

DMP-11, DMR-11,
M8207

M8207 STATIC DIAG#2
CZDMQCO

AH-E229C-MC
FICHE 1 OF 1

OCT 1981
COPYRIGHT © 79-81
MADE IN USA



CZDMQC M8207 STATIC DIAG. #2
CZDMQC.P11 21-JUL-81 14:36

MACY 1 30A(1052) 21-JUL-81 14:48 PAGE 2
PROGRAM DOCUMENT

SEQ 0001

2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231

.REM @

IDENTIFICATION

PRODUCT CODE: AC-E228C-MC
PRODUCT NAME: CZDMQC0 M8207 STATIC DIAG #2
PRODUCT DATE: JULY 1981
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) 1979,1981 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	VNIBUS	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

2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341

1.0 INTRODUCTION

1.1 PROGRAM ABSTRACT

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

THE PROGRAM WAS IMPLEMENTED USING THE DIAGNOSTIC SUPERVISOR.

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

1.2 HARDWARE INTRODUCTION

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

2.0 HARDWARE REQUIREMENTS

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

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

3.0 PRELIMINARY PROGRAM REQUIREMENTS

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

4.0 GENERAL PROGRAM CONSIDERATIONS

4.1 DIAGNOSTIC SUPERVISOR

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

2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397

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 120 SECONDS PER PASS FOR EACH UNIT.

4.3 XXDP+

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

4.4 ACT/SLIDE

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

4.5 APT

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

4.6 MEMORY MANAGEMENT

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

4.7 MEMORY PARITY OPTION

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

4.8 ERROR LOGGING

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

5.0 PROGRAM LOAD MEDIA

THIS PROGRAM CAN BE LOADED FROM PAPER TAPE USING THE 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+,

2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453

THE DIAGNOSTIC SUPERVISOR SHOULD BE LOADED FIRST, FOLLOWED BY THE DIAGNOSTIC PROGRAM.

6.0 OPERATING INSTRUCTIONS

6.1 LOADING AND STARTING PROCEDURES

6.1.1 LOADING PROCEDURES

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

6.1.2 STARTING PROCEDURES

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

6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

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

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

6.2 INITIAL DIALOGUE

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

DRS LOADED
DIAG. RUN-TIME SERVICES
CZDMQ-C-0
M8207 DIAG. #2 OF 2
UNIT IS M8200,M8204,OR M8207
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:

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

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 COMMAND.

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>

2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677

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>

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

<FLAG-LIST> IS AS IN THE START COMMAND, BUT UNSPECIFIED FIAGS 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.

2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733

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

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

6.3.7 PRINT COMMAND

PRI(NT)

6.3.7.1 EFFECT OF PRINT COMMAND

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

6.3.8 DISPLAY COMMAND

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

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

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

2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789

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.

6.3.10 ZFLAGS COMMAND

ZFL(AGS)

6.3.10.1 EFFECT OF ZFLAGS COMMAND

ALL FLAGS ARE CLEARED.

6.3.11 CONTROL CHARACTERS

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

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

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

6.3.12 HARDWARE PARAMETERS

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

2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845

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

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

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

3. MICRO-PROCESSOR RUN SWITCH-TYPE 1 IF ON, IF OFF: (0) 0?

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

6.3.13 SOFTWARE PARAMETERS

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

6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

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

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

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

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

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

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

2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900

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 16 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).

7.0 TEST DESCRIPTIONS

..... TEST
.....

2902 *VERIFY THAT REFERENCING UNIBUS DEVICE REGISTERS
2903 *DOES NOT CAUSE A TIME OUT TRAP
2904 *****
2905
2906
2907 ***** TEST 2 *****
2908 *TEST OF BR RIGHT SHIFT
2909 *VERIFY THAT A DEST OF BR RSH (011) OF A MICRO-INSTRUCTION
2910 *SHIFTS THE RESULTING BR DATA RIGHT ONCE.
2911 *****
2912
2913
2914 ***** TEST 3 *****
2915 *IOP CRAM WRITE/READ TEST
2916 *FLOAT A 1 THROUGH EACH CRAM LOCATION
2917 *****
2918
2919
2920 ***** TEST 4 *****
2921 *IOP CRAM WRITE/READ TEST
2922 *FLOAT A 0 THROUGH EACH CRAM LOCATION
2923 *****
2924
2925
2926 ***** TEST 5 *****
2927 *IOP CRAM DUAL ADDRESSING TEST
2928 *WRITE EACH ADDRESS INTO ITSELF, READ EACH
2929 *ADDRESS TO VERIFY CORRECT ADDRESSING
2930 *****
2931
2932
2933 ***** TEST 6 *****
2934 *IOP MAIN MEMORY TEST
2935 *FLOAT A 1 THROUGH ALL MAIN MEMORY LOCATIONS
2936 *****
2937
2938
2939 ***** TEST 7 *****
2940 *IOP MAIN MEMORY TEST
2941 *FLOAT A 0 THROUGH ALL MAIN MEMORY LOCATIONS
2942 *****
2943
2944
2945 ***** TEST 8 *****
2946 *IOP MAIN MEMORY DUAL ADDRESSING TEST
2947 *LOAD EACH MEMORY LOCATION WITH ITS OWN ADDRESS
2948 *READ BACK EACH LOCATION TO VERIFY CORRECT ADDRESSING
2949 *****
2950
2951
2952 ***** TEST 9 *****
2953 *IOP MAR TEST
2954 *PERFORM DUAL ADDRESSING TEST
2955 *USING MAR AUTO-INC FEATURE
2956 *****
2957

2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013

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

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

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

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

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

3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069

***** TEST 15 *****
*CRAM TEST OF JUMP(I) ON BR0 SET MICRO-PROCESSOR INSTRUCTION.
*SET THE BR0 BIT, PERFORM THE JUMP INSTRUCTION.
*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
*THEN THE PORT4 WILL CONTAIN A 37

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

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

***** TEST 18 *****
*CRAM TEST OF JUMP(I) ON BR7 SET MICRO-PROCESSOR INSTRUCTION.
*SET THE BR7 BIT, PERFORM THE JUMP INSTRUCTION
*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
*THEN PORT4 WILL CONTAIN A 37

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

3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125

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

***** TEST 20 *****
*CRAM TEST OF JUMP(1) ON Z BIT CLEAR MICRO-PROCESSOR INSTRUCTION.
*CLEAR THE Z BIT, PERFORM THE JUMP INSTRUCTION.
*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
*THEN PORT4 WILL CONTAIN A 37

***** TEST 21 *****
*CRAM TEST OF JUMP(1) ON BRO CLEAR MICRO-PROCESSOR INSTRUCTION.
*CLEAR THE BRO BIT, PERFORM THE JUMP INSTRUCTION.
*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
*THEN PORT4 WILL CONTAIN A 37

***** TEST 22 *****
*CRAM TEST OF JUMP(1) ON BR1 CLEAR MICRO-PROCESSOR INSTRUCTION.
*CLEAR THE BR1 BIT, PERFORM THE JUMP INSTRUCTION.
*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
*THEN PORT4 WILL CONTAIN A 37

***** TEST 23 *****
*CRAM TEST OF JUMP(1) ON BR4 CLEAR MICRO-PROCESSOR INSTRUCTION.
*CLEAR THE BR4 BIT, PERFORM THE JUMP INSTRUCTION.
*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
*THE CRAM PC IS CORRECT, IF THE CRAM PC IS NOT RIGHT
*THEN PORT4 CONTAINS A 37

***** TEST 24 *****
*CRAM TEST OF JUMP(1) ON BR7 CLEAR MICRO-PROCESSOR INSTRUCTION.
*CLEAR THE BR7 BIT, PERFORM THE JUMP INSTRUCTION.

3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181

*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
*THE CRAM PC IS CORRECT, IF THE CRAM PC IS NOT RIGHT
*THEN PORT4 CONTAINS A 37

***** TEST 25 *****

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

***** TEST 26 *****

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

***** TEST 27 *****

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

3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237

```
*          3          3
*          10         7
*          11         11
*          :10
*          17         377
*****
```

***** TEST 28 *****

* JUMP TEST, JUMP ALWAYS, JUMP CHANGE FIELD

* IN THIS TEST, WE WILL CHECK THE ABILITY OF THE
* MICRO-PROCESSOR TO JUMP (BRANCH AND ALWAYS INSTRUCTION)
* TO LOCATIONS, FIELDS FROM OTHER LOCATIONS FIELDS.
* WE ALREADY KNOW THAT THE BRANCH INSTR WORKS FROM
* OTHER TESTS. PROCEDURE:

- * 1. START ADDR 0, FIELD 0
- * 2. **CALCULATE NEW ADDR, FIELD VIA DEC.
- * 3. CAUSE JUMP (BRANCH) TO NEW ADDRESS
- * 4. READ PC FROM IBUS*12 AND IBUS*13
- * 5. REPEAT STEP 2-4 256.TIMES

* TO CALCULATE NEW ADDRESS:
* 1. DEC LOW BYTE OF ADDRESS FOR PC ADDRESS 0-7
* 2. DEC LOW BYTE OF N ADDRESS FOR PC ADDRESS 8-11
* BITS REPRESENTED AS BITS 0-3. WHEN 0-3 OVERFLOWS,
* RESTARTS AT ZERO

* NET RESULT IS JUMPS FROM:
* FIELD,PAGE LOC:
* 0 0
* 17 377
* 16 376
* 15 375
* :10 :
* 00 000

***** TEST 29 *****

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

***** TEST 30 *****

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

3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293

***** TEST 31 *****

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

***** TEST 32 *****

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

***** TEST 33 *****

*MICRO-PROCESSOR NOISE TEST
*WRITE ALL ZERO'S THEN ALL ONE'S THEN A DATA PATTERN
TO THE IBUS AND IBUS REGISTERS AND TO THE SP AND MAIN MEM
*THEN GO BACK AND READ THE DATA PATTERNS TO VERIFY THAT
*READING AND WRITING OF OTHER LOCATIONS AND REGISTERS
*DID NOT CHANGE THE DATA.

***** TEST 34 *****

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

***** TEST 35 *****

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

***** TEST 36 *****

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

***** TEST 37 *****

*
*NEGATIVE ADDRESS TEST.
* IN THIS TEST, WE'LL MAKE SURE THAT THE M8207

3294
3295
3296
3297
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
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349

* DOES NOT RESPOND TO AN ADDRESS THAT ISN'T ASSIGNED
* TO IT
*

***** TEST 38 *****

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

***** TEST 39 *****

*
* IN THIS TEST WE'RE GOING TO MAKE SURE THAT THE PC
* REG COUNTS UP PROPERLY. THE PC REG SHOULD INCREMENT
* ONCE AFTER EACH INSTRUCTION.
*

***** TEST 40 *****

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

***** TEST 41 *****

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

***** TEST 42 *****

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

3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405

***** TEST 43 *****
*TEST THAT MASTER CLEAR, CLEARS BITS IN THE NPR CONTROL REGISTER AND
*MICROPROCESSOR MISCELLANEOUS REGISTER-FIRST WE'LL SET THE
*PRIORITY UP SO THAT WHEN WE SET THE BUS REQUEST BIT THAT IT WON'T BUG US
*THEN WE'LL SET ALL THE BITS IN BOTH REGS EXCEPT THE
*NPR REQUEST. WE'LL LOOK TO SEE THAT ALL GOT SET, NEXT
*WE'LL DO A MASTER CLEAR AND BE SURE THAT THEY ALL CLEAR.

8.0 ERROR INFORMATION

8.1 ERROR REPORTING

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

THE FOLLOWING EXAMPLES PROVIDE TYPICAL ERROR REPORTS:

CZDMQ DVC FTL ERR 00045 TST 027 SUB 000 PC:022572

MASTER CLEAR FAILED TO CLEAR PC REG, CONTENTS-000624
CZDMQ DVC FTL ERR 00015 TST 042 SUB 000 PC:027234

UNIT=00, FAILING UNIT ADDRESS=160170
JUMP TEST ERROR
FROM ADDR TO ADDR BAD ADDR
000402 000000 000114

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

9.0 HISTORY

- MODIFIED AUGUST 1980 FOR THE FOLLOWING REASONS:

- 1) CANCEL DEPO CZDMQA1
- 2) CANCEL DEPO CZDMQA2
- 3) DETECT BAD TIMING ON INTERNAL CLOCK.

CZDMQC M8207 STATIC DIAG. #2 MACY11 30A(1052) 21-JUL-81 14:48 ^{K 2} PAGE 4-20
CZDMQC.P11 21-JUL-81 14:36 PROGRAM DOCUMENT

SEQ 0023

3406
3407
3408
3409
3410

- MODIFIED JULY 1981 TO FIX TEST 43 MAR BITS IN IBUS* 10.

a

CZDMQC M8207 STATIC DIAG. #2 MACY11 30A(1052) 21-JUL-81 14:48 L 2
CZDMQC.P11 21-JUL-81 14:36 PROGRAM DOCUMENT PAGE 5

SEQ 0024

3412
3413
3414
3415
3416

3418
 3426 002000
 3427
 3428
 3429
 3430
 3431
 3432
 3433
 3434 002000
 3435
 3436
 3437
 3438
 3439
 3440 002000
 3441
 3442
 3443 000000
 3444 000000
 3445 000000
 3446 000000
 3447 000000
 3448 000000
 3449 000000
 3450
 3451
 3452
 3453
 3454
 3455
 3456

.TITLE CZDMQCO M8207 STATIC DIAG #2
 .=2000

.MCALL SVC
 SVC

; INITIALIZE SUPERVISOR MACROS

BGNMOD CZDMQ

\$LSTIN= 0
 \$LSTTAG= 0
 SVCINS= 0 ; LIST INSTRUCTIONS, SHIFTED RIGHT
 SVCTST= 0 ; LIST TEST TAGS, SHIFTED RIGHT
 SVCSUB= 0 ; LIST SUBTEST TAGS, SHIFTED RIGHT
 SVLGBL= 0 ; LIST GLOBAL TAGS, SHIFTED RIGHT
 SVCTAG= 0 ; LIST OTHER TAGS, SHIFTED RIGHT

: CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
 : TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS. CHANGE THE
 : SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS. YOU MAY
 : CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.

3458
3459
3460
3461
3462
3463
3464 002000
3465
3473
3474 002000
(4) 002000
(4) 002000 103
(4) 002001 132
(4) 002002 104
(4) 002003 115
(4) 002004 121
(6) 002005 000
(6) 002006 000
(5) 002007 000
(5) 002010
(4) 002010 103
(5) 002011
(4) 002011 060
(5) 002012
(4) 002012 000000
(5) 002014
(4) 002014 000360
(5) 002016
(4) 002016 027344
(5) 002020
(4) 002020 000000
(5) 002022
(4) 002022 002262
(5) 002024
(4) 002024 000000
(5) 002026
(4) 002026 030144
(5) 002030
(4) 002030 000000
(5) 002032
(4) 002032 000000
(5) 002034
(4) 002034 000000
(5) 002036
(4) 002036 000000
(5) 002040
(4) 002040 002132
(5) 002042
(4) 002042 000000
(5) 002044
(4) 002044 000000
(5) 002046
(4) 002046 000000
(5) 002050
(4) 002050 003
(3) 002051 003

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

        POINTER BGNAU,BGNDU

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

(5)	002052		L\$EF::		;DIAG. EVENT FLAGS
(4)	002052	000000		.WORD 0	
(5)	002054	000000		.WORD 0	
(5)	002056		L\$SPC::		
(4)	002056	000000		.WORD 0	
(5)	002060		L\$DEVP::		; POINTER TO DEVICE TYPE LIST
(4)	002060	002730		.WORD L\$DVTYP	
(5)	002062		L\$REPP::		;PTR. TO REPORT CODE
(4)	002062	000000		.WORD 0	
(5)	002064		L\$EXP4::		
(4)	002064	000000		.WORD 0	
(5)	002066		L\$EXP5::		
(4)	002066	000000		.WORD 0	
(5)	002070		L\$AUT::		;PTR. TO ADD UNIT CODE
(4)	002070	012144		.WORD L\$AU	
(5)	002072		L\$DUT::		;PTR. TO DROP UNIT CODE
(4)	002072	012140		.WORD L\$DU	
(5)	002074		L\$LUN::		;LUN FOR EXERCISERS TO FILL
(4)	002074	000000		.WORD 0	
(5)	002076		L\$DESP::		;POINTER TO DIAG. DESCRIPTION
(4)	002076	002312		.WORD L\$DESC	
(5)	002100		L\$LOAD::		;GENERATE SPECIAL AUTOLOAD EMT
(4)	002100	104035		EMT E\$LOAD	
(5)	002102		L\$ETP::		;POINTER TO ERRTABL
(4)	002102	000000		.WORD 0	
(5)	002104		L\$IICP::		;PTR. TO INIT CODE
(4)	002104	011340		.WORD L\$INIT	
(5)	002106		L\$CCP::		;PTR. TO CLEAN-UP CODE
(4)	002106	012134		.WORD L\$CLEAN	
(5)	002110		L\$ACP::		;PTR. TO AUTO CODE
(4)	002110	012042		.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	
3475					
3476					
3482	002122			BGNPROT	
(3)	002122		L\$PROT::		
3483	002122	177777		.WORD -1	
3484	002124	177777		.WORD -1	
3485	002126	177777		.WORD -1	
3486	002130			ENDPROT	
3487					

3489
3490
3491
3492
3493
3494
3495
3496 002130
(4) 002130 000053
(3) 002132
(6) 002132 012146
(6) 002134 012256
(6) 002136 012422
(6) 002140 012552
(6) 002142 012712
(6) 002144 013154
(6) 002146 013356
(6) 002150 013570
(6) 002152 014112
(6) 002154 014470
(6) 002156 014736
(6) 002160 015202
(6) 002162 015432
(6) 002164 015676
(6) 002166 016142
(6) 002170 016406
(6) 002172 016652
(6) 002174 017116
(6) 002176 017362
(6) 002200 017642
(6) 002202 020122
(6) 002204 020376
(6) 002206 020654
(6) 002210 021130
(6) 002212 021404
(6) 002214 021576
(6) 002216 021744
(6) 002220 022322
(6) 002222 022560
(6) 002224 022774
(6) 002226 023210
(6) 002230 023452
(6) 002232 024044
(6) 002234 025116
(6) 002236 025216
(6) 002240 025364
(6) 002242 026042
(6) 002244 026202
(6) 002246 026310
(6) 002250 026446
(6) 002252 026544
(6) 002254 026644
(6) 002256 026770
3497
3504
3505

.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 43
.WORD 43
L\$DISPATCH::
.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

CZDMQCO M8207 STATIC DIAG #2 MACY11 30A(1052) 21-JUL-81 14:48 D 3
CZDMQC.P11 21-JUL-81 14:36 DISPATCH TABLE PAGE 5-5

SEQ 0029

3506
3507
3508

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 PAGE 5-6
DEFAULT HARDWARE P-TABLE

SEQ 0030

3510
3511
3512
3513
3514
3515
3516
3517
3518
3519 002260
(3) 002260 000013
(3) 002262
(3) 002262
3520 002262 000007
3521 002264 160170
3522 002266 000300
3523 002270 005000
3524 002272 000003
3525 002274 000056
3526 002276 000000
3527 002300 000000
3528 002302 000000
3529 002304 000004
3530
3531
3532 002306 000000
3533
3534 002310
(3) 002310

.SBTTL DEFAULT HARDWARE P-TABLE

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

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

L\$HW::
DFPTBL::

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

ENDHW
L10001:

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 PAGE 5-7
SOFTWARE P-TABLE

SEQ 0031

3536
3537
3538
3539
3540
3541
3542
3543 002310
(3) 002310 000000
(3) 002312
(3) 002312
3544
3545
3546 002312
(3) 002312
3547
3548
3549
3550
3551
3552

```
.SBTTL SOFTWARE P-TABLE

:////////////////////////
:/ THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
:/ PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
:////////////////////////

          BGNSW  SFPTBL
          .WORD  L10002-L$SW/2

L$SW::
SFPTBL::

          ENDSW

L10002:
```

[illegible]

OC2312

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

001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

SUBTTL GLOBAL EQUATES SECTION

[illegible]

EQUALS

: BIT DEFINITIONS

```

BIT15== 100000
BIT14== 40000
BIT13== 20000
BIT12== 10000
BIT11== 4000
BIT10== 2000
BIT09== 1000
BIT08== 400
BIT07== 200
BIT06== 100
BIT05== 40
BIT04== 20
BIT03== 10
BIT02== 4
BIT01== 2
BIT00== 1

```

8119==	81109
8118==	81108
8117==	81107
8116==	81106
8115==	81105
8114==	81104
8113==	81103
8112==	81102
8111==	81101
8110==	81100

```

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

```

```
EF.START--      32.      ; START COMMAND WAS ISSUED
EF.RESTART      31.      ; RESTART COMMAND WAS ISSUED
```

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

(1)	000036	EF.CONTINUE==	30.
(1)	000035	EF.NEW==	29.
(1)	000034	EF.PWR==	28.
(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	FNT==	1000
(1)	002000	PRI==	2000
(1)	004000	IXE==	4000
(1)	010000	IBE==	10000
(1)	020000	IER==	20000
(1)	040000	LOE==	40000
(1)	100000	HOE==	100000

3574
3575
3576
3577
3578
3579
3580
3581
3582
3583

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

```

3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595 002312
      (4) 002312
      (3) 002312 034115 030062 020067
      (3) 002320 044504 043501 020056
      (3) 002326 031043 047440 020106
      (3) 002334 000062
      (2)
3596
3597
3598
3599
3600 002336 000000
3601 002340 000000
3602
3603
3604
3605
3606 002342 000000
3607 002344 000000
3608 002346 000000
3609 002350 000000
3610 002352 000000
3611 002354 000000
3612 002356 000000
3613 002360 000000
3614 002362 000000
3615 002364 000000
3616 002366 000000
3617 002370 000000
3618 002372 000001
3619 002374 000000
3620 002376 000001
3621 002400 000001
3622 002402 000001
3623 002404 000001
3624 002406 000000
3625 002410 000000
3626 002412 000000
3627 002414 000000
3628 002416 000000
3629 002420 000000
3630 002422 000000
3631 002424 000000
3632 002426 000000
3633 002430 000000
3634 002432 000000

```

```

.SBTTL GLOBAL DATA SECTION

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

:*****
:* STORAGE FOR DEVICE REGISTERS
:*****
      DESCRIPT      <M8207 DIAG. #2 OF 2>
L$DESC::
      .ASCIIZ /M8207 DIAG. #2 OF 2/

      .EVEN

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

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

```

3635 002434 000000
 3636 002436 003777
 3637 002440 000000
 3638 002442 000000
 3639 002444 000000
 3640 002446 000000
 3641 002450 000000
 3642 002452 000000
 3643 002454 000000
 3644 002456 000000
 3645 002460 000000
 3646 002462 000000
 3647 002464 000000
 3648 002466 000000
 3649 002470 000000
 3650 002472 000000

MRO: .WORD 0 ;MEMLOC USED INSTEAD OF RO.
 MEMSZ: .WORD 3777 ;INDICATES MEMORIE SIZE, LAST ADDR.
 TEMP: .WORD 0
 \$TEMPO: .WORD 0
 \$TMP0: .WORD 0
 \$GDADR: .WORD 0 ;CONTAINS ADDRESS OF 'GOOD' DATA
 \$BDADR: .WORD 0 ;CONTAINS ADDRESS OF 'BAD' DATA
 \$GDDAT: .WORD 0 ;CONTAINS 'GOOD' DATA
 \$BDDAT: .WORD 0 ;CONTAINS 'BAD' DATA
 .WORD 0 ;RESERVED--NOT TO BE USED
 .WORD 0
 FTIME: .WORD 0
 SAVE4: .WORD 0
 SAVE6: .WORD 0
 RUNB: .WORD 0 ;0= RUN OFF, 1= RUN SW ON
 RUNINH: .WORD 0 ;0=RUN SW OFF, 1=RUN SW ON

3651
 3652
 3653
 3654
 3655 002474 000
 3656 002476 000
 3657 002476 000
 3658 002477 000
 3659
 3660
 3661
 3662
 3663
 3664
 3665
 3666
 3667
 3668
 3669
 3670
 3671
 3672
 3673
 3674
 3675
 3676
 3677
 3678
 3679
 3680 002500 000000
 3681 002502 000000
 3682 002504 000000
 3683
 3684
 3685
 3686
 3687 002506 000000
 3688 002510 000000
 3689 002512 000000
 3690 002514 000000

 ;* 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

 ;* 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
 STAT3: .WORD 0

 ;* POINTERS TO M8200,4,7 VECTORS AND REGISTERS

 KMRVEC: 0 ;POINTER TO M8200,4,7 RCV INTRPT VECTOR
 KMRLVL: 0 ;POINTER TO M8200,4,7 RCV INTRPT SERVICE PS
 KMTVEC: 0 ;POINTER TO M8200,4,7 TX INTRPT VECTOR
 KMTLVL: 0 ;POINTER TO M8200,4,7 TX INTRPT SERVICE PS

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 K 3 PAGE 5-12
GLOBAL DATA SECTION

SEQ 0036

3691	002516	000000	KMCSR:	0	;POINTER TO M8200,4,7 CONTROL STATUS REGISTER
3692	002520	000000	KMCSRH:	0	;POINTER TO M8200,4,7 CONTROL STATUS REGISTER HIGH BYTE
3693	002522	000000	KMCTL:	0	;POINTER TO M8200,4,7 CONTROL OUT REGISTER
3694	002524	000000	KMPO4:	0	;POINTER TO M8200,4,7 PORT REGISTER - SEL4
3695	002526	000000	KMPO6:	0	;POINTER TO M8200,4,7 PORT REGISTER - SEL6
3696					
3697			;:**** PRIMARY REG ADRS STORAGE FOR THIS UNIT ****		
3698			;THESE LOCATIONS WILL BE LOADED FOR THE CURRENT UNIT, IN INIT CODE		
3699	002530		REGADR:		
3700					
3701			;:**** STACK USED FOR SUBROUTINE LINKAGE ****		
3702	002530	000100			
3703	002730		SSTACK:	.BLKW 100	
3704					
3705					
3706					
3707					
3708					
3709					
3710					

```
3712 .SBTTL GLOBAL TEXT SECTION
3713
3714 :XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
3715 :% THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
3716 :% MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
3717 :% MORE THAN ONE TEST.
3718 :XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
3719
3720 :*****
3721 :* NAMES OF DEVICES SUPPORTED BY PROGRAM
3722 :*****
3723 DEVTYP <M8200,M8204,OR M8207>
3724 (4) 002730
3725 (3) 002730 034115 030062 026060
3726 (3) 002736 034115 030062 026064
3727 (3) 002744 051117 046440 031070
3728 (3) 002752 033460 000
3729 (2) 002756
3730
3731 .EVEN
3732
3733 :
3734 : FORMAT STATEMENTS USED IN PRINT CALLS
3735 :
3736
3737
3738
3739
3740
```

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 PAGE 5-14
GLOBAL SUBROUTINES

SEQ 0038

3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756

.SBTTL GLOBAL SUBROUTINES

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

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

.MACRO POPSP2
22626
.ENDM

CZDMQCO M8207 STATIC DIAG #2 MACY11 30A(1052) 21-JUL-81 14:48 N 3
CZDMQC.P11 21-JUL-81 14:36 GLOBAL SUBROUTINES PAGE 6

SEQ 0039

3758
3759
3760
3761
3762

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 PAGE 6-1
GLOBAL SUBROUTINES

SEQ 0040

3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789

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

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

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

```

3791 .MACRO MYINT
3792 .LIST
3793 MOV KMCSR,R1 ;RECORD DEVICE ADDR.
3794 .NLIST
3795 .ENDM
3796
3797 .MACRO MACEX ?N2
3798 .LIST
3799 ;DO NOT DO TEST IF M8200
3800 .NLIST
3801 TST TYPE
3802 BNE N2
3803 EXIT TST
3804 N2:
3805 .ENDM
3806 .MACRO MACEX2 ?N2
3807 .LIST
3808 ;DO NOT DO TEST IF M8200
3809 .NLIST
3810 CMP WTYPE,#0
3811 BNE N2
3812 EXIT TST
3813 N2:
3814 .ENDM
3815 .MACRO K4ONLY ?N2
3816 .LIST
3817 ;DO NOT DO TEST IF M8200, OR M8204
3818 .NLIST
3819 CMP MEMSZ,#2000
3820 BNE N2
3821 EXIT TST
3822 N2:
3823 ;NOTE THIS TEST IS ONLY DESIGNED FOR 4K MODULE.
3824 .FNDM
3825
3826 .MACRO CLRMAR
3827 ROMCLK
3828 004000
3829 .ENDM
3830 .MACRO ROMCLK
3831 .LIST
3832 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
3833 .NLIST
3834 .ENDM
3835
3836 .MACRO SROMCLK
3837 .LIST
3838 JSR R5,.SROMCLK
3839 .NLIST
3840 .ENDM
3841 .MACRO SKIP06 NNN
3842 .LIST
3843 ;GOTO 'NNN' IF M8206
3844 .NLIST
3845 CMP WTYPE,#6 ;SEE IF M8206
3846 BEQ NNN

```

```
3847 .ENDM
3848 .MACRO SKIP07 NNN
3849 .LIST
3850 ;GOTO 'NNN' IF M8207
3851 .NLIST
3852 CMP WTYPE,#7 ;SEE IF M8200,4,7
3853 BEQ NNN
3854 .ENDM
3855 .MACRO SKIP04 NNN
3856 .LIST
3857 ;GOTO 'NNN' IF M8204
3858 .NLIST
3859 CMP WTYPE,#4 ;SEE IF M8204
3860 BEQ NNN
3861 .ENDM
3862 .MACRO MSTCLR
3863 JSR R5,,MSTCLR ;CLEAR M8200,4,7
3864 .ENDM
3865
3866 002756 .MSTCLR:
3867 002756 112777 000100 177534 MCVB #BIT6,@KMCSRH ;SET INST.
3868 002764 142777 000300 177526 BICB #BIT6,BIT7,@KMCSRH
3869 002772 000205 RTS R5
3870
3871 002774 000024 PATCH: .BLKW 20. ;PATCH AREA.
3872
3873
3874
3875 003044 ENDBUG:
3876 ; UNSAFE TO PATCH ANY OTHER AREA.
3877 003044 .ROMCLK:
3878 003044 000240 NOP
3879 003046 000240 NOP
3880 003050 152777 000002 177442 .REGT: BISB #BIT1,@KMCSRH
3881 003056 012577 177444 MOV (R5)+,@KMPO6
3882 003062 152777 000003 177430 BISB #BIT1!BIT0,@KMCSRH
3883 003070 142777 000007 177422 BICB #BIT2!BIT1!BIT0,@KMCSRH
3884 003076 000205 RTS R5
3885
3886 003100 .SRMCLK:
3887 003100 000240 NOP
3888 003102 022737 000006 002414 CMP #6,WTYPE
3889 003110 001357 .REGT
3890 003112 152777 000002 177400 BISB #BIT1,@KMCSRH
3891 003120 012577 177402 MOV (R5)+,@KMPO6
3892 003124 000240 NOP
3893 003126 000240 NOP
3894 003130 142777 000007 177362 BICB #7,@KMCSRH
3895 003136 1$:
3896 003136 152777 000001 177354 BISB #BIT0,@KMCSRH ;STEP INSTR.
3897 003144 142777 000007 177346 BICB #BIT2!BIT1!BIT0,@KMCSRH
3898 003152 000240 NOP
3899 003154 000240 NOP
3900 003156 152777 000002 177334 2$: BISB #2,@KMCSRH
3901 003164 RTS R5
3902 003164 000205
```

3903 003166
3904
3905 003166
(1) 003166 004537 003044
3906 003172 000400
3907 003174
(1) 003174 004537 003044
3908 003200 063220
3909 003202
(1) 003202 004537 003044
3910 003206 060400
3911 003210
(1) 003210 004537 003100
3912 003214 000000
3913 003216 000207
3914
3915 003220
3916
3917 003220
(1) 003220 004537 003044
3918 003224 000401
3919 003226 000207
3920
3921 003230
3922
3923
3924 003230
(1) 003230 004537 003044
3925 003234 000402
3926 003236 000207
3927
3928 003240
3929
3930
3931 003240
(1) 003240 004537 003044
3932 003244 000420
3933 003246 000207
3934
3935 003250
3936
3937
3938 003250
(1) 003250 004537 003044
3939 003254 000600
3940 003256 000207
3941
3942 003260
3943
3944
3945 003260
(1) 003260 004537 003044
3946 003264 000777
3947 003266
(1) 003266 004537 003044
3948 003272 063220

CLRALL:

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

SETBR0:

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

SETBR1:

;THIS SUBROUTINE SETS BR1 BIT

ROMCLK

;NEXT WORD IS INSTRUCTION
JSR R5,.ROMCLK ;CLOCK INSTRUCTION
000402 ;BR_002
RTS PC

SETBR4:

;THIS SUBROUTINE SETS BR4 BIT

ROMCLK ;NEXT WORD IS INSTRUCTION
JSR R5,.ROMCLK ;CLOCK INSTRUCTION
420
RTS PC

SETBR7:

;THIS SUBROUTINE SETS BR7 BIT

ROMCLK ;NEXT WORD IS INSTRUCTION
JSR R5,.ROMCLK ;CLOCK INSTRUCTION
600
RTS PC

SETC:

;THIS SUBROUTINE SETS THE C BIT

ROMCLK ;NEXT WORD IS INSTRUCTION
JSR R5,.ROMCLK ;CLOCK INSTRUCTION
000777 ;BR 377
ROMCLK ;NEXT WORD IS INSTRUCTION
JSR R5,.ROMCLK ;CLOCK INSTRUCTION
063220 ;SP(0)_BR

```

3949 003274 ROMCLK ;NEXT WORD IS INSTRUCTION
(1) 003274 004537 003044 JSR R5,,ROMCLK ;CLOCK INSTRUCTION
3950 003300 060400 060400 ;BR SP(0)+BR
3951 003302 SROMCLK ;NOW WE MUST CLOCK THE BITS INTO IBUS <13>
(1) 003302 004537 003100 JSR R5,,SROMCLK
3952 003306 000000 0
3953 003310 000207 RTS PC
3954
3955 003312 SETZ: ;THIS SUBROUTINE SETS THE Z BIT
3956
3957
3958 003312 ROMCLK ;NEXT WORD IS INSTRUCTION
(1) 003312 004537 003044 JSR R5,,ROMCLK ;CLOCK INSTRUCTION
3959 003316 000777 000777 ;BR 377
3960 003320 SROMCLK ;NOW CLOCK THE BITS INTO IBUS<13>
(1) 003320 004537 003100 JSR R5,,SROMCLK
3961 003324 000777 0777
3962 003326 000207 RTS PC
3963
3964 003330 PAMDAT: ;THIS SUBROUTINE LOADS R4 WITH THE LOWEST
3965 ;8 BITS OF THE CRAM PC.
3966
3967
3968 003330 005004 CLR R4
3969 003332 017605 000000 MOV @SP,R5 ;GOOD DATA
3970 003336 062716 000002 ADD #2,(SP) ;ADJUST STACK
3971 003342 SKIP06 1$ ;IF M8206,WE'LL GET PC A DIFFERENT WAY.
(1) ;GOTO 1$ IF M8206
3972 003352 SKIP07 1$ ;IF M8200,4,7 WE'LL GET PC A DIFFERENT WAY.
(1) ;GOTO 1$ IF M8207
3973 003362 005011 CLR (R1) ;CLEAR BIT10
3974 003364 052711 000400 BIS #BIT8,(R1) ;CLOCK INSTRUCTION IN CRAM THAT
3975 ;JUMPED TO, IT LOADS BR WITH IT
3976 003370 005011 CLR (R1) ;CLR BIT8
3977 003372 ROMCLK ;NEXT WORD IS INSTRUCTION
(1) 003372 004537 003044 JSR R5,,ROMCLK ;CLOCK INSTRUCTION
3978 003376 061225 061225 ;MOV BR TO PORT 5
3979 003400 116104 000005 MOVB 5(R1),R4 ;PUT 'FOUND' IN R4
3980 003404 000207 RTS PC ;RETURN
3981
3982 003406 1$: ROMCLK ;READ PC LOW REG DIRECTLY.
(1) 003406 004537 003044 JSR R5,,ROMCLK ;CLOCK INSTRUCTION
3983 003412 121244 121244 ;IBUS<12> TO PORT 4
3984 003414 116104 000004 MOVB 4(R1),R4 ;PUT INTO R4
3985 003420 000207 RTS PC ;EXIT
3986
3987 003422 WROM: ;THIS SUBROUTINE WRITES THE ROMMAP INTO THE CRAM
3988
3989
3990 ; BIT #BIT15,STAT1 ;BE SURE M8200,4,7 HAS CRAM
3991 ; BEQ 2$ ;SKIP IF NO CRAM
3992 003422 SKIP07 2$
(1) ;GOTO 2$ IF M8207
3993 003432 005000 CLR R0 ;R0=CRAM ADDRESS
3994 003434 012702 012146 MOV #ROMMAP,R2 ;R2 POINTS TO ROMMAP
3995 003440 012711 002000 1$: MOV #BIT10,(R1) ;SET ROM0

```

```

3996 003444 010061 000004      MOV      R0,4(R1)      ;LOAD CRAM ADDRESS
3997 003450 012261 000006      MOV      (R2)+,6(R1)    ;LOAD WORD TO BE WRITTEN
3998 003454 052711 020000      BIS      #BIT13,(R1)    ;WRITE IT!
3999 003460 005200              INC      R0      ;NEXT ADDRESS
4000 003462 023700 002436      CMP      MEMSZ,R0      ;DONE YET?
4001 003466 001364              BNE      1$      ;BR IF NO
4002 003470 005011              CLR      (R1)      ;CLEAR SEL0
4003 003472 000207      2$:      RTS      PC      ;RETURN
4004
4005 003474      MEMSET:
4006      ;THIS SUBROUTINE LOADS CRAM WITH SPECIAL INSTRUCTIONS
4007      ;FOR THE CRAM JUMP TEST. ALL CRAM LOCATIONS ARE LOADED
4008      ;WITH INSTRUCTIONS THAT MOVE A 37 TO THE BR, EXCEPT THE
4009      ;FOLLOWING CRAM ADDRESSES: 0,1,4,7,525,1777. THESE LOCATIONS
4010      ;CONTAIN INSTRUCTIONS WHICH LOAD THE BR WITH THE LOWEST
4011      ;8 BITS OF THAT CRAM ADDRESS.
4012
4013 003474      SKIP07 3$      ;IF M8200,4,7 CAN'T WRITE CRAM.
      (1)      ;GOTO 3$ IF M8207
4014 003504 005000              CLR      R0      ;R0 = CRAM ADDRESS
4015 003506 012711 002000      1$:      MOV      #BIT10,(R1)    ;SET ROMO
4016 003512 010061 000004      MOV      R0,4(R1)    ;LOAD CRAM ADDRESS
4017 003516 012761 000437 000006      MOV      #437,6(R1)    ;LOAD INSTRUCTION
4018 003524 052711 020000      BIS      #BIT13,(R1)    ;WRITE INSTRUCTION IN CRAM
4019 003530 005200              INC      R0      ;NEXT ADDRESS
4020 003532 023700 002436      CMP      MEMSZ,R0      ;DONE YET?
4021 003536 001363              BNE      1$      ;BR IF NO
4022 003540 005000              CLR      R0      ;INDEX REGISTER
4023 003542 012711 002000      2$:      MOV      #BIT10,(R1)    ;SET ROMO
4024 003546 016061 003602 000004      MOV      (CRAM(R0),4(R1) ;LOAD CRAM ADDRESS IN SEL4
4025 003554 016061 003616 000006      MOV      (INSTU(R0),6(R1) ;LOAD INSTRUCTION TO BE WRITTEN
4026 003562 052711 020000      BIS      #BIT13,(R1)    ;WRITE CRAM!
4027 003566 005720              TST      (R0)+      ;NEXT
4028 003570 022700 000014      CMP      #4,R0      ;DONE YET?
4029 003574 001362              BNE      2$      ;BR IF NO
4030 003576 005011              CLR      (R1)      ;CLEAR ALL BITS
4031 003600 000207      3$:      RTS      PC      ;RETURN
4032
4033 003602 000000 000001 000004      CRAMA- .WORD 0,1,4,7,1777,525
      003610 000007 001777 000525
4034
4035 003616 000400      INSTU: 000400      ;BR_0
4036 003620 000401      000401      ;BR_1
4037 003622 000404      000404      ;BR_4
4038 003624 000407      000407      ;BR_7
4039 003626 000777      000777      ;BR_377
4040 003630 000525      000525      ;BR_125
4041
4042
4043      ;ROUTINE TO SAVE GENERAL REGISTERS FOR ERROR ROUTINE.
4044      ;CALL JSR PC,SV05
4045 003632 010537 002416      SV05: MOV      R5,$REG5
4046 003636 010437 002420      MOV      R4,$REG4
4047 003642 010337 002422      MOV      R3,$REG3
4048 003646 010237 002424      MOV      R2,$REG2
4049 003652 010137 002426      MOV      R1,$REG1

```

CZDMO CO M8207 STATIC DIAG #2 MACV11 30A(1052) 21-JUL-81 14:48 H 4
CZDMO C.P11 21-JUL-81 14:36 GLOBAL SUBROUTINES PAGE 7-5

SEQ 0046

4050	003656	013737	002434	002430	MOV	MRO, SREG0
4051	003664	000207			RTS	PC
4052						
4053						

4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065

.SBT'L GLOBAL ERROR REPORT SECTION

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

4066	003666	047045	047445	022466	TFM1:	.ASCIIZ	/%N%06%S4%06%S4%04%N/
	003674	032123	047445	022466			
	003702	032123	047445	022464			
	003710	000116					
4067	003712	047045	047445	022463	TFM2:	.ASCIIZ	/%N%03%S7%03%N/
	003720	033523	047445	022463			
	003726	000116					
4068	003730	047045	047445	022463	TFM3:	.ASCIIZ	/%N%03%S10%03%S4%04%N/
	003736	030523	022460	031517			
	003744	051445	022464	032117			
	003752	047045	000				
4069	003755	045	022516	031517	TFM4:	.ASCIIZ	/%N%03%S7%03%N/
	003762	051445	022467	031517			
	003770	047045	000				
4070	003773	045	022516	033117	TFM5:	.ASCIIZ	/%N%06%S5%06%S3%06%N/
	004000	051445	022465	033117			
	004006	051445	022463	033117			
	004014	047045	000				
4071	004017	045	022516	051101	TFM36:	.ASCIIZ	/%N%AREGISTER ADDRESS ERROR,ADDRESS = %06%A,UNIT - %02/
	004024	043505	051511	042524			
	004032	020122	042101	051104			
	004040	051505	020123	051105			
	004046	047522	026122	042101			
	004054	051104	051505	020123			
	004062	020075	047445	022466			
	004070	026101	047125	052111			
	004076	036440	022440	031117			
	004104	000					
4072	004105	045	022516	020101	TFM41:	.ASCIIZ	/%N%A CSR HIGH BYTE GOT WRITTEN INTO ON A LOW BYTE XFER/
	004112	051503	020122	044510			
	004120	044107	041040	052131			
	004126	020105	047507	020124			
	004134	051127	052111	042524			
	004142	020116	047111	047524			
	004150	047440	020116	020101			
	004156	047514	020127	054502			
	004164	042524	054040	042506			
	004172	000122					
4073	004174	047045	040445	041440	TFM42:	.ASCIIZ	/%N%A CSR LOW BYTE GOT WRITTEN INTO ON A HIGH BYTE XFER/
	004202	051123	046040	053517			
	004210	041040	052131	020105			
	004216	047507	020124	051127			
	004224	052111	042524	020116			
	004232	047111	047524	047440			
	004240	020116	020101	044510			

	004246	044107	041040	052131	
	004254	020105	043130	051105	
	004262	000			
4074	004263	045	022516	047101	TFM40: .ASCIZ /%N%ANEG ADDR TEST DUAL ADDR ERROR-BAD ADDR = %06/
	004270	043505	040440	042104	
	004276	020122	042524	052123	
	004304	042040	040525	020114	
	004312	042101	051104	042440	
	004320	051122	051117	041055	
	004326	042101	040440	042104	
	004334	020122	020075	047445	
	004342	000066			
4075	004344	040445	051440	051103	TFM43: .ASCIZ /%A SCRATCH PAD %03%A DUAL ADDRESS ERROR WITH SP%02/
	004352	052101	044103	050040	
	004360	042101	022440	031517	
	004366	040445	042040	040525	
	004374	020114	042101	051104	
	004402	051505	020123	051105	
	004410	047522	020122	044527	
	004416	044124	051440	022520	
	004424	031117	000		
4076	004427	045	022524	052101	TFM44: .ASCIZ /%T%ATHE MAR REG, CONTENTS- %06/
	004434	042510	046440	051101	
	004442	051040	043505	020054	
	004450	047503	052116	047105	
	004456	051524	020075	047445	
	004464	000066			
4077	004466	052045	040445	044124	TFM45: .ASCIZ /%T%ATHE PC REG, CONTENTS= %06/
	004474	020105	041520	051040	
	004502	043505	020054	047503	
	004510	052116	047105	051524	
	004516	020075	047445	000066	
4078	004524	047045	040445	047516	TFM45A: .ASCII /%N%ANOTE: THIS ERROR MAY BE FALSELY GENERATED IF THE/
	004532	042524	020072	044124	
	004540	051511	042440	051122	
	004546	051117	046440	054501	
	004554	041040	020105	040506	
	004562	051514	046105	020131	
	004570	042507	042516	040522	
	004576	042524	020104	043111	
	004604	052040	042510		
4079	004610	047045	040445	052522	.ASCIZ /%N%ARJN BIT (SW7 OF E28) IS ON/
	004616	020116	044502	020124	
	004624	051450	033527	047440	
	004632	020106	031105	024470	
	004640	044440	020123	047117	
	004646	000			
4080	004647	045	047101	051120	TFM46: .ASCIZ '%ANPR/MISC REGS DATA FAILURE, GOOD =%06%A, BAD =%06''
	004654	046457	051511	020103	
	004662	042522	051507	042040	
	004670	052101	020101	040506	
	004676	046111	051125	026105	
	004704	043440	047517	020104	
	004712	022475	033117	040445	
	004720	020054	040502	020104	
	004726	022475	033117	000	

4081	004733	045	050101	020103	TFM47:	.ASCIZ	'%APC INCR. INCORRECT: S/B= %06%A ; WAS = %06''
	004740	047111	051103	020056			
	004746	047111	047503	051122			
	004754	041505	035124	051440			
	004762	041057	020075	047445			
	004770	022466	020101	020073			
	004776	040527	020123	020075			
	005004	047445	000066				
4082	005010	040515	052123	051105	TWMC:	.ASCIZ	/MASTER CLEAR FAILED TO CLEAR /
	005016	041440	042514	051101			
	005024	043040	044501	042514			
	005032	020104	047524	041440			
	005040	042514	051101	000040			
4083	005046	047045	052045	047045	FM1:	.ASCIZ	/NXTN/
	005054	000					
4084							
4085							
4086							
4087	005055	000			EM0:	.ASCIZ	//
4088	005056	051103	046501	042040	EM1:	.ASCIZ	/CRAM DATA ERROR/
	005064	052101	020101	051105			
	005072	047522	000122				
4089	005076	051103	046501	042040	EM2:	.ASCIZ	/CRAM DJAL ADDRESSING ERROR/
	005104	040525	020114	042101			
	005112	051104	051505	044523			
	005120	043516	042440	051122			
	005126	051117	000				
4090	005131	112	046525	020120	EM3:	.ASCIZ	/JUMP ERROR/
	005136	051105	047522	000122			
4091	005144	051103	046501	045040	EM4:	.ASCIZ	/CRAM JUMP TEST FAULT/
	005152	046525	020120	042524			
	005160	052123	043040	052501			
	005166	052114	000				
4092	005171	111	050117	046440	EM5:	.ASCIZ	/IOP MAIN MEMORY TEST/
	005176	044501	020116	042515			
	005204	047515	054522	052040			
	005212	051505	000124				
4093	005216	047511	020120	040515	EM6:	.ASCIZ	/IOP MAR TEST/
	005224	020122	042524	052123			
	005232	000					
4094	005233	102	020122	044522	EM7:	.ASCIZ	/BR RIGHT SHIFT ERROR/
	005240	044107	020124	044123			
	005246	043111	020124	051105			
	005254	047522	000122				
4095	005260	040515	020122	052504	EM10:	.ASCIZ	/MAR DUAL ADDRESSING ERROR/
	005266	046101	040440	042104			
	005274	042522	051523	047111			
	005302	020107	051105	047522			
	005310	000122					
4096	005312	052512	050115	043040	EM11:	.ASCIZ	/JUMP FIELD ERROR/
	005320	042511	042114	042440			
	005326	051122	051117	000			
4097	005333	112	046525	020120	EM12:	.ASCIZ	/JUMP TEST ERROR/
	005340	042524	052123	042440			
	005346	051122	051117	000			
4098	005353	103	047117	044504	EM16:	.ASCIZ	/CONDITION CODE TESTING,Z & C/

	005360	044524	047117	041440	
	005366	042117	020105	042524	
	005374	052123	047111	026107	
	005402	020132	020046	000103	
4099	005410	046103	041517	020113	EMB1: .ASCIZ /CLOCK TIME TOO FAST/
	005416	044524	042515	052040	
	005424	047517	043040	051501	
	005432	000124			
4100	005434				EM35:
4101	005434	047506	041522	020105	EM17: .ASCIZ /FORCE POWER FAIL ERROR/
	005442	047520	042527	020122	
	005450	040506	046111	042440	
	005456	051122	051117	000	
4102	005463	111	052502	025123	EM27: .ASCIZ ''IBUS* WRITE/READ ERROR''
	005470	053440	044522	042524	
	005476	051057	040505	020104	
	005504	051105	047522	000122	
4103					
4104	005512	041111	051525	047457	EM29: .ASCIZ 'IBUS/OBUS WRITE/READ ERROR'
	005520	052502	020123	051127	
	005526	052111	027505	042522	
	005534	042101	042440	051122	
	005542	051117	000		
4105					
4106	005545	120	046507	041440	EMB50: .ASCIZ 'PGM CLOCK WOULD NOT CLEAR'
	005552	047514	045503	053440	
	005560	052517	042114	047040	
	005566	052117	041440	042514	
	005574	051101	000		
4107	005577	120	046507	041440	EMB51: .ASCIZ 'PGM CLOCK WOULD NOT SET'
	005604	047514	045503	053440	
	005612	052517	042114	047040	
	005620	052117	051440	052105	
	005626	000			
4108	005627	045	022516	025101	STM: .ASCIZ '%N%A*****'
	005634	025052	025052	025052	
	005642	025052	025052	025052	
	005650	025052	025052	025052	
	005656	025052	025052	025052	
	005664	025052	025052	025052	
	005672	025052	025052	025052	
	005700	025052	025052	025052	
	005706	025052	025052	025052	
	005714	025052	025052	025052	
	005722	000			
4109	005723	000			DH0: .ASCIZ //
4110					
4111					
4112	005724	054105	042520	052103	DH1: .ASCIZ /EXPECTED FOUND ADDRESS/
	005732	042105	020040	047506	
	005740	047125	020104	040440	
	005746	042104	042527	051523	
	005754	000			
4113	005755	105	050130	041505	DH2: .ASCIZ /EXPECTED FOUND/
	005762	042524	020104	043040	
	005770	052517	042116	000	

4114 005775 106 047522 020115 DH3: .ASCIZ /FROM ADDR TO ADDR BAD ADDR/
 006002 042101 051104 020040
 006010 047524 040440 042104
 006016 020122 041040 042101
 006024 040440 042104 000122

4115
 4116
 4117 .EVEN
 4118
 4119

4120
 4121
 4122 :-----
 4123 : MACRO'S NEEDED TO REPORT ERRORS
 4124 :-----

4125 .MACRO MDT1
 4126 PRINTB #TFM1,\$REG2,\$REG4,\$REG0
 4127 .ENDM

4128 .MACRO MDT2
 4129 PRINTB #TFM1,\$REG5,\$REG4,\$REG2
 4130 .ENDM

4131 .MACRO MDT3
 4132 PRINTB #TFM2,\$REG5,\$REG4
 4133 .ENDM

4134 .MACRO MDT4
 4135 PRINTB #TFM3,\$REG5,\$REG4,FLAG
 4136 .ENDM

4137 .MACRO MDT5
 4138 PRINTB #TFM3,\$REG5,\$REG4,\$REG2
 4139 .ENDM

4140 .MACRO MDT0
 4141 .ENDM
 4142 .MACRO MDT6
 4143 PRINTB #TFM4,\$REG2,\$REG4
 4144 .ENDM

4145 .MACRO MDT7
 4146 PRINTB #TFM4,\$REG5,\$REG4
 4147
 4148
 4149
 4150
 4151
 4152

CZDMQCO M8207 STATIC DIAG #2 MACY11 30A(1052) 21-JUL-81 14:48 N 4
CZDMQC.P11 21-JUL-81 14:36 GLOBAL ERROR REPORT SECTION PAGE 8

