	5579	5580
	5581	5582	5583	5584	5585	5586	5587	5588	5589	5591	5607	5615	5638
	5756	5784	5801	5819	5838	5845	5856	5857	5863	5871	5891	5896	5903
	5911	5917	5923	5931	5935	5943	5949	5953	5968	5970	5978	5980	5986
	5989	5997	5999	6007	6009	6017	6025	6027	6035	6037	6045	6052	6056
	6065	6071	6075	6089	6094	6101	6107	6115	6121	6129	6134	6141	6147
	6156	6162	6170	6174	6181	6189	6195	6201	6209	6213	6221	6224	6230
	6232	6245	6259	6261	6269	6275	6286	6293	6305	6311	6319	6324	6335
	6342	6354	6360	6368	6373	6384	6391	6403	6409	6417	6422	6433	6440
	6452	6458	6466	6471	6485	6493	6508	6515	6523	6528	6544	6552	6569
	6576	6584	6589	6600	6607	6619	6625	6633	6638	6649	6656	6668	6674
	6682	6687	6698	6705	6717	6723	6731	6736	6747	6754	6766	6772	6780
	6785	6796	6803	6815	6821	6829	6834	6845	6852	6864	6870	6878	6883
	6894	6901	6913	6919	6927	6932	6943	6950	6962	6968	6976	6981	7000
	7008	7018	7025	7033	7038	7049	7056	7069	7093	7101	7106	7107	7124
	7130	7149	7159	7160	7168	7173	7193	7201	7214	7220	7227	7246	7253
	7271	7279	7308	7316	7346	7353	7377	7378	7381	7388	7411	7412	7415
	7422	7444	7445	7448	7456	7498	7510	7542	7551	7587	7595	7619	7627
	7654	7668	7672	7683	7690	7705	7707	7720	7729	7746	7752	7756	7766
	7775	7792	7798	7802	7812	7821	7838	7844	7848	7858	7867	7884	7890
	7894	7904	7913	7930	7936	7940	7950	7959	7976	7982	7986	7996	8005
	8022	8028	8032	8042	8051	8068	8074	8078	8088	8097	8114	8120	8125
	8136	8145	8162	8168	8172	8183	8193	8210	8216	8221	8232	8241	8258
	8264	8268	8279	8288	8305	8311	8315	8326	8335	8352	8358	8362	8373
	8382	8399	8405	8409	8421	8430	8447	8453	8457	8468	8477	8494	8500
	8504	8517	8526	8543	8549	8553	8564	8573	8590	8596	8600	8611	8620
	8637	8643	8647	8658	8667	8684	8690	8694	8705	8714	8731	8737	8741
	8752	8761	8778	8784	8788	8799	8808	8825	8831	8835	8846	8855	8872
	8878	8882	8893	8902	8919	8925	8929	8941	8950	8967	8973	8977	8988
	8997	9014	9020	9024	9035	9044	9061	9067	9071	9082	9091	9108	9114
	9118	9128	9138	9155	9161	9165	9176	9185	9202	9208	9212	9236	9279
	9290												
F\$CLEA= 000007	4807#	5784	5787										
F\$DU = 000016	4807#	5801	5804										
F\$END = 000041	4807#	4813	5556	5557	5558	5559	5560	5561	5562	5563	5564	5565	5566
	5568	5570	5571	5572	5573	5574	5575	5576	5577	5578	5579	5580	5581
	5582	5583	5584	5585	5586	5587	5588	5589	5594	5615	5624	5753	5774
	5787	5804	5820	5838	5845	5856	5857	5863	5871	5891	5903	5908	5917
	5922	5923	5931	5943	5949	5952	5953	5968	5978	5979	5986	5987	5989
	5997	6007	6008	6015	6017	6025	6035	6036	6043	6045	6052	6065	6071
	6074	6075	6089	6101	6105	6115	6120	6121	6129	6141	6145	6156	6161
	6162	6170	6181	6188	6195	6200	6201	6209	6221	6222	6230	6231	6232
	6245	6259	6261	6269	6286	6290	6305	6310	6311	6319	6335	6339	6354
	6359	6360	6368	6384	6388	6403	6408	6409	6417	6433	6437	6452	6457
	6458	6466	6485	6490	6508	6514	6515	6523	6544	6549	6569	6575	6576
	6584	6600	6604	6619	6624	6625	6633	6649	6653	6668	6673	6674	6682
	6698	6702	6717	6722	6723	6731	6747	6751	6766	6771	6772	6780	6796
	6800	6815	6820	6821	6829	6845	6849	6864	6869	6870	6878	6894	6898
	6913	6918	6919	6927	6943	6947	6962	6967	6968	6976	7000	7005	7018
	7024	7025	7033	7049	7053	7069	7092	7093	7101	7106	7124	7128	7149

CZDMPAO M8207 STATIC DIAG #1  
CZDMPA.P11 17-JUL-79 14:33

MACY11 30A(1052) 17-JUL-79 14:39 PAGE 60-5  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0208

F\$HARD= 000004  
F\$HW = 000013  
F\$INIT= 000006  
F\$JMP = 000050

F\$MOD = 000000  
F\$MSG = 000011

F\$PROT= 000021  
F\$PWR = 000017  
F\$RPT = 000012  
F\$SEG = 000003

F\$SOFT= 000005  
F\$SRV = 000010  
F\$SUB = 000002  
F\$SW = 000014  
F\$TEST= 000001

7154	7159	7160	7168	7193	7198	7214	7219	7220	7227	7246	7253	7271
7279	7308	7316	7346	7353	7377	7378	7381	7388	7411	7412	7415	7422
7444	7445	7448	7456	7498	7510	7542	7551	7587	7595	7619	7627	7654
7668	7672	7683	7705	7706	7707	7720	7746	7751	7752	7756	7766	7792
7797	7798	7802	7812	7838	7843	7844	7848	7858	7884	7889	7890	7894
7904	7930	7935	7936	7940	7950	7976	7981	7982	7986	7996	8022	8027
8028	8032	8042	8068	8073	8074	8078	8088	8114	8119	8120	8125	8136
8162	8167	8168	8172	8183	8210	8215	8216	8221	8232	8258	8263	8264
8268	8279	8305	8310	8311	8315	8326	8352	8357	8358	8362	8373	8399
8404	8405	8409	8421	8447	8452	8453	8457	8468	8494	8499	8500	8504
8517	8543	8548	8549	8553	8564	8590	8595	8596	8600	8611	8637	8642
8643	8647	8658	8684	8689	8690	8694	8705	8731	8736	8737	8741	8752
8778	8783	8784	8788	8799	8825	8830	8831	8835	8846	8872	8877	8878
8882	8893	8919	8924	8925	8929	8941	8967	8972	8973	8977	8988	9014
9019	9020	9024	9035	9061	9066	9067	9071	9082	9108	9113	9114	9118
9128	9155	9160	9161	9165	9176	9202	9207	9208	9212	9247	9282	9290
4807#	9236	9247										
4807#	4894	4909										
4807#	5638	5753										
4807#	5615	7378	7412	7445	7668	7752	7798	7844	7890	7936	7982	8028
8074	8120	8168	8216	8264	8311	8358	8405	8453	8500	8549	8596	8643
8690	8737	8784	8831	8878	8925	8973	9020	9067	9114	9161	9208	
4807#	4813	9290										
4807#	5556	5557	5558	5559	5560	5561	5562	5563	5564	5565	5566	5568
5570	5571	5572	5573	5574	5575	5576	5577	5578	5579	5580	5581	5582
5583	5584	5585	5586	5587	5588	5589	5591	5594				
4807#	4857	4861										
4807#												
4807#	5607	5624										
4807#	5896	5908	5911	5922	5935	5952	5970	5979	5980	5987	5999	6008
6009	6015	6027	6036	6037	6043	6056	6074	6094	6105	6107	6120	6134
6145	6147	6161	6174	6188	6189	6200	6213	6222	6224	6231	6275	6290
6293	6310	6324	6339	6342	6359	6373	6388	6391	6408	6422	6437	6440
6457	6471	6490	6493	6514	6528	6549	6552	6575	6589	6604	6607	6624
6638	6653	6656	6673	6687	6702	6705	6722	6736	6751	6754	6771	6785
6800	6803	6820	6834	6849	6852	6869	6883	6898	6901	6918	6932	6947
6950	6967	6981	7005	7008	7024	7038	7053	7056	7092	7107	7128	7130
7154	7173	7198	7201	7219	7690	7706	7729	7751	7775	7797	7821	7843
7867	7889	7913	7935	7959	7981	8005	8027	8051	8073	8097	8119	8145
8167	8193	8215	8241	8263	8288	8310	8335	8357	8382	8404	8430	8452
8477	8499	8526	8548	8573	8595	8620	8642	8667	8689	8714	8736	8761
8783	8808	8830	8855	8877	8902	8924	8950	8972	8997	9019	9044	9066
9091	9113	9138	9160	9185	9207							
4807#	9279	9282										
4807#												
4807#	7106	7159										
4807#	4923	4926										
4807#	5838	5857	5863	5871	5891	5923	5931	5953	5968	5989	5997	6017
6025	6045	6052	6075	6089	6121	6129	6162	6170	6201	6209	6232	6245
6261	6269	6311	6319	6360	6368	6409	6417	6458	6466	6515	6523	6576
6584	6625	6633	6674	6682	6723	6731	6772	6780	6821	6829	6870	6878
6919	6927	6968	6976	7025	7033	7093	7101	7160	7168	7220	7227	7246
7253	7271	7279	7308	7316	7346	7353	7381	7388	7415	7422	7448	7456
7498	7510	7542	7551	7587	7595	7619	7627	7672	7683	7707	7720	7756
7766	7802	7812	7848	7858	7894	7904	7940	7950	7986	7996	8032	8042
8078	8088	8125	8136	8172	8183	8221	8232	8268	8279	8315	8326	8362

CZDMPAO MB207 STATIC DIAG #1  
 CZDMPA.P11 17-JUL-79 14:33

 MACY11 30A(1052) 17-JUL-79 14:39 PAGE 60-6  
 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0209

	8373	8409	8421	8457	8468	8504	8517	8553	8564	8600	8611	8647	8658
	8694	8705	8741	8752	8788	8799	8835	8846	8882	8893	8929	8941	8977
	8988	9024	9035	9071	9082	9118	9128	9165	9176	9212			
GETPRM 010676	5663	5671#	5676										
GSNTQ= 000200	4807#												
GSDISP= 000003	4807#												
GSEXCP= 000400	4807#												
GSMLI= 000002	4807#												
GSLOLI= 000001	4807#												
GSNO = 000000	4807#												
GSOFFS= 000400	4807#	9238	9239	9240	9241								
GSOF SI= 000376	4807#	9238	9239	9240	9241								
GSPRMA= 000001	4807#	9239	9240										
GSPRMD= 000002	4807#	9238	9241										
GSPRML= 000000	4807#												
GSRADA= 000140	4807#												
GSRADB= 000000	4807#												
GSRADD= 000040	4807#												
GSRADL= 000120	4807#												
GSRADO= 000020	4807#	9238	9239	9240	9241								
GSXFER= 000004	4807#												
GSYES - 000010	4807#	9238	9239	9240	9241								
HELP = 000000	4794#	4840	4851	4874	5123	5609	5617						
HOE = 100000 G	4953#												
IBE - 010000 G	4953#												
IDU = 000040 G	4953#												
IER = 020000 G	4953#												
INIFLG 002674	5048#												
INSTU 003522	5308	5318#											
ISR = 000100 G	4953#												
IXE = 004000 G	4953#												
ISAU = 000041	4807#	5819#	5820#										
ISAUTO= 000041	4807#	5756#	5774#										
ISCLN - 000041	4807#	5784#	5787#										
ISDU = 000041	4807#	5801#	5804#										
ISHRD = 000041	9236#	9247#											
ISINIT= 000041	4807#	5638#	5753#										
ISMOD = 000041	4807#	4813#	9290#										
ISMSG = 000041	4807#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#	5566#	5568#
	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#
	5583#	5584#	5585#	5586#	5587#	5588#	5589#	5591#	5594#				
	4807#	4857#											
ISPROT= 000040	4807#												
ISPTAB= 000041	4807#												
ISPR = 000041	4807#												
ISRPT - 000041	4807#	5607#	5624#										
ISSEG = 000041	4807#	5838	5863	5891	5896#	5903	5908#	5911#	5917	5922#	5931	5935#	5943
	5949	5952#	5968	5970#	5978	5979#	5980#	5986	5987#	5997	5999#	6007	6008#
	6009#	6015#	6025	6027#	6035	6036#	6037#	6043#	6052	6056#	6065	6071	6074#
	6089	6094#	6101	6105#	6107#	6115	6120#	6129	6134#	6141	6145#	6147#	6156
	6161#	6170	6174#	6181	6188#	6189#	6195	6200#	6209	6213#	6221	6222#	6224#
	6230	6231#	6245	6269	6275#	6286	6290#	6293#	6305	6310#	6319	6324#	6335
	6339#	6342#	6354	6359#	6368	6373#	6384	6388#	6391#	6403	6408#	6417	6422#
	6433	6437#	6440#	6452	6457#	6466	6471#	6485	6490#	6493#	6508	6514#	6523
	6528#	6544	6549#	6552#	6569	6575#	6584	6589#	6600	6604#	6607#	6619	6624#
	6633	6638#	6649	6653#	6656#	6668	6673#	6682	6687#	6698	6702#	6705#	6717
	6722#	6731	6736#	6747	6751#	6754#	6766	6771#	6780	6785#	6796	6800#	6803#

SEQ 0210

KMCSRH	002720
KMCTL	002722
KMINUM	002612
KMP04	002724
KMP06	002726
KMRLVL	002710
KMRVEC	002706

6815	6820#	6829	6834#	6845	6849#	6852#	6864	6869#	6878	6883#	6894	6898#
6901#	6913	6918#	6927	6932#	6943	6947#	6950#	6962	6967#	6976	6981#	7000
7005#	7008#	7018	7024#	7033	7038#	7049	7053#	7056#	7069	7092#	7101	7106
7107#	7124	7128#	7130#	7154#	7168	7173#	7193	7198#	7201#	7214	7219#	7227
7253	7279	7316	7353	7388	7422	7456	7510	7551	7595	7627	7683	7690#
7705	7706#	7720	7729#	7746	7751#	7766	7775#	7792	7797#	7812	7821#	7838
7843#	7858	7867#	7884	7889#	7904	7913#	7930	7935#	7950	7959#	7976	7981#
7996	8005#	8022	8027#	8042	8051#	8068	8073#	8088	8097#	8114	8119#	8136
8145#	8162	8167#	8183	8193#	8210	8215#	8232	8241#	8258	8263#	8279	8288#
8305	8310#	8326	8335#	8352	8357#	8373	8382#	8399	8404#	8421	8430#	8447
8452#	8468	8477#	8494	8499#	8517	8526#	8543	8548#	8564	8573#	8590	8595#
8611	8620#	8637	8642#	8658	8667#	8684	8689#	8705	8714#	8731	8736#	8752
8761#	8778	8783#	8799	8808#	8825	8830#	8846	8855#	8872	8877#	8893	8902#
8919	8924#	8941	8950#	8967	8972#	8988	8997#	9014	9019#	9035	9044#	9061
9066#	9082	9091#	9108	9113#	9128	9138#	9155	9160#	9176	9185#	9202	9207#
4807#												
9279#	9282#											
4807#												
4807#	5838	5863	5891	5931	5968	5997	6025	6052	6089	6129	6170	6209
6245	6269	6319	6368	6417	6466	6523	6584	6633	6682	6731	6780	6829
6878	6927	6976	7033	7101	7106#	7159#	7168	7227	7253	7279	7316	7353
7388	7422	7456	7510	7551	7595	7627	7683	7720	7766	7812	7858	7904
7950	7996	8042	8088	8136	8183	8232	8279	8326	8373	8421	8468	8517
8564	8611	8658	8705	8752	8799	8846	8893	8941	8988	9035	9082	9128
9176												
4807#	5838#	5845	5856	5857#	5863#	5871#	5891#	5923#	5931#	5953#	5968#	5989#
5997#	6017#	6025#	6045#	6052#	6075#	6089#	6121#	6129#	6162#	6170#	6201#	6209#
6232#	6245#	6259	6261#	6269#	6311#	6319#	6360#	6368#	6409#	6417#	6458#	6466#
6515#	6523#	6576#	6584#	6625#	6633#	6674#	6682#	6723#	6731#	6772#	6810#	6821#
6829#	6870#	6878#	6919#	6927#	6968#	6976#	7025#	7033#	7093#	7101#	7106	7149
7160#	7168#	7220#	7227#	7246#	7253#	7271#	7279#	7308#	7316#	7346#	7353#	7377
7378	7381#	7388#	7411	7412	7415#	7422#	7444	7445	7448#	7456#	7498#	7510#
7542#	7551#	7587#	7595#	7619#	7627#	7654	7668	7672#	7683#	7707#	7720#	7752
7756#	7766#	7798	7802#	7812#	7844	7848#	7858#	7890	7894#	7904#	7936	7940#
7950#												



