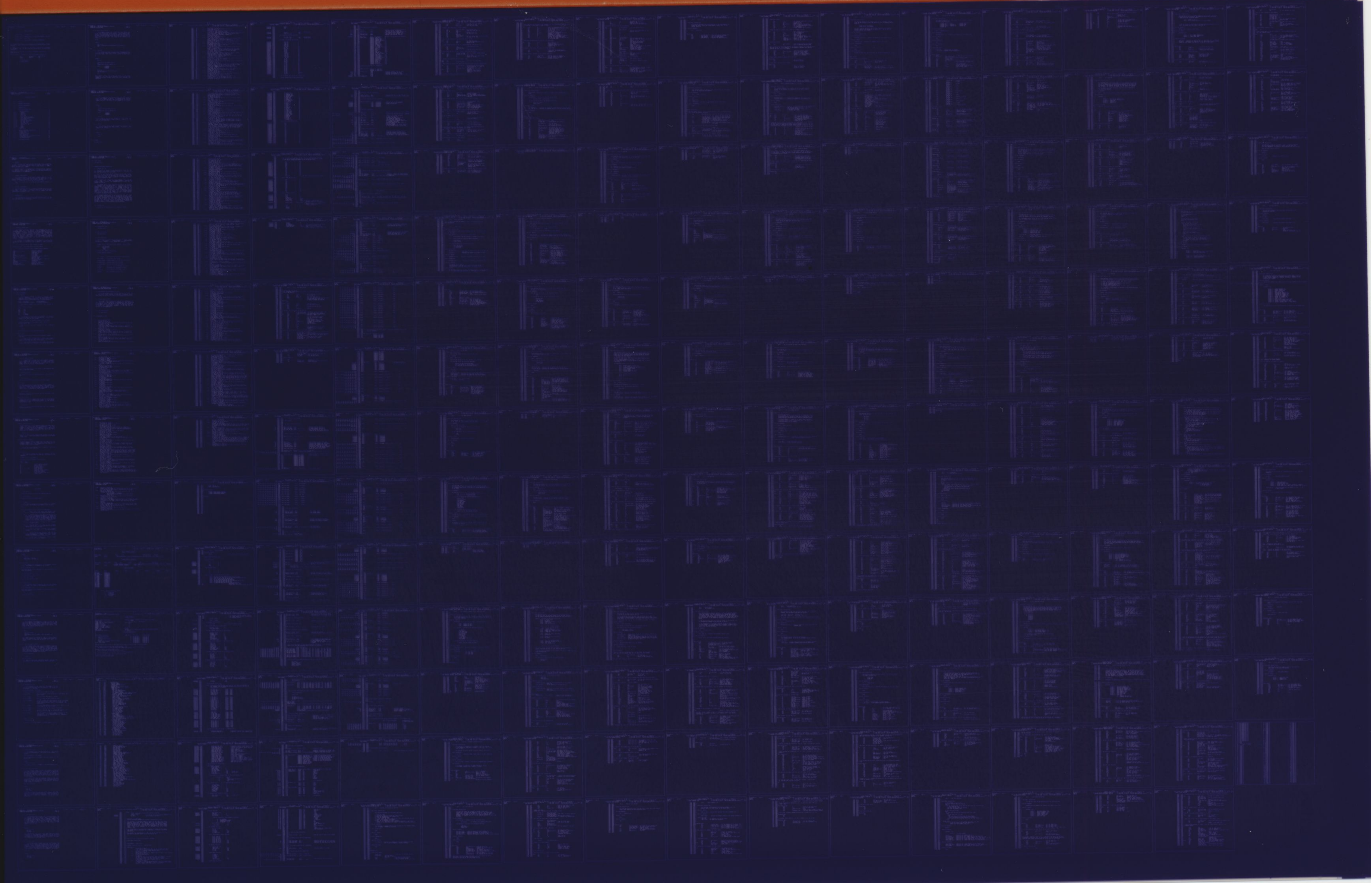
ECKAA-8.7

VAX 11/750 UDIAG MONITOR

EP-ECKAA-DL-8.7
1 OF 2 JUL 1986
COPYRIGHT© 1980-86

digital
MADE IN USA



ECKAA-8.7

VAX 11/750 UDIAG MONITOR

EP-ECKAA-DL-8.7
2 OF 2 JUL 1986
COPYRIGHT© 1980-86

digital
MADE IN USA

IDENTIFICATION

Product code: ZZ-ECKAA-8.7

Product name: ECKAA807 VAX 11/750 MICRO DIAGNOSTIC MONITOR

Product date: APRIL 1986

Maintainer: BASE SYSTEMS DIAGNOSTIC ENGINEERING

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

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) 1980,1986 by Digital Equipment Corporation. All Rights Reserved.

The following are trademarks of Digital Equipment Corporation.

```
DEC              DECsystem-10     DECSYSTEM-20
DECUS            MASSBUS          PDP
UNIBUS           VAX              VMS
```

```
|-|-|-|-|-|-|-|
|d|i|g|i|t|a|l|
|-|-|-|-|-|-|-|
```

Table of Contents

1.0 ABSTRACT

This program is the heart of the VAX-11/750 micro diagnostic package. It provides the mechanism for loading, controlling, and monitoring each micro diagnostic. This program resides in 8085 - RAM on the RDM module (L0006).

The micro monitor is responsible for the parsing and execution of commands typed in at the terminal. It controls the loading of each individual micro diagnostic overlay from the TU58 console drive into RAM memory on the RDM module.

2.0 HARDWARE REQUIREMENTS

This program requires the RDM module (L0006) installed in the CPU backplane, a local terminal, and a working TU58 drive. It is, however, necessary to have the CPU present to make use of the monitor for its purpose in executing micro diagnostics.

There is no restriction to the monitor when adding hardware to the system.

3.0 SOFTWARE REQUIREMENTS

This micro monitor runs as a standalone program residing in the RDM 8085 - RAM memory. It does not require or use the resident micro code. Any required micro code will be loaded with each seperate micro diagnostic test.

4.0 PREREQUISITES

The only prerequisites are that the RDM selftest, that is ran when the machine is powered up, executes with no errors, and that the TU58 console drive is working.

5.0 OPERATING INSTRUCTIONS

The following section describes the commands that can be performed under the micro monitor. Several commands control the microdiagnostic monitor. The monitor may be entered either directly or indirectly. To enter directly from the RDM console command mode (RDM> prompt), type TE/C. To enter indirectly, type TE and then await for the micro diagnostic to complete. To enter directly during test execution, type CTRL/C. The microdiagnostic program enters the monitor automatically if it finds either a TU58 drive error or a hardware error. When the monitor is waiting for a command, it prompts with MIC>.

Any numeric values (addresses and data) must be in hex. Any values enclosed in brackets ([]) is optional. Anytime that an exclamation point is used to describe a command it means exclusive OR.

The list below contains legal command op codes:

Diagnose Loop Set Clear Show Return Continue Examine

This list below contains legal command keyword arguments:

TEST
VBUS(visibility bus)
LOOP
STEP
PASS
NER(no error report)
CYCLE(one microcycle)
BELL
TICK(one-half microcycle)
FLAG
HALT
SOMM(stop-on-micromatch)
CONTINUE
IB(inhibit burst)
TR(trace flag)
DM(diagnostic module)

VA(virtual address)
MA(memory address)
PC(program counter)
PB(PC backup)
RT(R Temps)
MT(M Temps)
PS(processor status longword)
MD(memory data)
WD(write data)
SR(status and control registers)
SF(status flags)
INSTRUCTION
CF(control file)
QA(quality assurance)
QV(quick verify)
SB(single bit errors)

5.1 MONITOR COMMANDS

5.1.1 Diagnose -

The Diagnose command is used to initiate micro diagnostic execution. It initializes the program control flags and starts program execution. The execution of these diagnostics can be controlled by the following arguments:

```
TE[ST]  CO[NTINUE]               Q[UICK]V[ERIFY]
PA[SS]  S[INGLE]B[IT ERRORS]     D[IAGNOSTIC]M[ODULE]
```

The Diagnose command initializes the flag setting as follows:

Flag	Initial Setting
LOOP	clear
NER	clear
BELL	clear
HALT	set
IB	clear
QA	no change
TR	no change

If Diagnose is type without any arguments, the monitor executes all tests on the installed tape once. Control then returns to the monitor (MIC> prompt).

The format of the Diagnose command is as follows:

```
DI[TE:<test_number> [[<test_number] ![CO]] !SB !QV
PA:<pass_count> !DM]
```

The following are some examples of the Diagnose command:

```
    MIC> DI
```

Execute all tests on the installed tape once.

Diagnose Test

The TEST argument may be used with the Diagnose command in three ways.

1. Diagnose TEST followed by a single test number executes the specified test indefinitely. Type CTRL/C to excape from the loop and return to the monitor.

2. Diagnose TEST followed by a test number and Continue executes the specified test and all following tests on the

installed tape cartridge.

3. Diagnose TEST followed by two test numbers executes tests between and including the two numbers specified. Control then returns to the monitor. If the first test number typed is out of range, the program types an error message and returns control to the monitor.

```
    MIC> DI TE:2
```

Test 2 loops indefinitely. Use CTRL/C to stop execution and return to the MIC> prompt.

```
    MIC> DI TE:2 CO
```

Start with test 2 and run remaining tests on the installed tape.

```
    MIC> DI TE:3 6
```

Execute tests 3,4,5,6 then return to the MIC> prompt.

Diagnose Pass

The PASS argument may be used with the Diagnose command to set the pass count to a specified number. The default pass count is 1. A pass count of 0 causes the program to run indefinitely. At the end of the specified number of passes, control returns to the monitor (MIC> prompt).

```
    MIC> DI PA:2
```

Execute all tests on the installed tape for two passes then return to the monitor.

Diagnose Quick Verify

The QV (quick verify) argument may be used with the Diagnose command to run a predetermined subset of tests on the microdiagnostic tape. It was designed for use by DIGITAL manufacturing groups.

```
    MIC> DI QV
```

Execute all tests on the installed tape that are selected for Quick Verify only.

Diagnose Single Bit errors

The SB (single bit errors) argument enables the reporting of single bit errors from the microdiagnostic. If this argument is specified the microdiagnostic will report the number of single bit errors found when running the memory tests. If this argument is not specified, then there will be no report of single bit errors.

```
MIC> DI SB
```

Execute all tests on the installed tape and when running the memory tests, report the number of single bit errors found in the array.

Diagnose Diagnostic Module

The DM (diagnostic module) argument may be used with the diagnose command to run those tests that check out the diagnostic module only. This command should be used only if the VAX-11/750 is working properly.

MIC> DI DM Execute the RDM micro diagnostic from the tape installed in the TU58 drive.

5.1.2 Examine -

The Examine command is used to examine various registers and flags within the 750 CPU. The format of the command is:

```
EX <name_list>[:<reg_number>]
```

The name_list and reg_number are as follows:

name list	reg number	
VA		virtual address register
MA		memory address register
PC		program counter
PB		PC backup register
RT	0 --> 2F	RTemp registers
MT	0 --> 0F	MTemp registers
PS		processor status longword
MD		memory data register
WD		write data register
SR	0 --> 0E	status & control registers
SF		status flags

```
ST                      step counter
```

```
    MIC> EX VA
```

Examine the contents of the Virtual Address register.

```
    MIC> EX RT:C
```

Examine the contents of RTemp number C.

5.1.3 Show Flags -

This command displays the current states of program control flags. The format of the command is as follows:

```
SH FL
```

The current program control flags are listed below:

HALT Halt - This flag calls the monitor when error is detected and error message has been typed.

LOOP Loop on error - This flag is useful only in the event of a program detected error. Flag may be set with the Loop command, or manually by typing SE FL LO. When this flag is set and program detects an error, the test loops on minimum amount of code necessary to recreate the error. type CTRL/C to escape from the loop and return to the monitor.

NER No error report - If flag is set, program does not report errors.

BELL Bell on error - If flag is set, program rings terminal bell on first occurrence of an error and on every fifth occurrence.

QA Quality assurance - If flag is set, program responds to each test as if it had detected a failure.

IB Inhibit burst flag - If this flag and the loop flag are set, the program does not try to loop on DCS microcode only. Rather, it loops between ERRLOOP and IFERROR pseudo instructions.

TR Trace flag - If flag is set, monitor types test names as well as test numbers.

Example:

```
    MIC> SH FL

    FLAGS SET:  LO,NE,BE,
    FLAGS CLEAR:  HA,IB,QA,TR
```

5.1.4 Set Flag -

This command sets (enables) any of the program control flags. The format of the command is as follows:

```
SE FL <flag_name_list>
```

For a description of the flag_name_list, see the section on SHOW FLAG command.

```
    MIC> SE FL NE
```

Sets the No Error report flag.

```
    MIC> SE FL LO BE
```

Sets the Loop and Bell flags.

5.1.5 Clear Flag -

This command clears (disables) any of the program control flags. The format of the command is as follows:

```
CL FL <flag_name_list>
```

For a description of the flag_name_list, see the section on SHOW FLAG command.

```
    MIC> CL FL HA
```

Clears the Halt flag.

5.1.6 Set Stop-on-Micromatch -

This command stops execution of code in DCS at the specified address. DCS addresses range from 0 to 3F. Add 1800 to the desired DCS address. Type the Continue command after the MIC> prompt to proceed from your current halt, after typing SE SO <cs_address> to get to that address. The format of the command is as follows:

```
SE SO:<cs_address>
```

Example:

```
MIC> SE SO:1803
MIC> CO
MICRO BREAK MATCH, CS ADDR = 1803, DCS ADDR = 03
MIC>
```

This command causes the machine to stop at micro address 1803 (DCS address 03). After the command stops execution, the MIC> prompt is issued and the monitor will await your command.

5.1.7 Clear Stop-on-Micromatch -

This command clears the stop-on-micromatch function. Add 1800 to the desired DCS address to create a scope sync pulse at that address. If <cs_address> is specified, a scop sync pulse is generated on slot 6, pin C81 when the current address matches <cs_address>. The pulse occurs with the M clock that marks the beginning of the specified microcycle. The format of the command is as follows:

```
CL SO[:<cs_address>]
```

Example:

```
MIC> CL SO:1803
```

This command clears the stop-on-micromatch function and loads 1803 into the RDM match register for a scope sync.

5.1.8 Set Control File -

This command sets the specified <bit_number> in the control file of the specified <dcs_address>. The format for the command is as follows:

```
SE CF:<dcs_address> <bit_number>
```

The control file bit functions are defined as follows:

Bit <0> VSTB — This bit strobes visibility bus at rising edge of the next M clock

Bit <1> DCSACL — This bit clears DCS address register.

Bit <2> STPCLK — This bit stops the VAX-11/750 CPU clock.

Bit <3> ENTRACE — This bit latches the trace register at the rising edge of the next M clock. When this bit is not set, the trace register monitors control store address lines continously.

Bit <4> HRWBUS — This bit reads W-BUS by strobing the WH register at the rising edge of the next M clock.

Bit <5> WBUSDR — This bit enalbes the WD register onto W-BUS.

Bit <6> STBCMI — This bit reads CMI data by strobing MH register with CMI data at rising edge of the next M clock.

Bit <7> SATRIG — This bit inhibits clocking signature analyzer on the next M clock. This signal connects to a post on the module for use by signature analyzer (backplane pin C73 on slot 6).

Example:

```
    MIC> SE CF:3 4
```

Set control file bit 4 (HRWBUS) in the DCS address of 03.

5.1.9 Clear Control File -

This command clears the specified bits in the control file of the <dcs_address>. The format of the command is as follows:

```
CL CF:<dcs_address> <bit_number>
```

For a description of <bit_number>, see the section on SET CONTROL FILE.

Example:

```
    MIC> CL CF:3 4
```

Clear control file bit 4 (HRWBUS) in the DCS address of 03.

5.1.10 Set Step Instruction -

This command steps through the pseudo-instructions (8085 code) in the current test. If <test_pc> is specified, the step function does not start until the instruction at the <test_pc> address is ready to execute. If <test_pc> is not specified, the step function begins at the next pseudo instruction of the current test, following a Loop or Continue command.

In either case, when the program stops at the next pseudo instruction, press the space bar to continue stepping through the pseudo instructions. Type a carriage return to escape from the step mode back to the monitor. The format of the command is as follows:

```
SE ST IN[:<test_pc>]
```

Example:

```
    MIC> SE ST IN
    MIC> DI
    TPC = 001F
```

This example causes the step mode to be enabled and after typing DI to start program execution, the micro diagnostic stops in the first pseudo instruction. The typing of a space bar will cause the diagnostic to step one pseudo instruction.

5.1.11 Set Step Cycle -

This command steps through DCS microinstructions one CPU machine cycle at a time (M clock). After entering in this command, the Continue, Loop, or Diagnose command must be typed to begin stepping. After the program stops at the first DCS address in the current test, press the space bar to step through the test. Type a carriage return to escape and return control to the monitor. The format of the command is as follows:

```
SE ST CY
```

Example:

```
    MIC> SE ST CY
    MIC> DI
    DCS ADDR = 00
```

This example sets the cycle step mode and then starts execution of the diagnostic. When the first DCS microword is fetched for execution (DCS address 00), the monitor will stop execution, print the address on the terminal, and await for a space bar or a carriage return.

5.1.12 Set Step Tick -

This command steps through DCS micro instructions, one B clock at a time. After entering this command, the Continue, Loop, or Diagnose command must be typed to begin stepping. After the program stops at the first DCS address in the current test, press the space bar to step through the test. Type a carriage return to escape and return control to the monitor. The format of this command is as follows:

```
SE ST TI
```

Example:

```
    MIC> SE ST TI
    MIC> DI
    DCS ADDR = 00
```

5.1.13 Show Visibility Bus -

This command strobes the visibility bus (VBUS) and prints out the current signal states. This command is most useful after executing the Initialize pseudo instruction, and when stepping through microcode in DCS. Show Visibility Bus displays the 40 bits of the visibility bus, with the low order bit on the right. The format of this command is as follows:

```
SH VB
```

Example:

```
MIC> SH VB
VBUS = 1110100010111010000100001011111010010111
MIC>
```

5.1.14 Continue -

This command continues testing after the program stops, following error detection or CTRL/C. This command does not modify any flags. The format of this command is as follows:

```
CO
```

Example:

```
MIC> DI

ECKAA-V5.00 MIC(L0003)-V00.05
01,
?ERROR:0030 TEST:01 SUBTEST:01

DATA:         AAAAAAAA
              AAA8AAAA
              00000001

MIC> CO
02,03,04,05,06,07,08,09,0A,0B,0C,
END OF PASS 01

MIC>
```

This example shows that after an error in the micro diagnostic, the test can be resumed by typing the Continue command at the MIC> prompt after the error message has completed.

5.1.15 Loop -

This command puts the program in an error loop after it detect and reports an error. The command clears the HALT flag, sets the NER and LOOP flags, and performs a Continue command function. The format of the command is as follows:

```
LO
```

Example:

```
MIC> DI

ECKAA-V5.00 MIC(L0003)-V00.05
01,
?ERROR:0030 TEST:01 SUBTEST:01

DATA:       AAAAAAAA
            AAA8AAAA
            00000001

MIC> LO
```

In this example, when the Loop command is typed after the error, the diagnostic will loop on test 01 - subtest 01. To stop the loop type CTRL/C.

5.1.16 Return -

This command returns the microdiagnostic monitor to the RDM console command mode (RDM> prompt). The format of the command is as follows:

```
RE
```

Example:

```
MIC> RE

RDM>
```

5.2 OPTIONS

Not applicable

5.3 EVENT FLAGS

Not applicable

6.0 PROGRAM FUNCTIONAL DESCRIPTION

6.1 Program Overview

The monitor is loaded from the TU58 console drive by the RDM software then control is transferred to it. There are two ways that the monitor will take over.

The first is by using the TE command from the RDM prompt. When this is done, the monitor will start the execution of the micro diagnostic, that is in the TU58, for one pass. When the first pass is complete, control is passed to the command parser, which, issues the MIC> prompt.

The second is by using the TE/C command from the RDM prompt. When this is done, the command parser is entered and the MIC> prompt is issued. The command parser will wait for a command to be given.

The command parser takes characters entered on the terminal (as commands) and evaluates them. It has a table of possible commands that can be issued and if the command typed on the terminal is not supported then an error message is typed at the terminal. If the command entered is supported (one that is located in the table) then the parser dispatches to the appropriate routine. When the parser see's the DIagnose command typed on the terminal the parser will call the appropriate sub-routines to read the tu58. After the tu58 is read, control is passed to the Opcode Dispatch routine.

The Opcode Dispatch routine will interpret the test data (pseudo opcodes) and dispatch to the appropriate routine. After each routine has executed control is returned to the dispatcher. The Opcode dispatcher will continue to read test overlays from the tu58 drive until the last test has been reached and executed. At this point control is returned to the parser and the MIC> prompt is issued. Control can be either returned to the RDM or another micro diagnostic can be executed.

6.2 Program Size

The monitor is approximately 11.1 kbytes in length.

6.3 Program Run Times

Not Applicable.

6.4 Run-time Dynamics

Not Applicable.

6.5 Fault Detection

The monitor will report errors with the following header: test_pc, test_number, subtest_number, expected_data, received_data, indexes, r_temps, gatearrays, modules.

```
?ERROR: test_pc TEST: test_number SUBTEST: subtest_number

DATA:   expected_data
        received_data
        indexes
        r_temps (0-F)

FAILING GATE ARRAYS: gate_arrays

FAILING MODULES: modules
```

test_pc	Location of failure within test overlay.
test_number	Test number that failed.
subtest_number	Subtest number with the test that failed.
expected_data	Expected data from test code.
received_data	Received data from test code.
indexes	Loop I, J, and K (dependent on test).
rtemps	CPU-RTEMPS 0:F (dependent on test).
gatearrays	Failing gate array callout list.
modules	Failing module callout list.

6.6 Performance During Hardware Failures

If the RDM hardware starts failing then monitor should not be used until this problem is resolved. If a power failure takes place the monitor will have to be reloaded and restarted.

6.7 Program Applications

This program must be used to run any micro diagnostics on the VAX-11/750. Its application depends on the reason for executing the micro diagnostics. This detection and isolation of hardware fautls, and verifying that the machine hardware is operational after installation or repair. It can also be used to verify the "operational goodness" of the 750 cpu on a periodic or as needed basis.

6.8 Test Descriptions

Not Applicable.

7.0 MAINTENANCE HISTORY

```
1-00     Don Monroe 16-JUN-80
         Initial Release
1-01     Don Monroe 26-JUN-80
         Added pass count option to QA Flag. Conditions the QA flag
         on a specified pass count.
1-02     Don Monroe 3-Jul-80
         Fixed bug in BURST CLOCK routine related to stepping the
         clock with the LOOP & ERROR flags set and an INHIBIT on the
         Burst Clock call.
1-03     Don Monroe 3-Jul-80
         Fixed bug in typeout of Test PC.
1-04     Don Monroe 23-Jul-80
         Added a feature to BURST CLOCK that enables the detection
         of the clock being hung. Added CMI BUS name to module name
         list.
1-05     Don Monroe/Bill Landry 27-AUG-80
         Altered monitor to load DCS with NOPS before using any micro-
         code.
1-06     Don Monroe 9-Sep-80
         Modified LOADREG routine so that if the register is the RD
         Control Register, the state of the front panel lights is
         copied into it.
1-07     Don Monroe 10-Sep-80
         Added the ERRLOG and DUMPLOG pseudo instructions. Also added
```

```
        the CMA (M8728) to the module callout list.
1-08    Don Monroe 16-Sep-80
        Fixed bugs in DUMP LOG routine.
1-09    Don Monroe 16-Sep-80
        Fixed bugs in ERROR LOG routine. Added setting and clearing
        of error flags in ERROR LOG routine.
1-10    Bill Landry 19-Sep-80
        Fixed bug in ERROR LOG routine to clear INIT FLAG after first
        pass.
1-11    Don Monroe 23-Sep-1980
        Fixed more bugs in error log routine.
2-00    Don Monroe 29-Sep-1980
        Second Release
2-01    Don Monroe 7-Oct-80
        Modified Gate Array list to support bit slice algorithm in
        error report routine. Added the bit slice algorithm.
2-02    Don Monroe 20-Oct-80
        Fixed SHOW VBUS routine to ensure ctrl file is not holding
        vbus in load state.
2-03    Don Monroe 17-Nov-1980
        Fixed bug in CHECK_QV routine.
3-00    Don Monroe 17-NOV-1980
        Third Release
3-01    Don Monroe 29-Dec-1980
        Added DI DM command to invoke only the tests for the RDM.
3-02    Don Monroe 30-Dec-1980
        Fixed bugs.
3-03    Don Monroe 30-Dec-1980
        Fixed bugs.
3-04    Don Monroe 31-Dec-1980
        Fixed bugs.
4-00    Don Monroe 5-Jan-1980
        Added code to enable the clock interrupts if key switch is
        in remote position.
4-01    Don Monroe 17-Feb-81
        Added PRINT_N and PRINT_S pseudo instructions.
        Added NOTEQUAL condition to SKIP pseudo instruction.
5-00    Don Monroe  2-Mar-1981
        Release.
5-01    Bill Landry 19-MAY-1981
        Added code to fix ERROR_LOG ROUTINE. Routine would take
        fail exit if entry was already in log.
5-02    Don Monroe 21-May-81
        Started work on EXAMINE command.
5-03    Don Monroe 3-Jun-81
        Fixed bugs in EXAMINE routine.
5-04    Don Monroe 8-Jun-81
        Fixed bugs in EXAMINE routine.
        Added an initialize of DCS on entry to the monitor.
5-05    Don Monroe 8-Jun-81
        Moved the clock stop code from program init to entry
        point into monitor so EXAMINE's would work on TE/C cmd.
5-06    Don Monroe 12-Jun-81
        Fixed the RDM_WDR micro word.
```

```
5-07    Don Monroe 12-Jun-81
        Fixed mask on examine of SF and SC.
6-00    Don Monroe 23-Jun-81
        Changed revision for release.
6-01    Don Monroe 23-Jul-81
        Changed SET STEP CYCLE/TICK routines to clear the
        other flag. Modified typeout routine to automatically
        insert a Carriage Return Line Feed after 80 characters.
6-02    Don Monroe 24-Aug-81
        Added new Gate Array names for FPA module.
6-03    Don Monroe  2-Sep-81
        Masked PSL data on PSL EXAMINE.
6-04    Dave Spain  4-Sep-81
        Added missing FPA gate arrays.
6-05    Don Monroe 8-Sep-81
        Fixed bug in FPA bit-sliced name tables.
        Added ENABLE_STALL routine and modified BRSTCLOCK routine
        to support it.
6-06    Don Monroe 22-Sep-81
        Fixed bug in PSL Examine. The AND with immediate mask used
        the wrong instruction.
6-07    Don Monroe 24-Sep-81
        Modified Single Bit Error Messages. Changed Load Overlay
        Routine to do an Initialize before trying to load cache
        and other internals.
7-00    Don Monroe 10-Nov-81
        Added SBE keyword and flag to DIAGNOSE command to enable
        the DUMP_LOG routine to print the state of the error log.
7-01    Dave Spain 28-Jan-82
        Added L0016 memory controller and MAD gate array, and changed
        CMC and CMA names. L0011 controller is now MC0, L0016 is MC1.
        M8728 256K array card is now MA0, M8750 1M array card is MA1.
        Changed callouts of FCS, FMR and FFA to callout all slices.
7-02    Dave Spain 1-Feb-82
        Fixed bug in FCS, FMR and FFA callouts introduced in 7-01.
7-03    Dave Spain 3-Feb-82
        Replaced references to LOWER 256 KB in correctable bits
        messages to ARRAY 0.
7-04    Dave Spain 19-Feb-82
        Removed CMC and CMA module references from error callouts.
        Now calls out Memory Controller and Memory Array Cards under
        the L and M module numbers only.
7-05    Dave Spain 8-Mar-82
        Fixed bug in ERROR_LOG routine. Wasn't checking test address
        against top of error log correctly.
7-06    Dave Shull 10-Mar-82
        Changed CRD messages (LOG_MSG_2 & LOG_MSG_3) to reference
        the memory array card as ARRAY versus ARRAY 0.  This allows
        sharing of message for either array 0, array 1, etc.
7-07    Dave Shull 22-Mar-82
        Changed the DUMP_LOG routine so it would set the ERR_LOG_INIT
        flag.  This way the combinations of ERRLOG and DUMPLOG pseudo's
        can be used more then once within the same micro to check more
        than one array.
```

```
7-08     Dave Shull 06-Apr-82
         Changed the TYPE_ERROR and TYPE_GA_LIST routines so they would
         print the failing Gate Array(s) and Module(s) in column.
8-00     Dave Shull 08-Apr-82
         Removed the BEGINSA and ENDSA pseudo execution code which
         included the following:
                 -  Removed symbol SA_ADDRESS:
                 -  Removed BEGIN_SA and END_SA from INSTR_TABLE:
8-01     Dave Spain 05-May-82
         Changed bit-slice tables for FPA gate arrays to enhance
         isolation for error callouts.
8-02     Dave Shull 11-May-82
         Changed the LOAD_INDEX: routine so that it will handle the case
         where there is an address specified that points to the value to
         be loaded into the index, with No index specified for that
         address.
8-03     Dave Shull 03-Jun-82
         Fixed the COMPARE_VBUS: routine so it would work with an index
         greater than 80 (hex).  Prior to this fix, values greater than
         80 (hex) would no get processed correctly.
8-04     Dave Shull 14-Jun-82
         Changed CMK callout to CML for new gate array.
8-05     Dave Shull 03-Nov-82
         Changed the CCS module callout from L0005 to L0008.  This is for the
         new PCS module.
```

```
                                              +-------------------------+
                                              ! Object Module Synopsis !
                                              +-------------------------+

Module Name         Ident                   Bytes       File                                        Creation Date          Creator
-----------         -----                   -----       -----                                       -------------          -------
ECKAA               V08.05                   11358 WRKD$0:[SHULL.COMET.MICMON]ECKAA.OBJ;17-DEC-1982 16:38  VAX-11 Macro V03-00

                                             +---------------------------+
                                             ! Program Section Synopsis !
                                             +---------------------------+

Psect Name          Module Name            Base       End              Length                Align                 Attributes
----------          -----------            ----       ---              ------                -----                 ----------
. BLANK .                                00000000 00002C5D 00002C5E (        11358.) BYTE 0 NOPIC,USR,CON,REL,LCL,NOSHR,  EXE,  RD,  WRT,NOVEC
                    ECKAA                00000000 00002C5D 00002C5E (        11358.) BYTE 0

                                                  +-----------------+
                                                  ! Symbols By Name !
                                                  +-----------------+

Symbol              Value                   Symbol              Value                   Symbol              Value                   Symbol              Value
------              -----                   ------              -----                   ------              -----                   ------              -----
ADD_NAME            00000949-R
ALP_NAME            000006F9-R
BIT_SLICE_TBL       000006DB-R
FAC_NAME            00000B84-R
FCS_NAME            00000B96-R
FEX_NAME            00000B50-R
FFH_NAME            00000C2F-R
FFL_NAME            00000BF2-R
FIC_NAME            00000B6F-R
FIO_NAME            00000AD7-R
FMR_NAME            00000BD3-R
MDL_NAME            000009D5-R
MDR_NAME            00000860-R
MEC_NAME            000009A6-R
SRM_NAME            000007B2-R
UDP_NAME            00000A32-R


             Key for special characters above:
                    +------------------+
                    ! *  - Undefined    !
                    ! U  - Universal    !
                    ! R  - Relocatable !
                    ! X  - External     !
                    +------------------+
                                                  +----------------+
                                                  ! Image Synopsis !
                                                  +----------------+

Virtual memory allocated:                         00000000 00002DFF 00002E00 (11776. bytes, 23. pages)
Stack size:                                              0. pages
Image binary virtual block limits:                       1.        23. (    23. blocks)
Image name and identification:                    ECKAA V08.05
Number of files:                                         1.
Number of modules:                                       1.
Number of program sections:                              2.
Number of global symbols:                               16.
Number of image sections:                                1.
Image type:                                       SYSTEM.
Map format:                                       DEFAULT in file WRKD$0:[SHULL.COMET.MICMON]ECKAA.MAP;806
Estimated map length:                             10. blocks
                                               +----------------------+
                                               ! Link Run Statistics !
                                               +----------------------+

Performance Indicators                            Page Faults  CPU Time         Elapsed Time
----------------------                            -----------  --------         ------------
    Command processing:                                    30  00:00:00.07      00:00:00.12
    Pass 1:                                                25  00:00:00.21      00:00:00.47
    Allocation/Relocation:                                  4  00:00:00.05      00:00:00.22
    Pass 2:                                                24  00:00:02.19      00:00:03.11
    Map data after object module synopsis:                  7  00:00:00.06      00:00:00.10
    Symbol table output:                                    1  00:00:00.01      00:00:00.11
Total run values:                                          91  00:00:02.59      00:00:04.13

Using a working set limited to 300 pages and 29 pages of data storage (excluding image)

Total number object records read (both passes):   176
    of which 0 were in libraries and 2 were DEBUG data records containing 41 bytes

Number of modules extracted explicitly             = 0
    with 0 extracted to resolve undefined symbols

0 library searches were for symbols not in the library searched

A total of 0 global symbol table records was written

/SYST=0/EXE=ECKAA.EXE;805 ECKAA
```