SEQ 0052

4154
4155
4156
4157

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

CZDMQCO M8207 STATIC DIAG #2 MACY11 30A(1052) 21-JUL-81 14:48 PAGE 9
CZDMQC.P11 21-JUL-81 14:36 GLOBAL ERROR REPORT SECTION

SEQ 0053

4159
4160
4161

.MACRO SMD ERRNN ERNB ERHM ERFM
BGNMSG ERR'ERRVN

```
4163          PRINTB #FM1,#EM'ERNB
4164          PRINTB #FM1,#DH'ERHM
4165          MDT'ERFM
4166          PRINTB #STM
4167          ENDMSG
4168          .ENDM
4169
4170          .MACRO  ERROR  ECB
4171          JSR     PC,SV05
4172          ERRDF   'ECB',EMO,ERR'ECB'
4173          .ENDM
4174
4175
4176
4177 006032          $MD      1,1,1,1
      (4) 006032
      (9) 006032 012746 005056
      (8) 006036 012746 005046
      (7) 006042 012746 000002
      (4) 006046 010600
      (5) 006050 104414
      (5) 006052 062706 000006
      (9) 006056 012746 005724
      (8) 006062 012746 005046
      (7) 006066 012746 000002
      (4) 006072 010600
      (5) 006074 104414
      (5) 006076 062706 000006
      (12) 006102 013746 002430
      (11) 006106 013746 002420
      (10) 006112 013746 002424
      (9) 006116 012746 003666
      (8) 006122 012746 000004
      (5) 006126 010600
      (6) 006130 104414
      (6) 006132 062706 000012
      (8) 006136 012746 005627
      (7) 006142 012746 000001
      (4) 006146 010600
      (5) 006150 104414
      (5) 006152 062706 000004
      (4) 006156
      (4) 006156 104423
4178 006160
      (4) 006160
      (9) 006160 012746 005076
      (8) 006164 012746 005046
      (7) 006170 012746 000002
      (4) 006174 010600
      (5) 006176 104414
      (5) 006200 062706 000006
      (9) 006204 012746 005724
      (8) 006210 012746 005046
      (7) 006214 012746 000002
      (4) 006220 010600
      (5) 006222 104414

ERR1::      MOV     #EM1,-(SP)
            MOV     #FM1,-(SP)
            MOV     #2,-(SP)
            MOV     SP,R0
            TRAP    C$PNTB
            ADD     #6,SP
            MOV     #DH1,-(SP)
            MOV     #FM1,-(SP)
            MOV     #2,-(SP)
            MOV     SP,R0
            TRAP    C$PNTB
            ADD     #6,SP
            MOV     $REG0,-(SP)
            MOV     $REG4,-(SP)
            MOV     $REG2,-(SP)
            MOV     #TFM1,-(SP)
            MOV     #4,-(SP)
            MOV     SP,R0
            TRAP    C$PNTB
            ADD     #12,SP
            MOV     #STM,-(SP)
            MOV     #1,-(SP)
            MOV     SP,R0
            TRAP    C$PNTB
            ADD     #4,SP

L10003:      TRAP    C$MSG
            $MD      2,2,1,1

ERR2::      MOV     #EM2,-(SP)
            MOV     #FM1,-(SP)
            MOV     #2,-(SP)
            MOV     SP,R0
            TRAP    C$PNTB
            ADD     #6,SP
            MOV     #DH1,-(SP)
            MOV     #FM1,-(SP)
            MOV     #2,-(SP)
            MOV     SP,R0
            TRAP    C$PNTB
```

(5)	006224	062706	000006	ADD	#6,SP
(12)	006230	013746	002430	MOV	\$REG0,-(SP)
(11)	006234	013746	002420	MOV	\$REG4,-(SP)
(10)	006240	013746	002424	MOV	\$REG2,-(SP)
(9)	006244	012746	003666	MOV	#TFM1,-(SP)
(8)	006250	012746	000004	MOV	#4,-(SP)
(5)	006254	010600		MOV	SP,R0
(6)	006256	104414		TRAP	C\$PNTB
(6)	006260	062706	000012	ADD	#12,SP
(8)	006264	012746	005627	MOV	#STM,-(SP)
(7)	006270	012746	000001	MOV	#1,-(SP)
(4)	006274	010600		MOV	SP,R0
(5)	006276	104414		TRAP	C\$PNTB
(5)	006300	062706	000004	ADD	#4,SP
(4)	006304				
(4)	006304	104423		L10004: TRAP	C\$MSG
4179	006306			\$MD	3,1,1,2
(4)	006306			ERR3::	
(9)	006306	012746	005056	MOV	#EM1,-(SP)
(8)	006312	012746	005046	MOV	#FM1,-(SP)
(7)	006316	012746	000002	MOV	#2,-(SP)
(4)	006322	010600		MOV	SP,R0
(5)	006324	104414		TRAP	C\$PNTB
(5)	006326	062706	000006	ADD	#6,SP
(9)	006332	012746	005724	MOV	#DH1,-(SP)
(8)	006336	012746	005046	MOV	#FM1,-(SP)
(7)	006342	012746	000002	MOV	#2,-(SP)
(4)	006346	010600		MOV	SP,R0
(5)	006350	104414		TRAP	C\$PNTB
(5)	006352	062706	000006	ADD	#6,SP
(12)	006356	013746	002424	MOV	\$REG2,-(SP)
(11)	006362	013746	002420	MOV	\$REG4,-(SP)
(10)	006366	013746	002416	MOV	\$REG5,-(SP)
(9)	006372	012746	003666	MOV	#TFM1,-(SP)
(8)	006376	012746	000004	MOV	#4,-(SP)
(5)	006402	010600		MOV	SP,R0
(6)	006404	104414		TRAP	C\$PNTB
(6)	006406	062706	000012	ADD	#12,SP
(8)	006412	012746	005627	MOV	#STM,-(SP)
(7)	006416	012746	000001	MOV	#1,-(SP)
(4)	006422	010600		MOV	SP,R0
(5)	006424	104414		TRAP	C\$PNTB
(5)	006426	062706	000004	ADD	#4,SP
(4)	006432				
(4)	006432	104423		L10005: TRAP	C\$MSG
4180	006434			\$MD	4,3,2,3
(4)	006434			ERR4::	
(9)	006434	012746	005131	MOV	#EM3,-(SP)
(8)	006440	012746	005046	MOV	#FM1,-(SP)
(7)	006444	012746	000002	MOV	#2,-(SP)
(4)	006450	010600		MOV	SP,R0
(5)	006452	104414		TRAP	C\$PNTB
(5)	006454	062706	000006	ADD	#6,SP
(9)	006460	012746	005755	MOV	#DH2,-(SP)
(8)	006464	012746	005046	MOV	#FM1,-(SP)
(7)	006470	012746	000002	MOV	#2,-(SP)

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 E 5
GLOBAL ERROR REPORT SECTION PAGE 10-2

SEQ 0056

(4)	006474	010600	MOV	SP,R0
(5)	006476	104414	TRAP	C\$PNTB
(5)	006500	062706	ADD	#6,SP
(11)	006504	013746	MOV	\$REG4,-(SP)
(10)	006510	013746	MOV	\$REG5,-(SP)
(9)	006514	012746	MOV	#TFM2,-(SP)
(8)	006520	012746	MOV	#3,-(SP)
(5)	006524	010600	MOV	SP,R0
(6)	006526	104414	TRAP	C\$PNTB
(6)	006530	062706	ADD	#10,SP
(8)	006534	012746	MOV	#STM,-(SP)
(7)	006540	012746	MOV	#1,-(SP)
(4)	006544	010600	MOV	SP,R0
(5)	006546	104414	TRAP	C\$PNTB
(5)	006550	062706	ADD	#4,SP
(4)	006554			
(4)	006554	104423	L10006:	TRAP C\$MSG

4182 006556
(4) 006556
(9) 006556 012746 005144
(8) 006562 012746 005046
(7) 006566 012746 000002
(4) 006572 010600
(5) 006574 104414
(5) 006576 062706 000006
(9) 006602 012746 005755
(8) 006606 012746 005046
(7) 006612 012746 000002
(4) 006616 010600
(5) 006620 104414
(5) 006622 062706 000006
(11) 006626 013746 002420
(10) 006632 013746 002416
(9) 006636 012746 003712
(8) 006642 012746 000003
(5) 006646 010600
(6) 006650 104414
(6) 006652 062706 000010
(8) 006656 012746 005627
(7) 006662 012746 000001
(4) 006666 010600
(5) 006670 104414
(5) 006672 062706 000004
(4) 006676
(4) 006676 104423

SMD 5,4,2,3
ERR5::
MOV #EM4,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #6,SP
MOV #DH2,-(SP)
MOV #FM1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #6,SP
MOV \$REG4,-(SP)
MOV \$REG5,-(SP)
MOV #TFM2,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #10,SP
MOV #STM,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #4,SP
L'0007:
TRAP C\$MSG

4184 006700
(4) 006700
(9) 006700 012746 005171
(8) 006704 012746 005046
(7) 006710 012746 000002
(4) 006714 010600
(5) 006716 104414
(5) 006720 062706 000006
(9) 006724 012746 005724
(8) 006730 012746 005046
(7) 006734 012746 000002
(4) 006740 010600
(5) 006742 104414
(5) 006744 062706 000006
(12) 006750 013746 002406
(11) 006754 013746 002420
(10) 006760 013746 002416
(9) 006764 012746 003730
(8) 006770 012746 000004
(5) 006774 010600
(6) 006776 104414
(6) 007000 062706 000012
(8) 007004 012746 005627
(7) 007010 012746 000001
(4) 007014 010600
(5) 007016 104414
(5) 007020 062706 000004
(4) 007024
(4) 007024 104423
4185 007026
(4) 007026
(9) 007026 012746 005216
(8) 007032 012746 005046
(7) 007036 012746 000002
(4) 007042 010600
(5) 007044 104414
(5) 007046 062706 000006
(9) 007052 012746 005724
(8) 007056 012746 005046
(7) 007062 012746 000002
(4) 007066 010600
(5) 007070 104414
(5) 007072 062706 000006
(12) 007076 013746 002424
(11) 007102 013746 002420
(10) 007106 013746 002416
(9) 007112 012746 003730
(8) 007116 012746 000004
(5) 007122 010600
(6) 007124 104414
(6) 007126 062706 000012
(8) 007132 012746 005627
(7) 007136 012746 000001
(4) 007142 010600
(5) 007144 104414
(5) 007146 062706 000004

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

(4) 007152
(4) 007152 104423
4186 007154
(4) 007154
(9) 007154 012746 005233
(8) 007160 012746 005046
(7) 007164 012746 000002
(4) 007170 010600
(5) 007172 104414
(5) 007174 062706 000006
(9) 007200 012746 005755
(8) 007204 012746 005046
(7) 007210 012746 000002
(4) 007214 010600
(5) 007216 104414
(5) 007220 062706 000006
(11) 007224 013746 002420
(10) 007230 013746 002416
(9) 007234 012746 003712
(8) 007240 012746 000003
(5) 007244 010600
(6) 007246 104414
(6) 007250 062706 000010
(8) 007254 012746 005627
(7) 007260 012746 000001
(4) 007264 010600
(5) 007266 104414
(5) 007270 062706 000004
(4) 007274
(4) 007274 104423
4187 007276
(4) 007276
(9) 007276 012746 005260
(8) 007302 012746 005046
(7) 007306 012746 000002
(4) 007312 010600
(5) 007314 104414
(5) 007316 062706 000006
(9) 007322 012746 005755
(8) 007326 012746 005046
(7) 007332 012746 000002
(4) 007336 010600
(5) 007340 104414
(5) 007342 062706 000006
(11) 007346 013746 002420
(10) 007352 013746 002424
(9) 007356 012746 003755
(8) 007362 012746 000003
(5) 007366 010600
(6) 007370 104414
(6) 007372 062706 000010
(8) 007376 012746 005627
(7) 007402 012746 000001
(4) 007406 010600
(5) 007410 104414
(5) 007412 062706 000004

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

(4) 007416
(4) 007416 104423
4188 007420
(4) 007420
(9) 007420 012746 005233
(8) 007424 012746 005046
(7) 007430 012746 000002
(4) 007434 010600
(5) 007436 104414
(5) 007440 062706 000006
(9) 007444 012746 005755
(8) 007450 012746 005046
(7) 007454 012746 000002
(4) 007460 010600
(5) 007462 104414
(5) 007464 062706 000006
(11) 007470 013746 002420
(10) 007474 013746 002416
(9) 007500 012746 003755
(8) 007504 012746 000003
(5) 007510 010600
(6) 007512 104414
(6) 007514 062706 000010
(8) 007520 012746 005627
(7) 007524 012746 000001
(4) 007530 010600
(5) 007532 104414
(5) 007534 062706 000004
(4) 007540
(4) 007540 104423
4189 007542
(4) 007542
(9) 007542 012746 005260
(8) 007546 012746 005046
(7) 007552 012746 000002
(4) 007556 010600
(5) 007560 104414
(5) 007562 062706 000006
(9) 007566 012746 005755
(8) 007572 012746 005046
(7) 007576 012746 000002
(4) 007602 010600
(5) 007604 104414
(5) 007606 062706 000006
(11) 007612 013746 002420
(10) 007616 013746 002416
(9) 007622 012746 003712
(8) 007626 012746 000003
(5) 007632 010600
(6) 007634 104414
(6) 007636 062706 000010
(8) 007642 012746 005627
(7) 007646 012746 000001
(4) 007652 010600
(5) 007654 104414
(5) 007656 062706 000004

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

(4) 007662
(4) 007662 104423
4190 007664
(4) 007664
(9) 007664 012746 005312
(8) 007670 012746 005046
(7) 007674 012746 000002
(4) 007700 010600
(5) 007702 104414
(5) 007704 062706 000006
(9) 007710 012746 005755
(8) 007714 012746 005046
(7) 007720 012746 000002
(4) 007724 010600
(5) 007726 104414
(5) 007730 062706 000006
(11) 007734 013746 002420
(10) 007740 013746 002424
(9) 007744 012746 003755
(8) 007750 012746 000003
(5) 007754 010600
(6) 007756 104414
(6) 007760 062706 000010
(8) 007764 012746 005627
(7) 007770 012746 000001
(4) 007774 010600
(5) 007776 104414
(5) 010000 062706 000004
(4) 010004
(4) 010004 104423

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

4192 010006
(4) 010006
(9) 010006 012746 005333
(8) 010012 012746 005046
(7) 010016 012746 000002
(4) 010022 010600
(5) 010024 104414
(5) 010026 062706 000006
(9) 010032 012746 005775
(8) 010036 012746 005046
(7) 010042 012746 000002
(4) 010046 010600
(5) 010050 104414
(5) 010052 062706 000006
(12) 010056 013746 002420
(11) 010062 013746 002416
(10) 010066 013746 002412
(9) 010072 012746 003773
(8) 010076 012746 000004
(5) 010102 010600
(6) 010104 104414
(6) 010106 062706 000012
(8) 010112 012746 005627
(7) 010116 012746 000001
(4) 010122 010600
(5) 010124 104414
(5) 010126 062706 000004
(4) 010132
(4) 010132 104423
4193 010134
(4) 010134
(9) 010134 012746 005353
(8) 010140 012746 005046
(7) 010144 012746 000002
(4) 010150 010600
(5) 010152 104414
(5) 010154 062706 000006
(9) 010160 012746 005755
(8) 010164 012746 005046
(7) 010170 012746 000002
(4) 010174 010600
(5) 010176 104414
(5) 010200 062706 000006
(11) 010204 013746 002420
(10) 010210 013746 002416
(9) 010214 012746 003755
(8) 010220 012746 000003
(5) 010224 010600
(6) 010226 104414
(6) 010230 062706 000010
(8) 010234 012746 005627
(7) 010240 012746 000001
(4) 010244 010600
(5) 010246 104414
(5) 010250 062706 000004
(4) 010254

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

CZDMQCO M8207 STATIC DIAG #2 MACY11 30A(1052) 21-JUL-81 14:48 L 5
CZDMQC.P11 21-JUL-81 14:36 GLOBAL ERROR REPORT SECTION PAGE 13-1

SEQ 0063

(4) 010254 104423
4194

TRAP C\$MSG

4196					
4197	010256			\$MD	17,17,0,0
(4)	010256			ERR17::	
(9)	010256	012746	005434	MOV	#EM17,-(SP)
(8)	010262	012746	005046	MOV	#FM1,-(SP)
(7)	010266	012746	000002	MOV	#2,-(SP)
(4)	010272	010600		MOV	SP,R0
(5)	010274	104414		TRAP	C\$PNTB
(5)	010276	062706	000006	ADD	#6,SP
(9)	010302	012746	005723	MOV	#DH0,-(SP)
(8)	010306	012746	005046	MOV	#FM1,-(SP)
(7)	010312	012746	000002	MOV	#2,-(SP)
(4)	010316	010600		MOV	SP,R0
(5)	010320	104414		TRAP	C\$PNTB
(5)	010322	062706	000006	ADD	#6,SP
(8)	010326	012746	005627	MOV	#STM,-(SP)
(7)	010332	012746	000001	MOV	#1,-(SP)
(4)	010336	010600		MOV	SP,R0
(5)	010340	104414		TRAP	C\$PNTB
(5)	010342	062706	000004	ADD	#4,SP
(4)	010346			L10021:	
(4)	010346	104423		TRAP	C\$MSG
4198	010350			\$MD	29,29,2,3
(4)	010350			ERR29::	
(9)	010350	012746	005512	MOV	#EM29,-(SP)
(8)	010354	012746	005046	MOV	#FM1,-(SP)
(7)	010360	012746	000002	MOV	#2,-(SP)
(4)	010364	010600		MOV	SP,R0
(5)	010366	104414		TRAP	C\$PNTB
(5)	010370	062706	000006	ADD	#6,SP
(9)	010374	012746	005755	MOV	#DH2,-(SP)
(8)	010400	012746	005046	MOV	#FM1,-(SP)
(7)	010404	012746	000002	MOV	#2,-(SP)
(4)	010410	010600		MOV	SP,R0
(5)	010412	104414		TRAP	C\$PNTB
(5)	010414	062706	000006	ADD	#6,SP
(11)	010420	013746	002420	MOV	\$REG4,-(SP)
(10)	010424	013746	002416	MOV	\$REG5,-(SP)
(9)	010430	012746	003712	MOV	#TFM2,-(SP)
(8)	010434	012746	000003	MOV	#3,-(SP)
(5)	010440	010600		MOV	SP,R0
(6)	010442	104414		TRAP	C\$PNTB
(6)	010444	062706	000010	ADD	#10,SP
(8)	010450	012746	005627	MOV	#STM,-(SP)
(7)	010454	012746	000001	MOV	#1,-(SP)
(4)	010460	010600		MOV	SP,R0
(5)	010462	104414		TRAP	C\$PNTB
(5)	010464	062706	000004	ADD	#4,SP
(4)	010470			L10022:	
(4)	010470	104423		TRAP	C\$MSG
4199	010472			\$MD	35,35,2,3
(4)	010472			ERR35::	
(9)	010472	012746	005434	MOV	#EM35,-(SP)
(8)	010476	012746	005046	MOV	#FM1,-(SP)
(7)	010502	012746	000002	MOV	#2,-(SP)
(4)	010506	010600		MOV	SP,R0

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 PAGE 14-1
GLOBAL ERROR REPORT SECTION

SEQ 0065

(5)	010510	104414		TRAP	C\$PNTB
(5)	010512	062706	000006	ADD	#6,SP
(9)	010516	012746	005755	MOV	#DH2,-(SP)
(8)	010522	012746	005046	MOV	#FM1,-(SP)
(7)	010526	012746	000002	MOV	#2,-(SP)
(4)	010532	010600		MOV	SP,R0
(5)	010534	104414		TRAP	C\$PNTB
(5)	010536	062706	000006	ADD	#6,SP
(11)	010542	013746	002420	MOV	\$REG4,-(SP)
(10)	010546	013746	002416	MOV	\$REG5,-(SP)
(9)	010552	012746	003712	MOV	#TFM2,-(SP)
(8)	010556	012746	000003	MOV	#3,-(SP)
(5)	010562	010600		MOV	SP,R0
(6)	010564	104414		TRAP	C\$PNTB
(6)	010566	062706	000010	ADD	#10,SP
(8)	010572	012746	005627	MOV	#STM,-(SP)
(7)	010576	012746	000001	MOV	#1,-(SP)
(4)	010602	010600		MOV	SP,R0
(5)	010604	104414		TRAP	C\$PNTB
(5)	010606	062706	000004	ADD	#4,SP
(4)	010612				
(4)	010612	104423		L10023: TRAP	C\$MSG
4200					
4201	010614			BGNMSG	ERR36
(3)	010614			ERR36::	
4202	010614			PRINTB	#STM
(7)	010614	012746	005627	MOV	#STM,-(SP)
(6)	010620	012746	000001	MOV	#1,-(SP)
(3)	010624	010600		MOV	SP,R0
(4)	010626	104414		TRAP	C\$PNTB
(4)	010630	062706	000004	ADD	#4,SP
4203	010634			ENDMSG	
(3)	010634			L10024:	
(3)	010634	104423		TRAP	C\$MSG
4204					
4205	010636			BGNMSG	ERR40
(3)	010636			ERR40::	
4206	010636			PRINTF	#TFM40,R2
(8)	010636	010246		MOV	R2,-(SP)
(7)	010640	012746	004263	MOV	#TFM40,-(SP)
(6)	010644	012746	000002	MOV	#2,-(SP)
(3)	010650	010600		MOV	SP,R0
(4)	010652	104417		TRAP	C\$PNTF
(4)	010654	062706	000006	ADD	#6,SP
4207	010660			PRINTB	#STM
(7)	010660	012746	005627	MOV	#STM,-(SP)
(6)	010664	012746	000001	MOV	#1,-(SP)
(3)	010670	010600		MOV	SP,R0
(4)	010672	104414		TRAP	C\$PNTB
(4)	010674	062706	000004	ADD	#4,SP
4208	010700			ENDMSG	
(3)	010700			L10025:	
(3)	010700	104423		TRAP	C\$MSG
4209	010702			BGNMSG	ERR41
(3)	010702			ERR41::	
4210	010702			PRINTF	#TFM41

(7)	010702	012746	004105		MOV	#TFM41,-(SP)
(6)	010706	012746	000001		MOV	#1,-(SP)
(3)	010712	010600			MOV	SP,R0
(4)	010714	104417			TRAP	C\$PNTF
(4)	010716	062706	000004		ADD	#4,SP
4211	010722			PRINTB	#STM	
(7)	010722	012746	005627		MOV	#STM,-(SP)
(6)	010726	012746	000001		MOV	#1,-(SP)
(3)	010732	010600			MOV	SP,R0
(4)	010734	104414			TRAP	C\$PNTB
(4)	010736	062706	000004		ADD	#4,SP
4212	010742			ENDMSG		
(3)	010742			L10026:		
(3)	010742	104423			TRAP	C\$MSG
4213	010744			BGNMSG	ERR42	
(3)	010744			ERR42::		
4214	010744			PRINTF	#TFM42	
(7)	010744	012746	004174		MOV	#TFM42,-(SP)
(6)	010750	012746	000001		MOV	#1,-(SP)
(3)	010754	010600			MOV	SP,R0
(4)	010756	104417			TRAP	C\$PNTF
(4)	010760	062706	000004		ADD	#4,SP
4215	010764			PRINTB	#STM	
(7)	010764	012746	005627		MOV	#STM,-(SP)
(6)	010770	012746	000001		MOV	#1,-(SP)
(3)	010774	010600			MOV	SP,R0
(4)	010776	104414			TRAP	C\$PNTB
(4)	011000	062706	000004		ADD	#4,SP
4216	011004			ENDMSG		
(3)	011004			L10027:		
(3)	011004	104423			TRAP	C\$MSG
4217	011006			BGNMSG	ERR43	
(3)	011006			ERR43::		
4219	011006			PRINTF	#TFM43,R5,R4	
(9)	011006	010446			MOV	R4,-(SP)
(8)	011010	010546			MOV	R5,-(SP)
(7)	011012	012746	004344		MOV	#TFM43,-(SP)
(6)	011016	012746	000003		MOV	#3,-(SP)
(3)	011022	010600			MOV	SP,R0
(4)	011024	104417			TRAP	C\$PNTF
(4)	011026	062706	000010		ADD	#10,SP
4220	011032			PRINTB	#STM	
(7)	011032	012746	005627		MOV	#STM,-(SP)
(6)	011036	012746	000001		MOV	#1,-(SP)
(3)	011042	010600			MOV	SP,R0
(4)	011044	104414			TRAP	C\$PNTB
(4)	011046	062706	000004		ADD	#4,SP
4221	011052			ENDMSG		
(3)	011052			L10030:		
(3)	011052	104423			TRAP	C\$MSG
4222	011054			BGNMSG	ERR44	
(3)	011054			ERR44::		
4223	011054			PRINTF	#TFM44,#TMMC,R4	
(9)	011054	010446			MOV	R4,-(SP)
(8)	011056	012746	005010		MOV	#TMMC,-(SP)

```

(7) 011062 012746 004427
(6) 011066 012746 000003
(3) 011072 010600
(4) 011074 104417
(4) 011076 062706 000010
4224 011102
(7) 011102 012746 005627
(6) 011106 012746 000001
(3) 011112 010600
(4) 011114 104414
(4) 011116 062706 000004
4225 011122
(3) 011122
(3) 011122 104423
4226 011124
(3) 011124
4227 011124
(9) 011124 010446
(8) 011126 012746 005010
(7) 011132 012746 004466
(6) 011136 012746 000003
(3) 011142 010600
(4) 011144 104417
(4) 011146 062706 000010
4228 011152
(7) 011152 012746 004524
(6) 011156 012746 000001
(3) 011162 010600
(4) 011164 104414
(4) 011166 062706 000004
4229 011172
(7) 011172 012746 005627
(6) 011176 012746 000001
(3) 011202 010600
(4) 011204 104414
(4) 011206 062706 000004
4230 011212
(3) 011212
(3) 011212 104423
4231 011214
(3) 011214
4232 011214
(9) 011214 010446
(8) 011216 013746 002452
(7) 011222 012746 004647
(6) 011226 012746 000003
(3) 011232 010600
(4) 011234 104417
(4) 011236 062706 000010
4233 011242
(7) 011242 012746 005627
(6) 011246 012746 000001
(3) 011252 010600
(4) 011254 104414
(4) 011256 062706 000004
4234 011262

```

```

MOV #TFM44,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #10,SP
PRINTB #STM
MOV #STM,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP
ENDMSG
L10031:
TRAP C$MSG
BGNMSG ERR45
PRINTF #TFM45,#TMMC,R4
MOV R4,-(SP)
MOV #TMMC,-(SP)
MOV #TFM45,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #10,SP
PRINTB #TFM45A
MOV #TFM45A,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP
PRINTB #STM
MOV #STM,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP
ENDMSG
L10032:
TRAP C$MSG
BGNMSG ERR46
PRINTF #TFM46,$GDDAT,R4
MOV R4,-(SP)
MOV $GDDAT,-(SP)
MOV #TFM46,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #10,SP
PRINTB #STM
MOV #STM,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP
ENDMSG

```

(3)	011262			L10033:	
(3)	011262	104423		TRAP	C\$MSG
4235					
4236	011264			BGNMSG	ERR47
(3)	011264			ERR47::	
4237	011264			PRINTF	#TFM47,R5,R4
(9)	011264	010446		MOV	R4,-(SP)
(8)	011266	010546		MOV	R5,-(SP)
(7)	011270	012746	004733	MOV	#TFM47,-(SP)
(6)	011274	012746	000003	MOV	#3,-(SP)
(3)	011300	010600		MOV	SP,R0
(4)	011302	104417		TRAP	C\$PNTF
(4)	011304	062706	000010	ADD	#10,SP
4238	011310			PRINTB	#STM
(7)	011310	012746	005627	MOV	#STM,-(SP)
(6)	011314	012746	000001	MOV	#1,-(SP)
(3)	011320	010600		MOV	SP,R0
(4)	011322	104414		TRAP	C\$PNTB
(4)	011324	062706	000004	ADD	#4,SP
4239	011330			ENDMSG	
(3)	011330			L10034:	
(3)	011330	104423		TRAP	C\$MSG
4240					

CZDMQCO M8207 STATIC DIAG #2 MACY11 30A(1052) 21-JUL-81 14:48 E 6
CZDMQC.P11 21-JUL-81 14:36 REPORT CODING SECTION PAGE 14-5

SEQ 0069

4242
4243
4244

.SBTTL REPORT CODING SECTION

{
CZDMCO M8207 STATIC DIAG #2
CZDMOC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 F 6
REPORT CODING SECTION PAGE 15

SEQ 0070

4245

:::

CZDMQCO MB207 STATIC DIAG #2
 CZDMQCO.P11 21-JUL-81 14:36

MARY11 30A(1052) 21-JUL-81 14:48 PAGE 16
 REPORT CODING SECTION

SEQ 0071

4248
 4249
 4250
 4251
 4252
 3)
 4253
 4259
 4260
 (4)
 (3)
 4261

011332
 011332
 011332
 011332 00016'
 011334 000000

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

BGNRPT
 LSRPT::

EXIT RPT
 .WORD JSJMP
 .WORD L'U035-2-.

CZDMQCO MB207 STATIC DIAG #2 MACY11 30A(1052) 21-JUL-81 14:48 M 6
CZDMQC.P11 21-JUL-81 14:36 REPORT CODING SECTION PAGE 17

SEQ 0072

4269
4270 011336
(3) 011336
(3) 011336 104425
4271

L10035: ENDRPT
TRAP CSRPT

CZDMQCO M8207 STATIC DIAG #2 MACY11 30A(1052) 21-JUL-81 14:48 I 6
CZDMQC.P11 21-JUL-81 14:36 REPORT CODING SECTION PAGE 18

SFO 0073

4273
4274

CZDMQCO M8207 STATIC DIAG #2 MACY11 30A(1052) 21-JUL-81 14:48 J 6
CZDMQC.P11 21-JUL-81 14.36 REPORT CODING SECTION PAGE 19

SEQ 0074

4276
4277

CZDMQCO M8207 STATIC DIAG #2 MACY11 30A(1052) 21-JUL-81 14:48 ^{K 6} PAGE 19-1
CZDMQC.P11 21-JUL-81 14:36 INITIALIZE SECTION

SEQ 0075

4279

.SBTTL INITIALIZE SECTION

```
4281
4282 ;////////////////////
4283 ;/ THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
4284 ;/ AT THE BEGINNING OF EACH PASS.
4285 ;////////////////////
4286
4287 011340          BGNINIT
(3) 011340 L$INIT::
4288
4289 ;INITIALIZE SUBROUTINE STACK
4290 011340 012705 002730      MOV      #SSTACK,R5
4291 ;STORE BASE LEVEL PROGRAM STACK POINTER
4292 011344 010637 002344      MOV      SP,PSTACK
4293 011350 005737 002462      TST      FTIME
4294 011354 001011            BNE      1$
4295 011356 013737 000004 002464      MOV      @#4,SAVE4
4296 011364 013737 000006 002466      MOV      @#6,SAVE6
4297 011372 012737 000001 002462      MOV      #1,FTIME
4298 011400 013737 002464 000004 1$: MOV      SAVE4,@#4
4299 011406 013737 002466 000006      MOV      SAVE6,@#6
4300
4301 ;SEE IF PROGRAM JUST STARTED, BR IF YES
4302 011414          READEF      #EF.START
(3) 011414 012700 000040      MOV      #EF.START,R0
(3) 011420 104447          TRAP      C$REFG
4303 011422          BCOMPLETE      NEWST
(2) 011422 103414          BCS      NEWST
4304 ;SEE IF THIS IS A NEW PASS, BR IF YES
4305 011424          READEF      #EF.NEW
(3) 011424 012700 000035      MOV      #EF.NEW,R0
(3) 011430 104447          TRAP      C$REFG
4306 011432          BCOMPLETE      NEWST
(2) 011432 103410          BCS      NEWST
4307 ;SEE IF PROGRAM WAS JUST CONTINUED
4308 011434          READEF      #EF.CONTINUE
(3) 011434 012700 000036      MOV      #EF.CONTINUE,R0
(3) 011440 104447          TRAP      C$REFG
4309 011442          BCOMPLETE      ENDIT
(2) 011442 103576          BCS      ENDIT
4310 ;SEE IF PROGRAM JUST RESTARTED, BR IF NOT
4311 011444          READEF      #EF.RESTART
(3) 011444 012700 000037      MOV      #EF.RESTART,R0
(3) 011450 104447          TRAP      C$REFG
4312 011452          BNCOMPLETE      GETPRM
(2) 011452 103003          BCC      GETPRM
4313
4314 011454          NEWST:
4315 ;RESET LOGICAL DEVICE TO -1
4316 011454 012737 177777 002342      MOV      #-1,LOGDEV
4317
4318 ;GET UNIBUS ADRS, VECTOR, PRIORITY LEVEL, LINE UNIT, SWITCH
4319 ; PACKS, TEST CONNECTOR INFO. FOR THIS M8200,4,7 (CURRENT LOGICAL
4320 ; DEVICE).
4321 011462          GETPRM:
4322 011462 005237 002342          INC      LOGDEV
4323 011466 023737 002342 002012      CMP      LOGDEV,L$UNIT
```

4324	011474	002367		BGE	NEWST
4325	011476			GPHARD	LOGDEV,R1
(3)	011476	013700	002342	MOV	LOGDEV,R0
(3)	011502	104442		TRAP	C\$GPHRD
(3)	011504	010001		MOV	R0,R1
4326	011506			BNCOMPLETE	GETPRM
(2)	011506	103365		BCC	GETPRM
4327	011510	012137	002414	MOV	(R1)+,WTYPE
4328				;GET ADDRESS OF	M8200,4,7
4329	011514	011137	002516	MOV	(R1),KMCSR
4330				;GET POINTER TO	M8200,4,7 CSR HI BYTE
4331	011520	011137	002520	MOV	(R1),KMCSRH
4332	011524	005237	002520	INC	KMCSRH
4333				;GET POINTER TO	M8200,4,7 CTL OUT REG
4334	011530	011137	002522	MOV	(R1),KMCTL
4335	011534	062737	000002 002522	ADD	#2,KMCTL
4336				;GET POINTER TO	M8200,4,7 PORT REG - SEL 4
4337	011542	011137	002524	MOV	(R1),KMP04
4338	011546	062737	000004 002524	ADD	#4,KMP04
4339				;GET POINTER TO	M8200,4,7 PORT REG - SEL 6
4340	011554	012137	002526	MOV	(R1)+,KMP06
4341	011560	062737	000006 002526	ADD	#6,KMP06
4342				;GET POINTER TO	RCV VECTOR
4343	011566	011137	002506	MOV	(R1),KMRVEC
4344				;GET POINTER TO	RCV PRIORITY LEVEL
4345	011572	011137	002510	MOV	(R1),KMRLVL
4346	011576	062737	000002 002510	ADD	#2,KMRLVL
4347				;GET POINTER TO	TX VECTOR
4348	011604	011137	002512	MOV	(R1),KMTVEC
4349	011610	062737	000004 002512	ADD	#4,KMTVEC
4350				;GET POINTER TO	TX PRIORITY LEVEL
4351	011616	011137	002514	MOV	(R1),KMTLVL
4352	011622	062737	000006 002514	ADD	#6,KMTLVL
4353				;PUT VECTOR INTO	STAT1
4354	011630	016137	000020 002472	MOV	20(R1),RUNINH
4355	011636	012137	002500	MOV	(R1)+,STAT1
4356				;PUT PRIORITY INTO	STAT1
4357	011642	052137	002500	BIS	(R1)+,STAT1
4358				;SEE IF NO LINE	UNIT, SET BIT IF YES
4359	011646	005711		TST	(R1)
4360	011650	001004		BNE	50000\$
4361	011652	052737	010000 002500	BIS	#BIT12,STAT1
4362	011660	000416		BR	4\$
4363	011662			50000\$:	
4364				;SEE IF M8201	LINE UNIT, SET BIT IF YES
4365	011662	021127	000001	CMP	(R1),#1
4366	011666	001001		BNE	50001\$
4367	011670	000412		BR	4\$
4368	011672			50001\$:	
4369				;SEE IF M8202	LINE UNIT, SET BIT IF YES
4370	011672	021127	000002	CMP	(R1),#2
4371	011676	001004		BNE	50002\$
4372	011700	052737	020000 002500	BIS	#BIT13,STAT1
4373	011706	000403		BR	4\$
4374	011710			50002\$:	
4375				;SET BIT FOR	M8203 LINE UNIT

4376 011710 052737 100000 0025J0 BIS #BIT15,STAT1
4377 011716 4\$:
4378 ;SET BIT IN STAT1 FOR TEST CONNECTOR
4379 011716 056137 000006 002500 BIS 6(R1),STAT1
4380 011724 062701 000002 ADD #2,R1
4381 ;SET SWITCH PACK #1 IN STAT2 LOW BYTE
4382 011730 012137 002502 MOV (R1)+,STAT2
4383 ;SET SWITCH PACK #2 IN STAT2 HIGH BYTE
4384 011734 111137 002503 MOVB (R1),STAT2+1
4385
4386 ;INCREMENT LOGICAL UNIT (DEVICE) NUMBER
4387 : INC LOGDEV
4388 011740 000240 NOP
4389 011742 000240 NOP
4390
4391 011744 012737 002000 002436 MOV #2000,MEMSZ
4392 011752 005037 002432 CLR TYPE
4393 011756 123727 002414 000000 CMPB WTYPE,#0
4394 011764 001425 REQ ENDIT
4395 011766 123727 002414 000004 CMPB WTYPE,#4 ;KMC?
4396 011774 001004 BNE 5\$
4397 011776 012737 000001 002432 MOV #1,TYPE
4398 012004 000415 BR ENDIT
4399 012006 012737 007777 002436 5\$: MOV #7777,MEMSZ
4400 012014 123727 002414 000006 CMPB WTYPE,#6
4401 012022 001003 BNE 7\$
4402 012024 012737 000001 002432 MOV #1,TYPE
4403 012032 013737 002472 002470 7\$: MOV RUNINH,RUNB
4404 012040 6\$:
4405 012040 ENDIT:
4406 012040 ENDINIT
(3) 012040 L10036:
(3) 012040 104411 TRAP C\$INIT
4407
4408 .FVEN
4409 012042 BGNAUTO
(3) 012042 L\$AUTO: :
4410 ;DEVICE DOES NOT HAVE A 'READY'
4411 012042 013701 002516 MOV KMCSR,R1 ;R1 CONTAINS BASE M8200,4,7 ADDRESS
4412 012046 012705 000004 MOV #4,R5 ;4 REGISTERS TO BE TESTED
4413 012052 012737 012104 000004 MOV #2\$,4 ;SET OUT TIMEOUT TRAP
4414 012060 012737 000240 000006 MOV #240,6 ;LEVEL 7
4415 012066 005711 1\$: TST (R1) ;REFERENCE DEVICE REGISTERS
4416 012070 000240 NOP
4417 012072 062701 000002 ADD #2,R1 ;NEXT REGISTER
4418 012076 005305 DEC R5 ;DEC REGISTER COUNT
4419 012100 001372 BNE 1\$;BR IF NOT LAST REGISTER
4420 012102 000405 BR 3\$
4421
4422 012104 062706 000004 2\$: ADD #4,SP
4423 012110 DODU LOGDEV
(3) 012110 013700 002342 MCV LOGDEV,R0
(3) 012114 104451 TRAP C\$DODU
4424
4425 012116 013737 002464 000004 3\$: MOV SAVE4,4
4426 012124 013737 002466 000006 MOV SAVE6,6

CZDMQCO M8207 STATIC DIAG #2 MACY11 30A(1052) 21-JUL-81 14:48 ^{B 7} PAGE 20-3
CZDMQC.P11 21-JUL-81 14:36 INITIALIZE SECTION

SEQ 0079

4427 012132
(3) 012132
(3) 012132 104461
4428

ENDAUTO
L10037: *RAP C\$AUTO

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 PAGE 20-4
CLEANUP CODING SECTION

SEQ 0080

```
4430      .SBTTL  CLEANUP CODING SECTION
4431
4432      ://////////
4433      :// THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
4434      :// AT THE END OF EACH PASS.
4435      ://////////
4436
4437      BGNCLN
4438      (3) 012134
4439      L$CLEAN::
4440      (3) 012134 104433
4441      BRESET
4442      TRAP  C$RESET
4443
4444      ENDCLN
4445      L10040:
4446      TRAP  C$CLEAN
```

```
4447 .SBTTL DROP UNIT SECTION
4448
4449 ://
4450 :// THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
4451 :// TO NO LONGER BE TESTED.
4452 ://
4453
4454 012140 BGNDU
4455 (3) 012140 L$DU::
4456 012140 :ISSUE UNIBUS RESET TO CLEAN UP
4457 (3) 012140 104433 BRESET
4458 012142 TRAP C$RESET
4459 (3) 012142 ENDDU
4460 (3) 012142 104453 L10041:
4461 TRAP C$DU
4462
```

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 E 7 PAGE 20-6
ADD UNIT SECTION

SEQ 0082

4464
4465
4466
4467
4468
4469
4470
4471
4472 012144
(3) 012144
4473 012144
(3) 012144
(3) 012144 104452
4474
4475
4476
4477
4478
4479

.SBTTL ADD UNIT SECTION
:////////////////////
:// THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
:// TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING. IF
:// 'EF.AUNIT' IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
:////////////////////
L\$AU:: BGNAU
L10042: ENDAU
TRAP (\$AU

```

4481
4482
4483
4484
4485
4486 012146
4487
4488 012146
(2)
4489
4490
4491 012146
(2)
4492
4493 012146
(3) 012146
4494 012146 013701 002516
4495 012152 012705 000004
4496 012156 012737 012214 000004
4497 012164 012737 000240 000006
4498 012172 005711
4499 012174 000240
4500 012176
(3) 012176 104410
(3) 012200 000054
4501 012202 062701 000002
4502 012206 005305
4503 012210 001370
4504 012212 000410
4505
4506 012214 062706 000004
4507 012220
(5) 012224 104455
(6) 012226 000044
(6) 012230 005055
(6) 012232 010614
4508
4509 012234 013737 002464 000004
4510 012242 013737 002466 000006
4511 012250
(3) 012250 104410
(3) 012252 000002
4512
4513 012254
(3) 012254
(3) 012254 104401
4514
4515
4516
4517 012256
(2)
4518
4519
4520
4521 012256
(2)

```

.SBTTL HARDWARE TESTS

:START OF CODE BLOCK WHICH IS USED AS DATA
ROMMAP:

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

BGNTST
T1::

1\$: MOV KMCSR,R1 ;R1 CONTAINS BASE M8200,4,7 ADDRESS
MOV #4,R5 ;4 REGISTERS TO BE TESTED
MOV #2\$,4 ;SET OUT TIMEOUT TRAP
MOV #240,6 ;LEVEL 7
TST (R1) ;REFERENCE DEVICE REGISTERS
NOP
ESCAPE TST
TRAP C\$ESCAPE
.WORD L10043-
ADD #2,R1 ;NEXT REGISTER
DEC R5 ;DEC REGISTER COUNT
BNE 1\$;BR IF NOT LAST REGISTER
BR 3\$

2\$: ADD #4,SP
ERROR 36 ;TIME OUT ERROR
TRAP C\$ERDF
.WORD 36
.WORD EM0
.WORD ERR36

3\$: MOV SAVE4,4
MOV SAVE6,6
ESCAPE TST
TRAP C\$ESCAPE
.WORD L10043-
ENDTST
_10043:
TRAP C\$ETST
.EVEN

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

4522
4523 012256
(3) 012256
4524
4525 012256
4526 012262 013701 002516
4527 012266 005011
4528 012270 012705 052525
4529 012274 010561 000004
4530 012300
(1) 012300 004537 003044
4531 012304 120500
4532 012306
(1) 012306 004537 003044
4533 012312 061620
4534 012314
(1) 012314 004537 003044
4535 012320 061225
4536 012322 006005
4537 012324 005004
4538 012326 116104 000005
4539 012332 120504
4540 012334 001410
4541 012336
(5) 012342 104455
(6) 012344 000014
(6) 012346 005055
(6) 012350 007420
4542
4543 012352
(3) 012352 104410
(3) 012354 000044
4544 012356
(1) 012356 004537 003044
4545 012356
4546 012362 061620
4547 012364
(1) 012364 004537 003044
4548 012370 061225
4549 012372 006005
4550 012374 116104 000005
4551 012400 120504
4552 012402 001406
4553 012404
(5) 012410 104455
(6) 012412 000014
(6) 012414 005055
(6) 012416 007420
4554
4555 012420
4556 012420
(3) 012420
(3) 012420 104401
4557
4558 012422
(2)

BGNTST
T2::

MSTCLR
MOV KMCSR,R1
CLR (R1)
MOV #52525,R5
MOV R5,4(R1)
ROMCLK
JSR R5,,ROMCLK
120500
ROMCLK
JSR R5,,ROMCLK
061620
ROMCLK
JSR R5,,ROMCLK
061225
ROR R5
CLR R4
MOVB 5(R1),R4
CMPB R5,R4
BEQ 1\$
ERROR 12
TRAP C\$ERDF
.WORD 12
.WORD EMO
.WORD ERR12

;R1 CONTAINS BASE M8200.4.7 ADDRESS
;MASTER CLEAR M8200.4.7
;R1 = M8200.4.7 BASE ADDRESS
;CLEAR SELO
;START WITH 125
;PORT4 125
;NEXT WORD IS INSTRUCTION
;CLOCK INSTRUCTION
;PORT4 TO BR-REG
;NEXT WORD IS INSTRUCTION
;CLOCK INSTRUCTION
;BR RSH BR, SHIFT BR RIGHT
;NEXT WORD IS INSTRUCTION
;CLOCK INSTRUCTION
;PORT5 BR
;R5 = "EXPECTED"
;R4 = "FOUND"
;DID BR SHIFT RIGHT (NCE?)
;BR IF YES
;BR RIGHT SHIFT ERROR

;SHOULD BE 52

ESCAPE TST
TRAP C\$ESCAPE
.WORD L10044-

1\$:

ROMCLK
JSR R5,,ROMCLK
061620
ROMCLK
JSR R5,,ROMCLK
061225
ROR R5
MOVB 5(R1),R4
CMPB R5,R4
BEQ 2\$
ERROR 12
TRAP C\$ERDF
.WORD 12
.WORD EMO
.WORD ERR12

;NEXT WORD IS INSTRUCTION
;CLOCK INSTRUCTION
;BR RSH BR, SHFT BR RIGHT AGAIN
;NEXT WORD IS INSTRUCTION
;CLOCK INSTRUCTION
;PORT5 BR
;R5 = "EXPECTED"
;R4 = "FOUND"
;DID BR SHIFT RIGHT?
;BR IF YES
;BR RIGHT SHIFT ERROR

;S/B 25

2\$:
ENDTST
L10044:

TRAP C\$ETST

BADHEAD
;***** TEST 3 *****

```

4559                                     : *IOP CRAM WRITE/READ TEST
4560                                     : *FLOAT A 1 THROUGH EACH CRAM LOCATION
4561 012422                                BADHEAD
4562                                     : ***** TEST 3 *****
4563 012422                                BGNST
4564 012422                                T3::
4565                                     MACEX
4566                                     : DO NOT DO TEST IF M8200
4567 012430 104432                                TRAP C$EXIT
4568 012432 000116                                .WORD L10045-.
4569 012434 013701 002516                                MYINT
4570 012434 013701 002516                                MOV KMCSR,R1
4571                                     : RECORD DEVICE ADDR.
4572 012440 005037 002434                                : R1 CONTAINS BASE M8200,4,7 ADDRESS
4573 012444 012702 000001                                CLR MRO
4574 012450                                     : MRO = CRAM ADDRESS
4575 012450                                     : R2 = WRITE DATA
4576 012450                                     BGNSEG
4577 012450 104404                                TRAP C$BSEG
4578 012452 012711 002000                                MOV #BIT10,(R1)
4579 012452 013761 002434 000004                                MCV MRO,4(R1)
4580 012456 010261 000006                                MOV R2,6(R1)
4581 012470 052711 020000                                BIC #BIT13,(R1)
4582 012474 016104 000006                                MOV 6(R1),R4
4583 012500 020204                                CMP R2,R4
4584 012502 001410                                BEQ 4$
4585 012504 104455                                ERROR 1
4586 012510 000001                                TRAP C$ERDF
4587 012512 005055                                .WORD 1
4588 012514 006032                                .WORD EMO
4589 012516 104410                                .WORD ERR1
4590 012520 000002                                ESCAPE SEG
4591 012520 104410                                TRAP C$ESCAPE
4592 012522 000002                                .WORD 10000$-.
4593 012524 104405                                4$:
4594 012524 000241                                10000$:
4595 012526 006102                                TRAP C$ESEG
4596 012530 001346                                CLC
4597 012532 005237 002434                                ROL R2
4598 012534 023737 002436 002434                                BNE ADR5
4599 012540 001336                                INC MRO
4600 012540 001336                                CMP MEMSZ,MRO
4601 012540 001336                                BNE ADR4
4602 012550                                     : CLEAR CARRY
4603 012550                                     : SHIFT WRITE DATA
4604 012550                                     : BSR IF NOT DONE THIS ADDRESS
4605 012550                                     : BUMP TO NEXT CRAM ADDRESS
4606 012550                                     : DONE YET?
4607 012550                                     : BR IF NO
4608 012550                                     5$:
4609 012550                                     ENDTST
4610 012550                                     L10045:
4611 012550 104401                                TRAP C$ETST
4612 012552                                BADHEAD
4613 012552                                     : ***** TEST 4 *****
4614 012552                                     : *IOP CRAM WRITE/READ TEST
4615 012552                                     : *FLOAT A 0 THROUGH EACH CRAM LOCATION
4616 012552                                BADHEAD
4617 012552                                     : ***** TEST 4 *****
4618 012552                                BGNST

```

```

(3) 012552
4596 012552
(1)
(4) 012560 104432
(4) 012562 000126
4597 012564
(1) 012564 013701 002516
4598 012570
4599 012574 005037 002434
4600 012600 012702 000001
4601 012604
4602 012604
(3) 012604 104404
4603 012606 005102
4604 012610 012711 002000
4605 012614 013761 002434 000004
4606 012622 010261 000006
4607 012626 052711 020000
4608 012632 016104 000006
4609 012636 020204
4610 012640 001410
4611 012642
(5) 012646 104455
(6) 012650 000001
(6) 012652 005055
(6) 012654 006032
4612 012656
(3) 012656 104410
(3) 012660 000002
4613 012662
(3) 012662
(3) 012662 104405
4614 012664 005102
4615 012666 000241
4616 012670 006102
4617 012672 001344
4618 012674 005237 002434
4619 012700 023737 002436 002434
4620 012706 001334
4621 012710
4622 012710
(3) 012710
(3) 012710 104401
4623
4624 012712
(2)
4625
4626
4627
4628 012712
(2)
4629
4630 012712
(3) 012712
4631 012712
(1)

T4::
MACEX
;DO NOT DO TEST IF M8200
TRAP C$EXIT
;WORD L10046-.
MYINT
MOV KMCSR,R1 ;RECORD DEVICE ADDR.
MSTCLR ;MASTER CLEAR M8200,4,7
CLR MRO ;MRO = CRAM ADDRESS
MOV #1,R2 ;R2 = WRITE DATA

ADR1:
ADR2:
BGNSEG
TRAP C$BSEG
COM R2 ;MAKE IT A FLOATING ZERO
MOV #BIT10,(R1) ;SET ROMO
MOV MRO,4(R1) ;WRITE ADDRESS TO SEL4
MOV R2,6(R1) ;LOAD SEL6 WITH WRITE DATA
BIS #BIT13,(R1) ;WRITE SEL6 INTO CRAM
MOV 6(R1),R4 ;READ CRAM INTO 'FOUND'
CMP R2,R4 ;IS DATA CORRECT?
BEQ 4$ ;BR IF OK
ERROR 1 ;ERROR
TRAP C$ERDF
;WORD 1
;WORD EMO
;WORD ERR1
ESCAPE SEG
TRAP C$ESCAPE
;WORD 10000$-.
4$:
10000$:
ENDSEG
TRAP C$ESEG
COM R2 ;BACK TO FLOATING ONE
CLC ;CLEAR CARRY
ROL R2 ;SHIFT WRITE DATA
BNE ADR2 ;BR IF NOT DONE THIS ADDRESS
INC MRO ;BUMP TO NEXT CRAM ADDRESS
CMP MEMSZ,MRO ;DONE YET?
BNE ADR1 ;BR IF NO

5$:
ENDTST
L10046:
TRAP C$ETST

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

BGNST
T5::
MACEX
;DO NOT DO TEST IF M8200