KMTLVL	002714	5083#	5331*	5701*	5702*	
KMTVEC	002712	5082#	5329*	5698*	5699*	
LNUNIT	035172	9254#				
LOCK	002442	4989#				
LOE =	040000 G	4953#				
LOGDEV	002552	5003#	5666*	5672*	5673	5675 5770
LOKFLG	002676	5050#				
LOOPBK	035367	9257#				
LOT =	000010 G	4953#				
LTN.ED=	000114	9299#				
L\$ACP	002110 G	4848#				
L\$APT	002036 G	4848#				
L\$AU	0*1350 G	4848	5819#			
L\$AUT	002070 G	4848#				
L\$AUTO	011242 G	4848	5756#			
L\$CCP	002106 G	4848#				
L\$CLEA	011340 G	4848	5784#			
L\$CO	002032 G	4848#				
L\$DEPO	002011 G	4848#				
L\$DESC	002414 G	4849	4983#			
L\$DESP	002076 G	48 8#				
L\$DEVP	002060 G	4548#				
L\$DISP	002132 G	4848	4871#			
L\$DLY	002116 G	4848#				
L\$DTP	002040 G	4848#				
L\$DTYP	002034 G	4848#				
L\$DU	011344 G	4848	5801#			
L\$DUT	002072 G	4848#				
L\$DVTY	003130 G	4848	5116#			
L\$EF	002052 G	4848#				
L\$ETP	002102 G	4848#				
L\$EXP1	002042 G	4848#				
L\$EXP2	002044 G	4848#				
L\$EXP3	002046 G	4848#				
L\$EXP4	002064 G	4848#				
L\$EXP5	002066 G	4848#				
L\$HARD	034704 G	4848	9236#			
L\$HIME	002120 G	4848#				
L\$HPCP	002016 G	4848#				
L\$HPTP	002022 G	4848#				
L\$HW	002364 G	4848	4894#			
L\$ICP	002104 G	4848#				
L\$INIT	010554 G	4848	5638#			
L\$LANDP	002026 G	4848#				
L\$LAST	040004 G	4848	9298#			
L\$LOAD	002100 G	4848#				
L\$LUN	002074 G	4848#				
L\$MREV	002050 G	4848#				
L\$NAME	002000 G	4848#				
L\$PROT	002122 G	4848	4857#			
L\$PRT	002112 G	4848#				
L\$REPP	002062 G	4848#				
L\$REV	002010 G	4848#				
L\$RPT	010546 G	5607#				
L\$SOFT	035450 G	9279#				
L\$SPC	002056 G	4848#				

CZDMPAO M8207 STATIC DIAG #1  
CZDMPA.P11 17-JUL-79 14:33

MACY11 30A(1052) 17-JUL-79 14:39 F 1  
PAGE 60-9  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0212

L\$SPCP	002020	G	4848#		
L\$SPTP	002024	G	4848#		
L\$STA	002030	G	4848#		
L\$SW	002414	G	4923#		
L\$TEST	002114	G	4848#		
L\$TIML	002014	G	4848#		
L\$UNIT	002012	G	4848#	5673	
L10001	002412		4894	4909#	
L10002	002414		4923	4926#	
L10003	006114		5556#		
L10004	006172		5557#		
L10005	006250		5558#		
L10006	006332		5559#		
L10007	006414		5560#		
L10010	006476		5561#		
L10011	006560		5562#		
L10012	006642		5563#		
L10013	006720		5564#		
L10014	006776		5565#		
L10015	007024		5566#		
L10016	007102		5568#		
L10017	007160		5570#		
L10020	007206		5571#		
L10021	007234		5572#		
L10022	007312		5573#		
L10023	007340		5574#		
L10024	007366		5575#		
L10025	007444		5576#		
L10026	007472		5577#		
L10027	007550		5578#		
L10030	007626		5579#		
L10031	007710		5580#		
L10032	007766		5581#		
L10033	010050		5582#		
L10034	010126		5583#		
L10035	010154		5584#		
L10036	010202		5585#		
L10037	010260		5586#		
L10040	010336		5587#		
L10041	010414		5588#		
L10042	010472		5589#		
L10043	010544		5594#		
L10044	010552		5615	5624#	
L10045	011240		5753#		
L10046	011336		5774#		
L10047	011342		5787#		
L10050	011346		5804#		
L10051	011350		5820#		
L10052	011476		5845	5856	5857#
L10053	011542		5871#		
L10054	011730		5923#		
L10055	012074		5953#		
L10056	012230		5989#		
L10057	012360		6017#		
L10060	012510		6045#		
L10061	012652		6075#		

CZDMPAO M820/ STATIC DIAG #1  
CZDMPA.P11 17-JUL-79 14:33

MACY11 30A(1052) 17-JUL-79 14:39 PAGE 60-10  
CROSS REFERENCE TABLE -- USER SYMBOLS

G 1

SEQ 0213

L10062	013036	6121#		
L10063	013222	6162#		
L10064	013410	6201#		
L10065	013560	6232#		
L10066	013666	6259	6261#	
L10067	014116	6311#		
L10070	014346	6360#		
L10071	014576	6409#		
L10072	015026	6458#		
L10073	015312	6515#		
L10074	015606	6576#		
L10075	016036	6625#		
L10076	016266	6674#		
L10077	016516	6723#		
L10100	016746	6772#		
L10101	017176	6821#		
L10102	017426	6870#		
L10103	017656	6919#		
L10104	020106	6968#		
L10105	020404	7025#		
L10106	020724	7093#		
L10107	021270	7149	7160#	
L10110	021266	7159#		
L10111	021602	7220#		
L10112	021744	7246#		
L10113	022106	7271#		
L10114	022262	7308#		
L10115	022466	7346#		
L10116	022632	7377	7378	7381#
L10117	023002	7411	7412	7415#
L10120	023146	7444	7445	7448#
L10121	023334	7498#		
L10122	023550	7542#		
L10123	023766	7587#		
L10124	024120	7619#		
L10125	024336	7654	7668	7672#
L10126	024466	7707#		
L10127	024702	7752	7756#	
L10130	025106	7798	7802#	
L10131	025312	7844	7848#	
L10132	025516	7890	7894#	
L10133	025722	7936	7940#	
L10134	026126	7982	7986#	
L10135	026332	8028	8032#	
L10136	026536	8074	8078#	
L10137	026744	8120	8125#	
L10140	027150	8168	8172#	
L10141	027354	8216	8221#	
L10142	027560	8264	8268#	
L10143	027764	8311	8315#	
L10144	030170	8358	8362#	
L10145	030374	8405	8409#	
L10146	030600	8453	8457#	
L10147	031004	8500	8504#	
L10150	031210	8549	8553#	
L10151	031414	8596	8600#	

SEQ 0214

[illegible]

CZDMPAO M8207 STATIC DIAG #1  
CZDMPA.P11 17-JUL-79 14:33

J 1  
MACY11 30A(1052) 17-JUL-79 14:39 PAGE 60-13  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0216

6261	6275	6285	6286	6290	6293	6304	6305	6310	6311	6324	6334	6335
6339	6342	6353	6354	6359	6360	6373	6383	6384	6388	6391	6402	6403
6408	6409	6422	6432	6433	6437	6440	6451	6452	6457	6458	6471	6484
6485	6490	6493	6507	6508	6514	6515	6528	6542	6544	6549	6552	6568
6569	6575	6576	6589	6599	6600	6604	6607	6618	6619	6624	6625	6638
6648	6649	6653	6656	6667	6668	6673	6674	6687	6697	6698	6702	6705
6716	6717	6722	6723	6736	6746	6747	6751	6754	6765	6766	6771	6772
6785	6795	6796	6800	6803	6814	6815	6820	6821	6834	6844	6845	6849
6852	6863	6864	6869	6870	6883	6893	6894	6898	6901	6912	6913	6918
6919	6932	6942	6943	6947	6950	6961	6962	6967	6968	6981	6999	7000
7005	7008	7017	7018	7024	7025	7038	7048	7049	7053	7056	7068	7069
7084	7085	7092	7093	7106	7107	7123	7124	7128	7130	7148	7149	7154
7159	7160	7173	7192	7193	7198	7201	7213	7214	7219	7220	7229	7235
7239	7241	7243	7246	7260	7264	7266	7268	7271	7283	7298	7304	7305
7308	7320	7336	7338	7342	7346	7354	7376	7377	7378	7381	7410	7411
7412	7415	7443	7444	7445	7448	7496	7498	7537	7542	7574	7584	7587
7617	7619	7653	7654	7668	7672	7690	7704	7705	7706	7707	7729	7745
7746	7751	7752	7756	7775	7791	7792	7797	7798	7802	7821	7837	7838
7843	7844	7848	7867	7883	7884	7889	7890	7894	7913	7929	7930	7935
7936	7940	7959	7975	7976	7981	7982	7986	8005	8021	8022	8027	8028
8032	8051	8067	8068	8073	8074	8078	8097	8113	8114	8119	8120	8125
8145	8161	8162	8167	8168	8172	8193	8209	8210	8215	8216	8221	8241
8257	8258	8263	8264	8268	8288	8304	8305	8310	8311	8315	8335	8351
8352	8357	8358	8362	8382	8398	8399	8404	8405	8409	8430	8446	8447
8452	8453	8457	8477	8493	8494	8499	8500	8504	8526	8542	8543	8548
8549	8553	8573	8589	8590	8595	8596	8600	8620	8636	8637	8642	8643
8647	8667	8683	8684	8689	8690	8694	8714	8730	8731	8736	8737	8741
8761	8777	8778	8783	8784	8788	8808	8824	8825	8830	8831	8835	8855
8871	8872	8877	8878	8882	8902	8918	8919	8924	8925	8929	8950	8966
8967	8972	8973	8977	8997	9013	9014	9019	9020	9024	9044	9060	9061
9066	9067	9071	9091	9107	9108	9113	9114	9118	9138	9154	9155	9160
9161	9165	9185	9201	9202	9207	9208	9212	9236	9238	9239	9240	9241
9247	9279	9282	9298									
4807#	4820#	7106										
4807#	4822#	4909	4926	5556	5557	5558	5559	5560	5561	5562	5563	5564
5565	5566	5568	5570	5571	5572	5573	5574	5575	5576	5577	5578	5579
5580	5581	5582	5583	5584	5585	5586	5587	5588	5589	5594	5624	5753
5774	5787	5804	5820	5857	5871	5908	5922	5923	5952	5953	5979	5987
5989	6008	6015	6017	6036	6043	6045	6074	6075	6105	6120	6121	6145
6161	6162	6188	6200	6201	6222	6231	6232	6261	6290	6310	6311	6339
6359	6360	6388	6408	6409	6437	6457	6458	6490	6514	6515	6549	6575
6576	6604	6624	6625	6653	6673	6674	6702	6722	6723	6751	6771	6772
6800	6820	6821	6849	6869	6870	6898	6918	6919	6947	6967	6968	7005
7024	7025	7053	7092	7093	7128	7154	7159	7160	7198	7219	7220	7246
7271	7308	7346	7381	7415	7448	7498	7542	7587	7619	7672	7706	7707
7751	7756	7797	7802	7843	7848	7889	7894	7935	7940	7981	7986	8027
8032	8073	8078	8119	8125	8167	8172	8215	8221	8263	8268	8310	8315
8357	8362	8404	8409	8452	8457	8499	8504	8548	8553	8595	8600	8642
8647	8689	8694	8736	8741	8783	8788	8830	8835	8877	8882	8924	8929
8972	8977	9019	9024	9066	9071	9113	9118	9160	9165	9207	9212	9247
9282												
4807#	4819#	5838	5863	5891	5931	5968	5997	6025	6052	6089	6129	6170
6209	6245	6269	6319	6368	6417	6466	6523	6584	6633	6682	6731	6780
6829	6878	6927	6976	7033	7101	7168	7227	7253	7279	7316	7353	7388
7422	7456	7510	7551	7595	7627	7683	7720	7766	7812	7858	7904	7950
7996	8042	8088	8136	8183	8232	8279	8326	8373	8421	8468	8517	8564

SVCSUB= 000000  
SVCTAG= 000000

SVCTST 000000

K 1

CROSS REFERENCE TABLE -- USER SYMBOLS

```

TSHILI= 000007
TSLAST= 000001
TSLOLI= 000004
TSLSYM= 010000

```

9238#	9239#	9240#	9241#									
4807#	9298#											
9238#	9239#	9240#	9241#									
4807#	4909	4926	5556	5557	5558	5559	5560	5561	5562	5563	5564	5565
5566	5568	5570	5571	5572	5573	5574	5575	5576	5577	5578	5579	5580
5581	5582	5583	5584	5585	5586	5587	5588	5589	5594	5624	5753	5774
5787	5804	5820	5857	5871	5923	5953	5989	6017	6045	6075	6121	6162
6201	6232	6261	6311	6360	6409	6458	6515	6576	6625	6674	6723	6772
6821	6870	6919	6968	7025	7093	7159	7160	7220	7246	7271	7308	7346
7381	7415	7448	7498	7542	7587	7619	7672	7707	7756	7802	7848	7894
7940	7986	8032	8078	8125	8172	8221	8268	8315	8362	8409	8457	8504
8553	8600	8647	8694	8741	8788	8835	8882	8929	8977	9024	9071	9118
9165	9212	9247	9282									