Table of contents

```
                0000    ;DEBUG = 0
                0000            .TITLE  ECKAA VAX-11/750 MICRO DIAGNOSTIC MONITOR
                0000            .IDENT  /V08.07/
      00000008  0000            PRIM_REV = 8                ; If these go GTR 9 then change GENERATE_TITLE
      00000007  0000            SEC_REV =  7                ; macro (remove leading 0).
                0000
                0000    ;
                0000    ; COPYRIGHT (C) 1980,1982
                0000    ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS 01754
                0000    ;
                0000    ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE  ONLY ON A SINGLE
                0000    ; COMPUTER  SYSTEM AND  MAY BE  COPIED ONLY WITH  THE INCLUSION OF THE
                0000    ; ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE, OR ANY OTHER COPIES THEREOF,
                0000    ; MAY NOT BE PROVIDED OR  OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON
                0000    ; EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE
                0000    ; TERMS.  TITLE TO AND  OWNERSHIP OF THE  SOFTWARE  SHALL AT ALL TIMES
                0000    ; REMAIN IN DEC.
                0000    ;
                0000    ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
                0000    ; AND SHOULD  NOT BE CONSTRUED  AS A COMMITMENT  BY DIGITAL  EQUIPMENT
                0000    ; CORPORATION.
                0000    ;
                0000    ; DEC ASSUMES  NO  RESPONSIBILITY  FOR  THE USE OR  RELIABILITY OF ITS
                0000    ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
                0000    ;
                0000
                0000    ;++
                0000    ;
                0000    ; FACILITY:  VAX 11/750 Micro Diagnostic Package
                0000    ;
                0000    ; FUNCTIONAL DESCRIPTION:
                0000    ;
                0000    ;
                0000    ; ENVIRONMENT:  Stand alone.
                0000    ;
                0000    ; AUTHOR:  Donald W. Monroe,  CREATION DATE:
                0000    ;
                0000    ; MODIFIED BY:
                0000    ;       0-2     Don Monroe 9 Aug 79
                0000    ;               Fixed load of RTEMP data. QA flag was missing from ASCII
                0000    ;               list. Also fixed macro package to create the test files
                0000    ;               correctly for RTEMP, CACHE, and DCS data.
                0000    ;       0-3     Don Monroe 9 Aug 79
                0000    ;               Added load of CACHE data. Added disable of CMI in INIT routine.
                0000    ;       0-4     Don Monroe 14 Aug 79
                0000    ;               Fixed the FETCH routine.
                0000    ;       0-5     Don Monroe
                0000    ;               Changed the INIT routine to not force a CS Address. Deleted
                0000    ;               the call to disable the CMI from the INIT routine.
                0000    ;               Fixed deposit with line feed terminator in debug routine.
                0000    ;               Fixed a bug in the module typeout routine.
                0000    ;       0-6     Don Monroe 16 Aug 79
                0000    ;               Fixed bugs in ENDLOOP and BURST CLOCK routines. Added typeout
                0000    ;               of TRACE register and PARITY ERROR on unexpected clock stop.
                0000    ;               Added the ENABLE MICRO TRAP pseudo instruction.
```

```
0000  ;       0-7     Don Monroe 17 Aug 79
0000  ;               Fixed bugs with the WRITE$_DCS macro calls.
0000  ;       0-8     Don Monroe 21 Aug 79
0000  ;               Fixed bug in Start_Execution routine that left clear control
0000  ;               file bit set when clock was ticked.
0000  ;               Fixed bug in Read_overlay routine when loading RTEMP data into
0000  ;               the scratch pads. Same bug in Type_Error routine also.
0000  ;               Fixed a bug to make the BREAK command work.
0000  ;       0-9     Don Monroe 22 Aug 79
0000  ;               Fixed bug in WRITE$_DCS macro.
0000  ;       0-10    Don Monroe 22 Aug 79
0000  ;               Fixed a bug in the TYPE_ERROR routine when reading RTEMP's.
0000  ;               Added test number to TU58 error typeout.
0000  ;               Changed RTEMP typeout to start at R0 instead of R10(X).
0000  ;       0-11    Don Monroe 23 Aug 79
0000  ;               Added typeout of Loop flags on Unexpected Stop Clock.
0000  ;       0-12    Don Monroe 9 Oct 79
0000  ;               Made ENABLE MICRO TRAP instruction work. Added an argument
0000  ;               to BRSTCLK that inhibits a RET to 0.
0000  ;       0-13    Don Monroe 22 Oct 1979
0000  ;               Cleared the CYCLE and TICK flags in BURST_CLOCK routine if
0000  ;               a space is not typed.
0000  ;               Added a CLOCKX to the micro code that writes the CACHE.
0000  ;               Fixed bugs in micro words to disable the CMI and write
0000  ;               the cache. Added a disable cmi to INIT_THE WORLD.
0000  ;       0-14    Don Monroe 29 Oct 1979
0000  ;               Patched test number typeout routine to type all numbers.
0000  ;               Removed disable CMI call from INIT routine.
0000  ;               Fixed BAD DATA in Compare V Bus Routine.
0000  ;               Fixed Burst Clock routine to work correctly when the stop
0000  ;               clock bit is set inlocation 3F.
0000  ;       0-15    Don Monroe 31 Oct 1979
0000  ;               Fixed set/clr control file routines to restore the correct
0000  ;               control file (current dcs address - 1).
0000  ;       0-16    Don Monroe 13 Nov 79
0000  ;               Added micro instruction to CACHE WRITE sequence to hold data
0000  ;               for another cycle.
0000  ;       0-17    Don Monroe 19 Nov 79
0000  ;               Added routine to disable cmi routine to disable memory mgmt.
0000  ;               Modified single cycle and single tick routines to leave
0000  ;               MASTER HALT ENABLE off while ticking the clock.
0000  ;               Fixed bug in BURST CLOCK routine for loop on error.
0000  ;       0-18    Don Monroe 26 Nov 79
0000  ;               Added PROC INIT to disable CMI micro code. Added call to
0000  ;               DISABLE CMI routine in the INITIALIZE routine.
0000  ;       0-19    Don Monroe 26 Nov 79
0000  ;               Modified startup code for real environment.
0000  ;       0-20    Don Monroe 26 Nov 79
0000  ;               Changed COMPARE V BUS routine to put value of VBUS bit in
0000  ;               bit position 8 (for typeout) instead of 7.
0000  ;       0-21    Don Monroe 27 Nov 79
0000  ;               Made QA flag set the ERROR and LOOP ERROR flags
0000  ;               unconditionally in the IF ERROR routine.
0000  ;               Fixed INIT routine to leave DCS deselected when it exits.
0000  ;       0-22    Don Monroe 29 Nov 79
```

```
                0000    111 ;                   Fixed BURST CLOCK routine to stop ticking the clock if a
                0000    112 ;                   micro trap occurs during single cycle or single tick.
                0000    113 ;                   Added code to enable/disable signature analyzer in the
                0000    114 ;                   BEGIN/END SA routines.
                0000    115 ;           0-23    Don Monroe 30 Nov 79
                0000    116 ;                   Fixed bug in BEGIN SA ROUTINE to save current PC rather
                0000    117 ;                   than PC+1.
                0000    118 ;           0-24    Bill Landry 5 Dec 79
                0000    119 ;                   Added code to not test for trap condition at dcs address
                0000    120 ;                   0 when in single cycle mode.
                0000    121 ;           0-25    Don Monroe 10 Dec 79
                0000    122 ;                   Modified for the Second Pass Multi-Wire RDM.
                0000    123 ;           0-26    Don Monroe 11 Dec 79
                0000    124 ;                   Forgot to rearrange the parity bit mask for the scrambled
                0000    125 ;                   micro word.
                0000    126 ;           0-27    Don Monroe 17 Dec 79
                0000    127 ;                   Fixed bug in startup code to scramble the builtin micro code.
                0000    128 ;           0-28    Don Monroe 18 Dec 79
                0000    129 ;                   Found bug in scramble table. !'s instead of ;'s.
                0000    130 ;           0-29    Don Monroe 31 Dec 79
                0000    131 ;                   Fixed GO command bug in ODT routine to start at the correct
                0000    132 ;                   place. Changed NOP micro word to not have STOP CLOCK bit set.
                0000    133 ;           0-30    Don Monroe 2 Jan 80
                0000    134 ;                   Fixed bug in startup code that lost the state of the C bit.
                0000    135 ;                   Also fixed bug in ODT routine that lost the C bit and did
                0000    136 ;                   not save the PSW correctly.
                0000    137 ;           0-31    Don Monroe 3 Jan 80
                0000    138 ;                   Fixed bug in ODT routine that unconditionally set the C
                0000    139 ;                   bit on a 'G' command. Changed the BURST CLOCK routine to
                0000    140 ;                   read the DCS ADDRESS register for the typeout instead of
                0000    141 ;                   using an internal counter.
                0000    142 ;           0-32    Don Monroe 7 Jan 80
                0000    143 ;                   Changed format of show flags typeout. Added load of VBUS to
                0000    144 ;                   SHOW VBUS command. Added typeout of remaining VBUS bits to
                0000    145 ;                   error typeout if CMP was a CMPVBUS. Added a TYPE_UNCOUNTED
                0000    146 ;                   macro to type an uncounted ascii string. Changed format
                0000    147 ;                   of SET/CLR CF data from the bit mask to the bit number.
                0000    148 ;           0-33    Don Monroe 8 Jan 80
                0000    149 ;                   Fixed bugs introduced in version 32.
                0000    150 ;           0-34    Don Monroe 9 Jan 80
                0000    151 ;                   Added EXAMINE and DEPOSIT commands for test overlays.
                0000    152 ;                   Fixed bug in set step cycle.
                0000    153 ;                   Fixed bug in somm and jazzed-up print out.
                0000    154 ;           0-35    Don Monroe 10 Jan 80
                0000    155 ;                   Fixed bug in CPU_INIT that forgot about SOMM_FLAG.
                0000    156 ;           0-36    Don Monroe 10 Jan 80
                0000    157 ;                   Modified Directory Search routine to support the new
                0000    158 ;                   directory format.
                0000    159 ;           0-37    Don Monroe 23 Jan 80
                0000    160 ;                   Modified Directory Search routine to support the file version
                0000    161 ;                   number encoded in the directory. Also modified BURST_CLOCK
                0000    162 ;                   routine to try and fix control c problem.
                0000    163 ;           0-38    Don Monroe 24 Jan 80
                0000    164 ;                   Save and restore SOFT_CTRL_REG in BURST_CLOCK routine so
                0000    165 ;                   that a CTRL C typed before the BURST_CLOCK will be remembered.
```

```
0000  ;       0-39    Don Monroe 29 Jan 80
0000  ;               Added a clock tick to the ENDSA pseudo instruction to shut off
0000  ;               the signature analyzer.
0000  ;       0-40    Don Monroe 29 Jan 80
0000  ;               Forgot to enable the control file bit in EDIT 39.
0000  ;       0-41    Don Monroe 30 Jan 80
0000  ;               Reversed the order of loading the DCS ADDRESS register and
0000  ;               clearing the control file in the START_EXECUTION routine.
0000  ;               Changed END_SA routine to actually execute a micro word
0000  ;               instead of just fetching it.
0000  ;       0-42    Don Monroe 4 Feb 80
0000  ;               Fixed bug in DISABLE_CMI routine introduced when the DCS
0000  ;               SCRATCH area was moved.
0000  ;       0-43    Don Monroe 4 Feb 80
0000  ;               Added DCS BUS ENABLE bit to maintenance register.
0000  ;       0-44    Don Monroe 5 Feb 80
0000  ;               Added micro code to INIT_CPU routine to clear the PSL.
0000  ;       0-45    Don Monroe 6 Feb 80
0000  ;               Modified DIRECT_SEARCH to save the total number of tests
0000  ;               in the file so the END_TEST pseudo op knows when not to
0000  ;               try and read the next test.
0000  ;       0-46    Don Monroe 7 Feb 80
0000  ;               Fixed bug in DIRECTORY_SEARCH routine.
0000  ;       0-47    Don Monroe 29 Feb 80
0000  ;               Added new pseudo op called ONERROR. Added additional
0000  ;               parameters and functionality to LDFIELD pseudo op.
0000  ;       0-48    Don Monroe 3 Mar 80
0000  ;               Deleated the ONERROR pseudo op and made the same functionality
0000  ;               part of the SKIP pseudo op. Refer to the macro library for
0000  ;               details on the SKIP pseudo op.
0000  ;       0-49    Don Monroe 3 Mar 80
0000  ;               Fixed bug in SKIP introduced in version 0-48.
0000  ;       0-50    Dave Spain 3 Mar 80
0000  ;               Fixed bug in SKIP left over from version 0-49.
0000  ;       0-51    Don Monroe 7 Mar 80
0000  ;               Fixed bug in LOAD FIELD introduced in version 47.
0000  ;               Changed SUBTEST routine to get the subtest number from
0000  ;               the test stream instead of incrementing. Fixed bug in
0000  ;               RETURN TO RDM routine causing the clock not to start.
0000  ;       0-52    Don Monroe 27 Mar 80
0000  ;               Added an inhibit scramble parameter to the LDFIELD pseudo
0000  ;               instruction. Change module typeout to DCxxx numbers.
0000  ;       0-53    Don Monroe 28 Mar 80
0000  ;               Fixed bug introduced by version 52.
0000  ;       0-54    Don Monroe 2-Apr-80
0000  ;               Changed the LDINDEX pseudo op. See the description in the
0000  ;               Library for more details.
0000  ;       0-55    Don Monroe 3-Apr-80
0000  ;               Fixed bug introduced in version 54.
0000  ;       0-56    Don Monroe 30-Apr-80
0000  ;               Changed source operand index on LOADDCS instruction to
0000  ;               index by the size of the load.
0000  ;               Added CCS module name and renamed MCT to CMC. Also changed
0000  ;               return after end of pass to micro-monitor instead of RDM.
0000  ;       0-57    Dave Spain 5-May-80
```

```
 0000     221 ;                    Added missing CCS module file name string. Also rearranged
 0000     222 ;                    the file names so that CCS = ECKAE and CMC = ECKAG.
 0000     223 ;          0-58      Bill Landry 6-May-80
 0000     224 ;                    Deleted UBI and CCS from file name strings. Changed file
 0000     225 ;                    names so that CMC = ECKAE and FPA = ECKAD.
 0000     226 ;          0-59      Don Monroe 18-May-80
 0000     227 ;                    Added trace flag and typeout of test titles.
 0000     228 ;          0-60      Don Monroe 23-May-80
 0000     229 ;                    Fixed bug in Trace Flag.
 0000     230 ;          0-61      Don Monroe 3-Jun-80
 0000     231 ;                    Fixed bug in Type Test Number routine.
 0000     232 ;          0-62      Don Monroe 4-Jun-80
 0000     233 ;                    Fixed more bugs in type test number routine.
 0000     234 ;          0-63      Don Monroe 6-JUN-80
 0000     235 ;                    Fixed more bugs in test number typeout.
 0000     236 ;                    Added L000x numbers to module name typeouts.
 0000     237 ;          0-64      Don Monroe 9-JUN-80
 0000     238 ;                    Changed RDM tape name to ECKAF.
 0000     239 ;          0-65      Don Monroe 11-Jun-80
 0000     240 ;                    Added DC6xx numbers to INT and CON.
 0000     241 ;                    Fixed test number typeout so lines contain same number of tests
 0000     242 ;          0-66      Don Monroe 11-Jun-80
 0000     243 ;                    Fixed module typeout code to index the correct table.
 0000     244 ;          0-67      Don Monroe 12-JUN-80
 0000     245 ;                    Fixed bugs in module typeout code.
 0000     246 ;          1-00      Don Monroe 16-JUN-80
 0000     247 ;                    Initial Release
 0000     248 ;          1-01      Don Monroe 26-JUN-80
 0000     249 ;                    Added pass count option to QA Flag. Conditions the QA flag
 0000     250 ;                    on a specified pass count.
 0000     251 ;          1-02      Don Monroe 3-Jul-80
 0000     252 ;                    Fixed bug in BURST CLOCK routine related to stepping the
 0000     253 ;                    clock with the LOOP & ERROR flags set and an INHIBIT on the
 0000     254 ;                    Burst Clock call.
 0000     255 ;          1-03      Don Monroe 3-Jul-80
 0000     256 ;                    Fixed bug in typeout of Test PC.
 0000     257 ;          1-04      Don Monroe 23-Jul-80
 0000     258 ;                    Added a feature to BURST CLOCK that enables the detection
 0000     259 ;                    of the clock being hung. Added CMI BUS name to module name
 0000     260 ;                    list.
 0000     261 ;          1-05      Don Monroe/Bill Landry 27-AUG-80
 0000     262 ;                    Altered monitor to load DCS with NOPS before using any micro-
 0000     263 ;                    code.
 0000     264 ;          1-06      Don Monroe 9-Sep-80
 0000     265 ;                    Modified LOADREG routine so that if the register is the RD
 0000     266 ;                    Control Register, the state of the front panel lights is
 0000     267 ;                    copied into it.
 0000     268 ;          1-07      Don Monroe 10-Sep-80
 0000     269 ;                    Added the ERRLOG and DUMPLOG pseudo instructions. Also added
 0000     270 ;                    the CMA (M8728) to the module callout list.
 0000     271 ;          1-08      Don Monroe 16-Sep-80
 0000     272 ;                    Fixed bugs in DUMP LOG routine.
 0000     273 ;          1-09      Don Monroe 16-Sep-80
 0000     274 ;                    Fixed bugs in ERROR LOG routine. Added setting and clearing
 0000     275 ;                    of error flags in ERROR LOG routine.
```

```
0000  ;      1-10     Bill Landry 19-Sep-80
0000  ;               Fixed bug in ERROR LOG routine to clear INIT FLAG after first
0000  ;               pass.
0000  ;      1-11     Don Monroe 23-Sep-1980
0000  ;               Fixed more bugs in error log routine.
0000  ;      2-00     Don Monroe 29-Sep-1980
0000  ;               Second Release
0000  ;      2-01     Don Monroe 7-Oct-80
0000  ;               Modified Gate Array list to support bit slice algorithm in
0000  ;               error report routine. Added the bit slice algorithm.
0000  ;      2-02     Don Monroe 20-Oct-80
0000  ;               Fixed SHOW VBUS routine to ensure ctrl file is not holding
0000  ;               vbus in load state.
0000  ;      2-03     Don Monroe 17-Nov-1980
0000  ;               Fixed bug in CHECK_QV routine.
0000  ;      3-00     Don Monroe 17-NOV-1980
0000  ;               Third Release
0000  ;      3-01     Don Monroe 29-Dec-1980
0000  ;               Added DI DM command to invoke only the tests for the RDM.
0000  ;      3-02     Don Monroe 30-Dec-1980
0000  ;               Fixed bugs.
0000  ;      3-03     Don Monroe 30-Dec-1980
0000  ;               Fixed bugs.
0000  ;      3-04     Don Monroe 31-Dec-1980
0000  ;               Fixed bugs.
0000  ;      4-00     Don Monroe 5-Jan-1980
0000  ;               Added code to enable the clock interrupts if key switch is
0000  ;               in remote position.
0000  ;      4-01     Don Monroe 17-Feb-81
0000  ;               Added PRINT_N and PRINT_S pseudo instructions.
0000  ;               Added NOTEQUAL condition to SKIP pseudo instruction.
0000  ;      5-00     Don Monroe  2-Mar-1981
0000  ;               Release.
0000  ;      5-01     Bill Landry 19-MAY-1981
0000  ;               Added code to fix ERROR_LOG ROUTINE. Routine would take
0000  ;               fail exit if entry was already in log.
0000  ;      5-02     Don Monroe 21-May-81
0000  ;               Started work on EXAMINE command.
0000  ;      5-03     Don Monroe 3-Jun-81
0000  ;               Fixed bugs in EXAMINE routine.
0000  ;      5-04     Don Monroe 8-Jun-81
0000  ;               Fixed bugs in EXAMINE routine.
0000  ;               Added an initialize of DCS on entry to the monitor.
0000  ;      5-05     Don Monroe 8-Jun-81
0000  ;               Moved the clock stop code from program init to entry
0000  ;               point into monitor so EXAMINE's would work on TE/C cmd.
0000  ;      5-06     Don Monroe 12-Jun-81
0000  ;               Fixed the RDM_WDR micro word.
0000  ;      5-07     Don Monroe 12-Jun-81
0000  ;               Fixed mask on examine of SF and SC.
0000  ;      6-00     Don Monroe 23-Jun-81
0000  ;               Changed revision for release.
0000  ;      6-01     Don Monroe 23-Jul-81
0000  ;               Changed SET STEP CYCLE/TICK routines to clear the
0000  ;               other flag. Modified typeout routine to automatically
```

```
0000  ;                 insert a Carriage Return Line Feed after 80 characters.
0000  ;        6-02     Don Monroe 24-Aug-81
0000  ;                 Added new Gate Array names for FPA module.
0000  ;        6-03     Don Monroe  2-Sep-81
0000  ;                 Masked PSL data on PSL EXAMINE.
0000  ;        6-04     Dave Spain  4-Sep-81
0000  ;                 Added missing FPA gate arrays.
0000  ;        6-05     Don Monroe 8-Sep-81
0000  ;                 Fixed bug in FPA bit-sliced name tables.
0000  ;                 Added ENABLE_STALL routine and modified BRSTCLOCK routine
0000  ;                 to support it.
0000  ;        6-06     Don Monroe 22-Sep-81
0000  ;                 Fixed bug in PSL Examine. The AND with immediate mask used
0000  ;                 the wrong instruction.
0000  ;        6-07     Don Monroe 24-Sep-81
0000  ;                 Modified Single Bit Error Messages. Changed Load Overlay
0000  ;                 Routine to do an Initialize before trying to load cache
0000  ;                 and other internals.
0000  ;        7-00     Don Monroe 10-Nov-81
0000  ;                 Added SBE keyword and flag to DIAGNOSE command to enable
0000  ;                 the DUMP_LOG routine to print the state of the error log.
0000  ;        7-01     Dave Spain 28-Jan-82
0000  ;                 Added L0016 memory controller and MAD gate array, and changed
0000  ;                 CMC and CMA names. L0011 controller is now MC0, L0016 is MC1.
0000  ;                 M8728 256K array card is now MA0, M8750 1M array card is MA1.
0000  ;                 Changed callouts of FCS, FMR and FFA to callout all slices.
0000  ;        7-02     Dave Spain 1-Feb-82
0000  ;                 Fixed bug in FCS, FMR and FFA callouts introduced in 7-01.
0000  ;        7-03     Dave Spain 3-Feb-82
0000  ;                 Replaced references to LOWER 256 KB in correctable bits
0000  ;                 messages to ARRAY 0.
0000  ;        7-04     Dave Spain 19-Feb-82
0000  ;                 Removed CMC and CMA module references from error callouts.
0000  ;                 Now calls out Memory Controller and Memory Array Cards under
0000  ;                 the L and M module numbers only.
0000  ;        7-05     Dave Spain 8-Mar-82
0000  ;                 Fixed bug in ERROR_LOG routine. Wasn't checking test address
0000  ;                 against top of error log correctly.
0000  ;        7-06     Dave Shull 10-Mar-82
0000  ;                 Changed CRD messages (LOG_MSG_2 & LOG_MSG_3) to reference
0000  ;                 the memory array card as ARRAY versus ARRAY 0.  This allows
0000  ;                 sharing of message for either array 0, array 1, etc.
0000  ;        7-07     Dave Shull 22-Mar-82
0000  ;                 Changed the DUMP_LOG routine so it would set the ERR_LOG_INIT
0000  ;                 flag.  This way the combinations of ERRLOG and DUMPLOG pseudo's
0000  ;                 can be used more then once within the same micro to check more
0000  ;                 than one array.
0000  ;        7-08     Dave Shull 06-Apr-82
0000  ;                 Changed the TYPE_ERROR and TYPE_GA_LIST routines so they would
0000  ;                 print the failing Gate Array(s) and Module(s) in column.
0000  ;        8-00     Dave Shull 08-Apr-82
0000  ;                 Removed the BEGINSA and ENDSA pseudo execution code which
0000  ;                 included the following:
0000  ;                         -  Removed symbol SA_ADDRESS:
0000  ;                         -  Removed BEGIN_SA and END_SA from INSTR_TABLE:
```

```
0000  ;        8-01    Dave Spain 05-May-82
0000  ;                Changed bit-slice tables for FPA gate arrays to enhance
0000  ;                isolation for error callouts.
0000  ;        8-02    Dave Shull 11-May-82
0000  ;                Changed the LOAD_INDEX: routine so that it will handle the case
0000  ;                where there is an address specified that points to the value to
0000  ;                be loaded into the index, with No index specified for that
0000  ;                address.
0000  ;        8-03    Dave Shull 03-Jun-82
0000  ;                Fixed the COMPARE_VBUS: routine so it would work with an index
0000  ;                greater than 80 (hex).  Prior to this fix, values greater than
0000  ;                80 (hex) would no get processed correctly.
0000  ;        8-04    Dave Shull 14-Jun-82
0000  ;                Changed CMK callout to CML for new gate array.
0000  ;        8-05    Dave Shull 03-Nov-82
0000  ;                Changed the CCS module callout from L0005 to L0008.  This is
0000  ;                for the new PCS module.
0000  ;        8-06    Joe Zagarella 20-dec-85
0000  ;                Modified to be compatible with the L0003-YA module. Added PC_WB
0000  ;                to WCTRL field of microword that does LONLIT_[0] in DISABLE_CMI
0000  ;                subroutine. By loading the PC, the XB does a prefetch and is
0000  ;                filled with valid data and therefore will not try to do any
0000  ;                more prefetching. It was the prefetching of the XB that
0000  ;                indirectly caused a glitch on the PAD lines during a write to
0000  ;                cache.
0000  ;        8-07    Joe Zagarella 11-feb-86
0000  ;                Added module callouts for new memory controller(L0022) and
0000  ;                new 4mb array card(M7199)
0000  ;
0000  ;--
```

```
                         0000     417            .LIST    MC
                         0000     418            .NLIST   CND,ME,MD
                         0000     419            .SBTTL   DECLARATIONS
                         0000     420
                         0000     421 ;
                         0000     422 ; INCLUDE FILES:
                         0000     423 ;
                         0000     424            .LIBRARY /WRK$LIBRARY:8085MAC/
                         0000     425            .LIBRARY /WRK$LIBRARY:COMETMAC/
                         0000     426            .LIBRARY /WRK$LIBRARY:RDMMAC/
                         0000     427
                         0000     428 ;
                         0000     429 ; MACROS:
                         0000     430 ;
                         0000     431
                         0000     463
                         0000     489
                         0000     517
                         0000     548
                         0000     579
                         0000     609
                         0000     655
                         0000     706
                         0000     729
                         0000     753
                         0000     766
                         0000     784
                         0000     785
                         0000     817
                         0000     842
                         0000     850
```

```
                          0000    852         .SBTTL  "EQUATED SYMBOLS
                          0000    853 ;
                          0000    854 ; EQUATED SYMBOLS:
                          0000    855 ;
                          0000    856         EQUATE                          ; THIS MACRO IS IN 8085MAC. DEFINES
                          0000
                          0000
                          0000          ;
                          0000          ; 8085 REGISTER DEFINITIONS
                          0000          ;
                          0000          ;
                          0000          ; REGISTER PAIRS
                          0000          ;
                00000000  0000                  B       =       0
                00000002  0000                  D       =       2
                00000004  0000                  H       =       4
                00000006  0000                  $SP     =       6
                00000006  0000                  $PSW    =       6
                          0000          ;
                          0000          ; NON REGISTER PAIRS
                          0000          ;
                00000001  0000                  C       =       1
                00000003  0000                  E       =       3
                00000005  0000                  L       =       5
                00000006  0000                  M       =       6
                00000007  0000                  A       =       7
                          0000          ;
                          0000          ; 8085 ASSEMBLY LANGUAGE DEFINITION MACRO'S
                          0000          ;
                          0000                  .MCALL  MOV,MVI,LXI,STAX,LDAX,STA,LDA,SHLD,LHLD,XCHG
                          0000                  .MCALL  PUSH,POP,XTHL,SPHL,JMP,JC,JNC,JZ,JNZ,JP,JM,JPE,JPO,PCHL
                          0000                  .MCALL  CALL,CC,CNC,CZ,CNZ,CP,CM,CPE,CPO
                          0000                  .MCALL  RET,RC,RNC,RZ,RNZ,RP,RM,RPE,RPO,RST
                          0000                  .MCALL  INR,DCR,INX,DCX,ADD,ADC,ADI,ACI,DAD,SUB,SBB,SUI,SBI
                          0000                  .MCALL  ANA,XRA,ORA,CMP,ADI,XRI,ORI,CPI
                          0000                  .MCALL  RLC,RRC,RAL,RAR,CMA,STC,CMC,NOP,HLT,RIM,SIM
                          0000    857                                           ; 8085 REGISTERS. THIS
                          0000    858                                           ; MACRO CALL MUST APPEAR BEFORE ANY CODE
                0000    860                                                     ; IN COMETMAC. DEFINES EQUATED SYMBOLS
                0000    861                                                     ; AND "HEAD" WHICH IS USED TO CALCULATE
                0000    862                                                     ; ALL ADDRESS OFFSETS.
                0000    863             EQUATE_SYMBOLS  1
                0000
       FFFF7C00 0000                    HEAD = -^X8400
                0000            ;
                0000            ; RDM ADDRESS DEFINITIONS:
                0000            ;
       00001800 0000                    DCS_START       =       ^X1800  ; START ADDRESS OF DCS
                0000            ;
                0000            ; RDM RETURN ERROR CODE DEFINITIONS:
                0000            ;
       0000000A 0000                    $CONTROL_C_CODE =       ^XA
       00000009 0000                    $FILE_NOT_FOUND =       9
                0000            ;
                0000            ; SOFTWARE CONTROL REGISTER BIT DEFINITIONS:
                0000            ;
       00000001 0000                    CONTROL_C_FLAG  =       1
       00000002 0000                    CONTINUE_FLAG   =       2
       00000004 0000                    DIAGNOSE_FLAG   =       4
       00000008 0000                    FETCH_FLAG      =       8
       00000010 0000                    LOST_FLAG       =       ^X10
       00000020 0000                    CONT_FLAG       =       ^X20
       00000040 0000                    TSTSPAN_FLAG    =       ^X40
       00000080 0000                    BURST_STOP_FLAG =       ^X80
                0000            ;
                0000            ; SOFTWARE CONTROL REGISTER 'A' BIT DEFINITIONS:
                0000            ;
       00000001 0000                    SOMM_FLAG       =       1
       00000002 0000                    TICK_FLAG       =       2
       00000004 0000                    INSTR_FLAG      =       4
       00000008 0000                    CYCLE_FLAG      =       8
       00000010 0000                    ERROR_FLAG      =       ^X10
       00000020 0000                    LOOP_ERROR_FLAG =       ^X20
       00000040 0000                    EXP_UTRAP_FLAG  =       ^X40
       00000080 0000                    VBUS_COMPARE    =       ^X80
                0000            ;
                0000            ; SOFTWARE CONTROL REGISTER 'B' BIT DEFINITIONS:
                0000            ;
       00000001 0000                    ERR_LOG_INIT    =       1
       00000002 0000                    QV_FLAG         =       2
       00000004 0000                    RDM_FLAG        =       4
       00000008 0000                    RDM_LAST        =       8
       00000010 0000                    EXP_STALL_FLAG  =       ^X10
       00000020 0000                    SBE_FLAG        =       ^X20
                0000            ;
                0000            ; PROGRAM CONTROL REGISTER BIT DEFINITIONS:
                0000            ;
       00000001 0000                    HALT_FLAG       =       1
       00000002 0000                    LOOP_FLAG       =       2
       00000004 0000                    NER_FLAG        =       4
       00000008 0000                    BELL_FLAG       =       8
       00000010 0000                    SA_FLAG         =       ^X10
       00000020 0000                    IB_FLAG         =       ^X20
```

```
00000040  0000               QA_FLAG          =         ^X40
00000080  0000               TR_FLAG          =         ^X80
          0000      ;
          0000      ; RDM REGISTER DEFINITIONS:
          0000      ;
          0000      ; NOTE: THESE REGITER DEFINITIONS ARE OFFSET BY THE LOAD/START ADDRESS OF
          0000      ;       THE PROGRAM. FOR THE SIMULATOR VERSION THIS IS 400(X) AND FOR THE
          0000      ;       REAL VERSION IT IS 8400(X).
          0000      ;
00007400  0000               DCS_DATA_REG     =         ^XF800 + HEAD
00007401  0000               DCS_ADDR_REG_WO  =         ^XF801 + HEAD
00007402  0000               DCS_CTRL_REG     =         ^XF802 + HEAD
00007403  0000               DCS_CTRL_FILE    =         ^XF803 + HEAD
00007404  0000               ADDR_MATCH_LO    =         ^XF804 + HEAD
00007405  0000               ADDR_MATCH_HI    =         ^XF805 + HEAD
00007406  0000               STATUS_REG       =         ^XF806 + HEAD
          0000      ;
00007410  0000               A_REG_BYTE_0     =         ^XF810 + HEAD
00007411  0000               A_REG_BYTE_1     =         ^XF811 + HEAD
00007412  0000               A_REG_BYTE_2     =         ^XF812 + HEAD
00007413  0000               A_REG_BYTE_3     =         ^XF813 + HEAD
00007414  0000               MD_REG_BYTE_0    =         ^XF814 + HEAD
00007415  0000               MD_REG_BYTE_1    =         ^XF815 + HEAD
00007416  0000               MD_REG_BYTE_2    =         ^XF816 + HEAD
00007417  0000               MD_REG_BYTE_3    =         ^XF817 + HEAD
00007418  0000               MH_REG_BYTE_0    =         ^XF818 + HEAD
00007419  0000               MH_REG_BYTE_1    =         ^XF819 + HEAD
0000741A  0000               MH_REG_BYTE_2    =         ^XF81A + HEAD
0000741B  0000               MH_REG_BYTE_3    =         ^XF81B + HEAD
0000741C  0000               CMI_CTRL_REG     =         ^XF81C + HEAD
0000741D  0000               MAINT_REG        =         ^XF81D + HEAD
0000741E  0000               RD_CTRL_REG      =         ^XF81E + HEAD
          0000      ;
00007420  0000               FRONT_PNL_1      =         ^XF820 + HEAD
00007421  0000               FRONT_PNL_2      =         ^XF821 + HEAD
00007422  0000               CS_ADD_TRAP_LOW  =         ^XF822 + HEAD
00007423  0000               CS_ADD_TRAP_HGH  =         ^XF823 + HEAD
00007424  0000               CS_ADD_BUFF_LOW  =         ^XF824 + HEAD
00007425  0000               CS_ADD_BUFF_HGH  =         ^XF825 + HEAD
00007426  0000               BUFF_ADDR_LOW    =         ^XF826 + HEAD
00007427  0000               DCS_ADDR_REG_RO  =         ^XF827 + HEAD
          0000      ;
0000742C  0000               WD_REG_BYTE_0    =         ^XF82C + HEAD
0000742D  0000               WD_REG_BYTE_1    =         ^XF82D + HEAD
0000742E  0000               WD_REG_BYTE_2    =         ^XF82E + HEAD
0000742F  0000               WD_REG_BYTE_3    =         ^XF82F + HEAD
00007430  0000               WH_REG_BYTE_0    =         ^XF830 + HEAD
00007431  0000               WH_REG_BYTE_1    =         ^XF831 + HEAD
00007432  0000               WH_REG_BYTE_2    =         ^XF832 + HEAD
00007433  0000               WH_REG_BYTE_3    =         ^XF833 + HEAD
          0000      ;
          0000      ; RDM REGISTER NAMES AS DEFINED IN HARDWARE SPEC
          0000      ;
00007400  0000               DCSDA=DCS_DATA_REG                ;Diagnostic control store data register
00007401  0000               DCSAD=DCS_ADDR_REG_WO             ;Diagnostic control store address reg
```

```
        00007402  0000                  DCSCR=DCS_CTRL_REG                ;Diagnostic control store control reg
        00007403  0000                  DCSCF=DCS_CTRL_FILE               ;Diagnostic control store control file
        00007404  0000                  CSAMR=ADDR_MATCH_LO               ;Control store address match register
        00007406  0000                  STATUS=STATUS_REG                 ;Status register
        00007410  0000                  MAREG=A_REG_BYTE_0                ;Memory address register
        00007414  0000                  MDREG=MD_REG_BYTE_0               ;Memory data register
        00007418  0000                  MHREG=MH_REG_BYTE_0               ;Memory holding register
        0000741C  0000                  CMICTL=CMI_CTRL_REG               ;CMI control
        0000741D  0000                  MAIN32=MAINT_REG                  ;32 Bit path maintenance control
        0000741E  0000                  RDCTRL=RD_CTRL_REG                ;Remote diagnosis control
        00007420  0000                  FRONT1=FRONT_PNL_1                ;Front panel readback 1
        00007421  0000                  FRONT2=FRONT_PNL_2                ;Front panel readback 2
        00007422  0000                  CSTRP=CS_ADD_TRAP_LOW             ;Control store address trap register
        00007424  0000                  CSBUF=CS_ADD_BUFF_LOW             ;Control store address buffer
        00007426  0000                  BADD1=BUFF_ADDR_LOW               ;Control store address trace readback
        00007427  0000                  CLKDCS=DCS_ADDR_REG_RO            ;Clock control & DCS address register
        0000742C  0000                  WDREG=WD_REG_BYTE_0               ;WBUS data register
        00007430  0000                  WHREG=WH_REG_BYTE_0               ;WBUS holding register
                  0000          ;
                  0000          ; RDM REGISTER BIT DEFINITIONS:
                  0000          ;
                  0000          ; DCS CONTROL REGISTER
        00000001  0000                  HALT_ON_MATCH    =      1
        00000002  0000                  CLEAR_CTRL_FILE  =      2
        00000004  0000                  TRACE_ENABL      =      4
        00000008  0000                  PAR_CHK_ENABL    =      8
        00000010  0000                  PAR_STOP_ENABL   =      ^X10
        00000020  0000                  CLK_CTRL_0       =      ^X20     ; ACTIVE LOW
        00000040  0000                  CLK_CTRL_1       =      ^X40     ; ACTIVE LOW
        00000080  0000                  MICRO_ADDR_INH   =      ^X80
                  0000              ;
                  0000              ; FOLLOWING IS A FUNCTION TABLE OF THE CLOCK CONTROL BITS
                  0000              ;
                  0000                  ;       CLK_CTRL 1,0    FUNCTION
                  0000                  ;            11            HALT
                  0000                  ;            01            SINGLE MICRO INSTRUCTION
                  0000                  ;            10            SINGLE TICK
                  0000                  ;            00            RUN
                  0000          ;
                  0000          ; DCS CONTROL FILE      ALL OF THESE BITS ARE ACTIVE LOW
                  0000          ;
        00000001  0000                  V_BUS_STROBE     =      1
        00000002  0000                  DCS_ADDR_CLEAR   =      2
        00000004  0000                  STOP_CLOCK       =      4
        00000008  0000                  EN_TRACE_REG     =      8
        00000010  0000                  H_FROM_WBUS      =      ^X10
        00000020  0000                  WBUS_FROM_D      =      ^X20
        00000040  0000                  STROBE_CMI       =      ^X40
        00000080  0000                  SA_CLOCK         =      ^X80
                  0000          ;
                  0000          ; CS ADDRESS MATCH REGISTER (HIGH)
                  0000          ;
        00000040  0000                  VBUS_SERIAL_IN   =      ^X40     ; ACTIVE LOW
        00000080  0000                  VBUS_CLOCK       =      ^X80
                  0000          ;
```

```
          0000        ; STATUS REGISTER
          0000        ;
00000001  0000                VBUS_SERIAL_OUT =       1
00000002  0000                TRAP_OFF        =       2
00000008  0000                CMI_STATUS_0    =       8
00000010  0000                CMI_STATUS_1    =       ^X10
          0000                ;
          0000                ; THE CMI STATUS BITS ARE ENCODED AS FOLLOWS:
          0000                ;
          0000                ;         00 = NO RESPONSE
          0000                ;         01 = UNCORRECTABLE ERROR
          0000                ;         10 = CORRECTABLE ERROR
          0000                ;         11 = NO ERRORS
00000020  0000                CMI_COMPLETE    =       ^X20
          0000        ;
00000080  0000                CLOCK_STOPED    =       ^X80
          0000        ;
          0000        ; CMI CONTROL REGISTER
          0000        ;
00000001  0000                CMI_CTRL_CLEAR  =       1
00000008  0000                CMI_GO          =       8
00000020  0000                CMI_WRITE       =       ^X20
00000040  0000                CMI_READ        =       ^X40
00000080  0000                SA_START_STOP   =       ^X80
          0000        ;
          0000        ; 32 BIT MAINTENANCE CONTROL REGISTER
          0000        ;
00000001  0000                MAINT_TRAP_OFF  =       1
00000002  0000                MAINT_STB_INH   =       2
00000004  0000                MAINT_DCS_ENABL =       4
00000008  0000                MAINT_WB_TO_WH  =       8
00000010  0000                MAINT_WD_TO_WB  =       ^X10
00000020  0000                MAINT_CMI_TO_MH =       ^X20
00000040  0000                MAINT_MD_TO_CMI =       ^X40
00000080  0000                MAINT_A_TO_CMI  =       ^X80
          0000        ;
          0000        ; RD CONTROL REGISTER
          0000        ;
00000008  0000                MASTER_HALT_EN  =       8
00000010  0000                VBUS_LOAD       =       ^X10
00000020  0000                TRAP_HALT_EN    =       ^X20
00000040  0000                DC_LOW          =       ^X40
00000080  0000                AC_LOW          =       ^X80
          0000        ;
          0000        ; FRONT PANEL REGISTER 1
          0000        ;
00000001  0000                FP_BOOT_0       =       1
00000002  0000                FP_BOOT_1       =       2
00000004  0000                FP_START_0      =       4
00000008  0000                FP_START_1      =       8
00000010  0000                CPU_RUN         =       ^X10
00000020  0000                PARITY_ERROR    =       ^X20
          0000        ;
          0000        ; FRONT PANEL REGISTER 2
          0000        ;
          00000001  0000                REMOTE          =       1
          00000002  0000                FAULT           =       2
          00000004  0000                TEST            =       4
                    0000        ;
                    0000        ;
                    0000        ;
          00000040  0000                PB_INIT         =       ^X40    ; ACTIVE LOW
          00000080  0000                FRONT_PNL_LOCK  =       ^X80
                    0000        ;
                    0000        ; DCS ADDRESS REGISTER READ ONLY
                    0000        ;
          00000040  0000                CLK_CTRL_0_RO   =       ^X40    ; ACTIVE HIGH
          00000080  0000                CLK_CTRL_1_RO   =       ^X80    ; ACTIVE HIGH
                    0000        ;
                    0000        ; KEYWORD LIST OFFSET DEFINTIONS:
                    0000        ;
          00000000  0000                TEST_KEY        =       0
          00000002  0000                PASS_KEY        =       2
          00000004  0000                FLAG_KEY        =       4
          00000006  0000                SOMM_KEY        =       6
          00000008  0000                HALT_KEY        =       8
          0000000A  0000                LOOP_KEY        =       10
          0000000C  0000                NER_KEY         =       12
          0000000E  0000                BELL_KEY        =       14
          00000010  0000                SA_KEY          =       16
          00000012  0000                IB_KEY          =       18
          00000014  0000                QA_KEY          =       20
          00000016  0000                TR_KEY          =       22
          00000018  0000                INSTR_KEY       =       24
          0000001A  0000                LOST_KEY        =       26
          0000001C  0000                CONT_KEY        =       28
          0000001E  0000                STEP_KEY        =       30
          00000020  0000                CYCLE_KEY       =       32
          00000022  0000                TICK_KEY        =       34
          00000024  0000                CF_KEY          =       36
          00000026  0000                VB_KEY          =       38
          00000028  0000                QV_KEY          =       40
          0000002A  0000                RDM_KEY         =       42
          0000002C  0000                RT_KEY          =       44
          0000002E  0000                MT_KEY          =       46
          00000030  0000                PC_KEY          =       48
          00000032  0000                VA_KEY          =       50
          00000034  0000                MD_KEY          =       52
          00000036  0000                PB_KEY          =       54
          00000038  0000                MA_KEY          =       56
          0000003A  0000                WD_KEY          =       58
          0000003C  0000                SR_KEY          =       60
          0000003E  0000                PS_KEY          =       62
          00000040  0000                SF_KEY          =       64
          0000001E  0000                ST_KEY          =       STEP_KEY
          00000042  0000                SBE_KEY         =       66
                    0000        ;
                    0000        ; THE FOLLOWING EQUATES ARE NOT USED BY THE 1
                    0000        ;
                    0000        ; PSEUDO INSTRUCTION OP CODE DEFINITIONS:
```

```
                   0000        ;
         00000000  0000                OP_LOAD_DCS      =       0
         00000002  0000                OP_BURST_CLOCK   =       2
         00000004  0000                OP_COMPARE_REG   =       4
         00000006  0000                OP_COMPARE_REGM  =       6
         00000008  0000                OP_BEGIN_LOOP    =       8
         0000000A  0000                OP_END_LOOP      =       10
         0000000C  0000                OP_END_TEST      =       12
         0000000E  0000                OP_ERROR_LOOP    =       14
         00000010  0000                OP_PATTERN_GEN   =       16
         00000012  0000                OP_IF_ERROR      =       18
         00000014  0000                OP_INIT          =       20
         00000016  0000                OP_LOAD_REG      =       22
         00000018  0000                OP_MASK          =       24
         0000001A  0000                OP_NEW_TEST      =       26
         0000001C  0000                OP_SKIP          =       28
         0000001E  0000                OP_SUB_TEST      =       30
         00000020  0000                OP_COMPARE_VB    =       32
         00000022  0000                OP_FETCH         =       34
         00000024  0000                OP_END_FILE      =       36
         00000026  0000                OP_LOAD_FIELD    =       38
         00000028  0000                OP_LOAD_INDEX    =       40
         0000002A  0000                OP_INC_INDEX     =       42
         0000002C  0000                OP_SAVE_INDEX    =       44
         0000002E  0000                OP_ENABL_TRAP    =       46
         00000030  0000                OP_DUMPLOG       =       48
         00000032  0000                OP_ERRLOG        =       50
         00000034  0000                OP_PRINT_S       =       52
         00000036  0000                OP_PRINT_N       =       54
         00000038  0000                OP_ENABL_STALL   =       56
                   0000        ;
                   0000        ; GATE ARRAY NAME DEFINITIONS:
                   0000        ;
         00000000  0000                ALK              =       0
         00000001  0000                SRK              =       1
         00000002  0000                SPA              =       2
         00000003  0000                CLA              =       3
         00000004  0000                CCC              =       4
         00000005  0000                IRD              =       5
         00000006  0000                TOK              =       6
         00000007  0000                PHB              =       7
         00000008  0000                MSQ              =       8
         00000009  0000                SAC              =       9
         0000000A  0000                CAK              =       10
         0000000B  0000                ADK              =       11
         0000000C  0000                PRK              =       12
         0000000D  0000                UTR              =       13
         0000000E  0000                CML              =       14
         0000000F  0000                ACV              =       15
         00000010  0000                INT              =       16
         00000011  0000                CON              =       17
         00000012  0000                MAP              =       18
         00000013  0000                MAD              =       19
         00000014  0000                FQA              =       20
         00000015  0000                FCC              =       21
```

```
                 0000      ;
                 0000      ; THE FOLLOWING GATE ARRAY'S ARE IN A SPECIAL GROUP BECAUSE THEY ARE
                 0000      ; BIT SLICED. THE ERROR ROUTINE TRYS TO FIGURE OUT WHICH SLICE IS BAD
                 0000      ; IF ANY ONE OF THESE GATE ARRAYS IS IN THE CALLOUT LIST.
                 0000      ;
        00000016 0000              ALP             =       22
        00000017 0000              SRM             =       23
        00000018 0000              MDR             =       24
        00000019 0000              ADD             =       25
        0000001A 0000              MEC             =       26
        0000001B 0000              MDL             =       27
        0000001C 0000              UDP             =       28
        0000001D 0000              FIO             =       29
        0000001E 0000              FEX             =       30
        0000001F 0000              FIC             =       31
        00000020 0000              FAC             =       32
        00000021 0000              FCS             =       33
        00000022 0000              FMR             =       34
        00000023 0000              FFL             =       35
        00000024 0000              FFH             =       36
        00000025 0000              END_SLICE       =       37 ; ONE MORE THAN LAST BIT-SLICED ARRAY
                 0000    ;
                 0000    ; MODULE NAME DEFINITIONS:
                 0000    ;
        00000000 0000              DPM             =       0
        00000001 0000              MIC             =       1
        00000002 0000              FPA             =       2
        00000003 0000              MC0             =       3
        00000004 0000              RDM             =       4
        00000005 0000              CCS             =       5
        00000006 0000              UBI             =       6
        00000007 0000              CMI             =       7
        00000008 0000              MA0             =       8
        00000009 0000              MC1             =       9
        0000000A 0000              MA1             =       10
        0000000B 0000              MC2             =       11
        0000000C 0000              MA2             =       12
                 0000    ;
                 0000    ; MISCELLANEOUS EQUATES:
                 0000    ;
        00000001 0000              BYTE            =       1
        00000002 0000              WORD            =       2
        00000004 0000              LONG            =       4
        0000000A 0000              MICROWORD       =       10
        00002C5E 0000              MONITOR_SIZE    =       ^X2C5E
        00000400 0000              STACK_SIZE      =       1024
        000007A2 0000              M$TEST_SIZE     =       14336 - STACK_SIZE - MONITOR_SIZE
                 0000                                              ; MAXIMUM SIZE OF A TEST OVERLAY
        000003E2 0000              M$TEST_SIZE_D   = M$TEST_SIZE - 960
                 0000                                              ; MAXIMUM SIZE OF TEST OVERLAY WITH
                 0000                                              ; 'DEBUG' ENABLED IN 1
        00000001 0000              HIGH            =       1
        00000000 0000              LOW             =       0
        00000000 0000              ONERROR         =       0
        00000001 0000              NOERROR         =       1
```

```
00000002  0000              ONEQUAL           =     2
00000003  0000              NOTEQUAL          =     3
00000010  0000              MAX_ARGUMENTS     =     16       ; BYTE LENGTH OF LARGEST PSEUDO OP
00000036  0000              DCS_SCRATCH_ADR =       54       ; DCS SCRATCH ADDRESS START
000032FC  0000              ERROR_LOG_ADR     =     ^XB6FC + HEAD
          0000                                               ; START ADDRESS OF MEMORY ERROR LOG
          0000     864
```

```
0000               .SBTTL  "LINKAGE FROM RDM SYSTEM
0000
0000        ;
0000        ; LINKAGE FROM RDM SYSTEM
0000        ; THE "COMETMAC" MACRO PACKAGE WILL CAUSE THE STARTING ADDRESS OF THIS
0000        ; FILE TO BE 8400(X).
0000        ;
0000               JMP         START1              ; GETS ASSEMBLED AT 8400(X)
0003               LXI         H,0                 ; GET CURRENT SP
0006               DAD         $SP                 ; ...
0007               SHLD        SAVED_SP            ; SAVE FOR ABORT ROUTINE
000A               ABORT                           ; THIS GETS ASSEMBLED AT LOCATION 8403(X)
000D                                               ; THE RDM CODE WILL COME HERE IF
000D                                               ; IT NEEDS TO ABORT THE MONITOR
000D       CMD_ENTRY:
000D               RET
000E
000E       START1: PUSH        $PSW                ; SAVE THE C BIT
000F               LXI         H,0                 ; GET THE STACK POINTER
0012               DAD         $SP                 ; ...
0013               INX         H                   ; ACCOUNT FOR THE PSW PUSH
0014               INX         H                   ; ...
0015               SHLD        SAVED_SP            ; SAVE IT
0018        ;
0018        ; SCRAMBLE THE MICRO CODE IN THE MONITOR
0018        ;
0018               LXI         H,NOP_MICRO_WORD ; GET START ADDRESS OF MICRO CODE
001B               MVI         A,NO_MICRO_WORDS ; GET NUMBER OF MICRO WORDS
001D       1$:     PUSH        $PSW                ; SAVE THE NUMBER OF MICRO WORDS
001E               PUSH        H                   ; COPY H & L INTO D & E
001F               POP         D                   ; ...
0020               MVI         B,0                 ; SET THE STARTING BIT NUMBER
0022               MVI         C,79                ; AND THE ENDING BIT NUMBER
0024               PUSH        H                   ; SAVE H & L
0025               XRA         A                   ; CLEAR INHIBIT PARITY FLAG
0026               PUSH        $PSW                ; CLEAR INHIBIT SCRAMBLE FLAG
0027               CALL        INSERT_FIELD        ; SCRAMBLE THE MICRO WORD
002A               POP         H                   ; RESTORE H & L
002B               MVI         E,11                ; ADD 11 TO H & L
002D               MVI         D,0                 ; ...
002F               DAD         D                   ; ...
0030               POP         $PSW                ; GET THE NUMBER OF MICRO WORDS
0031               DCR         A                   ; DECREMENT
0032               JNZ         1$                  ; BRANCH IF MORE TO SCRAMBLE
0035        ;
0035        ; STOP THE CLOCK AND INITIALIZE SOME MORE PARAMETERS
0035        ;
0035               MVI         A,<CLK_CTRL_0!CLK_CTRL_1> ; GET DATA TO STOP THE CLOCK
0037               STA         DCS_CTRL_REG        ; STOP THE CLOCK
003A               STA         DCS_CTRL_RE_CPY ; SAVE A COPY OF THE CONTROL REGISTER
003D               LDA         FRONT_PNL_2         ; GET TEST, FAULT, AND REMOTE BITS
0040               ANI         <TEST!FAULT!REMOTE> ; ...
0042               STA         RD_CTRL_REG         ; CLEAR MASTER HALT ENABLE
0045               ORI         MASTER_HALT_EN      ; SET MASTER HALT ENABLE
0047               STA         RD_CTRL_REG         ; ...
```

```
004A          MVI           A,<MAINT_TRAP_OFF!-
004A                        MAINT_DCS_ENABL>; GET DATA TO INIT THE MAINT REG
004C          STA           MAINT_REG        ; LOAD THE MAINT REG
004F          CALL          FILL_DCS         ; FILL DCS WITH NOP'S
0052   ;
0052   ; CHECK ENTRY FLAG
0052   ;
0052          POP           $PSW             ; RESTORE THE C BIT
0053          CC            GET_CMD_LINE     ; ENTER HERE IF 'T/C' COMMAND USED
0056          JMP           PROGRAM_INIT     ; START EXECUTION
0059
```

```
                                    0059         .SBTTL  "OWN STORAGE
                                    0059  ;
                                    0059  ; OWN STORAGE
                                    0059  ;
                                    0059
                                    0059  ;
                                    0059  ; SOFTWARE CONTROL REGISTERS
                                    0059  ;
                                    0059
                               00   0059  SOFT_CTRL_REG:   .BYTE                        ; SOFTWARE CONTROL REGISTER
                               00   005A  SOFT_CTRL_REGA:  .BYTE                        ; SOFTWARE CONTROL REGISTER A
                               01   005B  SOFT_CTRL_REGB:  .BYTE   ERR_LOG_INIT         ; SOFTWARE CONTROL REGISTER B
                               00   005C  DCS_CTRL_RE_CPY: .BYTE                        ; USED TO SIMULATE BIT SETS AND BIT
                                    005D                                                ; CLEARS TO THE DCS CONTROL REGISTER
                                    005D
                                    005D  ;
                                    005D  ; PROGRAM CONTROL REGISTER
                                    005D  ;
                                    005D
                               01   005D  PROG_CTRL_REG:   .BYTE   HALT_FLAG            ; HALT FLAG IS INITIALLY SET
                                    005E
                                    005E  ;
                                    005E  ; PROGRAM_INIT PARAMETERS:
                                    005E  ;
                                    005E
                             0000   005E  SAVED_SP:        .WORD                        ; STACK POINTER ON CALL FROM RDM
                               00   0060  TEST_NUMBER:     .BYTE                        ; CONTAINS THE CURRENT TEST NUMBER
                               00   0061  SUB_TEST_NUMB:   .BYTE                        ; CONTAINS THE CURRENT SUB TEST NUMBER
                               00   0062  PASS_COUNT:      .BYTE                        ; CONTAINS THE CURRENT PASS COUNT
                               00   0063  FILE_NAM_INDEX:  .BYTE                        ; CONTAINS AN INDEX FOR FILE NAME LIST
                             FFFF   0064  STARTING_RECORD: .WORD   -1                   ; CONTAINS THE STARTING RECORD NUMBER
                                    0066                                                ; OF THE TEST FILE LAST OPENED
                               00   0066  TOTAL_NMB_TESTS: .BYTE                        ; NUMBER OF TESTS IN THE FILE
                                    0067
                                    0067  PROGRAM_TITLE:  GENERATE_TITLE  \PRIM_REV,\SEC_REV
45 54 20 44 49 4C 41 56 4E 49 3F 00' 0076  DIRECT_MSG:      .ASCIC  /?INVALID TEST NUMBER/ ; SPECIAL TEST IS NOT IN FILE
         52 45 42 4D 55 4E 20 54 53 0082
                               14   0076
                                    008B  MODULE_NAM_TBL: ADDRESS <DPM_NAME>            ; TABLE OF POINTERS TO MODULE NAME
                                    008D                  ADDRESS <MIC_NAME>            ; ASCII STRINGS
                                    008F                  ADDRESS <FPA_NAME>            ; ...
                                    0091                  ADDRESS <MC0_NAME>            ; ...
                                    0093                  ADDRESS <RDM_NAME>            ; ...
                                    0095                  ADDRESS <CCS_NAME>            ; ...
                                    0097                  ADDRESS <UBI_NAME>            ; ...
                                    0099                  ADDRESS <CMI_NAME>            ; ...
                                    009B                  ADDRESS <MA0_NAME>            ; ...
                                    009D                  ADDRESS <MC1_NAME>            ; ...
                                    009F                  ADDRESS <MA1_NAME>            ; ...
                                    00A1                  ADDRESS <MC2_NAME>            ; ...
                                    00A3                  ADDRESS <MA2_NAME>            ; ...
                                    00A5
                                    00A5  MODULE_LST:
  29 32 30 30 30 4C 28 4D 50 44 00' 00A5  DPM_NAME:        .ASCIC  /DPM(L0002)/         ; ASCII MODULE NAMES
                               0A   00A5
```

```
   29 33 30 30 30 4C 28 43 49 4D 00' 00B0  1486 MIC_NAME:       .ASCIC  /MIC(L0003)/     ; ...
                                 0A  00B0
   29 31 30 30 30 4C 28 41 50 46 00' 00BB  1487 FPA_NAME:       .ASCIC  /FPA(L0001)/     ; ...
                                 0A  00BB
   20 29 31 31 30 30 4C 28 20 20 00' 00C6  1488 MCO_NAME:       .ASCIC  /  (L0011) /     ; ...
                                 0A  00C6
   29 36 30 30 30 4C 28 4D 44 52 00' 00D1  1489 RDM_NAME:       .ASCIC  /RDM(L0006)/     ; ...
                                 0A  00D1
   29 38 30 30 30 4C 28 53 43 43 00' 00DC  1490 CCS_NAME:       .ASCIC  /CCS(L0008)/     ; ...
                                 0A  00DC
   29 34 30 30 30 4C 28 49 42 55 00' 00E7  1491 UBI_NAME:       .ASCIC  /UBI(L0004)/     ; ...
                                 0A  00E7
   20 20 53 55 42 20 20 49 4D 43 00' 00F2  1492 CMI_NAME:       .ASCIC  /CMI  BUS  /     ; ...
                                 0A  00F2
   20 29 38 32 37 38 4D 28 20 20 00' 00FD  1493 MA0_NAME:       .ASCIC  /  (M8728) /     ; ...
                                 0A  00FD
   20 29 36 31 30 30 4C 28 20 20 00' 0108  1494 MC1_NAME:       .ASCIC  /  (L0016) /     ; ...
                                 0A  0108
   20 29 30 35 37 38 4D 28 20 20 00' 0113  1495 MA1_NAME:       .ASCIC  /  (M8750) /     ; ...
                                 0A  0113
   20 29 32 32 30 30 4C 28 20 20 00' 011E  1496 MC2_NAME:       .ASCIC  /  (L0022) /     ; ...
                                 0A  011E
   20 29 39 39 31 37 4D 28 20 20 00' 0129  1497 MA2_NAME:       .ASCIC  /  (M7199) /     ; ...
                                 0A  0129
                                     0134  1498
                                     0134  1499 ;
                                     0134  1500 ; TU58 READ PARAMETERS
                                     0134  1501 ;
                                     0134  1502
                               0000  0134  1503 TU58_BUFF_ADDR: .WORD                    ; TU58 BUFFER ADDRESS
                                 00  0136  1504 TU58_REC_COUNT: .BYTE                    ; TU58 RECORD COUNT
                               0000  0137  1505 TU58_REC_NUMB:  .WORD                    ; TU58 RECORD NUMBER
                                     0139  1506
                                     0139  1507 ;
                                     0139  1508 ; TU58 OPEN PARAMETERS
                                     0139  1509 ;
                                     0139  1510
                               0000  0139  1511 FILE_NAM_PTR:   .WORD                    ; CONTAINS ADDRESS OF FILE NAME
                               0000  013B  1512 CURRENT_REC_NMB:.WORD                    ; CONTAINS THE CURRENT TU58 RECORD #
                               0000  013D  1513 TOTAL_BLOCKS:   .WORD                    ; CONTAINS FILE LENGTH IN BLOCKS
                                     013F  1514
                                     013F  1515 ;
                                     013F  1516 ; TU58 ERROR PARAMETERS
                                     013F  1517 ;
                                     013F  1518
                                 00  013F  1519 TU58_ERR_CODE:  .BYTE                    ; TU58 ERROR CODE
52 4F 52 52 45 20 38 35 55 54 3F 00' 0140  1520 TU58_ERROR:     .ASCIC  /?TU58 ERROR: /  ; TU58 ERROR MESSAGE
                              20 3A  014C
                                 0D  0140
                                     014E  1521
                                     014E  1522 ;
                                     014E  1523 ; TU58 FILE NAME STRINGS
                                     014E  1524 ;
                           0000014E  014E  1525                 FILE_NAME_HEAD = .
                                     014E  1526 FILE_FOR_DPM:   RAD50   <ECK>
```

```
                                    0150                  RAD50    <AB >
                                    0152                  RAD50    <EXE>
                                    0154  FILE_FOR_MIC:   RAD50    <ECK>
                                    0156                  RAD50    <AC >
                                    0158                  RAD50    <EXE>
                                    015A  FILE_FOR_FPA:   RAD50    <ECK>
                                    015C                  RAD50    <AD >
                                    015E                  RAD50    <EXE>
                                    0160  FILE_FOR_CMC:   RAD50    <ECK>
                                    0162                  RAD50    <AE >
                                    0164                  RAD50    <EXE>
                                    0166  FILE_FOR_RDM:   RAD50    <ECK>
                                    0168                  RAD50    <AF >
                                    016A                  RAD50    <EXE>
                          0000016C  016C  RDM_TAIL = .                                ; THIS EQUATE MUST ALWAYS IMMEDIATELY
                                    016C                                              ; FOLLOW THE RDM FILE NAME.
                          00000004  016C  RDM_INDEX = <<RDM_TAIL - FILE_NAME_HEAD> / 6> - 1
                                    016C
                          0000016C  016C          FILE_NAME_TAIL = .
                          00000005  016C          NUMBER_OF_FILES = <FILE_NAME_TAIL-FILE_NAME_HEAD>/6
                                    016C
                                    016C  ;
                                    016C  ; TERMINAL OUTPUT PARAMETERS
                                    016C  ;
                                    016C
                              0000  016C  MESSAGE_POINTER:.WORD                       ; CONTAINS POINTER TO ASCII STRING
                                00  016E  MESSAGE_LENGTH: .BYTE                       ; CONTAINS THE LENGTH OF ASCII MESSAGE
                                    016F
                                    016F  ;
                                    016F  ; TERMINAL INPUT PARAMETERS
                                    016F  ;
                                    016F
                          0000018D  016F  TERM_INP_BUFF:  .BLKB   30                  ; INPUT BUFFER FOR TERMINAL READS
                                    018D
                                    018D  ;
                                    018D  ; TERMINAL ERROR PARAMETERS:
                                    018D  ;
                                    018D
                                00  018D  TERM_ERR_CODE:  .BYTE                       ; CONTAINS THE TERMINAL ERROR CODE
45 20 4C 41 4E 49 4D 52 45 54 3F 00'  018E  TERM_ERROR:     .ASCIC  /?TERMINAL ERROR: / ; TERMINAL ERROR MESSAGE
                 20 3A 52 4F 52 52  019A
                                11  018E
                                    01A0  ;
                                    01A0  ; TYPE_NUMERIC PARAMETERS
                                    01A0  ;
                                    01A0
                              0000  01A0  TYPEN_DATA_PTR: .WORD                       ; CONTAINS POINTER TO DATA
                                00  01A2  TYPEN_DATA_TYPE:.BYTE                       ; CONTAINS THE DATA TYPE FLAG
                          000001AC  01A3  TEMP_BUFFER:    .BLKB   9                   ; BUFFER FOR CONVERT CALL
                                    01AC
                                    01AC  ;
                                    01AC  ; WRITE_DCS PARAMETERS
                                    01AC  ;
                                    01AC
                              0000  01AC  DCS_WRITE_ADR:  .WORD                       ; CONTAINS DCS ADDRESS TO START AT
```

```
                                    0000   01AE   1580 DCS_WRITE_PTR:   .WORD                       ; CONTAINS ADDRESS OF DATA TO WRITE
                                      00   01B0   1581 DCS_WRITE_WDCNT:.BYTE                        ; CONTAINS NO. OF MICRO WORDS TO WRITE
                                           01B1   1582
                                           01B1   1583 ;
                                           01B1   1584 ; SINGLE_MIC_INSTR PARAMETERS:
                                           01B1   1585 ;
                                           01B1   1586
                                      00   01B1   1587 MIC_INSTR_CNT:   .BYTE                       ; CONTAINS THE CYCLE COUNT
                                           01B2   1588
                                           01B2   1589 ;
                                           01B2   1590 ; SINGLE_TICK PARAMETERS:
                                           01B2   1591 ;
                                           01B2   1592
                                      00   01B2   1593 MIC_TICK_CNT:    .BYTE                       ; CONTAINS THE TICK COUNT
                                           01B3   1594
                                           01B3   1595 ;
                                           01B3   1596 ; SET/CLEAR CONTROL FILE PARAMETERS:
                                           01B3   1597 ;
                                           01B3   1598
                                      00   01B3   1599 SETCLR_CF_ADDR: .BYTE                        ; CONTAINS DCS ADDRESS
                                      00   01B4   1600 SETCLR_CF_DATA: .BYTE                        ; CONTAINS THE DATA TO SET AND CLEAR
                                           01B5   1601
                                           01B5   1602 ;
                                           01B5   1603 ; READ OVERLAY PARAMETERS:
                                           01B5   1604 ;
                                           01B5   1605
                                    0000   01B5   1606 INIT_TEST_PC:    .WORD                       ; GETS LOADED WITH BASE ADDRESS OF TEST
                                    0000   01B7   1607 C_INIT_TEST_PC: .WORD                        ; GETS LOADED WITH TWO'S COMPLIMENT
                                           01B9   1608                                              ; OF BASE ADDRESS OF TEST
                                      00   01B9   1609 TEST_SIZE:       .BYTE                       ; CONTAINS SIZE OF CURRENT TEST IN
                                           01BA   1610                                              ; RECORDS
                                           01BA   1611 ;
                                           01BA   1612 ; NOTE: ALL MICRO WORDS IN THIS PROGRAM MUST BE CONTIGUOUS IN THE FOLLOWING
                                           01BA   1613 ; BLOCK SINCE THEY ARE SCRAMBLED WHEN THE MONITOR STARTSUP.
                                           01BA   1614 ;
                                000001BA   01BA   1615                  MICRO_CODE_HEAD = .
 F7 08 00 03 64 02 00 04 70 18 00   01BA   1616 NOP_MICRO_WORD: .BYTE    0,^X18,^X70,  04,  00,  02,^X64,  03,  00,^X08,^XF7
 D7 48 40 03 64 02 00 04 70 18 00   01C5   1617 R_RDM_MIC_WORD: .BYTE    0,^X18,^X70,  04,  00,  02,^X64,  03,^X40,^X48,^XD7
 E7 08 00 02 47 02 00 04 70 18 00   01D0   1618 RDM_R_MIC_WORD: .BYTE    0,^X18,^X70,  04,  00,  02,^X47,  02,  00,  08,^XE7
 67 08 00 59 24 03 00 04 70 18 00   01DB   1619 RDM_M_MIC_WORD: .BYTE    0,^X18,^X70,  04,  00,  03,^X24,^X59,  00,  08,^X67
 67 48 13 59 24 03 00 4C 70 18 00   01E6   1620 RDM_WD_MIC_WORD:.BYTE    0,^X18,^X70,^X4C,  00,  03,^X24,^X59,^X13,^X48,^X67
 57 48 00 03 64 03 00 68 70 18 00   01F1   1621 RDM_SR_MIC_WORD:.BYTE    0,^X18,^X70,^X68,  00,  03,^X64,  03,  00,^X48,^X57
 67 48 00 03 64 03 00 64 70 18 00   01FC   1622                 .BYTE    0,^X18,^X70,^X64,  00,  03,^X64,  03,  00,^X48,^X67
 67 48 00 03 64 03 00 08 70 18 00   0207   1623 RDM_PS_MIC_WORD:.BYTE    0,^X18,^X70,  08,  00,  03,^X64,  03,  00,^X48,^X67
 67 C8 00 03 64 03 00 18 70 18 00   0212   1624 RDM_ST_MIC_WORD:.BYTE    0,^X18,^X70,^X18,  00,  03,^X64,  03,  00,^XC8,^X67
 67 48 00 03 64 03 00 38 70 18 00   021D   1625 RDM_SF_MIC_WORD:.BYTE    0,^X18,^X70,^X38,  00,  03,^X64,  03,  00,^X48,^X67
                                           0228   1626
                                           0228   1627 ;
                                           0228   1628 ; THE FOLLOWING MICRO WORDS DISABLE THE CMI AND CLEAR THE PSL
                                           0228   1629 ;
                                           0228   1630 ;          BUS/PRINIT
                                           0228   1631 ;          LONLIT_[0E000000]
                                           0228   1632 ;          MEMSCAR_LONLIT
                                           0228   1633 ;          LONLIT_[01000000]
                                           0228   1634 ;          MEMSCR_LONLIT
```

```
                                  0228  ;          LONLIT_[0],WCTRL/PC_WB
                                  0228  ;          PSL_R[LONLIT]
                                  0228
  F7 08 00 03 64 02 00 04 10 18 00  0228  LOAD_MEMSCAR:    .BYTE   0,^X18,^X10,  04,  00,  02,^X64,  03,  00,  08,^XF7
  F7 B8 00 78 FF FF FF 84 70 18 00  0233                   .BYTE   0,^X18,^X70,^X84,^XFF,^XFF,^XFF,^X78,  00,^XB8,^XF7
  F7 48 00 00 47 02 D4 68 70 18 00  023E                   .BYTE   0,^X18,^X70,^X68,^XD4,  02,^X47,  00,  00,^X48,^XF7
  F7 B8 00 7F 7F FF FF 84 70 18 00  0249  LOAD_MEMSCR:     .BYTE   0,^X18,^X70,^X84,^XFF,^XFF,^X7F,^X7F,  00,^XB8,^XF7
  F7 C8 00 00 47 02 D4 60 70 18 00  0254                   .BYTE   0,^X18,^X70,^X60,^XD4,  02,^X47,  00,  00,^XC8,^XF7
  F7 78 00 7F FF FF FF C8 70 18 00  025F                   .BYTE   0,^X18,^X70,^XC8,^XFF,^XFF,^XFF,^X7F,  00,^X78,^XF7
  F7 C8 00 5B F4 02 D4 00 70 18 00  026A                   .BYTE   0,^X18,^X70,  00,^XD4,  02,^XF4,^X5B,  00,^XC8,^XF7
                                  0275
                                  0275  ;
                                  0275  ; THE FOLLOWING MICRO WORDS ARE USED TO WRITE THE CACHE
                                  0275  ;
                                  0275  ;          LONLIT_[06000000]
                                  0275  ;
                                  0275  ;          LONLIT_[0]
                                  0275  ;
                                  0275  ;
                                  0275  ;          VA_RDM
                                  0275  ;          WB_RDM,WRITE,CLKX
                                  0275  ;          WB_RDM,WCTRL/WDR_WB
                                  0275  ;          VA_VA+4
                                  0275
  F7 F8 00 7C FF FF FF 84 70 18 00  0275  CACHE_ENA_ADR:   .BYTE   0,^X18,^X70,^X84,^XFF,^XFF,^XFF,^X7C,  00,^XF8,^XF7
                                  0280  MME_DIS_ADR:
                                  0280  MME_DIS_DATA:
  F7 F8 00 7F FF FF FF 84 70 18 00  0280  CACHE_ENA_DATA:  .BYTE   0,^X18,^X70,^X84,^XFF,^XFF,^XFF,^X7F,  00,^XF8,^XF7
                                  028B
  D7 08 00 03 64 02 00 4A 70 18 00  028B  CACHE_WRITE:     .BYTE   0,^X18,^X70,^X4A,  00,  02,^X64,  03,  00,  08,^XD7
  D7 88 00 03 64 02 00 5D 80 98 00  0296                   .BYTE   0,^X98,^X80,^X5D,  00,  02,^X64,  03,  00,^X88,^XD7
  D7 88 00 03 64 02 00 5C 70 18 00  02A1                   .BYTE   0,^X18,^X70,^X5C,  00,  02,^X64,  03,  00,^X88,^XD7
  F7 88 00 03 64 02 00 44 70 18 00  02AC                   .BYTE   0,^X18,^X70,^X44,  00,  02,^X64,  03,  00,^X88,^XF7
                         000002B7  02B7                   MICRO_CODE_TAIL = .
                         00000017  02B7                   NO_MICRO_WORDS = <MICRO_CODE_TAIL - MICRO_CODE_HEAD>/11
                                  02B7
                                  02B7  ;
                                  02B7  ; TYPE_TEST_NUMBER PARAMETERS:
                                  02B7  ;
                               00  02B7  TYPE_TEST_FLAG: .BYTE   0                       ; INDICATES IF FIRST TEST NUMBER HAS
                                  02B8                                                   ; BEEN TYPED
                               00  02B8  TESTS_PER_LINE: .BYTE   0                       ; CONTAINS THE NUMBER OF TEST NUMBERS
                                  02B9                                                   ; TYPED ON THE CURRENT LINE
                             0000  02B9  TITLE_LENGTH:   .WORD   0                       ; CONTAINS LENGTH OF TEST TITLE
                 20 54 53 45 54 00'  02BB  TEST_MSG_1:     .ASCIC  /TEST /
                               05  02BB
                        20 3A 00'  02C1  TEST_MSG_2:     .ASCIC  /: /
                               02  02C1
                                  02C4
                                  02C4  ;
                                  02C4  ; READ_V_BUS PARAMETERS:
                                  02C4  ;
                                  02C4
                                  02C4  BIT_SLICE_TEMP:                                  ; TEMPORARY STORAGE FOR BIT SLICE ROUTINE
                         000002EC  02C4  V_BUS_BUFFER:   .BLKB   40                      ; BUFFER FOR THE V BUS BITS
                                  02EC
                                  02EC  1688 ;
                                  02EC  1689 ; GET_CMD_LINE PARAMETERS:
                                  02EC  1690 ;
                                  02EC  1691
                   3E 43 49 4D 00' 02EC  1692 PROMPT:          .ASCIC  /MIC>/             ; DIAGNOSTIC PROMPT
                               04  02EC
                               1E  02F1  1693 CMD_LINE_LENGTH:.BYTE    30                 ; NUMBER OF CHARACTERS IN COMMAND LINE
                               01  02F2  1694 ONE:            .BYTE    1                  ; NUMBER OF CHARACTERS FOR OTHER READS
4F 43 20 44 49 4C 41 56 4E 49 3F 00' 02F3  1695 INVALID_COMMAND:.ASCIC  /?INVALID COMMAND/ ; COMMAND LINE ERROR MESSAGE
                   44 4E 41 4D 4D  02FF
                               10  02F3
                                  0304  1696 CMD_DISPATCH:    ADDRESS <DIAGNOSE_ACTION> ; POINTER TO DIAGNOSE ACTION ROUTINE
                                  0306  1697                  ADDRESS <SHOW_ACTION>    ; POINTER TO SHOW ACTION ROUTINE
                                  0308  1698                  ADDRESS <CONTINUE_ACTION> ; POINTER TO CONTINUE ACTION ROUTINE
                                  030A  1699                  ADDRESS <RETURN_ACTION> ; POINTER TO RETURN ACTION ROUTINE
                                  030C  1700                  ADDRESS <SET_ACTION>    ; POINTER TO SET ACTION ROUTINE
                                  030E  1701                  ADDRESS <CLEAR_ACTION>  ; POINTER TO CLEAR ACTION ROUTINE
                                  0310  1702                  ADDRESS <LOOP_ACTION>   ; POINTER TO LOOP ACTION ROUTINE
                                  0312  1703                  ADDRESS <EXAMINE_ACTION>
                                  0314  1704                  ADDRESS <DEPOSIT_ACTION>
                                  0316  1705                  ADDRESS <BREAK_ACTION>
                                  0318  1706
                                  0318  1707 ;
                                  0318  1708 ; PARSE_CMD_LINE PARAMETERS:
                                  0318  1709 ;
                                  0318  1710
                                  0318  1711 PARSE_TBL_PTR:   ADDRESS <PARSE_TABLE>     ; CONTAINS ADDRESS OF PARSE TABLE
                         0000032E 031A  1712 PARSE_TABLE:     .BLKB   20                ; PARSE TABLE
                            49 44 032E  1713 COMMAND_LIST:    .ASCII  /DI/              ; DIAGNOSE
                            48 53 0330  1714                  .ASCII  /SH/              ; SHOW
                            4F 43 0332  1715                  .ASCII  /CO/              ; CONTINUE
                            45 52 0334  1716                  .ASCII  /RE/              ; RETURN
                            45 53 0336  1717                  .ASCII  /SE/              ; SET
                            4C 43 0338  1718                  .ASCII  /CL/              ; CLEAR
                            4F 4C 033A  1719                  .ASCII  /LO/              ; LOOP
                            58 45 033C  1720                  .ASCII  /EX/              ; EXAMINE
                            45 44 033E  1721                  .ASCII  /DE/              ; DEPOSIT
                            52 42 0340  1722                  .ASCII  /BR/
                               FF 0342  1723                  .BYTE   -1                ; LIST TERMINATOR
                                  0343  1724
                            45 54 0343  1725 KEYWORD_LIST:    .ASCII  /TE/              ; TEST
                            41 50 0345  1726                  .ASCII  /PA/              ; PASS
                            4C 46 0347  1727                  .ASCII  /FL/              ; FLAG
                            4F 53 0349  1728                  .ASCII  /SO/              ; SOMM
                                  034B  1729
                            41 48 034B  1730 HALT_FLAG_NAME: .ASCII  /HA/              ; HALT
                            4F 4C 034D  1731                  .ASCII  /LO/              ; LOOP
                            45 4E 034F  1732                  .ASCII  /NE/              ; NER
                            45 42 0351  1733                  .ASCII  /BE/              ; BELL
                            41 53 0353  1734                  .ASCII  /SA/              ; SINGATURE ANALYZER
                            42 49 0355  1735                  .ASCII  /IB/              ; INHIBIT BURST FLAG
                            41 51 0357  1736                  .ASCII  /QA/              ; QA FLAG
                            52 54 0359  1737                  .ASCII  /TR/              ; TRACE FLAG
                                  035B  1738
                            4E 49 035B  1739                  .ASCII  /IN/              ; INSTRUCTION
```

```
                              54 4C  035D  1740                  .ASCII  /LT/              ; LOOP ON TEST
                              4F 43  035F  1741                  .ASCII  /CO/              ; CONTINUE
                              54 53  0361  1742                  .ASCII  /ST/              ; STEP
                              59 43  0363  1743                  .ASCII  /CY/              ; CYCLE
                              49 54  0365  1744                  .ASCII  /TI/              ; TICK
                              46 43  0367  1745                  .ASCII  /CF/              ; CONTROL FILE
                              42 56  0369  1746                  .ASCII  /VB/              ; VISIBILITY BUS
                              56 51  036B  1747                  .ASCII  /QV/              ; QUICK VERIFY
                              4D 44  036D  1748                  .ASCII  /DM/              ; RDM TESTS
                              54 52  036F  1749                  .ASCII  /RT/              ; R TEMPS
                              54 4D  0371  1750                  .ASCII  /MT/              ; M TEMPS
                              43 50  0373  1751                  .ASCII  /PC/              ; PC REGISTER
                              41 56  0375  1752                  .ASCII  /VA/              ; VA
                              44 4D  0377  1753                  .ASCII  /MD/              ; MDR
                              42 50  0379  1754                  .ASCII  /PB/              ; PC BACKUP
                              41 4D  037B  1755                  .ASCII  /MA/              ; MEMORY ADDRESS REG
                              44 57  037D  1756                  .ASCII  /WD/              ; WDR
                              52 53  037F  1757                  .ASCII  /SR/              ; STATUS & CONTROL
                              53 50  0381  1758                  .ASCII  /PS/              ; PSL
                              46 53  0383  1759                  .ASCII  /SF/              ; FLAGS
                              42 53  0385  1760                  .ASCII  /SB/              ; ENABLE SINGLE BIT ERROR REPORT
                                 FF  0387  1761                  .BYTE   -1                ; LIST TERMINATOR
                                     0388  1762
                                     0388  1763 ;
                                     0388  1764 ; PARSE_HEX_NUMB PARAMETERS:
                                     0388  1765 ;
                                     0388  1766
                           00000000  0388  1767 PARSE_HEX_TEMP: .LONG   0                 ; WORKING SPACE TO MAKE 32 BIT NUMBER
                                     038C  1768
                                     038C  1769 ;
                                     038C  1770 ; DIAGNOSE ACTION ROUTINE PARAMETERS:
                                     038C  1771 ;
                                     038C  1772
                                 00  038C  1773 SPEC_TEST_NUMB: .BYTE                     ; CONTAINS STARTING TEST NO. FOR LOST
                                 00  038D  1774 TEST_SPAN_END:  .BYTE                     ; CONTAINS END TEST NO. FO 'TEST SPAN'
                                 01  038E  1775 USER_PASS_CNT:  .BYTE   1                 ; CONTAINS THE USER SPECIFIED PASS CNT
                                     038F  1776
                                     038F  1777 ;
                                     038F  1778 ; SET/CLEAR ACTION ROUTINE PARAMETERS:
                                     038F  1779 ;
                                     038F  1780
                               0000  038F  1781 SOMM_ADDRESS:   .WORD                     ; CONTAINS ADDRESS OF A SOMM
                                     0391  1782
                                     0391  1783 ;
                                     0391  1784 ; SHOW ACTION ROUTINE PARAMETERS:
                                     0391  1785 ;
                                     0391  1786
20 3A 54 45 53 20 53 47 41 4C 46 00' 0391  1787 SHOW_MESSAGE_1: .ASCIC  /FLAGS SET: /
                                 0B  0391
52 41 45 4C 43 20 53 47 41 4C 46 00' 039D  1788 SHOW_MESSAGE_2: .ASCIC  /FLAGS CLEAR: /
                              20 3A  03A9
                                 0D  039D
                                 00  03AB  1789 SHOW_TEMP:      .BYTE                     ; TEMP LOCATION FOR THIS ROUTINE
      20 3D 53 55 42 56 20 20 20 00' 03AC  1790 VBUS_MSG:       .ASCIC  /   VBUS= /
                                 09  03AC
```

```
                  03B6  1791
                  03B6  1792 ;
                  03B6  1793 ; OP_CODE_DISPATCH PARAMETERS:
                  03B6  1794 ;
                  03B6  1795
            0000  03B6  1796 TEST_PC:        .WORD                      ; CONTAINS THE TEST STREAM POINTER OF
                  03B8  1797                                            ; THE NEXT PSEUDO INSTRUCTION
            0000  03B8  1798 CURRENT_PC:     .WORD                      ; CONTAINS THE TEST STREAM POINTER OF
                  03BA  1799                                            ; THE CURRENT PSEUDO INSTRUCTION
            0000  03BA  1800 SPEC_TEST_PC:   .WORD                      ; CONTAINS THE USER DEFINED TEST PC
            0000  03BC  1801 EXAMINE_TEMP:   .WORD                      ; WORKING LOACTION FOR 'EXAMINE'
                  03BE  1802
        000003CE  03BE  1803 ARG_LIST:       .BLKB   MAX_ARGUMENTS      ; RECEIVES THE PSEUDO INSTR ARGUMENTS
                  03CE  1804 INSTR_TABLE:    ADDRESS <LOAD_DCS>         ; 0
                  03D0  1805                 ADDRESS <BURST_CLOCK>      ; 2
                  03D2  1806                 ADDRESS <COMPARE_REG>      ; 4
                  03D4  1807                 ADDRESS <COMPARE_REG_MSK>; 6
                  03D6  1808                 ADDRESS <BEGIN_LOOP>       ; 8
                  03D8  1809                 ADDRESS <END_LOOP>         ; 10
                  03DA  1810                 ADDRESS <END_TEST>         ; 12
                  03DC  1811                 ADDRESS <ERROR_LOOP>       ; 14
                  03DE  1812                 ADDRESS <PATTERN_GEN>      ; 16
                  03E0  1813                 ADDRESS <IF_ERROR>         ; 18
                  03E2  1814                 ADDRESS <INIT_THE_WORLD>; 20
                  03E4  1815                 ADDRESS <LOAD_REG>         ; 22
                  03E6  1816                 ADDRESS <MASK>             ; 24
                  03E8  1817                 ADDRESS <NEW_TEST>         ; 26
                  03EA  1818                 ADDRESS <SKIP>             ; 28
                  03EC  1819                 ADDRESS <SUB_TEST>         ; 30
                  03EE  1820                 ADDRESS <COMPARE_VBUS>     ; 32
                  03F0  1821                 ADDRESS <FETCH_MIC_INSTR>; 34
                  03F2  1822                 ADDRESS <END_PASS>         ; 36
                  03F4  1823                 ADDRESS <LOAD_FIELD>       ; 42
                  03F6  1824                 ADDRESS <LOAD_INDEX>       ; 44
                  03F8  1825                 ADDRESS <INCREMENT_INDEX>; 46
                  03FA  1826                 ADDRESS <SAVE_INDEX>       ; 48
                  03FC  1827                 ADDRESS <ENABLE_TRAP>      ; 50
                  03FE  1828                 ADDRESS <DUMP_LOG>         ; 52
                  0400  1829                 ADDRESS <ERROR_LOG>        ; 54
                  0402  1830                 ADDRESS <PRINT_S>          ; 56
                  0404  1831                 ADDRESS <PRINT_N>          ; 58
                  0406  1832                 ADDRESS <ENABLE_STALL>     ; 60
                  0408  1833 ;
                  0408  1834 ; BEGIN_LOOP AND END_LOOP PARAMETERS:
                  0408  1835 ;
                  0408  1836
                  0408  1837 INDEX_NAM_TBL:  ADDRESS <I_INDEX_TBL>
                  040A  1838                 ADDRESS <J_INDEX_TBL>
                  040C  1839                 ADDRESS <K_INDEX_TBL>
            0000  040E  1840 I_INDEX_TBL:    .WORD                      ; CONTAINS CURRENT VALUE OF INDEX
            0000  0410  1841                 .WORD                      ; CONTAINS END VALUE OF INDEX
              00  0412  1842                 .BYTE                      ; CONTAINS THE INCREMENT VALUE
            0000  0413  1843                 .WORD                      ; CONTAINS THE TEST PC OF START OF LOOP
        0000041C  0415  1844 J_INDEX_TBL:    .BLKB   7
        00000423  041C  1845 K_INDEX_TBL:    .BLKB   7
```

```
                                     0423  1846
                                     0423  1847 ;
                                     0423  1848 ; NEW_TEST ROUTINE PARAMETERS:
                                     0423  1849 ;
                                     0423  1850
                               0000  0423  1851 LOOP_ADDRESS:   .WORD                    ; CONTAINS THE TEST PC FOR LOOP ON TEST
                               0000  0425  1852 ERROR_LOOP_ADDR:.WORD                    ; CONTAINS THE TEST PC FOR A 'SCOPE' LOOP
                                     0427  1853
                                     0427  1854 ;
                                     0427  1855 ; PATTERN GENERATE ROUTINE PARAMETERS:
                                     0427  1856 ;
                                     0427  1857
                           AAAAAAAA  0427  1858 PATTERN_ADDRESS:.LONG   ^XAAAAAAAA       ; THESE ARE THE 6 TEST PATTERNS
                           55555555  042B  1859                 .LONG   ^X55555555       ; USED BY THE PATTERN GENERATE
                           33333333  042F  1860                 .LONG   ^X33333333       ; PSEUDO INSTRUCTION
                           0F0F0F0F  0433  1861                 .LONG   ^X0F0F0F0F       ; ...
                           00FF00FF  0437  1862                 .LONG   ^X00FF00FF
                           0000FFFF  043B  1863                 .LONG   ^X0000FFFF       ; ...
                                     043F  1864
                                     043F  1865 ;
                                     043F  1866 ; COMPARE PARAMETERS:
                                     043F  1867 ;
                                     043F  1868
                           00000443  043F  1869 GOOD_DATA:      .BLKB   4                ; CONTAINS THE EXPECTED DATA OF A COMPARE  -
cont> ROUTINE
                           00000447  0443  1870 BAD_DATA:       .BLKB   4                ; CONTAINS THE RECIEVED DATA OF A COMPARE  -
cont> ROUTINE
                                 00  0447  1871 DATA_TYPE:      .BYTE                    ; CONTAINS THE NUMBER OF BYTES
                                     0448  1872                                          ; IN GOOD AND BAD DATA
                                 00  0448  1873 MODE_TYPE:      .BYTE                    ; CONTAINS THE MODE FLAG OF THE
                                     0449  1874                                          ; LAST CMPREG OR CMPREGMSK
                               FFFF  0449  1875 QA_COUNT:       .WORD   -1               ; CONTAINS INDEX COUNT FOR QA FLAG
                               0000  044B  1876 QA_INDEX:       .WORD                    ; GETS LOADED WITH CURRENT INDEX VALUE
                                     044D  1877                                          ; IN 'COMPARE' ROUTINES
                               0000  044D  1878 VBUS_TBL_ADDR:  .WORD                    ; CONTAINS ADDRESS OF VBUS TABLE IN
                                     044F  1879                                          ; THE TEST OVERLAY
                                 00  044F  1880 VBUS_TBL_COUNT: .BYTE                    ; CONTAINS NUMBER OF BITS NOT COMPARED
                                     0450  1881                                          ; BY THE CMPVBUS PSEUDO INSTRUCTION
                                     0450  1882
                                     0450  1883 ;
                                     0450  1884 ; BURST CLOCK PARAMETERS:
                                     0450  1885 ;
                                     0450  1886
                                 00  0450  1887 STEP_ADDRESS:   .BYTE                    ; CONTAINS THE DCS ADDRESS THAT IS
                                     0451  1888                                          ; TYPED FOR SINGLE TICK AND CYCLE
                               0000  0451  1889 BURST_TEMP:     .WORD                    ; TEMPORARY STORAGE FOR THIS ROUTINE
                               0000  0453  1890 CLK_MSG_ADDR:   .WORD                    ; GETS LOADED WITH A MESSAGE ADDRESS
                                 00  0455  1891 M_CLK_ADDR:     .BYTE                    ; USED TO CHECK IF M CLOCK STILL RUNNING
52 44 44 41 20 53 43 44 20 20 20 00' 0456  1892 DCS_ADDR_MSG:   .ASCIC  /   DCS ADDR= /
                              20 3D  0462
                                 0D  0456
4B 41 45 52 42 20 4F 52 43 49 4D 00' 0464  1893 SOMM_MSG:       .ASCIC  /MICRO BREAK MATCH, CS ADDR= /
41 20 53 43 20 2C 48 43 54 41 4D 20  0470
                     20 3D 52 44 44  047C
                                 1C  0464
20 44 45 54 43 45 50 58 45 4E 55 00' 0481  1894 CLK_STOP_MSG1:  .ASCIC  /UNEXPECTED CLOCK STOP. DCS ADDR= /
20 2E 50 4F 54 53 20 4B 43 4F 4C 43  048D
```

```
      20 3D 52 44 44 41 20 53 43 44  0499
                                21   04B1
4D 55 4E 20 54 53 45 54 20 20 20 00' 04A3   1895 CLK_STOP_MSG2:  .ASCIC  /   TEST NUMBER= /
                  20 3D 52 45 42     04AF
                                10   04A3
                  20 3D 43 50 54 00' 04B4   1896 CLK_STOP_MSG3:  .ASCIC  /TPC= /
                                05   04B4
44 44 41 20 50 41 52 54 20 20 20 00' 04BA   1897 CLK_STOP_MSG4:  .ASCIC  /   TRAP ADDRESS= /
                  20 3D 53 53 45 52  04C6
                                11   04BA
4E 55 48 20 4B 43 4F 4C 43 20 4D 00' 04CC   1898 CLK_STOP_MSG5:  .ASCIC  /M CLOCK HUNG. DCS ADDR= /
3D 52 44 44 41 20 53 43 44 20 2E 47  04D8
                                20   04E4
                                18   04CC
                                     04E5   1899
                                     04E5   1900 ;
                                     04E5   1901 ; ERROR LOG AND LOG DUMP PARAMETERS AND MESSAGES
                                     04E5   1902 ;
                                     04E5   1903
                          000004E8   04E5   1904 LOG_ADDRESS:    .BLKB   3                   ; TEMPORARY STORAGE FOR ERROR ADDRESS
                                00   04E8   1905 LOG_SYNDROME:   .BYTE                       ; TEMPORARY STORAGE FOR ERROR SYNDROME
4C 46 52 45 56 4F 20 54 53 45 54 00' 04E9   1906 LOG_MSG_1:      .ASCIC  /TEST OVERFLOWS INTO ERROR LOG/
52 52 45 20 4F 54 4E 49 20 53 57 4F  04F5
                  47 4F 4C 20 52 4F  0501
                                1D   04E9
43 20 46 4F 20 52 45 42 4D 55 4E 00' 0507   1907 LOG_MSG_2:      .ASCIC  /NUMBER OF CORRECTABLE BITS IN THIS ARRAY = /
42 20 45 4C 42 41 54 43 45 52 52 4F  0513
20 53 49 48 54 20 4E 49 20 53 54 49  051F
            20 3D 20 59 41 52 52 41  052B
                                2B   0507
41 48 54 20 52 45 54 41 45 52 47 00' 0533   1908 LOG_MSG_3:      .ASCIC  /GREATER THAN 64 CORRECTABLE BITS IN THIS ARRAY, /-
54 43 45 52 52 4F 43 20 34 36 20 4E  053F
4E 49 20 53 54 49 42 20 45 4C 42 41  054B
2C 59 41 52 52 41 20 53 49 48 54 20  0557
45 54 52 4F 42 41 20 54 53 45 54 20  0563
                                44   056F
                                3C   0533
                                     0570   1909                                 /TEST ABORTED/
                                     0570   1910
                                     0570   1911 ;
                                     0570   1912 ; PARITY CALCULATION PARAMETERS:
                                     0570   1913 ;
                                     0570   1914
      DD 09 F1 B3 BF F6 99 00 95 00  0570   1915 P0_PARITY_MSK:  .BYTE      00,^X95,  00,^X99,^XF6,^XBF,^XB3,^XF1,  09,^XDD
      22 F6 0E 4C 00 09 66 F7 6A FF  057A   1916 P1_PARITY_MSK:  .BYTE    ^XFF,^X6A,^XF7,^X66,  09,  00,^X4C,^X0E,^XF6,^X22
                                     0584   1917
                                     0584   1918 ;
                                     0584   1919 ; END PASS PARAMETERS:
                                     0584   1920 ;
                                     0584   1921
53 53 41 50 20 46 4F 20 44 4E 45 00' 0584   1922 ENDPASS_MSG:    .ASCIC  /END OF PASS /  ; END OF PASS MESSAGE
                                20   0590
                                0C   0584
                                     0591   1923
                                     0591   1924 ;
```

```
                                          0591  1925 ; RING_BELL PARAMETERS:
                                          0591  1926 ;
                                          0591  1927
                                      00  0591  1928 BELL_COUNT:      .BYTE                         ; CONTAINS THE NUMBER OF TIMES THE
                                          0592  1929                                                ; RING_BELL ROUTINE HAS BEEN CALLED
                                   07 01  0592  1930 BELL:            .BYTE   1,7                   ; ASCII MESSAGE FOR A BELL
                                          0594  1931
                                          0594  1932 ;
                                          0594  1933 ; TYPE_ERROR PARAMETERS:
                                          0594  1934 ;
                                          0594  1935
                                00000596  0594  1936 ERROR_PC:        .BLKB   2                     ; CONTAINS THE TEST PC OF AN 'IFERROR'
                                          0596  1937                                                ; INSTRUCTION FOR FAILURE TYPE OUT
                                0000059A  0596  1938 TYPE_ERR_BUFFER:.BLKB    4                     ; BUFFER FOR ERROR DATA TYPE OUT
                                      00  059A  1939 TYPE_ERR_TEMP:   .BYTE                         ; TEMP STORAGE FOR ROUTINE
          20 3A 52 4F 52 52 45 3F 00' 059B  1940 ERROR_MSG:       .ASCIC  /?ERROR: /
                                      08  059B
             20 3A 54 53 45 54 20 00' 05A4  1941 TEST_MSG:        .ASCIC  / TEST: /
                                      07  05A4
    20 3A 54 53 45 54 42 55 53 20 00' 05AC  1942 SUB_TEST_MSG:    .ASCIC  / SUBTEST: /
                                      0A  05AC
          20 20 20 3A 41 54 41 44 00' 05B7  1943 DATA_MSG:        .ASCIC  /DATA:   /
                                      08  05B7
                      54 4F 4E 20 00' 05C0  1944 NOT_MSG:         .ASCIC  / NOT/
                                      04  05C0
          20 20 20 20 20 20 20 20 00' 05C5  1945 TAB:             .ASCIC  /        /
                                      08  05C5
                                   2D 00' 05CE  1946 HYPHEN:          .ASCIC  /-/
                                      01  05CE
       41 54 41 44 20 53 55 42 56 00' 05D0  1947 VBUS_MSG_2:      .ASCIC  /VBUS DATA/
                                      09  05D0
54 41 47 20 47 4E 49 4C 49 41 46 00' 05DA  1948 FAIL_ARRAY_MSG: .ASCIC  /FAILING GATE ARRAYS: /
       20 3A 53 59 41 52 52 41 20 45  05E6
                                      15  05DA
20 20 20 20 20 20 20 20 20 20 20 00' 05F0  1949 FAIL_ARR_INDEX: .ASCIC  /                     /
       20 20 20 20 20 20 20 20 20 20  05FC
                                      15  05F0
44 4F 4D 20 47 4E 49 4C 49 41 46 00' 0606  1950 MODULE_MSG:      .ASCIC  /FAILING MODULES: /
                      20 3A 53 45 4C 55  0612
                                      11  0606
20 20 20 20 20 20 20 20 20 20 20 00' 0618  1951 FAIL_MOD_INDEX: .ASCIC  /                 /
                      20 20 20 20 20 20  0624
                                      11  0618
                                   2C 00' 062A  1952 COMMA_MSG:       .ASCIC  /,/
                                      01  062A
                                          062C  1953
                                          062C  1954 ;
                                          062C  1955 ; GATE ARRAY NAME LIST
                                          062C  1956 ;
                                          062C  1957
                                43 44 00' 062C  1958 DC_MSG:          .ASCIC  /DC/
                                      02  062C
          29 4B 4C 41 28 35 31 36 00' 062F  1959 GATE_ARRAY_LST: .ASCIC  /615(ALK)/              ; ALK
                                      08  062F
          29 4B 52 53 28 34 31 36 00' 0638  1960                  .ASCIC  /614(SRK)/              ; SRK
```

```
                                  08  0638
         29 41 50 53 28 36 31 36  00' 0641   1961                  .ASCIC  /616(SPA)/              ; SPA
                                  08  0641
         29 41 4C 43 28 32 31 36  00' 064A   1962                  .ASCIC  /612(CLA)/              ; CLA
                                  08  064A
         29 43 43 43 28 30 31 36  00' 0653   1963                  .ASCIC  /610(CCC)/              ; CCC
                                  08  0653
         29 44 52 49 28 32 32 36  00' 065C   1964                  .ASCIC  /622(IRD)/              ; IRD
                                  08  065C
         29 4B 4F 54 28 30 32 36  00' 0665   1965                  .ASCIC  /620(TOK)/              ; TOK
                                  08  0665
         29 42 48 50 28 39 32 36  00' 066E   1966                  .ASCIC  /629(PHB)/              ; PHB
                                  08  066E
         29 51 53 4D 28 31 32 36  00' 0677   1967                  .ASCIC  /621(MSQ)/              ; MSQ
                                  08  0677
         29 43 41 53 28 37 31 36  00' 0680   1968                  .ASCIC  /617(SAC)/              ; SAC
                                  08  0680
                                      0689   1969
         29 4B 41 43 28 37 32 36  00' 0689   1970                  .ASCIC  /627(CAK)/              ; CAK
                                  08  0689
         29 4B 44 41 28 36 32 36  00' 0692   1971                  .ASCIC  /626(ADK)/              ; ADK
                                  08  0692
         29 4B 52 50 28 34 32 36  00' 069B   1972                  .ASCIC  /624(PRK)/              ; PRK
                                  08  069B
         29 52 54 55 28 38 32 36  00' 06A4   1973                  .ASCIC  /628(UTR)/              ; UTR
                                  08  06A4
         29 4C 4D 43 28 31 35 36  00' 06AD   1974                  .ASCIC  /651(CML)/              ; CML
                                  08  06AD
         29 56 43 41 28 35 32 36  00' 06B6   1975                  .ASCIC  /625(ACV)/              ; ACV
                                  08  06B6
                                      06BF   1976
         29 54 4E 49 28 30 33 36  00' 06BF   1977                  .ASCIC  /630(INT)/              ; INT
                                  08  06BF
         29 4E 4F 43 28 31 31 36  00' 06C8   1978                  .ASCIC  /611(CON)/              ; CON
                                  08  06C8
                                      06D1   1979
         29 50 41 4D 28 32 33 36  00' 06D1   1980                  .ASCIC  /632(MAP)/              ; MAP
                                  08  06D1
         29 44 41 4D 28 30 35 36  00' 06DA   1981                  .ASCIC  /650(MAD)/              ; MAD
                                  08  06DA
                                      06E3   1982
         29 41 51 46 28 32 34 36  00' 06E3   1983                  .ASCIC  /642(FQA)/              ; FQA
                                  08  06E3
         29 43 43 46 28 33 34 36  00' 06EC   1984                  .ASCIC  /643(FCC)/              ; FCC
                                  08  06EC
                                      06F5   1985      ;
                                      06F5   1986      ; THE FOLLOWING GATE ARRAYS ARE BIT SLICED AND THEREFORE HAVE UNIQUE
                                      06F5   1987      ; CALLOUT MESSAGES
                                      06F5   1988      ;
                                      06F5   1989
                                      06F5   1990  BIT_SLICE_TBL:: ADDRESS <ALP_NAME>
                                      06F7   1991                  ADDRESS <SRM_NAME>
                                      06F9   1992                  ADDRESS <MDR_NAME>
                                      06FB   1993                  ADDRESS <ADD_NAME>
                                      06FD   1994                  ADDRESS <MEC_NAME>
```

```
                                     06FF  1995                   ADDRESS <MDL_NAME>
                                     0701  1996                   ADDRESS <UDP_NAME>
                                     0703  1997                   ADDRESS <FIO_NAME>
                                     0705  1998                   ADDRESS <FEX_NAME>
                                     0707  1999                   ADDRESS <FIC_NAME>
                                     0709  2000                   ADDRESS <FAC_NAME>
                                     070B  2001                   ADDRESS <FCS_NAME>
                                     070D  2002                   ADDRESS <FMR_NAME>
                                     070F  2003                   ADDRESS <FFL_NAME>
                                     0711  2004                   ADDRESS <FFH_NAME>
                                     0713  2005
                                 08  0713  2006 ALP_NAME::        .BYTE   8                       ; NUMBER OF SLICES
                           0000000F  0714  2007                   .LONG   ^X0000000F              ; BIT MASK'S PER SLICE
                           000000F0  0718  2008                   .LONG   ^X000000F0
                           00000F00  071C  2009                   .LONG   ^X00000F00
                           0000F000  0720  2010                   .LONG   ^X0000F000
                           000F0000  0724  2011                   .LONG   ^X000F0000
                           00F00000  0728  2012                   .LONG   ^X00F00000
                           0F000000  072C  2013                   .LONG   ^X0F000000
                           F0000000  0730  2014                   .LONG   ^XF0000000
                                     0734  2015
3C 20 31 20 50 4C 41 28 38 30 36 00' 0734  2016                   .ASCIC  /608(ALP 1 <03:00>)/            ; ALP
               29 3E 30 30 3A 33 30  0740
                                 12  0734
3C 20 32 20 50 4C 41 28 38 30 36 00' 0747  2017                   .ASCIC  /608(ALP 2 <07:04>)/            ; ALP
               29 3E 34 30 3A 37 30  0753
                                 12  0747
3C 20 33 20 50 4C 41 28 38 30 36 00' 075A  2018                   .ASCIC  /608(ALP 3 <11:08>)/            ; ALP
               29 3E 38 30 3A 31 31  0766
                                 12  075A
3C 20 34 20 50 4C 41 28 38 30 36 00' 076D  2019                   .ASCIC  /608(ALP 4 <15:12>)/            ; ALP
               29 3E 32 31 3A 35 31  0779
                                 12  076D
3C 20 35 20 50 4C 41 28 38 30 36 00' 0780  2020                   .ASCIC  /608(ALP 5 <19:16>)/            ; ALP
               29 3E 36 31 3A 39 31  078C
                                 12  0780
3C 20 36 20 50 4C 41 28 38 30 36 00' 0793  2021                   .ASCIC  /608(ALP 6 <23:20>)/            ; ALP
               29 3E 30 32 3A 33 32  079F
                                 12  0793
3C 20 37 20 50 4C 41 28 38 30 36 00' 07A6  2022                   .ASCIC  /608(ALP 7 <27:24>)/            ; ALP
               29 3E 34 32 3A 37 32  07B2
                                 12  07A6
3C 20 38 20 50 4C 41 28 38 30 36 00' 07B9  2023                   .ASCIC  /608(ALP 8 <31:28>)/            ; ALP
               29 3E 38 32 3A 31 33  07C5
                                 12  07B9
                                     07CC  2024
                                 04  07CC  2025 SRM_NAME::        .BYTE   4
                           11111111  07CD  2026                   .LONG   ^X11111111
                           22222222  07D1  2027                   .LONG   ^X22222222
                           44444444  07D5  2028                   .LONG   ^X44444444
                           88888888  07D9  2029                   .LONG   ^X88888888
                                     07DD  2030
3C 20 31 20 4D 52 53 28 33 31 36 00' 07DD  2031                   .ASCIC  /613(SRM 1 <00,04,08,12,16,20,24,28,32>)/; SRM
2C 32 31 2C 38 30 2C 34 30 2C 30 30  07E9
2C 38 32 2C 34 32 2C 30 32 2C 36 31  07F5
```

```
                        29 3E 32 33  0801
                                 27  07DD
3C 20 32 20 4D 52 53 28 33 31 36 00' 0805  2032                 .ASCIC  /613(SRM 2 <01,05,09,13,17,21,25,29,33>)/; SRM
2C 33 31 2C 39 30 2C 35 30 2C 31 30  0811
2C 39 32 2C 35 32 2C 31 32 2C 37 31  081D
                        29 3E 33 33  0829
                                 27  0805
3C 20 33 20 4D 52 53 28 33 31 36 00' 082D  2033                 .ASCIC  /613(SRM 3 <02,06,10,14,18,22,26,30,34>)/; SRM
2C 34 31 2C 30 31 2C 36 30 2C 32 30  0839
2C 30 33 2C 36 32 2C 32 32 2C 38 31  0845
                        29 3E 34 33  0851
                                 27  082D
3C 20 34 20 4D 52 53 28 33 31 36 00' 0855  2034                 .ASCIC  /613(SRM 4 <03,07,11,15,19,23,27,31>)/   ; SRM
2C 35 31 2C 31 31 2C 37 30 2C 33 30  0861
3E 31 33 2C 37 32 2C 33 32 2C 39 31  086D
                                 29  0879
                                 24  0855
                                     087A  2035
                                 08  087A  2036 MDR_NAME::       .BYTE   8
                           01010101  087B  2037                 .LONG   ^X01010101
                           02020202  087F  2038                 .LONG   ^X02020202
                           04040404  0883  2039                 .LONG   ^X04040404
                           08080808  0887  2040                 .LONG   ^X08080808
                           10101010  088B  2041                 .LONG   ^X10101010
                           20202020  088F  2042                 .LONG   ^X20202020
                           40404040  0893  2043                 .LONG   ^X40404040
                           80808080  0897  2044                 .LONG   ^X80808080
                                     089B  2045
3C 20 31 20 52 44 4D 28 37 30 36 00' 089B  2046                 .ASCIC  /607(MDR 1 <00,08,16,24>)/       ; MDR
3E 34 32 2C 36 31 2C 38 30 2C 30 30  08A7
                                 29  08B3
                                 18  089B
3C 20 32 20 52 44 4D 28 37 30 36 00' 08B4  2047                 .ASCIC  /607(MDR 2 <01,09,17,25>)/       ; MDR
3E 35 32 2C 37 31 2C 39 30 2C 31 30  08C0
                                 29  08CC
                                 18  08B4
3C 20 33 20 52 44 4D 28 37 30 36 00' 08CD  2048                 .ASCIC  /607(MDR 3 <02,10,18,26>)/       ; MDR
3E 36 32 2C 38 31 2C 30 31 2C 32 30  08D9
                                 29  08E5
                                 18  08CD
3C 20 34 20 52 44 4D 28 37 30 36 00' 08E6  2049                 .ASCIC  /607(MDR 4 <03,11,19,27>)/       ; MDR
3E 37 32 2C 39 31 2C 31 31 2C 33 30  08F2
                                 29  08FE
                                 18  08E6
3C 20 35 20 52 44 4D 28 37 30 36 00' 08FF  2050                 .ASCIC  /607(MDR 5 <04,12,20,28>)/       ; MDR
3E 38 32 2C 30 32 2C 32 31 2C 34 30  090B
                                 29  0917
                                 18  08FF
3C 20 36 20 52 44 4D 28 37 30 36 00' 0918  2051                 .ASCIC  /607(MDR 6 <05,13,21,29>)/       ; MDR
3E 39 32 2C 31 32 2C 33 31 2C 35 30  0924
                                 29  0930
                                 18  0918
3C 20 37 20 52 44 4D 28 37 30 36 00' 0931  2052                 .ASCIC  /607(MDR 7 <06,14,22,30>)/       ; MDR
3E 30 33 2C 32 32 2C 34 31 2C 36 30  093D
                                 29  0949
```

```
                                   18  0931
3C 20 38 20 52 44 4D 28 37 30 36 00'  094A   2053                  .ASCIC  /607(MDR 8 <07,15,23,31>)/      ; MDR
3E 31 33 2C 33 32 2C 35 31 2C 37 30   0956
                                   29  0962
                                   18  094A
                                       0963   2054
                                   04  0963   2055 ADD_NAME::      .BYTE   4
                             000000FF  0964   2056                 .LONG   ^X000000FF
                             0000FF00  0968   2057                 .LONG   ^X0000FF00
                             00FF0000  096C   2058                 .LONG   ^X00FF0000
                             FF000000  0970   2059                 .LONG   ^XFF000000
                                       0974   2060
3C 20 31 20 44 44 41 28 39 30 36 00'  0974   2061                  .ASCIC  /609(ADD 1 <07:00>)/            ; ADD
            29 3E 30 30 3A 37 30      0980
                                   12  0974
3C 20 32 20 44 44 41 28 39 30 36 00'  0987   2062                  .ASCIC  /609(ADD 2 <15:08>)/            ; ADD
            29 3E 38 30 3A 35 31      0993
                                   12  0987
3C 20 33 20 44 44 41 28 39 30 36 00'  099A   2063                  .ASCIC  /609(ADD 3 <23:16>)/            ; ADD
            29 3E 36 31 3A 33 32      09A6
                                   12  099A
3C 20 34 20 44 44 41 28 39 30 36 00'  09AD   2064                  .ASCIC  /609(ADD 4 <31:24>)/            ; ADD
            29 3E 34 32 3A 31 33      09B9
                                   12  09AD
                                       09C0   2065
                                   02  09C0   2066 MEC_NAME::      .BYTE   2
                             0000FFFF  09C1   2067                 .LONG   ^X0000FFFF
                             FFFF0000  09C5   2068                 .LONG   ^XFFFF0000
                                       09C9   2069
3C 20 31 20 43 45 4D 28 31 33 36 00'  09C9   2070                  .ASCIC  /631(MEC 1 <15:00>)/            ; MEC
            29 3E 30 30 3A 35 31      09D5
                                   12  09C9
3C 20 32 20 43 45 4D 28 31 33 36 00'  09DC   2071                  .ASCIC  /631(MEC 2 <31:16>)/            ; MEC
            29 3E 36 31 3A 31 33      09E8
                                   12  09DC
                                       09EF   2072
                                   04  09EF   2073 MDL_NAME::      .BYTE   4
                             000000FF  09F0   2074                 .LONG   ^X000000FF
                             0000FF00  09F4   2075                 .LONG   ^X0000FF00
                             00FF0000  09F8   2076                 .LONG   ^X00FF0000
                             FF000000  09FC   2077                 .LONG   ^XFF000000
                                       0A00   2078
3C 20 31 20 4C 44 4D 28 33 33 36 00'  0A00   2079                  .ASCIC  /633(MDL 1 <07:00>)/            ; MDL
            29 3E 30 30 3A 37 30      0A0C
                                   12  0A00
3C 20 32 20 4C 44 4D 28 33 33 36 00'  0A13   2080                  .ASCIC  /633(MDL 2 <15:08>)/            ; MDL
            29 3E 38 30 3A 35 31      0A1F
                                   12  0A13
3C 20 33 20 4C 44 4D 28 33 33 36 00'  0A26   2081                  .ASCIC  /633(MDL 3 <23:16>)/            ; MDL
            29 3E 36 31 3A 33 32      0A32
                                   12  0A26
3C 20 34 20 4C 44 4D 28 33 33 36 00'  0A39   2082                  .ASCIC  /633(MDL 4 <31:24>)/            ; MDL
            29 3E 34 32 3A 31 33      0A45
                                   12  0A39
                                       0A4C   2083
                                   04  0A4C  2084 UDP_NAME::        .BYTE   4
                             03030303  0A4D  2085                   .LONG   ^X03030303
                             0C0C0C0C  0A51  2086                   .LONG   ^X0C0C0C0C
                             30303030  0A55  2087                   .LONG   ^X30303030
                             C0C0C0C0  0A59  2088                   .LONG   ^XC0C0C0C0
                                       0A5D  2089
3C 20 31 20 50 44 55 28 38 31 36 00'   0A5D  2090                   .ASCIC  /618(UDP 1 <00,01,08,09,16,17,24,25>)/   ; UDP
2C 39 30 2C 38 30 2C 31 30 2C 30 30    0A69
3E 35 32 2C 34 32 2C 37 31 2C 36 31    0A75
                                   29  0A81
                                   24  0A5D
3C 20 32 20 50 44 55 28 38 31 36 00'   0A82  2091                   .ASCIC  /618(UDP 2 <02,03,10,11,18,19,26,27>)/   ; UDP
2C 31 31 2C 30 31 2C 33 30 2C 32 30    0A8E
3E 37 32 2C 36 32 2C 39 31 2C 38 31    0A9A
                                   29  0AA6
                                   24  0A82
3C 20 33 20 50 44 55 28 38 31 36 00'   0AA7  2092                   .ASCIC  /618(UDP 3 <04,05,12,13,20,21,28,29>)/   ; UDP
2C 33 31 2C 32 31 2C 35 30 2C 34 30    0AB3
3E 39 32 2C 38 32 2C 31 32 2C 30 32    0ABF
                                   29  0ACB
                                   24  0AA7
3C 20 34 20 50 44 55 28 38 31 36 00'   0ACC  2093                   .ASCIC  /618(UDP 4 <06,07,14,15,22,23,30,31>)/   ; UDP
2C 35 31 2C 34 31 2C 37 30 2C 36 30    0AD8
3E 31 33 2C 30 33 2C 33 32 2C 32 32    0AE4
                                   29  0AF0
                                   24  0ACC
                                       0AF1  2094
                                   08  0AF1  2095 FIO_NAME::        .BYTE   8
                             0000000F  0AF2  2096                   .LONG   ^X0000000F
                             000000F0  0AF6  2097                   .LONG   ^X000000F0
                             00000F00  0AFA  2098                   .LONG   ^X00000F00
                             0000F000  0AFE  2099                   .LONG   ^X0000F000
                             000F0000  0B02  2100                   .LONG   ^X000F0000
                             00F00000  0B06  2101                   .LONG   ^X00F00000
                             0F000000  0B0A  2102                   .LONG   ^X0F000000
                             F0000000  0B0E  2103                   .LONG   ^XF0000000
                                       0B12  2104
   29 30 20 4F 49 46 28 36 33 36 00'   0B12  2105                   .ASCIC  /636(FIO 0)/     ; FIO
                                   0A  0B12
   29 31 20 4F 49 46 28 36 33 36 00'   0B1D  2106                   .ASCIC  /636(FIO 1)/     ; FIO
                                   0A  0B1D
   29 32 20 4F 49 46 28 36 33 36 00'   0B28  2107                   .ASCIC  /636(FIO 2)/     ; FIO
                                   0A  0B28
   29 33 20 4F 49 46 28 36 33 36 00'   0B33  2108                   .ASCIC  /636(FIO 3)/     ; FIO
                                   0A  0B33
   29 34 20 4F 49 46 28 36 33 36 00'   0B3E  2109                   .ASCIC  /636(FIO 4)/     ; FIO
                                   0A  0B3E
   29 35 20 4F 49 46 28 36 33 36 00'   0B49  2110                   .ASCIC  /636(FIO 5)/     ; FIO
                                   0A  0B49
   29 36 20 4F 49 46 28 36 33 36 00'   0B54  2111                   .ASCIC  /636(FIO 6)/     ; FIO
                                   0A  0B54
   29 37 20 4F 49 46 28 36 33 36 00'   0B5F  2112                   .ASCIC  /636(FIO 7)/     ; FIO
                                   0A  0B5F
                                       0B6A  2113
                                   02  0B6A  2114 FEX_NAME::        .BYTE   2
```

```
                           00007000  0B6B  2115                          .LONG   ^X00007000
                           00000F80  0B6F  2116                          .LONG   ^X00000F80
                                     0B73  2117
   29 31 20 58 45 46 28 31 34 36 00' 0B73  2118                          .ASCIC  /641(FEX 1)/    ; FEX
                                 0A  0B73
   29 30 20 58 45 46 28 31 34 36 00' 0B7E  2119                          .ASCIC  /641(FEX 0)/    ; FEX
                                 0A  0B7E
                                     0B89  2120
                                 01  0B89  2121 FIC_NAME::               .BYTE   1
                           FFFFFFFF  0B8A  2122                          .LONG   ^XFFFFFFFF
                                     0B8E  2123
20 52 54 52 43 4E 49 28 32 31 36 00' 0B8E  2124                          .ASCIC  /612(INCRTR CLA)/        ; FIC
                        29 41 4C 43  0B9A
                                 0F  0B8E
                                     0B9E  2125
                                 01  0B9E  2126 FAC_NAME::               .BYTE   1
                           FFFFFFFF  0B9F  2127                          .LONG   ^XFFFFFFFF
                                     0BA3  2128
41 4C 43 20 55 4C 41 28 32 31 36 00' 0BA3  2129                          .ASCIC  /612(ALU CLA)/  ; FAC
                                 29  0BAF
                                 0C  0BA3
                                     0BB0  2130
                                 04  0BB0  2131 FCS_NAME::               .BYTE   4
                           11111111  0BB1  2132                          .LONG   ^X11111111
                           22222222  0BB5  2133                          .LONG   ^X22222222
                           44444444  0BB9  2134                          .LONG   ^X44444444
                           88888888  0BBD  2135                          .LONG   ^X88888888
                                     0BC1  2136
   29 30 20 53 43 46 28 37 33 36 00' 0BC1  2137                          .ASCIC  /637(FCS 0)/    ; FCS
                                 0A  0BC1
   29 31 20 53 43 46 28 37 33 36 00' 0BCC  2138                          .ASCIC  /637(FCS 1)/    ; FCS
                                 0A  0BCC
   29 32 20 53 43 46 28 37 33 36 00' 0BD7  2139                          .ASCIC  /637(FCS 2)/    ; FCS
                                 0A  0BD7
   29 33 20 53 43 46 28 37 33 36 00' 0BE2  2140                          .ASCIC  /637(FCS 3)/    ; FCS
                                 0A  0BE2
                                     0BED  2141
                                 02  0BED  2142 FMR_NAME::               .BYTE   2
                           0F0F0F0F  0BEE  2143                          .LONG   ^X0F0F0F0F
                           F0F0F0F0  0BF2  2144                          .LONG   ^XF0F0F0F0
                                     0BF6  2145
   29 30 20 52 4D 46 28 39 33 36 00' 0BF6  2146                          .ASCIC  /639(FMR 0)/    ; FMR
                                 0A  0BF6
   29 31 20 52 4D 46 28 39 33 36 00' 0C01  2147                          .ASCIC  /639(FMR 1)/    ; FMR
                                 0A  0C01
                                     0C0C  2148
                                 04  0C0C  2149 FFL_NAME::               .BYTE   4               ; FFA LOW HALF
                           000000FF  0C0D  2150                          .LONG   ^X000000FF
                           0000FF00  0C11  2151                          .LONG   ^X0000FF00
                           00FF0000  0C15  2152                          .LONG   ^X00FF0000
                           FF000000  0C19  2153                          .LONG   ^XFF000000
                                     0C1D  2154