```

(4)	012720	104432			TRAP	C\$EXIT	
(4)	012722	000230			.WORD	L10047-	
4632	012724				MYINT		
(1)	012724	013701	002516		MOV	KMCSR,R1	:RECORD DEVICE ADDR.
4633							:R1 CONTAINS BASE M8200,4,7 ADDRESS
4634	012730				MSTCLR		:MASTER CLEAR M8200,4,7
4635	012734	005037	002434		CLR	0	:MRO =CRAM ADDRESS
4636	012740				BGNSEG		
(3)	012740	104404			TRAP	C\$BSEG	
4637	012742	013702	002434	1\$:	MOV	MRO,R2	:SAVE R2 FOR TYPEOUT
4638	012746	012711	002000		MOV	#BIT10,(R1)	:SET ROMO
4639	012752	013761	002434	000004	MOV	MRO,4(R1)	:WRITE ADDRESS TO SEL4
4640	012760	013761	002434	000006	MOV	MRO,6(R1)	:LOAD SEL6 WITH WRITE DATA
4641	012766	052711	020000		BIS	#BIT13,(R1)	:WRITE CRAM
4642	012772				SKIP06	15\$	:IF M8206,SKIP NEXT INSTR.
(1)					:GOTO 15\$ IF M8206		
4643	013002	005061	000006		CLR	6(R1)	:CLEAR SEL 6
4644	013006			15\$:			
4645	013006	016104	000006		MOV	6(R1),R4	:SHOULD READ BACK OWN ADDRESS
4646	013012	023704	002434		CMP	MRO,R4	:IS DATA CORRECT?
4647	013016	001410			BEQ	2\$	:BR IF YES
4648	013020				ERROR	1	:DATA ERROR
(5)	013024	104455			TRAP	C\$ERDF	
(6)	013026	000001			.WORD	1	
(6)	013030	005055			.WORD	EMO	
(6)	013032	006032			.WORD	ERR1	
4649	013034				ESCAPE	SEG	
(3)	013034	104410			TRAP	C\$ESCAPE	
(3)	013036	000002			.WORD	10000\$-	
4650	013040			2\$:	ENDSEG		
(3)	013040			10000\$:			
(3)	013040	104405			TRAP	C\$ESEG	
4651	013042				BGNSEG		
(3)	013042	104404			TRAP	C\$BSEG	
4652	013044	005237	002434		INC	MRO	:BUMP TO NEXT ADDRESS
4653	013050	023737	002436	002434	CMP	MEMSZ,MRO	:DONE WRITING YET?
4654	013056	001331			BNE	1\$	:BR IF NO
4655	013060	005037	002434		CLR	MRO	:RESTART AT ADDRESS 0
4656	013064	013702	002434	3\$:	MOV	MRO,R2	:SAVE R2 FOR TYPEOUT
4657	013070	012711	002000		MOV	#BIT10,(R1)	:SET ROMO
4658	013074	013761	002434	000004	MOV	MRO,4(R1)	:SEL4 = CRAM ADDRESS
4659	013102	016104	000006		MOV	6(R1),R4	:READ CRAM INTO 'FOUND'
4660	013106	023704	002434		CMP	MRO,R4	:IS DATA CORRECT?
4661	013112	001411			BEQ	4\$	:BR IF YES
4662	013114				ERROR	2	:DUAL ADDRESSING ERROR
(5)	013120	104455			TRAP	C\$ERDF	
(6)	013122	000002			.WORD	2	
(6)	013124	005055			.WORD	EMO	
(6)	013126	006160			.WORD	ERR2	
4663	013130				ESCAPE	SEG	
(3)	013130	104410			TRAP	C\$ESCAPE	
(3)	013132	000002			.WORD	10001\$-	
4664	013134				ENDSEG		
(3)	013134			10001\$:			
(3)	013134	104405			TRAP	C\$ESEG	
4665	013136			4\$:			:LOOP TO 3\$ IF SW09=1

CZDMQCO M8207 STATIC DIAG #2 MACY11 30A(1052) 21-JUL-81 14:48 K 7
CZDMQC.P11 21-JUL-81 14:36 HARDWARE TESTS PAGE 20-12

SEQ 0088

```
4666 013136 005237 002434 INC MRO ;BUMP TO NEXT ADDRESS
4667 013142 023737 002436 002434 CMP MEMSZ,MRO ;DONE WRITING YET?
4668 013150 001345 BNE 3$ ;BR IF NO
4669 013152 5$:
4670 013152 ENDTST
(3) 013152 L10047:
(3) 013152 104401 TRAP C$ETST
4671
4672
4673 013154 BADHEAD
(2) ;***** TEST 6 *****
4674 ;*IOP MAIN MEMORY TEST
4675 ;*FLOAT A 1 THROUGH ALL MAIN MEMORY LOCATIONS
4676 013154 BADHEAD
(2) ;***** TEST 6 *****
4677
```

4679	013154					BGNTST			
(3)	013154					T6::			
4680	013154						MYINT		
(1)	013154	013701	002516			MOV	KMCSR,R1		:RECORD DEVICE ADDR.
4681									:R1 CONTAINS BASE M8200,4,7 ADDRESS
4682	013160					MSTCLR			:MASTER CLEAR M8200,4,7
4683	013164	005037	002406			CLR	FLAG		:START WITH ADDRESS 0
4684	013170	012737	000001	002434	1\$:	MOV	#1,MRO		:START WITH BIT 0
4685	013176	042737	003777	013232	65\$:	BIC	#3777,66\$		:CLEAR ADDRESS FIELD OF INSTRUCTION

CZDMQCO M8207 STATIC DIAG #2 MACY11 30A(1052) 21-JUL-81 14:48 M 7
CZDMQC.P11 21-JUL-81 14:36 HARDWARE TESTS PAGE 22

SEQ 0090

4687 013204 042737 000037 013240 BIC #37.68\$:CLEAR ADDRESS FIELD OF INSTRUCTION

4689	013212	153737	002406	013232	BISB	FLAG,66\$	;ADD ADDRESS TO INSTRUCTION7
4690	013220	153737	002407	013240	BISB	FLAG+1,68\$	;ADD ADDRESS TO INSTRUCTION
4691	013226				ROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	013226	004537	003044		JSR	R5,,ROMCLK	;CLOCK INSTRUCTION
4692	013232	010000			66\$:	010000	
4693	013234				ROMCLK		
(1)	013234	004537	003044		JSR	R5,,ROMCLK	;CLOCK INSTRUCTION
4694	013240	004000			68\$:	004000	;LOAD MAR HI
4695	013242	013761	002434	000004	MOV	MRO,4(R1)	;WRITE PATTERN IN PORT4
4696	013250				ROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	013250	004537	003044		JSR	R5,,ROMCLK	;CLOCK INSTRUCTION
4697	013254	122500				122500	;MOVE PORT4 TO MEMORY
4698	013256				ROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	013256	004537	003044		JSR	R5,,ROMCLK	;CLOCK INSTRUCTION
4699	013262	040620				040620	;MOVE MEMORY TO BR
4700	013264				ROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	013264	004537	003044		JSR	R5,,ROMCLK	;CLOCK INSTRUCTION
4701	013270	061225				61225	;MOVE BR TO PORT5
4702	013272	013705	002434		MOV	MRO,R5	;PUT 'EXPECTED' IN R5
4703	013276	116104	000005		MOVB	5(R1),R4	;PUT 'FOUND' IN R4
4704	013302	120504			CMPB	R5,R4	;DATA CORRECT?
4705	013304	001410			BEQ	67\$	;BR IF YES
4706	013306				ERROR	6	;DATA ERROR
(5)	013312	104455			TRAP	C\$ERDF	
(6)	013314	000006			.WORD	6	
(6)	013316	005055			.WORD	EMO	
(6)	013320	006700			.WORD	ERR6	
4707	013322				ESCAPE	TST	
(3)	013322	104410			TRAP	C\$ESCAPE	
(3)	013324	000030			.WORD	L10050-	
4708	013326				67\$:		;SW09=1?
4709	013326	000241			CLC		;CLEAR CARRY
4710	013330	106137	002434		ROLB	MRO	;SHIFT BIT IN MRO
4711	013334	001320			BNE	65\$	;DONE IF MRO-0
4712	013336				BREAK		
(3)	013336	104422			TRAP	C\$BRK	
4713	013340	005237	002406		INC	FLAG	;NEXT ADDRESS
4714	013344	023737	002436	002406	CMP	MEMSZ,FLAG	;LAST ADDRESS?
4715	013352	001306			BNE	1\$	;BR IF NO
4716	013354				2\$:		
4717	013354				ENDTST		
(3)	013354				L10050:		
(3)	013354	104401			TRAP	C\$ETST	
4718							
4719	013356				BADHEAD		
(2)					;***** TEST 7 *****		
4720					;*IOP MAIN MEMORY TEST		
4721					;*FLOAT A 0 THROUGH ALL MAIN MEMORY LOCATIONS		
4722	013356				BADHEAD		
(2)					;***** TEST 7 *****		
4723							
4724	013356				BGNTST		
(3)	013356				T7:		
4725	013356				MYINT		
(1)	013356	013701	002516		MOV	KMCSR,R1	;RECORD DEVICE ADDR.
4726							;R1 CONTAINS BASE M8200,4,7 ADDRESS

4727	013362					MSTCLR		:MASTER CLEAR M8200,4,7
4728	013366	005037	002406			CLR	FLAG	:START WITH ADDRESS 0
4729	013372	012737	000001	002434	1\$:	MOV	#1,MRO	:START WITH BIT 0
4730	013400	005137	002434		64\$:	COM	MRO	:CHANGE TO FLOATING 0
4731	013404	042737	003777	013440	65\$:	BIC	#3777,66\$	:CLEAR ADDRESS FIELD OF INSTRUCTION
4732	013412	042737	000037	013446		BIC	#37,68\$	:CLEAR ADDRESS FIELD OF INSTRUCTION
4733	013420	153737	002406	013440		BISB	FLAG,66\$	:ADD ADDRESS TO INSTRUCTION
4734	013426	153737	002407	013446		BISB	FLAG+1,68\$	:ADD ADDRESS TO INSTRUCTION
4735	013434					ROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	013434	004537	003044			JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
4736	013440	010000			66\$:	010000		:LOAD MAR LO WITH ADDRESS IN FLAG
4737	013442					ROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	013442	004537	003044			JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
4738	013446	004000			68\$:	004000		:LOAD MAR HI
4739	013450	013761	002434	000004		MOV	MRO,4(R1)	:WRITE PATTERN IN PORT4
4740	013456					ROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	013456	004537	003044			JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
4741	013462	122500				122500		:MOVE PORT4 TO MEMORY
4742	013464					ROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	013464	004537	003044			JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
4743	013470	040620				040620		:MOVE MEMORY TO BR
4744	013472					ROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	013472	004537	003044			JSR	R5,.ROMCLK	:CLOCK INSTRUCTION
4745	013476	061225				61225		:MOVE BR TO PORT5
4746	013500	013705	002434			MOV	MRO,R5	:PUT 'EXPECTED' IN R5
4747	013504	116104	000005			MOVB	5(R1),R4	:PUT 'FOUND' IN R4
4748	013510	120504				CMPS	R5,R4	:DATA CORRECT?
4749	013512	001406				BEQ	67\$	:BR IF YES
4750	013514					ERROR	6	:DATA ERROR
(5)	013520	104455				TRAP	C\$ERDF	
(6)	013522	000006				.WORD	6	
(6)	013524	005055				.WORD	EMO	
(6)	013526	006700				.WORD	ERR6	
4751	013530				67\$:	ESCAPE	TST	
(3)	013530	104410				TRAP	C\$ESCAPE	
(3)	013532	000034				.WORD	L10051-	
4752	013534	005137	002434			COM	MRO	:CHANGE TO FLOATING 1
4753	013540	000241				CLC		:CLEAR CARRY
4754	013542	106137	002434			ROLB	MRO	:SHIFT BIT IN MRO
4755	013546	001314				BNE	64\$	:DONE IF MRO 0
4756	013550					BREAK		
(3)	013550	104422				TRAP	C\$BRK	
4757	013552	005237	002406			INC	FLAG	:NEXT ADDRESS
4758	013556	023737	002436	002406		CMP	MEMSZ,FLAG	:LAST ADDRESS?
4759	013564	001302				BNE	1\$	:BR IF NO
4760	013566				2\$:			
4761	013566				ENDTST			
(3)	013566				L10051:			
(3)	013566	104401				TRAP	C\$ETST	
4762								
4763	013570					BADHEAD		
(2)						:***** TEST 8 *****		
4764						:*IOP MAIN MEMORY DUAL ADDRESSING TEST		
4765						:*LOAD EACH MEMORY LOCATION WITH ITS OWN ADDRESS		
4766						:*READ BACK EACH LOCATION TO VERIFY CORRECT ADDRESSING		
4767	013570					BADHEAD		

```

(2)
4768
4769 013570 BGNTST
(3) 013570 T8::
4770 013570 MYINT
(1) 013570 013701 002516 MOV KMCSR,R1 ;RECORD DEVICE ADDR.
4771 ;R1 CONTAINS BASE M8200,4,7 ADDRESS
4772 013574 MSTCLR ;MASTER CLEAR M8200,4,7
4773 013600 005037 002406 CLR FLAG ;START WITH ADDRESS 0
4774 013604 013702 002406 1$: MOV FLAG,R2 ;PUT DATA IN R2
4775 013610 042737 003777 013644 BIC #3777,2$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
4776 013616 042737 000037 013652 BIC #37,7$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
4777 013624 153737 002406 013644 BISB FLAG,2$ ;ADD ADDRESS TO INSTRUCTION
4778 013632 153737 002407 013652 BISB FLAG+1,7$ ;ADD ADDRESS TO INSTRUCTION
4779 013640 ROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 013640 004537 003044 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
4780 013644 010000 2$: 010000 ;LOAD MAR LO
4781 013646 ROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 013646 004537 003044 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
4782 013652 004000 7$: 004000 ;LOAD MAR HI
4783 013654 010261 000004 MOV R2,4(R1)
4784 013660 ROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 013660 004537 003044 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
4785 013664 122500 122500 ;MOVE PORT4 TO MEMORY
4786 013666 ROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 013666 004537 003044 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
4787 013672 040620 040620 ;MOVE MEMORY TO THE BR
4788 013674 ROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 013674 004537 003044 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
4789 013700 061225 61225 ;MOVE BR TO PORT5
4790 013702 010205 MOV R2,R5 ;PUT 'EXPECTED' IN R5
4791 013704 116104 000005 MOVB 5(R1),R4 ;PUT 'FOUND' IN R4
4792 013710 120504 CMPB R5,R4 ;DATA CORRECT?
4793 013712 001406 BEQ 3$ ;BR IF YES
4794 013714 ERROR 6 ;DATA ERROR
(5) 013720 104455 TRAP C$ERDF
(6) 013722 000006 .WORD 6
(6) 013724 005055 .WORD EMO
(6) 013726 006700 .WORD ERR6
4795 013730 3$: ESCAPE TST
(3) 013730 104410 TRAP C$ESCAPE
(3) 013732 000156 .WORD L10052-.
4796 013734 005237 002406 INC FLAG ;NEXT ADDRESS
4797 013740 023737 002436 002406 CMP MEMSZ,FLAG ;LAST ADDRESS?
4798 013746 001316 BNE 1$ ;BR IF NO
4799 013750 012737 013762 002340 MOV #4$,LOCK ;NEW SCOPE 1
4800 013756 005037 002406 CLR FLAG ;RESTART AT ADDRESS 0
4801 013762 013702 002406 4$: MOV FLAG,R2 ;PUT DATA IN R2
4802 013766 042737 003777 014014 BIC #3777,5$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
4803 013774 042737 000037 014022 BIC #37,8$ ;CLEAR ADDRESS FIELD OF INSTRUCTION
4804 014002 153737 002406 014014 BISB FLAG,5$ ;ADD ADDRESS TO INSTRUCTION
4805 014010 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC+5304
(1) 014010 004537 003044 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
4806 014014 010000 5$: 010000 ;LOAD THE MAR LO
4807 014016 ROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 014016 004537 003044 JSR R5,.ROMCLK ;CLOCK INSTRUCTION

```

```
4808 014022 004000      8$: 004000      ;LOAD MAR HI
4809 014024      ROMCLK      ;NEXT WORD IS INSTRUCTION,
(1) 014024 004537 003044 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
4810 014030 040620      040620 ;MOVE MEMORY TO THE BR
4811 014032      ROMCLK      ;NEXT WORD IS INSTRUCTION,
(1) 014032 004537 003044 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
4812 014036 061225      61225 ;MOV BR TO PORT5
4813 014040 010205      MOV R2,R5 ;PUT 'EXPECTED' IN R5
4814 014042 116104 000005 MOVB 5(R1),R4 ;PUT 'FOUND' IN R4
4815 014046 120504      CMPB R5,R4 ;DATA CORRECT?
4816 014050 001406      BEQ 6$ ;BR IF YES
4817 014052      ERROR 6 ;ADDRESSING ERROR
(5) 014056 104455      TRAP C$ERDF
(6) 014060 000006      .WORD 6
(6) 014062 005055      .WORD EMO
(6) 014064 006700      .WORD ERR6
4818 014066      6$: ESCAPE TST
(3) 014066 104410      TRAP C$ESCAPE
(3) 014070 000020      .WORD L10052-.
4819 014072      BREAK
(3) 014072 104422      TRAP C$BRK
4820 014074 005237 002406 INC FLAG ;NEXT ADDRESS
4821 014100 023737 002436 002406 CMP MEMSZ,FLAG ;IS IT THE LAST
4822 014106 001325      BNE 4$ ;BR IF NO
4823 014110      9$:
4824 014110      ENDTST
(3) 014110      L10052:
(3) 014110 104401      TRAP C$ETST
4825
4826 014112      BADHEAD
(2) ;***** TEST 9 *****
4827 ;*IOP MAR TEST
4828 ;*PERFORM DUAL ADDRESSING TEST
4829 ;*USING MAR AUTO-INC FEATURE
4830 014112      BADHEAD
(2) ;***** TEST 9 *****
4831
4832 014112      BGNTST
(3) 014112      T9::
4833 014112      K4ONLY ;FOR 4K CPUS ONLY.
(1) ;DO NOT DO TEST IF M8200, OR M8204
(4) 014122 104432      TRAP C$EXIT
(4) 014124 000342      .WORD L10053-.
4834 014126      MYINT
(1) 014126 013701 002516 MOV KMCSR,R1 ;RECORD DEVICE ADDR.
4835 ;R1 CONTAINS BASE M8200,4,7 ADDRESS
4836 014132      MSTCLR ;MASTER CLEAR M8200,4,7
4837 014136 005002      CLR R2 ;START WITH A ZERO
4838 014140 013703 002436 MOV MEMSZ,R3 ;GET MEMORY SIZE
4839 014144 005203      INC R3 ;STOP ADDR=MEMSZ+1
4840 014146      ROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 014146 004537 003044 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
4841 014152 010000      010000 ;LOAD MAR WITH A ZERO
4842 014154      CLRMAR
(2) 014154 004537 003044 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
4843 014162 010261 000004 1$: MOV R2,4(R1) ;WRITE DATA TO PORT4
```

4844	014166			ROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	014166	004537	003044	JSR	R5,,ROMCLK	;CLOCK INSTRUCTION
4845	014172	136500		136500		;MEM PORT4, AUTO-INC MAR
4846	014174	005202		INC	R2	;INCREMENT DATA
4847	014176	020302		CMP	R3,R2 ;DONE YET?	
4848	014200	001370		BNE	1\$	;BR IF NO
4849	014202	005002		CLR	R2	;RESTART WITH A ZERO
4850	014204			ROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	014204	004537	003044	JSR	R5,,ROMCLK	;CLOCK INSTRUCTION
4851	014210	010000		010000		;LOAD MAR WITH A ZERO
4852	014212			CLRMAR		
(2)	014212	004537	003044	JSR	R5,,ROMCLK	;CLOCK INSTRUCTION
4853	014220					
4854	014220			SROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	014220	004537	003100	JSR	R5,,SROMCLK	
4855	014224	055224		055224		;MOVE MEM TO PORT4
4856	014226	010205		MOV	R2,R5	;PUT 'EXPECTED' IN R5
4857	014230	116104	000004	MOVB	4(R1),R4	;PUT 'FOUND' IN R4
4858	014234	120504		CMPB	R5,R4	;DATA CORRECT?
4859	014236	001406		BEQ	3\$	;BR IF YES
4860	014240			ERROR	11	;MAR ERROR
(5)	014244	104455		TRAP	C\$ERDF	
(6)	014246	000013		.WORD	11	
(6)	014250	005055		.WORD	EMO	
(6)	014252	007276		.WORD	ERR11	
4861	014254			SROMCLK		
(1)	014254	004537	003100	JSR	R5,,SROMCLK	
4862	014260	000000		0		;DUMP NOP INSTR. TO CLK AUTO INC IN MAR.
4863	014262	005004		CLR	R4	
4864	014264			ROMCLK		;READ IBUS* <15> (MAR HIGH)
(1)	014264	004537	003044	JSR	R5,,ROMCLK	;CLOCK INSTRUCTION
4865	014270	121325		121325		;MAR HIGH _POT 5
4866						
4867	014272			ROMCLK		;READ IBUS* <14> (MAR LOW)
(1)	014272	004537	003044	JSR	R5,,ROMCLK	;CLOCK INSTRUCTION
4868	014276	121304		121304		
4869	014300	016104	000004	MOV	4(R1),R4	;ADD TO MAR HIGH.
4870	014304	042704	160000	BIC	#160000,R4	
4871	014310	005202		INC	R2	
4872	014312	020237	002436	CMP	R2,MEMSZ	
4873	014316	001002		BNE	35\$	
4874	014320	052702	010000	BIS	#10000,R2	;IF AT HIGH LIMIT,ADD IN OVERFLOW BIT.
4875	014324					
4876	014324	020204		CMP	R2,R4 ;ADDR. OK?	
4877	014326	001406		BEQ	4\$	
4878	014330			ERROR	11	;ERROR MAR ADDR. BAD IN IBUS <14>AND <15>
(5)	014334	104455		TRAP	C\$ERDF	
(6)	014336	000013		.WORD	11	
(6)	014340	005055		.WORD	EMO	
(6)	014342	007276		.WORD	ERR11	
4879						;EXPECTED (R4) IS COMBINATION OF
4880						;IBUS* <14> AND <15>
4881	014344					
4882	014344			ESCAPE	TST	
(3)	014344	104410		TRAP	C\$ESCAPE	
(3)	014346	000120		.WORD	L10053-	

4883	014350			BREAK		
(3)	014350	104422		TRAP	C\$BRK	
4884	014352	032702	010000	BIT	#10000,R2	;DONE YET?
4885	014356	001720		BEQ	2\$	
4886				;		
4887				;		
4888				;		
4889				;		
4890				;		
4891	014360			SKIP06	40\$	
(1)				;	GOTO 40\$ IF M8206	
4892	014370	005737	002470	TST	RUNB	
4893	014374	001034		BNE	40\$	
4894	014376	005737	002472	TST	RUNINH	
4895	014402	001031		BNE	40\$	
4896	014404	052711	040000	BIS	#40000,(R1)	;SET MASTER CLEAR
4897	014410	005011		CLR	(R1)	;CLEAR MASTER CLEAR
4898	014412			ROMCLK		;WE MUST FIRST CLOCK
(1)	014412	004537	003044	JSR	R5,.ROMCLK	;CLOCK INSTRUCTION
4899	014416	121325		121325		;THE MAR LATCH REGS
4900	014420			ROMCLK		;BEFORE WE CAN READ THEM
(1)	014420	004537	003044	JSR	R5,.ROMCLK	;CLOCK INSTRUCTION
4901	014424	121304		121304		
4902	014426			ROMCLK		;READ IBUS* <15> PUT IN PORT5
(1)	014426	004537	003044	JSR	R5,.ROMCLK	;CLOCK INSTRUCTION
4903	014432	121325		121325		;MAR HIGH
4904	014434			ROMCLK		;READ IBUS* <14>, PUT IN PORT4
(1)	014434	004537	003044	JSR	R5,.ROMCLK	;CLOCK INSTRUCTION
4905	014440	121304		121304		;MAR LOW
4906	014442	005002		CLR	R2	;EXPECT MAR CLEAR
4907	014444	016104	000004	MOV	4(R1),R4	;READ PORTS 4&5. THEY CONTAIN
4908						;THE CONTENTS OF THE MAR
4909						;MASTER CLEAR SHOULD HAVE
4910						;CLEARED THE MAR
4911						;BRANCH END TST IF CLEAR
4912	014450	001406		BEQ	40\$	
4913	014452			ERROR	44	
(5)	014456	104455		TRAP	C\$ERDF	
(6)	014460	000054		.WORD	44	
(6)	014462	005055		.WORD	EMO	
(6)	014464	011054		.WORD	ERR44	
4914	014466					
4915	014466					
(3)	014466					
(3)	014466	104401		TRAP	C\$ETST	
4916						
4917	014470			BADHEAD		
(2)				;	***** TEST 10 *****	
4918				;	*IOP (CRAM) ODT BITS TEST	
4919				;	*LOAD MAR WITH A 0 INC MAR UNTIL IT OVERFLOWS	
4920				;	*VERIFY THAT IBUS* 10 BITS IS SET ONLY WHEN MAR BIT 8 IS A ONE	
4921				;	*AND THAT IBUS* 10 BIT6 IS SET ON MAR OVERFLOW	
4922	014470			BADHEAD		
(2)				;	***** TEST 10 *****	
4923						
4924	014470					

BGNTST

(3)	014470			110::	MACEX		
4925	014470				:DO NOT DO TEST IF M8200		
(1)					TRAP C\$EXIT		
(4)	014476	104432			.WORD L10054-		
(4)	014500	000234			MYJNT		
4926	014502				MOV KMCSR,R1		:RECORD DEVICE ADDR.
(1)	014502	013701	002516				:R1 CONTAINS BASE M8200,4,7 ADDRESS
4927							:MASTER CLEAR M8200,4,7
4928	014506				MSTCLR		:R2-SAME AS MAR CONTENTS
4929	014512	005002			CLR R2		:NEXT WORD IS INSTRUCTION,
4930	014514				ROMCLK		:CLOCK INSTRUCTION
(1)	014514	004537	003044		JSR R5,.ROMCLK		:MAR_0
4931	014520	010000			010000		
4932	014522			15:			
4933	014522				ROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	014522	004537	003044		JSR R5,.ROMCLK		:CLOCK INSTRUCTION
4934	014526	121204			121204		:PORT4=IBUS*10
4935	014530	005005			CLR R5		:R5 'EXPECTED'
4936	014532	032702	000400		BIT #BIT8,R2		:IS BIT8 SET IN MAR?
4937	014536	001402			BEQ .+6		:BR IF NO
4938	014540	012705	000040		MCV #BIT5,R5		:IF YES THEN SET BITS
4939	014544	016104	000004		MOV 4(R1),R4		:R4='FOUND'
4940	014550	042704	177637		BIC #177637,R4		:CLEAR UNWANTED BITS
4941	014554	020504			CMP R5,R4		:BITS 5&6 SHOULD BE CLEAR
4942	014556	001410			BEQ 15\$		:BR IF OK
4943	014560				ERROR 7		:ERROR BITS 5&6 NOT CLEAR
(5)	014564	104455			TRAP C\$ERDF		
(6)	014566	000007			.WORD 7		
(6)	014570	005055			.WORD EMO		
(6)	014572	007026			.WORD ERR7		
4944	014574				ESCAPE TST		
(3)	014574	104410			TRAP C\$ESCAPE		
(3)	014576	000136			.WORD L10054-		
4945	014600			15\$:			
4946	014600				ROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	014600	004537	003044		JSR R5,.ROMCLK		:CLOCK INSTRUCTION
4947	014604	014000			014000		:INC MAR
4948	014606	005202			INC R2		:BUMP MEM ADDRESS
4949	014610	022702	002000		CMP #2000,R2		:OVERFLOWED YET?(OVFL PAGE BITS).
4950	014614	001342			BNE 15\$		:BR IF NO
4951	014616				ROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	014616	004537	003044		JSR R5,.ROMCLK		:CLOCK INSTRUCTION
4952	014622	121204			121204		:PART4 IBUS* 10
4953	014624	012705	000100		MOV #BIT6,R5		:R5='EXPECTED'
4954	014630	016104	000004		MOV 4(R1),R4		:R4='FOUND'
4955	014634	042704	177627		BIC #177627,R4		:CLEAR UNWANTED BITS
4956	014640	020504			CMP R5,R4		:BIT6 SHOULD BE SET
4957	014642	001406			BEQ 17\$		:BR IF OK
4958	014644				ERROR 7		:ERROR, BIT6 NOT SET
(5)	014650	104455			TRAP C\$ERDF		
(6)	014652	000007			.WORD 7		
(6)	014654	005055			.WORD EMO		
(6)	014656	007026			.WORD ERR7		
4959	014660			17\$:			
(1)	014660	004537	003044		ROMCLK		:NEXT WORD IS INSTRUCTION,
4960	014664	010000			JSR R5,.ROMCLK		:CLOCK INSTRUCTION
					010000		:MAR_0

4961	014666			ROMCLK		:NEXT WORD IS INSTRUCTION.
(1)	014666	004537	003044	JSR	R5,ROMCLK	:CLOCK INSTRUCTION
4962	014672	004000		004000		:MAR HI 0
4963	014674			ROMCLK		:NEXT WORD IS INSTRUCTION.
(1)	014674	004537	003044	JSR	R5,ROMCLK	:CLOCK INSTRUCTION
4964	014700	121204		121204		:PORT4 IBUS* 10
4965	014702	005005		CLR	R5	:R5='EXPECTED'
4966	014704	016104	000004	MOV	4(R1),R4	:R4='FOUND'
4967	014710	042704	177637	BIC	#177637,R4	:CLEAR UNWANTED BITS
4968	014714	020504		CMP	R5,R4	:BITS 5&6 SHOULD BE CLEAR
4969	014716	001406		BEQ	2\$	:BR IF OK
4970	014720			ERROR	7	:ERROR 5&6 NOT BOTH CLEAR
(5)	014724	104455		TRAP	C\$ERDF	
(6)	014726	000007		.WORD	7	
(6)	014730	005055		.WORD	EMO	
(6)	014732	007026		.WORD	ERR7	
4971	014734					
4972	014734					
(3)	014734					
(3)	014734	104401		TRAP	C\$ETST	
4973						
4974	014736			BADHEAD		
(2)				:***** TEST 11 *****		
4975				:*(CRAM TEST OF JUMP(J) NEVER MICRO-PROCESSOR INSTRUCTION.		
4976				:*PERFORM THE JUMP INSTRUCTION		
4977				:*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION		
4978				:*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE		
4979				:*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT		
4980				:*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT		
4981				:*THE CRAM PC IS CORRECT. IF THE CRAM PC IS NOT RIGHT,		
4982				:*THEN PORT4 CONTAINS A 37		
4983	014736			BADHEAD		
(2)				:***** TEST 11 *****		
4984						
4985	014736			BGNTST		
(3)	014736			T11::		
4986	014736			SKIP04	10\$	
(1)				:GOTO 10\$ IF M8204		
4987	014746			EXIT	TST	:CAN'T DO IF ROM,4K
(3)	014746	104432		TRAP	C\$EXIT	
(3)	014750	000230		.WORD	L10055-	
4988	014752					
4989	014752			10\$:		
(1)	014752	013701	002516	MYINT		
4990				MOV	KMCSR,R1	:RECORD DEVICE ADDR.
4991	014756					:R1 CONTAINS BASE M8200,4,7 ADDRESS
4992	014762			MSTCLR		:MASTER CLEAR M8200,4,7
(3)	014762	104404		BGNSEG		
4993	014764	004737	003474	TRAP	C\$BSEG	
4994	014770			JSR	PC,MEMSET	:SET MEM AND RAM
4995	014770	004737	003166			
4996	014774			JSR	PC,CLRALL	:CLEAR ALL CONDITIONS
(1)	014774	004537	003100	SROMCLK		:NEXT WORD IS INSTRUCTION.
4997	015000	100400		JSR	R5,SROMCLK	
4998	015002			100400		:START AT ROM PC=0
(1)	015002	004537	003100	SROMCLK		:NEXT WORD IS INSTRUCTION.
				JSR	R5,SROMCLK	

4999	015006	114377		114377.<400*0>	;JUMP TO ROM PC OF 1777
5000	015010	004737	003330	JSR PC,RAMDAT	;R4=CRAM PC (LSB 8 BITS)
5001	015014	000001		1	;EXPECTED DATA
5002	015016	120504		CMPB R5,R4	;IS ROM PC CORRECT?
5003	015020	001406		BEQ 2\$	;BR IF NO
5004	015022			ERROR 5	;ERROR, CRAM PC IS WRONG
(5)	015026	104455		TRAP C\$ERDF	
(6)	015030	000005		.WORD 5	
(6)	015032	005055		.WORD EM0	
(6)	015034	006556		.WORD ERR5	
5005	015036			2\$: ESCAPE SEG	
(3)	015036	104410		TRAP C\$ESCAPE	
(3)	015040	000002		.WORD 10000\$-	
5006	015042			ENDSEG	
(3)	015042			10000\$: TRAP C\$ESEG	
(3)	015042	104405		BGNSEG	
5007	015044			TRAP C\$BSEG	
(3)	015044	104404		JSR PC,CLRALL	;CLEAR ALL CONDITIONS
5008	015046	004737	003166	SROMCLK	;NEXT WORD IS INSTRUCTION,
5009	015052			JSR R5,.SROMCLK	
(1)	015052	004537	003100	100403	;START AT ROM PC=3
5010	015056	100403		SROMCLK	;NEXT WORD IS INSTRUCTION,
(1)	015060	004537	003100	JSR R5,.SROMCLK	
5012	015064	100000		100000!<400*0>	;JUMP TO ROM PC OF 0
5013	015066	004737	003330	JSR PC,RAMDAT	;R4=CRAM PC (LSB 8 BITS)
5014	015072	000004		4	;EXPECTED DATA
5015	015074	120504		CMPB R5,R4	;IS ROM PC CORRECT?
5016	015076	001406		BEQ 4\$	;BR IF YES
5017	015100			ERROR 5	;ERROR, CROM PC IS WRONG
(5)	015104	104455		TRAP C\$ERDF	
(6)	015106	000005		.WORD 5	
(6)	015110	005055		.WORD EM0	
(6)	015112	006556		.WORD ERR5	
5018	015114			4\$: ESCAPE SEG	
(3)	015114	104410		TRAP C\$ESCAPE	
(3)	015116	000002		.WORD 10001\$-	
5019	015120			ENDSEG	
(3)	015120			10001\$: TRAP C\$ESEG	
(3)	015120	104405		BGNSEG	
5020	015122			TRAP C\$BSEG	
(3)	015122	104404		JSR PC,CLRALL	;CLEAR ALL CONDITINS
5021	015124	004737	003166	SROMCLK	;NEXT WORD IS INSTRUCTION,
5022	015130			JSR R5,.SROMCLK	
(1)	015130	004537	003100	100406	;START AT ROM PC=6
5023	015134	100406		SROMCLK	;NEXT WORD IS INSTRUCTION,
5024	015136			JSR R5,.SROMCLK	
(1)	015136	004537	003100	104125!<400*0>	;JUMP TO ROM PC OF 525
5025	015142	104125		JSR PC,RAMDAT	;R4=CRAM PC (LSB 8 BITS)
5026	015144	004737	003330	7	;EXPECTED DATA
5027	015150	000007		CMPB R5,R4	;IS ROM PC CORRECT?
5028	015152	120504		BEQ 6\$	;BR IF YES
5029	015154	001406		ERROR 5	;ERROR, CRAM PC IS WRONG
5030	015156			TRAP C\$ERDF	
(5)	015162	104455		.WORD 5	
(6)	015164	000005			

(6)	015166	005055			.WORD	EM0	
(6)	015170	006556			.WORD	ERR5	
5031	015172			6\$:	ESCAPE	SEG	
(3)	015172	104410			TRAP	C\$ESCAPE	
(3)	015174	000002			.WORD	10002\$-	
5032	015176				ENDSEG		
(3)	015176			10002\$:			
(3)	015176	104405			TRAP	C\$ESEG	
5033	015200			ENDTST			
(3)	015200			L10055:			
(3)	015200	104401			TRAP	C\$ETST	
5034							
5035	015202				BADHEAD		
(2)					***** TEST 12 *****		
5036					*CRAM TEST OF JUMP(I) ALWAYS MICRO-PROCESSOR INSTRUCTION.		
5037					*PERFORM THE JUMP INSTRUCTION		
5038					*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION		
5039					*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE		
5040					*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT		
5041					*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT		
5042					*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL		
5043					*THEN PORT4 WILL CONTAIN A 37		
5044	015202				BADHEAD		
(2)					***** TEST 12 *****		
5045							
5046	015202			BGNTST			
(3)	015202			T12::			
5047	015202				MACEX2		;DON'T DO IF M8200
(1)					;DO NOT DO TEST IF M8200		
(4)	015212	104432			TRAP	C\$EXIT	
(4)	015214	000214			.WORD	L10056-	
5048	015216				MYINT		
(1)	015216	013701	002516		MOV	KMCSR,R1	;RECORD DEVICE ADDR.
5049							;R1 CONTAINS BASE M8200,4,7 ADDRESS
5050	015222				MSTCLR		;MASTER CLEAR M8200,4,7
5051	015226	004737	003474		JSR	PC,MEMSET	;SET MEM AND RAM
5052	015232			1\$:	BGNSEG		
(3)	015232	104404			TRAP	C\$BSEG	
5053	015234				SROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	015234	004537	003100		JSR	R5,.SROMCLK	
5054	015240	100400			100400		;START AT ROM PC=0
5055	015242				SROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	015242	004537	003100		JSR	R5,.SROMCLK	
5056	015246	114777			114377!<400*1>		;JUMP TO ROM PC OF 1777
5057	015250	004737	003330		JSR	PC,RAMDAT	;R4=CRAM PC (LSB 8 BITS)
5058	015254	000377			377		;EXPECTED DATA
5059	015256	120504			CMPB	R5,R4	;IS ROM PC CORRECT?
5060	015260	001406			BEQ	2\$	;BR IF YES
5061	015262				ERROR	5	;ERROR, CRAM PC IS WRONG
(5)	015266	104455			TRAP	C\$ERDF	
(6)	015270	000005			.WORD	5	
(6)	015272	005055			.WORD	EM0	
(6)	015274	006556			.WORD	ERR5	
5062	015276			2\$:	ESCAPE	SEG	
(3)	015276	104410			TRAP	C\$ESCAPE	
(3)	015300	000002			.WORD	10000\$-	

5063	015302			ENDSEG	
(3)	015302			10000\$:	
(3)	015302	104405		TRAP	C\$ESEG
5064	015304			BGNSEG	
(3)	015304	104404		TRAP	C\$BSEG
5065	015306			SROMCLK	
(1)	015306	004537	003100	JSR	R5, .SROMCLK ;NEXT WORD IS INSTRUCTION,
5066	015312	100403		100403	;START AT ROM PC=3
5067	015314			SROMCLK	;NEXT WORD IS INSTRUCTION,
(1)	015314	004537	003100	JSR	R5, .SROMCLK
5068	015320	100400		100000!<400*1>	;JUMP TO ROM PC OF 0
5069	015322	004737	003330	JSR	PC, RAMDAT ;R4=CRAM PC (LSB 8 BITS)
5070	015326	000000		0	;EXPECTED DATA
5071	015330	120504		CMPB	R5, R4 ;IS ROM PC CORRECT?
5072	015332	001406		BEQ	4\$;BR IF YES
5073	015334			ERROR	5 ;ERROR, CRAM PC IS WRONG
(5)	015340	104455		TRAP	C\$ERDF
(6)	015342	000005		.WORD	5
(6)	015344	005055		.WORD	EMO
(6)	015346	006556		.WORD	ERR5
5074	015350			4\$:	ESCAPE SEG
(3)	015350	104410		TRAP	C\$ESCAPE
(3)	015352	000002		.WORD	10001\$-
5075	015354			ENDSEG	
(3)	015354			10001\$:	
(3)	015354	104405		TRAP	C\$ESEG
5076	015356			BGNSEG	
(3)	015356	104404		TRAP	C\$BSEG
5077	015360			SROMCLK	
(1)	015360	004537	003100	JSR	R5, .SROMCLK ;NEXT WORD IS INSTRUCTION,
5078	015364	100406		100406	;START AT ROM PC=6
5079	015366			SROMCLK	;NEXT WORD IS INSTRUCTION,
(1)	015366	004537	003100	JSR	R5, .SROMCLK
5080	015372	104525		104125!<400*1>	;JUMP TO ROM PC OF 525
5081	015374	004737	003330	JSR	PC, RAMDAT ;R4=CRAM PC (LSB 8 BITS)
5082	015400	000125		125	;EXPECTED DATA
5083	015402	120504		CMPB	R5, R4 ;IS ROM PC CORRECT?
5084	015404	001406		BEQ	6\$;BR IF YES
5085	015406			ERROR	5 ;ERROR, CRAM PC IS WRONG
(5)	015412	104455		TRAP	C\$ERDF
(6)	015414	000005		.WORD	5
(6)	015416	005055		.WORD	EMO
(6)	015420	006556		.WORD	ERR5
5086	015422			6\$:	ESCAPE SEG
(3)	015422	104410		TRAP	C\$ESCAPE
(3)	015424	000002		.WORD	10002\$-
5087	015426			ENDSEG	
(3)	015426			10002\$:	
(3)	015426	104405		TRAP	C\$ESEG
5088	015430			ENDTST	
(3)	015430			L10056:	
(3)	015430	104401		TRAP	C\$ETST
5089				BADHEAD	
(2)				***** TEST 13 *****	
5091				;(CRAM TEST OF JUMP(I) ON C BIT SET MICRO-PROCESSOR INSTRUCTION.	

```

5092 ;*SET THE C BIT, PERFORM THE JUMP INSTRUCTION.
5093 ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
5094 ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
5095 ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
5096 ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
5097 ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
5098 ;*THEN PORT4 WILL CONTAIN A 37
5099 015432 BADHEAD
(2) ;***** TEST 13 *****
5100
5101 015432 BGNTST
(3) 015432 T13::
5102 015432 MACEX2 ;DON'T DO IF M8200
(1) ;DO NOT DO TEST IF M8200
(4) 015442 104432 TRAP C$EXIT
(4) 015444 000230 .WORD L10057-.
5103 015446 MYINT
(1) 015446 013701 002516 MOV KMCSR,R1 ;RECORD DEVICE ADDR.
5104 ;R1 CONTAINS BASE M8200,4,7 ADDRESS
5105 015452 MSTCLR ;MASTER CLEAR M8200,4,7
5106 015456 004737 003474 JSR PC,MEMSET ;SET MEM AND RAM
5107 015462 1$: BGNSEG
(3) 015462 104404 TRAP C$BSEG
5108 015464 004737 003260 JSR PC,SETC ;SET THE C BIT'
5109 015470 SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 015470 004537 003100 JSR R5,.SROMCLK
5110 015474 100400 ;START AT ROM PC=0
5111 015476 SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 015476 004537 003100 JSR R5,.SROMCLK
5112 015502 115377 114377!<400*2> ;JUMP TO ROM PC OF 1777
5113 015504 004737 003330 JSR PC,RAMDAT ;R4=CRAM PC (LSB 8 BITS)
5114 015510 000377 ;EXPECTED DATA
5115 015512 120504 CMPB R5,R4 ;IS ROM PC CORRECT?
5116 015514 001406 BEQ 2$ ;BR IF YES
5117 015516 ERROR 5 ;ERROR, CRAM PC IS WRONG
(5) 015522 104455 TRAP C$ERDF
(6) 015524 000005 .WORD 5
(6) 015526 005055 .WORD EMO
(6) 015530 006556 .WORD ERR5
5118 015532 2$: ;LOOP TO 1$ IF SW09=1
5119 015532 ESCAPE SEG
(3) 015532 104410 TRAP C$ESCAPE
(3) 015534 000002 .WORD 10000$-.
5120 015536 ENDSEG
(3) 015536 104405 10000$: TRAP C$ESEG
5121 015540 BGNSEG
(3) 015540 104404 TRAP C$BSEG
5122 015542 004737 003260 JSR PC,SETC ;SET THE C BIT'
5123 015546 SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 015546 004537 003100 JSR R5,.SROMCLK
5124 015552 100403 ;START AT ROM PC=3
5125 015554 SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 015554 004537 003100 JSR R5,.SROMCLK
5126 015560 101000 100000!<400*2> ;JUMP TO ROM PC OF 0
5127 015562 004737 003330 JSR PC,RAMDAT ;R4=CRAM PC (LSB 8 BITS)

```

```

0                                ;EXPECTED DATA
CMPB    R5,R4                  ;IS ROM PC CORRECT?
BEQ     4$                      ;BR IF YES
ERROR   5                      ;ERROR, CRAM PC IS WRONG
TRAP    C$ERDF
.WORD   5
.WORD   EMO
.WORD   ERR5

4$:                                ;LOOP TO 3$ IF SW09=1
ESCAPE  SEG
TRAP    C$ESCAPE
.WORD   10001$-.
ENDSEG

10001$:
TRAP    C$ESEG
BGNSEG
TRAP    C$BSEG
JSR     PC,SETC                ;SET THE C BIT'
SR0MCLK JSR     R5,.SR0MCLK      ;NEXT WORD IS INSTRUCTION,
100406  ;START AT ROM PC=6
SR0MCLK ;NEXT WORD IS INSTRUCTION,
JSR     R5,.SR0MCLK
104125!<400*2>                ;JUMP TO ROM PC OF 525
JSR     PC,RAMDAT              ;R4=CRAM PC (LSB 8 BITS)
125     ;EXPECTED DATA
CMPB    R5,R4                  ;IS ROM PC CORRECT?
BEQ     6$                      ;BR IF YES
ERROR   5                      ;ERROR, CRAM PC IS WRONG
TRAP    C$ERDF
.WORD   5
.WORD   EMO
.WORD   ERR5

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

10002$:
TRAP    C$ESEG

ENDTST
L10057:
TRAP    C$ETST

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

```

5161	015676			BGNTST		
(3)	015676			T14::		
5162	015676			MACEX2		;DON'T DO IF M8200.
(1)				;DO NOT DO TEST IF M8200		
(4)	015706	104432		TRAP	C\$EXIT	
(4)	015710	000230		.WORD	L10060-	
5163	015712			MYINT		
(1)	015712	013701	002516	MOV	KMCSR,R1	;RECORD DEVICE ADDR.
5164						;R1 CONTAINS BASE M8200,4,7 ADDRESS
5165	015716			MSTCLR		;MASTER CLEAR M8200,4,7
5166	015722	004737	003474	JSR	PC,MEMSET	;SET MEM AND RAM
5167	015726			1\$: BGNSEG		
(3)	015726	104404		TRAP	C\$BSEG	
5168	015730	004737	003312	JSR	PC,SETZ	;SET THE Z BIT'
5169	015734			SROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	015734	004537	003100	JSR	R5,.SROMCLK	
5170	015740	100400		100400		;START AT ROM PC=0
5171	015742			SROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	015742	004537	003100	JSR	R5,.SROMCLK	
5172	015746	115777		114377!<400*3>		;JUMP TO ROM PC OF 1777
5173	015750	004737	003330	JSR	PC,RAMDAT	;R4=CRAM PC (LSB 8 BITS)
5174	015754	000377		377		;EXPECTED DATA
5175	015756	120504		CMPB	R5,R4	;IS ROM PC CORRECT?
5176	015760	001406		BEQ	2\$	;BR IF YES
5177	015762			ERROR	5	;ERROR, CRAM PC IS WRONG
(5)	015766	104455		TRAP	C\$ERDF	
(6)	015770	000005		.WORD	5	
(6)	015772	005055		.WORD	EMO	
(6)	015774	006556		.WORD	ERR5	
5178	015776			2\$: ESCAPE	SEG	
(3)	015776	104410		TRAP	C\$ESCAPE	
(3)	016000	000002		.WORD	10000\$-	
5179	016002			ENDSEG		
(3)	016002			10000\$: TRAP	C\$ESEG	
(3)	016002	104405		BGNSEG		
5180	016004			TRAP	C\$BSEG	
(3)	016004	104404		JSR	PC,SETZ	;SET THE Z BIT'
5181	016006	004737	003312	SROMCLK		;NEXT WORD IS INSTRUCTION,
5182	016012			JSR	R5,.SROMCLK	
(1)	016012	004537	003100	100403		;START AT ROM PC=3
5183	016016	100403		SROMCLK		;NEXT WORD IS INSTRUCTION,
5184	016020			JSR	R5,.SROMCLK	
(1)	016020	004537	003100	100000!<400*3>		;JUMP TO ROM PC OF 0
5185	016024	101400		JSR	PC,RAMDAT	;R4=CRAM PC (LSB 8 BITS)
5186	016026	004737	003330	0		;EXPECTED DATA
5187	016032	000000		CMPB	R5,R4	;IS ROM PC CORRECT?
5188	016034	120504		BEQ	4\$	;BR IF YES
5189	016036	001406		ERROR	5	;ERROR, CRAM PC IS WRONG
5190	016040			TRAP	C\$ERDF	
(5)	016044	104455		.WORD	5	
(6)	016046	000005		.WORD	EMO	
(6)	016050	005055		.WORD	ERR5	
(6)	016052	006556		4\$: ESCAPE	SEG	
5191	016054			TRAP	C\$ESCAPE	
(3)	016054	104410		.WORD	10001\$-	
(3)	016056	000002				

```
5192 016060          ENDSEG
(3) 016060          10001$:
(3) 016060 104405    TRAP    C$ESEG
5193 016062          BGNSEG
(3) 016062 104404    TRAP    C$BSEG
5194 016064 004737 003312 JSR    PC,SETZ      ;SET THE Z BIT'
5195 016070          SRGMCLK      ;NEXT WORD IS INSTRUCTION,
(1) 016070 004537 003100 JSR    R5,.SRGMCLK
5196 016074 100406    100406      ;START AT ROM PC=6
5197 016076          SRGMCLK      ;NEXT WORD IS INSTRUCTION,
(1) 016076 004537 003100 JSR    R5,.SRGMCLK
5198 016102 105525    104125!<400*3> ;JUMP TO ROM PC OF 525
5199 016104 004737 003330 JSR    PC,RAMDAT ;R4=CRAM PC (LSB 8 BITS)
5200 016110 000125    125          ;EXPECTED DATA
5201 016112 120504    CMPB    R5,R4    ;IS ROM PC CORRECT?
5202 016114 001406    BEQ     6$      ;BR IF YES
5203 016116          ERROR    5      ;ERROR, CRAM PC IS WRONG
(5) 016122 104455    TRAP    C$ERDF
(6) 016124 000005    .WORD    5
(6) 016126 005055    .WORD    EMO
(6) 016130 006556    .WORD    ERR5
5204 016132          6$:
(3) 016132 104410    ESCAPE SEG
(3) 016134 000002    TRAP    C$ESCAPE
5205 016136          .WORD    10002$-.
(3) 016136          ENDSEG
(3) 016136 104405    10002$:
5206 016140          TRAP    C$ESEG
(3) 016140          ENDTST
(3) 016140 104401    L10060:
5207          TRAP    C$ETST
5208 016142          BADHEAD
(2)          ;***** TEST 15 *****
5209          ;*CRAM TEST OF JUMP(I) ON BRO SET MICRO-PROCESSOR INSTRUCTION.
5210          ;*SET THE BRO BIT, PERFORM THE JUMP INSTRUCTION.
5211          ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
5212          ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
5213          ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
5214          ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
5215          ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
5216          ;*THEN PORT4 WILL CONTAIN A 37
5217 016142          BADHEAD
(2)          ;***** TEST 15 *****
5218          BGNTEST
5219 016142          T15::
(3) 016142          MACEX2      ;DON'T DO IF M8200.
5220 016142          ;DO NOT DO TEST IF M8200
(1)          TRAP    C$EXIT
(4) 016152 104432    .WORD    L10061-.
(4) 016154 000230    MYINT
5221 016156          MOV     KMCSR,R1      ;RECORD DEVICE ADDR.
(1) 016156 013701 002516      ;R1 CONTAINS BASE M8200,4,7 ADDRESS
5222          MSTCLR      ;MASTER CLEAR M8200,4,7
5223 016162          JSR    PC,MEMSET      ;SET MEM AND RAM
5224 016166 004737 003474    1$:
5225 016172
```