TSLTNO= 000114  
TSNEST- 177777

9298#	9212	9247	9282									
4807#	4813#	4857#	4861#	4894#	4909#	4923#	4926#	5556#	5557#	5558#	5559#	5560#
5561#	5562#	5563#	5564#	5565#	5566#	5568#	5570#	5571#	5572#	5573#	5574#	5575#
5576#	5577#	5578#	5579#	5580#	5581#	5582#	5583#	5584#	5585#	5586#	5587#	5588#
5589#	5591#	5594#	5607#	5624#	5638#	5753#	5756#	5774#	5784#	5787#	5801#	5804#
5819#	5820#	5838#	5857#	5863#	5871#	5891#	5896#	5908#	5911#	5922#	5923#	5931#
5935#	5952#	5953#	5968#	5970#	5979#	5980#	5987#	5989#	5997#	5999#	6008#	6009#
6015#	6017#	6025#	6027#	6036#	6037#	6043#	6045#	6052#	6056#	6074#	6075#	6089#
6094#	6105#	6107#	6120#	6121#	6129#	6134#	6145#	6147#	6161#	6162#	6170#	6174#
6188#	6189#	6200#	6201#	6209#	6213#	6222#	6224#	6231#	6232#	6245#	6261#	6269#
6275#	6290#	6293#	6310#	6311#	6319#	6324#	6339#	6342#	6359#	6360#	6368#	6373#
6278#	6391#	6408#	6409#	6417#	6422#	6437#	6440#	6457#	6458#	6466#	6471#	6490#
6443#	6514#	6515#	6523#	6528#	6549#	6552#	6575#	6576#	6584#	6593#	6604#	6607#
6624#	6625#	6633#	6638#	6653#	6656#	6673#	6674#	6682#	6687#	6702#	6705#	6722#
6723#	6731#	6736#	6751#	6754#	6771#	6772#	6780#	6785#	6800#	6803#	6820#	6821#
6829#	6834#	6849#	6852#	6869#	6870#	6878#	6883#	6898#	6901#	6918#	6919#	6927#
6932#	6947#	6950#	6967#	6968#	6976#	6981#	7005#	7008#	7024#	7025#	7033#	7038#
7053#	7056#	7092#	7093#	7101#	7106#	7107#	7128#	7130#	7154#	7159#	7160#	7168#
7173#	7198#	7201#	7219#	7220#	7227#	7246#	7253#	7271#	7279#	7308#	7316#	7346#
7353#	7381#	7388#	7415#	7422#	7448#	7456#	7498#	7510#	7542#	7551#	7587#	7595#
7619#	7627#	7672#	7683#	7690#	7706#	7707#	7720#	7729#	7751#	7756#	7766#	7775#
7797#	7802#	7812#	7821#	7843#	7848#	7858#	7867#	7889#	7894#	7904#	7913#	7935#
7940#	7950#	7959#	7981#	7986#	7996#	8005#	8027#	8032#	8042#	8051#	8073#	8078#
8088#	8097#	8119#	8125#	8136#	8145#	8167#	8172#	8183#	8193#	8215#	8221#	8232#
8241#	8263#	8268#	8279#	8288#	8310#	8315#	8326#	8335#	8357#	8362#	8373#	8382#
8404#	8409#	8421#	8430#	8452#	8457#	8468#	8477#	8499#	8504#	8517#	8526#	8548#
8553#	8564#	8573#	8595#	8600#	8611#	8620#	8642#	8647#	8658#	8667#	8689#	8694#
8705#	8714#	8736#	8741#	8752#	8761#	8783#	8788#	8799#	8808#	8830#	8835#	8846#
8855#	8877#	8882#	8893#	8902#	8924#	8929#	8941#	8950#	8972#	8977#	8988#	8997#
9019#	9024#	9035#	9044#	9066#	9071#	9082#	9091#	9113#	9118#	9128#	9138#	9160#
9165#	9176#	9185#	9207#	9212#	9236#	9247#	9279#	9282#	9290#			

T\$NS0 = 000000  
T\$NS1 = 000005

4813#	9290											
4857#	4861	4894#	4909	4923#	4926	5556#	5557#	5558#	5559#	5560#	5561#	5562#
5563#	5564#	5565#	5566#	5568#	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#
5578#	5579#	5580#	5581#	5582#	5583#	5584#	5585#	5586#	5587#	5588#	5589#	5591#
5594	5607#	5624	5638#	5753	5756#	5774	5784#	5787	5801#	5804	5819#	5820
5838#	5857	5863#	5871	5891#	5923	5931#	5953	5968#	5989	5997#	6017	6025#
6045	6052#	6075	6089#	6121	6129#	6162	6170#	6201	6209#	6232	6245#	6261
6269#	6311	6319#	6360	6368#	6409	6417#	6458	6466#	6515	6523#	6576	6584#
6625	6633#	6674	6682#	6723	6731#	6772	6780#	6821	6829#	6870	6878#	6919
6927#	6968	6976#	7025	7033#	7093	7101#	7160	7168#	7220	7227#	7246	7253#
7271	7279#	7308	7316#	7346	7353#	7381	7388#	7415	7422#	7448	7456#	7498
7510#	7542	7551#	7587	7595#	7619	7627#	7672	7683#	7707	7720#	7756	7766#



TSSEK0= 010000

	7797	7821#	7838	7843	7867#	7884	7889	7913#	7930	7935	7959#	7976	7981
	8005#	8022	8027	8051#	8068	8073	8097#	8114	8119	8145#	8162	8167	8193#
	8210	8215	8241#	8258	8263	8288#	8305	8310	8335#	8352	8357	8382#	8399
	8404	8430#	8447	8452	8477#	8494	8499	8526#	8543	8548	8573#	8590	8595
	8620#	8637	8642	8667#	8684	8689	8714#	8731	8736	8761#	8778	8783	8808#
	8825	8830	8855#	8872	8877	8902#	8919	8924	8950#	8967	8972	8997#	9014
	9019	9044#	9061	9066	9091#	9108	9113	9138#	9155	9160	9185#	9202	9207
T\$SUBN= C00000	4807#	5838#	5863#	5891#	5931#	5968#	5997#	6025#	6052#	6089#	6129#	6170#	6209#
	6245#	6269#	6319#	6368#	6417#	6466#	6523#	6584#	6633#	6682#	6731#	6780#	6829#
	6878#	6927#	6976#	7033#	7101#	7106#	7168#	7227#	7253#	7279#	7316#	7353#	7388#
	7422#	7456#	7510#	7551#	7595#	7627#	7683#	7720#	7766#	7812#	7858#	7904#	7950#
	7996#	8042#	8088#	8136#	8183#	8232#	8279#	8326#	8373#	8421#	8468#	8517#	8564#
	8611#	8658#	8705#	8752#	8799#	8846#	8893#	8941#	8988#	9035#	9082#	9128#	9176#
T\$TAGL= 177777	4807#	4857#	4894#	4923#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#
T\$TAGN= 010171	5565#	5566#	5568#	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#
	5580#	5581#	5582#	5583#	5584#	5585#	5586#	5587#	5588#	5589#	5591#	5607#	5638#
	5756#	5784#	5801#	5819#	5838#	5863#	5891#	5931#	5968#	5997#	6025#	6052#	6089#
	6129#	6170#	6209#	6245#	6269#	6319#	6368#	6417#	6466#	6523#	6584#	6633#	6682#
	6731#	6780#	6829#	6878#	6927#	6976#	7033#	7101#	7106#	7168#	7227#	7253#	7279#
	7316#	7353#	7388#	7422#	7456#	7510#	7551#	7595#	7627#	7683#	7720#	7766#	7812#
	7858#	7904#	7950#	7996#	8042#	8088#	8136#	8183#	8232#	8279#	8326#	8373#	8421#
	8468#	8517#	8564#	8611#	8658#	8705#	8752#	8799#	8846#	8893#	8941#	8988#	9035#
T\$TEMP= 000000	9082#	9128#	9176#	9236#	9279#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#
	4861#	4871#	4909#	4926#	5556#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#
	5565#	5566#	5568#	5570#	5571#	5585#	5586#	5587#	5588#	5589#	5594#	5615#	5624#
	5580#	5581#	5582#	5583#	5584#	5845#	5856#	5857#	5871#	5903#	5908#	5917#	5922#
	5753#	5774#	5787#	5804#	5820#	5845#	5856#	5857#	5871#	5903#	5908#	5917#	5922#
	5923#	5943#	5949#	5952#	5953#	5978#	5979#	5986#	5987#	5989#	6007#	6008#	6015#
	6017#	6035#	6036#	6043#	6045#	6065#	6071#	6074#	6075#	6101#	6105#	6115#	6120#
	6121#	6141#	6145#	6156#	6161#	6162#	6181#	6188#	6195#	6200#	6201#	6221#	6222#
	6230#	6231#	6232#	6259#	6261#	6286#	6290#	6305#	6310#	6311#	6335#	6339#	6354#
	6359#	6360#	6384#	6388#	6403#	6408#	6409#	6433#	6437#	6452#	6457#	6458#	6485#
	6490#	6508#	6514#	6515#	6544#	6549#	6569#	6575#	6576#	6600#	6604#	6619#	6624#
	6625#	6649#	6653#	6668#	6673#	6674#	6698#	6702#	6717#	6722#	6723#	6747#	6751#
	6766#	6771#	6772#	6796#	6800#	6815#	6820#	6821#	6845#	6849#	6864#	6869#	6870#
	6894#	6898#	6913#	6918#	6919#	6943#	6947#	6962#	6967#	6968#	7000#	7005#	7018#
	7024#	7025#	7049#	7053#	7069#	7092#	7093#	7124#	7128#	7149#	7154#	7159#	7160#
	7193#	7198#	7214#	7219#	7220#	7246#	7271#	7308#	7346#	7377#	7378#	7381#	7411#
	7412#	7415#	7444#	7445#	7448#	7498#	7542#	7587#	7619#	7654#	7668#	7672#	7705#
	7706#	7707#	7746#	7751#	7752#	7756#	7792#	7797#	7798#	7802#	7838#	7843#	7844#
	7848#	7884#	7889#	7890#	7894#	7930#	7935#	7936#	7940#	7976#	7981#	7982#	7986#
	8022#	8027#	8028#	8032#	8068#	8073#	8074#	8078#	8114#	8119#	8120#	8125#	8162#
	8167#	8168#	8172#	8210#	8215#	8216#	8221#	8258#	8263#	8264#	8268#	8305#	8310#
	8311#	8315#	8352#	8357#	8358#	8362#	8399#	8404#	8405#	8409#	8447#	8452#	8453#
	8457#	8494#	8499#	8500#	8504#	8543#	8548#	8549#	8553#	8590#	8595#	8596#	8600#
	8637#	8642#	8643#	8647#	8684#	8689#	8690#	8694#	8731#	8736#	8737#	8741#	8778#
	8783#	8784#	8788#	8825#	8830#	8831#	8835#	8872#	8877#	8878#	8882#	8919#	8924#
	8925#	8929#	8967#	8972#	8973#	8977#	9014#	9019#	9020#	9024#	9061#	9066#	9067#
	9071#	9108#	9113#	9114#	9118#	9155#	9160#	9161#	9165#	9202#	9207#	9208#	9212#
	9238#	9239#	9240#	9241#	9247#	9282#	9290#						
T\$TEST= 000114	4807#	5833	5836	5838#	5859	5861	5863#	5873	5889	5891#	5925	5929	5931#
	5962	5966	5968#	5991	5995	5997#	6019	6023	6025#	6047	6050	6052#	6082
	6086	6089#	6123	6127	6129#	6164	6168	6170#	6203	6207	6209#	6237	6243
	6245#	6263	6267	6269#	6313	6317	6319#	6362	6366	6368#	6411	6415	6417#
	6460	6464	6466#	6517	6521	6523#	6578	6582	6584#	6627	6631	6633#	6676

TSTSTM= 177777

6680	6682#	6725	6729	6731#	6774	6778	6780#	6823	6827	6829#	6872	6876
6878#	6921	6925	6927#	6970	6974	6976#	7027	7031	7033#	7095	7099	7101#
7106	7162	7166	7168#	7222	7225	7227#	7248	7251	7253#	7273	7277	7279#
7310	7314	7316#	7348	7351	7353#	7383	7386	7388#	7417	7420	7422#	7450
7454	7456#	7500	7508	7510#	7544	7549	7551#	7589	7593	7595#	7621	7625
7627#	7678	7681	7683#	7712	7718	7720#	7758	7764	7766#	7804	7810	7812#
7850	7856	7858#	7896	7902	7904#	7942	7948	7950#	7988	7994	7996#	8034
8040	8042#	8080	8086	8088#	8128	8134	8136#	8175	8181	8183#	8224	8230
8232#	8271	8277	8279#	8318	8324	8326#	8365	8371	8373#	8412	8419	8421#
8460	8466	8468#	8507	8513	8517#	8556	8562	8564#	8603	8609	8611#	8650
8656	8658#	8697	8704	8705#	8744	8750	8752#	8791	8797	8799#	8838	8844
8846#	8885	8891	8893#	8932	8938	8941#	8980	8986	8988#	9027	9033	9035#
9074	9080	9082#	9120	9126	9128#	9168	9174	9176#	9298	9299		
4807#	5383	5556	5557	5558	5559	5560	5561	5562	5563	5564	5565	5566
5568	5570	5571	5572	5573	5574	5575	5576	5577	5578	5579	5580	5581
5582	5583	5584	5585	5586	5587	5588	5589	5592	5593	5594	5624	5652
5655	5658	5662	5675	5753	5770	5774	5785	5787	5803	5804	5820	5845
5852	5856	5857	5869	5871	5896	5902	5903	5908	5911	5916	5917	5922
5923	5935	5942	5943	5948	5949	5952	5953	5970	5977	5978	5979	5980
5985	5986	5987	5989	5999	6006	6007	6008	6009	6014	6015	6017	6027
6034	6035	6036	6037	6042	6043	6045	6056	6064	6065	6070	6071	6074
6075	6094	6100	6101	6105	6107	6114	6115	6120	6121	6134	6140	6141
6145	6147	6155	6156	6161	6162	6174	6180	6181	6188	6189	6194	6195
6200	6201	6213	6220	6221	6222	6224	6229	6230	6231	6232	6257	6259
6261	6275	6285	6286	6290	6293	6304	6305	6310	6311	6324	6334	6335
6339	6342	6353	6354	6359	6360	6373	6383	6384	6388	6391	6402	6403
6408	6409	6422	6432	6433	6437	6440	6451	6452	6457	6458	6471	6484
6485	6490	6493	6507	6508	6514	6515	6528	6542	6544	6549	6552	6568
6569	6575	6576	6589	6599	6600	6604	6607	6618	6619	6624	6625	6638
6648	6649	6653	6656	6667	6668	6673	6674	6687	6697	6698	6702	6705
6716	6717	6722	6723	6736	6746	6747	6751	6754	6765	6766	6771	6772
6785	6795	6796	6800	6803	6814	6815	6820	6821	6834	6844	6845	6849
6852	6863	6864	6869	6870	6883	6893	6894	6898	6901	6912	6913	6918
6919	6932	6942	6943	6947	6950	6961	6962	6967	6968	6981	6999	7000
7005	7008	7017	7018	7024	7025	7038	7048	7049	7053	7056	7068	7069
7084	7085	7092	7093	7106	7107	7123	7124	7128	7130	7148	7149	7154
7159	7160	7173	7192	7193	7198	7201	7213	7214	7219	7220	7229	7235
7239	7241	7243	7246	7260	7264	7266	7268	7271	7283	7298	7304	7305
7308	7320	7336	7338	7342	7346	7354	7376	7377	7378	7381	7410	7411
7412	7415	7443	7444	7445	7448	7496	7498	7537	7542	7574	7584	7587
7617	7619	7653	7654	7668	7672	7690	7704	7705	7706	7707	7729	7745
7746	7751	7752	7756	7775	7791	7792	7797	7798	7802	7821	7837	7838
7843	7844	7848	7867	7883	7884	7889	7890	7894	7913	7929	7930	7935
7936	7940	7959	7975	7976	7981	7982	7986	8005	8021	8022	8027	8028
8032	8051	8067	8068	8073	8074	8078	8097	8113	8114	8119	8120	8125
8145	8161	8162	8167	8168	8172	8193	8209	8210	8215	8216	8221	8241
8257	8258	8263	8264	8268	8288	8304	8305	8310	8311	8315	8335	8351
8352	8357	8358	8362	8382	8398	8399	8404	8405	8409	8430	8446	8447
8452	8453	8457	8477	8493	8494	8499	8500	8504	8526	8542	8543	8548
8549	8553	8573	8589	8590	8595	8596	8600	8620	8636	8637	8642	8643
8647	8667	8683	8684	8689	8690	8694	8714	8730	8731	8736	8737	8741
8761	8777	8778	8783	8784	8788	8808	8824	8825	8830	8831	8835	8855
8871	8872	8877	8878	8882	8902	8918	8919	8924	8925	8929	8950	8966
8967	8972	8973	8977	8997	9013	9014	9019	9020	9024	9044	9060	9061
9066	9067	9071	9091	9107	9108	9113	9114	9118	9138	9154	9155	9160
9161	9165	9185	9201	9202	9207	9208	9212					

CZDMPAO M8207 STATIC DIAG #1 MACY11 30A(1052) 17-JUL-79 14:39 PAGE 60-19  
 CZDMPA.P11 17-JUL-79 14:33 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0222