   29 30 20 41 46 46 28 38 33 36 00' 0C1D  2155                          .ASCIC  /638(FFA 0)/
                                 0A  0C1D
   29 31 20 41 46 46 28 38 33 36 00' 0C28  2156                          .ASCIC  /638(FFA 1)/
```

```
                             0A  0C28
  29 32 20 41 46 46 28 38 33 36 00' 0C33   2157                    .ASCIC  /638(FFA 2)/
                             0A  0C33
  29 33 20 41 46 46 28 38 33 36 00' 0C3E   2158                    .ASCIC  /638(FFA 3)/
                             0A  0C3E
                                 0C49   2159
                             04  0C49   2160 FFH_NAME::          .BYTE   4                       ; FFA HIGH HALF
                       000000FF  0C4A   2161                     .LONG   ^X000000FF
                       0000FF00  0C4E   2162                     .LONG   ^X0000FF00
                       00FF0000  0C52   2163                     .LONG   ^X00FF0000
                       FF000000  0C56   2164                     .LONG   ^XFF000000
                                 0C5A   2165
  29 34 20 41 46 46 28 38 33 36 00' 0C5A   2166                    .ASCIC  /638(FFA 4)/
                             0A  0C5A
  29 35 20 41 46 46 28 38 33 36 00' 0C65   2167                    .ASCIC  /638(FFA 5)/
                             0A  0C65
  29 36 20 41 46 46 28 38 33 36 00' 0C70   2168                    .ASCIC  /638(FFA 6)/
                             0A  0C70
  29 37 20 41 46 46 28 38 33 36 00' 0C7B   2169                    .ASCIC  /638(FFA 7)/
                             0A  0C7B
                                 0C86   2170
                                 0C86   2171 ;
                                 0C86   2172 ; SINGLE PSEUDO INSTRUCTION PARAMETERS:
                                 0C86   2173 ;
                                 0C86   2174
     20 3D 43 50 54 20 20 20 20 00' 0C86   2175 TPC_MSG:           .ASCIC  /    TPC= /
                             09  0C86
                                 0C90   2176
                                 0C90   2177 ;
                                 0C90   2178 ; INSERT FIELD PARAMETERS:
                                 0C90   2179 ;
                                 0C90   2180
                       00000C9A  0C90   2181 INS_FLD_BUFF:      .BLKB   10                      ; TEMPORARY BUFFER
                             00  0C9A   2182 CURRENT_BIT:       .BYTE                           ; CONTAINS CURRENT BIT NUMBER
                             00  0C9B   2183 BIT_COUNT:         .BYTE                           ; CONTAINS NUMBER OF BITS IN FIELD
                             00  0C9C   2184 SCRAMBLE_FLAG:     .BYTE                           ; INHIBIT SCRAMBLE FLAG
                                 0C9D   2185
                                 0C9D   2186 ;
                                 0C9D   2187 ; READ RTEMP PARAMETERS:
                                 0C9D   2188 ;
                             00  0C9D   2189 READ_R_TEMP:       .BYTE                           ; TEMP STORAGE
                                 0C9E   2190
                                 0C9E   2191 ;
                                 0C9E   2192 ; THE FOLLOWING 80 BYTES CONTAIN THE BIT NUMBERS OF THE SCRAMBLED DCS.
                                 0C9E   2193 ;
                                 0C9E   2194 ;                             BITS IN DCS MICRO-WORD          CORRESPONDING BITS
                                 0C9E   2195 ;                                                             IN COMET MICRO-WORD
                                 0C9E   2196 ;
         01 00 45 1D 46 1E 1A 44  0C9E   2197 DCS_BIT_MAP:       .BYTE   68,26,30,70,29,69,00,01         ; 0-7
         23 19 07 06 05 04 03 02  0CA6   2198                    .BYTE   02,03,04,05,06,07,25,35         ; 8-15
         15 11 16 17 09 0B 3B 14  0CAE   2199                    .BYTE   20,59,11,09,23,22,17,21         ; 16-23
         0F 39 12 0D 0E 3A 10 49  0CB6   2200                    .BYTE   73,16,58,14,13,18,57,15         ; 24-31
         29 21 4C 31 4B 2D 36 38  0CBE   2201                    .BYTE   56,54,45,75,49,76,33,41         ; 32-39
         2B 2F 26 30 4E 4A 48 22  0CC6   2202                    .BYTE   34,72,74,78,48,38,47,43         ; 40-47
         27 3D 3E 28 37 40 35 3F  0CCE   2203                    .BYTE   63,53,64,55,40,62,61,39         ; 48-55
```

```
          08 1B 1C 1F 0C 0A 3C 18  0CD6   2204                  .BYTE   24,60,10,12,31,28,27,08           ; 56-63
          41 24 25 4F 43 34 2C 2A  0CDE   2205                  .BYTE   42,44,52,67,79,37,36,65           ; 64-71
          13 2E 20 4D 32 42 33 47  0CE6   2206                  .BYTE   71,51,66,50,77,32,46,19           ; 72-79
                                   0CEE   2207
                                   0CEE   2208 ;
                                   0CEE   2209 ; MISCELLANEOUS ASCII MESSAGES
                                   0CEE   2210 ;
                                   0CEE   2211
                         0A 0D 02  0CEE   2212 CRLF:            .BYTE   2,13,10                 ; ASCII FOR CARRIAGE RET, LINE FEED
                                   0CF1   2213
```

```
0CF1        .SBTTL  "PROGRAM INITIALIZATION ROUTINE
0CF1  ;++
0CF1  ;
0CF1  ; FUNCTIONAL DESCRIPTION:
0CF1  ;
0CF1  ;         THIS ROUTINE PERFORMS THE FOLLOWING FUNCTIONS:
0CF1  ;
0CF1  ;         0) STOP THE CLOCK
0CF1  ;         1) INITIALIZE THE TEST PC AND THE STACK POINTER
0CF1  ;         2) OPEN THE APPROPRIATE TEST STREAM FILE
0CF1  ;         3) TYPE THE PROGRAM NAME
0CF1  ;         4) IF THE 'LOSS' OR 'LOST' FLAG IS SET, CALL THE DIRECTORY ROUTINE
0CF1  ;         5) READ A TEST OVERLAY INTO MEMORY
0CF1  ;         6) CALL THE INTERPRETER TO START TESTING
0CF1  ;         7) CHECK IS ALL SPECIFIED PASSES HAVE BEEN EXECUTED. I THEY HAVE,
0CF1  ;            RETURN TO THE RDM SYSTEM VIA 'ABORT', OTHERWISE RESTART EXECUTION.
0CF1  ;
0CF1  ; CALLING SEQUENCE:
0CF1  ;
0CF1  ;         THIS ROUTINE IS ENTERED FROM THE RDM SOFTWARE OR THE COMMAND PARSER
0CF1  ;         IF A 'DIAGNOSE' COMMAND IS EXECUTED.
0CF1  ;
0CF1  ; INPUT PARAMETERS:
0CF1  ;
0CF1  ;         NONE
0CF1  ;
0CF1  ; IMPLICIT INPUTS:
0CF1  ;
0CF1  ;         NONE
0CF1  ;
0CF1  ; OUTPUT PARAMETERS:
0CF1  ;
0CF1  ;         NONE
0CF1  ;
0CF1  ; IMPLICIT OUTPUTS:
0CF1  ;
0CF1  ;         FILE_NAM_PTR - CONTAINS THE ADDRESS OF THE TEST STREAM FILE NAME
0CF1  ;                        THAT IS ON THE CURRENTLY LOADED CARTRIDGE
0CF1  ;
0CF1  ; COMPLETION CODES:
0CF1  ;
0CF1  ;         NONE
0CF1  ;
0CF1  ;--
0CF1
0CF1  PROGRAM_INIT:
0CF1  ;
0CF1  ; SETUP INPUT REQUEST
0CF1  ;
0CF1            $TERM_INIT                              ; CLEAR OUT THE TERMINAL DRIVER
0CF6            $TERM_READ      TERM_INP_BUFF,-
0CF6                            ONE,TERM_INTERRUPT
0D01            LHLD            SAVED_SP                ; GET INITIAL STACK POINTER
0D04            SPHL                                    ; INIT THE STACK POINTER
0D05            TYPE            CRLF                    ; TYPE A CARRIAGE RETURN LINE FEED
```

```
0D0E          TYPE    PROGRAM_TITLE    ; TYPE THE TITLE AND VERSION
0D17          XRA     A                ; CLEAR THE 'A' REG
0D18          STA     PASS_COUNT       ; INIT THE PASS COUNT
0D1B   ;
0D1B   ; CHECK IF GOT HERE WITH A "DI DM" COMMAND
0D1B   ;
0D1B          LDA     SOFT_CTRL_REGB   ; GET THE RDM FLAG
0D1E          MOV     B,A              ; SAVE LAST TIME RDM FLAG
0D1F          ANI     RDM_FLAG         ; ...
0D21          JZ      21$              ; BRANCH IF NOT SET
0D24          MOV     A,B              ; SET RDM_LAST FLAG
0D25          ORI     RDM_LAST         ; ...
0D27          STA     SOFT_CTRL_REGB   ; ...
0D2A          MVI     A,RDM_INDEX      ; SET FILE NAME INDEX TO RDM FILE NAME
0D2C          JMP     22$              ; GO CHECK IF ON TAPE
0D2F   ;
0D2F   ; NOT "DI DM". CHECK IF LAST EXECUTION WAS A "DI DM"
0D2F   ;
0D2F   21$:   MOV     A,B              ; GET RDM_LAST FLAG
0D30          ANI     RDM_LAST         ; WAS LAST EXECUTION RDM TESTS?
0D32          JZ      23$              ; BRANCH IF NO
0D35     ;
0D35     ; LAST EXECUTION WAS "DI DM" SO SEARCH DIRECTORY FOR DEFAULT FILE
0D35     ;
0D35          MOV     A,B              ; CLEAR RDM_LAST FLAG
0D36          ANI     <^CRDM_LAST>     ; ...
0D38          STA     SOFT_CTRL_REGB   ; ...
0D3B          JMP     20$              ; FIND OUT WHICH TAPE THIS IS
0D3E   ;
0D3E   ; LAST EXECUTION NOT "DI DM". CHECK IF PREVIOUS EXECUTION
0D3E   ;
0D3E   23$:   LHLD    STARTING_RECORD  ; INIT THE CURRENT RECORD NUMBER
0D41          SHLD    CURRENT_REC_NMB  ; ...
0D44          XRA     A                ; INIT A REG
0D45          ORA     H                ; SEE IF A FILE HAS BEEN OPENED
0D46          JP      6$               ; BRANCH IF IT HAS
0D49   ;
0D49   ; SEARCH THE DIRECTORY FOR THE DEFAULT FILE
0D49   ;
0D49   20$:   XRA     A                ; INIT THE FILE NAME INDEX
0D4A   22$:   STA     FILE_NAM_INDEX   ; ...
0D4D   ;
0D4D   ; NOW FIGURE OUT WHICH TAPE THIS IS BY TRYING TO OPEN THE TEST STREAM FILES.
0D4D   ;
0D4D   1$:    LXI     H,FILE_FOR_DPM   ; GET ADDRESS OF DPM FILE NAME
0D50          MVI     D,0              ; SETUP A CONSTANT 6 TO ADD TO
0D52          MVI     E,6              ; H & L REG'S
0D54          LDA     FILE_NAM_INDEX   ; GET THE FILE NAME INDEX
0D57   2$:    DCR     A                ; DECREMENT
0D58          JM      3$               ; BRANCH IF DONE INDEXING
0D5B          DAD     D                ; INDEX FILE NAME ADDRESS
0D5C          JMP     2$               ; CONTINUE
0D5F   ;
0D5F   ; POINTER TO RAD50 FILE NAME IS IN H & L
0D5F   ;
```

```
0D5F  3$:     SHLD        FILE_NAM_PTR     ; SAVE FOR 'OPEN FILE' ROUTINE
0D62  ;
0D62  ; TRY AND OPEN THE FILE
0D62  ;
0D62          CALL        OPEN_TU58        ; TRY AND OPEN THE FILE
0D65          JC          7$               ; BRANCH IF ERROR
0D68          LHLD        CURRENT_REC_NMB  ; GET THE CURRENT RECORD NUMBER
0D6B          SHLD        STARTING_RECORD  ; INIT THE STARTING RECORD NUMBER
0D6E          JMP         6$               ; CONTINUE
0D71  ;
0D71  ; FILE IS NOT ON TAPE. SEE IF WE'VE TRIED ALL THE FILE NAMES
0D71  ;
0D71
0D71  7$:     LDA         FILE_NAM_INDEX   ; GET THE CURRENT INDEX
0D74          INR         A                ; INCREMENT
0D75          STA         FILE_NAM_INDEX   ; SAVE
0D78          CPI         NUMBER_OF_FILES  ; TRIED ALL FILE NAMES?
0D7A          JNZ         1$               ; BRANCH IF NO (TRY NEXT NAME)
0D7D          CALL        TYPE_TU58_ERROR  ; TYPE AN ERROR MESSAGE
0D80  5$:     CALL        GET_CMD_LINE     ; GO TO THE COMMAND PARSER
0D83          JMP         5$               ; DON'T ALLOW CONTINUE
0D86  ;
0D86  ; FOUND A FILE. FIND THE RECORD NUMBER OF THE SPECIFIED TEST
0D86  ;
0D86  6$:     CALL        DIRECT_SEARCH    ; GET THE RECORD NUMBER OF THE
0D89                                       ; SPECIFIED TEST
0D89          JNC         8$               ; BRANCH IF NO ERROR
0D8C  ;
0D8C  ; SEE IF ERROR CODE IS -1
0D8C  ;
0D8C          CPI         -1               ; IF -1, GO BACK AND SCAN THE
0D8E                                       ; DIRECTORY AGAIN
0D8E          JZ          20$              ; ...
0D91  ;
0D91  ; TEST SELECTED IS NOT IN THIS FILE. TYPE AN ERROR MESSAGE
0D91  ;
0D91          TYPE        DIRECT_MSG       ; TYPE THE ERROR MESSAGE
0D9A          TYPE        CRLF
0DA3          JMP         5$               ; CALL THE COMMAND PARSER
0DA6  ;
0DA6  ; LOCATION CURRENT_REC_NMB CONTAINS THE RECORD NUMBER FOR THE START OF
0DA6  ; THE OVERLAY. NOW LOAD THE OVERLAY INTO MEMORY.
0DA6  ;
0DA6  8$:     CALL        READ_OVERLAY     ; READ THE OVERLAY INTO MEMORY
0DA9          LHLD        INIT_TEST_PC     ; GET THE INITIAL TEST PC
0DAC          SHLD        TEST_PC          ; INIT THE TEST PC
0DAF  ;
0DAF  ; FIRST OVERLAY IS LOADED, START THE TEST.
0DAF  ;
0DAF          CALL        OP_CODE_DISP     ; START TESTING
0DB2  ;
0DB2  ; RETURN HERE FROM END OF PASS ROUTINE
0DB2  ;
0DB2          TYPE        CRLF             ; TYPE THE END OF PASS MESSAGE
0DBB          TYPE        ENDPASS_MSG      ; ...
```

```
0DC4             TYPEB           PASS_COUNT       ; TYPE THE PASS COUNT
0DCF             TYPE            CRLF             ; ...
0DD8             LDA             PASS_COUNT       ; SEE IF WE HAVE EXECUTED
0DDB             MOV             B,A              ; THE REQUIRED NUMBER OF PASSES
0DDC             LDA             USER_PASS_CNT    ; ...
0DDF             CMP             B                ; ...
0DE0             JZ              9$               ; BRANCH IF NO MORE PASSES
0DE3     10$:    XRA             A                ; INITIALZE THE PARAMETERS FOR
0DE4             STA             TESTS_PER_LINE   ; TYPING THE TEST NUMBERS
0DE7             STA             TYPE_TEST_FLAG   ; ...
0DEA             STA             TEST_NUMBER      ; INIT THE INITIAL TEST NUMBER
0DED             CALL            OPEN_TU58        ; RE-OPEN THE TEST FILE
0DF0             JNC             6$               ; BRANCH IF NO ERRORS TO START
0DF3                                              ; ANOTHER PASS
0DF3     ;
0DF3     ; ERROR WHEN REOPENING THE FILE. TYPE THE MESSAGE AND CALL COMMAND PARSER
0DF3     ;
0DF3             CALL            TYPE_TU58_ERROR  ; TYPE AN ERROR MESSAGE
0DF6     9$:     CALL            GET_CMD_LINE     ; CALL THE COMMAND PARSER
0DF9             JMP             9$               ; DON'T ALLOW CONTINUE
0DFC
```

```
                                          0DFC  2402         .SBTTL  "COMMON ROUTINES
                                          0DFC  2403 ;
                                          0DFC  2404 ; THE FOLLOWING ROUTINES ARE COMMON TO BOTH THE COMMAND PARSER AND
                                          0DFC  2405 ; THE INTERPRETER.
                                          0DFC  2406 ;
                                          0DFC  2407
                                          0DFC  2408
                                          0DFC  2409         .SBTTL  "  READ TU58
                                          0DFC  2410 ;++
                                          0DFC  2411 ;
                                          0DFC  2412 ; FUNCTIONAL DESCRIPTION:
                                          0DFC  2413 ;
                                          0DFC  2414 ;       THIS ROUTINE IS USED TO READ RECORDS FROM THE TU58 SERIAL LINE.
                                          0DFC  2415 ;
                                          0DFC  2416 ; CALLING SEQUENCE:
                                          0DFC  2417 ;
                                          0DFC  2418 ;       THIS ROUTINE IS CALLED BY THE FOLLOWING MACRO:
                                          0DFC  2419 ;
                                          0DFC  2420 ;               READ_RECORD [<BUFFER ADDRESS>],[<NUMBER OF RECORDS>]
                                          0DFC  2421 ;
                                          0DFC  2422 ;       THIS ROUTINE IS CALLED BY THE FOLLOWING ROUTINES:
                                          0DFC  2423 ;
                                          0DFC  2424 ;               READ_OVERLAY
                                          0DFC  2425 ;               DIRECT_SEARCH
                                          0DFC  2426 ;
                                          0DFC  2427 ;       THIS ROUTINE CALLS THE FOLLOWING ROUTINES:
                                          0DFC  2428 ;
                                          0DFC  2429 ;               RDM TU58 DRIVER
                                          0DFC  2430 ;               TYPE_TU58_ERROR
                                          0DFC  2431 ;               GET_CMD_LINE
                                          0DFC  2432 ;
                                          0DFC  2433 ; INPUT PARAMETERS:
                                          0DFC  2434 ;
                                          0DFC  2435 ;       REGISTER PAIR H & L - CONTAINS THE ADDRESS OF A BUFFER TO READ THE
                                          0DFC  2436 ;                             DATA INTO.
                                          0DFC  2437 ;       REGISTER A          - CONTAINS THE NUMBER FO RECORDS TO READ.
                                          0DFC  2438 ;
                                          0DFC  2439 ; IMPLICIT INPUTS:
                                          0DFC  2440 ;
                                          0DFC  2441 ;       CURRENT_REC_NMB   - CONTAINS THE CURRENT POSITION OF THE TAPE
                                          0DFC  2442 ;
                                          0DFC  2443 ; OUPUT PARAMETERS:
                                          0DFC  2444 ;
                                          0DFC  2445 ;       REGISTER PAIR H & L - CONTAINS THE ADDRESS OF THE READ BUFFER.
                                          0DFC  2446 ;
                                          0DFC  2447 ; IMPLICIT OUTPUTS:
                                          0DFC  2448 ;
                                          0DFC  2449 ;       DATA MOVED TO THE BUFFER
                                          0DFC  2450 ;       CURRENT_REC_NMB - CONTAINS THE CURRENT POSITION OF THE TAPE
                                          0DFC  2451 ;       TU58_ERR_CODE - GETS THE ERROR CODE IF AN ERROR OCCURS
                                          0DFC  2452 ;
                                          0DFC  2453 ; SIDE EFFECTS:
                                          0DFC  2454 ;
                                          0DFC  2455 ;       IF AN ERROR OCCURS ON THE READ, AN ERROR MESSAGE IS TYPED FOLLOWED
                                          0DFC  2456 ;       BY THE ERROR CODE, AND EXECUTION IS TRANSFERED TO THE COMMAND
```

```
0DFC  ;         PARSER.
0DFC  ;
0DFC  ;--
0DFC
0DFC
0DFC  READ_TU58:
0DFC            SHLD           TU58_BUFF_ADDR   ; SAVE THE BUFFER ADDRESS
0DFF            STA            TU58_REC_COUNT   ; AND THE RECORD COUNT
0E02            LHLD           CURRENT_REC_NMB  ; GET THE CURRENT RECORD NUMBER
0E05            SHLD           TU58_REC_NUMB    ; SAVE
0E08            MOV            C,A              ; UPDATE CURRENT RECORD NUMBER
0E09            MVI            B,0              ; ...
0E0B            DAD            B                ; ...
0E0C            SHLD           CURRENT_REC_NMB  ; ...
0E0F            $TU58_READ     TU58_BUFF_ADDR,TU58_REC_NUMB,TU58_REC_COUNT
0E1C            JNC            1$               ; EXIT IF NO ERROR
0E1F            STA            TU58_ERR_CODE    ; SAVE THE ERROR CODE
0E22            CALL           TYPE_TU58_ERROR  ; TYPE THE ERROR MESSAGE AND CODE
0E25  2$:       CALL           GET_CMD_LINE     ; GO TO THE COMMAND PARSER
0E28            JMP            2$               ; DON'T ALLOW CONTINUE COMMAND
0E2B  1$:       LHLD           TU58_BUFF_ADDR   ; RESTORE THE BUFFER ADDRESS TO H & L
0E2E            RET                             ; RETURN TO CALLER
0E2F
```

```
0E2F          .SBTTL  "  OPEN FILE ON TU58
0E2F  ;++
0E2F  ; FUNCTIONAL DESCRIPTION
0E2F  ;
0E2F  ;       THIS ROUTINE TRIES TO OPEN A FILE ON THE TU58.
0E2F  ;
0E2F  ; CALLING SEQUENCE
0E2F  ;
0E2F  ;       CALL    OPEN_TU58
0E2F  ;
0E2F  ;       THIS ROUTINE IS CALLED BY THE 'PROGRAM_INIT' ROUTINE
0E2F  ;
0E2F  ;       THIS ROUTINE CALLS THE FOLLOWING ROUTINES:
0E2F  ;
0E2F  ;               TPOPEN (IN RDM SYSTEM SOFTWARE)
0E2F  ;               TYPE_TU58_ERROR (IF DEVICE ERROR OCCURS)
0E2F  ;               GET_CMD_LINE (IF DEVICE ERROR OCCURS)
0E2F  ;
0E2F  ; IMPLICIT INPUTS
0E2F  ;
0E2F  ;       FILE_NAM_PTR - CONTAINS ADDRESS OF RADIX 50 FILE NAME
0E2F  ;
0E2F  ; IMPLICIT OUTPUTS
0E2F  ;
0E2F  ;       CURRENT_REC_NMB - RECEIVES THE STARTING RECORD NUMBER OF THE FILE
0E2F  ;       TU58_ERR_CODE   - RECEIVES AN ERROR CODE IF OPEN FAILED
0E2F  ;       TOTAL_BLOCKS    - RECEIVES THE LENGTH OF THE FILE IN BLOCKS
0E2F  ;
0E2F  ; COMPLETION CODES:
0E2F  ;
0E2F  ;       C BIT SET   - FILE WAS NOT FOUND.
0E2F  ;       C BIT CLEAR - NO ERROR
0E2F  ;
0E2F  ;--
0E2F
0E2F
0E2F  OPEN_TU58:
0E2F          $TU58_OPEN      FILE_NAM_PTR,CURRENT_REC_NMB,TOTAL_BLOCKS
0E3A          JNC             1$              ; BRANCH IF NO ERROR
0E3D          STA             TU58_ERR_CODE   ; SAVE THE ERROR CODE
0E40          CPI             $FILE_NOT_FOUND ; WAS IT A FILE NOT FOUND?
0E42          JZ              3$              ; BRANCH IF YES
0E45          CALL            TYPE_TU58_ERROR ; TYPE AN ERROR MESSAGE
0E48  2$:     CALL            GET_CMD_LINE    ; CALL THE COMMAND PARSER
0E4B          JMP             2$              ; DON'T ALLOW CONTINUE
0E4E  3$:     STC                             ; FILE NOT FOUND
0E4F  1$:     RET                             ; RETURN TO CALLER
0E50
```

```
0E50          .SBTTL  "  TYPE TU58 ERROR MESSAGE
0E50  ;++
0E50  ;
0E50  ; FUNCTIONAL DESCRIPTION:
0E50  ;
0E50  ;       THIS ROUTINE IS USED TO TYPE AN ERROR MESSAGE IF AN ERROR OCCURS
0E50  ;       ON AN 'OPEN' OR 'READ' REQUEST FROM THE TU58.
0E50  ;
0E50  ; CALLING SEQUENCE:
0E50  ;
0E50  ;       CALL    TYPE_TU58_ERROR
0E50  ;
0E50  ;       THIS ROUTINE IS CALLED BY THE FOLLOWING ROUTINES:
0E50  ;
0E50  ;               OPEN_TU58
0E50  ;               READ_TU58
0E50  ;
0E50  ;       THIS ROUTINE CALLS THE FOLLOWING ROUTINES:
0E50  ;
0E50  ;               TYPE
0E50  ;               TYPEB
0E50  ;
0E50  ; IMPLICIT INPUTS:
0E50  ;
0E50  ;       TU58_ERR_CODE - CONTAINS THE ERROR CODE
0E50  ;
0E50  ; IMPLICIT OUTPUTS:
0E50  ;
0E50  ;       NONE
0E50  ;
0E50  ; SIDE EFFECTS:
0E50  ;
0E50  ;       NONE
0E50  ;
0E50  ;--
0E50
0E50
0E50  TYPE_TU58_ERROR:
0E50          TYPE            CRLF            ; TYPE A CARRIAGE RETURN LINE FEED
0E59          TYPE            TU58_ERROR      ; TYPE THE ERROR MESSAGE
0E62          TYPEB           TU58_ERR_CODE   ; TYPE THE ERROR CODE
0E6D          TYPE            TEST_MSG        ; TYPE ' TEST: '
0E76          TYPEB           TEST_NUMBER     ; TYPE THE TEST NUMBER
0E81          RET                             ; RETURN TO CALLER
0E82
```

```
0E82          .SBTTL  "  TYPE ASCII ROUTINE
0E82    ;++
0E82    ;
0E82    ; FUNCTIONAL DESCRIPTION
0E82    ;
0E82    ;       THIS ROUTINE TYPES AN ASCII STRING ON THE CONSOLE TERMINAL.
0E82    ;       THE STRING MUST HAVE ITS CHARACTER COUNT IN THE FIRST BYTE.
0E82    ;       AN ALTERNATE ENTRY POINT (TYPE_ASCII_U) ALLOWS UNCOUNTED STRINGS
0E82    ;       TO BE TYPED.
0E82    ;
0E82    ; CALLING SEQUENCE:
0E82    ;
0E82    ;       THE ROUTINE IS CALLED WITH THE FOLLOWING MACRO:
0E82    ;
0E82    ;               TYPE            MESSAGE_ADDRESS
0E82    ;               TYPE_UNCOUNTED  CHARACTER_COUNT
0E82    ;
0E82    ;       THE ROUTINE IS CALLED BY THE FOLLOWING ROUTINES:
0E82    ;
0E82    ;               PROGRAM_INIT
0E82    ;               TYPE_TU58_ERROR
0E82    ;               TYPE_NUMERIC
0E82    ;               TYPE_TEST_NUMB
0E82    ;               GET_CMD_LINE
0E82    ;               SHOW_ACTION
0E82    ;               BURST CLOCK
0E82    ;               RING_BELL
0E82    ;               TYPE_ERROR
0E82    ;               SINGLE_INSTR
0E82    ;
0E82    ;       THE ROUTINE CALLS THE FOLLOWING ROUTINES:
0E82    ;
0E82    ;               TMWRIT (IN THE RDM SYSTEM SOFTWARE)
0E82    ;
0E82    ; IMPLICIT INPUTS:
0E82    ;
0E82    ;       MESSAGE_POINTER - CONTAINS THE ADDRESS OF THE ASCII STRING.
0E82    ;
0E82    ; IMPLICIT OUTPUTS:
0E82    ;
0E82    ;       NONE
0E82    ;
0E82    ; SIDE EFFECTS:
0E82    ;
0E82    ;       IF AN ERROR OCCURS ON THE TERMINAL WRITE, EXECUTION OF THE
0E82    ;       DIAGNOSTIC IS ABORTED.
0E82    ;
0E82    ;--
0E82
0E82
0E82    TYPE_ASCII:
0E82          LHLD            MESSAGE_POINTER ; GET THE ADDRESS OF THE MESSAGE
0E85          MOV             A,M             ; GET THE STRING LENGTH
0E86          STA             MESSAGE_LENGTH  ; PUT IN IMPLICIT STORAGE
0E89          INX             H               ; POINT PAST THE CHAR COUNT
```

```
0E8A  2631            SHLD           MESSAGE_POINTER ; ...
0E8D  2632 TYPE_ASCII_U:
0E8D  2633            $TERM_WRITE    MESSAGE_POINTER,-
0E8D  2634                           MESSAGE_LENGTH  ; WRITE THE STRING
0E98  2635            JNC            1$              ; EXIT IF NO ERROR
0E9B  2636            ABORT                          ; ABORT, FATAL ERROR
0E9E  2637 1$:        RET                            ; RETURN TO THE CALLER
0E9F  2638
```

```
0E9F          .SBTTL  " TYPE NUMERIC (NIBBLE, BYTE, WORD, OR LONG)
0E9F    ;++
0E9F    ;
0E9F    ; FUNCTIONAL DESCRIPTION:
0E9F    ;
0E9F    ;       THIS ROUTINE IS USED TO TYPE A NUMERIC VALUE ON THE TERMINAL.
0E9F    ;       THE ROUTINE CONVERTS THE VALUE TO AN ASCII STRING AND CALLS THE
0E9F    ;       'TYPE ASCII' ROUTINE.
0E9F    ;
0E9F    ; CALLING SEQUENCE:
0E9F    ;
0E9F    ;       THE ROUTINE IS CALLED WITH ANY ONE OF THE THREE MACRO'S:
0E9F    ;
0E9F    ;               TYPEB   ADDRESS OF DATA
0E9F    ;               TYPEW   ADDRESS OF DATA
0E9F    ;               TYPEL   ADDRESS OF DATA
0E9F    ;
0E9F    ;       THIS ROUTINE IS CALLED BY THE FOLLOWING ROUTINES:
0E9F    ;
0E9F    ;                   (BYTE)
0E9F    ;               PROGRAM_INIT
0E9F    ;               TYPE_TU58_ERROR
0E9F    ;               TYPE_TEST_NUMB
0E9F    ;               GET_CMD_LINE
0E9F    ;               SHOW_ACTION
0E9F    ;               BURST_CLOCK
0E9F    ;                   (WORD)
0E9F    ;               SINGLE_INSTR
0E9F    ;               (BYTE, WORD, AND LONG)
0E9F    ;               TYPE_ERROR
0E9F    ;
0E9F    ;       THE ROUTINE CALLS THE FOLLOWING ROUTINES:
0E9F    ;
0E9F    ;               CNVHEX (IN RDM SYSTEM SOFTWARE)
0E9F    ;
0E9F    ;
0E9F    ; IMPLICIT INPUTS:
0E9F    ;
0E9F    ;       TYPEN_DATA_PTR  - CONTAINS THE ADDRESS OF THE DATA TO TYPE.
0E9F    ;
0E9F    ; EXPLICIT INPUTS:
0E9F    ;
0E9F    ;       A REGISTER      - 81(X) IF TYPE NIBBLE
0E9F    ;                       - 1 IF TYPE BYTE
0E9F    ;                       - 2 IF TYPE WORD
0E9F    ;                       - 4 IF TYPE LONG
0E9F    ;
0E9F    ; IMPLICIT OUTPUTS:
0E9F    ;
0E9F    ;       NONE
0E9F    ;
0E9F    ;--
0E9F
0E9F
0E9F    TYPE_NUMERIC:
```

```
0E9F  2695          PUSH        $PSW              ; SAVE DATA TYPE
0EA0  2696          ANI         <^X7F>            ; DISCARD BIT 7
0EA2  2697          STA         TYPEN_DATA_TYPE   ; SAVE THE DATA TYPE
0EA5  2698          RLC                           ; GENERATE ASCII BYTE COUNT
0EA6  2699          STA         TEMP_BUFFER       ; PUT IN CONVERT BUFFER
0EA9  2700          $CONVERT    TYPEN_DATA_PTR,-; CONVERT THE NUMERIC TO AN
0EA9  2701                      TEMP_BUFFER+1,-   ; AN ASCII STRING
0EA9  2702                      TYPEN_DATA_TYPE   ; ...
0EB4  2703          POP         $PSW              ; GET THE DATA TYPE BACK
0EB5  2704          ANI         <^X80>            ; NIBBLE DATA TYPE?
0EB7  2705          JZ          1$                ; BRANCH IF NO
0EBA  2706          MVI         A,1               ; SET THE ASCII BYTE COUNT
0EBC  2707          STA         TEMP_BUFFER+1     ; TO ONE DIGIT
0EBF  2708          TYPE        TEMP_BUFFER+1     ; TYPE THE SINGLE DIGIT
0EC8  2709          JMP         2$
0ECB  2710 1$:      TYPE        TEMP_BUFFER       ; TYPE THE STRING
0ED4  2711 2$:      RET                           ; RETURN TO CALLER
0ED5  2712
```

```
0ED5                .SBTTL  "  TERMINAL INTERRUPT SERVICE ROUTINE
0ED5        ;++
0ED5        ;
0ED5        ; FUNCTIONAL DESCRIPTION:
0ED5        ;
0ED5        ;       THIS ROUTINE IS INVOKED BY AN INTERRUPT FROM THE TERMINAL.
0ED5        ;       IF THE CHARACTER TYPED WAS A CONTROL C, THE CTRL C BIT IN THE
0ED5        ;       'SWR' IS SET.
0ED5        ;
0ED5        ;       THE ROUTINE INITIATES ANOTHER TERMINAL REQUEST, AND EXITS.
0ED5        ;
0ED5        ; INPUT PARAMETERS:
0ED5        ;
0ED5        ;       C BIT SET   - INDICATES ERROR. ERROR CODE IS IN THE A REGISTER.
0ED5        ;
0ED5        ; IMPLICIT INPUTS:
0ED5        ;
0ED5        ;       NONE
0ED5        ;
0ED5        ; IMPLICIT OUTPUTS:
0ED5        ;
0ED5        ;       SOFT_CTRL_REG - CONTROL_C_FLAG SET IF CHARACTER WAS A CONTROL C.
0ED5        ;
0ED5        ; SIDE EFFECTS:
0ED5        ;
0ED5        ;       IF AN ERROR IS DETECTED, THE ABORT ROUTINE IS CALLED.
0ED5        ;       IF NO ERROR, ANOTHER INPUT REQUEST IS QUEUED.
0ED5        ;
0ED5        ;--
0ED5
0ED5        TERM_INTERRUPT:
0ED5                JNC             10$               ; EXIT IF NO ERROR
0ED8                CPI             $CONTROL_C_CODE   ; WAS IT A CONTROL C?
0EDA                JZ              2$                ; BRANCH IF YES
0EDD                ABORT                             ; ABORT THE DIAGNOSTIC
0EE0        2$:     LDA             SOFT_CTRL_REG     ; GET THE SWITCH REGISTER
0EE3                ORI             CONTROL_C_FLAG    ; SET THE CONTROL C BIT
0EE5                STA             SOFT_CTRL_REG     ; ...
0EE8        10$:    $TERM_READ      TERM_INP_BUFF,-   ; QUE ANOTHER INPUT REQUEST
0EE8                                ONE,TERM_INTERRUPT; ...
0EF3                RET                               ; RETURN TO CALLING SEQUENCE
0EF4
```

```
0EF4            .SBTTL  "  WRITE DCS ROUTINE
0EF4    ;++
0EF4    ;
0EF4    ; FUNCTIONAL DESCRIPTION:
0EF4    ;
0EF4    ;       THIS ROUTINE LOADS THE DIAGNOSTIC CONTROL STORE (DCS). IT IS
0EF4    ;       ENTERED WITH THE STARTING ADDRESS OF THE DCS, THE MICRO WORD
0EF4    ;       COUNT, AND A POINTER TO THE DATA TO LOAD. IT LOADS THE SPECIFIED
0EF4    ;       NUMBER OF MICRO WORDS INTO DCS AT THE STARTING ADDRESS AND
0EF4    ;       RETURNS. THE CURRENT VALUE OF THE DCS ADDRESS REGISTER IS SAVED AND
0EF4    ;       RESTORED.
0EF4    ;
0EF4    ; CALLING SEQUENCE:
0EF4    ;
0EF4    ;       THIS ROUTINE IS INVOKED WITH THE FOLLOWING MACRO:
0EF4    ;
0EF4    ;               WRITE$_DCS      DCS_ADDRESS,DATA_ADDRESS,WORD_COUNT
0EF4    ;
0EF4    ;       THE ROUTINE IS CALLED BY THE FOLLOWING ROUTINES:
0EF4    ;
0EF4    ;               LOAD_DCS
0EF4    ;               READ_OVERLAY
0EF4    ;               DIABLE_CMI
0EF4    ;
0EF4    ; IMPLICIT IMPUTS:
0EF4    ;
0EF4    ;       DCS_WRITE_ADR   - CONTAINS ADDRESS TO START LOADING AT
0EF4    ;       DCS_WRITE_WDCNT - CONTAINS THE NUMBER OF MICRO WORDS TO LOAD
0EF4    ;       DCS_WRITE_PTR   - CONTAINS ADDRESS OF THE DATA TO LOAD
0EF4    ;       DCS_CTRL_RE_CPY - CONTAINS THE CURRENT VALUE OF THE DCS CONTROL REGISTER
0EF4    ;
0EF4    ; IMPLICIT OUTPUTS:
0EF4    ;
0EF4    ;       NONE
0EF4    ;
0EF4    ; SIDE EFFECTS:
0EF4    ;
0EF4    ;       NONE
0EF4    ;
0EF4    ;--
0EF4
0EF4    WRITE_DCS:
0EF4            LDA             DCS_ADDR_REG_RO ; READ AND SAVE THE
0EF7            PUSH            $PSW            ; CONTENTS OF THE DCS ADR REG
0EF8            CALL            DESELECT_DCS    ; DESELECT DCS
0EFB            LDA             DCS_WRITE_ADR   ; GET ADDRESS TO START WRITE
0EFE            STA             DCS_ADDR_REG_WO ; SET THE DCS ADDR REG
0F01            LDA             DCS_WRITE_WDCNT ; GET THE # OF MICRO WRDS TO WRITE
0F04            MOV             B,A             ; SAVE IN REG B
0F05            LHLD            DCS_WRITE_PTR   ; GET THE ADDRESS OF THE DATA
0F08            LXI             D,DCS_DATA_REG  ; GET ADDRESS OF DCS DATA REG
0F0B    ;
0F0B    ; NOW START LOADING THE DCS. REGISTERS H & L HAVE ADDRESS OF DATA.
0F0B    ; REGISTERS D & E HAVE ADDRESS OF DCS DATA REG.
0F0B    ; REGISTER B CONTAINS NO. OF MICRO WORDS TO LOAD.
```

```
0F0B  ;
0F0B  ;
0F0B  ; THE FIRST BYTE OF THE DCS IS THE CONTROL FILE. THE DATA FOR THE CONTROL FILE
0F0B  ; IS IN BYTE 11 SO WE HAVE TO LOAD IT FIRST.
0F0B  ;
0F0B  1$:     PUSH        B                   ; SAVE THE MICRO WORD COUNT
0F0C          MVI         C,10                ; SETUP TO INCREMENT DATA ADDRESS
0F0E          MVI         B,0                 ; BY 10 BYTES
0F10          DAD         B                   ; POINT AT THE CONTROL FILE BYTE
0F11          MOV         A,M                 ; GET THE DATA
0F12          STAX        D                   ; PUT IN THE CONTROL FILE
0F13          MVI         C,<^C10>            ; NOW SETUP TO DECREMENT THE DATA
0F15          MVI         B,-1                ; ADDRESS BY 10 BYTES
0F17          DAD         B                   ; DECREMENT DATA ADDRESS BY 10
0F18          INX         H                   ; BYTES (TWO'S COMPLIMENTS SUBTRACT)
0F19          POP         B                   ; RESTORE THE MICRO WORD COUNT
0F1A          MVI         C,10                ; SET THE BYTE COUNT TO LOAD THE REST
0F1C                                          ; OF THE MICRO WORD
0F1C  ;
0F1C  ; NOW LOAD THE REMAINING 10 BYTES OF THE MICRO WORD
0F1C  ;
0F1C  2$:     MOV         A,M                 ; GET A BYTE OF DATA
0F1D          STAX        D                   ; WRITE THE DCS
0F1E          INX         H                   ; INCREMENT THE DATA ADDRESS
0F1F          DCR         C                   ; LOADED 10 BYTES YET?
0F20          JNZ         2$                  ; BRANCH IF NO
0F23  ;
0F23  ; LOADED A MICRO WORD. NOW CHECK THE MICRO WORD COUNT
0F23  ;
0F23          INX         H                   ; POINT PAST THE ELEVENTH BYTE
0F24          DCR         B                   ; LOADED ALL THE MICRO WORDS YET?
0F25          JNZ         1$                  ; BRANCH IF NO
0F28  ;
0F28  ; DONE LOADING. RESTORE THE DCS ADDRESS AND EXIT.
0F28  ;
0F28          CALL        SELECT_DCS          ; RESTORE THE MATCH REGISTER
0F2B          POP         $PSW                ; GET THE SAVED DCS ADDRESS REG
0F2C          STA         DCS_ADDR_REG_WO     ; RESTORE THE DCS ADD REG
0F2F          RET                             ; RETURN TO CALLER
0F30
```

```
0F30          .SBTTL  "  SINGLE MICRO INSTRUCTION ROUTINE
0F30   ;++
0F30   ;
0F30   ; FUNCTIONAL DESCRIPTION:
0F30   ;
0F30   ;       THIS ROUTINE TICKS THE CPU CLOCK IN 'SINGLE MICRO INSTRUCTION'
0F30   ;       MODE.
0F30   ;
0F30   ; CALLING SEQUENCE:
0F30   ;
0F30   ;       THIS ROUTINE IS CALLED WITH THE FOLLOWING MACRO:
0F30   ;
0F30   ;               SGL_MIC_INSTR   CYCLE_COUNT
0F30   ;
0F30   ;       THE ROUTINE IS CALLED BY THE FOLLOWING ROUTINES:
0F30   ;
0F30   ;               FETCH_MIC_INSTR
0F30   ;               BURST_CLOCK
0F30   ;
0F30   ; IMPLICIT INPUTS:
0F30   ;
0F30   ;       MIC_INSTR_CNT    - CONTAINS THE NUMBER OF MICRO CYCLES TO TICK THE CLK.
0F30   ;       DCS_CTRL_RE_CPY - CONTAINS THE CURRENT VALUE OF THE DCS CONTROL REGISTER
0F30   ;
0F30   ; IMPLICIT OUTPUTS:
0F30   ;
0F30   ;       NONE
0F30   ;
0F30   ;--
0F30
0F30   SINGLE_MIC_INST:
0F30   ;
0F30   ; CLEAR MASTER HALT ENABLE
0F30   ;
0F30           LDA             FRONT_PNL_2       ; GET THE STATE OF THE LIGHTS
0F33           ANI             <TEST!FAULT!REMOTE> ; MASK
0F35           STA             RD_CTRL_REG       ; CLEAR MASTER HALT ENABLE
0F38   ;
0F38           LDA             MIC_INSTR_CNT     ; GET THE COUNT
0F3B           MOV             B,A               ; SAVE IN 'B' REGISTER
0F3C           LXI             H,DCS_CTRL_REG    ; GET ADDRESS OF CONTROL REG
0F3F
0F3F   1$:     LDA             DCS_CTRL_RE_CPY   ; GET CONTROL REG COPY
0F42           ANI             <^CCLK_CTRL_1>    ; ACTIVATE CONTROL BIT ONE
0F44           MOV             M,A               ; WRITE THE CONTROL REGISTER
0F45           XRI             CLK_CTRL_1        ; INACTIVATE CONTROL BIT 1
0F47           MOV             M,A               ; WRITE THE CONTROL REGISTER
0F48           DCR             B                 ; DECREMENT THE COUNT
0F49           JNZ             1$                ; CONTINUE UNTIL 0
0F4C           LDA             FRONT_PNL_2       ; GET STATE OF THE LIGHTS
0F4F           ANI             <TEST!FAULT!REMOTE> ; ...
0F51           ORI             <MASTER_HALT_EN!-; SET MASTER HALT ENABLE AND TRAP
0F51                           TRAP_HALT_EN>     ; HALT ENABLE
0F53           STA             RD_CTRL_REG       ; ...
0F56           RET                               ; RETURN TO CALLING SEQUENCE
```

```
                              0F57  2908
```

```
0F57       .SBTTL  "  SINGLE CLOCK TICK ROUTINE
0F57  ;++
0F57  ;
0F57  ; FUNCTIONAL DESCRIPTION:
0F57  ;
0F57  ;     THIS ROUTINE STEPS THE CPU CLOCK THE SPECIFIED NUMBER OF TIMES.
0F57  ;
0F57  ; CALLING SEQUENCE:
0F57  ;
0F57  ;     THIS ROUTINE IS CALLED BY THE FOLLOWING MACRO:
0F57  ;
0F57  ;          SGL_TICK      TICKS
0F57  ;
0F57  ;     THE ROUTINE IS CALLED BY THE 'BURST_CLOCK' ROUTINE.
0F57  ;
0F57  ; IMPLICIT INPUTS:
0F57  ;
0F57  ;     MIC_TICK_CNT    - CONTAINS THE NUMBER OF TICKS TO STEP THE CLOCK.
0F57  ;     DCS_CTRL_RE_CPY - CONTAINS THE CURRENT VALUE OF THE DCS CONTROL REGISTER
0F57  ;
0F57  ; IMPLICIT OUTPUTS:
0F57  ;
0F57  ;     NONE
0F57  ;
0F57  ;--
0F57
0F57  SINGLE_TICK:
0F57       LDA       FRONT_PNL_2        ; GET STATE OF LIGHTS
0F5A       ANI       <TEST!FAULT!REMOTE> ; ...
0F5C       STA       RD_CTRL_REG        ; CLEAR MASTER HALT ENABLE
0F5F       LDA       MIC_TICK_CNT       ; GET THE COUNT
0F62       MOV       B,A                ; SAVE IN B REGISTER
0F63       LXI       H,DCS_CTRL_REG     ; GET ADDRESS OF CONTROL REG
0F66
0F66  1$:  LDA       DCS_CTRL_RE_CPY    ; READ THE CONTROL REGISTER COPY
0F69       ANI       <^CCLK_CTRL_0>     ; ACTIVATE CONTROL BIT 0
0F6B       MOV       M,A                ; WRITE THE CONTROL REG
0F6C       ORI       CLK_CTRL_0         ; INACTIVATE CONTROL BIT 0
0F6E       MOV       M,A                ; WRITE THE CONTROL REG
0F6F       MVI       A,<^XFF>           ; SETUP TO DECREMENT THE COUNT
0F71       ADD       B                  ; DECREMENT THE COUNT
0F72       MOV       B,A                ; ...
0F73       JNZ       1$                 ; CONTINUE UNTIL 0
0F76       LDA       FRONT_PNL_2        ; GET STATE OF LIGHTS
0F79       ANI       <TEST!FAULT!REMOTE> ; ...
0F7B       ORI       <MASTER_HALT_EN!-  ; SET MASTER HALT ENABLE AND TRAP
0F7B                 TRAP_HALT_EN>      ; HALT ENABLE
0F7D       STA       RD_CTRL_REG        ; CLEAR MASTER HALT ENABLE
0F80       RET                          ; RETURN TO CALLING SEQUENCE
0F81
```

```
0F81            .SBTTL  "  ABORT ROUTINE
0F81    ;++
0F81    ;
0F81    ; FUNCTIONAL DESCRIPTION:
0F81    ;
0F81    ;       THIS ROUTINE INITIALIZES THE CPU AND THE STACK POINTER AND
0F81    ;       RETURNS TO THE RDM MONITOR.
0F81    ;
0F81    ; CALLING SEQUENCE:
0F81    ;
0F81    ;       THIS ROUTINE IS CALLED BY THE FOLLOWING MACRO:
0F81    ;
0F81    ;               ABORT
0F81    ;
0F81    ;       THE ROUTINE IS CALLED BY THE FOLLOWING ROUTINES:
0F81    ;
0F81    ;               PROGRAM_INIT
0F81    ;               TYPE_ASCII
0F81    ;               TERM_INTERRUPT
0F81    ;               RETURN_ACTION
0F81    ;
0F81    ; IMPLICIT INPUTS:
0F81    ;
0F81    ;       NONE
0F81    ;
0F81    ; IMPLICIT OUTPUTS:
0F81    ;
0F81    ;       NONE
0F81    ;
0F81    ; SIDE EFFECTS:
0F81    ;
0F81    ;       THE DIAGNOSTIC IS EXITED.
0F81    ;
0F81    ;--
0F81
0F81    RETURN_TO_RDM:
0F81            CALL            CPU_INIT        ; ASCERT DC LOW
0F84            CALL            ENABLE_CMI      ; ENABLE THE CMI WITH PROC INIT
0F87            CALL            FETCH_0         ; FETCH CS LOCATION 0
0F8A            MVI             A,0             ; TURN ON THE CLOCK
0F8C            STA             MAINT_REG       ; ENABLE THE CLOCK INTERRUPT
0F8F            STA             DCS_CTRL_REG    ; TURN ON THE CPU CLOCK
0F92            LDA             FRONT_PNL_2     ; GET STATE OF LIGHTS
0F95            ANI             <REMOTE!FAULT!TEST>; ...
0F97            STA             RD_CTRL_REG     ; CLEAR MASTER HALT ENABLE
0F9A            LHLD            SAVED_SP        ; GET ADDRESS OF RDM RETURN POINT
0F9D            SPHL                            ; LOAD THE STACK POINTER
0F9E            RET                             ; RETURN TO RDM
0F9F
```

```
0F9F              .SBTTL  "  SET DCS CONTROL FILE ROUTINE
0F9F       ;++
0F9F       ;
0F9F       ; FUNCTIONAL DESCRIPTION:
0F9F       ;
0F9F       ;          THIS ROUTINE SETS BITS IN THE DCS CONTROL FILE. IT IS ENTERED WITH
0F9F       ;          THE DCS ADDRESS AND A BIT MAP.
0F9F       ;
0F9F       ; CALLING SEQUENCE:
0F9F       ;
0F9F       ;          CALL     SET_DCS_CF
0F9F       ;
0F9F       ;          THE ROUTINE IS CALLED BY THE FOLLOWING ROUTINES:
0F9F       ;
0F9F       ;                   SET_ACTION
0F9F       ;                   BURST_CLOCK
0F9F       ;
0F9F       ; IMPLICIT INPUTS:
0F9F       ;
0F9F       ;          SETCLR_CF_ADDR - CONTAINS THE DCS ADDRESS TO SET THE BITS AT.
0F9F       ;          SETCLR_CF_DATA - CONTAINS THE BIT MAP OF THE BITS TO SET.
0F9F       ;          DCS_CTRL_RE_CPY- CONTAINS THE CURRENT VALUE OF THE DCS CONTROL REGISTER
0F9F       ;
0F9F       ; IMNPLICIT OUTPUTS:
0F9F       ;
0F9F       ;          NONE
0F9F       ;
0F9F       ;--
0F9F
0F9F      SET_DCS_CF:
0F9F              LDA                DCS_ADDR_REG_RO ; GET THE CURRENT DCS ADDRESS
0FA2              DCR                A               ; BACKUP TO CURRENT MICRO INSTR
0FA3              PUSH               $PSW            ; SAVE IT ON THE STACK
0FA4              CALL               DESELECT_DCS    ; MAKE SURE DCS IS DESELECTED
0FA7              LDA                SETCLR_CF_ADDR  ; GET ADDRESS TO SET BITS AT
0FAA              STA                DCS_ADDR_REG_WO ; LOAD THE DCS ADDRESS REGISTER
0FAD              STA                DCS_ADDR_REG_WO ; ...
0FB0              LDA                SETCLR_CF_DATA  ; GET BITS TO SET
0FB3              CMA                                ; ...
0FB4              MOV                B,A             ; SAVE IN B REG
0FB5              LDA                DCS_CTRL_FILE   ; GET THE CURRENT BITS
0FB8              ANA                B               ; INSERT NEW BITS
0FB9              STA                DCS_DATA_REG    ; LOAD THE CONTROL FILE
0FBC              CALL               SELECT_DCS      ; RESTORE THE MATCH REGISTER
0FBF              POP                $PSW            ; GET SAVED DCS ADDRESS REG
0FC0              STA                DCS_ADDR_REG_WO ; RESTORE THE SAVED ADDRESS
0FC3              STA                DCS_ADDR_REG_WO ; ...
0FC6              PUSH               $PSW            ; SAVE A REG
0FC7              MVI                A,<MAINT_TRAP_OFF!- ; DISABLE LOADING THE CONTROL FILE
0FC7                                 MAINT_STB_INH!-
0FC7                                 MAINT_DCS_ENABL>;
0FC9              STA                MAINT_REG       ; ...
0FCC              POP                $PSW            ; GET SAVED DCS ADDRESS
0FCD              INR                A               ; POINT AT ADDRESS BEING FETCHED
0FCE              STA                DCS_ADDR_REG_WO ; LOAD THE ADDRESS REGISTER
```

```
                                   0FD1  3066        MVI                 A,<MAINT_TRAP_OFF!-
                                   0FD1  3067                            MAINT_DCS_ENABL>; DEASCERT STROBE INHIBIT
                                   0FD3  3068        STA                 MAINT_REG        ; ...
                                   0FD6  3069        RET                                  ; EXIT
                                   0FD7  3070
```

```
0FD7          .SBTTL  "  CLEAR DCS CONTROL FILE ROUTINE
0FD7  ;++
0FD7  ;
0FD7  ; FUNCTIONAL DESCRIPTION:
0FD7  ;
0FD7  ;       THIS ROUTINE CLEARS BITS IN THE DCS CONTROL FILE. IT IS ENTERED
0FD7  ;       WITH THE DCS ADDRESS AND A BIT MAP OF THE BITS TO CLEAR.
0FD7  ;
0FD7  ; CALLING SEQUENCE:
0FD7  ;
0FD7  ;       CALL    CLEAR_DCS_CF
0FD7  ;
0FD7  ;       THE ROUTINE IS CALLED BY THE FOLLOWING ROUTINES:
0FD7  ;
0FD7  ;               CLEAR_ACTION
0FD7  ;               BURST_CLOCK
0FD7  ;
0FD7  ; IMPLICIT INPUTS:
0FD7  ;
0FD7  ;       SETCLR_CF_ADDR - CONTAINS THE DCS ADDRESS TO CLEAR THE BITS AT
0FD7  ;       SETCLR_CF_DATA - CONTAINS THE BIT MAP OF THE BITS TO CLEAR
0FD7  ;
0FD7  ; IMPLICIT OUTPUTS:
0FD7  ;
0FD7  ;       NONE
0FD7  ;
0FD7  ;--
0FD7
0FD7  CLEAR_DCS_CF:
0FD7          LDA             DCS_ADDR_REG_RO ; GET THE CURRENT DCS ADDRESS
0FDA          DCR             A               ; POINT AT CURRENTLY LATCHED ADDRESS
0FDB          PUSH            $PSW            ; SAVE
0FDC          CALL            DESELECT_DCS    ; MAKE SURE DCS IS DESELECTED
0FDF          LDA             SETCLR_CF_ADDR  ; GET ADDRESS TO CLEAR THE BITS AT
0FE2          STA             DCS_ADDR_REG_WO ; LOAD THE DCS ADDR REG
0FE5          STA             DCS_ADDR_REG_WO ; ...
0FE8          LDA             SETCLR_CF_DATA  ; GET BIT MAP TO CLEAR
0FEB          MOV             B,A             ; SAVE IN B REG
0FEC          LDA             DCS_CTRL_FILE   ; GET THE CURRENT BITS IN THE CTRL FILE
0FEF          ORA             B               ; CLEAR THE BITS IN THE CF
0FF0          STA             DCS_DATA_REG    ; WRITE THE CONTROL FILE
0FF3          CALL            SELECT_DCS      ; RESTORE THE MATCH REGISTER
0FF6          POP             $PSW            ; GET SAVED DCS ADDRESS
0FF7          STA             DCS_ADDR_REG_WO ; RESTORE THE DCS ADDRESS
0FFA          STA             DCS_ADDR_REG_WO ; ...
0FFD          PUSH            $PSW            ; SAVE A REG
0FFE          MVI             A,<MAINT_TRAP_OFF!- ; DISABLE LOADING THE
0FFE                          MAINT_STB_INH!-
0FFE                          MAINT_DCS_ENABL>; CONTROL FILE
1000          STA             MAINT_REG       ; OF CONTROL FILE
1003          POP             $PSW            ; GET SAVED DCS ADDRESS
1004          INR             A               ; POINT AT ADDRESS BEING FETCHED
1005          STA             DCS_ADDR_REG_WO ; LOAD THE ADDRESS REGISTER
1008          MVI             A,<MAINT_TRAP_OFF!-
1008                          MAINT_DCS_ENABL>; DISABLE THE STROBE INHIBIT
```

```
          100A  3127          STA                MAINT_REG         ; ...
          100D  3128          RET                                  ; EXIT
          100E  3129
```

```
100E           .SBTTL  "  READ OVERLAY ROUTINE
100E   ;++
100E   ;
100E   ; FUNCTIONAL DESCRIPTION:
100E   ;
100E   ;       THIS ROUTINE READS A TEST OVERLAY FROM THE CURRENTLY OPEN
100E   ;       FILE ON THE TU58. THE OVERLAY FORMAT IS AS FOLLOWS. THE OVERLAY
100E   ;       HAS TWO FORMATS DEPENDING ON THE CONTENTS OF THE FIRST BYTE:
100E   ;
100E   ;       FORMAT 1 - NO DATA TO LOAD INTO CPU OR DCS
100E   ;               .BYTE   1 (BITS <6:0>)
100E   ;               .BYTE   NOT USED
100E   ;               .BYTE   NUMBER OF RECORDS IN THE TEST
100E   ;               .ASCIC  TEST TITLE
100E   ;               .BLKB   TEST INSTRUCTIONS
100E   ;                 .
100E   ;                 .
100E   ;               .BLKB   FILL TO A RECORD BOUNDRY
100E   ;
100E   ;       FORMAT 2 - DATA TO LOAD INTO CPU OR DCS
100E   ;               .BYTE   0 (BITS <6:0>)
100E   ;               .BYTE   NUMBER OF RECORDS OF DATA
100E   ;               .BYTE   NUMBER OF R TEMP LONG WORDS
100E   ;               .BYTE   R TEMP DATA
100E   ;                 .
100E   ;               .BYTE   NUMBER OF CACHE LONG WORDS
100E   ;               .BYTE   CACHE DATA
100E   ;                 .
100E   ;               .BYTE   NUMBER OF DCS MICRO WORDS
100E   ;               .BYTE   DCS DATA
100E   ;                 .
100E   ;                 .
100E   ;               .BLKB   FILL TO A RECORD BOUNDRY
100E   ;               .BYTE   NUMBER OF RECORDS IN THE TEST
100E   ;               .ASCIC  TEST TITLE
100E   ;               .BYTE   TEST INSTRUCTIONS
100E   ;                 .
100E   ;                 .
100E   ;               .BLKB   FILL TO A RECORD BOUNDRY
100E   ;
100E   ;       THE RTEMP, CACHE, AND DCS DATA IS LOADED INTO THE RESPECTIVE
100E   ;       PLACES, THEN THE THE TEST IS READ INTO THE BUFFER.
100E   ;
100E   ;       IF NO DCS DATA IS PRESENT IN THE OVERLAY, THE DCS IS LOADED WITH NOP
100E   ;       MICRO INSTRUCTIONS.
100E   ;
100E   ; CALLING SEQUENCE:
100E   ;
100E   ;       CALL    READ_OVERLAY
100E   ;
100E   ;       THE ROUTINE IS CALLED BY THE FOLLOWING ROUTINES:
100E   ;
100E   ;               PROGRAM_INIT
100E   ;               END_TEST
100E   ;
```

```
100E   ;       THE ROUTINE CALLS THE FOLLOWING ROUTINES:
100E   ;
100E   ;               READ_TU58
100E   ;               WRITE_DCS
100E   ;               TYPE_TEST_NUMB
100E   ;
100E   ; IMPLICIT INPUTS:
100E   ;
100E   ;       CURRENT_REC_NMB - CONTAINS THE RECORD NUMBER OF THE CURRENTLY
100E   ;                         OPEN FILE.
100E   ;       TEST_BUFFER     - IS THE ADDRESS OF THE READ BUFFER
100E   ;
100E   ; IMPLICIT OUTPUTS:
100E   ;
100E   ;       CURRENT_REC_NMB - POINTS TO THE CURRENT RECORD OF THE CURRENTLY
100E   ;                         OPEN FILE.
100E   ;       INIT_TEST_PC    - GETS LOADED WITH THE PC OF THE BASE OF THE TEST
100E   ;
100E   ;--
100E
100E   READ_OVERLAY:
100E           READ_RECORD     TEST_BUFFER,1     ; READ THE FIRST RECORD OF THE
1016                                             ; OVERLAY INTO 'TEST_BUFFER'
1016   ;
1016   ; BUFFER POINTER IS IN H & L
1016   ;
1016           MOV             A,M               ; GET THE DATA FLAG OF THE OVERLAY
1017           PUSH            $PSW              ; SAVE
1018           ANI             1                 ; CPU DATA?
101A           JNZ             40$               ; BRANCH IF NO
101D           INX             H                 ; POINT AT RECORD COUNT FOR DATA
101E           MOV             A,M               ; GET THE RECORD COUNT
101F           DCR             A                 ; SUBTRACT RECORD JUST READ
1020           JZ              1$                ; BRANCH IF ALL RECORDS IN MEMORY
1023   ;
1023   ; READ THE REST OF THE DATA INTO THE BUFFER
1023   ;
1023           MOV             B,A               ; SAVE A
1024           POP             $PSW              ; GET TEST SECTION FLAG
1025           PUSH            $PSW              ; SAVE AGAIN
1026           CALL            CHECK_QV          ; CHECK IF QV PASS
1029           JC              30$               ; BRANCH IF TEST IS TO BE SKIPPED
102C           MOV             A,B               ; RESTORE RECORD NUMBER
102D           PUSH            H                 ; SAVE THE BUFFER POINTER
102E           READ_RECORD     TEST_BUFFER+128   ; READ THE REST OF THE DATA
1034           POP             H
1035   ;
1035   ;
1035   ; FILL THE DCS WITH NOP INSTRUCTIONS.
1035   ;
1035   1$:     PUSH            H
1036           CALL            FILL_DCS          ; FILL DCS WITH NOP'S
1039           POP             H                 ; RESTORE THE BUFFER POINTER
103A   ;
103A   ; NOW LOAD THE R TEMP DATA
```

```
103A  ;
103A          INX             H                ; INCREMENT BUFFER POINTER
103B          MOV             A,M              ; GET THE WORD COUNT
103C          CPI             -1               ; IS THERE ANY R TEMP DATA?
103E          JZ              10$              ; BRANCH IF NO
1041
1041          INX             H                ; POINT AT FIRST BYTE OF DATA
1042          PUSH            $PSW             ; SAVE THE COUNT
1043          XRA             A                ; INIT THE REGISTER NUMBER
1044          STA             TYPE_ERR_TEMP    ; ...
1047  5$:     PUSH            H                ; SAVE DATA POINTER
1048          LXI             D,R_RDM_MIC_WORD; GET ADDRESS OF MICRO WORD
104B          LXI             H,TYPE_ERR_TEMP  ; AND ADDESS OF REGISTER NUMBER
104E          MVI             B,34             ; PUT STARTING BIT NUMBER IN B REG
1050          MVI             C,39             ; AND ENDING BIT NUMBER
1052          XRA             A                ; CLEAR INHIBIT PARITY FLAG
1053          PUSH            $PSW             ; CLEAR INHIBIT SCRAMBLE FLAG
1054          CALL            INSERT_FIELD     ; PUT REGISTER NUMBER IN MICRO WORD
1057          WRITE$_DCS      DCS_SCRATCH_ADR,-
1057                          R_RDM_MIC_WORD,1; LOAD THE DCS
106A          WRITE$_DCS      DCS_SCRATCH_ADR+1,-
106A                          NOP_MICRO_WORD,1 ; ...
107D          POP             H                ; RESTORE H & L
107E  ;
107E  ; PUT THE DATA IN THE D REGISTER
107E  ;
107E          LXI             D,WD_REG_BYTE_0  ; GET ADDRESS OF D REGISTER
1081          MVI             C,4              ; SETUP LOOP FOR 4 BYTES
1083  7$:     MOV             A,M              ; GET BYTE OF DATA
1084          STAX            D                ; PUT IN D REGISTER
1085          INX             H                ; INCREMENT POINTERS
1086          INX             D                ; ...
1087          DCR             C                ; DONE 4 BYTES?
1088          JNZ             7$               ; BRANCH IF NO
108B  ;
108B  ; EXECUTE THE MICRO INSTRUCTION
108B  ;
108B          EXECUTE         DCS_SCRATCH_ADR,1 ; EXECUTE ONE MICRO INSTRUCTION
109E          LDA             TYPE_ERR_TEMP    ; INCREMENT THE REGISTER NUMBER
10A1          INR             A                ; ...
10A2          STA             TYPE_ERR_TEMP    ; ...
10A5          POP             $PSW             ; GET THE REGISTER COUNT BACK
10A6          DCR             A                ; LOADED ALL THE REGISTERS?
10A7          PUSH            $PSW             ; SAVE THE REGISTER COUNT
10A8          JNZ             5$               ; BRANCH IF NO
10AB          POP             $PSW             ; CLEANUP THE STACK
10AC          DCX             H                ; BACKUP POINTER FOR CACHE DATA
10AD  ;
10AD  ; NOW LOAD THE CACHE DATA
10AD  ;
10AD  10$:    INX             H                ; INC POINTER
10AE          MOV             A,M              ; GET THE COUNT
10AF          CPI             -1               ; IS THERE ANY CACHE DATA?
10B1          JZ              20$              ; BRANCH IF NO
10B4
```

```
10B4         INX             H               ; POINT AT THE FIRST DATA WORD
10B5         PUSH            $PSW            ; SAVE THE WORD COUNT
10B6         PUSH            H               ; SAVE THE DATA POINTER
10B7         CALL            DISABLE_CMI     ; DISABLE THE CMI
10BA ;
10BA ; ENABLE THE CACHE
10BA ; MICRO CODE TO DISABLE THE CMI IS ALREADY IN DCS
10BA ;
10BA         WRITE$_DCS      DCS_SCRATCH_ADR+1,-
10BA                         CACHE_ENA_ADR,1 ; LOAD LONLIT TO SELECT REGISTER
10CD         WRITE$_DCS      DCS_SCRATCH_ADR+3,-
10CD                         CACHE_ENA_DATA,1; AND THE DATA TO WRITE
10E0         EXECUTE         DCS_SCRATCH_ADR+1,4 ; EXECUTE THE INSTRUCTIONS
10F3 ;
10F3 ; NOW LOAD THE MICRO CODE TO LOAD THE CACHE
10F3 ;
10F3         WRITE$_DCS      DCS_SCRATCH_ADR,-
10F3                         CACHE_WRITE,4   ; LOAD THE MICRO CODE
1106         LXI             H,0             ; INIT THE VA TO ZERO
1109         SHLD            WD_REG_BYTE_0   ; ...
110C         SHLD            WD_REG_BYTE_2   ; ...
110F         EXECUTE         DCS_SCRATCH_ADR,1 ; ...
1122         POP             H               ; GET THE ADDRESS OF THE DATA
1123 ;
1123 ; PUT A DATA WORD IN THE D REGISTER
1123 ;
1123 13$:    MVI             B,4             ; SET THE LOOP COUNT
1125         LXI             D,WD_REG_BYTE_0 ; GET ADDRESS OF D REGISTER
1128 15$:    MOV             A,M             ; GET A BYTE OF DATA
1129         STAX            D               ; PUT IN D REGISTER
112A         INX             H               ; INCREMENT THE POINTERS
112B         INX             D               ; ...
112C         DCR             B               ; DONE 4 BYTES?
112D         JNZ             15$             ; BRANCH IF NO
1130 ;
1130 ; LOAD THE CACHE
1130 ;
1130         EXECUTE         DCS_SCRATCH_ADR+1,3
1143 ;
1143 ; CHECK IF ALL DATA LOADED
1143 ;
1143         POP             $PSW            ; GET THE WORD COUNT
1144         DCR             A               ; DONE?
1145         PUSH            $PSW            ; SAVE THE WORD COUNT
1146         JNZ             13$             ; BRANCH IF MORE WORDS
1149         POP             $PSW            ; CLEANUP THE STACK
114A         DCX             H               ; BACKUP THE POINTER
114B ;
114B ; NOW LOAD THE DCS
114B ;
114B ; NOW LOAD THE DCS IF THERE IS DATA TO LOAD
114B ;
114B 20$:    INX             H               ; INC POINTER
114C         MOV             A,M             ; GET THE DCS WORD COUNT
114D         MOV             B,A             ; SAVE IN B REGISTER
```

```
114E             CPI             -1                ; IS THERE ANY DCS DATA?
1150             JZ              30$               ; BRANCH IF NO
1153             INX             H                 ; POINT AT FIRST DCS BYTE
1154             WRITE$_DCS      0                 ; LOAD DCS WITH THE MICRO WORDS
1163     ;
1163     ; NOW READ THE TEST INTO THE TEST BUFFER
1163     ;
1163     30$:    LXI             H,TEST_BUFFER     ; INIT THE INITIAL TEST PC
1166             INX             H                 ; ...
1167             SHLD            INIT_TEST_PC      ; ...
116A             MOV             A,L               ; AND THE TWO'S COMPLIMENT OF THE
116B             CMA                               ; INITIAL TEST PC
116C             MOV             L,A               ; ...
116D             MOV             A,H               ; ...
116E             CMA                               ; ...
116F             MOV             H,A               ; ...
1170             INX             H                 ; ...
1171             SHLD            C_INIT_TEST_PC    ; ...
1174             READ_RECORD     TEST_BUFFER,1     ; READ THE NEXT RECORD
117C     32$:    MOV             A,M               ; GET THE NUMBER OF RECORDS IN THE TEST
117D             STA             TEST_SIZE         ; SAVE
1180             DCR             A                 ; ACCOUNT FOR ONE JUST READ
1181             MOV             B,A               ; SAVE NUMBER OF RECORDS
1182             POP             $PSW              ; GET TEST SECTION FLAG
1183             PUSH            $PSW              ; SAVE AGAIN
1184             CALL            CHECK_QV          ; CHECK IF QV PASS
1187             JNC             33$               ; BRANCH IF TEST IS NOT TO BE SKIPPED
118A             MVI             B,1               ; SETUP FOR NO TYPEOUT
118C             CALL            TYPE_TEST_NUMB    ; UPDATE TEST NUMBER
118F             POP             $PSW              ; DISCARD TEST SECTION FLAG
1190             LDA             TEST_NUMBER       ; UPDATE TEST NUMBER
1193             INR             A                 ; ...
1194             STA             TEST_NUMBER       ; ...
1197             MOV             B,A               ; SAVE
1198             LDA             TOTAL_NMB_TESTS   ; GET TOTAL NUMBER OF TESTS ON TAPE
119B             CMP             B                 ; DONE YET?
119C             JNZ             READ_OVERLAY      ; BRANCH IF NO
119F     ;
119F     ; SKIPPED THE LAST TEST, SIMULATE AN END_FILE PSEUDO OP.
119F     ;
119F             LHLD            INIT_TEST_PC      ; GET INITIAL TEST PC
11A2             SHLD            TEST_PC
11A5             MVI             M,OP_END_FILE     ; PUT END FILE OPCODE IN MEMORY
11A7             INX             H                 ; ...
11A8             MVI             M,0               ; ...
11AA             JMP             36$               ; EXIT
11AD     ;
11AD     ; NOT A QV PASS OR TEST IS PART OF QV SECTION. READ IT IN
11AD     ;
11AD     33$:    MOV             A,B               ; RESTORE NUMBER OF RECORDS TO READ
11AE             ORA             A                 ; ANY MORE RECORDS TO READ?
11AF             JZ              31$               ; BRANCH IF NO
11B2             READ_RECORD     TEST_BUFFER+128   ; READ THE REST OF THE TEST INTO MEMORY
11B8     31$:    MVI             B,0               ; SETUP TO TYPE TEST NUMBER
11BA             CALL            TYPE_TEST_NUMB    ; TYPE THE TEST NUMBER
```

```
11BD      35$:     POP         $PSW               ; RETURN THE TEST SECTION FLAG
11BE      36$:     RET                            ; EXIT
11BF
11BF      ;
11BF      ; THERE WAS NO DATA FOR THIS TEST. READ THE REST OF THE TEST INTO MEMORY.
11BF      ;
11BF      40$:     INX         H                  ; POINT AT THE FIRST INSTRUCTION
11C0               INX         H                  ; FOR THE TEST
11C1               PUSH        H                  ; SAVE H & L
11C2               INX         H                  ; ...
11C3               SHLD        INIT_TEST_PC       ; INIT THE INITIAL TEST PC
11C6               MOV         A,L                ; AND THE TWO'S COMPLIMENT INITIAL
11C7               CMA                            ; TEST PC
11C8               MOV         L,A                ; ...
11C9               MOV         A,H                ; ...
11CA               CMA                            ; ...
11CB               MOV         H,A                ; ...
11CC               INX         H                  ; ...
11CD               SHLD        C_INIT_TEST_PC     ; ...
11D0               POP         H                  ; POINT BACK AT RECORD COUNT
11D1               JMP         32$                ; READ THE REST OF THE TEST INTO MEMORY
11D4
```

```
11D4        .SBTTL  "  CHECK QV ROUTINE"
11D4  ;++
11D4  ;
11D4  ; FUNCTIONAL DESCRIPTION:
11D4  ;
11D4  ;     THIS ROUTINE IS USED TO CHECK IF A QV PASS HAS BEEN SPECIFIED AND
11D4  ;     IF THE TEST TO BE EXECUTED IS IN THE QV SECTION.
11D4  ;
11D4  ; CALLING SEQUENCE:
11D4  ;
11D4  ;     CALL    CHECK_QV
11D4  ;
11D4  ;     THE ROUTINE IS CALLED BY THE "READ_OVERLAY" ROUTINE.
11D4  ;
11D4  ; EXPLICIT INPUTS:
11D4  ;
11D4  ;     A - CONTAINS THE TEST SECTION FLAG
11D4  ;     B - CONTAINS THE NUMBER OF RECORDS TO SKIP
11D4  ;
11D4  ; EXPLICIT OUTPUTS:
11D4  ;
11D4  ;     B - UNCHANGED
11D4  ;
11D4  ; IMPLICIT OUTPUTS:
11D4  ;
11D4  ;     CURRENT_REC_NMB - UPDATED IF THIS IS A QV PASS AND THE TEST IS
11D4  ;                       NOT PART OF THE QV SECTION.
11D4  ;
11D4  ; ROUTINE VALUE:
11D4  ;
11D4  ;     CF - SET IF THE CURRENT_REC_NMB WAS UPDATED.
11D4  ;--
11D4  CHECK_QV:
11D4        PUSH        $PSW            ; SAVE A REG
11D5        LDA         SOFT_CTRL_REGB  ; GET QV FLAG
11D8        ANI         QV_FLAG         ; ...
11DA        JZ          10$             ; BRANCH IF NOT QV PASS
11DD        POP         $PSW
11DE        PUSH        $PSW
11DF        ANI         <^X80>          ; IS THIS TEST IN QV SECTION?
11E1        JNZ         10$             ; BRANCH IF YES
11E4   ;
11E4   ; SKIP THIS TEST
11E4   ;
11E4        PUSH        H
11E5        LHLD        CURRENT_REC_NMB ; GET CURRENT TAPE RECORD NUMBER
11E8        PUSH        B
11E9        MOV         C,B
11EA        MVI         B,0             ; SKIP DATA FOR THIS TEST
11EC        DAD         B
11ED        POP         B
11EE        SHLD        CURRENT_REC_NMB ; RESTORE
11F1        POP         H
11F2        POP         $PSW
11F3        STC
```

```
11F4  3484          JMP         20$                ; EXIT
11F7  3485 10$:     POP         $PSW
11F8  3486          STC
11F9  3487          CMC
11FA  3488 20$:     RET
11FB  3489
```

```
11FB              .SBTTL  "  READ V BUS ROUTINE
11FB      ;++
11FB      ;
11FB      ; FUNCTIONAL DESCRIPTION:
11FB      ;
11FB      ;       THIS ROUTINE READS THE CURRENT STATE OF THE VISIBILITY BUS
11FB      ;       INTO A TEMPORARY BUFFER. EACH BYTE OF THE BUFFER CONTAINS ONE BIT
11FB      ;       OF THE V BUS IN BIT POSITION 0.
11FB      ;
11FB      ; CALLING SEQUENCE:
11FB      ;
11FB      ;       CALL    READ_V_BUS
11FB      ;
11FB      ;       THE ROUTINE IS CALLED BY THE FOLLOWING ROUTINES:
11FB      ;
11FB      ;               SHOW_ACTION
11FB      ;               COMPARE_VBUS
11FB      ;
11FB      ; IMPLICIT INPUTS:
11FB      ;
11FB      ;       NONE
11FB      ;
11FB      ; IMPLICIT OUTPUTS:
11FB      ;
11FB      ;       V_BUS_BUFFER - CONTAINS THE CURRENT STATE OF THE V BUS.
11FB      ;
11FB      ;--
11FB
11FB      READ_V_BUS:
11FB              MVI             B,40                    ; SET A LOOP COUNT
11FD              LXI             H,V_BUS_BUFFER          ; GET ADDRESS OF BUFFER
1200      1$:     LDA             STATUS_REG              ; READ THE V BUS
1203              ANI             VBUS_SERIAL_OUT         ; DISCARD UPPER BITS
1205              MOV             M,A                     ; PUT IN BUFFER
1206              INX             H                       ; INCREMENT BUFFER ADDRESS
1207              MVI             A,VBUS_CLOCK            ; GET DATA TO CLOCK THE V BUS
1209              STA             ADDR_MATCH_HI           ; CLOCK THE BUS
120C              XRA             A                       ; ...
120D              STA             ADDR_MATCH_HI           ; ...
1210              DCR             B                       ; DECREMENT THE LOOP COUNT
1211              JNZ             1$                      ; BRANCH IF NOT DONE
1214      ;
1214      ; RESTORE THE MATCH REGISTER TO THE SOMM ADDRESS
1214      ;
1214              LDA             SOMM_ADDRESS+1          ; GET UPPER ORDER SOMM ADDRESS
1217              STA             ADDR_MATCH_HI           ; RESTORE THE MATCH REGISTER
121A              RET                                     ; EXIT
121B
```

```
121B                    .SBTTL  "  DIRECTORY SEARCH ROUTINE
121B            ;++
121B            ;
121B            ; FUNCTIONAL DESCRIPTION:
121B            ;
121B            ;       THIS ROUTINE IS USED TO FIND THE STARTING RECORD NUMBER OF AN
121B            ;       OVERLAY. THE ROUTINE MUST BE CALLED WITH THE STARTING RECORD
121B            ;       NUMBER OF THE FILE AND THE SPECIFIED TEST NUMBER IF THE LOST
121B            ;       FLAG IS SET. THE ROUTINE RETURNS THE STARTING RECORD NUMBER
121B            ;       OF THE SPECIFIED TEST OVERLAY.
121B            ;
121B            ;       THE ROUTINE ALSO CHECKS THAT THE FILE ID CODE MATCHES THE
121B            ;       CURRENT FILE_NAM_INDEX. IF IT DOES, THE MODULE NAME IS TYPED
121B            ;       AND THE VERSION NUMBER OF THE FILE.
121B            ;       IF IT DOES NOT MATCH, A FAILURE RETURN IS MADE WITH -1 IN THE
121B            ;       A REGISTER.
121B            ;
121B            ;       THE DIRECTORY FORMAT IS AS FOLLOWS:
121B            ;
121B            ;               .BYTE   RECORD LENGTH OF DIRECTORY
121B            ;               .BYTE   TOTAL NUMBER OF TESTS IN FILE
121B            ;               .BYTE   NUMBER OF CHARACTERS IN VERSION NUMBER
121B            ;               .BLKB   ASCII VERSION NUMBER
121B            ;               .BYTE   FILE ID CODE
121B            ;               .BYTE   RECORD LENGTH OF TEST 1
121B            ;               .BYTE   RECORD LENGTH OF TEST 2
121B            ;                 .
121B            ;                 .
121B            ;               .BYTE   RECORD LENGTH OF TEST N
121B            ;               .BYTE   FF
121B            ;               .BYTE   FF
121B            ;                 .
121B            ;                 .
121B            ;               FILLED TO END OF RECORD
121B            ;
121B            ; CALLING SEQUENCE:
121B            ;
121B            ;       CALL    DIRECT_SEARCH
121B            ;
121B            ;       THE ROUTINE IS CALLED BY THE 'PROGRAM_INIT' ROUTINE
121B            ;
121B            ;       THE ROUTINE CALLS THE FOLLOWING ROUTINES:
121B            ;
121B            ;               READ_TU58
121B            ;
121B            ; IMPLICIT INPUTS:
121B            ;
121B            ;       FILE_NAM_INDEX - CONTAINS THE FILE NAME CODE
121B            ;       CURRENT_REC_NMB - CONTAINS THE STARTING RECORD NUMBER OF THE FILE
121B            ;
121B            ; IMPLICIT OUTPUTS:
121B            ;
121B            ;       CURRENT_REC_NMB - CONTAINS THE STARTING RECORD NUMBER OF THE TEST OVERLAY
121B            ;       TOTAL_NMB_TESTS - CONTAINS THE NUMBER OF TESTS IN THE FILE
121B            ;
```

```
121B  ; COMPLETION CODES:
121B  ;
121B  ;       C BIT SET   - TEST NUMBER IS OUT OF RANGE OR FILE_NAM_INDEX DOES
121B  ;                     NOT MATCH FILE ID. A REGISTER CONTIANS 0 FOR TEST
121B  ;                     NUMBER OUT OF RANGE, -1 FOR FILE NAME CHECK.
121B  ;       C BIT CLEAR - NO ERROR
121B  ;--
121B
121B  DIRECT_SEARCH:
121B          LHLD            CURRENT_REC_NMB ; GET THE STARTING RECORD NUMBER
121E          PUSH            H               ; SAVE
121F          READ_RECORD     TEST_BUFFER,1   ; READ THE FIRST RECORD OF THE DIRECTORY
1227          INX             H               ; POINT AT FILE ID CODE
1228          INX             H               ; ...
1229          LDA             FILE_NAM_INDEX  ; GET THE CURRENT FILE NAME CODE
122C          CMP             M               ; FILE ID OK?
122D          DCX             H               ; BACKUP DIRECTORY POINTER
122E          DCX             H               ; ...
122F          JZ              10$             ; BRANCH IF FILE ID OK
1232  ;
1232  ; FILE NAME INDEX NOT THE SAME AS FILE ID CODE. RETURN WITH
1232  ; -1 IN A REGISTER
1232  ;
1232          POP             H               ; CLEANUP THE STACK
1233          MVI             A,-1            ; SET RETURN CODE
1235          STC
1236          RET
1237
1237  ;
1237  ; TYPE THE MODULE NAME
1237  ;
1237  10$:    PUSH            H               ; SAVE DIRECTORY POINTER
1238          LXI             H,MODULE_NAM_TBL ; GET ADDRESS OF MODULE NAME TABLE
123B          RLC                             ; MAKE FILE NAME INDEX A 'WORD' INDEX
123C          MOV             E,A             ; PUT IN D & E
123D          MVI             D,0
123F          DAD             D               ; INDEX THE H & L REG'S
1240          MOV             E,M             ; GET LOW ADDRESS OF MODULE NAME
1241          INX             H               ; STRING
1242          MOV             D,M             ; AND THE HI ADDRESS
1243          XCHG                            ; PUT IN H & L
1244          TYPE                            ; TYPE THE MODULE NAME
124A          TYPE            HYPHEN          ; TYPE TWO SPACES
1253  ;
1253  ; NOW TYPE THE FILE VERSION NUMBER
1253  ;
1253          POP             H               ; GET THE DIRECTORY POINTER
1254          MOV             A,M             ; GET THE DIRECTORY RECORD SIZE
1255          PUSH            $PSW            ; SAVE
1256          INX             H               ; POINT AT TOTAL TESTS
1257          MOV             A,M             ; GET TOTAL NUMBER OF TESTS
1258          STA             TOTAL_NMB_TESTS ; SAVE
125B          INX             H               ; POINT AT VERSION NUMBER
125C          INX             H               ; ...
125D          TYPE                            ; TYPE THE VERSION NUMBER
```

```
1263            TYPE            CRLF
126C    ;
126C    ; CHECK IF MORE DIRECTORY TO READ
126C    ;
126C            POP             $PSW                    ; GET DIRECTORY LENGTH
126D            PUSH            $PSW                    ; SAVE
126E            DCR             A                       ; ACCOUNT FOR RECORD JUST READ
126F            JZ              1$                      ; BRANCH IF ALL DIRECTORY IS HERE
1272    ;
1272    ; READ THE REST OF THE DIRECTORY
1272    ;
1272            READ_RECORD     TEST_BUFFER+128 ; READ THE REST OF THE DIRECTORY
1278    ;
1278    ; CHECK IF THE LOST FLAG IS SET
1278    ;
1278    1$:     LDA             SOFT_CTRL_REG   ; GET THE LOST FLAG
127B            ANI             LOST_FLAG       ; IS IT SET?
127D            JNZ             3$              ; BRANCH IF YES
1280    ;
1280    ; SPECIAL TEST IS NOT SPECIFIED SO JUST GET RECORD NUMBER OF FIRST TEST
1280    ;
1280            MVI             D,0             ; SET THE RECORD NUMBER OF THE
1282            MVI             E,0             ; TEST TO ZERO
1284            JMP             5$              ; GO CALCULATE PHYSICAL RECOD NUMBER
1287    ;
1287    ; SEARCH THE DIRECTORY FOR THE RELATIVE RECORD NUMBER OF THE SPECIFIED TEST.
1287    ;
1287    3$:     LDA             SPEC_TEST_NUMB  ; GET THE SPECIFIED TEST NUMBER
128A            MOV             B,A             ; SAVE AS A LOOP COUNT
128B            LXI             H,TEST_BUFFER+10; GET ADDRESS OF DIRECTORY
128E            LXI             D,0             ; INIT THE RECORD NUMBER
1291    4$:     DCR             B               ; FOUND THE TEST YET?
1292            JZ              8$              ; BRANCH IF YES
1295            PUSH            H               ; SAVE THE REGISTERS
1296            PUSH            D               ; ...
1297            MOV             A,M             ; GET LOW BYTE OF TEST LENGTH
1298            ORA             A               ; CHECK IF END OF FILE
1299            JM              7$              ; BRANCH IF END OF FILE
129C            MOV             E,A             ; PUT TEST LENGTH IN D & E
129D            MVI             D,0             ; ...
129F            POP             H               ; GET ACCUMULATING RECORD NUMBER
12A0            DAD             D               ; ADD LENGTH OF THIS TEST
12A1            XCHG                            ; PUT IN D & E
12A2            POP             H               ; GET DIRECTORY ADDRESS
12A3            INX             H               ; POINT AT NEXT TEST ENTRY
12A4            JMP             4$              ; CHECK IF DONE
12A7    ;
12A7    ; CHECK IF THIS TEST IS IN THE FILE
12A7    ;
12A7    8$:     MOV             A,M             ; GET THE DIRECTORY ENTRY FOR THIS
12A8            ORA             A               ; TEST
12A9            JP              5$              ; BRANCH IF TEST EXISTS
12AC            JMP             9$              ; TEST DOES NOT EXIST
12AF    ;
12AF    ; FOUND THE END OF FILE BEFORE THE SPECIFIED TEST
```

```
12AF  3705 ;
12AF  3706 7$:     POP         H                 ; CLEANUP THE STACK
12B0  3707         POP         H                 ; ...
12B1  3708
12B1  3709 9$:     POP         $PSW              ; REMOVE SAVED DIRECTORY LENGTH FROM STACK
12B2  3710         POP         H                 ; REMOVE SAVED RECORD NUMBER FOM STACK
12B3  3711         XRA         A                 ; SET RETURN CODE
12B4  3712         STC                           ; RETURN WITH C BIT SET
12B5  3713         RET                           ; ...
12B6  3714 ;
12B6  3715 ; CALCULATE THE STARTING RECORD NUMBER OF THE TEST OVERLAY
12B6  3716 ;
12B6  3717 5$:     POP         $PSW              ; GET DIRECTORY LENGTH
12B7  3718         MOV         L,A               ; ...
12B8  3719         MVI         H,0               ; ...
12BA  3720         DAD         D                 ; ADD THE RECORD NUMBER OF THE TEST
12BB  3721         XCHG
12BC  3722         POP         H                 ; GET THE RECORD NUMBER OF THE FILE
12BD  3723         DAD         D                 ; ADD THE RELATIVE TEST RECORD NO.
12BE  3724         SHLD        CURRENT_REC_NMB   ; SAVE AS CURRENT RECORD NUMBER
12C1  3725         STC
12C2  3726         CMC
12C3  3727         RET                           ; EXIT WITH C BIT CLEAR
12C4  3728
```

```
12C4           .SBTTL  "  TYPE TEST NUMBER ROUTINE
12C4     ;++
12C4     ;
12C4     ; FUNCTIONAL DESCRIPTION:
12C4     ;
12C4     ;       THIS ROUTINE IS USED TO TYPE THE TEST NUMBER OF THE TERMINAL. IT
12C4     ;       IS CALLED BY THE READ OVERLAY ROUTINE.
12C4     ;
12C4     ;       IF IT'S THE FIRST CALL, THE LOST FLAG IS TESTED AND IF SET, THE
12C4     ;       TEST NUMBER IS OBTAINED FROM THE USER SPECIFIED TEST NUMBER. IN ALL
12C4     ;       OTHER CASES, THE TEST NUMBER IS OBTAINED FROM LOCATION 'TEST_NUMBER'.
12C4     ;
12C4     ;       IF THE TRACE (TR) FLAG IS SET, THE ENTIRE TEST TITLE IS PRINTED.
12C4     ;
12C4     ; CALLING SEQUNCE:
12C4     ;
12C4     ;       CALL    TYPE_TEST_NUMB
12C4     ;
12C4     ;       THE ROUTINE IS CALLED BY THE 'READ_OVERLAY' ROUTINE.
12C4     ;
12C4     ;       THE ROUTINE CALLS THE FOLLOWING ROUTINES:
12C4     ;
12C4     ;               TYPE_ASCII (TYPE)
12C4     ;               TYPE_NUMERIC (TYPEB)
12C4     ;
12C4     ; IMPLICIT INPUTS:
12C4     ;
12C4     ;       TESTS_PER_LINE  - CONTAINS THE NUMBER OF TEST NUMBERS TYPED ON THE
12C4     ;                         CURRET LINE.
12C4     ;       TEST_NUMBER     - CONTAINS THE NUMBER (MINUS ONE) OF THE TEST THAT
12C4     ;                         WAS JUST LOADED EXCEPT WHEN THE LOST FLAG IS SET.
12C4     ;       SPEC_TEST_NUMB  - CONTAINS THE FIRST TEST NUMBER THAT GETS EXECUTED
12C4     ;                         IF THE LOST FLAG IS SET.
12C4     ;
12C4     ; EXPLICIT INPUTS:
12C4     ;
12C4     ;       B - NON ZERO MEANS DON'T TYPE TEST NUMBER
12C4     ;
12C4     ; IMPLICIT OUTPUTS:
12C4     ;
12C4     ;       TESTS_PER_LINE  - SAME AS ABOVE
12C4     ;
12C4     ;--
12C4
12C4     TYPE_TEST_NUMB:
12C4     ;
12C4     ; CHECK IF THIS IS THE INITIAL CALL. IF SO, CHECK THE LOST FLAG. IF IT'S
12C4     ; SET, GET THE TEST NUMBER FROM THE USER SPECIFIED TEST NUMBER.
12C4     ;
12C4            LDA             TYPE_TEST_FLAG  ; GET THE FLAG
12C7            ORA             A               ; SET THE CONDITION CODES
12C8            JNZ             10$             ; BRANCH IF NOT INITIAL CALL
12CB     ;
12CB     ; INITIAL CALL. CHECK IF THE LOST FLAG IS SET.
12CB     ;
```

```
12CB             LDA             SOFT_CTRL_REG    ; GET THE LOST FLAG
12CE             ANI             LOST_FLAG        ; IS IT SET?
12D0             JZ              1$               ; BRANCH IF NO
12D3     ;
12D3     ; LOST FLAG IS SET. GET THE TEST NUMBER FROM THE USER SPECIFIED NUMBER
12D3     ;
12D3             LDA             SPEC_TEST_NUMB   ; GET THE USER SPECIFIED TEST NUMBER
12D6             JMP             20$              ; GO TYPE IT
12D9     ;
12D9     ; LOST FLAG IS NOT SET. TYPE TEST NUMBER 1
12D9     ;
12D9     1$:     MVI             A,1              ; SET TEST NUMBER TO 1
12DB             JMP             20$              ; TYPE IT
12DE     10$:    LDA             TEST_NUMBER      ; GET THE LAST TEST NUMBER
12E1             INR             A                ; SET TO NEW TEST NUMBER
12E2     20$:    STA             ARG_LIST         ; SAVE TEST NUMBER TO TYPE
12E5             MOV             A,B              ; GET TYPEOUT FLAG
12E6             ORA             A                ; SET CONDITION CODES
12E7             JZ              21$              ; BRANCH IF GOING TO TYPE
12EA             LDA             TESTS_PER_LINE   ; SETUP TESTS PER LINE SO
12ED             DCR             A                ; IT DOSEN;T CHANGE
12EE             STA             TESTS_PER_LINE   ; INIT TESTS PER LINE
12F1             JMP             17$              ; EXIT
12F4     21$:    LDA             PROG_CTRL_REG    ; GET THE TRACE FLAG
12F7             ANI             TR_FLAG          ; ...
12F9             JZ              30$              ; BRANCH IF ITS CLEAR
12FC             TYPE            TEST_MSG_1       ; TYPE 'TEST '
1305             TYPEB           ARG_LIST         ; TYPE THE TEST NUMBER
1310             TYPE            TEST_MSG_2       ; TYPE ': '
1319             LHLD            INIT_TEST_PC     ; GET POINTER TO TEST TITLE
131C             SHLD            MESSAGE_POINTER  ; PASS TO TYPE ASCII ROUTINE
131F             CALL            TYPE_ASCII       ; TYPE THE TEST TITLE
1322             TYPE            CRLF             ; ...
132B     ;
132B     ; UPDATE THE INITIAL TEST PC AND COMPLIMENT INITIAL TEST PC
132B     ;
132B     17$:    LHLD            INIT_TEST_PC     ; GET POINTER TO TEST TITLE
132E             MOV             L,M              ; GET THE LENGTH OF THE STRING
132F             INR             L                ; COUNT THE COUNT BYTE
1330             MOV             C,L              ; SAVE IN C
1331             MVI             H,0              ; ...
1333             SHLD            TITLE_LENGTH     ; SAVE
1336             XCHG
1337             LHLD            INIT_TEST_PC
133A             DAD             D                ; ADD TO INITIAL TEST PC
133B             SHLD            INIT_TEST_PC     ; ...
133E             XRA             A                ; INIT A REG
133F             SUB             C                ; GET TWO'S COMPLIMENT OF TITLE
1340             MOV             C,A              ; LENGTH
1341             MVI             B,-1             ; ...
1343             LHLD            C_INIT_TEST_PC   ; GET COMPLIMENT INIT TEST PC
1346             DAD             B                ; ADJUST BY TITLE LENGTH
1347             SHLD            C_INIT_TEST_PC   ; ...
134A     ;
134A     ; INCREMENT THE TESTS PER LINE AND SET THE TYPE_TEST_FLAG
```

```
134A      ;
134A              LDA             TESTS_PER_LINE  ; GET NUMBER OF TESTS PER LINE
134D              INR             A               ; INCREMENT IT
134E              STA             TESTS_PER_LINE  ; SAVE NEW VALUE OF TESTS ON LINE
1351              MVI             A,1             ; SET THE TYPE FLAG
1353              STA             TYPE_TEST_FLAG  ; ...
1356              RET                             ; EXIT
1357      ;
1357      ; TYPE THE TEST NUMBER THAT IS IN THE ARG_LIST.
1357      ;
1357      30$:    LDA             TESTS_PER_LINE  ; GET NUMBER OF TEST NUMBERS TYPED
135A              CPI             20              ; TYPED 20 TESTS YET?
135C              JNZ             25$             ; BRANCH IF NO
135F      ;
135F      ; TIME TO TYPE A CARRIAGE RETURN LINE FEED
135F      ;
135F              TYPE            CRLF            ; TYPE CARRIAGE RETURN LINE FEED
1368              XRA             A               ; RESET THE COUNT
1369              STA             TESTS_PER_LINE  ; ...
136C      ;
136C      ; TYPE THE TEST NUMBER FOLLOWED BY A COMMA
136C      ;
136C      25$:    TYPEB           ARG_LIST        ; TYPE THE NUMBER
1377              TYPE            COMMA_MSG       ; TYPE A COMMA
1380              JMP             17$             ; ...
1383
```

```
1383                 .SBTTL  "  DESELECT DCS ROUTINE
1383      ;++
1383      ;
1383      ; FUNCTIONAL DESCRIPTION:
1383      ;
1383      ;          THIS ROUTINE DESELECTS DCS BY SETTING PARITY CHECK AND MICRO ADDRESS
1383      ;          INHIBIT AND LOADING THE MATCH REGISTER WITH 0
1383      ;
1383      ; CALLING_SEQUENCE:
1383      ;
1383      ;          CALL    DESELECT_DCS
1383      ;
1383      ; IMPLICIT INPUTS:
1383      ;
1383      ;          NONE
1383      ;
1383      ; IMPLICIT OUTPUTS:
1383      ;
1383      ;          NONE
1383      ;
1383      ;--
1383
1383      DESELECT_DCS:
1383                 LDA               DCS_CTRL_RE_CPY ; GET A COPY OF THE CONTROL REG
1386                 ORI               <PAR_CHK_ENABL!-; SET PARITY CHECK AND MICRO
1386                                   MICRO_ADDR_INH> ; ADDRESS INHIBIT
1388                 STA               DCS_CTRL_REG    ; LOAD THE CONTROL REGISTER
138B                 LXI               H,0             ; PUT ADDRESS 0 IN THE MATCH
138E                 SHLD              ADDR_MATCH_LO   ; REGISTER
1391                 RET
1392
```

```
1392          .SBTTL  "  SELECT DCS
1392  ;++
1392  ;
1392  ; FUNCTIONAL DESCRIPTION:
1392  ;
1392  ;       THIS ROUTINE RESTORES THE MATCH REGISTER TO THE SOMM ADDRESS AND
1392  ;       CLEARS THE PARITY CHECK AND MICRO ADDRESS INHIBIT BITS IN THE
1392  ;       CONTROL REGISTER.
1392  ;
1392  ; CALLING SEQUENCE:
1392  ;
1392  ;       CALL    SELECT_DCS
1392  ;
1392  ;--
1392
1392  SELECT_DCS:
1392          LDA             DCS_CTRL_RE_CPY ; GET A COPY OF THE CONTROL REG
1395          STA             DCS_CTRL_REG    ; LOAD THE CONTROL REGISTER
1398          LHLD            SOMM_ADDRESS    ; GET THE SOMM ADDRESS
139B          SHLD            ADDR_MATCH_LO   ; RESTORE THE MATCH REGISTER
139E          RET
139F
```

```
139F          .SBTTL  "  START DCS EXECUTION ROUTINE
139F    ;++
139F    ;
139F    ; FUNCTIONAL DESCRIPTION:
139F    ;
139F    ;         THIS ROUTINE LOADS THE CONTROL STORE LATCHES WITH THE DCS MICRO
139F    ;         WORD AT THE SPECIFIED DCS ADDRESS.
139F    ;
139F    ; CALLING SEQUENCE:
139F    ;
139F    ;         CALL    START_EXECUTION
139F    ;
139F    ; IMPLICIT INPUTS:
139F    ;
139F    ;         NONE
139F    ;
139F    ; INPUT PARAMETERS:
139F    ;
139F    ;         A REGISTER - CONTAINS THE DCS ADDRESS OF THE MICRO WORD
139F    ;
139F    ; OUTPUT PARAMETERS:
139F    ;
139F    ;         NONE
139F    ;
139F    ;--
139F
139F    START_EXECUTION:
139F    ;
139F    ; SET THE 'PARITY CHECK', 'MICRO ADDRESS INHIBIT', AND 'CLEAR CONTROL FILE'
139F    ; BITS IN THE DCS CONTROL REGISTER.
139F    ;
139F              STA             DCS_ADDR_REG_WO ; LOAD THE DCS ADDRESS REGISTER
13A2
13A2              LDA             DCS_CTRL_RE_CPY ; GET CONTENTS OF DCS CONTROL REG
13A5              ORI             <PAR_CHK_ENABL!-; SET THE PARITY CHECK AND MICRO
13A5                              MICRO_ADDR_INH!-; ADDRESS INHIBIT AND CLEAR THE
13A5                              CLEAR_CTRL_FILE>; CONTROL FILE
13A7              STA             DCS_CTRL_REG    ; LOAD THE CONTROL REGISTER
13AA              ANI             <^CCLEAR_CTRL_FILE>; DEASCERT CLEAR CONTROL FILE
13AC              STA             DCS_CTRL_REG    ; ...
13AF              STA             DCS_CTRL_RE_CPY ; KEEP OTHER BITS ASCERTED WHEN CLOCK
13B2                                              ; IS TICKED
13B2    ;
13B2    ; TICK THE CLOCK TO LATCH THE SPECIFIED MICRO WORD
13B2    ;
13B2              PUSH            H               ; SAVE H & L
13B3              MVI             L,<DCS_START&255> ; GET THE DCS BASE ADDRESS
13B5              MVI             H,<DCS_START/256> ; ...
13B7              SHLD            ADDR_MATCH_LO   ; LOAD THE ADDRESS MATCH REGISTER
13BA              SGL_MIC_INSTR   1               ; STEP THE CLOCK
13C2    ;
13C2    ; NOW DEASCERT 'PARITY CHECK' AND 'MICRO ADDRESS INHIBIT'
13C2    ;
13C2              LDA             DCS_CTRL_RE_CPY ; GET THE CONTROL REGISTER
13C5              ANI             <^C<PAR_CHK_ENABL!-; CLEAR PARITY CHECK ENABLE
```

```
    13C5  3977                          MICRO_ADDR_INH>>; AND MICRO ADDRESS INHIBIT
    13C7  3978         STA              DCS_CTRL_REG    ; LOAD THE CONTROL REGISTER
    13CA  3979         STA              DCS_CTRL_RE_CPY ; SAVE A COPY
    13CD  3980  ;
    13CD  3981  ; LOAD THE SOMM ADDRESS IN THE MATCH REGISTER
    13CD  3982  ;
    13CD  3983         LHLD             SOMM_ADDRESS    ; GET THE SOMM ADDRESS
    13D0  3984         SHLD             ADDR_MATCH_LO   ; LOAD THE MATCH REGISTER
    13D3  3985         POP              H               ; RESTORE H & L
    13D4  3986         RET                              ; EXIT
    13D5  3987
```

```
13D5               .SBTTL  "  CMI DISABLE ROUTINE
13D5       ;++
13D5       ;
13D5       ; FUNCTIONAL DESCRIPTION:
13D5       ;
13D5       ;       THIS ROUTINE LOADS AND EXECUTES MICRO CODE TO DISABLE THE CMI AND
13D5       ;       MEMORY MANAGEMENT.
13D5       ;
13D5       ; CALLING SEQUENCE:
13D5       ;
13D5       ;       CALL    DISABLE_CMI
13D5       ;
13D5       ;--
13D5
13D5       DISABLE_CMI:
13D5               PUSH            B               ; SAVE THE REGISTERS
13D6               PUSH            D               ; ...
13D7               PUSH            H               ; ...
13D8               PUSH            $PSW            ; ...
13D9               WRITE$_DCS      DCS_SCRATCH_ADR,-
13D9                               LOAD_MEMSCAR,7 ; LOAD THE MICRO CODE
13EC               WRITE$_DCS      DCS_SCRATCH_ADR+7,-
13EC                               NOP_MICRO_WORD,1; ...
13FF               EXECUTE         DCS_SCRATCH_ADR,7 ; EXECUTE THE MICRO CODE
1412               POP             $PSW            ; RESTORE THE REGISTERS
1413               POP             H               ; ...
1414               POP             D               ; ...
1415               POP             B               ; ...
1416               RET                             ; EXIT
1417
```

```
1417         .SBTTL  "  CMI ENABLE ROUTINE
1417   ;++
1417   ;
1417   ; FUNCTIONAL DESCRIPTION:
1417   ;
1417   ;      THIS ROUTINE LOADS AND EXECUTES MICRO CODE TO ENABLE THE CMI
1417   ;      BY EXECUTING A PROCESSOR INIT BUS FUNCTION.
1417   ;
1417   ; CALLING SEQUENCE:
1417   ;
1417   ;      CALL    ENABLE_CMI
1417   ;
1417   ;--
1417
1417   ENABLE_CMI:
1417           PUSH            B                ; SAVE THE REGISTERS
1418           PUSH            D                ; ...
1419           PUSH            H                ; ...
141A           PUSH            $PSW             ; ...
141B           WRITE$_DCS      DCS_SCRATCH_ADR,-
141B                           LOAD_MEMSCAR,1 ; LOAD THE MICRO CODE
142E           WRITE$_DCS      DCS_SCRATCH_ADR+1,-
142E                           NOP_MICRO_WORD,1; ...
1441           EXECUTE         DCS_SCRATCH_ADR,1; EXECUTE THE MICRO CODE
1454           POP             $PSW             ; RESTORE THE REGISTERS
1455           POP             H                ; ...
1456           POP             D                ; ...
1457           POP             B                ; ...
1458           RET                              ; EXIT
1459
```

```
1459            .SBTTL  "  INIT CPU ROUTINE
1459    ;++
1459    ;
1459    ; FUNCTIONAL DESCRIPTION:
1459    ;
1459    ;       THIS ROUTINE SETS THEN CLEARS DC LOW IN THE CPU AND THEN FETCHES
1459    ;       LOCATION 0 IN THE CPU CONTROL STORE.
1459    ;
1459    ; CALLING SEQUENCE:
1459    ;
1459    ;       CALL    CPU_INIT
1459    ;
1459    ;--
1459
1459    CPU_INIT:
1459            MVI             A,<MAINT_TRAP_OFF!-
1459                            MAINT_DCS_ENABL>; GET DATA TO INIT THE MAINTENANCE REG
145B            STA             MAINT_REG       ; INIT THE MAINTENANCE REG
145E            LDA             SOFT_CTRL_REGA  ; GET THE SOMM FLAG
1461            ANI             SOMM_FLAG       ; ...
1463            ORI             <CLK_CTRL_0!-   ; GET DATA TO INIT THE DCS CONTROL REG
1463                            CLK_CTRL_1>     ; ...
1465            STA             DCS_CTRL_REG    ; INIT THE CONTROL REG
1468            STA             DCS_CTRL_RE_CPY ; SAVE A COPY
146B            LDA             FRONT_PNL_2     ; GET REMOTE, FAULT, AND TEST BITS
146E            ANI             <REMOTE!FAULT!TEST>; ...
1470            ORI             DC_LOW          ; SET THE DC LOW BIT
1472            STA             RD_CTRL_REG     ; ...
1475            XRI             DC_LOW          ; CLEAR DC LOW BIT
1477            STA             RD_CTRL_REG     ; ...
147A            RET
147B
```

```
147B            .SBTTL  "  FETCH CS LOCATION 0
147B    ;++
147B    ;
147B    ; FUNCTIONAL DESCRIPTION:
147B    ;
147B    ;       THIS ROUTINE FETCHES LOCATION 0 OF THE CPU CONTROL STORE
147B    ;
147B    ;--
147B
147B    FETCH_0:
147B            CALL            DESELECT_DCS    ; FORCE CS ADDRESS ZERO
147E            LDA             DCS_CTRL_RE_CPY ; SET PARITY CHECK AND UADDR INH
1481            ORI             <PAR_CHK_ENABL!-; ...
1481                            MICRO_ADDR_INH> ; ...
1483            STA             DCS_CTRL_RE_CPY ; ...
1486            SGL_MIC_INSTR   1               ; TICK THE CLOCK
148E            LDA             DCS_CTRL_RE_CPY ; DEASCERT PARITY CHECK AND MICRO
1491            ANI             <^C<PAR_CHK_ENABL!- ; ADDRESS INHIBIT
1491                            MICRO_ADDR_INH>>; ...
1493            STA             DCS_CTRL_RE_CPY ; ...
1496            STA             DCS_CTRL_REG    ; ...
1499            RET
149A
```

```
149A          .SBTTL  "  TYPE FLAGS ROUTINE
149A    ;++
149A    ;
149A    ; FUNCTIONAL DESCRIPTION:
149A    ;
149A    ;       THIS ROUTINE IS CALLED WITH THE PROGRAM CONTROL REGISTER BIT MASK
149A    ;       IN THE A REGISTER. IT CONVERTS THE BITS INTO THE FLAG MNEMONICS
149A    ;       AND TYPES THEM.
149A    ;--
149A
149A    TYPE_FLAGS:
149A            ORA             A                       ; ANY FLAGS TO TYPE?
149B            JZ              20$                     ; BRANCH IF NO
149E            MVI             B,8                     ; SET THE LOOP COUNT FOR 7 FLAGS
14A0            LXI             H,HALT_FLAG_NAME; GET ADDRESS OF ASCII STRING
14A3    1$:     RAR                                     ; SEE IF BIT IS SET
14A4            JNC             10$                     ; BRANCH IF CLEAR
14A7            PUSH            B                       ; SAVE PARAMETERS
14A8            PUSH            H                       ; ...
14A9            PUSH            $PSW                    ; ...
14AA            TYPE_UNCOUNTED  2                       ; TYPE THE FLAG NAME
14B5            TYPE            COMMA_MSG               ; TYPE A COMMA
14BE            POP             $PSW                    ; RESTORE THE PARAMTERS
14BF            POP             H                       ; ...
14C0            POP             B                       ; ...
14C1    10$:    INX             H                       ; POINT AT NEXT FLAG NAME
14C2            INX             H                       ; ...
14C3            DCR             B                       ; CHECKED ALL 7 BITS?
14C4            JNZ             1$                      ; BRANCH IF NO
14C7    20$:    TYPE            CRLF                    ; TERMINATE THE LINE
14D0            RET                                     ; EXIT
```

```
14D1                  .SBTTL  "  FILL DCS ROUTINE
14D1  ;++
14D1  ;
14D1  ; FUNCTIONAL DESCRIPTION:
14D1  ;
14D1  ;               THIS ROUTINE IS USED TO FILL DCS WITH 'NOP' MICRO INSTRUCTIONS.
14D1  ;
14D1  ; CALLING SEQUENCE:
14D1  ;
14D1  ;               CALL    FILL_DCS
14D1  ;
14D1  ;--
14D1  FILL_DCS:
14D1                  MVI             B,64              ; SET THE LOOP COUNT
14D3                  MVI             C,0               ; INIT THE DCS ADDRESS TO LOAD AT
14D5  1$:             PUSH            B                 ; SAVE THE LOOP COUNT
14D6                  MOV             A,C               ; PUT DCS ADDRESS IN A REG
14D7                  WRITE$_DCS      ,NOP_MICRO_WORD,1 ; LOAD A MICRO WORD
14E8                  POP             B                 ; GET THE LOOP COUNT
14E9                  INR             C                 ; INCREMENT THE DCS ADDRESS
14EA                  DCR             B                 ; LOADED 64 WORDS YET?
14EB                  JNZ             1$                ; BRANCH IF NO
14EE                  RET                               ; EXIT
```

```
14EF                .SBTTL  "COMMAND PARSER
14EF                .SBTTL  "  GET COMMAND LINE
14EF      ;++
14EF      ;
14EF      ; FUNCTIONAL DESCRIPTION:
14EF      ;
14EF      ;         THIS ROUTINE INITIALIZES THE CONTROL C FLAG, THE TERMINAL DRIVER,
14EF      ;         TYPES THE DIAGNOSTIC PROMPT, AND CALLS THE RDM TERMINAL DRIVER
14EF      ;         TO PERFORM A READ. WHEN THE DRIVER RETURNS, ERRORS ARE CHECKED.
14EF      ;         IF A CONTROL C WAS TYPED, THE ROUTINE RESTARTS. IF ANY OTHER ERROR
14EF      ;         WAS DETECTED, AN ERROR MESSAGE IS TYPED WITH THE ERROR CODE AND THE
14EF      ;         ROUTINE RESTARTS. IF NO ERROR WAS DETECTED, THE PARSE ROUTINE IS
14EF      ;         CALLED.
14EF      ;
14EF      ;         IF THE PARSE ROUTINE DOES NOT FIND ANY ERRORS IN THE COMMAND LINE,
14EF      ;         THE APPROPRIATE COMMAND ACTION ROUTINE IS CALLED.
14EF      ;
14EF      ;         IF THE COMMAND WAS A 'CONTINUE' COMMAND, RETURN IS MADE TO THE
14EF      ;         CALLING SEQUENCE. IF THE COMMAND WAS A 'DIAGNOSE' COMMAND, RETURN
14EF      ;         IS MADE TO THE 'INTERPRETER'. ANY OTHER COMMAND (EXCEPT 'RETURN')
14EF      ;         CAUSES THIS ROUTINE TO RESTART.
14EF      ;
14EF      ; CALLING SEQUENCE:
14EF      ;
14EF      ;         CALL    GET_CMD_LINE
14EF      ;
14EF      ; IMPLICIT INPUTS:
14EF      ;
14EF      ;         SOFT_CTRL_REG - CONTAINS THE SOFTWARE CONTROL FLAGS.
14EF      ;
14EF      ; IMPLICIT OUTPUTS:
14EF      ;
14EF      ;         REFER TO THE INDIVIDUAL COMMAND EXECUTION ROUTINES.
14EF      ;
14EF      ;--
14EF
14EF      GET_CMD_LINE:
14EF              LDA             SOFT_CTRL_REG   ; GET THE SWITCH REGISTER
14F2              ANI             <^C<CONTROL_C_FLAG!- ; CLEAR SOME FLAGS
14F2                              DIAGNOSE_FLAG!CONTINUE_FLAG>> ; ...
14F4              STA             SOFT_CTRL_REG   ; ...
14F7              XRA             A               ; INITIALIZE THE BELL COUNT
14F8              STA             BELL_COUNT      ; FOR THE RING _BELL ROUTINE
14FB              STA             TESTS_PER_LINE  ; ...
14FE      1$:     $TERM_INIT                      ; INIT THE TERMINAL DRIVER
1503              TYPE            CRLF            ; TYPE A CARRIAGE RETURN LINE FEED
150C              TYPE            PROMPT          ; TYPE THE PROMPT
1515              $TERM_READ      TERM_INP_BUFF,- ; GET A COMMAND LINE
1515                              CMD_LINE_LENGTH
1520              JNC             2$              ; BRANCH IF NO ERROR
1523              CPI             $CONTROL_C_CODE ; WAS IT A CONTROL C?
1525              JZ              1$              ; BRANCH IF YES
1528              STA             TERM_ERR_CODE   ; SAVE THE ERROR CODE
152B              TYPE            CRLF            ; TYPE A CARRIAGE RETURN LINE FEED
1534              TYPE            TERM_ERROR      ; TYPE THE TERMINAL ERROR MSG
```

```
153D           TYPEB          TERM_ERR_CODE    ; TYPE THE ERROR CODE
1548           JMP            1$               ; RESTART THIS ROUTINE
154B
154B      ;
154B      ; GOT A COMMAND LINE. CALL THE PARSER.
154B      ;
154B
154B      2$:  LDA            TERM_INP_BUFF+1  ; GET FIRST CHARACTER
154E           CPI            <^XD>            ; IS IT A CARRIAGE RETURN?
1550           JZ             1$               ; BRANCH IF  YES
1553           CALL           COMMAND_PARSE    ; PARSE THE COMMAND LINE
1556           JNC            3$               ; BRANCH IF NO ERROR
1559      5$:  TYPE           INVALID_COMMAND  ; TYPE AN ERROR MESSAGE
1562           JMP            1$               ; RESTART THIS ROUTINE
1565      ;
1565      ; COMMAND PARSE WAS SUCCESSFUL. CALL THE COMMAND ACTION ROUTINE.
1565      ;
1565
1565      3$:  LDA            PARSE_TABLE      ; GET THE COMMAND CODE IN 'A' REG
1568           ANI            <^X7F>           ; THROW AWAY KEYWORD FLAG BIT
156A           MOV            L,A              ; SAVE IN 'L' REG
156B           MVI            H,0              ; CLEAR THE 'H' REG
156D           LXI            B,CMD_DISPATCH   ; GET ADDRESS OF COMMAND DISPATCH
1570                                           ; TABLE
1570           DAD            B                ; GET POINTER TO COMMAND ACTION
1571                                           ; ADDRESS IN H & L
1571           XCHG                            ; PUT IN D & E
1572           LDAX           D                ; GET LOW PART OF COMMAND ACTION RTN
1573           MOV            L,A              ; ADDRESS AND PUT IN 'L' REG
1574           INX            D
1575           LDAX           D                ; GET HIGH PART AND PUT IN
1576           MOV            H,A              ; H REGISTER
1577           LXI            B,4$             ; GET A RETURN ADDRESS FOR THE
157A                                           ; SIMULATED CALL THRU H AND L REG'S
157A           PUSH           B                ; PUT IT ON THE STACK
157B           LXI            D,PARSE_TABLE+1  ; GET ADDRESS OF PARSE TABLE
157E           PCHL                            ; GO TO THE COMMAND ACTION ROUTINE
157F      ;
157F      ; COMMAND ACTION ROUTINES WILL RETURN HERE.
157F      ;
157F      4$:  JC             5$               ; BRANCH IF ERROR ON COMMAND EXEC.
1582      ;
1582      ; SEE WHAT KIND OF COMMAND IT WAS. IF 'CONTINUE', RETURN TO CALLING
1582      ; SEQUENCE. IF 'DIAGNOSE', RETURN TO THE 'PROGRAM INIT'. ELSE RESTART
1582      ; THIS ROUTINE.
1582      ;
1582           LXI            H,SOFT_CTRL_REG  ; GET ADDRESS OF SOFTWARE CTRL REG
1585           MOV            A,M              ; GET CONTROL FLAGS
1586           ANI            CONTINUE_FLAG    ; CONTINUE FLAG SET?
1588           JNZ            7$               ; BRANCH IF YES
158B      6$:  MOV            A,M              ; GET CONTROL FLAGS AGAIN
158C           ANI            DIAGNOSE_FLAG    ; DIAGNOSE FLAG SET?
158E           JZ             1$               ; BRANCH IF NO
1591           LXI            H,PROGRAM_INIT   ; GET ADDRESS OF INIT ROUTINE
1594           XTHL                            ; PUT IT ON THE STACK
```

```
1595  4274 7$:      $TERM_READ      TERM_INP_BUFF,- ; QUE A TERMINAL READ REQUEST
1595  4275                          ONE,TERM_INTERRUPT ; ...
15A0  4276          RET                             ; GO TO APPROPRIATE ROUTINE
15A1  4277
15A1  4278
15A1  4279 BREAK_ACTION:
15A1  4280          CALL            CMD_ENTRY
15A4  4281          POP             H               ; DICARD RETURN PC
15A5  4282          JMP             GET_CMD_LINE
15A8  4283
15A8  4284
```

```
15A8                    .SBTTL  "  PARSE COMMAND LINE
15A8            ;++
15A8            ;
15A8            ; FUNCTIONAL DESCRIPTION:
15A8            ;
15A8            ;       THIS ROUTINE IS RESPONSIBLE FOR PARSING COMMAND LINES INTO
15A8            ;       NUMERIC CODES THAT ARE PLACED IN THE PARSE TABLE.
15A8            ;
15A8            ; IMPLICIT INPUTS:
15A8            ;
15A8            ;       TERM_INP_BUFF - BUFFER THAT CONTAINS THE ASCII COMMAND LINE
15A8            ;
15A8            ; IMPLICIT OUTPUTS:
15A8            ;
15A8            ;       PARSE_TABLE   - CONTAINS THE NUMERIC CODES OF THE PARSED COMMAND LINE
15A8            ;
15A8            ; COMPLETION CODES:
15A8            ;       C BIT SET   - THE COMMAND LINE DID NOT PARSE
15A8            ;       C BIT CLEAR - THE PARSE WAS SUCCESSFUL
15A8            ;
15A8            ;--
15A8
15A8            COMMAND_PARSE:
15A8                    LXI             H,PARSE_TABLE   ; GET THE ADDRESS OF THE PARSE TABLE
15AB                    SHLD            PARSE_TBL_PTR   ; INIT THE PARSE TABLE POINTER
15AE                    LXI             H,TERM_INP_BUFF ; GET THE ADDRESS OF THE INPUT STRING
15B1                    INX             H               ; POINT AT FIRST CHARACTER
15B2                    LXI             B,COMMAND_LIST  ; GET THE ADDRESS OF THE COMMAND LIST
15B5            ;
15B5            ; NOW SEARCH THE COMMAND LIST FOR A MATCH
15B5            ;
15B5                    CALL            SEARCH_LIST     ; SEARCH THE COMMAND LIST
15B8                    JC              PARSE_ERR       ; IF NOT IN LIST, EXIT WITH C SET
15BB                    MOV             A,D             ; GET THE CODE
15BC                    ORI             <^X80>          ; INSERT FLAG BIT
15BE                    MOV             D,A             ; PUT BACK IN D REG
15BF                    CALL            PUT_D_IN_TBL    ; PUT COMMAND CODE IN PARSE TABLE
15C2            ;
15C2            ; THROW AWAY SPACES
15C2            ;
15C2                    CALL            SKIP_SPACES     ; BUMP STRING PTR PAST SPACES
15C5                    JNC             1$              ; BRANCH IF CARRIAGE RETURN NOT FOUND
15C8            2$:     MVI             D,<^X55>        ; PUT PARSE TABLE TERMINATOR IN THE
15CA                    CALL            PUT_D_IN_TBL    ; PARSE TABLE
15CD                    JMP             PARSE_DONE      ; EXIT WITH C BIT CLEAR
15D0            ;
15D0            ; NOW SEARCH THE KEYWORD LIST FOR A MATCH
15D0            ;
15D0            1$:     LXI             B,KEYWORD_LIST  ; GET ADDRESS OF LIST
15D3                    CALL            SEARCH_LIST     ; SEARCH THE KEYWORD LIST
15D6                    JC              PARSE_ERR       ; IF NOT IN LIST, EXIT WITH C SET
15D9                    MOV             A,D             ; GET THE CODE
15DA                    ORI             <^X80>          ; INSERT KEYWORD CODE BIT
15DC                    MOV             D,A             ; RETURN TO 'D' REG
15DD                    CALL            PUT_D_IN_TBL    ; PUT THE KEYWORD CODE IN PARSE TABLE
```

```
15E0  ;
15E0  ; THROW AWAY SPACES
15E0  ;
15E0          MOV         A,M              ; GET THE NEXT CHARACTER IN STRING
15E1          CPI         <^X3A>           ; IS CHARACTER A ':' ?
15E3          JZ          4$               ; BRANCH IF YES
15E6          CALL        SKIP_SPACES      ; BUMP STRING PTR PAST SPACES
15E9          JC          2$               ; EXIT IF CARRIAGE RETURN FOUND
15EC          MOV         A,M              ; GET THE NEXT CHARACTER IN STRING
15ED          CPI         <^X3A>           ; IS CHARACTER A ':' ?
15EF          JNZ         1$               ; IF NOT, SEARCH FOR A KEYWORD
15F2  ;
15F2  ; FOUND THE START OF A COLON LIST. CHECK IF A NUMBER FOLLOWS.
15F2  ;
15F2  4$:     INX         H                ; BUMP STRING PTR PAST THE ':'
15F3          MOV         A,M              ; GET THE CHARACTER
15F4          CPI         <^X20>           ; IS IT A SPACE?
15F6          JNZ         3$               ; BRANCH IF NO
15F9          CALL        SKIP_SPACES      ; SKIP OVER ANY SPACES
15FC          JC          2$               ; EXIT IF CARRIAGE RETURN FOUND
15FF  3$:     CPI         <^XD>            ; IS IT A CARRIAGE RETURN?
1601          JZ          PARSE_ERR        ; BRANCH IF YES
1604          CALL        PARSE_HEX_NUMB   ; PARSE THE HEX NUMBER
1607          JC          PARSE_ERR        ; IF ILLEGAL NO., EXIT WITH C SET
160A  ;
160A  ; THROW AWAY SPACES
160A  ;
160A  5$:     CALL        SKIP_SPACES      ; BUMP STRING POINTER PAST SPACES
160D          JC          2$               ; EXIT IF CARRIAGE RETURN FOUND
1610  ;
1610  ; NEXT LEXIME COULD BE ANOTHER NUMERIC OR ANOTHER KEYWORD. USE PARSE_HEX_NUMB
1610  ; ROUTINE TO SEE IF IT'S A NUMERIC. IF NOT NUMERIC, RESTORE INPUT BUFFER
1610  ; POINTER AND SEE IF IT'S A KEYWORD.
1610  ;
1610          PUSH        H                ; SAVE INPUT STRING POINTER
1611          CALL        PARSE_HEX_NUMB   ; SEE IF LEXIME IS NUMERIC
1614          POP         D                ; GET SAVED H AND L BACK
1615          JNC         5$               ; BRANCH IF IT WAS A NUMBER
1618          XCHG                         ; RESTORE INPUT STRING POINTER
1619          JMP         1$               ; SEE IF IT'S A KEYWORD
161C  ;
161C  ; PARSE ERROR. RETURN TO CALLER WITH C BIT SET
161C  ;
161C  PARSE_ERR:
161C          STC                          ; SET THE CARRY BIT
161D          RET                          ; RETURN TO CALLER
161E  ;
161E  ; PARSE DONE. EXIT WITH C BIT CLEAR
161E  ;
161E  PARSE_DONE:
161E          STC                          ; SET THE CARRY BIT
161F          CMC                          ; CLEAR THE CARRY BIT
1620          RET                          ; RETURN TO CALLER
1621
1621
```

```
1621              .SBTTL  "  SEARCH COMMAND/KEYWORD LIST ROUTINE
1621       ;++
1621       ;
1621       ; FUNCTIONAL DESCRIPTION:
1621       ;
1621       ;      THIS ROUTINE SEARCHES THE COMMAND LIST OR A KEYWORD LIST FOR
1621       ;      A MATCH ON THE CURRENT TWO CHARACTERS OF THE INPUT COMMAND
1621       ;      STRING.
1621       ;
1621       ; CALLING SEQUENCE:
1621       ;
1621       ;      CALL    SEARCH_LIST
1621       ;
1621       ; INPUT PARAMETERS:
1621       ;
1621       ;      REGISTER PAIR H & L - CONTAIN THE ADDRESS OF THE INPUT STRING.
1621       ;      REGISTER PARI B & C - CONTAIN THE ADDRESS OF THE SEARCH LIST.
1621       ;
1621       ; IMPLICIT INPUTS:
1621       ;
1621       ;      NONE
1621       ;
1621       ; OUTPUT PARAMETERS:
1621       ;
1621       ;      REGISTER PAIR H & L - POINT AT FIRST CHARACTER PAST THE MATCHED STRING
1621       ;      REGISTER D          - CONTAINS THE WORD POSITION IN THE LIST OF
1621       ;                          - THE MATCHED STRING.
1621       ;      C BIT SET           - IF NO MATCH WAS FOUND IN THE LIST
1621       ;      C BIT CLEAR         - IF A MATCH WAS FOUND.
1621       ;
1621       ;--
1621
1621       SEARCH_LIST:
1621              MVI          D,0               ; INIT THE WORD POSITION
1623       2$:    MVI          E,2               ; WANT TO MATCH TWO CHARACTERS
1625       1$:    LDAX         B                 ; GET CHARACTER FROM SEARCH LIST
1626              ORA          A                 ; CHECK IF LIST TERMINATOR
1627              JM           SEARCH_ERR        ; IF TERMINATOR, EXIT WITH C SET
162A              CMP          M                 ; DOES CHARACTER MATCH?
162B              JZ           3$                ; BRANCH IF YES
162E       ;
162E       ; CHARACTER DID NOT MATCH. IF LOOP COUNT = 2, BUMP MATCH STRING ADDRESS BY
162E       ; TWO BYTES. INCREMENT WORD POSITION BY TWO AND RESTART.
162E       ; IF LOOP COUNT = 1, BUMP MATCH STRING ADDRESS BY ONE, INCREMENT THE WORD
162E       ; POSITION BY TWO, DECREMENT THE INPUT STRING ADDRESS BY ONE, AND RESTART.
162E       ;
162E              INR          D                 ; INCREMENT WORD POSITION BY TWO
162F              INR          D                 ; ...
1630              INX          B                 ; INCREMENT MATCH STR ADR BY ONE
1631              MOV          A,E               ; GET THE LOOP COUNT
1632              CPI          2                 ; LOOP COUNT = 2?
1634              JNZ          4$                ; BRANCH IF NO
1637              INX          B                 ; INCREMENT MATCH STR ADR BY TWO
1638              JMP          1$                ; KEEP TRYING TO MATCH
163B       4$:    DCX          H                 ; DECREMENT INPUT STRING ADDRESS
```

```
163C          JMP             2$              ; KEEP TRYING TO MATCH
163F    ;
163F    ; GOT A MATCH. IF LOOP COUNT = 2, BUMP MATCH STRING ADDRESS AND INPUT
163F    ; STRING ADDRESS BY ONE, DECREMENT THE LOOP COUNT, AND TRY SECOND CHARAC.
163F    ; IF LOOP COUNT = 1, BUMP THE INPUT STRING ADDRESS BY ONE AND RETURN
163F    ; TO CALLER WITH C BIT CLEAR
163F    ;
163F  3$:     DCR             E               ; DECREMENT LOOP COUNT BY 1
1640          JZ              SEARCH_DONE     ; GOT TWO MATCHES, EXIT WITH C CLEAR
1643          INX             B               ; INCREMENT MATCH STR ADDR
1644          INX             H               ; INCREMENT INPUT STRING ADDR
1645          JMP             1$              ; SEE IF SECOND CHAR MATCHES
1648    ;
1648    ; FOUND A MATCH. BUMP INPUT STRING POINTER TO NEXT CHARACTER AND RETURN
1648    ; WITH C CLEAR
1648    ;
1648  SEARCH_DONE:
1648          INX             H               ; INCREMENT INPUT STRING PTR
1649          STC                             ; SET C BIT
164A          CMC                             ; CLEAR THE C BIT
164B          RET                             ; RETURN TO CALLER
164C    ;
164C    ; SEARCH FAILED. RETURN WITH C BIT SET.
164C    ;
164C  SEARCH_ERR:
164C          STC
164D          RET
164E
```

```
164E           .SBTTL  "  SKIP SPACES ROUTINE
164E     ;++
164E     ;
164E     ; FUNCTIONAL DESCRIPTION:
164E     ;
164E     ;         THIS ROUTINE INCREMENTS THE INPUT STRING POINTER TO THE FIRST
164E     ;         CHARACTER AFTER A STRING OF SPACES. IF A CARRIAGE RETURN IS
164E     ;         FOUND, RETURN IS MADE WITH THE C BIT SET.
164E     ;
164E     ; CALLING SEQUENCE:
164E     ;
164E     ;         CALL    SKIP_SPACES
164E     ;
164E     ; INPUT PARAMETERS:
164E     ;
164E     ;         REGISTER PAIR H & L - CONTAINS ADDRESS OF INPUT STRING
164E     ;
164E     ; IMPLICIT INPUTS:
164E     ;
164E     ;         NONE
164E     ;
164E     ; OUTPUT PARAMETERS:
164E     ;
164E     ;         REGISTER PAIR H & L - CONTAINS ADDRESS OF FIRST CHARACTER AFTER A
164E     ;                               STRING OF SPACES.
164E     ;
164E     ; IMPLICIT OUTPUTS:
164E     ;
164E     ;         NONE
164E     ;
164E     ;--
164E
164E     SKIP_SPACES:
164E              MOV             A,M                ; GET CHARACTER IN INPUT BUFFER
164F              CPI             <^XD>              ; IS IT A CARRIAGE RETURN?
1651              JZ              2$                 ; BRANCH IF YES
1654              CPI             <^X20>             ; IS IT A SPACE?
1656              INX             H                  ;
1657              JNZ             SKIP_SPACES        ; BRANCH IF NO
165A     ;
165A     ; FOUND THE FIRST SPACE. NOW SKIP UNTIL NO MORE SPACES.
165A     ;
165A     1$:      MOV             A,M                ; GET THE CHARACTER
165B              CPI             <^X20>             ; IS IT A SPACE?
165D              JNZ             3$                 ; BRANCH IF NO
1660              INX             H                  ; INCREMENT INPUT BUFFER PTR
1661              JMP             1$                 ; CONTINUE SKIPPING SPACES
1664     ;
1664     ; FOUND A CHARACTER THAT IS NOT A SPACE. CHECK IF IT'S A CARRIAGE RETURN.
1664     ;
1664     3$:      CPI             <^XD>              ; IS CHAR. A CARRIAGE RETURN?
1666              JZ              2$                 ; BRANCH IF YES
1669              STC                                ; SET THE C BIT
166A              CMC                                ; CLEAR THE C BIT
166B              RET                                ; RETURN TO CALLER
```

```
                              166C  4536 2$:     STC                           ; SET THE C BIT
                              166D  4537         RET                           ; RETURN TO CALLER
                              166E  4538
```

```
166E           .SBTTL  " PUT D REGISTER IN PARSE TABLE ROUTINE
166E      ;++
166E      ;
166E      ; FUNCTIONAL DESCRIPTION:
166E      ;
166E      ;        THIS ROUTINE PUTS THE D REGISTER INTO THE CURRENT POSITION OF
166E      ;        THE PARSE TABLE.
166E      ;
166E      ; CALLING SEQUENCE:
166E      ;
166E      ;        CALL    PUT_D_IN_TBL
166E      ;
166E      ; INPUT PARAMETERS:
166E      ;
166E      ;        D REGISTER - CONTAINS PARSE TABLE CODE
166E      ;
166E      ; IMPLICIT INPUTS:
166E      ;
166E      ;        PARSE_TBL_PTR - CONTAINS THE ADDRESS OF THE CURRENT POSITION
166E      ;                        IN THE PARSE TABLE
166E      ;
166E      ; IMPLICIT OUTPUTS:
166E      ;
166E      ;        PARSE_TBL_PTR - IS INCREMENTED BY ONE
166E      ;
166E      ;--
166E
166E      PUT_D_IN_TBL:
166E             PUSH          H                  ; SAVE H & L REGISTERS
166F             LHLD          PARSE_TBL_PTR      ; GET ADDR OF PARSE TABLE
1672             MOV           M,D                ; STORE DATA IN PARSE TABLE
1673             INX           H                  ; INCREMENT THE TABLE ADDRESS
1674             SHLD          PARSE_TBL_PTR      ; PUT ADDR BACK IN POINTER
1677             POP           H                  ; RESTORE H & L REGISTERS
1678             RET                              ; RETURN TO CALLING SEQUENCE
1679
```

```
1679                   .SBTTL  " PARSE HEX NUMBER ROUTINE
1679         ;++
1679         ;
1679         ; FUNCTIONAL DESCRIPTION:
1679         ;
1679         ;         THIS ROUTINE CONVERTS AN ASCII STRING OF CHARACTERS INTO A
1679         ;         BINARY NUMBER. IF THE ASCII STRING IS NOT COMPOSED OF LEGAL
1679         ;         HEXIDECIMAL DIGITS, RETURN IS MADE WITH THE C BIT SET. THE
1679         ;         STRING MUST BE TERMINATED WITH A SPACE OR CARRIAGE RETURN.
1679         ;
1679         ;         THE BINARY NUMBER IS PLACED IN THE PARSE TABLE FROM MOST
1679         ;         SIGNIFICANT BYTE TO LEAST SIGNIFICANT BYTE. THE PARSE TABLE
1679         ;         ALWAYS RECEIVES A 32 BIT NUMBER.
1679         ;
1679         ; CALLING SEQUENCE:
1679         ;
1679         ;         CALL    PARSE_HEX_NUMB
1679         ;
1679         ; INPUT PARAMETERS:
1679         ;
1679         ;         REGISTER PAIR H & L - CONTAINS THE ADDRESS OF THE INPUT STRING.
1679         ;
1679         ; IMPLICIT INPUTS:
1679         ;
1679         ;         NONE
1679         ;
1679         ; IMPLICIT OUTPUTS:
1679         ;
1679         ;         REGISTER PAIR H & L - CONTAINS THE ADDRESS OF THE INPUT CHARACTER
1679         ;                               AFTER THE ASCII STRING OF DIGITS.
1679         ;
1679         ; COMPLETION CODES:
1679         ;
1679         ;         C BIT SET   - ILLEGAL HEXIDECIMAL STRING
1679         ;         C BIT CLEAR - STRING PARSE WAS SUCCESSFUL
1679         ;
1679         ;--
1679
1679         PARSE_HEX_NUMB:
1679                   LXI             B,PARSE_HEX_TEMP ; GET ADDRESS OF TEMP LOCATION
167C                   XRA             A                ; CLEAR THE A REGISTER
167D                   STAX            B                ; INITIALIZE THE TEMP LOCATION
167E                   INX             B                ; ...
167F                   STAX            B                ; ...
1680                   INX             B                ; ...
1681                   STAX            B                ; ...
1682                   INX             B                ; ...
1683                   STAX            B                ; ...
1684         ;
1684         ; NOW START CONVERTING THE INPUT STRING
1684         ;
1684         1$:       MOV             A,M              ; GET A CHARACTER
1685                   CPI             <^XD>            ; IS IT A CARRIAGE RETURN?
1687                   JZ              PARSE_HEX_DONE   ; BRANCH IF YES
168A                   CPI             <^X20>           ; IS IT A SPACE?
```

```
168C             JZ      PARSE_HEX_DONE  ; BRANCH IF YES
168F             INX     H               ; INCREMENT POINTER TO INPUT STRING
1690             SUI     <^X30>          ; IS DIGIT GREATER THAN 0?
1692             JM      PARSE_HEX_ERR   ; BRANCH IF NO
1695             CPI     <^XA>           ; IS DIGIT NUMERIC?
1697             JM      2$              ; BRANCH IF YES
169A    ;
169A    ; CHECK IF DIGIT IS A LEGAL ALPHA CHARACTER.
169A    ;
169A             SUI     <^X11>          ; IS DIGIT ALPHA?
169C             JM      PARSE_HEX_ERR   ; BRANCH IF NO
169F             CPI     6               ; IS DIGIT LEGAL HEX?
16A1             JP      PARSE_HEX_ERR   ; BRANCH IF NO
16A4             ADI     <^XA>           ; MAKE DIGIT BINARY HEXIDECIMAL
16A6    ;
16A6    ; BINARY DIGIT IS IN 'A' REG. SHIFT WORKING 32 LOCATION AND INSERT DIGIT.
16A6    ;
16A6    2$:      PUSH    $PSW            ; SAVE 'A' REG
16A7             MVI     E,4             ; SET A LOOP COUNT FOR 4 SHIFTS
16A9    4$:      MVI     D,4             ; SET A LOOP COUNT FOR 4 BYTES
16AB             LXI     B,PARSE_HEX_TEMP ; GET ADDRESS OF TEMP LOCATION
16AE             PUSH    $PSW            ; SAVE THE C BIT (THIS IS A DUMMY
16AF                                     ; PUSH TO INIT THE FOLLOWING LOOP)
16AF    3$:      POP     $PSW            ; RESTORE THE C BIT
16B0             LDAX    B               ; GET A BYTE OF TEMP LOCATION
16B1             RAL                     ; SHIFT IT LEFT
16B2             PUSH    $PSW            ; SAVE THE C BIT
16B3             STAX    B               ; PUT IT BACK IN TEMP LOCATION
16B4             INX     B               ; POINT AT NEXT BYTE
16B5             DCR     D               ; DECREMENT THE COUNT
16B6             JNZ     3$              ; CONTINUE FOR 4 BYTES
16B9    ;
16B9    ; FINISHED 4 BYTES, CHECK IF 4 SHIFTS HAVE BEEN DONE
16B9    ;
16B9             POP     $PSW            ; FIXUP STACK
16BA             DCR     E               ; DECREMENT
16BB             JNZ     4$              ; CONTINUE FOR 4 SHIFTS
16BE    ;
16BE    ; TEMP IS NOW SHIFTED. PUT THE NEW 4 BITS INTO THE TEMP LOCATION
16BE    ;
16BE             LXI     B,PARSE_HEX_TEMP ; GET ADDR OF TEMP LOCATION
16C1             LDAX    B               ; GET THE LOW BYTE
16C2             ANI     <^XF0>          ; THROW AWAY GARBAGE CARRY BIT
16C4             MOV     D,A             ; SAVE IN D REGISTER
16C5             POP     $PSW            ; GET THE DATA TO INSERT
16C6             ORA     D               ; INSERT INTO THE LOW BYTE
16C7             STAX    B               ; PUT BACK IN TEMP LOCATION
16C8             JMP     1$              ; GO GET NEXT CHARACTER
16CB    ;
16CB    ; FOUND A CARRIAGE RETURN OR SPACE. NOW INSERT THE TEMP LOCATION
16CB    ; INTO THE PARSE TABLE.
16CB    ;
16CB    PARSE_HEX_DONE:
16CB             LXI     B,PARSE_HEX_TEMP+3 ; GET ADDR OF HIGH BYTE
16CE             PUSH    H               ; SAVE H & L REGISTERS
```

```
16CF  4687          LHLD          PARSE_TBL_PTR    ; GET ADDR OF PARSE TABLE
16D2  4688          MVI           D,4              ; SET A LOOP COUNT FOR 4 BYTES
16D4  4689 1$:      LDAX          B                ; GET A BYTE OF DATA
16D5  4690          MOV           M,A              ; PUT IN PARSE TABLE
16D6  4691          INX           H                ; INCREMENT PARSE TABLE ADDRESS
16D7  4692          DCX           B                ; DECREMENT TEMP ADDRESS
16D8  4693          DCR           D                ; DECREMENT THE LOOP COUNT
16D9  4694          JNZ           1$               ; MOVE 4 BYTES
16DC  4695          SHLD          PARSE_TBL_PTR    ; SAVE THE CURRENT PARSE TBL ADDRESS
16DF  4696          POP           H                ; RESTORE THE H & L REGISTERS
16E0  4697          STC                            ; SET THE C BIT
16E1  4698          CMC                            ; CLEAR THE C BIT
16E2  4699          RET                            ; RETURN TO THE CALLER
16E3  4700 ;
16E3  4701 ; AN ERROR WAS FOUND PARSING THE STRING. RETURN WITH C BIT SET.
16E3  4702 ;
16E3  4703 PARSE_HEX_ERR:
16E3  4704          STC
16E4  4705          RET
16E5  4706
```

```
16E5                .SBTTL  "ACTION ROUTINES
16E5                .SBTTL  "  DIAGNOSE ACTION ROUTINE
16E5        ;++
16E5        ;
16E5        ; FUNCTIONAL DESCRIPTION:
16E5        ;
16E5        ;       THIS ROUTINE PEFORMS THE NECESSARY FUNCTIONS FOR A 'DIGNOSE'
16E5        ;       COMMAND. THE FOLLOWING FLAGS ARE INITIALIZED:
16E5        ;
16E5        ;       LOST_FLAG   CONT_FLAG   TSTSPAN_FLAG
16E5        ;       LOOP_FLAG   NER_FLAG    BELL_FLAG   HALT_FLAG
16E5        ;
16E5        ;       THE PARSE TABLE IS THEN INTERPRETED INTO THE APPROPRIATE ACTIONS.
16E5        ;
16E5        ; CALLING SEQUENCE:
16E5        ;
16E5        ;       CALL    DIAGNOSE_ACTION
16E5        ;
16E5        ; INPUT PARAMETERS:
16E5        ;
16E5        ;       REGISTER PAIR D & E - CONTAIN ADDRESS OF PARSE TABLE PAST THE
16E5        ;                             COMMAND CODE.
16E5        ;
16E5        ; IMPLICIT INPUTS:
16E5        ;
16E5        ;       NONE
16E5        ;
16E5        ; OUTPUT PARAMETERS:
16E5        ;
16E5        ;       NONE
16E5        ;
16E5        ; IMPLICIT OUTPUTS:
16E5        ;
16E5        ;       THE PROGRAM CONTROL FLAGS ARE INITIALIZED.
16E5        ;       THE 'DIAGNOSE_FLAG' IS SET IN THE SOFTWARE CONTROL REGISTER
16E5        ;
16E5        ; COMPLETION CODES:
16E5        ;
16E5        ;       C BIT SET   - ILLEGAL QUALIFIER WITH THE 'DIAGNOSE' COMMAND
16E5        ;       C BIT CLEAR - DIAGNOSE COMMAND EXECUTION WAS SUCCESSFUL
16E5        ;
16E5        ;--
16E5
16E5
16E5        DIAGNOSE_ACTION:
16E5                LDA             PROG_CTRL_REG   ; GET PROGRAM CONTROL REG
16E8                ANI             <^C<LOOP_FLAG!NER_FLAG!- ; CLEAR THE LOOP, NER,
16E8                                BELL_FLAG>>     ; AND BELL FLAGS
16EA                ORI             HALT_FLAG       ; SET THE 'HALT' FLAG
16EC                STA             PROG_CTRL_REG   ; ...
16EF                LDA             SOFT_CTRL_REG   ; GET THE SOFTWARE CONTROL REG
16F2                ANI             <^C<LOST_FLAG!- ; CLEAR THE LOST,
16F2                                CONT_FLAG!TSTSPAN_FLAG!- ; CONTINUE, TEST SPAN,
16F2                                DIAGNOSE_FLAG>> ; AND DIAGNOSE FLAGS
16F4                STA             SOFT_CTRL_REG   ; ...
```

```
16F7          LDA       SOFT_CTRL_REGB    ; SET THE ERROR LOG INIT FLAG
16FA          ORI       ERR_LOG_INIT      ; ...
16FC          ANI       <^C<QV_FLAG!-     ; AND CLEAR THE QV AND RDM FLAGS
16FC                    RDM_FLAG!-        ; AND SBE FLAG
16FC                    SBE_FLAG>>        ; ...
16FE          STA       SOFT_CTRL_REGB    ; ...
1701          MVI       A,1               ; GET THE INITIAL PASS COUNT
1703          STA       USER_PASS_CNT     ; INIT THE PASS COUNT
1706          XCHG                        ; PUT PARSE TBL ADDR IN H & L REG'S
1707          XRA       A                 ; CLEAR THE A REG
1708          STA       TEST_NUMBER       ; INIT THE TEST NUMBER
170B          STA       TYPE_TEST_FLAG    ; INIT THE TYPE TEST FLAG
170E          MOV       D,A               ; INITIALIZE THE TEST FLAG
170F          MVI       C,3               ; PUT CONSTANT 3 IN B & C REG'S
1711          MOV       B,A               ; ...
1712    ;
1712    ; NOW START DECODING THE CODES IN THE PARSE TABLE
1712    ;
1712    DIAGNOSE_LOOP:
1712          MOV       A,M               ; GET A KEYWORD
1713          ANI       <^X7F>            ; DISCARD KEYWORD CODE BIT
1715          INX       H                 ; INCREMENT PARSE TABLE POINTER
1716          CPI       <^X55>            ; END OF PARSE TABLE?
1718          JZ        DIAGNOSE_DONE     ; BRANCH IF YES
171B    ;
171B    ; SEE WHICH KEYWORD IS IN PARSE TABLE.
171B    ;
171B          CPI       TEST_KEY          ; 'TEST' KEYWORD?
171D          JZ        TEST_SUB_ACTION   ; BRANCH IF YES
1720          CPI       PASS_KEY          ; 'PASS' KEYWORD?
1722          JZ        PASS_SUB_ACTION   ; BRANCH IF YES
1725          CPI       CONT_KEY          ; 'CONTINUE' KEYWORD?
1727          JZ        CONT_SUB_ACTION   ; BRANCH IF YES
172A          CPI       QV_KEY            ; 'QUICK VERIFY' KEYWORD?
172C          JZ        QV_SUB_ACTION     ; BRANCH IF YES
172F          CPI       RDM_KEY           ; 'RDM' KEYWORD?
1731          JZ        RDM_SUB_ACTION    ; BRANCH IF YES
1734          CPI       SBE_KEY           ; SINGLE BIT ERROR ENABLE?
1736          JNZ       DIAGNOSE_ERR      ; BRANCH IF NO
1739    ;
1739    ; SBE KEYWORD. SET THE SBE FLAG.
1739    ;
1739          LDA       SOFT_CTRL_REGB    ; GET CURRENT FLAGS
173C          ORI       SBE_FLAG          ; SET THE SBE FLAG
173E          STA       SOFT_CTRL_REGB    ; ...
1741          JMP       DIAGNOSE_LOOP     ; CONTINUE PARSING
1744    ;
1744    ; RDM KEYWORD. SET THE RDM FLAG.
1744    ;
1744    RDM_SUB_ACTION:
1744          LDA       SOFT_CTRL_REGB    ; GET CURRENT FLAGS
1747          ORI       RDM_FLAG          ; SET THE RDM FLAG
1749          STA       SOFT_CTRL_REGB    ; ...
174C          JMP       DIAGNOSE_LOOP     ; CONTINUE PARSING
174F    ;
```

```
174F  ; QUICK VERIFY KEYWORD. SET THE QV FLAG.
174F  ;
174F  QV_SUB_ACTION:
174F          LDA             SOFT_CTRL_REGB  ; GET CURRENT FLAGS
1752          ORI             QV_FLAG         ; SET THE QV FLAG
1754          STA             SOFT_CTRL_REGB  ; ...
1757          JMP             DIAGNOSE_LOOP   ; CONTINUE PARSING
175A  ;
175A  ; CONTINUE KEYWORD WAS RECOGNIZED. SET THE 'CONT' FLAG AND KEEP PARSING.
175A  ;
175A  CONT_SUB_ACTION:
175A          LDA             SOFT_CTRL_REG   ; GET CONTENTS OF SOFTWARE CTRL REG
175D          ORI             CONT_FLAG       ; SET THE 'CONT' FLAG
175F          STA             SOFT_CTRL_REG   ; UPDATE SOFTWARE CONTROL REG
1762          JMP             DIAGNOSE_LOOP   ; CONTINUE SEARCHING PARSE TABLE
1765  ;
1765  ; 'TEST' KEYWORD WAS RECOGNIZED. SET THE 'LOST' FLAG AND GET THE TEST
1765  ; NUMBER(S).
1765  ;
1765  TEST_SUB_ACTION:
1765          MOV             A,D             ; GET THE TEST/SECTION FLAG
1766          ORA             A               ; SET THE CONDITION CODES
1767          JNZ             DIAGNOSE_ERR    ; BRANCH IF SECTION ALREADY SPECIFIED
176A          MVI             D,1             ; SET THE TEST/SECTION FLAG
176C          DAD             B               ; POINT AT THE TEST NUMBER IN PARSE TBL
176D          MOV             A,M             ; GET THE TEST NUMBER
176E          STA             SPEC_TEST_NUMB  ; SAVE
1771          INX             H               ; POINT AT NEXT ENTRY IN PARSE TBL
1772          MVI             E,LOST_FLAG     ; GET THE 'LOST' FLAG IN 'E' REG
1774          MOV             A,M             ; GET NEXT ELEMENT IN PARSE TBL
1775          ORA             A               ; SET THE CONDITION CODES
1776          MVI             A,0
1778          JM              1$              ; BRANCH IF IT'S A KEYWORD
177B          MOV             A,M             ; GET NEXT ELEMENT IN PARSE TBL
177C          CPI             <^X55>          ; END OF PARSE TABLE?
177E          MVI             A,0             ; CLEAR THE A REG
1780          JZ              1$              ; BRANCH IF YES
1783  ;
1783  ; AN END TEST HAS BEEN SPECIFIED. GET THE NUMBER OUT OF THE PARSE TABLE.
1783  ;
1783          DAD             B               ; POINT AT THE NUMBER
1784          MOV             A,M             ; GET THE END TEST NUMBER
1785          STA             TEST_SPAN_END   ; SAVE
1788          INX             H               ; UPDATE PARSE TBL POINTER
1789          MVI             A,TSTSPAN_FLAG  ; GET THE TEST SPAN FLAG
178B  ;
178B  ; 'E' REG HAS 'LOST' FLAG. 'A' REG HAS 'TSTSPAN'  FLAG OR ZERO.
178B  ; OR TOGETHER AND LOAD SOFT_CTRL_REG.
178B  ;
178B  1$:
178B          ORA             E               ; OR IN THE 'LOST' FLAG
178C          MOV             E,A             ; SAVE
178D          LDA             SOFT_CTRL_REG   ; GET THE CONTROL REGISTER
1790          ORA             E               ; OR IN THE TWO FLAGS
1791          STA             SOFT_CTRL_REG   ; SAVE
```

```
1794            JMP             DIAGNOSE_LOOP   ; CONTINUE PARSING DIAGNOSE COMMAND
1797    ;
1797    ; 'PASS' KEYWORD WAS RECOGNIZED. GET THE USER PASS COUNT FROM THE
1797    ; PARSE TABLE.
1797    ;
1797    PASS_SUB_ACTION:
1797            DAD             B               ; POINT AT THE PASS COUNT
1798            MOV             A,M             ; GET THE PASS COUNT
1799            STA             USER_PASS_CNT   ; SAVE
179C            INX             H               ; POINT AT NEXT ELEMENT IN PARSE TBL
179D            JMP             DIAGNOSE_LOOP   ; CONTINUE PARSING DIAGNOSE COMMAND
17A0    ;
17A0    ; AN ILLEGAL DIAGNOSE COMMAND WAS DETECTED. RETURN WITH C BIT SET
17A0    ;
17A0    DIAGNOSE_ERR:
17A0            STC
17A1            RET
17A2    ;
17A2    ; DIAGNOSE PARSE COMPLETE. SET THE 'DIAGNOSE_FLAG' AND RETURN.
17A2    ;
17A2    DIAGNOSE_DONE:
17A2            LDA             SOFT_CTRL_REG   ; GET THE CONTROL REG
17A5            ORI             DIAGNOSE_FLAG   ; SET THE 'DIAGNOSE' FLAG
17A7            STA             SOFT_CTRL_REG   ; ...
17AA            STC
17AB            CMC
17AC            RET
17AD
```

```
17AD          .SBTTL  "  SHOW ACTION ROUTINE
17AD  ;++
17AD  ;
17AD  ; FUNCTIONAL DESCTIPTION:
17AD  ;
17AD  ;       THIS ROUTINE DISPLAYS THE CURRENT VALUE OF THE 'PROGRAM' CONTROL