5226	016172			BGNSEG		
(3)	016172	104404		TRAP	C\$BSEG	
5227	016174	004737	003220	JSR	PC,SETBRO	;SET THE BRO BIT'
5228	016200			SROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	016200	004537	003100	JSR	R5,..SROMCLK	
5229	016204	100400		100400		;START AT ROM PC=0
5230	016206			SROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	016206	004537	003100	JSR	R5,..SROMCLK	
5231	016212	116377		114377!<400*4>		;JUMP TO ROM PC OF 1777
5232	016214	004737	003330	JSR	PC,RAMDAT	;R4=CRAM PC (LSB 8 BITS)
5233	016220	000377		377		;EXPECTED DATA
5234	016222	120504		CMPB	R5,R4	;IS ROM PC CORRECT?
5235	016224	001406		BEQ	2\$	;BR IF YES
5236	016226			ERROR	5	;ERROR, CRAM PC IS WRONG
(5)	016232	104455		TRAP	C\$ERDF	
(6)	016234	000005		.WORD	5	
(6)	016236	005055		.WORD	EMO	
(6)	016240	006556		.WORD	ERR5	
5237	016242			2\$: ESCAPE SEG		
(3)	016242	104410		TRAP	C\$ESCAPE	
(3)	016244	000002		.WORD	10000\$-	
5238	016246			ENDSEG		
(3)	016246			00000\$: TRAP	C\$ESEG	
(3)	016246	104405		BGNSEG		
5239	016250			TRAP	C\$BSEG	
(3)	016250	104404		JSR	PC,SETBRO	;SET THE BRO BIT'
5240	016252	004737	003220	SROMCLK		;NEXT WORD IS INSTRUCTION,
5241	016256			JSR	R5,..SROMCLK	
(1)	016256	004537	003100	100403		;START AT ROM PC=3
5242	016262	100403		SROMCLK		;NEXT WORD IS INSTRUCTION,
5243	016264			JSR	R5,..SROMCLK	
(1)	016264	004537	003100	100000!<400*4>		;JUMP TO ROM PC OF 0
5244	016270	102000		JSR	PC,RAMDAT	;R4=CRAM PC (LSB 8 BITS)
5245	016272	004737	003330	0		;EXPECTED DATA
5246	016276	000000		CMPB	R5,R4	;IS ROM PC CORRECT?
5247	016300	120504		BEQ	4\$	;BR IF YES
5248	016302	001406		ERROR	5	;ERROR, CRAM PC IS WRONG
5249	016304			TRAP	C\$ERDF	
(5)	016310	104455		.WORD	5	
(6)	016312	000005		.WORD	EMO	
(6)	016314	005055		.WORD	ERR5	
(6)	016316	006556		4\$: ESCAPE SEG		
5250	016320			TRAP	C\$ESCAPE	
(3)	016320	104410		.WORD	10001\$-	
(3)	016322	000002		ENDSEG		
5251	016324			10001\$: TRAP	C\$ESEG	
(3)	016324			BGNSEG		
(3)	016324	104405		TRAP	C\$BSEG	
5252	016326			JSR	PC,SETBRO	;SET THE BRO BIT'
(3)	016326	104404		SROMCLK		;NEXT WORD IS INSTRUCTION,
5253	016330	004737	003220	JSR	R5,..SROMCLK	
5254	016334			100406		;START AT ROM PC=6
(1)	016334	004537	003100	SROMCLK		;NEXT WORD IS INSTRUCTION,
5255	016340	100406		JSR	R5,..SROMCLK	
5256	016342					
(1)	016342	004537	003100			

5257	016346	106125		104125!<400*4>	:JUMP TO ROM PC OF 525
5258	016350	004737	003330	JSR PC,RAMDAT	:R4=CRAM PC (LSB 8 BITS)
5259	016354	000125		125	:EXPECTED DATA
5260	016356	120504		CMPB R5,R4	:IS ROM PC CORRECT?
5261	016360	001406		BEQ 6\$	:BR IF YES
5262	016362			ERROR 5	:ERROR, CRAM PC IS WRONG
(5)	016366	104455		TRAP C\$ERDF	
(6)	016370	000005		.WORD 5	
(6)	016372	005055		.WORD EMO	
(6)	016374	006556		.WORD ERR5	
5263	016376			6\$: ESCAPE SEG	
(3)	016376	104410		TRAP C\$ESCAPE	
(3)	016400	000002		.WORD 10002\$-	
5264	016402			ENDSEG	
(3)	016402			10002\$: TRAP C\$ESEG	
(3)	016402	104405			
5265	016404			ENDTST	
(3)	016404			L10061: TRAP C\$ETST	
(3)	016404	104401			
5266					
5267	016406			BADHEAD	
(2)				:***** TEST 16 *****	
5268				:*CRAM TEST OF JUMP(I) ON BR1 SET MICRO-PROCESSOR INSTRUCTION.	
5269				:*SET THE BR1 BIT, PERFORM THE JUMP INSTRUCTION.	
5270				:*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION	
5271				:*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE	
5272				:*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT	
5273				:*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT	
5274				:*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL	
5275				:*THEN PORT4 WILL CONTAIN A 37	
5276	016406			BADHEAD	
(2)				:***** TEST 16 *****	
5277					
5278	016406			BGNTST	
(3)	016406			T16::	
5279	016406			MACEX2	:DON'T DO IF M8200.
(1)				:DO NOT DO TEST IF M8200	
(4)	016416	104432		TRAP C\$EXIT	
(4)	016420	000230		.WORD L10062-	
5280	016422			MYINT	
(1)	016422	013701	002516	MOV KMCSR,R1	:RECORD DEVICE ADDR.
5281					:R1 CON'AINS BASE M8200,4,7 ADDRESS
5282	016426			MSTCLR	:MASTER CLEAR M8200,4,7
5283	016432	004737	003474	JSR PC,MEMSET	:SET MEM AND RAM
5284	016436				
5285	016436			1\$: BGNSEG	
(3)	016436	104404		TRAP C\$BSEG	
5286	016440	004737	003230	JSR PC,SETBR1	:SET THE BR1 BIT'
5287	016444			SROMCLK	:NEXT WORD IS INSTRUCTION.
(1)	016444	004537	003100	JSR R5,.SROMCLK	
5288	016450	100400		100400	:START AT ROM PC=0
5289	016452			SROMCLK	:NEXT WORD IS INSTRUCTION.
(1)	016452	004537	003100	JSR R5,.SROMCLK	
5290	016456	116777		114377!<400*5>	:JUMP TO ROM PC OF 1777
5291	016460	004737	003330	JSR PC,RAMDAT	:R4=CRAM PC (LSB 8 BITS)
5292	016464	000377		377	:EXPECTED DATA

5293	016466	120504		CMPB	R5,R4	:IS ROM PC CORRECT?
5294	016470	001406		BEQ	2\$	:BR IF YES
5295	016472			ERROR	5	:ERROR, CRAM PC IS WRONG
(5)	016476	104455		TRAP	C\$ERDF	
(6)	016500	000005		.WORD	5	
(6)	016502	005055		.WORD	EMO	
(6)	016504	006556		.WORD	ERR5	
5296	016506			2\$: ESCAPE	SEG	
(3)	016506	104410		TRAP	C\$ESCAPE	
(3)	016510	000002		.WORD	10000\$-	
5297	016512			ENDSEG		
(3)	016512			10000\$: TRAP	C\$ESEG	
(3)	016512	104405		BGNSEG		
5298	016514			TRAP	C\$BSEG	
(3)	016514	104404		JSR	PC,SETBR1	:SET THE BR1 BIT'
5299	016516	004737	003230	SROMCLK		:NEXT WORD IS INSTRUCTION,
5300	016522			JSR	R5,.SROMCLK	
(1)	016522	004537	003100	100403		:START AT ROM PC=3
5301	016526	100403		SROMCLK		:NEXT WORD IS INSTRUCTION,
(.)	016530	004537	003100	JSR	R5,.SROMCLK	
5303	016534	102400		100000!<400*5>		:JUMP TO ROM PC OF 0
5304	016536	004737	003330	JSR	PC,RAMDAT	:R4=CRAM PC (LSB 8 BITS)
5305	016542	000000		J		:EXPECTED DATA
5306	016544	120504		CMPB	R5,R4	:IS ROM PC CORRECT?
5307	016546	001406		BEQ	4\$	:BR IF YES
5308	016550			ERROR	5	:ERROR, CRAM PC IS WRONG
(5)	016554	104455		TRAP	C\$ERDF	
(6)	016556	000005		.WORD	5	
(6)	016560	005055		.WORD	EMO	
(6)	016562	006556		.WORD	ERR5	
5309	016564			4\$: ESCAPE	SEG	
(3)	016564	104410		TRAP	C\$ESCAPE	
(3)	016566	000002		.WORD	10001\$-	
5310	016570			ENDSEG		
(3)	016570			10001\$: TRAP	C\$ESEG	
(3)	016570	104405		BGNSEG		
5311	016572			TRAP	C\$BSEG	
(3)	016572	104404		JSR	PC,SETBR1	:SET THE BR1 BIT'
5312	016574	004737	003230	SROMCLK		:NEXT WORD IS INSTRUCTION,
5313	016600			JSR	R5,.SROMCLK	
(1)	016600	004537	003100	100406		:START AT ROM PC=6
5314	016604	100406		SROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	016606	004537	003100	JSR	R5,.SROMCLK	
5316	016612	106525		104125!<400*5>		:JUMP TO ROM PC OF 525
5317	016614	004737	003330	JSR	PC,RAMDAT	:R4=CRAM PC (LSB 8 BITS)
5318	016620	000125		125		:EXPECTED DATA
5319	016622	120504		CMPB	R5,R4	:IS ROM PC CORRECT?
5320	016624	001406		BEQ	6\$	:BR IF YES
5321	016626			ERROR	5	:ERROR, CRAM PC IS WRONG
(5)	016632	104455		TRAP	C\$ERDF	
(6)	016634	000005		.WORD	5	
(6)	016636	005055		.WORD	EMO	
(6)	016640	006556		.WORD	ERR5	
5322	016642			6\$: ESCAPE	SEG	

CZDMQCO M8207 STATIC DIAG #2
CZDMQCO.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 PAGE 23-18
HARDWARE TESTS

SEQ 0109

(3) 016642 104410
(3) 016644 000002
5323 016646
(3) 016646
(3) 016646 104405
5324 016650
(3) 016650
(3) 016650 104401
5325
5326 016652
(2)
5327
5328
5329
5330
5331
5332
5333
5334
5335 016652
(2)
5336
5337 016652
(3) 016652
5338 016652
(1)
(4) 016662 104432
(4) 016664 000230
5339 016666
(1) 016666 013701 002516
5340 016672
5341 016676 004737 003474
5342 016702
5343 016702
(3) 016702 104404

TRAP C\$ESCAPE
.WORD 100028-
ENDSEG
100028:
TRAP C\$ESEG
ENDTST
L100062:
TRAP C\$ETST
BADHEAD
:***** TEST 17 *****
:*CRAM TEST OF JUMP(I) ON BR4 SET MICRO-PROCESSOR INSTRUCTION.
:*SET THE BR4 BIT, PERFORM THE JUMP INSTRUCTION.
:*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
:*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
:*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
:*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
:*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
:*THEN PORT4 WILL CONTAIN A 37
BADHEAD
:***** TEST 17 *****
BGN TST
117::
MACEX2
:DO NOT DO TEST IF M8200 :DON'T DO IF M8200.
TRAP C\$EXIT
.WORD L100063-
MYINT
MOV KMCSR,R1 :RECORD DEVICE ADDR.
MSTCLR :MASTER CLEAR M8200,4,7
JSR PC, MEMSET :SET MEM AND RAM
18:
BGNSEG
TRAP C\$BSEG

200000 MB207 STATIC DIAG #2
200000.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 PAGE 24
HARDWARE TESTS

SEQ 0110

5345 016704 004737 003240
5346 016710
() 016710 004537 003100
5347 016714 100400
5348 016716
() 016716 004537 003100

JSR PC,SETHR4
SRMCLK
JSR R5,.SRMCLK
100400
SRMCLK
JSR R5,.SRMCLK

:SET THE BR4 BIT'
:NEXT WORD IS INSTRUCTION.
:START AT ROM PC=0
:NEXT WORD IS INSTRUCTION.

CZDPOCO M8207 STATIC DIAG #2 MACV11 30A(1052) 21-JUL-81 14:48 N 9
CZDPOC.P11 21-JUL-81 14:36 HARDWARE TESTS PAGE 25

SEQ 0111

5350 016722 117377
5351 016724 004737 003330

114377:<400*6>
JSR PC,RANDAT

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

CZDMOC0 MB207 STATIC DIAG #2
CZDMOC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48
HARDWARE TESTS

1 9
PAGE 26

SEQ 0112

5353 016730 000377
5354 016732 120504

377
(MPB) R5,R4

:EXPECTED DATA
:IS ROM PC CORRECT?

CZDMQCO M8207 STATIC DIAG #2 MACY11 30A(1052) 21-JUL-81 14:48 J 9
CZDMQC.P11 21-JUL-81 14:36 HARDWARE TESTS PAGE 27

SEQ 0113

5356 016734 001406

BEQ 28

;BR IF YES

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48
HARDWARE TESTS

K 9
PAGE 28

SEQ 0114

5358 016736
(5) 016742 104455
(6) 016744 000005
(6) 016746 005055
(6) 016750 006556

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

;ERROR, CRAM PC IS WRONG

5360	016752		2\$:	ESCAPE SEG	
(3)	016752	104410		TRAP C\$ESCAPE	
(3)	016754	000002		.WORD 10000\$-	
5361	016756			ENDSEG	
(3)	016756		10000\$:		
(3)	016756	104405		TRAP C\$ESEG	
5362	016760			BGNSEG	
(3)	016760	104404		TRAP C\$BSEG	
5363	016762	004737 003240		JSR PC,SETBR4	:SET THE BR4 BIT'
5364	016766			SROMCLK	:NEXT WORD IS INSTRUCTION,
(1)	016766	004537 003100		JSR R5,.SROMCLK	
5365	016772	100403		100403	:START AT ROM PC=3
5366	016774			SROMCLK	:NEXT WORD IS INSTRUCTION,
(1)	016774	004537 003100		JSR R5,.SROMCLK	
5367	017000	103000		100000!<400*6>	:JUMP TO ROM PC OF 0
5368	017002	004737 003330		JSR PC,RAMDAT	:R4=CRAM PC (LSB 8 BITS)
5369	017006	000000		0	:EXPECTED DATA
5370	017010	120504		CMPB R5,R4	:IS ROM PC CORRECT?
5371	017012	001406		BEQ 4\$	:BR IF YES
5372	017014			ERROR 5	:ERROR, CRAM PC IS WRONG
(5)	017020	104455		TRAP C\$ERDF	
(6)	017022	000005		.WORD 5	
(6)	017024	005055		.WORD EM0	
(6)	017026	006556		.WORD ERR5	
5373	017030		4\$:	ESCAPE SEG	
(3)	017030	104410		TRAP C\$ESCAPE	
(3)	017032	000002		.WORD 10001\$-	
5374	017034			ENDSEG	
(3)	017034		10001\$:		
(3)	017034	104405		TRAP C\$ESEG	
5375	017036			BGNSEG	
(3)	017036	104404		TRAP C\$BSEG	
5376	017040	004737 003240		JSR PC,SETBR4	:SET THE BR4 BIT'
5377	017044			SROMCLK	:NEXT WORD IS INSTRUCTION,
(1)	017044	004537 003100		JSR R5,.SROMCLK	
5378	017050	100406		100406	:START AT ROM PC=6
5379	017052			SROMCLK	:NEXT WORD IS INSTRUCTION,
(1)	017052	004537 003100		JSR R5,.SROMCLK	
5380	017056	107125		104125!<400*6>	:JUMP TO ROM PC OF 525
5381	017060	004737 003330		JSR PC,RAMDAT	:R4=CRAM PC (LSB 8 BITS)
5382	017064	000125		125	:EXPECTED DATA
5383	017066	120504		CMPB R5,R4	:IS ROM PC CORRECT?
5384	017070	001406		BEQ 6\$	:BR IF YES
5385	017072			ERROR 5	:ERROR, CRAM PC IS WRONG
(5)	017076	104455		TRAP C\$ERDF	
(6)	017100	000005		.WORD 5	
(6)	017102	005055		.WORD EM0	
(6)	017104	006556		.WORD ERR5	
5386	017106		6\$:	ESCAPE SEG	
(3)	017106	104410		TRAP C\$ESCAPE	
(3)	017110	000002		.WORD 10002\$-	
5387	017112			ENDSEG	
(3)	017112		10002\$:		
(3)	017112	104405		TRAP C\$ESEG	
5388	017114		ENDTST		
(3)	017114		L10063:		

```

(3) 017114 104401 TRAP C$ETST
5389
5390 017116 BADHEAD
(2) ;***** TEST 18 *****
5391 ;*CRAM TEST OF JUMP(I) ON BR7 SET MICRO-PROCESSOR INSTRUCTION.
5392 ;*SET THE BR7 BIT, PERFORM THE JUMP INSTRUCTION.
5393 ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
5394 ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
5395 ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
5396 ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
5397 ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
5398 ;*THEN PORT4 WILL CONTAIN A 37
5399 017116 BADHEAD
(2) ;***** TEST 18 *****
5400
5401 017116 BGNTST
(3) 017116 T18::
5402 017116 MACEX2 ;DON'T DO IF M8200.
(1) ;DO NOT DO TEST IF M8200
(4) 017126 104432 TRAP C$EXIT
(4) 017130 000230 .WORD L10064-.
5403 017132 MYINT
(1) 017132 013701 002516 MOV KMCSR,R1 ;RECORD DEVICE ADDR.
5404 ;R1 CONTAINS BASE M8200,4,7 ADDRESS
5405 017136 MSTCLR ;MASTER CLEAR M8200,4,7
5406 017142 004737 003474 JSR PC,MEMSET ;SET MEM AND RAM
5407 017146 1$: BGNSEG
(3) 017146 104404 TRAP C$BSEG
5408 017150 004737 003250 JSR PC,SETR7 ;SET THE BR7 BIT'
5409 017154 SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 017154 004537 003100 JSR R5,.SROMCLK
5410 017160 100400 ;START AT ROM PC=0
5411 017162 SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 017162 004537 003100 JSR R5,.SROMCLK
5412 017166 117777 114377!<400*7> ;JUMP TO ROM PC OF 1777
5413 017170 004737 003330 JSR PC,RAMDAT ;R4=CRAM PC (LSB 8 BITS)
5414 017174 000377 377 ;EXPECTED DATA
5415 017176 120504 CMPB R5,R4 ;IS ROM PC CORRECT?
5416 017200 001406 BEQ 2$ ;BR IF YES
5417 017202 ERROR 5 ;ERROR, CRAM PC IS WRONG
(5) 017206 104455 TRAP C$ERDF
(6) 017210 000005 .WORD 5
(6) 017212 005055 .WORD EMO
(6) 017214 006556 .WORD ERR5
5418 017216 2$: ESCAPE SEG
(3) 017216 104410 TRAP C$ESCAPE
(3) 017220 000002 .WORD 10000$-.
5419 017222 ENDSEG
(3) 017222 10000$:
(3) 017222 104405 TRAP C$ESEG
5420 017224 BGNSEG
(3) 017224 104404 TRAP C$BSEG
5421 017226 004737 003250 JSR PC,SETR7 ;SET THE BR7 BIT'
5422 017232 SROMCLK ;NEXT WORD IS INSTRUCTION,
(1) 017232 004537 003100 JSR R5,.SROMCLK
5423 017236 100403 ;START AT ROM PC=3

```

5424	017240			SROMCLK		;NEXT WORD IS INSTRUCTION,
(1)	017240	004537	003100	JSR	R5, SROMCLK	
5425	017244	103400		100000!	<400*7>	;JUMP TO ROM PC OF 0
5426	017246	004737	003330	JSR	PC, RAMDAT	;R4=CRAM PC (LSB 8 BITS)
5427	017252	000000		0		;EXPECTED DATA
5428	017254	120504		CMPB	R5, R4	;IS ROM PC CORRECT?
5429	017256	001406		BEQ	4\$	;BR IF YES
5430	017260			ERROR	5	;ERROR, CRAM PC IS WRONG
(5)	017264	104455		TRAP	C\$ERDF	
(6)	017266	000005		.WORD	5	
(6)	017270	005055		.WORD	EMO	
(6)	017272	006556		.WORD	ERR5	
5431	017274			4\$: ESCAPE	SEG	
(3)	017274	104410		TRAP	C\$ESCAPE	
(3)	017276	000002		.WORD	10001\$-	
5432	017300			ENDSEG		
(3)	017300			10001\$: TRAP	C\$ESEG	
(3)	017300	104405		BGNSEG		
5433	017302			TRAP	C\$BSEG	
(3)	017302	104404		JSR	PC, SETBR7	;SET THE BR7 BIT'
5434	017304	004737	003250	SROMCLK		;NEXT WORD IS INSTRUCTION,
5435	017310			JSR	R5, SROMCLK	
(1)	017310	004537	003100	100406		;START AT ROM PC=6
5436	017314	100406		SROMCLK		;NEXT WORD IS INSTRUCTION,
5437	017316			JSR	R5, SROMCLK	
(1)	017316	004537	003100	104125!	<400*7>	;JUMP TO ROM PC OF 525
5438	017322	107525		JSR	PC, RAMDAT	;R4=CRAM PC (LSB 8 BITS)
5439	017324	004737	003330	125		;EXPECTED DATA
5440	017330	000125		CMPB	R5, R4	;IS ROM PC CORRECT?
5441	017332	120504		BEQ	6\$	;BR IF YES
5442	017334	001406		ERROR	5	;ERROR, CRAM PC IS WRONG
5443	017336			TRAP	C\$ERDF	
(5)	017342	104455		.WORD	5	
(6)	017344	000005		.WORD	EMO	
(6)	017346	005055		.WORD	ERR5	
(6)	017350	006556		6\$: ESCAPE	SEG	
5444	017352			TRAP	C\$ESCAPE	
(3)	017352	104410		.WORD	10002\$-	
(3)	017354	000002		ENDSEG		
5445	017356			10002\$: TRAP	C\$ESEG	
(3)	017356					
(3)	017356	104405		ENDTST		
5446	017360			L10064: TRAP	C\$ETST	
(3)	017360	104401				
5447				BADHEAD		
5448	017362			;***** TEST 19 *****		
(2)				;*CRAM TEST OF JUMP(1) ON C BIT CLEAR MICRO-PROCESSOR INSTRUCTION.		
5449				;*SET THE C BIT, PERFORM THE JUMP INSTRUCTION.		
5450				;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION		
5451				;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE		
5452				;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT		
5453				;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT		
5454				;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL		
5455				;*THEN PORT4 WILL CONTAIN A 37		
5456						

5457 017362
(2)
5458
5459 017362
(3) 017362
5460 017362
(1)
(4) 017372 104432
(4) 017374 000244
5461 017376 013701 002576
5462 017402
5463 017406 004737 003474
5464 017412
(3) 017412 104404
5465 017414 004737 003260
5466 017420 004737 003166
5467 017424
(1) 017424 004537 003100
5468 017430 100400
5469 017432
(1) 017432 004537 003100
5470 017436 115377
5471 017440 004737 003330
5472 017444 000001
5473 017446 120504
5474 017450 001406
5475 017452
(5) 017456 104455
(6) 017460 000005
(6) 017462 005055
(6) 017464 006556
5476 017466
(3) 017466 104410
(3) 017470 000002
5477 017472
(3) 017472
(3) 017472 104405
5478 017474
(3) 017474 104404
5479 017476
(1)
5480 017506 004737 003166
5481 017512
(1) 017512 004537 003100
5482 017516 100403
5483 017520
(1) 017520 004537 003100
5484 017524 101000
5485 017526 004737 003330
5486 017532 000004
5487 017534 120504
5488 017536 001406
5489 017540
(5) 017544 104455
(6) 017546 000005

BADHEAD
:***** TEST 19 *****

BGNTEST
:19::
MACEX2 :DON'T DO IF MB200.
:DO NOT DO TEST IF MB200
TRAP C\$EXIT
.WORD L10065-.
MYINT
MOV KMCSR,R1 :RECORD DEVICE ADDR.
MSTCLR :MASTER CLEAR MB200,4,7
JSR PC,MEMSET :SET MEM AND RAM
:S:
BGNSEG
TRAP C\$BSEG
JSR PC,SETC
JSR PC,CLRALL
SROMCLK :NEXT WORD IS INSTRUCTION,
JSR R5,.SROMCLK
100400 :START AT ROM PC=0
SROMCLK :NEXT WORD IS INSTRUCTION,
JSR R5,.SROMCLK
114377.<400*2> :JUMP TO ROM PC OF 1777
JSR PC,RAMDAT :R4=CRAM PC (LSB 8 BITS)
1 :EXPECTED DATA
CMPB R5,R4 :IS ROM PC CORRECT?
BEQ 28 :BR IF YES
ERROR 5 :ERROR, CRAM PC IS WRONG
TRAP C\$ERDF
.WORD 5
.WORD EMO
.WORD ERR5
:S:
ESCAPE SEG
TRAP C\$ESCAPE
.WORD 100008-.
ENDSEG
:00008:
TRAP C\$ESEG
BGNSEG
TRAP C\$BSEG
SKIP06 68
:GOTO 68 IF MB206
JSR PC,CLRALL :CLEAR ALL CONDITIONS
SROMCLK :NEXT WORD OF INSTRUCTION
JSR R5,.SROMCLK
100403 :START AT ROM PC=3
SROMCLK :NEXT WORD OF INSTRUCTION
JSR R5,.SROMCLK
100000!<400*2> :JUMP TO ROM PC OF 0
JSR PC,RAMDAT :R4=CRAM PC (LSB 8 BITS)
4 :EXPECTED DATA
CMPB R5,R4 :IS ROM PC CORRECT?
BEQ 48 :BR IF YES
ERROR 5 :ERROR, CRAM PC IS WRONG
TRAP C\$ERDF
.WORD 5

(6) 017550 005055
(6) 017552 006556
5490 017554 104410
(3) 017554 104410
(3) 017556 000002
5491 017560
(3) 017560
(3) 017560 104405
5492 017562
(3) 017562 104404
5493 017564 004737 003166
5494 017570
(1) 017570 004537 003100
5495 017574 100406
5496 017576
(1) 017576 004537 003100
5497 017602 105125
5498 017604 004737 003330
5499 017610 000007
5500 017612 120504
5501 017614 001406
5502 017616
(5) 017622 104455
(6) 017624 000005
(6) 017626 005055
(6) 017630 006556
5503 017632
(3) 017632 104405
(3) 017634 000002
5504 017636
(3) 017636
(3) 017636 104405
5505 017640
(3) 017640
(3) 017640 104405
5506
5507 017642
(2)
5508
5509
5510
5511
5512
5513
5514
5515
5516 017642
(2)
5517
5518 017642
(3) 017642
5519 017642
(1)
(4) 017652 104432
(4) 017654 000244
5520 017656

```

48:      .WORD    EMO
      .WORD    ERR5
      ESCAPE SEG
      TRAP      C$ESCAPE
      .WORD      10001$-.
      ENDSEG

10001$:  TRAP      C$ESEG
      BGNSEG
      TRAP      C$BSEG
      JSR        PC,CLRALL          ;CLEAR ALL CONDITIONS
                                          ;NEXT WORD IS INSTRUCTION,
      SROMCLK
      JSR        R5,.,SROMCLK
      100406
      SROMCLK
      JSR        R5,.,SROMCLK
      104 25: <400*2>              ;JUMP TO ROM PC OF 525
      JSR        PC,RAMDAT          ;R4=CRAM PC (LSB 8 BITS)
      7
      C$RFB      R5,R4              ;EXPECTED DATA
      BFO        6$                 ;IS ROM PC CORRECT?
      ERROR      5                   ;BR IF YES
      TRAP      C$ERDF              ;ERROR, CRAM PC IS WRONG
      .WORD      5
      .WORD      EMO
      .WORD      ERR5
48:      ESCAPE SEG
      TRAP      C$ESCAPE
      .WORD      10002$-.
      ENDSEG

10002$:  TRAP      C$ESEG

ENDIST
10065:  TRAP      C$ETST

BADHEAD
;***** TEST 20 *****
;*(CRAM TEST OF JUMP(1) ON Z BIT CLEAR MICRO-PROCESSOR INSTRUCTION.
;*(CLEAR THE Z BIT, PERFORM THE JUMP INSTRUCTION.
;*(VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
;*(IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
;*(BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
;*(THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
;*(THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
;*(THEN PORT4 WILL CONTAIN A 37
BADHEAD
;***** TEST 20 *****

BGNIST
T20::
MACEX2
;DO NOT DO TEST IF M8200
;DON'T DO IF M8200.
      TRAP      C$EXIT
      .WORD      L10066-.
      MYINT

```

(1)	017656	013701	002516	MOV	KMCSR,R1	:RECORD DEVICE ADDR.
5521	017662			MSTCLR		:MASTER CLEAR M8200.4,7
5522	017666	004737	003474	JSR	PC, MEMSET	:SET MEM AND RAM
5523	017672			18:	BGNSEG	
(3)	017672	104404		TRAP	C\$BSEG	
5524	017674	004737	003312	JSR	PC, SETZ	
5525	017700	004737	003166	JSR	PC, CLRALL	: CLEAR CONDITION CODES :*** 80
5526	017704			SROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	017704	004537	003100	JSR	R5, .SROMCLK	
5527	017710	100400		100400		:START AT ROM PC=0
5528	017712			SROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	017712	004537	003100	JSR	R5, .SROMCLK	
5529	017716	115777		114377. <400+3>		:JUMP TO ROM PC OF 1777
5530	017720	004737	003330	JSR	PC, RAMDAT	:R4=CRAM PC (LSB 8 BITS)
5531	017724	000001		1		:EXPECTED DATA
5532	017726	120504		CMPS	R5,R4	:IS ROM PC CORRECT?
5533	017730	001406		BEO	28	:BR IF YES
5534	017732			ERROR	5	:ERROR, CRAM PC IS WRONG
(5)	017736	104455		TRAP	C\$ERDF	
(6)	017740	000005		.WORD	5	
(6)	017742	005055		.WORD	EMO	
(6)	017744	006556		.WORD	ERR5	
5535	017746			28:	ESCAPE SEG	
(3)	017746	104410		TRAP	C\$ESCAPE	
(3)	017750	000002		.WORD	100008-	
5536	017752			ENDSEG		
(3)	017752			100008:		
(3)	017752	104405		TRAP	C\$ESEG	
5537	017754			BGNSEG		
(3)	017754	104404		TRAP	C\$BSEG	
5538	017756			SKIPO6	68	
(1)				:GOTO 68 IF M8206		
5539	017766	004737	003166	JSR	PC, CLRALL	:CLEAR ALL CONDITIONS
5540	017772			SROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	017772	004537	003100	JSR	R5, .SROMCLK	
5541	017776	100403		100403		:START AT ROM PC=3
5542	020000			SROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	020000	004537	003100	JSR	R5, .SROMCLK	
5543	020004	101400		100000! <400+3>		:JUMP TO ROM PC OF 0
5544	020006	004737	003330	JSR	PC, RAMDAT	:R4=CRAM PC (LSB 8 BITS)
5545	020012	000004		4		:EXPECTED DATA
5546	020014	120504		CMPS	R5,R4	:IS ROM PC CORRECT?
5547	020016	001406		BEO	48	:BR IF YES
5548	020020			ERROR	5	:ERROR, CRAM PC IS WRONG
(5)	020024	104455		TRAP	C\$ERDF	
(6)	020026	000005		.WORD	5	
(6)	020030	005055		.WORD	EMO	
(6)	020032	006556		.WORD	ERR5	
5549	020034			48:	ESCAPE SEG	
(3)	020034	104410		TRAP	C\$ESCAPE	
(3)	020036	000002		.WORD	100018-	
5550	020040			ENDSEG		
(3)	020040			100018:		
(3)	020040	104405		TRAP	C\$ESEG	
5551	020042			BGNSEG		
(3)	020042	104404		TRAP	C\$BSEG	

CZDMOCC M8207 STATIC DIAG #2
CZDMOCC.P11 21-JUL-81 14:36

MACV11 30A10521 21-JUL-81 14:48 PAGE 29-6
HARDWARE TESTS

SEQ 0121

```

5552 020044 004737 003166 JSR PC,CLRALL ;CLEAR ALL CONDITIONS
5553 020050 SRMCLK ;NEXT WORD IS INSTRUCTION,
(1) 020050 004537 003100 JSR R5,.SRMCLK
5554 020054 100406 ;START AT ROM PC=6
5555 020056 SRMCLK ;NEXT WORD IS INSTRUCTION,
(1) 020056 004537 003100 JSR R5,.SRMCLK
5556 020062 105525 ;JUMP TO ROM PC OF 525
5557 020064 004737 003330 JSR PC,RAMDAT ;R4=CRAM PC (LSB 8 BITS)
5558 020070 000007 ;EXPECTED DATA
5559 020072 120504 ;IS ROM PC CORRECT?
5560 020074 001406 BEQ 68 ;BR IF YES
5561 020076 ERROR 5 ;ERROR, CRAM PC IS WRONG
(5) 020102 104455 TRAP C$ERDF
(6) 020104 000005 .WORD 5
(6) 020106 005055 .WORD EMO
(6) 020110 006556 .WORD ERR5
5562 020112 18: ESCAPE SEG
(3) 020112 104400 TRAP C$ESCAPE
(3) 020114 000002 .WORD 100028-.
5563 020116 ENDSEG
(3) 020116 100028: TRAP C$ESEG
(3) 020116 104405 END*ST
(3) 020120 100066: TRAP C$E*ST
5565 020122 BADHEAD
5566 (2) ;***** TEST 21 *****
5567 ;*CRAM TEST OF JUMP(I) ON BRO (CLEAR MICRO-PROCESSOR INSTRUCTION.
5568 ;*CLEAR THE BRO BIT, PERFORM THE JUMP INSTRUCTION.
5569 ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
5570 ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
5571 ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
5572 ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
5573 ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
5574 ;*THEN PORT4 WILL CONTAIN A 37
5575 020122 BADHEAD
5576 (2) ;***** TEST 21 *****
5577 020122 BGN*ST
5578 (3) 020122 121::
5579 (1) 020132 104432 MACEX2 ;DON'T DO IF M8200.
5580 (4) 020134 000240 ;DO NOT DO TEST IF M8200
5581 (4) 020136 013701 002516 TRAP C$EXIT
5582 (3) 020152 104404 .WORD L10067-.
5583 020154 004737 003166 MYINT
5584 (1) 020160 004537 003100 MOV KMCSR,R1 ;RECORD DEVICE ADDR.
5585 020164 100400 MSTCLR ;MASTER CLEAR M8200.4,7
5586 020166 SRMCLK ;SET MEM AND RAM
18: BGNSEG
TRAP C$BSEG
JSR PC,CLRALL ;CLEAR ALL CONDITIONS
SRMCLK ;NEXT WORD IS INSTRUCTION,
JSR R5,.SRMCLK
100400 ;START AT ROM PC=0
SRMCLK ;NEXT WORD IS INSTRUCTION,

```

1) 020166 004537 003100
5587 020172 116377
5588 020174 004737 003330
5589 020200 000001
5590 020202 120504
5591 020204 001406
5592 020206
(5) 020212 104455
(6) 020214 000005
(6) 020216 005055
(6) 020220 006556
5593 020222
(3) 020222 104410
(3) 020224 000002
5594 020226
(3) 020226
(3) 020226 104405
5595 020230
(3) 020230 104404
5596 020232
(1) 020242 004737 003100
5597 020246
(1) 020246 004537 003100
5599 020252 100403
5600 020254
(1) 020254 004537 003100
5601 020260 102000
5602 020262 004737 003330
5603 020266 000004
5604 020270 120504
5605 020272 001406
5606 020274
(5) 020300 104455
(6) 020302 000005
(6) 020304 005055
(6) 020306 006556
5607 020310
(3) 020310 104410
(3) 020312 000002
5608 020314
(3) 020314
(3) 020314 104405
5609 020316
(3) 020316 104404
5610 020320 004737 003100
5611 020324
(1) 020324 004537 003100
5612 020330 100406
5613 020332
(1) 020332 004537 003100
5614 020336 106125
5615 020340 004737 003330
5616 020344 000007
5617 020346 120504
5618 020350 001406

JSR R5,SR0MCLK
114377.<400*4>
JSR PC,RAMDAT
1
CMPB R5,R4
BEQ 28
ERROR 5
TRAP (SERDF
.WORD 5
.WORD EMO
.WORD ERR5
ESCAPE SEG
TRAP (SESCAPE
.WORD 100008-
ENDSEG
100008: TRAP (SESEG
BGNSEG
TRAP (SBSEG
SKIP06 68
;GOTO 68 IF M8206
JSR PC,CLRALL
SR0MCLK
JSR R5,SR0MCLK
100403
SR0MCLK
JSR R5,SR0MCLK
100000.<400*4>
JSR PC,RAMDAT
4
CMPB R5,R4
BEQ 48
ERROR 5
TRAP (SERDF
.WORD 5
.WORD EMO
.WORD ERR5
ESCAPE SEG
TRAP (SESCAPE
.WORD 100018-
ENDSEG
48: 100018: TRAP (SESEG
BGNSEG
TRAP (SBSEG
JSR PC,CLRALL
SR0MCLK
JSR R5,SR0MCLK
100406
SR0MCLK
JSR R5,SR0MCLK
104125.<400*4>
JSR PC,RAMDAT
7
CMPB R5,R4
BEQ 68

;JUMP TO ROM PC OF 1777
;R4=CRAM PC (LSB 8 BITS)
;EXPECTED DATA
;IS ROM PC CORRECT?
;BR IF YES
;ERROR, CRAM PC IS WRONG
;CLEAR ALL CONDITIONS
;NEXT WORD IS INSTRUCTION.
;START AT ROM PC=3
;NEXT WORD IS INSTRUCTION.
;JUMP TO ROM PC OF 0
;R4=CRAM PC (LSB 8 BITS)
;EXPECTED DATA
;IS ROM PC CORRECT?
;BR IF YES
;ERROR, CRAM PC IS WRONG
;CLEAR ALL CONDITIONS
;NEXT WORD IS INSTRUCTION.
;START AT ROM PC=6
;NEXT WORD IS INSTRUCTION.
;JUMP TO ROM PC OF 525
;R4=CRAM PC (LSB 8 BITS)
;EXPECTED DATA
;IS ROM PC CORRECT?
;BR IF YES

20M000 MB207 STATIC DIAG #2
20M000.P11 21-JUL-81 14:36

20M000 3041052) 21-JUL-81 14:48 PAGE 29-8
HARDWARE TESTS

SEQ 0123

```

5619 020352          ERROR 5          ;ERROR, CRAM PC IS WRONG
      (5) 020356 104455 TRAP (SERDF
      (6) 020360 000005 .WORD 5
      (6) 020362 005055 .WORD EMO
      (6) 020364 006556 .WORD ERR5
5620 020366          ESCAPE SEG
      (3) 020366 104410 TRAP (SESCAPE
      (3) 020370 000002 .WORD 100028-.
5621 020372          ENDSEG
      (3) 020372 100028: TRAP (SESEG
      (3) 020372 104415
5622 020374          ENDTEST
      (3) 020374 100028: TRAP (SETST
      (3) 020374 104415
5623
5624 020376          BADHEAD
      (2)
5625          ;***** TEST 22 *****
5626          ;*CRAM TEST OF JUMP(1) ON BR1 CLEAR MICRO-PROCESSOR INSTRUCTION.
5627          ;*CLEAR THE BR1 BIT, PERFORM THE JUMP INSTRUCTION.
5628          ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
5629          ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
5630          ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
5631          ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
5632          ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
5633          ;*THEN PORT4 WILL CONTAIN A 37
5634 020376          BADHEAD
      (2)
5635          ;***** TEST 22 *****
5636          ;*****
5637 020376          MACEX2          ;DON'T DO IF MB200.
      (3) 020376          ;DO NOT DO TEST IF MB200
5638 020376          TRAP (SEXIT
      (4) 020406 104432 .WORD L10070-.
5639 020410 000240 MYINT
5640 020412 013701 002516 MOV KMCSR,R1          ;RECORD DEVICE ADDR.
5641 020416 004737 003474 MTLCLR          ;MASTER CLEAR MB200,4,7
5642 020422 004737 003474 JSR PC,MEMSET          ;SET MEM AND RAM
5643 020426 104406          BGNSEG
      (3) 020426 104406 TRAP (SBSEG
5644 020430 004737 003166 JSR PC,CLRALL          ;CLEAR ALL CONDITIONS
5645 020434 004537 003100 SROMCLK          ;NEXT WORD IS INSTRUCTION.
      (1) 020434 004537 003100 JSR R5,.SROMCLK
5646 020440 100400          ;START AT ROM PC=0
5647 020442 004537 003100 SROMCLK          ;NEXT WORD IS INSTRUCTION.
      (1) 020442 004537 003100 JSR R5,.SROMCLK
5648 020446 116777 003330 114377!<400*5>          ;JUMP TO ROM PC OF 1777
5649 020450 004737 003330 JSR PC,RANDAT          ;R4=CRAM PC (LSB 8 BITS)
5650 020454 000001          ;EXPECTED DATA
5651 020456 120506          ;IS ROM PC CORRECT?
5652 020460 001406          ;BR IF YES
5653 020462          ERROR 5          ;ERROR, CRAM PC IS WRONG
      (5) 020466 104455 TRAP (SERDF
      (6) 020470 000005 .WORD 5
      (6) 020472 005055 .WORD EMO
      (6) 020474 006556 .WORD ERR5

```

5651	020476			28:	ESCAPE SEG	
3)	020476	104410		TRAP	(BESCAPE	
3)	020500	000002		WORD	100008-	
5652	020502			ENDSEG		
3)	020502			100008:		
3)	020502	104405		TRAP	(BSESEG	
5653	020504			BGNSEG		
3)	020504	104404		TRAP	(BSESEG	
5654	020508			SKIP06	68	
3)				:GOTO 68 IF MB206		
5655	020516	004737	003166	JSR	PC,CLRAL	:CLEAR ALL CONDITIONS
5656	020522			SRORCLK		:NEXT WORD IS INSTRUCTION.
3)	020522	004537	003100	JSR	R5,SRORCLK	
5657	020526	004003		100403		:START AT ROM PC=3
5658	020530			SRORCLK		:NEXT WORD IS INSTRUCTION.
3)	020530	004537	003100	JSR	R5,SRORCLK	
5659	020534	102400		100000	<400*5>	:JUMP TO ROM PC OF 0
5660	020536	004737	003330	JSR	PC,RANDAT	:R4=CRAM PC (LSB 8 BITS)
5661	020542	000004		4		:EXPECTED DATA
5662	020544	120504		CMPI	R5,R4	:IS ROM PC CORRECT?
5663	020546	001406		BEQ	48	:BR IF YES
5664	020550			ERROR	5	:ERROR, CRAM PC IS WRONG
3)	020554	104455		TRAP	(BERDF	
3)	020556	000005		WORD	5	
6)	020560	005055		WORD	END	
6)	020562	006556		WORD	ERR5	
5665	020564			ESCAPE SEG		
3)	020564	104410		TRAP	(BESCAPE	
3)	020566	000002		WORD	100018-	
5666	020570			ENDSEG		
3)	020570			100018:		
3)	020570	104405		TRAP	(BSESEG	
5667	020572			BGNSEG		
3)	020572	104404		TRAP	(BSESEG	
5668	020574	004737	003166	JSR	PC,CLRAL	:CLEAR ALL CONDITIONS
5669	020600			SRORCLK		:NEXT WORD IS INSTRUCTION.
3)	020600	004537	003100	JSR	R5,SRORCLK	
5670	020604	100406		100406		:START AT ROM PC=6
5671	020608			SRORCLK		:NEXT WORD IS INSTRUCTION.
3)	020608	004537	003100	JSR	R5,SRORCLK	
5672	020612	106525		104125	<400*5>	:JUMP TO ROM PC OF 525
5673	020614	004737	003330	JSR	PC,RANDAT	:R4=CRAM PC (LSB 8 BITS)
5674	020620	000007		7		:EXPECTED DATA
5675	020622	120504		CMPI	R5,R4	:IS ROM PC CORRECT?
5676	020624	001406		BEQ	68	:BR IF YES
5677	020626			ERROR	5	:ERROR, CRAM PC IS WRONG
3)	020632	104455		TRAP	(BERDF	
6)	020634	000005		WORD	5	
6)	020636	005055		WORD	END	
6)	020640	006556		WORD	ERR5	
5678	020642			ESCAPE SEG		
3)	020642	104410		TRAP	(BESCAPE	
3)	020644	000002		WORD	100028-	
5679	020646			ENDSEG		
3)	020646			100028:		
3)	020646	104405		TRAP	(BSESEG	

7DM0C0 0207 STATIC DIAG #2
7DM0C.P11 21-JUL-81 14:36

MACV11 30A(1052) 21-JUL-81 14:48 PAGE 29-10
HARDWARE TESTS

SEQ 0125

5680 020650
(3) 020650
(3) 020650 10440
5681
5682 020652
(2)
5683
5684
5685
5686
5687
5688
5689 020652 11140
5690
5691
5692 020654
(2)
5693

ENDTST
L'0070:
*RAP CSETST

BADHEAD

***** TEST 23 *****
:*(RAM TEST OF JUMP(1) ON BR4 CLEAR MICRO-PROCESSOR INSTRUCTION.
:*(CLEAR THE BR4 BIT, PERFORM THE JUMP INSTRUCTION.
:*(VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
:*(IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
:*(BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
:*(THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT

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

BADHEAD

***** TEST 23 *****

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACV11 30A(1052) 21-JUL-81 14:48 J 10
HARDWARE TESTS PAGE 30

SEQ 0126

5695 020654
(3) 020654
5696 020654
(1)
(4) 020664 104432
(4) 020666 000240
5697 020670
(1) 020670 013701 002516
5698 020674
5699 020700 004737 003474
5700 020704
(3) 020704 104404
5701 020706 004737 003166
5702 020712
(1) 020712 004537 003100
5703 020716 100400
5704 020720
(1) 020720 004537 003100
5705 020724 117377
5706 020726 004737 003330

BGNTST
T23::

MACEX2
;DO NOT DO TEST IF M8200 ;DON'T DO IF M8200.
TRAP C\$EXIT
.WORD L10071-
MYINT
MOV KMCSR,R1 ;RECORD DEVICE ADDR.
MSTCLR ;MASTER CLEAR M8200,4,7
JSR PC,MEMSET ;SET MEM AND RAM
BGNSEG
TRAP C\$BSEG
JSR PC,CLRALL ;CLEAR ALL CONDITIONS
SROMCLK ;NEXT WORD IS INSTRUCTION,
JSR R5,.SROMCLK
100400 ;START AT ROM PC=0
SROMCLK ;NEXT WORD IS INSTRUCTION,
JSR R5,.SROMCLK
114377.<400*6> ;JUMP TO ROM PC OF 1777
JSR PC,RAMDAT ;R4=CRAM PC (LSB 8 BITS)

5708	020732	000001		1		;EXPECTED DATA
5709	020734	120504		CMPB	R5,R4	:IS ROM PC CORRECT?
5710	020736	001406		BEQ	2\$	:BR IF YES
5711	020740			ERROR	5	:ERROR, CRAM PC IS WRONG
(5)	020744	104455		TRAP	C\$ERDF	
(6)	020746	000005		.WORD	5	
(6)	020750	005055		.WORD	EMO	
(6)	020752	006556		.WORD	ERR5	
5712	020754			ESCAPE	SEG	
(3)	020754	104410		TRAP	C\$ESCAPE	
(3)	020756	000002		.WORD	10000\$-	
5713	020760			ENDSEG		
(3)	020760					
(3)	020760	104405		TRAP	C\$ESEG	
5714	020762			BGNSEG		
(3)	020762	104404		TRAP	C\$BSEG	
5715	020764			SKIP06	6\$	
(1)				:GOTO 6\$	IF M8206	
5716	020774	004737	003166	JSR	PC,CLRALL	:CLEAR ALL CONDITIONS
5717	021000			SROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	021000	004537	003100	JSR	R5,..SROMCLK	
5718	021004	100403		100403		:START AT ROM PC=3
5719	021006			SROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	021006	004537	003100	JSR	R5,..SROMCLK	
5720	021012	103000		100000.<400*6>	:JUMP TO ROM PC OF 0	
5721	021014	004737	003330	JSR	PC,RAMDAT	:R4=CRAm PC (LSB 8 BITS)
5722	021020	000004		4		:EXPECTED DATA

CZDMQCO M8207 STATIC DIAG #2 MACY11 30A(1052) 21-JUL-81 14:48 L 10
CZDMQC.P11 21-JUL-81 14:36 HARDWARE TESTS PAGE 32

SEQ 0128

5724 021022 120504
5725 021024 001406

CMPB R5,R4
BEQ 4\$

;IS ROM PC CORRECT?
;BSR IF YES

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 M 10
HARDWARE TESTS PAGE 33

SEQ 0129

5727 021026
(5) 021032 104455
(6) 021034 000005
(6) 021036 005055
(6) 021040 006556

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

;ERROR, CRAM PC IS WRONG

5729 021042
(3) 021042 104410
(3) 021044 000002
5730 021046
(3) 021046
(3) 021046 104405
5731 021050
(3) 021050 104404
5732 021052 004737 003166
5733 021056
(1) 021056 004537 003100
5734 021062 100406
5735 021064
(1) 021064 004537 003100
5736 021070 107125
5737 021072 004737 003330
5738 021076 000007
5739 021100 120504
5740 021102 001406
5741 021104
(5) 021110 104455
(6) 021112 000005
(6) 021114 005055
(6) 021116 006556
5742 021120
(3) 021120 104410
(3) 021122 000002
5743 021124
(3) 021124
(3) 021124 104405
5744 021126
(3) 021126
(3) 021126 104401
5745
5746 021130
(2)
5747
5748
5749
5750
5751
5752
5753
5754
5755 021130
(2)
5756
5757 021130
(3) 021130
5758 021130
(1)
(4) 021140 104432
(4) 021142 000240
5759 021144
(1) 021144 013701 002516
5760 021150