T\$TSTS= 000001	4807#	5838#	5863#	5891#	5931#	5968#	5997#	6025#	6052#	6089#	6129#	6170#	6209#
	6245#	6269#	6319#	6368#	6417#	6466#	6523#	6584#	6633#	6682#	6731#	6780#	6829#
	6878#	6927#	6976#	7033#	7101#	7168#	7227#	7253#	7279#	7316#	7353#	7388#	7422#
	7456#	7510#	7551#	7595#	7627#	7683#	7720#	7766#	7812#	7858#	7904#	7950#	7996#
	8042#	8088#	8136#	8183#	8232#	8279#	8326#	8373#	8421#	8468#	8517#	8564#	8611#
	8658#	8705#	8752#	8799#	8846#	8893#	8941#	8988#	9035#	9082#	9128#	9176#	
T\$SAU = 010051	5819#	5820											
T\$SAUT= 010046	5756#	5774											
T\$SCLE= 010047	5784#	5787											
T\$SDU = 010050	5801#	5804											
T\$SHAR= 010167	9236#	9247											
T\$SHW = 010001	4894#	4909											
T\$SINI= 010045	5638#	5753											
T\$MSG= 010043	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#	5566#	5568#	5570#
	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#	5583#
	5584#	5585#	5586#	5587#	5588#	5589#	5591#	5594					
T\$PRO= 010000	4857#												
T\$RPT= 010044	5607#	5615	5624										
T\$SEG= 010000	5896#	5903	5908#	5911#	5917	5922#	5935#	5943	5949	5952#	5970#	5978	5979#
	5980#	5986	5987#	5999#	6007	6008#	6009#	6015#	6027#	6035	6036#	6037#	6043#
	6056#	6065	6071	6074#	6094#	6101	6105#	6107#	6115	6120#	6134#	6141	6145#
	6147#	6156	6161#	6174#	6181	6188#	6189#	6195	6200#	6213#	6221	6222#	6224#
	6230	6231#	6275#	6286	6290#	6293#	6305	6310#	6324#	6335	6339#	6342#	6354
	6359#	6373#	6384	6388#	6391#	6403	6408#	6422#	6433	6437#	6440#	6452	6457#
	6471#	6485	6490#	6493#	6508	6514#	6528#	6544	6549#	6552#	6569	6575#	6589#
	6600	6604#	6607#	6619	6624#	6638#	6649	6653#	6656#	6668	6673#	6687#	6698
	6702#	6705#	6717	6722#	6736#	6747	6751#	6754#	6766	6771#	6785#	6796	6800#
	6803#	6815	6820#	6834#	6845	6849#	6852#	6864	6869#	6883#	6894	6898#	6901#
	6913	6918#	6932#	6943	6947#	6950#	6962	6967#	6981#	7000	7005#	7008#	7018
	7024#	7038#	7049	7053#	7056#	7069	7092#	7107#	7124	7128#	7130#	7154#	7173#
	7193	7198#	7201#	7214	7219#	7290#	7705	7706#	7729#	7746	7751#	7775#	7792
	7797#	7821#	7838	7843#	7867#	7884	7889#	7913#	7930	7935#	7959#	7976	7981#
	8005#	8022	8027#	8051#	8068	8073#	8097#	8114	8119#	8145#	8162	8167#	8193#
	8210	8215#	8241#	8258	8263#	8288#	8305	8310#	8335#	8352	8357#	8382#	8399
	8404#	8430#	8447	8452#	8477#	8494	8499#	8526#	8543	8548#	8573#	8590	8595#
	8620#	8637	8642#	8667#	8684	8689#	8714#	8731	8736#	8761#	8778	8783#	8808#
	8825	8830#	8855#	8872	8877#	8902#	8919	8924#	8950#	8967	8972#	8997#	9014
	9019#	9044#	9061	9066#	9091#	9108	9113#	9138#	9155	9160#	9185#	9202	9207#
T\$SOF= 010170	9279#	9282											
T\$SUB= 010110	7106#	7159											
T\$SW = 010002	4923#	4926											
T\$TES= 010166	5838#	5845	5856	5857	5863#	5871	5891#	5923	5931#	5953	5968#	5989	5997#
	6017	6025#	6045	6052#	6075	6089#	6121	6129#	6162	6170#	6201	6209#	6232
	6245#	6259	6261	6269#	6311	6319#	6360	6368#	6409	6417#	6458	6466#	6515
	6523#	6576	6584#	6625	6633#	6674	6682#	6723	6731#	6772	6780#	6821	6829#
	6870	6878#	6919	6927#	6968	6976#	7025	7033#	7093	7101#	7149	7160	7168#
	7220	7227#	7246	7253#	7271	7279#	7308	7316#	7346	7353#	7377	7378	7381
	7388#	7411	7412	7415	7422#	7444	7445	7448	7456#	7498	7510#	7542	7551#
	7587	7595#	7619	7627#	7654	7668	7672	7683#	7707	7720#	7752	7756	7766#
	7798	7802	7812#	7844	7848	7858#	7890	7894	7904#	7936	7940	7950#	7982
	7986	7996#	8028	8032	8042#	8074	8078	8088#	8120	8125	8136#	8168	8172
	8183#	8216	8221	8232#	8264	8268	8279#	8311	8315	8326#	8358	8362	8373#
	8405	8409	8421#	8453	8457	8468#	8500	8504	8517#	8549	8553	8564#	8596
	8600	8611#	8643	8647	8658#	8690	8694	8705#	8737	8741	8752#	8784	8788
	8799#	8831	8835	8846#	8878	8882	8893#	8925	8929	8941#	8973	8977	8988#
	9020	9024	9035#	9067	9071	9082#	9114	9118	9128#	9161	9165	9176#	9208

CZDMPAO M8207 STATIC DIAG #1  
CZDMPA.P11 17-JUL-79 14:33

MACY11 30A(1052) 17-JUL-79 14:39 PAGE 60-20  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0223

		9212	
T1	011352 G	4871	5838#
T10	013040 G	4871	6129#
T11	013224 G	4871	6170#
T12	013412 G	4871	6209#
T13	013562 G	4871	6245#
T14	013670 G	4871	6269#
T15	014120 G	4871	6319#
T16	014350 G	4871	6368#
T17	014600 G	4871	6417#
T18	015030 G	4871	6466#
T19	015314 G	4871	6523#
T2	011500 G	4871	5863#
T20	015610 G	4871	6584#
T21	016040 G	4871	6633#
T22	016270 G	4871	6682#
T23	016520 G	4871	6731#
T24	016750 G	4871	6780#
T25	017200 G	4871	6829#
T26	017430 G	4871	6878#
T27	017660 G	4871	6927#
T28	020110 G	4871	6976#
T29	020406 G	4871	7037#
T3	011544 G	4871	5891#
T30	020726 G	4871	7101#
T30.1	020744	7106#	
T31	021272 G	4871	7168#
T32	021604 G	4871	7227#
T33	021746 G	4871	7253#
T34	022110 G	4871	7279#
T35	022264 G	4871	7316#
T36	022470 G	4871	7353#
T37	022634 G	4871	7388#
T38	023004 G	4871	7422#
T39	023150 G	4871	7456#
T4	011732 G	4871	5931#
T40	023336 G	4871	7510#
T41	023552 G	4871	7551#
T42	023770 G	4871	7595#
T43	024122 G	4871	7627#
T44	024340 G	4871	7683#
T45	024500 G	4871	7720#
T46	024704 G	4871	7766#
T47	025110 G	4871	7812#
T48	025314 G	4871	7858#
T49	025520 G	4871	7904#
T5	012076 G	4871	5968#
T50	025724 G	4871	7950#
T51	026130 G	4871	7996#
T52	026334 G	4871	8042#
T53	026540 G	4871	8088#
T54	026746 G	4871	8136#
T55	027152 G	4871	8183#
T56	027356 G	4871	8232#
T57	027562 G	4871	8279#
T58	027766 G	4871	8326#

[illegible]

	7123*	7144*	7146	7148*	7188*	7190	7192*	7209*	7211	7213*	7241*	7243*	7266*
	7268*	7290*	7305*	7338*	7342*	7372*	7374	7402*	7408	7439*	7441	7474*	7494
	7533*	7534*	7535	7569*	7572	7580*	7612*	7615	7649*	7651	7700*	7702	7741*
	7743	7787*	7789	7833*	7835	7879*	7881	7925*	7927	7971*	7973	8017*	8019
	8063*	8065	8109*	8111	8157*	8159	8205*	8207	8253*	8255	8300*	8302	8347*
	8349	8394*	8396	8442*	8444	8489*	8491	8538*	8540	8585*	8587	8632*	8634
	8679*	8681	8726*	8728	8773*	8775	8820*	8822	8867*	8869	8914*	8916	8962*
	8964	9009*	9011	9056*	9058	9103*	9105	9150*	9152	9197*	9199		
\$LSTIN= 000000	4816#												
\$LSTTA- 000000	4817#												
\$MPO 002550	5002#	5357*	5360	5375									
- 040004	4799#	4995#	4997#	5019#	5020#	5021#	5022#	5049#	5095#	5212#	5615	5845	5856
	5903	5917	5943	5949	5978	5986	6007	6035	6065	6071	6101	6115	6141
	6156	6181	6195	6221	6230	6259	6286	6305	6335	6354	6384	6403	6433
	6452	6485	6508	6544	6569	6600	6619	6649	6668	6698	6717	6747	6766
	6796	6815	6845	6864	6894	6913	6943	6962	7000	7018	7049	7069	7124
	7149	7193	7214	7377	7378	7411	7412	7444	7445	7637	7654	7668	7705
	7746	7752	7792	7798	7838	7844	7884	7890	7930	7936	7976	7982	8022
	8028	8068	8074	8114	8120	8162	8168	8210	8216	8258	8264	8305	8311
	8352	8358	8399	8405	8447	8453	8494	8500	8543	8549	8590	8596	8637
	8643	8684	8690	8731	8737	8778	8784	8825	8831	8872	8878	8919	8925
	8967	8973	9014	9020	9061	9067	9108	9114	9155	9161	9202	9208	9296#
.MSTCL 003142	5206#	5894	5932	5969	5998	6026	6053	6091	6131	6172	6211	6247	6270
	6320	6369	6418	6467	6524	6585	6634	6683	6732	6781	6830	6879	6928
	6977	7035	7103	7169	7255	7281	7307	7318	7345	7390	7424	7457	7512
	7553	7597	7629	7685	7722	7768	7814	7860	7906	7952	7998	8044	8090
	8138	8186	8234	8281	8328	8375	8423	8470	8519	8566	8613	8660	8707
	8754	8801	8848	8895	8943	8990	9037	9084	9131	9178			
.ROMCL 003230	5221#	5230	5232	5234	5240	5247	5254	5261	5269	5283	5344	5362	5364
	5367	5372	5376	5403	5414	5416	5424	5426	6278	6280	6297	6299	6327
	6329	6346	6348	6376	6378	6395	6397	6425	6427	6444	6446	6475	6477
	6498	6500	6532	6534	6557	6559	6592	6594	6611	6613	6641	6643	6660
	6662	6690	6692	6709	6711	6739	6741	6758	6760	6788	6790	6807	6809
	6837	6839	6856	6858	6886	6888	6905	6907	6935	6937	6954	6956	6986
	6994	7012	7041	7043	7060	7062	7079	7111	7115	7117	7136	7140	7142
	7178	7183	7186	7205	7207	7237	7262	7296	7334	7366	7369	7399	7403
	7405	7433	7436	7467	7489	7491	7528	7530	7561	7564	7567	7578	7604
	7607	7610	7646	7692	7694	7696	7698	7733	7737	7739	7779	7783	7785
	7825	7829	7831	7871	7875	7877	7917	7921	7923	7963	7967	7969	8009
	8013	8015	8055	8059	8061	8101	8105	8107	8149	8153	8155	8197	8201
	8203	8245	8249	8251	8292	8296	8298	8339	8343	8345	8386	8390	8392
	8434	8438	8440	8481	8485	8487	8530	8534	8536	8577	8581	8583	8624
	8628	8630	8671	8675	8677	8718	8722	8724	8765	8769	8771	8812	8816
	8818	8859	8863	8865	8906	8910	8912	8954	8958	8960	9001	9005	9007
	9048	9052	9054	9095	9099	9101	9142	9146	9148	9189	9193	9195	



BADHEA	5174#	5833	5836	5859	5861	5873	5889	5925	5929	5962	5966	5991	5995	6019	6023
	6047	6050	6082	6086	6123	6127	6164	6168	6203	6207	6237	6243	6263	6267	6313
	6317	6362	6366	6411	6415	6460	6464	6517	6521	6578	6582	6627	6631	6676	6680
	6725	6729	6774	6778	6823	6827	6872	6876	6921	6925	6970	6974	7027	7031	7095
	7099	7162	7166	7222	7225	7248	7251	7273	7277	7310	7314	7348	7351	7383	7386
	7417	7420	7450	7454	7500	7508	7544	7549	7589	7593	7621	7625	7678	7681	7712
	7718	7758	7764	7804	7810	7850	7856	7896	7902	7942	7948	7988	7994	8034	8040
	8080	8086	8128	8134	8175	8181	8224	8230	8271	8277	8318	8324	8365	8371	8412
	8419	8460	8466	8507	8513	8556	8562	8603	8609	8650	8656	8697	8704	8744	8750
	8791	8797	8838	8844	8885	8891	8932	8938	8980	8986	9027	9033	9074	9080	9120
	9126	9168	9174												
BCOMPL	102#	4807#	5653	5656	5659										
BERROR	110#	4807#	5162#	6285	6304	6334	6353	6383	6402	6432	6451	6484	6507	6542	6568
	6599	6618	6648	6667	6697	6716	6746	6765	6795	6814	6844	6863	6893	6912	6942
	6961	6999	7017	7048	7068	7084									
BGNAU	118#	4807#	5819												
BGNAUT	132#	4807#	5756												
BGNCLN	146#	4807#	5784												
BGNDU	159#	4807#	5801												
BGNHRD	172#	4807#	9236												
BGNHW	187#	4807#	4894												
BGNINI	203#	4807#	5638												
BGNMOD	215#	4807#	4813												
BGNMSG	232#	4807#	5556	5557	5558	5559	5560	5561	5562	5563	5564	5565	5566	5568	5570
	5571	5572	5573	5574	5575	5576	5577	5578	5579	5580	5581	5582	5583	5584	5585
	5586	5587	5588	5589	5591										
BGNPRO	245#	4807#	4857												
BGNPTA	259#	4807#													
BGNRPT	297#	4807#	5607												
BGNSEG	311#	4807#	5896	5911	5935	5970	5980	5999	6009	6027	6037	6056	6094	6107	6134
	6147	6174	6189	6213	6224	6275	6293	6324	6342	6373	6391	6422	6440	6471	6493
	6528	6552	6589	6607	6638	6656	6687	6705	6736	6754	6785	6803	6834	6852	6883
	6901	6932	6950	6981	7008	7038	7056	7107	7130	7173	7201	7690	7729	7775	7821
	7867	7913	7959	8005	8051	8097	8145	8193	8241	8288	8335	8382	8430	8477	8526
	8573	8620	8667	8714	8761	8808	8855	8902	8950	8997	9044	9091	9138	9185	
BGNSET	325#	4807#													
BGNSFT	352#	4807#	9279												
BGNSRV	367#	4807#													
BGNSUB	380#	4807#	7106												
BGNSW	408#	4807#	4923												
BGNTST	423#	4807#	5838	5863	5891	5931	5968	5997	6025	6052	6089	6129	6170	6209	6245
	6269	6319	6368	6417	6466	6523	6584	6633	6682	6731	6780	6829	6878	6927	6976
	7033	7101	7168	7227	7253	7279	7316	7353	7388	7422	7456	7510	7551	7595	7627
	7683	7720	7766	7812	7858	7904	7950	7996	8042	8088	8136	8183	8232	8279	8326
	8373	8421	8468	8517	8564	8611	8658	8705	8752	8799	8846	8893	8941	8988	9035
	9082	9128	9176												
BNCOMP	457#	4807#	5663	5676											
BNERRO	465#	4807#													
BREAK	473#	4807#													
BRESET	481#	4807#	5785	5803	7229	7304	7354								
CKLOOP	491#	4807#	7085												
CLOCK	503#	4807#													
CLOSE	514#	4807#													
CLRVEC	528#	4807#													
COMMEN	537#	4807#													
DELAY	568#	4807#													