17AD  ;       FLAGS OR THE VISIBILITY BUS. THE PRINT OUT FOR THE PROGRAM CONTROL
17AD  ;       FLAGS IS:
17AD  ;
17AD  ;               FLAGS SET: <FLAG NAMES>
17AD  ;               FLAGS CLEAR: <FLAG NAMES>
17AD  ;
17AD  ;       THE VBUS PRINTOUT DISPLAYS THE VALUE OF EACH VBUS BIT IN BINARY.
17AD  ;       GROUPS OF 8 BITS ARE SEPARATED BY COMMAS. BIT 0 OF THE BUS IS
17AD  ;       ON THE RIGHT OF THE TYPEOUT.
17AD  ;
17AD  ; CALLING SEQUENCE:
17AD  ;
17AD  ;       CALL    SHOW_ACTION
17AD  ;
17AD  ; INPUT PARAMETERS:
17AD  ;
17AD  ;       REGISTER PAIR D & E - CONTAIN THE ADDRESS OF THE PARSE_TABLE+1
17AD  ;
17AD  ; IMPLICIT INPUTS:
17AD  ;
17AD  ;       PROG_CTRL_REG - CONTAINS THE CURRENT STATE OF THE FLAGS.
17AD  ;
17AD  ; OUTPUT PARAMETERS:
17AD  ;
17AD  ;       NONE
17AD  ;
17AD  ; IMPLICIT OUTPUTS:
17AD  ;
17AD  ;       NONE
17AD  ;
17AD  ; COMPLETION CODES:
17AD  ;
17AD  ;       RETURN IS MADE WITH THE C BIT CLEAR IF NO ERROR OTHERWISE IT IS SET.
17AD  ;
17AD  ;--
17AD
17AD  SHOW_ACTION:
17AD  ;
17AD  ; SEE IF THIS IS A FLAG SHOW
17AD  ;
17AD          LDAX            D               ; GET THE KEYWORD CODE
17AE          CPI             <FLAG_KEY!^X80> ; IS IT 'FLAG'?
17B0          JNZ             10$             ; BRANCH IF NO
17B3  ;
17B3  ; DISPLAY THE PROGRAM CONTROL FLAGS
17B3  ;
17B3          TYPE            SHOW_MESSAGE_1  ; TYPE THE 'FLAGS SET: ' MESSAGE
17BC          LDA             PROG_CTRL_REG   ; GET THE FLAGS
17BF          CALL            TYPE_FLAGS      ; TYPE THE FLAG NAMES THAT ARE SET
```

```
17C2            TYPE    SHOW_MESSAGE_2   ; TYPE THE 'FLAGS CLEAR: ' MESSAGE
17CB            LDA     PROG_CTRL_REG    ; GET THE FLAGS
17CE            CMA                      ; GET THE FLAGS THAT ARE CLEAR
17CF            CALL    TYPE_FLAGS       ; TYPE THE FLAG NAMES THAT ARE CLEAR
17D2            JMP     50$              ; EXIT
17D5    ;
17D5    ; THIS IS A VISIBILITY BUS 'SHOW'. READ THE V BUS.
17D5    ;
17D5    10$:    CPI     <VB_KEY!^X80>    ; IS IT A VISIBILITY BUS SHOW?
17D7            JNZ     40$              ; BRANCH IF NO
17DA      ;
17DA      ; FIRST CLEAR THE CONTROL FILE
17DA      ;
17DA            LDA     DCS_CTRL_RE_CPY  ; GET CURRENT CONTENTS OF CTRL REG
17DD            ORI     CLEAR_CTRL_FILE  ; SET CONTROL FILE CLEAR BIT
17DF            STA     DCS_CTRL_REG     ; ...
17E2            ANI     <^C<CLEAR_CTRL_FILE>>
17E4            STA     DCS_CTRL_REG     ; CLEAR CONTROL FILE CLEAR BIT
17E7            LDA     FRONT_PNL_2      ; FIRST LOAD THE VBUS
17EA            ANI     <FAULT!TEST!REMOTE> ; ...
17EC            ORI     <MASTER_HALT_EN!-
17EC                    VBUS_LOAD!TRAP_HALT_EN>
17EE            STA     RD_CTRL_REG      ; LOAD THE VBUS
17F1            ANI     <^CVBUS_LOAD>    ; ...
17F3            STA     RD_CTRL_REG      ; ...
17F6            CALL    READ_V_BUS       ; READ THE V BUS INTO THE BUFFER
17F9            TYPE    VBUS_MSG         ; TYPE 'VBUS= '
1802    ;
1802    ; NOW TYPE THE BITS
1802    ;
1802            LXI     H,V_BUS_BUFFER+39 ; GET THE BUFFER ADDRESS
1805            MVI     B,40             ; SET THE LOOP COUNT
1807    20$:    MOV     A,M              ; GET THE BIT VALUE
1808            DCX     H                ; POINT AT NEXT BIT
1809            PUSH    H                ; SAVE REGISTERS
180A            PUSH    B                ; ...
180B            STA     SHOW_TEMP
180E            TYPEN   SHOW_TEMP        ; TYPE THE BIT VALUE
1819            POP     B                ; RESTORE THE REGISTERS
181A            POP     H                ; ...
181B            DCR     B                ; DONE 40 BITS YET?
181C            JZ      50$              ; BRANCH IF YES
181F            MOV     A,B              ; GET THE LOOP COUNT
1820            ANI     7                ; TIME FOR A COMMA?
1822            JNZ     20$              ; BRANCH IF NO
1825            PUSH    H                ; SAVE REGISTERS
1826            PUSH    B                ; ...
1827            TYPE    COMMA_MSG        ; TYPE A COMMA
1830            POP     B                ; RESTORE THE REGISTERS
1831            POP     H                ; ...
1832            JMP     20$              ; CONTINUE
1835    ;
1835    ; RETURN WITH C BIT CLEAR
1835    ;
1835    50$:    STC
```

```
1836  5012          CMC
1837  5013          RET
1838  5014 ;
1838  5015 ; ILLEGAL KEYWORD. RETURN WITH C BIT SET
1838  5016 ;
1838  5017 40$:     STC
1839  5018          RET
183A  5019
```

```
183A            .SBTTL  "  CONTINUE ACTION ROUTINE
183A    ;++
183A    ;
183A    ; FUNCTIONAL DESCTIPTION:
183A    ;
183A    ;       THIS ROUTINE SETS THE 'CONTINUE' FLAG IN THE SOFTWARE CONTROL
183A    ;       REGISTER.
183A    ;
183A    ; CALLING SEQUENCE:
183A    ;
183A    ;       CALL    CONTINUE_ACTION
183A    ;
183A    ; INPUT PARAMETERS:
183A    ;
183A    ;       NONE
183A    ;
183A    ; IMPLICIT INPUTS:
183A    ;
183A    ;       NONE
183A    ;
183A    ; OUTPUT PARAMETERS:
183A    ;
183A    ;       NONE
183A    ;
183A    ; IMPLICIT OUTPUTS:
183A    ;
183A    ;       SOFT_CTRL_REG - 'CONTINUE_FLAG' SI SET
183A    ;
183A    ; COMPLETION CODES:
183A    ;
183A    ;       RETURN IS MADE WITH THE C BIT CLEAR.
183A    ;
183A    ;--
183A
183A    CONTINUE_ACTION:
183A            LDA             SOFT_CTRL_REG   ; GET THE CURRENT FLAGS
183D            ORI             CONTINUE_FLAG   ; SET THE 'CONTINUE' FLAG
183F            STA             SOFT_CTRL_REG   ; ...
1842            STC
1843            CMC
1844            RET
1845
```

```
1845                 .SBTTL  "  RETURN ACTION ROUTINE
1845         ;++
1845         ;
1845         ; FUNCTIONAL DESCRIPTION:
1845         ;
1845         ;       THIS ROUTINE CAUSES A RETURN TO THE RDM SOFTWARE. THIS IS DONE
1845         ;       BY CALLING THE 'ABORT' ROUTINE.
1845         ;
1845         ; CALLING SEQUENCE:
1845         ;
1845         ;       CALL    RETURN_ACTION
1845         ;
1845         ;--
1845
1845         RETURN_ACTION:
1845                 ABORT
1848
```

```
1848          .SBTTL  "  LOOP ACTION ROUTINE
1848  ;++
1848  ;
1848  ; FUNCTION DESCRIPTION:
1848  ;
1848  ;       THIS ROUTINE SETS THE 'LOOP' AND 'NER' AND CLEARS THE 'HALT'
1848  ;       FLAG IN THE PROGRAM CONTROL REGISTER. IT ALSO TRANSFERS THE
1848  ;       THE CONTENTS OF 'CURRENT_PC' TO 'ERROR_PC' FOR THE IFERROR ROUTINE.
1848  ;
1848  ; CALLING SEQUENCE:
1848  ;
1848  ;       CALL    LOOP_ACTION
1848  ;
1848  ;--
1848
1848  LOOP_ACTION:
1848          LDA             PROG_CTRL_REG    ; GET THE CURRENT FLAGS
184B          ORI             <NER_FLAG!LOOP_FLAG> ; SET THE NER & LOOP FLAGS
184D          ANI             <^CHALT_FLAG>    ; CLEAR THE 'HALT' FLAG
184F          STA             PROG_CTRL_REG    ; ...
1852          LDA             SOFT_CTRL_REG    ; NOW SET THE 'CONTINUE' FLAG
1855          ORI             CONTINUE_FLAG    ; TO SIMULATE A 'CONTINUE'
1857          STA             SOFT_CTRL_REG    ; COMMAND
185A          LHLD            CURRENT_PC       ; GET THE CURRENT PC
185D          SHLD            ERROR_PC         ; SAVE IN ERROR_PC
1860          STC
1861          CMC
1862          RET
1863
```

```
1863            .SBTTL  "  SET ACTION ROUTINE
1863     ;++
1863     ;
1863     ; FUNCTIONAL DESCRIPTION:
1863     ;
1863     ;       THIS ROUTINE EXECUTES THE MANY FORMS OF THE 'SET' COMMAND. THE
1863     ;       FORMS SUPPORTED ARE:
1863     ;
1863     ;               SET FLAG
1863     ;               SET SOMM
1863     ;               SET CF
1863     ;               SET STEP
1863     ;
1863     ; INPUT PARAMETERS:
1863     ;
1863     ;       REGISTER PAIR D & E - CONTAIN ADDRESS OF THE PARSE TABLE.
1863     ;
1863     ; IMPLICIT INPUTS:
1863     ;
1863     ;       NONE
1863     ;
1863     ; OUTPUT PARAMETERS:
1863     ;
1863     ;       NONE
1863     ;
1863     ; IMPLICIT OUTPUTS:
1863     ;
1863     ;       NONE
1863     ;
1863     ; COMPLETION CODES:
1863     ;
1863     ;       C BIT SET   - SYNTAX ERROR IN SET COMMAND
1863     ;       C BIT CLEAR - SET COMMAND EXECUTED SUCCESSFULLY
1863     ;
1863     ;--
1863
1863     SET_ACTION:
1863            XCHG                            ; PUT PARSE TBL ADDR IN H & L
1864            MVI             C,3             ; GET CONSTANT TO INCREMENT PARSE
1866            MVI             B,0             ; TABLE ADDRESS
1868            MOV             A,M             ; GET A KEYWORD
1869            ANI             <^X7F>          ; DISCARD KEYWORD FLAG BIT
186B            INX             H               ; POINT AT NEXT ELEMENT OF PARSE TBL
186C            CPI             <^X55>          ; END OF PARSE TABLE?
186E            JZ              SET_ERR         ; BRANCH IF YES
1871            CPI             FLAG_KEY        ; IS IT A 'FLAG' SET?
1873            JZ              SFLAG_SUB_ACT   ; BRANCH IF YES
1876            CPI             STEP_KEY        ; IS IT A 'STEP' SET?
1878            JZ              SSTEP_SUB_ACT   ; BRANCH IF YES
187B            CPI             SOMM_KEY        ; IS IT A 'SOMM' SET?
187D            JZ              SSOMM_SUB_ACT   ; BRANCH IF YES
1880            CPI             CF_KEY          ; IS TI A 'CF' SET?
1882            JNZ             SET_ERR         ; BRANCH IF NO
1885     ;
1885     ; CONTROL FILE (CF) KEYWORD RECOGNIZED. GET THE DCS ADDRESS AND BIT NUMBER
```

```
1885  ; AND LOAD THE CONTROL FILE.
1885  ;
1885            DAD             B                 ; POINT AT DCS ADDRESS
1886            MOV             A,M               ; GET THE ADDRESS
1887            STA             SETCLR_CF_ADDR    ; SAVE
188A            INX             H                 ; INCREMENT PARSE TBL PTR
188B            DAD             B                 ; POINT AT BIT(S) TO SET
188C            MOV             A,M               ; GET THE DATA
188D            CALL            BIT_NMB_TO_MSK    ; DECODE THE BIT NUMBER
1890            STA             SETCLR_CF_DATA    ; SAVE
1893            CALL            SET_DCS_CF        ; SET THE BITS IN THE CONTROL FILE
1896            JMP             SET_DONE          ; EXIT
1899  ;
1899  ; FLAG KEYWORD RECOGNIZED. SET THE SPECIFIED FLAG.
1899  ;
1899  SFLAG_SUB_ACT:
1899            MOV             A,M               ; GET THE KEYWORD FLAG NAME
189A            ANI             <^X7F>            ; DISCARD KEYWORD FLAG BIT
189C            INX             H                 ; POINT AT NEXT ELEMENT OF PARSE TBL
189D            CPI             <^X55>            ; END OF PARSE TABLE?
189F            JZ              SET_DONE          ; BRANCH IF YES
18A2            CALL            DECODE_FLAG_NAM   ; GO DECODE THE FLAG NAME
18A5            JC              SET_ERR           ; EXIT WITH C SET IF ERROR
18A8            LDA             PROG_CTRL_REG     ; GET THE PROGRAM CONTROL FLAGS
18AB            ORA             C                 ; OR IN THE NEW FLAG
18AC            STA             PROG_CTRL_REG     ; SAVE
18AF  ;
18AF  ; CHECK IF QA FLAG.
18AF  ;
18AF            MOV             A,C
18B0            CPI             QA_FLAG           ; QA FLAG?
18B2            JNZ             SFLAG_SUB_ACT     ; BRANCH IF NO
18B5  ;
18B5  ; QA FLAG, CHECK IF LOOP COUNT SPECIFIED
18B5  ;
18B5            MVI             A,-1              ; INIT QA COUNT
18B7            STA             QA_COUNT          ; ...
18BA            STA             QA_COUNT+1        ; ...
18BD            MOV             A,M               ; GET NEXT ELEMENT IN PARSE TABLE
18BE            ORA             A                 ; SET THE CONDITION CODES
18BF            JM              SFLAG_SUB_ACT     ; BRANCH IF KEYWORD
18C2            CPI             <^X55>            ; END OF PARSE TABLE?
18C4            JZ              SET_DONE          ; BRANCH IF YES
18C7            INX             H                 ; POINT TO BYTE 1 OF LOOP COUNT
18C8            INX             H                 ; ...
18C9            MOV             A,M               ; GET HIGH BYTE OF LOOP COUNT
18CA            STA             QA_COUNT+1        ; SAVE
18CD            INX             H
18CE            MOV             A,M               ; GET BYTE 0 OF COUNT
18CF            STA             QA_COUNT          ; SAVE
18D2            INX             H                 ; POINT TO NEXT ENTRY IN PARSE TABLE
18D3            JMP             SFLAG_SUB_ACT     ; SEE IF MORE FLAGS
18D6  ;
18D6  ; STEP KEYWORD WAS RECOGNIZED. DECODE THE NEXT KEYWORD AND SET THE
18D6  ; APPROPRIATE BIT IN SOFTWARE CONTROL REG. 1.
```

```
18D6   ;
18D6   SSTEP_SUB_ACT:
18D6            MOV     A,M                ; GET THE KEYWORD
18D7            ANI     <^X7F>             ; DISCARD KEYWORD FLAG BIT
18D9            INX     H                  ; POINT AT NEXT ELEMENT IN PARSE TBL
18DA            CPI     <^X55>             ; END OF PARSE TABLE?
18DC            JZ      SET_ERR            ; BRANCH IF YES
18DF            MVI     D,CYCLE_FLAG       ; INIT 'D' REG WITH CYCLE FLAG
18E1            CPI     CYCLE_KEY          ; IS IT 'CYCLE'?
18E3            JZ      1$                 ; BRANCH IF YES
18E6            MVI     D,TICK_FLAG        ; INIT 'D' REG WITH TICK FLAG
18E8            CPI     TICK_KEY           ; IS IT 'TICK'?
18EA            JZ      1$                 ; BRANCH IF YES
18ED            MVI     D,INSTR_FLAG       ; INIT 'D' REG WITH INSTRUCTION FLAG
18EF            CPI     INSTR_KEY          ; IS IT 'INSTRUCTION'?
18F1            JNZ     SET_ERR            ; BRANCH IF NO
18F4   ;
18F4   ; SET STEP INSTRUCTION. SEE IF A SPECIAL TEST PC WAS SPECIFIED.
18F4   ;
18F4            MVI     A,-1               ; INIT THE SPECIAL TEST PC
18F6            STA     SPEC_TEST_PC+1     ; ...
18F9            MOV     A,M                ; GET NEXT ELEMENT IN PARSE TABLE
18FA            CPI     <^X55>             ; END OF PARSE TABLE?
18FC            JZ      1$                 ; BRANCH IF YES
18FF            INX     H                  ; POINT AT SPECIFIED TEST PC
1900            INX     H                  ; ...
1901            MOV     A,M                ; GET UPPER ORDER BITS
1902            STA     SPEC_TEST_PC+1     ; SAVE
1905            INX     H                  ; POINT AT LOW BITS
1906            MOV     A,M                ; GET THEM
1907            STA     SPEC_TEST_PC       ; SAVE
190A   1$:      LDA     SOFT_CTRL_REGA     ; GET CONTROL REGISTER 1
190D            ANI     <^C<CYCLE_FLAG!-
190D                    TICK_FLAG>>        ; CLEAR BOTH CYCLE AND TICK FLAGS
190F            ORA     D                  ; OR IN THE APPROPRIATE BIT
1910            STA     SOFT_CTRL_REGA     ; SAVE
1913            JMP     SET_DONE           ; EXIT
1916   ;
1916   ; SOMM KEYWORD WAS RECOGNIZED. SET THE STOP ON MATCH BIT AND LOAD THE
1916   ; ADDRESS INTO THE MATCH REGISTER IF IT WAS SPECIFIED.
1916   ;
1916   SSOMM_SUB_ACT:
1916            LDA     SOFT_CTRL_REGA     ; SET THE SOMM FLAG
1919            ORI     SOMM_FLAG          ; ...
191B            STA     SOFT_CTRL_REGA     ; ...
191E   ;
191E   ; SET THE STOP ON MATCH BIT
191E   ;
191E            LDA     DCS_CTRL_RE_CPY    ;
1921            ORI     HALT_ON_MATCH      ;
1923            STA     DCS_CTRL_RE_CPY    ;
1926            STA     DCS_CTRL_REG       ;
1929   ;
1929   ; CHECK IF ADDRESS SPECIFIED
1929   ;
```

```
1929           MOV     A,M             ; GET NEXT KEYWORD
192A           CPI     <^X55>          ; END OF PARSE TABLE?
192C           JZ      SET_DONE        ; BRANCH IF YES
192F      ;
192F      ; GET THE ADDRESS AND LOAD THE MATCH REGISTER
192F      ;
192F           INX     H               ; POINT TO THE ADDRESS
1930           INX     H               ; BYTE 1
1931           MOV     A,M             ; GET UPPER BYTE OF ADDRESS
1932           STA     ADDR_MATCH_HI   ; LOAD ADDRESS MATCH REG
1935           STA     SOMM_ADDRESS+1  ; AND SAVE
1938           INX     H               ; POINT AT BYTE 0
1939           MOV     A,M             ; GET BYTE 0 OF ADDRESS
193A           STA     ADDR_MATCH_LO   ; LOAD ADDRESS MATCH LO
193D           STA     SOMM_ADDRESS    ; AND SAVE
1940      ;
1940      ; EXIT WITH C BIT CLEAR
1940      ;
1940      SET_DONE:
1940           STC
1941           CMC
1942           RET
1943      ;
1943      ; EXIT WITH C BIT SET
1943      ;
1943      SET_ERR:
1943           STC
1944           RET
1945
```

```
1945             .SBTTL  "  CLEAR ACTION ROUTINE
1945      ;++
1945      ;
1945      ; FUNCTIONAL DESCRIPTION:
1945      ;
1945      ;      THIS ROUTINE EXECUTES THE MANY FORMS OF THE 'CLEAR' COMMAND.
1945      ;      THE FORMS SUPPORTED ARE:
1945      ;
1945      ;               CLEAR FLAG
1945      ;               CLEAR SOMM
1945      ;               CLEAR CF
1945      ;
1945      ; CALLING SEQUENCE:
1945      ;
1945      ;      CALL    CLEAR_ACTION
1945      ;
1945      ; INPUT PARAMETERS:
1945      ;
1945      ;      REGISTER PAIR D & E - CONTAINS THE ADDRESS OF THE PARSE TABLE.
1945      ;
1945      ; IMPLICIT INPUTS:
1945      ;
1945      ;      NONE
1945      ;
1945      ; OUTPUT PARAMETERS:
1945      ;
1945      ;      NONE
1945      ;
1945      ; IMPLICIT OUTPUTS:
1945      ;
1945      ;      NONE
1945      ;
1945      ; COMPLETION CODES:
1945      ;
1945      ;      C BIT SET   - SYNTAX ERROR IN CLEAR COMMAND
1945      ;      C BIT CLEAR - CLEAR COMMAND EXECUTED SUCCESSFULLY
1945      ;
1945      ;--
1945
1945      CLEAR_ACTION:
1945             XCHG                                  ; PUT PARSE TABLE ADDR IN H & L
1946             MVI              B,0                  ; PUT A CONSTANT 3 IN B & C REG
1948             MVI              C,3                  ; ...
194A             MOV              A,M                  ; GET KEYWORD CODE
194B             ANI              <^X7F>               ; DISCARD KEYWORD FLAG BIT
194D             INX              H                    ; POINT AT NEXT ENTRY IN PARSE TBL
194E             CPI              <^X55>               ; END OF PARSE TABLE?
1950             JZ               CLEAR_ERR            ; BRANCH IF YES
1953             CPI              FLAG_KEY             ; IS IT A 'FLAG' CLEAR?
1955             JZ               CFLAG_SUB_ACT        ; BRANCH IF YES
1958             CPI              CF_KEY               ; IS IT A 'CF' CLEAR?
195A             JZ               CCF_SUB_ACT          ; BRANCH IF YES
195D             CPI              SOMM_KEY             ; IS TI A 'SOMM' CLEAR?
195F             JNZ              CLEAR_ERR            ; BRANCH IF NO
1962      ;
```

```
1962  ; SOMM KEYWORD WAS DETECTED. CLEAR THE SOMM FLAG AND THE HALT ON MATCH BIT
1962  ; AND LOAD THE MATCH REGISTER IF AN ADDRESS IS SPECIFIED
1962  ;
1962          LDA     SOFT_CTRL_REGA    ; GET THE SOMM FLAG
1965          ANI     <^CSOMM_FLAG>     ; CLEAR IT
1967          STA     SOFT_CTRL_REGA    ; ...
196A          LDA     DCS_CTRL_RE_CPY   ; CLEAR THE HALT ON MATCH BIT
196D          ANI     <^CHALT_ON_MATCH> ; ...
196F          STA     DCS_CTRL_RE_CPY   ; ...
1972          STA     DCS_CTRL_REG      ; ...
1975  ;
1975  ; NOW LOAD THE MATCH REGISTER IF AN ADDRESS IS SPECIFIED.
1975  ;
1975  1$:     MOV     A,M               ; GET NEXT PARSE TABLE ENTRY
1976          CPI     <^X55>            ; END OF PARSE TABLE?
1978          JZ      CLEAR_DONE        ; BRANCH IF YES
197B          INX     H
197C          INX     H
197D          MOV     A,M               ; GET THE ADDRESS
197E          STA     ADDR_MATCH_HI     ; LOAD THE HI ORDER BITS
1981          STA     SOMM_ADDRESS+1    ; AND SAVE
1984          INX     H
1985          MOV     A,M               ; GET THE LOW ORDER BITS
1986          STA     ADDR_MATCH_LO     ; LOAD THE LOW ORDER BITS
1989          STA     SOMM_ADDRESS      ; AND SAVE
198C          JMP     CLEAR_DONE        ; EXIT
198F  ;
198F  ; FLAG KEYWORD WAS DETECTED. DECODE AND CLEAR THE APPROPRIATE PROGRAM
198F  ; FLAG.
198F  ;
198F  CFLAG_SUB_ACT:
198F          MOV     A,M               ; GET THE KEYWORD FLAG NAME
1990          ANI     <^X7F>            ; DISCARD KEYWORD FLAG BIT
1992          INX     H                 ; POINT AT NEXT PARSE TBL ENTRY
1993          CPI     <^X55>            ; END OF PARSE TABLE?
1995          JZ      CLEAR_DONE        ; BRANCH IF YES
1998          CALL    DECODE_FLAG_NAM   ; DECODE THE FLAG NAME
199B          JC      CLEAR_ERR         ; BRANCH IF ERROR
199E          MOV     A,C               ; GET THE FLAG BIT
199F          CMA                       ; INVERT FOR THE CLEAR OPERATION
19A0          MOV     C,A               ; SAVE
19A1          LDA     PROG_CTRL_REG     ; GET PROGRAM CONTROL FLAGS
19A4          ANA     C                 ; CLEAR THE FLAG
19A5          STA     PROG_CTRL_REG     ; SAVE.
19A8          JMP     CFLAG_SUB_ACT     ; SEE IF MORE FLAGS TO CLEAR
19AB  ;
19AB  ; CONTROL FILE (CF) KEYWORD WAS RECOGNIZED. GET THE ADDRESS AND DATA AND
19AB  ; CLEAR THE CONTROL FILE.
19AB  ;
19AB  CCF_SUB_ACT:
19AB          DAD     B                 ; POINT AT THE ADDRESS
19AC          MOV     A,M               ; GET THE ADDRESS
19AD          STA     SETCLR_CF_ADDR    ; SAVE
19B0          INX     H
19B1          DAD     B                 ; POINT AT THE DATA
```

```
19B2  5417          MOV         A,M               ; GET THE DATA
19B3  5418          CALL        BIT_NMB_TO_MSK    ; DECODE THE BIT VALUE
19B6  5419          STA         SETCLR_CF_DATA    ; SAVE
19B9  5420          CALL        CLEAR_DCS_CF      ; CLEAR THE SPECIFIED BITS
19BC  5421 ;
19BC  5422 ; EXIT WITH C BIT CLEAR
19BC  5423 ;
19BC  5424 CLEAR_DONE:
19BC  5425          STC
19BD  5426          CMC
19BE  5427          RET
19BF  5428 ;
19BF  5429 ; EXIT WITH C BIT SET
19BF  5430 ;
19BF  5431 CLEAR_ERR:
19BF  5432          STC
19C0  5433          RET
19C1  5434
```

```
19C1              .SBTTL  "  EXAMINE ACTION ROUTINE
19C1        ;++
19C1        ;
19C1        ; FUNCTIONAL DESCRIPTION:
19C1        ;
19C1        ;       THIS ROUTINE EXECUTES THE MANY FORMS OF THE 'EXAMINE' COMMAND.
19C1        ;       THE FORMS SUPPORTED ARE:
19C1        ;
19C1        ;               EXAMINE RTEMPS:<2F:0>           EXAMINE MTEMPS:<F:0>
19C1        ;               EXAMINE PC      EXAMINE VA      EXAMINE MDR
19C1        ;               EXAMINE PBACK   EXAMINE MA      EXAMINE WDR
19C1        ;               EXAMINE SR:<n>  EXAMINE PSL     EXAMINE ST
19C1        ;               EXAMINE SF
19C1        ;
19C1        ; CALLING SEQUENCE:
19C1        ;
19C1        ;       CALL    EXAMINE_ACTION
19C1        ;
19C1        ; INPUT PARAMETERS:
19C1        ;
19C1        ;       REGISTER PAIR D & E - CONTAINS THE ADDRESS OF THE PARSE TABLE.
19C1        ;
19C1        ; IMPLICIT INPUTS:
19C1        ;
19C1        ;       NONE
19C1        ;
19C1        ; OUTPUT PARAMETERS:
19C1        ;
19C1        ;       NONE
19C1        ;
19C1        ; IMPLICIT OUTPUTS:
19C1        ;
19C1        ;       NONE
19C1        ;
19C1        ; COMPLETION CODES:
19C1        ;
19C1        ;       C BIT SET   - SYNTAX ERROR IN COMMAND
19C1        ;       C BIT CLEAR - COMMAND EXECUTED SUCCESSFULLY
19C1        ;
19C1        ;--
19C1
19C1        EXAMINE_ACTION:
19C1                PUSH            D                 ; SAVE D & E
19C2                WRITE$_DCS      DCS_SCRATCH_ADR+1,-; INIT DCS SCRATCH LOCATION
19C2                                NOP_MICRO_WORD,1; ...
19D5                LXI             H,0               ; INIT 'EXAMINE_TEMP'
19D8                SHLD            EXAMINE_TEMP      ; ...
19DB                SHLD            EXAMINE_TEMP+2    ; ...
19DE                POP             H                 ; PUT PARSE TABLE ADDRESS IN H & L
19DF                MVI             B,0               ; PUT A CONSTANT 3 IN B & C REG
19E1                MVI             C,3               ; ...
19E3                MOV             A,M               ; GET KEYWORD CODE
19E4                ANI             <^X7F>            ; DISCARD KEYWORD FLAG BIT
19E6                INX             H                 ; POINT AT NEXT ENTRY IN PARSE TBL
19E7                CPI             <^X55>            ; END OF PARSE TABLE?
```

```
19E9            JZ          EXAMINE_ERR     ; BRANCH IF YES
19EC            CPI         RT_KEY          ; RTEMP?
19EE            JZ          EXAMINE_RT
19F1            CPI         MT_KEY          ; MTEMP?
19F3            JZ          EXAMINE_MT
19F6            CPI         PC_KEY          ; PC?
19F8            JZ          EXAMINE_PC
19FB            CPI         VA_KEY          ; VA?
19FD            JZ          EXAMINE_VA
1A00            CPI         MD_KEY          ; MDR?
1A02            JZ          EXAMINE_MD
1A05            CPI         PB_KEY          ; PC BACKUP?
1A07            JZ          EXAMINE_PB
1A0A            CPI         MA_KEY          ; MA?
1A0C            JZ          EXAMINE_MA
1A0F            CPI         WD_KEY          ; WDR?
1A11            JZ          EXAMINE_WD
1A14            CPI         SR_KEY          ; STATUS & CONTROL?
1A16            JZ          EXAMINE_SR
1A19            CPI         PS_KEY          ; PSL?
1A1B            JZ          EXAMINE_PS
1A1E            CPI         ST_KEY          ; STEP COUNTER?
1A20            JZ          EXAMINE_ST
1A23            CPI         SF_KEY          ; FLAGS?
1A25            JZ          EXAMINE_SF
1A28            JMP         EXAMINE_ERR     ; INVALID KEYWORD
1A2B    ;
1A2B    ; EXAMINE R TEMPS
1A2B    ;
1A2B    EXAMINE_RT:
1A2B            MOV         A,M             ; GET NEXT ELEMENT IN PARSE TABLE
1A2C            CPI         <^X55>          ; REGISTER NUMBER SPECIFIED?
1A2E            JZ          EXAMINE_ERR     ; BRANCH IF NO
1A31            DAD         B               ; POINT AT NUMBER
1A32            MOV         A,M             ; ...
1A33            CALL        READ_RTEMP      ; GET CONTENTS OF REGISTER
1A36            JMP         EX_DISP_WH      ; DISPLAY WH REGISTER
1A39    ;
1A39    ; EXAMINE M TEMPS
1A39    ;
1A39    EXAMINE_MT:
1A39            MOV         A,M             ; GET NEXT ELEMENT IN PARSE TABLE
1A3A            CPI         <^X55>          ; REGISTER NUMBER SPECIFIED?
1A3C            JZ          EXAMINE_ERR     ; BRANCH IF NO
1A3F            DAD         B               ; POINT AT ADDRESS
1A40            MOV         A,M             ; ...
1A41    MTEMP_COMMON:
1A41            CALL        READ_MTEMP      ; GET CONTENTS OF REGISTER
1A44            JMP         EX_DISP_WH      ; DISPLAY WH REGISTER
1A47    ;
1A47    ; EXAMINE PC
1A47    ;
1A47    EXAMINE_PC:
1A47            MVI         A,<^X1A>        ; GET MSRC FIELD VALUE
1A49            JMP         MTEMP_COMMON    ; READ THE REGISTER
```

```
1A4C   ;
1A4C   ; EXAMINE VA:
1A4C   ;
1A4C   EXAMINE_VA:
1A4C            MVI             A,<^X1B>        ; GET MSRC FIELD VALUE
1A4E            JMP             MTEMP_COMMON    ; READ THE REGISTER
1A51   ;
1A51   ; EXAMINE MDR
1A51   ;
1A51   EXAMINE_MD:
1A51            MVI             A,<^X12>        ; GET MSRC FIELD VALUE
1A53            JMP             MTEMP_COMMON    ; READ THE REGISTER
1A56   ;
1A56   ; EXAMINE PC BACKUP
1A56   ;
1A56   EXAMINE_PB:
1A56            MVI             A,<^X19>        ; GET MSRC FIELD VALUE
1A58            JMP             MTEMP_COMMON    ; READ THE REGISTER
1A5B   ;
1A5B   ; EXAMINE MA
1A5B   ;
1A5B   EXAMINE_MA:
1A5B            MVI             A,<^X18>        ; GET MSRC FIELD VALUE
1A5D            JMP             MTEMP_COMMON    ; READ THE REGISTER
1A60   ;
1A60   ; EXAMINE WDR
1A60   ;
1A60   EXAMINE_WD:
1A60            WRITE$_DCS      DCS_SCRATCH_ADR,-
1A60                            RDM_WD_MIC_WORD,1; LOAD THE MICRO WORDS
1A73            EXECUTE         DCS_SCRATCH_ADR,1; EXECUTE THE MICRO INSTRUCTIONS
1A86            JMP             EX_DISP_WH      ; DISPLAY WH REGISTER
1A89   ;
1A89   ; EXAMINE STATUS AND CONTROL
1A89   ;
1A89   EXAMINE_SR:
1A89            MOV             A,M             ; GET NEXT ELEMENT IN PARSE TABLE
1A8A            CPI             <^X55>          ; REGISTER NUMBER SPECIFIED?
1A8C            JZ              EXAMINE_ERR     ; BRANCH IF NO
1A8F            DAD             B               ; POINT AT ADDRESS
1A90            MOV             A,M             ; ...
1A91            STA             WD_REG_BYTE_3   ; PUT IN WD REGISTER
1A94            WRITE$_DCS      DCS_SCRATCH_ADR,-
1A94                            RDM_SR_MIC_WORD,2; LOAD THE MICRO WORDS
1AA7            WRITE$_DCS      DCS_SCRATCH_ADR+2,-
1AA7                            NOP_MICRO_WORD,1; ...
1ABA            EXECUTE         DCS_SCRATCH_ADR,2; EXECUTE THE MICRO INSTRUCTIONS
1ACD            LDA             WH_REG_BYTE_3   ; GET RETURNED DATA
1AD0            MVI             B,15            ; GET MASK
1AD2            JMP             MASK_DISPLAY    ; DISPLAY WH REGISTER
1AD5   ;
1AD5   ; EXAMINE PSL
1AD5   ;
1AD5   EXAMINE_PS:
1AD5            WRITE$_DCS      DCS_SCRATCH_ADR,-
```

```
1AD5                                RDM_PS_MIC_WORD,1; LOAD THE MICRO WORDS
1AE8            EXECUTE             DCS_SCRATCH_ADR,1; EXECUTE THE MICRO INSTRUCTIONS
1AFB            LDA                 WH_REG_BYTE_2    ; GET BYTE 2
1AFE            ANI                 <^XDF>           ; MASK
1B00            STA                 EXAMINE_TEMP+2   ; SAVE FOR DISPLAY
1B03            LDA                 WH_REG_BYTE_3    ; GET BYTE 3
1B06            ANI                 <^XCF>           ; MASK
1B08            STA                 EXAMINE_TEMP+3   ; SAVE FOR DISPLAY
1B0B            LDA                 WH_REG_BYTE_0    ; GET BYTE 0
1B0E            MVI                 B,255            ; GET MASK
1B10            JMP                 MASK_DISPLAY     ; DISPLAY 'EXAMINE_TEMP'
1B13      ;
1B13      ; EXAMINE STEP COUNTER
1B13      ;
1B13      EXAMINE_ST:
1B13            WRITE$_DCS          DCS_SCRATCH_ADR,-
1B13                                RDM_ST_MIC_WORD,1; LOAD THE MICRO WORDS
1B26            EXECUTE             DCS_SCRATCH_ADR,1; EXECUTE THE MICRO INSTRUCTIONS
1B39            LDA                 WH_REG_BYTE_0    ; GET DATA READ
1B3C            MVI                 B,31             ; ONLY NEED BITS<4:0>
1B3E            JMP                 MASK_DISPLAY     ; DISPLAY DATA
1B41      ;
1B41      ; EXAMINE FLAGS
1B41      ;
1B41      EXAMINE_SF:
1B41            WRITE$_DCS          DCS_SCRATCH_ADR,-
1B41                                RDM_SF_MIC_WORD,1; LOAD THE MICRO WORDS
1B54            EXECUTE             DCS_SCRATCH_ADR,1; EXECUTE THE MICRO INSTRUCTIONS
1B67            LDA                 WH_REG_BYTE_0    ; GET DATA READ
1B6A            MVI                 B,63             ; ONLY NEED BITS<5:0>
1B6C      ;     JMP                 MASK_DISPLAY     ; DISPLAY DATA
1B6C
1B6C      ;
1B6C      ; MASK AND DISPLAY A REGISTER
1B6C      ;
1B6C      MASK_DISPLAY:
1B6C            ANA                 B                ; MASK THE DATA
1B6D            STA                 EXAMINE_TEMP     ; SAVE
1B70            LXI                 H,EXAMINE_TEMP   ; GET ADDRESS OF DATA
1B73            JMP                 EXAMINE_DISPLAY  ; DISPLAY THE DATA
1B76      ;
1B76      ; DISPLAY WH REGISTER
1B76      ;
1B76      EX_DISP_WH:
1B76            LXI                 H,WH_REG_BYTE_0 ; GET ADDRESS OF DATA
1B79      ;
1B79      ; DISPLAY REGISTER WHOSE POINTER IS IN H & L
1B79      ;
1B79      EXAMINE_DISPLAY:
1B79            CALL                DISPLAY_DATA
1B7C            JMP                 EXAMINE_EX
1B7F      EXAMINE_ERR:
1B7F            STC
1B80            RET
1B81      EXAMINE_EX:
   1B81  5656              STC
   1B82  5657              CMC
   1B83  5658              RET
   1B84  5659
   1B84  5660 DEPOSIT_ACTION:
   1B84  5661              RET
   1B85  5662
   1B85  5663
```

```
1B85          .SBTTL  " DECODE FLAG NAME ROUTINE
1B85  ;++
1B85  ;
1B85  ; FUNCTIONAL DESCRIPTION:
1B85  ;
1B85  ;       THIS ROUTINE DECODES A PROGRAM FLAG NAME INTO A BIT POSITION.
1B85  ;       THE FLAG NAME MUST BE ONE OF THE FOLLOWING:
1B85  ;
1B85  ;               HALT  LOOP  NER  BELL  SA  IB  QA  TR
1B85  ;
1B85  ; CALLING SEQUENCE:
1B85  ;
1B85  ;       CALL    DECODE_FLAG_NAM
1B85  ;
1B85  ; INPUT PARAMETERS:
1B85  ;
1B85  ;       REGISTER A - CONTAINS THE OFFSET POSITION OF THE FLAG NAME
1B85  ;                    IN THE KEYWORD TABLE.
1B85  ;
1B85  ; IMPLICIT INPUTS:
1B85  ;
1B85  ;       NONE
1B85  ;
1B85  ; OUTPUT PARAMETERS:
1B85  ;
1B85  ;       REGISTER C - CONTAINS THE FLAG BIT DECODED FROM THE NAME
1B85  ;
1B85  ; IMPLICIT OUTPUTS:
1B85  ;
1B85  ;       NONE
1B85  ;
1B85  ; COMPLETION CODES:
1B85  ;
1B85  ;       C BIT SET   - INDICATES AN ILLEGAL FLAG NAME
1B85  ;       C BIT CLEAR - CONVERSION IS COMPLETE
1B85  ;
1B85  ;--
1B85
1B85  DECODE_FLAG_NAM:
1B85          SUI             HALT_KEY        ; IS IT A LEGAL FLAG NAME?
1B87          JM              DECODE_ERR      ; BRANCH IF NO
1B8A          CPI             TR_KEY+2        ; IS IT LEGAL FLAG NAME?
1B8C          JP              DECODE_ERR      ; BRANCH IF NO
1B8F  ;
1B8F  ; CONVERT FLAG NAME OFFSET INTO A BIT POSITION.
1B8F  ;
1B8F          RRC                             ; DIVIDE OFFSET BY 2
1B90          CALL            BIT_NMB_TO_MSK  ; GET THE BIT MASK
1B93  ;
1B93  ; DECODE COMPLETE. RETURN WITH C BIT CLEAR.
1B93  ;
1B93  2$:     MOV             C,A             ; PUT FLAG BIT IN C REGISTER
1B94          STC
1B95          CMC
1B96          RET
```

```
1B97  5720 ;
1B97  5721 ; ILLEGAL FLAG NAME. RETURN WITH C BIT SET.
1B97  5722 ;
1B97  5723 DECODE_ERR:
1B97  5724          STC
1B98  5725          RET
1B99  5726
```

```
1B99              .SBTTL  "THE INTERPRETER
1B99              .SBTTL  "   OP CODE DISPATCH ROUTINE
1B99      ;++
1B99      ;
1B99      ; FUNCTIONAL DESCRIPTION:
1B99      ;
1B99      ;       THIS ROUTINE USES THE TEST PC TO PICKUP AN OPCODE AND ITS
1B99      ;       ASSOCIATED ARGUMENTS AND CALL THE SUBROUTINE SPECIFIED BY THE
1B99      ;       OP CODE.
1B99      ;
1B99      ;       BEGIN
1B99      ;       IF THE 'SINGLE_INSTR' FLAG IS SET
1B99      ;       THEN
1B99      ;         IF THE 'LOSS' FLAG IS SET OR THE 'LOST' FLAG IS SET AND THE
1B99      ;            CURRENT TEST_PC IS EQUAL TO OR GREATER THAN THE
1B99      ;            SPECIFIED TEST PC OR THERE IS NO SPECIFIED
1B99      ;            TEST PC
1B99      ;         THEN CALL THE SINGLE PSEUDO INSTRUCTION ROUTINE
1B99      ;         ELSE DISPATCH
1B99      ;       ELSE DISPATCH
1B99      ;       END
1B99      ;
1B99      ; CALLING SEQUENCE:
1B99      ;
1B99      ;       CALL    OP_CODE_DISP
1B99      ;
1B99      ; INPUT PARAMETERS:
1B99      ;
1B99      ;       NONE
1B99      ;
1B99      ; IMPLICIT INPUTS:
1B99      ;
1B99      ;       TEST_PC         - CONTAINS THE POINTER TO THE CURRENT PSEUDO INSTR
1B99      ;       SPEC_TEST_PC    - CONTAINS THE TEST PC TO BEGIN SINGLE INSTRUCTION
1B99      ;       SOFT_CTRL_REG   - CONTAINS THE LOSS AND LOST FLAGS
1B99      ;       SPEC_TEST_NUMB  - CONTAINS THE TEST NUMBER FOR LOOP IN SPECIAL TEST
1B99      ;       TEST_NUMBER     - CONTAINS THE CURRENT TEST NUMBER
1B99      ;
1B99      ; OUTPUT PARAMETERS:
1B99      ;
1B99      ;       NONE
1B99      ;
1B99      ; IMPLICIT OUTPUTS:
1B99      ;
1B99      ;       NONE
1B99      ;
1B99      ; COMPLETION CODES:
1B99      ;
1B99      ;       NONE
1B99      ;
1B99      ;--
1B99
1B99      OP_CODE_DISP:
1B99      ;
1B99      ; CHECK THE SINGLE INSTRUCTION FLAG
```

```
1B99  ;
1B99            LDA     SOFT_CTRL_REGA   ; GET THE SINGLE INSTRUCTION FLAG
1B9C            ANI     INSTR_FLAG       ; IS IT SET?
1B9E            JZ      1$               ; BRANCH IF NO
1BA1  ;
1BA1  ; CHECK THE 'LOST' FLAG
1BA1  ;
1BA1            LDA     SOFT_CTRL_REG    ; GET THE LOST FLAG
1BA4            ANI     LOST_FLAG        ; IS IT SET?
1BA6            JZ      2$               ; BRANCH IF NO
1BA9  ;
1BA9  ; LOST FLAG IS SET. SEE IF TEST PC
1BA9  ; IS THE SPECIFIED TEST PC.
1BA9  ;
1BA9            LHLD    TEST_PC          ; GET THE CURRENT TEST PC
1BAC            XCHG                     ; SAVE IN D & E
1BAD            LHLD    C_INIT_TEST_PC   ; GET TWO'S COMPLIMENT OF INIT TEST PC
1BB0            DAD     D                ; H & L NOW CONTAIN VIRTUAL TEST PC
1BB1            INX     H                ; ACCOUNT FOR LENGTH BYTE
1BB2            XCHG                     ; SAVE IN D & E
1BB3            LHLD    TITLE_LENGTH     ; GET LENGHT OF TEST TITLE
1BB6            DAD     D
1BB7            XCHG
1BB8            LHLD    SPEC_TEST_PC     ; GET THE SPECIAL TEST PC
1BBB            INR     H                ; HAS ONE BEEN SPECIFIED?
1BBC            JZ      2$               ; BRANCH IF NO
1BBF            DCR     H                ; RESTORE SPECIAL TEST PC
1BC0  6$:       MOV     A,H              ; NOW COMPARE THE SPECIAL TEST
1BC1            CMP     D                ; TEST PC TO THE CURRENT TEST PC
1BC2            JC      2$               ; BRANCH IF GREATER THAN
1BC5            JNZ     1$               ; BRANCH IF NOT THE SAME
1BC8            MOV     A,L              ; ...
1BC9            CMP     E                ; ...
1BCA            JC      2$               ; BRANCH IF GREATER THAN
1BCD            JNZ     1$               ; BRANCH IF LESS THAN
1BD0  2$:       CALL    SINGLE_INSTR     ; GO TO THE SINGLE INSTRUCTION ROUTINE
1BD3            CC      GET_CMD_LINE     ; CALL CMD PARSER IF NOT A SPACE
1BD6  ;
1BD6  ; NOW PICKUP THE OP CODE AND ARGUMENTS TO THE PSEUDO INSTRUCTION
1BD6  ;
1BD6  1$:       LHLD    TEST_PC          ; GET THE CURRENT TEST PC
1BD9            SHLD    CURRENT_PC       ; SAVE IN CASE IT'S AN IFERROR INSTR
1BDC            MOV     C,M              ; GET THE OP CODE
1BDD            INX     H                ; POINT AT THE ARGUMENT COUNT
1BDE            LXI     D,ARG_LIST       ; GET POINTER TO ARGUMENT LIST
1BE1            MOV     A,M              ; GET THE NUMBER OF ARGUMENTS
1BE2            CPI     MAX_ARGUMENTS+1  ; LESS THAN ARGUMENT LIST?
1BE4            JM      5$               ; BRANCH IF YES
1BE7            MVI     A,MAX_ARGUMENTS  ; SET AT MAX
1BE9  5$:       MOV     B,A              ; PUT ARGUMENT COUNT IN B REG
1BEA            INX     H                ; POINT AT FIRST ARGUMENT
1BEB  3$:       DCR     B                ; DECREMENT ARGUMENT COUNT
1BEC            JM      4$               ; BRANCH IF ALL ARGUMENTS PICKED UP
1BEF            MOV     A,M              ; GET AN ARGUMENT
1BF0            STAX    D                ; SAVE IN ARGUMENT LIST
```

```
1BF1            INX         D               ; INCREMENT ARG LIST POINTER
1BF2            INX         H               ; INCREMENT TEST STREAM POINTER
1BF3            JMP         3$              ; CONTINUE
1BF6      ;
1BF6      ; GOT ALL THE ARGUMENTS. UPDATE THE TEST PC
1BF6      ;
1BF6      4$:   SHLD        TEST_PC         ; SAVE THE CURRENT TEST PC
1BF9      ;
1BF9      ; USE THE OPCODE TO GET THE ADDRESS OF THE APPROPRIATE SUB ROUTINE
1BF9      ; AND CALL THE ROUTINE.
1BF9      ;
1BF9            LXI         H,INSTR_TABLE   ; GET THE ADDRESS OF THE INSTRUCTION TABLE
1BFC            MVI         B,0             ; INIT B REGISTER
1BFE            DAD         B               ; INDEX TO THE ROUTINE ADDRESS
1BFF            MOV         A,M             ; GET LOW BITS OF ROUTINE ADDRESS
1C00            MOV         E,A             ; PUT IN 'E' REG
1C01            INX         H               ; POINT AT UPPER BITS
1C02            MOV         A,M             ; GET UPPER BITS OF ADDRESS
1C03            MOV         D,A             ; PUT IN D REG
1C04            LXI         B,OP_CODE_DISP  ; GET RETURN ADDRESS FOR SIMULATED
1C07            PUSH        B               ; SUB ROUTINE CALL
1C08            XCHG                        ; PUT ROUTINE ADDRESS IN H & L
1C09            PCHL                        ; JMP TO THE ROUTINE
1C0A
```

```
1C0A                      .SBTTL  "  LOAD DCS ROUTINE
1C0A            ;++
1C0A            ;
1C0A            ; FUNCTIONAL DESCRIPTION:
1C0A            ;
1C0A            ;         THIS ROUTINE INTERPRETS THE 'LOADDCS' PSEUDO INSTRUCTION.
1C0A            ;         IT MOVES <WRD CNT> NUMBER OF MICRO WORDS STARTING AT <SRC ADR>
1C0A            ;         (INDEXED BY <SRC INDEX> TIMES THE <WRD CNT>) TO
1C0A            ;         DCS STARTING AT <DCS ADR>. IF <SRC
1C0A            ;         INDEX> IS SPECIFIED, <SRC ADR> IS INDEXED BY MICRO WORDS (11 BYTES)
1C0A            ;         TIMES THE CURRENT VALUE-1 OF THE INDEX. THIS ROUTINE CALLS THE
1C0A            ;         'WRITE_DCS' ROUTINE.
1C0A            ;
1C0A            ; CALLING SEQUENCE:
1C0A            ;
1C0A            ;         LOADDCS <SRC ADR>,[<SRC INDEX>],<DCS ADR>,<WRD CNT>
1C0A            ;
1C0A            ; IMPLICIT INPUTS:
1C0A            ;
1C0A            ;         ARGUMENT LIST:
1C0A            ;
1C0A            ;                 OFFSET 0 - SOURCE ADDRESS LOW
1C0A            ;                 OFFSET 1 - SOURCE ADDRESS HIGH
1C0A            ;                 OFFSET 2 - SOURCE INDE CODE
1C0A            ;                 OFFSET 3 - DCS ADDRESS
1C0A            ;                 OFFSET 4 - MICRO WORD COUNT
1C0A            ;
1C0A            ; OUTPUT PARAMETERS:
1C0A            ;
1C0A            ;         NONE
1C0A            ;
1C0A            ; IMPLICIT OUTPUTS:
1C0A            ;
1C0A            ;         NONE
1C0A            ;
1C0A            ; COMPLETION CODES:
1C0A            ;
1C0A            ;         NONE
1C0A            ;
1C0A            ;--
1C0A
1C0A            LOAD_DCS:
1C0A                      LDA             ARG_LIST+2        ; GET THE SRC INDEX BYTE
1C0D                      CALL            GET_INDEX_VALUE   ; GET THE CURRENT INDEX VALUE
1C10            ;
1C10            ; NOW MULTIPLY BY THE WORD COUNT
1C10            ;
1C10                      LDA             ARG_LIST+4        ; GET THE WORD COUNT
1C13                      CALL            MULTIPLY_A_BC
1C16                      MOV             B,D               ; PUT RESULT IN B & C
1C17                      MOV             C,E               ; ...
1C18            ;
1C18            ; INDEX VALUE IS IN B REGISTER. GET THE SRC ADDRESS AND INDEX IT.
1C18            ;
1C18                      MVI             A,11              ; MULTIPLY 11 BYTES TIMES THE
```

```
1C1A  5918          CALL            MULTIPLY_A_BC    ; CURRENT INDEX VALUE
1C1D  5919          LHLD            ARG_LIST         ; GET THE SOURCE DATA ADDRESS
1C20  5920          DAD             D                ; INDEX IT
1C21  5921          XCHG
1C22  5922          LHLD            INIT_TEST_PC     ; GET INITIAL TEST PC
1C25  5923          DAD             D                ; RELOCATE SOURCE ADDRESS
1C26  5924 ;
1C26  5925 ; INDEXED SRC ADDRESS IS IN H & L. GET THE DCS ADDRESS AND WORD COUNT
1C26  5926 ;
1C26  5927 3$:     LDA             ARG_LIST+4       ; GET THE WORD COUNT
1C29  5928          MOV             B,A              ; PUT IN 'B' REG FOR MACRO CALL
1C2A  5929          LDA             ARG_LIST+3       ; GET THE DCS ADDRESS
1C2D  5930          WRITE$_DCS                       ; WRITE THE DCS
1C3A  5931          RET                              ; EXIT
1C3B  5932
```

```
1C3B          .SBTTL  " NEW TEST ROUTINE
1C3B    ;++
1C3B    ;
1C3B    ; FUNCTIONAL DESCRIPTION:
1C3B    ;
1C3B    ;       IF THE CONTROL_C_FLAG IS SET
1C3B    ;       THEN CALL THE GET_CMD_LINE ROUTINE
1C3B    ;       ELSE
1C3B    ;          IF THE TSTSPAN_FLAG IS SET AND THE LAST TEST WAS THE SPECIFIED
1C3B    ;             TEST_SPAN_END TEST
1C3B    ;          THEN CALL THE GET_CMD_LINE ROUTINE
1C3B    ;          ELSE TAKE THE NORMAL EXIT
1C3B    ;
1C3B    ;       IF THE LOST_FLAG IS SET
1C3B    ;       THEN
1C3B    ;         IF THE LAST TEST WAS THE SPECIFIED SPEC_TEST_NUMB
1C3B    ;         THEN
1C3B    ;           IF THE CONT_FLAG IS CLEAR
1C3B    ;           THEN
1C3B    ;             SET THE TEST_PC TO THE VALUE OF LOOP_ADDRESS
1C3B    ;             CLEAR THE ERROR_FLAG AND LOOP_ERR_FLAG AND EXIT
1C3B    ;           ELSE CLEAR THE LOST_FLAG AND CONT_FLAG AND TAKE NORMAL EXIT
1C3B    ;         ELSE
1C3B    ;
1C3B    ;   NORMAL EXIT:
1C3B    ;       ELSE SAVE THE TEST_PC IN LOOP_ADDRESS AND ERROR_LOOP_ADDR
1C3B    ;               CLEAR THE ERROR_FLAG AND LOOP_ERROR_FLAG
1C3B    ;               UPDATE THE TEST_NUMBER, AND EXIT
1C3B    ;
1C3B    ; CALLING SEQUENCE:
1C3B    ;
1C3B    ;       NEWTEST
1C3B    ;
1C3B    ; IMPLICIT INPUTS:
1C3B    ;
1C3B    ;       ARGUMENT LIST:
1C3B    ;
1C3B    ;               OFFSET 0 - THE TEST NUMBER
1C3B    ;
1C3B    ;       TEST_PC         - CONTAINS THE CURRENT TEST PC
1C3B    ;       TEST_NUMBER     - CONTAINS THE TEST NUMBER OF THE PREVIOUS TEST
1C3B    ;       SPEC_TEST_NUMB  - CONTAINS THE TEST NUMBER REQUEST FOR 'LOOP ON TEST'
1C3B    ;       TEST_SPAN_END   - CONTAINS THE ENDING TEST NUMBER FOR A 'TEST SPAN' FUNCTION
1C3B    ;       LOOP_ADDRESS    - CONTAINS THE 'LOOP ADDRESS' OF THE PREVIOUS TEST
1C3B    ;       ERROR_LOOP_ADDR - CONTAINS THE 'ERROR LOOP ADDRESS' OF THE PREVIOUS TEST
1C3B    ;
1C3B    ; IMPLICIT OUTPUTS:
1C3B    ;
1C3B    ;       LOOP_ADDRESS    - CONTAINS THE 'LOOP ADDRESS' OF THE CURRENT TEST
1C3B    ;       ERROR_LOOP_ADDR - CONTAINS THE 'ERROR LOOP ADDRESS' OF THE CURRENT TEST
1C3B    ;       TEST_NUMBER     - CONTAINS THE TEST NUMBER OF THE CURRENT TEST
1C3B    ;
1C3B    ; COMPLETION CODES:
1C3B    ;
1C3B    ;       NONE
```

```
1C3B  ;
1C3B  ;--
1C3B
1C3B  NEW_TEST:
1C3B  ;
1C3B  ; CHECK THE CONTROL_C_FLAG
1C3B  ;
1C3B            LDA             SOFT_CTRL_REG    ; GET THE CONTROL C FLAG
1C3E            ANI             CONTROL_C_FLAG   ; IS IT SET?
1C40            CNZ             GET_CMD_LINE     ; IF YES, CALL THE COMMAND PARSER
1C43  ;
1C43  ; INITIALIZE THE SUB TEST NUMBER
1C43  ;
1C43            XRA             A                ; GET INITIAL DATA
1C44            STA             SUB_TEST_NUMB    ; INIT THE SUB TEST NUMBER
1C47  ;
1C47  ; INITIALIZE THE CURRENT VALUE OF THE LOOP NAMES
1C47  ;
1C47            CALL            INIT_LOOP_VALUE ;
1C4A  ;
1C4A  ; INITIALIZE THE TRAP FLAG
1C4A  ;
1C4A            LDA             SOFT_CTRL_REGA   ; GET THE FLAGS REGISTER
1C4D            ANI             <^CEXP_UTRAP_FLAG> ; CLEAR THE FLAG
1C4F            STA             SOFT_CTRL_REGA   ; RESTORE THE FLAGS REGISTER
1C52  ;
1C52  ; CHECK THE TSTSPAN_FLAG
1C52  ;
1C52            LDA             SOFT_CTRL_REG    ; GET THE TSTSPAN_FLAG
1C55            ANI             TSTSPAN_FLAG     ; IS IT SET?
1C57            JZ              1$               ; BRANCH IF NO
1C5A            LDA             TEST_NUMBER      ; GET THE NUMBER OF THE LAST TEST
1C5D            MOV             B,A              ; SAVE
1C5E            LDA             TEST_SPAN_END    ; GET THE SPECIFIED END TEST NUMBER
1C61            CMP             B                ; SAME AS LAST TEST NUMBER?
1C62            JNZ             40$              ; BRANCH IF NO
1C65            CALL            GET_CMD_LINE     ; CALL THE COMMAND PARSER
1C68            LDA             SOFT_CTRL_REG    ; CLEAR THE TEST SPAN FLAG
1C6B            ANI             <^C<TSTSPAN_FLAG!- ; AND THE LOST FLAG
1C6B                            LOST_FLAG>>      ; ...
1C6D            STA             SOFT_CTRL_REG    ; ...
1C70            JMP             40$              ; IF CONTINUE CMD, TAKE NORMAL EXIT
1C73  ;
1C73  ; CHECK THE 'LOST_FLAG'
1C73  ;
1C73  1$:       LDA             SOFT_CTRL_REG    ; GET THE 'LOST_FLAG'
1C76            ANI             LOST_FLAG        ; IS IT SET?
1C78            JZ              40$              ; BRANCH IF NO
1C7B  ;
1C7B  ; CHECK THE LAST TEST NUMBER AGAINST SPECIFIED NUMBER
1C7B  ;
1C7B            LDA             TEST_NUMBER      ; GET THE NUMBER OF THE LAST TEST
1C7E            MOV             B,A              ; SAVE
1C7F            LDA             SPEC_TEST_NUMB   ; GET THE SPECIFIED TEST NUMBER
1C82            CMP             B                ; IS LAST TEST SPECIFIED TEST?
```

```
1C83             JNZ        40$                  ; BRANCH IF NO
1C86       ;
1C86       ; LAST TEST WAS SPECIFIED TEST. CHECK IF 'CONT_FLAG' IS SET.
1C86       ;
1C86             LDA        SOFT_CTRL_REG        ; GET THE CONT_FLAG
1C89             ANI        CONT_FLAG            ; IS IT SET?
1C8B             JNZ        5$                   ; BRANCH IF YES
1C8E             LHLD       LOOP_ADDRESS         ; GET THE LOOP ADDRESS
1C91             SHLD       TEST_PC              ; SET THE TEST PC
1C94             LDA        SOFT_CTRL_REGA       ; CLEAR THE ERROR AND LOOP
1C97             ANI        <^C<LOOP_FLAG!-      ; ERROR FLAGS
1C97                        LOOP_ERROR_FLAG>>    ; ...
1C99             STA        SOFT_CTRL_REGA       ; ...
1C9C             JMP        45$                  ; EXIT
1C9F       ;
1C9F       ; CONT_FLAG WAS SET
1C9F       ;
1C9F       5$:   LDA        SOFT_CTRL_REG        ; GET THE LOST_FLAG & CONT_FLAG
1CA2             ANI        <^C<LOST_FLAG!-      ; CLEAR THEM
1CA2                        CONT_FLAG>>          ; ...
1CA4             STA        SOFT_CTRL_REG        ; ...
1CA7       ;
1CA7       ; NORMAL EXIT
1CA7       ;
1CA7       40$:  LHLD       TEST_PC              ; GET THE TEST PC
1CAA             SHLD       LOOP_ADDRESS         ; SAVE AS THE LOOP ADDRESS
1CAD             SHLD       ERROR_LOOP_ADDR      ; AND THE ERROR LOOP ADDRESS
1CB0             LDA        SOFT_CTRL_REGA       ; GET THE ERROR FLAG AND LOOP ERROR FLAG
1CB3             ANI        <^C<ERROR_FLAG!-     ; CLEAR THEM
1CB3                        LOOP_ERROR_FLAG>>    ; ...
1CB5             STA        SOFT_CTRL_REGA       ; ...
1CB8             LDA        ARG_LIST             ; GET THE TEST NUMBER
1CBB             STA        TEST_NUMBER          ; SAVE
1CBE       45$:  RET                             ; EXIT
1CBF
```

```
1CBF          .SBTTL  "  INITIALIZE ROUTINE
1CBF  ;++
1CBF  ;
1CBF  ; FUNCTIONAL DESCRIPTION:
1CBF  ;
1CBF  ;       THIS ROUTINE INITIALIZES THE 11/750 CPU. THIS IS DONE BY SETTING
1CBF  ;       THE 'DC LOW' BIT IN THE RD CONTROL REGISTER WITH MICRO ADDRESS
1CBF  ;       INHIBIT ASCERTED.
1CBF  ;
1CBF  ;       THE RDM CONTROL FILE IS CLEARED, AND THE MASTER_HALT_ENABLE IS
1CBF  ;       TOGGLED. THE V BUS LOAD BIT IS ALSO CLEARED.
1CBF  ;
1CBF  ; CALLING SEQUENCE:
1CBF  ;
1CBF  ;       INITIALIZE
1CBF  ;
1CBF  ; IMPLICIT INPUTS:
1CBF  ;
1CBF  ;       NONE
1CBF  ;
1CBF  ; IMPLICIT OUTPUTS:
1CBF  ;
1CBF  ;       NONE
1CBF  ;
1CBF  ; COMPLETION CODES:
1CBF  ;
1CBF  ;       NONE
1CBF  ;
1CBF  ;--
1CBF
1CBF  INIT_THE_WORLD:
1CBF          CALL            CPU_INIT         ; INIT THE MAINT REG, FORCE DC LOW,
1CC2                                           ; AND FETCH CS LOCATION 0
1CC2  ;
1CC2  ; CHECK IF IN REMOTE MODE
1CC2  ;
1CC2          LDA             FRONT_PNL_2      ; GET REMOTE BIT
1CC5          ANI             REMOTE           ; ...
1CC7          JZ              10$              ; BRANCH IF NOT REMOTE
1CCA          MVI             A,MAINT_DCS_ENABL; GET DATA TO ENABLE CLOCK INTERRUPTS
1CCC          STA             MAINT_REG        ; ENABLE THE RDM CLOCK
1CCF
1CCF  10$:    CALL            DISABLE_CMI      ; DISABLE THE CMI
1CD2          CALL            FETCH_0          ; FETCH CS LOCATION 0
1CD5          LDA             FRONT_PNL_2      ; GET THE REMOTE, FAULT, AND TEST BITS
1CD8          ANI             <REMOTE!FAULT!TEST> ;...
1CDA          ORI             <MASTER_HALT_EN!- ; SET MASTER HALT ENABLE
1CDA                          TRAP_HALT_EN>    ; AND TRAP HALT ENABLE
1CDC          STA             RD_CTRL_REG      ; ...
1CDF          RET                              ; EXIT
1CE0
```

```
1CE0                .SBTTL  "  LOOP ROUTINE
1CE0        ;++
1CE0        ;
1CE0        ; FUNCTIONAL DESCRIPTION:
1CE0        ;
1CE0        ;       THIS ROUTINE INITIALIZES THE 'LOOP_NAME' PARAMETERS TO THE
1CE0        ;       'STARTING_VALUE', 'END_VALUE', AND 'INCREMENT_VALUE'. IT ALSO
1CE0        ;       SAVES THE CURRENT TEST_PC FOR USE BY THE "ENDLOOP" ROUTINE.
1CE0        ;
1CE0        ;       THE 'INCREMENT_VALUE' IS SET TO +1 IF THE STARTING VALUE IS LESS
1CE0        ;       THAN THE ENDING VALUE OTHERWISE IT IS SET TO -1.
1CE0        ;
1CE0        ; CALLING SEQUENCE:
1CE0        ;
1CE0        ;       LOOP    LOOP_NAME,STARTING_VALUE,ENDING_VALUE
1CE0        ;
1CE0        ; IMPLICIT INPUTS:
1CE0        ;
1CE0        ;       ARGUMENT LIST:
1CE0        ;
1CE0        ;               OFFSET 0 - INDEX NAME CODE
1CE0        ;               OFFSET 1 - STARTING VALUE LOW
1CE0        ;               OFFSET 2 - STARTING VALUE HIGH
1CE0        ;               OFFSET 3 - ENDING VALUE LOW
1CE0        ;               OFFSET 4 - ENDING VALUE HIGH
1CE0        ;
1CE0        ;       INDEX_NAM_TBL   - TABLE THAT CONTAINS THE ADDRESS OF THE I, J, OR
1CE0        ;                         K INDEX TABLES.
1CE0        ;       TEST_PC         - CONTAINS THE CURRENT TEST PC.
1CE0        ;
1CE0        ; IMPLICIT OUTPUTS:
1CE0        ;
1CE0        ;       THE I, J, OR K INDEX TABLE IS INITIALIZED.
1CE0        ;
1CE0        ;--
1CE0
1CE0        BEGIN_LOOP:
1CE0                LDA             ARG_LIST        ; GET THE INDEX NAME BYTE
1CE3                CALL            GET_INDEX_VALUE ; GET ADDRESS OF INDEX TABLE
1CE6                LXI             D,ARG_LIST+1    ; GET ADDRESS OF STARTING VALUE
1CE9                MVI             B,4             ; SET A LOOP COUNT
1CEB        10$:    LDAX            D               ; GET LOW BYTE
1CEC                MOV             M,A             ; PUT IN TABLE
1CED                INX             D               ; INCREMENT POINTERS
1CEE                INX             H               ; ...
1CEF                DCR             B               ; DONE 4 BYTES?
1CF0                JNZ             10$             ; BRANCH IF NO
1CF3        ;
1CF3        ; NOW CALCULATE THE INCREMENT VALUE
1CF3        ;
1CF3                LDA             ARG_LIST+2      ; GET HIGH BYTE OF START VALUE
1CF6                MOV             B,A             ; SAVE
1CF7                LDA             ARG_LIST+4      ; GET HIGH BYTE OF END VALUE
1CFA                CMP             B               ; IS START LESS THAN END?
1CFB                JNZ             20$             ; NOT EQUAL, CHECK IF POSITIVE OR NEG
```

```
1CFE  ;
1CFE  ; HIGH BYTES ARE EQUAL, CHECK THE LOW BYTES
1CFE  ;
1CFE            LDA            ARG_LIST+1          ; GET START LOW BYTE
1D01            MOV            B,A                 ; SAVE
1D02            LDA            ARG_LIST+3          ; GET END LOW BYTE
1D05            CMP            B                   ; IS START LESS THAN END?
1D06            JNC            1$                  ; BRANCH IF YES
1D09            JZ             1$                  ; ...
1D0C  ;
1D0C  ; END IS LESS THAN START. SET THE INCREMENT VALUE TO MINUS 1
1D0C  ;
1D0C  3$:       MVI            A,<^XFF>            ; GET INCREMENT VALUE
1D0E            MOV            M,A                 ; PUT IN INDEX TABLE
1D0F            JMP            2$                  ; GO PUT TEST PC IN TABLE
1D12  ;
1D12  ; START IS LESS THAN END. SET THE INCREMENT VALUE TO PLUS 1
1D12  1$:       MVI            A,1                 ; GET THE INCREMENT VALUE
1D14            MOV            M,A                 ; PUT IN INDEX TABLE
1D15            JMP            2$                  ; GO GET THE TEST PC
1D18  ;
1D18  ; HIGH BYTES ARE NOT EQUAL, BRANCH TO APPROPRIATE SECTION TO SET INCREMENT VALUE
1D18  ;
1D18  20$:      JNC            1$                  ; START IS LESS THAN END
1D1B            JMP            3$                  ; END IS LESS THAN START
1D1E  ;
1D1E  ; NOW GET THE TEST PC AND PUT IT IN THE INDEX TABLE
1D1E  ;
1D1E  2$:       INX            H                   ; POINT AT LOW BYTE OF TEST PC FIELD
1D1F            LDA            TEST_PC             ; GET LOW BYTE OF TEST PC
1D22            MOV            M,A                 ; PUT IN INDEX TABLE
1D23            INX            H                   ; POINT AT HIGH BYTE FIELD
1D24            LDA            TEST_PC+1           ; GET HIGH BYTE OF TEST PC
1D27            MOV            M,A                 ; PUT IN INDEX TABLE
1D28            RET                                ; EXIT
1D29
```

```
1D29          .SBTTL  "  END LOOP ROUTINE
1D29    ;++
1D29    ;
1D29    ; FUNCTIONAL DESCRIPTION:
1D29    ;
1D29    ;       THE ERROR LOOP FLAG IS CLEARED
1D29    ;
1D29    ;       THE ROUTINE ADDS THE 'LOOP_INCREMENT' VALUE TO THE CURRENT VALUE
1D29    ;       OF THE SPECIFIED LOOP_NAME.
1D29    ;       THEN
1D29    ;          IF THE LOOP INCREMENT VALUE IS +1
1D29    ;          THEN
1D29    ;            IF THE CURRENT VALUE IS LESS THAN OR EQUAL TO THE END VALUE
1D29    ;            THEN LOAD THE TEST_PC WITH THE LOOP TEST PC AND EXIT
1D29    ;            ELSE SET THE CURRENT VALUE TO MINUS ONE AND EXIT
1D29    ;          ELSE
1D29    ;            IF THE CURRENT VALUE IS GREATER THAN OR EQUAL TO THE END VALUE
1D29    ;            THEN LOAD THE TEST_PC WITH THE LOOP TEST PC AND EXIT
1D29    ;            ELSE SET THE CURRENT VALUE TO MINUS ONE AND EXIT
1D29    ;
1D29    ; CALLING SEQUENCE:
1D29    ;
1D29    ;       ENDLOOP LOOP_NAME
1D29    ;
1D29    ; IMPLICIT INPUTS:
1D29    ;
1D29    ;       ARGUMENT LIST:
1D29    ;
1D29    ;               OFFSET 0 - INDEX NAME CODE
1D29    ;
1D29    ;       I, J, OR K INDEX TABLE
1D29    ;
1D29    ; IMPLICIT OUTPUTS:
1D29    ;
1D29    ;       TEST_PC - CONTAINS THE ADDRESS OF THE NEXT PSEUDO INSTRUCTION TO EXECUTE
1D29    ;
1D29    ;--
1D29
1D29    END_LOOP:
1D29            LDA             SOFT_CTRL_REGA  ; CLEAR THE LOOP ERROR FLAG
1D2C            ANI             <^CLOOP_ERROR_FLAG>; ...
1D2E            STA             SOFT_CTRL_REGA  ; ...
1D31            LDA             ARG_LIST        ; GET THE INDEX CODE
1D34            CALL            GET_INDEX_VALUE ; GET CURRENT VALUE AND TABLE ADDRESS
1D37            INX             H               ; POINT AT THE INCREMENT VALUE
1D38            INX             H               ; ...
1D39            INX             H               ; ...
1D3A            INX             H               ; ...
1D3B            XCHG                            ; PUT IN D & E REG'S
1D3C            LDAX            D               ; GET THE INCREMENT VALUE
1D3D            MOV             L,A             ; SAVE IN L REGISTER
1D3E            MVI             H,0             ; INIT H FOR POSITIVE INCREMENT
1D40            ORA             A               ; IS INCREMENT POSITIVE?
1D41            JP              20$             ; BRANCH IF YES
1D44            MVI             H,<^XFF>        ; SET H FOR NEGATIVE INCREMENT
```

```
1D46  20$:    DAD     B               ; ADD TO CURRENT VALUE
1D47          DCX     D               ; POINT AT CURRENT VALUE
1D48          DCX     D               ; ...
1D49          DCX     D               ; ...
1D4A          MOV     A,H             ; SAVE NEW CURRENT VALUE IN TABLE
1D4B          STAX    D               ; ...
1D4C          DCX     D               ; ...
1D4D          MOV     A,L             ; ...
1D4E          STAX    D               ; ...
1D4F  ;
1D4F  ; NOW SEE IF THE LOOP IS DONE
1D4F  ;
1D4F          INX     D               ; POINT AT END VALUE
1D50          INX     D               ; ...
1D51          INX     D               ; HIGH BYTE
1D52          LDAX    D               ; GET HIGH BYTE OF END VALUE
1D53          INX     D               ; POINT AT INCREMENT VALUE
1D54          CMP     H               ; COMPARE WITH NEW CURRENT VALUE HIGH BYTE
1D55          PUSH    $PSW            ; SAVE CONDITION CODES
1D56          JNZ     30$             ; BRANCH IF NOT EQUAL
1D59          POP     $PSW            ; THROW AWAY CONDITION CODES
1D5A  ;
1D5A  ; HIGH BYTES ARE EQUAL. CHECK THE LOW BYTES
1D5A  ;
1D5A          DCX     D               ; POINT AT LOW BYTE OF END VALUE
1D5B          DCX     D               ; ...
1D5C          LDAX    D               ; GET THE LOW BYTE
1D5D          INX     D               ; POINT AT INCREMENT VALUE
1D5E          INX     D               ; ...
1D5F          CMP     L               ; COMPARE WITH LOW BYTE OF CURRENT VALUE
1D60          PUSH    $PSW            ; SAVE THE CONDITION CODES
1D61  ;
1D61  ; SEE IF INCREMENT VALUE IS POSITIVE OR NEGATIVE
1D61  ;
1D61  30$:    LDAX    D               ; GET THE INCREMENT VALUE
1D62          ORA     A               ; SET THE CONDITION CODES
1D63          JM      1$              ; BRANCH IF INCREMENT WAS NEGATIVE
1D66  ;
1D66  ; INCREMENT VALUE IS +1, SEE IF LOOP IS DONE
1D66  ;
1D66          POP     $PSW            ; GET COND CODES FROM COMPARE
1D67          JC      10$             ; BRANCH IF LOOP IS DONE
1D6A          JMP     5$              ; SET THE LOOP TEST PC
1D6D  ;
1D6D  ; INCREMENT VALUE IS -1, SEE IF LOOP IS DONE
1D6D  ;
1D6D  1$:     POP     $PSW            ; GET CONDITION CODES FROM COMPARE
1D6E          JNC     10$             ; BRANCH IF LOOP IS DONE
1D71  ;
1D71  ; SET THE TEST_PC TO THE PC IN THE LOOP TABLE
1D71  ;
1D71  5$:     INX     D               ; POINT AT LOW BYTE OF TEST PC
1D72          LDAX    D               ; GET THE LOW BYTE
1D73          STA     TEST_PC         ; PUT IN TEST PC
1D76          INX     D               ; POINT AT HIGH BYTE OF TEST PC
```

```
1D77         LDAX       D                 ; GET IT
1D78         STA        TEST_PC+1         ; PUT IN TEST PC
1D7B         RET                          ; EXIT
1D7C    ;
1D7C    ; LOOP IS DONE, SET THE CURRENT VALUE TO MINUS ONE
1D7C    ;
1D7C    10$:  LDA        ARG_LIST          ; GET THE LOOP NAME CODE
1D7F         CALL       GET_INDEX_VALUE   ; GET POINTER TO CURRENT VALUE
1D82         MVI        A,-1              ; GET DATA TO SET IN CURRENT VALUE
1D84         MOV        M,A               ; SET THE CURRENT VALUE TO MINUS ONE
1D85         INX        H                 ; ...
1D86         MOV        M,A               ; ...
1D87         RET                          ; EXIT
1D88
```

```
1D88          .SBTTL  "  ERROR LOOP ROUTINE
1D88   ;++
1D88   ;
1D88   ; FUNCTIONAL DESCRIPTION:
1D88   ;
1D88   ;      THIS ROUTINE SAVES THE CURRENT TEST PC IN LOCATION 'ERROR_LOOP_ADDR'.
1D88   ;      THIS TEST PC IS USED WHEN PERFORMING A 'SCOPE LOOP'.
1D88   ;
1D88   ; CALLING SEQUENCE:
1D88   ;
1D88   ;      ERRLOOP
1D88   ;
1D88   ; IMPLICIT INPUTS:
1D88   ;
1D88   ;      TEST_PC          - CONTAINS THE CURRENT TEST PC
1D88   ;
1D88   ; IMPLICIT OUTPUTS:
1D88   ;
1D88   ;      ERROR_LOOP_ADDR  - GETS LOADED WITH THE CURRENT TEST PC
1D88   ;
1D88   ;--
1D88
1D88   ERROR_LOOP:
1D88           LHLD            TEST_PC            ; GET THE TEST PC
1D8B           SHLD            ERROR_LOOP_ADDR    ; SAVE
1D8E           RET                                ; EXIT
1D8F
```

```
1D8F             .SBTTL  "  PATTERN GENERATE ROUTINE
1D8F        ;++
1D8F        ;
1D8F        ; FUNCTIONAL DESCRIPTION:
1D8F        ;
1D8F        ;       THIS ROUTINE IS USED TO SELECT ONE OF SIX TEST PATTERNS AND MOVE
1D8F        ;       THE SELECTED PATTERN TO THE SPECIFIED LOCATIONS. IF 'LONGLIT' IS
1D8F        ;       SPECIFIED, THE PATTERN IS COMPLIMENTED AND PLACED IN BITS 31 THRU
1D8F        ;       62 OF THE MICRO WORD SPECIFIED BY THE DESTINATION ADDRESS.
1D8F        ;       IF DESTINATION_ADDRESS_X IS SPECIFIED, THE PATTERN IS ALSO MOVED
1D8F        ;       TO THIS ADDRESS.
1D8F        ;
1D8F        ;       THE PATTERNS ARE:
1D8F        ;
1D8F        ;               AAAAAAAA
1D8F        ;               55555555
1D8F        ;               33333333
1D8F        ;               0F0F0F0F
1D8F        ;               00FF00FF
1D8F        ;               0000FFFF
1D8F        ;
1D8F        ; CALLING SEQUENCE:
1D8F        ;
1D8F        ;       SA01PAT LOOP_NAME,DESTINATION_ADDRESS,[LONGLIT],[<DESTINATION_ADDRESS_X>]
1D8F        ;
1D8F        ; IMPLICIT INPUTS:
1D8F        ;
1D8F        ;       ARGUMENT LIST:
1D8F        ;
1D8F        ;               OFFSET 0 - INDEX NAME CODE
1D8F        ;               OFFSET 1 - DESTINATION ADDRESS LOW
1D8F        ;               OFFSET 2 - DESTINATION ADDRESS HIGH
1D8F        ;               OFFSET 3 - LONG LITERAL CODE
1D8F        ;
1D8F        ;       PATTERN_ADDRESS - BASE ADDRESS OF THE SIX PATTERNS
1D8F        ;       I, J, OR K INDEX TABLE
1D8F        ;
1D8F        ; IMPLICIT OUTPUTS:
1D8F        ;
1D8F        ;       THE SPECIFIED PATTERN IS MOVED TO THE SPECIFIED ADDRESS
1D8F        ;
1D8F        ;--
1D8F
1D8F        PATTERN_GEN:
1D8F                LDA     ARG_LIST        ; GET THE INDEX CODE
1D92                CALL    GET_INDEX_VALUE ; GET THE INDEX VALUE
1D95        ;
1D95        ; NOW GENERATE THE ADDRESS OF THE PATTERN
1D95        ;
1D95                MVI     A,4             ; MULTIPLY INDEX VALUE BY 4
1D97                CALL    MULTIPLY_A_BC   ; GET INDEX INTO PATTERN TABLE
1D9A                LXI     H,PATTERN_ADDRESS ; GET THE BASE ADDRESS OF THE PATTERNS
1D9D                DAD     D               ; INDEX THE BASE ADDRESS
1D9E        ;
1D9E        ; SEE IF DESTINATION_ADDRESS_X IS SPECIFIED
```

```
1D9E  ;
1D9E          XCHG
1D9F          LHLD            ARG_LIST+4      ; GET OPTIONAL DESTINATION ADDRESS
1DA2          INR             H               ; IS UPPER BYTE MINUS 1?
1DA3          JZ              5$              ; BRANCH IF YES, ADDRESS NOT SPECIFIED
1DA6          DCR             H               ; RESTORE UPPER BYTE OF ADDRESS
1DA7          PUSH            D               ; SAVE PATTERN ADDRESS
1DA8          XCHG
1DA9          LHLD            INIT_TEST_PC    ; GET INITIAL TEST PC
1DAC          DAD             D               ; RELOCATE DESTINATION ADDRESS
1DAD          POP             D               ; RESTORE PATTERN ADDRESS
1DAE  ;
1DAE  ; NOW MOVE THE PATTERN TO THE ALTERNATE DESTINATION
1DAE  ;
1DAE          MVI             B,4             ; SET A BYTE LOOP COUNT
1DB0  3$:     LDAX            D               ; GET BYTE OF PATTERN
1DB1          MOV             M,A             ; PUT IN DESTINATION
1DB2          INX             D               ; INCREMENT ADDRESSES
1DB3          INX             H               ; ...
1DB4          DCR             B               ; DONE 4 BYTES?
1DB5          JNZ             3$              ; BRANCH IF NO
1DB8          DCX             D               ; BACKUP PATTERN ADDRESS
1DB9          DCX             D               ; ...
1DBA          DCX             D               ; ...
1DBB          DCX             D               ; ...
1DBC  ;
1DBC  ; GET THE ADDRESS OF THE DESTINATION
1DBC  ;
1DBC  5$:     PUSH            D               ; SAVE PATTERN ADDRESS
1DBD          LHLD            ARG_LIST+1      ; GET THE DESTINATION ADDRESS
1DC0          XCHG
1DC1          LHLD            INIT_TEST_PC    ; GET INITIAL TEST PC
1DC4          DAD             D               ; RELOCATE DESTINATION ADDRESS
1DC5          POP             D               ; RESTORE PATTERN ADDRESS
1DC6          XCHG                            ; PUT DEST. ADDR IN D & E
1DC7  ;
1DC7  ; SEE IF THE DESTINATION IS THE LONGLIT FIELD OF A MICRO WORD
1DC7  ;
1DC7          LDA             ARG_LIST+3      ; GET THE LONGLIT CODE
1DCA          ORA             A               ; SET THE CONDITION CODES
1DCB          JNZ             10$             ; BRANCH IF LONG LIT
1DCE  ;
1DCE  ; LONG LITERAL WAS NOT SPECIFIED, MOVE THE PATTERN TO THE DESTINATION
1DCE  ;
1DCE          MVI             B,4             ; SET A LOOP COUNT FOR NO. OF BYTES
1DD0  6$:     MOV             A,M             ; GET A BYTE OF THE PATTERN
1DD1          INX             H               ; ...
1DD2          STAX            D               ; PUT IN THE DESTINATION
1DD3          INX             D               ; ...
1DD4          DCR             B               ; DECREMENT LOOP COUNT
1DD5          JNZ             6$              ; BRANCH IF NOT DONE WITH 4 BYTES
1DD8          JMP             20$             ; EXIT
1DDB  ;
1DDB  ; LONG LITERAL WAS SPECIFIED. MUST PUT THE PATTERN IN BITS 31 THRU 62 OF
1DDB  ; THE DESTINATION. THIS IS DONE BY MOVING THE PATTERN TO A TEMPORARY
```

```
1DDB      ; BUFFER, LEFT SHIFTING THE BUFFER 7 BITS, THEN MOVING THE BUFFER TO
1DDB      ; THE DESTINATION.
1DDB      ;
1DDB      ; FIRST COMPLIMENT THE PATTERN FOR THE LONG LIT FIELD
1DDB      ;
1DDB      10$:     MVI       B,4              ; SET THE BYTE COUNT
1DDD               PUSH      D                ; SAVE DESTINATION ADDRESS
1DDE               LXI       D,TEMP_BUFFER    ; GET ADDRESS OF TEMPORARY BUFFER
1DE1      11$:     MOV       A,M              ; GET A BYTE OF THE PATTERN
1DE2               CMA                        ; COMPLIMENT
1DE3               STAX      D                ; PUT IN TEMP BUFFER
1DE4               INX       D
1DE5               INX       H
1DE6               DCR       B                ; DONE 4 BYTES?
1DE7               JNZ       11$              ; BRANCH IF NO
1DEA               POP       D                ; RESTORE DESTINATION ADDRESS
1DEB               LXI       H,TEMP_BUFFER    ; PASS ADDRESS OF PATTERN
1DEE               MVI       B,31             ; PUT STARTING BIT NUMBER IN B REG
1DF0               MVI       C,62             ; AND ENDING BIT NUMBER IN C REG
1DF2               XRA       A                ; CLEAR INHIBIT PARITY FLAG
1DF3               PUSH      $PSW             ; CLEAR INHIBIT SCRAMBLE FLAG
1DF4               CALL      INSERT_FIELD     ; GENERATE THE FIELD
1DF7      ;
1DF7      ; ALL DONE !!
1DF7      ;
1DF7      20$:     RET                        ; EXIT
1DF8
```

```
1DF8          .SBTTL  "  LOAD REGISTER ROUTINE
1DF8    ;++
1DF8    ;
1DF8    ; FUNCTIONAL DESCRIPTION:
1DF8    ;
1DF8    ;       THIS ROUTINE LOADS AN RDM REGISTER WITH THE SPECIFIED DATA.
1DF8    ;
1DF8    ; CALLING SEQUENCE:
1DF8    ;
1DF8    ;       LOADREG REGISTER_ADDRESS,SRC_ADDRESS,[SRC_INDEX],[DATA_TYPE]
1DF8    ;
1DF8    ; IMPLICIT INPUTS:
1DF8    ;
1DF8    ;       ARGUMENT LIST:
1DF8    ;
1DF8    ;               OFFSET 0 - REGISTER ADDRESS LOW
1DF8    ;               OFFSET 1 - REGISTER ADDRESS HIGH
1DF8    ;               OFFSET 2 - SOURCE ADDRESS LOW
1DF8    ;               OFFSET 3 - SOURCE ADDRESS HIGH
1DF8    ;               OFFSET 4 - SOURCE INDEX CODE
1DF8    ;               OFFSET 5 - DATA TYPE CODE
1DF8    ;
1DF8    ; IMPLICIT OUTPUTS:
1DF8    ;
1DF8    ;       NONE
1DF8    ;
1DF8    ;--
1DF8
1DF8    LOAD_REG:
1DF8            LDA             ARG_LIST+4      ; GET THE INDEX CODE
1DFB            CALL            GET_INDEX_VALUE ; GET THE INDEX VALUE
1DFE            LDA             ARG_LIST+5      ; GET THE DATA TYPE
1E01            CALL            MULTIPLY_A_BC   ; MULTIPLY DATA TYPE BY INDEX VALUE
1E04    ;
1E04    ; D & E NOW CONTAIN THE INDEX VALUE FOR THE SPECIFIED DATA TYPE
1E04    ;
1E04    2$:     LHLD            ARG_LIST+2      ; GET THE SRC ADDRESS
1E07            DAD             D               ; INDEX IT
1E08            XCHG
1E09            LHLD            INIT_TEST_PC    ; GET INITIAL TEST PC
1E0C            DAD             D               ; RELOCATE SOURCE ADDRESS
1E0D    ;
1E0D    ; REAL SOURCE ADDRESS IS NOW IN H & L. NOW WRITE THE SPECIFIED REGISTER
1E0D    ; ACCORDING TO THE DATA TYPE. IF THE REGISTER IS THE RD CONTROL REGISTER,
1E0D    ; READ THE FRONT PANEL REGISTER AND OR IN THE STATE OF THE LIGHTS.
1E0D    ;
1E0D            XCHG                            ; PUT SRC ADDR IN D & E
1E0E            LDA             ARG_LIST+5      ; GET THE DATA TYPE
1E11            MOV             B,A             ; SAVE
1E12            LHLD            ARG_LIST        ; GET THE REGISTER ADDRESS
1E15    3$:     LDAX            D               ; GET A BYTE OF DATA
1E16      ;
1E16      ; CHECK IF RD_CTRL_REG
1E16      ;
1E16            PUSH            $PSW            ; SAVE DATA
```

```
1E17  6570         MOV             A,L               ; GET LOW BYTE OF REGISTER ADDRESS
1E18  6571         CPI             <RD_CTRL_REG & 255>
1E1A  6572                                           ; RD CONTROL REG?
1E1A  6573         JNZ             5$                ; BRANCH IF NO
1E1D  6574         LDA             FRONT_PNL_2       ; GET STATE OF LIGHTS
1E20  6575         ANI             <REMOTE!FAULT!TEST>
1E22  6576         MOV             C,A               ; ...
1E23  6577         POP             $PSW              ; GET DATA THAT IS BEING WRITTEN
1E24  6578         ORA             C                 ; INSERT FRONT PANEL CONDITIONS
1E25  6579         JMP             7$                ; PUT DATA IN REGISTER
1E28  6580 5$:     POP             $PSW              ; RESTORE THE DATA
1E29  6581 7$:     INX             D                 ; ...
1E2A  6582         MOV             M,A               ; WRITE THE REGISTER
1E2B  6583         INX             H                 ; ...
1E2C  6584         DCR             B                 ; DECREMENT THE BYTE COUNT
1E2D  6585         JNZ             3$                ; CONTINUE FOR DATA TYPE NO OF BYTES
1E30  6586         RET                               ; EXIT
1E31  6587
```

```
1E31                 .SBTTL   "  SKIP ROUTINE
1E31        ;++
1E31        ;
1E31        ; FUNCTIONAL DESCRIPTION:
1E31        ;
1E31        ;        IF THE CONDITION IS SPECIFIED AND IS ONE OF 'ONERROR' OR 'NOERROR'
1E31        ;        THE LOOP ERROR FLAG IS TESTED AND IF IN THE SPECIFIED STATE, THE
1E31        ;        TEST PC IS LOADED WITH THE DESTINATION ADDRESS. IF THE CONDITION
1E31        ;        IS 'ONEQUAL', THE CURRENT VALUES OF THE TWO INDEXES ARE COMPARED
1E31        ;        AND IF EQUAL, THE TEST PC IS LOADED WITH THE SPECIFIED ADDRESS.
1E31        ;
1E31        ;        IF THE CONDITION IS NOT SPECIFIED, THE TEST PC IS LOADED WITH
1E31        ;        THE SPECIFIED ADDRESS.
1E31        ;
1E31        ; CALLING SEQUENCE:
1E31        ;
1E31        ;        SKIP    [DESTINATION ADDRESS],[CONDITION],[INDEX NAME],[INDEX NAME]
1E31        ;
1E31        ; IMPLICIT INPUT:
1E31        ;
1E31        ;        ARGUMENT LIST:
1E31        ;
1E31        ;                OFFSET 0 - DESTINATION ADDRESS LOW
1E31        ;                OFFSET 1 - DESTINATION ADDRESS HIGH
1E31        ;                OFFSET 2 - CONDITION
1E31        ;                OFFSET 3 - INDEX FLAG
1E31        ;                OFFSET 4 - INDEX FLAG
1E31        ;
1E31        ; IMPLICIT OUTPUTS:
1E31        ;
1E31        ;        TEST_PC - SEE FUNCTIONAL DESCRIPTION.
1E31        ;
1E31        ;--
1E31
1E31        SKIP:
1E31                 LDA             SOFT_CTRL_REGA   ; GET THE LOOP ERROR FLAG
1E34                 ANI             LOOP_ERROR_FLAG  ; ...
1E36                 MOV             B,A              ; SAVE IN B
1E37                 LDA             ARG_LIST+2       ; GET THE CONDITION
1E3A                 ORA             A                ; SET THE CONDITION CODES
1E3B                 JM              1$               ; BRANCH IF NONE SPECIFIED
1E3E                 CPI             ONEQUAL          ; INDEX CONDITION?
1E40                 JZ              5$               ; BRANCH IF YES
1E43                 CPI             NOTEQUAL         ; INDEX CONDITION?
1E45                 JZ              5$               ; BRANCH IF YES
1E48                 CPI             ONERROR          ; ONERROR CONDITION?
1E4A                 JZ              15$              ; BRANCH IF ON
1E4D        ;
1E4D        ; MUST BE NOERROR CONDITION
1E4D        ;
1E4D                 MOV             A,B              ; GET LOOP ERROR FLAG
1E4E                 ORA             A                ; IS IT SET?
1E4F                 JZ              1$               ; BRANCH IF NO
1E52                 JMP             10$              ; EXIT
1E55        ;
```

```
1E55  ; ONERROR CONDITION
1E55  ;
1E55  15$:    MOV     A,B               ; GET LOOP ERROR FLAG
1E56          ORA     A                 ; IS TI SET?
1E57          JNZ     1$                ; BRANCH IF YES
1E5A          JMP     10$               ; EXIT
1E5D  ;
1E5D  ; ONEQUAL OR NOTEQUAL CONDITION
1E5D  ;
1E5D  5$:     LDA     ARG_LIST+3        ; GET THE FIRST INDEX CODE
1E60          CALL    GET_INDEX_VALUE   ; GET THE CURRENT VALUE
1E63          PUSH    B                 ; SAVE CURRENT VALUE
1E64          LDA     ARG_LIST+4        ; GET THE SECOND INDEX CODE
1E67          CALL    GET_INDEX_VALUE   ; GET THE CURRENT VALUE
1E6A  ;
1E6A  ; NOW COMPARE THE CURRENT VALUES OF THE TWO INDEXES
1E6A  ;
1E6A          POP     D                 ; GET VALUE OF FIRST INDEX
1E6B          MOV     A,E               ; CHECK LOWER ORDER BITS
1E6C          CMP     C                 ; EQUAL?
1E6D          JNZ     7$                ; BRANCH IF NO
1E70          MOV     A,D               ; CHECK UPPER ORDER BITS
1E71          CMP     B                 ; EQUAL?
1E72          JNZ     7$                ; BRANCH IF NO
1E75      ;
1E75      ; INDEXES ARE EQUAL. CHECK THE CONDITION
1E75      ;
1E75          LDA     ARG_LIST+2        ; GET THE CONDITION
1E78          CPI     ONEQUAL           ; ONEQUAL CONDITION?
1E7A          JZ      1$                ; BRANCH IF YES
1E7D          JMP     10$               ; EXIT
1E80      ;
1E80      ; INDEXES ARE NOT EQUAL. CHECK THE CONDITION
1E80      ;
1E80  7$:     LDA     ARG_LIST+2        ; GET THE CONDITION
1E83          CPI     ONEQUAL           ; ONEQUAL CONDITION?
1E85          JZ      10$               ; BRANCH IF YES
1E88  ;
1E88  ; SET THE TEST PC TO THE SPECIFIED ADDRESS
1E88  ;
1E88  1$:     LHLD    INIT_TEST_PC      ; GET BASE ADDRESS OF TEST
1E8B          XCHG
1E8C          LHLD    ARG_LIST          ; GET THE SPECIFIED ADDRESS
1E8F          DAD     D                 ; RELOCATE THE DESTINATION ADDRESS
1E90          SHLD    TEST_PC           ; SET THE TEST PC TO THE DESIRED ADDRESS
1E93  10$:    RET                       ; EXIT
1E94
```

```
1E94          .SBTTL  "  SUB TEST ROUTINE
1E94    ;++
1E94    ;
1E94    ; FUNCTIONAL DESCRIPTION:
1E94    ;
1E94    ;       THIS ROUTINE INCREMENTS THE SUBTEST NUMBER AND INITIALIZES THE
1E94    ;       THE CURRENT VALUES OF THE INDEX NAMES.
1E94    ;
1E94    ; CALLING SEQUENCE:
1E94    ;
1E94    ;       SUBTEST
1E94    ;
1E94    ; IMPLICIT INPUTS:
1E94    ;
1E94    ;       ARGUMENT LIST:
1E94    ;
1E94    ;               OFFSET 0 - CONTAINS THE SUB TEST NUMBER
1E94    ;
1E94    ; IMPLICIT OUTPUTS:
1E94    ;
1E94    ;       SUB_TEST_NUMB - GETS THE NEW SUBTEST NUMBER
1E94    ;
1E94    ;--
1E94
1E94    SUB_TEST:
1E94    ;
1E94    ; INITIALIZE THE CURRENT VALUE OF THE LOOP NAMES
1E94    ;
1E94            CALL            INIT_LOOP_VALUE ;
1E97    ;
1E97    ; INITIALIZE THE TRAP FLAG
1E97    ;
1E97            LDA             SOFT_CTRL_REGA   ; GET THE FLAG
1E9A            ANI             <^CEXP_UTRAP_FLAG> ; CLEAR THE FLAG
1E9C            STA             SOFT_CTRL_REGA   ; RESTORE THE FLAGS
1E9F    ;
1E9F    ; SET THE SUBTEST NUMBER
1E9F    ;
1E9F            LDA             ARG_LIST         ; GET THE SUBTEST NUMBER
1EA2            STA             SUB_TEST_NUMB    ; SAVE THE NEW SUB TEST NUMBER
1EA5            RET                              ; EXIT
1EA6
```

```
1EA6              .SBTTL  "  FETCH ROUTINE
1EA6     ;++
1EA6     ;
1EA6     ; FUNCTIONAL DESCRIPTION:
1EA6     ;
1EA6     ;        THIS ROUTINE IS USED TO LOAD A DCS MICRO WORD INTO THE 11/750
1EA6     ;        CONTROL STORE LATCHES. THIS IS DONE WITH THE FOLLOWING SEQUENCE:
1EA6     ;
1EA6     ;        1) ASSERT 'MICRO ADDRESS INHIBIT' AND 'PARITY CHECK'
1EA6     ;        2) LOAD THE DCS ADDRESS REGISTER
1EA6     ;        3) LOAD THE 'ADDRESS MATCH' REGISTER WITH 1800(X).
1EA6     ;        4) STEP THE CLOCK ONE MICRO STATE.
1EA6     ;        5) DEASCERT 'PARITY CHECK'
1EA6     ;        6) IF 'INHIBIT' IS NOT SPECIFIED, DEASCERT 'MICRO ADDRESS INHIBIT'
1EA6     ;
1EA6     ;        THE CONTENTS OF SOMM ADDRESS IS REPLACED IN THE MATCH REGISTER
1EA6     ;        AFTER THE FETCH.
1EA6     ;
1EA6     ; CALLING SEQUENCE:
1EA6     ;
1EA6     ;        FETCH   DCS_ADDRESS,INHIBIT
1EA6     ;
1EA6     ; IMPLICIT INPUTS:
1EA6     ;
1EA6     ;        ARGUMENT LIST:
1EA6     ;
1EA6     ;                OFFSET 0 - DCS ADDRESS
1EA6     ;                OFFSET 1 - DESTINATION INHIBIT CODE
1EA6     ;
1EA6     ; IMPLICIT OUTPUTS:
1EA6     ;
1EA6     ;        NONE
1EA6     ;
1EA6     ;--
1EA6
1EA6    FETCH_MIC_INSTR:
1EA6              LDA             SOFT_CTRL_REG   ; SET THE FETCH FLAG
1EA9              ORI             FETCH_FLAG      ; ...
1EAB              STA             SOFT_CTRL_REG   ; ...
1EAE              LDA             ARG_LIST        ; GET THE DCS ADDRESS
1EB1              CALL            START_EXECUTION ; FETCH THE MICRO WORD
1EB4     ;
1EB4     ; SEE IF MICRO ADDRESS INHIBIT SHOULD BE ASCERTED
1EB4     ;
1EB4              LDA             ARG_LIST+1      ; GET THE INHIBIT FLAG
1EB7              ORA             A               ; SET THE CONDITION CODES
1EB8              LDA             DCS_CTRL_RE_CPY ; GET CTRL REG DATA
1EBB              JM              1$              ; BRANCH IF INHIBIT NOT SPECIFIED
1EBE              ORI             MICRO_ADDR_INH  ; LEAVE MICRO ADDR INHIBIT SET
1EC0    1$:       STA             DCS_CTRL_REG    ; LOAD THE CONTROL REGISTER
1EC3              STA             DCS_CTRL_RE_CPY ; SAVE A COPY
1EC6     ;
1EC6     ; LOAD THE SOMM ADDRESS IN THE MATCH REGISTER
1EC6     ;
1EC6              LHLD            SOMM_ADDRESS    ; GET THE SOMM ADDRESS
```

```
                                     1EC9  6790          SHLD              ADDR_MATCH_LO    ; LOAD THE MATCH REGISTER
                                     1ECC  6791          RET                                ; EXIT
                                     1ECD  6792
```

```
1ECD          .SBTTL  "  MASK ROUTINE
1ECD  ;++
1ECD  ;
1ECD  ; FUNCTIONAL DESCRIPTION:
1ECD  ;
1ECD  ;       THIS ROUTINE IS USED TO MASK AN ELEMENT OF DATA. THE MASK IS PERFORMED
1ECD  ;       BY DOING A LOGICAL 'AND' OF THE DATA WITH THE MASK.
1ECD  ;
1ECD  ; CALLING SEQUENCE:
1ECD  ;
1ECD  ;       MASK    SOURCE ADDRESS,SOURCE INDEX,MASK ADDRESS,MASK INDEX
1ECD  ;
1ECD  ; IMPLICIT INPUTS:
1ECD  ;
1ECD  ;       ARGUMENT LIST:
1ECD  ;
1ECD  ;               OFFSET 0 - SOURCE ADDRESS LOW
1ECD  ;               OFFSET 1 - SOURCE ADDRESS HIGH
1ECD  ;               OFFSET 2 - SOURCE INDEX CODE
1ECD  ;               OFFSET 3 - MASK ADDRESS LOW
1ECD  ;               OFFSET 4 - MASK ADDRESS HIGH
1ECD  ;               OFFSET 5 - MASK INDEX CODE
1ECD  ;               OFFSET 6 - DATA TYPE CODE
1ECD  ;
1ECD  ; IMPLICIT OUTPUTS:
1ECD  ;
1ECD  ;       THE CONTENTS OF THE SOURCE ADDRESS IS MODIFIED BY THE MASK.
1ECD  ;
1ECD  ;--
1ECD
1ECD  MASK:   LDA             ARG_LIST+2      ; GET THE SOURCE INDEX CODE
1ED0          CALL            GET_INDEX_VALUE ; GET THE CURRENT VALUE
1ED3  ;
1ED3  ; NOW MULTIPLY THE INDEX BY THE DATA TYPE
1ED3  ;
1ED3          LDA             ARG_LIST+6      ; GET THE DATA TYPE
1ED6          CALL            MULTIPLY_A_BC   ; MULTIPLY DATA TYPE AND LOOP VALUE
1ED9  ;
1ED9  ; D & E REGISTER NOW HAS SOURCE INDEX
1ED9  ;
1ED9          LHLD            ARG_LIST        ; GET THE SOURCE ADDRESS
1EDC          DAD             D               ; ADD THE INDEX
1EDD          XCHG
1EDE          LHLD            INIT_TEST_PC    ; GET INITIAL TEST PC
1EE1          DAD             D               ; RELOCATE SOURCE ADDRESS
1EE2          PUSH            H               ; SAVE THE SOURCE ADDRESS
1EE3  ;
1EE3  ; NOW GET THE MASK INDEX AND ADDRESS
1EE3  ;
1EE3          LDA             ARG_LIST+5      ; GET THE MASK INDEX CODE
1EE6          CALL            GET_INDEX_VALUE ; GET THE CURRENT VALUE
1EE9          LDA             ARG_LIST+6      ; GET THE DATA TYPE
1EEC          CALL            MULTIPLY_A_BC   ; MULTIPLY DATA TYPE TIMES INDEX VALUE
1EEF          LHLD            ARG_LIST+3      ; GET THE MASK ADDRESS
1EF2          DAD             D               ; INDEX THE MASK ADDRESS
```

```
1EF3  6849         XCHG
1EF4  6850         LHLD           INIT_TEST_PC     ; GET INITIAL TEST PC
1EF7  6851         DAD            D                ; RELOCATE MASK ADDRESS
1EF8  6852 ;
1EF8  6853 ; NOW START MASKING THE SOURCE DATA
1EF8  6854 ;
1EF8  6855         POP            D                ; GET SOURCE ADDRESS
1EF9  6856         LDA            ARG_LIST+6       ; GET THE DATA TYPE
1EFC  6857         MOV            B,A              ; SAVE
1EFD  6858 1$:     LDAX           D                ; GET A BYTE OF THE DATA
1EFE  6859         ANA            M                ; MASK THE BYTE
1EFF  6860         STAX           D                ; PUT BACK IN SOURCE ADDRESS
1F00  6861         INX            D                ; INCREMENT THE SOURCE AND
1F01  6862         INX            H                ; MASK ADDRESSES
1F02  6863         DCR            B                ; DECREMENT THE BYTE COUNT
1F03  6864         JNZ            1$               ; CONTINUE FOR DATA TYPE NO. OF BYTES
1F06  6865 ;
1F06  6866 ; ALL DONE
1F06  6867 ;
1F06  6868         RET                             ; EXIT
1F07  6869
```

```
1F07            .SBTTL  "  COMPARE REGISTER ROUTINE
1F07      ;++
1F07      ;
1F07      ; FUNCTIONAL DESCRIPTION:
1F07      ;
1F07      ;       THIS ROUTINE COMPARES THE CONTENTS OF AN RDM REGISTER(S) WITH
1F07      ;       THE CONTENTS OF A SPECIFIED LOCATION IN THE THE 'DATA' AREA OF THE
1F07      ;       TEST. IF THE TWO DATA ELEMENTS DO NOT SATISFY THE MODE ARGUMENT,
1F07      ;       THE ERROR FLAG AND LOOP ERROR FLAG ARE SET OTHERWISE, THE LOOP ERROR
1F07      ;       FLAG IS CLEARED.
1F07      ;
1F07      ; CALLING SEQUENCE:
1F07      ;
1F07      ;       CMPREG  REG ADDR,DEST ADDR,[DEST INDEX],[DATA TYPE]
1F07      ;
1F07      ; IMPLICIT INPUTS:
1F07      ;
1F07      ;       ARGUMENT LIST:
1F07      ;
1F07      ;               OFFSET 0 - REGISTER ADDRESS LOW
1F07      ;               OFFSET 1 - REGISTER ADDRESS HIGH
1F07      ;               OFFSET 2 - DESTINATION ADDRESS LOW
1F07      ;               OFFSET 3 - DESTINATION ADDRESS HIGH
1F07      ;               OFFSET 4 - DESTINATION INDEX CODE
1F07      ;               OFFSET 5 - DATA TYPE CODE
1F07      ;               OFFSET 6 - MODE TYPE
1F07      ;                               0 = COMPARE FOR EQUALITY
1F07      ;                              FF = COMPARE FOR NOT EQUAL
1F07      ;
1F07      ; IMPLICIT OUTPUTS:
1F07      ;
1F07      ;       GOOD_DATA       - GETS THE CONTENTS OF THE 'DATA' ELEMENT
1F07      ;       BAD_DATA        - GETS THE CONTENTS OF THE RDM REGISTER
1F07      ;       DATA_TYPE       - GETS THE DATA TYPE USED IN THE COMPARISON
1F07      ;
1F07      ;--
1F07
1F07      COMPARE_REG:
1F07              LHLD            ARG_LIST        ; GET ADDRESS OF THE RDM REGISTER
1F0A              LDA             ARG_LIST+5      ; GET THE DATA TYPE
1F0D              CALL            SAVE_BAD_DATA   ; PUT THE REGISTER DATA INTO IMPLICIT STOAGE
1F10      ;
1F10      ; GET THE DATA INDEX AND INDEX THE DATA ADDRESS
1F10      ;
1F10              LDA             ARG_LIST+4      ; GET THE INDEX CODE
1F13              CALL            GET_INDEX_VALUE ; GET THE CURRENT VALUE
1F16              MOV             A,C             ; SAVE
1F17              STA             QA_INDEX        ; ...
1F1A              MOV             A,B             ; ...
1F1B              STA             QA_INDEX+1      ; ...
1F1E              LDA             ARG_LIST+5      ; GET THE DATA TYPE
1F21              CALL            MULTIPLY_A_BC   ; ADJUST INDEX VALUE BY DATA TYPE
1F24              LHLD            ARG_LIST+2      ; GET THE DATA ADDRESS
1F27              DAD             D               ; INDEX THE DATA ADDRESS
1F28              XCHG
```

```
1F29  6926          LHLD        INIT_TEST_PC     ; GET INITIAL TEST PC
1F2C  6927          DAD         D                ; RELOCATE DATA ADDRESS
1F2D  6928          LDA         ARG_LIST+5       ; GET THE DATA TYPE
1F30  6929          CALL        SAVE_GOOD_DATA   ; SAVE THE GOOD DATA
1F33  6930 ;
1F33  6931 ; NOW COMPARE THE GOOD AND BAD DATA
1F33  6932 ;
1F33  6933          LDA         ARG_LIST+6       ; GET THE MODE TYPE
1F36  6934          STA         MODE_TYPE        ; SAVE FOR TYPEOUT ROUTINE
1F39  6935          MOV         B,A              ; PASS IN B REGISTER
1F3A  6936          LDA         ARG_LIST+5       ; GET THE DATA TYPE
1F3D  6937          CALL        CMP_GOOD_BAD     ; COMPARE THE DATA
1F40  6938          JNC         10$              ; BRANCH IF DATA IS EQUAL TO REGISTER
1F43  6939          LDA         SOFT_CTRL_REGA   ; GET THE CONTROL FLAGS
1F46  6940          ORI         <ERROR_FLAG!-    ; SET THE ERROR AND LOOP ERROR
1F46  6941                      LOOP_ERROR_FLAG>; FLAGS
1F48  6942          ANI         <^CVBUS_COMPARE>; ENSURE VBUS COMPARE FLAG CLEAR
1F4A  6943          STA         SOFT_CTRL_REGA   ; ...
1F4D  6944          RET                          ; EXIT
1F4E  6945 ;
1F4E  6946 ; DATA WAS EQUAL TO REGISTER. CLEAR THE LOOP ERROR FLAG
1F4E  6947 ;
1F4E  6948 10$:    LDA         SOFT_CTRL_REGA   ; GET THE CONTROL REG
1F51  6949          ANI         <^C<LOOP_ERROR_FLAG!-
1F51  6950                      VBUS_COMPARE>> ; ...
1F53  6951          STA         SOFT_CTRL_REGA   ; ...
1F56  6952          CALL        CHECK_QA_FLAG    ; CHECK IF QA FLAG SET
1F59  6953          RET                          ; EXIT
1F5A  6954
```

```
1F5A            .SBTTL  " COMPARE REGISTER MASKED ROUTINE
1F5A    ;++
1F5A    ;
1F5A    ; FUNCTIONAL DESCRIPTION:
1F5A    ;
1F5A    ;       THIS ROUTINE COMPARES THE CONTENTS OF AN RDM REGISTER(S) WITH THE
1F5A    ;       CONTENTS OF A SPECIFIED DATA ELEMENT. BEFORE THE COMPARISON, THE
1F5A    ;       CONTENTS OF THE REGISTER(S) IS MASKED BY A SPECIFIED MASK. THE
1F5A    ;       MASK IS PERFORMED BY LOGICALLY 'ANDING' THE MASK AND THE CONTENTS
1F5A    ;       OF THE REGISTER(S). IF THE RESULTING COMPARISON DOES NOT SATISFY
1F5A    ;       THE MODE ARGUMENT, THE ERROR AND LOOP ERROR FLAGS ARE SET OTHERWISE,
1F5A    ;       THE LOOP ERROR FLAG IS CLEARED.
1F5A    ;
1F5A    ; CALLING SEQUENCE:
1F5A    ;
1F5A    ;       CMPREGMSK REG ADDRESS,DST ADDRESS,[DST INDEX],MSK ADDRESS,[MSK INDEX],[DAT -
cont> A TYPE]
1F5A    ;
1F5A    ; IMPLICIT INPUTS:
1F5A    ;
1F5A    ;       ARGUMENT LIST:
1F5A    ;
1F5A    ;               OFFSET 0 - REGISTER ADDRESS LOW
1F5A    ;               OFFSET 1 - REGISTER ADDRESS HIGH
1F5A    ;               OFFSET 2 - DESTINATION ADDRESS LOW
1F5A    ;               OFFSET 3 - DESTINATION ADDRESS HIGH
1F5A    ;               OFFSET 4 - DESTINATION INDEX CODE
1F5A    ;               OFFSET 5 - MASK ADDRESS LOW
1F5A    ;               OFFSET 6 - MASK ADDRESS HIGH
1F5A    ;               OFFSET 7 - MASK INDEX CODE
1F5A    ;               OFFSET 8 - DATA TYPE CODE
1F5A    ;               OFFSET 9 - MODE TYPE
1F5A    ;                           0 = COMPARE FOR EQUALITY
1F5A    ;                          FF = COMPARE FOR NOT EQUAL
1F5A    ;
1F5A    ; IMPLICIT OUTPUTS:
1F5A    ;
1F5A    ;       GOOD_DATA       - GETS THE CONTENTS OF THE DATA ELEMENT
1F5A    ;       BAD_DATA        - GETS THE MASKED CONTENTS OF THE RDM REGISTER
1F5A    ;       DATA_TYPE       - GETS THE SPECIFIED DATA TYPE
1F5A    ;
1F5A    ;--
1F5A
1F5A    COMPARE_REG_MSK:
1F5A            LHLD            ARG_LIST        ; GET THE REGISTER ADDRESS
1F5D            LDA             ARG_LIST+8      ; GET THE DATA TYPE
1F60            CALL            SAVE_BAD_DATA   ; SAVE THE REGISTER DATA
1F63    ;
1F63    ; NOW GET THE DESTINATION ADDRESS AND SAVE THE DATA
1F63    ;
1F63            LDA             ARG_LIST+4      ; GET THE INDEX CODE
1F66            CALL            GET_INDEX_VALUE ; GET THE INDEX VALUE
1F69            MOV             A,C             ; SAVE
1F6A            STA             QA_INDEX        ; ...
1F6D            MOV             A,B             ; ...
1F6E            STA             QA_INDEX+1      ; ...
```

```
1F71             LDA             ARG_LIST+8       ; GET THE DATA TYPE
1F74             CALL            MULTIPLY_A_BC    ; MULTIPLY DATA TYPE BY CURRENT INDEX VALUE
1F77             LHLD            ARG_LIST+2       ; GET THE DATA ADDRESS
1F7A             DAD             D                ; INDEX THE ADDRESS
1F7B             XCHG
1F7C             LHLD            INIT_TEST_PC     ; GET INITIAL TEST PC
1F7F             DAD             D                ; RELOATE DATA ADDRESS
1F80             LDA             ARG_LIST+8       ; GET THE DATA TYPE
1F83             CALL            SAVE_GOOD_DATA   ; SAVE THE DATA ELEMENT
1F86       ;
1F86       ; NOW GET THE MASK ADDRESS
1F86       ;
1F86             LDA             ARG_LIST+7       ; GET THE MASK INDEX CODE
1F89             CALL            GET_INDEX_VALUE  ; GET THE CURRENT INDEX VALUE
1F8C             LDA             ARG_LIST+8       ; GET THE DATA TYPE
1F8F             CALL            MULTIPLY_A_BC    ; GET THE INDEX VALUE TIMES THE DATA TYPE
1F92             LHLD            ARG_LIST+5       ; GET THE MASK ADDRESS
1F95             DAD             D                ; INDEX THE MASK ADDRESS
1F96             XCHG
1F97             LHLD            INIT_TEST_PC     ; GET INITIAL TEST PC
1F9A             DAD             D                ; RELOCATE MASK ADDRESS
1F9B       ;
1F9B       ; NOW MASK THE 'GOOD_DATA'
1F9B       ;
1F9B             PUSH            H                ; SAVE THE MASK ADDRESS
1F9C             LXI             D,GOOD_DATA      ; GET ADDRESS OF GOOD DATA
1F9F             LDA             ARG_LIST+8       ; GET THE DATA TYPE
1FA2             MOV             B,A              ; SAVE
1FA3       1$:   LDAX            D                ; GET A BYTE OF THE GOOD DATA
1FA4             ANA             M                ; AND WITH THE MASK
1FA5             STAX            D                ; PUT BACK IN GOOD DATA
1FA6             INX             D                ; POINT AT NEXT BYTE
1FA7             INX             H                ; ...
1FA8             DCR             B                ; DECREMENT THE BYTE COUNT
1FA9             JNZ             1$               ; CONTINUE FOR THE DATA TYPE NO. OF BYTES
1FAC       ;
1FAC       ; NOW MASK THE 'BAD_DATA'
1FAC       ;
1FAC             POP             H                ; GET ADDRESS OF MASK
1FAD             LXI             D,BAD_DATA       ; GET ADDRESS OF BAD DATA
1FB0             LDA             ARG_LIST+8       ; GET THE DATA TYPE
1FB3             MOV             B,A              ; SAVE
1FB4       2$:   LDAX            D                ; GET A BYTE OF THE BAD DATA
1FB5             ANA             M                ; AND WITH THE MASK
1FB6             STAX            D                ; PUT BACK IN BAD DATA
1FB7             INX             D                ; POINT AT NEXT BYTE
1FB8             INX             H                ; ...
1FB9             DCR             B                ; DECREMENT THE BYTE COUNT
1FBA             JNZ             2$               ; CONTINUE FOR THE DATA TYPE NO. OF BYTES
1FBD       ;
1FBD       ; DATA IS ALREADY IN THE 'GOOD' AND 'BAD' DATA LOCATIONS. NOW COMPARE IT.
1FBD       ;
1FBD             LDA             ARG_LIST+9       ; GET THE MODE TYPE
1FC0             STA             MODE_TYPE        ; SAVE FOR TYPEOUT ROUTINE
1FC3             MOV             B,A              ; PASS IN B REGISTER
```

```
1FC4              LDA             ARG_LIST+8         ; GET THE DATA TYPE
1FC7              CALL            CMP_GOOD_BAD       ; COMPARE THE DATA
1FCA              JNC             5$                 ; BRANCH IF DATA COMPARED
1FCD              LDA             SOFT_CTRL_REGA     ; GET THE CONTROL REGISTER
1FD0              ORI             <ERROR_FLAG!-      ; SET THE ERROR AND LOOP ERROR
1FD0                              LOOP_ERROR_FLAG>;  FLAGS
1FD2              ANI             <^CVBUS_COMPARE>;  ENSURE VBUS COMPARE FLAG CLEAR
1FD4              STA             SOFT_CTRL_REGA     ; ...
1FD7              RET
1FD8        ;
1FD8        ; DATA IS OK. CLEAR THE LOOP ERROR FLAG.
1FD8        ;
1FD8        5$:   LDA             SOFT_CTRL_REGA
1FDB              ANI             <^C<LOOP_ERROR_FLAG!-
1FDB                              VBUS_COMPARE>>
1FDD              STA             SOFT_CTRL_REGA
1FE0              CALL            CHECK_QA_FLAG
1FE3              RET                                ; EXIT
1FE4
```

```
1FE4            .SBTTL  "  COMPARE V BUS ROUTINE
1FE4    ;++
1FE4    ;
1FE4    ; FUNCTIONAL DESCRIPTION:
1FE4    ;
1FE4    ;       THIS ROUTINE IS USED TO COMPARE VALUES OF BITS ON THE V BUS WITH
1FE4    ;       EXPECTED VALUES OF THE BITS. THE EXPECTED VALUES MUST BE PLACED IN
1FE4    ;       A TABLE WITH THE FOLLOWING FORMAT:
1FE4    ;
1FE4    ;               .BYTE   NUMBER OF BITS
1FE4    ;               .BYTE   BIT<7>=VALUE AND BITS <6:0>=BIT NUMBER
1FE4    ;                  .
1FE4    ;                  .
1FE4    ;
1FE4    ;       IF ANY BIT ON THE V BUS IS NOT IN THE SAME STATE AS SPECIFIED IN THE
1FE4    ;       TABLE, THE ERROR AND LOOP ERROR FLAGS ARE SET.
1FE4    ;
1FE4    ; CALLING SEQUENCE:
1FE4    ;
1FE4    ;       CMPVBUS TABLE_ADDRESS,[INDEX NAME]
1FE4    ;
1FE4    ; IMPLICIT INPUTS:
1FE4    ;
1FE4    ;       ARGUMENT LIST:
1FE4    ;
1FE4    ;               OFFSET 0 - DATA TABLE ADDRESS LOW
1FE4    ;               OFFSET 1 - DATA TABLE ADDRESS HIGH
1FE4    ;               OFFSET 2 - INDEX NAME CODE
1FE4    ;
1FE4    ; IMPLICIT OUTPUTS:
1FE4    ;
1FE4    ;       GOOD_DATA - CONTAINS THE EXPECTED BIT NO. AND VALUE IF A BIT FAILED
1FE4    ;       BAD_DATA  - CONTAINS THE RECEIVED BIT NO. AND VALUE IF A BIT FAILED
1FE4    ;       ERROR AND LOOP ERROR FLAGS ARE SET IF ANY BIT IS NOT CORRECT
1FE4    ;
1FE4    ;--
1FE4
1FE4    COMPARE_VBUS:
1FE4            MVI             A,WORD          ; INITIALIZE THE DATA TYPE
1FE6            STA             DATA_TYPE       ; ...
1FE9            XRA             A               ; INITIALIZE THE MODE TYPE
1FEA            STA             MODE_TYPE       ; 'EQUALS' FOR THIS COMPARE
1FED    ;
1FED    ; CLEAR THE GOOD AND BAD DATA LONG WORDS
1FED    ;
1FED            LXI             H,0             ; GET DATA TO CLEAR WITH
1FF0            SHLD            GOOD_DATA       ; CLEAR LOW 16 BITS
1FF3            SHLD            GOOD_DATA+2     ; AND UPPER 16 BITS
1FF6            SHLD            BAD_DATA        ; CLEAR BAD DATA
1FF9            SHLD            BAD_DATA+2      ; ...
1FFC    ;
1FFC    ; FIRST READ THE V BUS INTO A BUFFER
1FFC    ;
1FFC            CALL            READ_V_BUS      ; READ THE V BUS
1FFF            LHLD            ARG_LIST        ; GET ADDRESS OF EXPECTED TABLE
```

```
2002          XCHG
2003          LHLD            INIT_TEST_PC      ; GET INITIAL TEST PC
2006          DAD             D                 ; RELOCATE TABLE ADDRESS
2007 ;
2007 ; SEE IF TABLE ADDRESS IS INDEXED
2007 ;
2007          PUSH            H                 ; SAVE H & L
2008          LDA             ARG_LIST+2        ; GET THE INDEX FLAG
200B          CALL            GET_INDEX_VALUE   ; GET THE CURRENT VALUE
200E          MOV             A,C               ; SAVE
200F          STA             QA_INDEX          ; ...
2012          MOV             A,B               ; ...
2013          STA             QA_INDEX+1        ; ...
2016          POP             H                 ; RESTORE H & L
2017          JC              5$                ; BRANCH IF NO INDEX
201A ;
201A ; GENRATE THE ADDRESS OF THE INDEXED TABLE
201A ;
201A          INR             C                 ; BUMP THE REGISTER FOR COUNT TO ZERO
201B 3$:      DCR             C                 ; DONE INDEXING?
201C          JZ              5$                ; BRANCH IF YES
201F          MOV             E,M               ; GET NUMBER OF BITS IN CURRENT TABLE
2020          MVI             D,0               ; SETUP TO ADD TO TABLE ADDRESS
2022          INX             D                 ; ...
2023          DAD             D                 ; POINT TO NEXT TABLE
2024          JMP             3$                ; CHECK IF DONE INDEXING
2027 ;
2027 ; GET NUMBER OF BITS TO COMPARE
2027 ;
2027 5$:      MOV             B,M               ; GET THE NUMBER OF BITS TO COMPARE
2028          INX             H                 ; POINT AT FIRST EXPECTED BIT
2029 ;
2029 ; NOW START CHECKING THE BITS
2029 ;
2029 1$:      CALL            CHECK_VBUS        ; CHECK THE VBUS BIT
202C          JC              10$               ; BRANCH IF BIT DID NOT MATCH
202F          INX             H                 ; POINT AT NEXT EXPECTED BIT
2030          DCR             B                 ; CHECKED ALL THE BITS?
2031          JNZ             1$                ; BRANCH IF NO
2034 ;
2034 ; ALL BITS ARE OK. CLEAR THE LOOP ERROR FLAG
2034 ;
2034          LDA             SOFT_CTRL_REGA
2037          ANI             <^C<LOOP_ERROR_FLAG!-
2037                          VBUS_COMPARE>>
2039          STA             SOFT_CTRL_REGA
203C          CALL            CHECK_QA_FLAG     ; CHECK IF QA FLAG SET
203F          RET                               ; EXIT
2040 ;
2040 ; A BIT ON THE V BUS IS INCORRECT. SET THE ERROR AND LOOP ERROR FLAGS.
2040 ;
2040 10$:     LDA             SOFT_CTRL_REGA    ; GET THE FLAGS
2043          ORI             <ERROR_FLAG!-     ; SET THE ERROR AND LOOP
2043                          LOOP_ERROR_FLAG!-
2043                          VBUS_COMPARE>     ; ERROR FLAGS
```

```
        2045   7196         STA             SOFT_CTRL_REGA   ; ...
        2048   7197         INX             H                ; POINT AT NEXT BIT
        2049   7198         SHLD            VBUS_TBL_ADDR    ; SAVE THE TABLE ADDRESS
        204C   7199         MOV             A,B              ; AND THE BIT COUNT
        204D   7200         DCR             A                ; DECREMENT THE BIT COUNT
        204E   7201         STA             VBUS_TBL_COUNT   ; ...
        2051   7202         RET                              ; EXIT
        2052   7203
```

```
2052                .SBTTL  "  IF ERROR ROUTINE
2052        ;++
2052        ;
2052        ; FUNCTIONAL DESCRIPTION:
2052        ;
2052        ;       THIS ROUTINE IS USED TO TEST THE ERROR AND LOOP ERROR FLAGS. FOLLOWING
2052        ;       IS THE ALGORITHM:
2052        ;
2052        ;       IF THE ERROR FLAG IS CLEAR
2052        ;       THEN EXIT
2052        ;       ELSE
2052        ;
2052        ;       IF THE CONTROL_C_FLAG IS SET
2052        ;       THEN CALL THE COMMAND PARSER
2052        ;       ELSE
2052        ;
2052        ;       IF THE LOOP_ERROR_FLAG IS SET
2052        ;       THEN
2052        ;          IF THE BELL_FLAG IS SET
2052        ;          THEN CALL THE RING_BELL ROUTINE
2052        ;          ELSE
2052        ;
2052        ;          IF THE NER_FLAG IS CLEAR
2052        ;          THEN CALL THE TYPE ERROR ROUTINE
2052        ;          ELSE
2052        ;
2052        ;          IF THE HALT_FLAG IS SET
2052        ;          THEN CALL THE COMMAND PARSER
2052        ;          ELSE
2052        ;
2052        ;       IF THE LOOP_FLAG IS SET AND THE SA_FLAG IS CLEAR
2052        ;       THEN
2052        ;          IF THE TEST_PC IS EQUAL TO THE ERROR_PC
2052        ;          THEN SET THE TEST_PC TO THE ERROR LOOP ADDRESS
2052        ;               CLEAR THE ERROR LOOP FLAG, AND EXIT
2052        ;          ELSE EXIT
2052        ;       ELSE EXIT
2052        ;
2052        ; CALLING SEQUENCE:
2052        ;
2052        ;       IFERROR [DATA COUNT],[GATE ARRAY LIST],[MODULE LIST]
2052        ;
2052        ; IMPLICIT INPUTS:
2052        ;
2052        ;       ARGUMENT LIST:
2052        ;
2052        ;               OFFSET 0 - DATA COUNT
2052        ;               OFFSET 1 - NUMBER OF GATE ARRAY CODES IN LIST
2052        ;               OFFSET 2 - FIRST GATE ARRAY CODE
2052        ;                  .
2052        ;                  .
2052        ;               OFFSET N - LAST GATE ARRAY CODE
2052        ;               OFFSET N+1 - NUMBER OF MODULE NAME CODES IN LIST
2052        ;               OFFSET N+2 - MODULE NAME CODE
2052        ;                  .
```

```
2052  ;
2052  ;                       OFFSET N+M - LAST MODULE NAME CODE
2052  ;
2052  ;               ERROR_LOOP_ADDR - CONTAINS THE ADDRESS TO LOOP BACK TO IF THE LOOP
2052  ;                                 FLAG IS SET AND THE SA FLAG IS CLEAR.
2052  ;
2052  ; IMPLICIT OUTPUTS:
2052  ;
2052  ;               TEST_PC          - CONTAINS THE ADDRESS OF THE NEXT PSEUDO INSTRUCTION
2052  ;
2052  ;--
2052
2052  IF_ERROR:
2052                  LDA             SOFT_CTRL_REGA   ; GET THE ERROR FLAG
2055                  ANI             ERROR_FLAG       ; IS IT SET?
2057                  JZ              50$              ; BRANCH IF NO (EXIT)
205A  ;
205A  ; CHECK THE CONTROL C FLAG
205A  ;
205A                  LDA             SOFT_CTRL_REG    ; GET THE CONTROL C FLAG
205D                  ANI             CONTROL_C_FLAG   ; IS IT SET?
205F                  CNZ             GET_CMD_LINE     ; CALL COMMAND PARSER IF YES
2062  ;
2062  ; CHECK THE LOOP ERROR FLAG
2062  ;
2062                  LDA             SOFT_CTRL_REGA   ; GET THE LOOP ERROR FLAG
2065                  ANI             LOOP_ERROR_FLAG  ; IS IT SET?
2067                  JZ              10$              ; BRANCH IF NO
206A  ;
206A  ; CHECK THE BELL FLAG
206A  ;
206A                  LDA             PROG_CTRL_REG    ; GET THE BELL FLAG
206D                  ANI             BELL_FLAG        ; IS IT SET?
206F                  CNZ             RING_BELL        ; RING THE TERMINAL BELL IF YES
2072  ;
2072  ; CHECK THE NER FLAG
2072  ;
2072                  LXI             H,ARG_LIST       ; GET ADDRESS OF ARGUMENT LIST
2075                  LDA             PROG_CTRL_REG    ; GET THE NER FLAG
2078                  ANI             NER_FLAG         ; IS IT CLEAR?
207A                  CZ              TYPE_ERROR       ; YES,TYPE THE ERROR MESSAGE
207D  ;
207D  ; CHECK THE HALT FLAG
207D  ;
207D  5$:            LDA             PROG_CTRL_REG    ; GET THE HALT FLAG
2080                  ANI             HALT_FLAG        ; IS THE HALT FLAG SET?
2082                  CNZ             GET_CMD_LINE     ; CALL COMMAND PARSER IF YES
2085  ;
2085  ; CHECK THE LOOP FLAG AND THE SA FLAG
2085  ;
2085  10$:           LDA             PROG_CTRL_REG    ; GET THE LOOP AND SA FLAGS
2088                  MOV             B,A              ; SAVE
2089                  ANI             LOOP_FLAG        ; IS THE LOOP FLAG SET?
208B                  JZ              50$              ; BRANCH IF NO (EXIT)
208E                  MOV             A,B              ; GET THE SA FLAG
```

```
208F            ANI             SA_FLAG         ; IS THE SA FLAG SET?
2091            JNZ             50$             ; BRANCH IF YES (EXIT)
2094      ;
2094      ; SEE IF THIS IS THE CORRECT ERROR PC
2094      ;
2094            LXI             H,CURRENT_PC    ; GET ADDRESS OF CURRENT PC
2097            LDA             ERROR_PC        ; GET LOW BYTE OF ERROR PC
209A            CMP             M               ; LOW BYTE THE SAME?
209B            JNZ             50$             ; BRANCH IF NO
209E            INX             H               ; POINT AT HIGH BYTE
209F            LDA             ERROR_PC+1      ; GET HIGH BYTE
20A2            CMP             M               ; HIGH BYTE THE SAME?
20A3            JNZ             50$             ; BRANCH IF NO
20A6      ;
20A6      ; SET THE TEST PC TO THE ERROR LOOP ADDRESS
20A6      ;
20A6            LHLD            ERROR_LOOP_ADDR ; GET THE ERROR LOOP TEST PC
20A9            SHLD            TEST_PC         ; SET THE TEST PC
20AC            LDA             SOFT_CTRL_REGA  ; GET THE CONTROL REGISTER
20AF            ANI             <^CLOOP_ERROR_FLAG>; CLEAR THE ERROR LOOP FLAG
20B1            STA             SOFT_CTRL_REGA  ; ...
20B4      ;
20B4      ; EXIT
20B4      ;
20B4      50$:  RET                             ; EXIT
20B5
```

```
20B5          .SBTTL  "  BURST CLOCK ROUTINE
20B5    ;++
20B5    ;
20B5    ; FUNCTIONAL DESCRIPTION:
20B5    ;
20B5    ;       THIS ROUTINE IS USED TO BURST THE CPU CLOCK. THE ALGORITHM IS:
20B5    ;
20B5    ;       IF THE LOOP AND ERROR FLAGS ARE SET AND THE SA FLAG IS CLEAR AND
20B5    ;          THE INHIBIT FLAG IS CLEAR
20B5    ;       THEN CLEAR THE 'STOP CLOCK' BIT AND SET THE 'DCS ADDRESS CLEAR'
20B5    ;            BIT AT THE SPECIFIED DCS ADDRESS
20B5    ;       ELSE SET THE 'STOP CLOCK' BIT AND CLEAR THE 'DCS ADDRESS CLEAR'
20B5    ;            BIT AT THE SPECIFIED DCS ADDRESS
20B5    ;
20B5    ;       IF THE CYCLE OR TICK FLAG IS SET
20B5    ;       THEN WHILE THE OPERATOR TYPES SPACES, DO
20B5    ;(15$)     BEGIN
20B5    ;          SET THE TYPEOUT ADDRESS TO THE CURRENT DCS ADDRESS MINUS ONE
20B5    ;          TYPE THE TYPEOUT ADDRESS
20B5    ;          IF THE LOOP AND ERROR FLAGS ARE NOT SET
20B5    ;          THEN
20B5    ;             IF THE TYPEOUT ADDRESS EQUALS THE <DCS ADDRESS>
20B5    ;             THEN CALL THE COMMAND PARSER
20B5    ;(25$)     WAIT FOR OPERATOR INPUT
20B5    ;          CHECK FOR SPACE
20B5    ;          STEP THE CLOCK (CYCLE OR TICK)
20B5    ;          END
20B5    ;       CALL COMMAND PARSER
20B5    ;          IF RETURN FROM COMMAND PARSER
20B5    ;          THEN RESTART THIS ROUTINE
20B5    ;          ELSE
20B5    ;       ELSE
20B5    ;
20B5    ;(30$)  START THE CPU CLOCK
20B5    ;       UNTIL OPERATOR TYPES 'CONTROL C' OR THE CLOCK STOPS, DO
20B5    ;       BEGIN
20B5    ;          WAIT HERE
20B5    ;          END
20B5    ;       IF THE OPERATOR TYPED CONTROL C
20B5    ;       THEN CALL THE COMMAND PARSER. IF COMMAND PARSER RETURNS,
20B5    ;            RESTART THIS ROUTINE
20B5    ;       ELSE
20B5    ;(40$)  IF THE 'EXP_UTRAP_FLAG' IS SET
20B5    ;       THEN
20B5    ;        IF THE TRAP REGISTER IS STILL IN DCS SPACE
20B5    ;        THEN
20B5    ;(45$)    IF THE SOMM FLAG IS SET AND THE STOP CLOCK BIT IN THE CURRENT DCS
20B5    ;            ADDRESS IS NOT SET
20B5    ;         THEN TYPE 'STOP ON MICRO MATCH' MESSAGE AND CALL THE COMMAND PARSER
20B5    ;            IF THE COMMAND PARSER RETURNS
20B5    ;            THEN RESTART THE CLOCK AND WAIT AGAIN
20B5    ;            ELSE
20B5    ;         ELSE
20B5    ;(50$)       CLEAR THE CONTROL FILE
20B5    ;            IF THE SOMM FLAG IS CLEAR AND THE CURRENT DCS ADDRESS MINUS 1 IS
```

```
20B5   ;               EQUAL TO THE <DCS ADDRESS>
20B5   ;            THEN EXIT THIS ROUTINE
20B5   ;(60$)       ELSE TYPE 'UNEXPECTED CLOCK STOPPED' MESSAGE AND CALL THE COMMAND
20B5   ;                 PARSER. DON'T ALLOW CONTINUE.
20B5   ;        ELSE EXIT THIS ROUTINE.
20B5   ;       ELSE GO TO 45$
20B5   ;
20B5   ; CALLING SEQUENCE:
20B5   ;
20B5   ;       BRSTCLK <DCS ADDRESS>
20B5   ;
20B5   ; IMPLICIT INPUTS:
20B5   ;
20B5   ;       ARGUEMENT LIST:
20B5   ;
20B5   ;               OFFSET 0 - DCS ADDRESS TO STOP THE CLOCK AT
20B5   ;               OFFSET 1 - INHIBIT RETURN TO ZERO FLAG
20B5   ;                                = -1 IF SPECIFIED
20B5   ;                                =  0 IF NOT SPECIFIED
20B5   ;
20B5   ; IMPLICIT OUTPUTS:
20B5   ;
20B5   ;       NONE
20B5   ;
20B5   ;--
20B5
20B5   BURST_CLOCK:
20B5           LDA             ARG_LIST         ; GET THE SPECIFIED DCS ADDRESS
20B8           STA             SETCLR_CF_ADDR   ; SAVE FOR SET/CLR CF ROUTINES
20BB           LXI             H,CLK_STOP_MSG1  ; INITIALIZE THE ERROR MESSAGE
20BE           SHLD            CLK_MSG_ADDR     ; ...
20C1           LDA             DCS_ADDR_REG_RO  ; SAVE THE CURRENT DCS ADDRESS
20C4           ANI             <^C<CLK_CTRL_1_RO!-
20C4                           CLK_CTRL_0_RO>>  ; DISCARD CLOCK STATE BITS
20C6           STA             M_CLK_ADDR       ; ...
20C9           LDA             SOFT_CTRL_REG    ; CLEAR THE BURST_STOP FLAG
20CC           ANI             <^C<BURST_STOP_FLAG>>
20CE           STA             SOFT_CTRL_REG    ; ...
20D1   ;
20D1   ; CHECK THE INHIBIT BURST FLAG
20D1   ;
20D1           LDA             PROG_CTRL_REG    ; GET THE FLAG
20D4           ANI             IB_FLAG          ; IS THE FLAG SET?
20D6           JNZ             1$               ; BRANCH IF YES
20D9   ;
20D9   ; CHECK THE LOOP, ERROR, AND SA FLAGS
20D9   ;
20D9           LDA             SOFT_CTRL_REGA   ; GET THE ERROR FLAG
20DC           ANI             ERROR_FLAG       ; IS IT SET?
20DE           JZ              1$               ; BRANCH IF NO
20E1           LDA             PROG_CTRL_REG    ; GET THE LOOP FLAG
20E4           MOV             B,A              ; SAVE THE SA FLAG
20E5           ANI             LOOP_FLAG        ; IS IT SET?
20E7           JZ              1$               ; BRANCH IF NO
20EA           MOV             A,B              ; GET THE SA FLAG
```

```
20EB              ANI           SA_FLAG            ; IS IT CLEAR?
20ED              JNZ           1$                 ; BRANCH IF NO
20F0              LDA           ARG_LIST+1         ; GET THE INHIBIT FLAG
20F3              ORA           A                  ; SET THE CONDITION CODES
20F4              JNZ           1$                 ; BRANCH IF INHIBIT SPECIFIED
20F7      ;
20F7      ; LOOP AND ERROR FLAGS ARE SET, SA FLAG IS CLEAR. SET THE 'DCS ADDRESS
20F7      ; CLEAR' BIT AND CLEAR THE 'STOP CLOCK' BIT IN THE SPECIFIED DCS ADDRESS
20F7      ;
20F7              MVI           A,DCS_ADDR_CLEAR ; GET THE ADDRESS CLEAR BIT
20F9              STA           SETCLR_CF_DATA     ; PASS TO SET/CLR CF ROUTINE
20FC              CALL          SET_DCS_CF         ; SET THE BIT
20FF              MVI           A,STOP_CLOCK       ; CLEAR THE STOP CLOCK BIT
2101              STA           SETCLR_CF_DATA     ; ...
2104              CALL          CLEAR_DCS_CF       ; ...
2107              JMP           10$                ; GO TO NEXT STEP
210A      ;
210A      ; FLAGS WERE NOT IN THE CORRECT STATE. SET THE 'STOP CLOCK' BIT AND CLEAR
210A      ; THE 'DCS ADDRESS CLEAR' BIT IN THE CONTROL FILE.
210A      ;
210A      1$:     MVI           A,STOP_CLOCK       ; SET THE CLOCK STOP BIT
210C              STA           SETCLR_CF_DATA     ; ...
210F              CALL          SET_DCS_CF         ; ...
2112              MVI           A,DCS_ADDR_CLEAR ; GET THE ADDRESS CLEAR BIT
2114              STA           SETCLR_CF_DATA     ; CLEAR IT IN THE CONTROL FILE
2117              CALL          CLEAR_DCS_CF       ; ...
211A              LDA           SOFT_CTRL_REG      ; SET THE BURST_STOP FLAG
211D              ORI           BURST_STOP_FLAG ; ...
211F              STA           SOFT_CTRL_REG      ; ...
2122      ;
2122      ; CHECK IF SINGLE CYCLE OR SINGLE TICK HAS BEEN SPECIFIED
2122      ;
2122      10$:    LDA           SOFT_CTRL_REGA     ; GET THE TWO FLAGS
2125              ANI           <CYCLE_FLAG!TICK_FLAG> ; IS EITHER ONE SET?
2127              JZ            30$                ; BRANCH IF NO
212A      ;
212A      ; TYPE THE CURRENT DCS ADDRESS
212A      ;
212A      15$:    TYPE          CRLF               ;
2133              TYPE          DCS_ADDR_MSG       ; TYPE 'DCS ADDR= '
213C              LDA           DCS_ADDR_REG_RO ; GET CURRENT DCS ADDRESS
213F              ANI           <^C<CLK_CTRL_1_RO!-
213F                            CLK_CTRL_0_RO>> ; DISCARD CLOCK STATE BITS
2141              DCR           A                  ; BACKUP BY ONE
2142              STA           BURST_TEMP         ; SAVE FOR TYPEOUT
2145              MOV           B,A                ; SAVE A REG
2146              LDA           DCS_CTRL_FILE      ; SEE IF THE RETURN TO ZERO
2149              ANI           DCS_ADDR_CLEAR     ; BIT IS ACTIVE IN THE CONTROL FILE
214B              JNZ           16$                ; BRANCH IF NOT ACTIVE
214E              LDA           STEP_ADDRESS       ; GET THE PREVIOUS ADDRESS
2151              INR           A                  ; INCREMENT TO CURRENT ADDRESS
2152              STA           BURST_TEMP         ; SET FOR TYPEOUT
2155              MOV           B,A                ; SAVE FOR NEXT INSTRUCTION
2156      16$:    MOV           A,B                ; RESTORE THE DCS ADDRESS
2157              STA           STEP_ADDRESS       ; SAVE
```

```
215A          TYPEB          BURST_TEMP        ; TYPE THE CURRENT ADDRESS
2165     ;
2165     ; SEE IF WE ARE STILL IN DCS SPACE. IF NOT, TERMINATE TICKING THE CLOCK.
2165     ;
2165          LDA            SOFT_CTRL_REG     ; GET THE FETCH FLAG
2168          ANI            FETCH_FLAG        ; ...
216A          JNZ            19$               ; SKIP TEST IF JUST EXECUTED FETCH
216D          LDA            CS_ADD_TRAP_HGH   ; GET HIGH 8 BITS OF TRAP REGISTER
2170          ANI            <^X3F>            ; DISCARD UPPER TWO BITS
2172          CPI            <^X18>            ; CHECK IF IN THE RANGE 18 TO 1F
2174          JM             18$               ; BRANCH IF LESS THAN 18
2177          CPI            <^X20>            ; ...
2179          JM             19$               ; BRANCH IF LESS THAN 20
217C     ;
217C     ; WE LEFT DCS SPACE, TYPE THE TRAP REGISTER CONTENTS
217C     ;
217C     18$:     TYPE       CLK_STOP_MSG4     ; TYPE '  TRAP ADDRESS= '
2185          LHLD           CS_ADD_TRAP_LOW   ; GET TRAP REGISTER
2188          MOV            A,H               ; GET HIGH BYTE
2189          ANI            <^X3F>            ; DISCARD TOP TWOBITS
218B          MOV            H,A               ; PUT BACK
218C          SHLD           BURST_TEMP        ; SAVE
218F          TYPEW          BURST_TEMP        ; TYPE IT
219A          CALL           GET_CMD_LINE      ; CALL THE COMMAND PARSER
219D          RET                              ; EXIT THIS ROUTINE
219E     ;
219E     ; SEE IF THE LOOP AND ERROR FLAGS ARE SET
219E     ;
219E     19$:     LDA        ARG_LIST+1        ; GET THE INHIBIT ARGUMENT
21A1          ORA            A                 ; SET THE CONDITION CODES
21A2          JNZ            20$               ; BRANCH IF INHIBIT SPECIFIED
21A5          LDA            SOFT_CTRL_REGA    ; GET THE ERROR FLAG
21A8          MOV            B,A               ; SAVE
21A9          ANI            ERROR_FLAG        ; IS IT SET?
21AB          JZ             20$               ; BRANCH IF NO
21AE          LDA            PROG_CTRL_REG     ; GET THE LOOP FLAG
21B1          ANI            LOOP_FLAG         ; IS IT SET?
21B3          JNZ            25$               ; BRANCH IF YES & WAIT FOR OPERATOR
21B6                                           ; INPUT
21B6     ;
21B6     ; LOOP AND ERROR FLAGS WERE NOT BOTH SET OR THE INHIBIT ARGUMENT IS SPECIFIED.
21B6     ; DON'T ALLOW OPERATOR TO STEP BEYOND THE 'STOP CLOCK' MICRO INSTRUCTION.
21B6     ;
21B6     20$:     LDA        STEP_ADDRESS      ; GET THE TYPEOUT ADDRESS
21B9          MOV            B,A               ; SAVE
21BA          LDA            ARG_LIST          ; GET THE SPECIFIED ADDRESS
21BD          CMP            B                 ; DOES TYPEOUT ADDR=<DCS ADDRESS>?
21BE          JNZ            25$               ; BRANCH IF NO
21C1          CALL           GET_CMD_LINE      ; CALL THE COMMAND PARSER
21C4          RET                              ; CONTINUE WITH NEXT PSEUDO
21C5                                           ; INSTRUCT. IF OPERATOR TYPES 'CO'
21C5     ;
21C5     ; WAIT FOR OPERATOR INPUT AND CHECK FOR A SPACE.
21C5     ;
21C5     25$:     $TERM_INIT                   ; CLEAR ANY READ REQUESTS
```

```
21CA             $TERM_READ      TERM_INP_BUFF,ONE ; READ ONE CHARACTER
21D5             LDA             TERM_INP_BUFF+1 ; GET THE CHARACTER THAT WAS TYPED
21D8             CPI             <^X20>          ; WAS IT A SPACE?
21DA             JNZ             27$             ; BRANCH IF NO
21DD     ;
21DD     ; SPACE WAS TYPE, TICK THE CLOCK
21DD     ;
21DD             LDA             SOFT_CTRL_REG   ; CLEAR THE FETCH FLAG
21E0             ANI             <^CFETCH_FLAG>  ; ...
21E3             STA             SOFT_CTRL_REG   ; ...
21E5             LDA             SOFT_CTRL_REGA  ; GET THE CYCLE FLAG
21E8             ANI             CYCLE_FLAG      ; IS THIS A CYCLE TICK?
21EA             JZ              26$             ; BRANCH IF NO
21ED             SGL_MIC_INSTR   1               ; STEP THE CLOCK ONE CYCLE
21F5             JMP             15$             ; GO TYPE THE ADDRESS AGAIN
21F8     26$:    SGL_TICK        1               ; STEP THE CLOCK ONE TICK
2200             JMP             15$             ; GO TYPE THE ADDRESS AGAIN
2203     ;
2203     ; SPACE WAS NOT TYPED. CALL THE COMMAND PARSER
2203     ;
2203     27$:    LDA             SOFT_CTRL_REGA  ; CLEAR THE TWO FLAGS
2206             ANI             <^C<CYCLE_FLAG!-; ...
2206                             TICK_FLAG>>     ; ...
2208             STA             SOFT_CTRL_REGA  ; ...
220B             CALL            GET_CMD_LINE    ; CALL THE COMMAND PARSER
220E             JMP             BURST_CLOCK     ; RESTART THIS ROUTINE
2211
```

```
2211  ;
2211  ; NEITHER THE CYCLE NOR THE TICK FLAG WAS SET. START THE CLOCK
2211  ;
2211  30$:     LDA             SOFT_CTRL_REG    ; CLEAR THE FETCH FLAG
2214           ANI             <^CFETCH_FLAG>   ; ...
2216           STA             SOFT_CTRL_REG    ; ...
2219           LDA             DCS_CTRL_RE_CPY  ; GET COPY OF DCS CONTROL REGISTER
221C           ANI             <^C<CLK_CTRL_0!CLK_CTRL_1>> ; ACTIVATE BITS TO START CLOCK
221E           STA             DCS_CTRL_REG     ; START THE CLOCK
2221  ;
2221  ; START INTERRUPT DRIVEN INPUT REQUEST AND WAIT FOR EITHER A CONTROL C OR
2221  ; THE CLOCK TO STOP.
2221  ;
2221           LDA             SOFT_CTRL_REG    ; MAKE SURE CONTROL C FLAG
2224           STA             BURST_TEMP       ; SAVE INCASE CTRL C FLAG SET
2227           ANI             <^CCONTROL_C_FLAG>; IS CLEAR
2229           STA             SOFT_CTRL_REG    ; ...
222C  36$:     LDA             SOFT_CTRL_REG    ; GET THE CONTROL C FLAG
222F           ANI             CONTROL_C_FLAG   ; IS IT SET?
2231           JNZ             37$              ; BRANCH IF YES
2234           LDA             DCS_ADDR_REG_RO  ; READ THE CLOCK CONTROL BITS
2237           MOV             C,A              ; SAVE DCS ADDRESS
2238           ANI             <CLK_CTRL_0_RO!- ; DISCARD OTHER BITS
2238                           CLK_CTRL_1_RO>
223A           JNZ             37$              ; GET OUT IF CLOCK STOPPED
223D           LDA             SOFT_CTRL_REG    ; SEE OF BURST_STOP FLAG IS SET
2240           ANI             BURST_STOP_FLAG  ; ...
2242           JZ              36$              ; BRANCH IF IT'S NOT SET
2245    ;
2245    ; CHECK IF DCS ADDRESS REGISTER IS STILL CHANGING
2245    ;
2245           LDA             M_CLK_ADDR       ; GET THE PREVIOUS DCS ADDRESS
2248           MOV             B,A              ; PUT IN B REG
2249           MOV             A,C              ; GET CURRENT DCS ADDRESS
224A           ANI             <^C<CLK_CTRL_1_RO!-
224A                           CLK_CTRL_0_RO>>  ; DISCARD CLOCK STATE BITS
224C           STA             M_CLK_ADDR       ; SAVE FOR NEXT CHECK
224F           CMP             B                ; HAS IT CHANGED?
2250           JNZ             36$              ; BRANCH IF YES (CLOCK STILL RUNNING)
2253           LXI             H,CLK_STOP_MSG5  ; GET ADDRESS OF ERROR MESSAGE
2256           SHLD            CLK_MSG_ADDR     ; SAVE FOR TYPEOUT ROUTINE
2259  ;
2259  ; CONTROL C TYPED OR CLOCK STOPPED/HUNG. UNCONDITIONALLY STOP THE CLOCK
2259  ;
2259  37$:     LDA             DCS_CTRL_RE_CPY  ; GET COPY OF CONTROL REGISTER
225C           STA             DCS_CTRL_REG     ; STOP THE CLOCK
225F  ;
225F  ; SEE IF CONTROL C WAS TYPED
225F  ;
225F           LDA             SOFT_CTRL_REG    ; GET THE CONTROL C FLAG
2262           ANI             CONTROL_C_FLAG   ; IS IT SET?
2264           JZ              40$              ; BRANCH IF NO (CLOCK STOPPED)
2267           CALL            GET_CMD_LINE     ; CALL THE COMMAND PARSER
226A           JMP             30$              ; IF 'CO', START THE CLOCK AGAIN
226D  ;
```

```
226D  ; IF EXPECTED TRAP FLAG IS SET, SEE IF TRACE REGISTER IS OUTSIDE DCS SPACE.
226D  ;
226D  40$:    LDA         BURST_TEMP        ; RESTORE THE CONTROL C FLAG
2270          STA         SOFT_CTRL_REG     ; ...
2273          LDA         SOFT_CTRL_REGA    ; GET THE EXPECTED TRAP FLAG
2276          ANI         EXP_UTRAP_FLAG    ; ...
2278          JZ          42$               ; BRANCH IF FLAG NOT SET
227B          LDA         CS_ADD_TRAP_HGH   ; GET UPPER BYTE OF TRAP REGISTER
227E          ANI         <^X3F>            ; MASK
2280          CPI         <^X18>            ; GTR THAN 18?
2282          JM          55$               ; BRANCH IF OUT OF DCS SPACE
2285          CPI         <^X20>            ; LESS THAN 20?
2287          JP          55$               ; BRANCH IF OUT OF DCS SPACE
228A  ;
228A  ; SEE IF 'EXPECT STALL' FLAG IS SET
228A  ;
228A  42$:    LDA         SOFT_CTRL_REGB    ; GET THE EXPECTED STALL FLAG
228D          ANI         EXP_STALL_FLAG    ; ...
228F          JNZ         55$               ; BRANCH IF FLAG SET
2292  ;
2292  ; SEE IF THE CLOCK STOPPED BECAUSE OF A 'SOMM'
2292  ;
2292  45$:    LDA         SOFT_CTRL_REGA    ; GET THE SOMM FLAG
2295          ANI         SOMM_FLAG         ; IS IT SET?
2297          JZ          50$               ; BRANCH IF NO
229A  ;
229A  ; SOMM FLAG IS SET. SEE IF CLOCK STOPPED AT THE CORRECT PLACE
229A  ;
229A          LDA         DCS_CTRL_FILE     ; CHECK STOP CLOCK BIT IN CONTROL FILE
229D          ANI         STOP_CLOCK        ; ...
229F          JZ          50$               ; BRANCH IF NOT SOMM STOP
22A2  ;
22A2  ; CLOCK STOPPED BECAUSE OF 'SOMM'. TYPE THE 'SOMM MESSAGE' AND CALL THE
22A2  ; COMMAND PARSER
22A2  ;
22A2          TYPE        CRLF              ; ...
22AB          TYPE        SOMM_MSG          ; TYPE THE SOMM MESSAGE
22B4          TYPEW       SOMM_ADDRESS      ; TYPE THE ADDRESS
22BF          LDA         DCS_CTRL_FILE     ; CHECK IF RETURN TO ZERO BIT IS SET
22C2          ANI         DCS_ADDR_CLEAR    ; ...
22C4          JZ          47$               ; BRANCH IF ACTIVE
22C7          TYPE        COMMA_MSG         ; TYPE A COMMA
22D0          TYPE        DCS_ADDR_MSG      ; TYPE DCS ADDR =
22D9          LDA         DCS_ADDR_REG_RO   ; GET CURRENT DCS ADDR
22DC          ANI         <^C<CLK_CTRL_1_RO!-
22DC                      CLK_CTRL_0_RO>>   ; DISCARD CLOCK STATE BITS
22DE          DCR         A                 ; DECREMENT BY 1
22DF          STA         BURST_TEMP        ; SAVE THE ADDR
22E2          TYPEB       BURST_TEMP        ; PRINT THE CURRENT DCS ADDR
22ED  47$:    CALL        GET_CMD_LINE      ; CALL THE COMMAND PARSER
22F0          LDA         FRONT_PNL_2       ; GET REMOTE, FAULT, & TEST
22F3          ANI         <FAULT!TEST!REMOTE>
22F5          STA         RD_CTRL_REG       ; CLEAR MASTER HALT ENABLE
22F8          ORI         <TRAP_HALT_EN!-
22F8                      MASTER_HALT_EN>   ; RESET MASTER HALT ENABLE
```

```
22FA          STA             RD_CTRL_REG     ; ...
22FD          JMP             30$             ; IF 'CO' RESTART THE CLOCK
2300  ;
2300  ; SEE IF CLOCK STOPPED AT <DCS ADDRESS>
2300  ;
2300  50$:    LDA             DCS_CTRL_RE_CPY ; CLEAR THE CONTROL FILE
2303          ORI             CLEAR_CTRL_FILE ; ...
2305          STA             DCS_CTRL_REG    ; ...
2308          ANI             <^CCLEAR_CTRL_FILE>; ...
230A          STA             DCS_CTRL_REG    ; ...
230D          LDA             DCS_ADDR_REG_RO ; GET THE CURRENT DCS ADDRESS
2310          DCR             A               ; BACKUP BY ONE
2311          ANI             <^C<CLK_CTRL_1_RO!-
2311                          CLK_CTRL_0_RO>> ; DISCARD CLOCK STATE BITS
2313          MOV             B,A             ; SAVE
2314          LDA             ARG_LIST        ; GET THE <DCS ADDRESS>
2317          CMP             B               ; DID CLOCK STOP AT CORRECT PLACE?
2318          JNZ             60$             ; BRANCH IF NO
231B  ;
231B  ; CLOCK STOPPED CORRECTLY. EXIT
231B  ;
231B  55$:    RET                             ; EXIT
231C  ;
231C  ; CLOCK STOPPED WHEN IT WAS NOT SUPPOSE TO. TYPE AN ERROR MESSAGE AND
231C  ; CALL THE COMMAND PARSER
231C  ;
231C  60$:    TYPE            CRLF
2325          LHLD            CLK_MSG_ADDR    ; GET THE ERROR MESSAGE ADDRESS
2328          TYPE                            ; TYPE THE ERROR MESSAGE
232E          LDA             DCS_ADDR_REG_RO ; GET ADDRESS AT WHICH IT STOPPED
2331                                          ; OR HUNG
2331          ANI             <^C<CLK_CTRL_1_RO!-
2331                          CLK_CTRL_0_RO>> ; DISCARD CLOCK STATE BITS
2333          DCR             A               ; ...
2334          STA             BURST_TEMP      ; SAVE
2337          TYPEB           BURST_TEMP      ; TYPE THE ADDRESS
2342          TYPE            CLK_STOP_MSG2   ; TYPE '  TEST NUMBER= '
234B          TYPEB           TEST_NUMBER     ; TYPE THE TEST NUMBER
2356          TYPE            CRLF
235F          TYPE            CLK_STOP_MSG3   ; TYPE 'TPC= '
2368          LHLD            C_INIT_TEST_PC  ; GET COMPLIMENT INITIAL TEST PC
236B          XCHG
236C          LHLD            CURRENT_PC      ; GET THE CURRENT TEST PC
236F          DAD             D               ; SUBTRACT INITIAL TEST PC
2370          XCHG
2371          LHLD            TITLE_LENGTH    ; GET THE TITLE LENGTH
2374          DAD             D               ; ...
2375          INX             H               ; ACCOUNT FOR LENGTH BYTE
2376          SHLD            TYPE_ERR_BUFFER ; SAVE
2379          TYPEW           TYPE_ERR_BUFFER ; TYPE IT
2384          TYPE            CLK_STOP_MSG4   ; TYPE '  TRAP ADDRESS= '
238D          LHLD            CS_ADD_TRAP_LOW ; GET CONTENTS OF TRAP REGISTER
2390          MOV             A,H             ; DISCARD UPPER TWO BITS
2391          ANI             <^X3F>          ; OF TRAP REGISTER
2393          MOV             H,A             ; ...
```

```
2394            SHLD        TYPE_ERR_BUFFER ;
2397            TYPEW       TYPE_ERR_BUFFER ; TYPE THE TRAP ADDRESS
23A2    ;
23A2    ; type the index values if they are in use
23A2    ;
23A2            TYPE        TAB             ; TYPE 8 SPACES
23AB            LXI         H,0             ; CLEAR THE TEMP BUFFER
23AE            SHLD        TYPE_ERR_BUFFER ; ...
23B1            SHLD        TYPE_ERR_BUFFER+2 ; ...
23B4            XRA         A               ; GET THE I INDEX CODE IN A REGISTER
23B5            CALL        GET_INDEX_VALUE ; GET THE CURRENT VALUE OF I INDEX
23B8            JC          70$             ; BRANCH IF I NOT BEING USED
23BB            PUSH        B               ; PUT VALUE IN H & L
23BC            POP         H               ; ...
23BD            INX         H               ; MAKE LOOP COUNT START AT ONE
23BE            SHLD        TYPE_ERR_BUFFER ; SAVE I INDEX VALUE
23C1            TYPEW       TYPE_ERR_BUFFER ; TYPE THE I INDEX VALUE
23CC
23CC    70$:    MVI         A,2             ; GET J INDEX CODE
23CE            CALL        GET_INDEX_VALUE ; GET THE CURRENT VALUE OF I INDEX
23D1            JC          71$             ; BRANCH IF I NOT BEING USED