4\$: ESCAPE SEG
TRAP C\$ESCAPE
.WORD 10001\$-.
ENDSEG
10001\$: TRAP C\$ESEG
BGNSEG
TRAP C\$BSEG
JSR PC,CLRALL ;CLEAR ALL CONDITIONS
SROMCLK ;NEXT WORD IS INSTRUCTION,
JSR R5,.SROMCLK
100406 ;START AT ROM PC=6
SROMCLK ;NEXT WORD IS INSTRUCTION,
JSR R5,.SROMCLK
104125!<400*6> ;JUMP TO ROM PC OF 525
JSR PC,RAMDAT ;R4=CRAM PC (LSB 8 BITS)
7 ;EXPECTED DATA
CMPB R5,R4 ;IS ROM PC CORRECT?
BEJ 6\$;BR IF YES
ERROR 5 ;ERROR, CRAM PC IS WRONG
TRAP C\$ERDF
.WORD 5
.WORD EM0
.WORD ERR5
6\$: ESCAPE SEG
TRAP C\$ESCAPE
.WORD 10002\$-.
ENDSEG
10002\$: TRAP C\$ESEG
ENDTST
L10071: TRAP C\$ETST
BADHEAD
:***** TEST 24 *****
:*CRAM TEST OF JUMP(I) ON BR7 CLEAR MICRO-PROCESSOR INSTRUCTION.
:*CLEAR THE BR7 BIT, PERFORM THE JUMP INSTRUCTION.
:*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
:*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
:*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
:*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
:*THE CRAM PC IS CORRECT, IF THE CRAM PC IS NOT RIGHT.
:*THEN PORT4 CONTAINS A 37
BADHEAD
:***** TEST 24 *****
BGNFST
T24:: MACEX2 ;DON'T DO IF M8200.
;DO NOT DO TEST IF M8200
TRAP C\$EXIT
.WORD L10072-.
MYINT
MOV KMCSR,R1 ;RECORD DEVICE ADDR.
MSTCLR ;MASTER CLEAR M8200,4,7

```
5761 021154 004737 003474      JSR    PC, MEMSET      ;SET MEM AND RAM
5762 021160      1$:  BGNSEG
      (3) 021160 104404      TRAP    C$BSEG
5763 021162 004737 003166      JSR    PC, CLRALL      ;CLEAR ALL CONDITIONS
5764 021166      SROMCLK      ;NEXT WORD IS INSTRUCTION.
      (1) 021166 004537 003100 JSR    R5, .SROMCLK
5765 021172 100400      100400      ;START AT ROM PC=0
5766 021174      SROMCLK      ;NEXT WORD IS INSTRUCTION.
      (1) 021174 004537 003100 JSR    R5, .SROMCLK
5767 021200 117777      114377! <400*7> ;JUMP TO ROM PC OF 1777
5768 021202 004737 003330      JSR    PC, RAMDAT      ;R4=CRAM PC (LSB 8 BITS)
5769 021206 000001      1      ;EXPECTED DATA
5770 021210 120504      CMPB    R5, R4      ;IS ROM PC CORRECT?
5771 021212 001406      BEQ     2$      ;BR IF YES
5772 021214      ERROR    5      ;ERROR, CRAM PC IS WRONG
      (5) 021220 104455      TRAP    C$ERDF
      (6) 021222 000005      .WORD   5
      (6) 021224 005055      .WORD   EMO
      (6) 021226 006556      .WORD   ERR5
5773 021230      2$:  ESCAPE SEG
      (3) 021230 104410      TRAP    C$ESCAPE
      (3) 021232 000002      .WORD   10000$-.
5774 021234      10000$: ENDSEG
      (3) 021234 104405      TRAP    C$ESEG
5775 021236      BGNSEG
      (3) 021236 104404      TRAP    C$BSEG
5776 021240      SKIP06 6$
      (1)      ;GOTO 6$ IF M8206
5777 021250 004737 003166      JSR    PC, CLRALL      ;CLEAR ALL CONDITIONS
5778 021254      SROMCLK      ;NEXT WORD IS INSTRUCTION.
      (1) 021254 004537 003100 JSR    R5, .SROMCLK
5779 021260 100403      100403      ;START AT ROM PC=3
5780 021262      SROMCLK      ;NEXT WORD IS INSTRUCTION.
      (1) 021262 004537 003100 JSR    R5, .SROMCLK
5781 021266 103400      100000! <400*7> ;JUMP TO ROM PC OF 0
5782 021270 004737 003330      JSR    PC, RAMDAT      ;R4=CRAM PC (LSB 8 BITS)
5783 021274 000004      4      ;EXPECTED DATA
5784 021276 120504      CMPB    R5, R4      ;IS ROM PC CORRECT?
5785 021300 001406      BEQ     4$      ;BR IF YES
5786 021302      ERROR    5      ;ERROR, CRAM PC IS WRONG
      (5) 021306 104455      TRAP    C$ERDF
      (6) 021310 000005      .WORD   5
      (6) 021312 005055      .WORD   EMO
      (6) 021314 006556      .WORD   ERR5
5787 021316      4$:  ESCAPE SEG
      (3) 021316 104410      TRAP    C$ESCAPE
      (3) 021320 000002      .WORD   10001$-.
5788 021322      10001$: ENDSEG
      (3) 021322 104405      TRAP    C$ESEG
5789 021324      BGNSEG
      (3) 021324 104404      TRAP    C$BSEG
5790 021326 004737 003166      JSR    PC, CLRALL      ;CLEAR ALL CONDITIONS
5791 021332      SROMCLK      ;NEXT WORD IS INSTRUCTION.
      (1) 021332 004537 003100 JSR    R5, .SROMCLK
```

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 PAGE 34-2
HARDWARE TESTS

SEQ 0132

```

5792 021336 100406          100406          ;START AT ROM PC=6
5793 021340          SROMCLK          ;NEXT WORD IS INSTRUCTION,
(1) 021340 004537 003100      JSR      R5,,SROMCLK
5794 021344 107525          104125!<400*7>      ;JUMP TO ROM PC OF 525
5795 021346 004737 003330      JSR      PC,RAMDAT          ;R4=CRAM PC (LSB 8 BITS)
5796 021352 000007          7          ;EXPECTED DATA
5797 021354 120504          CMPB      R5,R4          ;IS ROM PC CORRECT?
5798 021356 001406          BEQ      6$          ;BR IF YES
5799 021360          ERROR      5          ;ERROR, CRAM PC IS WRONG
(5) 021364 104455          TRAP      C$ERDF
(6) 021366 000005          .WORD      5
(6) 021370 005055          .WORD      EMO
(6) 021372 006556          .WORD      ERR5
5800 021374          6$:      ESCAPE SEG
(3) 021374 104410          TRAP      C$ESCAPE
(3) 021376 0000C2          .WORD      10002$-.
5801 021400          ENDSEG
(3) 021400          10002$:
(3) 021400 104405          TRAP      C$ESEG
5802 021402          ENDTST
(3) 021402          L10072:
(3) 021402 104401          TRAP      C$ETST
5803 021404          BADHEAD
5804 021404          :***** TEST 25 *****
(2) 5805          :*
5806          :*MAIN MEMORY PAGE DUAL ADDRESS TEST.
5807          :*IN THIS TEST WE WILL VERIFY THAT PAGES DO
5808          :*NOT DUAL ADDRESS. THIS TEST IS DIFFERENT FROM THE
5809          :*PREVIOUS DUAL ADDRESS TESTS IN THAT THE OTHER
5810          :*TEST REALLY DIDN'T CHECK PAGE DUAL ADDRESSING
5811 021404          BADHEAD
(2) 5812          :***** TEST 25 *****
5813 021404          BGNTST
(3) 021404          T25::
5814 021404          K4ONLY          ;FOR 4K CPUS ONLY.
(1) 5815          ;DO NOT DO TEST IF M8200, OR M8204
(4) 021414 104432          TRAP      C$EXIT
(4) 021416 000156          .WORD      L10073-.
5815 021420          MYINT
(1) 021420 013701 002516      MOV      KMCSR,R1          ;RECORD DEVICE ADDR.
5816 021424          MSTCLR
5817 021430 005002          CLR R2          ;R2 WILL BE PAGE #
5818 021432 042737 000037 021456 1$:      BIC #37,2$          ;CLEAR UNUSED BITS
5819 021440 050237 021456      BIS R2,2$          ;ADD CURRENT PAGE MARKER.
5820 021444          ROMCLK          ;SET ADDR D
(1) 021444 004537 003044      JSR      R5,,ROMCLK          ;CLOCK INSTRUCTION
5821 021450 010000          10000
5822 021452          ROMCLK          ;OF PAGE X
(1) 021452 004537 003044      JSR      R5,,ROMCLK          ;CLOCK INSTRUCTION
5823 021456 004000          2$:      4000          ;THIS LOCATION MODIFIED BY LOST
5824          ;FEW INSTRUCTIONS
5825 021460 010261 000004      MOV R2,4(R1)          ;PUT PAGE # INTO PART 4
5826 021464          ROMCLK          ;CLOCK PART 4 INTO MEMORY
(1) 021464 004537 003044      JSR      R5,,ROMCLK          ;CLOCK INSTRUCTION

```

5827	021470	122500				122500			;WHOSE PAGE # IS IN R2
5828	021472	005202				INC R2			;UPDATE PAGE #
5829	021474	032702	000020			BIT #20,R2			;DONE ALL PAGES?
5830	021500	001754				BEQ 1\$			;NO-DO NEXT ONE
5831									
5832									;OK, ALL LOADS OF ALL PAGES
5833									;CONTAIN THIER PAGE NUMBER, NOW
5834									;WE'LL GO BACK THROUGH
5835									;THEM CHECKING THEM OUT.
5836									
5837	021502	005002				CLR R2			;R2 STILL HAS PAGE NUMBER
5838									
5839	021504	042737	000037	021522	3\$:	BIC #37,4\$			
5840	021512	050237	021522			BIS R2,4\$			
5841	021516					ROMCLK			;LOAD PAGE NUMBER
(1)	021516	004537	003044			JSR R5,ROMCLK			;CLOCK INSTRUCTION
5842	021522	004000			4\$:	4000			
5843	021524					ROMCLK			;MOVE MEM TO PART 4
(1)	021524	004537	003044			JSR R5,ROMCLK			;CLOCK INSTRUCTION
5844	021530	041224				041224			
5845	021532	116104	000004			MOVB 4(R1),R4			;'FOUND'
5846	021536	110205				MOVB R2,R5			;'EXPECTED'
5847	021540	120504				CMPEB R5,R4			;ADDRESS PROBLEM?
5848	021542	001406				BEQ 5\$			
5849									
5850	021544					ERROR 13			;PAGE ADDRESSING ERROR IN MAIN
(5)	021550	104455				TRAP C\$ERDF			
(6)	021552	000015				.WORD 13			
(6)	021554	005055				.WORD EMO			
(6)	021556	007542				.WORD ERR13			
5851									;MEMORY.
5852									;MAR BITS 8 THROUGH 12 ARE REPRESENTED
5853									;BY R2 ('EXP'ED')BITS 0-4)
5854									
5855	021560				5\$:	ESCAPE TST			
(3)	021560	104410				TRAP C\$ESCAPE			
(3)	021562	000012				.WORD L10073-			
5856	021564	005202				INC R2			;UPDATE PAGE ADDRESS
5857	021566	032702	000020			BIT #20,R2			;ALL DONE?
5858	021572	001744				BEQ 3\$			;NO-CHECK NEXT PAGE.
5859									;YES-EXIT.
5860	021574				ENDTST				
(3)	021574				L10073:				
(3)	021574	104401				TRAP C\$ETST			
5861									
5862	021576					BADHEAD			
5863						***** TEST 26 *****			
(2)						;			
5864						;			
5865						*JUMP FIELD,PAGE TEST			
5866						;			
5867						*IN THIS TEST WILL MAKE SURE A JUMP FIELD INSTRUCTION			
5868						*WORKS. TO DO THIS, WE'LL PUT THE DESIRED PAGE,FIELD			
5869						*INORMATION IN !BUS*<13> THEN ISSUE A JUMP FIELD			
5870						*THEN WF'LL READ PC REG. AND VERIFY.			
5871	021576					BADHEAD			

```

(2)
5872
5873 021576
(3) 021576
5874 021576
(1)
(4) 021606 104432
(4) 021610 000132
5875 021612
(1) 021612 013701 002516
5876 021616
5877
5878 021622 005002
5879
5880 021624 042737 000017 021642 1$:
5881 021632 050237 021642
5882
5883 021636
(1) 021636 004537 003044
5884 021642 000400 2$:
5885 021644
(1) 021644 004537 003044
5886 021650 061233
5887 021652
(1) 021652 004537 003100
5888 021656 100000
5889
5890
5891 021660 142761 000002 000001
5892 021666
(1) 021666 004537 003044
5893 021672 121264
5894 021674 116104 000004
5895 021700 042704 177760
5896 021704 120402
5897 021706 001407
5898 021710 010205
5899 021712
(5) 021716 104455
(6) 021720 000016
(6) 021722 005055
(6) 021724 007664
5900
5901
5902
5903 021726
(3) 021726 104410
(3) 021730 000012
5904
5905
5906 021732 005202
5907 021734 032702 000020
5908 021740 001731
5909
5910 021742
(3) 021742

***** TEST 26 *****
BCNTST
T26::
K4ONLY
;DO NOT DO TEST IF M8200, OR M8204
TRAP C$EXIT
;WORD L10074-.
MYINT
MOV KMCSR,R1 ;RECORD DEVICE ADDR.
MSTCLR
CLR R2 ;R2 TO CONTAIN FIELD #
BIC #17,2$ ;CLEAR ANY JUNK
BIS R2,2$ ;SET FIELD # INTO INSTR.
ROMCLK ;CLOCK FIELD BITS INTO BREG.
JSR R5,ROMCLK ;CLOCK INSTRUCTION
000400 ;CONTAINS FIELD,PAGE BITS
ROMCLK ;XFER BREG INTO IBUS*13>
JSR R5,ROMCLK ;CLOCK INSTRUCTION
061233
SROMCLK ;GET INSTRUCTION Clocked.
JSR R5,SROMCLK ;BAS FORM FOR JUM FIELD INSTR.
100000
BICB #BIT1,1(R1) ;CLEAR ROMI
ROMCLK ;CLOCK NEXT INSTR.
JSR R5,ROMCLK ;CLOCK INSTRUCTION
121264 ;MOVE IBUS*TO PORT 4
MOVB 4(R1),R4 ;GET IT.
BIC #^C<17>,R4
CMPB R4,R2 ;FIELD OK?
BEQ 3$ ;IF OK GO AHEAD
MOV R2,R5
ERROR 14 ;CHANGE FIELD INSTRUCTION ;*** B0
TRAP C$ERDF
;WORD 14
;WORD EM0
;WORD ERR14
;FAILED. FOR FIELD,PAGE INDICATES
;BY 'EXPECTED' BITS 0,1,2,3 OF
;EXPECTED REPRESENT FIELD BITS.
3$:
ESCAPE TST
TRAP C$ESCAPE
;WORD L10074-.
INC R2
BIT #20,R2
BEQ 1$
;UPDATE TO NEXT FIELD
;DONE ALL FIELDS?
ENDTST
L10074:

```

(3) 021742 104401

5911

5912 021744

(2)

5913

5914

5915

5916

5917

5918

5919

5920

5921

5922

5923

5924

5925

5926

5927

5928

5929

5930

5931

5932

5933

5934

5935

5936

5937

5938

5939

5940

5941

5942

5943 021744

(2)

5944

5945 021744

(3) 021744

5946 021744

(1) 021744 J13701 002516

5947 021750

(1)

(4) 021760 104432

(4) 021762 000336

5948 021764

5949

5950 021770 012737 000000 002406

5951

5952

5953

5954 021776 012702 000000

5955

5956

5957 022002 012737 000000 002412

5958

TRAP CSETST

BADHEAD

***** TEST 27 *****

;

;*JUMP TEST, JUMP ALWAYS, JUMP CHANGE FIELD

;

;*IN THIS TEST, WE WILL CHECK THE ABILITY OF THE

;*MICRO PROCESSOR TO JUMP (BRANCH & ALWAYS INSTRUCTION)

;*TO LOCATIONS, FIELDS FROM OTHER LOCATIONS FIELDS.

;*WE ALREADY KNOW THAT THE BRANCH INSTR WORKS FROM

;*OTHER TEST. PROCEDURE:

1. START ADDR 0, FIELD 0

2. **CALCULATE NEW ADDR, FIELD VIA INC,

3. CAUSE JUMP (BRANCH) TO NEW ADDRESS

4. READ PC FROM IBUS*12 AND IBUS*13

5. REPEAT STEP 2-4 256.TIMES

;

TO CALCULATE NEW ADDRESS:

1. INC LOW BYTE OF ADDRESS FOR PC ADDRESS 0-7

2. INC LOW BYTE OF NADDRESS FOR PC ADDRESS 8-1

BITS REPRESENTED AS BITS 0-3. WHEN 0-3 OVERFLOWS,

RESTARTS AT ZERO.

NET RESULT IS JUMPS FROM:

FIELD,PAGE	LOC
0	0
1	1
2	2
3	3
10	7
11	11
16	15
17	377

BADHEAD

***** TEST 27 *****

BGNTST

127::

MYINT

MOV KMCSR,R1 ;RECORD DEVICE ADDR.

K4ONLY ;4K CPUS ONLY.

DO NOT DO TEST IF M8200, OR M8204

TRAP C\$EXIT

WORD L10075-

MSTCLR

MOV #0, FLAG ;FLAG TO REPRESENT

FIELD,PAGE

TO VARIE STARTING PAGE,FIELD,

CHANGE #0 PORTION OF INSTR.

R2 TO CONTAIN JUMPED

TO CHANGE STARTING IMM ADDR.,

VARIE #0 PORTIONS OF INSTR.

ADDRESS

MOV #0, FADR

LOOP HERE

5959	022010			18:		
5960	022010	042737	000017	022050	BIC #17,28	:CLEAR JUNK FROM FIELD
5961						:PORTION OF CHANGE FIELD INSTR
5962	022016	013700	002406		MOV FLAG,R0	:INORDER TO INC, DEC FIELD,PAGE
5963	022022	042700	177760		BIC #17,28	
5964	022026	050037	022050		BIS R0,28	:NOW POSITION IN INSTR.
5965	022032	042737	077777	022064	BIC #077777,38	:NOW FOR IMMED. BR INSTR.
5966	022040	050237	022064		BIS R2,38	:NOW ADD IMMEDIATE ADDR
5967						
5968						
5969						
5970	022044				ROMCLK	
(1)	022044	004537	003044	28:	JSR R5,ROMCLK	:CLOCK INSTRUCTION
5971	022050	000400			000400	:MOVE PAGE, FIELD # TO BREG.
5972	022052				ROMCLK	
(1)	022052	004537	003044		JSR R5,ROMCLK	:CLOCK INSTRUCTION
5973	022056	061233			61233	:MOV BREG TO PC HIGH REG.
5974	022060				SROMCLK	
(1)	022060	004537	003100		JSR R5,SROMCLK	
5975	022064	100000		38:	100000	:NOW CLOCK IT IN BY JMP FIELD INSTR.
5976						
5977	022066				ROMCLK	:READ PC REG HI
(1)	022066	004537	003044		JSR R5,ROMCLK	:CLOCK INSTRUCTION
5978	022072	121265			121265	
5979	022074				ROMCLK	:READ PC REG LOW
(1)	022074	004537	003044		JSR R5,ROMCLK	:CLOCK INSTRUCTION
5980	022100	121244			121244	
5981						
5982	022102	016104	000004		MOV 4(R1),R4	:READ PC REG (NOW IN SEL 4)
5983	022106	042704	170000		BIC #170000,R4	:STRIP FOR ONLY PAGE, FIELD BITS.
5984						
5985	022112	013705	022050		MOV 28,R5	:NOW FROM ADDR WE WANTED TO
5986	022116	000305			SWAB R5	:JUMP TO
5987	022120	042705	170377		BIC #170377,R5	:CLEAR JUNK
5988	022124	050205			BIS R2,R5	:ADD IMMED ADDR
5989	022126				SKIP06 58	
(1)					:GOTO 58 IF MB206	
5990	022136	105205		58:	INCB R5	:UPDATE ADDR. EXPECTED SENCE THE READ
5991	022140					:OF THE IBUS <13> INC THE PC.
5992						
5993						
5994	022140	020504			CMP R5,R4	:JUMP GO OK?
5995	022142	001406			BEQ 48	:YEA, CONTINUES
5996	022144				ERROR 15	:FAILED TO JUMP PROPERLY.
(5)	022150	104455			TRAP (SERDF	
(6)	022152	000017			.WORD 15	
(6)	022154	005055			.WORD EMC	
(6)	022156	010006			.WORD ERR15	
5997						:FROM ADDR" REPRESENTS
5998						:THE ADDRESS WE STARTED AT
5999						:TO ADDR" REPRESENTS WHERE
6000						:WE EXPECTED TO JUMP TO.
6001						:BAD ADDR" REPRESENTS WHERE
6002						:WE WENT TO.
6003						
6004						

6005
6006
6007
6008
6009
6010
6011
6012
6013
6014
6015
6016
6017
6018
6019
6020
6021
6022
6023
6024
6025
6026
6027
(3)
(3)
6028
6029
6030
6031
6032
6033
6034
6035
6036
6037
(1)
6038
6039
6040
6041
6042
6043
6044
6045
6046
6047
6048
6049
(1)
6050
6051
(1)
6052
6053
(1)
6054

022160
022160 104410
022162 000136
022164 010437 002412
022170 005237 002406
022174 105202
022176 001304

022200

022210 005737 002470
022214 001041
022216 005737 002472
022222 001036
022224 052711 040000
022230 105761 000001
022234 042711 040000

022240
(1) 022240 004537 003044
022244 121265
022246
(1) 022246 004537 003044
022252 121244
022254
(1) 022254 004537 003044
022260 121265

2
48:

ESCAPE TST
TRAP CSESCAPE
WORD L10075-
MOV R4,FADR
INC FLAG
INCB R2
BNE 18

:UPDATE PAGE, FIELD
:UPDATE JUMPED. ADDR
:LOOP IF NOT DONE.

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

SKIP06 408
:GOTO 408 IF MB206
TST RUMB
BNE 408
TST RUMINH
BNE 408
BIS #40000,(R1) :SET MASTER CLEAR
TSTB ?(R1)
BIC #40000,(R1)

:TO RUN THIS SECTION OF CODE YOU MUST TURN SW7 OF SWITCH PACK #E28
:OFF SO THAT MB207 NOT SELFSTARTING.

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

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

:THIS IS A PICTURE OF THE P.C. REG.
BITS 0-7 ARE IN IBUS<12>
BITS 8-11 ARE IN IBUS<13>
THEY GOT CLOCK IN THERE VIA JUMPS TAKEN
THE FIELD BITS
ARE IN BIT POSITION 0,1 OF THE INSTRUCTION AT 28.

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

```

6055
6056 022262 ROMCLK :REG PC REG LOW, PUT IN PORT4
        ) 022262 004537 003044 JSR R5, ROMCLK :CLOCK INSTRUCTION
6057 022266 121244 :
6058 022270 005005 CLR R5 :EXPECT ZERO
6059 022272 016104 000004 MOV 4(R1), R4 :READ PC REG FROM PORT 485
6060 022276 042704 170003 BIC #170003, R4
6061 022302 001406 BEG 408 :IF CLEARED, EXIT
6062 :NOTE WE ALSO CLEARED BIT 1 OF THE
6063 :PC REG, BECAUSE AFTER THE MASTER
6064 :CLEAR, WE DID TWO INSTRUCTIONS TO
6065 :READ IT, THUS CAUSING THE PC REG
6066 :TO GET BUMPED.
6067
6068 022304 ERROR 45 :MASTER CLEAR FAILED TO CLEAR
        (5) 022310 104455 TRAP CSEDEF
        (6) 022312 000055 .WORD 45
        (6) 022314 005055 .WORD EMO
        (6) 022316 011124 .WORD EHR45
6069 :PC REG
6070 022320
6071 022320 408:
        (3) 022320 FNDTST:
        (3) 022320 104401 L10075:
6072 022322 TRAP CSE1ST
        (2) BADHEAD
6073 :***** TEST 28 *****
6074 :
6075 :JUMP TEST, JUMP ALWAYS, JUMP CHANGE FIELD
6076 :
6077 :IN THIS TEST, WE WILL CHECK THE ABILITY OF THE
6078 :MICRO PROCESSOR TO JUMP (BRANCH & ALWAYS INSTRUCTION)
6079 :TO LOCATIONS, FIELDS FROM OTHER LOCATIONS FIELDS.
6080 :WE ALREADY KNOW THAT THE BRANCH INSTR WORKS FROM
6081 :OTHER TEST. PROCEDURE:
6082 :1. START ADDR 0, FIELD 0
6083 :2. **CALCULATE NEW ADDR, FIELD VIA DEC.
6084 :3. CAUSE JUMP (BRANCH) TO NEW ADDRESS
6085 :4. READ PC FROM IBUS*12 AND IBUS*13
6086 :5. REPEAT STEP 2-4 256.TIMES
6087 :
6088 :TO CALCULATE NEW ADDRESS:
6089 :1. DEC LOW BYTE OF ADDRESS FOR PC ADDRESS 0-7
6090 :2. DEC LOW BYTE OF NADDRESS FOR PC ADDRESS 8-11
6091 :BITS REPRESENTED AS BITS 0-3. WHEN 0-3 OVERFLOWS,
6092 :RESTARTS AT ZERO.
6093 :NET RESULT IS JUMPS FROM:
6094 :FIELD, PAGE LOC
6095 :0 0
        :17 377
    
```

6097						16	376
6098						15	375
6099						:TO	:
6100						00	000
6101							
6102	022322					BADHEAD	
(2)						***** TEST 28 *****	
6103							
6104	022322					BGNTST	
(3)	022322						
6105	022322					MYINT	
(1)	022322	013701	002516			MOV KMCSR,R1	:RECORD DEVICE ADDR.
6106	022326					K4ONLY	:4K CPUS ONLY.
(1)						:DO NOT DO TEST IF M8200, OR M8204	
(4)	022336	104432				TRAP C\$EXIT	
(4)	022340	000216				.WORD L10076-	
6107	022342					MSTCLR	
6108							
6109	022346	012737	000000	002406		MOV #0, FLAG	:FLAG TO REPRESENT
6110							:FIELD,PAGE
6111							:TO VARIE STARTING PAGE,FIELD,
6112							:CHANGE #0 PORTION OF INSTR.
6113	022354	012702	000000			MOV #0, R2	:R2 TO CONTAIN JUMPED
6114							:TO CHANGE STARTING IMM ADDR.,
6115							:VARIE #0 PORTIONS OF INSTR.
6116	022360	012737	000000	002412		MOV #0, FADR	:ADDRESS
6117						:LOOP HERE	
6118	022366						
6119	022366	042737	000017	022426	1\$:	BIC #17,2\$	:CLEAR JUNK FROM FIELD
6120							:PORTION OF CHANGE FIELD INSTR
6121	022374	013700	002406			MOV FLAG,R0	:INORDER TO INC, DEC FIELD,PAGE
6122	022400	042700	177760			BIC #^C<17>,R0	
6123	022404	050037	022426			BIS R0,2\$	:NOW POSITION IN INSTR.
6124	022410	042737	077777	022442		BIC #077777,3\$	:NOW FOR IMMED. BR INSTR.
6125	022416	050237	022442			BIS R2,3\$	:NOW ADD IMMEDIATE ADDR
6126							
6127							
6128							
6129	022422					ROMCLK	
(1)	022422	004537	003044			JSR R5,.ROMCLK	:CLOCK INSTRUCTION
6130	022426	000400			2\$:	000400	:MOVE PAGE,FIELD # TO BREG.
6131	022430					ROMCLK	
(1)	022430	004537	003044			JSR R5,.ROMCLK	:CLOCK INSTRUCTION
6132	022434	061233				61233	:MOV BREG TO PC HIGH REG.
6133	022436					SROMCLK	
(1)	022436	004537	003100			JSR R5,.SROMCLK	
6134	022442	100000			3\$:	100000	:NOW CLOCK IT IN BY JMP FIELD INSTR.
6135							
6136	022444					ROMCLK	:READ PC REG HI
(1)	022444	004537	003044			JSR R5,.ROMCLK	:CLOCK INSTRUCTION
6137	022450	121265				121265	
6138	022452					ROMCLK	:READ PC REG LOW
(1)	022452	004537	003044			JSR R5,.ROMCLK	:CLOCK INSTRUCTION
6139	022456	121244				121244	
6140							
6141	022460	016104	000004			MOV 4(R1),R4	:READ PC REG (NOW IN SEL 4)

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36MACY11 30A(1052) 21-JUL-81 14:48 PAGE 35-1
HARDWARE TESTS

SEQ 0140

```

6142 022404 042704 170000      BIC      #170000,R4      ;STRIP FOR ONLY PAGE, FIELD BITS.
6143
6144 022470 013705 022426      MOV 2$,R5      ;NOW FROM ADDR WE WANTED TO
6145 022474 000305              SWAB R5      ;JUMP TO
6146 022476 042705 170377      BIC #170377,R5      ;CLEAR JUNK
6147 022502 050205              BIS R2,R5      ;ADD IMMED ADDR
6148 022504
(1)
6149 022514 105205              ;GOTO 5$ IF M8206
6150 022516      5$:      INCB      R5      ;UPDATE ADDR. EXPECTED SENCE THE READ
6151
6152
6153 022516 020504              CMP R5,R4      ;JUMP GO OK?
6154 022520 001406              BEQ 4$      ;YEA, CONTINUES
6155 022522      ERROR 15      ;FAILED TO JUMP PROPERLY.
(5) 022526 104455              TRAP      C$ERDF
(6) 022530 000017              .WORD      15
(6) 022532 005055              .WORD      EMO
(6) 022534 010006              .WORD      ERR15

6156
6157
6158
6159
6160
6161
6162
6163      .REM %
6164
6165
6166
6167
6168
6169
6170
6171
6172
6173
6174
6175
6176
6177
6178
6179
6180
6181
6182
6183
6184
6185
6186 022536
(3) 022536 104410
(3) 022540 000016
6187 022542 010437 002412
6188 022546 005337 002406
6189 022552 105302
6190 022554 001304

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

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

%
4$:      ESCAPE TST
      TRAP      C$ESCAPE
      .WORD      L10076-.
      MOV R4,FADR
      DEC FLAG
      DECB R2
      BNE 1$
;UPDATE PAGE, FIELD
;UPDATE IMMED. ADDR
;LOOP IF NOT DONE.

```

15	12	11	10	9	8	7	0

UNUSED		FIELD		PAGE		IMMED. ADDR	
		BITS		BITS			

6191
6192
6193 022556
(3) 022556
(3) 022556 104401
6194 022560
(2)
6195
6196
6197
6198
6199
6200 022560
(2)
6201
6202 022560
(3) 022560
6203 022560
(1)
(4) 022570 104432
(4) 022572 000200
6204 022574
6205 022600
(1) 022600 013701 002516
6206 022604 004737 003166
6207 022610
(1) 022610 004537 003044
6208 022614 121264
6209 022616 116104 000004
6210 022622 042704 177477
6211 022626 012705 000000
6212 022632 120405
6213 022634 001410
6214 022636
(5) 022642 104455
(6) 022644 000020
(6) 022646 005055
(6) 022650 010134
6215
6216 022652
(3) 022652 104410
(3) 022654 000116
6217 022656 004737 003312
6218 022662
(1) 022662 004537 003044
6219 022666 121264
6220
6221 022670 016104 000004
6222 022674 042704 177477
6223 022700 012705 000200
6224 022704 120405
6225 022706 001410
6226 022710
(5) 022714 104455
(6) 022716 000020
(6) 022720 005055

ENDTST
L10076:

TRAP C\$ETST
BADHEAD
;***** TEST 29 *****
;
;* IN THIS TEST WE'LL VERIFY THAT THE Z BIT CAN BE READ FROM
;* IBUS*<13>. WE ALLREADY KNOW THAT THE Z BIT WORKS PROPERLY.
;* ALL WE WANT TO KNOW HERE IS THAT IT CAN BE READ.
;
BADHEAD
;***** TEST 29 *****

T29::

BGNTST
K4ONLY ;M8206 &M8207 ONLY!
;DO NOT DO TEST IF M8200, OR M8204
TRAP C\$EXIT
.WORD L10077-.
MSTCLR
MYINT
MOV KMCSR,R1 ;RECORD DEVICE ADDR.
JSR PC,CLRALL ;CLR CONDITION CODES.
ROMCLK ;NOW READ IBUS*<15>PUT IN PORT 4
JSR R5,.ROMCLK ;CLOCK INSTRUCTION
121264
MOVB 4(R1),R4 ;READ IT FROM PORT 4
BIC #177477,R4 ;STRIP ANY JUNK,C&Z BITS 6,7
MOV #0,R5 ;EXPECT IT CLEAR
CMPB R4,R5 ;OK?
BEQ 1\$
ERROR 16 ;FAILURE OF Z&C TO BE CLEAR.
TRAP C\$ERDF
.WORD 16
.WORD EMO
.WORD ERR16

1\$:

ESCAPE TST
TRAP C\$ESCAPE
.WORD L10077-.
JSR PC,SETZ ;SET Z BIT.
ROMCLK ;NOW GO BACK AND CHECK Z BIT SET.
JSR R5,.ROMCLK ;CLOCK INSTRUCTION
121264
MOV 4(R1),R4 ;GET INFO.
BIC #^C<300>,R4 ;STRIP FOR C&Z BITS.
MOV #200,R5 ;EXPECT ONLY Z BIT SET.
CMPB R4,R5 ;SET OK?
BEQ 2\$
ERROR 16 ;Z BIT FAILED TO SET PROPERLY.
TRAP C\$ERDF
.WORD 16
.WORD EMO

```
(6) 022722 010134 .WORD ERR16
6227
6228 022724 ESCAPE TST
(3) 022724 104410 TRAP C$ESCAPE
(3) 022726 000044 .WORD L10077-
6229 022730 004737 003166 2$: JSR PC,CLRALL ;NOW TRY TO CLEAR Z BIT.
6230 022734 ROMCLK
(1) 022734 004537 003044 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
6231 022740 121264 121264
6232 022742 016104 000004 MOV 4(R1),R4
6233 022746 042704 177477 BIC #^C<300>,R4 ;STRIP FOR C&Z BITS
6234 022752 001407 BEQ 3$ ;IF ZERO,WE'RE OK
6235 022754 005005 CLR R5 ;ELSE REPORT ERROR
6236 022756 ERROR 16 ;Z BIT FAILED TO CLEAR PROPERLY.
(5) 022762 104455 TRAP C$ERDF
(6) 022764 000020 .WORD 16
(6) 022766 005055 .WORD EMO
(6) 022770 010134 .WORD ERR16
6237 022772 3$:
6238 022772 L10077: ENDTST
(3) 022772 TRAP C$ETST
(3) 022772 104401 ;FINDFAST
6239
6240 022774 BADHEAD
(2) ;***** TEST 30 *****
6241 ;*
6242 ;* IN THIS TEST WE'LL VERIFY THAT THE C BIT CAN BE READ FROM
6243 ;* IBUS*<13>. WE ALLREADY KNOW THAT THE C BIT WORKS PROPERLY,
6244 ;* ALL WE WANT TO KNOW HERE IS THAT IT CAN BE READ.
6245 ;*
6246 022774 BADHEAD
(2) ;***** TEST 30 *****
6247
6248 022774 BGNTST
(3) 022774 T30::
6249 022774 K4ONLY ;M8206 &M8207 ONLY!
(1) ;DO NOT DO TEST IF M8200, OR M8204
(4) 023004 104432 TRAP C$EXIT
(4) 023006 000200 .WORD L10100-
6250 023010 MSTCLR
6251 023014 MYINT
(1) 023014 013701 002516 MOV KMCSR,R1 ;RECORD DEVICE ADDR.
6252 023020 004737 003166 JSR PC,CLRALL ;CLR CONDITION CODES.
6253 023024 ROMCLK ;NOW READ IBUS*<13>PUT IN PORT 4
(1) 023024 004537 003044 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
6254 023030 121264 121264
6255 023032 116104 000004 MOV 4(R1),R4 ;READ IT FROM PORT 4
6256 023036 042704 177477 BIC #177477,R4 ;STRIP ANY JUNK,C&Z BITS 6,7
6257 023042 012705 000000 MOV #0,R5 ;EXPECT IT CLEAR
6258 023046 120405 CMPB R4,R5 ;OK?
6259 023050 001410 BEQ 1$
6260 023052 ERROR 16 ;FAILURE OF Z&C TO BE CLEAR.
(5) 023056 104455 TRAP C$ERDF
(6) 023060 000020 .WORD 16
(6) 023062 005055 .WORD EMO
(6) 023064 010134 .WORD ERR16
```

```
6261
6262 023066          ESCAPE TST
(3) 023066 104410    TRAP C$ESCAPE
(3) 023070 000116    .WORD L10100-.
6263 023072 004737 003260 1$: JSR PC,SETC          ;SET C BIT.
6264 023076          ROMCLK          ;NOW GO BACK AND CHECK C BIT SET.
(1) 023076 004537 003044 JSR R5,.ROMCLK      ;CLOCK INSTRUCTION
6265 023102 121264    121264
6266 023104 016104    000004 MOV 4(R1),R4          ;GET INFO.
6267 023110 042704    177477 BIC #^C<300>,R4      ;STRIP FOR C&Z BITS.
6268 023114 012705    000100 MOV #100,R5          ;EXPECT ONLY C BIT SET.
6269 023120 120405    CMPB R4,R5        ;SET OK?
6270 023122 001410    BEQ 2$
6271 023124          ERROR 16           ;C BIT FAILED TO SET PROPERLY.
(5) 023130 104455    TRAP C$ERDF
(6) 023132 000020    .WORD 16
(6) 023134 005055    .WORD EMO
(6) 023136 010134    .WORD ERR16
6272
6273 023140          ESCAPE TST
(3) 023140 104410    TRAP C$ESCAPE
(3) 023142 000044    .WORD L10100-.
6274 023144 004737 003166 2$: JSR PC,CLRALL        ;NOW TRY TO CLEAR C BIT.
6275 023150          ROMCLK          ;CLOCK INSTRUCTION
(1) 023150 004537 003044 JSR R5,.ROMCLK
6276 023154 121264    121264
6277 023156 016104    000004 MOV 4(R1),R4
6278 023162 042704    177477 BIC #^C<300>,R4      ;STRIP FOR C&Z BITS
6279 023166 001407    BEQ 3$           ;IF ZERO,WE'RE OK
6280 023170 005005    CLR R5           ;ELSE REPORT ERROR
6281 023172          ERROR 16           ;C BIT FAILED TO CLEAR PROPERLY.
(5) 023176 104455    TRAP C$ERDF
(6) 023200 000020    .WORD 16
(6) 023202 005055    .WORD EMO
(6) 023204 010134    .WORD ERR16
6282 023206          3$:
6283 023206          L10100:
(3) 023206 104401    TRAP C$ETST
(3) 023206          BADHEAD
6284 023210          ;***** TEST 31 *****
(2) 6285          ;*TEST OF PROGRAM CLOCK BIT
6286          ;*DO A MASTER CLEAR, VERIFY THAT PROGRAM CLOCK IS SET
6287          ;*WRITE PROGRAM CLOCK BIT TO A ONE, VERIFY THAT IT CLEARS,
6288          ;*AND THEN SETS SOME TIME LATER
6289 023210          BADHEAD
(2) 6290          ;***** TEST 31 *****
6291 023210          BGNTST
(3) 023210          T31::
6292 023210          MYINT
(1) 023210 013701 002516 MOV KMCSR,R1          ;RECORD DEVICE ADDR.
6293 023214          MSTCLR          ;MASTER CLEAR M8200,4,7
6294 023220 005037 002440 CLR TEMP          ;PREPARE FOR
6295 023224 005037 002444 CLR $TMP0        ;DELAY
6296 023230 012761 000020 000004 1$: MOV #20,4(R1) ;LOAD PORT 4
```

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACV11 30A(1052) 21-JUL-81 14:48 PAGE 35-5
HARDWARE TESTS

SEQ 0144

```

6297 023236 152761 000002 000001 BISB #BIT1,1(R1) ;SET ROMI
6298 023244 012761 121111 000006 MOV #121111,6(R1) ;SEL6 INSTRUCTION
6299 023252 152761 000003 000001 BISB #BIT1!BIT0,1(R1);SET CLOCK BIT
6300 023260 012761 121224 000006 MOV #121224,6(R1) ;LOAD NEXT INSTRUCTION
6301 023266 152761 000003 000001 BISB #BIT1!BIT0,1(R1);READ CLOCK BIT
6302 023274 142761 030001 000001 BICB #BIT!BIT0,1(R1);CLEAR MAINT BITS
6303 023302 016104 000004 MOV 4(R1),R4 ;PUT 'FOUND' IN R4
6304 023306 005037 002452 CLR $GDDAT ;PUT 'EXPECTED' IN $GDDAT
6305 023312 123704 002452 CMPB $GDDAT,R4 ;IS PGM CLOCK CLEAR?
6306 023316 001406 BEQ 2$
6307 023320 013702 002452 MOV $GDDAT,R2
6308 023324 ERRDF 50,EMB50 ;ERROR, PGM CLOCK IS NOT CLEAR
(4) 023324 104455 TRAP C$ERDF
(5) 023326 000062 .WORD 50
(5) 023330 005545 .WORD EMB50
(5) 023332 0000C0 .WORD 0
6309 023334 2$: ROMCLK ;NEXT WORD IS INSTRUCTION,
6310 023334 JSR R5,ROMCLK ;CLOCK INSTRUCTION
(1) 023334 004537 003044 121224 ;PORT4 LU11
6311 023340 121224 CMPB #20,4(R1) ;IS PGM CLOCK SET?
6312 023342 122761 000020 000004 BEQ 3$ ;BR IF YES
6313 023350 001420 INC TEMP ;INCREMENT DELAY
6314 023352 005237 002440 ADC $TMP0 ;INCREMENT DELAY
6315 023356 005537 002444 CMP #6,$TMP0 ;IS DELAY DONE
6316 023362 022737 000006 002444 BNE 2$ ;BR IF NO
6317 023370 001361 MOV #6,R2
6318 023372 012702 000006 MOV $TMP0,R4
6319 023376 013704 002444 ERRDF 51,EMB51 ;ERROR PGM CLOCK NOT SET
6320 023402 104455 TRAP C$ERDF
(4) 023402 104455 .WORD 51
(5) 023404 000063 .WORD EMB51
(5) 023406 005577 .WORD 0
(5) 023410 000000 3$:
6321 023412 6322
6323 023412 122737 000007 002414 CMPB #7,WIYPE ; ONLY DO NEXT TEST IF M8207
6324 023420 001013 BNE 4$ ; EXIT IF NOT.
6325
6326 023422 005737 002444 TST $TMP0 ; IF ANY LARGE COUNT, WE'RE OK
6327 023426 001010 BNE 4$ ; THEN EXIT
6328
6329 023430 042737 000007 002440 BIC #7,TEMP ; CLEAR OUT ANY SMALL COUNT
6330 023436 001004 BNE 4$ ; IF LARGE COUNT LEFT OVER, WE'RE OK.
6331
6332 023440 ERRDF 100,EMB1 ; ERROR
(4) 023440 104455 TRAP C$ERDF
(5) 023442 000144 .WORD 100
(5) 023444 005410 .WORD EMB1
(5) 023446 000000 .WORD 0
6333
6334 ; TIME T1000 SHORT FOR CLOCK. MUST BE
6335 023450 4$: ; DEFECTIVE CAPACITOR IN TIMEING CIRCUIT.
6336
6337 023450 ENDTST
(3) 023450 L10101:
(3) 023450 104401 TRAP C$ETST

```

```

6338
6339 023452      BADHEAD
(2)              :***** TEST 32 *****
6340              :*FORCE POWER FAIL TEST
6341              :*SET FORCE POWER FAIL BIT VERIFY THAT PROCESSOR TRAPS TO 24
6342              :*GOING DOWN AND COMING UP. VERIFY ALSO THAT BUS INIT WAS
6343              :*BLOCKED FROM GETTING TO THE M8200,4,7 DURING THE POWER FAIL
6344              :*THIS TEST WILL TAKE LONGER THAN 2 SECONDS TO RUN. THIS TEST
6345              :*SHOULD NOT BE RUN IF YOU HAVE VOLATILABLE MEMORY IN YOUR SYSTEM.
6346 023452      BADHEAD
(2)              :***** TEST 32 *****
6347
6348 023452      BGNTST
(3) 023452      T32::
6349 023452      BRESET          ;STALL FOR TIME
(3) 023452 104433 TRAP          C$RESET
6350 023454      MYINT          ;
(1) 023454 013701 002516 MOV          KMCSR,R1          ;RECORD DEVICE ADDR.
6351              ;R1 CONTAINS BASE M8200,4,7 ADDRESS
6352 023460      MSTCLR          ;MASTER CLEAR M8200,4,7
6353 023464 005037 002440 CLR          TEMP          ;PREPARE FOR DELAY
6354 023470 013737 000024 002444 MOV          @#24,$TMP0      ;SAVE POWER FAIL ADDRESS
6355 023476 013746 000024      MOV          @#24,-(SP)      ;STORE POWER FAIL ADDRESS
6356 023502 012737 023564 000024 MOV          #1$,@#24-      ;SET U FOPR FORCE POWER FAIL
6357 023510 012761 000002 000004 MOV          #2,4(R1)      ;LOAD PORT4
6358 023516 012711 001000      MOV          #BIT9,(R1)      ;SET ROMI
6359 023522 012761 121111 000006 MOV          #121111,6(R1)      ;LOAD INSTRUCTION
6360 023530 012711 005400      MOV          #BIT9!BIT8.BIT11,(R1) ;CLOCK INSTRUCTION
6361 023534 005237 002440 5$: INC          TEMP          ;WAIT FOR POWER FAIL
6362 023540 001375      BNE          5$          ;BR IF DELAY NOT DONE
6363 023542      MSTCLR
6364 023546      ERROR          17          ;ERROR, NO POWER FAIL
(5) 023552 104455 TRAP          C$ERDF
(6) 023554 000021      .WORD          17
(6) 023556 005055      .WORD          EMO
(6) 023560 010256      .WORD          ERR17
6365 023562 000445 BR          4$
6366 023564 012737 023602 000024 1$: MOV          #3$,@#24      ;POWER UP ADDRESS
6367 023572 010637 023600      MOV          SP,2$          ;STORE STACK
6368 023576 000000      HALT          ;WAIT FOR POWER UP SEQUENCE
6369 023600 000000      2$: 0
6370 023602 013706 023600 3$: MOV          2$,SP          ;RESTORE STACK
6371 023606 012737 024002 000024 MOV          #10$,@#24      ;PUT IN CASE OF FALSE POWER-UP.
6372 023614 005037 024000      CLR          11$
6373 023620 005237 024000 12$: INC          11$          ;STALL ON POWER UP.
6374 023624 001375      BNE          12$          ;WAIT HERE IF BAD,WILL POWER OUT OF HERE.
6375              ;ELSE PROCEED.
6376
6377 023626      POPSP2          ;POP STACK TWICE2
6378 023630 013701 002516 MOV          KMCSR,R1
6379 023634 012637 000024      MOV          (SP)+,@#24
6380 023640 023737 002444 000024 CMP          $TMP0,@#24
6381 023646 001413      BEQ          4$          ;IS IT CORRECT?
6382 023650      ERROR          17          ;BR IF YES
(5) 023654 104455 TRAP          C$ERDF          ;ERROR, STACK IS INCORRECT
(6) 023656 000021      .WORD          17

```

(6)	023660	005055		.WORD	EM0	
(6)	023662	010256		.WORD	ERR17	
6383	023664	013737	002444	MOV	\$TMP0,@#24	:RESTORE TRUE POWER FAIL ADDRESS
6384	023672	013706	002344	MOV	PSTACK,SP	:RESTORE STACK
6385	023676	032711	004000	4\$: BIT	#BIT11,(R1)	:BIT11 STILL SET?
6386	023702	001016		BNE	7\$	
6387	023704	005737	002470	TST	RUNB	
6388	023710	001013		BNE	7\$	
6389	023712	011104		MOV	(R1),R4	
6390	023714	012705	004000	MOV	#BIT11,R5	
6391	023720			ERROR	35	:OAC FAILED
(5)	023724	104455		TRAP	C\$ERDF	
(6)	023726	000043		.WORD	35	
(6)	023730	005055		.WORD	EM0	
(6)	023732	010472		.WORD	ERR35	
6392						:TO PREVENT
6393						:INIT FROM
6394						:CLEARING CSR
6395	023734			EXIT	TST	
(3)	023734	104432		TRAP	C\$EXIT	
(3)	023736	000104		.WORD	L10102-	
6396	023740	012711	003000	7\$: MOV	#BIT9:BIT10,(R1)	:SEL6 = MAINT IR
6397	023744	012705	121111	MOV	#121111,R5	:R5 = EXPECTED
6398	023750	016104	000006	MOV	6(R1),R4	:R4 = FOUND
6399	023754	020504		CMP	R5,R4	:MAINT IR SHOULD = 12111
6400	023756	001431		BEQ	6\$	:BR IF OK
6401	023760			MSTCLR		
6402	023764			ERROR	35	:IF = 0 THEN BUS INIT WAS
(5)	023770	104455		TRAP	C\$ERDF	
(6)	023772	000043		.WORD	35	
(6)	023774	005055		.WORD	EM0	
(6)	023776	010472		.WORD	ERR35	
6403						:NOT BLOCKED FROM CLEARING
6404						:THE M8200,4,7
6405						
6406	024000	000000		11\$: .WORD	0	:TEMP COUNT FOR STALL ON POWER UP.
6407						
6408	024002	052711	040000	10\$: BIS	#BIT14,(R1)	:CLR THE THING SO IT CAN'T ASSIRT AC LOW
6409						:AGAIN!
6410	024006			MSTCLR		
6411	024012			ERROR	17	:ERROR GLIP GAVE US SECOUND UNEXPECTED
(5)	024016	104455		TRAP	C\$ERDF	
(6)	024020	000021		.WORD	17	
(6)	024022	005055		.WORD	EM0	
(6)	024024	010256		.WORD	ERR17	
6412						:ASSERTION OF AC LOW ON UNIBUS.
6413						:FATEL TYPE OF ERROR.
6414	024026	062706	000004	ADD	#4,SP	:RESTORE STACK.
6415	024032	012637	000024	MOV	(SP)+,@#24	
6416	024036			MSTCLR		
6417	024042			6\$: ENDTST		
6418	024042			L10102:		
(3)	024042			TRAP	C\$ETST	
(3)	024042	104401				
6419						
6420	024044			BADHEAD		

```
(2)
6421
6422
6423
6424
6425
6426
6427 024044
(2)
6428
6429 024044
(3) 024044
6430 024044
(1) 024044 013701 002516
6431 024050
6432 024054 005002
6433 024056 042737 000017 024104 1$:
6434 024064 156237 025100 024104
6435 024072 116261 025106 000004
6436 024100
( ) 024100 004537 003044
6437 024104 121100 2$:
6438 024106 005202
6439 024110 022702 000005
6440 024114 001360
6441 024116 005002
6442 024120 042737 000017 024166 3$:
6443 024126 042737 000017 024202
6444 024134 042737 000017 024214
6445 024142 050237 024166
6446 024146 050237 024202
6447 024152 050237 024214
6448 024156 105061 000004
6449 024162
(1) 024162 004537 003044
6450 024166 122100 4$:
6451 024170 112761 000377 000004
6452 024176
(1) 024176 004537 003044
6453 024202 122100 5$:
6454 024204 110261 000004
6455 024210
(1) 024210 004537 003044
6456 024214 122100 6$:
6457 024216 005202
6458 024220 022702 000010
6459 024224 001335
6460 024226 005002
6461 024230 042737 000017 024276 7$:
6462 024236 042737 000017 024312
6463 024244 042737 000017 024324
6464 024252 050237 024276
6465 024256 050237 024312
6466 024262 050237 024324
6467 024266 105061 000004
6468 024272
```

```
***** TEST 33 *****
:MICRO-PROCESSOR NOISE TEST
:WRITE ALL ZERO'S THEN ALL ONE'S THEN A DATA PATTERN
:TO THE IBUS* AND IBUS REGISTERS AND TO THE SP AND MAIN MEM
:THEN GO BACK AND READ THE DATA PATTERNS TO VERIFY THAT
:READING AND WRITING OF OTHER LOCATIONS AND REGISTERS
: DID NOT CHANGE THE DATA.
BADHEAD
***** TEST 33 *****

BGNTST
T33::

MYINT
MOV KMCSR,R1 ;RECORD DEVICE ADDR.
MSTCLR ;MASTER CLEAR M8200,4,7
CLR R2 ;R2 IS INDEX REGISTER
BIC #17,2$ ;CLEAR ADDRESS FIELD
BISB 30$(R2),2$ ;ADD IBUS* REG ADDRESS TO INSTRUCTION
MOVB 31$(R2),4(R1) ;LOAD PORT4
ROMCLK ;NEXT WORD IS INSTRUCTION.
JSR R5,ROMCLK ;CLOCK INSTRUCTION
121100 ;WRITE IBUS* REGISTER
INC R2 ;INC INDEX REGISTER
CMP #5,R2 ;DONE YET?
BNE 1$ ;BR IF NO
CLR R2 ;R2 IS IBUS REGISTER ADDRESS
BIC #17,4$ ;CLEAR ADDRESS FIELD OF INSTRUCTIONS
BIC #17,5$
BIC #17,6$
BIS R2,4$ ;ADD IBUS REG ADDRESS TO INSTRUCTION
BIS R2,5$
BIS R2,6$
CLRB 4(R1) ;CLEAR PORT4
ROMCLK ;NEXT WORD IS INSTRUCTION.
JSR R5,ROMCLK ;CLOCK INSTRUCTION
122100 ;WRITE 0 TO IBUS REG
MOVB #377,4(R1) ;LOAD PORT4
ROMCLK ;NEXT WORD IS INSTRUCTION.
JSR R5,ROMCLK ;CLOCK INSTRUCTION
122100 ;WRITE ALL ONES TO IBUS REG
MOVB R2,4(R1) ;LOAD PORT4
ROMCLK ;NEXT WORD IS INSTRUCTION.
JSR R5,ROMCLK ;CLOCK INSTRUCTION
122100 ;WRITE ITS OWN ADDRESS TO IBUS REG
INC R2 ;NEXT ADDRESS
CMP #10,R2 ;DONE YET?
BNE 3$ ;BR IF NO
CLR R2 ;START AT SP ADDRESS 0
BIC #17,8$ ;CLEAR ADDRESS FIELD
BIC #17,9$
BIC #17,10$
BIS R2,8$ ;ADD ADDRESS TO INSTRUCTION
BIS R2,9$
BIS R2,10$
CLRB 4(R1) ;CLEAR PORT4
ROMCLK ;NEXT WORD IS INSTRUCTION.
```