DESCRI	558#	4807#	4983												
DEVYTP	588#	4807#	5116												
DISPAT	597#	4807#	4871												
DISPLA	615#	4807#													
DOCLN	635#	4807#													
DODU	643#	4807#	5770												
DORPT	651#	4807#													
ED\$CAL	5169#	5833	5836	5859	5861	5873	5889	5925	5929	5962	5966	5991	5995	6019	6023
	6047	6050	6082	6086	6123	6127	6164	6168	6203	6207	6237	6243	6263	6267	6313
	6317	6362	6366	6411	6415	6460	6464	6517	6521	6578	6582	6627	6631	6676	6680
	6725	6729	6774	6778	6823	6827	6872	6876	6921	6925	6970	6974	7027	7031	7095
	7099	7162	7166	7222	7225	7248	7251	7273	7277	7310	7314	7348	7351	7383	7386
	7417	7420	7450	7454	7500	7508	7544	7549	7589	7593	7621	7625	7678	7681	7712
	7718	7758	7764	7804	7810	7850	7856	7896	7902	7942	7948	7988	7994	8034	8040
	8080	8086	8128	8134	8175	8181	8224	8230	8271	8277	8318	8324	8365	8371	8412
	8419	8460	8466	8507	8513	8556	8562	8603	8609	8650	8656	8697	8704	8744	8750
	8791	8797	8838	8844	8885	8891	8932	8938	8980	8986	9027	9033	9074	9080	9120
	9126	9168	9174												
ENDAU	659#	4807#	5820												
ENDAUT	675#	4807#	5774												
ENDCLN	692#	4807#	5787												
ENDCOM	708#	4807#													
ENDDU	730#	4807#	5804												
ENDHRD	749#	4807#	9247												
ENDHW	765#	4807#	4909												
ENDINI	779#	4807#	5753												
ENDMOD	796#	4807#	9290												
ENDMSG	813#	4807#	5556	5557	5558	5559	5560	5561	5562	5563	5564	5565	5566	5568	5570
	5571	5572	5573	5574	5575	5576	5577	5578	5579	5580	5581	5582	5583	5584	5585
	5586	5587	5588	5589	5594										
ENDPRO	830#	4807#	4861												
ENDPTA	844#	4807#													
FNRPT	857#	4807#	5624												
ENDSEG	876#	4807#	5908	5922	5952	5979	5987	6008	6015	6036	6043	6074	6105	6120	6145
	6161	6188	6200	6222	6231	6290	6310	6339	6359	6388	6408	6437	6457	6490	6514
	6549	6575	6604	6624	6653	6673	6702	6722	6751	6771	6800	6820	6849	6869	6898
	6918	6947	6967	7005	7024	7053	7092	7128	7154	7198	7219	7706	7751	7797	7843
	7889	7935	7981	8027	8073	8119	8167	8215	8263	8310	8357	8404	8452	8499	8548
	8595	8642	8689	8736	8783	8830	8877	8924	8972	9019	9066	9113	9160	9207	
ENDSET	895#	4807#													
ENDSFT	915#	4807#	9282												
ENDSRV	932#	4807#													
ENDSUB	952#	4807#	7159												
ENDSW	974#	4807#	4926												
ENDTST	988#	4807#	5857	5871	5923	5953	5989	6017	6045	6075	6121	6162	6201	6232	6261
	6311	6360	6409	6458	6515	6576	6625	6674	6723	6772	6821	6870	6919	6968	7025
	7093	7160	7220	7246	7271	7308	7346	7381	7415	7448	7498	7542	7587	7619	7672
	7707	7756	7802	7848	7894	7940	7986	8032	8078	8125	8172	8221	8268	8315	8362
	8409	8457	8504	8553	8600	8647	8694	8741	8788	8835	8882	8929	8977	9024	9071
	9118	9165	9212												
EQUALS	1009#	4807#	4953												
ERRDF	1087#	4807#	5383	5852	5869	5902	5916	5942	5948	5977	5985	6006	6014	6034	6042
	6064	6070	6100	6114	6140	6155	6180	6194	6220	6229	6257	6285	6304	6334	6353
	6383	6402	6432	6451	6484	6507	6542	6568	6599	6618	6648	6667	6697	6716	6746
	6765	6795	6814	6844	6863	6893	6912	6942	6961	6999	7017	7048	7068	7084	7123
	7148	7192	7213	7241	7243	7266	7268	7305	7338	7342	7376	7410	7443	7496	7537

	7574	7584	7617	7653	7704	7745	7791	7837	7883	7929	7975	8021	8067	8113	8161
	8209	8257	8304	8351	8398	8446	8493	8542	8589	8636	8683	8730	8777	8824	8871
	8918	8966	9013	9060	9107	9154	9201								
ERRHRD	1099#	4807#													
ERROR	1109#	4807#	5147#	5383	5852	5869	5902	5916	5942	5948	5977	5985	6006	6014	6034
	6042	6064	6070	6100	6114	6140	6155	6180	6194	6220	6229	6257	7241	7243	7266
	7268	7305	7338	7342	7376	7410	7443	7496	7537	7574	7584	7617	7653	7704	7745
	7791	7837	7883	7929	7975	8021	8067	8113	8161	8209	8257	8304	8351	8398	8446
	8493	8542	8589	8636	8683	8730	8777	8824	8871	8918	8966	9013	9060	9107	9154
	9201														
ERRSF	1118#	4807#													
ERRSOF	1130#	4807#													
ERRTBL	1140#	4807#													
ESCAPE	1156#	4807#	5845	5856	5903	5917	5943	5949	5978	5986	6007	6035	6065	6071	6101
	6115	6141	6156	6181	6195	6221	6230	6259	6286	6305	6335	6354	6384	6403	6433
	6452	6485	6508	6544	6569	6600	6619	6649	6668	6698	6717	6747	6766	6796	6815
	6845	6864	6894	6913	6943	6962	7000	7018	7049	7069	7124	7149	7193	7214	7377
	7411	7444	7654	7705	7746	7792	7838	7884	7930	7976	8022	8068	8114	8162	8210
	8258	8305	8352	8399	8447	8494	8543	8590	8637	8684	8731	8778	8825	8872	8919
	8967	9014	9061	9108	9155	9202									
EXIT	1186#	4807#	5615	7378	7412	7445	7668	7752	7798	7844	7890	7936	7982	8028	8074
	8120	8168	8216	8264	8311	8358	8405	8453	8500	8549	8596	8643	8690	8737	8784
	8831	8878	8925	8973	9020	9067	9114	9161	9208						
FEQUAL	1228#	4807#													
GETBYT	1246#	4807#													
GETPRI	1264#	4807#													
GETWOR	1256#	4807#													
GMANIA	1286#	4807#													
GMANID	1299#	4807#													
GMANIL	1315#	4807#													
GPHARD	1328#	4807#	5675												
GPRMA	1340#	4807#	9239	9240											
GPRMD	1372#	4807#	9238	9241											
GPRML	1407#	4807#													
HEADER	1432#	4807#	4848												
INLGOP	1446#	4807#													
IOSETU	1453#	4807#													
IOSTAR	1466#	4807#													
KT11	1488#	4807#													
K4ONLY	5179#														
LASTAD	1659#	4807#	9298												
MANUAL	1677#	4807#													
MDTO	5520#	5566	5571	5572	5574	5575	5577	5584	5585						
MDT1	5523#	5559	5560	5561	5562	5563									
MDT2	5527#	5556	5557	5558	5564	5565	5568	5573	5578	5579	5581	5583	5586	5587	5588
	5589														
MDT27	5536#	5580	5582												
MDT5	5532#	5570	5576												
MEMORY	1685#	4807#													
MSTCLR	5200#	5894	5932	5969	5998	6026	6053	6091	6131	6172	6211	6247	6270	6320	6369
	6418	6467	6524	6585	6634	6683	6732	6781	6830	6879	6928	6977	7035	7103	7169
	7255	7281	7307	7318	7345	7390	7424	7457	7512	7553	7597	7629	7685	7722	7768
	7814	7860	7906	7952	7998	8044	8090	8138	8186	8234	8281	8328	8375	8423	8470
	8519	8566	8613	8660	8707	8754	8801	8848	8895	8943	8990	9037	9084	9131	9178
MYINT	5188#	5864	5892	6171	6212	6246	6274	6323	6372	6421	6470	6527	6588	6637	6686
	6735	6784	6833	6882	6931	6980	7037	7102	7172	7228	7254	7280	7317	7356	7389

	7423	7458	7511	7552	7596	7628	7684	7721	7767	7813	7859	7905	7951	7997	8043
	8089	8137	8184	8233	8280	8327	8374	8422	8469	8518	8565	8612	8659	8706	8753
	8800	8847	8894	8942	8989	9036	9083	9130	9177						
MSBYTE	2901#	4807#	4848#												
MSCHEC	3206#	4807#	5615#	7378#	7412#	7445#	7668#	7752#	7798#	7844#	7890#	7936#	7982#	8028#	8074#
	8120#	8168#	8216#	8264#	8311#	8358#	8405#	8453#	8500#	8549#	8596#	8643#	8690#	8737#	8784#
MSCNTO	8831#	8878#	8925#	8973#	9020#	9067#	9114#	9161#	9208#						
MSCOUN	3279#	4807#	9238#	9239#	9240#	9241#									
	3124#	4807#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#	5566#	5568#	5570#
	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#	5583#	5584#	5585#
MSDATA	5586#	5587#	5588#	5589#	5592#	5593#									
MSDECR	2614#	4807#	4848#	4983#	5116#										
	3063#	4807#	4861#	4909#	4926#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#
	5566#	5568#	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#
	5583#	5584#	5585#	5586#	5587#	5588#	5589#	5594#	5624#	5753#	5774#	5787#	5804#	5820#	5857#
	5871#	5908#	5922#	5923#	5952#	5953#	5979#	5987#	5989#	6008#	6015#	6017#	6036#	6043#	6045#
	6074#	6075#	6105#	6120#	6121#	6145#	6161#	6162#	6188#	6200#	6201#	6222#	6231#	6232#	6261#
	6290#	6310#	6311#	6339#	6359#	6360#	6388#	6408#	6409#	6437#	6457#	6458#	6490#	6514#	6515#
	6549#	6575#	6576#	6604#	6624#	6625#	6653#	6673#	6674#	6702#	6722#	6723#	6751#	6771#	6772#
	6800#	6820#	6821#	6849#	6869#	6870#	6898#	6918#	6919#	6947#	6967#	6968#	7005#	7024#	7025#
	7053#	7092#	7093#	7128#	7154#	7159#	7160#	7198#	7219#	7220#	7246#	7271#	7308#	7346#	7381#
	7415#	7448#	7498#	7542#	7587#	7619#	7672#	7706#	7707#	7751#	7756#	7797#	7802#	7843#	7848#
	7889#	7894#	7935#	7940#	7981#	7986#	8027#	8032#	8073#	8078#	8119#	8125#	8167#	8172#	8215#
	8221#	8263#	8268#	8310#	8315#	8357#	8362#	8404#	8409#	8452#	8457#	8499#	8504#	8548#	8553#
	8595#	8600#	8642#	8647#	8689#	8694#	8736#	8741#	8783#	8788#	8830#	8835#	8877#	8882#	8924#
	8929#	8972#	8977#	9019#	9024#	9066#	9071#	9113#	9118#	9160#	9165#	9207#	9212#	9247#	9282#
MSDEFA	9290#														
MSENDE	3263#	4807#	9238#	9239#	9240#	9241#									
	3145#	4807#	4909#	4926#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#	5566#
	5568#	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#	5583#
	5584#	5585#	5586#	5587#	5588#	5589#	5594#	5624#	5753#	5774#	5787#	5804#	5820#	5857#	5871#
	5908#	5922#	5923#	5952#	5953#	5979#	5987#	5989#	6008#	6015#	6017#	6036#	6043#	6045#	6074#
	6075#	6105#	6120#	6121#	6145#	6161#	6162#	6188#	6200#	6201#	6222#	6231#	6232#	6261#	6290#
	6310#	6311#	6339#	6359#	6360#	6388#	6408#	6409#	6437#	6457#	6458#	6490#	6514#	6515#	6549#
	6575#	6576#	6604#	6624#	6625#	6653#	6673#	6674#	6702#	6722#	6723#	6751#	6771#	6772#	6800#
	6820#	6821#	6849#	6869#	6870#	6898#	6918#	6919#	6947#	6967#	6968#	7005#	7024#	7025#	7053#
	7092#	7093#	7128#	7154#	7159#	7160#	7198#	7219#	7220#	7246#	7271#	7308#	7346#	7381#	7415#
	7448#	7498#	7542#	7587#	7619#	7672#	7706#	7707#	7751#	7756#	7797#	7802#	7843#	7848#	7889#
	7894#	7935#	7940#	7981#	7986#	8027#	8032#	8073#	8078#	8119#	8125#	8167#	8172#	8215#	8221#
	8263#	8268#	8310#	8315#	8357#	8362#	8404#	8409#	8452#	8457#	8499#	8504#	8548#	8553#	8595#
	8600#	8642#	8647#	8689#	8694#	8736#	8741#	8783#	8788#	8830#	8835#	8877#	8882#	8924#	8929#
MSERRI	8972#	8977#	9019#	9024#	9066#	9071#	9113#	9118#	9160#	9165#	9207#	9212#	9247#	9282#	9290#
	2365#	4807#	5383#	5852#	5869#	5902#	5916#	5942#	5948#	5977#	5985#	6006#	6014#	6034#	6042#
	6064#	6070#	6100#	6114#	6140#	6155#	6180#	6194#	6220#	6229#	6257#	6285#	6304#	6334#	6353#
	6383#	6402#	6432#	6451#	6484#	6507#	6542#	6568#	6599#	6618#	6648#	6667#	6697#	6716#	6746#
	6765#	6795#	6814#	6844#	6863#	6893#	6912#	6942#	6961#	6999#	7017#	7048#	7068#	7084#	7123#
	7148#	7192#	7213#	7241#	7243#	7266#	7268#	7305#	7338#	7342#	7376#	7410#	7443#	7496#	7537#
	7574#	7584#	7617#	7653#	7704#	7745#	7791#	7837#	7883#	7929#	7975#	8021#	8067#	8113#	8161#
	8209#	8257#	8304#	8351#	8398#	8446#	8493#	8542#	8589#	8636#	8683#	8730#	8777#	8824#	8871#
MSERCA	8918#	8966#	9013#	9060#	9107#	9154#	9201#								
	2921#	4807#	5845#	5856#	5903#	5917#	5943#	5949#	5978#	5986#	6007#	6035#	6065#	6071#	6101#
	6115#	6141#	6156#	6181#	6195#	6221#	6230#	6259#	6286#	6305#	6335#	6354#	6384#	6403#	6433#
	6452#	6485#	6508#	6544#	6569#	6600#	6619#	6649#	6668#	6698#	6717#	6747#	6766#	6796#	6815#
	6845#	6864#	6894#	6913#	6943#	6962#	7000#	7018#	7049#	7069#	7124#	7149#	7193#	7214#	7377#
	7411#	7444#	7654#	7705#	7746#	7792#	7838#	7884#	7930#	7976#	8022#	8068#	8114#	8162#	8210#
	8258#	8305#	8352#	8399#	8447#	8494#	8543#	8590#	8637#	8684#	8731#	8778#	8825#	8872#	8919#