23D4            PUSH        B               ; PUT VALUE IN H & L
23D5            POP         H               ; ...
23D6            INX         H               ; MAKE LOOP COUNT START AT ONE
23D7            SHLD        TYPE_ERR_BUFFER ; SAVE I INDEX VALUE
23DA            TYPE        COMMA_MSG       ; TYPE A COMMA SPACE
23E3            TYPEW       TYPE_ERR_BUFFER ; TYPE THE I INDEX VALUE
23EE
23EE    71$:    MVI         A,4             ; GET K INDEX CODE
23F0            CALL        GET_INDEX_VALUE ; GET THE CURRENT VALUE OF I INDEX
23F3            JC          61$             ; BRANCH IF I NOT BEING USED
23F6            PUSH        B               ; PUT VALUE IN H & L
23F7            POP         H               ; ...
23F8            INX         H               ; MAKE LOOP COUNT START AT ONE
23F9            SHLD        TYPE_ERR_BUFFER ; SAVE I INDEX VALUE
23FC            TYPE        COMMA_MSG       ; TYPE A COMMA SPACE
2405            TYPEW       TYPE_ERR_BUFFER ; TYPE THE I INDEX VALUE
2410    61$:    CALL        GET_CMD_LINE    ; CALL THE COMMAND PARSER
2413            JMP         61$             ; DON'T ALLOW CONTINUE
2416
2416
```

```
2416            .SBTTL  "  END TEST ROUTINE
2416    ;++
2416    ;
2416    ; FUNCTIONAL DESCRIPTION:
2416    ;
2416    ;       THIS ROUTINE IS USED TO TERMINATE A TEST. IT SHOULD ALWAYS BE
2416    ;       THE LAST INSTRUCTION OF THE TEST (EXCEPT FOR THE LAST TEST
2416    ;       OF A TEST SEQUENCE). FOLLOWING IS THE ALGORITHM FOR THIS ROUTINE:
2416    ;
2416    ;       READ THE NEXT OVERLAY AND EXIT.
2416    ;
2416    ; CALLING SEQUENCE:
2416    ;
2416    ;       ENDTEST
2416    ;
2416    ; IMPLICIT INPUTS:
2416    ;
2416    ;       TEST_PC         - CONTAINS THE CURRENT TEST PC
2416    ;
2416    ; IMPLICIT OUTPUTS:
2416    ;
2416    ;       NONE
2416    ;
2416    ;--
2416
2416    END_TEST:
2416            LDA             TEST_NUMBER     ; GET THE CURRENT TEST NUMBER
2419            MOV             B,A             ; ...
241A            LDA             TOTAL_NMB_TESTS ; GET THE TOTAL NUMBER OF TESTS
241D            CMP             B               ; MORE TESTS TO EXECUTE?
241E            JZ              10$             ; BRANCH IF NO
2421            CALL            CPU_INIT        ; INIT THE CPU
2424            CALL            READ_OVERLAY    ; READ IN THE NEXT TEST
2427            LHLD            INIT_TEST_PC    ; INITIALIZE THE TEST PC
242A            SHLD            TEST_PC         ; ...
242D    10$:    RET                             ; EXIT
242E
```

```
242E          .SBTTL  "  END PASS ROUTINE
242E  ;++
242E  ;
242E  ; FUNCTIONAL DESCRIPTION:
242E  ;
242E  ;       THIS ROUTINE INCREMENTS THE PASS COUNT AND RETURNS TO THE
242E  ;       PROGRAM_INIT ROUTINE.
242E  ;
242E  ; CALLING SEQUENCE:
242E  ;
242E  ;       ENDPASS
242E  ;
242E  ; IMPLICIT INPUTS:
242E  ;
242E  ;       PASS_COUNT      - CONTAINS THE CURRENT PASS COUNT
242E  ;
242E  ; IMPLICIT OUTPUTS:
242E  ;
242E  ;       PASS_COUNT      - CONTAINS THE CURRENT PASS COUNT PLUS 1
242E  ;
242E  ;--
242E
242E  END_PASS:
242E          LDA             PASS_COUNT      ; GET THE PASS COUNT
2431          INR             A               ; INCREMENT
2432          STA             PASS_COUNT      ; SAVE
2435          POP             B               ; THROW AWAY PC OF CALL
2436          RET                             ; RETURN TO PROGRAM_INIT ROUTINE
2437
```

```
2437             .SBTTL  "  LOAD FIELD ROUTINE
2437       ;++
2437       ;
2437       ; FUNCTIONAL DESCRIPTION:
2437       ;
2437       ;        THIS ROUTINE IS USED TO LOAD A FIELD OF A MICRO WORD. IF BIT 15
2437       ;        OF EITHER THE STARTING BIT NUMBER OR ENDING BIT NUMBER ARGUEMNT
2437       ;        IS SET, THE BIT NUMBER IS A POINTER (OPTIONALLY INDEXED) TO THE
2437       ;        REQUESTED BIT NUMBER.
2437       ;
2437       ; CALLING SEQUENCE:
2437       ;
2437       ;        LDFIELD SRC_ADDRESS,[SRC_INDEX],[DATA_TYPE],DEST_ADDRESS,START_BIT,END_BIT -
cont> ,[START_BIT_INDEX],[END_BIT_INDEX],[INHIBIT_PARITY]
2437       ;
2437       ; IMPLICIT INPUTS:
2437       ;
2437       ;        ARGUMENT LIST:
2437       ;
2437       ;                OFFSET 0 - SOURCE ADDRESS LOW
2437       ;                OFFSET 1 - SOURCE ADDRESS HIGH
2437       ;                OFFSET 2 - SOURCE INDEX CODE
2437       ;                OFFSET 3 - DATA TYPE CODE
2437       ;                OFFSET 4 - DESTINATION ADDRESS LOW
2437       ;                OFFSET 5 - DESTINATION ADDRESS HIGH
2437       ;                OFFSET 6 - STARTING BIT NUMBER LOW
2437       ;                OFFSET 7 - STARTING BIT NUMBER HIGH
2437       ;                OFFSET 8 - ENDING BIT NUMBER LOW
2437       ;                OFFSET 9 - ENDING BIT NUMBER HIGH
2437       ;                OFFSET 10- LONLIT FLAG
2437       ;                OFFSET 11- STARTING BIT NUMBER INDEX CODE
2437       ;                OFFSET 12- ENDING BIT NUMBER INDEX CODE
2437       ;                OFFSET 13- INHIBIT PARITY CODE (-1 EQL INHIBIT)
2437       ;                OFFSET 14- INHIBIT SCRAMBLE CODE (-1 EQL INHIBIT)
2437       ;
2437       ; IMPLICIT OUTPUTS:
2437       ;
2437       ;        THE SPECIFIED FIELD OF THE DESTINATION IS LOADED WITH THE SOURCE DATA
2437       ;
2437       ;--
2437
2437       LOAD_FIELD:
2437             LDA        ARG_LIST+2       ; GET THE INDEX CODE
243A             CALL       GET_INDEX_VALUE  ; GET THE CURRENT VALUE
243D             LDA        ARG_LIST+3       ; GET THE DATA TYPE
2440             CALL       MULTIPLY_A_BC    ; MULTIPLY DATA TYPE BY INDEX VALUE
2443       ;
2443       ; D & E NOW CONTAIN THE INDEX VALUE FOR THE SPECIFIED DATA TYPE
2443       ;
2443             LHLD       ARG_LIST         ; GET THE SOURCE ADDRESS
2446             DAD        D                ; INDEX IT
2447             XCHG
2448             LHLD       INIT_TEST_PC     ; GET INITIAL TEST PC
244B             DAD        D                ; RELOCATE SOURCE ADDRESS
244C       ;
244C       ; CHECK IF LONLIT SPECIFIED
```

```
244C   ;
244C                LDA          ARG_LIST+10      ; GET THE LONLIT FLAG
244F                ORA          A                ; SET THE CONDITION CODES
2450                JM           5$               ; BRANCH IF NO LONLIT
2453                LXI          D,TEMP_BUFFER    ; PUT COMPLIMENT DATA IN TEMP BUFFER
2456                MVI          C,4              ; SETUP LOOP TO MOVE 4 BYTES
2458   3$:          MOV          A,M              ; GET A BYTE OF THE SOURCE
2459                CMA                           ; COMPLIMENT IT
245A                STAX         D                ; PUT IN TEMP BUFFER
245B                INX          D                ; INCREMENT THE POINTERS
245C                INX          H                ; ...
245D                DCR          C                ; DONE 4 BYTES?
245E                JNZ          3$               ; BRANCH IF NO
2461                LXI          H,TEMP_BUFFER    ; PUT ADDRESS OF DATA IN H & L
2464   ;
2464   ; GET THE DESTINATION ADDRESS AND INSERT THE FIELD
2464   ;
2464   5$:          PUSH         H                ; SAVE SOURCE ADDRESS
2465                LHLD         ARG_LIST+4       ; GET THE DESTINATION ADDRESS
2468                XCHG
2469                LHLD         INIT_TEST_PC     ; GET INITIAL TEST PC
246C                DAD          D                ; RELOCATE DESTINATION ADDRESS
246D                XCHG
246E   ;
246E   ; D & E CONTAIN DESTINATION ADDRESS
246E   ; (SP)  CONTAINS THE SOURCE ADDRESS
246E   ;
246E   ; GET THE STARTING AND ENDING BIT NUMBERS
246E   ;
246E                PUSH         D                ; SAVE D & E
246F                LHLD         ARG_LIST+6       ; GET THE STARTING BIT NUMBER
2472                MOV          A,H              ; SEE IF BIT 15 SET
2473                ANI          <^X80>           ; ...
2475                JZ           10$              ; BRANCH IF ABSOLUTE DATA
2478                MOV          A,H              ; CLEAR BIT 15
2479                ANI          <^X7F>           ; ...
247B                MOV          H,A              ; ...
247C                PUSH         H                ; SAVE ADDRESS
247D                LDA          ARG_LIST+11      ; GET STARTING INDEX CODE
2480                CALL         GET_INDEX_VALUE  ; GET THE CURRENT INDEX VALUE
2483                POP          H                ; GET ADDRESS BACK
2484                DAD          B                ; ADD CURRENT INDEX VALUE
2485                XCHG
2486                LHLD         INIT_TEST_PC     ; GET THE INITIAL TEST PC
2489                DAD          D                ; ADD TO INDEX VALUE ADDRESS
248A                MOV          L,M              ; GET STARTING BIT NUMBER
248B   10$:         MOV          B,L              ; PUT STARTING BIT IN B REG
248C   ;
248C   ; GET THE END VALUE
248C   ;
248C                LHLD         ARG_LIST+8       ; GET THE ENDING BIT NUMBER
248F                MOV          A,H              ; SEE IF BIT 15 SET
2490                ANI          <^X80>           ; ...
2492                JZ           15$              ; BRANCH IF ABSOLUTE DATA
2495                MOV          A,H              ; CLEAR BIT 15
```

```
2496               ANI     <^X7F>           ; ...
2498               MOV     H,A              ; ...
2499               PUSH    B                ; SAVE STARTING BIT VALUE
249A               PUSH    H                ; SAVE ADDRESS
249B               LDA     ARG_LIST+12      ; GET STARTING INDEX CODE
249E               CALL    GET_INDEX_VALUE  ; GET THE CURRENT INDEX VALUE
24A1               POP     H                ; GET ADDRESS BACK
24A2               DAD     B                ; ADD CURRENT INDEX VALUE
24A3               XCHG
24A4               LHLD    INIT_TEST_PC     ; GET THE INITIAL TEST PC
24A7               DAD     D                ; ADD TO INDEX VALUE ADDRESS
24A8               MOV     L,M              ; GET STARTING BIT NUMBER
24A9               POP     B                ; RESTORE STARTING BIT VALUE
24AA      15$:     MOV     C,L              ; PUT ENDING BIT VALUE IN C REG
24AB               POP     D                ; RESTORE DESTINATION ADDRESS
24AC               POP     H                ; RESTORE SOURCE ADDRESS
24AD               LDA     ARG_LIST+14      ; GET THE INHIBIT SCRAMBLE FLAG
24B0               PUSH    $PSW             ; PASS IT ON THE STACK
24B1               LDA     ARG_LIST+13      ; GET THE INHIBIT PARITY FLAG
24B4               CALL    INSERT_FIELD     ; GENERATE THE FIELD
24B7               RET
24B8
```

```
24B8              .SBTTL  "  LOAD INDEX ROUTINE
24B8      ;++
24B8      ;
24B8      ; FUNCTIONAL DESCRIPTION:
24B8      ;
24B8      ;       THIS ROUTINE INITIALIZES THE 'STARTING_VALUE' OF THE 'LOOP_NAME'
24B8      ;       PARAMETER.
24B8      ;
24B8      ; CALLING SEQUENCE:
24B8      ;
24B8      ;       LDINDEX         LOOP_NAME,STARTING_VALUE
24B8      ;
24B8      ; IMPLICIT INPUTS:
24B8      ;
24B8      ;       ARGUMENT LIST:
24B8      ;
24B8      ;               OFFSET 0 - INDEX NAME CODE
24B8      ;               OFFSET 1 - STARTING VALUE LOW
24B8      ;               OFFSET 2 - STARTING VALUE HIGH
24B8      ;               OFFSET 3 - VALUE FLAG (0  = LITERAL OTHERWISE, ADDRESS)
24B8      ;               OFFSET 4 - VALUE INDEX CODE (ONLY USED IF VALUE FLAG IS NON 0)
24B8      ;
24B8      ;       INDEX_NAM_TBL - TABLE THAT CONTAINS THE ADDRESS OF THE I, J, OR
24B8      ;                       K INDEX TABLES.
24B8      ;
24B8      ; IMPLICIT OUTPUTS:
24B8      ;
24B8      ;       THE I, J, OR K STARTING VALUE IS INITIALIZED.
24B8      ;
24B8      ;--
24B8
24B8      LOAD_INDEX:
24B8              LDA             ARG_LIST        ; GET THE INDEX NAME BYTE
24BB              CALL            GET_INDEX_VALUE ; GET ADDRESS OF THE INDEX TABLE
24BE              PUSH            H               ; SAVE ADDRESS OF TABLE
24BF              LDA             ARG_LIST+3      ; GET THE VALUE FLAG
24C2              ORA             A               ; SET THE CONDITION CODES
24C3              JZ              10$             ; BRANCH IF VALUE IS A LITERAL
24C6      ;
24C6      ; VALUE TO LOAD IS IN A TABLE
24C6      ;
24C6              LDA             ARG_LIST+4      ; GET THE INDEX CODE FOR THE VALUE
24C9              CALL            GET_INDEX_VALUE ; GET THE CURRENT VALUE
24CC              JC              5$              ; BRANCH IF NO INDEX SPECIFIED
24CF              MVI             A,2             ; MULTIPLY INDEX VALUE BY 2
24D1              CALL            MULTIPLY_A_BC   ; ...
24D4              LHLD            ARG_LIST+1      ; GET THE TABLE ADDRESS
24D7              DAD             D               ; INDEX BY INDEX VALUE
24D8              XCHG                            ; PUT ADDRESS IN D & E
24D9              LHLD            INIT_TEST_PC    ; GET RELOCATION FACTOR
24DC              DAD             D               ; RELOCATE TABLE ADDRESS
24DD              XCHG                            ; PUT IN D & E
24DE              POP             H               ; RESTORE ADDRESS OF INDEX TABLE
24DF              JMP             20$             ; LOAD THE VALUE
24E2      ;
```

```
24E2   ; COME HERE WHEN THERE IS AND ADDRESS THAT POINTS TO THE VALUE TO BE LOADED
24E2   ;          AND
24E2   ; THERE IS NO INDEX SPECIFIED TO INDEX INTO THAT ADDRESS WITH
24E2   ;
24E2   5$:        LHLD          ARG_LIST+1        ; GET THE ADDRESS
24E5              XCHG                            ; PUT THE ADDRESS IN D & E
24E6              LHLD          INIT_TEST_PC      ; GET THE RELOCATION FACTOR
24E9              DAD           D                 ; RELOCATE THE ADDRESS
24EA              XCHG                            ; PUT IN D & E
24EB              POP           H                 ; RESTORE ADDRESS OF INDEX TABLE
24EC              JMP           20$               ; LOAD THE VALUE
24EF   ;
24EF   ; (SP) - ADDRESS OF INDEX TABLE
24EF   ; ARG LIST CONTAINS THE VALUE TO LOAD
24EF   ;
24EF   10$:       POP           H                 ; RESTORE ADDRESS OF INDEX TABLE
24F0              LXI           D,ARG_LIST+1      ; GET ADDRESS OF STARTING VALUE
24F3   ;
24F3   ; H & L - CONTAIN THE ADDRESS OF THE INDEX TABLE
24F3   ; D & E - CONTAIN THE ADDRESS OF THE VALUE TO LOAD
24F3   ;
24F3   20$:       LDAX          D                 ; GET LOW BYTE OF STARTING VALUE
24F4              MOV           M,A               ; PUT IN INDEX TABLE
24F5              INX           D                 ; INCREMENT THE POINTERS
24F6              INX           H                 ; ...
24F7              LDAX          D                 ; GET HIGH BYTE OF STARTING VALUE
24F8              MOV           M,A               ; STORE IN INDEX TABLE
24F9              RET                             ; EXIT
24FA
```

```
24FA            .SBTTL  "  INCREMENT INDEX ROUTINE
24FA    ;++
24FA    ;
24FA    ; FUNCTIONAL DESCRIPTION:
24FA    ;
24FA    ;       THIS ROUTINE INCREMENTS THE CURRENT VALUE OF THE 'LOOP_NAME'
24FA    ;       INDEX.
24FA    ;
24FA    ; CALLING SEQUENCE:
24FA    ;
24FA    ;       INCINDEX        LOOP_NAME
24FA    ;
24FA    ; IMPLICIT INPUTS:
24FA    ;
24FA    ;       ARGUMENT LIST:
24FA    ;
24FA    ;               OFFSET 0 - INDEX NAME CODE
24FA    ;
24FA    ;       INDEX_NAME_TBL
24FA    ;
24FA    ;--
24FA
24FA    INCREMENT_INDEX:
24FA            LDA             ARG_LIST        ; GET THE INDEX NAME CODE
24FD            CALL            GET_INDEX_VALUE ; GET THE CURRENT INDEX VALUE
2500            INX             B               ; INCREMENT THE CURRENT VALUE
2501            MOV             M,C             ; PUT BACK IN THE TABLE
2502            INX             H               ; ...
2503            MOV             M,B             ; ...
2504            RET                             ; EXIT
2505
```

```
2505          .SBTTL  "  SAVE INDEX ROUTINE
2505    ;++
2505    ;
2505    ; FUNCTIONAL DESCRIPTION:
2505    ;
2505    ;       THIS ROUTINE SAVES THE CURRENT VALUE OF THE SPECIFIED
2505    ;       INDEX NAME IN THE SPECIFIED ADDRESS.
2505    ;
2505    ; CALLING SEQUENCE:
2505    ;
2505    ;       SAVEINDEX       INDEX_NAME,ADDRESS
2505    ;
2505    ; IMPLICIT INPUTS:
2505    ;
2505    ;       ARGUMENT LIST:
2505    ;
2505    ;               OFFSET 0 - INDEX NAME CODE
2505    ;               OFFSET 1 - ADDRESS LOW
2505    ;               OFFSET 2 - ADDRESS HIGH
2505    ;
2505    ;--
2505
2505    SAVE_INDEX:
2505            LDA             ARG_LIST        ; GET INDEX NAME CODE
2508            CALL            GET_INDEX_VALUE ; GET THE CURRENT VALUE
250B            LHLD            ARG_LIST+1      ; GET DESTINATION ADDRESS
250E            XCHG                            ; PUT IN D & E
250F            LHLD            INIT_TEST_PC    ; RELOCATE THE ADDRESS
2512            DAD             D               ; ...
2513            MOV             M,C             ; SAVE THE INDEX VALUE
2514            INX             H               ; ...
2515            MOV             M,B             ; ...
2516            RET                             ; EXIT
2517
```

```
2517           .SBTTL  "  ENABLE MICRO TRAP ROUTINE
2517    ;++
2517    ;
2517    ; FUNCTIONAL DESCRIPTION:
2517    ;
2517    ;       THIS PSEUDO INSTRCUTION SETS A SOFTWARE FLAG THAT CONTROLS THE
2517    ;       BRSTCLOCK ROUTINE.
2517    ;
2517    ; CALLING SEQUENCE:
2517    ;
2517    ;       EXPUTRAP
2517    ;
2517    ; IMPLICIT INPUTS:
2517    ;
2517    ;       ARGUMENT LIST:
2517    ;
2517    ;               NONE
2517    ;
2517    ;--
2517
2517    ENABLE_TRAP:
2517            LDA             SOFT_CTRL_REGA  ; GET THE CONTROL REGISTER
251A            ORI             EXP_UTRAP_FLAG  ; SET THE FLAG
251C            STA             SOFT_CTRL_REGA  ; RESTORE THE CONTROL REGISTER
251F            RET
2520
```

```
2520            .SBTTL  "  ENABLE MICRO STALL ROUTINE
2520    ;++
2520    ;
2520    ; FUNCTIONAL DESCRIPTION:
2520    ;
2520    ;       THIS PSEUDO INSTRCUTION SETS A SOFTWARE FLAG THAT CONTROLS THE
2520    ;       BRSTCLOCK ROUTINE.
2520    ;
2520    ; CALLING SEQUENCE:
2520    ;
2520    ;       EXPSTALL
2520    ;
2520    ; IMPLICIT INPUTS:
2520    ;
2520    ;       ARGUMENT LIST:
2520    ;
2520    ;               NONE
2520    ;
2520    ;--
2520
2520    ENABLE_STALL:
2520            LDA             SOFT_CTRL_REGB  ; GET THE CONTROL REGISTER
2523            ORI             EXP_STALL_FLAG  ; SET THE FLAG
2525            STA             SOFT_CTRL_REGB  ; RESTORE THE CONTROL REGISTER
2528            RET
2529
```

```
2529            .SBTTL  "  ERROR LOG ROUTINE
2529      ;++
2529      ;
2529      ; FUNCTIONAL DESCRIPTION:
2529      ;
2529      ;       THIS ROUTINE IS USED TO INSERT ENTRIES IN THE ERROR LOG. THE ERROR
2529      ;       LOG KEEPS TRACK OF SINGLE BIT ERRORS IN THE DYNAMIC MEMORY TEST.
2529      ;
2529      ;       THE LOG CONSISTS OF 260 BYTES LOCATED AT THE VERY TOP OF MEMORY.
2529      ;       THE FIRST 4 BYTES CONTAINS THE NUMBER OF ENTRIES IN THE LOG.
2529      ;
2529      ;       IF THE INIT_ERR_LOG FLAG IS SET WHEN THIS ROUTINE IS CALLED, THE
2529      ;       NUMBER OF ENTRIES IN THE LOG IS SET TO ZERO AND THE FLAG CLEARED
2529      ;       BEFORE THE CURRENT DATA IS PLACED IN THE LOG. THE INIT_ERR_LOG FLAG
2529      ;       IS SET BY A 'DIAGNOSE' COMMAND.
2529      ;
2529      ; CALLING SEQUENCE:
2529      ;
2529      ;       ERRLOG
2529      ;
2529      ; IMPLICIT INPUTS:
2529      ;
2529      ;       ARGUMENT LIST:
2529      ;
2529      ;               NONE
2529      ;
2529      ;       TEST_SIZE - CONTAINS THE NUMBER OF RECORDS (128 BYTES EACH) IN THE
2529      ;                   CURRENT TEST. THE TEST MUST NOT OVERFLOW INTO THE LOG.
2529      ;
2529      ;       WH-REGISTER - CONTAINS PHYSICAL ADDRESS OF ERROR
2529      ;       MH-REGISTER - CONTAINS SYNDROME OF FAILURE IN BITS <6:0>
2529      ;
2529      ; ROUTINE VALUE:
2529      ;
2529      ;       ERROR_FLAG - SET IF ERROR LOG IS FULL.
2529      ;--
2529      ERROR_LOG:
2529            LDA             TEST_SIZE       ; GET CURRENT TEST SIZE
252C            MOV             C,A             ; SETUP TO CALCULATE # OF BYTES
252D            XRA             A               ; ...
252E            MOV             B,A             ; ...
252F            MVI             A,128           ; ...
2531            CALL            MULTIPLY_A_BC   ; GET BYTE LENGTH OF TEST IN D & E
2534            LXI             H,TEST_BUFFER   ; GET STARTING ADDRESS OF TEST
2537            DAD             D               ; GET ENDING ADDRESS OF TEST
2538            MOV             A,H             ; CHECK IF ERROR LOG OVERFLOW
2539            CPI             <^XB6>          ; CHECK UPPER 8 BITS
253B            JZ              6$              ; BRANCH TO CHECK LOW HALF
253E            JNC             1$              ; BRANCH IF TOO BIG
2541            JMP             5$              ; OTHERWISE ITS OK
2544      6$:   MOV             A,L             ; GET LOW HALF
2545            CPI             <^XFC>          ; CHECK LOW HALF
2547            JC              5$              ; BRANCH IF OK
254A      ;
254A      ; TEST OVERFLOWS ERROR LOG
```

```
254A  ;
254A  1$:     TYPE        CRLF
2553          TYPE        LOG_MSG_1          ; TYPE THE ERROR MESSAGE
255C          TYPE        CRLF
2565          JMP         LOG_FAIL           ; EXIT
2568  ;
2568  ; CHECK IF ERR_LOG_INIT SET
2568  ;
2568  5$:     LXI         H,ERROR_LOG_ADR    ; GET START ADDRESS OF ERROR LOG B6FC(X)
256B          MVI         E,4                ; SETUP ERROR LOG INCREMENT
256D          MVI         D,0                ; ...
256F          LDA         SOFT_CTRL_REGB     ; GET THE FLAG
2572          MOV         B,A                ; SAVE
2573          ANI         ERR_LOG_INIT       ; IS FLAG SET?
2575          JZ          10$                ; BRANCH IF NO
2578          MOV         M,D                ; SET LOG COUNT TO ZERO
2579          MOV         A,B                ; CLEAR THE FLAG
257A          ANI         <^C<ERR_LOG_INIT>>; ...
257C          STA         SOFT_CTRL_REGB     ; ...
257F  ;
257F  ; CHECK IF ERROR LOG FULL
257F  ;
257F  10$:    MOV         A,M                ; GET NUMBER OF ENTRIES IN LOG
2580          CPI         64                 ; LOG FULL?
2582          JZ          LOG_FAIL           ; BRANCH IF YES
2585  ;
2585  ; SEARCH THE ERROR LOG FOR THIS PHYSICAL ADDRESS
2585  ;
2585          MOV         B,A                ; GET NUMBER OF ENTRIES IN LOG
2586          LDA         WH_REG_BYTE_0      ; GET THE PHYSICAL ADDRESS
2589          STA         LOG_ADDRESS        ; ...
258C          LDA         WH_REG_BYTE_1      ; ...
258F          STA         LOG_ADDRESS+1      ; ...
2592          LDA         WH_REG_BYTE_2      ; ...
2595          STA         LOG_ADDRESS+2      ; ...
2598          LDA         MH_REG_BYTE_0      ; GET THE SYNDROME
259B          ANI         <^X7F>             ; ...
259D          STA         LOG_SYNDROME       ; SAVE
25A0          MOV         A,B                ; CHECK IF LOG EMPTY
25A1          ORA         A                  ; ...
25A2          JZ          25$                ; BRANCH IF LOG EMPTY
25A5  ;
25A5  ; SCAN THE LOG
25A5  ;
25A5  15$:    DAD         D                  ; POINT AT NEXT ENTRY
25A6          LDA         LOG_ADDRESS        ; GET CURRENT LOG ADDRESS BYTE 0
25A9          CMP         M                  ; THIS ITEM IN LOG?
25AA          JNZ         20$                ; BRANCH IF NO
25AD          LDA         LOG_ADDRESS+1      ; GET NEXT BYTE OF CURRENT ADDRESS
25B0          INX         H
25B1          CMP         M                  ; STILL THIS ITEM?
25B2          DCX         H
25B3          JNZ         20$                ; BRANCH IF NO
25B6          LDA         LOG_ADDRESS+2      ; GET 3RD BYTE OF CURRENT ADDRESS
25B9          INX         H
```

```
25BA          INX     H
25BB          CMP     M                 ; STILL THIS ITEM?
25BC          DCX     H
25BD          DCX     H
25BE          JNZ     20$               ; BRANCH IF NO
25C1          LDA     LOG_SYNDROME      ; GET CURRENT SYNDROME
25C4          INX     H
25C5          INX     H
25C6          INX     H
25C7          CMP     M                 ; THIS SYNDROME?
25C8          JZ      LOG_SUCCESS       ; BRANCH IF YES
25CB          DCX     H
25CC          DCX     H
25CD          DCX     H
25CE  20$:    DCR     B                 ; CHECKED ALL ENTRIES IN LOG?
25CF          JNZ     15$               ; BRANCH IF NO
25D2  ;
25D2  ; CHECKED ALL ENTRIES IN ERROR LOG AND THIS ADDRESS AND SYNDROME ARE NOT
25D2  ; THERE. PUT THEM IN THE ERROR LOG
25D2  ;
25D2  25$:    DAD     D                 ; POINT AT FREE ELEMENT IN LOG
25D3          LDA     LOG_ADDRESS       ; PUT ADDRESS AND SYNDROME IN LOG
25D6          MOV     M,A               ; ...
25D7          INX     H
25D8          LDA     LOG_ADDRESS+1     ; ...
25DB          MOV     M,A               ; ...
25DC          INX     H
25DD          LDA     LOG_ADDRESS+2     ; ...
25E0          MOV     M,A               ; ...
25E1          INX     H
25E2          LDA     LOG_SYNDROME      ; ...
25E5          MOV     M,A               ; ...
25E6          LXI     H,ERROR_LOG_ADR ; GET ADDRESS OF LOG COUNT B6FC(X)
25E9          MOV     A,M               ; GET LOG COUNT
25EA          INR     A                 ; COUNT THIS ENTRY
25EB          MOV     M,A               ; SAVE IN LOG
25EC          JMP     LOG_SUCCESS       ; EXIT
25EF  LOG_FAIL:
25EF          LDA     SOFT_CTRL_REGA    ; GET ERROR FLAGS
25F2          ORI     <ERROR_FLAG!-
25F2                  LOOP_ERROR_FLAG>; SET THE FLAGS
25F4          ANI     <^CVBUS_COMPARE>; ...
25F6          STA     SOFT_CTRL_REGA    ; ...
25F9          JMP     LOG_EXIT
25FC  LOG_SUCCESS:
25FC          LDA     SOFT_CTRL_REGA    ; GET THE FLAGS
25FF          ANI     <^C<LOOP_ERROR_FLAG!-
25FF                  VBUS_COMPARE>>    ; CLEAR THEM
2601          STA     SOFT_CTRL_REGA    ; ...
2604          CALL    CHECK_QA_FLAG     ; CHECK IF QA FLAG SET
2607  LOG_EXIT:
2607          RET                       ; EXIT
2608
```

```
2608            .SBTTL  "  PRINT STRING ROUTINE
2608    ;++
2608    ;
2608    ; FUNCTIONAL DESCRIPTION:
2608    ;
2608    ;       THIS ROUTINE IS USED TO PRINT A COUNTED ASCII STRING
2608    ;
2608    ; CALLING SEQUENCE:
2608    ;
2608    ;       PRINT_S         POINTER
2608    ;
2608    ; IMPLICIT INPUTS:
2608    ;
2608    ;       ARGUMENT LIST:
2608    ;
2608    ;               OFFSET 0 - POINTER OFFSET LOW
2608    ;               OFFSET 1 - POINTER OFFSET HIGH
2608    ;--
2608    PRINT_S:
2608            LHLD            ARG_LIST                ; GET OFFSET TO STRING
260B            XCHG                                    ; PUT IN D REG
260C            LHLD            INIT_TEST_PC            ; GET OFFSET TO TEST DATA
260F            DAD             D                       ; GET ADDRESS OF STRING
2610            TYPE                                    ; TYPE THE STRING
2616            RET                                     ; EXIT
2617
```

```
2617          .SBTTL   " PRINT NUMERIC ROUTINE
2617  ;++
2617  ;
2617  ; FUNCTIONAL DESCRIPTION:
2617  ;
2617  ;       THIS ROUTINE IS USED TO CONVERT A BINARY NUMBER TO A HEXIDECIMAL
2617  ;       ASCII STRING AND TYPE IT.
2617  ;
2617  ; CALLING SEQUENCE:
2617  ;
2617  ;       PRINT_N         REG_ADDRESS,DATA_TYPE
2617  ;
2617  ; IMPLICIT INPUTS:
2617  ;
2617  ;       ARGUMENT LIST:
2617  ;
2617  ;               OFFSET 0 - REGISTER ADDRESS LOW
2617  ;               OFFSET 1 - REGISTER ADDRESS HIGH
2617  ;               OFFSET 2 - DATA TYPE
2617  ;--
2617  PRINT_N:
2617          LHLD            ARG_LIST                ; GET THE REGISTER ADDRESS
261A          LDA             ARG_LIST+2              ; GET THE DATA TYPE
261D          CPI             1                       ; BYTE?
261F          JNZ             10$                     ; BRANCH IF NO
2622          TYPEB                                   ; PRINT THE DATA
262A          JMP             PRINT_N_EX              ; EXIT
262D  10$:    CPI             2                       ; WORD?
262F          JNZ             20$                     ; BRANCH IF NO
2632          TYPEW                                   ; PRINT THE DATA
263A          JMP             PRINT_N_EX              ; EXIT
263D  20$:    TYPEL                                   ; TYPE THE DATA
2645  PRINT_N_EX:
2645          RET                                     ; EXIT
```

```
2646          .SBTTL  DUMP ERROR LOG ROUTINE
2646  ;++
2646  ;
2646  ; FUNCTIONAL DESCRIPTION:
2646  ;
2646  ;       THIS ROUTINE IS USED TO TYPE THE NUMBER OF ERRORS IN THE
2646  ;       ERROR LOG.
2646  ;--
2646  DUMP_LOG:
2646          LDA             SOFT_CTRL_REGB  ; GET ENABLE FLAG
2649          ANI             SBE_FLAG        ; IS FLAG SET?
264B          JZ              DUMP_RET        ; BRANCH IF NO
264E          TYPE            CRLF
2657          LXI             H,ERROR_LOG_ADR ; GET ADDRESS OF ERROR LOG COUNT B6FC(X)
265A          LDA             SOFT_CTRL_REGB  ; SEE IF ERROR LOG WAS INITIALIZED
265D          ANI             ERR_LOG_INIT    ; ...
265F          JZ              2$              ; BRANCH IF IT WAS INITIALIZED
2662          XRA             A               ; INIT IT
2663          MOV             M,A             ; ...
2664  2$:     MOV             A,M             ; GET NUMBER OF ERRORS
2665          CPI             64              ; IS IT FULL?
2667          JNZ             1$              ; BRANCH IF NO
266A          TYPE            LOG_MSG_3       ; TYPE ERROR LOG FULL MESSAGE
2673          JMP             DUMP_EXIT       ; EXIT
2676  1$:     PUSH            H               ; SAVE ADDRESS OF LOG
2677          TYPE            LOG_MSG_2       ; TYPE THE MESSAGE
2680          POP             H               ; GET ADDRESS OF LOG
2681          TYPEB                           ; TYPE NUMBER OF ERRORS (H&L CONTAIN
2689                                          ; ADDRESS)
2689  DUMP_EXIT:
2689          TYPE            CRLF
2692  DUMP_RET:
2692          LDA             SOFT_CTRL_REGB  ; SET THE ERROR LOG INIT FLAG
2695          ORI             ERR_LOG_INIT    ; ...
2697          STA             SOFT_CTRL_REGB  ; ...
269A          RET
269B
```

```
269B            .SBTTL  "INTERPRETER SUBROUTINES
269B            .SBTTL  "  INITIALIZE LOOP VALUE ROUTINE
269B    ;++
269B    ;
269B    ; FUNCTIONAL DESCRIPTION:
269B    ;
269B    ;       THIS ROUTINE SETS THE CURRENT VALUE OF ALL THE LOOP NAMES TO
269B    ;       MINUS 1.
269B    ;
269B    ; CALLING SEQUENCE:
269B    ;
269B    ;       CALL    INIT_LOOP_VALUE
269B    ;
269B    ;       THIS ROUTINE IS CALLED BY:
269B    ;
269B    ;               NEW_TEST
269B    ;               SUB_TEST
269B    ;
269B    ; IMPLICIT INPUTS:
269B    ;
269B    ;       NONE
269B    ;
269B    ; IMPLICIT OUTPUTS:
269B    ;
269B    ;       THE CURRENT VALUE OFALL THELOOP NAMES IS INITIALIZED
269B    ;
269B    ;--
269B
269B    INIT_LOOP_VALUE:
269B            LXI             H,I_INDEX_TBL   ; GET ADDRESS OF I INDEX TABLE
269E            MVI             A,-1            ; GET INITIAL DATA
26A0            MVI             C,6             ; SETUP TO INCREMENT H & L
26A2            MVI             B,0             ; ...
26A4            MVI             D,3             ; SET A LOOP COUNT
26A6    10$:    MOV             M,A             ; INIT I CURRENT VALUE
26A7            INX             H               ; ...
26A8            MOV             M,A             ; ...
26A9            DAD             B               ; POINT AT J INDEX TABLE
26AA            DCR             D               ; DONE ALL 3 INDEXES?
26AB            JNZ             10$             ; BRANCH IF NO
26AE            RET                             ; EXIT
26AF
```

```
26AF         .SBTTL  "  GET INDEX VALUE ROUTINE
26AF ;++
26AF ;
26AF ; FUNCTIONAL DESCRIPTION:
26AF ;
26AF ;       THIS ROUTINE GETS THE CURRENT VALUE OF A SPECIFIED INDEX NAME.
26AF ;
26AF ; CALLING SEQUENCE:
26AF ;
26AF ;       CALL    GET_INDEX_VALUE
26AF ;
26AF ; INPUT PARAMETERS:
26AF ;
26AF ;       REGISTER A - CONTAINS THE INDEX NAME VALUE (I=0, J=2, K=4)
26AF ;
26AF ; IMPLICIT INPUTS:
26AF ;
26AF ;       NONE
26AF ;
26AF ; OUTPUT PARAMETERS:
26AF ;
26AF ;       REGISTER B AND C - CONTAINS THE CURRENT VALUE OF THE SPECIFIED INDEX NAME
26AF ;       REGISTER PAIR H & L - CONTAIN ADDRESS OF SPECIFIED INDEX TABLE
26AF ;
26AF ; IMPLICIT OUTPUTS:
26AF ;
26AF ;       NONE
26AF ;
26AF ; COMPLETION CODES:
26AF ;
26AF ;       C BIT SET IF CURRENT VALUE OF INDEX IS MINUS 1 (INDICATES UNUSED).
26AF ;       OTHERWISE, C BIT CLEAR.
26AF ;
26AF ;--
26AF
26AF GET_INDEX_VALUE:
26AF ;
26AF ; SEE IF AN INDEX NAME WAS SPECIFIED
26AF ;
26AF         MVI             B,0                 ; INITIALIZE THE INDEX VALUE
26B1         MOV             C,B                 ; ...
26B2         ORA             A                   ; SET THE CONDITION CODES
26B3         JM              5$                  ; EXIT IF INDEX NAME NOT SPECIFIED
26B6 ;
26B6 ; SETUP TO INDEX THRU THE INDEX_NAM_TBL WITH THE 'A' REGISTER
26B6 ;
26B6         LXI             H,INDEX_NAM_TBL     ; GET ADDRESS OF THE TABLE
26B9         MOV             E,A                 ; SETUP TO DO THE ADD
26BA         MVI             D,0                 ; ...
26BC         DAD             D                   ; INCREMENT H & L TO CORRECT PLACE
26BD ;
26BD ; NOW GET THE ADDRESS OF THE SPECIFIED INDEX
26BD ;
26BD         MOV             E,M                 ; GET LOW PART OF ADDRESS
26BE         INX             H                   ; POINT AT HI PART OF ADDRESS
```

```
26BF  8566          MOV           D,M                 ; GET HIGH PART OF ADDRESS
26C0  8567          XCHG                              ; PUT ADDRESS IN H & L
26C1  8568 ;
26C1  8569 ; REGISTER PAIR D & E NOW CONTAINS ADDRESS OF THE INDEX TABLE. GET THE
26C1  8570 ; CURRENT VALUE FROM THE TABLE AND RETURN
26C1  8571 ;
26C1  8572          MOV           C,M                 ; GET THE CURRENT VALUE (LOW BYTE)
26C2  8573          INX           H                   ; POINT AT HIGH BYTE
26C3  8574          MOV           B,M                 ; GET HIGH BYTE
26C4  8575          DCX           H                   ; POINT H & L AT CURRENT VALUE
26C5  8576          MOV           A,B                 ; TEST FOR MINUS ONE
26C6  8577          CPI           -1                  ; ...
26C8  8578          JNZ           10$                 ; HIGH BYTE NOT MINUS ONE
26CB  8579          MOV           A,C                 ; ...
26CC  8580          CPI           -1                  ; ...
26CE  8581          JNZ           10$                 ; LOW BYTE NOT MINUS ONE
26D1  8582 5$:      STC                               ; SET THE C BIT
26D2  8583          RET                               ; RETURN WITH C BIT SET
26D3  8584 10$:     STC                               ; RETURN WITH C BIT CLEAR
26D4  8585          CMC                               ; ...
26D5  8586          RET                               ; RETURN TO CALLER
26D6  8587
```

```
26D6            .SBTTL  " MULTIPLY A TIMES B&C ROUTINE
26D6    ;++
26D6    ;
26D6    ; FUNCTIONAL DESCRIPTION:
26D6    ;
26D6    ;       THIS ROUTINE MULTIPLIES THE CONTENTS OF THE A REGISTER BY THE
26D6    ;       CONTENTS OF THE B AND C REGISTERS WHERE B CONTAINS THE HIGH ORDER
26D6    ;       BITS. THE 16 LEAST SIGNIFICANT BITS OF THE PRODUCT ARE RETURNED IN
26D6    ;       THE D & E REGISTER PAIR. THE LOW ORDER BITS ARE IN THE E REGISTER.
26D6    ;
26D6    ; CALLING SEQUENCE:
26D6    ;
26D6    ;       CALL    MULTIPLY_A_BC
26D6    ;
26D6    ; INPUT PARAMETERS:
26D6    ;
26D6    ;       REGISTER A - CONTAINS THE MULTIPLIER
26D6    ;       REGISTER PAIR B & C - CONTAINS THE MULTIPLICAND
26D6    ;
26D6    ; IMPLICIT INPUTS:
26D6    ;
26D6    ;       NONE
26D6    ;
26D6    ; OUTPUT PARAMETERS:
26D6    ;
26D6    ;       REGISTER PAIR D & E - CONTAIN THE 16 BIT PRODUCT
26D6    ;
26D6    ; IMPLICIT OUTPUTS:
26D6    ;
26D6    ;       NONE
26D6    ;
26D6    ;--
26D6
26D6    MULTIPLY_A_BC:
26D6            MVI             H,0                     ; INITIALIZE THE PRODUCT
26D8            MOV             L,0                     ; ...
26D9    ;
26D9    ; START THE MULTIPLY
26D9    ;
26D9    1$:     DCR             A                       ; DECREMENT THE MULTIPLIER
26DA            JM              5$                      ; BRANCH IF DONE WITH MULTIPLY
26DD            DAD             B                       ; ADD MULTIPLICAND TO P.P.
26DE            JMP             1$                      ; CONTINUE
26E1    5$:     XCHG                                    ; MOVE RESULT TO D & E
26E2            RET                                     ; EXIT
26E3
```

```
26E3            .SBTTL  "  SAVE BAD DATA ROUTINE
26E3    ;++
26E3    ;
26E3    ; FUNCTIONAL DESCRIPTION:
26E3    ;
26E3    ;       THIS ROUTINE SAVES THE CONTENTS OF THE LOCATION(S) POINTED TO BY
26E3    ;       THE H & L REGISTERS IN IMPLICIT STORAGE.
26E3    ;
26E3    ; CALLING SEQUENCE:
26E3    ;
26E3    ;       CALL    SAVE_BAD_DATA
26E3    ;
26E3    ; INPUT PARAMETERS:
26E3    ;
26E3    ;       REGISTER PAIR H & L - CONTAIN THE ADDRESS OF THE DATA TO SAVE
26E3    ;       REGISTER A          - CONTAINS THE NUMBER OF BYTES TO SAVE
26E3    ;
26E3    ; IMPLICIT INPUTS:
26E3    ;
26E3    ;       NONE
26E3    ;
26E3    ; OUTPUT PARAMETERS:
26E3    ;
26E3    ;       NONE
26E3    ;
26E3    ; IMPLICIT OUTPUTS:
26E3    ;
26E3    ;       BAD_DATA - GETS LOADED WITH THE SPECIFIED DATA
26E3    ;       DATA_TYPE- IS LOADED WITH THE BYTE COUNT
26E3    ;
26E3    ;--
26E3
26E3    SAVE_BAD_DATA:
26E3            STA             DATA_TYPE               ; SAVE THE DATA TYPE
26E6            MOV             B,A                     ; SAVE THE BYTE COUNT
26E7            LXI             D,BAD_DATA              ; GET ADDRESS OF BAD DATA LOCATION
26EA    ;
26EA    ; FIRST CLEAR THE ENTIRE LONG WORD
26EA    ;
26EA            PUSH            H                       ; SAVE H & L
26EB            LXI             H,0                     ; GET DATA TO CLEAR WITH
26EE            SHLD            BAD_DATA                ; CLEAR LOW 16 BITS
26F1            SHLD            BAD_DATA+2              ; AND UPPER 16 BITS
26F4            POP             H                       ; RESTORE H & L
26F5    1$:     MOV             A,M                     ; GET A BYTE OF THE DATA TO SAVE
26F6            STAX            D                       ; SAVE THE DATA
26F7            INX             D                       ; POINT TO THE NEXT BYTE
26F8            INX             H                       ; ...
26F9            DCR             B                       ; DECREMENT THE BYTE COUNT
26FA            JNZ             1$                      ; CONTINUE FOR THE SPECIFIED NO. OF BYTES
26FD            RET                                     ; EXIT
26FE
```

```
26FE            .SBTTL  "  SAVE GOOD DATA ROUTINE
26FE    ;++
26FE    ;
26FE    ; FUNCTIONAL DESCRIPTION:
26FE    ;
26FE    ;       THIS ROUTINE SAVES THE CONTENTS OF THE LOCATION(S) POINTED TO BY
26FE    ;       THE H & L REGISTERS IN IMPLICIT STORAGE.
26FE    ;
26FE    ; CALLING SEQUENCE:
26FE    ;
26FE    ;       CALL    SAVE_GOOD_DATA
26FE    ;
26FE    ; INPUT PARAMETERS:
26FE    ;
26FE    ;       REGISTER PAIR H & L - CONTAIN THE ADDRESS OF THE DATA TO SAVE
26FE    ;       REGISTER A          - CONTAINS THE NUMBER OF BYTES TO SAVE
26FE    ;
26FE    ; IMPLICIT INPUTS:
26FE    ;
26FE    ;       NONE
26FE    ;
26FE    ; OUTPUT PARAMETERS:
26FE    ;
26FE    ;       NONE
26FE    ;
26FE    ; IMPLICIT OUTPUTS:
26FE    ;
26FE    ;       GOOD_DATA - GETS LOADED WITH THE SPECIFIED DATA
26FE    ;       DATA_TYPE - IS LOADED WITH THE BYTE COUNT
26FE    ;
26FE    ;--
26FE
26FE    SAVE_GOOD_DATA:
26FE            STA             DATA_TYPE       ; SAVE DATA TYPE
2701            MOV             B,A             ; SAVE THE BYTE COUNT
2702            LXI             D,GOOD_DATA     ; GET ADDRESS OF GOOD DATA LOCATION
2705    ;
2705    ; FIRST CLEAR THE ENTIRE LONG WORD
2705    ;
2705            PUSH            H               ; SAVE H & L
2706            LXI             H,0             ; GET DATA TO CLEAR WITH
2709            SHLD            GOOD_DATA       ; CLEAR LOW 16 BITS
270C            SHLD            GOOD_DATA+2     ; AND UPPER 16 BITS
270F            POP             H               ; RESTORE H & L
2710    1$:     MOV             A,M             ; GET A BYTE OF THE DATA TO SAVE
2711            STAX            D               ; SAVE THE DATA
2712            INX             D               ; POINT TO THE NEXT BYTE
2713            INX             H               ; ...
2714            DCR             B               ; DECREMENT THE BYTE COUNT
2715            JNZ             1$              ; CONTINUE FOR THE SPECIFIED NO. OF BYTES
2718            RET                             ; EXIT
2719
```

```
2719                    .SBTTL  "  COMPARE GOOD AND BAD ROUTINE
2719            ;++
2719            ;
2719            ; FUNCTIONAL DESCRIPTION:
2719            ;
2719            ;       THIS ROUTINE COMPARES THE CONTENTS OF GOOD_DATA AND BAD_DATA.
2719            ;
2719            ; CALLING SEQUENCE:
2719            ;
2719            ;       CALL    CMP_GOOD_BAD
2719            ;
2719            ; INPUT PARAMETERS:
2719            ;
2719            ;       REGISTER A - CONTAINS THE NUMBER OF BYTES TO COMPARE
2719            ;       REGISTER B - CONTAINS THE MODE FLAG
2719            ;                         0 = COMPARE FOR EQUALITY
2719            ;                        FF = COMPARE FOR NOT EQUAL
2719            ;
2719            ; INPUT PARAMETERS:
2719            ;
2719            ;       GOOD_DATA - CONTAINS THE EXPECTED DATA
2719            ;       BAD_DATA  - CONTAINS THE RECIEVED DATA
2719            ;
2719            ; OUTPUT PARAMETERS:
2719            ;
2719            ;       NONE
2719            ;
2719            ; IMPLICIT OUTPUTS:
2719            ;
2719            ;       NONE
2719            ;
2719            ; COMPLETION CODES:
2719            ;
2719            ;       C BIT SET   - ALL (OR ONE) BYTE FAILED THE COMPARE
2719            ;       C BIT CLEAR - ALL (OR ONE) BYTE PASSED THE COMPARE
2719            ;
2719            ;--
2719
2719            CMP_GOOD_BAD:
2719                    MOV             C,A                     ; SAVE THE BYTE COUNT
271A                    LXI             D,GOOD_DATA             ; GET ADDRESS OF GOOD DATA
271D                    LXI             H,BAD_DATA              ; AND ADDRESS OF BAD DATA
2720                    MOV             A,B                     ; WHAT IS THE MODE?
2721                    ORA             A                       ; ...
2722                    JNZ             10$                     ; BRANCH IF "NOT EQUAL"
2725            1$:     LDAX            D                       ; GET A BYTE OF GOOD DATA
2726                    CMP             M                       ; COMPARE WITH BAD DATA
2727                    JNZ             5$                      ; EXIT WITH C BIT SET
272A                    INX             D                       ; POINT AT NEXT BYTE
272B                    INX             H                       ; ...
272C                    DCR             C                       ; DECREMENT THE BYTE COUNT
272D                    JNZ             1$                      ; CONTINUE FOR BYTE CNT NO. OF TIMES
2730            15$:    STC                                     ; EXIT WITH C BIT CLEAR
2731                    CMC                                     ; ...
2732                    RET                                     ; ...
```

```
 2733  8797 5$:      STC                                      ; EXIT WITH C BIT SET
 2734  8798          RET                                      ; ...
 2735  8799 ;
 2735  8800 ; MODE IS "NOT EQUAL"
 2735  8801 ;
 2735  8802 10$:     LDAX            D                        ; GET A BYTE OF GOOD DATA
 2736  8803          CMP             M                        ; COMPARE WITH BAD DATA
 2737  8804          JNZ             15$                      ; EXIT WITH C BIT CLEAR
 273A  8805          INX             D                        ; POINT AT NEXT BYTE
 273B  8806          INX             H                        ; ...
 273C  8807          DCR             C                        ; DECREMENT THE BYTE COUNT
 273D  8808          JNZ             10$                      ; CONTINUE FOR BYTE CNT NO. OF TIMES
 2740  8809          JMP             5$                       ; EXIT WITH C SET (ALL BYTES EQUAL)
 2743  8810
```

```
2743               .SBTTL  "  RING BELL ROUTINE
2743      ;++
2743      ;
2743      ; FUNCTIONAL DESCRIPTION:
2743      ;
2743      ;        THIS ROUTINE TRANSMITS AN ASCII 7 (BELL) TO THE TERMINAL THE FIRST
2743      ;        TIME IT IS CALLED AND ON EVERY 5TH CALL THERE AFTER.
2743      ;
2743      ; CALLING SEQUENCE:
2743      ;
2743      ;        CALL    RING_BELL
2743      ;
2743      ; IMPLICIT INPUTS:
2743      ;
2743      ;        BELL_COUNT - CONTAINS THE NUMBER OF TIMES THE ROUTINE HAS BEEN CALLED.
2743      ;
2743      ; IMPLICIT OUTPUTS:
2743      ;
2743      ;        BELL_COUNT - IS INCREMENTED BY ONE OR INITIALIZED TO ONE.
2743      ;
2743      ;--
2743
2743      RING_BELL:
2743               LDA             BELL_COUNT      ; GET THE CALL COUNT
2746               ORA             A               ; CHECK FOR ZERO
2747               JZ              1$              ; BRANCH IF FIRST CALL
274A               CPI             5               ; IS IT EQUAL TO 5 YET?
274C               JNZ             10$             ; EXIT IF NO
274F               XRA             A               ; CLEAR A
2750               STA             BELL_COUNT      ; INITIALIZE THE BELL COUNT
2753      ;
2753      ; RING THE BELL
2753      ;
2753      1$:      TYPE            BELL            ; RING THE BELL
275C      ;
275C      ; INCREMENT THE BELL COUNT AND EXIT
275C      ;
275C      10$:     LDA             BELL_COUNT
275F               INR             A               ; INCREMENT
2760               STA             BELL_COUNT      ; ...
2763               RET                             ;
2764
```

```
2764             .SBTTL  "  TYPE ERROR ROUTINE
2764     ;++
2764     ;
2764     ; FUNCTIONAL DESCRIPTION:
2764     ;
2764     ;       THIS ROUTINE IS USED TO TYPE AN ERROR MESSAGE. THE ERROR MESSAGE
2764     ;       HAS THE FOLLOWING FORMAT:
2764     ;
2764     ;       ?ERROR: <ERROR_PC> TEST: <TEST_NUMBER> SUBTEST: <SUB_TEST_NUMB>
2764     ;
2764     ;       DATA:   XXXXXXXX [NOT]
2764     ;               YYYYYYYY
2764     ;                   .
2764     ;                   .
2764     ;
2764     ;       FAILING GATE ARRAYS: AAA,BBB,CCC,...
2764     ;       FAILING MODULES: M8NXX
2764     ;
2764     ;       THE DATA THAT GETS TYPED OUT AFTER 'DATA:' COMES FROM LOCATIONS
2764     ;       'GOOD_DATA', 'BAD_DATA', CURRENT LOOP VALUES, AND INTERNAL CPU
2764     ;       SCRATCH PADS. THE NUMBER OF SCRATCH PADS TO TYPE IS CONTAINED IN
2764     ;       THE FOLLOWING TABLE. IF THE MODE TYPE OF THE LAST COMPARISON WAS
2764     ;       "NOT EQUAL", THE WORD 'NOT' IS TYPED AFTER THE GOOD DATA.
2764     ;
2764     ;       THE NAMES OF THE GATE ARRAYS AND MODULES TO TYPE MUST BE IN A TABLE
2764     ;       WITH THE FOLLOWING FORMAT:
2764     ;
2764     ;               .BYTE   NUMBER OF SCRATCH PADS
2764     ;               .BYTE   NUMBER OF GATE ARRAYS
2764     ;               .BYTE   GATE ARRAY CODE
2764     ;                 .
2764     ;                 .
2764     ;               .BYTE   NUMBER OF MODULES
2764     ;               .BYTE   MODULE CODE
2764     ;                 .
2764     ;
2764     ; CALLING SEQUENCE:
2764     ;
2764     ;       CALL    TYPE_ERROR
2764     ;
2764     ; INPUT PARAMETERS:
2764     ;
2764     ;       REGISTER PAIR H & L - CONTAIN THE ADDRESS OF THE GATE ARRAY/MODULE LIST
2764     ;
2764     ; IMPLICIT INPUTS:
2764     ;
2764     ;       CURRENT_PC      - CONTAINS THE ADDRESS OF THE 'IFERROR' INSTRUCTION
2764     ;       TEST_NUMBER     - CONTAINS THE CURRENT TEST NUMBER
2764     ;       SUB_TEST_NUMB   - CONTAINS THE CURRENT SUBTEST NUMBER
2764     ;       GOOD_DATA       - CONTAINS THE EXPECTED DATA
2764     ;       BAD_DATA        - CONTAINS THE RECEIVED DATA
2764     ;       CURRENT LOOP VALUES
2764     ;
2764     ; IMPLICIT OUTPUTS:
2764     ;
```

```
2764 ;       NONE
2764 ;
2764 ;--
2764
2764 TYPE_ERROR:
2764         PUSH            H               ; SAVE POINTER TO LIST
2765         TYPE            CRLF            ; START A NEW LINE
276E         TYPE            ERROR_MSG       ; TYPE '?ERROR: '
2777 ;
2777 ; GET THE ERROR TEST PC, SUBTRACT THE BASE ADDRESS, AND TYPE IT
2777 ;
2777         LHLD            C_INIT_TEST_PC  ; GET TWO'S COMPLIMENT INIT TEST PC
277A         XCHG
277B         LHLD            CURRENT_PC      ; GET THE ERROR PC
277E         DAD             D               ; SUBTRACT THE BASE ADDRESS OF TEST
277F         XCHG                            ; ADD IN THE TITLE LENGTH
2780         LHLD            TITLE_LENGTH    ; ...
2783         DAD             D               ; ...
2784         INX             H               ; ACCOUNT FOR LENGTH BYTE
2785         SHLD            TYPE_ERR_BUFFER ; PUT ERROR PC IN BUFFER
2788         TYPEW           TYPE_ERR_BUFFER ; TYPE THE ERROR PC
2793 ;
2793 ; NOW TYPE THE TEST NUMBER
2793 ;
2793         TYPE            TEST_MSG        ; TYPE ' TEST: '
279C         TYPEB           TEST_NUMBER     ; TYPE THE TEST NUMBER
27A7 ;
27A7 ; NOW TYPE THE SUB TEST NUMBER
27A7 ;
27A7         TYPE            SUB_TEST_MSG    ; TYPE ' SUBTEST: '
27B0         LDA             SUB_TEST_NUMB   ; IF SUB TEST NUMBER IS ZERO, MAKE
27B3         ORA             A               ; IT EQUAL TO ONE
27B4         JNZ             1$              ; NOT ZERO
27B7         MVI             A,1             ; ...
27B9         STA             SUB_TEST_NUMB   ; ...
27BC 1$:     TYPEB           SUB_TEST_NUMB   ; TYPE THE SUB TEST NUMBER
27C7         TYPE            CRLF            ; TERMINATE THE LINE
27D0         TYPE            CRLF            ; SKIP A LINE
27D9 ;
27D9 ; NOW TYPE THE DATA
27D9 ;
27D9         TYPE            DATA_MSG        ; TYPE 'DATA:    '
27E2         TYPEL           GOOD_DATA       ; TYPE THE EXPECTED DATA
27ED         LDA             MODE_TYPE       ; WAS THIS AN 'EQUALS' CHECK?
27F0         ORA             A               ; ...
27F1         JZ              2$              ; BRANCH IF YES
27F4         TYPE            NOT_MSG         ; TYPE 'NOT' AFTER GOOD DATA
27FD 2$:     TYPE            CRLF
2806         LXI             H,BAD_DATA
2809         CALL            DISPLAY_DATA    ; TYPE THE RECEIVED DATA
280C ;
280C ; NOW TYPE THE LOOP VALUES OF I, J, AND K IF THEY ARE BEING USED.
280C ;
280C         LXI             H,0             ; CLEAR THE TEMP BUFFER
280F         SHLD            TYPE_ERR_BUFFER ; ...
```

```
2812             SHLD        TYPE_ERR_BUFFER+2 ; ...
2815             XRA         A                 ; GET THE I INDEX CODE IN A REGISTER
2816             CALL        GET_INDEX_VALUE   ; GET THE CURRENT VALUE OF I INDEX
2819             JC          5$                ; BRANCH IF I NOT BEING USED
281C             PUSH        B                 ; PUT VALUE IN H & L
281D             POP         H                 ; ...
281E             INX         H                 ; MAKE LOOP COUNT START AT ONE
281F             SHLD        TYPE_ERR_BUFFER   ; SAVE I INDEX VALUE
2822             LXI         H,TYPE_ERR_BUFFER
2825             CALL        DISPLAY_DATA      ; TYPE THE I INDEX VALUE
2828
2828   5$:       MVI         A,2               ; GET THE J INDEX CODE
282A             CALL        GET_INDEX_VALUE   ; GET THE CURRENT VALUE
282D             JC          10$               ; BRANCH IF J INDEX NOT USED
2830             PUSH        B                 ; MOVE VALUE TO H & L
2831             POP         H                 ; ...
2832             INX         H                 ; MAKE VALUE START AT ONE
2833             SHLD        TYPE_ERR_BUFFER   ; SAVE THE CURRENT VALUE
2836             LXI         H,TYPE_ERR_BUFFER
2839             CALL        DISPLAY_DATA      ; TYPE THE J INDEX VALUE
283C
283C   10$:      MVI         A,4               ; GET THE K INDEX CODE
283E             CALL        GET_INDEX_VALUE   ; GET THE CURRENT VALUE
2841             JC          15$               ; BRANCH IF NOT USED
2844             PUSH        B                 ; PUT VALUE IN H & L
2845             POP         H                 ; ...
2846             INX         H                 ; MAKE VALUE START AT ONE
2847             SHLD        TYPE_ERR_BUFFER   ; SAVE CURRENT J INDEX VALUE
284A             LXI         H,TYPE_ERR_BUFFER
284D             CALL        DISPLAY_DATA      ; TYPE THE K INDEX VALUE
2850         ;
2850         ; NOW TYPE THE SCRATCH PAD DATA IF THERE IS ANY
2850         ;
2850   15$:      POP         H                 ; GET POINTER TO LIST
2851             MOV         A,M               ; GET THE NUMBER OF SCRATCH PADS TO TYPE
2852             ORA         A                 ; SET THE CONDITION CODES
2853             JZ          30$               ; BRANCH IF NO SCRATCH PADS TO TYPE
2856             PUSH        $PSW              ; SAE THE COUNT
2857             XRA         A                 ; INIT THE REGISTER NUMBER
2858             STA         TYPE_ERR_TEMP     ; ...
285B   20$:      PUSH        H                 ; SAVE POINTER
285C             LDA         TYPE_ERR_TEMP     ; GET REGISTER NUMBER
285F             CALL        READ_RTEMP        ; GET CONTENTS OF REGISTER
2862         ;
2862         ; NOW TYPE THE CONTENTS OF THE WH REGISTER
2862         ;
2862             LXI         H,WH_REG_BYTE_0 ; GET ADDRESS OF REGISTER
2865             CALL        DISPLAY_DATA    ; TYPE THE DATA
2868         ;
2868         ; CHECK IF MORE DATA TO TYPE
2868         ;
2868             POP         H                 ; RESTORE H & L
2869             LDA         TYPE_ERR_TEMP     ; INCREMENT THE REGISTER NUMBER
286C             INR         A                 ; ...
286D             STA         TYPE_ERR_TEMP     ; ...
```

```
2870              POP     $PSW               ; GET THE WORD COUNT
2871              DCR     A                  ; DONE YET?
2872              PUSH    $PSW               ; SAVE WORD COUNT
2873              JNZ     20$                ; BRANCH IF NO
2876              POP     $PSW               ; CLEANUP THE STACK
2877      ;
2877      ; SEE IF THIS WAS A VBUS COMPARE THAT FAILED
2877      ;
2877      30$:    LDA     SOFT_CTRL_REGA     ; GET THE VBUS COMPARE FLAG
287A              ANI     VBUS_COMPARE       ; IS IT SET?
287C              JZ      40$                ; BRANCH IF NO
287F      ;
287F      ; TYPE THE REMAINING VBUS BITS THAT WERE NOT CHECKED
287F      ;
287F              PUSH    H                  ; SAVE THE H & L REGISTERS
2880              LHLD    VBUS_TBL_ADDR      ; GET THE ADDRESS OF THE VBUS TABLE
2883              LDA     VBUS_TBL_COUNT     ; AND THE REMAINING BITS
2886              MOV     B,A                ; ...
2887              DCR     B                  ; MORE BITS TO TYPE?
2888              JM      39$                ; BRANCH IF NO
288B              PUSH    H
288C              PUSH    B
288D              TYPE    VBUS_MSG_2         ; TYPE THE VBUS MESSAGE
2896              TYPE    CRLF
289F              POP     B
28A0              POP     H
28A1      33$:    CALL    CHECK_VBUS         ; GET VBUS DATA INTO 'BAD_DATA'
28A4              PUSH    H                  ; SAVE H & L
28A5              PUSH    B                  ; AND B
28A6              LXI     H,BAD_DATA
28A9              CALL    DISPLAY_DATA       ; TYPE THE VBUS BIT AND VALUE
28AC              POP     B
28AD              POP     H
28AE              INX     H                  ; POINT AT NEXT VBUS ENTRY
28AF              DCR     B                  ; MORE TO TYPE?
28B0              JP      33$                ; BRANCH IF YES
28B3      39$:    POP     H                  ; RESTORE H & L
28B4
28B4      ;
28B4      ; NOW TYPE THE FAILING GATE ARRAY LIST
28B4      ;
28B4      40$:    INX     H                  ; POINT AT THE GATE ARRAY COUNT
28B5              MOV     A,M                ; GET NUMBER OF GATE ARRAYS
28B6              ORA     A                  ; SET THE CODITION CODES
28B7              CNZ     TYPE_GA_LIST       ; DON'T CALL IF NO GATE ARRAYS IN CALLOUT
28BA
28BA      ;
28BA      ; NOW SEE IF THERE IS A MODULE LIST TO TYPE
28BA      ;
28BA      50$:    INX     H                  ; POINT AT THE MODULE COUNT
28BB              MOV     A,M                ; GET THE COUNT
28BC              ORA     A                  ; SET THE CONDITION CODES
28BD              JZ      60$                ; BRANCH IF NO MODULES TO TYPE
28C0              PUSH    $PSW               ; SAVE THE COUNT
28C1              INX     H                  ; POINT AT THE MODULE CODE
```

```
28C2            PUSH    H               ; SAVE THE POINTER
28C3            TYPE    MODULE_MSG      ; TYPE 'FAILING MODULES: '
28CC            POP     H               ; GET THE POINTER
28CD    51$:    MOV     A,M             ; GET THE MODULE CODE
28CE            PUSH    H               ; SAVE THE POINTER
28CF            RLC                     ; MAKE CODE A WORD INDEX
28D0            LXI     H,MODULE_NAM_TBL; GET ADDRESS OF MODULE NAME TABLE
28D3            MOV     C,A             ; SETUP TO INDEX THRU THE TABLE
28D4            MVI     B,0             ; ...
28D6            DAD     B               ; INDEX THE LIST ADDRESS
28D7            MOV     E,M             ; GET ADDRESS OF ASCII STRING
28D8            INX     H               ; ...
28D9            MOV     D,M             ; ...
28DA            XCHG                    ; ...
28DB            TYPE                    ; TYPE THE MODULE NAME
28E1            TYPE    CRLF            ; TERMINATE THE LINE
28EA            TYPE    FAIL_MOD_INDEX  ; SET CURSOR FOR NEXT MODULE PRINTOUT
28F3            POP     H               ; GET LIST POINTER
28F4            POP     $PSW            ; GET LOOP COUNT
28F5            DCR     A               ; DECREMENT
28F6            PUSH    $PSW            ; SAVE
28F7            INX     H               ; POINT AT NEXT MODULE CODE
28F8            JNZ     51$             ; TYPE NEXT MODULE NAME
28FB            POP     $PSW            ; THROW AWAY SAVED LOOP COUNT
28FC            TYPE    CRLF            ; TERMINATE THE LAST LINE
2905    ;
2905    ; DONE
2905    ;
2905    60$:    RET                     ; EXIT
2906
```

```
2906            .SBTTL  "  SINGLE PSEUDO INSTRUCTION ROUTINE
2906      ;++
2906      ;
2906      ; FUNCTIONAL DESCRIPTION:
2906      ;
2906      ;       THIS ROUTINE IS USED TO STEP BY SINGLE PSEUDO INSTRUCTIONS. IT IS
2906      ;       CALLED BY THE OP CODE DISPATCH ROUTINE. THE ROUTINE TYPES THE
2906      ;       CURRENT TEST PC AND WAITS FOR OPERATOR INPUT. IF A 'SPACE' IS TYPED
2906      ;       THE ROUTINE EXITS WITH THE C BIT CLEAR. IF ANY OTHER CHARACTER
2906      ;       IS TYPED, THE SINGLE INSTRUCTION FLAG IS CLEARED AND RETURN IS MADE
2906      ;       WITH THE C BIT SET.
2906      ;
2906      ; CALLING SEQUENCE:
2906      ;
2906      ;       CALL    SINGLE_INSTR
2906      ;
2906      ; IMPLICIT INPUTS:
2906      ;
2906      ;       TEST_PC - CONTAINS THE CURRENT TEST PC
2906      ;
2906      ; OUTPUT PARAMETERS:
2906      ;
2906      ;       NONE
2906      ;
2906      ; COMPLETION CODES:
2906      ;
2906      ;       C BIT SET   - A SPACE WAS NOT TYPED
2906      ;       C BIT CLEAR - A SPACE WAS TYPED
2906      ;
2906      ;--
2906
2906      SINGLE_INSTR:
2906            $TERM_INIT                        ; ABORT ANY INPUT REQUESTS
290B            TYPE            CRLF              ; START A NEW LINE
2914      1$:   TYPE            TPC_MSG           ; TYPE '   TPC= '
291D            LHLD            TEST_PC           ; GET THE TEST PC
2920            XCHG
2921            LHLD            C_INIT_TEST_PC    ; GET TWO'S COMPLIMENT INIT TEST PC
2924            DAD             D                 ; SUBTRACT FROM TEST PC
2925            INX             H                 ; ACCOUNT FOR LENGTH BYTE
2926            XCHG
2927            LHLD            TITLE_LENGTH
292A            DAD             D
292B            SHLD            TYPE_ERR_BUFFER   ; SAVE THE TEST PC
292E            TYPEW           TYPE_ERR_BUFFER   ; TYPE THE TEST PC
2939            $TERM_READ      TERM_INP_BUFF,ONE  ; GET ONE CHARACTER FROM TERMINAL
2944            LDA             TERM_INP_BUFF+1   ; GET THE CHARACTER THAT WAS TYPED
2947            CPI             <^X20>            ; WAS IT A SPACE?
2949            JZ              5$                ; BRANCH IF YES
294C            LDA             SOFT_CTRL_REGA    ; GET SINGLE INSTR FLAG
294F            ANI             <^CINSTR_FLAG>    ; CLEAR THE FLAG
2951            STA             SOFT_CTRL_REGA    ; ...
2954            STC
2955            JMP             10$               ; RETURN WITH C BIT SET
2958      ;
```

```
    2958  9161 ; EXIT
    2958  9162 ;
    2958  9163 5$:
    2958  9164          $TERM_READ       TERM_INP_BUFF,- ; ISSUE AN INTERRUPT DRIVEN READ
    2958  9165                           ONE,TERM_INTERRUPT; ...
    2963  9166          STC
    2964  9167          CMC                                 ; RETURN WITH C BIT CLEAR
    2965  9168 10$:     RET                                 ; EXIT
    2966  9169
```

```
2966              .SBTTL  "  INSERT FIELD ROUTINE
2966      ;++
2966      ;
2966      ; FUNCTIONAL DESCRIPTION:
2966      ;
2966      ;       THIS ROUTINE IS USED TO INSERT A DATA ELEMENT INTO A FIELD OF A
2966      ;       MICRO WORD. IT USES THE TABLE "DCS_BIT_MAP" TO SCRAMBLE THE BITS
2966      ;       IN THE MICRO WORD. AFTER THE FIELD IS INSERTED, THE 'CALC_PARITY'
2966      ;       ROUTINE IS CALLED TO GENERATE THE NEW PARITY BITS.
2966      ;
2966      ; CALLINE SEQUENCE:
2966      ;
2966      ;       CALL    INSERT_FIELD
2966      ;
2966      ; INPUT PARAMETERS:
2966      ;
2966      ;       REGISTER PAIR H & L - CONTAINS ADDRESS OF THE DATA TO INSERT
2966      ;       REGISTER PAIR D & E - CONTAINS ADDRESS OF BIT 0 OF THE ELEMENT
2966      ;                             TO INSER THE FIELD IN.
2966      ;       REGISTER B          - CONTAINS THE STARTING BIT NUMBER OF THE FIELD
2966      ;       REGISTER C          - CONTAINS THE ENDING BIT NUMBER OF THE FIELD
2966      ;       REGISTER A          - CONTAINS THE INHIBIT PARITY FLAG. NON ZERO
2966      ;                             MEANS DON'T SET THE PARITY.
2966      ;
2966      ; IMPLICIT INPUTS:
2966      ;
2966      ;       (SP)     - INHIBIT SCRAMBLE FLAG
2966      ;
2966      ; OUTPUT PARAMETERS:
2966      ;
2966      ;       NONE
2966      ;
2966      ; IMPLICIT OUTPUTS:
2966      ;
2966      ;       NONE
2966      ;
2966      ;--
2966
2966      INSERT_FIELD:
2966              SHLD            INS_FLD_BUFF    ; SAVE H & L
2969              STA             CURRENT_BIT     ; SAVE A REGISTER
296C              POP             H               ; GET RETURN PC
296D              POP             $PSW            ; GET SCRAMBLE FLAG
296E              STA             SCRAMBLE_FLAG   ; SAVE
2971              PUSH            H               ; RESTORE THE RETURN PC
2972              LHLD            INS_FLD_BUFF    ; RESTORE H & L
2975              LDA             CURRENT_BIT     ; RESTORE THE A REGISTER
2978              PUSH            $PSW            ; SAVE THE A REGISTER
2979              PUSH            D               ; SAVE THE DESTINATION ADDRESS
297A              PUSH            H               ; SAVE THE SOURCE DATA ADDRESS
297B              PUSH            B               ; SAVE STARTING AND ENDING BIT NUMBERS
297C              XRA             A               ; INIT THE CURRENT BIT NUMBER
297D              STA             CURRENT_BIT     ; ...
2980              LXI             H,INS_FLD_BUFF  ; GET ADDRESS OF TEMP BUFFER
2983              MVI             B,10            ; SET A LOOP COUNT FOR 10 BYTES
```

```
2985  1$:     LDAX      D                 ; GET A BYTE OF THE MICRO WORD
2986          MOV       M,A               ; PUT IN TEMP BUFFER
2987          INX       H                 ; ...
2988          INX       D                 ; ...
2989          DCR       B                 ; DECREMENT BYTE COUNT
298A          JNZ       1$                ; CONTINUE FOR 10 BYTES
298D  ;
298D  ; CALCULATE THE NUMBER OF BITS IN THE FIELD
298D  ;
298D          POP       B                 ; GET THE STARTING AND ENDING BIT #'S
298E          MOV       A,C               ; SUBTRACT STARTING FROM THE
298F          SUB       B                 ; ENDING BIT NUMBER
2990          INR       A                 ; INCREMENT BY ONE
2991          STA       BIT_COUNT         ;SAVE
2994  ;
2994  ; B = UNSCRAMBLED BIT NUMBER
2994  ; GET THE SCRAMBLED BIT NUMBER
2994  ;
2994  2$:     LDA       SCRAMBLE_FLAG     ; GET THE SCRAMBLE FLAG
2997          ORA       A                 ; IS IT SET?
2998          JP        4$                ; BRANCH IF NO
299B          MOV       A,B               ; PUT UNSCRAMBLED NUMBER IN A
299C          JMP       6$                ; CONTINUE BELOW
299F  4$:     LXI       H,DCS_BIT_MAP     ; GET ADDRESS OF THE BIT MAP
29A2          MOV       E,B               ; INCREMENT BY THE UNSCRAMBLED BIT #
29A3          MVI       D,0               ; ...
29A5          DAD       D                 ; ...
29A6          MOV       A,M               ; GET THE SCRAMBLED BIT NUMBER
29A7  ;
29A7  ; A = SCRAMBLED BIT NUMBER
29A7  ; B = UNSCRAMBLED BIT NUMBER
29A7  ; CALCULATE THE BYTE AND BIT NUMBER OF THE MICRO WORD
29A7  ;
29A7  6$:     LXI       D,0               ; INIT D & E
29AA  3$:     SUI       8                 ; DIVIDE SCRAMBLED BIT NUMBER BY 8
29AC          JM        5$                ; BRANCH IF DIVIDE DONE
29AF          INR       E                 ; INCREMENT THE QUOTIENT
29B0          JMP       3$                ; CONTINUE THE DIVIDE
29B3  5$:     ADI       8                 ; GET THE REMAINDER
29B5  ;
29B5  ; A = BIT NUMBER WITHIN THE BYTE OF THE MICRO WORD
29B5  ; B = UNSCRAMBLED BIT NUMBER
29B5  ; D = 0
29B5  ; E = BYTE NUMBER WITHIN THE MICRO WORD
29B5  ; GET THE BYTE ADDRESS AND BIT MASK FOR THE MICRO WORD
29B5  ;
29B5          LXI       H,INS_FLD_BUFF    ; GET ADDRESS OF BUFFER
29B8          DAD       D                 ; GET ADDRESS OF BYTE TO MODIFY
29B9          CALL      BIT_NMB_TO_MSK    ; GET BIT MASK
29BC  ;
29BC  ; A = MACRO WORD BIT MASK
29BC  ; B = UNSCRAMBLED BIT NUMBER
29BC  ; H & L = MICRO WORD BYTE ADDRESS
29BC  ; NOW GET THE BYTE ADDRESS AND BIT MASK FOR THE SOURCE DATA
29BC  ;
```

```
29BC                 POP        D                ; GET THE SOURCE ADDRESS
29BD                 PUSH       D                ; SAVE IT AGAIN
29BE                 PUSH       H                ; SAVE MICRO WORD ADDRESS
29BF                 XCHG
29C0                 PUSH       $PSW             ; SAVE THE UNSCRAMBLED BIT NUMBER
29C1                 LDA        CURRENT_BIT      ; GET THE CURRENT BIT NUMBER
29C4                 LXI        D,0              ; INIT D & E
29C7      15$:       SUI        8                ; DIVIDE THE CURRENT BIT NUMBER BY 8
29C9                 JM         20$              ; BRANCH IF DIVIDE DONE
29CC                 INR        E                ; INCREMENT THE QUOTIENT
29CD                 JMP        15$              ; CONTINUE THE DIVIDE
29D0      20$:       ADI        8                ; BACKUP THE REMAINDER
29D2                 CALL       BIT_NMB_TO_MSK   ; GET A MASK FOR THE BIT NUMBER
29D5                 MOV        C,A              ; PUT MASK IN C REG
29D6      ;
29D6      ; C = SOURCE DATA BIT MASK
29D6      ; D = 0
29D6      ; E = SOURCE DATA BYTE NUMBER
29D6      ; H & L = SOURCE DATA ADDRESS
29D6      ;
29D6                 DAD        D                ; GET BYTE ADDRESS OF SOURCE DATA
29D7                 POP        $PSW             ; GET MICRO WORD BIT MASK
29D8                 POP        D                ; GET MICRO WORD BYTE ADDRESS
29D9      ;
29D9      ; NO INSERT THE BIT FROM THE SOURCE INTO THE MICRO WORD
29D9      ;
29D9                 PUSH       B                ; SAVE B & C
29DA                 MOV        B,A              ; SAVE MICRO WORD BIT MASK
29DB                 LDAX       D                ; GET THE BYTE OF MICRO WORD DATA
29DC                 ORA        B                ; SET THE SPECIFIED BIT
29DD                 PUSH       B                ; SAVE THE BIT MASK
29DE                 MOV        B,A              ; PUT THE DATA IN B
29DF                 MOV        A,M              ; GET THE BYTE OF SOURCE DATA
29E0                 ANA        C                ; SEE IF BIT SHOULD BE SET
29E1                 MOV        A,B              ; PUT MICRO WORD BYTE BACK INTO A
29E2                 POP        B                ; RESTORE B & C
29E3                 JNZ        25$              ; BRANCH IF BIT SHOULD BE SET
29E6                 XRA        B                ; CLEAR THE BIT
29E7      25$:       POP        B                ; RESTORE THE UNSCRAMBLED BIT NUMBER
29E8                 STAX       D                ; PUT THE BYTE IN THE MICRO WORD
29E9      ;
29E9      ; NOW CHECK IF THERE ARE MORE BITS TO INSERT
29E9      ;
29E9                 LDA        CURRENT_BIT      ; INCREMENT TH CURRENT BIT NUMBER
29EC                 INR        A                ; ...
29ED                 STA        CURRENT_BIT      ; ...
29F0                 INR        B                ; INCREMENT THE UNSCRAMBLED BIT #
29F1                 LDA        BIT_COUNT        ; DECREMENT THE BIT COUNT
29F4                 DCR        A                ; ...
29F5                 STA        BIT_COUNT        ; ...
29F8                 JNZ        2$               ; BRANCH IF MORE BITS TO INSERT
29FB      ;
29FB      ; SETUP THE REGISTERS TO MOVE THE MICRO WORD TO ITS ORIGINAL DESTINATION.
29FB      ;
29FB                 POP        D                ; CLEANUP THE STACK
```

```
29FC                 POP         D                 ; GET THE DESTINATION ADDRESS
29FD                 LXI         H,INS_FLD_BUFF    ; GET ADDRESS OF THE MICRO WORD
2A00                 MVI         C,10              ; SET THE BYTE COUNT
2A02      ;
2A02      ; NOW MOVE THE TEMP BUFFER TO THE DESTINATION
2A02      ;
2A02                 PUSH        D                 ; SAVE ADDRESS OF MICRO WORD
2A03      30$:       MOV         A,M               ; GET A BYTE FROM TEMP BUFFER
2A04                 STAX        D                 ; PUT IN DESTINATION
2A05                 INX         D                 ; ...
2A06                 INX         H                 ; ...
2A07                 DCR         C                 ; DECREMENT THE BYTE COUNT
2A08                 JNZ         30$               ; CONTINUE FOR 10 BYTES
2A0B                 POP         H                 ; GET MICRO WORD ADDRESS
2A0C                 POP         $PSW              ; GET THE INHIBIT PARITY FLAG
2A0D                 ORA         A                 ; SET THE CONDITION CODES
2A0E                 JNZ         INSERT_FIELD_X    ; BRANCH IF INHIBIT
2A11                 LDA         SCRAMBLE_FLAG     ; GET THE SCRAMBLE FLAG
2A14                 ORA         A                 ; IS IT SET?
2A15                 JM          INSERT_FIELD_X    ; BRANCH IF YES
2A18                 CALL        CALC_PARITY       ; CALCULATE THE NEW PARITY BITS
2A1B      INSERT_FIELD_X:
2A1B                 RET                           ; EXIT
2A1C
```

```
2A1C          .SBTTL  "  CALCULATE PARITY ROUTINE
2A1C  ;++
2A1C  ;
2A1C  ; FUNCTIONAL DESCRIPTION:
2A1C  ;
2A1C  ;       THIS ROUTINE CALCULATES AND LOADS THE P0 AND P1 PARITY BITS OF
2A1C  ;       A MICRO WORD. A BIT MASK IS USED TO DETERMINE THE PARITY FOR THE
2A1C  ;       CORRESPONDING PARITY BITS. P0 IS SET TO MAKE EVEN PARITY AND P1
2A1C  ;       IS SET TO MAKE ODD PARITY.
2A1C  ;
2A1C  ; CALLING SEQUENCE:
2A1C  ;
2A1C  ;       CALL    CALC_PARITY
2A1C  ;
2A1C  ; INPUT PARAMETERS:
2A1C  ;
2A1C  ;       REGISTERS H & L - CONTAIN THE ADDRESS OF THE MICRO WORD TO CALCULATE
2A1C  ;                         THE PARITY BITS ON.
2A1C  ;
2A1C  ; IMPLICIT INPUTS:
2A1C  ;
2A1C  ;       P0_PARITY_MSK - CONTAINS THE 10 BYTE BIT MAP OF THE P0 PARITY BIT
2A1C  ;       P1_PARITY_MSK - CONTAINS THE 10 BYTE BIT MAP OF THE P1 PARITY BIT
2A1C  ;
2A1C  ; IMPLICIT OUTPUTS:
2A1C  ;
2A1C  ;       THE MICRO WORD'S PARITY BITS ARE SET TO THE CORRECT VALUE.
2A1C  ;
2A1C  ;--
2A1C
2A1C  CALC_PARITY:
2A1C          PUSH            H               ; SAVE ADDRESS OF DATA
2A1D  ;
2A1D  ; FIRST CALCULATE THE P0 PARITY BIT
2A1D  ;
2A1D          LXI             D,P0_PARITY_MSK ; GET ADDRESS OF P0 BIT MAP
2A20          MVI             B,0             ; INIT A PARITY FLAG
2A22          MVI             C,10            ; SET A LOOP COUNT
2A24  1$:     LDAX            D               ; GET A BYTE OF THE BIT MAP
2A25          ANA             M               ; AND WITH THE MICRO WORD
2A26          JPE             2$              ; BRANCH IF PARITY EVEN
2A29          MOV             A,B             ; GET THE PARITY FLAG
2A2A          CMA                             ; COMPLIMENT IT
2A2B          MOV             B,A             ; SAVE
2A2C  2$:     INX             D               ; INCREMENT BIT MAP ADDRESS
2A2D          INX             H               ; INCREMENT DATA ADDRESS
2A2E          DCR             C               ; DONE 10 BYTES?
2A2F          JNZ             1$              ; BRANCH IF NO
2A32  ;
2A32  ; NOW SET OR CLEAR THE P0 BIT
2A32  ;
2A32          POP             H               ; GET THE MICRO WORD ADDRESS
2A33          PUSH            H               ; SAVE AGAIN
2A34          MVI             E,5             ; POINT AT BYTE 5
2A36          MVI             D,0             ; ...
```

```
2A38          DAD     D                 ; ...
2A39          MOV     A,M               ; GET THE BYTE
2A3A          ANI     <^XBF>            ; CLEAR THE P0 BIT
2A3C          MOV     C,A               ; SAVE
2A3D          MOV     A,B               ; GET THE PARITY FLAG
2A3E          ORA     A                 ; SEE IF ITS ZERO
2A3F          MOV     A,C               ; GET DATA BACK
2A40          JZ      5$                ; BRANCH IF ALREADY ZERO
2A43          ORI     <^X40>            ; SET THE P0 BIT
2A45    5$:   MOV     M,A               ; LOAD THE MICRO WORD
2A46    ;
2A46    ; NOW CALCULATE THE P1 PARITY BIT
2A46    ;
2A46          POP     H                 ; GET ADDRESS OF DATA AGAIN
2A47          PUSH    H                 ; SAVE
2A48          LXI     D,P1_PARITY_MSK   ; GET ADDRESS OF P1 BIT MAP
2A4B          MVI     B,0               ; INIT A PARITY FLAG
2A4D          MVI     C,10              ; SET A LOOP COUNT
2A4F    10$:  LDAX    D                 ; GET A BYTE OF THE BIT MAP
2A50          ANA     M                 ; AND WITH THE MICRO WORD
2A51          JPE     12$               ; BRANCH IF PARITY EVEN
2A54          MOV     A,B               ; GET THE PARITY FLAG
2A55          CMA                       ; COMPLIMENT IT
2A56          MOV     B,A               ; SAVE
2A57    12$:  INX     D                 ; INCREMENT BIT MAP ADDRESS
2A58          INX     H                 ; INCREMENT DATA ADDRESS
2A59          DCR     C                 ; DONE 10 BYTES?
2A5A          JNZ     10$               ; BRANCH IF NO
2A5D    ;
2A5D    ; NOW SET OR CLEAR THE P1 BIT
2A5D    ;
2A5D          POP     H                 ; GET THE MICRO WORD ADDRESS
2A5E          INX     H                 ; POINT AT BYTE 2
2A5F          INX     H                 ; ...
2A60          MOV     A,M               ; GET THE BYTE
2A61          ANI     <^XF7>            ; CLEAR THE P1 BIT
2A63          MOV     C,A               ; SAVE
2A64          MOV     A,B               ; GET THE PARITY FLAG
2A65          ORA     A                 ; SEE IF ITS ZERO
2A66          MOV     A,C               ; GET DATA BACK
2A67          JNZ     15$               ; BRANCH IF NOT ZERO
2A6A          ORI     8                 ; SET THE P1 BIT
2A6C    15$:  MOV     M,A               ; LOAD THE MICRO WORD
2A6D          RET                       ; EXIT
2A6E
```

```
2A6E            .SBTTL  "  CONVERT BIT NUMBER TO BIT MASK ROUTINE
2A6E     ;++
2A6E     ;
2A6E     ; FUNCTIONAL DESCRIPTION:
2A6E     ;
2A6E     ;      THIS ROUTINE CONVERTS A BIT NUMBER (BETWEEN 0 AND 7) INTO A MASK
2A6E     ;      WITH ONE BIT SET IN THE POSITION DEFINED BY THE NUMBER
2A6E     ;
2A6E     ; INPUT PARAMETERS:
2A6E     ;
2A6E     ;      A REGISTER - CONTAINS THE BIT NUMBER
2A6E     ;
2A6E     ; OUTPUT PARAMETERS:
2A6E     ;
2A6E     ;      A REGISTER - CONTAINS THE BIT MASK
2A6E     ;
2A6E     ;--
2A6E
2A6E     BIT_NMB_TO_MSK:
2A6E            PUSH            B                       ; SAVE THE B REGISTER
2A6F            MOV             B,A                     ; PUT BIT NUMBER IN B REG
2A70            MVI             A,1                     ; INIT THE BIT MASK
2A72     1$:    DCR             B                       ; FOUND THE BIT YET?
2A73            JM              5$                      ; BRANCH IF YES
2A76            RLC                                     ; ROTATE THE MASK
2A77            JMP             1$                      ; CONTINUE
2A7A     5$:    POP             B                       ; RESTORE THE B REGISTER
2A7B            RET                                     ; EXIT
2A7C
2A7C
```

```
2A7C             .SBTTL  "  CHECK VBUS ROUTINE
2A7C        ;++
2A7C        ;
2A7C        ; FUNCTIONAL DESCRIPTION:
2A7C        ;
2A7C        ;        THIS ROUTINE MOVES A BIT FROM THE VBUS BUFFER TO 'BAD_DATA' AND
2A7C        ;        THE EXPECTED VALUE FROM A VBUS TABLE TO 'GOOD_DATA', COMPARES
2A7C        ;        THE TWO AND RETURNS WITH THE RESULT OF THE COMPARE.
2A7C        ;
2A7C        ; IMPLICIT INPUTS:
2A7C        ;
2A7C        ;        V_BUS_BUFFER    - CONTAINS THE CURRENT STATE OF THE VBUS
2A7C        ;
2A7C        ; EXPLICIT INPUTS:
2A7C        ;
2A7C        ;        H & L REGISTERS - CONTAINS THE ADDRESS OF THE EXPECTED VBUS BIT
2A7C        ;                          AND VALUE.
2A7C        ;
2A7C        ; ROUTINE VALUE:
2A7C        ;
2A7C        ;        C BIT CLEAR IF COMPARE IS OK
2A7C        ;        C BIT SET IF COMPARE FAILED.
2A7C        ;
2A7C        ;--
2A7C
2A7C        CHECK_VBUS:
2A7C                 LXI             D,V_BUS_BUFFER  ; GET ADDRESS OF V BUS BUFFER
2A7F                 MOV             A,M             ; GET EXPECTED BIT NUMBER AND VALUE
2A80                 MOV             C,A             ; SAVE
2A81                 ANI             <^X7F>          ; DISCARD VALUE
2A83                 STA             GOOD_DATA       ; SAVE BIT NUMBER
2A86                 STA             BAD_DATA        ; SAVE BIT NUMBER IN BAD DATA
2A89                 MOV             A,C             ; GET BIT VALUE
2A8A                 RLC                             ; PUT IN POSITION 0
2A8B                 ANI             1               ; DISCARD GARBAGE
2A8D                 STA             GOOD_DATA+1     ; SAVE VALUE IN GOOD DATA
2A90                 MOV             A,M             ; GET BIT NUMBER AGAIN
2A91                 ANI             <^X7F>          ; THROW AWAY THE VALUE
2A93                 ADD             E               ; INDEX THE BUFFER ADDRESS TO
2A94                 MOV             E,A             ; THE SPECIFIED BIT NUMBER
2A95                 MVI             A,0             ; ...
2A97                 ADC             D               ; ...
2A98                 MOV             D,A             ; ...
2A99                 LDAX            D               ; GET THE RECEIVED VALUE OF THE BIT
2A9A                 STA             BAD_DATA+1      ; SAVE AS BAD DATA
2A9D                 LDA             GOOD_DATA+1     ; GET EXPECTED VALUE
2AA0                 MOV             C,A             ; SAVE
2AA1                 LDA             BAD_DATA+1      ; GET RECEIVED VALUE
2AA4                 CMP             C               ; DOES EXPECTED = RECEIVED?
2AA5                 JNZ             10$             ; BRANCH IF NO
2AA8                 STC
2AA9                 CMC
2AAA                 RET                             ; RETURN WITH C BIT CLEAR
2AAB        10$:     STC
2AAC                 RET                             ; RETURN WITH C BIT SET
```

```
                                  2AAD  9548
                                  2AAD  9549
```

```
2AAD               .SBTTL  "  CHECK QA FLAG
2AAD      ;++
2AAD      ;
2AAD      ; FUNCTIONAL DESCRIPTION:
2AAD      ;
2AAD      ;        THIS ROUTINE CHECKS IF THE QA FLAG IS SET. IF II IS, AND THE
2AAD      ;        QA COUNT HAS BEEN REACHED, THE ERROR FLAG AND LOOP ERROR FLAG
2AAD      ;        IS SET.
2AAD      ;
2AAD      ; INPUT PARAMETERS:
2AAD      ;
2AAD      ;        NONE
2AAD      ;
2AAD      ; IMPLICIT INPUTS:
2AAD      ;
2AAD      ;        QA_COUNT - CONTAINS THE USER SPECIFIED COUNT TO START THE QA FUNCTION
2AAD      ;        QA_INDEX - CONTAINS THE CURRENT INDEX USED IN THE 'COMPARE' ROUTINES
2AAD      ;
2AAD      ;--
2AAD      CHECK_QA_FLAG:
2AAD               LDA            PROG_CTRL_REG   ; GET THE QA FLAG
2AB0               ANI            QA_FLAG         ; IS THE FLAG SET?
2AB2               JZ             10$             ; BRANCH IF NO
2AB5               LDA            QA_COUNT+1      ; GET HIGH BYTE OF COUNT
2AB8               CPI            -1              ; WAS IT SPECIFIED?
2ABA               JNZ            2$              ; BRANCH IF YES
2ABD               LDA            QA_COUNT        ; GET LOW BYTE
2AC0               CPI            -1              ; WAS IT SPECIFIED?
2AC2               JZ             5$              ; BRANCH IF NO
2AC5      ;
2AC5      ; QA COUNT WAS SPECIFIED. CHECK IF WERE THERE YET
2AC5      ;
2AC5      2$:      LHLD           QA_COUNT        ; GET THE USER SPECIFIED COUNT
2AC8               DCX            H               ; ADJUST TO INDEX RANGE
2AC9               LDA            QA_INDEX+1      ; GET HIGH BYTE OF CURRENT INDEX VALUE
2ACC               CMP            H               ; HIGH BYTE THERE YET?
2ACD               JC             10$             ; BRANCH IF NO
2AD0               JNZ            5$              ; BRANCH IF YES
2AD3               LDA            QA_INDEX        ; GET LOW BYTE OF INDEX
2AD6               CMP            L               ; LOW BYTE THERE YET?
2AD7               JC             10$             ; BRANCH IF NO
2ADA      ;
2ADA      ; SET THE ERROR AND LOOP ERROR FLAGS
2ADA      ;
2ADA      5$:      LDA            SOFT_CTRL_REGA  ; GET THE FLAGS
2ADD               ORI            <ERROR_FLAG!-
2ADD                              LOOP_ERROR_FLAG>
2ADF               STA            SOFT_CTRL_REGA  ; SET THE ERROR FLAGS
2AE2      10$:     RET
2AE3
2AE3
```

```
2AE3                 .SBTTL  "  TYPE GATE ARRAY LIST
2AE3     ;++
2AE3     ;
2AE3     ; FUNCTIONAL DESCRIPTION:
2AE3     ;
2AE3     ;           THIS ROUTINE IS USED TO TYPE A LIST OF GATE ARRAY NAMES.
2AE3     ;
2AE3     ; IMPLICIT INPUTS:
2AE3     ;
2AE3     ;           H & L - CONTAIN ADDRESS OF LIST OF GATE ARRAY CODES
2AE3     ;           A     - CONTAINS THE NUMBER OF CODES IN THE LIST
2AE3     ;
2AE3     ; IMPLICIT OUTPUTS:
2AE3     ;
2AE3     ;           H & L - CONTAIN ADDRESS OF END OF LIST OF GATE ARRAY CODES
2AE3     ;
2AE3     ;--
2AE3     TYPE_GA_LIST:
2AE3
2AE3     ;
2AE3     ; TYPE THE GATE ARRAY MESSAGE AND THE LIST
2AE3     ;
2AE3                 PUSH            $PSW            ; SAVE THE GATE ARRAY COUNT
2AE4                 INX             H               ; POINT AT FIRST GATE ARRAY CODE
2AE5                 PUSH            H               ; SAVE LIST POINTER
2AE6                 TYPE            CRLF
2AEF                 TYPE            FAIL_ARRAY_MSG  ; TYPE 'FAILING GATE ARRAYS: '
2AF8                 POP             H               ; GET THE GATE ARRAY LIST POINTER
2AF9     1$:         PUSH            H               ; SAVE THE POINTER
2AFA                 PUSH            $PSW            ; SAVE CODE
2AFB                 TYPE            DC_MSG          ; TYPE THE DC PREFIX
2B04                 POP             $PSW            ;
2B05                 POP             H               ;
2B06                 MOV             A,M             ; GET THE GATE ARRAY CODE
2B07                 CPI             ALP             ; CHECK IF BIT SLICE GATE ARRAY
2B09                 JC              5$              ; BRANCH IF NOT
2B0C                 CPI             END_SLICE       ; CHECK IF BIT SLICE CODE
2B0E                 JNC             5$              ; BRANCH IF NOT
2B11                 PUSH            H
2B12                 CALL            BIT_SLICE       ; TYPE THE DC NUMBER
2B15                 JMP             10$             ; CONTINUE
2B18     5$:         PUSH            H               ; SAVE THE POINTER
2B19                 MOV             C,A             ; SAVE IN C REG
2B1A                 RLC                             ; MAKE CODE A 9 BYTE INDEX
2B1B                 RLC                             ; ...
2B1C                 RLC                             ; ...
2B1D                 ADD             C               ; ...
2B1E                 MOV             C,A             ; ...
2B1F                 MVI             B,0             ; ...
2B21                 LXI             H,GATE_ARRAY_LST ; GET ADDRESS OF ASCII GATE ARRAY NAMES
2B24                 DAD             B               ; POINT AT ASCII NAME
2B25                 TYPE                            ; TYPE THE NAME
2B2B     10$:        TYPE            CRLF            ; TERMINATE THE LINE
2B34                 TYPE            FAIL_ARR_INDEX  ; SET CURSOR FOR NEXT ARRAY NAME
2B3D                 POP             H               ; GET THE LOOP COUNT FROM THE STACK
```

```
2B3E  9658          POP     $PSW            ; ...
2B3F  9659          DCR     A               ; DECREMENT
2B40  9660          PUSH    $PSW            ; SAVE
2B41  9661          INX     H               ; POINT AT THE NEXT CODE
2B42  9662          JNZ     1$              ; CONTINUE LOOP
2B45  9663          POP     $PSW            ; THROW AWAY LOOP COUNT
2B46  9664          PUSH    H               ; SAVE THE POINTER
2B47  9665          TYPE    CRLF            ; TERMINATE THE LINE
2B50  9666          POP     H               ; RESTORE THE POINTER
2B51  9667          DCX     H               ; BACKUP TO LAST GATE ARRAY CODE
2B52  9668          RET                     ; EXIT
2B53  9669
```

```
2B53          .SBTTL  "  BIT SLICE ALGORITHM
2B53  ;++
2B53  ;
2B53  ; FUNCTIONAL DESCRIPTION:
2B53  ;
2B53  ;       THIS ROUTINE CHECKS TO SEE IF THE XOR OF THE GOOD AND BAD DATA
2B53  ;       FITS THE BIT SLICE FOR THE SPECIFIED GATE ARRAY.
2B53  ;
2B53  ; EXPLICIT INPUTS:
2B53  ;
2B53  ;       A - CONTAINS GATE ARRAY CODE
2B53  ;
2B53  ; IMPLICIT INPUTS:
2B53  ;
2B53  ;       BIT_SLICE_TBL - TABLE OF POINTERS TO BIT SLICE STRINGS
2B53  ;                       INDEXED BY GATE ARRAY NUMBER
2B53  ;
2B53  ;         THE BIT SLICE STRINGS HAVE THE FOLLOWING FORMAT:
2B53  ;
2B53  ;               .BYTE   Number of Slices
2B53  ;               .LONG   Mask for slice # 1
2B53  ;               .LONG   Mask for slice # 2
2B53  ;                 .
2B53  ;               .LONG   Mask for slice # n
2B53  ;               .ASCIC  /Chip name string for slice # 1/
2B53  ;               .ASCIC  /Chip name string for slice # 2/
2B53  ;                 .
2B53  ;               .ASCIC  /Chip name string for slice # n/
2B53  ;--
2B53
2B53  BIT_SLICE:
2B53  ;
2B53  ; CREATE THE INDEX TO THE SPECIFIED GATE ARRAY TABLE
2B53  ;
2B53          SUI             ALP             ; NORMALIZE GATE ARRAY NUMBER
2B55          LXI             H,BIT_SLICE_TBL ; GET ADDRESS OF POINTER TABLE
2B58          RLC                             ; MAKE GATE ARRAY A WORD INDEX
2B59          MOV             E,A
2B5A          MVI             D,0
2B5C          DAD             D               ; INDEX THE TABLE ADDRESS
2B5D  ;
2B5D  ; NOW GET THE ADDRESS OF THE SPECIFIED GATE ARRAY TABLE
2B5D  ;
2B5D          MOV             E,M             ; GET ADDRESS OF GATE ARRAY BIT SLICE
2B5E          INX             H               ; ...
2B5F          MOV             D,M             ; TABLE
2B60          XCHG                            ; PUT IN H & L
2B61
2B61          MOV             B,M             ; GET NUMBER OF SLICES IN THIS ARRAY
2B62          INX             H               ; POINT AT FIRST SLICE MASK
2B63  ;
2B63  ; NOW GET THE XOR OF THE GOOD AND BAD DATA
2B63  ;
2B63          PUSH            B
2B64          PUSH            H
```

```
2B65  9726          LXI      D,BIT_SLICE_TEMP; GET ADDRESS OF TEMP STORAGE
2B68  9727          LXI      B,BAD_DATA       ; GET ADDRESS OF BAD DATA
2B6B  9728          LXI      H,GOOD_DATA      ; GET ADDRESS OF GOOD DATA
2B6E  9729          LDAX     B                ; GET BAD DATA
2B6F  9730          XRA      M                ; GET XOR WITH GOOD DATA
2B70  9731          STAX     D                ; SAVE (BYTE 0)
2B71  9732          INX      B
2B72  9733          INX      D
2B73  9734          INX      H
2B74  9735          LDAX     B                ; BYTE 1
2B75  9736          XRA      M
2B76  9737          STAX     D
2B77  9738          INX      B
2B78  9739          INX      D
2B79  9740          INX      H
2B7A  9741          LDAX     B                ; BYTE 2
2B7B  9742          XRA      M
2B7C  9743          STAX     D
2B7D  9744          INX      B
2B7E  9745          INX      D
2B7F  9746          INX      H
2B80  9747          LDAX     B                ; BYTE 3
2B81  9748          XRA      M
2B82  9749          STAX     D
2B83  9750          POP      H
2B84  9751          POP      B
2B85  9752          DCX      D                ; BACKUP POINTER TO XOR DATA
2B86  9753          DCX      D
2B87  9754          DCX      D
2B88  9755  ;
2B88  9756  ; NOW CHECK XOR DATA WITH THE BIT SLICE MASKS
2B88  9757  ;
2B88  9758  ; B REG CONTAINS NUMBER OF SLICES
2B88  9759  ; H & L CONTAIN POINTER TO MASKS
2B88  9760  ; D & E CONTAIN POINTER TO XOR DATA
2B88  9761  ;
2B88  9762          PUSH     B                ; SAVE SLICE COUNT
2B89  9763          PUSH     H                ; AND POINTER
2B8A  9764          XCHG
2B8B  9765  1$:     MVI      C,4              ; SET COUNTER FOR NUMBER OF BYTES
2B8D  9766  5$:     LDAX     D                ; GET A BYTE OF THE MASK
2B8E  9767          INX      D
2B8F  9768          ANA      M                ; AND WITH XOR DATA
2B90  9769          INX      H
2B91  9770          JNZ      10$              ; BRANCH IF FOUND THE FAILING SLICE
2B94  9771          DCR      C                ; CHECK ALL FOUR BYTES
2B95  9772          JNZ      5$               ; ...
2B98  9773    ;
2B98  9774    ; CHECK NEXT BIT SLICE
2B98  9775    ;
2B98  9776          DCX      H                ; BACKUP POINTER TO XOR DATA
2B99  9777          DCX      H
2B9A  9778          DCX      H
2B9B  9779          DCX      H
2B9C  9780          DCR      B                ; CHECK ALL BIT SLICES
```

```
2B9D              JNZ         1$
2BA0              INR         B                     ; FORCE LAST SLICE OUTPUT
2BA1       ;
2BA1       ; FOUND THE FAILING BIT SLICE
2BA1       ; B REG CONTAINS THE FAILING LOOP COUNT
2BA1       ;
2BA1       10$:   POP         H                     ; GET POINTER TO SLICE COUNT
2BA2              POP         D                     ; AND NUMBER OF SLICES
2BA3              MOV         A,D                   ; GENERATE POINTER TO FIRST MESSAGE STRING
2BA4              RLC                               ; ...
2BA5              RLC                               ; ...
2BA6              ADD         L                     ; ...
2BA7              MOV         L,A                   ; ...
2BA8              MVI         A,0                   ; ...
2BAA              ADC         H                     ; ...
2BAB              MOV         H,A                   ; ...
2BAC              MOV         A,D                   ; GET SLICE NUMBER THAT FAILED
2BAD              SUB         B                     ; ...
2BAE              MOV         C,A                   ; PUT IN B & C
2BAF              MVI         B,0                   ; ...
2BB1              MOV         A,M                   ; GET LENGTH OF STRING
2BB2              PUSH        H                     ; SAVE H & L
2BB3              PUSH        B                     ; SAVE SLICE NUMBER
2BB4              CALL        MULTIPLY_A_BC         ; GET POINTER TO APPROPRIATE STRING
2BB7              POP         H                     ; GET SLICE NUMBER
2BB8              DAD         D                     ; ADD TO STRING INDEX
2BB9              XCHG                              ; PUT IN D & E
2BBA              POP         H                     ; RESTORE H & L
2BBB              DAD         D                     ; POINT TO SPECIFIED STRING
2BBC              TYPE                              ; TYPE THE DC NUMBER
2BC2              RET                               ; EXIT
```

```
2BC3              .SBTTL  "  READ RTEMP REGISTER
2BC3      ;++
2BC3      ;
2BC3      ; FUNCTIONAL DESCRIPTION:
2BC3      ;
2BC3      ;       THIS ROUTINE IS USED TO READ THE CONTENTS OF AN RTEMP REGISTER
2BC3      ;       IN THE 11/750 CPU.
2BC3      ;
2BC3      ; EXPLICIT INPUTS:
2BC3      ;
2BC3      ;       A REG - CONTAINS THE REGISTER NUMBER TO READ
2BC3      ;
2BC3      ; EXPLICIT OUTPUTS:
2BC3      ;
2BC3      ;       WH REG - CONTAINS THE CONTENTS OF THE SPECIFIED R TEMP REGISTER
2BC3      ;
2BC3      ; SIDE EFFECTS:
2BC3      ;
2BC3      ;       ALL 8085 REGISTERS ARE LOST
2BC3      ;--
2BC3      READ_RTEMP:
2BC3              STA             READ_R_TEMP      ; SAVE REGISTER NUMBER
2BC6              LXI             D,RDM_R_MIC_WORD; GET ADDRESS OF MICRO WORD
2BC9              LXI             H,READ_R_TEMP    ; GET ADDRESS OF REGISTER NUMBER
2BCC              MVI             B,34             ; SET STARTING BIT NUMBER
2BCE              MVI             C,39             ; AND ENDING BIT NUMBER
2BD0              XRA             A                ; CLEAR INHIBIT PARITY FLAG
2BD1              PUSH            $PSW             ; CLEAR INHIBIT SCRAMBLE FLAG
2BD2              CALL            INSERT_FIELD     ; PUT REGISTER NUMBER IN MICRO WORD
2BD5              WRITE$_DCS      DCS_SCRATCH_ADR,-
2BD5                              RDM_R_MIC_WORD,1; LOAD THE MICRO WORD
2BE8              WRITE$_DCS      DCS_SCRATCH_ADR+1,-
2BE8                              NOP_MICRO_WORD,1; ...
2BFB              EXECUTE         DCS_SCRATCH_ADR,1; GET CONTENTS OF REGISTER
2C0E              RET                              ; EXIT
```

```
2C0F          .SBTTL  "  DISPLAY DATA
2C0F    ;++
2C0F    ;
2C0F    ; FUNCTIONAL DESCRIPTION:
2C0F    ;
2C0F    ;       THIS ROUTINE IS USED TO TYPE A <TAB> FOLLOWED BY THE LONG WORD
2C0F    ;       CONTENTS OF THE SPECIFIED LOCATION, FOLLOWED BY A CRLF.
2C0F    ;
2C0F    ; EXPLICIT INPUTS:
2C0F    ;
2C0F    ;       H & L - CONTAIN ADDRESS OF DATA TO TYPE
2C0F    ;
2C0F    ;--
2C0F    DISPLAY_DATA:
2C0F            PUSH            H                       ; SAVE ADDRESS
2C10            TYPE            TAB                     ; TYPE THE TAB
2C19            POP             H
2C1A            TYPEL                                   ; TYPE THE DATA
2C22            TYPE            CRLF                    ; TERMINATE THE LINE
2C2B            RET                                     ; EXIT
```

```
2C2C                .SBTTL  "  READ MSRC FIELD
2C2C        ;++
2C2C        ;
2C2C        ; FUNCTIONAL DESCRIPTION:
2C2C        ;
2C2C        ;       THIS ROUTINE IS USED TO READ THE CONTENTS OF AN MSRC FIELD
2C2C        ;       IN THE 11/750 CPU.
2C2C        ;
2C2C        ; EXPLICIT INPUTS:
2C2C        ;
2C2C        ;       A REG - CONTAINS THE REGISTER NUMBER TO READ
2C2C        ;
2C2C        ; EXPLICIT OUTPUTS:
2C2C        ;
2C2C        ;       WH REG - CONTAINS THE CONTENTS OF THE SPECIFIED M SRC REGISTER
2C2C        ;
2C2C        ; SIDE EFFECTS:
2C2C        ;
2C2C        ;       ALL 8085 REGISTERS ARE LOST
2C2C        ;--
2C2C        READ_MTEMP:
2C2C                STA             READ_R_TEMP     ; SAVE REGISTER NUMBER
2C2F                LXI             D,RDM_M_MIC_WORD; GET ADDRESS OF MICRO WORD
2C32                LXI             H,READ_R_TEMP   ; GET ADDRESS OF REGISTER NUMBER
2C35                MVI             B,64            ; SET STARTING BIT NUMBER
2C37                MVI             C,68            ; AND ENDING BIT NUMBER
2C39                XRA             A               ; CLEAR INHIBIT PARITY FLAG
2C3A                PUSH            $PSW            ; CLEAR INHIBIT SCRAMBLE FLAG
2C3B                CALL            INSERT_FIELD    ; PUT REGISTER NUMBER IN MICRO WORD
2C3E                WRITE$_DCS      DCS_SCRATCH_ADR,-
2C3E                                RDM_M_MIC_WORD,1; LOAD THE MICRO WORD
2C51                WRITE$_DCS      DCS_SCRATCH_ADR+1,-
2C51                                NOP_MICRO_WORD,1; ...
2C64                EXECUTE         DCS_SCRATCH_ADR,1; GET CONTENTS OF REGISTER
2C77                RET                             ; EXIT
2C78
2C78        ;
2C78        ; THE TEST OVERLAYS GET LOADED STARTING HERE
2C78        ;
2C78
2C78
2C78
2C78        TEST_BUFFER:
2C78
2C78                .END
```

```
$CONTROL_C_CODE= 0000000A             CLA            = 00000003             CS_ADD_TRAP_HGH= 00007423
$FILE_NOT_FOUND= 00000009             CLEAR_ACTION     00001945 R    01     CS_ADD_TRAP_LOW= 00007422
$PSW           = 00000006             CLEAR_CTRL_FILE= 00000002             CURRENT_BIT      00000C9A R    01
$SP            = 00000006             CLEAR_DCS_CF     00000FD7 R    01     CURRENT_PC       000003B8 R    01
A              = 00000007             CLEAR_DONE       000019BC R    01     CURRENT_REC_NMB  0000013B R    01
ACV            = 0000000F             CLEAR_ERR        000019BF R    01     CYCLE_FLAG     = 00000008
AC_LOW         = 00000080             CLKDCS         = 00007427             CYCLE_KEY      = 00000020
ADD            = 00000019             CLK_CTRL_0     = 00000020             C_INIT_TEST_PC   000001B7 R    01
ADDR_MATCH_HI  = 00007405             CLK_CTRL_0_RO  = 00000040             D              = 00000002
ADDR_MATCH_LO  = 00007404             CLK_CTRL_1     = 00000040             DATA_MSG         000005B7 R    01
ADD_NAME         00000963 RG   01     CLK_CTRL_1_RO  = 00000080             DATA_TYPE        00000447 R    01
ADK            = 0000000B             CLK_MSG_ADDR     00000453 R    01     DCSAD          = 00007401
ALK            = 00000000             CLK_STOP_MSG1    00000481 R    01     DCSCF          = 00007403
ALP            = 00000016             CLK_STOP_MSG2    000004A3 R    01     DCSCR          = 00007402
ALP_NAME         00000713 RG   01     CLK_STOP_MSG3    000004B4 R    01     DCSDA          = 00007400
ARG_LIST         000003BE R    01     CLK_STOP_MSG4    000004BA R    01     DCS_ADDR_CLEAR = 00000002
A_REG_BYTE_0   = 00007410             CLK_STOP_MSG5    000004CC R    01     DCS_ADDR_MSG     00000456 R    01
A_REG_BYTE_1   = 00007411             CLOCK_STOPED   = 00000080             DCS_ADDR_REG_RO= 00007427
A_REG_BYTE_2   = 00007412             CMD_DISPATCH     00000304 R    01     DCS_ADDR_REG_WO= 00007401
A_REG_BYTE_3   = 00007413             CMD_ENTRY        0000000D R    01     DCS_BIT_MAP      00000C9E R    01
B              = 00000000             CMD_LINE_LENGTH  000002F1 R    01     DCS_CTRL_FILE  = 00007403
BADD1          = 00007426             CMI            = 00000007             DCS_CTRL_REG   = 00007402
BAD_DATA         00000443 R    01     CMICTL         = 0000741C             DCS_CTRL_RE_CPY  0000005C R    01
BEGIN_LOOP       00001CE0 R    01     CMI_COMPLETE   = 00000020             DCS_DATA_REG   = 00007400
BELL             00000592 R    01     CMI_CTRL_CLEAR = 00000001             DCS_SCRATCH_ADR= 00000036
BELL_COUNT       00000591 R    01     CMI_CTRL_REG   = 0000741C             DCS_START      = 00001800
BELL_FLAG      = 00000008             CMI_GO         = 00000008             DCS_WRITE_ADR    000001AC R    01
BELL_KEY       = 0000000E             CMI_NAME         000000F2 R    01     DCS_WRITE_PTR    000001AE R    01
BIT_COUNT        00000C9B R    01     CMI_READ       = 00000040             DCS_WRITE_WDCNT  000001B0 R    01
BIT_NMB_TO_MSK   00002A6E R    01     CMI_STATUS_0   = 00000008             DC_LOW         = 00000040
BIT_SLICE        00002B53 R    01     CMI_STATUS_1   = 00000010             DC_MSG           0000062C R    01
BIT_SLICE_TBL    000006F5 RG   01     CMI_WRITE      = 00000020             DECODE_ERR       00001B97 R    01
BIT_SLICE_TEMP   000002C4 R    01     CML            = 0000000E             DECODE_FLAG_NAM  00001B85 R    01
BREAK_ACTION     000015A1 R    01     CMP_GOOD_BAD     00002719 R    01     DEPOSIT_ACTION   00001B84 R    01
BUFF_ADDR_LOW  = 00007426             COMMAND_LIST     0000032E R    01     DESELECT_DCS     00001383 R    01
BURST_CLOCK      000020B5 R    01     COMMAND_PARSE    000015A8 R    01     DIAGNOSE_ACTION  000016E5 R    01
BURST_STOP_FLAG= 00000080             COMMA_MSG        0000062A R    01     DIAGNOSE_DONE    000017A2 R    01
BURST_TEMP       00000451 R    01     COMPARE_REG      00001F07 R    01     DIAGNOSE_ERR     000017A0 R    01
BYTE           = 00000001             COMPARE_REG_MSK  00001F5A R    01     DIAGNOSE_FLAG  = 00000004
C              = 00000001             COMPARE_VBUS     00001FE4 R    01     DIAGNOSE_LOOP    00001712 R    01
CACHE_ENA_ADR    00000275 R    01     CON            = 00000011             DIRECT_MSG       00000076 R    01
CACHE_ENA_DATA   00000280 R    01     CONTINUE_ACTION  0000183A R    01     DIRECT_SEARCH    0000121B R    01
CACHE_WRITE      0000028B R    01     CONTINUE_FLAG  = 00000002             DISABLE_CMI      000013D5 R    01
CAK            = 0000000A             CONTROL_C_FLAG = 00000001             DISPLAY_DATA     00002C0F R    01
CALC_PARITY      00002A1C R    01     CONT_FLAG      = 00000020             DPM            = 00000000
CCC            = 00000004             CONT_KEY       = 0000001C             DPM_NAME         000000A5 R    01
CCF_SUB_ACT      000019AB R    01     CONT_SUB_ACTION  0000175A R    01     DUMP_EXIT        00002689 R    01
CCS            = 00000005             CPU_INIT         00001459 R    01     DUMP_LOG         00002646 R    01
CCS_NAME         000000DC R    01     CPU_RUN        = 00000010             DUMP_RET         00002692 R    01
CFLAG_SUB_ACT    0000198F R    01     CRLF             00000CEE R    01     E              = 00000003
CF_KEY         = 00000024             CSAMR          = 00007404             ENABLE_CMI       00001417 R    01
CHECK_QA_FLAG    00002AAD R    01     CSBUF          = 00007424             ENABLE_STALL     00002520 R    01
CHECK_QV         000011D4 R    01     CSTRP          = 00007422             ENABLE_TRAP      00002517 R    01
CHECK_VBUS       00002A7C R    01     CS_ADD_BUFF_HGH= 00007425             ENDPASS_MSG      00000584 R    01
CHR            = 00000005             CS_ADD_BUFF_LOW= 00007424             END_LOOP         00001D29 R    01
```

```
END_PASS         0000242E R      01       FILE_FOR_MIC     00000154 R      01       J_INDEX_TBL      00000415 R      01
END_SLICE      = 00000025                 FILE_FOR_RDM     00000166 R      01       KEYWORD_LIST     00000343 R      01
END_TEST         00002416 R      01       FILE_NAME_HEAD = 0000014E R      01       K_INDEX_TBL      0000041C R      01
EN_TRACE_REG   = 00000008                 FILE_NAME_TAIL = 0000016C R      01       L              = 00000005
ERROR_FLAG     = 00000010                 FILE_NAM_INDEX   00000063 R      01       LITERAL        = 00000001
ERROR_LOG        00002529 R      01       FILE_NAM_PTR     00000139 R      01       LOAD_DCS         00001C0A R      01
ERROR_LOG_ADR  = 000032FC                 FILL_DCS         000014D1 R      01       LOAD_FIELD       00002437 R      01
ERROR_LOOP       00001D88 R      01       FIO            = 0000001D                 LOAD_INDEX       000024B8 R      01
ERROR_LOOP_ADDR  00000425 R      01       FIO_NAME         00000AF1 RG     01       LOAD_MEMSCAR     00000228 R      01
ERROR_MSG        0000059B R      01       FLAG_KEY       = 00000004                 LOAD_MEMSCR      00000249 R      01
ERROR_PC         00000594 R      01       FMR            = 00000022                 LOAD_REG         00001DF8 R      01
ERR_LOG_INIT   = 00000001                 FMR_NAME         00000BED RG     01       LOG_ADDRESS      000004E5 R      01
EXAMINE_ACTION   000019C1 R      01       FPA            = 00000002                 LOG_EXIT         00002607 R      01
EXAMINE_DISPLAY  00001B79 R      01       FPA_NAME         000000BB R      01       LOG_FAIL         000025EF R      01
EXAMINE_ERR      00001B7F R      01       FP_BOOT_0      = 00000001                 LOG_MSG_1        000004E9 R      01
EXAMINE_EX       00001B81 R      01       FP_BOOT_1      = 00000002                 LOG_MSG_2        00000507 R      01
EXAMINE_MA       00001A5B R      01       FP_START_0     = 00000004                 LOG_MSG_3        00000533 R      01
EXAMINE_MD       00001A51 R      01       FP_START_1     = 00000008                 LOG_SUCCESS      000025FC R      01
EXAMINE_MT       00001A39 R      01       FQA            = 00000014                 LOG_SYNDROME     000004E8 R      01
EXAMINE_PB       00001A56 R      01       FRONT1         = 00007420                 LONG           = 00000004
EXAMINE_PC       00001A47 R      01       FRONT2         = 00007421                 LOOP_ACTION      00001848 R      01
EXAMINE_PS       00001AD5 R      01       FRONT_PNL_1    = 00007420                 LOOP_ADDRESS     00000423 R      01
EXAMINE_RT       00001A2B R      01       FRONT_PNL_2    = 00007421                 LOOP_ERROR_FLAG= 00000020
EXAMINE_SF       00001B41 R      01       FRONT_PNL_LOCK = 00000080                 LOOP_FLAG      = 00000002
EXAMINE_SR       00001A89 R      01       GATE_ARRAY_LST   0000062F R      01       LOOP_KEY       = 0000000A
EXAMINE_ST       00001B13 R      01       GET_CMD_LINE     000014EF R      01       LOST_FLAG      = 00000010
EXAMINE_TEMP     000003BC R      01       GET_INDEX_VALUE  000026AF R      01       LOST_KEY       = 0000001A
EXAMINE_VA       00001A4C R      01       GOOD_DATA        0000043F R      01       LOW            = 00000000
EXAMINE_WD       00001A60 R      01       H              = 00000004                 M              = 00000006
EXP_STALL_FLAG = 00000010                 HALT_FLAG      = 00000001                 M$TEST_SIZE    = 000007A2
EXP_UTRAP_FLAG = 00000040                 HALT_FLAG_NAME   0000034B R      01       M$TEST_SIZE_D  = 000003E2
EX_DISP_WH       00001B76 R      01       HALT_KEY       = 00000008                 MA0            = 00000008
FAC            = 00000020                 HALT_ON_MATCH  = 00000001                 MA0_NAME         000000FD R      01
FAC_NAME         00000B9E RG     01       HEAD           = FFFF7C00                 MA1            = 0000000A
FAIL_ARRAY_MSG   000005DA R      01       HIGH           = 00000001                 MA1_NAME         00000113 R      01
FAIL_ARR_INDEX   000005F0 R      01       HYPHEN           000005CE R      01       MA2            = 0000000C
FAIL_MOD_INDEX   00000618 R      01       H_FROM_WBUS    = 00000010                 MA2_NAME         00000129 R      01
FAULT          = 00000002                 IB_FLAG        = 00000020                 MAD            = 00000013
FCC            = 00000015                 IB_KEY         = 00000012                 MAIN32         = 0000741D
FCS            = 00000021                 IF_ERROR         00002052 R      01       MAINT_A_TO_CMI = 00000080
FCS_NAME         00000BB0 RG     01       INCREMENT_INDEX  000024FA R      01       MAINT_CMI_TO_MH= 00000020
FETCH_0          0000147B R      01       INDEX_NAM_TBL    00000408 R      01       MAINT_DCS_ENABL= 00000004
FETCH_FLAG     = 00000008                 INIT_LOOP_VALUE  0000269B R      01       MAINT_MD_TO_CMI= 00000040
FETCH_MIC_INSTR  00001EA6 R      01       INIT_TEST_PC     000001B5 R      01       MAINT_REG      = 0000741D
FEX            = 0000001E                 INIT_THE_WORLD   00001CBF R      01       MAINT_STB_INH  = 00000002
FEX_NAME         00000B6A RG     01       INSERT_FIELD     00002966 R      01       MAINT_TRAP_OFF = 00000001
FFH            = 00000024                 INSERT_FIELD_X   00002A1B R      01       MAINT_WB_TO_WH = 00000008
FFH_NAME         00000C49 RG     01       INSTR_FLAG     = 00000004                 MAINT_WD_TO_WB = 00000010
FFL            = 00000023                 INSTR_KEY      = 00000018                 MAP            = 00000012
FFL_NAME         00000C0C RG     01       INSTR_TABLE      000003CE R      01       MAREG          = 00007410
FIC            = 0000001F                 INS_FLD_BUFF     00000C90 R      01       MASK             00001ECD R      01
FIC_NAME         00000B89 RG     01       INT            = 00000010                 MASK_DISPLAY     00001B6C R      01
FILE_FOR_CMC     00000160 R      01       INVALID_COMMAND  000002F3 R      01       MASTER_HALT_EN = 00000008
FILE_FOR_DPM     0000014E R      01       IRD            = 00000005                 MAX_ARGUMENTS  = 00000010
FILE_FOR_FPA     0000015A R      01       I_INDEX_TBL      0000040E R      01       MA_KEY         = 00000038
```

```
MC0            = 00000003                ONEQUAL        = 00000002                PRIM_REV       = 00000008
MC0_NAME         000000C6 R      01      ONERROR        = 00000000                PRINT_N          00002617 R      01
MC1            = 00000009                OPEN_TU58        00000E2F R      01      PRINT_N_EX       00002645 R      01
MC1_NAME         00000108 R      01      OP_BEGIN_LOOP  = 00000008                PRINT_S          00002608 R      01
MC2            = 0000000B                OP_BURST_CLOCK = 00000002                PRK            = 0000000C
MC2_NAME         0000011E R      01      OP_CODE_DISP     00001B99 R      01      PROGRAM_INIT     00000CF1 R      01
MDL            = 0000001B                OP_COMPARE_REG = 00000004                PROGRAM_TITLE    00000067 R      01
MDL_NAME         000009EF RG     01      OP_COMPARE_REGM= 00000006                PROG_CTRL_REG    0000005D R      01
MDR            = 00000018                OP_COMPARE_VB  = 00000020                PROMPT           000002EC R      01
MDREG          = 00007414                OP_DUMPLOG     = 00000030                PS_KEY         = 0000003E
MDR_NAME         0000087A RG     01      OP_ENABL_STALL = 00000038                PUT_D_IN_TBL     0000166E R      01
MD_KEY         = 00000034                OP_ENABL_TRAP  = 0000002E                QA_COUNT         00000449 R      01
MD_REG_BYTE_0  = 00007414                OP_END_FILE    = 00000024                QA_FLAG        = 00000040
MD_REG_BYTE_1  = 00007415                OP_END_LOOP    = 0000000A                QA_INDEX         0000044B R      01
MD_REG_BYTE_2  = 00007416                OP_END_TEST    = 0000000C                QA_KEY         = 00000014
MD_REG_BYTE_3  = 00007417                OP_ERRLOG      = 00000032                QV_FLAG        = 00000002
MEC            = 0000001A                OP_ERROR_LOOP  = 0000000E                QV_KEY         = 00000028
MEC_NAME         000009C0 RG     01      OP_FETCH       = 00000022                QV_SUB_ACTION    0000174F R      01
MESSAGE_LENGTH   0000016E R      01      OP_IF_ERROR    = 00000012                RDCTRL         = 0000741E
MESSAGE_POINTER  0000016C R      01      OP_INC_INDEX   = 0000002A                RDM            = 00000004
MHREG          = 00007418                OP_INIT        = 00000014                RDM_FLAG       = 00000004
MH_REG_BYTE_0  = 00007418                OP_LOAD_DCS    = 00000000                RDM_INDEX      = 00000004
MH_REG_BYTE_1  = 00007419                OP_LOAD_FIELD  = 00000026                RDM_KEY        = 0000002A
MH_REG_BYTE_2  = 0000741A                OP_LOAD_INDEX  = 00000028                RDM_LAST       = 00000008
MH_REG_BYTE_3  = 0000741B                OP_LOAD_REG    = 00000016                RDM_M_MIC_WORD   000001DB R      01
MIC            = 00000001                OP_MASK        = 00000018                RDM_NAME         000000D1 R      01
MICROWORD      = 0000000A                OP_NEW_TEST    = 0000001A                RDM_PS_MIC_WORD  00000207 R      01
MICRO_ADDR_INH = 00000080                OP_PATTERN_GEN = 00000010                RDM_R_MIC_WORD   000001D0 R      01
MICRO_CODE_HEAD= 000001BA R      01      OP_PRINT_N     = 00000036                RDM_SF_MIC_WORD  0000021D R      01
MICRO_CODE_TAIL= 000002B7 R      01      OP_PRINT_S     = 00000034                RDM_SR_MIC_WORD  000001F1 R      01
MIC_INSTR_CNT    000001B1 R      01      OP_SAVE_INDEX  = 0000002C                RDM_ST_MIC_WORD  00000212 R      01
MIC_NAME         000000B0 R      01      OP_SKIP        = 0000001C                RDM_SUB_ACTION   00001744 R      01
MIC_TICK_CNT     000001B2 R      01      OP_SUB_TEST    = 0000001E                RDM_TAIL       = 0000016C R      01
MME_DIS_ADR      00000280 R      01      P0_PARITY_MSK    00000570 R      01      RDM_WD_MIC_WORD  000001E6 R      01
MME_DIS_DATA     00000280 R      01      P1_PARITY_MSK    0000057A R      01      RD_CTRL_REG    = 0000741E
MODE_TYPE        00000448 R      01      PARITY_ERROR   = 00000020                READ_MTEMP       00002C2C R      01
MODULE_LST       000000A5 R      01      PARSE_DONE       0000161E R      01      READ_OVERLAY     0000100E R      01
MODULE_MSG       00000606 R      01      PARSE_ERR        0000161C R      01      READ_RTEMP       00002BC3 R      01
MODULE_NAM_TBL   0000008B R      01      PARSE_HEX_DONE   000016CB R      01      READ_R_TEMP      00000C9D R      01
MONITOR_SIZE   = 00002C5E                PARSE_HEX_ERR    000016E3 R      01      READ_TU58        00000DFC R      01
MSQ            = 00000008                PARSE_HEX_NUMB   00001679 R      01      READ_V_BUS       000011FB R      01
MTEMP_COMMON     00001A41 R      01      PARSE_HEX_TEMP   00000388 R      01      REMOTE         = 00000001
MT_KEY         = 0000002E                PARSE_TABLE      0000031A R      01      RETURN_ACTION    00001845 R      01
MULTIPLY_A_BC    000026D6 R      01      PARSE_TBL_PTR    00000318 R      01      RETURN_TO_RDM    00000F81 R      01
M_CLK_ADDR       00000455 R      01      PAR_CHK_ENABL  = 00000008                RING_BELL        00002743 R      01
NER_FLAG       = 00000004                PAR_STOP_ENABL = 00000010                RT_KEY         = 0000002C
NER_KEY        = 0000000C                PASS_COUNT       00000062 R      01      R_RDM_MIC_WORD   000001C5 R      01
NEW_TEST         00001C3B R      01      PASS_KEY       = 00000002                SAC            = 00000009
NOERROR        = 00000001                PASS_SUB_ACTION  00001797 R      01      SAVED_SP         0000005E R      01
NOP_MICRO_WORD   000001BA R      01      PATTERN_ADDRESS  00000427 R      01      SAVE_BAD_DATA    000026E3 R      01
NOTEQUAL       = 00000003                PATTERN_GEN      00001D8F R      01      SAVE_GOOD_DATA   000026FE R      01
NOT_MSG          000005C0 R      01      PB_INIT        = 00000040                SAVE_INDEX       00002505 R      01
NO_MICRO_WORDS = 00000017                PB_KEY         = 00000036                SA_CLOCK       = 00000080
NUMBER_OF_FILES= 00000005                PC_KEY         = 00000030                SA_FLAG        = 00000010
ONE              000002F2 R      01      PHB            = 00000007                SA_KEY         = 00000010
```

```
SA_START_STOP   = 00000080                  SUB_TEST_NUMB     00000061 R      01        UTR            = 0000000D
SBE_FLAG        = 00000020                  TAB               000005C5 R      01        VAL            = 00002305
SBE_KEY         = 00000042                  TEMP_BUFFER       000001A3 R      01        VA_KEY         = 00000032
SCRAMBLE_FLAG     00000C9C R      01        TERM_ERROR        0000018E R      01        VBUS_CLOCK     = 00000080
SEARCH_DONE       00001648 R      01        TERM_ERR_CODE     0000018D R      01        VBUS_COMPARE   = 00000080
SEARCH_ERR        0000164C R      01        TERM_INP_BUFF     0000016F R      01        VBUS_LOAD      = 00000010
SEARCH_LIST       00001621 R      01        TERM_INTERRUPT    00000ED5 R      01        VBUS_MSG         000003AC R      01
SEC_REV         = 00000007                  TEST            = 00000004                  VBUS_MSG_2       000005D0 R      01
SELECT_DCS        00001392 R      01        TESTS_PER_LINE    000002B8 R      01        VBUS_SERIAL_IN = 00000040
SETCLR_CF_ADDR    000001B3 R      01        TEST_BUFFER       00002C78 R      01        VBUS_SERIAL_OUT= 00000001
SETCLR_CF_DATA    000001B4 R      01        TEST_KEY        = 00000000                  VBUS_TBL_ADDR    0000044D R      01
SET_ACTION        00001863 R      01        TEST_MSG          000005A4 R      01        VBUS_TBL_COUNT   0000044F R      01
SET_DCS_CF        00000F9F R      01        TEST_MSG_1        000002BB R      01        VB_KEY         = 00000026
SET_DONE          00001940 R      01        TEST_MSG_2        000002C1 R      01        V_BUS_BUFFER     000002C4 R      01
SET_ERR           00001943 R      01        TEST_NUMBER       00000060 R      01        V_BUS_STROBE   = 00000001
SFLAG_SUB_ACT     00001899 R      01        TEST_PC           000003B6 R      01        WBUS_FROM_D    = 00000020
SF_KEY          = 00000040                  TEST_SIZE         000001B9 R      01        WDREG          = 0000742C
SHOW_ACTION       000017AD R      01        TEST_SPAN_END     0000038D R      01        WD_KEY         = 0000003A
SHOW_MESSAGE_1    00000391 R      01        TEST_SUB_ACTION   00001765 R      01        WD_REG_BYTE_0  = 0000742C
SHOW_MESSAGE_2    0000039D R      01        TICK_FLAG       = 00000002                  WD_REG_BYTE_1  = 0000742D
SHOW_TEMP         000003AB R      01        TICK_KEY        = 00000022                  WD_REG_BYTE_2  = 0000742E
SINGLE_INSTR      00002906 R      01        TITLE_LENGTH      000002B9 R      01        WD_REG_BYTE_3  = 0000742F
SINGLE_MIC_INST   00000F30 R      01        TOK             = 00000006                  WHREG          = 00007430
SINGLE_TICK       00000F57 R      01        TOTAL_BLOCKS      0000013D R      01        WH_REG_BYTE_0  = 00007430
SKIP              00001E31 R      01        TOTAL_NMB_TESTS   00000066 R      01        WH_REG_BYTE_1  = 00007431
SKIP_SPACES       0000164E R      01        TPC_MSG           00000C86 R      01        WH_REG_BYTE_2  = 00007432
SOFT_CTRL_REG     00000059 R      01        TRACE_ENABL     = 00000004                  WH_REG_BYTE_3  = 00007433
SOFT_CTRL_REGA    0000005A R      01        TRAP_HALT_EN    = 00000020                  WORD           = 00000002
SOFT_CTRL_REGB    0000005B R      01        TRAP_OFF        = 00000002                  WRITE_DCS        00000EF4 R      01
SOMM_ADDRESS      0000038F R      01        TR_FLAG         = 00000080                  X              = 00000003
SOMM_FLAG       = 00000001                  TR_KEY          = 00000016                  Y              = 0000004E
SOMM_KEY        = 00000006                  TSTSPAN_FLAG    = 00000040
SOMM_MSG          00000464 R      01        TU58_BUFF_ADDR    00000134 R      01
SPA             = 00000002                  TU58_ERROR        00000140 R      01
SPEC_TEST_NUMB    0000038C R      01        TU58_ERR_CODE     0000013F R      01
SPEC_TEST_PC      000003BA R      01        TU58_REC_COUNT    00000136 R      01
SRK             = 00000001                  TU58_REC_NUMB     00000137 R      01
SRM             = 00000017                  TYPEN_DATA_PTR    000001A0 R      01
SRM_NAME          000007CC RG     01        TYPEN_DATA_TYPE   000001A2 R      01
SR_KEY          = 0000003C                  TYPE_ASCII        00000E82 R      01
SSOMM_SUB_ACT     00001916 R      01        TYPE_ASCII_U      00000E8D R      01
SSTEP_SUB_ACT     000018D6 R      01        TYPE_ERROR        00002764 R      01
STACK_SIZE      = 00000400                  TYPE_ERR_BUFFER   00000596 R      01
START1            0000000E R      01        TYPE_ERR_TEMP     0000059A R      01
STARTING_RECORD   00000064 R      01        TYPE_FLAGS        0000149A R      01
START_EXECUTION   0000139F R      01        TYPE_GA_LIST      00002AE3 R      01
STATUS          = 00007406                  TYPE_NUMERIC      00000E9F R      01
STATUS_REG      = 00007406                  TYPE_TEST_FLAG    000002B7 R      01
STEP_ADDRESS      00000450 R      01        TYPE_TEST_NUMB    000012C4 R      01
STEP_KEY        = 0000001E                  TYPE_TU58_ERROR   00000E50 R      01
STOP_CLOCK      = 00000004                  UBI             = 00000006
STROBE_CMI      = 00000040                  UBI_NAME          000000E7 R      01
ST_KEY          = 0000001E                  UDP             = 0000001C
SUB_TEST          00001E94 R      01        UDP_NAME          00000A4C RG     01
SUB_TEST_MSG      000005AC R      01        USER_PASS_CNT     0000038E R      01
```

```
                                                   +-----------------+
                                                   ! Psect synopsis !
                                                   +-----------------+

PSECT name                      Allocation          PSECT No.  Attributes
-----------                     -----------         ---------  ----------
.  ABS  .                       00000000  (      0.)  00 (   0.)  NOPIC    USR    CON    ABS    LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
. BLANK .                       00002C78  (11384.)  01 (   1.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD      WRT NOVEC BYTE
```

```
                                                        +-----------------------------+
                                                        ! Symbol Cross Reference !
                                                        +-----------------------------+

SYMBOL              VALUE               DEFINITION          REFERENCES...
------              -----               ----------          -------------
$CONTROL_C_CODE=0000000A            863        (4)          2746     (13)       4214     (34)
$FILE_NOT_FOUND=00000009            863        (4)          2521     (9)
$PSW               =00000006        856        (3)          1384     (5)        1395     (5)        1402     (5)        1408     (5)        1429     (5)
                                                            2695     (12)       2703     (12)       2800     (14)       2848     (14)       3043     (18)
                                                            3055     (18)       3058     (18)       3063     (18)       3103     (19)       3114     (19)
                                                            3117     (19)       3122     (19)       3213     (20)       3224     (20)       3225     (20)
                                                            3248     (20)       3257     (20)       3282     (20)       3284     (20)       3286     (20)
                                                            3297     (20)       3337     (20)       3339     (20)       3341     (20)       3374     (20)
                                                            3375     (20)       3380     (20)       3406     (20)       3462     (21)       3466     (21)
                                                            3467     (21)       3482     (21)       3485     (21)       3643     (23)       3654     (23)
                                                            3655     (23)       3709     (23)       3717     (23)       4007     (28)       4013     (28)
                                                            4038     (29)       4044     (29)       4127     (32)       4130     (32)       4649     (39)
                                                            4653     (39)       4655     (39)       4658     (39)       4666     (39)       4676     (39)
                                                            6297     (54)       6299     (54)       6309     (54)       6319     (54)       6325     (54)
                                                            6507     (56)       6569     (57)       6577     (57)       6580     (57)       7992     (70)
                                                            9002     (87)       9020     (87)       9022     (87)       9024     (87)       9073     (87)
                                                            9093     (87)       9095     (87)       9098     (87)       9213     (89)       9218     (89)
                                                            9285     (89)       9302     (89)       9350     (89)       9625     (94)       9632     (94)
                                                            9634     (94)       9658     (94)       9660     (94)       9663     (94)       9840     (96)
                                                            9897     (98)
$SP                =00000006        856        (3)          1375     (5)        1384     (5)        1386     (5)        1395     (5)        1402     (5)
                                                            1408     (5)        1429     (5)        2695     (12)       2703     (12)       2800     (14)
                                                            2848     (14)       3043     (18)       3055     (18)       3058     (18)       3063     (18)
                                                            3103     (19)       3114     (19)       3117     (19)       3122     (19)       3213     (20)
                                                            3224     (20)       3225     (20)       3248     (20)       3257     (20)       3282     (20)
                                                            3284     (20)       3286     (20)       3297     (20)       3337     (20)       3339     (20)
                                                            3341     (20)       3374     (20)       3375     (20)       3380     (20)       3406     (20)
                                                            3462     (21)       3466     (21)       3467     (21)       3482     (21)       3485     (21)
                                                            3643     (23)       3654     (23)       3655     (23)       3709     (23)       3717     (23)
                                                            4007     (28)       4013     (28)       4038     (29)       4044     (29)       4127     (32)
                                                            4130     (32)       4649     (39)       4653     (39)       4655     (39)       4658     (39)
                                                            4666     (39)       4676     (39)       6297     (54)       6299     (54)       6309     (54)
                                                            6319     (54)       6325     (54)       6507     (56)       6569     (57)       6577     (57)
                                                            6580     (57)       7992     (70)       9002     (87)       9020     (87)       9022     (87)
                                                            9024     (87)       9073     (87)       9093     (87)       9095     (87)       9098     (87)
                                                            9213     (89)       9218     (89)       9285     (89)       9302     (89)       9350     (89)
                                                            9625     (94)       9632     (94)       9634     (94)       9658     (94)       9660     (94)
                                                            9663     (94)       9840     (96)       9897     (98)
A                  =00000007        856        (3)          1394     (5)        1401     (5)        1409     (5)        1414     (5)        1423     (5)
                                                            2271     (7)        2277     (7)        2280     (7)        2283     (7)        2288     (7)
                                                            2294     (7)        2303     (7)        2309     (7)        2318     (7)        2339     (7)
                                                            2380     (7)        2383     (7)        2387     (7)        2467     (8)        2570     (10)
                                                            2572     (10)       2628     (11)       2706     (12)       2805     (14)       2821     (14)
                                                            2833     (14)       2892     (15)       2897     (15)       2899     (15)       2941     (16)
                                                            2946     (16)       2948     (16)       2949     (16)       2951     (16)       3000     (17)
                                                            3042     (18)       3050     (18)       3061     (18)       3064     (18)       3067     (18)
                                                            3102     (19)       3109     (19)       3120     (19)       3123     (19)       3126     (19)
                                                            3207     (20)       3212     (20)       3217     (20)       3218     (20)       3223     (20)
```

```
                                  3228     (20)      3243     (20)      3249     (20)      3256     (20)      3260     (20)
                                  3262     (20)      3269     (20)      3278     (20)      3280     (20)      3283     (20)
                                  3292     (20)      3305     (20)      3307     (20)      3308     (20)      3313     (20)
                                  3317     (20)      3324     (20)      3333     (20)      3338     (20)      3349     (20)
                                  3350     (20)      3354     (20)      3361     (20)      3363     (20)      3364     (20)
                                  3366     (20)      3369     (20)      3370     (20)      3372     (20)      3373     (20)
                                  3382     (20)      3384     (20)      3400     (20)      3401     (20)      3417     (20)
                                  3419     (20)      3420     (20)      3422     (20)      3524     (22)      3526     (22)