(1)	024272	004537	003044			JSR	R5,,ROMCLK	:CLOCK INSTRUCTION
6469	024276	123100			8\$:	123100		:WRITE ZERO TO SP
6470	024300	112761	000377	000004		MOVB	#377,4(R1)	:LOAD PORT4
6471	024306					ROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	024306	004537	003044			JSR	R5,,ROMCLK	:CLOCK INSTRUCTION
6472	024312	123100			9\$:	123100		:WRITE ALL ONES TO SP
6473	024314	110261	000004			MOVB	R2,4(R1)	:LOAD PORT4
6474	024320					ROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	024320	004537	003044			JSR	R5,,ROMCLK	:CLOCK INSTRUCTION
6475	024324	123100			10\$:	123100		:WRITE SP ADDRESS TO ITSELF
6476	024326	005202				INC	R2	:NEXT SP ADDRESS
6477	024330	022702	000020			CMP	#20,R2	:DONE YET?
6478	024334	001335				BNE	7\$	:BR IF NO
6479	024336	005002				CLR	R2	:R2 = ,AOM ,E, ADDRESS
6480	024340					ROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	024340	004537	003044			JSR	R5,,ROMCLK	:CLOCK INSTRUCTION
6481	024344	010000				010000		:MAR _ 0
6482	024346					ROMCLK		
(1)	024346	004537	003044			JSR	R5,,ROMCLK	:CLOCK INSTRUCTION
6483	024352	004000				4000		
6484	024354	105061	000004		11\$:	(LRB	4(R1)	:CLEAR PORT4
6485	024360					ROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	024360	004537	003044			JSR	R5,,ROMCLK	:CLOCK INSTRUCTION
6486	024364	122500				122500		:WRITE ZEROS TO MEM
6487	024366	112761	000377	000004		MOVB	#377,4(R1)	:LOAD PORT4
6488	024374					ROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	024374	004537	003044			JSR	R5,,ROMCLK	:CLOCK INSTRUCTION
6489	024400	122500				122500		:WRITE ONES TO MEM
6490	024402	110261	000004			MOVB	R2,4(R1)	:LOAD PORT4
6491	024406					ROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	024406	004537	003044			JSR	R5,,ROMCLK	:CLOCK INSTRUCTION
6492	024412	136500				136500		:WRITE TO MEM IT OWN ADDRESS
6493	024414	005202				INC	R2	:NEXT MEM ADDRESS
6494	024416	022702	001000			CMP	#1000,R2	:DONE YET?
6495	024422	001354				BNE	11\$	:BR IF NO
6496								
6497								
6498								
6499	024424					ROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	024424	004537	003044			JSR	R5,,ROMCLK	:CLOCK INSTRUCTION
6500	024430	010000				010000		:MAR 0
6501	024432					ROMCLK		:NEXT WORD IS INSTRUCTION,
(1)	024432	004537	003044			JSR	R5,,ROMCLK	:CLOCK INSTRUCTION
6502	024436	004000				4000		:MAR HI _ 0 (M8200,4,7 ONLY)
6503								
6504	024440	005737	002432			:WOULD BE (RAM CODE		
6505	024444	001452				ST	TYPE	
6506	024446	005005				BEQ	40\$	
6507	024450	042737	000360	024512	12\$:	CLR	R5	:R5 IS INDEX REGISTER
6508	024456	116502	025100			BIC	#360,13\$	:CLEAR ADDRESS FIELD
6509	024462	010203				MOVB	30\$(R5),R2	:R2 = IBUS* ADDRESS
6510	024464	006303				MOV	R2,R3	:PUT IBUS* ADDRESS IN R3
6511	024466	006303				ASL	R3	:SHIFT ADDRESS TO BITS 4-7
6512	024470	006303				ASL	R3	
6513	024472	006303				ASL	R3	
6514	024474	050337	024512			BIS	R3,13\$	:ADD ADDRESS TO INSTRUCTION

6515	024500	116537	025106	002452	MOV	318(R5),SGDDAT	:SGDDAT = 'EXPECTED'
6516	024506				ROMCLK		:NEXT WORD IS INSTRUCTION.
(1)	024506	004537	003044		JSR	R5,,ROMCLK	:CLOCK INSTRUCTION
6517	024512	121004			138:	121004	:PORT4 = IBUS* REGISTER
6518	024514	016104	000004		MOV	4(R1),R4	:R4 = 'FOUND'
6519	024520	123704	002452		CMPIB	SGDDAT,R4	:IBUS* CONTENTS OK?
6520	024524	001416			BEO	208	:BR IF YES
6521	024526	010237	002434		MOV	R2,MRO	
6522	024532	105037	002453		CLRB	SGDDAT+1	
6523	024536	013705	002452		MOV	SGDDAT,R5	
6524	024542				ERROR	29	:IBUS* DATA ERROR
(5)	024546	104455			TRAP	C\$ERDF	
(6)	024550	000035			.WORD	29	
(6)	024552	005055			.WORD	EMO	
(6)	024554	010350			.WORD	ERR29	
6525	024556				ESCAPE	TST	
(3)	024556	104410			TRAP	C\$ESCAPE	
(3)	024560	000334			.WORD	L10103-	
6526	024562	005205			208:	INC	:INC COUNTER
6527	024564	022705	000005		CMPI	#5,R5	:DONE YET?
6528	024570	001327			BNE	128	:BR IF NO
6529							
6530	024572				408:		
6531					:END CRAM,GENERAL TESTS		
6532							
6533	024572	005002			CLR	R2	:R2 = IBUS REG ADDRESS
6534	024574	042737	000360	024630	148:	BIC	#360,158
6535	024602	010203			MOV	R2,R3	:CLEAR ADDRESS FIELD OF INSTRUCTION
6536	024604	006303			ASL	R3	:R3 = IBUS ADDRESS
6537	024606	006303			ASL	R3	:SHIFT ADDRESS TO BITS 4-7
6538	024610	006303			ASL	R3	
6539	024612	006303			ASL	R3	
6540	024614	050337	024630		BIS	R3,158	:ADD ADDRESS TO INSTRUCTION
6541	024620	010237	002452		MOV	R2,SGDDAT	:SGDDAT = 'EXPECTED'
6542	024624				ROMCLK		:NEXT WORD IS INSTRUCTION.
(1)	024624	004537	003044		JSR	R5,,ROMCLK	:CLOCK INSTRUCTION
6543	024630	021004			158:	021004	:PORT4 = IBUS REG
6544	024632	016104	000004		MOV	4(R1),R4	:IBUS = 'FOUND'
6545	024636	123704	002452		CMPIB	SGDDAT,R4	:IBUS CONTENTS OK?
6546	024642	001410			BEO	218	:BR IF YES
6547	024644	013705	002452		MOV	SGDDAT,R5	
6548	024650				ERROR	29	:IBUS DATA ERROR
(5)	024654	104455			TRAP	C\$ERDF	
(6)	024656	000035			.WORD	29	
(6)	024660	005055			.WORD	EMO	
(6)	024662	010350			.WORD	ERR29	
6549	024664	005202			218:	INC	:NEXT IBUS REGISTER
6550	024666	022702	000010		CMPI	#10,R2	:DONE YET?
6551	024672	001340			BNE	148	:BR IF NO
6552	024674	005002			CLR	R2	:R2 = SP ADDRESS
6553	024676	042737	000017	024714	168:	BIC	#17,178
6554	024704	050237	024714		BIS	R2,178	:CLEAR ADDRESS FIELD OF INSTRUCTION
6555	024710				ROMCLK		:ADD ADDRESS TO INSTRUCTION
(1)	024710	004537	003044		JSR	R5,,ROMCLK	:NEXT WORD IS INSTRUCTION.
6556	024714	040600			178:	040600	:CLOCK INSTRUCTION
6557	024716	010237	002452		MOV	R2,SGDDAT	:BR - SP
							:SGDDAT = 'EXPECTED'


```

6592
6593      025114      .EVEN
6594
6595      025114      ENDTST
6596      (3) 025114      L10103:
6597      (3) 025114      104401      TRAP      CSETST
6598
6599      025116      BADHEAD
6600      (2)
6601      :***** TEST 34 *****
6602      : THIS TEST IS DESIGNED TO MAKE SURE THAT A NODST INSTRUCTION
6603      : DOES NOT WRITE INTO PORT B OF THE MULTI PORT RAM.
6604      : TO DO THIS, WE'LL PUT A 125 INTO INDAT2, THEN WE'LL PUT A
6605      : 125 INTO BOTH SP1 AND BR. LAST WE'LL DO A NODST BR, SUBOC, SP1
6606      : IF THERE IS A WRITE INTO PORTB, INDAT2 WILL CONTAIN A 377.
6607      BADHEAD
6608      :***** TEST 34 *****
6609
6610      025116      BGN TST
6611      (3) 025116      T34::
6612      (1) 025116      013701      002516      MOV NT
6613      025122      004537      003044      MOV      KPCR, R1      ; RECORD DEVICE ADDR.
6614      025122      000525      JSR      R5, .ROMCLK      ; CLOCK INSTRUCTION
6615      025130      004537      003044      00525      ; PUT A 125 INTO BRG.
6616      025130      004537      003044      ROMCLK      JSR      R5, .ROMCLK      ; CLOCK INSTRUCTION
6617      025134      062221      062221      ; NOW INTO OI DAT2
6618      025136      004537      003044      ROMCLK      JSR      R5, .ROMCLK      ; CLOCK INSTRUCTION
6619      025142      063221      63221      ; NOW INTO SP1
6620      025144      004537      003044      ROMCLK      JSR      R5, .ROMCLK      ; CLOCK INSTRUCTION
6621      025150      060361      060361      ; NOW THE 'NODST BR, SUBOC, SP1'
6622      025152      004537      003044      JSR      R5, .ROMCLK      ; THE NODST SHOULD NOT MODIFY INDAT2:
6623      025156      020420      020420      ; CLOCK INSTRUCTION
6624      025160      004537      003044      JSR      R5, .ROMCLK      ; PUT CONTENT OF INDAT2 IN BRG.
6625      025164      061220      061220      ; CLOCK INSTRUCTION
6626      025166      111104      111104      ; PUT BRG INTO BSEL0
6627      025170      012705      000125      MOV      (R1), R4      ; SEE WHAT CAME BACK.
6628      025174      020405      020405      MOV      #125, R5      ; SHOULD BE 125 IF 377 CAME BACK.
6629      025176      001406      001406      ; YOU CAN BET THAT THE 'NODST' WROTE
6630      025200      104455      104455      ; INTO THE MULTI PORT RAM! WATCH SIGNAL
6631      025204      000007      000007      ; 'D1 WRITE OUT L'
6632      025210      005055      005055      CMP      R4, R5      ; NOW LOOK.
6633      025210      005055      005055      BEQ      10$
6634      025210      005055      005055      ERROR      7
6635      025210      005055      005055      TRAP      CSE RDF
6636      025210      005055      005055      .WORD      7
6637      025210      005055      005055      .WORD      EMO

```

(6)	025212	007026		.WORD	ERR7	
6633						
6634	025214			10\$:		
6635	025214				ENDTST	
(3)	025214			L10104:		
(3)	025214	104401			TRAP C\$ETST	
6636						
6637	025216				BADHEAD	
(2)					;***** TEST 35 *****	
6638					;*	
6639					;* EXTENDED CRAM TEST FOR M8206. IN THIS TEST WE WILL LOAD DATA	
6640					;* THROUGHOUT THE CRAM (TEST DATA IS JUST 4K OF DIAG. CODE) AND	
6641					;* THEN READ IT BACK AND VERIFY THAT IT IS CORRECT.	
6642	025216				BADHEAD	
(2)					;***** TEST 35 *****	
6643						
6644	025216				BGNTST	
(3)	025216			135::		
6645	025216				SKIP06 10\$	;DO TEST ONLY IF IT IS A M8206
(1)					;GOTO 10\$ IF M8206	
6646	025226				EXIT TST	;OTHERWISE,SKIP TEST.
(3)	025226	104432			TRAP C\$EXIT	
(3)	025230	000132			.WORD L10105-	
6647						
6648	025232			10\$:	MYINT	
(1)	025232	013701	002516		MOV KMCSR,R1	;RECORD DEVICE ADDR.
6649						
6650	025236	012702	012146		MOV #ROMMAP,R2	;GET ADDR. OF LIST.
6651						
6652	025242	012711	002000		MOV #2000,(R1)	;SET TO WRITE DATA.
6653	025246	005003			CLR R3	;CRAM ADDR ZERO.
6654						
6655	025250	010361	000004	15\$:	MOV R3,4(R1)	;SET ADDR.
6656	025254	012261	000006		MOV (R2)+,6(R1)	;WRITE DATA.
6657						
6658	025260	020337	002436		CMP R3,MEMSZ	;DONE WHOLE CRAM?
6659	025264	001402			BEQ 20\$	;YES,EXIT THIS LOOP.
6660	025266	005203			INC R3	;NO,UPDAT ADDR.
6661	025270	000767			BR 15\$	
6662	025272	005003		20\$:	CLR R3	;NOW WE WILL READ BACK,STARTING AT
6663						;CRAM ADDR. ZERO.
6664	025274	012705	012146		MOV #ROMMAP,R5	;GET ADDR. LIST OF DATA
6665						
6666	025300	010361	000004	30\$:	MOV R3,4(R1)	;SET ADDR.
6667						
6668	025304	011502			MOV (R5),R2	;PUT EXPECTED INTO R2
6669	025306	016104	000006		MOV 6(R1),R4	;READ ACCUAL
6670	025312	020204			CMP R2,R4	;EQUAL?
6671	025314	001411			BEQ 40\$	;YES,CONTINUE.
6672	025316	010300			MOV R3,R0	
6673						
6674	025320				ERROR 1	;ERROR CRAM DATA TEST,DATA
(5)	025324	104455			TRAP C\$ERDF	
(6)	025326	000001			.WORD 1	
(6)	025330	005055			.WORD EMO	
(6)	025332	006032			.WORD ERR1	

```

6675                                     ;READ NOT DATA THAT WAS WRITTEN.
6676
6677 025334      ESCAPE TST
(3) 025334      104410      TRAP      C$ESCAPE
(3) 025336      000024      .WORD      L10105-
6678 025340      020337      002436      40$:      CMP      R3, MEMS2
6679 025344      001002      BNE      50$      ;ALL DONE?
6680
6681 025346      EXIT      TST
(3) 025346      104432      TRAP      C$EXIT
(3) 025350      000012      .WORD      L10105-
6682
6683 025352      005203      50$:      INC      R3      ;UPDATE ADDR.
6684 025354      062705      000002      ADD      #2, R5
6685 025360      000747      BR      30$
6686
6687 025362      ENDTST
(3) 025362      L10105:      TRAP      C$ETST
(3) 025362      104401
6688
6689
6690 025364      BADHEAD
(2)      ;***** TEST 36 *****
6691      ;*
6692      ;* THIS TEST LOADS MICRO-CODE INTO A M8206 MCPU THEN EXECTUES IT.
6693      ;* THE MICRO CODE IS DESIGNED TO WRITE ALL ONES INTO THE SEL REGS.
6694      ;* THIS TEST IS ONLY PERFORMED ON AN M8206.
6695 025364      BADHEAD
(2)      ;***** TEST 36 *****
6696
6697 025364      BGNTST
(3) 025364      T36::
6698
6699 025364      SKIP06 1$      ;ONLY DO THIS TEST IF M8206
(1)      ;GOTO 1$ IF M8206
6700 025374      EXIT      TST
(3) 025374      104432      TRAP      C$EXIT
(3) 025376      000442      .WORD      L10106-
6701
6702 025400      1$:      MYINT
(1) 025400      013701      002516      MOV      KMCSR, R1      ;RECORD DEVICE ADDR.
6703
6704 025404      004537      026006      JSR      R5, LOADER      ;LOAD THE MICRO CODE
6705
6706 025410      000777      777      ;MOVE #377, BRG
6707 025412      061220      061220      ;MOVE BRG, BSEL0
6708 025414      061222      061222      ;MOVE BRG, BSEL2
6709 025416      061223      061223      ;MOVE BRG, BSEL3
6710 025420      061224      061224      ;MOVE BRG, BSEL4
6711 025422      061225      061225      ;MOVE BRG, BSEL5
6712 025424      061226      061226      ;MOVE BRG, BSEL6
6713 025426      061227      061227      ;MOVE BRG, BSEL7
6714 025430      123000      ;MOVE BSEL0, SP0
6715 025432      101410      ;BRANCH BACK ONE UNTIL <>377
6716
6717 025434      000400      400      ;MOVE #0, BRG
  
```

6718	025436	061220		61220		;MOVE BRG,BSEL0
6719	025440	061222		61222		;MOVE BRG,BSEL2
6720	025442	061223		61223		;MOVE BRG,BSEL3
6721	025444	061224		61224		;MOVE BRG,BSEL4
6722	025446	061225		61225		;MOVE BRG,BSEL5
6723	025450	061226		61226		;MOVE BRG,BSEL6
6724	025452	061227		61227		;MOVE BRG,BSEL7
6725	025454	123000		123000		;MOVE BSEL0,SP0
6726	025456	104022		104022		;BRANCH BACK ONE LOCATION.
6727	025460	177777		177777		
6728						
6729	025462	012711	040000	MOV	#040000,(R1)	;INITIALIZE MCPU
6730	025466	012711	100000	MOV	#100000,(R1)	;START CPU.
6731						
6732	025472	012700	000062	MOV	#50.,R0	;THE CYCLE TIME ON THE M8206 IS
6733						;200NS. WE ARE ASKING THE MCPU TO
6734						;DO 8 INSTRUCTIONS. WE'LL DELAY
6735						;100 PDP11 INSTRUCTIONS
6736						;THIS REALLY SHOULD BE PLENTY OF TIME.
6737						
6738	025476	005300		20\$:	DEC R0	
6739	025500	001376			BNE 20\$	
6740						
6741	025502	005005		CLR	R5	;JUST FOR TYPEOUT.
6742	025504	012705	000377	MOV	#377,R5	;EXPECT 377
6743	025510	111104		MOV	(R1),R4	;READ MCPU
6744	025512	120405		CMPS	R4,R5	;SEE IF OK.
6745	025514	001410		BEQ	30\$	
6746						
6747	025516			ERROR	29	;ERROR! MCPU WAS TO WRITE ALL
(5)	025522	104455		TRAP	C\$ERDF	
(6)	025524	000035		.WORD	29	
(6)	025526	005055		.WORD	EM0	
(6)	025530	010350		.WORD	ERR29	
6748						;ONES INTO BSEL0,BUT INSTEAD FAILED.
6749	025532			ESCAPE	TST	
(3)	025532	104410		TRAP	C\$ESCAPE	
(3)	025534	000304		.WORD	L10106-	
6750						
6751	025536	012705	177777	30\$:	MOV	#177777,R5
6752	025542	016104	000002		MOV	2(R1),R4
6753	025546	020405			CMP	R4,R5
6754	025550	001410			BEQ	40\$
6755						
6756	025552			ERROR	29	;ERROR! MCPU WAS TO WRITE ALL ONES
(5)	025556	104455		TRAP	C\$ERDF	
(6)	025560	000035		.WORD	29	
(6)	025562	005055		.WORD	EM0	
(6)	025564	010350		.WORD	ERR29	
6757						;INTO BSEL 2&3
6758						
6759	025566			ESCAPE	TST	
(3)	025566	104410		TRAP	C\$ESCAPE	
(3)	025570	000250		.WORD	L10106-	
6760						
6761	025572	016104	000004	40\$:	MOV	4(R1),R4
						;READ BSEL 4&5

6762	025576	020405			CMP	R4,R5		;READ OK?
6763	025600	001410			BEQ	50\$		
6764								
6765	025602				ERROR	29		;ERROR! FAILED TO WRITE BSEL \$85
(5)	025606	104455			TRAP	C\$ERDF		
(6)	025610	000035			.WORD	29		
(6)	025612	005055			.WORD	EM0		
(6)	025614	010350			.WORD	ERR29		
6766								; TO ALL ONES.
6767	025616				ESCAPE	TST		
(3)	025616	104410			TRAP	C\$ESCAPE		
(3)	025620	000220			.WORD	L10106-		
6768								
6769	025622	016104	000006	50\$:	MOV	6(R1),R4		;READ BSEL 6&7
6770	025626	020405			CMP	R4,R5		;READ OK?
6771	025630	001410			BEQ	60\$		
6772								
6773	025632				ERROR	29		;ERROR! FAILED TO WRITE BSEL 6&7
(5)	025636	104455			TRAP	C\$ERDF		
(6)	025640	000035			.WORD	29		
(6)	025642	005055			.WORD	EM0		
(6)	025644	010350			.WORD	ERR29		
6774								; TO ALL ONES.
6775	025646				ESCAPE	TST		
(3)	025646	104410			TRAP	C\$ESCAPE		
(3)	025650	000170			.WORD	L10106-		
6776	025652	105011		60\$:	CLRB	(R1)		;SIGNAL MCPU TO WRITE ALL ZEROS.
6777	025654	005005			CLR	R5		;EXPECT TO READ ALL ZEROS.
6778								
6779	025656	005004			CLR	R4		
6780	025660	111104			MOVB	(R1),R4		;READ BSELO
6781	025662	001410			BEQ	70\$		;EXPECT ZERO.
6782								
6783	025664				ERROR	29		;MCPU FAILED TO CLEAR BSELO
(5)	025670	104455			TRAP	C\$ERDF		
(6)	025672	000035			.WORD	29		
(6)	025674	005055			.WORD	EM0		
(6)	025676	010350			.WORD	ERR29		
6784								
6785	025700				ESCAPE	TST		
(3)	025700	104410			TRAP	C\$ESCAPE		
(3)	025702	000136			.WORD	L10106-		
6786	025704	016104	000002	70\$:	MOV	2(R1),R4		;READ BSEL 2&3
6787	025710	001410			BEQ	80\$		;IF ZERO,OK
6788								
6789	025712				ERROR	29		;MCPU FAILED TO CLEAR BSEL 2&3
(5)	025716	104455			TRAP	C\$ERDF		
(6)	025720	000035			.WORD	29		
(6)	025722	005055			.WORD	EM0		
(6)	025724	010350			.WORD	ERR29		
6790	025726				ESCAPE	TST		
(3)	025726	104410			TRAP	C\$ESCAPE		
(3)	025730	000110			.WORD	L10106-		
6791	025732			80\$:				
6792	025732	016104	000004		MOV	4(R1),R4		;READ BSEL 4&5
6793	025736	001410			BEQ	90\$		

```
6794
6795 025740          ERROR 29          ;MCPU FAILED TO CLEAR BSEL 4&5
(5) 025744 104455   TRAP  C$ERDF
(6) 025746 000035   .WORD 29
(6) 025750 005055   .WORD EM0
(6) 025752 010350   .WORD ERR29
6796 025754          ESCAPE TST
(3) 025754 104410   TRAP  C$ESCAPE
(3) 025756 000062   .WORD L10106-.
6797 025760
6798 025760 016104 000006   90$: MOV 6(R1),R4          ;READ BSEL 6&7
6799 025764 001406        BEQ 95$
6800
6801 025766          ERROR 29          ;MCPU FAILED TO CLEAR BSEL 6&7
(5) 025772 104455   TRAP  C$ERDF
(6) 025774 000035   .WORD 29
(6) 025776 005055   .WORD EM0
(6) 026000 010350   .WORD ERR29
6802
6803 026002          95$: EXIT TST
(3) 026002 104432   TRAP  C$EXIT
(3) 026004 000034   .WORD L10106-.
6804
6805
6806          ;
6807          ;LOADER  SUBROUTINE USED BY THIS TEST TO LOAD MICRO CODE INTO A M8206
6808          ;
6809
6810 026006 012711 002000   LOADER: MOV #2000,(R1)
6811
6812 026012 005000        CLR R0
6813
6814 026014 010061 000004   10$: MOV R0,4(R1)          ;SET ADDR.
6815 026020 005200        INC R0
6816 026022 011561 000006   MOV (R5),6(R1)          ;WRITE MICRO CODE.
6817 026026 022527 177777   CMP (R5)+,#177777    ;SEE IF TERM.
6818 026032 001370        BNE 10$
6819 026034 005011        CLR (R1)
6820 026036 000205        RTS R5
6821
6822 026040          ENDTST
(3) 026040          L10106: TRAP C$ETST
(3) 026040 104401
6823
6824 026042          BADHEAD
(2)          ;***** TEST 37 *****
6825          ;*
6826          ;*NEGATIVE ADDRESS TEST.
6827          ;*   IN THIS TEST, WE'LL MAKE SURE THAT THE M8207
6828          ;*   DOES NOT RESPOND TO AN ADDRESS THAT ISN'T ASSIGNED
6829          ;*   TO IT
6830          ;*
6831 026042          BADHEAD
(2)          ;***** TEST 37 *****
6832
6833 026042          BGNTEST
```

```
(3) 026042
6834 026042
(1) 026042 013701 002516
6835
6836 026046 012711 000641
6837 026052 012737 026130 000004
6838 026060 005037 000006
6839 026064 012702 160000
6840
6841 026070 022712 000641 10$: CMP #641,(R2) ;SEE IF CONTENTS OF THE ADDRESS
6842 ;POINTED TO BY R2 EQUALS THE CONTENTS
6843 ;OF THE MCPU CSR
6844 026074 001420 BEQ 40$
6845
6846 026076 062702 000002 15$: ADD #2,R2 ;UPDATE ADDRESS.
6847 026102 020227 177700 CMP R2,#177700 ;DONE? ;BC
6848 026106 001370 BNE 10$ ;NO-LOOP
6849
6850 026110 013737 002464 000004 17$: MOV SAVE4,#4 ;RESTORE TRAP CATCHER
6851 026116 013737 002466 000006 MOV SAVE6,#6 ;FROM VALUES SAVED BY INIT SECTION
6852 026124 EXIT TST ;EXIT, ALL DONE
(3) 026124 104432 TRAP C$EXIT
(3) 026126 000052 .WORD L10107-
6853
6854 026130 062706 000004 20$: ADD #4,SP ;SAVE FROM TRAP
6855 026134 000760 BR 15$ ;LOOP
6856
6857 026136 40$: ;*OH NO, WE MAY HAVE A DUAL ADDRESS PROBLEM.
6858
6859 026136 012711 000174 MOV #174,(R1) ;WRITE NEW PATTERN IN MCPU CSR
6860 026142 022712 000174 CMP #174,(R2) ;DID NEW PATTERN SHOW UP IN ADDR?
6861 026146 001403 BEQ 60$
6862
6863 026150 012711 000641 50$: MOV #641,(R1) ;PUT OLD PATTERN BACK IN MCPU CSR.
6864 026154 000750 BR 15$ ;LOOP
6865
6866 026156 020102 60$: CMP R1,R2 ;IS THIS THE MCPU ADDRESS?
6867 026160 001773 BEQ 50$ ;YES-NO ERROR
6868
6869 026162 ERROR 40 ;DUAL ADDRESS ERROR
(5) 026166 104455 TRAP C$ERDF
(6) 026170 000050 .WORD 40
(6) 026172 005055 .WORD EMO
(6) 026174 010636 .WORD ERR40
6870 026176 000744 BR 17$
6871
6872 026200
(3) 026200
(3) 026200 104401 L10107: ENDTST
6873 TRAP C$ETST
6874 026202
(2) BADHEAD
6875 ;***** TEST 38 *****
6876 ;*
6877 ;*BYTE ADDRESSING TEST
6878 ;* HERE, WE'RE GOING TO MAKE SURE THAT WE CAN
;* WRITE INTO ONLY A HIGH OR LOW BYTE OF THE MCPU.
```

```
6879
6880 026202
(2)
6881
6882 026202
(3) 026202
6883 026202
(1) 026202 013701 002516
6884 026206 005061 000002
6885 026212 112761 177777 000002
6886
6887 026220 032761 177400 000002
6888 026226 001410
6889
6890 026230
(5) 026234 104455
(6) 026236 000051
(6) 026240 005055
(6) 026242 010702
6891 026244
(3) 026244 104410
(3) 026246 000040
6892
6893 026250 005061 000002
6894 026254 112761 177777 000003
6895 026262 032761 000377 000002
6896 026270 001406
6897
6898 026272
(5) 026276 104455
(6) 026300 000052
(6) 026302 005055
(6) 026304 010744
6899
6900
6901 026306
6902 026306
(3) 026306
(3) 026306 104401
6903
6904 026310
(2)
6905
6906
6907
6908
6909
6910 026310
(2)
6911
6912 026310
(3) 026310
6913 026310
(1)
6914 026320
(3) 026320 104432

; *
; BADHEAD
; ***** TEST 38 *****
T38:: BGNTST
MYINT
MOV KMCSR,R1 ;RECORD DEVICE ADDR.
CLR 2(R1) ;CLEAR CSR
MOVB #-1,2(R1) ;WRITE ALL ONES INTO LOW BYTE
;OF CSR
BIT #177400,2(R1) ;SEE IF HIGH BYTE GOT WRITTEN
BEQ 10$
ERROR 41 ;HIGH BYTE GOT WRITTEN INTO ON A LOW BYTE
TRAP C$ERDF
.WORD 41
.WORD EMO
.WORD ERR41
ESCAPE TST ;OPERATION
TRAP C$ESCAPE
.WORD L10110-.
10$: CLR 2(R1)
MOVB #-1,3(R1) ;WRITE INTO HIGH BYTE
BIT #377,2(R1) ;SEE IF LOW BYTE GOT WRITTEN
BEQ 20$
ERROR 42 ;LOW BYTE GOT WRITTEN INTO ON A
TRAP C$ERDF
.WORD 42
.WORD EMO
.WORD ERR42
;HIGH BYTE OPERATION.
20$:
L10110: ENDIST
TRAP C$ETST
BADHEAD
; ***** TEST 39 *****
; *
; *IN THIS TEST WE'RE GOING TO MAKE SURE THAT THE PC
; *REG COUNTS UP PROPERLY. THE PC REG SHOULD INCREMENT
; *ONCE AFTER EACH INSTRUCTION.
; *
; BADHEAD
; ***** TEST 39 *****
T39:: BGNTST
SKIP07 10$ ;ONLY DO IF M8207
;GOTO 10$ IF M8207
EXIT TST
TRAP C$EXIT
```

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 PAGE 36-8
HARDWARE TESTS

SEQ 0159

```

(3) 026322 000122 .WORD L10111-.
6915
6916 026324 10$: MYINT
(1) 026324 013701 002516 MOV KMCSR,R1 ;RECORD DEVICE ADDR.
6917 026330 MSTCLR
6918 026334 ROMCLK
(1) 026334 004537 003044 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
6919 026340 000400 400
6920 026342 ROMCLK
(1) 026342 004537 003044 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
6921 026346 061233 61233
6922 026350 SROMCLK
(1) 026350 004537 003100 JSR R5,.SROMCLK
6923 026354 100000 100000
6924 026356 012705 000001 MOV #1,R5 ;START AT ZERO
6925
6926 026362 20$: ROMCLK ;READ PC HIGH REG.
(1) 026362 004537 003044 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
6927 026366 121265 121265
6928
6929 026370 ROMCLK ;READ PC LOW REG.
(1) 026370 004537 003044 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
6930 026374 121244 121244
6931 026376 016104 000004 MOV 4(R1),R4 ;GET WHOLE PICTURE
6932 026402 042704 170000 BIC #170000,R4
6933 026406 020405 CMP R4,R5 ;INCREMENT OK?
6934 026410 001410 BEQ 30$
6935
6936 026412 ERROR 47 ;PC FAILED TO INCREMENT PROPERLY
(5) 026416 104455 TRAP C$ERDF
(6) 026420 000057 .WORD 47
(6) 026422 005055 .WORD EMO
(6) 026424 011264 .WORD ERR47
6937
6938 ;SHOULD INCREMENT BY ONE
6939 026426 ;FOR EACH INSTRUCTION.
(3) 026426 104410 ESCAPE TST
(3) 026430 000014 TRAP C$ESCAPE
6940 .WORD L10111-.
6941 026432 062705 000002 30$: ADD #2,R5 ;UPDATE EXPECTED ADDRESS BY 2.
6942 026436 020527 000777 CMP R5,#777
6943 026442 001347 BNE 20$
6944
6945 026444 ENDTST
(3) 026444 L10111: TRAP C$ETST
(3) 026444 104401
6946
6947 026446 BADHEAD
(2) ;***** TEST 40 *****
6948 ;*
6949 ;*IN THIS TEST WE'LL MAKE SURE THAT 'BRANCH FIELD H' DOESN'T
6950 ;*GET SUCH HIGH.
6951 ;*FIRST WE'LL CLEAR THE PC HIGH REG. THEN WE'LL DO A BRANCH INSTR
6952 ;*WITH BAB BITS 11&12 SET. IF PCR BITS 8&9 SET THEN WE'LL KNOW
6953 ;*WE WERE SUCCESSFUL IF PCR BITS 8&9 FAIL TO SET, WE'LL KNOW
6954 ;*THAT THE MUX SELECTED THE WRONG INPUT TO BE CLOCKED INTO THE PCR.

```

```

6955
6956 026446          : *
                        : BADHEAD
                        : ***** TEST 40 *****
6957 (2)
6958 026446          : BGN1ST
6959 (3) 026446          :
6960 026446          : T40::
6961 (1) 026446          : SKIP07 10$          ; ONLY DO IF M8207
6962 (3) 026446 104432   : ;GOTO 10$ IF M8207
6963 (3) 026460 000062   : EXIT TST
6964 026460 000062   : TRAP C$EXIT
6965 026462          : .WORD L10112-.
6966 (1) 026462 013701 002516 : 10$: MYINT
6967 (1) 026462 013701 002516 : MOV KMCSR,R1
6968 026466          : MSTCLR
6969 026472          : ; INITIALIZE PARAMETERS
6970 (1) 026472 004537 003044 : ; RECORD DEVICE ADDR.
6971 026476 114400     : ; CLEAR DEVICE.
6972 026476 114400     : ROMCLK
6973 (1) 026476 114400     : JSR R5,.ROMCLK
6974 026476 114400     : ; DO A 'BRANCH ALWAYS' WITH
6975 026500          : ; CLOCK INSTRUCTION
6976 (1) 026500 004537 003044 : ; BAB BITS 11&12 SET THIS SHOULD CLOCK
6977 026504 121265     : ; THESE BITS INTO BITS 8&9 OF THE PCR.
6978 026504 121265     : ROMCLK
6979 (1) 026504 121265     : JSR R5,.ROMCLK
6980 026506 116105 000005 : ; NOW READ THE PCR HIGH
6981 026512 112704 000003 : ; CLOCK INSTRUCTION
6982 026516 042705 000374 : ; AND PUT INTO PORT5.
6983 026522 020405     : ; REG. BR NO CLK OF BAB BITS
6984 026524 001406     : ; READ THE PCR.
6985 026526          : MOVB 5(R1),R5
6986 (5) 026532 104455     : MOVB #3,R4
6987 (6) 026534 000017     : BIC #374,R5
6988 (6) 026536 005055     : CMP R4,R5
6989 (6) 026540 010006     : BEQ 20$
6990 026542          : ; EXPECT BITS 8,9 TO BE SET.
6991 026542 104401     : ERROR 15
6992 026544          : TRAP C$ERDF
6993 (2) 026544          : .WORD 15
6994 026544          : .WORD EMO
6995 026544          : .WORD ERR15
6996 026544          : ; 'BRANCH FIELD H' STUCK HIGH OR
6997 026544          : ; OTHER PROBLEM IN THIS AREA.
6998 026542          : 20$:
6999 (3) 026542 104401     : L10112: ENDTST
7000 026544          : TRAP C$ETST
7001 026544          : BADHEAD
7002 (2) 026544          : ***** TEST 41 *****
7003 026544          : *
7004 026544          : ; IN THIS TEST WE'RE GOING TO MAKE SURE THAT ONLY SPO
7005 026544          : ; IS SELECTED FOR SOURCE WHEN THE DESTINATION
7006 026544          : ; IS THE OUTBUS
7007 026544          : ; FIRST WE'LL WRITE EACH SP ADDRS INTO ITSELF THEN WE'LL
7008 026544          : ; MOV SP TO OBUS4. THAT SHOULD SELECT
7009 026544          : ; SP ADDRESS 0. IF ANY OTHER DATA SHOWS UP, WE'LL
7010 026544          : ; BLAME IT ON THE SELECTION OF A DIFFERENT SCRATCH PAD.
7011 026544          : BADHEAD
7012 026544          : ***** TEST 41 *****
7013 026544          : BGN1ST

```

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 PAGE 36-10
HARDWARE TESTS

SEQ 0161

```

(3) 026544
6995 026544
(1) 026544 013701 002516
6996 026550 005005
6997
6998 026552 042737 000017 026574 10$: BIC #17,20$ ;STRIP SP ADDR FIELD FROM INSTR
6999 026560 010561 000004 MOV R5,4(R1) ;PUT SP ADDR INTO PORT4.
7000 026564 050537 026574 BIS R5,20$ ;ADD SP ADDR TO INSTR.
7001 026570
(1) 026570 004537 003044 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
7002 026574 123100 20$: 123100 ;WRITE TO SP
7003 026576 005205 INC R5 ;UPDATE ADDRESS
7004 026600 120527 000020 CMPB R5,#20 ;IF NOT THROUGH, REPEAT.
7005 026604 001362 BNE 10$
7006
7007 026606 ROMCLK ;NOW MOV SPO TO OBUS* PORT4
(1) 026606 004537 003044 JSR R5,.ROMCLK ;CLOCK INSTRUCTION
7008 026612 061204 061204 ;
7009 026614 116104 000004 MOVB 4(R1),R4 ;READ PORT4 IT S/B ZERO
7010 026620 001410 BEQ 30$
7011 026622 012705 000000 MCV #0,R5
7012 026626 ERROR 43 ;SPO NOT SELECTED FOR SOURCE-SEE
(5) 026632 104455 TRAP C$ERDF
(6) 026634 000053 .WORD 43
(6) 026636 005055 .WORD EMO
(6) 026640 011006 .WORD ERR43
7013 ;DISCUSSION IN HEADER.
7014
7015 026642 30$: ENDTST
(3) 026642 L10113: TRAP C$ETST
(3) 026642 104401
7016
7017 026644 BADHEAD
(2) ;***** TEST 42 *****
7018 ;*
7019 ;*IN THIS TEST WE ARE GOING TO MAKE SURE THAT THE
7020 ;*SIGNAL 'MOV INST H' (AND ITS ASSOC. TRIBS) DOESN'T GET
7021 ;*STUCK HIGH. IN ORDER TO DO THIS WE'LL CLEAR THE PC HIGH REG
7022 ;*PUT KNOWN DATA IN THE BREG AND SP1 THEN WE'LL A BRANCH
7023 ;*WITH CROM BITS 0-3 SET AS WELL AS CROM BIT 9 WITH CROM BITS 8 AND 11 CLEAR.
7024 ;*IF 'MOV INST H' GETS STUCK HIGH, THE PC REG HIGH WILL GET LOADED
7025 ;*WITH THE CONTENTS OF THE ALU
7026 026644 BADHEAD
(2) ;***** TEST 42 *****
7027
7028 026644 BGNTST
(3) 026644
7029 026644 r42:: SKIP07 10$ ;ONLY DO IF M8207
(1) ;GOTO 10$ IF M8207
7030 026654 104432 EXIT TST ;ELSE EXIT
(3) 026654 000110 TRAP C$EXIT
(3) 026656 .WORD L10114-.
7031
7032 026660 10$: MYINT ;DO INITIAL TEST SET-UP.
(1) 026660 013701 002516 MOV KMCSR,R1 ;RECORD DEVICE ADDR.
7033 026664 MSTCLR ;DO A MASTER CLEAR.

```

CZDMQCO M8207 STATIC DIAG #2
CZDMQCO.P11 21-JUL-81 14:36

MAC(Y11 30A(1052) 21-JUL-81 14:48 PAGE 36-11
HARDWARE TESTS

SEQ 0162

7034	026670	005737	002470	TST	RUNB	
7035	026674	001034		BNE	208	
7036						
7037				;TO RUN THIS SECTION OF CODE YOU MUST TURN SW7 OF SWITCH PACK #E28		
7038				;OFF SO THAT M8207 NOT SELFSTARTING.		
7039						
7040	026676	012761	000002 000004	MOV	#2,4(R1)	;PUT A 2 INTO SP1
7041	026704			ROMCLK		;PORT4 TO SCRATCH PAD 1
(1)	026704	004537 003044		JSR	R5,.ROMCLK	;CLOCK INSTRUCTION
7042	026710	123101		123101		
7043	026712	012761 000004 000004		MOV	#4,4(R1)	
7044	026720			ROMCLK		
(1)	026720	004537 003044		JSR	R5,.ROMCLK	;CLOCK INSTRUCTION
7045	026724	123100		123100		
7046	026726			ROMCLK		;NOW DO A BRANCH ON C-BIT SET
(1)	026726	004537 003044		JSR	R5,.ROMCLK	;CLOCK INSTRUCTION
7047	026732	141201		141201		;BASED ON SP CONTENTS
7048						;OK-WHAT WE ARE REALLY
7049						;INTERESTED IN IS SEEING IF THE
7050						;PC HIGH REG GETS LOADED WITH
7051						;THE CONTENTS OF THE ALU (2)
7052						;IF THIS OCCURS, WE CAN PROBABLY
7053						;SAY THAT 'MOV INSTR' REMAINED
7054						;HIGH.
7055	026734			ROMCLK		;READ PC HIGH, PUT INTO PORT5
(1)	026734	004537 003044		JSR	R5,.ROMCLK	;CLOCK INSTRUCTION
7056	026740	121265		121265		
7057	026742	116104 000005		MOVB	5(R1),R4	;READ PC REG HIGH FROM PORT
7058	026746	001407		BEQ	208	;SHOULD BE CLEAR
7059	026750	005005		CLR	R5	
7060						
7061	026752			ERROR	15	;ERROR-PC REG HIGH S/B CLEAR-SEE HEADER
(5)	026756	104455		TRAP	C\$ERDF	
(6)	026760	000017		.WORD	15	
(6)	026762	005055		.WORD	EMO	
(6)	026764	010006		.WORD	ERR15	
7062						;DISCUSSION.
7063						
7064	026766					
7065	026766					
(3)	026766					
(3)	026766	104401				
7066						
7067	026770			BADHEAD		
(2)				;***** TEST 43 *****		
7068				;*TEST THAT MASTER CLEAR, CLEARS BITS IN THE NPR CONTROL REGISTER AND		
7069				;*MICROPROCESSOR MISCELLANEOUS REGISTER-FIRST WE'LL SET THE		
7070				;*PRIORITY UP SO THAT WHEN WE SET THE BUS REQUEST BIT THAT IT WON'T BUG US		
7071				;*THEN WE'LL SET ALL THE BITS IN BOTH REGS EXCEPT THE		
7072				;*NPR REQUEST. WE'LL LOOK TO SEE THAT ALL GOT SET, NEXT		
7073				;*WE'LL DO A MASTER CLEAR AND BE SURE THAT THEY ALL		
7074				;*CLEAR.		
7075	026770			BADHEAD		
(2)				;***** TEST 43 *****		
7076						
7077	026770			BGNTST		

143::

(3)	026770					MYINT		
7078	026770					MOV	KMCSR,R1	;RECORD DEVICE ADDR.
(1)	026770	013701	002516			MSTCLR		
7079	026774					SETPRI	#PRI07	;DON'T ALLOW INTERRUPTS.
7080	027000					MOV	#PRI07,R0	
(3)	027000	012700	000340			TRAP	(SSPRI)	
(3)	027004	104441				MOV	#-1,4(R1)	;DATA TO BE SET
7081	027006	012761	177777	000004		BIC	#2,4(R1)	;DON'T SET AC LOW
7082	027014	042761	000002	000004		ROMCLK		
7083	027022					JSR	R5,,ROMCLK	;CLOCK INSTRUCTION
(1)	027022	004537	003044			121111		;PUT INTO MISC REG.
7084	027026	121111				BIC	#400,4(R1)	;DON'T SET NPR BIT
7085	027030	042761	000400	000004		ROMCLK		
7086	027036					JSR	R5,,ROMCLK	;CLOCK INSTRUCTION
(1)	027036	004537	003044			121130		;PUT INTO NPR REG
7087	027042	121130				ROMCLK		
7088	027044					JSR	R5,,ROMCLK	;CLOCK INSTRUCTION
(1)	027044	004537	003044			121225		;MOV MISC REG (11) TO PORT5
7089	027050	121225						
7090								
7091	027052					ROMCLK		
(1)	027052	004537	003044			JSR	R5,,ROMCLK	;CLOCK INSTRUCTION
7092	027056	121204				121204		;MOVE NPR REG (10) TO PORT4
7093	027060	012737	146636	002452		MOV	#146636,SGDDAT	;EXPECT ALL TO SET
7094	027066	016104	000004			MOV	4(R1),R4	;READ WHAT HAPPEN
7095	027072	042704	030140			BIC	#C30140,R4	;MASK UNUSED BITS
7096	027076	023704	002452			CMF	SGDDAT,R4	;DID ALL BITS GET SET?
7097	027102	001410				BEQ	108	;YES CONTINUE.
7098	027104					BRESET		
(3)	027104	104433				TRAP	(SRESET	
7099	027106					ERROR	46	;SO SORT OF PROBLEM SETTING BITS
(5)	027112	104455				TRAP	(SERDF	
(6)	027114	000056				.WORD	46	
(6)	027116	005055				.WORD	EMO	
(6)	027120	011214				.WORD	ERR46	
7100								;IN THE NPR AND/OR MISC REG.
7101	027122					(KLOOP		
(3)	027122	104406				TRAP	(SCLP1	
7102								
7103	027124	152761	000100	000001	108:	BISB	#100,1(R1)	;SET MASTER CLEAR
7104	027132	142761	000300	000001		BICB	#300,1(R1)	;CLEAR MASTER CLEAR
7105								
7106	027140					ROMCLK		
(1)	027140	004537	003044			JSR	R5,,ROMCLK	;CLOCK INSTRUCTION
7107	027144	121225				121225		;MOV MISC REG (11) TO PORT5
7108								
7109	027146					ROMCLK		
(1)	027146	004537	003044			JSR	R5,,ROMCLK	;CLOCK INSTRUCTION
7110	027152	121204				121204		;MOV NPR REG (10) TO PORT4
7111	027154	016104	000004			MOV	4(R1),R4	;READ RESULTS
7112	027160	005037	002452			CLR	SGDDAT	;EXPECT ZERO
7113	027164	042704	010140			BIC	#C10140,R4	;MASK UNUSED BITS
7114	027170	001407				BEQ	208	;IF ALL ZERO, EVERYTHING COOL.
7115								
7116	027172					ERROR	46	;MASTER CLEAR FAILED TO CLEAR
(5)	027176	104455				TRAP	(SERDF	

7117	(6)	027200	000056		.WORD	46		
7118	(6)	027202	005055		.WORD	EMO		
7119	(6)	027204	011214		.WORD	ERR46		
7120		027206			(KLOOP			:SOME BITS IN THE NPR AND/OR MISC REGS.
7121	(3)	027206	104406		TRAP	(SCLP1		
7122		027210		208:	MOV	#14,4(R1)		:NOW WE ARE GOING TO TRY TO
7123	(1)	027216	004537	003044	ROMCLK			:SET THE EXT BITS (16&17) IN THE NPR REG.
7124		027222	121110		JSR	R5,,ROMCLK		:CLOCK INSTRUCTION
7125		027224	004537	003044	ROMCLK			:IF MASTER CLEAR FAILED TO CLEAR ITSELF
7126		027230	121205		JSR	R5,,ROMCLK		:THEN WE WILL BE UNABLE TO SET
7127		027232	116104	000005	MOV	5(R1),R4		:CLOCK INSTRUCTION
7128		027236	042704	000140	BIC	#140,R4		:THESE BITS
7129		027242	012737	000014	MOV	#14,\$GDDAT		:READ REG
7130		027250	023704	002452	CMF	\$GDDAT,R4		:MASK UNUSED BITS
7131		027254	001407		BEO	308		:STORE GOOD
7132		027256			ERROR	46		:DID BITS SET?
7133	(5)	027262	104455		TRAP	(SERDF		:YES-CONTINUE
7134	(6)	027264	000056		.WORD	46		:MASTER CLEAR FAILED TO CLEAR
7135	(6)	027266	005055		.WORD	EMO		
7136	(6)	027270	011214		.WORD	ERR46		
7137		027272			(KLOOP			:ITSELF, THUS PROHIBITING US FROM
7138	(3)	027272	104406		TRAP	(SCLP1		:FURTHER SETTING BITS IN THE NPR REG.
7139		027274		308:	BRESET			
7140	(3)	027274	104433		TRAP	(BRESET		:NOW WE'LL SEE IF A BUS RESET CLEARS
7141		027276	005737	002470	TST	RUNB		:THESE BITS.
7142		027302	001016		BNE	408		:CAN'T DO THIS
7143	(1)	027304	004537	003044	ROMCLK			:TEST IF RUN SW SET.
7144		027310	121204		JSR	R5,,ROMCLK		:CLOCK INSTRUCTION
7145		027312	116104	000004	MOV	4(R1),R4		:READ MISC REG
7146		027316	001410		BEO	408		:IF ZERO-END TST
7147		027320	005037	002452	CLR	\$GDDAT		:S/B ZERO
7148		027324			ERROR	46		:BUS RESET FAILED TO CLEAR NPR REG
7149	(5)	027330	104455		TRAP	(SERDF		
7150	(6)	027332	000056		.WORD	46		
7151	(6)	027334	005055		.WORD	EMO		
7152	(6)	027336	011214		.WORD	ERR46		
7153		027340		408:				:MASTER CLEAR WAS ABLE TO LOOK TO THE
7154		027340		L10115:	ENDTST			:CIRCUITRY THAT CONVERTS BUS INIT
7155	(3)	027340						:TO "CLEAR"

CZDMQCC MB207 STATIC DIAG #2 MACV11 30A(1052) 21-JUL-81 14:48 J 13
CZDMQCC.P11 21-JUL-81 14:36 HARDWARE TESTS PAGE 36-14

SEQ 0165

(3) 027340 104401

TRAP (SETST

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 PAGE 37
HARDWARE PARAMETER CODING SECTION

SEQ 0166

.SBTTL HARDWARE PARAMETER CODING SECTION

```

7157
7158
7159
7160
7161
7162
7163
7164
7165
7166
7167
7168
7169
7170 027342      BGNHRD
      (3) 027342 000016      .WORD L10116-L$HARD/2
      (3) 027344      L$HARD::
7171
7172 027344      GPRMD      WPM,0,0,7,0,7,YES
      (4) 027344 000032      .WORD      T$CODE
      (4) 027346 027400      .WORD      WPM
      (4) 027350 000007      .WORD      7
      (4) 027352 000000      .WORD      T$LLOLIM
      (4) 027354 000007      .WORD      T$HILIM
7173 027356      GPRMA      ADDRES,2,0,160000,177776,YES
      (4) 027356 001031      .WORD      T$CODE
      (4) 027360 027452      .WORD      ADDRES
      (4) 027362 160000      .WORD      T$LLOLIM
      (4) 027364 177776      .WORD      T$HILIM
7174      :      GPRMA      VECTOR,4,0,0,674,YES
7175      :      GPRMD      PRIRTY,6,0,7000,4,7,YES
7176      :      GPRMD      LNUNIT,10,0,3,0,3,YES
7177      :      GPRMD      SWPAC1,12,0,377,0,377,YES
7178      :      GPRMD      SWPAC2,14,0,377,0,377,YES
7179      :      GPRMD      LOOPBK,16,0,40000,0,1,YES
7180 027366      GPRMD      ISRUN,24,0,7,0,1,YES
      (4) 027366 012032      .WORD      T$CODE
      (4) 027370 030050      .WORD      ISRUN
      (4) 027372 000007      .WORD      7
      (4) 027374 000000      .WORD      T$LLOLIM
      (4) 027376 000001      .WORD      T$HILIM
7181 027400      ENDHRD
      (2)
      (3) 027400      L10116:
7182
7183 027400 044127 041511 020110 WPM:      .ASCIIZ  'WHICH MICRO-CPU? (0=M8200,4=M8204,7=M8207)'
      027406 044515 051103 026517
      027414 050103 037525 024040
      027422 036460 034115 030062
      027430 026060 036464 034115
      027436 030062 026064 036467
      027444 034115 030062 000067
7184 027452 044515 051103 026517 ADDRES: .ASCIIZ  /MICRO-CPU  CSR ADDRESS : /
      027460 050103 020125 041440
      027466 051123 040440 042104
      027474 042522 051523 035040
      027502 000040