MSESCS	8967#	9014#	9061#	9108#	9155#	9202#										
	2932#	4807#	5845#	5856#	5903#	5917#	5943#	5949#	5978#	5986#	6007#	6035#	6065#	6071#	6101#	
	6115#	6141#	6156#	6181#	6195#	6221#	6230#	6259#	6286#	6305#	6335#	6354#	6384#	6403#	6433#	
	6452#	6485#	6508#	6544#	6569#	6600#	6619#	6649#	6668#	6698#	6717#	6747#	6766#	6796#	6815#	
	6845#	6864#	6894#	6913#	6943#	6962#	7000#	7018#	7049#	7069#	7124#	7149#	7193#	7214#	7377#	
	7411#	7444#	7654#	7705#	7746#	7792#	7838#	7884#	7930#	7976#	8022#	8068#	8114#	8162#	8210#	
	8258#	8305#	8352#	8399#	8447#	8494#	8543#	8590#	8637#	8684#	8731#	8778#	8825#	8872#	8919#	
MSEXCP	8967#	9014#	9061#	9108#	9155#	9202#										
	3186#	4807#	9238#	9239#	9240#	9241#										
MSEXIT	2943#	4807#	5615#	7378#	7412#	7445#	7668#	7752#	7798#	7844#	7890#	7936#	7982#	8028#	8074#	
	8120#	8168#	8216#	8264#	8311#	8358#	8405#	8453#	8500#	8549#	8596#	8643#	8690#	8737#	8784#	
	8831#	8878#	8925#	8973#	9020#	9067#	9114#	9161#	9208#							
MSEXSE	2965#	4807#	5615#	7378#	7412#	7445#	7668#	7752#	7798#	7844#	7890#	7936#	7982#	8028#	8074#	
	8120#	8168#	8216#	8264#	8311#	8358#	8405#	8453#	8500#	8549#	8596#	8643#	8690#	8737#	8784#	
	8831#	8878#	8925#	8973#	9020#	9067#	9114#	9161#	9208#							
MSEXTJ	2954#	4807#	5615#	7378#	7412#	7445#	7668#	7752#	7798#	7844#	7890#	7936#	7982#	8028#	8074#	
	8120#	8168#	8216#	8264#	8311#	8358#	8405#	8453#	8500#	8549#	8596#	8643#	8690#	8737#	8784#	
	8831#	8878#	8925#	8973#	9020#	9067#	9114#	9161#	9208#							
MSGEN	3087#	4807#	4813#	4848#	4857#	4871#	4894#	4909#	4923#	4926#	4983#	5116#	5556#	5557#	5558#	
	5559#	5560#	5561#	5562#	5563#	5564#	5565#	5566#	5568#	5570#	5571#	5572#	5573#	5574#	5575#	
	5576#	5577#	5578#	5579#	5580#	5581#	5582#	5583#	5584#	5585#	5586#	5587#	5588#	5589#	5591#	
	5594#	5607#	5624#	5638#	5753#	5756#	5774#	5784#	5787#	5801#	5804#	5819#	5820#	5838#	5857#	
	5863#	5871#	5891#	5908#	5922#	5923#	5931#	5952#	5953#	5968#	5979#	5987#	5989#	5997#	6008#	
	6015#	6017#	6025#	6036#	6043#	6045#	6052#	6074#	6075#	6089#	6105#	6120#	6121#	6129#	6145#	
	6161#	6162#	6170#	6188#	6200#	6201#	6209#	6222#	6231#	6232#	6245#	6261#	6269#	6290#	6310#	
	6311#	6319#	6339#	6359#	6360#	6368#	6388#	6408#	6409#	6417#	6437#	6457#	6458#	6466#	6490#	
	6514#	6515#	6523#	6549#	6575#	6576#	6584#	6604#	6624#	6625#	6633#	6653#	6673#	6674#	6682#	
	6702#	6722#	6723#</													

MSGTT	8399#	8404#	8409#	8447#	8452#	8457#	8494#	8499#	8504#	8543#	8548#	8553#	8590#	8595#	8600#
	8637#	8642#	8647#	8684#	8689#	8694#	8731#	8736#	8741#	8778#	8783#	8788#	8825#	8830#	8835#
	8872#	8877#	8882#	8919#	8924#	8929#	8967#	8972#	8977#	9014#	9019#	9024#	9061#	9066#	9071#
	9108#	9113#	9118#	9155#	9160#	9165#	9202#	9207#	9212#	9247#	9282#	9290#			
	2634#	4807#	5615#	5845#	5856#	5903#	5917#	5943#	5949#	5978#	5986#	6007#	6035#	6065#	6071#
	6101#	6115#	6141#	6156#	6181#	6195#	6221#	6230#	6259#	6286#	6305#	6335#	6354#	6384#	6403#
	6433#	6452#	6485#	6508#	6544#	6569#	6600#	6619#	6649#	6668#	6698#	6717#	6747#	6766#	6796#
	6815#	6845#	6864#	6894#	6913#	6943#	6962#	7000#	7018#	7049#	7069#	7124#	7149#	7193#	7214#
	7377#	7378#	7411#	7412#	7444#	7445#	7654#	7668#	7705#	7746#	7752#	7792#	7798#	7838#	7844#
	7884#	7890#	7930#	7936#	7976#	7982#	8022#	8028#	8068#	8074#	8114#	8120#	8162#	8168#	8210#
MSGNGB	8216#	8258#	8264#	8305#	8311#	8352#	8358#	8399#	8405#	8447#	8453#	8494#	8500#	8543#	8549#
	8590#	8596#	8637#	8643#	8684#	8690#	8731#	8737#	8778#	8784#	8825#	8831#	8872#	8878#	8919#
	8925#	8967#	8973#	9014#	9020#	9061#	9067#	9108#	9114#	9155#	9161#	9202#	9208#		
	2689#	4807#	4813#	4848#	4857#	4871#	4894#	4923#	4983#	5116#	5556#	5557#	5558#	5559#	5560#
	5561#	5562#	5563#	5564#	5565#	5566#	5568#	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#
MSGNIN	5578#	5579#	5580#	5581#	5582#	5583#	5584#	5585#	5586#	5587#	5588#	5589#	5591#	5607#	5638#
	5756#	5784#	5801#	5819#	9236#	9279#	9298#								
	3101#	4807#	4848#	4871#	4894#	4923#	4983#	5116#	5383#	5556#	5557#	5558#	5559#	5560#	5561#
	5562#	5563#	5564#	5565#	5566#	5568#	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#
	5579#	5580#	5581#	5582#	5583#	5584#	5585#	5586#	5587#	5588#	5589#	5592#	5593#	5594#	5615#
	5624#	5652#	5653#	5655#	5656#	5658#	5659#	5662#	5663#	5675#	5676#	5753#	5770#	5774#	5785#
	5787#	5803#	5804#	5820#	5845#	5852#	5856#	5857#	5869#	5871#	5896#	5902#	5903#	5908#	5911#
	5916#	5917#	5922#	5923#	5935#	5942#	5943#	5948#	5949#	5952#	5953#	5970#	5977#	5978#	5979#
	5980#	5985#	5986#	5987#	5989#	5999#	6006#	6007#	6008#	6009#	6014#	6015#	6017#	6027#	6034#
	6035#	6036#	6037#	6042#	6043#	6045#	6056#	6064#	6065#	6070#	6071#	6074#	6075#	6094#	6100#
	6101#	6105#	6107#	6114#	6115#	6120#	6121#	6134#	6140#	6141#	6145#	6147#	6155#	6156#	6161#
	6162#	6174#	6180#	6181#	6188#	6189#	6194#	6195#	6200#	6201#	6213#	6220#	6221#		
	6229#	6230#	6231#	6232#	6257#	6259#	6261#	6275#	6285#	6286#	6290#	6293#	6304#	6305#	6310#
	6311#	6324#	6334#	6335#	6339#	6342#	6353#	6354#	6359#	6360#	6373#	6383#	6384#	6388#	6391#
	6402#	6403#	6408#	6409#	6422#	6432#	6433#	6437#	6440#	6451#	6452#	6457#	6458#	6471#	6484#
	6485#	6490#	6493#	6507#	6508#	6514#	6515#	6528#	6542#	6544#	6549#	6552#	6568#	6569#	6575#
	6576#	6589#	6599#	6600#	6604#	6607#	6618#	6619#	6624#	6625#	6638#	6648#	6649#	6653#	6656#
	6667#	6668#	6673#	6674#	6687#	6697#	6698#	6702#	6705#	6716#	6717#	6722#	6723#	6736#	6746#
	6747#	6751#	6754#	6765#	6766#	6771#	6772#	6785#	6795#	6796#	6800#	6803#	6814#	6815#	6820#
	6821#	6834#	6844#	6845#	6849#	6852#	6863#	6864#	6869#	6870#	6883#	6893#	6894#	6898#	6901#
	6912#	6913#	6918#	6919#	6932#	6942#	6943#	6947#	6950#	6961#	6962#	6967#	6968#	6981#	6999#
	7000#	7005#	7008#	7017#	7018#	7024#	7025#	7038#	7048#	7049#	7053#	7056#	7068#	7069#	7084#
	7085#	7092#	7093#	7106#	7107#	7123#	7124#	7128#	7130#	7148#	7149#	7154#	7159#	7160#	7173#
	7192#	7193#	7198#	7201#	7213#	7214#	7219#	7220#	7229#	7235#	7239#	7241#	7243#	7246#	7260#
	7264#	7266#	7268#	7271#	7283#	7298#	7304#	7305#	7308#	7320#	7336#	7338#	7342#	7346#	7354#
	7376#	7377#	7378#	7381#	7410#	7411#	7412#	7415#	7443#	7444#	7445#	7448#	7496#	7498#	7537#
	7542#	7574#	7584#	7587#	7617#	7619#	7653#	7654#	7668#	7672#	7690#	7704#	7705#	7706#	7707#
	7729#	7745#	7746#	7751#	7752#	7756#	7775#	7791#	7792#	7797#	7798#	7802#	7821#	7837#	7838#
	7843#	7844#	7848#	7867#	7883#	7884#	7889#	7890#	7894#	7913#	7929#	7930#	7935#	7936#	7940#
7959#	7975#	7976#	7981#	7982#	7986#	8005#	8021#	8022#	8027#	8028#	8032#	8051#	8067#	8068#	
8073#	8074#	8078#	8097#	8113#	8114#	8119#	8120#	8125#	8145#	8161#	8162#	8167#	8168#	8172#	
8193#	8209#	8210#	8215#	8216#	8221#	8241#	8257#	8258#	8263#	8264#	8268#	8288#	8304#	8305#	
8310#	8311#	8315#	8335#	8351#	8352#	8357#	8358#	8362#	8382#	8398#	8399#	8404#	8405#	8409#	
8430#	8446#	8447#	8452#	8453#	8457#	8477#	8493#	8494#	8499#	8500#	8504#	8526#	8542#	8543#	
8548#	8549#	8553#	8573#	8589#	8590#	8595#	8596#	8600#	8620#	8636#	8637#	8642#	8643#	8647#	
8667#	8683#	8684#	8689#	8690#	8694#	8714#	8730#	8731#	8736#	8737#	8741#	8761#	8777#	8778#	
8783#	8784#	8788#	8808#	8824#	8825#	8830#	8831#	8835#	8855#	8871#	8872#	8877#	8878#	8882#	
8902#	8918#	8919#	8924#	8925#	8929#	8950#	8966#	8967#	8972#	8973#	8977#	8997#	9013#	9014#	
9019#	9020#	9024#	9044#	9060#	9061#	9066#	9067#	9071#	9091#	9107#	9108#	9113#	9114#	9118#	
9138#	9154#	9155#	9160#	9161#	9165#	9185#	9201#	9202#	9207#	9208#	9212#	9236#	9238#	9239#	
9240#	9241#	9247#	9279#	9282#	9298#										

CZDMPAO M8207 STATIC DIAG #1				MACY11 30A(1052) 17-JUL-79 14:39				PAGE 61-6									
CZDMPA.P11 17-JUL-79 14:33				CROSS REFERENCE TABLE -- MACRO NAMES												SEQ 0232	
MSGNLS	2717#	4807#	5908#	5922#	5952#	5979#	5987#	6008#	6015#	6036#	6043#	6074#	6105#	6120#	6145#		
	6161#	6188#	6200#	6222#	6231#	6290#	6310#	6339#	6359#	6388#	6408#	6437#	6457#	6490#	6514#		
	6549#	6575#	6604#	6624#	6653#	6673#	6702#	6722#	6751#	6771#	6800#	6820#	6849#	6869#	6898#		
	6918#	6947#	6967#	7005#	7024#	7053#	7092#	7128#	7154#	7198#	7219#	7706#	7751#	7797#	7843#		
	7889#	7935#	7981#	8027#	8073#	8119#	8167#	8215#	8263#	8310#	8357#	8404#	8452#	8499#	8548#		
MSGNSU MSGNTA	8595#	8642#	8689#	8736#	8783#	8830#	8877#	8924#	8972#	9019#	9066#	9113#	9160#	9207#			
	2679#	4807#	7106#														
	2659#	4807#	4909#	4926#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#	5566#		
	5568#	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#	5583#		
	5584#	5585#	5586#	5587#	5588#	5589#	5594#	5624#	5753#	5774#	5787#	5804#	5820#	5857#	5871#		
	5923#	5953#	5989#	6017#	6045#	6075#	6121#	6162#	6201#	6232#	6261#	6311#	6360#	6409#	6458#		
	6515#	6576#	6625#	6674#	6723#	6772#	6821#	6870#	6919#	6968#	7025#	7093#	7159#	7160#	7220#		
	7246#	7271#	7308#	7346#	7381#	7415#	7448#	7498#	7542#	7587#	7619#	7672#	7707#	7756#	7802#		
	7848#	7894#	7940#	7986#	8032#	8078#	8125#	8172#	8221#	8268#	8315#	8362#	8409#	8457#	8504#		
	8553#	8600#	8647#	8694#	8741#	8788#	8835#	8882#	8929#	8977#	9024#	9071#	9118#	9165#	9212#		
MSGNTE	9247#	9282#															
	2669#	4807#	5838#	5863#	5891#	5931#	5968#	5997#	6025#	6052#	6089#	6129#	6170#	6209#	6245#		
	6269#	6319#	6368#	6417#	6466#	6523#	6584#	6633#	6682#	6731#	6780#	6829#	6878#	6927#	6976#		
	7033#	7101#	7168#	7227#	7253#	7279#	7316#	7353#	7388#	7422#	7456#	7510#	7551#	7595#	7627#		
	7683#	7720#	7766#	7812#	7858#	7904#	7950#	7996#	8042#	8088#	8136#	8183#	8232#	8279#	8326#		
MSHAPT MSHNAP MSINCR	8373#	8421#	8468#	8517#	8564#	8611#	8658#	8705#	8752#	8799#	8846#	8893#	8941#	8988#	9035#		
	9082#	9128#	9176#														
	2477#	4807#	4848#														
	2565#	4807#	4848#														
	3054#	4807#	4813#	4857#	4894#	4923#	5383#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#		
	5564#	5565#	5566#	5568#	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#		
	5581#	5582#	5583#	5584#	5585#	5586#	5587#	5588#	5589#	5591#	5592#	5593#	5594#	5607#	5624#		
	5638#	5652#	5655#	5658#	5662#	5675#	5753#	5756#	5770#	5774#	5784#	5785#	5787#	5801#	5803#		
	5804#	5819#	5820#	5838#	5845#	5852#	5856#	5857#	5863#	5869#	5871#	5891#	5896#	5902#	5903#		
	5908#	5911#	5916#	5917#	5922#	5923#	5931#	5935#	5942#	5943#	5948#	5949#	5952#	5953#	5968#		
	5970#	5977#	5978#	5979#	5980#	5985#	5986#	5987#	5989#	5997#	5999#	6006#	6007#	6008#	6009#		
	6014#	6015#	6017#	6025#	6027#	6034#	6035#	6036#	6037#	6042#	6043#	6045#	6052#	6056#	6064#		
	6065#	6070#	6071#	6074#	6075#	6089#	6094#	6100#	6101#	6105#	6107#	6114#	6115#	6120#	6121#		
	6129#	6134#	6140#	6141#	6145#	6147#	6155#	6156#	6161#	6162#	6170#	6174#	6180#	6181#	6188#		
	6189#	6194#	6195#	6200#	6201#	6209#	6213#	6220#	6221#	6222#	6224#	6229#	6230#	6231#	6232#		
	6245#	6257#	6259#	6261#	6269#	6275#	6285#	6286#	6290#	6293#	6304#	6305#	6310#	6311#	6319#		
	6324#	6334#	6335#	6339#	6342#	6353#	6354#	6359#	6360#	6368#	6373#	6383#	6384#	6388#	6391#		
	6402#	6403#	6408#	6409#	6417#	6422#	6432#	6433#	6437#	6440#	6451#	6452#	6457#	6458#	6466#		
	6471#	6484#	6485#	6490#	6493#	6507#	6508#	6514#	6515#	6523#	6528#	6542#	6544#	6549#	6552#		
	6568#	6569#	6575#	6576#	6584#	6589#	6599#	6600#	6604#	6607#	6618#	6619#	6624#	6625#	6633#		
	6638#	6648#	6649#	6653#	6656#	6667#	6668#	6673#	6674#	6682#	6687#	6697#	6698#	6702#	6705#		
	6716#	6717#	6722#	6723#	6731#	6736#	6746#	6747#	6751#	6754#	6765#	6766#	6771#	6772#	6780#		
	6785#	6795#	6796#	6800#	6803#	6814#	6815#	6820#	6821#	6829#	6834#	6844#	6845#	6849#	6852#		
	6863#	6864#	6869#	6870#	6878#	6883#	6893#	6894#	6898#	6901#	6912#	6913#	6918#	6919#	6927#		
	6932#	6942#	6943#	6947#	6950#	6961#	6962#	6967#	6968#	6976#	6981#	6999#	7000#	7005#	7008#		
	7017#	7018#	7024#	7025#	7033#	7038#	7048#	7049#	7053#	7056#	7068#	7069#	7084#	7085#	7092#		
	7093#	7101#	7106#	7107#	7123#	7124#	7128#	7130#	7148#	7149#	7154#	7159#	7160#	7168#	7173#		
	7192#	7193#	7198#	7201#	7213#	7214#	7219#	7220#	7227#	7229#	7235#	7239#	7241#	7243#	7246#		
	7253#	7260#	7264#	7266#	7268#	7271#	7279#	7283#	7298#	7304#	7305#	7308#	7316#	7320#	7336#		
	7338#	7342#	7346#	7353#	7354#	7376#	7377#	7378#	7381#	7388#	7410#	7411#	7412#	7415#	7422#		
	7443#	7444#	7445#	7448#	7456#	7496#	7498#	7510#	7537#	7542#	7551#	7574#	7584#	7587#	7595#		
	7617#	7619#	7627#	7653#	7654#	7668#	7672#	7683#	7690#	7704#	7705#	7706#	7707#	7720#	7729#		
	7745#	7746#	7751#	7752#	7756#	7766#	7775#	7791#	7792#	7797#	7798#	7802#	7812#	7821#	7837#		
	7838#	7843#	7844#	7848#	7858#	7867#	7883#	7884#	7889#	7890#	7894#	7904#	7913#	7929#	7930#		
	7935#	7936#	7940#	7950#	7959#	7975#	7976#	7981#	7982#	7986#	7996#	8005#	8021#	8022#	8027#		
	8028#	8032#	8042#	8051#	8067#	8068#	8073#	8074#	8078#	8088#	8097#	8113#	8114#	8119#	8120#		