                                  3528     (22)      3606     (23)      3619     (23)      3629     (23)      3642     (23)
                                  3645     (23)      3656     (23)      3678     (23)      3685     (23)      3686     (23)
                                  3688     (23)      3699     (23)      3700     (23)      3711     (23)      3718     (23)
                                  3780     (24)      3796     (24)      3799     (24)      3801     (24)      3802     (24)
                                  3805     (24)      3812     (24)      3831     (24)      3833     (24)      3842     (24)
                                  3844     (24)      3857     (24)      3862     (24)      3971     (27)      4009     (28)
                                  4011     (28)      4012     (28)      4040     (29)      4042     (29)      4043     (29)
                                  4067     (30)      4099     (31)      4119     (32)      4128     (32)      4156     (33)
                                  4157     (33)      4205     (34)      4219     (34)      4239     (34)      4247     (34)
                                  4250     (34)      4266     (34)      4269     (34)      4319     (35)      4321     (35)
                                  4337     (35)      4339     (35)      4344     (35)      4349     (35)      4356     (35)
                                  4433     (36)      4446     (36)      4514     (37)      4523     (37)      4617     (39)
                                  4628     (39)      4675     (39)      4690     (39)      4769     (40)      4772     (40)
                                  4775     (40)      4777     (40)      4782     (40)      4838     (40)      4839     (40)
                                  4843     (40)      4847     (40)      4848     (40)      4849     (40)      4851     (40)
                                  4853     (40)      4859     (40)      4862     (40)      4869     (40)      4880     (40)
                                  4989     (41)      4994     (41)      4999     (41)      5152     (45)      5170     (45)
                                  5174     (45)      5183     (45)      5196     (45)      5202     (45)      5205     (45)
                                  5206     (45)      5212     (45)      5215     (45)      5224     (45)      5241     (45)
                                  5243     (45)      5248     (45)      5251     (45)      5277     (45)      5285     (45)
                                  5289     (45)      5350     (46)      5375     (46)      5380     (46)      5384     (46)
                                  5393     (46)      5400     (46)      5402     (46)      5413     (46)      5417     (46)
                                  5480     (47)      5487     (47)      5521     (47)      5525     (47)      5532     (47)
                                  5536     (47)      5544     (47)      5550     (47)      5556     (47)      5562     (47)
                                  5568     (47)      5575     (47)      5576     (47)      5582     (47)      5586     (47)
                                  5589     (47)      5591     (47)      5592     (47)      5601     (47)      5602     (47)
                                  5617     (47)      5618     (47)      5627     (47)      5628     (47)      5716     (48)
                                  5810     (49)      5814     (49)      5828     (49)      5831     (49)      5832     (49)
                                  5836     (49)      5852     (49)      5853     (49)      5855     (49)      5856     (49)
                                  5917     (50)      5928     (50)      5930     (50)      6002     (51)      6021     (51)
                                  6041     (51)      6119     (52)      6174     (53)      6183     (53)      6191     (53)
                                  6199     (53)      6200     (53)      6204     (53)      6205     (53)      6217     (53)
                                  6220     (53)      6274     (54)      6276     (54)      6283     (54)      6286     (54)
                                  6314     (54)      6342     (54)      6343     (54)      6345     (54)      6426     (56)
                                  6448     (56)      6471     (56)      6477     (56)      6495     (56)      6506     (56)
                                  6563     (57)      6570     (57)      6576     (57)      6582     (57)      6626     (58)
                                  6628     (58)      6639     (58)      6640     (58)      6646     (58)      6647     (58)
                                  6662     (58)      6665     (58)      6780     (60)      6857     (61)      6917     (62)
                                  6919     (62)      6935     (62)      7007     (63)      7009     (63)      7038     (63)
                                  7052     (63)      7065     (63)      7124     (64)      7126     (64)      7150     (64)
                                  7152     (64)      7199     (64)      7200     (64)      7311     (65)      7314     (65)
                                  7448     (66)      7451     (66)      7455     (66)      7461     (66)      7464     (66)
                                  7472     (66)      7475     (66)      7495     (66)      7497     (66)      7502     (66)
                                  7504     (66)      7505     (66)      7507     (66)      7525     (66)      7527     (66)
                                  7529     (66)      7536     (66)      7539     (66)      7551     (66)      7575     (66)
                                  7577     (66)      7611     (67)      7622     (67)      7623     (67)      7682     (67)
                                  7691     (67)      7693     (67)      7711     (67)      7714     (67)      7733     (67)
```

```
                                             7735   (67)     7737   (67)     7749   (67)     7752   (67)     7754   (67)
                                             7756   (67)     7764   (67)     7771   (67)     7773   (67)     7781   (67)
                                             7783   (67)     7791   (67)     7824   (68)     7859   (69)     7922   (70)
                                             7926   (70)     7951   (70)     7954   (70)     7956   (70)     7971   (70)
                                             7974   (70)     7976   (70)     8034   (71)     8042   (71)     8075   (71)
                                             8079   (71)     8242   (76)     8243   (76)     8244   (76)     8245   (76)
                                             8249   (76)     8254   (76)     8271   (76)     8275   (76)     8281   (76)
                                             8287   (76)     8297   (76)     8298   (76)     8336   (76)     8339   (76)
                                             8342   (76)     8345   (76)     8347   (76)     8348   (76)     8349   (76)
                                             8420   (78)     8424   (78)     8426   (78)     8447   (79)     8448   (79)
                                             8449   (79)     8457   (79)     8498   (80)     8502   (80)     8504   (80)
                                             8552   (81)     8558   (81)     8576   (81)     8579   (81)     8628   (82)
                                             8670   (83)     8680   (83)     8723   (84)     8733   (84)     8781   (85)
                                             8784   (85)     8785   (85)     8836   (86)     8840   (86)     8850   (86)
                                             8930   (87)     8935   (87)     8941   (87)     8943   (87)     8945   (87)
                                             8952   (87)     8954   (87)     8966   (87)     8976   (87)     8986   (87)
                                             8999   (87)     9000   (87)     9003   (87)     9018   (87)     9021   (87)
                                             9037   (87)     9062   (87)     9063   (87)     9070   (87)     9071   (87)
                                             9078   (87)     9082   (87)     9094   (87)     9150   (88)     9222   (89)
                                             9227   (89)     9236   (89)     9238   (89)     9245   (89)     9247   (89)
                                             9253   (89)     9294   (89)     9308   (89)     9312   (89)     9313   (89)
                                             9315   (89)     9325   (89)     9329   (89)     9343   (89)     9351   (89)
                                             9354   (89)     9402   (90)     9404   (90)     9417   (90)     9419   (90)
                                             9420   (90)     9421   (90)     9422   (90)     9425   (90)     9437   (90)
                                             9439   (90)     9450   (90)     9452   (90)     9453   (90)     9454   (90)
                                             9455   (90)     9458   (90)     9482   (91)     9483   (91)     9520   (92)
                                             9521   (92)     9525   (92)     9529   (92)     9532   (92)     9533   (92)
                                             9535   (92)     9539   (92)     9636   (94)     9645   (94)     9650   (94)
                                             9659   (94)     9708   (95)     9789   (95)     9793   (95)     9794   (95)
                                             9796   (95)     9797   (95)     9799   (95)     9801   (95)     9839   (96)
                                             9843   (96)     9845   (96)     9846   (96)     9866   (97)     9896   (98)
                                             9900   (98)     9902   (98)     9903   (98)
ACV             =0000000F    863    (4)
AC_LOW          =00000080    863    (4)
ADD             =00000019    863    (4)
ADDR_MATCH_HI   =00007405    863    (4)      3527   (22)     3529   (22)     3536   (22)     5286   (45)     5381   (46)
ADDR_MATCH_LO   =00007404    863    (4)      3895   (25)     3918   (26)     3970   (27)     3984   (27)     5290   (45)
                                             5385   (46)     6790   (60)     863    (4)
ADD_NAME         00000963-R  2055   (6)      1993   (6)
ADK             =0000000B    863    (4)
ALK             =00000000    863    (4)
ALP             =00000016    863    (4)      9637   (94)     9705   (95)
ALP_NAME         00000713-R  2006   (6)      1990   (6)
ARG_LIST         000003BE-R  1803   (6)      3800   (24)     3812   (24)     3862   (24)     5827   (49)     5905   (50)
                                             5910   (50)     5919   (50)     5927   (50)     5929   (50)     6075   (51)
                                             6169   (53)     6171   (53)     6182   (53)     6184   (53)     6190   (53)
                                             6192   (53)     6266   (54)     6340   (54)     6421   (56)     6434   (56)
                                             6461   (56)     6470   (56)     6544   (57)     6546   (57)     6551   (57)
                                             6562   (57)     6564   (57)     6627   (58)     6653   (58)     6656   (58)
                                             6671   (58)     6678   (58)     6686   (58)     6730   (59)     6774   (60)
                                             6779   (60)     6824   (61)     6829   (61)     6834   (61)     6843   (61)
                                             6845   (61)     6847   (61)     6856   (61)     6909   (62)     6910   (62)
                                             6915   (62)     6921   (62)     6923   (62)     6928   (62)     6933   (62)
                                             6936   (62)     6999   (63)     7000   (63)     7005   (63)     7011   (63)
                                             7013   (63)     7018   (63)     7023   (63)     7025   (63)     7027   (63)
```

```
                                                7037    (63)      7051    (63)      7063    (63)      7066    (63)      7140    (64)
                                                7148    (64)      7297    (65)      7424    (66)      7454    (66)      7535    (66)
                                                7552    (66)      7715    (67)      7906    (70)      7908    (70)      7913    (70)
                                                7921    (70)      7938    (70)      7950    (70)      7958    (70)      7970    (70)
                                                7979    (70)      7991    (70)      7993    (70)      8030    (71)      8033    (71)
                                                8039    (71)      8044    (71)      8057    (71)      8069    (71)      8106    (72)
                                                8138    (73)      8140    (73)      8387    (77)      8416    (78)      8417    (78)
A_REG_BYTE_0     =00007410       863       (4)  863     (4)
A_REG_BYTE_1     =00007411       863       (4)
A_REG_BYTE_2     =00007412       863       (4)
A_REG_BYTE_3     =00007413       863       (4)
B                =00000000       856       (3)  1375    (5)       1384    (5)       1386    (5)       1387    (5)       1388    (5)
                                                1395    (5)       1396    (5)       1397    (5)       1398    (5)       1400    (5)
                                                1402    (5)       1404    (5)       1407    (5)       1408    (5)       1429    (5)
                                                2277    (7)       2280    (7)       2288    (7)       2294    (7)       2320    (7)
                                                2383    (7)       2385    (7)       2468    (8)       2469    (8)       2630    (11)
                                                2695    (12)      2703    (12)      2800    (14)      2805    (14)      2817    (14)
                                                2819    (14)      2820    (14)      2822    (14)      2824    (14)      2825    (14)
                                                2826    (14)      2827    (14)      2834    (14)      2835    (14)      2841    (14)
                                                2842    (14)      2848    (14)      2892    (15)      2900    (15)      2941    (16)
                                                2950    (16)      2951    (16)      3043    (18)      3050    (18)      3052    (18)
                                                3055    (18)      3058    (18)      3063    (18)      3103    (19)      3109    (19)
                                                3111    (19)      3114    (19)      3117    (19)      3122    (19)      3213    (20)
                                                3216    (20)      3223    (20)      3224    (20)      3225    (20)      3228    (20)
                                                3229    (20)      3231    (20)      3236    (20)      3238    (20)      3242    (20)
                                                3247    (20)      3248    (20)      3251    (20)      3254    (20)      3257    (20)
                                                3263    (20)      3270    (20)      3271    (20)      3272    (20)      3278    (20)
                                                3282    (20)      3284    (20)      3286    (20)      3287    (20)      3291    (20)
                                                3296    (20)      3297    (20)      3298    (20)      3308    (20)      3317    (20)
                                                3318    (20)      3322    (20)      3325    (20)      3326    (20)      3327    (20)
                                                3328    (20)      3333    (20)      3337    (20)      3339    (20)      3341    (20)
                                                3342    (20)      3348    (20)      3350    (20)      3353    (20)      3354    (20)
                                                3359    (20)      3367    (20)      3373    (20)      3374    (20)      3375    (20)
                                                3378    (20)      3380    (20)      3384    (20)      3386    (20)      3394    (20)
                                                3400    (20)      3404    (20)      3406    (20)      3412    (20)      3413    (20)
                                                3414    (20)      3415    (20)      3423    (20)      3425    (20)      3462    (21)
                                                3466    (21)      3467    (21)      3473    (21)      3475    (21)      3476    (21)
                                                3477    (21)      3478    (21)      3479    (21)      3481    (21)      3482    (21)
                                                3485    (21)      3520    (22)      3525    (22)      3530    (22)      3605    (23)
                                                3607    (23)      3608    (23)      3611    (23)      3612    (23)      3618    (23)
                                                3626    (23)      3631    (23)      3633    (23)      3641    (23)      3643    (23)
                                                3644    (23)      3647    (23)      3648    (23)      3654    (23)      3655    (23)
                                                3678    (23)      3681    (23)      3683    (23)      3684    (23)      3690    (23)
                                                3691    (23)      3693    (23)      3694    (23)      3706    (23)      3707    (23)
                                                3709    (23)      3710    (23)      3717    (23)      3720    (23)      3722    (23)
                                                3723    (23)      3801    (24)      3829    (24)      3834    (24)      3836    (24)
                                                3967    (27)      3985    (27)      4004    (28)      4005    (28)      4006    (28)
                                                4007    (28)      4012    (28)      4013    (28)      4014    (28)      4015    (28)
                                                4016    (28)      4035    (29)      4036    (29)      4037    (29)      4038    (29)
                                                4043    (29)      4044    (29)      4045    (29)      4046    (29)      4047    (29)
                                                4121    (32)      4125    (32)      4126    (32)      4127    (32)      4130    (32)
                                                4131    (32)      4132    (32)      4133    (32)      4134    (32)      4135    (32)
                                                4153    (33)      4155    (33)      4158    (33)      4160    (33)      4241    (34)
                                                4243    (34)      4246    (34)      4248    (34)      4249    (34)      4251    (34)
                                                4253    (34)      4281    (34)      4312    (35)      4313    (35)      4334    (35)
```

```
        4355   (35)      4375   (35)      4377   (35)      4432   (36)      4445   (36)
        4449   (36)      4451   (36)      4461   (36)      4462   (36)      4469   (36)
        4518   (37)      4526   (37)      4568   (38)      4571   (38)      4573   (38)
        4616   (39)      4618   (39)      4619   (39)      4620   (39)      4621   (39)
        4622   (39)      4623   (39)      4624   (39)      4633   (39)      4649   (39)
        4652   (39)      4653   (39)      4655   (39)      4656   (39)      4658   (39)
        4659   (39)      4660   (39)      4666   (39)      4672   (39)      4673   (39)
        4676   (39)      4678   (39)      4685   (39)      4686   (39)      4689   (39)
        4691   (39)      4692   (39)      4696   (39)      4777   (40)      4784   (40)
        4842   (40)      4845   (40)      4858   (40)      4861   (40)      4879   (40)
        4882   (40)      4948   (41)      4988   (41)      4990   (41)      4991   (41)
        4992   (41)      4995   (41)      4996   (41)      4997   (41)      4999   (41)
        5002   (41)      5003   (41)      5005   (41)      5006   (41)      5151   (45)
        5154   (45)      5169   (45)      5172   (45)      5173   (45)      5185   (45)
        5210   (45)      5211   (45)      5214   (45)      5217   (45)      5226   (45)
        5246   (45)      5247   (45)      5250   (45)      5283   (45)      5284   (45)
        5288   (45)      5348   (46)      5352   (46)      5378   (46)      5379   (46)
        5383   (46)      5395   (46)      5412   (46)      5415   (46)      5416   (46)
        5478   (47)      5484   (47)      5485   (47)      5489   (47)      5524   (47)
        5535   (47)      5576   (47)      5585   (47)      5592   (47)      5594   (47)
        5602   (47)      5610   (47)      5618   (47)      5620   (47)      5628   (47)
        5630   (47)      5637   (47)      5800   (49)      5801   (49)      5804   (49)
        5826   (49)      5832   (49)      5833   (49)      5834   (49)      5837   (49)
        5838   (49)      5839   (49)      5850   (49)      5851   (49)      5854   (49)
        5857   (49)      5858   (49)      5912   (50)      5920   (50)      5923   (50)
        5928   (50)      5930   (50)      6021   (51)      6023   (51)      6041   (51)
        6043   (51)      6172   (53)      6173   (53)      6175   (53)      6176   (53)
        6177   (53)      6183   (53)      6185   (53)      6191   (53)      6193   (53)
        6215   (53)      6218   (53)      6268   (54)      6269   (54)      6270   (54)
        6271   (54)      6273   (54)      6279   (54)      6280   (54)      6281   (54)
        6282   (54)      6284   (54)      6285   (54)      6287   (54)      6291   (54)
        6292   (54)      6293   (54)      6294   (54)      6295   (54)      6297   (54)
        6299   (54)      6303   (54)      6304   (54)      6305   (54)      6306   (54)
        6307   (54)      6309   (54)      6313   (54)      6319   (54)      6325   (54)
        6330   (54)      6331   (54)      6333   (54)      6334   (54)      6344   (54)
        6429   (56)      6438   (56)      6441   (56)      6442   (56)      6446   (56)
        6447   (56)      6449   (56)      6450   (56)      6451   (56)      6453   (56)
        6454   (56)      6455   (56)      6456   (56)      6460   (56)      6464   (56)
        6465   (56)      6476   (56)      6478   (56)      6479   (56)      6480   (56)
        6481   (56)      6492   (56)      6493   (56)      6497   (56)      6498   (56)
        6499   (56)      6500   (56)      6502   (56)      6504   (56)      6507   (56)
        6552   (57)      6555   (57)      6563   (57)      6565   (57)      6569   (57)
        6577   (57)      6580   (57)      6581   (57)      6583   (57)      6584   (57)
        6626   (58)      6639   (58)      6646   (58)      6655   (58)      6661   (58)
        6666   (58)      6687   (58)      6835   (61)      6838   (61)      6839   (61)
        6848   (61)      6851   (61)      6855   (61)      6857   (61)      6858   (61)
        6860   (61)      6861   (61)      6862   (61)      6863   (61)      6919   (62)
        6924   (62)      6927   (62)      6935   (62)      7009   (63)      7014   (63)
        7017   (63)      7028   (63)      7031   (63)      7035   (63)      7038   (63)
        7039   (63)      7041   (63)      7042   (63)      7043   (63)      7044   (63)
        7049   (63)      7052   (63)      7053   (63)      7055   (63)      7056   (63)
        7057   (63)      7058   (63)      7065   (63)      7143   (64)      7147   (64)
        7152   (64)      7154   (64)      7164   (64)      7165   (64)      7170   (64)
        7171   (64)      7177   (64)      7178   (64)      7197   (64)      7199   (64)
        7311   (65)      7314   (65)      7324   (65)      7448   (66)      7451   (66)
```

```
                                          7497    (66)       7504    (66)       7505    (66)       7539    (66)       7551    (66)
                                          7553    (66)       7622    (67)       7627    (67)       7714    (67)       7716    (67)
                                          7743    (67)       7746    (67)       7747    (67)       7767    (67)       7768    (67)
                                          7769    (67)       7776    (67)       7777    (67)       7778    (67)       7786    (67)
                                          7787    (67)       7788    (67)       7824    (68)       7826    (68)       7861    (69)
                                          7914    (70)       7917    (70)       7928    (70)       7929    (70)       7930    (70)
                                          7937    (70)       7941    (70)       7949    (70)       7957    (70)       7960    (70)
                                          7961    (70)       7964    (70)       7966    (70)       7977    (70)       7978    (70)
                                          7981    (70)       7982    (70)       7985    (70)       7987    (70)       7989    (70)
                                          7990    (70)       7992    (70)       8032    (71)       8045    (71)       8048    (71)
                                          8050    (71)       8060    (71)       8062    (71)       8068    (71)       8074    (71)
                                          8076    (71)       8077    (71)       8078    (71)       8108    (72)       8110    (72)
                                          8111    (72)       8143    (73)       8145    (73)       8146    (73)       8244    (76)
                                          8248    (76)       8271    (76)       8275    (76)       8287    (76)       8297    (76)
                                          8303    (76)       8308    (76)       8310    (76)       8313    (76)       8314    (76)
                                          8316    (76)       8317    (76)       8320    (76)       8321    (76)       8322    (76)
                                          8325    (76)       8326    (76)       8327    (76)       8328    (76)       8334    (76)
                                          8337    (76)       8340    (76)       8343    (76)       8390    (77)       8454    (79)
                                          8456    (79)       8500    (80)       8503    (80)       8505    (80)       8550    (81)
                                          8551    (81)       8560    (81)       8565    (81)       8573    (81)       8574    (81)
                                          8575    (81)       8576    (81)       8630    (82)       8670    (83)       8675    (83)
                                          8679    (83)       8681    (83)       8682    (83)       8683    (83)       8684    (83)
                                          8723    (84)       8728    (84)       8732    (84)       8734    (84)       8735    (84)
                                          8736    (84)       8737    (84)       8784    (85)       8787    (85)       8790    (85)
                                          8791    (85)       8802    (85)       8805    (85)       8806    (85)       8915    (87)
                                          8924    (87)       8927    (87)       8928    (87)       8969    (87)       8970    (87)
                                          8971    (87)       8979    (87)       8980    (87)       8981    (87)       8989    (87)
                                          8990    (87)       8991    (87)       8998    (87)       9002    (87)       9005    (87)
                                          9016    (87)       9020    (87)       9022    (87)       9024    (87)       9034    (87)
                                          9037    (87)       9038    (87)       9040    (87)       9041    (87)       9044    (87)
                                          9045    (87)       9047    (87)       9048    (87)       9051    (87)       9052    (87)
                                          9053    (87)       9054    (87)       9056    (87)       9061    (87)       9069    (87)
                                          9073    (87)       9074    (87)       9075    (87)       9077    (87)       9079    (87)
                                          9083    (87)       9084    (87)       9086    (87)       9092    (87)       9093    (87)
                                          9095    (87)       9096    (87)       9098    (87)       9144    (88)       9145    (88)
                                          9148    (88)       9212    (89)       9213    (89)       9215    (89)       9218    (89)
                                          9219    (89)       9220    (89)       9221    (89)       9225    (89)       9226    (89)
                                          9228    (89)       9229    (89)       9230    (89)       9235    (89)       9237    (89)
                                          9247    (89)       9250    (89)       9252    (89)       9273    (89)       9281    (89)
                                          9282    (89)       9283    (89)       9285    (89)       9301    (89)       9302    (89)
                                          9303    (89)       9307    (89)       9308    (89)       9309    (89)       9310    (89)
                                          9311    (89)       9312    (89)       9315    (89)       9316    (89)       9318    (89)
                                          9319    (89)       9320    (89)       9327    (89)       9335    (89)       9336    (89)
                                          9342    (89)       9344    (89)       9345    (89)       9346    (89)       9349    (89)
                                          9350    (89)       9392    (90)       9397    (90)       9399    (90)       9402    (90)
                                          9404    (90)       9405    (90)       9406    (90)       9412    (90)       9413    (90)
                                          9416    (90)       9420    (90)       9429    (90)       9430    (90)       9432    (90)
                                          9434    (90)       9437    (90)       9439    (90)       9440    (90)       9441    (90)
                                          9447    (90)       9448    (90)       9449    (90)       9453    (90)       9481    (91)
                                          9482    (91)       9484    (91)       9488    (91)       9536    (92)       9584    (93)
                                          9625    (94)       9626    (94)       9627    (94)       9630    (94)       9631    (94)
                                          9632    (94)       9634    (94)       9635    (94)       9641    (94)       9644    (94)
                                          9651    (94)       9653    (94)       9657    (94)       9658    (94)       9660    (94)
                                          9661    (94)       9663    (94)       9664    (94)       9666    (94)       9667    (94)
                                          9710    (95)       9715    (95)       9719    (95)       9720    (95)       9724    (95)
```

```
                                             9725   (95)     9727   (95)     9729   (95)     9731   (95)     9732   (95)
                                             9733   (95)     9734   (95)     9735   (95)     9737   (95)     9738   (95)
                                             9739   (95)     9740   (95)     9741   (95)     9743   (95)     9744   (95)
                                             9745   (95)     9746   (95)     9747   (95)     9749   (95)     9750   (95)
                                             9751   (95)     9752   (95)     9753   (95)     9754   (95)     9762   (95)
                                             9763   (95)     9766   (95)     9767   (95)     9769   (95)     9776   (95)
                                             9777   (95)     9778   (95)     9779   (95)     9780   (95)     9782   (95)
                                             9787   (95)     9788   (95)     9798   (95)     9800   (95)     9802   (95)
                                             9803   (95)     9805   (95)     9806   (95)     9808   (95)     9809   (95)
                                             9837   (96)     9840   (96)     9846   (96)     9863   (97)     9865   (97)
                                             9894   (98)     9897   (98)     9903   (98)
BADD1          =00007426     863    (4)
BAD_DATA        00000443-R   1870   (6)      7050   (63)     7134   (64)     7135   (64)     8671   (83)     8677   (83)
                                             8678   (83)     8783   (85)     8958   (87)     9049   (87)     9524   (92)
                                             9537   (92)     9540   (92)     9727   (95)
BEGIN_LOOP      00001CE0-R   6168   (53)     1808   (6)
BELL            00000592-R   1930   (6)      8845   (86)
BELL_COUNT      00000591-R   1928   (6)      4206   (34)     8835   (86)     8841   (86)     8849   (86)     8851   (86)
BELL_FLAG      =00000008     863    (4)      4755   (40)     7292   (65)
BELL_KEY       =0000000E     863    (4)
BIT_COUNT       00000C9B-R   2183   (6)      9239   (89)     9328   (89)     9330   (89)
BIT_NMB_TO_MSK  00002A6E-R   9480   (91)     5175   (45)     5418   (46)     5712   (48)     9274   (89)     9293   (89)
BIT_SLICE       00002B53-R   9701   (95)     9642   (94)
BIT_SLICE_TBL   000006F5-R   1990   (6)      9706   (95)
BIT_SLICE_TEMP  000002C4-R   1685   (6)      9726   (95)
BREAK_ACTION    000015A1-R   4279   (34)     1705   (6)
BUFF_ADDR_LOW  =00007426     863    (4)      863    (4)
BURST_CLOCK     000020B5-R   7423   (66)     1805   (6)      7587   (66)
BURST_STOP_FLAG=00000080     863    (4)      7433   (66)     7479   (66)     7616   (67)
BURST_TEMP      00000451-R   1889   (6)      7496   (66)     7503   (66)     7507   (66)     7528   (66)     7529   (66)
                                             7604   (67)     7647   (67)     7692   (67)     7693   (67)     7734   (67)
                                             7735   (67)
BYTE           =00000001     863    (4)
C              =00000001     856    (3)      1399   (5)      2467   (8)      2818   (14)     2823   (14)     2828   (14)
                                             2836   (14)     3255   (20)     3268   (20)     3273   (20)     3476   (21)
                                             3824   (24)     3832   (24)     3833   (24)     4154   (33)     4156   (33)
                                             4159   (33)     4776   (40)     5150   (45)     5191   (45)     5196   (45)
                                             5349   (46)     5400   (46)     5402   (46)     5404   (46)     5486   (47)
                                             5716   (48)     5825   (49)     5913   (50)     6505   (56)     6576   (57)
                                             6578   (57)     6663   (58)     6917   (62)     7007   (63)     7150   (64)
                                             7159   (64)     7160   (64)     7611   (67)     7623   (67)     7925   (70)
                                             7931   (70)     7988   (70)     8109   (72)     8144   (73)     8242   (76)
                                             8499   (80)     8551   (81)     8572   (81)     8579   (81)     8781   (85)
                                             8792   (85)     8807   (85)     9082   (87)     9236   (89)     9294   (89)
                                             9314   (89)     9338   (89)     9347   (89)     9398   (90)     9407   (90)
                                             9419   (90)     9422   (90)     9433   (90)     9442   (90)     9452   (90)
                                             9455   (90)     9521   (92)     9525   (92)     9539   (92)     9541   (92)
                                             9645   (94)     9649   (94)     9650   (94)     9765   (95)     9771   (95)
                                             9799   (95)     9838   (96)     9895   (98)
CACHE_ENA_ADR   00000275-R   1658   (6)      3305   (20)
CACHE_ENA_DATA  00000280-R   1661   (6)      3307   (20)
CACHE_WRITE     0000028B-R   1663   (6)      3313   (20)
CAK            =0000000A     863    (4)
CALC_PARITY     00002A1C-R   9391   (90)     9356   (89)
CCC            =00000004     863    (4)
```

```
CCF_SUB_ACT     000019AB-R   5411  (46)      5358  (46)
CCS            =00000005     863   (4)
CCS_NAME        000000DC-R   1490  (6)       1475  (6)
CFLAG_SUB_ACT   0000198F-R   5392  (46)      5356  (46)      5406  (46)
CF_KEY         =00000024     863   (4)       5163  (45)      5357  (46)
CHECK_QA_FLAG   00002AAD-R   9570  (93)      6952  (62)      7082  (63)      7187  (64)      8363  (76)
CHECK_QV        000011D4-R   3461  (21)      3226  (20)      3376  (20)
CHECK_VBUS      00002A7C-R   9518  (92)      7175  (64)      9046  (87)
CHR            =00000005     1540  (6)       1526  (6)       1527  (6)       1528  (6)       1529  (6)       1530  (6)
                                             1531  (6)       1532  (6)       1533  (6)       1534  (6)       1535  (6)
                                             1536  (6)       1537  (6)       1538  (6)       1539  (6)       1540  (6)
CLA            =00000003     863   (4)
CLEAR_ACTION    00001945-R   5346  (46)      1701  (6)
CLEAR_CTRL_FILE=00000002     863   (4)       3958  (27)      3960  (27)      4971  (41)      4973  (41)      7706  (67)
                                             7708  (67)
CLEAR_DCS_CF    00000FD7-R   3100  (19)      5420  (46)      7466  (66)      7477  (66)
CLEAR_DONE      000019BC-R   5424  (46)      5377  (46)      5387  (46)      5397  (46)
CLEAR_ERR       000019BF-R   5431  (46)      5354  (46)      5360  (46)      5399  (46)
CLKDCS         =00007427     863   (4)
CLK_CTRL_0     =00000020     863   (4)       1414  (5)       2945  (16)      2947  (16)      4072  (30)      7597  (67)
CLK_CTRL_0_RO  =00000040     863   (4)       7430  (66)      7494  (66)      7613  (67)      7625  (67)      7690  (67)
                                             7713  (67)      7732  (67)
CLK_CTRL_1     =00000040     863   (4)       1414  (5)       2896  (15)      2898  (15)      4072  (30)      7597  (67)
CLK_CTRL_1_RO  =00000080     863   (4)       7430  (66)      7494  (66)      7613  (67)      7625  (67)      7690  (67)
                                             7713  (67)      7732  (67)
CLK_MSG_ADDR    00000453-R   1890  (6)       7427  (66)      7630  (67)      7727  (67)
CLK_STOP_MSG1   00000481-R   1894  (6)       7426  (66)
CLK_STOP_MSG2   000004A3-R   1895  (6)       7736  (67)
CLK_STOP_MSG3   000004B4-R   1896  (6)       7739  (67)
CLK_STOP_MSG4   000004BA-R   1897  (6)       7523  (66)      7750  (67)
CLK_STOP_MSG5   000004CC-R   1898  (6)       7629  (67)
CLOCK_STOPED   =00000080     863   (4)
CMD_DISPATCH    00000304-R   1696  (6)       4241  (34)
CMD_ENTRY       0000000D-R   1380  (5)       4280  (34)
CMD_LINE_LENGTH 000002F1-R   1693  (6)       4212  (34)
CMI            =00000007     863   (4)
CMICTL         =0000741C     863   (4)
CMI_COMPLETE   =00000020     863   (4)
CMI_CTRL_CLEAR =00000001     863   (4)
CMI_CTRL_REG   =0000741C     863   (4)       863   (4)
CMI_GO         =00000008     863   (4)
CMI_NAME        000000F2-R   1492  (6)       1477  (6)
CMI_READ       =00000040     863   (4)
CMI_STATUS_0   =00000008     863   (4)
CMI_STATUS_1   =00000010     863   (4)
CMI_WRITE      =00000020     863   (4)
CML            =0000000E     863   (4)
CMP_GOOD_BAD    00002719-R   8780  (85)      6937  (62)      7067  (63)
COMMAND_LIST    0000032E-R   1713  (6)       4313  (35)
COMMAND_PARSE   000015A8-R   4308  (35)      4229  (34)
COMMA_MSG       0000062A-R   1952  (6)       3863  (24)      4129  (32)      5004  (41)      7686  (67)      7780  (67)
                                             7790  (67)
COMPARE_REG     00001F07-R   6908  (62)      1806  (6)
COMPARE_REG_MSK 00001F5A-R   6998  (63)      1807  (6)
COMPARE_VBUS    00001FE4-R   7123  (64)      1820  (6)
```

```
CON            =00000011     863    (4)
CONTINUE_ACTION 0000183A-R   5055   (42)     1698   (6)
CONTINUE_FLAG  =00000002     863    (4)      4203   (34)     4267   (34)     5057   (42)     5103   (44)
CONTROL_C_FLAG =00000001     863    (4)      2750   (13)     4203   (34)     5997   (51)     7280   (65)     7605   (67)
                                             7608   (67)     7640   (67)
CONT_FLAG      =00000020     863    (4)      4761   (40)     4830   (40)     6049   (51)     6063   (51)
CONT_KEY       =0000001C     863    (4)      4794   (40)
CONT_SUB_ACTION 0000175A-R   4828   (40)     4795   (40)
CPU_INIT        00001459-R   4065   (30)     2997   (17)     6111   (52)     7828   (68)
CPU_RUN        =00000010     863    (4)
CRLF            00000CEE-R   2212   (6)      2269   (7)      2362   (7)      2378   (7)      2381   (7)      2568   (10)
                                             3650   (23)     3817   (24)     3856   (24)     4137   (32)     4209   (34)
                                             4217   (34)     7490   (66)     7680   (67)     7726   (67)     7738   (67)
                                             8260   (76)     8262   (76)     8442   (79)     8460   (79)     8916   (87)
                                             8946   (87)     8947   (87)     8957   (87)     9043   (87)     9090   (87)
                                             9099   (87)     9139   (88)     9628   (94)     9655   (94)     9665   (94)
                                             9867   (97)
CSAMR          =00007404     863    (4)
CSBUF          =00007424     863    (4)
CSTRP          =00007422     863    (4)
CS_ADD_BUFF_HGH=00007425     863    (4)
CS_ADD_BUFF_LOW=00007424     863    (4)      863    (4)
CS_ADD_TRAP_HGH=00007423     863    (4)      7514   (66)     7652   (67)
CS_ADD_TRAP_LOW=00007422     863    (4)      7524   (66)     7751   (67)     863    (4)
CURRENT_BIT     00000C9A-R   2182   (6)      9211   (89)     9217   (89)     9223   (89)     9286   (89)     9324   (89)
                                             9326   (89)
CURRENT_PC      000003B8-R   1798   (6)      5105   (44)     5824   (49)     7320   (65)     7742   (67)     8923   (87)
CURRENT_REC_NMB 0000013B-R   1512   (6)      2302   (7)      2331   (7)      2465   (8)      2470   (8)      2518   (9)
                                             3474   (21)     3480   (21)     3604   (23)     3724   (23)
CYCLE_FLAG     =00000008     863    (4)      5229   (45)     5255   (45)     7485   (66)     7573   (66)     7584   (66)
CYCLE_KEY      =00000020     863    (4)      5230   (45)
C_INIT_TEST_PC  000001B7-R   1607   (6)      3368   (20)     3424   (20)     3835   (24)     3837   (24)     5799   (49)
                                             7740   (67)     8921   (87)     9143   (88)
D              =00000002     856    (3)      1375   (5)      1384   (5)      1386   (5)      1387   (5)      1388   (5)
                                             1395   (5)      1396   (5)      1397   (5)      1400   (5)      1402   (5)
                                             1404   (5)      1406   (5)      1407   (5)      1408   (5)      1429   (5)
                                             2315   (7)      2320   (7)      2630   (11)     2695   (12)     2703   (12)
                                             2800   (14)     2807   (14)     2822   (14)     2826   (14)     2834   (14)
                                             2835   (14)     2841   (14)     2848   (14)     3043   (18)     3055   (18)
                                             3058   (18)     3063   (18)     3103   (19)     3114   (19)     3117   (19)
                                             3122   (19)     3213   (20)     3216   (20)     3224   (20)     3225   (20)
                                             3229   (20)     3231   (20)     3236   (20)     3238   (20)     3242   (20)
                                             3247   (20)     3248   (20)     3251   (20)     3252   (20)     3257   (20)
                                             3263   (20)     3267   (20)     3270   (20)     3271   (20)     3272   (20)
                                             3278   (20)     3282   (20)     3284   (20)     3286   (20)     3287   (20)
                                             3291   (20)     3296   (20)     3297   (20)     3298   (20)     3308   (20)
                                             3317   (20)     3318   (20)     3323   (20)     3325   (20)     3326   (20)
                                             3327   (20)     3333   (20)     3337   (20)     3339   (20)     3341   (20)
                                             3342   (20)     3348   (20)     3353   (20)     3359   (20)     3367   (20)
                                             3374   (20)     3375   (20)     3380   (20)     3394   (20)     3406   (20)
                                             3412   (20)     3413   (20)     3414   (20)     3415   (20)     3423   (20)
                                             3425   (20)     3462   (21)     3466   (21)     3467   (21)     3473   (21)
                                             3481   (21)     3482   (21)     3485   (21)     3525   (22)     3605   (23)
                                             3607   (23)     3608   (23)     3611   (23)     3612   (23)     3618   (23)
                                             3626   (23)     3630   (23)     3631   (23)     3633   (23)     3634   (23)
```

```
                                         3641    (23)    3643    (23)    3644    (23)    3647    (23)    3648    (23)
                                         3654    (23)    3655    (23)    3671    (23)    3680    (23)    3683    (23)
                                         3684    (23)    3689    (23)    3690    (23)    3691    (23)    3693    (23)
                                         3694    (23)    3706    (23)    3707    (23)    3709    (23)    3710    (23)
                                         3717    (23)    3720    (23)    3722    (23)    3723    (23)    3829    (24)
                                         3967    (27)    3985    (27)    4005    (28)    4006    (28)    4007    (28)
                                         4012    (28)    4013    (28)    4014    (28)    4015    (28)    4036    (29)
                                         4037    (29)    4038    (29)    4043    (29)    4044    (29)    4045    (29)
                                         4046    (29)    4126    (32)    4127    (32)    4130    (32)    4131    (32)
                                         4133    (32)    4134    (32)    4246    (34)    4248    (34)    4249    (34)
                                         4254    (34)    4281    (34)    4312    (35)    4319    (35)    4321    (35)
                                         4328    (35)    4337    (35)    4339    (35)    4355    (35)    4375    (35)
                                         4377    (35)    4430    (36)    4443    (36)    4444    (36)    4451    (36)
                                         4462    (36)    4469    (36)    4518    (37)    4526    (37)    4568    (38)
                                         4570    (38)    4571    (38)    4573    (38)    4633    (39)    4649    (39)
                                         4651    (39)    4653    (39)    4655    (39)    4658    (39)    4661    (39)
                                         4666    (39)    4675    (39)    4676    (39)    4677    (39)    4686    (39)
                                         4688    (39)    4691    (39)    4693    (39)    4696    (39)    4775    (40)
                                         4784    (40)    4838    (40)    4841    (40)    4845    (40)    4861    (40)
                                         4882    (40)    4948    (41)    4990    (41)    4991    (41)    4996    (41)
                                         5002    (41)    5006    (41)    5154    (45)    5172    (45)    5185    (45)
                                         5210    (45)    5211    (45)    5214    (45)    5217    (45)    5226    (45)
                                         5229    (45)    5232    (45)    5235    (45)    5246    (45)    5247    (45)
                                         5250    (45)    5256    (45)    5283    (45)    5284    (45)    5288    (45)
                                         5352    (46)    5378    (46)    5379    (46)    5383    (46)    5395    (46)
                                         5415    (46)    5478    (47)    5484    (47)    5489    (47)    5576    (47)
                                         5592    (47)    5602    (47)    5618    (47)    5628    (47)    5800    (49)
                                         5801    (49)    5804    (49)    5811    (49)    5826    (49)    5827    (49)
                                         5833    (49)    5837    (49)    5838    (49)    5839    (49)    5854    (49)
                                         5856    (49)    5912    (50)    5920    (50)    5923    (50)    6171    (53)
                                         6173    (53)    6175    (53)    6176    (53)    6215    (53)    6218    (53)
                                         6268    (54)    6269    (54)    6270    (54)    6271    (54)    6273    (54)
                                         6280    (54)    6281    (54)    6282    (54)    6284    (54)    6285    (54)
                                         6287    (54)    6291    (54)    6292    (54)    6293    (54)    6294    (54)
                                         6295    (54)    6297    (54)    6299    (54)    6303    (54)    6304    (54)
                                         6305    (54)    6306    (54)    6307    (54)    6309    (54)    6313    (54)
                                         6319    (54)    6325    (54)    6330    (54)    6331    (54)    6333    (54)
                                         6334    (54)    6344    (54)    6429    (56)    6438    (56)    6441    (56)
                                         6442    (56)    6447    (56)    6449    (56)    6450    (56)    6453    (56)
                                         6454    (56)    6455    (56)    6456    (56)    6460    (56)    6464    (56)
                                         6465    (56)    6478    (56)    6479    (56)    6480    (56)    6493    (56)
                                         6494    (56)    6497    (56)    6498    (56)    6499    (56)    6502    (56)
                                         6507    (56)    6552    (57)    6555    (57)    6565    (57)    6569    (57)
                                         6577    (57)    6580    (57)    6581    (57)    6583    (57)    6661    (58)
                                         6665    (58)    6687    (58)    6835    (61)    6838    (61)    6839    (61)
                                         6848    (61)    6851    (61)    6855    (61)    6858    (61)    6860    (61)
                                         6861    (61)    6862    (61)    6924    (62)    6927    (62)    7014    (63)
                                         7017    (63)    7028    (63)    7031    (63)    7035    (63)    7036    (63)
                                         7039    (63)    7041    (63)    7042    (63)    7043    (63)    7049    (63)
                                         7050    (63)    7053    (63)    7055    (63)    7056    (63)    7057    (63)
                                         7143    (64)    7147    (64)    7154    (64)    7163    (64)    7164    (64)
                                         7165    (64)    7171    (64)    7177    (64)    7197    (64)    7324    (65)
                                         7743    (67)    7746    (67)    7747    (67)    7768    (67)    7769    (67)
                                         7777    (67)    7778    (67)    7787    (67)    7788    (67)    7914    (70)
                                         7917    (70)    7924    (70)    7928    (70)    7929    (70)    7930    (70)
```

```
                                              7937   (70)     7941   (70)     7949   (70)     7957   (70)     7960   (70)
                                              7964   (70)     7978   (70)     7981   (70)     7985   (70)     7989   (70)
                                              7990   (70)     7992   (70)     8032   (71)     8045   (71)     8048   (71)
                                              8050   (71)     8060   (71)     8062   (71)     8068   (71)     8069   (71)
                                              8074   (71)     8076   (71)     8077   (71)     8078   (71)     8110   (72)
                                              8143   (73)     8145   (73)     8248   (76)     8269   (76)     8274   (76)
                                              8303   (76)     8308   (76)     8310   (76)     8313   (76)     8314   (76)
                                              8316   (76)     8317   (76)     8320   (76)     8321   (76)     8322   (76)
                                              8325   (76)     8326   (76)     8327   (76)     8334   (76)     8337   (76)
                                              8340   (76)     8343   (76)     8390   (77)     8454   (79)     8456   (79)
                                              8501   (80)     8503   (80)     8506   (80)     8559   (81)     8560   (81)
                                              8565   (81)     8566   (81)     8573   (81)     8575   (81)     8671   (83)
                                              8675   (83)     8679   (83)     8681   (83)     8682   (83)     8683   (83)
                                              8724   (84)     8728   (84)     8732   (84)     8734   (84)     8735   (84)
                                              8736   (84)     8782   (85)     8787   (85)     8790   (85)     8791   (85)
                                              8802   (85)     8805   (85)     8806   (85)     8915   (87)     8924   (87)
                                              8927   (87)     8928   (87)     8970   (87)     8971   (87)     8980   (87)
                                              8981   (87)     8990   (87)     8991   (87)     8998   (87)     9002   (87)
                                              9005   (87)     9016   (87)     9020   (87)     9022   (87)     9024   (87)
                                              9034   (87)     9040   (87)     9045   (87)     9047   (87)     9052   (87)
                                              9053   (87)     9056   (87)     9061   (87)     9069   (87)     9073   (87)
                                              9074   (87)     9075   (87)     9077   (87)     9079   (87)     9086   (87)
                                              9087   (87)     9092   (87)     9093   (87)     9095   (87)     9096   (87)
                                              9098   (87)     9144   (88)     9145   (88)     9148   (88)     9212   (89)
                                              9213   (89)     9215   (89)     9218   (89)     9219   (89)     9220   (89)
                                              9226   (89)     9228   (89)     9229   (89)     9251   (89)     9252   (89)
                                              9259   (89)     9273   (89)     9281   (89)     9282   (89)     9283   (89)
                                              9285   (89)     9287   (89)     9301   (89)     9302   (89)     9303   (89)
                                              9309   (89)     9320   (89)     9335   (89)     9336   (89)     9342   (89)
                                              9344   (89)     9345   (89)     9346   (89)     9349   (89)     9350   (89)
                                              9392   (90)     9396   (90)     9399   (90)     9405   (90)     9406   (90)
                                              9412   (90)     9413   (90)     9415   (90)     9416   (90)     9429   (90)
                                              9430   (90)     9431   (90)     9434   (90)     9440   (90)     9441   (90)
                                              9447   (90)     9448   (90)     9449   (90)     9519   (92)     9534   (92)
                                              9535   (92)     9536   (92)     9584   (93)     9625   (94)     9626   (94)
                                              9627   (94)     9630   (94)     9631   (94)     9632   (94)     9634   (94)
                                              9635   (94)     9641   (94)     9644   (94)     9657   (94)     9658   (94)
                                              9660   (94)     9661   (94)     9663   (94)     9664   (94)     9666   (94)
                                              9667   (94)     9709   (95)     9710   (95)     9715   (95)     9716   (95)
                                              9720   (95)     9725   (95)     9726   (95)     9731   (95)     9733   (95)
                                              9734   (95)     9737   (95)     9739   (95)     9740   (95)     9743   (95)
                                              9745   (95)     9746   (95)     9749   (95)     9750   (95)     9752   (95)
                                              9753   (95)     9754   (95)     9763   (95)     9766   (95)     9767   (95)
                                              9769   (95)     9776   (95)     9777   (95)     9778   (95)     9779   (95)
                                              9787   (95)     9788   (95)     9789   (95)     9797   (95)     9802   (95)
                                              9805   (95)     9806   (95)     9808   (95)     9809   (95)     9835   (96)
                                              9840   (96)     9846   (96)     9863   (97)     9865   (97)     9892   (98)
                                              9897   (98)     9903   (98)
DATA_MSG        000005B7-R    1943   (6)      8951   (87)
DATA_TYPE       00000447-R    1871   (6)      7125   (64)     8669   (83)     8722   (84)
DCSAD          =00007401      863    (4)
DCSCF          =00007403      863    (4)
DCSCR          =00007402      863    (4)
DCSDA          =00007400      863    (4)
DCS_ADDR_CLEAR =00000002      863    (4)      7461   (66)     7475   (66)     7499   (66)     7684   (67)
```

```
DCS_ADDR_MSG    00000456-R  1892   (6)      7491   (66)     7687   (67)
DCS_ADDR_REG_RO=00007427    863    (4)      2799   (14)     3041   (18)     3101   (19)     7428   (66)     7492   (66)
                                            7610   (67)     7688   (67)     7710   (67)     7729   (67)     863    (4)
DCS_ADDR_REG_WO=00007401    863    (4)      2803   (14)     2849   (14)     3046   (18)     3047   (18)     3056   (18)
                                            3057   (18)     3065   (18)     3106   (19)     3107   (19)     3115   (19)
                                            3116   (19)     3124   (19)     3953   (27)     863    (4)
DCS_BIT_MAP     00000C9E-R  2197   (6)      9249   (89)
DCS_CTRL_FILE  =00007403    863    (4)      3051   (18)     3110   (19)     7498   (66)     7673   (67)     7683   (67)
                                            863    (4)
DCS_CTRL_REG   =00007402    863    (4)      1415   (5)      2893   (15)     2942   (16)     3002   (17)     3893   (25)
                                            3916   (26)     3959   (27)     3961   (27)     3978   (27)     4073   (30)
                                            4104   (31)     4972   (41)     4974   (41)     5273   (45)     5371   (46)
                                            6784   (60)     7598   (67)     7635   (67)     7707   (67)     7709   (67)
                                            863    (4)
DCS_CTRL_RE_CPY 0000005C-R  1446   (6)      1416   (5)      2895   (15)     2944   (16)     3890   (25)     3915   (26)
                                            3955   (27)     3962   (27)     3975   (27)     3979   (27)     4074   (30)
                                            4095   (31)     4098   (31)     4100   (31)     4103   (31)     4970   (41)
                                            5270   (45)     5272   (45)     5368   (46)     5370   (46)     6781   (60)
                                            6785   (60)     7596   (67)     7634   (67)     7705   (67)
DCS_DATA_REG   =00007400    863    (4)      2807   (14)     3053   (18)     3112   (19)     863    (4)
DCS_SCRATCH_ADR=00000036    863    (4)      3260   (20)     3262   (20)     3278   (20)     3305   (20)     3307   (20)
                                            3308   (20)     3313   (20)     3317   (20)     3333   (20)     4009   (28)
                                            4011   (28)     4012   (28)     4040   (29)     4042   (29)     4043   (29)
                                            5480   (47)     5575   (47)     5576   (47)     5589   (47)     5591   (47)
                                            5592   (47)     5601   (47)     5602   (47)     5617   (47)     5618   (47)
                                            5627   (47)     5628   (47)     9843   (96)     9845   (96)     9846   (96)
                                            9900   (98)     9902   (98)     9903   (98)
DCS_START      =00001800    863    (4)      3968   (27)     3969   (27)
DCS_WRITE_ADR   000001AC-R  1579   (6)      2802   (14)     3260   (20)     3262   (20)     3305   (20)     3307   (20)
                                            3313   (20)     3354   (20)     4009   (28)     4011   (28)     4040   (29)
                                            4042   (29)     4157   (33)     5480   (47)     5575   (47)     5589   (47)
                                            5591   (47)     5601   (47)     5617   (47)     5627   (47)     5930   (50)
                                            9843   (96)     9845   (96)     9900   (98)     9902   (98)
DCS_WRITE_PTR   000001AE-R  1580   (6)      2806   (14)     3260   (20)     3262   (20)     3305   (20)     3307   (20)
                                            3313   (20)     3354   (20)     4009   (28)     4011   (28)     4040   (29)
                                            4042   (29)     4157   (33)     5480   (47)     5575   (47)     5589   (47)
                                            5591   (47)     5601   (47)     5617   (47)     5627   (47)     5930   (50)
                                            9843   (96)     9845   (96)     9900   (98)     9902   (98)
DCS_WRITE_WDCNT 000001B0-R  1581   (6)      2804   (14)     3260   (20)     3262   (20)     3305   (20)     3307   (20)
                                            3313   (20)     3354   (20)     4009   (28)     4011   (28)     4040   (29)
                                            4042   (29)     4157   (33)     5480   (47)     5575   (47)     5589   (47)
                                            5591   (47)     5601   (47)     5617   (47)     5627   (47)     5930   (50)
                                            9843   (96)     9845   (96)     9900   (98)     9902   (98)
DC_LOW         =00000040    863    (4)      4077   (30)     4079   (30)
DC_MSG          0000062C-R  1958   (6)      9633   (94)
DECODE_ERR      00001B97-R  5723   (48)     5705   (48)     5707   (48)
DECODE_FLAG_NAM 00001B85-R  5703   (48)     5188   (45)     5398   (46)
DEPOSIT_ACTION  00001B84-R  5660   (47)     1704   (6)
DESELECT_DCS    00001383-R  3889   (25)     2801   (14)     3044   (18)     3104   (19)     4094   (31)
DIAGNOSE_ACTION 000016E5-R  4752   (40)     1696   (6)
DIAGNOSE_DONE   000017A2-R  4893   (40)     4786   (40)
DIAGNOSE_ERR    000017A0-R  4887   (40)     4801   (40)     4840   (40)
DIAGNOSE_FLAG  =00000004    863    (4)      4203   (34)     4270   (34)     4761   (40)     4895   (40)
DIAGNOSE_LOOP   00001712-R  4781   (40)     4808   (40)     4816   (40)     4824   (40)     4832   (40)     4873   (40)
                                            4883   (40)
```

```
DIRECT_MSG      00000076-R   1469   (6)      2361   (7)
DIRECT_SEARCH   0000121B-R   3603   (23)     2349   (7)
DISABLE_CMI     000013D5-R   4003   (28)     3299   (20)     6122   (52)
DISPLAY_DATA    00002C0F-R   9862   (97)     5650   (47)     8959   (87)     8974   (87)     8984   (87)     8994   (87)
                                             9012   (87)     9050   (87)
DPM            =00000000     863    (4)
DPM_NAME        000000A5-R   1485   (6)      1470   (6)
DUMP_EXIT       00002689-R   8459   (79)     8453   (79)
DUMP_LOG        00002646-R   8438   (79)     1828   (6)
DUMP_RET        00002692-R   8461   (79)     8441   (79)
E              =00000003     856    (3)      1405   (5)      2316   (7)      3629   (23)     3632   (23)     3672   (23)
                                             3688   (23)     4431   (36)     4446   (36)     4459   (36)     4650   (39)
                                             4667   (39)     4846   (40)     4868   (40)     4869   (40)     4871   (40)
                                             5815   (49)     5853   (49)     5913   (50)     6662   (58)     7162   (64)
                                             8268   (76)     8558   (81)     8564   (81)     9085   (87)     9250   (89)
                                             9262   (89)     9290   (89)     9414   (90)     9531   (92)     9532   (92)
                                             9708   (95)     9714   (95)
ENABLE_CMI      00001417-R   4034   (29)     2998   (17)
ENABLE_STALL    00002520-R   8197   (75)     1832   (6)
ENABLE_TRAP     00002517-R   8170   (74)     1827   (6)
ENDPASS_MSG     00000584-R   1922   (6)      2379   (7)
END_LOOP        00001D29-R   6262   (54)     1809   (6)
END_PASS        0000242E-R   7857   (69)     1822   (6)
END_SLICE      =00000025     863    (4)      9639   (94)
END_TEST        00002416-R   7822   (68)     1810   (6)
EN_TRACE_REG   =00000008     863    (4)
ERROR_FLAG     =00000010     863    (4)      6073   (51)     6941   (62)     7071   (63)     7195   (64)     7274   (65)
                                             7445   (66)     7540   (66)     8354   (76)     9597   (93)
ERROR_LOG       00002529-R   8240   (76)     1829   (6)
ERROR_LOG_ADR  =000032FC     863    (4)      8267   (76)     8346   (76)     8443   (79)
ERROR_LOOP      00001D88-R   6371   (55)     1811   (6)
ERROR_LOOP_ADDR 00000425-R   1852   (6)      6070   (51)     6373   (55)     7331   (65)
ERROR_MSG       0000059B-R   1940   (6)      8917   (87)
ERROR_PC        00000594-R   1936   (6)      5106   (44)     7321   (65)     7325   (65)
ERR_LOG_INIT   =00000001     863    (4)      1445   (6)      4764   (40)     8272   (76)     8276   (76)     8445   (79)
                                             8463   (79)
EXAMINE_ACTION  000019C1-R   5477   (47)     1703   (6)
EXAMINE_DISPLAY 00001B79-R   5649   (47)     5640   (47)
EXAMINE_ERR     00001B7F-R   5652   (47)     5491   (47)     5516   (47)     5523   (47)     5534   (47)     5584   (47)
EXAMINE_EX      00001B81-R   5655   (47)     5651   (47)
EXAMINE_MA      00001A5B-R   5567   (47)     5505   (47)
EXAMINE_MD      00001A51-R   5555   (47)     5501   (47)
EXAMINE_MT      00001A39-R   5531   (47)     5495   (47)
EXAMINE_PB      00001A56-R   5561   (47)     5503   (47)
EXAMINE_PC      00001A47-R   5543   (47)     5497   (47)
EXAMINE_PS      00001AD5-R   5599   (47)     5511   (47)
EXAMINE_RT      00001A2B-R   5520   (47)     5493   (47)
EXAMINE_SF      00001B41-R   5625   (47)     5515   (47)
EXAMINE_SR      00001A89-R   5581   (47)     5509   (47)
EXAMINE_ST      00001B13-R   5615   (47)     5513   (47)
EXAMINE_TEMP    000003BC-R   1801   (6)      5482   (47)     5483   (47)     5605   (47)     5608   (47)     5638   (47)
                                             5639   (47)
EXAMINE_VA      00001A4C-R   5549   (47)     5499   (47)
EXAMINE_WD      00001A60-R   5573   (47)     5507   (47)
EXP_STALL_FLAG =00000010     863    (4)      7662   (67)     8199   (75)
```

```
EXP_UTRAP_FLAG =00000040    863     (4)       6012     (51)      6725     (59)      7650     (67)      8172     (74)
EX_DISP_WH      00001B76-R  5644    (47)      5527     (47)      5539     (47)      5577     (47)
FAC            =00000020    863     (4)
FAC_NAME        00000B9E-R  2126    (6)       2000     (6)
FAIL_ARRAY_MSG  000005DA-R  1948    (6)       9629     (94)
FAIL_ARR_INDEX  000005F0-R  1949    (6)       9656     (94)
FAIL_MOD_INDEX  00000618-R  1951    (6)       9091     (87)
FAULT          =00000002    863     (4)       1418     (5)       2888     (15)      2903     (15)      2938     (16)      2954     (16)
                                              3004     (17)      4076     (30)      4976     (41)      6125     (52)      6575     (57)
                                              7696     (67)
FCC            =00000015    863     (4)
FCS            =00000021    863     (4)
FCS_NAME        00000BB0-R  2131    (6)       2001     (6)
FETCH_0         0000147B-R  4093    (31)      2999     (17)      6123     (52)
FETCH_FLAG     =00000008    863     (4)       6772     (60)      7512     (66)      7570     (66)      7594     (67)
FETCH_MIC_INSTR 00001EA6-R  6770    (60)      1821     (6)
FEX            =0000001E    863     (4)
FEX_NAME        00000B6A-R  2114    (6)       1998     (6)
FFH            =00000024    863     (4)
FFH_NAME        00000C49-R  2160    (6)       2004     (6)
FFL            =00000023    863     (4)
FFL_NAME        00000C0C-R  2149    (6)       2003     (6)
FIC            =0000001F    863     (4)
FIC_NAME        00000B89-R  2121    (6)       1999     (6)
FILE_FOR_CMC    00000160-R  1535    (6)
FILE_FOR_DPM    0000014E-R  1526    (6)       2314     (7)
FILE_FOR_FPA    0000015A-R  1532    (6)
FILE_FOR_MIC    00000154-R  1529    (6)
FILE_FOR_RDM    00000166-R  1538    (6)
FILE_NAME_HEAD =0000014E-R  1525    (6)       1543     (6)       1546     (6)
FILE_NAME_TAIL =0000016C-R  1545    (6)       1546     (6)
FILE_NAM_INDEX  00000063-R  1463    (6)       2310     (7)       2317     (7)       2338     (7)       2340     (7)       3609     (23)
FILE_NAM_PTR    00000139-R  1511    (6)       2325     (7)       2518     (9)
FILL_DCS        000014D1-R  4152    (33)      1425     (5)       3237     (20)
FIO            =0000001D    863     (4)
FIO_NAME        00000AF1-R  2095    (6)       1997     (6)
FLAG_KEY       =00000004    863     (4)       4949     (41)      5157     (45)      5355     (46)
FMR            =00000022    863     (4)
FMR_NAME        00000BED-R  2142    (6)       2002     (6)
FPA            =00000002    863     (4)
FPA_NAME        000000BB-R  1487    (6)       1472     (6)
FP_BOOT_0      =00000001    863     (4)
FP_BOOT_1      =00000002    863     (4)
FP_START_0     =00000004    863     (4)
FP_START_1     =00000008    863     (4)
FQA            =00000014    863     (4)
FRONT1         =00007420    863     (4)
FRONT2         =00007421    863     (4)
FRONT_PNL_1    =00007420    863     (4)       863      (4)
FRONT_PNL_2    =00007421    863     (4)       1417     (5)       2887     (15)      2902     (15)      2937     (16)      2953     (16)
                                              3003     (17)      4075     (30)      4975     (41)      6116     (52)      6124     (52)
                                              6574     (57)      7695     (67)      863      (4)
FRONT_PNL_LOCK =00000080    863     (4)
GATE_ARRAY_LST  0000062F-R  1959    (6)       9652     (94)
GET_CMD_LINE    000014EF-R  4200    (34)      1430     (5)       2344     (7)       2398     (7)       2475     (8)       2524     (9)
```

```
                                             4282   (34)     5819   (49)     5998   (51)     6025   (51)     7281   (65)
                                             7306   (65)     7530   (66)     7555   (66)     7586   (66)     7642   (67)
                                             7694   (67)     7792   (67)
GET_INDEX_VALUE 000026AF-R   8546   (81)     5906   (50)     6170   (53)     6267   (54)     6341   (54)     6422   (56)
                                             6545   (57)     6654   (58)     6657   (58)     6825   (61)     6844   (61)
                                             6916   (62)     7006   (63)     7024   (63)     7149   (64)     7765   (67)
                                             7774   (67)     7784   (67)     7907   (70)     7959   (70)     7980   (70)
                                             8031   (71)     8040   (71)     8107   (72)     8139   (73)     8967   (87)
                                             8977   (87)     8987   (87)
GOOD_DATA       0000043F-R   1869   (6)      7036   (63)     7132   (64)     7133   (64)     8724   (84)     8730   (84)
                                             8731   (84)     8782   (85)     8952   (87)     9523   (92)     9528   (92)
                                             9538   (92)     9728   (95)
H              =00000004     856    (3)      1374   (5)      1375   (5)      1384   (5)      1385   (5)      1386   (5)
                                             1387   (5)      1388   (5)      1393   (5)      1395   (5)      1396   (5)
                                             1400   (5)      1402   (5)      1404   (5)      1408   (5)      1429   (5)
                                             2269   (7)      2270   (7)      2304   (7)      2314   (7)      2361   (7)
                                             2362   (7)      2378   (7)      2379   (7)      2380   (7)      2381   (7)
                                             2568   (10)     2569   (10)     2570   (10)     2571   (10)     2572   (10)
                                             2630   (11)     2695   (12)     2703   (12)     2708   (12)     2710   (12)
                                             2800   (14)     2826   (14)     2835   (14)     2841   (14)     2848   (14)
                                             2893   (15)     2942   (16)     3043   (18)     3055   (18)     3058   (18)
                                             3063   (18)     3103   (19)     3114   (19)     3117   (19)     3122   (19)
                                             3207   (20)     3213   (20)     3216   (20)     3224   (20)     3225   (20)
                                             3229   (20)     3230   (20)     3231   (20)     3236   (20)     3238   (20)
                                             3242   (20)     3247   (20)     3248   (20)     3251   (20)     3253   (20)
                                             3257   (20)     3260   (20)     3262   (20)     3263   (20)     3271   (20)
                                             3278   (20)     3282   (20)     3284   (20)     3286   (20)     3287   (20)
                                             3291   (20)     3296   (20)     3297   (20)     3298   (20)     3305   (20)
                                             3307   (20)     3308   (20)     3313   (20)     3314   (20)     3317   (20)
                                             3318   (20)     3326   (20)     3333   (20)     3337   (20)     3339   (20)
                                             3341   (20)     3342   (20)     3348   (20)     3353   (20)     3358   (20)
                                             3359   (20)     3364   (20)     3366   (20)     3367   (20)     3369   (20)
                                             3374   (20)     3375   (20)     3380   (20)     3394   (20)     3403   (20)
                                             3406   (20)     3412   (20)     3413   (20)     3414   (20)     3415   (20)
                                             3420   (20)     3422   (20)     3423   (20)     3425   (20)     3462   (21)
                                             3466   (21)     3467   (21)     3473   (21)     3481   (21)     3482   (21)
                                             3485   (21)     3521   (22)     3525   (22)     3605   (23)     3606   (23)
                                             3607   (23)     3608   (23)     3611   (23)     3612   (23)     3618   (23)
                                             3626   (23)     3627   (23)     3633   (23)     3637   (23)     3641   (23)
                                             3643   (23)     3644   (23)     3647   (23)     3648   (23)     3650   (23)
                                             3654   (23)     3655   (23)     3661   (23)     3679   (23)     3683   (23)
                                             3690   (23)     3693   (23)     3694   (23)     3706   (23)     3707   (23)
                                             3709   (23)     3710   (23)     3717   (23)     3719   (23)     3722   (23)
                                             3811   (24)     3812   (24)     3813   (24)     3817   (24)     3825   (24)
                                             3856   (24)     3862   (24)     3863   (24)     3894   (25)     3967   (27)
                                             3969   (27)     3985   (27)     4006   (28)     4007   (28)     4009   (28)
                                             4011   (28)     4012   (28)     4013   (28)     4014   (28)     4037   (29)
                                             4038   (29)     4040   (29)     4042   (29)     4043   (29)     4044   (29)
                                             4045   (29)     4122   (32)     4126   (32)     4127   (32)     4129   (32)
                                             4130   (32)     4131   (32)     4133   (32)     4134   (32)     4137   (32)
                                             4157   (33)     4209   (34)     4210   (34)     4217   (34)     4218   (34)
                                             4219   (34)     4231   (34)     4240   (34)     4250   (34)     4265   (34)
                                             4272   (34)     4281   (34)     4309   (35)     4311   (35)     4312   (35)
                                             4355   (35)     4375   (35)     4451   (36)     4462   (36)     4469   (36)
                                             4518   (37)     4526   (37)     4568   (38)     4571   (38)     4573   (38)
```

```
                                                  4633    (39)      4649    (39)      4653    (39)      4655    (39)      4658    (39)
                                                  4666    (39)      4676    (39)      4686    (39)      4691    (39)      4696    (39)
                                                  4784    (40)      4845    (40)      4861    (40)      4882    (40)      4954    (41)
                                                  4957    (41)      4983    (41)      4987    (41)      4990    (41)      4991    (41)
                                                  4994    (41)      4996    (41)      5002    (41)      5004    (41)      5006    (41)
                                                  5154    (45)      5172    (45)      5185    (45)      5210    (45)      5211    (45)
                                                  5214    (45)      5217    (45)      5226    (45)      5246    (45)      5247    (45)
                                                  5250    (45)      5283    (45)      5284    (45)      5288    (45)      5352    (46)
                                                  5378    (46)      5379    (46)      5383    (46)      5395    (46)      5415    (46)
                                                  5480    (47)      5481    (47)      5484    (47)      5489    (47)      5575    (47)
                                                  5576    (47)      5589    (47)      5591    (47)      5592    (47)      5601    (47)
                                                  5602    (47)      5617    (47)      5618    (47)      5627    (47)      5628    (47)
                                                  5639    (47)      5645    (47)      5801    (49)      5807    (49)      5809    (49)
                                                  5810    (49)      5826    (49)      5833    (49)      5839    (49)      5849    (49)
                                                  5854    (49)      6176    (53)      6215    (53)      6218    (53)      6268    (54)
                                                  6269    (54)      6270    (54)      6271    (54)      6275    (54)      6278    (54)
                                                  6283    (54)      6296    (54)      6297    (54)      6299    (54)      6309    (54)
                                                  6319    (54)      6325    (54)      6344    (54)      6428    (56)      6435    (56)
                                                  6437    (56)      6450    (56)      6478    (56)      6499    (56)      6503    (56)
                                                  6507    (56)      6569    (57)      6577    (57)      6580    (57)      6583    (57)
                                                  6839    (61)      6862    (61)      7035    (63)      7043    (63)      7049    (63)
                                                  7057    (63)      7131    (64)      7147    (64)      7154    (64)      7171    (64)
                                                  7177    (64)      7197    (64)      7297    (65)      7320    (65)      7324    (65)
                                                  7426    (66)      7490    (66)      7491    (66)      7507    (66)      7523    (66)
                                                  7525    (66)      7527    (66)      7529    (66)      7629    (67)      7680    (67)
                                                  7681    (67)      7682    (67)      7686    (67)      7687    (67)      7693    (67)
                                                  7726    (67)      7735    (67)      7736    (67)      7737    (67)      7738    (67)
                                                  7739    (67)      7747    (67)      7749    (67)      7750    (67)      7752    (67)
                                                  7754    (67)      7756    (67)      7760    (67)      7761    (67)      7768    (67)
                                                  7769    (67)      7771    (67)      7777    (67)      7778    (67)      7780    (67)
                                                  7781    (67)      7787    (67)      7788    (67)      7790    (67)      7791    (67)
                                                  7930    (70)      7933    (70)      7937    (70)      7951    (70)      7954    (70)
                                                  7956    (70)      7957    (70)      7960    (70)      7971    (70)      7974    (70)
                                                  7976    (70)      7978    (70)      7981    (70)      7990    (70)      7992    (70)
                                                  8032    (71)      8050    (71)      8062    (71)      8068    (71)      8077    (71)
                                                  8110    (72)      8145    (73)      8247    (76)      8249    (76)      8260    (76)
                                                  8261    (76)      8262    (76)      8267    (76)      8308    (76)      8310    (76)
                                                  8313    (76)      8314    (76)      8316    (76)      8317    (76)      8320    (76)
                                                  8321    (76)      8322    (76)      8325    (76)      8326    (76)      8327    (76)
                                                  8337    (76)      8340    (76)      8343    (76)      8346    (76)      8442    (79)
                                                  8443    (79)      8452    (79)      8454    (79)      8455    (79)      8456    (79)
                                                  8460    (79)      8497    (80)      8503    (80)      8557    (81)      8565    (81)
                                                  8573    (81)      8575    (81)      8623    (82)      8675    (83)      8676    (83)
                                                  8679    (83)      8683    (83)      8728    (84)      8729    (84)      8732    (84)
                                                  8736    (84)      8783    (85)      8791    (85)      8806    (85)      8845    (86)
                                                  8915    (87)      8916    (87)      8917    (87)      8928    (87)      8930    (87)
                                                  8934    (87)      8935    (87)      8939    (87)      8945    (87)      8946    (87)
                                                  8947    (87)      8951    (87)      8952    (87)      8956    (87)      8957    (87)
                                                  8958    (87)      8963    (87)      8970    (87)      8971    (87)      8973    (87)
                                                  8980    (87)      8981    (87)      8983    (87)      8990    (87)      8991    (87)
                                                  8993    (87)      8998    (87)      9002    (87)      9005    (87)      9011    (87)
                                                  9016    (87)      9020    (87)      9022    (87)      9024    (87)      9034    (87)
                                                  9040    (87)      9042    (87)      9043    (87)      9045    (87)      9047    (87)
                                                  9049    (87)      9052    (87)      9053    (87)      9056    (87)      9061    (87)
                                                  9069    (87)      9073    (87)      9074    (87)      9075    (87)      9076    (87)
```

```
                                                       9077   (87)         9079   (87)         9081   (87)         9086   (87)         9090   (87)
                                                       9091   (87)         9092   (87)         9093   (87)         9095   (87)         9096   (87)
                                                       9098   (87)         9099   (87)         9139   (88)         9140   (88)         9145   (88)
                                                       9150   (88)         9212   (89)         9213   (89)         9215   (89;         9218   (89)
                                                       9220   (89)         9224   (89)         9228   (89)         9249   (89)         9272   (89)
                                                       9283   (89)         9285   (89)         9302   (89)         9337   (89)         9346   (89)
                                                       9349   (89)         9350   (89)         9392   (90)         9406   (90)         9412   (90)
                                                       9413   (90)         9429   (90)         9430   (90)         9441   (90)         9447   (90)
                                                       9448   (90)         9449   (90)         9584   (93)         9586   (93)         9625   (94)
                                                       9626   (94)         9627   (94)         9628   (94)         9629   (94)         9630   (94)
                                                       9631   (94)         9632   (94)         9633   (94)         9634   (94)         9635   (94)
                                                       9641   (94)         9644   (94)         9652   (94)         9655   (94)         9656   (94)
                                                       9657   (94)         9658   (94)         9660   (94)         9661   (94)         9663   (94)
                                                       9664   (94)         9665   (94)         9666   (94)         9667   (94)         9706   (95)
                                                       9715   (95)         9720   (95)         9725   (95)         9728   (95)         9734   (95)
                                                       9740   (95)         9746   (95)         9750   (95)         9763   (95)         9769   (95)
                                                       9776   (95)         9777   (95)         9778   (95)         9779   (95)         9787   (95)
                                                       9795   (95)         9796   (95)         9802   (95)         9805   (95)         9808   (95)
                                                       9836   (96)         9840   (96)         9843   (96)         9845   (96)         9846   (96)
                                                       9863   (97)         9864   (97)         9865   (97)         9867   (97)         9893   (98)
                                                       9897   (98)         9900   (98)         9902   (98)         9903   (98)
HALT_FLAG         =00000001      863       (4)         1453   (6)          4756   (40)         5100   (44)         7305   (65)
HALT_FLAG_NAME     0000034B-R    1730      (6)         4122   (32)
HALT_KEY          =00000008      863       (4)         5704   (48)
HALT_ON_MATCH     =00000001      863       (4)         5271   (45)         5369   (46)
HEAD              =FFFF7C00      863       (4)         1373   (5)          1376   (5)          1377   (5)          1389   (5)          1393   (5)
                                                       1403   (5)          1410   (5)          1415   (5)          1416   (5)          1417   (5)
                                                       1419   (5)          1421   (5)          1424   (5)          1425   (5)          1430   (5)
                                                       1431   (5)          1470   (6)          1471   (6)          1472   (6)          1473   (6)
                                                       1474   (6)          1475   (6)          1476   (6)          1477   (6)          1478   (6)
                                                       1479   (6)          1480   (6)          1481   (6)          1482   (6)          1696   (6)
                                                       1697   (6)          1698   (6)          1699   (6)          1700   (6)          1701   (6)
                                                       1702   (6)          1703   (6)          1704   (6)          1705   (6)          1711   (6)
                                                       1804   (6)          1805   (6)          1806   (6)          1807   (6)          1808   (6)
                                                       1809   (6)          1810   (6)          1811   (6)          1812   (6)          1813   (6)
                                                       1814   (6)          1815   (6)          1816   (6)          1817   (6)          1818   (6)
                                                       1819   (6)          1820   (6)          1821   (6)          1822   (6)          1823   (6)
                                                       1824   (6)          1825   (6)          1826   (6)          1827   (6)          1828   (6)
                                                       1829   (6)          1830   (6)          1831   (6)          1832   (6)          1837   (6)
                                                       1838   (6)          1839   (6)          1990   (6)          1991   (6)          1992   (6)
                                                       1993   (6)          1994   (6)          1995   (6)          1996   (6)          1997   (6)
                                                       1998   (6)          1999   (6)          2000   (6)          2001   (6)          2002   (6)
                                                       2003   (6)          2004   (6)          2264   (7)          2266   (7)          2267   (7)
                                                       2269   (7)          2270   (7)          2272   (7)          2276   (7)          2279   (7)
                                                       2282   (7)          2284   (7)          2290   (7)          2296   (7)          2297   (7)
                                                       2301   (7)          2302   (7)          2305   (7)          2310   (7)          2314   (7)
                                                       2317   (7)          2319   (7)          2321   (7)          2325   (7)          2329   (7)
                                                       2330   (7)          2331   (7)          2332   (7)          2333   (7)          2338   (7)
                                                       2340   (7)          2342   (7)          2343   (7)          2344   (7)          2345   (7)
                                                       2349   (7)          2351   (7)          2357   (7)          2361   (7)          2362   (7)
                                                       2363   (7)          2368   (7)          2369   (7)          2370   (7)          2374   (7)
                                                       2378   (7)          2379   (7)          2380   (7)          2381   (7)          2382   (7)
                                                       2384   (7)          2386   (7)          2388   (7)          2389   (7)          2390   (7)
                                                       2391   (7)          2392   (7)          2397   (7)          2398   (7)          2399   (7)
                                                       2463   (8)          2464   (8)          2465   (8)          2466   (8)          2470   (8)
```

```
                                    2471     (8)       2472     (8)       2473     (8)       2474     (8)       2475     (8)
                                    2476     (8)       2477     (8)       2518     (9)       2519     (9)       2520     (9)
                                    2522     (9)       2523     (9)       2524     (9)       2525     (9)       2568     (10)
                                    2569     (10)      2570     (10)      2571     (10)      2572     (10)      2627     (11)
                                    2629     (11)      2631     (11)      2634     (11)      2635     (11)      2636     (11)
                                    2697     (12)      2699     (12)      2702     (12)      2705     (12)      2707     (12)
                                    2708     (12)      2709     (12)      2710     (12)      2745     (13)      2747     (13)
                                    2748     (13)      2749     (13)      2751     (13)      2753     (13)      2799     (14)
                                    2801     (14)      2802     (14)      2803     (14)      2804     (14)      2806     (14)
                                    2807     (14)      2837     (14)      2843     (14)      2847     (14)      2849     (14)
                                    2887     (15)      2889     (15)      2891     (15)      2893     (15)      2895     (15)
                                    2901     (15)      2902     (15)      2906     (15)      2937     (16)      2939     (16)
                                    2940     (16)      2942     (16)      2944     (16)      2952     (16)      2953     (16)
                                    2957     (16)      2997     (17)      2998     (17)      2999     (17)      3001     (17)
                                    3002     (17)      3003     (17)      3005     (17)      3006     (17)      3041     (18)
                                    3044     (18)      3045     (18)      3046     (18)      3047     (18)      3048     (18)
                                    3051     (18)      3053     (18)      3054     (18)      3056     (18)      3057     (18)
                                    3062     (18)      3065     (18)      3068     (18)      3101     (19)      3104     (19)
                                    3105     (19)      3106     (19)      3107     (19)      3108     (19)      3110     (19)
                                    3112     (19)      3113     (19)      3115     (19)      3116     (19)      3121     (19)
                                    3124     (19)      3127     (19)      3207     (20)      3215     (20)      3219     (20)
                                    3226     (20)      3227     (20)      3230     (20)      3237     (20)      3245     (20)
                                    3250     (20)      3252     (20)      3253     (20)      3258     (20)      3260     (20)
                                    3262     (20)      3267     (20)      3274     (20)      3278     (20)      3279     (20)
                                    3281     (20)      3285     (20)      3294     (20)      3299     (20)      3305     (20)
                                    3307     (20)      3308     (20)      3313     (20)      3315     (20)      3316     (20)
                                    3317     (20)      3323     (20)      3329     (20)      3333     (20)      3340     (20)
                                    3352     (20)      3354     (20)      3358     (20)      3360     (20)      3368     (20)
                                    3369     (20)      3371     (20)      3376     (20)      3377     (20)      3379     (20)
                                    3381     (20)      3383     (20)      3385     (20)      3387     (20)      3391     (20)
                                    3392     (20)      3396     (20)      3402     (20)      3403     (20)      3405     (20)
                                    3416     (20)      3424     (20)      3426     (20)      3463     (21)      3465     (21)
                                    3469     (21)      3474     (21)      3480     (21)      3484     (21)      3521     (22)
                                    3522     (22)      3527     (22)      3529     (22)      3531     (22)      3535     (22)
                                    3536     (22)      3604     (23)      3606     (23)      3609     (23)      3613     (23)
                                    3627     (23)      3636     (23)      3637     (23)      3646     (23)      3649     (23)
                                    3650     (23)      3657     (23)      3661     (23)      3665     (23)      3667     (23)
                                    3673     (23)      3677     (23)      3679     (23)      3682     (23)      3687     (23)
                                    3695     (23)      3701     (23)      3702     (23)      3724     (23)      3779     (24)
                                    3781     (24)      3785     (24)      3787     (24)      3791     (24)      3792     (24)
                                    3797     (24)      3798     (24)      3800     (24)      3803     (24)      3804     (24)
                                    3806     (24)      3807     (24)      3808     (24)      3810     (24)      3811     (24)
                                    3812     (24)      3813     (24)      3814     (24)      3815     (24)      3816     (24)
                                    3817     (24)      3821     (24)      3826     (24)      3828     (24)      3830     (24)
                                    3835     (24)      3837     (24)      3841     (24)      3843     (24)      3845     (24)
                                    3850     (24)      3852     (24)      3856     (24)      3858     (24)      3862     (24)
                                    3863     (24)      3864     (24)      3890     (25)      3893     (25)      3895     (25)
                                    3915     (26)      3916     (26)      3917     (26)      3918     (26)      3953     (27)
                                    3955     (27)      3959     (27)      3961     (27)      3962     (27)      3970     (27)
                                    3971     (27)      3975     (27)      3978     (27)      3979     (27)      3983     (27)
                                    3984     (27)      4009     (28)      4011     (28)      4012     (28)      4040     (29)
                                    4042     (29)      4043     (29)      4068     (30)      4069     (30)      4073     (30)
                                    4074     (30)      4075     (30)      4078     (30)      4080     (30)      4094     (31)
                                    4095     (31)      4098     (31)      4099     (31)      4100     (31)      4103     (31)
                                    4104     (31)      4120     (32)      4122     (32)      4124     (32)      4128     (32)
```

```
                                              4129    (32)    4136    (32)    4137    (32)    4157    (33)    4161    (33)
                                              4201    (34)    4204    (34)    4206    (34)    4207    (34)    4208    (34)
                                              4209    (34)    4210    (34)    4212    (34)    4213    (34)    4215    (34)
                                              4216    (34)    4217    (34)    4218    (34)    4219    (34)    4220    (34)
                                              4226    (34)    4228    (34)    4229    (34)    4230    (34)    4231    (34)
                                              4232    (34)    4237    (34)    4241    (34)    4251    (34)    4254    (34)
                                              4259    (34)    4265    (34)    4268    (34)    4271    (34)    4272    (34)
                                              4275    (34)    4280    (34)    4282    (34)    4309    (35)    4310    (35)
                                              4311    (35)    4313    (35)    4317    (35)    4318    (35)    4322    (35)
                                              4326    (35)    4327    (35)    4329    (35)    4330    (35)    4334    (35)
                                              4335    (35)    4336    (35)    4340    (35)    4346    (35)    4347    (35)
                                              4348    (35)    4351    (35)    4358    (35)    4359    (35)    4360    (35)
                                              4362    (35)    4363    (35)    4364    (35)    4368    (35)    4369    (35)
                                              4376    (35)    4378    (35)    4380    (35)    4434    (36)    4436    (36)
                                              4448    (36)    4450    (36)    4452    (36)    4460    (36)    4463    (36)
                                              4516    (37)    4519    (37)    4525    (37)    4527    (37)    4532    (37)
                                              4569    (38)    4572    (38)    4616    (39)    4630    (39)    4632    (39)
                                              4635    (39)    4637    (39)    4642    (39)    4644    (39)    4652    (39)
                                              4662    (39)    4668    (39)    4672    (39)    4679    (39)    4685    (39)
                                              4687    (39)    4694    (39)    4695    (39)    4753    (40)    4757    (40)
                                              4758    (40)    4762    (40)    4763    (40)    4768    (40)    4770    (40)
                                              4773    (40)    4774    (40)    4786    (40)    4791    (40)    4793    (40)
                                              4795    (40)    4797    (40)    4799    (40)    4801    (40)    4805    (40)
                                              4807    (40)    4808    (40)    4813    (40)    4815    (40)    4816    (40)
                                              4821    (40)    4823    (40)    4824    (40)    4829    (40)    4831    (40)
                                              4832    (40)    4840    (40)    4844    (40)    4850    (40)    4854    (40)
                                              4860    (40)    4870    (40)    4872    (40)    4873    (40)    4881    (40)
                                              4883    (40)    4894    (40)    4896    (40)    4950    (41)    4954    (41)
                                              4955    (41)    4956    (41)    4957    (41)    4958    (41)    4960    (41)
                                              4961    (41)    4966    (41)    4970    (41)    4972    (41)    4974    (41)
                                              4975    (41)    4979    (41)    4981    (41)    4982    (41)    4983    (41)
                                              4987    (41)    4993    (41)    4994    (41)    4998    (41)    5001    (41)
                                              5004    (41)    5007    (41)    5056    (42)    5058    (42)    5079    (43)
                                              5098    (44)    5101    (44)    5102    (44)    5104    (44)    5105    (44)
                                              5106    (44)    5156    (45)    5158    (45)    5160    (45)    5162    (45)
                                              5164    (45)    5171    (45)    5175    (45)    5176    (45)    5177    (45)
                                              5178    (45)    5187    (45)    5188    (45)    5189    (45)    5190    (45)
                                              5192    (45)    5198    (45)    5203    (45)    5204    (45)    5207    (45)
                                              5209    (45)    5213    (45)    5216    (45)    5218    (45)    5228    (45)
                                              5231    (45)    5234    (45)    5237    (45)    5242    (45)    5245    (45)
                                              5249    (45)    5252    (45)    5253    (45)    5257    (45)    5258    (45)
                                              5264    (45)    5266    (45)    5270    (45)    5272    (45)    5273    (45)
                                              5279    (45)    5286    (45)    5287    (45)    5290    (45)    5291    (45)
                                              5354    (46)    5356    (46)    5358    (46)    5360    (46)    5365    (46)
                                              5367    (46)    5368    (46)    5370    (46)    5371    (46)    5377    (46)
                                              5381    (46)    5382    (46)    5385    (46)    5386    (46)    5387    (46)
                                              5397    (46)    5398    (46)    5399    (46)    5403    (46)    5405    (46)
                                              5406    (46)    5414    (46)    5418    (46)    5419    (46)    5420    (46)
                                              5480    (47)    5482    (47)    5483    (47)    5491    (47)    5493    (47)
                                              5495    (47)    5497    (47)    5499    (47)    5501    (47)    5503    (47)
                                              5505    (47)    5507    (47)    5509    (47)    5511    (47)    5513    (47)
                                              5515    (47)    5516    (47)    5523    (47)    5526    (47)    5527    (47)
                                              5534    (47)    5538    (47)    5539    (47)    5545    (47)    5551    (47)
                                              5557    (47)    5563    (47)    5569    (47)    5575    (47)    5576    (47)
                                              5577    (47)    5584    (47)    5587    (47)    5589    (47)    5591    (47)
```

```
          5592    (47)        5593    (47)        5595    (47)        5601    (47)        5602    (47)
          5603    (47)        5605    (47)        5606    (47)        5608    (47)        5609    (47)
          5611    (47)        5617    (47)        5618    (47)        5619    (47)        5621    (47)
          5627    (47)        5628    (47)        5629    (47)        5638    (47)        5639    (47)
          5640    (47)        5645    (47)        5650    (47)        5651    (47)        5705    (48)
          5707    (48)        5712    (48)        5784    (49)        5786    (49)        5790    (49)
          5792    (49)        5797    (49)        5799    (49)        5803    (49)        5806    (49)
          5808    (49)        5812    (49)        5813    (49)        5816    (49)        5817    (49)
          5818    (49)        5819    (49)        5823    (49)        5824    (49)        5827    (49)
          5830    (49)        5835    (49)        5840    (49)        5844    (49)        5849    (49)
          5857    (49)        5905    (50)        5906    (50)        5910    (50)        5911    (50)
          5918    (50)        5919    (50)        5922    (50)        5927    (50)        5929    (50)
          5930    (50)        5996    (51)        5998    (51)        6003    (51)        6007    (51)
          6011    (51)        6013    (51)        6017    (51)        6019    (51)        6020    (51)
          6022    (51)        6024    (51)        6025    (51)        6026    (51)        6029    (51)
          6030    (51)        6034    (51)        6036    (51)        6040    (51)        6042    (51)
          6044    (51)        6048    (51)        6050    (51)        6051    (51)        6052    (51)
          6053    (51)        6056    (51)        6057    (51)        6061    (51)        6064    (51)
          6068    (51)        6069    (51)        6070    (51)        6071    (51)        6074    (51)
          6075    (51)        6076    (51)        6111    (52)        6116    (52)        6118    (52)
          6120    (52)        6122    (52)        6123    (52)        6124    (52)        6128    (52)
          6169    (53)        6170    (53)        6171    (53)        6178    (53)        6182    (53)
          6184    (53)        6186    (53)        6190    (53)        6192    (53)        6194    (53)
          6195    (53)        6201    (53)        6206    (53)        6210    (53)        6211    (53)
          6216    (53)        6219    (53)        6263    (54)        6265    (54)        6266    (54)
          6267    (54)        6277    (54)        6298    (54)        6315    (54)        6320    (54)
          6321    (54)        6326    (54)        6332    (54)        6335    (54)        6340    (54)
          6341    (54)        6372    (55)        6373    (55)        6421    (56)        6422    (56)
          6427    (56)        6428    (56)        6434    (56)        6436    (56)        6440    (56)
          6452    (56)        6461    (56)        6463    (56)        6470    (56)        6472    (56)
          6482    (56)        6483    (56)        6494    (56)        6501    (56)        6503    (56)
          6508    (56)        6544    (57)        6545    (57)        6546    (57)        6547    (57)
          6551    (57)        6554    (57)        6562    (57)        6564    (57)        6573    (57)
          6574    (57)        6579    (57)        6585    (57)        6624    (58)        6627    (58)
          6629    (58)        6631    (58)        6633    (58)        6635    (58)        6641    (58)
          6642    (58)        6648    (58)        6649    (58)        6653    (58)        6654    (58)
          6656    (58)        6657    (58)        6664    (58)        6667    (58)        6671    (58)
          6673    (58)        6674    (58)        6678    (58)        6680    (58)        6684    (58)
          6686    (58)        6688    (58)        6720    (59)        6724    (59)        6726    (59)
          6730    (59)        6731    (59)        6771    (60)        6773    (60)        6774    (60)
          6775    (60)        6779    (60)        6781    (60)        6782    (60)        6784    (60)
          6785    (60)        6789    (60)        6790    (60)        6824    (61)        6825    (61)
          6829    (61)        6830    (61)        6834    (61)        6837    (61)        6843    (61)
          6844    (61)        6845    (61)        6846    (61)        6847    (61)        6850    (61)
          6856    (61)        6864    (61)        6909    (62)        6910    (62)        6911    (62)
          6915    (62)        6916    (62)        6918    (62)        6920    (62)        6921    (62)
          6922    (62)        6923    (62)        6926    (62)        6928    (62)        6929    (62)
          6933    (62)        6934    (62)        6936    (62)        6937    (62)        6938    (62)
          6939    (62)        6943    (62)        6948    (62)        6951    (62)        6952    (62)
          6999    (63)        7000    (63)        7001    (63)        7005    (63)        7006    (63)
          7008    (63)        7010    (63)        7011    (63)        7012    (63)        7013    (63)
          7016    (63)        7018    (63)        7019    (63)        7023    (63)        7024    (63)
          7025    (63)        7026    (63)        7027    (63)        7030    (63)        7036    (63)
          7037    (63)        7045    (63)        7050    (63)        7051    (63)        7059    (63)
          7063    (63)        7064    (63)        7066    (63)        7067    (63)        7068    (63)
```

```
                                   7069    (63)      7073    (63)      7078    (63)      7081    (63)      7082    (63)
                                   7125    (64)      7127    (64)      7132    (64)      7133    (64)      7134    (64)
                                   7135    (64)      7139    (64)      7140    (64)      7142    (64)      7148    (64)
                                   7149    (64)      7151    (64)      7153    (64)      7155    (64)      7161    (64)
                                   7166    (64)      7175    (64)      7176    (64)      7179    (64)      7183    (64)
                                   7186    (64)      7187    (64)      7192    (64)      7196    (64)      7198    (64)
                                   7201    (64)      7273    (65)      7275    (65)      7279    (65)      7281    (65)
                                   7285    (65)      7287    (65)      7291    (65)      7293    (65)      7297    (65)
                                   7298    (65)      7300    (65)      7304    (65)      7306    (65)      7310    (65)
                                   7313    (65)      7316    (65)      7320    (65)      7321    (65)      7323    (65)
                                   7325    (65)      7327    (65)      7331    (65)      7332    (65)      7333    (65)
                                   7335    (65)      7424    (66)      7425    (66)      7426    (66)      7427    (66)
                                   7428    (66)      7431    (66)      7432    (66)      7434    (66)      7438    (66)
                                   7440    (66)      7444    (66)      7446    (66)      7447    (66)      7450    (66)
                                   7453    (66)      7454    (66)      7456    (66)      7462    (66)      7463    (66)
                                   7465    (66)      7466    (66)      7467    (66)      7473    (66)      7474    (66)
                                   7476    (66)      7477    (66)      7478    (66)      7480    (66)      7484    (66)
                                   7486    (66)      7490    (66)      7491    (66)      7492    (66)      7496    (66)
                                   7498    (66)      7500    (66)      7501    (66)      7503    (66)      7506    (66)
                                   7507    (66)      7511    (66)      7513    (66)      7514    (66)      7517    (66)
                                   7519    (66)      7523    (66)      7524    (66)      7528    (66)      7529    (66)
                                   7530    (66)      7535    (66)      7537    (66)      7538    (66)      7541    (66)
                                   7542    (66)      7544    (66)      7550    (66)      7552    (66)      7554    (66)
                                   7555    (66)      7561    (66)      7562    (66)      7563    (66)      7565    (66)
                                   7569    (66)      7571    (66)      7572    (66)      7574    (66)      7575    (66)
                                   7576    (66)      7577    (66)      7578    (66)      7582    (66)      7585    (66)
                                   7586    (66)      7587    (66)      7593    (67)      7595    (67)      7596    (67)
                                   7598    (67)      7603    (67)      7604    (67)      7606    (67)      7607    (67)
                                   7609    (67)      7610    (67)      7614    (67)      7615    (67)      7617    (67)
                                   7621    (67)      7626    (67)      7628    (67)      7629    (67)      7630    (67)
                                   7634    (67)      7635    (67)      7639    (67)      7641    (67)      7642    (67)
                                   7643    (67)      7647    (67)      7648    (67)      7649    (67)      7651    (67)
                                   7652    (67)      7655    (67)      7657    (67)      7661    (67)      7663    (67)
                                   7667    (67)      7669    (67)      7673    (67)      7675    (67)      7680    (67)
                                   7681    (67)      7682    (67)      7683    (67)      7685    (67)      7686    (67)
                                   7687    (67)      7688    (67)      7692    (67)      7693    (67)      7694    (67)
                                   7695    (67)      7697    (67)      7700    (67)      7701    (67)      7705    (67)
                                   7707    (67)      7709    (67)      7710    (67)      7715    (67)      7717    (67)
                                   7726    (67)      7727    (67)      7728    (67)      7729    (67)      7734    (67)
                                   7735    (67)      7736    (67)      7737    (67)      7738    (67)      7739    (67)
                                   7740    (67)      7742    (67)      7745    (67)      7748    (67)      7749    (67)
                                   7750    (67)      7751    (67)      7755    (67)      7756    (67)      7760    (67)
                                   7762    (67)      7763    (67)      7765    (67)      7766    (67)      7770    (67)
                                   7771    (67)      7774    (67)      7775    (67)      7779    (67)      7780    (67)
                                   7781    (67)      7784    (67)      7785    (67)      7789    (67)      7790    (67)
                                   7791    (67)      7792    (67)      7793    (67)      7823    (68)      7825    (68)
                                   7827    (68)      7828    (68)      7829    (68)      7830    (68)      7831    (68)
                                   7858    (69)      7860    (69)      7906    (70)      7907    (70)      7908    (70)
                                   7909    (70)      7913    (70)      7916    (70)      7921    (70)      7923    (70)
                                   7924    (70)      7932    (70)      7933    (70)      7938    (70)      7940    (70)
                                   7950    (70)      7953    (70)      7958    (70)      7959    (70)      7963    (70)
                                   7970    (70)      7973    (70)      7979    (70)      7980    (70)      7984    (70)
                                   7991    (70)      7993    (70)      7994    (70)      8030    (71)      8031    (71)
                                   8033    (71)      8035    (71)      8039    (71)      8040    (71)      8041    (71)
                                   8043    (71)      8044    (71)      8047    (71)      8051    (71)      8057    (71)
```

```
                                                            8059    (71)      8063    (71)      8069    (71)      8106    (72)      8107    (72)
                                                            8138    (73)      8139    (73)      8140    (73)      8142    (73)      8171    (74)
                                                            8173    (74)      8198    (75)      8200    (75)      8241    (76)      8246    (76)
                                                            8247    (76)      8251    (76)      8252    (76)      8253    (76)      8256    (76)
                                                            8260    (76)      8261    (76)      8262    (76)      8263    (76)      8267    (76)
                                                            8270    (76)      8273    (76)      8277    (76)      8283    (76)      8288    (76)
                                                            8289    (76)      8290    (76)      8291    (76)      8292    (76)      8293    (76)
                                                            8294    (76)      8296    (76)      8299    (76)      8304    (76)      8306    (76)
                                                            8307    (76)      8311    (76)      8312    (76)      8318    (76)      8319    (76)
                                                            8324    (76)      8329    (76)      8335    (76)      8338    (76)      8341    (76)
                                                            8344    (76)      8346    (76)      8350    (76)      8352    (76)      8356    (76)
                                                            8357    (76)      8359    (76)      8362    (76)      8363    (76)      8387    (77)
                                                            8389    (77)      8391    (77)      8416    (78)      8417    (78)      8419    (78)
                                                            8420    (78)      8421    (78)      8423    (78)      8424    (78)      8425    (78)
                                                            8426    (78)      8439    (79)      8441    (79)      8442    (79)      8443    (79)
                                                            8444    (79)      8446    (79)      8451    (79)      8452    (79)      8453    (79)
                                                            8455    (79)      8457    (79)      8460    (79)      8462    (79)      8464    (79)
                                                            8497    (80)      8507    (80)      8553    (81)      8557    (81)      8578    (81)
                                                            8581    (81)      8629    (82)      863     (4)       8631    (82)      8669    (83)
                                                            8671    (83)      8677    (83)      8678    (83)      8685    (83)      8722    (84)
                                                            8724    (84)      8730    (84)      8731    (84)      8738    (84)      8782    (85)
                                                            8783    (85)      8786    (85)      8789    (85)      8793    (85)      8804    (85)
                                                            8808    (85)      8809    (85)      8835    (86)      8837    (86)      8839    (86)
                                                            8841    (86)      8845    (86)      8849    (86)      8851    (86)      8916    (87)
                                                            8917    (87)      8921    (87)      8923    (87)      8926    (87)      8929    (87)
                                                            8930    (87)      8934    (87)      8935    (87)      8939    (87)      8940    (87)
                                                            8942    (87)      8944    (87)      8945    (87)      8946    (87)      8947    (87)
                                                            8951    (87)      8952    (87)      8953    (87)      8955    (87)      8956    (87)
                                                            8957    (87)      8958    (87)      8959    (87)      8964    (87)      8965    (87)
                                                            8967    (87)      8968    (87)      8972    (87)      8973    (87)      8974    (87)
                                                            8977    (87)      8978    (87)      8982    (87)      8983    (87)      8984    (87)
                                                            8987    (87)      8988    (87)      8992    (87)      8993    (87)      8994    (87)
                                                            9001    (87)      9004    (87)      9006    (87)      9007    (87)      9011    (87)
                                                            9012    (87)      9017    (87)      9019    (87)      9023    (87)      9028    (87)
                                                            9030    (87)      9035    (87)      9036    (87)      9039    (87)      9042    (87)
                                                            9043    (87)      9046    (87)      9049    (87)      9050    (87)      9055    (87)
                                                            9064    (87)      9072    (87)      9076    (87)      9081    (87)      9089    (87)
                                                            9090    (87)      9091    (87)      9097    (87)      9099    (87)      9138    (88)
                                                            9139    (88)      9140    (88)      9141    (88)      9143    (88)      9147    (88)
                                                            9149    (88)      9150    (88)      9151    (88)      9152    (88)      9154    (88)
                                                            9155    (88)      9157    (88)      9159    (88)      9165    (88)      9210    (89)
                                                            9211    (89)      9214    (89)      9216    (89)      9217    (89)      9223    (89)
                                                            9224    (89)      9231    (89)      9239    (89)      9244    (89)      9246    (89)
                                                            9248    (89)      9249    (89)      9261    (89)      9263    (89)      9272    (89)
                                                            9274    (89)      9286    (89)      9289    (89)      9291    (89)      9293    (89)
                                                            9317    (89)      9324    (89)      9326    (89)      9328    (89)      9330    (89)
                                                            9331    (89)      9337    (89)      9348    (89)      9352    (89)      9353    (89)
                                                            9355    (89)      9356    (89)      9396    (90)      9401    (90)      9408    (90)
                                                            9423    (90)      9431    (90)      9436    (90)      9443    (90)      9456    (90)
                                                            9485    (91)      9487    (91)      9519    (92)      9523    (92)      9524    (92)
                                                            9528    (92)      9537    (92)      9538    (92)      9540    (92)      9542    (92)
                                                            9571    (93)      9573    (93)      9574    (93)      9576    (93)      9577    (93)
                                                            9579    (93)      9583    (93)      9585    (93)      9587    (93)      9588    (93)
                                                            9589    (93)      9591    (93)      9595    (93)      9598    (93)      9628    (94)
                                                            9629    (94)      9633    (94)      9638    (94)      9640    (94)      9642    (94)
```

```
                                             9643    (94)    9652    (94)    9654    (94)    9655    (94)    9656    (94)
                                             9662    (94)    9665    (94)    9706    (95)    9726    (95)    9727    (95)
                                             9728    (95)    9770    (95)    9772    (95)    9781    (95)    9804    (95)
                                             9810    (95)    9834    (96)    9835    (96)    9836    (96)    9841    (96)
                                             9843    (96)    9845    (96)    9846    (96)    9864    (97)    9866    (97)
                                             9867    (97)    9891    (98)    9892    (98)    9893    (98)    9898    (98)
                                             9900    (98)    9902    (98)    9903    (98)
HIGH            =00000001     863    (4)
HYPHEN           000005CE-R  1946    (6)     3637    (23)
H_FROM_WBUS     =00000010     863    (4)
IB_FLAG         =00000020     863    (4)     7439    (66)
IB_KEY          =00000012     863    (4)
IF_ERROR         00002052-R  7272    (65)    1813    (6)
INCREMENT_INDEX  000024FA-R  8105    (72)    1825    (6)
INDEX_NAM_TBL    00000408-R  1837    (6)     8557    (81)
INIT_LOOP_VALUE  0000269B-R  8496    (80)    6007    (51)    6720    (59)
INIT_TEST_PC     000001B5-R  1606    (6)     2369    (7)     3360    (20)    3391    (20)    3416    (20)    3814    (24)
                                             3821    (24)    3828    (24)    3830    (24)    5922    (50)    6440    (56)
                                             6463    (56)    6554    (57)    6684    (58)    6837    (61)    6850    (61)
                                             6926    (62)    7016    (63)    7030    (63)    7142    (64)    7830    (68)
                                             7916    (70)    7940    (70)    7963    (70)    7984    (70)    8047    (71)
                                             8059    (71)    8142    (73)    8389    (77)
INIT_THE_WORLD   00001CBF-R  6110    (52)    1814    (6)
INSERT_FIELD     00002966-R  9209    (89)    1403    (5)     3258    (20)    6508    (56)    7994    (70)    9841    (96)
                                             9898    (98)
INSERT_FIELD_X   00002A1B-R  9357    (89)    9352    (89)    9355    (89)
INSTR_FLAG      =00000004     863    (4)     5235    (45)    5785    (49)    9156    (88)
INSTR_KEY       =00000018     863    (4)     5236    (45)
INSTR_TABLE      000003CE-R  1804    (6)     5849    (49)
INS_FLD_BUFF     00000C90-R  2181    (6)     9210    (89)    9216    (89)    9224    (89)    9272    (89)    9337    (89)
INT             =00000010     863    (4)
INVALID_COMMAND  000002F3-R  1695    (6)     4231    (34)
IRD             =00000005     863    (4)
I_INDEX_TBL      0000040E-R  1840    (6)     1837    (6)     8497    (80)
J_INDEX_TBL      00000415-R  1844    (6)     1838    (6)
KEYWORD_LIST     00000343-R  1725    (6)     4334    (35)
K_INDEX_TBL      0000041C-R  1845    (6)     1839    (6)
L               =00000005     856    (3)     3361    (20)    3363    (20)    3417    (20)    3419    (20)    3718    (23)
                                             3822    (24)    3823    (24)    3824    (24)    3968    (27)    4239    (34)