```

7185	027504	044515	051103	026517	VECTOR: .ASCIIZ /MICRO-CPU VECTOR ADDRESS : /
	027512	050103	020125	042526	
	027520	052103	051117	040440	
	027526	042104	042522	051523	
	027534	035040	000040		
7186	027540	044515	051103	026517	PRIITY: .ASCIIZ /MICRO-CPU PRIORITY LEVEL : /
	027546	050103	020125	051120	
	027554	047511	044522	054524	
	027562	046040	053105	046105	
	027570	035040	000040		
7187	027574	044127	041511	020110	LNUNIT: .ASCIIZ /WHICH LINE UNIT (0-3)? 0=NONE,1=M8201,2=M8202,3=M8203 : /
	027602	044514	042516	052440	
	027610	044516	020124	030050	
	027616	031455	037451	030040	
	027624	047075	047117	026105	
	027632	036461	034115	030062	
	027640	026061	036462	034115	
	027646	030062	026062	036463	
	027654	034115	030062	020063	
	027662	020072	000		
7188	027665	123	044527	041524	SWPAC1: .ASCIIZ /SWITCH PACK #1 (DDCMP LINE #) : /
	027672	020110	040520	045503	
	027700	021440	020061	042050	
	027706	041504	050115	046040	
	027714	047111	020105	024443	
	027722	035040	000040		
7189	027726	053523	052111	044103	SWPAC2: .ASCIIZ /SWITCH PACK #2 (BM873 BOOT ADR) : /
	027734	050040	041501	020113	
	027742	031043	024040	046502	
	027750	033470	020063	047502	
	027756	052117	040440	051104	
	027764	020051	020072	000	
7190	027771	127	046111	020114	LOOPBK: .ASCIIZ /WILL TEST CONNECTOR(S) BE USED ? 0=NO,1=YFS : /
	027776	042524	052123	041440	
	030004	047117	042516	052103	
	030012	051117	051450	020051	
	030020	042502	052440	042523	
	030026	020104	020077	036460	
	030034	047516	030454	054475	
	030042	051505	035040	000040	
7191	030050	044515	051103	026517	ISRUN: .ASCIIZ 'MICRO-PROCESSOR RUN SWITCH TYPE 0 IF OFF, 1 IF ON :'
	030056	051120	041517	051505	
	030064	047523	020122	052522	
	030072	020116	053523	052111	
	030100	044103	020040	054524	
	030106	042520	030040	044440	
	030114	020106	043117	026106	
	030122	030440	044440	020106	
	030130	047117	035040	000	
7192					
7193		030136			.EVEN
7194					
7195					
7196					
7197					
7198					

CZDMQCO M8207 STATIC DIAG #2 MACY11 30A(1052) 21-JUL-81 14:48 M 13
CZDMQC.P11 21-JUL-81 14:36 PAGE 37-2
HARDWARE PARAMETER CODING SECTION

SEQ 0168

7199

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 PAGE 37-3
SOFTWARE PARAMETER CODING SECTION

SEQ 0169

```

7201      .SBTTL  SOFTWARE PARAMETER CODING SECTION
7202
7203
7204      ;////////////////////////////////////
7205      ;/  THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
7206      ;/  THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
7207      ;/  MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
7208      ;/  INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
7209      ;/  MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
7210      ;/  WITH THE OPERATOR.
7211      ;////////////////////////////////////
7212
7213      030136      BGNSFT
7214      (3) 030136  000000      .WORD L10117-L$SOFT/2
7215      (3) 030140
7216
7217      030140      ENDSFT
7218      (2)
7219      (3) 030140      L10117: .EVEN
7220
7221      .EVEN
7222
7223
7224
7225      030140      LASTAD
7226      (2)
7227      (4) 030140  000000      .EVEN
7228      (4) 030142  000000      .WORD 0
7229      (3) 030144      .WORD 0
7230      L$LAST::
7231
7232      .END

```

[illegible]

3434#	4206	4210	4214	4219	4223	4227	4232	4237
3434#								
3434#								
3434#								
3434#								
3434#	4302	4305	4308	4311				
3434#	4438	4456	6349	7098	7137			

[illegible]

ERR2	006160 G	4178#	4662												
ERR29	010350 G	4198#	6524	6548	6747	6756	6765	6773	6783	6789	6795	6801			
ERR3	006306 G	4179#													
ERR35	010472 G	4199#	6391	6402											
ERR36	010614 G	4201#	4507												
ERR4	006434 G	4180#													
ERR40	010636 G	4205#	6869												
ERR41	010702 G	4209#	6890												
ERR42	010744 G	4213#	6898												
ERR43	011006 G	4218#	7012												
ERR44	011054 G	4222#	4913												
ERR45	011124 G	4226#	6068												
ERR46	011214 G	4231#	7099	7116	7132	7148									
ERR47	011264 G	4236#	6936												
ERR5	006556 G	4182#	5004	5017	5030	5061	5073	5085	5117	5131	5145	5177	5190	5203	
		5236	5249	5262	5295	5308	5321	5358	5372	5385	5417	5430	5443	5475	
		5489	5502	5534	5548	5561	5592	5606	5619	5650	5664	5677	5711	5727	
		5741	5772	5786	5799										
ERR6	006700 G	4184#	4706	4750	4794	4817	6581								
ERR7	007026 G	4185#	4943	4958	4970	6564	6632								
EVL	- 000004 G	3573#													
ESEND =	002100	3434#													
ESLOAD=	000035	3434#	3474												
FADR	002412	3626#	4192	5957*	6028*	6116*	6187*								
FLAG	002406	3624#	4184	4683*	4689	4690	4713*	4714	4728*	4733	4734	4757*	4758	4773*	
		4774	4777	4778	4796*	4797	4800*	4801	4804	4820*	4821	5950*	5962	6029*	
		6109*	6121	6188*											
FM1	005046	4083#	4177	4178	4179	4180	4182	4184	4185	4186	4187	4188	4189	4190	
		4192	4193	4197	4198	4199									
FTIME	002462	3646#	4293	4297*											
FSAU =	000015	3434#	4472	4473											
FSAUTO=	000020	3434#	4409	4427											
FSBGN =	000040	3434#	3440	3482	4177	4178	4179	4180	4182	4184	4185	4186	4187	4188	
		4189	4190	4192	4193	4197	4198	4199	4201	4205	4209	4213	4218	4222	
		4226	4231	4236	4252	4260	4287	4409	4437	4454	4472	4493	4500	4511	
		4513	4523	4543	4556	4563	4564	4570	4579	4588	4595	4596	4602	4612	
		4622	4630	4631	4636	4649	4651	4663	4670	4679	4707	4717	4724	4751	
		4761	4769	4795	4818	4824	4832	4833	4882	4915	4924	4925	4944		

FSEND - 00C041

3434#	3440	4177	4178	4179	4180	4182	4184	4185	4186	4187	4188	4189
4190	4192	4193	4197	4198	4199	4203	4208	4212	4216	4221	4225	4230
4234	4239	4260	4270	4406	4427	4440	4457	4473	4493	4500	4511	4513
4523	4543	4556	4563	4564	4579	4580	4588	4595	4596	4612	4613	4622
4630	4631	4649	4650	4663	4664	4670	4679	4707	4717	4724	4751	4761
4769	4795	4818	4824	4832	4833	4882	4915	4924	4925	4944	4972	4985
4987	5005	5006	5018	5019	5031	5032	5033	5046	5047	5062	5063	5074
5075	5086	5087	5088	5101	5102	5119	5120	5133	5134	5146	5147	5148
5161	5162	5178	5179	5191	5192	5204	5205	5206	5219	5220	5237	5238
5250	5251	5263	5264	5265	5278	5279	5296	5297	5309	5310	5322	5323
5324	5337	5338	5360	5361	5373	5374	5386	5387	5388	5401	5402	5418
5419	5431	5432	5444	5445	5446	5459	5460	5476	5477	5490	5491	5503
5504	5505	5518	5519	5535	5536	5549	5550	5562	5563	5564	5577	5578
5593	5594	5607	5608	5620	5621	5622	5635	5636	5651	5652	5665	5666
5678	5679	5680	5695	5696	5712	5713	5729	5730	5742	5743	5744	5757
5758	5773	5774	5787	5788	5800	5801	5802	5813	5814	5855	5860	5873
5874	5903	5910	5945	5947	6027	6071	6104	6106	6186	6193	6202	6203
6216	6228	6238	6248	6249	6262	6273	6283	6291	6337	6348	6395	6418
6429	6525	6565	6582	6586	6595	6605	6635	6644	6646	6677	6681	6687
6697	6700	6749	6759	6767	6775	6785	6790	6796	6803	6822	6833	6852
6872	6882	6891	6902	6912	6914	6939	6945	6958	6960	6981	6994	7015
7028	7030	7065	7077	7154	7181	7216						

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

3434#	7170	7181										
3434#	3519	3534										
3434#	4287	4406										
3434#	4260	4564	4596	4631	4833	4925	4987	5047	5102	5162	5220	5279
5338	5402	5460	5519	5578	5636	5696	5758	5814	5874	5947	6106	6203
6249	6395	6586	6646	6681	6700	6803	6852	6914	6960	7030		

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

3434#	3440											
3434#	4177	4178	4179	4180	4182	4184	4185	4186	4187	4188	4189	4190
4192	4193	4197	4198	4199	4201	4203	4205	4208	4209	4212	4213	4216
4218	4221	4222	4225	4226	4230	4231	4234	4236	4239			

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

3434#	3482	3486										
3434#												
3434#	4252	4270										
3434#	4570	4580	4602	4613	4636	4650	4651	4664	4992	5006	5007	5019
5020	5032	5052	5063	5064	5075	5076	5087	5107	5120	5121	5134	5135
5147	5167	5179	5180	5192	5193	5205	5226	5238	5239	5251	5252	5264
5285	5297	5298	5310	5311	5323	5343	5361	5362	5374	5375	5387	5407
5419	5420	5432	5433	5445	5464	5477	5478	5491	5492	5504	5523	5536
5537	5550	5551	5563	5582	5594	5595	5608	5609	5621	5640	5652	5653
5666	5667	5679	5700	5713	5714	5730	5731	5743	5762	5774	5775	5788
5789	5801											

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

3434#	7213	7216										
3434#												
3434#												
3434#	3543	3546										
3434#	4493	4513	4523	4556	4563	4588	4595	4622	4630	4670	4679	4717
4724	4761	4769	4824	4832	4915	4924	4972	4985	5033	5046	5088	5101
5148	5161	5206	5219	5265	5278	5324	5337	5388	5401	5446	5459	5505
5518	5564	5577	5622	5635	5680	5695	5744	5757	5802	5813	5860	5873
5910	5945	6071	6104	6193	6202	6238	6248	6283	6291	6337	6348	6418
6429	6595	6605	6635	6644	6687	6697	6822	6833	6872	6882	6902	6912
6945	6958	6981	6994	7015	7028	7065	7077	7154				

GETPRM 011462
GSCNTO- 000200

4312	4321#	4326
3434#		

[illegible]

```

ISSFTJ= 000041
ISSFT = 000041
ISSRL = 000041
ISSLB = 000041

```

ZS'S' = 000041

```
JSJMP = 0C0167
KPIACTV 002376
KPICSR 002516
```

KMCSRM	002520	
KMCTL	002522	
KMINUM	002400	
KMPO4	002524	
KMPO6	002526	
KMRLVL	002510	
KMRVEC	002506	
KMTLVL	002514	
KMTVEC	002512	
LNUNIT	027574	
LOADER	026006	
LOCK	002340	
LOE =	040000	G
LOGDEV	002342	
LOKFLG	002476	
LOOPBK	027771	
LOT =	000010	G
LSACP	002110	G
LSAPT	002036	G
LSAU	012144	G
LSAUT	002070	G
LSAUTO	012042	G
LSCCP	002106	G
LSCLEA	012134	G
LSCO	002032	G
LSDEPO	002011	G
LSDESC	002312	G
LSDESP	002076	G
LSDEVP	002060	G
LSDISP	002132	G
LSIV	002116	G

[illegible]

CDMOCG M8207 STATIC DIAG #2
CDMOC.P1 21-JUL-81 14:36

MACV11 30A(1052) 21-JUL-81 14:48 PAGE 38-7
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0177

LSOTP	002040	G	3474#	
LSOTVP	002034	G	3474#	
LSOU	012140	G	3474	4454#
LSOLT	002072	G	3474#	
LSO.TV	002730	G	3474	3723#
LSF	002052	G	3474#	
LSENV1	002044	G	3474#	
SETP	002102	G	3474#	
SEXP1	002046	G	3474#	
SEXP4	002064	G	3474#	
SEXP5	002066	G	3474#	
SHARD	027344	G	3474	7170#
SHIME	002120	G	3474#	
SHPCP	002016	G	3474#	
SHPTP	002072	G	3474#	
SHW	002262	G	3474	3519#
SIP	002104	G	3474#	
SINT	011340	G	3474	4287#
SLADP	002026	G	3474#	
SLAST	030144	G	3474	7225#
SLOAD	002100	G	3474#	
SLJN	002074	G	3474#	
SMREV	002050	G	3474#	
SNAME	002000	G	3474#	
SPRIO	002042	G	3474#	
SPROT	002122	J	3474	3482#
SPRT	002112	G	3474#	
SREPP	002062	G	3474#	
SREV	002010	G	3474#	
SRPT	011332	G	4252#	
SSOFT	030140	G	7213#	
SSPC	002056	G	3474#	
SSPCP	002020	G	3474#	
SSPTP	002024	G	3474#	
SSTA	002030	G	3474#	
SSW	002312	G	3543#	
STEST	002114	G	3474#	
STIML	002014	G	3474#	
SUNIT	002012	G	3474#	4323
L10001	002310		3519	3534#
L10002	002312		3543	3546#
L10003	006156		4177#	
L10004	006304		4178#	
L10005	006432		4179#	
L10006	006554		4180#	
L10007	006676		4182#	
L10010	007024		4184#	
L10011	007152		4185#	
L10012	007274		4186#	
L10013	007416		4187#	
L10014	007540		4188#	
L10015	007662		4189#	
L10016	010004		4190#	
L10017	010132		4192#	
L10020	010254		4193#	
L10021	010346		4197#	

[illegible]

L10112	026542	6960	6981#												
L10113	026642	7015#													
L10114	026766	7030	7065#												
L10115	027340	7154#													
L10116	027400	7170	7181#												
L10117	030140	7213	7216#												
MASKX	002362	3614#													
MEM LIM	002374	3619#													
MEMSET	003474	4005#	4993	5051	5106	5166	5224	5283	5341	5406	5463	5522	5581	5639	
		5699	5761												
MEMSZ	002436	3636#	4000	4020	4391*	4399*	4585	4619	4653	4667	4714	4758	4797	4821	
		4833	4838	4872	5814	5874	5947	6106	6203	6249	6658	6678			
MRO	002434	3635#	4050	4567*	4572	4584*	4585	4599*	4605	4618*	4619	4635*	4637	4639	
		4640	4646	4652*	4653	4655*	4656	4658	4660	4666*	4667	4684*	4695	4702	
		4710*	4729*	4730*	4739	4746	4752*	4754*	6521*						
		4303	4306	4314#	4324										
NEWST	011454	3600#													
NEXT	002336	3618#													
ONE	002372	3434#	3474												
OSAPTS=	000000	3434#	3464#	3474											
OSAU =	000001	3434#	3474												
OSBGNR=	000000	3434#	3474												
OSBGNS=	000000	3434#	3474												
OSDU =	000001	3434#	3464#	3474											
OSERRT=	000000	3434#	3474												
OSGNSW=	000000	3434#	3474												
OSPOIN=	000001	3434#	3464#	3474											
OSSETU=	000000	3434#	3474	7225											
PATCH	002774	3871#													
PNT =	001000 G	3573#													
PRI =	002000 G	3573#													
PRI RTY	027540	7186#													
PRI00 =	000000 G	3573#													
PRI01 =	000040 G	3573#													
PRI02 =	000100 G	3573#													
PRI03 =	000140 G	3573#													
PRI04 =	000200 G	3573#													
PRI05 =	000240 G	3573#													
PRI06 =	000300 G	3573#													
PRI07 =	000340 G	3573#	7080												
PSTACK	002344	3607#	4292*	6384											
QV.FLG	002477	3658#													
RAMDAT	003330	3964#	5000	5013	5026	5057	5069	5081	5113	5127	5141	5173	5186	5199	
		5232	5245	5258	5291	5304	5317	5351	5368	5381	5413	5426	5439	5471	
		5485	5498	5530	5544	5557	5588	5602	5615	5646	5660	5673	5706	5721	
		5737	5768	5782	5795										
REGADR	002530	3699#													
RETADR	002352	3610#													
ROMMAP	012146	3994	4486#	6650	6664										
RUN	002410	3625#													
RUNB	002470	3649#	4403*	4892	6038	6387	7034	7139							
RUNINH	002472	3650#	4354*	4403	4894	6040									
SAVACT	002402	3622#													
SAVE4	002464	3647#	4295*	4298	4425	4509	6850								
SAVE6	002466	3648#	4296*	4299	4426	4510	6851								
SAVNUM	002404	3623#													
SAVPC	002366	3616#													

SAVSP	002364	3615#												
SETBRO	003220	3915#	5227	5240	5253									
SETBR1	003230	3921#	5286	5299	5312									
SETBR4	003240	3928#	5345	5363	5376									
SETBR7	003250	3935#	5408	5421	5434									
SETC	003260	3942#	5108	5122	5136	5465	6263							
SETZ	003312	3955#	5168	5181	5194	5524	6217							
SFPTBL	002312 G	3543#												
SSTACK	002730	3703#	4290											
STAT	002356	3612#												
STAT1	002500	3680#	4355*	4357*	4361*	4372*	4376*	4379*						
STAT2	002502	3681#	4382*	4384*										
STAT3	002504	3682#												
STM	005627	4108#	4177	4178	4179	4180	4182	4184	4185	4186	4187	4188	4189	4190
		4192	4193	4197	4198	4199	4202	4207	4211	4215	4220	4224	4229	4233
		4238												
STRTSW	002354	3611#												
SUBRPC	002346	3608#												
SVCGBL=	000000	3434#	3440	3448#	3474	3482	3496	3519	3543	3595	3723	4177	4178	4179
		4180	4182	4184	4185	4186	4187	4188	4189	4190	4192	4193	4197	4198
		4199	4201	4205	4209	4213	4218	4222	4226	4231	4236	4252	4287	4409
		4437	4454	4472	7170	7213	7225#							
SVCINS=	000000	3434#	3445#	3474	3496	3519	3543	3595	3723	4177	4178	4179	4180	4182
		4184	4185	4186	4187	4188	4189	4190	4192	4193	4197	4198	4199	4202
		4203	4206	4207	4208	4210	4211	4212	4214	4215	4216	4219	4220	4221
		4223	4224	4225	4227	4228	4229	4230	4232	4233	4234	4237	4238	4239
		4260	4270	4302	4303	4305	4306	4308	4309	4311	4312	4325	4326	4406
		4423	4427	4438	4440	4456	4457	4473	4500	4507	4511	4513	4541	4543
		4553	4556	4564	4570	4578	4579	4580	4588	4596	4602	4611	4612	4613
		4622	4631	4636	4648	4649	4650	4651	4662	4663	4664	4670	4706	4707
		4712	4717	4750	4751	4756	4761	4794	4795	4817	4818	4819	4824	4833
		4860	4878	4882	4883	4913	4915	4925	4943	4944	4958	4970	4972	4987
		4992	5004	5005	5006	5007	5017	5018	5019	5020	5030	5031	5032	5033
		5047	5052	5061	5062	5063	5064	5073	5074	5075	5076	5085	5086	5087
		5088	5102	5107	5117	5119	5120	5121	5131	5133	5134	5135	5145	5146
		5147	5148	5162	5167	5177	5178	5179	5180	5190	5191	5192	5193	5203
		5204	5205	5206	5220	5226	5236	5237	5238	5239	5249	5250	5251	5252
		5262	5263	5264	5265	5279	5285	5295	5296	5297	5298	5308	5309	5310
		5311	5321	5322	5323	5324	5338	5343	5358	5360	5361	5362	5372	5373
		5374	5375	5385	5386	5387	5388	5402	5407	5417	5418	5419	5420	5430
		5431	5432	5433	5443	5444	5445	5446	5460	5464	5475	5476	5477	5478
		5489	5490	5491	5492	5502	5503	5504	5505	5519	5523	5534	5535	5536
		5537	5548	5549	5550	5551	5561	5562	5563	5564	5578	5582	5592	5593
		5594	5595	5606	5607	5608	5609	5619	5620	5621	5622	5636	5640	5650
		5651	5652	5653	5664	5665	5666	5667	5677	5678	5679	5680	5696	5700
		5711	5712	5713	5714	5727	5729	5730	5731	5741	5742	5743	5744	5758
		5762	5772	5773	5774	5775	5786	5787	5788	5789	5799	5800	5801	5802
		5814	5850	5855	5860	5874	5899	5903	5910	5947	5996	6027	6068	6071
		6106	6155	6186	6193	6203	6214	6216	6226	6228	6236	6238	6249	6260
		6262	6271	6273	6281	6283	6308	6320	6332	6337	6349	6364	6382	6391
		6395	6402	6411	6418	6524	6525	6548	6564	6565	6581	6582	6586	6595
		6632	6635	6646	6674	6677	6681	6687	6700	6747	6749	6756	6759	6765
		6767	6773	6775	6783	6785	6789	6790	6795	6796	6801	6803	6822	6852
		6869	6872	6890	6891	6898	6902	6914	6936	6939	6945	6960	6977	6981
		7012	7015	7030	7061	7065	7080	7098	7099	7101	7116	7118	7132	7135
		7137	7148	7154	7170	7172	7173	7180	7181	7213	7216	7225		

CZDMU LU MB207 STATIC DIAG #2 MACY11 30A(1052) 21-JUL-81 14:48 PAGE 38-11
 CZDMQC.P11 21-JUL-81 14:36 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0181

SVCSUB= 000000
 SVCTAG= 000000

SVCTST= 000000

SV05 003632

SWPAC1 027665
 SWPAC2 027726
 SLSYM- 010000

TEMP 002440
 TFM1 003666
 TFM2 003712
 TFM3 003730
 TFM36 004017
 TFM4 003755
 TFM40 004263
 TFM41 004105
 TFM42 004174
 TFM43 004344
 TFM44 004427
 TFM45 004466
 TFM45A 004524
 TFM46 004647
 TFM47 004733
 TFM5 003773
 TMMC 005010
 TYPE 002432
 TSARGC= 000001

3434#	3447#												
3434#	3449#	3534	3546	4177	4178	4179	4180	4182	4184	4185	4186	4187	
4188	4189	4190	4192	4193	4197	4198	4199	4203	4208	4212	4216	4221	
4225	4230	4234	4239	4270	4406	4427	4440	4457	4473	4513	4556	4580	
4588	4613	4622	4650	4664	4670	4717	4761	4824	4915	4972	5006	5019	
5032	5033	5063	5075	5087	5088	5120	5134	5147	5148	5179	5192	5205	
5206	5238	5251	5264	5265	5297	5310	5323	5324	5361	5374	5387	5388	
5419	5432	5445	5446	5477	5491	5504	5505	5536	5550	5563	5564	5594	
5608	5621	5622	5652	5666	5679	5680	5713	5730	5743	5744	5774	5788	
5801	5802	5860	5910	6071	6193	6238	6283	6337	6418	6595	6635	6687	
6822	6872	6902	6945	6981	7015	7065	7154	7181	7216				
3434#	3446#	4493	4523	4563	4595	4630	4679	4724	4769	4832	4924	4985	
5046	5101	5161	5219	5278	5337	5401	5459	5518	5577	5635	5695	5757	
5813	5873	5945	6104	6202	6248	6291	6348	6429	6605	6644	6697	6833	
6882	6912	6958	6994	7028	7077								
4045#	4507	4541	4553	4578	4611	4648	4662	4706	4750	4794	4817	4860	
4878	4913	4943	4958	4970	5004	5017	5030	5061	5073	5085	5117	5131	
5145	5177	5190	5203	5236	5249	5262	5295	5308	5321	5358	5372	5385	
5417	5430	5443	5475	5489	5502	5534	5548	5561	5592	5606	5619	5650	
5664	5677	5711	5727	5741	5772	5786	5799	5850	5899	5996	6068	6155	
6214	6226	6236	6260	6271	6281	6364	6382	6391	6402	6411	6524	6548	
6564	6581	6632	6674	6747	6756	6765	6773	6783	6789	6795	6801	6869	
6890	6898	6936	6977	7012	7061	7099	7116	7132	7148				
7188#													
7189#													
3434#	3534#	3546#	4177#	4178#	4179#	4180#	4182#	4184#	4185#	4186#	4187#	4188#	
4189#	4190#	4192#	4193#	4197#	4198#	4199#	4203#	4208#	4212#	4216#	4221#	4225#	
4230#	4234#	4239#	4270#	4406#	4427#	4440#	4457#	4473#	4513#	4556#	4570#	4588#	
4602#	4622#	4636#	4651#	4670#	4717#	4761#	4824#	4915#	4972#	4992#	5007#	5020#	
5033#	5052#	5064#	5076#	5088#	5107#	5121#	5135#	5148#	5167#	5180#	5193#	5206#	
5226#	5239#	5252#	5265#	5285#	5298#	5311#	5324#	5343#	5362#	5375#	5388#	5407#	
5420#	5433#	5446#	5464#	5478#	5492#	5505#	5523#	5537#	5551#	5564#	5582#	5595#	
5609#	5622#	5640#	5653#	5667#	5680#	5700#	5714#	5731#	5744#	5762#	5775#	5789#	
5802#	5860#	5910#	6071#	6193#	6238#	6283#	6337#	6418#	6595#	6635#	6687#	6822#	
6872#	6902#	6945#	6981#	7015#	7065#	7154#	7181#	7216#					
3637#	6294*	6314*	6329*	6353*	6361*								
4066#	4177	4178	4179										
4067#	4180	4182	4186	4189	4198	4199							
4068#	4184	4185											
4071#													
4069#	4187	4188	4190	4193									
4074#	4206												
4072#	4210												
4073#	4214												
4075#	4219												
4076#	4223												
4077#	4227												
4078#	4228												
4080#	4232												
4081#	4237												
4070#	4192												
4082#	4223	4227											
3634#	4392*	4397*	4402*	4564	4596	4631	4925	6504					
3474#	4177#	4178#	4179#	4180#	4182#	4184#	4185#	4186#	4187#	4188#	4189#	4190#	
4192#	4193#	4197#	4198#	4199#	4202#	4206#	4207#	4210#	4211#	4214#	4215#	4219#	
4220#	4223#	4224#	4227#	4228#	4229#	4232#	4233#	4237#	4238#				

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 PAGE 38-12
CROSS REFERENCE TABLE -- USER SYMBOLS

N 14

SEQ 0182

TSCODE= 012032
TSERRN= 000056

TSEXCP= 000000
TSFLAG= 000040

TSGMAN= 000000
TSHILI= 000001
TSLAST= 000001
TSLOLI= 000000
TSLSYM= 010000

TSLTNO= 000053
TSNEST= 000000

TNSO - 000000
TNS1 000005

7172#	7173#	7180#											
3434#	4507#	4541#	4553#	4578#	4611#	4648#	4662#	4706#	4750#	4794#	4817#	4860#	
4878#	4913#	4943#	4958#	4970#	5004#	5017#	5030#	5061#	5073#	5085#	5117#	5131#	
5145#	5177#	5190#	5203#	5236#	5249#	5262#	5295#	5308#	5321#	5358#	5372#	5385#	
5417#	5430#	5443#	5475#	5489#	5502#	5534#	5548#	5561#	5592#	5606#	5619#	5650#	
5664#	5677#	5711#	5727#	5741#	5772#	5786#	5799#	5850#	5899#	5996#	6068#	6155#	
6214#	6226#	6236#	6260#	6271#	6281#	6308#	6320#	6332#	6364#	6382#	6391#	6402#	
6411#	6524#	6548#	6564#	6581#	6632#	6674#	6747#	6756#	6765#	6773#	6783#	6789#	
6795#	6801#	5869#	6890#	6898#	6936#	6977#	7012#	7061#	7099#	7116#	7132#	7148#	
7172#	7173#	7180#											
4260#	4500#	4511#	4543#	4564#	4579#	4596#	4612#	4631#	4649#	4663#	4707#	4751#	
4795#	4818#	4833#	4882#	4925#	4944#	4987#	5005#	5018#	5031#	5047#	5062#	5074#	
5086#	5102#	5119#	5133#	5146#	5162#	5178#	5191#	5204#	5220#	5237#	5250#	5263#	
5279#	5296#	5309#	5322#	5338#	5360#	5373#	5386#	5402#	5418#	5431#	5444#	5460#	
5476#	5490#	5503#	5519#	5535#	5549#	5562#	5578#	5593#	5607#	5620#	5636#	5651#	
5665#	5678#	5696#	5712#	5729#	5742#	5758#	5773#	5787#	5800#	5814#	5855#	5874#	
5903#	5947#	6027#	6106#	6186#	6203#	6216#	6228#	6249#	6262#	6273#	6395#	6525#	
6565#	6582#	6586#	6646#	6677#	6681#	6700#	6749#	6759#	6767#	6775#	6785#	6790#	
6796#	6803#	6852#	6891#	6914#	6939#	6960#	7030#						
3434#													
7172#	7173#	7180#											
3434#	7225#												
7172#	7173#	7180#											
3434#	3534	3546	4177	4178	4179	4180	4182	4184	4185	4186	4187	4188	
4189	4190	4192	4193	4197	4198	4199	4203	4208	4212	4216	4221	4225	
4230	4234	4239	4270	4406	4427	4440	4457	4473	4513	4556	4588	4622	
4670	4717	4761	4824	4915	4972	5033	5088	5148	5206	5265	5324	5388	
5446	5505	5564	5622	5680	5744	5802	5860	5910	6071	6193	6238	6283	
6337	6418	6595	6635	6687	6822	6872	6902	6945	6981	7015	7065	7154	
7181	7216												
7225#													
3434#	3440#	3482#	3486#	3519#	3534#	3543#	3546#	4177#	4178#	4179#	4180#	4182#	
4184#	4185#	4186#	4187#	4188#	4189#	4190#	4192#	4193#	4197#	4198#	4199#	4201#	
4203#	4205#	4208#	4209#	4212#	4213#	4216#	4218#	4221#	4222#	4225#	4226#	4230#	
4231#	4234#	4236#	4239#	4252#	4270#	4287#	4406#	4409#	4427#	4437#	4440#	4454#	
4457#	4472#	4473#	4493#	4513#	4523#	4556#	4563#	4570#	4580#	4588#	4595#	4602#	
4613#	4622#	4630#	4636#	4650#	4651#	4664#	4670#	4679#	4717#	4724#	4761#	4769#	
4824#	4832#	4915#	4924#	4972#	4985#	4992#	5006#	5007#	5019#	5020#	5032#	5033#	
5046#	5052#	5063#	5064#	5075#	5076#	5087#	5088#	5101#	5107#	5120#	5121#	5134#	
5135#	5147#	5148#	5161#	5167#	5179#	5180#	5192#	5193#	5205#	5206#	5219#	5226#	
5238#	5239#	5251#	5252#	5264#	5265#	5278#	5285#	5297#	5298#	5310#	5311#	5323#	
5324#	5337#	5343#	5361#	5362#	5374#	5375#	5387#	5388#	5401#	5407#	5419#	5420#	
5432#	5433#	5445#	5446#	5459#	5464#	5477#	5478#	5491#	5492#	5504#	5505#	5518#	
5523#	5536#	5537#	5550#	5551#	5563#	5564#	5577#	5582#	5594#	5595#	5608#	5609#	
5621#	5622#	5635#	5640#	5652#	5653#	5666#	5667#	5679#	5680#	5695#	5700#	5713#	
5714#	5730#	5731#	5743#	5744#	5757#	5762#	5774#	5775#	5788#	5789#	5801#	5802#	
5813#	5860#	5873#	5910#	5945#	6071#	6104#	6193#	6202#	6238#	6248#	6283#	6291#	
6337#	6348#	6418#	6429#	6595#	6605#	6635#	6644#	6687#	6697#	6822#	6833#	6872#	
6882#	6902#	6912#	6945#	6958#	6981#	6994#	7015#	7028#	7065#	7077#	7154#	7170#	
7181#	7213#	7216#											
3440#													
3482#	3486	3519#	3534	3543#	3546	4177#	4178#	4179#	4180#	4182#	4184#	4185#	
4186#	4187#	4188#	4189#	4190#	4192#	4193#	4197#	4198#	4199#	4201#	4203	4205#	
4208	4209#	4212	4213#	4216	4218#	4221	4222#	4225	4226#	4230	4231#	4234	
4236#	4239	4252#	4270	4287#	4406	4409#	4427	4437#	4440	4454#	4457	4472#	
4473	4493#	4513	4523#	4556	4563#	4588	4595#	4622	4630#	4670	4679#	4717	

T\$NS2 = 000003

T\$PTNU= 000000
T\$SAVL= 177777
T\$SEGL= 177777

T\$SEK0= 010002

T\$SUBN= 000000

T\$TAGL= 177777
T\$TAGN= 010120

T\$TEMP 000005

4724#	4761	4769#	4824	4832#	4915	4924#	4972	4985#	5033	5046#	5088	5101#
5148	5161#	5206	5219#	5265	5278#	5324	5337#	5388	5401#	5446	5459#	5505
5518#	5564	5577#	5622	5635#	5680	5695#	5744	5757#	5802	5813#	5860	5873#
5910	5945#	6071	6104#	6193	6202#	6238	6248#	6283	6291#	6337	6348#	6418
6429#	6595	6605#	6635	6644#	6687	6697#	6822	6833#	6872	6882#	6902	6912#
6945	6958#	6981	6994#	7015	7028#	7065	7077#	7154	7170#	7181	7213#	7216
4570#	4580	4602#	4613	4636#	4650	4651#	4664	4992#	5006	5007#	5019	5020#
5032	5052#	5063	5064#	5075	5076#	5087	5107#	5120	5121#	5134	5135#	5147
5167#	5179	5180#	5192	5193#	5205	5226#	5238	5239#	5251	5252#	5264	5285#
5297	5298#	5310	5311#	5323	5343#	5361	5362#	5374	5375#	5387	5407#	5419
5420#	5432	5433#	5445	5464#	5477	5478#	5491	5492#	5504	5523#	5536	5537#
5550	5551#	5563	5582#	5594	5595#	5608	5609#	5621	5640#	5652	5653#	5666
5667#	5679	5700#	5713	5714#	5730	5731#	5743	5762#	5774	5775#	5788	5789#
5801												
3434#												
3434#												
3434#	4570#	4579	4580#	4602#	4612	4613#	4636#	4649	4650#	4651#	4663	4664#
4992#	5005	5006#	5007#	5018	5019#	5020#	5031	5032#	5052#	5062	5063#	5064#
5074	5075#	5076#	5086	5087#	5107#	5119	5120#	5121#	5133	5134#	5135#	5146
5147#	5167#	5178	5179#	5180#	5191	5192#	5193#	5204	5205#	5226#	5237	5238#
5239#	5250	5251#	5252#	5263	5264#	5285#	5296	5297#	5298#	5309	5310#	5311#
5322	5323#	5343#	5360	5361#	5362#	5373	5374#	5375#	5386	5387#	5407#	5418
5419#	5420#	5431	5432#	5433#	5444	5445#	5464#	5476	5477#	5478#	5490	5491#
5492#	5503	5504#	5523#	5535	5536#	5537#	5549	5550#	5551#	5562	5563#	5582#
5593	5594#	5595#	5607	5608#	5609#	5620	5621#	5640#	5651	5652#	5653#	5665
5666#	5667#	5678	5679#	5700#	5712	5713#	5714#	5729	5730#	5731#	5742	5743#
5762#	5773	5774#	5775#	5787	5788#	5789#	5800	5801#				
4570#	4579	4580	4602#	4612	4613	4636#	4649	4650	4651#	4663	4664	4992#
5005	5006	5007#	5018	5019	5020#	5031	5032	5052#	5062	5063	5064#	5074
5075	5076#	5086	5087	5107#	5119	5120	5121#	5133	5134	5135#	5146	5147
5167#	5178	5179	5180#	5191	5192	5193#	5204	5205	5226#	5237	5238	5239#
5250	5251	5252#	5263	5264	5285#	5296	5297	5298#	5309	5310	5311#	5322
5323	5343#	5360	5361	5362#	5373	5374	5375#	5386	5387	5407#	5418	5419
5420#	5431	5432	5433#	5444	5445	5464#	5476	5477	5478#	5490	5491	5492#
5503	5504	5523#	5535	5536	5537#	5549	5550	5551#	5562	5563	5582#	5593
5594	5595#	5607	5608	5609#	5620	5621	5640#	5651	5652	5653#	5665	5666
5667#	5678	5679	5700#	5712	5713	5714#	5729	5730	5731#	5742	5743	5762#
5773	5774	5775#	5787	5788	5789#	5800	5801					
3434#	4493#	4523#	4563#	4595#	4630#	4679#	4724#	4769#	4832#	4924#	4985#	5046#
5101#	5161#	5219#	5278#	5337#	5401#	5459#	5518#	5577#	5635#	5695#	5757#	5813#
5873#	5945#	6104#	6202#	6248#	6291#	6348#	6429#	6605#	6644#	6697#	6833#	6882#
6912#	6958#	6994#	7028#	7077#								
3434#												
3434#	3482#	3519#	3543#	4177#	4178#	4179#	4180#	4182#	4184#	4185#	4186#	4187#
4188#	4189#	4190#	4192#	4193#	4197#	4198#	4199#	4201#	4205#	4209#	4213#	4218#
4222#	4226#	4231#	4236#	4252#	4287#	4409#	4437#	4454#	4472#	4493#	4523#	4563#
4595#	4630#	4679#	4724#	4769#	4832#	4924#	4985#	5046#	5101#	5161#	5219#	5278#
5337#	5401#	5459#	5518#	5577#	5635#	5695#	5757#	5813#	5873#	5945#	6104#	6202#
6248#	6291#	6348#	6429#	6605#	6644#	6697#	6833#	6882#	6912#	6958#	6994#	7028#
7077#	7170#	7213#										
3486#	3496#	3534#	3546#	4177#	4178#	4179#	4180#	4182#	4184#	4185#	4186#	4187#
4188#	4189#	4190#	4192#	4193#	4197#	4198#	4199#	4203#	4208#	4212#	4216#	4221#
4225#	4230#	4234#	4239#	4260#	4270#	4406#	4427#	4440#	4457#	4473#	4500#	4511#
4513#	4543#	4556#	4564#	4579#	4580#	4588#	4596#	4612#	4613#	4622#	4631#	4649#
4650#	4663#	4664#	4670#	4707#	4717#	4751#	4761#	4795#	4818#	4824#	4833#	4882#
4915#	4925#	4944#	4972#	4987#	5005#	5006#	5018#	5019#	5031#	5032#	5033#	5047#

TEST= 000053

TESTM= 177777

5062#	5063#	5074#	5075#	5086#	5087#	5088#	5102#	5119#	5120#	5133#	5134#	5146#
5147#	5148#	5162#	5178#	5179#	5191#	5192#	5204#	5205#	5206#	5220#	5237#	5238#
5250#	5251#	5263#	5264#	5265#	5279#	5296#	5297#	5309#	5310#	5322#	5323#	5324#
5338#	5360#	5361#	5373#	5374#	5386#	5387#	5388#	5402#	5418#	5419#	5431#	5432#
5444#	5445#	5446#	5460#	5476#	5477#	5490#	5491#	5503#	5504#	5505#	5519#	5535#
5536#	5549#	5550#	5562#	5563#	5564#	5578#	5593#	5594#	5607#	5608#	5620#	5621#
5622#	5636#	5651#	5652#	5665#	5666#	5678#	5679#	5680#	5696#	5712#	5713#	5729#
5730#	5742#	5743#	5744#	5758#	5773#	5774#	5787#	5788#	5800#	5801#	5802#	5814#
5855#	5860#	5874#	5903#	5910#	5947#	6027#	6071#	6106#	6186#	6193#	6203#	6216#
6228#	6238#	6249#	6262#	6273#	6283#	6337#	6395#	6418#	6525#	6565#	6582#	6586#
6595#	6635#	6646#	6677#	6681#	6687#	6700#	6749#	6759#	6767#	6775#	6785#	6790#
6796#	6803#	6822#	6852#	6872#	6891#	6902#	6914#	6939#	6945#	6960#	6981#	7015#
7030#	7065#	7154#	7172#	7173#	7180#	7181#	7216#					
3434#	4488	4491	4493#	4517	4521	4523#	4558	4561	4563#	4590	4593	4595#
4624	4628	4630#	4673	4676	4679#	4719	4722	4724#	4763	4767	4769#	4826
4830	4832#	4917	4922	4924#	4974	4983	4985#	5035	5044	5046#	5090	5099
5101#	5150	5159	5161#	5208	5217	5219#	5267	5276	5278#	5326	5335	5337#
5390	5399	5401#	5448	5457	5459#	5507	5516	5518#	5566	5575	5577#	5624
5633	5635#	5682	5692	5695#	5746	5755	5757#	5804	5811	5813#	5863	5871
5873#	5912	5943	5945#	6072	6102	6104#	6194	6200	6202#	6240	6246	6248#
6284	6289	6291#	6339	6346	6348#	6420	6427	6429#	6597	6603	6605#	6637
6642	6644#	6690	6695	6697#	6824	6831	6833#	6874	6880	6882#	6904	6910
6912#	6947	6956	6958#	6983	6992	6994#	7017	7026	7028#	7067	7075	7077#
7225												
3434#	4177	4178	4179	4180	4182	4184	4185	4186	4187	4188	4189	4190
4192	4193	4197	4198	4199	4202	4203	4206	4207	4208	4210	4211	4212
4214	4215	4216	4219	4220	4221	4223	4224	4225	4227	4228	4229	4230
4232	4233	4234	4237	4238	4239	4270	4302	4305	4308	4311	4325	4406
4423	4427	4438	4440	4456	4457	4473	4500	4507	4511	4513	4541	4543
4553	4556	4564	4570	4578	4579	4580	4588	4596	4602	4611	4612	4613
4622	4631	4636	4648	4649	4650	4651	4662	4663	4664	4670	4706	4707
4712	4717	4750	4751	4756	4761	4794	4795	4817	4818	4819	4824	4833
4860	4878	4882	4883	4913	4915	4925	4943	4944	4958	4970	4972	4987
4992	5004	5005	5006	5007	5017	5018	5019	5020	5030	5031	5032	5033
5047	5052	5061	5062	5063	5064	5073	5074	5075	5076	5085	5086	5087
5088	5102	5107	5117	5119	5120	5121	5131	5133	5134	5135	5145	5146
5147	5148	5162	5167	5177	5178	5179	5180	5190	5191	5192	5193	5203
5204	5205	5206	5220	5226	5236	5237	5238	5239	5249	5250	5251	5252
5262	5263	5264	5265	5279	5285	5295	5296	5297	5298	5308	5309	5310
5311	5321	5322	5323	5324	5338	5343	5358	5360	5361	5362	5372	5373
5374	5375	5385	5386	5387	5388	5402	5407	5417	5418	5419	5420	5430
5431	5432	5433	5443	5444	5445	5446	5460	5464	5475	5476	5477	5478
5489	5490	5491	5492	5502	5503	5504	5505	5519	5523	5534	5535	5536
5537	5548	5549	5550	5551	5561	5562	5563	5564	5578	5582	5592	5593
5594	5595	5606	5607	5608	5609	5619	5620	5621	5622	5636	5640	5650
5651	5652	5653	5664	5665	5666	5667	5677	5678	5679	5680	5696	5700
5711	5712	5713	5714	5727	5729	5730	5731	5741	5742	5743	5744	5758
5762	5772	5773	5774	5775	5786	5787	5788	5789	5799	5800	5801	5802
5814	5850	5855	5860	5874	5899	5903	5910	5947	5996	6027	6068	6071
6106	6155	6186	6193	6203	6214	6216	6226	6228	6236	6238	6249	6260
6262	6271	6273	6281	6283	6308	6320	6332	6337	6349	6364	6382	6391
6395	6402	6411	6418	6524	6525	6548	6564	6565	6581	6582	6586	6595
6632	6635	6646	6674	6677	6681	6687	6700	6747	6749	6756	6759	6765
6767	6773	6775	6783	6785	6789	6790	6795	6796	6801	6803	6822	6852
6869	6872	6890	6891	6898	6902	6914	6936	6939	6945	6960	6977	6981
7012	7015	7030	7061	7065	7080	7098	7099	7101	7116	7118	7132	7135

TSTSTS= 000001	7137	7148	7154										
	3434#	4493#	4523#	4563#	4595#	4630#	4679#	4724#	4769#	4832#	4924#	4985#	5046#
	5101#	5161#	5219#	5278#	5337#	5401#	5459#	5518#	5577#	5635#	5695#	5757#	5813#
	5873#	5945#	6104#	6202#	6248#	6291#	6348#	6429#	6605#	6644#	6697#	6833#	6882#
	6912#	6958#	6994#	7028#	7077#								
TSSAU = 010042	4472#	4473											
TSSAUT= 010037	4409#	4427											
TSSCLE= 010040	4437#	4440											
TSSDU = 010041	4454#	4457											
TSSHAR= 010116	7170#	7181											
TSSHW = 010001	3519#	3534											
TSSINI= 010036	4287#	4406											
TSSMSG= 010034	4177#	4178#	4179#	4180#	4182#	4184#	4185#	4186#	4187#	4188#	4189#	4190#	4192#
	4193#	4197#	4198#	4199#	4201#	4203	4205#	4208	4209#	4212	4213#	4216	4218#
	4221	4222#	4225	4226#	4230	4231#	4234	4236#	4239				
TSSPRO= 010000	3482#												
TSSRPT= 010035	4252#	4260	4270										
TSSSEG= 010002	4570#	4579	4580#	4602#	4612	4613#	4636#	4649	4650#	4651#	4663	4664#	4992#
	5005	5006#	5007#	5018	5019#	5020#	5031	5032#	5052#	5062	5063#	5064#	5074
	5075#	5076#	5086	5087#	5107#	5119	5120#	5121#	5133	5134#	5135#	5146	5147#
	5167#	5178	5179#	5180#	5191	5192#	5193#	5204	5205#	5226#	5237	5238#	5239#
	5250	5251#	5252#	5263	5264#	5285#	5296	5297#	5298#	5309	5310#	5311#	5322
	5323#	5343#	5360	5361#	5362#	5373	5374#	5375#	5386	5387#	5407#	5418	5419#
	5420#	5431	5432#	5433#	5444	5445#	5464#	5476	5477#	5478#	5490	5491#	5492#
	5503	5504#	5523#	5535	5536#	5537#	5549	5550#	5551#	5562	5563#	5582#	5593
	5594#	5595#	5607	5608#	5609#	5620	5621#	5640#	5651	5652#	5653#	5665	5666#
	5667#	5678	5679#	5700#	5712	5713#	5714#	5729	5730#	5731#	5742	5743#	5762#
	5773	5774#	5775#	5787	5788#	5789#	5800	5801#					
TSSSOF- 010117	7213#	7216											
TSSSW - 010002	3543#	3546											
TSTES- 010115	4493#	4500	4511	4513	4523#	4543	4556	4563#	4564	4588	4595#	4596	4622
	4630#	4631	4670	4679#	4707	4717	4724#	4751	4761	4769#	4795	4818	4824
	4832#	4833	4882	4915	4924#	4925	4944	4972	4985#	4987	5033	5046#	5047
	5088	5101#	5102	5148	5161#	5162	5206	5219#	5220	5265	5278#	5279	5324
	5337#	5338	5388	5401#	5402	5446	5459#	5460	5505	5518#	5519	5564	5577#
	5578	5622											

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36MACY11 30A(1052) 21-JUL-81 14:48 PAGE 38-16
(CROSS REFERENCE TABLE -- USER SYMBOLS)

SEQ 0186

T22	020376 G	3496	5635#											
T23	020654 G	3496	5695#											
T24	021130 G	3496	5757#											
T25	021404 G	3496	5813#											
T26	021576 G	3496	5873#											
T27	021744 G	3496	5945#											
T28	022322 G	3496	6104#											
T29	022560 G	3496	6202#											
T3	012422 G	3496	4563#											
T30	022774 G	3496	6248#											
T31	023210 G	3496	6291#											
T32	023452 G	3496	6348#											
T33	024044 G	3496	6429#											
T34	025116 G	3496	6605#											
T35	025216 G	3496	6644#											
T36	025364 G	3496	6697#											
T37	026042 G	3496	6833#											
T38	026202 G	3496	6882#											
T39	026310 G	3496	6912#											
T4	012552 G	3496	4595#											
T40	026446 G	3496	6958#											
T41	026544 G	3496	6994#											
T42	026644 G	3496	7028#											
T43	026770 G	3496	7077#											
T5	012712 G	3496	4630#											
T6	013154 G	3496	4679#											
T7	013356 G	3496	4724#											
T8	013570 G	3496	4769#											
T9	014112 G	3496	4832#											
UAM	= 000200 G	3573#												
VECTOR	027504	7185#												
WPM	027400	7172	7183#											
WROM	003422	3987#												
WTYPE	002414	3627#	3888	3971	3972	3992	4013	4327*	4393	4395	4400	4642	4891	4986
		5047	5102	5162	5220	5279	5338	5402	5460	5479	5519	5538	5578	5596
		5636	5654	5696	5715	5758	5776	5989	6037	6148	6323	6645	6699	6913
		6959	7029											
X\$ALWA-	000000	3434#												
X\$FALS=	000040	3434#												
X\$OFFS=	000400	3434#												
X\$TRUE=	000020	3434#												
ZERO	002370	3617#												
\$BDADR	002450	3641#												
\$BDDAT	002454	3643#												
\$GDADR	002446	3640#												
\$GDDAT	002452	3642#	4232	6304*	6305	6307	6515*	6519	6522*	6523	6541*	6545	6547	6557*
		6561	6563	6574*	6578	6580	7093*	7096	7112*	7128*	7129	7146*		
\$LSTIN-	000000	3443#												
\$LSTTA-	000000	3444#												
\$REG0	002430	3633#	4050*	4177	4178									
\$REG1	002426	3632#	4049*											
\$REG2	002424	3631#	4048*	4177	4178	4179	4185	4187	4190					
\$REG3	002422	3630#	4047*											
\$REG4	002420	3629#	4046*	4177	4178	4179	4180	4182	4184	4185	4186	4187	4188	4189
		4190	4192	4193	4198	4199								
\$REG5	002416	3628#	4045*	4179	4180	4182	4184	4185	4186	4188	4189	4192	4193	4198

.SR0MC 003100

[illegible]

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 PAGE 39
(CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0188

BADHEA	3785#	4488	4491	4517	4521	4558	4561	4590	4593	4624	4628	4673	4676	4719	4722
	4763	4767	4826	4830	4917	4922	4974	4983	5035	5044	5090	5099	5150	5159	5208
	5217	5267	5276	5326	5335	5390	5399	5448	5457	5507	5516	5566	5575	5624	5633
	5682	5692	5746	5755	5804	5811	5863	5871	5912	5943	6072	6102	6194	6200	6240
	6246	6284	6289	6339	6346	6420	6427	6597	6603	6637	6642	6690	6695	6824	6831
	6874	6880	6904	6910	6947	6956	6983	6992	7017	7026	7067	7075			
BAMPL	15#	3434#	4303	4306	4309										
BERROR	19#	3434#													
BGNAU	23#	3434#	4472												
BGNAUT	31#	3434#	4409												
BGNCLN	39#	3434#	4437												
BGNDU	47#	3434#	4454												
BGNHRD	55#	3434#	7170												
BGNHW	66#	3434#	3519												
BGNINI	77#	3434#	4287												
BGNMOD	85#	3434#	3440												
BGNMSG	98#	3434#	4177	4178	4179	4180	4182	4184	4185	4186	4187	4188	4189	4190	4192
	4193	4197	4198	4199	4201	4205	4209	4213	4218	4222	4226	4231	4236		
BGNPRO	106#	3434#	3482												
BGNPTA	114#	3434#													
BGNRPT	144#	3434#	4252												
BGNSEG	152#	3434#	4570	4602	4636	4651	4992	5007	5020	5052	5064	5076	5107	5121	5135
	5167	5180	5193	5226	5239	5252	5285	5298	5311	5343	5362	5375	5407	5420	5433
	5464	5478	5492	5523	5537	5551	5582	5595	5609	5640	5653	5667	5700	5714	5731
	5762	5775	5789												
BGNSET	161#	3434#													
BGNSFT	182#	3434#	7213												
BGNSRV	193#	3434#													
BGNSUB	201#	3434#													
BGNSW	225#	3434#	3543												
BGNTST	236#	3434#	4493	4523	4563	4595	4630	4679	4724	4769	4832	4924	4985	5046	5101
	5161	5219	5278	5337	5401	5459	5518	5577	5635	5695	5757	5813	5873	5945	6104
	6202	6248	6291	6348	6429	6605	6644	6697	6833	6882	6912	6958	6994	7028	7077
BNCOMP	266#	3434#	4312	4326											
BNFRRO	270#	3434#													
BREAK	274#	3434#	4712	4756	4819	4883									
BRESET	278#	3434#	4438	4456	6349	7098	7137								
CKLOOP	282#	3434#	7101	7118	7135										
CLOCK	286#	3434#													
CLOSE	292#	3434#													
CLRMAR	3826#	4842	4852												
CLRVEC	296#	3434#													
COMMEN	301#	3434#													
DELAY	322#	3434#													
DESCRI	317#	3434#	3595												
DEVTYP	341#	3434#	3723												
DISPAT	346#	3434#	3496												
DISPLA	360#	3434#													
DOCLN	376#	3434#													
DODU	380#	3434#	4423												
DORPT	385#	3434#													
ED\$CAL	3780#	4488	4491	4517	4521	4558	4561	4590	4593	4624	4628	4673	4676	4719	4722
	4763	4767	4826	4830	4917	4922	4974	4983	5035	5044	5090	5099	5150	5159	5208
	5217	5267	5276	5326	5335	5390	5399	5448	5457	5507	5516	5566	5575	5624	5633
	5682	5692	5746	5755	5804	5811	5863	5871	5912	5943	6072	6102	6194	6200	6240
	6246	6284	6289	6339	6346	6420	6427	6597	6603	6637	6642	6690	6695	6824	6831