	8125#	8136#	8145#	8161#	8162#	8167#	8168#	8172#	8183#	8193#	8209#	8210#	8215#	8216#	8221#
	8232#	8241#	8257#	8258#	8263#	8264#	8268#	8279#	8288#	8304#	8305#	8310#	8311#	8315#	8326#
	8335#	8351#	8352#	8357#	8358#	8362#	8373#	8382#	8398#	8399#	8404#	8405#	8409#	8421#	8430#
	8446#	8447#	8452#	8453#	8457#	8468#	8477#	8493#	8494#	8499#	8500#	8504#	8517#	8526#	8542#
	8543#	8548#	8549#	8553#	8564#	8573#	8589#	8590#	8595#	8596#	8600#	8611#	8620#	8636#	8637#
	8642#	8643#	8647#	8658#	8667#	8683#	8684#	8689#	8690#	8694#	8705#	8714#	8730#	8731#	8736#
	8737#	8741#	8752#	8761#	8777#	8778#	8783#	8784#	8788#	8799#	8808#	8824#	8825#	8830#	8831#
	8835#	8846#	8855#	8871#	8872#	8877#	8878#	8882#	8893#	8902#	8918#	8919#	8924#	8925#	8929#
	8941#	8950#	8966#	8967#	8972#	8973#	8977#	8988#	8997#	9013#	9014#	9019#	9020#	9024#	9035#
	9044#	9060#	9061#	9066#	9067#	9071#	9082#	9091#	9107#	9108#	9113#	9114#	9118#	9128#	9138#
	9154#	9155#	9160#	9161#	9165#	9176#	9185#	9201#	9202#	9207#	9208#	9212#	9236#	9279#	
MSIOSE	2431#	4807#													
MSLDRO	2771#	4807#	5652#	5655#	5658#	5662#	5675#	5770#	7235#	7239#	7260#	7264#	7283#	7298#	7320#
	7336#														
MSMASK	2390#	4807#													
MSMCHI	87#	4807#													
MSMCLO	2327#	4807#													
MSMSK1	2402#	4807#													
MSPOP	2646#	4807#	4861#	4909#	4926#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#
	5566#	5568#	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#
	5583#	5584#	5585#	5586#	5587#	5588#	5589#	5594#	5624#	5753#	5774#	5787#	5804#	5820#	5857#
	5871#	5908#	5922#	5923#	5952#	5953#	5979#	5987#	5989#	6008#	6015#	6017#	6036#	6043#	6045#
	6074#	6075#	6105#	6120#	6121#	6145#	6161#	6162#	6188#	6200#	6201#	6222#	6231#	6232#	6261#
	6290#	6310#	6311#	6339#	6359#	6360#	6388#	6408#	6409#	6437#	6457#	6458#	6490#	6514#	6515#
	6549#	6575#	6576#	6604#	6624#	6625#	6653#	6673#	6674#	6702#	6722#	6723#	6751#	6771#	6772#
	6800#	6820#	6821#	6849#	6869#	6870#	6898#	6918#	6919#	6947#	6967#	6968#	7005#	7024#	7025#
	7053#	7092#	7093#	7128#	7154#	7159#	7160#	7198#	7219#	7220#	7246#	7271#	7308#	7346#	7381#
	7415#	7448#	7498#	7542#	7587#	7619#	7672#	7706#	7707#	7751#	7756#	7797#	7802#	7843#	7848#
	7889#	7894#	7935#	7940#	7981#	7986#	8027#	8032#	8073#	8078#	8119#	8125#	8167#	8172#	8215#
	8221#	8263#	8268#	8310#	8315#	8357#	8362#	8404#	8409#	8452#	8457#	8499#	8504#	8548#	8553#
	8595#	8600#	8642#	8647#	8689#	8694#	8736#	8741#	8783#	8788#	8830#	8835#	8877#	8882#	8924#
	8929#	8972#	8977#	9019#	9024#	9066#	9071#	9113#	9118#	9160#	9165#	9207#	9212#	9247#	9282#
	9290#														
MSPRIN	2349#	4807#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#	5566#	5568#	5570#
	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#	5583#	5584#	5585#
	5586#	5587#	5588#	5589#	5592#	5593#									
MSPUSH	2337#	4807#	4813#	4857#	4894#	4923#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#
	5565#	5566#	5568#	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#
	5582#	5583#	5584#	5585#	5586#	5587#	5588#	5589#	5591#	5607#	5638#	5756#	5784#	5801#	5819#
	5838#	5863#	5891#	5896#	5911#	5931#	5935#	5968#	5970#	5980#	5997#	5999#	6009#	6025#	6027#
	6037#	6052#	6056#	6089#	6094#	6107#	6129#	6134#	6147#	6170#	6174#	6189#	6209#	6213#	6224#
	6245#	6269#	6275#	6293#	6319#	6324#	6342#	6368#	6373#	6391#	6417#	6422#	6440#	6466#	6471#
	6493#	6523#	6528#	6552#	6584#	6589#	6607#	6633#	6638#	6656#	6682#	6687#	6705#	6731#	6736#
	6754#	6780#	6785#	6803#	6829#	6834#	6852#	6878#	6883#	6901#	6927#	6932#	6950#	6976#	6981#
	7008#	7033#	7038#	7056#	7101#	7106#	7107#	7130#	7168#	7173#	7201#	7227#	7253#	7279#	7316#
	7353#	7388#	7422#	7456#	7510#	7551#	7595#	7627#	7683#	7690#	7720#	7729#	7766#	7775#	7812#
	7821#	7858#	7867#	7904#	7913#	7950#	7959#	7996#	8005#	8042#	8051#	8088#	8097#	8136#	8145#
	8183#	8193#	8232#	8241#	8279#	8288#	8326#	8335#	8373#	8382#	8421#	8430#	8468#	8477#	8517#
	8526#	8564#	8573#	8611#	8620#	8658#	8667#	8705#	8714#	8752#	8761#	8799#	8808#	8846#	8855#
	8893#	8902#	8941#	8950#	8988#	8997#	9035#	9044#	9082#	9091#	9128#	9138#	9176#	9185#	9236#
	9279#														
MSPUT	2819#	4807#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#	5566#	5568#	5570#
	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#	5583#	5584#	5585#
	5586#	5587#	5588#	5589#	5592#	5593#									
MSPUT1	2842#	4807#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#	5566#	5568#	5570#
	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#	5583#	5584#	5585#

MSRADI MSRBRO MSRNRO MSSETS	5586#	5587#	5588#	5589#	5592#	5593#									
	3151#	4807#	9238#	9239#	9240#	9241#									
	2787#	4807#													
	2802#	4807#	5675#												
	3071#	4807#	4813#	4857#	4894#	4923#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#
	5565#	5566#	5568#	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#
	5582#	5583#	5584#	5585#	5586#	5587#	5588#	5589#	5591#	5607#	5638#	5756#	5784#	5801#	5819#
	5838#	5863#	5891#	5896#	5911#	5931#	5935#	5968#	5970#	5980#	5997#	5999#	6009#	6025#	6027#
	6037#	6052#	6056#	6089#	6094#	6107#	6129#	6134#	6147#	6170#	6174#	6189#	6209#	6213#	6224#
	6245#	6269#	6275#	6293#	6319#	6324#	6342#	6368#	6373#	6391#	6417#	6422#	6440#	6466#	6471#
	6493#	6523#	6528#	6552#	6584#	6589#	6607#	6633#	6638#	6656#	6682#	6687#	6705#	6731#	6736#
	6754#	6780#	6785#	6803#	6829#	6834#	6852#	6878#	6883#	6901#	6927#	6932#	6950#	6976#	6981#
	7008#	7033#	7038#	7056#	7101#	7106#	7107#	7130#	7168#	7173#	7201#	7227#	7253#	7279#	7316#
	7353#	7388#	7422#	7456#	7510#	7551#	7595#	7627#	7683#	7690#	7720#	7729#	7766#	7775#	7812#
	7821#	7858#	7867#	7904#	7913#	7950#	7959#	7996#	8005#	8042#	8051#	8088#	8097#	8136#	8145#
	8183#	8193#	8232#	8241#	8279#	8288#	8326#	8335#	8373#	8382#	8421#	8430#	8468#	8477#	8517#
8526#	8564#	8573#	8611#	8620#	8658#	8667#	8705#	8714#	8752#	8761#	8799#	8808#	8846#	8855#	
8893#	8902#	8941#	8950#	8988#	8997#	9035#	9044#	9082#	9091#	9128#	9138#	9176#	9185#	9236#	
9279#															
MSSTAR MS SVC	2468#	4807#													
	2746#	4807#	5383	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#	5566#	5568#
	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#	5583#	5584#
	5585#	5586#	5587#	5588#	5589#	5592#	5593#	5594#	5615#	5624#	5652#	5655#	5658#	5662#	5675#
	5753#	5770#	5774#	5785#	5787#	5803#	5804#	5820#	5845#	5852	5856#	5857#	5869	5871#	5896#
	5902	5903#	5908#	5911#	5916	5917#	5922#	5923#	5935#	5942	5943#	5948	5949#	5952#	5953#
	5970#	5977	5978#	5979#	5980#	5985	5986#	5987#	5989#	5999#	6006	6007#	6008#	6009#	6014
	6015#	6017#	6027#	6034	6035#	6036#	6037#	6042	6043#	6045#	6056#	6064	6065#	6070	6071#
	6074#	6075#	6094#	6100	6101#	6105#	6107#	6114	6115#	6120#	6121#	6134#	6140	6141#	6145#
	6147#	6155	6156#	6161#	6162#	6174#	6180	6181#	6188#	6189#	6194	6195#	6200#	6201#	6213#
	6220	6221#	6222#	6224#	6229	6230#	6231#	6232#	6257	6259#	6261#	6275#	6285	6286#	6290#
	6293#	6304	6305#	6310#	6311#	6324#	6334	6335#	6339#	6342#	6353	6354#	6359#	6360#	6373#
	6383	6384#	6388#	6391#	6402	6403#	6408#	6409#	6422#	6432	6433#	6437#	6440#	6451	6452#
	6457#	6458#	6471#	6484	6485#	6490#	6493#	6507	6508#	6514#	6515#	6528#	6542	6544#	6549#
	6552#	6568	6569#	6575#	6576#	6589#	6599	6600#	6604#	6607#	6618	6619#	6624#	6625#	6638#
	6648	6649#	6653#	6656#	6667	6668#	6673#	6674#	6687#	6697	6698#	6702#	6701	6716	6717#
	6722#	6723#	6736#	6746	6747#	6751#	6754#	6765	6766#	6771#	6772#	6785#	6795	6796#	6800#
	6803#	6814	6815#	6820#	6821#	6834#	6844	6845#	6849#	6852#	6863	6864#	6869#	6870#	6883#
	6893	6894#	6898#	6901#	6912	6913#	6918#	6919#	6932#	6942	6943#	6947#	6950#	6961	6962#
	6967#	6968#	6981#	6999	7000#	7005#	7008#	7017	7018#	7024#	7025#	7038#	7048	7049#	7053#
	7056#	7068	7069#	7084	7085#	7092#	7093#	7106#	7107#	7123	7124#	7128#	7130#	7148	7149#
	7154#	7159#	7160#	7173#	7192	7193#	7198#	7201#	7213	7214#	7219#	7220#	7229#	7235#	7239#
	7241	7243	7246#	7260#	7264#	7266	7268	7271#	7283#	7298#	7304#	7305	7308#	7320#	7336#
	7338	7342	7346#	7354#	7376	7377#	7378#	7381#	7410	7411#	7412#	7415#	7443	7444#	7445#
	7448#	7496	7498#	7537	7542#	7574	7584	7587#	7617	7619#	7653	7654#	7668#	7672#	7690#
	7704	7705#	7706#	7707#	7729#	7745	7746#	7751#	7752#	7756#	7775#	7791	7792#	7797#	7798#
	7802#	7821#	7837	7838#	7843#	7844#	7848#	7867#	7883	7884#	7889#	7890#	7894#	7913#	7929
	7930#	7935#	7936#	7940#	7959#	7975	7976#	7981#	7982#	7986#	8005#	8021	8022#	8027#	8028#
	8032#	8051#	8067	8068#	8073#	8074#	8078#	8097#	8113	8114#	8119#	8120#	8125#	8145#	8161
	8162#	8167#	8168#	8172#	8193#	8209	8210#	8215#	8216#	8221#	8241#	8257	8258#	8263#	8264#
	8268#	8288#	8304	8305#	8310#	8311#	8315#	8335#	8351	8352#	8357#	8358#	8362#	8382#	8398
	8399#	8404#	8405#	8409#	8430#	8446	8447#	8452#	8453#	8457#	8477#	8493	8494#	8499#	8500#
	8504#	8526#	8542	8543#	8548#	8549#	8553#	8573#	8589	8590#	8595#	8596#	8600#	8620#	8636
	8637#	8642#	8643#	8647#	8667#	8683	8684#	8689#	8690#	8694#	8714#	8730	8731#	8736#	8737#
	8741#	8761#	8777	8778#	8783#	8784#	8788#	8808#	8824	8825#	8830#	8831#	8835#	8855#	8871
	8872#	8877#	8878#	8882#	8902#	8918	8919#	8924#	8925#	8929#	8950#	8966	8967#	8972#	8973#
	8977#	8997#	9013	9014#	9019#	9020#	9024#	9044#	9060	9061#	9066#	9067#	9071#	9091#	9107