                                             4247    (34)    5814    (49)    6274    (54)    6286    (54)    6308    (54)
                                             6570    (57)    7965    (70)    7966    (70)    7986    (70)    7988    (70)
                                             8254    (76)    8624    (82)    9590    (93)    9792    (95)    9793    (95)
LITERAL         =00000001    9902    (98)    1374    (5)     1385    (5)     1393    (5)     2269    (7)     2270    (7)
                                             2314    (7)     2361    (7)     2362    (7)     2378    (7)     2379    (7)
                                             2380    (7)     2381    (7)     2568    (10)    2569    (10)    2570    (10)
                                             2571    (10)    2572    (10)    2708    (12)    2710    (12)    2807    (14)
                                             2893    (15)    2942    (16)    3207    (20)    3230    (20)    3252    (20)
                                             3253    (20)    3260    (20)    3262    (20)    3267    (20)    3305    (20)
                                             3307    (20)    3313    (20)    3314    (20)    3323    (20)    3358    (20)
                                             3369    (20)    3403    (20)    3521    (22)    3606    (23)    3627    (23)
                                             3637    (23)    3650    (23)    3661    (23)    3679    (23)    3680    (23)
                                             3811    (24)    3812    (24)    3813    (24)    3817    (24)    3856    (24)
                                             3862    (24)    3863    (24)    3894    (25)    4009    (28)    4011    (28)
                                             4040    (29)    4042    (29)    4122    (32)    4129    (32)    4137    (32)
                                             4157    (33)    4209    (34)    4210    (34)    4217    (34)    4218    (34)
```

```
                                          4219   (34)    4231   (34)    4241   (34)    4251   (34)    4254   (34)
                                          4265   (34)    4272   (34)    4309   (35)    4311   (35)    4313   (35)
                                          4334   (35)    4616   (39)    4652   (39)    4672   (39)    4685   (39)
                                          4954   (41)    4957   (41)    4983   (41)    4987   (41)    4994   (41)
                                          5004   (41)    5480   (47)    5481   (47)    5575   (47)    5589   (47)
                                          5591   (47)    5601   (47)    5617   (47)    5627   (47)    5639   (47)
                                          5645   (47)    5827   (49)    5849   (49)    5857   (49)    6171   (53)
                                          6428   (56)    6494   (56)    6503   (56)    7036   (63)    7050   (63)
                                          7131   (64)    7297   (65)    7320   (65)    7426   (66)    7490   (66)
                                          7491   (66)    7507   (66)    7523   (66)    7529   (66)    7629   (67)
                                          7680   (67)    7681   (67)    7682   (67)    7686   (67)    7687   (67)
                                          7693   (67)    7726   (67)    7735   (67)    7736   (67)    7737   (67)
                                          7738   (67)    7739   (67)    7749   (67)    7750   (67)    7756   (67)
                                          7760   (67)    7761   (67)    7771   (67)    7780   (67)    7781   (67)
                                          7790   (67)    7791   (67)    7924   (70)    7933   (70)    8069   (71)
                                          8247   (76)    8260   (76)    8261   (76)    8262   (76)    8267   (76)
                                          8346   (76)    8442   (79)    8443   (79)    8452   (79)    8455   (79)
                                          8460   (79)    8497   (80)    8557   (81)    8671   (83)    8676   (83)
                                          8724   (84)    8729   (84)    8782   (85)    8783   (85)    8845   (86)
                                          8916   (87)    8917   (87)    8930   (87)    8934   (87)    8935   (87)
                                          8939   (87)    8945   (87)    8946   (87)    8947   (87)    8951   (87)
                                          8952   (87)    8956   (87)    8957   (87)    8958   (87)    8963   (87)
                                          8973   (87)    8983   (87)    8993   (87)    9011   (87)    9042   (87)
                                          9043   (87)    9049   (87)    9076   (87)    9081   (87)    9090   (87)
                                          9091   (87)    9099   (87)    9139   (88)    9140   (88)    9150   (88)
                                          9224   (89)    9249   (89)    9259   (89)    9272   (89)    9287   (89)
                                          9337   (89)    9396   (90)    9431   (90)    9519   (92)    9628   (94)
                                          9629   (94)    9633   (94)    9652   (94)    9655   (94)    9656   (94)
                                          9665   (94)    9706   (95)    9726   (95)    9727   (95)    9728   (95)
                                          9835   (96)    9836   (96)    9843   (96)    9845   (96)    9864   (97)
                                          9867   (97)    9892   (98)    9893   (98)    9900   (98)    9902   (98)
LOAD_DCS       00001C0A-R  5904   (50)    1804   (6)
LOAD_FIELD     00002437-R  7905   (70)    1823   (6)
LOAD_INDEX     000024B8-R  8029   (71)    1824   (6)
LOAD_MEMSCAR   00000228-R  1638   (6)     4009   (28)    4040   (29)
LOAD_MEMSCR    00000249-R  1641   (6)
LOAD_REG       00001DF8-R  6543   (57)    1815   (6)
LOG_ADDRESS    000004E5-R  1904   (6)     8289   (76)    8291   (76)    8293   (76)    8304   (76)    8307   (76)
                                          8312   (76)    8335   (76)    8338   (76)    8341   (76)
LOG_EXIT       00002607-R  8364   (76)    8357   (76)
LOG_FAIL       000025EF-R  8351   (76)    8263   (76)    8283   (76)
LOG_MSG_1      000004E9-R  1906   (6)     8261   (76)
LOG_MSG_2      00000507-R  1907   (6)     8455   (79)
LOG_MSG_3      00000533-R  1908   (6)     8452   (79)
LOG_SUCCESS    000025FC-R  8358   (76)    8324   (76)    8350   (76)
LOG_SYNDROME   000004E8-R  1905   (6)     8296   (76)    8319   (76)    8344   (76)
LONG          =00000004    863    (4)
LOOP_ACTION    00001848-R  5097   (44)    1702   (6)
LOOP_ADDRESS   00000423-R  1851   (6)     6051   (51)    6069   (51)
LOOP_ERROR_FLAG=00000020   863    (4)     6055   (51)    6073   (51)    6264   (54)    6625   (58)    6941   (62)
                                          6950   (62)    7071   (63)    7080   (63)    7185   (64)    7195   (64)
                                          7286   (65)    7334   (65)    8354   (76)    8361   (76)    9597   (93)
LOOP_FLAG     =00000002    863    (4)     4755   (40)    5099   (44)    6055   (51)    7312   (65)    7449   (66)
                                          7543   (66)
LOOP_KEY      =0000000A    863    (4)
```

```
LOST_FLAG       =00000010     863    (4)      3666    (23)     3786    (24)     4761    (40)     4846    (40)     5791    (49)
                                              6028    (51)     6035    (51)     6063    (51)

LOST_KEY        =0000001A     863    (4)
LOW             =00000000     863    (4)
M               =00000006     856    (3)      2628    (11)     2821    (14)     2833    (14)     2897    (15)     2899    (15)
                                              2946    (16)     2948    (16)     3212    (20)     3217    (20)     3243    (20)
                                              3269    (20)     3292    (20)     3324    (20)     3349    (20)     3370    (20)
                                              3393    (20)     3395    (20)     3524    (22)     3610    (23)     3632    (23)
                                              3634    (23)     3642    (23)     3645    (23)     3685    (23)     3699    (23)
                                              3822    (24)     4266    (34)     4269    (34)     4344    (35)     4349    (35)
                                              4356    (35)     4435    (36)     4514    (37)     4523    (37)     4570    (38)
                                              4628    (39)     4690    (39)     4782    (40)     4843    (40)     4847    (40)
                                              4851    (40)     4859    (40)     4880    (40)     4989    (41)     5152    (45)
                                              5170    (45)     5174    (45)     5183    (45)     5205    (45)     5212    (45)
                                              5215    (45)     5224    (45)     5243    (45)     5248    (45)     5251    (45)
                                              5277    (45)     5285    (45)     5289    (45)     5350    (46)     5375    (46)
                                              5380    (46)     5384    (46)     5393    (46)     5413    (46)     5417    (46)
                                              5487    (47)     5521    (47)     5525    (47)     5532    (47)     5536    (47)
                                              5582    (47)     5586    (47)     5825    (49)     5828    (49)     5836    (49)
                                              5852    (49)     5855    (49)     6174    (53)     6200    (53)     6205    (53)
                                              6217    (53)     6220    (53)     6343    (54)     6345    (54)     6448    (56)
                                              6477    (56)     6495    (56)     6582    (57)     6859    (61)     7040    (63)
                                              7054    (63)     7162    (64)     7170    (64)     7322    (65)     7326    (65)
                                              7926    (70)     7965    (70)     7986    (70)     8075    (71)     8079    (71)
                                              8109    (72)     8111    (72)     8144    (73)     8146    (73)     8274    (76)
                                              8281    (76)     8305    (76)     8309    (76)     8315    (76)     8323    (76)
                                              8336    (76)     8339    (76)     8342    (76)     8345    (76)     8347    (76)
                                              8349    (76)     8448    (79)     8449    (79)     8502    (80)     8504    (80)
                                              8564    (81)     8566    (81)     8572    (81)     8574    (81)     8680    (83)
                                              8733    (84)     8788    (85)     8803    (85)     8999    (87)     9062    (87)
                                              9070    (87)     9078    (87)     9085    (87)     9087    (87)     9227    (89)
                                              9253    (89)     9313    (89)     9343    (89)     9400    (90)     9417    (90)
                                              9425    (90)     9435    (90)     9450    (90)     9458    (90)     9520    (92)
                                              9529    (92)     9636    (94)     9714    (95)     9716    (95)     9719    (95)
                                              9730    (95)     9736    (95)     9742    (95)     9748    (95)     9768    (95)
                                              9801    (95)
M$TEST_SIZE     =000007A2     863    (4)      863     (4)
M$TEST_SIZE_D   =000003E2     863    (4)
MA0             =00000008     863    (4)
MA0_NAME         000000FD-R   1493   (6)      1478    (6)
MA1             =0000000A     863    (4)
MA1_NAME         00000113-R   1495   (6)      1480    (6)
MA2             =0000000C     863    (4)
MA2_NAME         00000129-R   1497   (6)      1482    (6)
MAD             =00000013     863    (4)
MAIN32          =0000741D     863    (4)
MAINT_A_TO_CMI  =00000080     863    (4)
MAINT_CMI_TO_MH =00000020     863    (4)
MAINT_DCS_ENABL =00000004     863    (4)      1423    (5)      3061    (18)     3067    (18)     3120    (19)     3126    (19)
                                              4067    (30)     6119    (52)

MAINT_MD_TO_CMI =00000040     863    (4)
MAINT_REG       =0000741D     863    (4)      1424    (5)      3001    (17)     3062    (18)     3068    (18)     3121    (19)
                                              3127    (19)     4068    (30)     6120    (52)     863     (4)

MAINT_STB_INH   =00000002     863    (4)      3061    (18)     3120    (19)
MAINT_TRAP_OFF  =00000001     863    (4)      1423    (5)      3061    (18)     3067    (18)     3120    (19)     3126    (19)
```

```
                                              4067     (30)

MAINT_WB_TO_WH  =00000008      863    (4)
MAINT_WD_TO_WB  =00000010      863    (4)
MAP             =00000012      863    (4)
MAREG           =00007410      863    (4)
MASK             00001ECD-R    6824   (61)    1816     (6)
MASK_DISPLAY     00001B6C-R    5636   (47)    5595     (47)     5611     (47)     5621     (47)
MASTER_HALT_EN  =00000008      863    (4)     1420     (5)      2905     (15)     2956     (16)     4978     (41)     6127     (52)
                                              7699     (67)
MAX_ARGUMENTS   =00000010      863    (4)     1803     (6)      5829     (49)     5831     (49)
MA_KEY          =00000038      863    (4)     5504     (47)
MC0             =00000003      863    (4)
MC0_NAME         000000C6-R    1488   (6)     1473     (6)
MC1             =00000009      863    (4)
MC1_NAME         00000108-R    1494   (6)     1479     (6)
MC2             =0000000B      863    (4)
MC2_NAME         0000011E-R    1496   (6)     1481     (6)
MDL             =0000001B      863    (4)
MDL_NAME         000009EF-R    2073   (6)     1995     (6)
MDR             =00000018      863    (4)
MDREG           =00007414      863    (4)
MDR_NAME         0000087A-R    2036   (6)     1992     (6)
MD_KEY          =00000034      863    (4)     5500     (47)
MD_REG_BYTE_0   =00007414      863    (4)     863      (4)
MD_REG_BYTE_1   =00007415      863    (4)
MD_REG_BYTE_2   =00007416      863    (4)
MD_REG_BYTE_3   =00007417      863    (4)
MEC             =0000001A      863    (4)
MEC_NAME         000009C0-R    2066   (6)     1994     (6)
MESSAGE_LENGTH   0000016E-R    1553   (6)     2629     (11)     2634     (11)     4128     (32)
MESSAGE_POINTER  0000016C-R    1552   (6)     2269     (7)      2270     (7)      2361     (7)      2362     (7)      2378     (7)
                                              2379     (7)      2381     (7)      2568     (10)     2569     (10)     2571     (10)
                                              2627     (11)     2631     (11)     2634     (11)     2708     (12)     2710     (12)
                                              3636     (23)     3637     (23)     3649     (23)     3650     (23)     3811     (24)
                                              3813     (24)     3815     (24)     3817     (24)     3856     (24)     3863     (24)
                                              4128     (32)     4129     (32)     4137     (32)     4209     (34)     4210     (34)
                                              4217     (34)     4218     (34)     4231     (34)     4954     (41)     4957     (41)
                                              4983     (41)     5004     (41)     7490     (66)     7491     (66)     7523     (66)
                                              7680     (67)     7681     (67)     7686     (67)     7687     (67)     7726     (67)
                                              7728     (67)     7736     (67)     7738     (67)     7739     (67)     7750     (67)
                                              7760     (67)     7780     (67)     7790     (67)     8260     (76)     8261     (76)
                                              8262     (76)     8391     (77)     8442     (79)     8452     (79)     8455     (79)
                                              8460     (79)     8845     (86)     8916     (87)     8917     (87)     8934     (87)
                                              8939     (87)     8946     (87)     8947     (87)     8951     (87)     8956     (87)
                                              8957     (87)     9042     (87)     9043     (87)     9076     (87)     9089     (87)
                                              9090     (87)     9091     (87)     9099     (87)     9139     (88)     9140     (88)
                                              9628     (94)     9629     (94)     9633     (94)     9654     (94)     9655     (94)
                                              9656     (94)     9665     (94)     9810     (95)     9864     (97)     9867     (97)
MHREG           =00007418      863    (4)
MH_REG_BYTE_0   =00007418      863    (4)     8294     (76)     863      (4)
MH_REG_BYTE_1   =00007419      863    (4)
MH_REG_BYTE_2   =0000741A      863    (4)
MH_REG_BYTE_3   =0000741B      863    (4)
MIC             =00000001      863    (4)
MICROWORD       =0000000A      863    (4)
```

```
MICRO_ADDR_INH =00000080      863     (4)      3892    (25)     3958    (27)     3977    (27)     4097    (31)     4102    (31)
                                               6783    (60)
MICRO_CODE_HEAD=000001BA-R    1615    (6)      1668    (6)
MICRO_CODE_TAIL=000002B7-R    1667    (6)      1668    (6)
MIC_INSTR_CNT   000001B1-R    1587    (6)      2891    (15)     3278    (20)     3308    (20)     3317    (20)     3333    (20)
                                               3971    (27)     4012    (28)     4043    (29)     4099    (31)     5576    (47)
                                               5592    (47)     5602    (47)     5618    (47)     5628    (47)     7575    (66)
                                               9846    (96)     9903    (98)
MIC_NAME        000000B0-R    1486    (6)      1471    (6)
MIC_TICK_CNT    000001B2-R    1593    (6)      2940    (16)     7577    (66)
MME_DIS_ADR     00000280-R    1659    (6)
MME_DIS_DATA    00000280-R    1660    (6)
MODE_TYPE       00000448-R    1873    (6)      6934    (62)     7064    (63)     7127    (64)     8953    (87)
MODULE_LST      000000A5-R    1484    (6)
MODULE_MSG      00000606-R    1950    (6)      9076    (87)
MODULE_NAM_TBL  0000008B-R    1470    (6)      3627    (23)     9081    (87)
MONITOR_SIZE   =00002C5E      863     (4)      863     (4)
MSQ            =00000008      863     (4)
MTEMP_COMMON    00001A41-R    5537    (47)     5545    (47)     5551    (47)     5557    (47)     5563    (47)     5569    (47)
MT_KEY         =0000002E      863     (4)      5494    (47)
MULTIPLY_A_BC   000026D6-R    8622    (82)     5911    (50)     5918    (50)     6427    (56)     6547    (57)     6830    (61)
                                               6846    (61)     6922    (62)     7012    (63)     7026    (63)     7909    (70)
                                               8043    (71)     8246    (76)     9804    (95)
M_CLK_ADDR      00000455-R    1891    (6)      7431    (66)     7621    (67)     7626    (67)
NER_FLAG       =00000004      863     (4)      4755    (40)     5099    (44)     7299    (65)
NER_KEY        =0000000C      863     (4)
NEW_TEST        00001C3B-R    5992    (51)     1817    (6)
NOERROR        =00000001      863     (4)
NOP_MICRO_WORD  000001BA-R    1616    (6)      1393    (5)      3262    (20)     4011    (28)     4042    (29)     4157    (33)
                                               5480    (47)     5591    (47)     9845    (96)     9902    (98)
NOTEQUAL       =00000003      863     (4)      6632    (58)
NOT_MSG         000005C0-R    1944    (6)      8956    (87)
NO_MICRO_WORDS =00000017      1668    (6)      1394    (5)
NUMBER_OF_FILES=00000005      1546    (6)      2341    (7)
ONE             000002F2-R    1694    (6)      2266    (7)      2753    (13)     4275    (34)     7562    (66)     9151    (88)
                                               9165    (88)
ONEQUAL        =00000002      863     (4)      6630    (58)     6672    (58)     6679    (58)
ONERROR        =00000000      863     (4)      6634    (58)
OPEN_TU58       00000E2F-R    2517    (9)      2329    (7)      2391    (7)
OP_BEGIN_LOOP  =00000008      863     (4)
OP_BURST_CLOCK =00000002      863     (4)
OP_CODE_DISP    00001B99-R    5780    (49)     2374    (7)      5857    (49)
OP_COMPARE_REG =00000004      863     (4)
OP_COMPARE_REGM=00000006      863     (4)
OP_COMPARE_VB  =00000020      863     (4)
OP_DUMPLOG     =00000030      863     (4)
OP_ENABL_STALL =00000038      863     (4)
OP_ENABL_TRAP  =0000002E      863     (4)
OP_END_FILE    =00000024      863     (4)      3393    (20)
OP_END_LOOP    =0000000A      863     (4)
OP_END_TEST    =0000000C      863     (4)
OP_ERRLOG      =00000032      863     (4)
OP_ERROR_LOOP  =0000000E      863     (4)
OP_FETCH       =00000022      863     (4)
OP_IF_ERROR    =00000012      863     (4)
```

```
OP_INC_INDEX    =0000002A    863    (4)
OP_INIT         =00000014    863    (4)
OP_LOAD_DCS     =00000000    863    (4)
OP_LOAD_FIELD   =00000026    863    (4)
OP_LOAD_INDEX   =00000028    863    (4)
OP_LOAD_REG     =00000016    863    (4)
OP_MASK         =00000018    863    (4)
OP_NEW_TEST     =0000001A    863    (4)
OP_PATTERN_GEN  =00000010    863    (4)
OP_PRINT_N      =00000036    863    (4)
OP_PRINT_S      =00000034    863    (4)
OP_SAVE_INDEX   =0000002C    863    (4)
OP_SKIP         =0000001C    863    (4)
OP_SUB_TEST     =0000001E    863    (4)
P0_PARITY_MSK    00000570-R  1915   (6)       9396   (90)
P1_PARITY_MSK    0000057A-R  1916   (6)       9431   (90)
PARITY_ERROR    =00000020    863    (4)
PARSE_DONE       0000161E-R  4390   (35)      4330   (35)
PARSE_ERR        0000161C-R  4384   (35)      4318   (35)      4336   (35)      4362   (35)      4364   (35)
PARSE_HEX_DONE   000016CB-R  4684   (39)      4630   (39)      4632   (39)
PARSE_HEX_ERR    000016E3-R  4703   (39)      4635   (39)      4642   (39)      4644   (39)
PARSE_HEX_NUMB   00001679-R  4615   (39)      4363   (35)      4376   (35)
PARSE_HEX_TEMP   00000388-R  1767   (6)       4616   (39)      4652   (39)      4672   (39)      4685   (39)
PARSE_TABLE      0000031A-R  1712   (6)       1711   (6)       4237   (34)      4254   (34)      4309   (35)
PARSE_TBL_PTR    00000318-R  1711   (6)       4310   (35)      4569   (38)      4572   (38)      4687   (39)      4695   (39)
PAR_CHK_ENABL   =00000008    863    (4)       3892   (25)      3958   (27)      3977   (27)      4097   (31)      4102   (31)
PAR_STOP_ENABL  =00000010    863    (4)
PASS_COUNT       00000062-R  1462   (6)       2272   (7)       2380   (7)       2382   (7)       7858   (69)      7860   (69)
PASS_KEY        =00000002    863    (4)       4792   (40)
PASS_SUB_ACTION  00001797-R  4878   (40)      4793   (40)
PATTERN_ADDRESS  00000427-R  1858   (6)       6428   (56)
PATTERN_GEN      00001D8F-R  6420   (56)      1812   (6)
PB_INIT         =00000040    863    (4)
PB_KEY          =00000036    863    (4)       5502   (47)
PC_KEY          =00000030    863    (4)       5496   (47)
PHB             =00000007    863    (4)
PRIM_REV        =00000008    4      (1)
PRINT_N          00002617-R  8415   (78)      1831   (6)
PRINT_N_EX       00002645-R  8427   (78)      8421   (78)      8425   (78)
PRINT_S          00002608-R  8386   (77)      1830   (6)
PRK             =0000000C    863    (4)
PROGRAM_INIT     00000CF1-R  2260   (7)       1431   (5)       4272   (34)
PROGRAM_TITLE    00000067-R  1468   (6)       2270   (7)
PROG_CTRL_REG    0000005D-R  1453   (6)       3808   (24)      4753   (40)      4757   (40)      4955   (41)      4958   (41)
                                              5098   (44)      5101   (44)      5190   (45)      5192   (45)      5403   (46)
                                              5405   (46)      7291   (65)      7298   (65)      7304   (65)      7310   (65)
                                              7438   (66)      7447   (66)      7542   (66)      9571   (93)
PROMPT           000002EC-R  1692   (6)       4210   (34)
PS_KEY          =0000003E    863    (4)       5510   (47)
PUT_D_IN_TBL     0000166E-R  4567   (38)      4322   (35)      4329   (35)      4340   (35)
QA_COUNT         00000449-R  1875   (6)       5203   (45)      5204   (45)      5213   (45)      5216   (45)      9574   (93)
                                              9577   (93)      9583   (93)
QA_FLAG         =00000040    863    (4)       5197   (45)      9572   (93)
QA_INDEX         0000044B-R  1876   (6)       6918   (62)      6920   (62)      7008   (63)      7010   (63)      7151   (64)
                                              7153   (64)      9585   (93)      9589   (93)
```

```
QA_KEY          =00000014        863     (4)
QV_FLAG         =00000002        863     (4)       3464    (21)      4767    (40)      4822    (40)
QV_KEY          =00000028        863     (4)       4796    (40)
QV_SUB_ACTION    0000174F-R      4820    (40)      4797    (40)
RDCTRL          =0000741E        863     (4)
RDM             =00000004        863     (4)
RDM_FLAG        =00000004        863     (4)       2278    (7)       4767    (40)      4814    (40)
RDM_INDEX       =00000004        1543    (6)       2283    (7)
RDM_KEY         =0000002A        863     (4)       4798    (40)
RDM_LAST        =00000008        863     (4)       2281    (7)       2289    (7)       2295    (7)
RDM_M_MIC_WORD   000001DB-R      1619    (6)       9892    (98)      9900    (98)
RDM_NAME         000000D1-R      1489    (6)       1474    (6)
RDM_PS_MIC_WORD  00000207-R      1623    (6)       5601    (47)
RDM_R_MIC_WORD   000001D0-R      1618    (6)       9835    (96)      9843    (96)
RDM_SF_MIC_WORD  0000021D-R      1625    (6)       5627    (47)
RDM_SR_MIC_WORD  000001F1-R      1621    (6)       5589    (47)
RDM_ST_MIC_WORD  00000212-R      1624    (6)       5617    (47)
RDM_SUB_ACTION   00001744-R      4812    (40)      4799    (40)
RDM_TAIL        =0000016C-R      1541    (6)       1543    (6)
RDM_WD_MIC_WORD  000001E6-R      1620    (6)       5575    (47)
RD_CTRL_REG     =0000741E        863     (4)       1419    (5)       1421    (5)       2889    (15)      2906    (15)      2939    (16)
                                                   2957    (16)      3005    (17)      4078    (30)      4080    (30)      4979    (41)
                                                   4981    (41)      6128    (52)      6571    (57)      7697    (67)      7700    (67)
                                                   863     (4)
READ_MTEMP       00002C2C-R      9890    (98)      5538    (47)
READ_OVERLAY     0000100E-R      3206    (20)      2368    (7)       3387    (20)      7829    (68)
READ_RTEMP       00002BC3-R      9833    (96)      5526    (47)      9007    (87)
READ_R_TEMP      00000C9D-R      2189    (6)       9834    (96)      9836    (96)      9891    (98)      9893    (98)
READ_TU58        00000DFC-R      2462    (8)       3207    (20)      3230    (20)      3369    (20)      3403    (20)      3606    (23)
                                                   3661    (23)
READ_V_BUS       000011FB-R      3519    (22)      4982    (41)      7139    (64)
REMOTE          =00000001        863     (4)       1418    (5)       2888    (15)      2903    (15)      2938    (16)      2954    (16)
                                                   3004    (17)      4076    (30)      4976    (41)      6117    (52)      6125    (52)
                                                   6575    (57)      7696    (67)
RETURN_ACTION    00001845-R      5078    (43)      1699    (6)
RETURN_TO_RDM    00000F81-R      2996    (17)      1377    (5)       2636    (11)      2748    (13)      5079    (43)
RING_BELL        00002743-R      8834    (86)      7293    (65)
RT_KEY          =0000002C        863     (4)       5492    (47)
R_RDM_MIC_WORD   000001C5-R      1617    (6)       3252    (20)      3260    (20)
SAC             =00000009        863     (4)
SAVED_SP         0000005E-R      1459    (6)       1376    (5)       1389    (5)       2267    (7)       3006    (17)
SAVE_BAD_DATA    000026E3-R      8668    (83)      6911    (62)      7001    (63)
SAVE_GOOD_DATA   000026FE-R      8721    (84)      6929    (62)      7019    (63)
SAVE_INDEX       00002505-R      8137    (73)      1826    (6)
SA_CLOCK        =00000080        863     (4)
SA_FLAG         =00000010        863     (4)       7315    (65)      7452    (66)
SA_KEY          =00000010        863     (4)
SA_START_STOP   =00000080        863     (4)
SBE_FLAG        =00000020        863     (4)       4767    (40)      4806    (40)      8440    (79)
SBE_KEY         =00000042        863     (4)       4800    (40)
SCRAMBLE_FLAG    00000C9C-R      2184    (6)       9214    (89)      9244    (89)      9353    (89)
SEARCH_DONE      00001648-R      4468    (36)      4460    (36)
SEARCH_ERR       0000164C-R      4476    (36)      4434    (36)
SEARCH_LIST      00001621-R      4429    (36)      4317    (35)      4335    (35)
SEC_REV         =00000007        5       (1)
```

```
SELECT_DCS       00001392-R  3914  (26)      2847   (14)    3054   (18)    3113   (19)
SETCLR_CF_ADDR   000001B3-R  1599  (6)       3045   (18)    3105   (19)    5171   (45)    5414   (46)    7425   (66)
SETCLR_CF_DATA   000001B4-R  1600  (6)       3048   (18)    3108   (19)    5176   (45)    5419   (46)    7462   (66)
                                             7465   (66)    7473   (66)    7476   (66)
SET_ACTION       00001863-R  5148  (45)      1700   (6)
SET_DCS_CF       00000F9F-R  3040  (18)      5177   (45)    7463   (66)    7474   (66)
SET_DONE         00001940-R  5295  (45)      5178   (45)    5187   (45)    5209   (45)    5258   (45)    5279   (45)
SET_ERR          00001943-R  5302  (45)      5156   (45)    5164   (45)    5189   (45)    5228   (45)    5237   (45)
SFLAG_SUB_ACT    00001899-R  5182  (45)      5158   (45)    5198   (45)    5207   (45)    5218   (45)
SF_KEY          =00000040    863   (4)       5514   (47)
SHOW_ACTION      000017AD-R  4944  (41)      1697   (6)
SHOW_MESSAGE_1   00000391-R  1787  (6)       4954   (41)
SHOW_MESSAGE_2   0000039D-R  1788  (6)       4957   (41)
SHOW_TEMP        000003AB-R  1789  (6)       4993   (41)    4994   (41)
SINGLE_INSTR     00002906-R  9137  (88)      5818   (49)
SINGLE_MIC_INST  00000F30-R  2883  (15)      3278   (20)    3308   (20)    3317   (20)    3333   (20)    3971   (27)
                                             4012   (28)    4043   (29)    4099   (31)    5576   (47)    5592   (47)
                                             5602   (47)    5618   (47)    5628   (47)    7575   (66)    9846   (96)
                                             9903   (98)
SINGLE_TICK      00000F57-R  2936  (16)      7577   (66)
SKIP             00001E31-R  6623  (58)      1818   (6)
SKIP_SPACES      0000164E-R  4513  (37)      4326   (35)    4347   (35)    4359   (35)    4368   (35)    4519   (37)
SOFT_CTRL_REG    00000059-R  1443  (6)       2749   (13)    2751   (13)    3665   (23)    3785   (24)    4201   (34)
                                             4204   (34)    4265   (34)    4758   (40)    4762   (40)    4829   (40)
                                             4831   (40)    4870   (40)    4872   (40)    4894   (40)    4896   (40)
                                             5056   (42)    5058   (42)    5102   (44)    5104   (44)    5790   (49)
                                             5996   (51)    6017   (51)    6026   (51)    6029   (51)    6034   (51)
                                             6048   (51)    6061   (51)    6064   (51)    6771   (60)    6773   (60)
                                             7279   (65)    7432   (66)    7434   (66)    7478   (66)    7480   (66)
                                             7511   (66)    7569   (66)    7571   (66)    7593   (67)    7595   (67)
                                             7603   (67)    7606   (67)    7607   (67)    7615   (67)    7639   (67)
                                             7648   (67)
SOFT_CTRL_REGA   0000005A-R  1444  (6)       4069   (30)    5253   (45)    5257   (45)    5264   (45)    5266   (45)
                                             5365   (46)    5367   (46)    5784   (49)    6011   (51)    6013   (51)
                                             6053   (51)    6056   (51)    6071   (51)    6074   (51)    6263   (54)
                                             6265   (54)    6624   (58)    6724   (59)    6726   (59)    6939   (62)
                                             6943   (62)    6948   (62)    6951   (62)    7069   (63)    7073   (63)
                                             7078   (63)    7081   (63)    7183   (64)    7186   (64)    7192   (64)
                                             7196   (64)    7273   (65)    7285   (65)    7333   (65)    7335   (65)
                                             7444   (66)    7484   (66)    7538   (66)    7572   (66)    7582   (66)
                                             7585   (66)    7649   (67)    7667   (67)    8171   (74)    8173   (74)
                                             8352   (76)    8356   (76)    8359   (76)    8362   (76)    9028   (87)
                                             9155   (88)    9157   (88)    9595   (93)    9598   (93)
SOFT_CTRL_REGB   0000005B-R  1445  (6)       2276   (7)     2282   (7)     2296   (7)     3463   (21)    4763   (40)
                                             4768   (40)    4805   (40)    4807   (40)    4813   (40)    4815   (40)
                                             4821   (40)    4823   (40)    7661   (67)    8198   (75)    8200   (75)
                                             8270   (76)    8277   (76)    8439   (79)    8444   (79)    8462   (79)
                                             8464   (79)
SOMM_ADDRESS     0000038F-R  1781  (6)       3535   (22)    3917   (26)    3983   (27)    5287   (45)    5291   (45)
                                             5382   (46)    5386   (46)    6789   (60)    7682   (67)
SOMM_FLAG       =00000001    863   (4)       4070   (30)    5265   (45)    5366   (46)    7668   (67)
SOMM_KEY        =00000006    863   (4)       5161   (45)    5359   (46)
SOMM_MSG         00000464-R  1893  (6)       7681   (67)
SPA             =00000002    863   (4)
SPEC_TEST_NUMB   0000038C-R  1773  (6)       3677   (23)    3791   (24)    4844   (40)    6042   (51)
```

```
SPEC_TEST_PC    000003BA-R    1800    (6)      5242    (45)     5249    (45)     5252    (45)     5806    (49)
SRK            =00000001      863     (4)
SRM            =00000017      863     (4)
SRM_NAME        000007CC-R    2025    (6)      1991    (6)
SR_KEY         =0000003C      863     (4)      5508    (47)
SSOMM_SUB_ACT   00001916-R    5263    (45)     5162    (45)
SSTEP_SUB_ACT   000018D6-R    5223    (45)     5160    (45)
STACK_SIZE     =00000400      863     (4)      863     (4)
START1          0000000E-R    1384    (5)      1373    (5)
STARTING_RECORD 00000064-R    1464    (6)      2301    (7)      2332    (7)
START_EXECUTION 0000139F-R    3948    (27)     3278    (20)     3308    (20)     3317    (20)     3333    (20)     4012    (28)
                                               4043    (29)     5576    (47)     5592    (47)     5602    (47)     5618    (47)
                                               5628    (47)     6775    (60)     9846    (96)     9903    (98)
STATUS         =00007406      863     (4)
STATUS_REG     =00007406      863     (4)      3522    (22)     863     (4)
STEP_ADDRESS    00000450-R    1887    (6)      7501    (66)     7506    (66)     7550    (66)
STEP_KEY       =0000001E      863     (4)      5159    (45)     863     (4)
STOP_CLOCK     =00000004      863     (4)      7464    (66)     7472    (66)     7674    (67)
STROBE_CMI     =00000040      863     (4)
ST_KEY         =0000001E      863     (4)      5512    (47)
SUB_TEST        00001E94-R    6716    (59)     1819    (6)
SUB_TEST_MSG    000005AC-R    1942    (6)      8939    (87)
SUB_TEST_NUMB   00000061-R    1461    (6)      6003    (51)     6731    (59)     8940    (87)     8944    (87)     8945    (87)
TAB             000005C5-R    1945    (6)      7760    (67)     9864    (97)
TEMP_BUFFER     000001A3-R    1573    (6)      2699    (12)     2702    (12)     2707    (12)     2708    (12)     2710    (12)
                                               6494    (56)     6503    (56)     7924    (70)     7933    (70)
TERM_ERROR      0000018E-R    1566    (6)      4218    (34)
TERM_ERR_CODE   0000018D-R    1565    (6)      4216    (34)     4219    (34)
TERM_INP_BUFF   0000016F-R    1559    (6)      2266    (7)      2753    (13)     4212    (34)     4226    (34)     4275    (34)
                                               4311    (35)     7562    (66)     7563    (66)     9151    (88)     9152    (88)
                                               9165    (88)
TERM_INTERRUPT  00000ED5-R    2744    (13)     2266    (7)      2753    (13)     4275    (34)     9165    (88)
TEST           =00000004      863     (4)      1418    (5)      2888    (15)     2903    (15)     2938    (16)     2954    (16)
                                               3004    (17)     4076    (30)     4976    (41)     6125    (52)     6575    (57)
                                               7696    (67)
TESTS_PER_LINE  000002B8-R    1675    (6)      2388    (7)      3804    (24)     3806    (24)     3841    (24)     3843    (24)
                                               3850    (24)     3858    (24)     4207    (34)
TEST_BUFFER     00002C78-R    9912    (98)     3207    (20)     3230    (20)     3358    (20)     3369    (20)     3403    (20)
                                               3606    (23)     3661    (23)     3679    (23)     8247    (76)
TEST_KEY       =00000000      863     (4)      4790    (40)
TEST_MSG        000005A4-R    1941    (6)      2571    (10)     8934    (87)
TEST_MSG_1      000002BB-R    1678    (6)      3811    (24)
TEST_MSG_2      000002C1-R    1679    (6)      3813    (24)
TEST_NUMBER     00000060-R    1460    (6)      2390    (7)      2572    (10)     3381    (20)     3383    (20)     3798    (24)
                                               4773    (40)     6020    (51)     6040    (51)     6076    (51)     7737    (67)
                                               7823    (68)     8935    (87)
TEST_PC         000003B6-R    1796    (6)      2370    (7)      3392    (20)     5797    (49)     5823    (49)     5844    (49)
                                               6052    (51)     6068    (51)     6216    (53)     6219    (53)     6332    (54)
                                               6335    (54)     6372    (55)     6688    (58)     7332    (65)     7831    (68)
                                               9141    (88)
TEST_SIZE       000001B9-R    1609    (6)      3371    (20)     8241    (76)
TEST_SPAN_END   0000038D-R    1774    (6)      4860    (40)     6022    (51)
TEST_SUB_ACTION 00001765-R    4837    (40)     4791    (40)
TICK_FLAG      =00000002      863     (4)      5232    (45)     5255    (45)     7485    (66)     7584    (66)
TICK_KEY       =00000022      863     (4)      5233    (45)
```

```
TITLE_LENGTH     000002B9-R  1677    (6)      3826    (24)      5803    (49)      7745    (67)      8926    (87)      9147    (88)
TOK             =00000006    863     (4)
TOTAL_BLOCKS     0000013D-R  1513    (6)      2518    (9)
TOTAL_NMB_TESTS  00000066-R  1466    (6)      3385    (20)      3646    (23)      7825    (68)
TPC_MSG          00000C86-R  2175    (6)      9140    (88)
TRACE_ENABL     =00000004    863     (4)
TRAP_HALT_EN    =00000020    863     (4)      2905    (15)      2956    (16)      4978    (41)      6127    (52)      7699    (67)
TRAP_OFF        =00000002    863     (4)
TR_FLAG         =00000080    863     (4)      3809    (24)
TR_KEY          =00000016    863     (4)      5706    (48)
TSTSPAN_FLAG    =00000040    863     (4)      4761    (40)      4862    (40)      6018    (51)      6028    (51)
TU58_BUFF_ADDR   00000134-R  1503    (6)      2463    (8)       2471    (8)       2477    (8)
TU58_ERROR       00000140-R  1520    (6)      2569    (10)
TU58_ERR_CODE    0000013F-R  1519    (6)      2473    (8)       2520    (9)       2570    (10)
TU58_REC_COUNT   00000136-R  1504    (6)      2464    (8)       2471    (8)
TU58_REC_NUMB    00000137-R  1505    (6)      2466    (8)       2471    (8)
TYPEN_DATA_PTR   000001A0-R  1571    (6)      2380    (7)       2570    (10)      2572    (10)      2702    (12)      3812    (24)
                                              3862    (24)      4219    (34)      4994    (41)      7507    (66)      7529    (66)
                                              7682    (67)      7693    (67)      7735    (67)      7737    (67)      7749    (67)
                                              7756    (67)      7771    (67)      7781    (67)      7791    (67)      8420    (78)
                                              8424    (78)      8426    (78)      8457    (79)      8930    (87)      8935    (87)
                                              8945    (87)      8952    (87)      9150    (88)      9866    (97)
TYPEN_DATA_TYPE  000001A2-R  1572    (6)      2697    (12)      2702    (12)
TYPE_ASCII       00000E82-R  2626    (11)     2269    (7)       2270    (7)       2361    (7)       2362    (7)       2378    (7)
                                              2379    (7)       2381    (7)       2568    (10)      2569    (10)      2571    (10)
                                              2708    (12)      2710    (12)      3636    (23)      3637    (23)      3649    (23)
                                              3650    (23)      3811    (24)      3813    (24)      3816    (24)      3817    (24)
                                              3856    (24)      3863    (24)      4129    (32)      4137    (32)      4209    (34)
                                              4210    (34)      4217    (34)      4218    (34)      4231    (34)      4954    (41)
                                              4957    (41)      4983    (41)      5004    (41)      7490    (66)      7491    (66)
                                              7523    (66)      7680    (67)      7681    (67)      7686    (67)      7687    (67)
                                              7726    (67)      7728    (67)      7736    (67)      7738    (67)      7739    (67)
                                              7750    (67)      7760    (67)      7780    (67)      7790    (67)      8260    (76)
                                              8261    (76)      8262    (76)      8391    (77)      8442    (79)      8452    (79)
                                              8455    (79)      8460    (79)      8845    (86)      8916    (87)      8917    (87)
                                              8934    (87)      8939    (87)      8946    (87)      8947    (87)      8951    (87)
                                              8956    (87)      8957    (87)      9042    (87)      9043    (87)      9076    (87)
                                              9089    (87)      9090    (87)      9091    (87)      9099    (87)      9139    (88)
                                              9140    (88)      9628    (94)      9629    (94)      9633    (94)      9654    (94)
                                              9655    (94)      9656    (94)      9665    (94)      9810    (95)      9864    (97)
                                              9867    (97)
TYPE_ASCII_U     00000E8D-R  2632    (11)     4128    (32)
TYPE_ERROR       00002764-R  8914    (87)     7300    (65)
TYPE_ERR_BUFFER  00000596-R  1938    (6)      7748    (67)      7749    (67)      7755    (67)      7756    (67)      7762    (67)
                                              7763    (67)      7770    (67)      7771    (67)      7779    (67)      7781    (67)
                                              7789    (67)      7791    (67)      8929    (87)      8930    (87)      8964    (87)
                                              8965    (87)      8972    (87)      8973    (87)      8982    (87)      8983    (87)
                                              8992    (87)      8993    (87)      9149    (88)      9150    (88)
TYPE_ERR_TEMP    0000059A-R  1939    (6)      3250    (20)      3253    (20)      3279    (20)      3281    (20)      9004    (87)
                                              9006    (87)      9017    (87)      9019    (87)
TYPE_FLAGS       0000149A-R  4118    (32)     4956    (41)      4960    (41)
TYPE_GA_LIST     00002AE3-R  9620    (94)     9064    (87)
TYPE_NUMERIC     00000E9F-R  2694    (12)     2380    (7)       2570    (10)      2572    (10)      3812    (24)      3862    (24)
                                              4219    (34)      4994    (41)      7507    (66)      7529    (66)      7682    (67)
                                              7693    (67)      7735    (67)      7737    (67)      7749    (67)      7756    (67)
```

```
                                             7771   (67)      7781   (67)      7791   (67)      8420   (78)      8424   (78)
                                             8426   (78)      8457   (79)      8930   (87)      8935   (87)      8945   (87)
                                             8952   (87)      9150   (88)      9866   (97)
TYPE_TEST_FLAG   000002B7-R    1673  (6)     2389   (7)       3779   (24)      3845   (24)      4774   (40)
TYPE_TEST_NUMB   000012C4-R    3774  (24)    3379   (20)      3405   (20)
TYPE_TU58_ERROR  00000E50-R    2567  (10)    2343   (7)       2397   (7)       2474   (8)       2523   (9)
UBI             =00000006      863   (4)
UBI_NAME         000000E7-R    1491  (6)     1476   (6)
UDP             =0000001C      863   (4)
UDP_NAME         00000A4C-R    2084  (6)     1996   (6)
USER_PASS_CNT    0000038E-R    1775  (6)     2384   (7)       4770   (40)      4881   (40)
UTR             =0000000D      863   (4)
VAL             =00002305      1540  (6)     1526   (6)       1527   (6)       1528   (6)       1529   (6)       1530   (6)
                                             1531   (6)       1532   (6)       1533   (6)       1534   (6)       1535   (6)
                                             1536   (6)       1537   (6)       1538   (6)       1539   (6)       1540   (6)
VA_KEY          =00000032      863   (4)     5498   (47)
VBUS_CLOCK      =00000080      863   (4)     3526   (22)
VBUS_COMPARE    =00000080      863   (4)     6942   (62)      6950   (62)      7072   (63)      7080   (63)      7185   (64)
                                             7195   (64)      8355   (76)      8361   (76)      9029   (87)
VBUS_LOAD       =00000010      863   (4)     4978   (41)      4980   (41)
VBUS_MSG         000003AC-R    1790  (6)     4983   (41)
VBUS_MSG_2       000005D0-R    1947  (6)     9042   (87)
VBUS_SERIAL_IN  =00000040      863   (4)
VBUS_SERIAL_OUT =00000001      863   (4)     3523   (22)
VBUS_TBL_ADDR    0000044D-R    1878  (6)     7198   (64)      9035   (87)
VBUS_TBL_COUNT   0000044F-R    1880  (6)     7201   (64)      9036   (87)
VB_KEY          =00000026      863   (4)     4965   (41)
V_BUS_BUFFER     000002C4-R    1686  (6)     3521   (22)      4987   (41)      9519   (92)
V_BUS_STROBE    =00000001      863   (4)
WBUS_FROM_D     =00000020      863   (4)
WDREG           =0000742C      863   (4)
WD_KEY          =0000003A      863   (4)     5506   (47)
WD_REG_BYTE_0   =0000742C      863   (4)     3267   (20)      3315   (20)      3323   (20)      863    (4)
WD_REG_BYTE_1   =0000742D      863   (4)
WD_REG_BYTE_2   =0000742E      863   (4)     3316   (20)
WD_REG_BYTE_3   =0000742F      863   (4)     5587   (47)
WHREG           =00007430      863   (4)
WH_REG_BYTE_0   =00007430      863   (4)     5609   (47)      5619   (47)      5629   (47)      5645   (47)      8288   (76)
                                             863    (4)       9011   (87)
WH_REG_BYTE_1   =00007431      863   (4)     8290   (76)
WH_REG_BYTE_2   =00007432      863   (4)     5603   (47)      8292   (76)
WH_REG_BYTE_3   =00007433      863   (4)     5593   (47)      5606   (47)
WORD            =00000002      863   (4)     7124   (64)
WRITE_DCS        00000EF4-R    2798  (14)    3260   (20)      3262   (20)      3305   (20)      3307   (20)      3313   (20)
                                             3354   (20)      4009   (28)      4011   (28)      4040   (29)      4042   (29)
                                             4157   (33)      5480   (47)      5575   (47)      5589   (47)      5591   (47)
                                             5601   (47)      5617   (47)      5627   (47)      5930   (50)      9843   (96)
                                             9845   (96)      9900   (98)      9902   (98)
X               =00000003      1540  (6)     1526   (6)       1527   (6)       1528   (6)       1529   (6)       1530   (6)
                                             1531   (6)       1532   (6)       1533   (6)       1534   (6)       1535   (6)
                                             1536   (6)       1537   (6)       1538   (6)       1539   (6)       1540   (6)
Y               =0000004E      9902  (98)    1374   (5)       1385   (5)       1393   (5)       1526   (6)       1527   (6)
                                             1528   (6)       1529   (6)       1530   (6)       1531   (6)       1532   (6)
                                             1533   (6)       1534   (6)       1535   (6)       1536   (6)       1537   (6)
                                             1538   (6)       1539   (6)       1540   (6)       2269   (7)       2270   (7)
```

```
                                              2314    (7)         2361    (7)         2362    (7)         2378    (7)         2379    (7)
                                              2380    (7)         2381    (7)         2568    (10)        2569    (10)        2570    (10)
                                              2571    (10)        2572    (10)        2708    (12)        2710    (12)        2807    (14)
                                              2893    (15)        2942    (16)        3207    (20)        3230    (20)        3252    (20)
                                              3253    (20)        3260    (20)        3262    (20)        3267    (20)        3305    (20)
                                              3307    (20)        3313    (20)        3314    (20)        3323    (20)        3358    (20)
                                              3369    (20)        3403    (20)        3521    (22)        3606    (23)        3627    (23)
                                              3637    (23)        3650    (23)        3661    (23)        3679    (23)        3680    (23)
                                              3811    (24)        3812    (24)        3813    (24)        3817    (24)        3856    (24)
                                              3862    (24)        3863    (24)        3894    (25)        4009    (28)        4011    (28)
                                              4040    (29)        4042    (29)        4122    (32)        4129    (32)        4137    (32)
                                              4157    (33)        4209    (34)        4210    (34)        4217    (34)        4218    (34)
                                              4219    (34)        4231    (34)        4241    (34)        4251    (34)        4254    (34)
                                              4265    (34)        4272    (34)        4309    (35)        4311    (35)        4313    (35)
                                              4334    (35)        4616    (39)        4652    (39)        4672    (39)        4685    (39)
                                              4954    (41)        4957    (41)        4983    (41)        4987    (41)        4994    (41)
                                              5004    (41)        5480    (47)        5481    (47)        5575    (47)        5589    (47)
                                              5591    (47)        5601    (47)        5617    (47)        5627    (47)        5639    (47)
                                              5645    (47)        5827    (49)        5849    (49)        5857    (49)        6171    (53)
                                              6428    (56)        6494    (56)        6503    (56)        7036    (63)        7050    (63)
                                              7131    (64)        7297    (65)        7320    (65)        7426    (66)        7490    (66)
                                              7491    (66)        7507    (66)        7523    (66)        7529    (66)        7629    (67)
                                              7680    (67)        7681    (67)        7682    (67)        7686    (67)        7687    (67)
                                              7693    (67)        7726    (67)        7735    (67)        7736    (67)        7737    (67)
                                              7738    (67)        7739    (67)        7749    (67)        7750    (67)        7756    (67)
                                              7760    (67)        7761    (67)        7771    (67)        7780    (67)        7781    (67)
                                              7790    (67)        7791    (67)        7924    (70)        7933    (70)        8069    (71)
                                              8247    (76)        8260    (76)        8261    (76)        8262    (76)        8267    (76)
                                              8346    (76)        8442    (79)        8443    (79)        8452    (79)        8455    (79)
                                              8460    (79)        8497    (80)        8557    (81)        8671    (83)        8676    (83)
                                              8724    (84)        8729    (84)        8782    (85)        8783    (85)        8845    (86)
                                              8916    (87)        8917    (87)        8930    (87)        8934    (87)        8935    (87)
                                              8939    (87)        8945    (87)        8946    (87)        8947    (87)        8951    (87)
                                              8952    (87)        8956    (87)        8957    (87)        8958    (87)        8963    (87)
                                              8973    (87)        8983    (87)        8993    (87)        9011    (87)        9042    (87)
                                              9043    (87)        9049    (87)        9076    (87)        9081    (87)        9090    (87)
                                              9091    (87)        9099    (87)        9139    (88)        9140    (88)        9150    (88)
                                              9224    (89)        9249    (89)        9259    (89)        9272    (89)        9287    (89)
                                              9337    (89)        9396    (90)        9431    (90)        9519    (92)        9628    (94)
                                              9629    (94)        9633    (94)        9652    (94)        9655    (94)        9656    (94)
                                              9665    (94)        9706    (95)        9726    (95)        9727    (95)        9728    (95)
                                              9835    (96)        9836    (96)        9843    (96)        9845    (96)        9864    (97)
                                              9867    (97)        9892    (98)        9893    (98)        9900    (98)        9902    (98)
```

```
                                                 +-------------------------+
                                                 ! Macros Cross Reference !
                                                 +-------------------------+

MACRO             SIZE          DEFINITION       REFERENCES...
-----             ----          ----------       -------------
$CONVERT          2             2700   (12)      2700   (12)
$TERM_INIT        1             2264   (7)       2264   (7)       4208   (34)      7561   (66)      9138   (88)
$TERM_READ        3             2265   (7)       2265   (7)       2752   (13)      4211   (34)      4274   (34)      7562   (66)
                                                 9151   (88)      9164   (88)

$TERM_WRITE       3             2633   (11)      2633   (11)
$TU58_OPEN        2             2518   (9)       2518   (9)
$TU58_READ        3             2471   (8)       2471   (8)
ABORT             1             754    (2)       1377   (5)       2636   (11)      2748   (13)      5079   (43)
ACI               1             856    (3)
ADC               1             856    (3)       9534   (92)      9795   (95)
ADD               1             856    (3)       2950   (16)      9531   (92)      9649   (94)      9792   (95)
ADDRESS           1             767    (2)       1470   (6)       1471   (6)       1472   (6)       1473   (6)       1474   (6)
                                                 1475   (6)       1476   (6)       1477   (6)       1478   (6)       1479   (6)
                                                 1480   (6)       1481   (6)       1482   (6)       1696   (6)       1697   (6)
                                                 1698   (6)       1699   (6)       1700   (6)       1701   (6)       1702   (6)
                                                 1703   (6)       1704   (6)       1705   (6)       1711   (6)       1804   (6)
                                                 1805   (6)       1806   (6)       1807   (6)       1808   (6)       1809   (6)
                                                 1810   (6)       1811   (6)       1812   (6)       1813   (6)       1814   (6)
                                                 1815   (6)       1816   (6)       1817   (6)       1818   (6)       1819   (6)
                                                 1820   (6)       1821   (6)       1822   (6)       1823   (6)       1824   (6)
                                                 1825   (6)       1826   (6)       1827   (6)       1828   (6)       1829   (6)
                                                 1830   (6)       1831   (6)       1832   (6)       1837   (6)       1838   (6)
                                                 1839   (6)       1990   (6)       1991   (6)       1992   (6)       1993   (6)
                                                 1994   (6)       1995   (6)       1996   (6)       1997   (6)       1998   (6)
                                                 1999   (6)       2000   (6)       2001   (6)       2002   (6)       2003   (6)
                                                 2004   (6)
ADI               1             856    (3)       4645   (39)      9264   (89)      9292   (89)
ANA               1             856    (3)       3052   (18)      5404   (46)      5637   (47)      6859   (61)      7040   (63)
                                                 7054   (63)      9314   (89)      9400   (90)      9435   (90)      9768   (95)

ANI               1             1418   (5)       1418   (5)       2278   (7)       2289   (7)       2295   (7)       2696   (12)
                                                 2704   (12)      2888   (15)      2896   (15)      2903   (15)      2938   (16)
                                                 2945   (16)      2954   (16)      3004   (17)      3214   (20)      3464   (21)
                                                 3468   (21)      3523   (22)      3666   (23)      3786   (24)      3809   (24)
                                                 3960   (27)      3976   (27)      4070   (30)      4076   (30)      4101   (31)
                                                 4202   (34)      4238   (34)      4267   (34)      4270   (34)      4674   (39)
                                                 4754   (40)      4759   (40)      4765   (40)      4783   (40)      4973   (41)
                                                 4976   (41)      4980   (41)      5000   (41)      5100   (44)      5153   (45)
                                                 5184   (45)      5225   (45)      5254   (45)      5351   (46)      5366   (46)
                                                 5369   (46)      5394   (46)      5488   (47)      5604   (47)      5607   (47)
                                                 5785   (49)      5791   (49)      5997   (51)      6012   (51)      6018   (51)
                                                 6027   (51)      6035   (51)      6049   (51)      6054   (51)      6062   (51)
                                                 6072   (51)      6117   (52)      6125   (52)      6264   (54)      6575   (57)
                                                 6625   (58)      6725   (59)      6942   (62)      6949   (62)      7072   (63)
                                                 7079   (63)      7184   (64)      7274   (65)      7280   (65)      7286   (65)
                                                 7292   (65)      7299   (65)      7305   (65)      7312   (65)      7315   (65)
                                                 7334   (65)      7429   (66)      7433   (66)      7439   (66)      7445   (66)
                                                 7449   (66)      7452   (66)      7485   (66)      7493   (66)      7499   (66)
                                                 7512   (66)      7515   (66)      7526   (66)      7540   (66)      7543   (66)
```

```
                                                  7570   (66)      7573   (66)      7583   (66)      7594   (67)      7597   (67)
                                                  7605   (67)      7608   (67)      7612   (67)      7616   (67)      7624   (67)
                                                  7640   (67)      7650   (67)      7653   (67)      7662   (67)      7668   (67)
                                                  7674   (67)      7684   (67)      7689   (67)      7696   (67)      7708   (67)
                                                  7712   (67)      7731   (67)      7753   (67)      7952   (70)      7955   (70)
                                                  7972   (70)      7975   (70)      8272   (76)      8276   (76)      8295   (76)
                                                  8355   (76)      8360   (76)      8440   (79)      8445   (79)      9029   (87)
                                                  9156   (88)      9418   (90)      9451   (90)      9522   (92)      9527   (92)
                                                  9530   (92)      9572   (93)
CALL              1             856     (3)       1377   (5)       1403   (5)       1425   (5)       2264   (7)       2266   (7)
                                                  2269   (7)       2270   (7)       2329   (7)       2343   (7)       2344   (7)
                                                  2349   (7)       2361   (7)       2362   (7)       2368   (7)       2374   (7)
                                                  2378   (7)       2379   (7)       2380   (7)       2381   (7)       2391   (7)
                                                  2397   (7)       2398   (7)       2471   (8)       2474   (8)       2475   (8)
                                                  2518   (9)       2523   (9)       2524   (9)       2568   (10)      2569   (10)
                                                  2570   (10)      2571   (10)      2572   (10)      2634   (11)      2636   (11)
                                                  2702   (12)      2708   (12)      2710   (12)      2748   (13)      2753   (13)
                                                  2801   (14)      2847   (14)      2997   (17)      2998   (17)      2999   (17)
                                                  3044   (18)      3054   (18)      3104   (19)      3113   (19)      3207   (20)
                                                  3226   (20)      3230   (20)      3237   (20)      3258   (20)      3260   (20)
                                                  3262   (20)      3278   (20)      3299   (20)      3305   (20)      3307   (20)
                                                  3308   (20)      3313   (20)      3317   (20)      3333   (20)      3354   (20)
                                                  3369   (20)      3376   (20)      3379   (20)      3403   (20)      3405   (20)
                                                  3606   (23)      3636   (23)      3637   (23)      3649   (23)      3650   (23)
                                                  3661   (23)      3811   (24)      3812   (24)      3813   (24)      3816   (24)
                                                  3817   (24)      3856   (24)      3862   (24)      3863   (24)      3971   (27)
                                                  4009   (28)      4011   (28)      4012   (28)      4040   (29)      4042   (29)
                                                  4043   (29)      4094   (31)      4099   (31)      4128   (32)      4129   (32)
                                                  4137   (32)      4157   (33)      4208   (34)      4209   (34)      4210   (34)
                                                  4212   (34)      4217   (34)      4218   (34)      4219   (34)      4229   (34)
                                                  4231   (34)      4275   (34)      4280   (34)      4317   (35)      4322   (35)
                                                  4326   (35)      4329   (35)      4335   (35)      4340   (35)      4347   (35)
                                                  4359   (35)      4363   (35)      4368   (35)      4376   (35)      4954   (41)
                                                  4956   (41)      4957   (41)      4960   (41)      4982   (41)      4983   (41)
                                                  4994   (41)      5004   (41)      5079   (43)      5175   (45)      5177   (45)
                                                  5188   (45)      5398   (46)      5418   (46)      5420   (46)      5480   (47)
                                                  5526   (47)      5538   (47)      5575   (47)      5576   (47)      5589   (47)
                                                  5591   (47)      5592   (47)      5601   (47)      5602   (47)      5617   (47)
                                                  5618   (47)      5627   (47)      5628   (47)      5650   (47)      5712   (48)
                                                  5818   (49)      5906   (50)      5911   (50)      5918   (50)      5930   (50)
                                                  6007   (51)      6025   (51)      6111   (52)      6122   (52)      6123   (52)
                                                  6170   (53)      6267   (54)      6341   (54)      6422   (56)      6427   (56)
                                                  6508   (56)      6545   (57)      6547   (57)      6654   (58)      6657   (58)
                                                  6720   (59)      6775   (60)      6825   (61)      6830   (61)      6844   (61)
                                                  6846   (61)      6911   (62)      6916   (62)      6922   (62)      6929   (62)
                                                  6937   (62)      6952   (62)      7001   (63)      7006   (63)      7012   (63)
                                                  7019   (63)      7024   (63)      7026   (63)      7067   (63)      7082   (63)
                                                  7139   (64)      7149   (64)      7175   (64)      7187   (64)      7463   (66)
                                                  7466   (66)      7474   (66)      7477   (66)      7490   (66)      7491   (66)
                                                  7507   (66)      7523   (66)      7529   (66)      7530   (66)      7555   (66)
                                                  7561   (66)      7562   (66)      7575   (66)      7577   (66)      7586   (66)
                                                  7642   (67)      7680   (67)      7681   (67)      7682   (67)      7686   (67)
                                                  7687   (67)      7693   (67)      7694   (67)      7726   (67)      7728   (67)
                                                  7735   (67)      7736   (67)      7737   (67)      7738   (67)      7739   (67)
                                                  7749   (67)      7750   (67)      7756   (67)      7760   (67)      7765   (67)
```

```
                                                 7771     (67)      7774     (67)      7780     (67)      7781     (67)      7784     (67)
                                                 7790     (67)      7791     (67)      7792     (67)      7828     (68)      7829     (68)
                                                 7907     (70)      7909     (70)      7959     (70)      7980     (70)      7994     (70)
                                                 8031     (71)      8040     (71)      8043     (71)      8107     (72)      8139     (73)
                                                 8246     (76)      8260     (76)      8261     (76)      8262     (76)      8363     (76)
                                                 8391     (77)      8420     (78)      8424     (78)      8426     (78)      8442     (79)
                                                 8452     (79)      8455     (79)      8457     (79)      8460     (79)      8845     (86)
                                                 8916     (87)      8917     (87)      8930     (87)      8934     (87)      8935     (87)
                                                 8939     (87)      8945     (87)      8946     (87)      8947     (87)      8951     (87)
                                                 8952     (87)      8956     (87)      8957     (87)      8959     (87)      8967     (87)
                                                 8974     (87)      8977     (87)      8984     (87)      8987     (87)      8994     (87)
                                                 9007     (87)      9012     (87)      9042     (87)      9043     (87)      9046     (87)
                                                 9050     (87)      9076     (87)      9089     (87)      9090     (87)      9091     (87)
                                                 9099     (87)      9138     (88)      9139     (88)      9140     (88)      9150     (88)
                                                 9151     (88)      9165     (88)      9274     (89)      9293     (89)      9356     (89)
                                                 9628     (94)      9629     (94)      9633     (94)      9642     (94)      9654     (94)
                                                 9655     (94)      9656     (94)      9665     (94)      9804     (95)      9810     (95)
                                                 9841     (96)      9843     (96)      9845     (96)      9846     (96)      9864     (97)
                                                 9866     (97)      9867     (97)      9898     (98)      9900     (98)      9902     (98)
                                                 9903     (98)
CC                  1             856     (3)    1430     (5)       5819     (49)
CHECKRP             1             1375    (5)    1375     (5)       1384     (5)       1386     (5)       1387     (5)       1388     (5)
                                                 1395     (5)       1396     (5)       1397     (5)       1400     (5)       1402     (5)
                                                 1404     (5)       1407     (5)       1408     (5)       1429     (5)       2320     (7)
                                                 2469     (8)       2630     (11)      2695     (12)      2703     (12)      2800     (14)
                                                 2817     (14)      2820     (14)      2825     (14)      2826     (14)      2827     (14)
                                                 2835     (14)      2841     (14)      2848     (14)      3043     (18)      3055     (18)
                                                 3058     (18)      3063     (18)      3103     (19)      3114     (19)      3117     (19)
                                                 3122     (19)      3213     (20)      3216     (20)      3224     (20)      3225     (20)
                                                 3229     (20)      3231     (20)      3236     (20)      3238     (20)      3242     (20)
                                                 3247     (20)      3248     (20)      3251     (20)      3257     (20)      3263     (20)
                                                 3271     (20)      3272     (20)      3278     (20)      3282     (20)      3284     (20)
                                                 3286     (20)      3287     (20)      3291     (20)      3296     (20)      3297     (20)
                                                 3298     (20)      3308     (20)      3317     (20)      3318     (20)      3326     (20)
                                                 3327     (20)      3333     (20)      3337     (20)      3339     (20)      3341     (20)
                                                 3342     (20)      3348     (20)      3353     (20)      3359     (20)      3367     (20)
                                                 3374     (20)      3375     (20)      3380     (20)      3394     (20)      3406     (20)
                                                 3412     (20)      3413     (20)      3414     (20)      3415     (20)      3423     (20)
                                                 3425     (20)      3462     (21)      3466     (21)      3467     (21)      3473     (21)
                                                 3475     (21)      3478     (21)      3479     (21)      3481     (21)      3482     (21)
                                                 3485     (21)      3525     (22)      3605     (23)      3607     (23)      3608     (23)
                                                 3611     (23)      3612     (23)      3618     (23)      3626     (23)      3631     (23)
                                                 3633     (23)      3641     (23)      3643     (23)      3644     (23)      3647     (23)
                                                 3648     (23)      3654     (23)      3655     (23)      3683     (23)      3684     (23)
                                                 3690     (23)      3691     (23)      3693     (23)      3694     (23)      3706     (23)
                                                 3707     (23)      3709     (23)      3710     (23)      3717     (23)      3720     (23)
                                                 3722     (23)      3723     (23)      3829     (24)      3836     (24)      3967     (27)
                                                 3985     (27)      4004     (28)      4005     (28)      4006     (28)      4007     (28)
                                                 4012     (28)      4013     (28)      4014     (28)      4015     (28)      4016     (28)
                                                 4035     (29)      4036     (29)      4037     (29)      4038     (29)      4043     (29)
                                                 4044     (29)      4045     (29)      4046     (29)      4047     (29)      4125     (32)
                                                 4126     (32)      4127     (32)      4130     (32)      4131     (32)      4132     (32)
                                                 4133     (32)      4134     (32)      4155     (33)      4158     (33)      4243     (34)
                                                 4248     (34)      4253     (34)      4281     (34)      4312     (35)      4355     (35)
                                                 4375     (35)      4377     (35)      4445     (36)      4449     (36)      4451     (36)
```

```
                                        4461   (36)      4462   (36)      4469   (36)      4518   (37)      4526   (37)
                                        4568   (38)      4571   (38)      4573   (38)      4619   (39)      4621   (39)
                                        4623   (39)      4633   (39)      4649   (39)      4653   (39)      4655   (39)
                                        4658   (39)      4660   (39)      4666   (39)      4676   (39)      4686   (39)
                                        4691   (39)      4692   (39)      4696   (39)      4784   (40)      4842   (40)
                                        4845   (40)      4858   (40)      4861   (40)      4879   (40)      4882   (40)
                                        4990   (41)      4991   (41)      4992   (41)      4995   (41)      4996   (41)
                                        5002   (41)      5003   (41)      5005   (41)      5006   (41)      5154   (45)
                                        5169   (45)      5172   (45)      5173   (45)      5185   (45)      5210   (45)
                                        5211   (45)      5214   (45)      5217   (45)      5226   (45)      5246   (45)
                                        5247   (45)      5250   (45)      5283   (45)      5284   (45)      5288   (45)
                                        5352   (46)      5378   (46)      5379   (46)      5383   (46)      5395   (46)
                                        5412   (46)      5415   (46)      5416   (46)      5478   (47)      5484   (47)
                                        5489   (47)      5524   (47)      5535   (47)      5576   (47)      5585   (47)
                                        5592   (47)      5602   (47)      5618   (47)      5628   (47)      5800   (49)
                                        5801   (49)      5804   (49)      5826   (49)      5833   (49)      5838   (49)
                                        5839   (49)      5851   (49)      5854   (49)      5858   (49)      5920   (50)
                                        5923   (50)      6175   (53)      6176   (53)      6215   (53)      6218   (53)
                                        6268   (54)      6269   (54)      6270   (54)      6271   (54)      6279   (54)
                                        6280   (54)      6281   (54)      6282   (54)      6285   (54)      6291   (54)
                                        6292   (54)      6293   (54)      6295   (54)      6297   (54)      6299   (54)
                                        6303   (54)      6304   (54)      6306   (54)      6307   (54)      6309   (54)
                                        6319   (54)      6325   (54)      6330   (54)      6333   (54)      6344   (54)
                                        6429   (56)      6438   (56)      6441   (56)      6442   (56)      6449   (56)
                                        6450   (56)      6453   (56)      6454   (56)      6455   (56)      6456   (56)
                                        6460   (56)      6464   (56)      6465   (56)      6478   (56)      6480   (56)
                                        6493   (56)      6498   (56)      6499   (56)      6502   (56)      6507   (56)
                                        6552   (57)      6555   (57)      6569   (57)      6577   (57)      6580   (57)
                                        6581   (57)      6583   (57)      6655   (58)      6661   (58)      6687   (58)
                                        6835   (61)      6838   (61)      6839   (61)      6848   (61)      6851   (61)
                                        6855   (61)      6861   (61)      6862   (61)      6924   (62)      6927   (62)
                                        7014   (63)      7017   (63)      7028   (63)      7031   (63)      7035   (63)
                                        7042   (63)      7043   (63)      7049   (63)      7056   (63)      7057   (63)
                                        7143   (64)      7147   (64)      7154   (64)      7164   (64)      7165   (64)
                                        7171   (64)      7177   (64)      7197   (64)      7324   (65)      7743   (67)
                                        7746   (67)      7747   (67)      7767   (67)      7768   (67)      7769   (67)
                                        7776   (67)      7777   (67)      7778   (67)      7786   (67)      7787   (67)
                                        7788   (67)      7861   (69)      7914   (70)      7917   (70)      7929   (70)
                                        7930   (70)      7937   (70)      7941   (70)      7949   (70)      7957   (70)
                                        7960   (70)      7961   (70)      7964   (70)      7977   (70)      7978   (70)
                                        7981   (70)      7982   (70)      7985   (70)      7987   (70)      7989   (70)
                                        7990   (70)      7992   (70)      8032   (71)      8045   (71)      8048   (71)
                                        8050   (71)      8060   (71)      8062   (71)      8068   (71)      8076   (71)
                                        8077   (71)      8108   (72)      8110   (72)      8143   (73)      8145   (73)
                                        8248   (76)      8303   (76)      8308   (76)      8310   (76)      8313   (76)
                                        8314   (76)      8316   (76)      8317   (76)      8320   (76)      8321   (76)
                                        8322   (76)      8325   (76)      8326   (76)      8327   (76)      8334   (76)
                                        8337   (76)      8340   (76)      8343   (76)      8390   (77)      8454   (79)
                                        8456   (79)      8503   (80)      8505   (80)      8560   (81)      8565   (81)
                                        8573   (81)      8575   (81)      8630   (82)      8675   (83)      8679   (83)
                                        8682   (83)      8683   (83)      8728   (84)      8732   (84)      8735   (84)
                                        8736   (84)      8790   (85)      8791   (85)      8805   (85)      8806   (85)
                                        8915   (87)      8924   (87)      8927   (87)      8928   (87)      8969   (87)
                                        8970   (87)      8971   (87)      8979   (87)      8980   (87)      8981   (87)
                                        8989   (87)      8990   (87)      8991   (87)      8998   (87)      9002   (87)
```

```
                                                 9005   (87)      9016   (87)      9020   (87)      9022   (87)      9024   (87)
                                                 9034   (87)      9040   (87)      9041   (87)      9044   (87)      9045   (87)
                                                 9047   (87)      9048   (87)      9051   (87)      9052   (87)      9053   (87)
                                                 9056   (87)      9061   (87)      9069   (87)      9073   (87)      9074   (87)
                                                 9075   (87)      9077   (87)      9079   (87)      9084   (87)      9086   (87)
                                                 9092   (87)      9093   (87)      9095   (87)      9096   (87)      9098   (87)
                                                 9144   (88)      9145   (88)      9148   (88)      9212   (89)      9213   (89)
                                                 9215   (89)      9218   (89)      9219   (89)      9220   (89)      9221   (89)
                                                 9228   (89)      9229   (89)      9235   (89)      9252   (89)      9273   (89)
                                                 9281   (89)      9282   (89)      9283   (89)      9285   (89)      9301   (89)
                                                 9302   (89)      9303   (89)      9307   (89)      9311   (89)      9316   (89)
                                                 9319   (89)      9335   (89)      9336   (89)      9342   (89)      9345   (89)
                                                 9346   (89)      9349   (89)      9350   (89)      9392   (90)      9405   (90)
                                                 9406   (90)      9412   (90)      9413   (90)      9416   (90)      9429   (90)
                                                 9430   (90)      9440   (90)      9441   (90)      9447   (90)      9448   (90)
                                                 9449   (90)      9481   (91)      9488   (91)      9584   (93)      9625   (94)
                                                 9626   (94)      9627   (94)      9630   (94)      9631   (94)      9632   (94)
                                                 9634   (94)      9635   (94)      9641   (94)      9644   (94)      9653   (94)
                                                 9657   (94)      9658   (94)      9660   (94)      9661   (94)      9663   (94)
                                                 9664   (94)      9666   (94)      9667   (94)      9710   (95)      9715   (95)
                                                 9720   (95)      9724   (95)      9725   (95)      9732   (95)      9733   (95)
                                                 9734   (95)      9738   (95)      9739   (95)      9740   (95)      9744   (95)
                                                 9745   (95)      9746   (95)      9750   (95)      9751   (95)      9752   (95)
                                                 9753   (95)      9754   (95)      9762   (95)      9763   (95)      9767   (95)
                                                 9769   (95)      9776   (95)      9777   (95)      9778   (95)      9779   (95)
                                                 9787   (95)      9788   (95)      9802   (95)      9803   (95)      9805   (95)
                                                 9806   (95)      9808   (95)      9809   (95)      9840   (96)      9846   (96)
                                                 9863   (97)      9865   (97)      9897   (98)      9903   (98)
CM                 1            856    (3)
CMA                1            856    (3)       3049   (18)      3362   (20)      3365   (20)      3418   (20)      3421   (20)
                                                 4959   (41)      5401   (46)      6496   (56)      7927   (70)      9403   (90)
                                                 9438   (90)
CMC                1            856    (3)       3487   (21)      3726   (23)      4392   (35)      4471   (36)      4534   (37)
                                                 4698   (39)      4898   (40)      5012   (41)      5060   (42)      5108   (44)
                                                 5297   (45)      5426   (46)      5657   (47)      5718   (48)      8585   (81)
                                                 8795   (85)      9167   (88)      9544   (92)
CMP                1            856    (3)       2385   (7)       3386   (20)      3610   (23)      4435   (36)      5811   (49)
                                                 5815   (49)      6023   (51)      6043   (51)      6185   (53)      6193   (53)
                                                 6296   (54)      6308   (54)      6663   (58)      6666   (58)      7322   (65)
                                                 7326   (65)      7553   (66)      7627   (67)      7716   (67)      7826   (68)
                                                 8305   (76)      8309   (76)      8315   (76)      8323   (76)      8788   (85)
                                                 8803   (85)      9541   (92)      9586   (93)      9590   (93)
CNC                1            856    (3)
CNZ                1            856    (3)       5998   (51)      7281   (65)      7293   (65)      7306   (65)      9064   (87)
CP                 1            856    (3)
CPE                1            856    (3)
CPI                1            856    (3)       2341   (7)       2355   (7)       2521   (9)       2746   (13)      3244   (20)
                                                 3293   (20)      3351   (20)      3851   (24)      4214   (34)      4227   (34)
                                                 4345   (35)      4350   (35)      4357   (35)      4361   (35)      4447   (36)
                                                 4515   (37)      4517   (37)      4524   (37)      4531   (37)      4629   (39)
                                                 4631   (39)      4636   (39)      4643   (39)      4785   (40)      4790   (40)
                                                 4792   (40)      4794   (40)      4796   (40)      4798   (40)      4800   (40)
                                                 4852   (40)      4949   (41)      4965   (41)      5155   (45)      5157   (45)
                                                 5159   (45)      5161   (45)      5163   (45)      5186   (45)      5197   (45)
                                                 5208   (45)      5227   (45)      5230   (45)      5233   (45)      5236   (45)
```

```
                                            5244  (45)     5278  (45)     5353  (46)     5355  (46)     5357  (46)
                                            5359  (46)     5376  (46)     5396  (46)     5490  (47)     5492  (47)
                                            5494  (47)     5496  (47)     5498  (47)     5500  (47)     5502  (47)
                                            5504  (47)     5506  (47)     5508  (47)     5510  (47)     5512  (47)
                                            5514  (47)     5522  (47)     5533  (47)     5583  (47)     5706  (48)
                                            5829  (49)     6571  (57)     6630  (58)     6632  (58)     6634  (58)
                                            6672  (58)     6679  (58)     7516  (66)     7518  (66)     7564  (66)
                                            7654  (67)     7656  (67)     8250  (76)     8255  (76)     8282  (76)
                                            8418  (78)     8422  (78)     8450  (79)     8577  (81)     8580  (81)
                                            8838  (86)     9153  (88)     9575  (93)     9578  (93)     9637  (94)
                                            9639  (94)
CPO             1           856    (3)
CZ              1           856    (3)      7300  (65)
DAD             1           856    (3)      1375  (5)      1386  (5)      1407  (5)      2320  (7)      2469  (8)
                                            2820  (14)     2825  (14)     3478  (21)     3631  (23)     3691  (23)
                                            3720  (23)     3723  (23)     3829  (24)     3836  (24)     4243  (34)
                                            4842  (40)     4858  (40)     4879  (40)     5169  (45)     5173  (45)
                                            5412  (46)     5416  (46)     5524  (47)     5535  (47)     5585  (47)
                                            5800  (49)     5804  (49)     5851  (49)     5920  (50)     5923  (50)
                                            6279  (54)     6429  (56)     6441  (56)     6464  (56)     6552  (57)
                                            6555  (57)     6687  (58)     6835  (61)     6838  (61)     6848  (61)
                                            6851  (61)     6924  (62)     6927  (62)     7014  (63)     7017  (63)
                                            7028  (63)     7031  (63)     7143  (64)     7165  (64)     7743  (67)
                                            7746  (67)     7914  (70)     7917  (70)     7941  (70)     7961  (70)
                                            7964  (70)     7982  (70)     7985  (70)     8045  (71)     8048  (71)
                                            8060  (71)     8143  (73)     8248  (76)     8303  (76)     8334  (76)
                                            8390  (77)     8505  (80)     8560  (81)     8630  (82)     8924  (87)
                                            8927  (87)     9084  (87)     9144  (88)     9148  (88)     9252  (89)
                                            9273  (89)     9301  (89)     9416  (90)     9653  (94)     9710  (95)
                                            9806  (95)     9809  (95)
DCR             1           856    (3)      1409  (5)      2318  (7)      2836  (14)     2842  (14)     2900  (15)
                                            3042  (18)     3102  (19)     3218  (20)     3273  (20)     3283  (20)
                                            3328  (20)     3338  (20)     3372  (20)     3530  (22)     3656  (23)
                                            3681  (23)     3805  (24)     4135  (32)     4160  (33)     4459  (36)
                                            4661  (39)     4667  (39)     4693  (39)     4997  (41)     5809  (49)
                                            5834  (49)     6177  (53)     6437  (56)     6451  (56)     6481  (56)
                                            6500  (56)     6584  (57)     6863  (61)     7044  (63)     7058  (63)
                                            7160  (64)     7178  (64)     7200  (64)     7495  (66)     7691  (67)
                                            7711  (67)     7733  (67)     7931  (70)     8328  (76)     8506  (80)
                                            8628  (82)     8684  (83)     8737  (84)     8792  (85)     8807  (85)
                                            9021  (87)     9038  (87)     9054  (87)     9094  (87)     9230  (89)
                                            9329  (89)     9347  (89)     9407  (90)     9442  (90)     9484  (91)
                                            9659  (94)     9771  (95)     9780  (95)
DCX             1           856    (3)      3287  (20)     3342  (20)     3611  (23)     3612  (23)     4451  (36)
                                            4692  (39)     4990  (41)     6280  (54)     6281  (54)     6282  (54)
                                            6285  (54)     6303  (54)     6304  (54)     6453  (56)     6454  (56)
                                            6455  (56)     6456  (56)     8310  (76)     8316  (76)     8317  (76)
                                            8325  (76)     8326  (76)     8327  (76)     8575  (81)     9584  (93)
                                            9667  (94)     9752  (95)     9753  (95)     9754  (95)     9776  (95)
                                            9777  (95)     9778  (95)     9779  (95)
EQUATE          2           856    (3)      856   (3)
EQUATE_SYMBOLS  21          863    (4)      863   (4)
EXECUTE         2           818    (2)      3278  (20)     3308  (20)     3317  (20)     3333  (20)     4012  (28)
                                            4043  (29)     5576  (47)     5592  (47)     5602  (47)     5618  (47)
                                            5628  (47)     9846  (96)     9903  (98)
```

```
GENERATE_TITLE   1             843    (2)      1468   (6)
HLT              1             856    (3)
INR              1             856    (3)      2339   (7)       3064   (18)      3123   (19)      3280   (20)      3382   (20)
                                               3799   (24)      3823   (24)      3842   (24)      4159   (33)      4443   (36)
                                               4444   (36)      5807   (49)      6435   (56)      7159   (64)      7502   (66)
                                               7859   (69)      8348   (76)      8850   (86)      9018   (87)      9238   (89)
                                               9262   (89)      9290   (89)      9325   (89)      9327   (89)      9782   (95)
INX              1             856    (3)      1387   (5)       1388   (5)       2630   (11)      2826   (14)      2835   (14)
                                               2841   (14)      3216   (20)      3242   (20)      3247   (20)      3271   (20)
                                               3272   (20)      3291   (20)      3296   (20)      3326   (20)      3327   (20)
                                               3348   (20)      3353   (20)      3359   (20)      3367   (20)      3394   (20)
                                               3412   (20)      3413   (20)      3415   (20)      3423   (20)      3525   (22)
                                               3607   (23)      3608   (23)      3633   (23)      3644   (23)      3647   (23)
                                               3648   (23)      3694   (23)      4133   (32)      4134   (32)      4248   (34)
                                               4312   (35)      4355   (35)      4445   (36)      4449   (36)      4461   (36)
                                               4462   (36)      4469   (36)      4518   (37)      4526   (37)      4571   (38)
                                               4619   (39)      4621   (39)      4623   (39)      4633   (39)      4660   (39)
                                               4691   (39)      4784   (40)      4845   (40)      4861   (40)      4882   (40)
                                               5154   (45)      5172   (45)      5185   (45)      5210   (45)      5211   (45)
                                               5214   (45)      5217   (45)      5226   (45)      5246   (45)      5247   (45)
                                               5250   (45)      5283   (45)      5284   (45)      5288   (45)      5352   (46)
                                               5378   (46)      5379   (46)      5383   (46)      5395   (46)      5415   (46)
                                               5489   (47)      5801   (49)      5826   (49)      5833   (49)      5838   (49)
                                               5839   (49)      5854   (49)      6175   (53)      6176   (53)      6215   (53)
                                               6218   (53)      6268   (54)      6269   (54)      6270   (54)      6271   (54)
                                               6291   (54)      6292   (54)      6293   (54)      6295   (54)      6306   (54)
                                               6307   (54)      6330   (54)      6333   (54)      6344   (54)      6449   (56)
                                               6450   (56)      6478   (56)      6480   (56)      6498   (56)      6499   (56)
                                               6581   (57)      6583   (57)      6861   (61)      6862   (61)      7042   (63)
                                               7043   (63)      7056   (63)      7057   (63)      7164   (64)      7171   (64)
                                               7177   (64)      7197   (64)      7324   (65)      7747   (67)      7769   (67)
                                               7778   (67)      7788   (67)      7929   (70)      7930   (70)      8076   (71)
                                               8077   (71)      8108   (72)      8110   (72)      8145   (73)      8308   (76)
                                               8313   (76)      8314   (76)      8320   (76)      8321   (76)      8322   (76)
                                               8337   (76)      8340   (76)      8343   (76)      8503   (80)      8565   (81)
                                               8573   (81)      8682   (83)      8683   (83)      8735   (84)      8736   (84)
                                               8790   (85)      8791   (85)      8805   (85)      8806   (85)      8928   (87)
                                               8971   (87)      8981   (87)      8991   (87)      9053   (87)      9061   (87)
                                               9069   (87)      9074   (87)      9086   (87)      9096   (87)      9145   (88)
                                               9228   (89)      9229   (89)      9345   (89)      9346   (89)      9405   (90)
                                               9406   (90)      9440   (90)      9441   (90)      9448   (90)      9449   (90)
                                               9626   (94)      9661   (94)      9715   (95)      9720   (95)      9732   (95)
                                               9733   (95)      9734   (95)      9738   (95)      9739   (95)      9740   (95)
                                               9744   (95)      9745   (95)      9746   (95)      9767   (95)      9769   (95)
JC               1             856    (3)      2330   (7)       3227   (20)      4259   (34)      4318   (35)      4336   (35)
                                               4348   (35)      4360   (35)      4364   (35)      4369   (35)      5189   (45)
                                               5399   (46)      5812   (49)      5816   (49)      6320   (54)      7155   (64)
                                               7176   (64)      7766   (67)      7775   (67)      7785   (67)      8041   (71)
                                               8256   (76)      8968   (87)      8978   (87)      8988   (87)      9587   (93)
                                               9591   (93)      9638   (94)
JM               1             856    (3)      2319   (7)       3687   (23)      4434   (36)      4635   (39)      4637   (39)
                                               4642   (39)      4850   (40)      5207   (45)      5705   (48)      5830   (49)
                                               5835   (49)      6315   (54)      6629   (58)      6782   (60)      7517   (66)
                                               7519   (66)      7655   (67)      7923   (70)      8553   (81)      8629   (82)
                                               9039   (87)      9261   (89)      9289   (89)      9355   (89)      9485   (91)
```

```
JMP               1             856    (3)       1373   (5)       1431   (5)       2284   (7)       2297   (7)       2321   (7)
                                                 2333   (7)       2345   (7)       2363   (7)       2399   (7)       2476   (8)
                                                 2525   (9)       2709   (12)      3396   (20)      3426   (20)      3484   (21)
                                                 3673   (23)      3695   (23)      3702   (23)      3792   (24)      3797   (24)
                                                 3807   (24)      3864   (24)      4220   (34)      4232   (34)      4282   (34)
                                                 4330   (35)      4380   (35)      4450   (36)      4452   (36)      4463   (36)
                                                 4527   (37)      4679   (39)      4808   (40)      4816   (40)      4824   (40)
                                                 4832   (40)      4873   (40)      4883   (40)      4961   (41)      5007   (41)
                                                 5178   (45)      5218   (45)      5258   (45)      5387   (46)      5406   (46)
                                                 5516   (47)      5527   (47)      5539   (47)      5545   (47)      5551   (47)
                                                 5557   (47)      5563   (47)      5569   (47)      5577   (47)      5595   (47)
                                                 5611   (47)      5621   (47)      5640   (47)      5651   (47)      5840   (49)
                                                 6030   (51)      6057   (51)      6201   (53)      6206   (53)      6211   (53)
                                                 6321   (54)      6483   (56)      6579   (57)      6642   (58)      6649   (58)
                                                 6674   (58)      7166   (64)      7467   (66)      7576   (66)      7578   (66)
                                                 7587   (66)      7643   (67)      7701   (67)      7793   (67)      8051   (71)
                                                 8063   (71)      8253   (76)      8263   (76)      8350   (76)      8357   (76)
                                                 8421   (78)      8425   (78)      8453   (79)      8631   (82)      8809   (85)
                                                 9159   (88)      9248   (89)      9263   (89)      9291   (89)      9487   (91)
                                                 9643   (94)
JNC               1             856    (3)       2351   (7)       2392   (7)       2472   (8)       2519   (9)       2635   (11)
                                                 2745   (13)      3377   (20)      4124   (32)      4213   (34)      4230   (34)
                                                 4327   (35)      4378   (35)      6194   (53)      6210   (53)      6326   (54)
                                                 6938   (62)      7068   (63)      8252   (76)      9640   (94)
JNZ               1             856    (3)       1410   (5)       2342   (7)       2837   (14)      2843   (14)      2901   (15)
                                                 2952   (16)      3215   (20)      3274   (20)      3285   (20)      3329   (20)
                                                 3340   (20)      3387   (20)      3469   (21)      3531   (22)      3667   (23)
                                                 3781   (24)      3852   (24)      4136   (32)      4161   (33)      4268   (34)
                                                 4351   (35)      4358   (35)      4448   (36)      4519   (37)      4525   (37)
                                                 4662   (39)      4668   (39)      4694   (39)      4801   (40)      4840   (40)
                                                 4950   (41)      4966   (41)      5001   (41)      5164   (45)      5198   (45)
                                                 5237   (45)      5360   (46)      5813   (49)      5817   (49)      6024   (51)
                                                 6044   (51)      6050   (51)      6178   (53)      6186   (53)      6298   (54)
                                                 6452   (56)      6472   (56)      6482   (56)      6501   (56)      6573   (57)
                                                 6585   (57)      6648   (58)      6664   (58)      6667   (58)      6864   (61)
                                                 7045   (63)      7059   (63)      7179   (64)      7316   (65)      7323   (65)
                                                 7327   (65)      7440   (66)      7453   (66)      7456   (66)      7500   (66)
                                                 7513   (66)      7537   (66)      7544   (66)      7554   (66)      7565   (66)
                                                 7609   (67)      7614   (67)      7628   (67)      7663   (67)      7717   (67)
                                                 7932   (70)      8306   (76)      8311   (76)      8318   (76)      8329   (76)
                                                 8419   (78)      8423   (78)      8451   (79)      8507   (80)      8578   (81)
                                                 8581   (81)      8685   (83)      8738   (84)      8786   (85)      8789   (85)
                                                 8793   (85)      8804   (85)      8808   (85)      8839   (86)      8942   (87)
                                                 9023   (87)      9097   (87)      9231   (89)      9317   (89)      9331   (89)
                                                 9348   (89)      9352   (89)      9408   (90)      9443   (90)      9456   (90)
                                                 9542   (92)      9576   (93)      9588   (93)      9662   (94)      9770   (95)
                                                 9772   (95)      9781   (95)
JP                1             856    (3)       2305   (7)       3701   (23)      4644   (39)      5707   (48)      6277   (54)
                                                 7657   (67)      9055   (87)      9246   (89)
JPE               1             856    (3)       9401   (90)      9436   (90)
JPO               1             856    (3)
JZ                1             856    (3)       2279   (7)       2290   (7)       2357   (7)       2386   (7)       2522   (9)
                                                 2705   (12)      2747   (13)      3219   (20)      3245   (20)      3294   (20)
                                                 3352   (20)      3402   (20)      3465   (21)      3613   (23)      3657   (23)
                                                 3682   (23)      3787   (24)      3803   (24)      3810   (24)      4120   (32)
```

```
                                        4215    (34)     4228    (34)     4271    (34)     4346    (35)     4362    (35)
                                        4436    (36)     4460    (36)     4516    (37)     4532    (37)     4630    (39)
                                        4632    (39)     4786    (40)     4791    (40)     4793    (40)     4795    (40)
                                        4797    (40)     4799    (40)     4854    (40)     4998    (41)     5156    (45)
                                        5158    (45)     5160    (45)     5162    (45)     5187    (45)     5209    (45)
                                        5228    (45)     5231    (45)     5234    (45)     5245    (45)     5279    (45)
                                        5354    (46)     5356    (46)     5358    (46)     5377    (46)     5397    (46)
                                        5491    (47)     5493    (47)     5495    (47)     5497    (47)     5499    (47)
                                        5501    (47)     5503    (47)     5505    (47)     5507    (47)     5509    (47)
                                        5511    (47)     5513    (47)     5515    (47)     5523    (47)     5534    (47)
                                        5584    (47)     5786    (49)     5792    (49)     5808    (49)     6019    (51)
                                        6036    (51)     6118    (52)     6195    (53)     6436    (56)     6631    (58)
                                        6633    (58)     6635    (58)     6641    (58)     6673    (58)     6680    (58)
                                        7161    (64)     7275    (65)     7287    (65)     7313    (65)     7446    (66)
                                        7450    (66)     7486    (66)     7541    (66)     7574    (66)     7617    (67)
                                        7641    (67)     7651    (67)     7669    (67)     7675    (67)     7685    (67)
                                        7827    (68)     7953    (70)     7973    (70)     8035    (71)     8251    (76)
                                        8273    (76)     8283    (76)     8299    (76)     8324    (76)     8441    (79)
                                        8446    (79)     8837    (86)     8955    (87)     9001    (87)     9030    (87)
                                        9072    (87)     9154    (88)     9423    (90)     9573    (93)     9579    (93)
LDA             1          856   (3)    1417    (5)      2276    (7)      2317    (7)      2338    (7)      2382    (7)
                                        2384    (7)      2749    (13)     2799    (14)     2802    (14)     2804    (14)
                                        2887    (15)     2891    (15)     2895    (15)     2902    (15)     2937    (16)
                                        2940    (16)     2944    (16)     2953    (16)     3003    (17)     3041    (18)
                                        3045    (18)     3048    (18)     3051    (18)     3101    (19)     3105    (19)
                                        3108    (19)     3110    (19)     3279    (20)     3381    (20)     3385    (20)
                                        3463    (21)     3522    (22)     3535    (22)     3609    (23)     3665    (23)
                                        3677    (23)     3779    (24)     3785    (24)     3791    (24)     3798    (24)
                                        3804    (24)     3808    (24)     3841    (24)     3850    (24)     3890    (25)
                                        3915    (26)     3955    (27)     3975    (27)     4069    (30)     4075    (30)
                                        4095    (31)     4100    (31)     4201    (34)     4226    (34)     4237    (34)
                                        4753    (40)     4758    (40)     4763    (40)     4805    (40)     4813    (40)
                                        4821    (40)     4829    (40)     4870    (40)     4894    (40)     4955    (41)
                                        4958    (41)     4970    (41)     4975    (41)     5056    (42)     5098    (44)
                                        5102    (44)     5190    (45)     5253    (45)     5264    (45)     5270    (45)
                                        5365    (46)     5368    (46)     5403    (46)     5593    (47)     5603    (47)
                                        5606    (47)     5609    (47)     5619    (47)     5629    (47)     5784    (49)
                                        5790    (49)     5905    (50)     5910    (50)     5927    (50)     5929    (50)
                                        5996    (51)     6011    (51)     6017    (51)     6020    (51)     6022    (51)
                                        6026    (51)     6034    (51)     6040    (51)     6042    (51)     6048    (51)
                                        6053    (51)     6061    (51)     6071    (51)     6075    (51)     6116    (52)
                                        6124    (52)     6169    (53)     6182    (53)     6184    (53)     6190    (53)
                                        6192    (53)     6216    (53)     6219    (53)     6263    (54)     6266    (54)
                                        6340    (54)     6421    (56)     6470    (56)     6544    (57)     6546    (57)
                                        6562    (57)     6574    (57)     6624    (58)     6627    (58)     6653    (58)
                                        6656    (58)     6671    (58)     6678    (58)     6724    (59)     6730    (59)
                                        6771    (60)     6774    (60)     6779    (60)     6781    (60)     6824    (61)
                                        6829    (61)     6843    (61)     6845    (61)     6856    (61)     6910    (62)
                                        6915    (62)     6921    (62)     6928    (62)     6933    (62)     6936    (62)
                                        6939    (62)     6948    (62)     7000    (63)     7005    (63)     7011    (63)
                                        7018    (63)     7023    (63)     7025    (63)     7037    (63)     7051    (63)
                                        7063    (63)     7066    (63)     7069    (63)     7078    (63)     7148    (64)
                                        7183    (64)     7192    (64)     7273    (65)     7279    (65)     7285    (65)
                                        7291    (65)     7298    (65)     7304    (65)     7310    (65)     7321    (65)
                                        7325    (65)     7333    (65)     7424    (66)     7428    (66)     7432    (66)
```

```
                                                    7438   (66)       7444   (66)       7447   (66)       7454   (66)       7478   (66)
                                                    7484   (66)       7492   (66)       7498   (66)       7501   (66)       7511   (66)
                                                    7514   (66)       7535   (66)       7538   (66)       7542   (66)       7550   (66)
                                                    7552   (66)       7563   (66)       7569   (66)       7572   (66)       7582   (66)
                                                    7593   (67)       7596   (67)       7603   (67)       7607   (67)       7610   (67)
                                                    7615   (67)       7621   (67)       7634   (67)       7639   (67)       7647   (67)
                                                    7649   (67)       7652   (67)       7661   (67)       7667   (67)       7673   (67)
                                                    7683   (67)       7688   (67)       7695   (67)       7705   (67)       7710   (67)
                                                    7715   (67)       7729   (67)       7823   (68)       7825   (68)       7858   (69)
                                                    7906   (70)       7908   (70)       7921   (70)       7958   (70)       7979   (70)
                                                    7991   (70)       7993   (70)       8030   (71)       8033   (71)       8039   (71)
                                                    8106   (72)       8138   (73)       8171   (74)       8198   (75)       8241   (76)
                                                    8270   (76)       8288   (76)       8290   (76)       8292   (76)       8294   (76)
                                                    8304   (76)       8307   (76)       8312   (76)       8319   (76)       8335   (76)
                                                    8338   (76)       8341   (76)       8344   (76)       8352   (76)       8359   (76)
                                                    8417   (78)       8439   (79)       8444   (79)       8462   (79)       8835   (86)
                                                    8849   (86)       8940   (87)       8953   (87)       9006   (87)       9017   (87)
                                                    9028   (87)       9036   (87)       9152   (88)       9155   (88)       9217   (89)
                                                    9244   (89)       9286   (89)       9324   (89)       9328   (89)       9353   (89)
                                                    9538   (92)       9540   (92)       9571   (93)       9574   (93)       9577   (93)
                                                    9585   (93)       9589   (93)       9595   (93)
LDAX               1              856    (3)        4246   (34)       4249   (34)       4432   (36)       4656   (39)       4673   (39)
                                                    4689   (39)       4948   (41)       6173   (53)       6273   (54)       6294   (54)
                                                    6305   (54)       6313   (54)       6331   (54)       6334   (54)       6447   (56)
                                                    6565   (57)       6858   (61)       7039   (63)       7053   (63)       8074   (71)
                                                    8078   (71)       8787   (85)       8802   (85)       9226   (89)       9309   (89)
                                                    9399   (90)       9434   (90)       9536   (92)       9729   (95)       9735   (95)
                                                    9741   (95)       9747   (95)       9766   (95)
LHLD               1              856    (3)        2267   (7)        2301   (7)        2331   (7)        2369   (7)        2465   (8)
                                                    2477   (8)        2627   (11)       2806   (14)       3006   (17)       3391   (20)
                                                    3474   (21)       3604   (23)       3814   (24)       3821   (24)       3828   (24)
                                                    3835   (24)       3917   (26)       3983   (27)       4569   (38)       4687   (39)
                                                    5105   (44)       5797   (49)       5799   (49)       5803   (49)       5806   (49)
                                                    5823   (49)       5919   (50)       5922   (50)       6051   (51)       6068   (51)
                                                    6372   (55)       6434   (56)       6440   (56)       6461   (56)       6463   (56)
                                                    6551   (57)       6554   (57)       6564   (57)       6684   (58)       6686   (58)
                                                    6789   (60)       6834   (61)       6837   (61)       6847   (61)       6850   (61)
                                                    6909   (62)       6923   (62)       6926   (62)       6999   (63)       7013   (63)
                                                    7016   (63)       7027   (63)       7030   (63)       7140   (64)       7142   (64)
                                                    7331   (65)       7524   (66)       7727   (67)       7740   (67)       7742   (67)
                                                    7745   (67)       7751   (67)       7830   (68)       7913   (70)       7916   (70)
                                                    7938   (70)       7940   (70)       7950   (70)       7963   (70)       7970   (70)
                                                    7984   (70)       8044   (71)       8047   (71)       8057   (71)       8059   (71)
                                                    8140   (73)       8142   (73)       8387   (77)       8389   (77)       8416   (78)
                                                    8921   (87)       8923   (87)       8926   (87)       9035   (87)       9141   (88)
                                                    9143   (88)       9147   (88)       9216   (89)       9583   (93)
LXI                1              856    (3)        1374   (5)        1385   (5)        1393   (5)        2269   (7)        2270   (7)
                                                    2314   (7)        2361   (7)        2362   (7)        2378   (7)        2379   (7)
                                                    2380   (7)        2381   (7)        2568   (10)       2569   (10)       2570   (10)
                                                    2571   (10)       2572   (10)       2708   (12)       2710   (12)       2807   (14)
                                                    2893   (15)       2942   (16)       3207   (20)       3230   (20)       3252   (20)
                                                    3253   (20)       3260   (20)       3262   (20)       3267   (20)       3305   (20)
                                                    3307   (20)       3313   (20)       3314   (20)       3323   (20)       3358   (20)
                                                    3369   (20)       3403   (20)       3521   (22)       3606   (23)       3627   (23)
                                                    3637   (23)       3650   (23)       3661   (23)       3679   (23)       3680   (23)
```

```
                                                 3811   (24)      3812   (24)      3813   (24)      3817   (24)      3856   (24)
                                                 3862   (24)      3863   (24)      3894   (25)      4009   (28)      4011   (28)
                                                 4040   (29)      4042   (29)      4122   (32)      4129   (32)      4137   (32)
                                                 4157   (33)      4209   (34)      4210   (34)      4217   (34)      4218   (34)
                                                 4219   (34)      4231   (34)      4241   (34)      4251   (34)      4254   (34)
                                                 4265   (34)      4272   (34)      4309   (35)      4311   (35)      4313   (35)
                                                 4334   (35)      4616   (39)      4652   (39)      4672   (39)      4685   (39)
                                                 4954   (41)      4957   (41)      4983   (41)      4987   (41)      4994   (41)
                                                 5004   (41)      5480   (47)      5481   (47)      5575   (47)      5589   (47)
                                                 5591   (47)      5601   (47)      5617   (47)      5627   (47)      5639   (47)
                                                 5645   (47)      5827   (49)      5849   (49)      5857   (49)      6171   (53)
                                                 6428   (56)      6494   (56)      6503   (56)      7036   (63)      7050   (63)
                                                 7131   (64)      7297   (65)      7320   (65)      7426   (66)      7490   (66)
                                                 7491   (66)      7507   (66)      7523   (66)      7529   (66)      7629   (67)
                                                 7680   (67)      7681   (67)      7682   (67)      7686   (67)      7687   (67)
                                                 7693   (67)      7726   (67)      7735   (67)      7736   (67)      7737   (67)
                                                 7738   (67)      7739   (67)      7749   (67)      7750   (67)      7756   (67)
                                                 7760   (67)      7761   (67)      7771   (67)      7780   (67)      7781   (67)
                                                 7790   (67)      7791   (67)      7924   (70)      7933   (70)      8069   (71)
                                                 8247   (76)      8260   (76)      8261   (76)      8262   (76)      8267   (76)
                                                 8346   (76)      8442   (79)      8443   (79)      8452   (79)      8455   (79)
                                                 8460   (79)      8497   (80)      8557   (81)      8671   (83)      8676   (83)
                                                 8724   (84)      8729   (84)      8782   (85)      8783   (85)      8845   (86)
                                                 8916   (87)      8917   (87)      8930   (87)      8934   (87)      8935   (87)
                                                 8939   (87)      8945   (87)      8946   (87)      8947   (87)      8951   (87)
                                                 8952   (87)      8956   (87)      8957   (87)      8958   (87)      8963   (87)
                                                 8973   (87)      8983   (87)      8993   (87)      9011   (87)      9042   (87)
                                                 9043   (87)      9049   (87)      9076   (87)      9081   (87)      9090   (87)
                                                 9091   (87)      9099   (87)      9139   (88)      9140   (88)      9150   (88)
                                                 9224   (89)      9249   (89)      9259   (89)      9272   (89)      9287   (89)
                                                 9337   (89)      9396   (90)      9431   (90)      9519   (92)      9628   (94)
                                                 9629   (94)      9633   (94)      9652   (94)      9655   (94)      9656   (94)
                                                 9665   (94)      9706   (95)      9726   (95)      9727   (95)      9728   (95)
                                                 9835   (96)      9836   (96)      9843   (96)      9845   (96)      9864   (97)
                                                 9867   (97)      9892   (98)      9893   (98)      9900   (98)      9902   (98)
MOV                1            856     (3)      2277   (7)       2280   (7)       2288   (7)       2294   (7)       2383   (7)
                                                 2467   (8)       2628   (11)      2805   (14)      2821   (14)      2833   (14)
                                                 2892   (15)      2897   (15)      2899   (15)      2941   (16)      2946   (16)
                                                 2948   (16)      2951   (16)      3050   (18)      3109   (19)      3212   (20)
                                                 3217   (20)      3223   (20)      3228   (20)      3243   (20)      3269   (20)
                                                 3292   (20)      3324   (20)      3349   (20)      3350   (20)      3354   (20)
                                                 3361   (20)      3363   (20)      3364   (20)      3366   (20)      3370   (20)
                                                 3373   (20)      3384   (20)      3400   (20)      3417   (20)      3419   (20)
                                                 3420   (20)      3422   (20)      3476   (21)      3524   (22)      3629   (23)
                                                 3632   (23)      3634   (23)      3642   (23)      3645   (23)      3678   (23)
                                                 3685   (23)      3688   (23)      3699   (23)      3718   (23)      3801   (24)
                                                 3822   (24)      3824   (24)      3833   (24)      4156   (33)      4239   (34)
                                                 4247   (34)      4250   (34)      4266   (34)      4269   (34)      4319   (35)
                                                 4321   (35)      4337   (35)      4339   (35)      4344   (35)      4349   (35)
                                                 4356   (35)      4446   (36)      4514   (37)      4523   (37)      4570   (38)
                                                 4628   (39)      4675   (39)      4690   (39)      4775   (40)      4777   (40)
                                                 4782   (40)      4838   (40)      4843   (40)      4847   (40)      4851   (40)
                                                 4859   (40)      4869   (40)      4880   (40)      4989   (41)      4999   (41)
                                                 5152   (45)      5170   (45)      5174   (45)      5183   (45)      5196   (45)
                                                 5205   (45)      5212   (45)      5215   (45)      5224   (45)      5243   (45)
```

```
                                                  5248   (45)      5251   (45)      5277   (45)      5285   (45)      5289   (45)