	6874	6880	6904	6910	6947	6956	6983	6992	7017	7026	7067	7075				
ENDAU	389#	3434#	4473													
ENDAUT	401#	3434#	4427													
ENDCLN	413#	3434#	4440													
ENDCOM	425#	3434#														
ENDCU	441#	3434#	4457													
ENDHRD	453#	3434#	7181													
ENDHW	465#	3434#	3534													
ENDINI	475#	3434#	4406													
ENDMOD	487#	3434#														
ENDMSG	500#	3434#	4177	4178	4179	4180	4182	4184	4185	4186	4187	4188	4189	4190	4192	
	4193	4197	4198	4199	4203	4208	4212	4216	4221	4225	4230	4234	4239			
ENDPRO	512#	3434#	3486													
ENDPTA	520#	3434#														
ENDRPT	529#	3434#	4270													
ENDSEG	541#	3434#	4580	4613	4650	4664	5006	5019	5032	5063	5075	5087	5120	5134	5147	
	5179	5192	5205	5238	5251	5264	5297	5310	5323	5361	5374	5387	5419	5432	5445	
	5477	5491	5504	5536	5550	5563	5594	5608	5621	5652	5666	5679	5713	5730	5743	
	5774	5788	5801													
ENDSET	555#	3434#														
ENDSFT	568#	3434#	7216													
ENDSRV	580#	3434#														
ENDSUB	596#	3434#														
ENDSW	614#	3434#	3546													
ENDTST	624#	3434#	4513	4556	4588	4622	4670	4717	4761	4824	4915	4972	5033	5088	5148	
	5206	5265	5324	5388	5446	5505	5564	5622	5680	5744	5802	5860	5910	6071	6193	
	6238	6283	6337	6418	6595	6635	6687	6822	6872	6902	6945	6981	7015	7065	7154	
EQUALS	642#	3434#	3573													
ERRDF	714#	3434#	4507	4541	4553	4578	4611	4648	4662	4706	4750	4794	4817	4860	4878	
	4913	4943	4958	4970	5004	5017	5030	5061	5073	5085	5117	5131	5145	5177	5190	
	5203	5236	5249	5262	5295	5308	5321	5358	5372	5385	5417	5430	5443	5475	5489	
	5502	5534	5548	5561	5592	5606	5619	5650	5664	5677	5711	5727	5741	5772	5786	
	5799	5850	5899	5996	6068	6155	6214	6226	6236	6260	6271	6281	6308	6320	6332	
	6364	6382	6391	6402	6411	6524	6548	6564	6581	6632	6674	6747	6756	6765	6773	
	6783	6789	6795	6801	6869	6890	6898	6936	6977	7012	7061	7099	7116	7132	7148	

CZDMQCO M8207 STATIC DIAG #2
CZDMQC.P11 21-JUL-81 14:36

MACY11 30A(1052) 21-JUL-81 14:48 PAGE 39-2
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0190

GETBYT	824#	3434#														
GETPRI	834#	3434#														
GETWUR	829#	3434#														
GMAVIA	839#	3434#														
GMAVID	848#	3434#														
GMAVIL	859#	3434#														
GPHARD	868#	3434#	4325													
GPRMA	874#	3434#	7173													
GPRMD	903#	3434#	7172	7180												
GPRML	934#	3434#														
HEADER	954#	3434#	3474													
INLOUP	962#	3434#														
IOSETU	966#	3434#														
IOSTAR	974#	3434#														
KT11	982#	3434#														
K4ONLY	3775#	3815#	4833	5814	5874	5947	6106	6203	6249							
LASTAD	1147#	3434#	7225													
MACEX	3797#	4564	4596	4631	4925											
MACEX2	3806#	5047	5102	5162	5220	5279	5338	5402	5460	5519	5578	5636	5696	5758		
MANUAL	1162#	3434#														
MDTO	4145#	4197														
MDT1	4125#	4177	4178													
MDT2	4129#	4179														
MDT3	4133#	4180	4182	4186	4189	4198	4199									
MDT4	4137#	4184														
MDT5	4141#	4185														
MDT6	4147#	4187	4190													
MDT7	4151#	4188	4193													
MDT8	4155#	4192														
MEMORY	1166#	3434#														
MSCLR	3862#	4525	4598	4634	4682	4727	4772	4836	4928	4991	5050	5105	5165	5223	5282	
	5340	5405	5462	5521	5580	5638	5698	5760	5816	5876	5948	6107	6204	6250	6293	
	6352	6363	6401	6410	6416	6431	6917	6963	7033	7079						
MYINT	3791#	4565	4597	4632	4680	4725	4770	4834	4926	4989	5048	5103	5163	5221	5280	
	5339	5403	5461	5520	5579	5637	5697	5759	5815	5875	5946	6105	6205	6251	6292	
	6350	6430	6606	6648	6702	6834	6883	6916	6962	6995	7032	7078				
MSBYTE	2000#	3434#	3474#													
MSCHEC	2118#	3434#	4260#	4564#	4596#	4631#	4833#	4925#	4987#	5047#	5102#	5162#	5220#	5279#	5338#	
	5402#	5460#	5519#	5578#	5636#	5696#	5758#	5814#	5874#	5947#	6106#	6203#	6249#	6395#	6586#	
	6646#	6681#	6700#	6803#	6852#	6914#	6960#	7030#								
MSCNTO	2182#	3434#	7172#	7173#	7180#											
MSCOUN	2066#	3434#	4177#	4178#	4179#	4180#	4182#	4184#	4185#	4186#	4187#	4188#	4189#	4190#	4192#	
	4193#	4197#	4198#	4199#	4202#	4206#	4207#	4210#	4211#	4214#	4215#	4219#	4220#	4223#	4224#	
	4227#	4228#	4229#	4232#	4233#	4237#	4238#									
MSDATA	1867#	3434#	3474#	3595#	3723#											
MSDECR	2029#	3434#	3486#	3534#	3546#	4177#	4178#	4179#	4180#	4182#	4184#	4185#	4186#	4187#	4189#	
	4189#	4190#	4192#	4193#	4197#	4198#	4199#	4203#	4208#	4212#	4216#	4221#	4225#	4230#	4234#	
	4239#	4270#	4406#	4427#	4440#	4457#	4473#	4513#	4556#	4580#	4588#	4613#	4622#	4650#	4664#	
	4670#	4717#	4761#	4824#	4915#	4972#	5006#	5019#	5032#	5033#	5063#	5075#	5087#	5088#	5120#	
	5134#	5147#	5148#	5179#	5192#	5205#	5206#	5238#	5251#	5264#	5265#	5297#	5310#	5323#	5324#	
	5361#	5374#	5387#	5388#	5419#	5432#	5445#	5446#	5477#	5491#	5504#	5505#	5536#	5550#	5563#	
	5564#	5594#	5608#	5621#	5622#	5652#	5666#	5679#	5680#	5713#	5730#	5743#	5744#	5774#	5788#	
	5801#	5802#	5860#	5910#	6071#	6193#	6238#	6283#	6337#	6418#	6595#	6635#	6687#	6822#	6872#	
	6902#	6945#	6981#	7015#	7065#	7154#	7181#	7216#								
MSDEFA	2170#	3434#	7172#	7173#	7180#											
MSENDE	2074#	3434#	3534#	3546#	4177#	4178#	4179#	4180#	4182#	4184#	4185#	4186#	4187#	4188#	4189#	

	4190#	4192#	4193#	4197#	4198#	4199#	4203#	4208#	4212#	4216#	4221#	4225#	4230#	4234#	4239#
	4270#	4406#	4427#	4440#	4457#	4473#	4513#	4556#	4580#	4588#	4613#	4622#	4650#	4664#	4670#
	4717#	4761#	4824#	4915#	4972#	5006#	5019#	5032#	5033#	5063#	5075#	5087#	5088#	5120#	5134#
	5147#	5148#	5179#	5192#	5205#	5206#	5238#	5251#	5264#	5265#	5297#	5310#	5323#	5324#	5361#
	5374#	5387#	5388#	5419#	5432#	5445#	5446#	5477#	5491#	5504#	5505#	5536#	5550#	5563#	5564#
	5594#	5608#	5621#	5622#	5652#	5666#	5679#	5680#	5713#	5730#	5743#	5744#	5774#	5788#	5801#
	5802#	5860#	5910#	6071#	6193#	6238#	6283#	6337#	6418#	6595#	6635#	6687#	6822#	6872#	6902#
	6945#	6981#	7015#	7065#	7154#	7181#	7216#								
MSERRI	1649#	3434#	4507#	4541#	4553#	4578#	4611#	4648#	4662#	4706#	4750#	4794#	4817#	4860#	4878#
	4913#	4943#	4958#	4970#	5004#	5017#	5030#	5061#	5073#	5085#	5117#	5131#	5145#	5177#	5190#
	5203#	5236#	5249#	5262#	5295#	5308#	5321#	5358#	5372#	5385#	5417#	5430#	5443#	5475#	5489#
	5502#	5534#	5548#	5561#	5592#	5606#	5619#	5650#	5664#	5677#	5711#	5727#	5741#	5772#	5786#
	5799#	5850#	5899#	5996#	6068#	6155#	6214#	6226#	6236#	6260#	6271#	6281#	6308#	6320#	6332#
	6364#	6382#	6391#	6402#	6411#	6524#	6548#	6564#	6581#	6632#	6674#	6747#	6756#	6765#	6773#
MSERCA	6783#	6789#	6795#	6801#	6869#	6890#	6898#	6936#	6977#	7012#	7061#	7099#	7116#	7132#	7148#
	2006#	3434#	4500#	4511#	4543#	4579#	4612#	4649#	4663#	4707#	4751#	4795#	4818#	4882#	4944#
	5005#	5018#	5031#	5062#	5074#	5086#	5119#	5133#	5146#	5178#	5191#	5204#	5237#	5250#	5263#
	5296#	5309#	5322#	5360#	5373#	5386#	5418#	5431#	5444#	5476#	5490#	5503#	5535#	5549#	5562#
	5593#	5607#	5620#	5651#	5665#	5678#	5712#	5729#	5742#	5773#	5787#	5800#	5855#	5903#	6027#
	6186#	6216#	6228#	6262#	6273#	6525#	6565#	6582#	6677#	6749#	6759#	6767#	6775#	6785#	6790#
	6796#	6891#	6939#												
MSERCS	2010#	3434#	4500#	4511#	4543#	4579#	4612#	4649#	4663#	4707#	4751#	4795#	4818#	4882#	4944#
	5005#	5018#	5031#	5062#	5074#	5086#	5119#	5133#	5146#	5178#	5191#	5204#	5237#	5250#	5263#
	5296#	5309#	5322#	5360#	5373#	5386#	5418#	5431#	5444#	5476#	5490#	5503#	5535#	5549#	5562#
	5593#	5607#	5620#	5651#	5665#	5678#	5712#	5729#	5742#	5773#	5787#	5800#	5855#	5903#	6027#
	6186#	6216#	6228#	6262#	6273#	6525#	6565#	6582#	6677#	6749#	6759#	6767#	6775#	6785#	6790#
	6796#	6891#	6939#												
MSEXCP	2101#	3434#	7172#	7173#	7180#										
MSEXIT	2014#	3434#	4260#	4564#	4596#	4631#	4833#	4925#	4987#	5047#	5102#	5162#	5220#	5279#	5338#
	5402#	5460#	5519#	5578#	5636#	5696#	5758#	5814#	5874#	5947#	6106#	6203#	6249#	6395#	6586#
	6646#	6681#	6700#	6803#	6852#	6914#	6960#	7030#							
MSXSE	2022#	3434#	4260#	4564#	4596#	4631#	4833#	4925#	4987#	5047#	5102#	5162#	5220#	5279#	5338#
	5402#	5460#	5519#	5578#	5636#	5696#	5758#	5814#	5874#	5947#	6106#	6203#	6249#	6395#	6586#
	6646#	6681#	6700#	6803#	6852#	6914#	6960#	7030#							
MSXTJ	2018#	3434#	4260#	4564#	4596#	4631#	4833#	4925#	4987#	5047#	5102#	5162#	5220#	5279#	5338#
	5402#	5460#	5519#	5578#	5636#	5696#	5758#	5814#	5874#	5947#	6106#	6203#	6249#	6395#	6586#
	6646#	6681#	6700#	6803#	6852#	6914#	6960#	7030#							
MSGEN	2038#	3434#	3440#	3474#	3482#	3496#	3519#	3534#	3543#	3546#	3595#	3723#	4177#	4178#	4179#
	4180#	4182#	4184#	4185#	4186#	4187#	4188#	4189#	4190#	4192#	4193#	4197#	4198#	4199#	4201#
	4203#	4205#	4208#	4209#	4212#	4213#	4216#	4218#	4221#	4222#	4225#	4226#	4230#	4231#	4234#
	4236#	4239#	4252#	4270#	4287#	4406#	4409#	4427#	4437#	4440#	4454#	4457#	4472#	4473#	4493#
	4513#	4523#	4556#	4563#	4580#	4588#	4595#	4613#	4622#	4630#	4650#	4664#	4670#	4679#	4717#
	4724#	4761#	4769#	4824#	4832#	4915#	4924#	4972#	4985#	5006#	5019#	5032#	5033#	5046#	5063#
	5075#	5087#	5088#	5101#	5120#	5134#	5147#	5148#	5161#	5179#	5192#	5205#	5206#	5219#	5238#
	5251#	5264#	5265#	5278#	5297#	5310#	5323#	5324#	5337#	5361#	5374#	5387#	5388#	5401#	5419#
	5432#	5445#	5446#	5459#	5477#	5491#	5504#	5505#	5518#	5536#	5550#	5563#	5564#	5577#	5594#
	5608#	5621#	5622#	5635#	5652#	5666#	5679#	5680#	5695#	5713#	5730#	5743#	5744#	5757#	5774#
	5788#	5801#	5802#	5813#	5860#	5873#	5910#	5945#	6071#	6104#	6193#	6202#	6238#	6248#	6283#
	6291#	6337#	6348#	6418#	6429#	6595#	6605#	6635#	6644#	6687#	6697#	6822#	6833#	6872#	6882#
	6902#	6912#	6945#	6958#	6981#	6994#	7015#	7028#	7065#	7077#	7154#	7170#	7181#	7213#	7216#
	7225#														
MSGENB	1938#	3434#													
MSGETS	2035#	3434#	3486#	3534#	3546#	4177#	4178#	4179#	4180#	4182#	4184#	4185#	4186#	4187#	4188#
	4189#	4190#	4192#	4193#	4197#	4198#	4199#	4203#	4208#	4212#	4216#	4221#	4225#	4230#	4234#
	4239#	4270#	4406#	4427#	4440#	4457#	4473#	4513#	4556#	4579#	4580#	4588#	4612#	4613#	4622#
	4649#	4650#	4663#	4664#	4670#	4717#	4761#	4824#	4915#	4972#	5005#	5006#	5018#	5019#	5031#

	5032#	5033#	5062#	5063#	5074#	5075#	5086#	5087#	5088#	5119#	5120#	5133#	5134#	5146#	5147#
	5148#	5178#	5179#	5191#	5192#	5204#	5205#	5206#	5237#	5238#	5250#	5251#	5263#	5264#	5265#
	5296#	5297#	5309#	5310#	5322#	5323#	5324#	5360#	5361#	5373#	5374#	5386#	5387#	5388#	5418#
	5419#	5431#	5432#	5444#	5445#	5446#	5476#	5477#	5490#	5491#	5503#	5504#	5505#	5535#	5536#
	5549#	5550#	5562#	5563#	5564#	5593#	5594#	5607#	5608#	5620#	5621#	5622#	5651#	5652#	5665#
	5666#	5678#	5679#	5680#	5712#	5713#	5729#	5730#	5742#	5743#	5744#	5773#	5774#	5787#	5788#
	5800#	5801#	5802#	5860#	5910#	6071#	6193#	6238#	6283#	6337#	6418#	6595#	6635#	6687#	6822#
	6872#	6902#	6945#	6981#	7015#	7065#	7154#	7181#	7216#						
MSGETT	1877#	3434#	4260#	4500#	4511#	4543#	4564#	4579#	4596#	4612#	4631#	4649#	4663#	4707#	4751#
	4795#	4818#	4833#	4882#	4925#	4944#	4987#	5005#	5018#	5031#	5047#	5062#	5074#	5086#	5102#
	5119#	5133#	5146#	5162#	5178#	5191#	5204#	5220#	5237#	5250#	5263#	5279#	5296#	5309#	5322#
	5338#	5360#	5373#	5386#	5402#	5418#	5431#	5444#	5460#	5476#	5490#	5503#	5519#	5535#	5549#
	5562#	5578#	5593#	5607#	5620#	5636#	5651#	5665#	5678#	5696#	5712#	5729#	5742#	5758#	5773#
	5787#	5800#	5814#	5855#	5874#	5903#	5947#	6027#	6106#	6186#	6203#	6216#	6228#	6249#	6262#
	6273#	6395#	6525#	6565#	6582#	6586#	6646#	6677#	6681#	6700#	6749#	6759#	6767#	6775#	6785#
	6790#	6796#	6803#	6852#	6891#	6914#	6939#	6960#	7030#						
MSGNGB	1902#	3434#	3440#	3474#	3482#	3496#	3519#	3543#	3595#	3723#	4177#	4178#	4179#	4180#	4182#
	4184#	4185#	4186#	4187#	4188#	4189#	4190#	4192#	4193#	4197#	4198#	4199#	4201#	4205#	4209#
	4213#	4218#	4222#	4226#	4231#	4236#	4252#	4287#	4409#	4437#	4454#	4472#	7170#	7213#	7225#
MSGNIN	2049#	3434#	3474#	3496#	3519#	3543#	3595#	3723#	4177#	4178#	4179#	4180#	4182#	4184#	4185#
	4186#	4187#	4188#	4189#	4190#	4192#	4193#	4197#	4198#	4199#	4202#	4203#	4206#	4207#	4208#
	4210#	4211#	4212#	4214#	4215#	4216#	4219#	4220#	4221#	4223#	4224#	4225#	4227#	4228#	4229#
	4230#	4232#	4233#	4234#	4237#	4238#	4239#	4260#	4270#	4302#	4303#	4305#	4306#	4308#	4309#
	4311#	4312#	4325#	4326#	4406#	4423#	4427#	4438#	4440#	4456#	4457#	4473#	4500#	4507#	4511#
	4513#	4541#	4543#	4553#	4556#	4564#	4570#	4578#	4579#	4580#	4588#	4596#	4602#	4611#	4612#
	4613#	4622#	4631#	4636#	4648#	4649#	4650#	4651#	4662#	4663#	4664#	4670#	4706#	4707#	4712#
	4717#	4750#	4751#	4756#	4761#	4794#	4795#	4817#	4818#	4819#	4824#	4833#	4860#	4878#	4882#
	4883#	4913#	4915#	4925#	4943#	4944#	4958#	4970#	4972#	4987#	4992#	5004#	5005#	5006#	5007#
	5017#	5018#	5019#	5020#	5030#	5031#	5032#	5033#	5047#	5052#	5061#	5062#	5063#	5064#	5073#
	5074#	5075#	5076#	5085#	5086#	5087#	5088#	5102#	5107#	5117#	5119#	5120#	5121#	5131#	5133#
	5134#	5135#	5145#	5146#	5147#	5148#	5162#	5167#	5177#	5178#	5179#	5180#	5190#	5191#	5192#
	5193#	5203#	5204#	5205#	5206#	5220#	5226#	5236#	5237#	5238#	5239#	5249#	5250#	5251#	5252#
	5262#	5263#	5264#	5265#	5279#	5285#	5295#	5296#	5297#	5298#	5308#	5309#	5310#	5311#	5321#
	5322#	5323#	5324#	5338#	5343#	5358#	5360#	5361#	5362#	5372#	5373#	5374#	5375#	5385#	5386#
	5387#	5388#	5402#	5407#	5417#	5418#	5419#	5420#	5430#	5431#	5432#	5433#	5443#	5444#	5445#
	5446#	5460#	5464#	5475#	5476#	5477#	5478#	5489#	5490#	5491#	5492#	5502#	5503#	5504#	5505#
	5519#	5523#	5534#	5535#	5536#	5537#	5548#	5549#	5550#	5551#	5561#	5562#	5563#	5564#	5578#
	5582#	5592#	5593#	5594#	5595#	5606#	5607#	5608#	5609#	5619#	5620#	5621#	5622#	5636#	5640#
	5650#	5651#	5652#	5653#	5664#	5665#	5666#	5667#	5677#	5678#	5679#	5680#	5696#	5700#	5711#
	5712#	5713#	5714#	5727#	5729#	5730#	5731#	5741#	5742#	5743#	5744#	5758#	5762#	5772#	5773#
	5774#	5775#	5786#	5787#	5788#	5789#	5799#	5800#	5801#	5802#	5814#	5850#	5855#	5860#	5874#
	5899#	5903#	5910#	5947#	5996#	6027#	6068#	6071#	6106#	6155#	6186#	6193#	6203#	6214#	6216#
	6226#	6228#	6236#	6238#	6249#	6260#	6262#	6271#	6273#	6281#	6283#	6308#	6320#	6332#	6337#
	6349#	6364#	6382#	6391#	6395#	6402#	6411#	6418#	6524#	6525#	6548#	6564#	6565#	6581#	6582#
	6586#	6595#	6632#	6635#	6646#	6674#	6677#	6681#	6687#	6700#	6747#	6749#	6756#	6759#	6765#
	6767#	6773#	6775#	6783#	6785#	6789#	6790#	6795#	6796#	6801#	6803#	6822#	6852#	6869#	6872#
	6890#	6891#	6898#	6902#	6914#	6936#	6939#	6945#	6960#	6977#	6981#	7012#	7015#	7030#	7061#
	7065#	7080#	7098#	7099#	7101#	7116#	7118#	7132#	7135#	7137#	7148#	7154#	7170#	7172#	7173#
	7180#	7181#	7213#	7216#	7225#										
MSGNLS	1913#	3434#	4580#	4613#	4650#	4664#	5006#	5019#	5032#	5063#	5075#	5087#	5120#	5134#	5147#
	5179#	5192#	5205#	5238#	5251#	5264#	5297#	5310#	5323#	5361#	5374#	5387#	5419#	5432#	5445#
	5477#	5491#	5504#	5536#	5550#	5563#	5594#	5608#	5621#	5652#	5666#	5670#	5713#	5730#	5743#
	5774#	5788#	5801#												
MSGNSU	1898#	3434#													
MSGNTA	1890#	3434#	3534#	3546#	4177#	4178#	4179#	4180#	4182#	4184#	4185#	4186#	4187#	4188#	4189#
	4190#	4192#	4193#	4197#	4198#	4199#	4203#	4208#	4212#	4216#	4221#	4225#	4230#	4234#	4239#

	4270#	4406#	4427#	4440#	4457#	4473#	4513#	4556#	4588#	4622#	4670#	4717#	4761#	4824#	4915#
	4972#	5033#	5088#	5148#	5206#	5265#	5324#	5388#	5446#	5505#	5564#	5622#	5680#	5744#	5802#
	5860#	5910#	6071#	6193#	6238#	6283#	6337#	6418#	6595#	6635#	6687#	6822#	6872#	6902#	6945#
MSGNTE	6981#	7015#	7065#	7154#	7181#	7216#									
	1894#	3434#	4493#	4523#	4563#	4595#	4630#	4679#	4724#	4769#	4832#	4924#	4985#	5046#	5101#
	5161#	5219#	5278#	5337#	5401#	5459#	5518#	5577#	5635#	5695#	5757#	5813#	5873#	5945#	6104#
	6202#	6248#	6291#	6348#	6429#	6605#	6644#	6697#	6833#	6882#	6912#	6958#	6994#	7028#	7077#
MSHAPT	1739#	3434#	3474#												
MSHNAP	1824#	3434#	3474#												
MSINCR	2026#	3434#	3440#	3482#	3519#	3543#	4177#	4178#	4179#	4180#	4182#	4184#	4185#	4186#	4187#
	4188#	4189#	4190#	4192#	4193#	4197#	4198#	4199#	4201#	4202#	4203#	4205#	4206#	4207#	4208#
	4209#	4210#	4211#	4212#	4213#	4214#	4215#	4216#	4218#	4219#	4220#	4221#	4222#	4223#	4224#
	4225#	4226#	4227#	4228#	4229#	4230#	4231#	4232#	4233#	4234#	4236#	4237#	4238#	4239#	4252#
	4270#	4287#	4302#	4305#	4308#	4311#	4325#	4406#	4409#	4423#	4427#	4437#	4438#	4440#	4454#
	4456#	4457#	4472#	4473#	4493#	4500#	4507#	4511#	4513#	4523#	4541#	4543#	4553#	4556#	4563#
	4564#	4570#	4578#	4579#	4580#	4588#	4595#	4596#	4602#	4611#	4612#	4613#	4622#	4630#	4631#
	4636#	4648#	4649#	4650#	4651#	4662#	4663#	4664#	4670#	4679#	4706#	4707#	4712#	4717#	4724#
	4750#	4751#	4756#	4761#	4769#	4794#	4795#	4817#	4818#	4819#	4824#	4832#	4833#	4860#	4878#
	4882#	4883#	4913#	4915#	4924#	4925#	4943#	4944#	4958#	4970#	4972#	4985#	4987#	4992#	5004#
	5005#	5006#	5007#	5017#	5018#	5019#	5020#	5030#	5031#	5032#	5033#	5046#	5047#	5052#	5061#
	5062#	5063#	5064#	5073#	5074#	5075#	5076#	5085#	5086#	5087#	5088#	5101#	5102#	5107#	5117#
	5119#	5120#	5121#	5131#	5133#	5134#	5135#	5145#	5146#	5147#	5148#	5161#	5162#	5167#	5177#
	5178#	5179#	5180#	5190#	5191#	5192#	5193#	5203#	5204#	5205#	5206#	5219#	5220#	5226#	5236#
	5237#	5238#	5239#	5249#	5250#	5251#	5252#	5262#	5263#	5264#	5265#	5278#	5279#	5285#	5295#
	5296#	5297#	5298#	5308#	5309#	5310#	5311#	5321#	5322#	5323#	5324#	5337#	5338#	5343#	5358#
	5360#	5361#	5362#	5372#	5373#	5374#	5375#	5385#	5386#	5387#	5388#	5401#	5402#	5407#	5417#
	5418#	5419#	5420#	5430#	5431#	5432#	5433#	5443#	5444#	5445#	5446#	5459#	5460#	5464#	5475#
	5476#	5477#	5478#	5489#	5490#	5491#	5492#	5502#	5503#	5504#	5505#	5518#	5519#	5523#	5534#
	5535#	5536#	5537#	5548#	5549#	5550#	5551#	5561#	5562#	5563#	5564#	5577#	5578#	5582#	5592#
	5593#	5594#	5595#	5606#	5607#	5608#	5609#	5619#	5620#	5621#	5622#	5635#	5636#	5640#	5650#
	5651#	5652#	5653#	5664#	5665#	5666#	5667#	5677#	5678#	5679#	5680#	5695#	5696#	5700#	5711#
	5712#	5713#	5714#	5727#	5729#	5730#	5731#	5741#	5742#	5743#	5744#	5757#	5758#	5762#	5772#
	5773#	5774#	5775#	5786#	5787#	5788#	5789#	5799#	5800#	5801#	5802#	5813#	5814#	5850#	5855#
	5860#	5873#	5874#	5899#	5903#	5910#	5945#	5947#	5996#	6027#	6068#	6071#	6104#	6106#	6155#
	6186#	6193#	6202#	6203#	6214#	6216#	6226#	6228#	6236#	6238#	6248#	6249#	6260#	6262#	6271#
	6273#	6281#	6283#	6291#	6308#	6320#	6332#	6337#	6348#	6349#	6364#	6382#	6391#	6395#	6402#
	6411#	6418#	6429#	6524#	6525#	6548#	6564#	6565#	6581#	6582#	6586#	6595#	6605#	6632#	6635#
	6644#	6646#	6674#	6677#	6681#	6687#	6697#	6700#	6747#	6749#	6756#	6759#	6765#	6767#	6773#
	6775#	6783#	6785#	6789#	6790#	6795#	6796#	6801#	6803#	6822#	6833#	6852#	6869#	6872#	6882#
	6890#	6891#	6898#	6902#	6912#	6914#	6936#	6939#	6945#	6958#	6960#	6977#	6981#	6994#	7012#
	7015#	7028#	7030#	7061#	7065#	7077#	7080#	7098#	7099#	7101#	7116#	7118#	7132#	7135#	7137#
	7148#	7154#	7170#	7213#											
MSIOSE	1700#	3434#													
MSLDRO	1942#	3434#	4302#	4305#	4308#	4311#	4325#	4423#	7080#						
MSMASK	1671#	3434#													
MSMCHI	4#	3434#													
MSMCLO	1624#	3434#													
MSMSK1	1677#	3434#													
MSPOP	1881#	3434#	3486#	3534#	3546#	4177#	4178#	4179#	4180#	4182#	4184#	4185#	4186#	4187#	4188#
	4189#	4190#	4192#	4193#	4197#	4198#	4199#	4203#	4208#	4212#	4216#	4221#	4225#	4230#	4234#
	4239#	4270#	4406#	4427#	4440#	4457#	4473#	4513#	4556#	4580#	4588#	4613#	4622#	4650#	4664#
	4670#	4717#	4761#	4824#	4915#	4972#	5006#	5019#	5032#	5033#	5063#	5075#	5087#	5088#	5120#
	5134#	5147#	5148#	5179#	5192#	5205#	5206#	5238#	5251#	5264#	5265#	5297#	5310#	5323#	5324#
	5361#	5374#	5387#	5388#	5419#	5432#	5445#	5446#	5477#	5491#	5504#	5505#	5536#	5550#	5563#
	5564#	5594#	5608#	5621#	5622#	5652#	5666#	5679#	5680#	5713#	5730#	5743#	5744#	5774#	5788#
	5801#	5802#	5860#	5910#	6071#	6193#	6238#	6283#	6337#	6418#	6595#	6635#	6687#	6822#	6872#

MSPRIN	6902#	6945#	6981#	7015#	7065#	7154#	7181#	7216#							
	1636#	3434#	4177#	4178#	4179#	4180#	4182#	4184#	4185#	4186#	4187#	4188#	4189#	4190#	4192#
	4193#	4197#	4198#	4199#	4202#	4206#	4207#	4210#	4211#	4214#	4215#	4219#	4220#	4223#	4224#
MSPUSH	4227#	4228#	4229#	4232#	4233#	4237#	4238#								
	1631#	3434#	3440#	3482#	3519#	3543#	4177#	4178#	4179#	4180#	4182#	4184#	4185#	4186#	4187#
	4188#	4189#	4190#	4192#	4193#	4197#	4198#	4199#	4201#	4205#	4209#	4213#	4218#	4222#	4226#
	4231#	4236#	4252#	4287#	4409#	4437#	4454#	4472#	4493#	4523#	4563#	4570#	4595#	4602#	4630#
	4636#	4651#	4679#	4724#	4769#	4832#	4924#	4985#	4992#	5007#	5020#	5046#	5052#	5064#	5076#
	5101#	5107#	5121#	5135#	5161#	5167#	5180#	5193#	5219#	5226#	5239#	5252#	5278#	5285#	5298#
	5311#	5337#	5343#	5362#	5375#	5401#	5407#	5420#	5433#	5459#	5464#	5478#	5492#	5518#	5523#
	5537#	5551#	5577#	5582#	5595#	5609#	5635#	5640#	5653#	5667#	5695#	5700#	5714#	5731#	5757#
	5762#	5775#	5789#	5813#	5873#	5945#	6104#	6202#	6248#	6291#	6348#	6429#	6605#	6644#	6697#
MSPUT	6833#	6882#	6912#	6958#	6994#	7028#	7077#	7170#	7213#						
	1972#	3434#	4177#	4178#	4179#	4180#	4182#	4184#	4185#	4186#	4187#	4188#	4189#	4190#	4192#
	4193#	4197#	4198#	4199#	4202#	4206#	4207#	4210#	4211#	4214#	4215#	4219#	4220#	4223#	4224#
MSPUT1	4227#	4228#	4229#	4232#	4233#	4237#	4238#								
	1981#	3434#	4177#	4178#	4179#	4180#	4182#	4184#	4185#	4186#	4187#	4188#	4189#	4190#	4192#
	4193#	4197#	4198#	4199#	4202#	4206#	4207#	4210#	4211#	4214#	4215#	4219#	4220#	4223#	4224#
	4227#	4228#	4229#	4232#	4233#	4237#	4238#								
MSRADI	2077#	3434#	7172#	7173#	7180#										
MSRBRO	1952#	3434#													
MSRNRO	1962#	3434#	4325#												
MSSETS	2032#	3434#	3440#	3482#	3519#	3543#	4177#	4178#	4179#	4180#	4182#	4184#	4185#	4186#	4187#
	4188#	4189#	4190#	4192#	4193#	4197#	4198#	4199#	4201#	4205#	4209#	4213#	4218#	4222#	4226#
	4231#	4236#	4252#	4287#	4409#	4437#	4454#	4472#	4493#	4523#	4563#	4570#	4595#	4602#	4630#
	4636#	4651#	4679#	4724#	4769#	4832#	4924#	4985#	4992#	5007#	5020#	5046#	5052#	5064#	5076#
	5101#	5107#	5121#	5135#	5161#	5167#	5180#	5193#	5219#	5226#	5239#	5252#	5278#	5285#	5298#
	5311#	5337#	5343#	5362#	5375#	5401#	5407#	5420#	5433#	5459#	5464#	5478#	5492#	5518#	5523#
	5537#	5551#	5577#	5582#	5595#	5609#	5635#	5640#	5653#	5667#	5695#	5700#	5714#	5731#	5757#
	5762#	5775#	5789#	5813#	5873#	5945#	6104#	6202#	6248#	6291#	6348#	6429#	6605#	6644#	6697#
	6833#	6882#	6912#	6958#	6994#	7028#	7077#	7170#	7213#						
MSSTAR	1733#	3434#													
MS SVC	1933#	3434#	4177#	4178#	4179#	4180#	4182#	4184#	4185#	4186#	4187#	4188#	4189#	4190#	4192#
	4193#	4197#	4198#	4199#	4202#	4203#	4206#	4207#	4208#	4210#	4211#	4212#	4214#	4215#	4216#
	4219#	4220#	4221#	4223#	4224#	4225#	4227#	4228#	4229#	4230#	4232#	4233#	4234#	4237#	4238#
	4239#	4260#	4270#	4302#	4305#	4308#	4311#	4325#	4406#	4423#	4427#	4438#	4440#	4456#	4457#
	4473#	4500#	4507	4511#	4513#	4541	4543#	4553	4556#	4564#	4570#	4578	4579#	4580#	4588#
	4596#	4602#	4611	4612#	4613#	4622#	4631#	4636#	4648	4649#	4650#	4651#	4662	4663#	4664#
	4670#	4706	4707#	4712#	4717#	4750	4751#	4756#	4761#	4794	4795#	4817	4818#	4819#	4824#
	4833#	4860	4878	4882#	4883#	4913	4915#	4925#	4943	4944#	4958	4970	4972#	4987#	4992#
	5004	5005#	5006#	5007#	5017	5018#	5019#	5020#	5030	5031#	5032#	5033#	5047#	5052#	5061
	5062#	5063#	5064#	5073	5074#	5075#	5076#	5085	5086#	5087#	5088#	5102#	5107#	5117	5119#
	5120#	5121#	5131	5133#	5134#	5135#	5145	5146#	5147#	5148#	5162#	5167#	5177	5178#	5179#
	5180#	5190	5191#	5192#	5193#	5203	5204#	5205#	5206#	5220#	5226#	5236	5237#	5238#	5239#
	5249	5250#	5251#	5252#	5262	5263#	5264#	5265#	5279#	5285#	5295	5296#	5297#	5298#	5308
	5309#	5310#	5311#	5321	5322#	5323#	5324#	5338#	5343#	5358	5360#	5361#	5362#	5372	5373#
	5374#	5375#	5385	5386#	5387#	5388#	5402#	5407#	5417	5418#	5419#	5420#	5430	5431#	5432#
	5433#	5443	5444#	5445#	5446#	5460#	5464#	5475	5476#	5477#	5478#	5489	5490#	5491#	5492#
	5502	5503#	5504#	5505#	5519#	5523#	5534	5535#	5536#	5537#	5548	5549#	5550#	5551#	5561
	5562#	5563#	5564#	5578#	5582#	5592	5593#	5594#	5595#	5606	5607#	5608#	5609#	5619	5620#
	5621#	5622#	5636#	5640#	5650	5651#	5652#	5653#	5664	5665#	5666#	5667#	5677	5678#	5679#
	5680#	5696#	5700#	5711	5712#	5713#	5714#	5727	5729#	5730#	5731#	5741	5742#	5743#	5744#
	5758#	5762#	5772	5773#	5774#	5775#	5786	5787#	5788#	5789#	5799	5800#	5801#	5802#	5814#
	5850	5855#	5860#	5874#	5899	5903#	5910#	5947#	5996	6027#	6068	6071#	6106#	6155	6186#
	6193#	6203#	6214	6216#	6226	6228#	6236	6238#	6249#	6260	6262#	6271	6273#	6281	6283#
	6308	6320	6332	6337#	6349#	6364	6382	6391	6395#	6402	6411	6418#	6524	6525#	6548

	6564#	6565#	6581#	6582#	6586#	6595#	6632#	6635#	6646#	6674#	6677#	6681#	6687#	6700#	6747#
	6749#	6756#	6759#	6765#	6767#	6773#	6775#	6783#	6785#	6789#	6790#	6795#	6796#	6801#	6803#
	6822#	6852#	6869#	6872#	6890#	6891#	6898#	6902#	6914#	6936#	6939#	6945#	6960#	6977#	6981#
	7012#	7015#	7030#	7061#	7065#	7080#	7098#	7099#	7101#	7116#	7118#	7132#	7135#	7137#	7148#
	7154#														
MSTLAB	1929#	3434#	4177#	4178#	4179#	4180#	4182#	4184#	4185#	4186#	4187#	4188#	4189#	4190#	4192#
	4193#	4197#	4198#	4199#	4202#	4203#	4206#	4207#	4208#	4210#	4211#	4212#	4214#	4215#	4216#
	4219#	4220#	4221#	4223#	4224#	4225#	4227#	4228#	4229#	4230#	4232#	4233#	4234#	4237#	4238#
	4239#	4270#	4302#	4305#	4308#	4311#	4325#	4406#	4423#	4427#	4438#	4440#	4456#	4457#	4473#
	4500#	4507#	4511#	4513#	4541#	4543#	4553#	4556#	4564#	4570#	4578#	4579#	4580#	4588#	4596#
	4602#	4611#	4612#	4613#	4622#	4631#	4636#	4648#	4649#	4650#	4651#	4662#	4663#	4664#	4670#
	4706#	4707#	4712#	4717#	4750#	4751#	4756#	4761#	4794#	4795#	4817#	4818#	4819#	4824#	4833#
	4860#	4878#	4882#	4883#	4913#	4915#	4925#	4943#	4944#	4958#	4970#	4972#	4987#	4992#	5004#
	5005#	5006#	5007#	5017#	5018#	5019#	5020#	5030#	5031#	5032#	5033#	5047#	5052#	5061#	5062#
	5063#	5064#	5073#	5074#	5075#	5076#	5085#	5086#	5087#	5088#	5102#	5107#	5117#	5119#	5120#
	5121#	5131#	5133#	5134#	5135#	5145#	5146#	5147#	5148#	5162#	5167#	5177#	5178#	5179#	5180#
	5190#	5191#	5192#	5193#	5203#	5204#	5205#	5206#	5220#	5226#	5236#	5237#	5238#	5239#	5249#
	5250#	5251#	5252#	5262#	5263#	5264#	5265#	5279#	5285#	5295#	5296#	5297#	5298#	5308#	5309#
	5310#	5311#	5321#	5322#	5323#	5324#	5338#	5343#	5358#	5360#	5361#	5362#	5372#	5373#	5374#
	5375#	5385#	5386#	5387#	5388#	5402#	5407#	5417#	5418#	5419#	5420#	5430#	5431#	5432#	5433#
	5443#	5444#	5445#	5446#	5460#	5464#	5475#	5476#	5477#	5478#	5489#	5490#	5491#	5492#	5502#
	5503#	5504#	5505#	5519#	5523#	5534#	5535#	5536#	5537#	5548#	5549#	5550#	5551#	5561#	5562#
	5563#	5564#	5578#	5582#	5592#	5593#	5594#	5595#	5606#	5607#	5608#	5609#	5619#	5620#	5621#
	5622#	5636#	5640#	5650#	5651#	5652#	5653#	5664#	5665#	5666#	5667#	5677#	5678#	5679#	5680#
MSTSTL	5696#	5700#	5711#	5712#	5713#	5714#	5727#	5729#	5730#	5731#	5741#	5742#	5743#	5744#	5758#
	5762#	5772#	5773#	5774#	5775#	5786#	5787#	5788#	5789#	5799#	5800#	5801#	5802#	5814#	5850#
	5855#	5860#	5874#	5899#	5903#	5910#	5947#	5996#	6027#	6068#	6071#	6106#	6155#	6186#	6193#
	6203#	6214#	6216#	6226#	6228#	6236#	6238#	6249#	6260#	6262#	6271#	6273#	6281#	6283#	6308#
	6320#	6332#	6337#	6349#	6364#	6382#	6391#	6395#	6402#	6411#	6418#	6524#	6525#	6548#	6564#
	6565#	6581#	6582#	6586#	6595#	6632#	6635#	6646#	6674#	6677#	6681#	6687#	6700#	6747#	6749#
	6756#	6759#	6765#	6767#	6773#	6775#	6783#	6785#	6789#	6790#	6795#	6796#	6801#	6803#	6822#
	6852#	6869#	6872#	6890#	6891#	6898#	6902#	6914#	6936#	6939#	6945#	6960#	6977#	6981#	7012#
	7015#	7030#	7061#	7065#	7080#	7098#	7099#	7101#	7116#	7118#	7132#	7135#	7137#	7148#	7154#
	1921#	3434#	4177#	4178#	4179#	4180#	4182#	4184#	4185#	4186#	4187#	4188#	4189#	4190#	4192#
	4193#	4197#	4198#	4199#	4202#	4203#	4206#	4207#	4208#	4210#	4211#	4212#	4214#	4215#	4216#
	4219#	4220#	4221#	4223#	4224#	4225#	4227#	4228#	4229#	4230#	4232#	4233#	4234#	4237#	4238#
	4239#	4270#	4302#	4305#	4308#	4311#	4325#	4406#	4423#	4427#	4438#	4440#	4456#	4457#	4473#
	4500#	4507#	4511#	4513#	4541#	4543#	4553#	4556#	4564#	4570#	4578#	4579#	4580#	4588#	4596#
	4602#	4611#	4612#	4613#	4622#	4631#	4636#	4648#	4649#	4650#	4651#	4662#	4663#	4664#	4670#
	4706#	4707#	4712#	4717#	4750#	4751#	4756#	4761#	4794#	4795#	4817#	4818#	4819#	4824#	4833#
	4860#	4878#	4882#	4883#	4913#	4915#	4925#	4943#	4944#	4958#	4970#	4972#	4987#	4992#	5004#
	5005#	5006#	5007#	5017#	5018#	5019#	5020#	5030#	5031#	5032#	5033#	5047#	5052#	5061#	5062#
	5063#	5064#	5073#	5074#	5075#	5076#	5085#	5086#	5087#	5088#	5102#	5107#	5117#	5119#	5120#
	5121#	5131#	5133#	5134#	5135#	5145#	5146#	5147#	5148#	5162#	5167#	5177#	5178#	5179#	5180#
	5190#	5191#	5192#	5193#	5203#	5204#	5205#	5206#	5220#	5226#	5236#	5237#	5238#	5239#	5249#
	5250#	5251#	5252#	5262#	5263#	5264#	5265#	5279#	5285#	5295#	5296#	5297#	5298#	5308#	5309#
	5310#	5311#	5321#	5322#	5323#	5324#	5338#	5343#	5358#	5360#	5361#	5362#	5372#	5373#	5374#
	5375#	5385#	5386#	5387#	5388#	5402#	5407#	5417#	5418#	5419#	5420#	5430#	5431#	5432#	5433#
	5443#	5444#	5445#	5446#	5460#	5464#	5475#	5476#	5477#	5478#	5489#	5490#	5491#	5492#	5502#
	5503#	5504#	5505#	5519#	5523#	5534#	5535#	5536#	5537#	5548#	5549#	5550#	5551#	5561#	5562#
	5563#	5564#	5578#	5582#	5592#	5593#	5594#	5595#	5606#	5607#	5608#	5609#	5619#	5620#	5621#
	5622#	5636#	5640#	5650#	5651#	5652#	5653#	5664#	5665#	5666#	5667#	5677#	5678#	5679#	5680#
	5696#	5700#	5711#	5712#	5713#	5714#	5727#	5729#	5730#	5731#	5741#	5742#	5743#	5744#	5758#
	5762#	5772#	5773#	5774#	5775#	5786#	5787#	5788#	5789#	5799#	5800#	5801#	5802#	5814#	5850#
	5855#	5860#	5874#	5899#	5903#	5910#	5947#	5996#	6027#	6068#	6071#	6106#	6155#	6186#	6193#
	6203#	6214#	6216#	6226#	6228#	6236#	6238#	6249#	6260#	6262#	6271#	6273#	6281#	6283#	6308#

	6320#	6332#	6337#	6349#	6364#	6382#	6391#	6395#	6402#	6411#	6418#	6524#	6525#	6548#	6564#
	6565#	6581#	6582#	6586#	6595#	6632#	6635#	6646#	6674#	6677#	6681#	6687#	6700#	6747#	6749#
	6756#	6759#	6765#	6767#	6773#	6775#	6783#	6785#	6789#	6790#	6795#	6796#	6801#	6803#	6822#
	6852#	6869#	6872#	6890#	6891#	6898#	6902#	6914#	6936#	6939#	6945#	6960#	6977#	6981#	7012#
	7015#	7030#	7061#	7065#	7080#	7098#	7099#	7101#	7116#	7118#	7132#	7135#	7137#	7148#	7154#
MSWORD	1994#	3434#	3474#	3496#	4260#	4507#	4541#	4553#	4564#	4578#	4596#	4611#	4631#	4648#	4662#
	4706#	4750#	4794#	4817#	4833#	4860#	4878#	4913#	4925#	4943#	4958#	4970#	4987#	5004#	5017#
	5030#	5047#	5061#	5073#	5085#	5102#	5117#	5131#	5145#	5162#	5177#	5190#	5203#	5220#	5236#
	5249#	5262#	5279#	5295#	5308#	5321#	5338#	5358#	5372#	5385#	5402#	5417#	5430#	5443#	5460#
	5475#	5489#	5502#	5519#	5534#	5548#	5561#	5578#	5592#	5606#	5619#	5636#	5650#	5664#	5677#
	5696#	5711#	5727#	5741#	5758#	5772#	5786#	5799#	5814#	5850#	5874#	5899#	5947#	5996#	6068#
	6106#	6155#	6203#	6214#	6226#	6236#	6249#	6260#	6271#	6281#	6308#	6320#	6332#	6364#	6382#
	6391#	6395#	6402#	6411#	6524#	6548#	6564#	6581#	6586#	6632#	6646#	6674#	6681#	6700#	6747#
	6756#	6765#	6773#	6783#	6789#	6795#	6801#	6803#	6852#	6869#	6890#	6898#	6914#	6936#	6960#
	6977#	7012#	7030#	7061#	7099#	7116#	7132#	7148#	7172#	7173#	7180#	7225			
MSXFER	1682#	3434#													
OPEN	1171#	3434#													
POINTE	1176#	3434#	3464												
POPSP2	3754#	6377													
PRINTB	1239#	3434#	4177	4178	4179	4180	4182	4184	4185	4186	4187	4188	4189	4190	4192
	4193	4197	4198	4199	4202	4207	4211	4215	4220	4224	4228	4229	4233	4238	
PRINTF	1279#	3434#	4206	4210	4214	4219	4223	4227	4232	4237					
PRINTS	1319#	3434#													
PRINTX	1359#	3434#													
READBU	1399#	3434#													
READEF	1403#	3434#	4302	4305	4308	4311									
RFLAGS	1408#	3434#													
ROMCLK	3830#	3905	3907	3909	3917	3924	3931	3938	3945	3947	3949	3958	3977	3982	4530
	4532	4534	4545	4547	4691	4693	4696	4698	4700	4735	4737	4740	4742	4744	4779
	4781	4784	4786	4788	4805	4807	4809	4811	4840	4842	4844	4850	4852	4864	4867
	4898	4900	4902	4904	4930	4933	4946	4951	4959	4961	4963	5820	5822	5826	5841
	5843	5883	5885	5892	5970	5972	5977	5979	6049	6051	6053	6056	6129	6131	6136
	6138	6207	6218	6230	6253	6264	6275	6310	6436	6449	6452	6455	6468	6471	6474
	6480	6482	6485	6488	6491	6499	6501	6516	6542	6555	6558	6570	6572	6575	6607
	6609	6611	6613	6617	6620	6918	6920	6926	6929	6965	6968	7001	7007	7041	7044
	7046	7055	7083	7086	7088	7091	7106	7109	7122	7124	7141				
SETPRI	1413#	3434#	7080												
SETVEC	1418#	3434#													
SKIP04	3855#	4986													
SKIP06	3841#	3971	4642	4891	5479	5538	5596	5654	5715	5776	5989	6037	6148	6645	6699
SKIP07	3848#	3972	3992	4013	6913	6959	7029								
SI 4SH	1424#	3434#													
SHOMCL	3836#	3911	3951	3960	4854	4861	4996	4998	5009	5011	5022	5024	5053	5055	5065
	5067	5077	5079	5109	5111	5123	5125	5137	5139	5169	5171	5182	5184	5195	5197
	5228	5230	5241	5243	5254	5256	5287	5289	5300	5302	5313	5315	5346	5348	5364
	5366	5377	5379	5409	5411	5422	5424	5435	5437	5467	5469	5481	5483	5494	5496
	5526	5528	5540	5542	5553	5555	5584	5586	5598	5600	5611	5613	5642	5644	5656
	5658	5669	5671	5702	5704	5717	5719	5733	5735	5764	5766	5778	5780	5791	5793
	5887	5974	6133	6922											
STARS	1438#	3434#													
SVC	1452#	3433#	3434												
XFER	1612#	3434#	4260#	4564#	4596#	4631#	4833#	4925#	4987#	5047#	5102#	5162#	5220#	5279#	5338#
	5402#	5460#	5519#	5578#	5636#	5696#	5758#	5814#	5874#	5947#	6106#	6203#	6249#	6395#	6586#
	6646#	6681#	6700#	6803#	6852#	6914#	6960#	7030#							
XFERF	1616#	3434#													
XFERT	1620#	3434#													

CZDMQCO MB207 STATIC DIAG #2 MACY11 30A(1052) 21-JUL-81 14:48 PAGE 39-9
CZDMQC.P11 21-JUL-81 14:36 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0197

SMC	4160#	4177	4178	4179	4180	4182	4184	4185	4186	4187	4188	4189	4190	4192	4193
	4197	4198	4199												

. ABS. 030144 000

ERRORS DETECTED: 0

CZDMQC.BIC,CZDMQC.SEQ/CRF/DOC/NL:TOC=SVC34R.MLB,CZDMQC.P11

RUN-TIME: 40 48 4 SECONDS

RUN-TIME RATIO: 106/93 1.1

CORE USED: 18K (35 PAGES)

DOCUMENT PAGES: 197