	9108#	9113#	9114#	9118#	9138#	9154	9155#	9160#	9161#	9165#	9185#	9201	9202#	9207#	9208#
MSTLAB	9212#														
	2739#	4807#	5383#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#	5566#	5568#
	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#	5583#	5584#
	5585#	5586#	5587#	5588#	5589#	5592#	5593#	5594#	5624#	5652#	5655#	5658#	5662#	5675#	5753#
	5770#	5774#	5785#	5787#	5803#	5804#	5820#	5845#	5852#	5856#	5857#	5869#	5871#	5896#	5902#
	5903#	5908#	5911#	5916#	5917#	5922#	5923#	5935#	5942#	5943#	5948#	5949#	5952#	5953#	5970#
	5977#	5978#	5979#	5980#	5985#	5986#	5987#	5989#	5999#	6006#	6007#	6008#	6009#	6014#	6015#
	6017#	6027#	6034#	6035#	6036#	6037#	6042#	6043#	6045#	6056#	6064#	6065#	6070#	6071#	6074#
	6075#	6094#	6100#	6101#	6105#	6107#	6114#	6115#	6120#	6121#	6134#	6140#	6141#	6145#	6147#
	6155#	6156#	6161#	6162#	6174#	6180#	6181#	6188#	6189#	6194#	6195#	6200#	6201#	6213#	6220#
	6221#	6222#	6224#	6229#	6230#	6231#	6232#	6257#	6259#	6261#	6275#	6285#	6286#	6290#	6293#
	6304#	6305#	6310#	6311#	6324#	6334#	6335#	6339#	6342#	6353#	6354#	6359#	6360#	6373#	6383#
	6384#	6388#	6391#	6402#	6403#	6408#	6409#	6422#	6432#	6433#	6437#	6440#	6451#	6452#	6457#
	6458#	6471#	6484#	6485#	6490#	6493#	6507#	6508#	6514#	6515#	6528#	6542#	6544#	6549#	6552#
	6568#	6569#	6575#	6576#	6589#	6599#	6600#	6604#	6607#	6618#	6619#	6624#	6625#	6638#	6648#
	6649#	6653#	6656#	6667#	6668#	6673#	6674#	6687#	6697#	6698#	6702#	6705#	6716#	6717#	6722#
	6723#	6736#	6746#	6747#	6751#	6754#	6765#	6766#	6771#	6772#	6785#	6795#	6796#	6800#	6803#
	6814#	6815#	6820#	6821#	6834#	6844#	6845#	6849#	6852#	6863#	6864#	6869#	6870#	6883#	6893#
	6894#	6898#	6901#	6912#	6913#	6918#	6919#	6932#	6942#	6943#	6947#	6950#	6961#	6962#	6967#
	6968#	6981#	6999#	7000#	7005#	7008#	7017#	7018#	7024#	7025#	7038#	7048#	7049#	7053#	7056#
	7068#	7069#	7084#	7085#	7092#	7093#	7106#	7107#	7123#	7124#	7128#	7130#	7148#	7149#	7154#
	7159#	7160#	7173#	7192#	7193#	7198#	7201#	7213#	7214#	7219#	7220#	7229#	7235#	7239#	7241#
	7243#	7246#	7260#	7264#	7266#	7268#	7271#	7283#	7298#	7304#	7305#	7308#	7320#	7336#	7338#
	7342#	7346#	7354#	7376#	7377#	7378#	7381#	7410#	7411#	7412#	7415#	7443#	7444#	7445#	7448#
	7496#	7498#	7537#	7542#	7574#	7584#	7587#	7617#	7619#	7653#	7654#	7668#	7672#	7690#	7704#
	7705#	7706#	7707#	7729#	7745#	7746#	7751#	7752#	7756#	7775#	7791#	7792#	7797#	7798#	7802#
	7821#	7837#	7838#	7843#	7844#	7848#	7867#	7883#	7884#	7889#	7890#	7894#	7913#	7929#	7930#
	7935#	7936#	7940#	7959#	7975#	7976#	7981#	7982#	7986#	8005#	8021#	8022#	8027#	8028#	8032#
	8051#	8067#	8068#	8073#	8074#	8078#	8097#	8113#	8114#	8119#	8120#	8125#	8145#	8161#	8162#
	8167#	8168#	8172#	8193#	8209#	8210#	8215#	8216#	8221#	8241#	8257#	8258#	8263#	8264#	8268#
	8288#	8304#	8305#	8310#	8311#	8315#	8335#	8351#	8352#	8357#	8358#	8362#	8382#	8398#	8399#
	8404#	8405#	8409#	8430#	8446#	8447#	8452#	8453#	8457#	8477#	8493#	8494#	8499#	8500#	8504#
	8526#	8542#	8543#	8548#	8549#	8553#	8573#	8589#	8590#	8595#	8596#	8600#	8620#	8636#	8637#
	8642#	8643#	8647#	8667#	8683#	8684#	8689#	8690#	8694#	8714#	8730#	8731#	8736#	8737#	8741#
	8761#	8777#	8778#	8783#	8784#	8788#	8808#	8824#	8825#	8830#	8831#	8835#	8855#	8871#	8872#
	8877#	8878#	8882#	8902#	8918#	8919#	8924#	8925#	8929#	8950#	8966#	8967#	8972#	8973#	8977#
	8997#	9013#	9014#	9019#	9020#	9024#	9044#	9060#	9061#	9066#	9067#	9071#	9091#	9107#	9108#
MSTSTL	9113#	9114#	9118#	9138#	9154#	9155#	9160#	9161#	9165#	9185#	9201#	9202#	9207#	9208#	9212#
	2728#	4807#	5383#	5556#	5557#	5558#	5559#	5560#	5561#	5562#	5563#	5564#	5565#	5566#	5568#
	5570#	5571#	5572#	5573#	5574#	5575#	5576#	5577#	5578#	5579#	5580#	5581#	5582#	5583#	5584#
	5585#	5586#	5587#	5588#	5589#	5592#	5593#	5594#	5624#	5652#	5655#	5658#	5662#	5675#	5753#
	5770#	5774#	5785#	5787#	5803#	5804#	5820#	5845#	5852#	5856#	5857#	5869#	5871#	5896#	5902#
	5903#	5908#	5911#	5916#	5917#	5922#	5923#	5935#	5942#	5943#	5948#	5949#	5952#	5953#	5970#
	5977#	5978#	5979#	5980#	5985#	5986#	5987#	5989#	5999#	6006#	6007#	6008#	6009#	6014#	6015#
	6017#	6027#	6034#	6035#	6036#	6037#	6042#	6043#	6045#	6056#	6064#	6065#	6070#	6071#	6074#
	6075#	6094#	6100#	6101#	6105#	6107#	6114#	6115#	6120#	6121#	6134#	6140#	6141#	6145#	6147#
	6155#	6156#	6161#	6162#	6174#	6180#	6181#	6188#	6189#	6194#	6195#	6200#	6201#	6213#	6220#
	6221#	6222#	6224#	6229#	6230#	6231#	6232#	6257#	6259#	6261#	6275#	6285#	6286#	6290#	6293#
	6304#	6305#	6310#	6311#	6324#	6334#	6335#	6339#	6342#	6353#	6354#	6359#	6360#	6373#	6383#
	6384#	6388#	6391#	6402#	6403#	6408#	6409#	6422#	6432#	6433#	6437#	6440#	6451#	6452#	6457#
	6458#	6471#	6484#	6485#	6490#	6493#	6507#	6508#	6514#	6515#	6528#	6542#	6544#	6549#	6552#
	6568#	6569#	6575#	6576#	6589#	6599#	6600#	6604#	6607#	6618#	6619#	6624#	6625#	6638#	6648#
	6649#	6653#	6656#	6667#	6668#	6673#	6674#	6687#	6697#	6698#	6702#	6705#	6716#	6717#	6722#
	6723#	6736#	6746#	6747#	6751#	6754#	6765#	6766#	6771#	6772#	6785#	6795#	6796#	6800#	6803#
	6814#	6815#	6820#	6821#	6834#	6844#	6845#	6849#	6852#	6863#	6864#	6869#	6870#	6883#	6893#

CZDMPAO M8207 STATIC DIAG #1  
CZDMPA.P11 17-JUL-79 14:33MACY11 30A(1052) 17-JUL-79 14:39 PAGE 61-10  
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0236

	6894#	6898#	6901#	6912#	6913#	6918#	6919#	6932#	6942#	6943#	6947#	6950#	6961#	6962#	6967#
	6968#	6981#	6999#	7000#	7005#	7008#	7017#	7018#	7024#	7025#	7038#	7048#	7049#	7053#	7056#
	7068#	7069#	7084#	7085#	7092#	7093#	7106#	7107#	7123#	7124#	7128#	7130#	7148#	7149#	7154#
	7159#	7160#	7173#	7192#	7193#	7198#	7201#	7213#	7214#	7219#	7220#	7229#	7235#	7239#	7241#
	7243#	7246#	7260#	7264#	7266#	7268#	7271#	7283#	7298#	7304#	7305#	7308#	7320#	7336#	7338#
	7342#	7346#	7354#	7376#	7377#	7378#	7381#	7410#	7411#	7412#	7415#	7443#	7444#	7445#	7448#
	7496#	7498#	7537#	7542#	7574#	7584#	7587#	7617#	7619#	7653#	7654#	7668#	7672#	7690#	7704#
	7705#	7706#	7707#	7729#	7745#	7746#	7751#	7752#	7756#	7775#	7791#	7792#	7797#	7798#	7802#
	7821#	7837#	7838#	7843#	7844#	7848#	7867#	7883#	7884#	7889#	7890#	7894#	7913#	7929#	7930#
	7935#	7936#	7940#	7959#	7975#	7976#	7981#	7982#	7986#	8005#	8021#	8022#	8027#	8028#	8032#
	8051#	8067#	8068#	8073#	8074#	8078#	8097#	8113#	8114#	8119#	8120#	8125#	8145#	8161#	8162#
	8167#	8168#	8172#	8193#	8209#	8210#	8215#	8216#	8221#	8241#	8257#	8258#	8263#	8264#	8268#
	8288#	8304#	8305#	8310#	8311#	8315#	8335#	8351#	8352#	8357#	8358#	8362#	8382#	8398#	8399#
	8404#	8405#	8409#	8430#	8446#	8447#	8452#	8453#	8457#	8477#	8493#	8494#	8499#	8500#	8504#
	8526#	8542#	8543#	8548#	8549#	8553#	8573#	8589#	8590#	8595#	8596#	8600#	8620#	8636#	8637#
	8642#	8643#	8647#	8667#	8683#	8684#	8689#	8690#	8694#	8714#	8730#	8731#	8736#	8737#	8741#
	8761#	8777#	8778#	8783#	8784#	8788#	8808#	8824#	8825#	8830#	8831#	8835#	8855#	8871#	8872#
	8877#	8878#	8882#	8902#	8918#	8919#	8924#	8925#	8929#	8950#	8966#	8967#	8972#	8973#	8977#
	8997#	9013#	9014#	9019#	9020#	9024#	9044#	9060#	9061#	9066#	9067#	9071#	9091#	9107#	9108#
	9113#	9114#	9118#	9138#	9154#	9155#	9160#	9161#	9165#	9185#	9201#	9202#	9207#	9208#	9212#
MSWORD	2888#	4807#	4848#	4871#	5383#	5615#	5852#	5869#	5902#	5916#	5942#	5948#	5977#	5985#	6006#
	6014#	6034#	6042#	6064#	6070#	6100#	6114#	6140#	6155#	6180#	6194#	6220#	6229#	6257#	6285#
	6304#	6334#	6353#	6383#	6402#	6432#	6451#	6484#	6507#	6542#	6568#	6599#	6618#	6648#	6667#
	6697#	6716#	6746#	6765#	6795#	6814#	6844#	6863#	6893#	6912#	6942#	6961#	6999#	7017#	7048#
	7068#	7084#	7123#	7148#	7192#	7213#	7241#	7243#	7266#	7268#	7305#	7338#	7342#	7376#	7378#
	7410#	7412#	7443#	7445#	7496#	7537#	7574#	7584#	7617#	7653#	7668#	7704#	7745#	7752#	7791#
	7798#	7837#	7844#	7883#	7890#	7929#	7936#	7975#	7982#	8021#	8028#	8067#	8074#	8113#	8120#
	8161#	8168#	8209#	8216#	8257#	8264#	8304#	8311#	8351#	8358#	8398#	8405#	8446#	8453#	8493#
	8500#	8542#	8549#	8589#	8596#	8636#	8643#	8683#	8690#	8730#	8737#	8777#	8784#	8824#	8831#
	8871#	8878#	8918#	8925#	8966#	8973#	9013#	9020#	9060#	9067#	9107#	9114#	9154#	9161#	9201#
	9208#	9238#	9239#	9240#	9241#	9298									
MSXFER	2410#	4807#													
OPEN	1694#	4807#													
POINTE	1702#	4807#	4838												
PRINTB	1768#	4807#	5556	5557	5558	5559	5560	5561	5562	5563	5564	5565	5566	5568	5570
	5571	5572	5573	5574	5575	5576	5577	5578	5579	5580	5581	5582	5583	5584	5585
	5586	5587	5588	5589											
	1811#	4807#	5592	5593											
PRINTS	1854#	4807#													
PRINTX	1897#	4807#													
READBU	1940#	4807#													
READEF	1949#	4807#	5652	5655	5658	5662									
RERROR	5155#	7123	7148	7192	7213										
RFLAGS	1967#	4807#													
ROMCLK	5194#	5230	5232	5234	5240	5247	5254	5261	5269	5283	5344	5362	5364	5367	5372
	5376	5403	5414	5416	5424	5426	6278	6280	6297	6299	6327	6329	6346	6348	6376
	6378	6395	6397	6425	6427	6444	6446	6475	6477	6498	6500	6532	6534	6557	6559
	6592	6594	6611	6613	6641	6643	6660	6662	6690	6692	6709	6711	6739	6741	6758
	6760	6788	6790	6807	6809	6837	6839	6856	6858	6886	6888	6905	6907	6935	6937
	6954	6956	6986	6994	7012	7041	7043	7060	7062	7079	7111	7115	7117	7136	7140
	7142	7178	7183	7186	7205	7207	7237	7262	7296	7334	7366	7369	7399	7403	7405
	7433	7436	7467	7489	7491	7528	7530	7561	7564	7567	7578	7604	7607	7610	7646
	7692	7694	7696	7698	7733	7737	7739	7779	7783	7785	7825	7829	7831	7871	7875
	7877	7917	7921	7923	7963	7967	7969	8009	8013	8015	8055	8059	8061	8101	8105
	8107	8149	8153	8155	8197	8201	8203	8245	8249	8251	8292	8296	8298	8339	8343
	8345	8386	8390	8392	8434	8438	8440	8481	8485	8487	8530	8534	8536	8577	8581

	8583	8624	8628	8630	8671	8675	8677	8718	8722	8724	8765	8769	8771	8812	8816
	8818	8859	8863	8865	8906	8910	8912	8954	8958	8960	9001	9005	9007	9048	9052
	9054	9095	9099	9101	9142	9146	9148	9189	9193	9195					
SETPRI	1977#	4807#	7235	7239	7260	7264	7283	7298	7320	7336					
SETVEC	1986#	4807#													
SLASH	1998#	4807#													
STARS	2015#	4807#													
SVC	2036#	4806#	4807												
XFER	2299#	4807#	5615#	7378#	7412#	7445#	7668#	7752#	7798#	7844#	7890#	7936#	7982#	8028#	8074#
	8120#	8168#	8216#	8264#	8311#	8358#	8405#	8453#	8500#	8549#	8596#	8643#	8690#	8737#	8784#
	8831#	8878#	8925#	8973#	9020#	9067#	9114#	9161#	9208#						
XFERF	2310#	4807#													
XFERT	2319#	4807#													
SMD	5540#	5556	5557	5558	5559	5560	5561	5562	5563	5564	5565	5566	5568	5570	5571
	5572	5573	5574	5575	5576	5577	5578	5579	5580	5581	5582	5583	5584	5585	5586
	5587	5588	5589												

. ABS. 040004 000

ERRORS DETECTED: 0

SAIL:CZDMPA,CZDMPA/CRF/NL:TOC=CZDMP.MLB,CZDMPA.P11  
RUN-TIME: 192 224 23 SECONDS  
RUN-TIME RATIO: 2338/441=5.2  
CORE USED: 21k (41 PAGES)