                                                  5350   (46)      5375   (46)      5380   (46)      5384   (46)      5393   (46)
                                                  5400   (46)      5402   (46)      5413   (46)      5417   (46)      5487   (47)
                                                  5521   (47)      5525   (47)      5532   (47)      5536   (47)      5582   (47)
                                                  5586   (47)      5716   (48)      5810   (49)      5814   (49)      5825   (49)
                                                  5828   (49)      5832   (49)      5836   (49)      5852   (49)      5853   (49)
                                                  5855   (49)      5856   (49)      5912   (50)      5913   (50)      5928   (50)
                                                  5930   (50)      6021   (51)      6041   (51)      6174   (53)      6183   (53)
                                                  6191   (53)      6200   (53)      6205   (53)      6217   (53)      6220   (53)
                                                  6274   (54)      6283   (54)      6286   (54)      6343   (54)      6345   (54)
                                                  6448   (56)      6477   (56)      6495   (56)      6563   (57)      6570   (57)
                                                  6576   (57)      6582   (57)      6626   (58)      6639   (58)      6646   (58)
                                                  6662   (58)      6665   (58)      6857   (61)      6917   (62)      6919   (62)
                                                  6935   (62)      7007   (63)      7009   (63)      7038   (63)      7052   (63)
                                                  7065   (63)      7150   (64)      7152   (64)      7162   (64)      7170   (64)
                                                  7199   (64)      7311   (65)      7314   (65)      7448   (66)      7451   (66)
                                                  7497   (66)      7504   (66)      7505   (66)      7525   (66)      7527   (66)
                                                  7539   (66)      7551   (66)      7611   (67)      7622   (67)      7623   (67)
                                                  7714   (67)      7752   (67)      7754   (67)      7824   (68)      7926   (70)
                                                  7951   (70)      7954   (70)      7956   (70)      7965   (70)      7966   (70)
                                                  7971   (70)      7974   (70)      7976   (70)      7986   (70)      7988   (70)
                                                  8075   (71)      8079   (71)      8109   (72)      8111   (72)      8144   (73)
                                                  8146   (73)      8242   (76)      8244   (76)      8249   (76)      8254   (76)
                                                  8271   (76)      8274   (76)      8275   (76)      8281   (76)      8287   (76)
                                                  8297   (76)      8336   (76)      8339   (76)      8342   (76)      8345   (76)
                                                  8347   (76)      8349   (76)      8448   (79)      8449   (79)      8502   (80)
                                                  8504   (80)      8551   (81)      8558   (81)      8564   (81)      8566   (81)
                                                  8572   (81)      8574   (81)      8576   (81)      8579   (81)      8624   (82)
                                                  8670   (83)      8680   (83)      8723   (84)      8733   (84)      8781   (85)
                                                  8784   (85)      8999   (87)      9037   (87)      9062   (87)      9070   (87)
                                                  9078   (87)      9082   (87)      9085   (87)      9087   (87)      9227   (89)
                                                  9236   (89)      9247   (89)      9250   (89)      9253   (89)      9294   (89)
                                                  9308   (89)      9312   (89)      9313   (89)      9315   (89)      9343   (89)
                                                  9402   (90)      9404   (90)      9417   (90)      9419   (90)      9420   (90)
                                                  9422   (90)      9425   (90)      9437   (90)      9439   (90)      9450   (90)
                                                  9452   (90)      9453   (90)      9455   (90)      9458   (90)      9482   (91)
                                                  9520   (92)      9521   (92)      9525   (92)      9529   (92)      9532   (92)
                                                  9535   (92)      9539   (92)      9636   (94)      9645   (94)      9650   (94)
                                                  9708   (95)      9714   (95)      9716   (95)      9719   (95)      9789   (95)
                                                  9793   (95)      9796   (95)      9797   (95)      9799   (95)      9801   (95)
MVI                1             856    (3)       1394   (5)       1398   (5)       1399   (5)       1405   (5)       1406   (5)
                                                  1414   (5)       1422   (5)       2283   (7)       2315   (7)       2316   (7)
                                                  2380   (7)       2468   (8)       2570   (10)      2572   (10)      2706   (12)
                                                  2818   (14)      2819   (14)      2823   (14)      2824   (14)      2828   (14)
                                                  2949   (16)      3000   (17)      3059   (18)      3066   (18)      3118   (19)
                                                  3125   (19)      3207   (20)      3254   (20)      3255   (20)      3260   (20)
                                                  3262   (20)      3268   (20)      3278   (20)      3305   (20)      3307   (20)
                                                  3308   (20)      3313   (20)      3317   (20)      3322   (20)      3333   (20)
                                                  3354   (20)      3369   (20)      3378   (20)      3393   (20)      3395   (20)
                                                  3404   (20)      3477   (21)      3520   (22)      3526   (22)      3606   (23)
                                                  3619   (23)      3630   (23)      3671   (23)      3672   (23)      3689   (23)
                                                  3719   (23)      3796   (24)      3812   (24)      3825   (24)      3834   (24)
                                                  3844   (24)      3862   (24)      3968   (27)      3969   (27)      3971   (27)
                                                  4009   (28)      4011   (28)      4012   (28)      4040   (29)      4042   (29)
                                                  4043   (29)      4066   (30)      4099   (31)      4121   (32)      4128   (32)
```

```
                                                 4153   (33)      4154   (33)      4157   (33)      4219   (34)      4240   (34)
                                                 4328   (35)      4430   (36)      4431   (36)      4650   (39)      4651   (39)
                                                 4688   (39)      4769   (40)      4776   (40)      4841   (40)      4846   (40)
                                                 4849   (40)      4853   (40)      4862   (40)      4988   (41)      4994   (41)
                                                 5150   (45)      5151   (45)      5202   (45)      5229   (45)      5232   (45)
                                                 5235   (45)      5241   (45)      5348   (46)      5349   (46)      5480   (47)
                                                 5485   (47)      5486   (47)      5544   (47)      5550   (47)      5556   (47)
                                                 5562   (47)      5568   (47)      5575   (47)      5576   (47)      5589   (47)
                                                 5591   (47)      5592   (47)      5594   (47)      5601   (47)      5602   (47)
                                                 5610   (47)      5617   (47)      5618   (47)      5620   (47)      5627   (47)
                                                 5628   (47)      5630   (47)      5831   (49)      5850   (49)      5917   (50)
                                                 6119   (52)      6172   (53)      6199   (53)      6204   (53)      6275   (54)
                                                 6278   (54)      6342   (54)      6426   (56)      6446   (56)      6476   (56)
                                                 6492   (56)      6504   (56)      6505   (56)      7124   (64)      7163   (64)
                                                 7461   (66)      7464   (66)      7472   (66)      7475   (66)      7507   (66)
                                                 7529   (66)      7575   (66)      7577   (66)      7682   (67)      7693   (67)
                                                 7735   (67)      7737   (67)      7749   (67)      7756   (67)      7771   (67)
                                                 7773   (67)      7781   (67)      7783   (67)      7791   (67)      7925   (70)
                                                 8042   (71)      8245   (76)      8268   (76)      8269   (76)      8420   (78)
                                                 8424   (78)      8426   (78)      8457   (79)      8498   (80)      8499   (80)
                                                 8500   (80)      8501   (80)      8550   (81)      8559   (81)      8623   (82)
                                                 8930   (87)      8935   (87)      8943   (87)      8945   (87)      8952   (87)
                                                 8976   (87)      8986   (87)      9083   (87)      9150   (88)      9225   (89)
                                                 9251   (89)      9338   (89)      9397   (90)      9398   (90)      9414   (90)
                                                 9415   (90)      9432   (90)      9433   (90)      9483   (91)      9533   (92)
                                                 9651   (94)      9709   (95)      9765   (95)      9794   (95)      9800   (95)
                                                 9837   (96)      9838   (96)      9843   (96)      9845   (96)      9846   (96)
                                                 9866   (97)      9894   (98)      9895   (98)      9900   (98)      9902   (98)
                                                 9903   (98)
NOP               1             856    (3)
ORA               1             856    (3)       2304   (7)       3111   (19)      3401   (20)      3686   (23)      3700   (23)
                                                 3780   (24)      3802   (24)      4119   (32)      4433   (36)      4677   (39)
                                                 4839   (40)      4848   (40)      4868   (40)      4871   (40)      5191   (45)
                                                 5206   (45)      5256   (45)      6276   (54)      6314   (54)      6471   (56)
                                                 6578   (57)      6628   (58)      6640   (58)      6647   (58)      6780   (60)
                                                 7455   (66)      7536   (66)      7922   (70)      8034   (71)      8298   (76)
                                                 8552   (81)      8785   (85)      8836   (86)      8941   (87)      8954   (87)
                                                 9000   (87)      9063   (87)      9071   (87)      9245   (89)      9310   (89)
                                                 9351   (89)      9354   (89)      9421   (90)      9454   (90)
ORI               1             856    (3)       1420   (5)       2281   (7)       2750   (13)      2904   (15)      2947   (16)
                                                 2955   (16)      3891   (25)      3956   (27)      4071   (30)      4077   (30)
                                                 4096   (31)      4320   (35)      4338   (35)      4756   (40)      4764   (40)
                                                 4806   (40)      4814   (40)      4822   (40)      4830   (40)      4895   (40)
                                                 4971   (41)      4977   (41)      5057   (42)      5099   (44)      5103   (44)
                                                 5265   (45)      5271   (45)      6126   (52)      6772   (60)      6783   (60)
                                                 6940   (62)      7070   (63)      7193   (64)      7479   (66)      7698   (67)
                                                 7706   (67)      8172   (74)      8199   (75)      8353   (76)      8463   (79)
                                                 9424   (90)      9457   (90)      9596   (93)
PCHL              1             856    (3)       4255   (34)      5860   (49)
POP               1             856    (3)       1397   (5)       1404   (5)       1408   (5)       1429   (5)       2703   (12)
                                                 2827   (14)      2848   (14)      3055   (18)      3063   (18)      3114   (19)
                                                 3122   (19)      3224   (20)      3231   (20)      3238   (20)      3263   (20)
                                                 3278   (20)      3282   (20)      3286   (20)      3308   (20)      3317   (20)
                                                 3318   (20)      3333   (20)      3337   (20)      3341   (20)      3374   (20)
                                                 3380   (20)      3406   (20)      3425   (20)      3466   (21)      3479   (21)
```

```
                                                         3481    (21)       3482    (21)       3485    (21)       3618    (23)       3641    (23)
                                                         3654    (23)       3690    (23)       3693    (23)       3706    (23)       3707    (23)
                                                         3709    (23)       3710    (23)       3717    (23)       3722    (23)       3985    (27)
                                                         4012    (28)       4013    (28)       4014    (28)       4015    (28)       4016    (28)
                                                         4043    (29)       4044    (29)       4045    (29)       4046    (29)       4047    (29)
                                                         4130    (32)       4131    (32)       4132    (32)       4158    (33)       4281    (34)
                                                         4377    (35)       4573    (38)       4655    (39)       4666    (39)       4676    (39)
                                                         4696    (39)       4995    (41)       4996    (41)       5005    (41)       5006    (41)
                                                         5484    (47)       5576    (47)       5592    (47)       5602    (47)       5618    (47)
                                                         5628    (47)       6299    (54)       6319    (54)       6325    (54)       6442    (56)
                                                         6465    (56)       6502    (56)       6577    (57)       6580    (57)       6661    (58)
                                                         6855    (61)       7049    (63)       7154    (64)       7768    (67)       7777    (67)
                                                         7787    (67)       7861    (69)       7960    (70)       7981    (70)       7987    (70)
                                                         7989    (70)       7990    (70)       8050    (71)       8062    (71)       8068    (71)
                                                         8456    (79)       8679    (83)       8732    (84)       8970    (87)       8980    (87)
                                                         8990    (87)       8998    (87)       9016    (87)       9020    (87)       9024    (87)
                                                         9044    (87)       9045    (87)       9051    (87)       9052    (87)       9056    (87)
                                                         9077    (87)       9092    (87)       9093    (87)       9098    (87)       9212    (89)
                                                         9213    (89)       9235    (89)       9281    (89)       9302    (89)       9303    (89)
                                                         9316    (89)       9319    (89)       9335    (89)       9336    (89)       9349    (89)
                                                         9350    (89)       9412    (90)       9429    (90)       9447    (90)       9488    (91)
                                                         9630    (94)       9634    (94)       9635    (94)       9657    (94)       9658    (94)
                                                         9663    (94)       9666    (94)       9750    (95)       9751    (95)       9787    (95)
                                                         9788    (95)       9805    (95)       9808    (95)       9846    (96)       9865    (97)
                                                         9903    (98)
PUSH                 1               856     (3)         1384    (5)        1395    (5)        1396    (5)        1400    (5)        1402    (5)
                                                         2695    (12)       2800    (14)       2817    (14)       3043    (18)       3058    (18)
                                                         3103    (19)       3117    (19)       3213    (20)       3225    (20)       3229    (20)
                                                         3236    (20)       3248    (20)       3251    (20)       3257    (20)       3278    (20)
                                                         3284    (20)       3297    (20)       3298    (20)       3308    (20)       3317    (20)
                                                         3333    (20)       3339    (20)       3375    (20)       3414    (20)       3462    (21)
                                                         3467    (21)       3473    (21)       3475    (21)       3605    (23)       3626    (23)
                                                         3643    (23)       3655    (23)       3683    (23)       3684    (23)       3967    (27)
                                                         4004    (28)       4005    (28)       4006    (28)       4007    (28)       4012    (28)
                                                         4035    (29)       4036    (29)       4037    (29)       4038    (29)       4043    (29)
                                                         4125    (32)       4126    (32)       4127    (32)       4155    (33)       4253    (34)
                                                         4375    (35)       4568    (38)       4649    (39)       4653    (39)       4658    (39)
                                                         4686    (39)       4991    (41)       4992    (41)       5002    (41)       5003    (41)
                                                         5478    (47)       5576    (47)       5592    (47)       5602    (47)       5618    (47)
                                                         5628    (47)       5858    (49)       6297    (54)       6309    (54)       6438    (56)
                                                         6460    (56)       6493    (56)       6507    (56)       6569    (57)       6655    (58)
                                                         6839    (61)       7035    (63)       7147    (64)       7767    (67)       7776    (67)
                                                         7786    (67)       7937    (70)       7949    (70)       7957    (70)       7977    (70)
                                                         7978    (70)       7992    (70)       8032    (71)       8454    (79)       8675    (83)
                                                         8728    (84)       8915    (87)       8969    (87)       8979    (87)       8989    (87)
                                                         9002    (87)       9005    (87)       9022    (87)       9034    (87)       9040    (87)
                                                         9041    (87)       9047    (87)       9048    (87)       9073    (87)       9075    (87)
                                                         9079    (87)       9095    (87)       9215    (89)       9218    (89)       9219    (89)
                                                         9220    (89)       9221    (89)       9282    (89)       9283    (89)       9285    (89)
                                                         9307    (89)       9311    (89)       9342    (89)       9392    (90)       9413    (90)
                                                         9430    (90)       9481    (91)       9625    (94)       9627    (94)       9631    (94)
                                                         9632    (94)       9641    (94)       9644    (94)       9660    (94)       9664    (94)
                                                         9724    (95)       9725    (95)       9762    (95)       9763    (95)       9802    (95)
                                                         9803    (95)       9840    (96)       9846    (96)       9863    (97)       9897    (98)
                                                         9903    (98)
```

```
RAD50              2           610     (2)        1526    (6)         1527    (6)         1528    (6)         1529    (6)         1530    (6)
                                                  1531    (6)         1532    (6)         1533    (6)         1534    (6)         1535    (6)
                                                  1536    (6)         1537    (6)         1538    (6)         1539    (6)         1540    (6)
RAL                1           856     (3)        4657    (39)
RAR                1           856     (3)        4123    (32)
RC                 1           856     (3)
READ_RECORD        2           786     (2)        3207    (20)        3230    (20)        3369    (20)        3403    (20)        3606    (23)
                                                  3661    (23)
RET                1           856     (3)        1381    (5)         2478    (8)         2527    (9)         2573    (10)        2637    (11)
                                                  2711    (12)        2754    (13)        2850    (14)        2907    (15)        2958    (16)
                                                  3008    (17)        3069    (18)        3128    (19)        3407    (20)        3488    (21)
                                                  3537    (22)        3621    (23)        3713    (23)        3727    (23)        3846    (24)
                                                  3896    (25)        3919    (26)        3986    (27)        4017    (28)        4048    (29)
                                                  4081    (30)        4105    (31)        4138    (32)        4162    (33)        4276    (34)
                                                  4386    (35)        4393    (35)        4472    (36)        4478    (36)        4535    (37)
                                                  4537    (37)        4574    (38)        4699    (39)        4705    (39)        4889    (40)
                                                  4899    (40)        5013    (41)        5018    (41)        5061    (42)        5109    (44)
                                                  5298    (45)        5304    (45)        5427    (46)        5433    (46)        5654    (47)
                                                  5658    (47)        5661    (47)        5719    (48)        5725    (48)        5931    (50)
                                                  6077    (51)        6129    (52)        6221    (53)        6336    (54)        6346    (54)
                                                  6374    (55)        6512    (56)        6586    (57)        6689    (58)        6732    (59)
                                                  6791    (60)        6868    (61)        6944    (62)        6953    (62)        7074    (63)
                                                  7083    (63)        7188    (64)        7202    (64)        7339    (65)        7531    (66)
                                                  7556    (66)        7721    (67)        7832    (68)        7862    (69)        7995    (70)
                                                  8080    (71)        8112    (72)        8147    (73)        8174    (74)        8201    (75)
                                                  8365    (76)        8392    (77)        8428    (78)        8465    (79)        8508    (80)
                                                  8583    (81)        8586    (81)        8633    (82)        8686    (83)        8739    (84)
                                                  8796    (85)        8798    (85)        8852    (86)        9103    (87)        9168    (88)
                                                  9358    (89)        9459    (90)        9489    (91)        9545    (92)        9547    (92)
                                                  9599    (93)        9668    (94)        9811    (95)        9847    (96)        9868    (97)
                                                  9904    (98)
RIM                1           856     (3)
RLC                1           856     (3)        2698    (12)        3628    (23)        9080    (87)        9486    (91)        9526    (92)
                                                  9646    (94)        9647    (94)        9648    (94)        9707    (95)        9790    (95)
                                                  9791    (95)
RM                 1           856     (3)
RNC                1           856     (3)
RNZ                1           856     (3)
RP                 1           856     (3)
RPE                1           856     (3)
RPO                1           856     (3)
RRC                1           856     (3)        5711    (48)
RST                1           856     (3)
RZ                 1           856     (3)
SBB                1           856     (3)
SBI                1           856     (3)
SGL_MIC_INSTR      1           707     (2)        3278    (20)        3308    (20)        3317    (20)        3333    (20)        3971    (27)
                                                  4012    (28)        4043    (29)        4099    (31)        5576    (47)        5592    (47)
                                                  5602    (47)        5618    (47)        5628    (47)        7575    (66)        9846    (96)
                                                  9903    (98)
SGL_TICK           1           730     (2)        7577    (66)
SHLD               1           856     (3)        1376    (5)         1389    (5)         2269    (7)         2270    (7)         2302    (7)
                                                  2325    (7)         2332    (7)         2361    (7)         2362    (7)         2370    (7)
                                                  2378    (7)         2379    (7)         2380    (7)         2381    (7)         2463    (8)
                                                  2466    (8)         2470    (8)         2568    (10)        2569    (10)        2570    (10)
```

```
                                              2571  (10)      2572  (10)      2631  (11)      2708  (12)      2710  (12)
                                              3260  (20)      3262  (20)      3305  (20)      3307  (20)      3313  (20)
                                              3315  (20)      3316  (20)      3354  (20)      3360  (20)      3368  (20)
                                              3392  (20)      3416  (20)      3424  (20)      3450  (21)      3636  (23)
                                              3637  (23)      3649  (23)      3650  (23)      3724  (23)      3811  (24)
                                              3812  (24)      3813  (24)      3815  (24)      3817  (24)      3826  (24)
                                              3830  (24)      3837  (24)      3856  (24)      3862  (24)      3863  (24)
                                              3895  (25)      3918  (26)      3970  (27)      3984  (27)      4009  (28)
                                              4011  (28)      4040  (29)      4042  (29)      4128  (32)      4129  (32)
                                              4137  (32)      4157  (33)      4209  (34)      4210  (34)      4217  (34)
                                              4218  (34)      4219  (34)      4231  (34)      4310  (35)      4572  (38)
                                              4695  (39)      4954  (41)      4957  (41)      4983  (41)      4994  (41)
                                              5004  (41)      5106  (44)      5480  (47)      5482  (47)      5483  (47)
                                              5575  (47)      5589  (47)      5591  (47)      5601  (47)      5617  (47)
                                              5627  (47)      5824  (49)      5844  (49)      5930  (50)      6052  (51)
                                              6069  (51)      6070  (51)      6373  (55)      6688  (58)      6790  (60)
                                              7132  (64)      7133  (64)      7134  (64)      7135  (64)      7198  (64)
                                              7332  (65)      7427  (66)      7490  (66)      7491  (66)      7507  (66)
                                              7523  (66)      7528  (66)      7529  (66)      7630  (67)      7680  (67)
                                              7681  (67)      7682  (67)      7686  (67)      7687  (67)      7693  (67)
                                              7726  (67)      7728  (67)      7735  (67)      7736  (67)      7737  (67)
                                              7738  (67)      7739  (67)      7748  (67)      7749  (67)      7750  (67)
                                              7755  (67)      7756  (67)      7760  (67)      7762  (67)      7763  (67)
                                              7770  (67)      7771  (67)      7779  (67)      7780  (67)      7781  (67)
                                              7789  (67)      7790  (67)      7791  (67)      7831  (68)      8260  (76)
                                              8261  (76)      8262  (76)      8391  (77)      8420  (78)      8424  (78)
                                              8426  (78)      8442  (79)      8452  (79)      8455  (79)      8457  (79)
                                              8460  (79)      8677  (83)      8678  (83)      8730  (84)      8731  (84)
                                              8845  (86)      8916  (87)      8917  (87)      8929  (87)      8930  (87)
                                              8934  (87)      8935  (87)      8939  (87)      8945  (87)      8946  (87)
                                              8947  (87)      8951  (87)      8952  (87)      8956  (87)      8957  (87)
                                              8964  (87)      8965  (87)      8972  (87)      8982  (87)      8992  (87)
                                              9042  (87)      9043  (87)      9076  (87)      9089  (87)      9090  (87)
                                              9091  (87)      9099  (87)      9139  (88)      9140  (88)      9149  (88)
                                              9150  (88)      9210  (89)      9628  (94)      9629  (94)      9633  (94)
                                              9654  (94)      9655  (94)      9656  (94)      9665  (94)      9810  (95)
                                              9843  (96)      9845  (96)      9864  (97)      9866  (97)      9867  (97)
                                              9900  (98)      9902  (98)
SIM              1            856    (3)
SPHL             1            856    (3)      2268  (7)       3007  (17)
STA              1            856    (3)      1415  (5)       1416  (5)       1419  (5)       1421  (5)       1424  (5)
                                              2272  (7)       2282  (7)       2296  (7)       2310  (7)       2340  (7)
                                              2388  (7)       2389  (7)       2390  (7)       2464  (8)       2473  (8)
                                              2520  (9)       2629  (11)      2697  (12)      2699  (12)      2707  (12)
                                              2751  (13)      2803  (14)      2849  (14)      2889  (15)      2906  (15)
                                              2939  (16)      2957  (16)      3001  (17)      3002  (17)      3005  (17)
                                              3046  (18)      3047  (18)      3053  (18)      3056  (18)      3057  (18)
                                              3062  (18)      3065  (18)      3068  (18)      3106  (19)      3107  (19)
                                              3112  (19)      3115  (19)      3116  (19)      3121  (19)      3124  (19)
                                              3127  (19)      3250  (20)      3260  (20)      3262  (20)      3278  (20)
                                              3281  (20)      3305  (20)      3307  (20)      3308  (20)      3313  (20)
                                              3317  (20)      3333  (20)      3354  (20)      3371  (20)      3383  (20)
                                              3527  (22)      3529  (22)      3536  (22)      3646  (23)      3800  (24)
                                              3806  (24)      3843  (24)      3845  (24)      3858  (24)      3893  (25)
                                              3916  (26)      3953  (27)      3959  (27)      3961  (27)      3962  (27)
```

```
                                                 3971   (27)      3978   (27)      3979   (27)      4009   (28)      4011   (28)
                                                 4012   (28)      4040   (29)      4042   (29)      4043   (29)      4068   (30)
                                                 4073   (30)      4074   (30)      4078   (30)      4080   (30)      4098   (31)
                                                 4099   (31)      4103   (31)      4104   (31)      4128   (32)      4157   (33)
                                                 4204   (34)      4206   (34)      4207   (34)      4216   (34)      4757   (40)
                                                 4762   (40)      4768   (40)      4770   (40)      4773   (40)      4774   (40)
                                                 4807   (40)      4815   (40)      4823   (40)      4831   (40)      4844   (40)
                                                 4860   (40)      4872   (40)      4881   (40)      4896   (40)      4972   (41)
                                                 4974   (41)      4979   (41)      4981   (41)      4993   (41)      5058   (42)
                                                 5101   (44)      5104   (44)      5171   (45)      5176   (45)      5192   (45)
                                                 5203   (45)      5204   (45)      5213   (45)      5216   (45)      5242   (45)
                                                 5249   (45)      5252   (45)      5257   (45)      5266   (45)      5272   (45)
                                                 5273   (45)      5286   (45)      5287   (45)      5290   (45)      5291   (45)
                                                 5367   (46)      5370   (46)      5371   (46)      5381   (46)      5382   (46)
                                                 5385   (46)      5386   (46)      5405   (46)      5414   (46)      5419   (46)
                                                 5480   (47)      5575   (47)      5576   (47)      5587   (47)      5589   (47)
                                                 5591   (47)      5592   (47)      5601   (47)      5602   (47)      5605   (47)
                                                 5608   (47)      5617   (47)      5618   (47)      5627   (47)      5628   (47)
                                                 5638   (47)      5930   (50)      6003   (51)      6013   (51)      6029   (51)
                                                 6056   (51)      6064   (51)      6074   (51)      6076   (51)      6120   (52)
                                                 6128   (52)      6265   (54)      6332   (54)      6335   (54)      6726   (59)
                                                 6731   (59)      6773   (60)      6784   (60)      6785   (60)      6918   (62)
                                                 6920   (62)      6934   (62)      6943   (62)      6951   (62)      7008   (63)
                                                 7010   (63)      7064   (63)      7073   (63)      7081   (63)      7125   (64)
                                                 7127   (64)      7151   (64)      7153   (64)      7186   (64)      7196   (64)
                                                 7201   (64)      7335   (65)      7425   (66)      7431   (66)      7434   (66)
                                                 7462   (66)      7465   (66)      7473   (66)      7476   (66)      7480   (66)
                                                 7496   (66)      7503   (66)      7506   (66)      7571   (66)      7575   (66)
                                                 7577   (66)      7585   (66)      7595   (67)      7598   (67)      7604   (67)
                                                 7606   (67)      7626   (67)      7635   (67)      7648   (67)      7692   (67)
                                                 7697   (67)      7700   (67)      7707   (67)      7709   (67)      7734   (67)
                                                 7860   (69)      8173   (74)      8200   (75)      8277   (76)      8289   (76)
                                                 8291   (76)      8293   (76)      8296   (76)      8356   (76)      8362   (76)
                                                 8464   (79)      8669   (83)      8722   (84)      8841   (86)      8851   (86)
                                                 8944   (87)      9004   (87)      9019   (87)      9157   (88)      9211   (89)
                                                 9214   (89)      9223   (89)      9239   (89)      9326   (89)      9330   (89)
                                                 9523   (92)      9524   (92)      9528   (92)      9537   (92)      9598   (93)
                                                 9834   (96)      9843   (96)      9845   (96)      9846   (96)      9891   (98)
                                                 9900   (98)      9902   (98)      9903   (98)
STAX              1             856    (3)       2822   (14)      2834   (14)      3270   (20)      3325   (20)      4618   (39)
                                                 4620   (39)      4622   (39)      4624   (39)      4659   (39)      4678   (39)
                                                 5837   (49)      6284   (54)      6287   (54)      6479   (56)      6497   (56)
                                                 6860   (61)      7041   (63)      7055   (63)      7928   (70)      8681   (83)
                                                 8734   (84)      9320   (89)      9344   (89)      9731   (95)      9737   (95)
                                                 9743   (95)      9749   (95)
STC               1             856    (3)       2526   (9)       3483   (21)      3486   (21)      3620   (23)      3712   (23)
                                                 3725   (23)      4385   (35)      4391   (35)      4470   (36)      4477   (36)
                                                 4533   (37)      4536   (37)      4697   (39)      4704   (39)      4888   (40)
                                                 4897   (40)      5011   (41)      5017   (41)      5059   (42)      5107   (44)
                                                 5296   (45)      5303   (45)      5425   (46)      5432   (46)      5653   (47)
                                                 5656   (47)      5717   (48)      5724   (48)      8582   (81)      8584   (81)
                                                 8794   (85)      8797   (85)      9158   (88)      9166   (88)      9543   (92)
                                                 9546   (92)
SUB               1             856    (3)       3832   (24)      9237   (89)      9798   (95)
SUI               1             856    (3)       4634   (39)      4641   (39)      5704   (48)      9260   (89)      9288   (89)
```

```
                                                9705  (95)
TYPE               2            432    (2)      2269  (7)       2270  (7)       2361  (7)       2362  (7)       2378  (7)
                                                2379  (7)       2381  (7)       2568  (10)      2569  (10)      2571  (10)
                                                2708  (12)      2710  (12)      3636  (23)      3637  (23)      3649  (23)
                                                3650  (23)      3811  (24)      3813  (24)      3817  (24)      3856  (24)
                                                3863  (24)      4129  (32)      4137  (32)      4209  (34)      4210  (34)
                                                4217  (34)      4218  (34)      4231  (34)      4954  (41)      4957  (41)
                                                4983  (41)      5004  (41)      7490  (66)      7491  (66)      7523  (66)
                                                7680  (67)      7681  (67)      7686  (67)      7687  (67)      7726  (67)
                                                7728  (67)      7736  (67)      7738  (67)      7739  (67)      7750  (67)
                                                7760  (67)      7780  (67)      7790  (67)      8260  (76)      8261  (76)
                                                8262  (76)      8391  (77)      8442  (79)      8452  (79)      8455  (79)
                                                8460  (79)      8845  (86)      8916  (87)      8917  (87)      8934  (87)
                                                8939  (87)      8946  (87)      8947  (87)      8951  (87)      8956  (87)
                                                8957  (87)      9042  (87)      9043  (87)      9076  (87)      9089  (87)
                                                9090  (87)      9091  (87)      9099  (87)      9139  (88)      9140  (88)
                                                9628  (94)      9629  (94)      9633  (94)      9654  (94)      9655  (94)
                                                9656  (94)      9665  (94)      9810  (95)      9864  (97)      9867  (97)
TYPEB              1            518    (2)      2380  (7)       2570  (10)      2572  (10)      3812  (24)      3862  (24)
                                                4219  (34)      7507  (66)      7693  (67)      7735  (67)      7737  (67)
                                                8420  (78)      8457  (79)      8935  (87)      8945  (87)
TYPEL              1            580    (2)      8426  (78)      8952  (87)      9866  (97)
TYPEN              1            490    (2)      4994  (41)
TYPEW              2            549    (2)      7529  (66)      7682  (67)      7749  (67)      7756  (67)      7771  (67)
                                                7781  (67)      7791  (67)      8424  (78)      8930  (87)      9150  (88)
TYPE_UNCOUNTED     1            464    (2)      4128  (32)
WRITE$_DCS         3            656    (2)      3259  (20)      3261  (20)      3304  (20)      3306  (20)      3312  (20)
                                                3354  (20)      4008  (28)      4010  (28)      4039  (29)      4041  (29)
                                                4157  (33)      5479  (47)      5574  (47)      5588  (47)      5590  (47)
                                                5600  (47)      5616  (47)      5626  (47)      5930  (50)      9842  (96)
                                                9844  (96)      9899  (98)      9901  (98)
XCHG               1            856    (3)      3635  (23)      3692  (23)      3721  (23)      3827  (24)      4245  (34)
                                                4379  (35)      4771  (40)      5149  (45)      5347  (46)      5798  (49)
                                                5802  (49)      5805  (49)      5859  (49)      5921  (50)      6272  (54)
                                                6433  (56)      6439  (56)      6462  (56)      6466  (56)      6553  (57)
                                                6561  (57)      6685  (58)      6836  (61)      6849  (61)      6925  (62)
                                                7015  (63)      7029  (63)      7141  (64)      7741  (67)      7744  (67)
                                                7915  (70)      7939  (70)      7942  (70)      7962  (70)      7983  (70)
                                                8046  (71)      8049  (71)      8058  (71)      8061  (71)      8141  (73)
                                                8388  (77)      8567  (81)      8632  (82)      8922  (87)      8925  (87)
                                                9088  (87)      9142  (88)      9146  (88)      9284  (89)      9717  (95)
                                                9764  (95)      9807  (95)
XRA                1            856    (3)      1401  (5)       2271  (7)       2303  (7)       2309  (7)       2387  (7)
                                                3249  (20)      3256  (20)      3528  (22)      3711  (23)      3831  (24)
                                                3857  (24)      4205  (34)      4617  (39)      4772  (40)      6002  (51)
                                                6506  (56)      7126  (64)      7764  (67)      8243  (76)      8447  (79)
                                                8840  (86)      8966  (87)      9003  (87)      9222  (89)      9318  (89)
                                                9730  (95)      9736  (95)      9742  (95)      9748  (95)      9839  (96)
                                                9896  (98)
XRI                1            856    (3)      2898  (15)      4079  (30)
XTHL               1            856    (3)      4273  (34)
```

```
                                   +---------------------------------+
                                   ! Directives Cross Reference !
                                   +---------------------------------+

DIRECTIVE       REFERENCES...
---------       -------------
.ASCIC          1468   (6)      1469   (6)      1485   (6)      1486   (6)      1487   (6)      1488   (6)      1489   (6)      1490   (6)
                1491   (6)      1492   (6)      1493   (6)      1494   (6)      1495   (6)      1496   (6)      1497   (6)      1520   (6)
                1566   (6)      1678   (6)      1679   (6)      1692   (6)      1695   (6)      1787   (6)      1788   (6)      1790   (6)
                1892   (6)      1893   (6)      1894   (6)      1895   (6)      1896   (6)      1897   (6)      1898   (6)      1906   (6)
                1907   (6)      1908   (6)      1922   (6)      1940   (6)      1941   (6)      1942   (6)      1943   (6)      1944   (6)
                1945   (6)      1946   (6)      1947   (6)      1948   (6)      1949   (6)      1950   (6)      1951   (6)      1952   (6)
                1958   (6)      1959   (6)      1960   (6)      1961   (6)      1962   (6)      1963   (6)      1964   (6)      1965   (6)
                1966   (6)      1967   (6)      1968   (6)      1970   (6)      1971   (6)      1972   (6)      1973   (6)      1974   (6)
                1975   (6)      1977   (6)      1978   (6)      1980   (6)      1981   (6)      1983   (6)      1984   (6)      2016   (6)
                2017   (6)      2018   (6)      2019   (6)      2020   (6)      2021   (6)      2022   (6)      2023   (6)      2031   (6)
                2032   (6)      2033   (6)      2034   (6)      2046   (6)      2047   (6)      2048   (6)      2049   (6)      2050   (6)
                2051   (6)      2052   (6)      2053   (6)      2061   (6)      2062   (6)      2063   (6)      2064   (6)      2070   (6)
                2071   (6)      2079   (6)      2080   (6)      2081   (6)      2082   (6)      2090   (6)      2091   (6)      2092   (6)
                2093   (6)      2105   (6)      2106   (6)      2107   (6)      2108   (6)      2109   (6)      2110   (6)      2111   (6)
                2112   (6)      2118   (6)      2119   (6)      2124   (6)      2129   (6)      2137   (6)      2138   (6)      2139   (6)
                2140   (6)      2146   (6)      2147   (6)      2155   (6)      2156   (6)      2157   (6)      2158   (6)      2166   (6)
                2167   (6)      2168   (6)      2169   (6)      2175   (6)
.ASCII          1713   (6)      1714   (6)      1715   (6)      1716   (6)      1717   (6)      1718   (6)      1719   (6)      1720   (6)
                1721   (6)      1722   (6)      1725   (6)      1726   (6)      1727   (6)      1728   (6)      1730   (6)      1731   (6)
                1732   (6)      1733   (6)      1734   (6)      1735   (6)      1736   (6)      1737   (6)      1739   (6)      1740   (6)
                1741   (6)      1742   (6)      1743   (6)      1744   (6)      1745   (6)      1746   (6)      1747   (6)      1748   (6)
                1749   (6)      1750   (6)      1751   (6)      1752   (6)      1753   (6)      1754   (6)      1755   (6)      1756   (6)
                1757   (6)      1758   (6)      1759   (6)      1760   (6)
.BLKB           1559   (6)      1573   (6)      1686   (6)      1712   (6)      1803   (6)      1844   (6)      1845   (6)      1869   (6)
                1870   (6)      1904   (6)      1936   (6)      1938   (6)      2181   (6)
.BYTE           1373   (5)      1374   (5)      1375   (5)      1376   (5)      1377   (5)      1381   (5)      1384   (5)      1385   (5)
                1386   (5)      1387   (5)      1388   (5)      1389   (5)      1393   (5)      1394   (5)      1395   (5)      1396   (5)
                1397   (5)      1398   (5)      1399   (5)      1400   (5)      1401   (5)      1402   (5)      1403   (5)      1404   (5)
                1405   (5)      1406   (5)      1407   (5)      1408   (5)      1409   (5)      1410   (5)      1414   (5)      1415   (5)
                1416   (5)      1417   (5)      1418   (5)      1419   (5)      1420   (5)      1421   (5)      1423   (5)      1424   (5)
                1425   (5)      1429   (5)      1430   (5)      1431   (5)      1443   (6)      1444   (6)      1445   (6)      1446   (6)
                1453   (6)      1460   (6)      1461   (6)      1462   (6)      1463   (6)      1466   (6)      1504   (6)      1519   (6)
                1553   (6)      1565   (6)      1572   (6)      1581   (6)      1587   (6)      1593   (6)      1599   (6)      1600   (6)
                1609   (6)      1616   (6)      1617   (6)      1618   (6)      1619   (6)      1620   (6)      1621   (6)      1622   (6)
                1623   (6)      1624   (6)      1625   (6)      1638   (6)      1639   (6)      1640   (6)      1641   (6)      1642   (6)
                1643   (6)      1644   (6)      1658   (6)      1661   (6)      1663   (6)      1664   (6)      1665   (6)      1666   (6)
                1673   (6)      1675   (6)      1693   (6)      1694   (6)      1723   (6)      1761   (6)      1773   (6)      1774   (6)
                1775   (6)      1789   (6)      1842   (6)      1871   (6)      1873   (6)      1880   (6)      1887   (6)      1891   (6)
                1905   (6)      1915   (6)      1916   (6)      1928   (6)      1930   (6)      1939   (6)      2006   (6)      2025   (6)
                2036   (6)      2055   (6)      2066   (6)      2073   (6)      2084   (6)      2095   (6)      2114   (6)      2121   (6)
                2126   (6)      2131   (6)      2142   (6)      2149   (6)      2160   (6)      2182   (6)      2183   (6)      2184   (6)
                2189   (6)      2197   (6)      2198   (6)      2199   (6)      2200   (6)      2201   (6)      2202   (6)      2203   (6)
                2204   (6)      2205   (6)      2206   (6)      2212   (6)      2264   (7)      2266   (7)      2267   (7)      2268   (7)
                2269   (7)      2270   (7)      2271   (7)      2272   (7)      2276   (7)      2277   (7)      2278   (7)      2279   (7)
                2280   (7)      2281   (7)      2282   (7)      2283   (7)      2284   (7)      2288   (7)      2289   (7)      2290   (7)
                2294   (7)      2295   (7)      2296   (7)      2297   (7)      2301   (7)      2302   (7)      2303   (7)      2304   (7)
                2305   (7)      2309   (7)      2310   (7)      2314   (7)      2315   (7)      2316   (7)      2317   (7)      2318   (7)
                2319   (7)      2320   (7)      2321   (7)      2325   (7)      2329   (7)      2330   (7)      2331   (7)      2332   (7)
```

```
                 2333     (7)      2338     (7)      2339     (7)      2340     (7)      2341     (7)      2342     (7)      2343     (7)      2344     (7)
                 2345     (7)      2349     (7)      2351     (7)      2355     (7)      2357     (7)      2361     (7)      2362     (7)      2363     (7)
                 2368     (7)      2369     (7)      2370     (7)      2374     (7)      2378     (7)      2379     (7)      2380     (7)      2381     (7)
                 2382     (7)      2383     (7)      2384     (7)      2385     (7)      2386     (7)      2387     (7)      2388     (7)      2389     (7)
                 2390     (7)      2391     (7)      2392     (7)      2397     (7)      2398     (7)      2399     (7)      2463     (8)      2464     (8)
                 2465     (8)      2466     (8)      2467     (8)      2468     (8)      2469     (8)      2470     (8)      2471     (8)      2472     (8)
                 2473     (8)      2474     (8)      2475     (8)      2476     (8)      2477     (8)      2478     (8)      2518     (9)      2519     (9)
                 2520     (9)      2521     (9)      2522     (9)      2523     (9)      2524     (9)      2525     (9)      2526     (9)      2527     (9)
                 2568     (10)     2569     (10)     2570     (10)     2571     (10)     2572     (10)     2573     (10)     2627     (11)     2628     (11)
                 2629     (11)     2630     (11)     2631     (11)     2634     (11)     2635     (11)     2636     (11)     2637     (11)     2695     (12)
                 2696     (12)     2697     (12)     2698     (12)     2699     (12)     2702     (12)     2703     (12)     2704     (12)     2705     (12)
                 2706     (12)     2707     (12)     2708     (12)     2709     (12)     2710     (12)     2711     (12)     2745     (13)     2746     (13)
                 2747     (13)     2748     (13)     2749     (13)     2750     (13)     2751     (13)     2753     (13)     2754     (13)     2799     (14)
                 2800     (14)     2801     (14)     2802     (14)     2803     (14)     2804     (14)     2805     (14)     2806     (14)     2807     (14)
                 2817     (14)     2818     (14)     2819     (14)     2820     (14)     2821     (14)     2822     (14)     2823     (14)     2824     (14)
                 2825     (14)     2826     (14)     2827     (14)     2828     (14)     2833     (14)     2834     (14)     2835     (14)     2836     (14)
                 2837     (14)     2841     (14)     2842     (14)     2843     (14)     2847     (14)     2848     (14)     2849     (14)     2850     (14)
                 2887     (15)     2888     (15)     2889     (15)     2891     (15)     2892     (15)     2893     (15)     2895     (15)     2896     (15)
                 2897     (15)     2898     (15)     2899     (15)     2900     (15)     2901     (15)     2902     (15)     2903     (15)     2905     (15)
                 2906     (15)     2907     (15)     2937     (16)     2938     (16)     2939     (16)     2940     (16)     2941     (16)     2942     (16)
                 2944     (16)     2945     (16)     2946     (16)     2947     (16)     2948     (16)     2949     (16)     2950     (16)     2951     (16)
                 2952     (16)     2953     (16)     2954     (16)     2956     (16)     2957     (16)     2958     (16)     2997     (17)     2998     (17)
                 2999     (17)     3000     (17)     3001     (17)     3002     (17)     3003     (17)     3004     (17)     3005     (17)     3006     (17)
                 3007     (17)     3008     (17)     3041     (18)     3042     (18)     3043     (18)     3044     (18)     3045     (18)     3046     (18)
                 3047     (18)     3048     (18)     3049     (18)     3050     (18)     3051     (18)     3052     (18)     3053     (18)     3054     (18)
                 3055     (18)     3056     (18)     3057     (18)     3058     (18)     3061     (18)     3062     (18)     3063     (18)     3064     (18)
                 3065     (18)     3067     (18)     3068     (18)     3069     (18)     3101     (19)     3102     (19)     3103     (19)     3104     (19)
                 3105     (19)     3106     (19)     3107     (19)     3108     (19)     3109     (19)     3110     (19)     3111     (19)     3112     (19)
                 3113     (19)     3114     (19)     3115     (19)     3116     (19)     3117     (19)     3120     (19)     3121     (19)     3122     (19)
                 3123     (19)     3124     (19)     3126     (19)     3127     (19)     3128     (19)     3207     (20)     3212     (20)     3213     (20)
                 3214     (20)     3215     (20)     3216     (20)     3217     (20)     3218     (20)     3219     (20)     3223     (20)     3224     (20)
                 3225     (20)     3226     (20)     3227     (20)     3228     (20)     3229     (20)     3230     (20)     3231     (20)     3236     (20)
                 3237     (20)     3238     (20)     3242     (20)     3243     (20)     3244     (20)     3245     (20)     3247     (20)     3248     (20)
                 3249     (20)     3250     (20)     3251     (20)     3252     (20)     3253     (20)     3254     (20)     3255     (20)     3256     (20)
                 3257     (20)     3258     (20)     3260     (20)     3262     (20)     3263     (20)     3267     (20)     3268     (20)     3269     (20)
                 3270     (20)     3271     (20)     3272     (20)     3273     (20)     3274     (20)     3278     (20)     3279     (20)     3280     (20)
                 3281     (20)     3282     (20)     3283     (20)     3284     (20)     3285     (20)     3286     (20)     3287     (20)     3291     (20)
                 3292     (20)     3293     (20)     3294     (20)     3296     (20)     3297     (20)     3298     (20)     3299     (20)     3305     (20)
                 3307     (20)     3308     (20)     3313     (20)     3314     (20)     3315     (20)     3316     (20)     3317     (20)     3318     (20)
                 3322     (20)     3323     (20)     3324     (20)     3325     (20)     3326     (20)     3327     (20)     3328     (20)     3329     (20)
                 3333     (20)     3337     (20)     3338     (20)     3339     (20)     3340     (20)     3341     (20)     3342     (20)     3348     (20)
                 3349     (20)     3350     (20)     3351     (20)     3352     (20)     3353     (20)     3354     (20)     3358     (20)     3359     (20)
                 3360     (20)     3361     (20)     3362     (20)     3363     (20)     3364     (20)     3365     (20)     3366     (20)     3367     (20)
                 3368     (20)     3369     (20)     3370     (20)     3371     (20)     3372     (20)     3373     (20)     3374     (20)     3375     (20)
                 3376     (20)     3377     (20)     3378     (20)     3379     (20)     3380     (20)     3381     (20)     3382     (20)     3383     (20)
                 3384     (20)     3385     (20)     3386     (20)     3387     (20)     3391     (20)     3392     (20)     3393     (20)     3394     (20)
                 3395     (20)     3396     (20)     3400     (20)     3401     (20)     3402     (20)     3403     (20)     3404     (20)     3405     (20)
                 3406     (20)     3407     (20)     3412     (20)     3413     (20)     3414     (20)     3415     (20)     3416     (20)     3417     (20)
                 3418     (20)     3419     (20)     3420     (20)     3421     (20)     3422     (20)     3423     (20)     3424     (20)     3425     (20)
                 3426     (20)     3462     (21)     3463     (21)     3464     (21)     3465     (21)     3466     (21)     3467     (21)     3468     (21)
                 3469     (21)     3473     (21)     3474     (21)     3475     (21)     3476     (21)     3477     (21)     3478     (21)     3479     (21)
                 3480     (21)     3481     (21)     3482     (21)     3483     (21)     3484     (21)     3485     (21)     3486     (21)     3487     (21)
                 3488     (21)     3520     (22)     3521     (22)     3522     (22)     3523     (22)     3524     (22)     3525     (22)     3526     (22)
                 3527     (22)     3528     (22)     3529     (22)     3530     (22)     3531     (22)     3535     (22)     3536     (22)     3537     (22)
                 3604     (23)     3605     (23)     3606     (23)     3607     (23)     3608     (23)     3609     (23)     3610     (23)     3611     (23)
```

```
                3612   (23)    3613   (23)    3618   (23)    3619   (23)    3620   (23)    3621   (23)    3626   (23)    3627   (23)
                3628   (23)    3629   (23)    3630   (23)    3631   (23)    3632   (23)    3633   (23)    3634   (23)    3635   (23)
                3636   (23)    3637   (23)    3641   (23)    3642   (23)    3643   (23)    3644   (23)    3645   (23)    3646   (23)
                3647   (23)    3648   (23)    3649   (23)    3650   (23)    3654   (23)    3655   (23)    3656   (23)    3657   (23)
                3661   (23)    3665   (23)    3666   (23)    3667   (23)    3671   (23)    3672   (23)    3673   (23)    3677   (23)
                3678   (23)    3679   (23)    3680   (23)    3681   (23)    3682   (23)    3683   (23)    3684   (23)    3685   (23)
                3686   (23)    3687   (23)    3688   (23)    3689   (23)    3690   (23)    3691   (23)    3692   (23)    3693   (23)
                3694   (23)    3695   (23)    3699   (23)    3700   (23)    3701   (23)    3702   (23)    3706   (23)    3707   (23)
                3709   (23)    3710   (23)    3711   (23)    3712   (23)    3713   (23)    3717   (23)    3718   (23)    3719   (23)
                3720   (23)    3721   (23)    3722   (23)    3723   (23)    3724   (23)    3725   (23)    3726   (23)    3727   (23)
                3779   (24)    3780   (24)    3781   (24)    3785   (24)    3786   (24)    3787   (24)    3791   (24)    3792   (24)
                3796   (24)    3797   (24)    3798   (24)    3799   (24)    3800   (24)    3801   (24)    3802   (24)    3803   (24)
                3804   (24)    3805   (24)    3806   (24)    3807   (24)    3808   (24)    3809   (24)    3810   (24)    3811   (24)
                3812   (24)    3813   (24)    3814   (24)    3815   (24)    3816   (24)    3817   (24)    3821   (24)    3822   (24)
                3823   (24)    3824   (24)    3825   (24)    3826   (24)    3827   (24)    3828   (24)    3829   (24)    3830   (24)
                3831   (24)    3832   (24)    3833   (24)    3834   (24)    3835   (24)    3836   (24)    3837   (24)    3841   (24)
                3842   (24)    3843   (24)    3844   (24)    3845   (24)    3846   (24)    3850   (24)    3851   (24)    3852   (24)
                3856   (24)    3857   (24)    3858   (24)    3862   (24)    3863   (24)    3864   (24)    3890   (25)    3892   (25)
                3893   (25)    3894   (25)    3895   (25)    3896   (25)    3915   (26)    3916   (26)    3917   (26)    3918   (26)
                3919   (26)    3953   (27)    3955   (27)    3958   (27)    3959   (27)    3960   (27)    3961   (27)    3962   (27)
                3967   (27)    3968   (27)    3969   (27)    3970   (27)    3971   (27)    3975   (27)    3977   (27)    3978   (27)
                3979   (27)    3983   (27)    3984   (27)    3985   (27)    3986   (27)    4004   (28)    4005   (28)    4006   (28)
                4007   (28)    4009   (28)    4011   (28)    4012   (28)    4013   (28)    4014   (28)    4015   (28)    4016   (28)
                4017   (28)    4035   (29)    4036   (29)    4037   (29)    4038   (29)    4040   (29)    4042   (29)    4043   (29)
                4044   (29)    4045   (29)    4046   (29)    4047   (29)    4048   (29)    4067   (30)    4068   (30)    4069   (30)
                4070   (30)    4072   (30)    4073   (30)    4074   (30)    4075   (30)    4076   (30)    4077   (30)    4078   (30)
                4079   (30)    4080   (30)    4081   (30)    4094   (31)    4095   (31)    4097   (31)    4098   (31)    4099   (31)
                4100   (31)    4102   (31)    4103   (31)    4104   (31)    4105   (31)    4119   (32)    4120   (32)    4121   (32)
                4122   (32)    4123   (32)    4124   (32)    4125   (32)    4126   (32)    4127   (32)    4128   (32)    4129   (32)
                4130   (32)    4131   (32)    4132   (32)    4133   (32)    4134   (32)    4135   (32)    4136   (32)    4137   (32)
                4138   (32)    4153   (33)    4154   (33)    4155   (33)    4156   (33)    4157   (33)    4158   (33)    4159   (33)
                4160   (33)    4161   (33)    4162   (33)    4201   (34)    4203   (34)    4204   (34)    4205   (34)    4206   (34)
                4207   (34)    4208   (34)    4209   (34)    4210   (34)    4212   (34)    4213   (34)    4214   (34)    4215   (34)
                4216   (34)    4217   (34)    4218   (34)    4219   (34)    4220   (34)    4226   (34)    4227   (34)    4228   (34)
                4229   (34)    4230   (34)    4231   (34)    4232   (34)    4237   (34)    4238   (34)    4239   (34)    4240   (34)
                4241   (34)    4243   (34)    4245   (34)    4246   (34)    4247   (34)    4248   (34)    4249   (34)    4250   (34)
                4251   (34)    4253   (34)    4254   (34)    4255   (34)    4259   (34)    4265   (34)    4266   (34)    4267   (34)
                4268   (34)    4269   (34)    4270   (34)    4271   (34)    4272   (34)    4273   (34)    4275   (34)    4276   (34)
                4280   (34)    4281   (34)    4282   (34)    4309   (35)    4310   (35)    4311   (35)    4312   (35)    4313   (35)
                4317   (35)    4318   (35)    4319   (35)    4320   (35)    4321   (35)    4322   (35)    4326   (35)    4327   (35)
                4328   (35)    4329   (35)    4330   (35)    4334   (35)    4335   (35)    4336   (35)    4337   (35)    4338   (35)
                4339   (35)    4340   (35)    4344   (35)    4345   (35)    4346   (35)    4347   (35)    4348   (35)    4349   (35)
                4350   (35)    4351   (35)    4355   (35)    4356   (35)    4357   (35)    4358   (35)    4359   (35)    4360   (35)
                4361   (35)    4362   (35)    4363   (35)    4364   (35)    4368   (35)    4369   (35)    4375   (35)    4376   (35)
                4377   (35)    4378   (35)    4379   (35)    4380   (35)    4385   (35)    4386   (35)    4391   (35)    4392   (35)
                4393   (35)    4430   (36)    4431   (36)    4432   (36)    4433   (36)    4434   (36)    4435   (36)    4436   (36)
                4443   (36)    4444   (36)    4445   (36)    4446   (36)    4447   (36)    4448   (36)    4449   (36)    4450   (36)
                4451   (36)    4452   (36)    4459   (36)    4460   (36)    4461   (36)    4462   (36)    4463   (36)    4469   (36)
                4470   (36)    4471   (36)    4472   (36)    4477   (36)    4478   (36)    4514   (37)    4515   (37)    4516   (37)
                4517   (37)    4518   (37)    4519   (37)    4523   (37)    4524   (37)    4525   (37)    4526   (37)    4527   (37)
                4531   (37)    4532   (37)    4533   (37)    4534   (37)    4535   (37)    4536   (37)    4537   (37)    4568   (38)
                4569   (38)    4570   (38)    4571   (38)    4572   (38)    4573   (38)    4574   (38)    4616   (39)    4617   (39)
                4618   (39)    4619   (39)    4620   (39)    4621   (39)    4622   (39)    4623   (39)    4624   (39)    4628   (39)
                4629   (39)    4630   (39)    4631   (39)    4632   (39)    4633   (39)    4634   (39)    4635   (39)    4636   (39)
                4637   (39)    4641   (39)    4642   (39)    4643   (39)    4644   (39)    4645   (39)    4649   (39)    4650   (39)
```

```
                    4651     (39)     4652     (39)     4653     (39)     4655     (39)     4656     (39)     4657     (39)     4658     (39)     4659     (39)
                    4660     (39)     4661     (39)     4662     (39)     4666     (39)     4667     (39)     4668     (39)     4672     (39)     4673     (39)
                    4674     (39)     4675     (39)     4676     (39)     4677     (39)     4678     (39)     4679     (39)     4685     (39)     4686     (39)
                    4687     (39)     4688     (39)     4689     (39)     4690     (39)     4691     (39)     4692     (39)     4693     (39)     4694     (39)
                    4695     (39)     4696     (39)     4697     (39)     4698     (39)     4699     (39)     4704     (39)     4705     (39)     4753     (40)
                    4755     (40)     4756     (40)     4757     (40)     4758     (40)     4761     (40)     4762     (40)     4763     (40)     4764     (40)
                    4767     (40)     4768     (40)     4769     (40)     4770     (40)     4771     (40)     4772     (40)     4773     (40)     4774     (40)
                    4775     (40)     4776     (40)     4777     (40)     4782     (40)     4783     (40)     4784     (40)     4785     (40)     4786     (40)
                    4790     (40)     4791     (40)     4792     (40)     4793     (40)     4794     (40)     4795     (40)     4796     (40)     4797     (40)
                    4798     (40)     4799     (40)     4800     (40)     4801     (40)     4805     (40)     4806     (40)     4807     (40)     4808     (40)
                    4813     (40)     4814     (40)     4815     (40)     4816     (40)     4821     (40)     4822     (40)     4823     (40)     4824     (40)
                    4829     (40)     4830     (40)     4831     (40)     4832     (40)     4838     (40)     4839     (40)     4840     (40)     4841     (40)
                    4842     (40)     4843     (40)     4844     (40)     4845     (40)     4846     (40)     4847     (40)     4848     (40)     4849     (40)
                    4850     (40)     4851     (40)     4852     (40)     4853     (40)     4854     (40)     4858     (40)     4859     (40)     4860     (40)
                    4861     (40)     4862     (40)     4868     (40)     4869     (40)     4870     (40)     4871     (40)     4872     (40)     4873     (40)
                    4879     (40)     4880     (40)     4881     (40)     4882     (40)     4883     (40)     4888     (40)     4889     (40)     4894     (40)
                    4895     (40)     4896     (40)     4897     (40)     4898     (40)     4899     (40)     4948     (41)     4949     (41)     4950     (41)
                    4954     (41)     4955     (41)     4956     (41)     4957     (41)     4958     (41)     4959     (41)     4960     (41)     4961     (41)
                    4965     (41)     4966     (41)     4970     (41)     4971     (41)     4972     (41)     4973     (41)     4974     (41)     4975     (41)
                    4976     (41)     4978     (41)     4979     (41)     4980     (41)     4981     (41)     4982     (41)     4983     (41)     4987     (41)
                    4988     (41)     4989     (41)     4990     (41)     4991     (41)     4992     (41)     4993     (41)     4994     (41)     4995     (41)
                    4996     (41)     4997     (41)     4998     (41)     4999     (41)     5000     (41)     5001     (41)     5002     (41)     5003     (41)
                    5004     (41)     5005     (41)     5006     (41)     5007     (41)     5011     (41)     5012     (41)     5013     (41)     5017     (41)
                    5018     (41)     5056     (42)     5057     (42)     5058     (42)     5059     (42)     5060     (42)     5061     (42)     5079     (43)
                    5098     (44)     5099     (44)     5100     (44)     5101     (44)     5102     (44)     5103     (44)     5104     (44)     5105     (44)
                    5106     (44)     5107     (44)     5108     (44)     5109     (44)     5149     (45)     5150     (45)     5151     (45)     5152     (45)
                    5153     (45)     5154     (45)     5155     (45)     5156     (45)     5157     (45)     5158     (45)     5159     (45)     5160     (45)
                    5161     (45)     5162     (45)     5163     (45)     5164     (45)     5169     (45)     5170     (45)     5171     (45)     5172     (45)
                    5173     (45)     5174     (45)     5175     (45)     5176     (45)     5177     (45)     5178     (45)     5183     (45)     5184     (45)
                    5185     (45)     5186     (45)     5187     (45)     5188     (45)     5189     (45)     5190     (45)     5191     (45)     5192     (45)
                    5196     (45)     5197     (45)     5198     (45)     5202     (45)     5203     (45)     5204     (45)     5205     (45)     5206     (45)
                    5207     (45)     5208     (45)     5209     (45)     5210     (45)     5211     (45)     5212     (45)     5213     (45)     5214     (45)
                    5215     (45)     5216     (45)     5217     (45)     5218     (45)     5224     (45)     5225     (45)     5226     (45)     5227     (45)
                    5228     (45)     5229     (45)     5230     (45)     5231     (45)     5232     (45)     5233     (45)     5234     (45)     5235     (45)
                    5236     (45)     5237     (45)     5241     (45)     5242     (45)     5243     (45)     5244     (45)     5245     (45)     5246     (45)
                    5247     (45)     5248     (45)     5249     (45)     5250     (45)     5251     (45)     5252     (45)     5253     (45)     5255     (45)
                    5256     (45)     5257     (45)     5258     (45)     5264     (45)     5265     (45)     5266     (45)     5270     (45)     5271     (45)
                    5272     (45)     5273     (45)     5277     (45)     5278     (45)     5279     (45)     5283     (45)     5284     (45)     5285     (45)
                    5286     (45)     5287     (45)     5288     (45)     5289     (45)     5290     (45)     5291     (45)     5296     (45)     5297     (45)
                    5298     (45)     5303     (45)     5304     (45)     5347     (46)     5348     (46)     5349     (46)     5350     (46)     5351     (46)
                    5352     (46)     5353     (46)     5354     (46)     5355     (46)     5356     (46)     5357     (46)     5358     (46)     5359     (46)
                    5360     (46)     5365     (46)     5366     (46)     5367     (46)     5368     (46)     5369     (46)     5370     (46)     5371     (46)
                    5375     (46)     5376     (46)     5377     (46)     5378     (46)     5379     (46)     5380     (46)     5381     (46)     5382     (46)
                    5383     (46)     5384     (46)     5385     (46)     5386     (46)     5387     (46)     5393     (46)     5394     (46)     5395     (46)
                    5396     (46)     5397     (46)     5398     (46)     5399     (46)     5400     (46)     5401     (46)     5402     (46)     5403     (46)
                    5404     (46)     5405     (46)     5406     (46)     5412     (46)     5413     (46)     5414     (46)     5415     (46)     5416     (46)
                    5417     (46)     5418     (46)     5419     (46)     5420     (46)     5425     (46)     5426     (46)     5427     (46)     5432     (46)
                    5433     (46)     5478     (47)     5480     (47)     5481     (47)     5482     (47)     5483     (47)     5484     (47)     5485     (47)
                    5486     (47)     5487     (47)     5488     (47)     5489     (47)     5490     (47)     5491     (47)     5492     (47)     5493     (47)
                    5494     (47)     5495     (47)     5496     (47)     5497     (47)     5498     (47)     5499     (47)     5500     (47)     5501     (47)
                    5502     (47)     5503     (47)     5504     (47)     5505     (47)     5506     (47)     5507     (47)     5508     (47)     5509     (47)
                    5510     (47)     5511     (47)     5512     (47)     5513     (47)     5514     (47)     5515     (47)     5516     (47)     5521     (47)
                    5522     (47)     5523     (47)     5524     (47)     5525     (47)     5526     (47)     5527     (47)     5532     (47)     5533     (47)
                    5534     (47)     5535     (47)     5536     (47)     5538     (47)     5539     (47)     5544     (47)     5545     (47)     5550     (47)
                    5551     (47)     5556     (47)     5557     (47)     5562     (47)     5563     (47)     5568     (47)     5569     (47)     5575     (47)
```

```
        5576   (47)   5577   (47)   5582   (47)   5583   (47)   5584   (47)   5585   (47)   5586   (47)   5587   (47)
        5589   (47)   5591   (47)   5592   (47)   5593   (47)   5594   (47)   5595   (47)   5601   (47)   5602   (47)
        5603   (47)   5604   (47)   5605   (47)   5606   (47)   5607   (47)   5608   (47)   5609   (47)   5610   (47)
        5611   (47)   5617   (47)   5618   (47)   5619   (47)   5620   (47)   5621   (47)   5627   (47)   5628   (47)
        5629   (47)   5630   (47)   5637   (47)   5638   (47)   5639   (47)   5640   (47)   5645   (47)   5650   (47)
        5651   (47)   5653   (47)   5654   (47)   5656   (47)   5657   (47)   5658   (47)   5661   (47)   5704   (48)
        5705   (48)   5706   (48)   5707   (48)   5711   (48)   5712   (48)   5716   (48)   5717   (48)   5718   (48)
        5719   (48)   5724   (48)   5725   (48)   5784   (49)   5785   (49)   5786   (49)   5790   (49)   5791   (49)
        5792   (49)   5797   (49)   5798   (49)   5799   (49)   5800   (49)   5801   (49)   5802   (49)   5803   (49)
        5804   (49)   5805   (49)   5806   (49)   5807   (49)   5808   (49)   5809   (49)   5810   (49)   5811   (49)
        5812   (49)   5813   (49)   5814   (49)   5815   (49)   5816   (49)   5817   (49)   5818   (49)   5819   (49)
        5823   (49)   5824   (49)   5825   (49)   5826   (49)   5827   (49)   5828   (49)   5829   (49)   5830   (49)
        5831   (49)   5832   (49)   5833   (49)   5834   (49)   5835   (49)   5836   (49)   5837   (49)   5838   (49)
        5839   (49)   5840   (49)   5844   (49)   5849   (49)   5850   (49)   5851   (49)   5852   (49)   5853   (49)
        5854   (49)   5855   (49)   5856   (49)   5857   (49)   5858   (49)   5859   (49)   5860   (49)   5905   (50)
        5906   (50)   5910   (50)   5911   (50)   5912   (50)   5913   (50)   5917   (50)   5918   (50)   5919   (50)
        5920   (50)   5921   (50)   5922   (50)   5923   (50)   5927   (50)   5928   (50)   5929   (50)   5930   (50)
        5931   (50)   5996   (51)   5997   (51)   5998   (51)   6002   (51)   6003   (51)   6007   (51)   6011   (51)
        6012   (51)   6013   (51)   6017   (51)   6018   (51)   6019   (51)   6020   (51)   6021   (51)   6022   (51)
        6023   (51)   6024   (51)   6025   (51)   6026   (51)   6028   (51)   6029   (51)   6030   (51)   6034   (51)
        6035   (51)   6036   (51)   6040   (51)   6041   (51)   6042   (51)   6043   (51)   6044   (51)   6048   (51)
        6049   (51)   6050   (51)   6051   (51)   6052   (51)   6053   (51)   6055   (51)   6056   (51)   6057   (51)
        6061   (51)   6063   (51)   6064   (51)   6068   (51)   6069   (51)   6070   (51)   6071   (51)   6073   (51)
        6074   (51)   6075   (51)   6076   (51)   6077   (51)   6111   (52)   6116   (52)   6117   (52)   6118   (52)
        6119   (52)   6120   (52)   6122   (52)   6123   (52)   6124   (52)   6125   (52)   6127   (52)   6128   (52)
        6129   (52)   6169   (53)   6170   (53)   6171   (53)   6172   (53)   6173   (53)   6174   (53)   6175   (53)
        6176   (53)   6177   (53)   6178   (53)   6182   (53)   6183   (53)   6184   (53)   6185   (53)   6186   (53)
        6190   (53)   6191   (53)   6192   (53)   6193   (53)   6194   (53)   6195   (53)   6199   (53)   6200   (53)
        6201   (53)   6204   (53)   6205   (53)   6206   (53)   6210   (53)   6211   (53)   6215   (53)   6216   (53)
        6217   (53)   6218   (53)   6219   (53)   6220   (53)   6221   (53)   6263   (54)   6264   (54)   6265   (54)
        6266   (54)   6267   (54)   6268   (54)   6269   (54)   6270   (54)   6271   (54)   6272   (54)   6273   (54)
        6274   (54)   6275   (54)   6276   (54)   6277   (54)   6278   (54)   6279   (54)   6280   (54)   6281   (54)
        6282   (54)   6283   (54)   6284   (54)   6285   (54)   6286   (54)   6287   (54)   6291   (54)   6292   (54)
        6293   (54)   6294   (54)   6295   (54)   6296   (54)   6297   (54)   6298   (54)   6299   (54)   6303   (54)
        6304   (54)   6305   (54)   6306   (54)   6307   (54)   6308   (54)   6309   (54)   6313   (54)   6314   (54)
        6315   (54)   6319   (54)   6320   (54)   6321   (54)   6325   (54)   6326   (54)   6330   (54)   6331   (54)
        6332   (54)   6333   (54)   6334   (54)   6335   (54)   6336   (54)   6340   (54)   6341   (54)   6342   (54)
        6343   (54)   6344   (54)   6345   (54)   6346   (54)   6372   (55)   6373   (55)   6374   (55)   6421   (56)
        6422   (56)   6426   (56)   6427   (56)   6428   (56)   6429   (56)   6433   (56)   6434   (56)   6435   (56)
        6436   (56)   6437   (56)   6438   (56)   6439   (56)   6440   (56)   6441   (56)   6442   (56)   6446   (56)
        6447   (56)   6448   (56)   6449   (56)   6450   (56)   6451   (56)   6452   (56)   6453   (56)   6454   (56)
        6455   (56)   6456   (56)   6460   (56)   6461   (56)   6462   (56)   6463   (56)   6464   (56)   6465   (56)
        6466   (56)   6470   (56)   6471   (56)   6472   (56)   6476   (56)   6477   (56)   6478   (56)   6479   (56)
        6480   (56)   6481   (56)   6482   (56)   6483   (56)   6492   (56)   6493   (56)   6494   (56)   6495   (56)
        6496   (56)   6497   (56)   6498   (56)   6499   (56)   6500   (56)   6501   (56)   6502   (56)   6503   (56)
        6504   (56)   6505   (56)   6506   (56)   6507   (56)   6508   (56)   6512   (56)   6544   (57)   6545   (57)
        6546   (57)   6547   (57)   6551   (57)   6552   (57)   6553   (57)   6554   (57)   6555   (57)   6561   (57)
        6562   (57)   6563   (57)   6564   (57)   6565   (57)   6569   (57)   6570   (57)   6571   (57)   6573   (57)
        6574   (57)   6575   (57)   6576   (57)   6577   (57)   6578   (57)   6579   (57)   6580   (57)   6581   (57)
        6582   (57)   6583   (57)   6584   (57)   6585   (57)   6586   (57)   6624   (58)   6625   (58)   6626   (58)
        6627   (58)   6628   (58)   6629   (58)   6630   (58)   6631   (58)   6632   (58)   6633   (58)   6634   (58)
        6635   (58)   6639   (58)   6640   (58)   6641   (58)   6642   (58)   6646   (58)   6647   (58)   6648   (58)
        6649   (58)   6653   (58)   6654   (58)   6655   (58)   6656   (58)   6657   (58)   6661   (58)   6662   (58)
        6663   (58)   6664   (58)   6665   (58)   6666   (58)   6667   (58)   6671   (58)   6672   (58)   6673   (58)
        6674   (58)   6678   (58)   6679   (58)   6680   (58)   6684   (58)   6685   (58)   6686   (58)   6687   (58)
```

```
                 6688    (58)    6689    (58)    6720    (59)    6724    (59)    6725    (59)    6726    (59)    6730    (59)    6731    (59)
                 6732    (59)    6771    (60)    6772    (60)    6773    (60)    6774    (60)    6775    (60)    6779    (60)    6780    (60)
                 6781    (60)    6782    (60)    6783    (60)    6784    (60)    6785    (60)    6789    (60)    6790    (60)    6791    (60)
                 6824    (61)    6825    (61)    6829    (61)    6830    (61)    6834    (61)    6835    (61)    6836    (61)    6837    (61)
                 6838    (61)    6839    (61)    6843    (61)    6844    (61)    6845    (61)    6846    (61)    6847    (61)    6848    (61)
                 6849    (61)    6850    (61)    6851    (61)    6855    (61)    6856    (61)    6857    (61)    6858    (61)    6859    (61)
                 6860    (61)    6861    (61)    6862    (61)    6863    (61)    6864    (61)    6868    (61)    6909    (62)    6910    (62)
                 6911    (62)    6915    (62)    6916    (62)    6917    (62)    6918    (62)    6919    (62)    6920    (62)    6921    (62)
                 6922    (62)    6923    (62)    6924    (62)    6925    (62)    6926    (62)    6927    (62)    6928    (62)    6929    (62)
                 6933    (62)    6934    (62)    6935    (62)    6936    (62)    6937    (62)    6938    (62)    6939    (62)    6941    (62)
                 6942    (62)    6943    (62)    6944    (62)    6948    (62)    6950    (62)    6951    (62)    6952    (62)    6953    (62)
                 6999    (63)    7000    (63)    7001    (63)    7005    (63)    7006    (63)    7007    (63)    7008    (63)    7009    (63)
                 7010    (63)    7011    (63)    7012    (63)    7013    (63)    7014    (63)    7015    (63)    7016    (63)    7017    (63)
                 7018    (63)    7019    (63)    7023    (63)    7024    (63)    7025    (63)    7026    (63)    7027    (63)    7028    (63)
                 7029    (63)    7030    (63)    7031    (63)    7035    (63)    7036    (63)    7037    (63)    7038    (63)    7039    (63)
                 7040    (63)    7041    (63)    7042    (63)    7043    (63)    7044    (63)    7045    (63)    7049    (63)    7050    (63)
                 7051    (63)    7052    (63)    7053    (63)    7054    (63)    7055    (63)    7056    (63)    7057    (63)    7058    (63)
                 7059    (63)    7063    (63)    7064    (63)    7065    (63)    7066    (63)    7067    (63)    7068    (63)    7069    (63)
                 7071    (63)    7072    (63)    7073    (63)    7074    (63)    7078    (63)    7080    (63)    7081    (63)    7082    (63)
                 7083    (63)    7124    (64)    7125    (64)    7126    (64)    7127    (64)    7131    (64)    7132    (64)    7133    (64)
                 7134    (64)    7135    (64)    7139    (64)    7140    (64)    7141    (64)    7142    (64)    7143    (64)    7147    (64)
                 7148    (64)    7149    (64)    7150    (64)    7151    (64)    7152    (64)    7153    (64)    7154    (64)    7155    (64)
                 7159    (64)    7160    (64)    7161    (64)    7162    (64)    7163    (64)    7164    (64)    7165    (64)    7166    (64)
                 7170    (64)    7171    (64)    7175    (64)    7176    (64)    7177    (64)    7178    (64)    7179    (64)    7183    (64)
                 7185    (64)    7186    (64)    7187    (64)    7188    (64)    7192    (64)    7195    (64)    7196    (64)    7197    (64)
                 7198    (64)    7199    (64)    7200    (64)    7201    (64)    7202    (64)    7273    (65)    7274    (65)    7275    (65)
                 7279    (65)    7280    (65)    7281    (65)    7285    (65)    7286    (65)    7287    (65)    7291    (65)    7292    (65)
                 7293    (65)    7297    (65)    7298    (65)    7299    (65)    7300    (65)    7304    (65)    7305    (65)    7306    (65)
                 7310    (65)    7311    (65)    7312    (65)    7313    (65)    7314    (65)    7315    (65)    7316    (65)    7320    (65)
                 7321    (65)    7322    (65)    7323    (65)    7324    (65)    7325    (65)    7326    (65)    7327    (65)    7331    (65)
                 7332    (65)    7333    (65)    7334    (65)    7335    (65)    7339    (65)    7424    (66)    7425    (66)    7426    (66)
                 7427    (66)    7428    (66)    7430    (66)    7431    (66)    7432    (66)    7433    (66)    7434    (66)    7438    (66)
                 7439    (66)    7440    (66)    7444    (66)    7445    (66)    7446    (66)    7447    (66)    7448    (66)    7449    (66)
                 7450    (66)    7451    (66)    7452    (66)    7453    (66)    7454    (66)    7455    (66)    7456    (66)    7461    (66)
                 7462    (66)    7463    (66)    7464    (66)    7465    (66)    7466    (66)    7467    (66)    7472    (66)    7473    (66)
                 7474    (66)    7475    (66)    7476    (66)    7477    (66)    7478    (66)    7479    (66)    7480    (66)    7484    (66)
                 7485    (66)    7486    (66)    7490    (66)    7491    (66)    7492    (66)    7494    (66)    7495    (66)    7496    (66)
                 7497    (66)    7498    (66)    7499    (66)    7500    (66)    7501    (66)    7502    (66)    7503    (66)    7504    (66)
                 7505    (66)    7506    (66)    7507    (66)    7511    (66)    7512    (66)    7513    (66)    7514    (66)    7515    (66)
                 7516    (66)    7517    (66)    7518    (66)    7519    (66)    7523    (66)    7524    (66)    7525    (66)    7526    (66)
                 7527    (66)    7528    (66)    7529    (66)    7530    (66)    7531    (66)    7535    (66)    7536    (66)    7537    (66)
                 7538    (66)    7539    (66)    7540    (66)    7541    (66)    7542    (66)    7543    (66)    7544    (66)    7550    (66)
                 7551    (66)    7552    (66)    7553    (66)    7554    (66)    7555    (66)    7556    (66)    7561    (66)    7562    (66)
                 7563    (66)    7564    (66)    7565    (66)    7569    (66)    7570    (66)    7571    (66)    7572    (66)    7573    (66)
                 7574    (66)    7575    (66)    7576    (66)    7577    (66)    7578    (66)    7582    (66)    7584    (66)    7585    (66)
                 7586    (66)    7587    (66)    7593    (67)    7594    (67)    7595    (67)    7596    (67)    7597    (67)    7598    (67)
                 7603    (67)    7604    (67)    7605    (67)    7606    (67)    7607    (67)    7608    (67)    7609    (67)    7610    (67)
                 7611    (67)    7613    (67)    7614    (67)    7615    (67)    7616    (67)    7617    (67)    7621    (67)    7622    (67)
                 7623    (67)    7625    (67)    7626    (67)    7627    (67)    7628    (67)    7629    (67)    7630    (67)    7634    (67)
                 7635    (67)    7639    (67)    7640    (67)    7641    (67)    7642    (67)    7643    (67)    7647    (67)    7648    (67)
                 7649    (67)    7650    (67)    7651    (67)    7652    (67)    7653    (67)    7654    (67)    7655    (67)    7656    (67)
                 7657    (67)    7661    (67)    7662    (67)    7663    (67)    7667    (67)    7668    (67)    7669    (67)    7673    (67)
                 7674    (67)    7675    (67)    7680    (67)    7681    (67)    7682    (67)    7683    (67)    7684    (67)    7685    (67)
                 7686    (67)    7687    (67)    7688    (67)    7690    (67)    7691    (67)    7692    (67)    7693    (67)    7694    (67)
                 7695    (67)    7696    (67)    7697    (67)    7699    (67)    7700    (67)    7701    (67)    7705    (67)    7706    (67)
```

```
                  7707     (67)     7708     (67)     7709     (67)     7710     (67)     7711     (67)     7713     (67)     7714     (67)     7715     (67)
                  7716     (67)     7717     (67)     7721     (67)     7726     (67)     7727     (67)     7728     (67)     7729     (67)     7732     (67)
                  7733     (67)     7734     (67)     7735     (67)     7736     (67)     7737     (67)     7738     (67)     7739     (67)     7740     (67)
                  7741     (67)     7742     (67)     7743     (67)     7744     (67)     7745     (67)     7746     (67)     7747     (67)     7748     (67)
                  7749     (67)     7750     (67)     7751     (67)     7752     (67)     7753     (67)     7754     (67)     7755     (67)     7756     (67)
                  7760     (67)     7761     (67)     7762     (67)     7763     (67)     7764     (67)     7765     (67)     7766     (67)     7767     (67)
                  7768     (67)     7769     (67)     7770     (67)     7771     (67)     7773     (67)     7774     (67)     7775     (67)     7776     (67)
                  7777     (67)     7778     (67)     7779     (67)     7780     (67)     7781     (67)     7783     (67)     7784     (67)     7785     (67)
                  7786     (67)     7787     (67)     7788     (67)     7789     (67)     7790     (67)     7791     (67)     7792     (67)     7793     (67)
                  7823     (68)     7824     (68)     7825     (68)     7826     (68)     7827     (68)     7828     (68)     7829     (68)     7830     (68)
                  7831     (68)     7832     (68)     7858     (69)     7859     (69)     7860     (69)     7861     (69)     7862     (69)     7906     (70)
                  7907     (70)     7908     (70)     7909     (70)     7913     (70)     7914     (70)     7915     (70)     7916     (70)     7917     (70)
                  7921     (70)     7922     (70)     7923     (70)     7924     (70)     7925     (70)     7926     (70)     7927     (70)     7928     (70)
                  7929     (70)     7930     (70)     7931     (70)     7932     (70)     7933     (70)     7937     (70)     7938     (70)     7939     (70)
                  7940     (70)     7941     (70)     7942     (70)     7949     (70)     7950     (70)     7951     (70)     7952     (70)     7953     (70)
                  7954     (70)     7955     (70)     7956     (70)     7957     (70)     7958     (70)     7959     (70)     7960     (70)     7961     (70)
                  7962     (70)     7963     (70)     7964     (70)     7965     (70)     7966     (70)     7970     (70)     7971     (70)     7972     (70)
                  7973     (70)     7974     (70)     7975     (70)     7976     (70)     7977     (70)     7978     (70)     7979     (70)     7980     (70)
                  7981     (70)     7982     (70)     7983     (70)     7984     (70)     7985     (70)     7986     (70)     7987     (70)     7988     (70)
                  7989     (70)     7990     (70)     7991     (70)     7992     (70)     7993     (70)     7994     (70)     7995     (70)     8030     (71)
                  8031     (71)     8032     (71)     8033     (71)     8034     (71)     8035     (71)     8039     (71)     8040     (71)     8041     (71)
                  8042     (71)     8043     (71)     8044     (71)     8045     (71)     8046     (71)     8047     (71)     8048     (71)     8049     (71)
                  8050     (71)     8051     (71)     8057     (71)     8058     (71)     8059     (71)     8060     (71)     8061     (71)     8062     (71)
                  8063     (71)     8068     (71)     8069     (71)     8074     (71)     8075     (71)     8076     (71)     8077     (71)     8078     (71)
                  8079     (71)     8080     (71)     8106     (72)     8107     (72)     8108     (72)     8109     (72)     8110     (72)     8111     (72)
                  8112     (72)     8138     (73)     8139     (73)     8140     (73)     8141     (73)     8142     (73)     8143     (73)     8144     (73)
                  8145     (73)     8146     (73)     8147     (73)     8171     (74)     8172     (74)     8173     (74)     8174     (74)     8198     (75)
                  8199     (75)     8200     (75)     8201     (75)     8241     (76)     8242     (76)     8243     (76)     8244     (76)     8245     (76)
                  8246     (76)     8247     (76)     8248     (76)     8249     (76)     8250     (76)     8251     (76)     8252     (76)     8253     (76)
                  8254     (76)     8255     (76)     8256     (76)     8260     (76)     8261     (76)     8262     (76)     8263     (76)     8267     (76)
                  8268     (76)     8269     (76)     8270     (76)     8271     (76)     8272     (76)     8273     (76)     8274     (76)     8275     (76)
                  8276     (76)     8277     (76)     8281     (76)     8282     (76)     8283     (76)     8287     (76)     8288     (76)     8289     (76)
                  8290     (76)     8291     (76)     8292     (76)     8293     (76)     8294     (76)     8295     (76)     8296     (76)     8297     (76)
                  8298     (76)     8299     (76)     8303     (76)     8304     (76)     8305     (76)     8306     (76)     8307     (76)     8308     (76)
                  8309     (76)     8310     (76)     8311     (76)     8312     (76)     8313     (76)     8314     (76)     8315     (76)     8316     (76)
                  8317     (76)     8318     (76)     8319     (76)     8320     (76)     8321     (76)     8322     (76)     8323     (76)     8324     (76)
                  8325     (76)     8326     (76)     8327     (76)     8328     (76)     8329     (76)     8334     (76)     8335     (76)     8336     (76)
                  8337     (76)     8338     (76)     8339     (76)     8340     (76)     8341     (76)     8342     (76)     8343     (76)     8344     (76)
                  8345     (76)     8346     (76)     8347     (76)     8348     (76)     8349     (76)     8350     (76)     8352     (76)     8354     (76)
                  8355     (76)     8356     (76)     8357     (76)     8359     (76)     8361     (76)     8362     (76)     8363     (76)     8365     (76)
                  8387     (77)     8388     (77)     8389     (77)     8390     (77)     8391     (77)     8392     (77)     8416     (78)     8417     (78)
                  8418     (78)     8419     (78)     8420     (78)     8421     (78)     8422     (78)     8423     (78)     8424     (78)     8425     (78)
                  8426     (78)     8428     (78)     8439     (79)     8440     (79)     8441     (79)     8442     (79)     8443     (79)     8444     (79)
                  8445     (79)     8446     (79)     8447     (79)     8448     (79)     8449     (79)     8450     (79)     8451     (79)     8452     (79)
                  8453     (79)     8454     (79)     8455     (79)     8456     (79)     8457     (79)     8460     (79)     8462     (79)     8463     (79)
                  8464     (79)     8465     (79)     8497     (80)     8498     (80)     8499     (80)     8500     (80)     8501     (80)     8502     (80)
                  8503     (80)     8504     (80)     8505     (80)     8506     (80)     8507     (80)     8508     (80)     8550     (81)     8551     (81)
                  8552     (81)     8553     (81)     8557     (81)     8558     (81)     8559     (81)     8560     (81)     8564     (81)     8565     (81)
                  8566     (81)     8567     (81)     8572     (81)     8573     (81)     8574     (81)     8575     (81)     8576     (81)     8577     (81)
                  8578     (81)     8579     (81)     8580     (81)     8581     (81)     8582     (81)     8583     (81)     8584     (81)     8585     (81)
                  8586     (81)     8623     (82)     8624     (82)     8628     (82)     8629     (82)     8630     (82)     8631     (82)     8632     (82)
                  8633     (82)     8669     (83)     8670     (83)     8671     (83)     8675     (83)     8676     (83)     8677     (83)     8678     (83)
                  8679     (83)     8680     (83)     8681     (83)     8682     (83)     8683     (83)     8684     (83)     8685     (83)     8686     (83)
                  8722     (84)     8723     (84)     8724     (84)     8728     (84)     8729     (84)     8730     (84)     8731     (84)     8732     (84)
                  8733     (84)     8734     (84)     8735     (84)     8736     (84)     8737     (84)     8738     (84)     8739     (84)     8781     (85)
```

```
        8782   (85)   8783   (85)   8784   (85)   8785   (85)   8786   (85)   8787   (85)   8788   (85)   8789   (85)
        8790   (85)   8791   (85)   8792   (85)   8793   (85)   8794   (85)   8795   (85)   8796   (85)   8797   (85)
        8798   (85)   8802   (85)   8803   (85)   8804   (85)   8805   (85)   8806   (85)   8807   (85)   8808   (85)
        8809   (85)   8835   (86)   8836   (86)   8837   (86)   8838   (86)   8839   (86)   8840   (86)   8841   (86)
        8845   (86)   8849   (86)   8850   (86)   8851   (86)   8852   (86)   8915   (87)   8916   (87)   8917   (87)
        8921   (87)   8922   (87)   8923   (87)   8924   (87)   8925   (87)   8926   (87)   8927   (87)   8928   (87)
        8929   (87)   8930   (87)   8934   (87)   8935   (87)   8939   (87)   8940   (87)   8941   (87)   8942   (87)
        8943   (87)   8944   (87)   8945   (87)   8946   (87)   8947   (87)   8951   (87)   8952   (87)   8953   (87)
        8954   (87)   8955   (87)   8956   (87)   8957   (87)   8958   (87)   8959   (87)   8963   (87)   8964   (87)
        8965   (87)   8966   (87)   8967   (87)   8968   (87)   8969   (87)   8970   (87)   8971   (87)   8972   (87)
        8973   (87)   8974   (87)   8976   (87)   8977   (87)   8978   (87)   8979   (87)   8980   (87)   8981   (87)
        8982   (87)   8983   (87)   8984   (87)   8986   (87)   8987   (87)   8988   (87)   8989   (87)   8990   (87)
        8991   (87)   8992   (87)   8993   (87)   8994   (87)   8998   (87)   8999   (87)   9000   (87)   9001   (87)
        9002   (87)   9003   (87)   9004   (87)   9005   (87)   9006   (87)   9007   (87)   9011   (87)   9012   (87)
        9016   (87)   9017   (87)   9018   (87)   9019   (87)   9020   (87)   9021   (87)   9022   (87)   9023   (87)
        9024   (87)   9028   (87)   9029   (87)   9030   (87)   9034   (87)   9035   (87)   9036   (87)   9037   (87)
        9038   (87)   9039   (87)   9040   (87)   9041   (87)   9042   (87)   9043   (87)   9044   (87)   9045   (87)
        9046   (87)   9047   (87)   9048   (87)   9049   (87)   9050   (87)   9051   (87)   9052   (87)   9053   (87)
        9054   (87)   9055   (87)   9056   (87)   9061   (87)   9062   (87)   9063   (87)   9064   (87)   9069   (87)
        9070   (87)   9071   (87)   9072   (87)   9073   (87)   9074   (87)   9075   (87)   9076   (87)   9077   (87)
        9078   (87)   9079   (87)   9080   (87)   9081   (87)   9082   (87)   9083   (87)   9084   (87)   9085   (87)
        9086   (87)   9087   (87)   9088   (87)   9089   (87)   9090   (87)   9091   (87)   9092   (87)   9093   (87)
        9094   (87)   9095   (87)   9096   (87)   9097   (87)   9098   (87)   9099   (87)   9103   (87)   9138   (88)
        9139   (88)   9140   (88)   9141   (88)   9142   (88)   9143   (88)   9144   (88)   9145   (88)   9146   (88)
        9147   (88)   9148   (88)   9149   (88)   9150   (88)   9151   (88)   9152   (88)   9153   (88)   9154   (88)
        9155   (88)   9156   (88)   9157   (88)   9158   (88)   9159   (88)   9165   (88)   9166   (88)   9167   (88)
        9168   (88)   9210   (89)   9211   (89)   9212   (89)   9213   (89)   9214   (89)   9215   (89)   9216   (89)
        9217   (89)   9218   (89)   9219   (89)   9220   (89)   9221   (89)   9222   (89)   9223   (89)   9224   (89)
        9225   (89)   9226   (89)   9227   (89)   9228   (89)   9229   (89)   9230   (89)   9231   (89)   9235   (89)
        9236   (89)   9237   (89)   9238   (89)   9239   (89)   9244   (89)   9245   (89)   9246   (89)   9247   (89)
        9248   (89)   9249   (89)   9250   (89)   9251   (89)   9252   (89)   9253   (89)   9259   (89)   9260   (89)
        9261   (89)   9262   (89)   9263   (89)   9264   (89)   9272   (89)   9273   (89)   9274   (89)   9281   (89)
        9282   (89)   9283   (89)   9284   (89)   9285   (89)   9286   (89)   9287   (89)   9288   (89)   9289   (89)
        9290   (89)   9291   (89)   9292   (89)   9293   (89)   9294   (89)   9301   (89)   9302   (89)   9303   (89)
        9307   (89)   9308   (89)   9309   (89)   9310   (89)   9311   (89)   9312   (89)   9313   (89)   9314   (89)
        9315   (89)   9316   (89)   9317   (89)   9318   (89)   9319   (89)   9320   (89)   9324   (89)   9325   (89)
        9326   (89)   9327   (89)   9328   (89)   9329   (89)   9330   (89)   9331   (89)   9335   (89)   9336   (89)
        9337   (89)   9338   (89)   9342   (89)   9343   (89)   9344   (89)   9345   (89)   9346   (89)   9347   (89)
        9348   (89)   9349   (89)   9350   (89)   9351   (89)   9352   (89)   9353   (89)   9354   (89)   9355   (89)
        9356   (89)   9358   (89)   9392   (90)   9396   (90)   9397   (90)   9398   (90)   9399   (90)   9400   (90)
        9401   (90)   9402   (90)   9403   (90)   9404   (90)   9405   (90)   9406   (90)   9407   (90)   9408   (90)
        9412   (90)   9413   (90)   9414   (90)   9415   (90)   9416   (90)   9417   (90)   9418   (90)   9419   (90)
        9420   (90)   9421   (90)   9422   (90)   9423   (90)   9424   (90)   9425   (90)   9429   (90)   9430   (90)
        9431   (90)   9432   (90)   9433   (90)   9434   (90)   9435   (90)   9436   (90)   9437   (90)   9438   (90)
        9439   (90)   9440   (90)   9441   (90)   9442   (90)   9443   (90)   9447   (90)   9448   (90)   9449   (90)
        9450   (90)   9451   (90)   9452   (90)   9453   (90)   9454   (90)   9455   (90)   9456   (90)   9457   (90)
        9458   (90)   9459   (90)   9481   (91)   9482   (91)   9483   (91)   9484   (91)   9485   (91)   9486   (91)
        9487   (91)   9488   (91)   9489   (91)   9519   (92)   9520   (92)   9521   (92)   9522   (92)   9523   (92)
        9524   (92)   9525   (92)   9526   (92)   9527   (92)   9528   (92)   9529   (92)   9530   (92)   9531   (92)
        9532   (92)   9533   (92)   9534   (92)   9535   (92)   9536   (92)   9537   (92)   9538   (92)   9539   (92)
        9540   (92)   9541   (92)   9542   (92)   9543   (92)   9544   (92)   9545   (92)   9546   (92)   9547   (92)
        9571   (93)   9572   (93)   9573   (93)   9574   (93)   9575   (93)   9576   (93)   9577   (93)   9578   (93)
        9579   (93)   9583   (93)   9584   (93)   9585   (93)   9586   (93)   9587   (93)   9588   (93)   9589   (93)
        9590   (93)   9591   (93)   9595   (93)   9597   (93)   9598   (93)   9599   (93)   9625   (94)   9626   (94)
        9627   (94)   9628   (94)   9629   (94)   9630   (94)   9631   (94)   9632   (94)   9633   (94)   9634   (94)
```

```
                9635  (94)   9636  (94)   9637  (94)   9638  (94)   9639  (94)   9640  (94)   9641  (94)   9642  (94)
                9643  (94)   9644  (94)   9645  (94)   9646  (94)   9647  (94)   9648  (94)   9649  (94)   9650  (94)
                9651  (94)   9652  (94)   9653  (94)   9654  (94)   9655  (94)   9656  (94)   9657  (94)   9658  (94)
                9659  (94)   9660  (94)   9661  (94)   9662  (94)   9663  (94)   9664  (94)   9665  (94)   9666  (94)
                9667  (94)   9668  (94)   9705  (95)   9706  (95)   9707  (95)   9708  (95)   9709  (95)   9710  (95)
                9714  (95)   9715  (95)   9716  (95)   9717  (95)   9719  (95)   9720  (95)   9724  (95)   9725  (95)
                9726  (95)   9727  (95)   9728  (95)   9729  (95)   9730  (95)   9731  (95)   9732  (95)   9733  (95)
                9734  (95)   9735  (95)   9736  (95)   9737  (95)   9738  (95)   9739  (95)   9740  (95)   9741  (95)
                9742  (95)   9743  (95)   9744  (95)   9745  (95)   9746  (95)   9747  (95)   9748  (95)   9749  (95)
                9750  (95)   9751  (95)   9752  (95)   9753  (95)   9754  (95)   9762  (95)   9763  (95)   9764  (95)
                9765  (95)   9766  (95)   9767  (95)   9768  (95)   9769  (95)   9770  (95)   9771  (95)   9772  (95)
                9776  (95)   9777  (95)   9778  (95)   9779  (95)   9780  (95)   9781  (95)   9782  (95)   9787  (95)
                9788  (95)   9789  (95)   9790  (95)   9791  (95)   9792  (95)   9793  (95)   9794  (95)   9795  (95)
                9796  (95)   9797  (95)   9798  (95)   9799  (95)   9800  (95)   9801  (95)   9802  (95)   9803  (95)
                9804  (95)   9805  (95)   9806  (95)   9807  (95)   9808  (95)   9809  (95)   9810  (95)   9811  (95)
                9834  (96)   9835  (96)   9836  (96)   9837  (96)   9838  (96)   9839  (96)   9840  (96)   9841  (96)
                9843  (96)   9845  (96)   9846  (96)   9847  (96)   9863  (97)   9864  (97)   9865  (97)   9866  (97)
                9867  (97)   9868  (97)   9891  (98)   9892  (98)   9893  (98)   9894  (98)   9895  (98)   9896  (98)
                9897  (98)   9898  (98)   9900  (98)   9902  (98)   9903  (98)   9904  (98)
.END            9914  (98)
.ENDC           1374  (5)    1375  (5)    1382  (5)    1384  (5)    1385  (5)    1386  (5)    1387  (5)    1388  (5)
                1393  (5)    1395  (5)    1396  (5)    1397  (5)    1400  (5)    1402  (5)    1404  (5)    1407  (5)
                1408  (5)    1429  (5)    1468  (6)    1526  (6)    1527  (6)    1528  (6)    1529  (6)    1530  (6)
                1531  (6)    1532  (6)    1533  (6)    1534  (6)    1535  (6)    1536  (6)    1537  (6)    1538  (6)
                1539  (6)    1540  (6)    2264  (7)    2266  (7)    2269  (7)    2270  (7)    2277  (7)    2280  (7)
                2288  (7)    2294  (7)    2314  (7)    2320  (7)    2361  (7)    2362  (7)    2378  (7)    2379  (7)
                2380  (7)    2381  (7)    2383  (7)    2467  (8)    2469  (8)    2471  (8)    2518  (9)    2568  (10)
                2569  (10)   2570  (10)   2571  (10)   2572  (10)   2628  (11)   2630  (11)   2634  (11)   2695  (12)
                2702  (12)   2703  (12)   2708  (12)   2710  (12)   2753  (13)   2800  (14)   2805  (14)   2807  (14)
                2817  (14)   2820  (14)   2821  (14)   2822  (14)   2825  (14)   2826  (14)   2827  (14)   2833  (14)
                2834  (14)   2835  (14)   2841  (14)   2848  (14)   2892  (15)   2893  (15)   2897  (15)   2899  (15)
                2941  (16)   2942  (16)   2946  (16)   2948  (16)   2951  (16)   3043  (18)   3050  (18)   3055  (18)
                3058  (18)   3063  (18)   3103  (19)   3109  (19)   3114  (19)   3117  (19)   3122  (19)   3207  (20)
                3212  (20)   3213  (20)   3216  (20)   3217  (20)   3223  (20)   3224  (20)   3225  (20)   3228  (20)
                3229  (20)   3230  (20)   3231  (20)   3236  (20)   3238  (20)   3242  (20)   3243  (20)   3247  (20)
                3248  (20)   3251  (20)   3252  (20)   3253  (20)   3257  (20)   3260  (20)   3262  (20)   3263  (20)
                3267  (20)   3269  (20)   3270  (20)   3271  (20)   3272  (20)   3278  (20)   3282  (20)   3284  (20)
                3286  (20)   3287  (20)   3291  (20)   3292  (20)   3296  (20)   3297  (20)   3298  (20)   3305  (20)
                3307  (20)   3308  (20)   3313  (20)   3314  (20)   3317  (20)   3318  (20)   3323  (20)   3324  (20)
                3325  (20)   3326  (20)   3327  (20)   3333  (20)   3337  (20)   3339  (20)   3341  (20)   3342  (20)
                3348  (20)   3349  (20)   3350  (20)   3353  (20)   3354  (20)   3358  (20)   3359  (20)   3361  (20)
                3363  (20)   3364  (20)   3366  (20)   3367  (20)   3369  (20)   3370  (20)   3373  (20)   3374  (20)
                3375  (20)   3380  (20)   3384  (20)   3394  (20)   3400  (20)   3403  (20)   3406  (20)   3412  (20)
                3413  (20)   3414  (20)   3415  (20)   3417  (20)   3419  (20)   3420  (20)   3422  (20)   3423  (20)
                3425  (20)   3462  (21)   3466  (21)   3467  (21)   3473  (21)   3475  (21)   3476  (21)   3478  (21)
                3479  (21)   3481  (21)   3482  (21)   3485  (21)   3521  (22)   3524  (22)   3525  (22)   3605  (23)
                3606  (23)   3607  (23)   3608  (23)   3611  (23)   3612  (23)   3618  (23)   3626  (23)   3627  (23)
                3629  (23)   3631  (23)   3632  (23)   3633  (23)   3634  (23)   3636  (23)   3637  (23)   3641  (23)
                3642  (23)   3643  (23)   3644  (23)   3645  (23)   3647  (23)   3648  (23)   3649  (23)   3650  (23)
                3654  (23)   3655  (23)   3661  (23)   3678  (23)   3679  (23)   3680  (23)   3683  (23)   3684  (23)
                3685  (23)   3688  (23)   3690  (23)   3691  (23)   3693  (23)   3694  (23)   3699  (23)   3706  (23)
                3707  (23)   3709  (23)   3710  (23)   3717  (23)   3718  (23)   3720  (23)   3722  (23)   3723  (23)
                3801  (24)   3811  (24)   3812  (24)   3813  (24)   3817  (24)   3822  (24)   3824  (24)   3829  (24)
                3833  (24)   3836  (24)   3856  (24)   3862  (24)   3863  (24)   3894  (25)   3967  (27)   3971  (27)
                3985  (27)   4004  (28)   4005  (28)   4006  (28)   4007  (28)   4009  (28)   4011  (28)   4012  (28)
```

```
                 4013   (28)    4014   (28)    4015   (28)    4016   (28)    4035   (29)    4036   (29)    4037   (29)    4038   (29)
                 4040   (29)    4042   (29)    4043   (29)    4044   (29)    4045   (29)    4046   (29)    4047   (29)    4099   (31)
                 4122   (32)    4125   (32)    4126   (32)    4127   (32)    4129   (32)    4130   (32)    4131   (32)    4132   (32)
                 4133   (32)    4134   (32)    4137   (32)    4155   (33)    4156   (33)    4157   (33)    4158   (33)    4208   (34)
                 4209   (34)    4210   (34)    4212   (34)    4217   (34)    4218   (34)    4219   (34)    4231   (34)    4239   (34)
                 4241   (34)    4243   (34)    4246   (34)    4247   (34)    4248   (34)    4249   (34)    4250   (34)    4251   (34)
                 4253   (34)    4254   (34)    4265   (34)    4266   (34)    4269   (34)    4272   (34)    4275   (34)    4281   (34)
                 4309   (35)    4311   (35)    4312   (35)    4313   (35)    4319   (35)    4321   (35)    4334   (35)    4337   (35)
                 4339   (35)    4344   (35)    4349   (35)    4355   (35)    4356   (35)    4375   (35)    4377   (35)    4432   (36)
                 4445   (36)    4446   (36)    4449   (36)    4451   (36)    4461   (36)    4462   (36)    4469   (36)    4514   (37)
                 4518   (37)    4523   (37)    4526   (37)    4568   (38)    4570   (38)    4571   (38)    4573   (38)    4616   (39)
                 4618   (39)    4619   (39)    4620   (39)    4621   (39)    4622   (39)    4623   (39)    4624   (39)    4628   (39)
                 4633   (39)    4649   (39)    4652   (39)    4653   (39)    4655   (39)    4656   (39)    4658   (39)    4659   (39)
                 4660   (39)    4666   (39)    4672   (39)    4673   (39)    4675   (39)    4676   (39)    4678   (39)    4685   (39)
                 4686   (39)    4689   (39)    4690   (39)    4691   (39)    4692   (39)    4696   (39)    4775   (40)    4777   (40)
                 4782   (40)    4784   (40)    4838   (40)    4842   (40)    4843   (40)    4845   (40)    4847   (40)    4851   (40)
                 4858   (40)    4859   (40)    4861   (40)    4869   (40)    4879   (40)    4880   (40)    4882   (40)    4948   (41)
                 4954   (41)    4957   (41)    4983   (41)    4987   (41)    4989   (41)    4990   (41)    4991   (41)    4992   (41)
                 4994   (41)    4995   (41)    4996   (41)    4999   (41)    5002   (41)    5003   (41)    5004   (41)    5005   (41)
                 5006   (41)    5152   (45)    5154   (45)    5169   (45)    5170   (45)    5172   (45)    5173   (45)    5174   (45)
                 5183   (45)    5185   (45)    5196   (45)    5205   (45)    5210   (45)    5211   (45)    5212   (45)    5214   (45)
                 5215   (45)    5217   (45)    5224   (45)    5226   (45)    5243   (45)    5246   (45)    5247   (45)    5248   (45)
                 5250   (45)    5251   (45)    5277   (45)    5283   (45)    5284   (45)    5285   (45)    5288   (45)    5289   (45)
                 5350   (46)    5352   (46)    5375   (46)    5378   (46)    5379   (46)    5380   (46)    5383   (46)    5384   (46)
                 5393   (46)    5395   (46)    5400   (46)    5402   (46)    5412   (46)    5413   (46)    5415   (46)    5416   (46)
                 5417   (46)    5478   (47)    5480   (47)    5481   (47)    5484   (47)    5487   (47)    5489   (47)    5521   (47)
                 5524   (47)    5525   (47)    5532   (47)    5535   (47)    5536   (47)    5575   (47)    5576   (47)    5582   (47)
                 5585   (47)    5586   (47)    5589   (47)    5591   (47)    5592   (47)    5601   (47)    5602   (47)    5617   (47)
                 5618   (47)    5627   (47)    5628   (47)    5639   (47)    5645   (47)    5716   (48)    5800   (49)    5801   (49)
                 5804   (49)    5810   (49)    5814   (49)    5825   (49)    5826   (49)    5827   (49)    5828   (49)    5832   (49)
                 5833   (49)    5836   (49)    5837   (49)    5838   (49)    5839   (49)    5849   (49)    5851   (49)    5852   (49)
                 5853   (49)    5854   (49)    5855   (49)    5856   (49)    5857   (49)    5858   (49)    5912   (50)    5913   (50)
                 5920   (50)    5923   (50)    5928   (50)    5930   (50)    6021   (51)    6041   (51)    6171   (53)    6173   (53)
                 6174   (53)    6175   (53)    6176   (53)    6183   (53)    6191   (53)    6200   (53)    6205   (53)    6215   (53)
                 6217   (53)    6218   (53)    6220   (53)    6268   (54)    6269   (54)    6270   (54)    6271   (54)    6273   (54)
                 6274   (54)    6279   (54)    6280   (54)    6281   (54)    6282   (54)    6283   (54)    6284   (54)    6285   (54)
                 6286   (54)    6287   (54)    6291   (54)    6292   (54)    6293   (54)    6294   (54)    6295   (54)    6297   (54)
                 6299   (54)    6303   (54)    6304   (54)    6305   (54)    6306   (54)    6307   (54)    6309   (54)    6313   (54)
                 6319   (54)    6325   (54)    6330   (54)    6331   (54)    6333   (54)    6334   (54)    6343   (54)    6344   (54)
                 6345   (54)    6428   (56)    6429   (56)    6438   (56)    6441   (56)    6442   (56)    6447   (56)    6448   (56)
                 6449   (56)    6450   (56)    6453   (56)    6454   (56)    6455   (56)    6456   (56)    6460   (56)    6464   (56)
                 6465   (56)    6477   (56)    6478   (56)    6479   (56)    6480   (56)    6493   (56)    6494   (56)    6495   (56)
                 6497   (56)    6498   (56)    6499   (56)    6502   (56)    6503   (56)    6507   (56)    6552   (57)    6555   (57)
                 6563   (57)    6565   (57)    6569   (57)    6570   (57)    6576   (57)    6577   (57)    6580   (57)    6581   (57)
                 6582   (57)    6583   (57)    6626   (58)    6639   (58)    6646   (58)    6655   (58)    6661   (58)    6662   (58)
                 6665   (58)    6687   (58)    6835   (61)    6838   (61)    6839   (61)    6848   (61)    6851   (61)    6855   (61)
                 6857   (61)    6858   (61)    6860   (61)    6861   (61)    6862   (61)    6917   (62)    6919   (62)    6924   (62)
                 6927   (62)    6935   (62)    7007   (63)    7009   (63)    7014   (63)    7017   (63)    7028   (63)    7031   (63)
                 7035   (63)    7036   (63)    7038   (63)    7039   (63)    7041   (63)    7042   (63)    7043   (63)    7049   (63)
                 7050   (63)    7052   (63)    7053   (63)    7055   (63)    7056   (63)    7057   (63)    7065   (63)    7131   (64)
                 7143   (64)    7147   (64)    7150   (64)    7152   (64)    7154   (64)    7162   (64)    7164   (64)    7165   (64)
                 7170   (64)    7171   (64)    7177   (64)    7197   (64)    7199   (64)    7297   (65)    7311   (65)    7314   (65)
                 7320   (65)    7324   (65)    7426   (66)    7448   (66)    7451   (66)    7490   (66)    7491   (66)    7497   (66)
                 7504   (66)    7505   (66)    7507   (66)    7523   (66)    7525   (66)    7527   (66)    7529   (66)    7539   (66)
                 7551   (66)    7561   (66)    7562   (66)    7575   (66)    7577   (66)    7611   (67)    7622   (67)    7623   (67)
```

```
                7629    (67)    7680    (67)    7681    (67)    7682    (67)    7686    (67)    7687    (67)    7693    (67)    7714    (67)
                7726    (67)    7728    (67)    7735    (67)    7736    (67)    7737    (67)    7738    (67)    7739    (67)    7743    (67)
                7746    (67)    7747    (67)    7749    (67)    7750    (67)    7752    (67)    7754    (67)    7756    (67)    7760    (67)
                7761    (67)    7767    (67)    7768    (67)    7769    (67)    7771    (67)    7776    (67)    7777    (67)    7778    (67)
                7780    (67)    7781    (67)    7786    (67)    7787    (67)    7788    (67)    7790    (67)    7791    (67)    7824    (68)
                7861    (69)    7914    (70)    7917    (70)    7924    (70)    7926    (70)    7928    (70)    7929    (70)    7930    (70)
                7933    (70)    7937    (70)    7941    (70)    7949    (70)    7951    (70)    7954    (70)    7956    (70)    7957    (70)
                7960    (70)    7961    (70)    7964    (70)    7965    (70)    7966    (70)    7971    (70)    7974    (70)    7976    (70)
                7977    (70)    7978    (70)    7981    (70)    7982    (70)    7985    (70)    7986    (70)    7987    (70)    7988    (70)
                7989    (70)    7990    (70)    7992    (70)    8032    (71)    8045    (71)    8048    (71)    8050    (71)    8060    (71)
                8062    (71)    8068    (71)    8069    (71)    8074    (71)    8075    (71)    8076    (71)    8077    (71)    8078    (71)
                8079    (71)    8108    (72)    8109    (72)    8110    (72)    8111    (72)    8143    (73)    8144    (73)    8145    (73)
                8146    (73)    8242    (76)    8244    (76)    8247    (76)    8248    (76)    8249    (76)    8254    (76)    8260    (76)
                8261    (76)    8262    (76)    8267    (76)    8271    (76)    8274    (76)    8275    (76)    8281    (76)    8287    (76)
                8297    (76)    8303    (76)    8308    (76)    8310    (76)    8313    (76)    8314    (76)    8316    (76)    8317    (76)
                8320    (76)    8321    (76)    8322    (76)    8325    (76)    8326    (76)    8327    (76)    8334    (76)    8336    (76)
                8337    (76)    8339    (76)    8340    (76)    8342    (76)    8343    (76)    8345    (76)    8346    (76)    8347    (76)
                8349    (76)    8390    (77)    8391    (77)    8420    (78)    8424    (78)    8426    (78)    8442    (79)    8443    (79)
                8448    (79)    8449    (79)    8452    (79)    8454    (79)    8455    (79)    8456    (79)    8457    (79)    8460    (79)
                8497    (80)    8502    (80)    8503    (80)    8504    (80)    8505    (80)    8551    (81)    8557    (81)    8558    (81)
                8560    (81)    8564    (81)    8565    (81)    8566    (81)    8572    (81)    8573    (81)    8574    (81)    8575    (81)
                8576    (81)    8579    (81)    8624    (82)    863     (4)     8630    (82)    8670    (83)    8671    (83)    8675    (83)
                8676    (83)    8679    (83)    8680    (83)    8681    (83)    8682    (83)    8683    (83)    8723    (84)    8724    (84)
                8728    (84)    8729    (84)    8732    (84)    8733    (84)    8734    (84)    8735    (84)    8736    (84)    8781    (85)
                8782    (85)    8783    (85)    8784    (85)    8787    (85)    8790    (85)    8791    (85)    8802    (85)    8805    (85)
                8806    (85)    8845    (86)    8915    (87)    8916    (87)    8917    (87)    8924    (87)    8927    (87)    8928    (87)
                8930    (87)    8934    (87)    8935    (87)    8939    (87)    8945    (87)    8946    (87)    8947    (87)    8951    (87)
                8952    (87)    8956    (87)    8957    (87)    8958    (87)    8963    (87)    8969    (87)    8970    (87)    8971    (87)
                8973    (87)    8979    (87)    8980    (87)    8981    (87)    8983    (87)    8989    (87)    8990    (87)    8991    (87)
                8993    (87)    8998    (87)    8999    (87)    9002    (87)    9005    (87)    9011    (87)    9016    (87)    9020    (87)
                9022    (87)    9024    (87)    9034    (87)    9037    (87)    9040    (87)    9041    (87)    9042    (87)    9043    (87)
                9044    (87)    9045    (87)    9047    (87)    9048    (87)    9049    (87)    9051    (87)    9052    (87)    9053    (87)
                9056    (87)    9061    (87)    9062    (87)    9069    (87)    9070    (87)    9073    (87)    9074    (87)    9075    (87)
                9076    (87)    9077    (87)    9078    (87)    9079    (87)    9081    (87)    9082    (87)    9084    (87)    9085    (87)
                9086    (87)    9087    (87)    9089    (87)    9090    (87)    9091    (87)    9092    (87)    9093    (87)    9095    (87)
                9096    (87)    9098    (87)    9099    (87)    9138    (88)    9139    (88)    9140    (88)    9144    (88)    9145    (88)
                9148    (88)    9150    (88)    9151    (88)    9165    (88)    9212    (89)    9213    (89)    9215    (89)    9218    (89)
                9219    (89)    9220    (89)    9221    (89)    9224    (89)    9226    (89)    9227    (89)    9228    (89)    9229    (89)
                9235    (89)    9236    (89)    9247    (89)    9249    (89)    9250    (89)    9252    (89)    9253    (89)    9259    (89)
                9272    (89)    9273    (89)    9281    (89)    9282    (89)    9283    (89)    9285    (89)    9287    (89)    9294    (89)
                9301    (89)    9302    (89)    9303    (89)    9307    (89)    9308    (89)    9309    (89)    9311    (89)    9312    (89)
                9313    (89)    9315    (89)    9316    (89)    9319    (89)    9320    (89)    9335    (89)    9336    (89)    9337    (89)
                9342    (89)    9343    (89)    9344    (89)    9345    (89)    9346    (89)    9349    (89)    9350    (89)    9392    (90)
                9396    (90)    9399    (90)    9402    (90)    9404    (90)    9405    (90)    9406    (90)    9412    (90)    9413    (90)
                9416    (90)    9417    (90)    9419    (90)    9420    (90)    9422    (90)    9425    (90)    9429    (90)    9430    (90)
                9431    (90)    9434    (90)    9437    (90)    9439    (90)    9440    (90)    9441    (90)    9447    (90)    9448    (90)
                9449    (90)    9450    (90)    9452    (90)    9453    (90)    9455    (90)    9458    (90)    9481    (91)    9482    (91)
                9488    (91)    9519    (92)    9520    (92)    9521    (92)    9525    (92)    9529    (92)    9532    (92)    9535    (92)
                9536    (92)    9539    (92)    9584    (93)    9625    (94)    9626    (94)    9627    (94)    9628    (94)    9629    (94)
                9630    (94)    9631    (94)    9632    (94)    9633    (94)    9634    (94)    9635    (94)    9636    (94)    9641    (94)
                9644    (94)    9645    (94)    9650    (94)    9652    (94)    9653    (94)    9654    (94)    9655    (94)    9656    (94)
                9657    (94)    9658    (94)    9660    (94)    9661    (94)    9663    (94)    9664    (94)    9665    (94)    9666    (94)
                9667    (94)    9706    (95)    9708    (95)    9710    (95)    9714    (95)    9715    (95)    9716    (95)    9719    (95)
                9720    (95)    9724    (95)    9725    (95)    9726    (95)    9727    (95)    9728    (95)    9729    (95)    9731    (95)
                9732    (95)    9733    (95)    9734    (95)    9735    (95)    9737    (95)    9738    (95)    9739    (95)    9740    (95)
```

```
                   9741     (95)     9743     (95)     9744     (95)     9745     (95)     9746     (95)     9747     (95)     9749     (95)     9750     (95)
                   9751     (95)     9752     (95)     9753     (95)     9754     (95)     9762     (95)     9763     (95)     9766     (95)     9767     (95)
                   9769     (95)     9776     (95)     9777     (95)     9778     (95)     9779     (95)     9787     (95)     9788     (95)     9789     (95)
                   9793     (95)     9796     (95)     9797     (95)     9799     (95)     9801     (95)     9802     (95)     9803     (95)     9805     (95)
                   9806     (95)     9808     (95)     9809     (95)     9810     (95)     9835     (96)     9836     (96)     9840     (96)     9843     (96)
                   9845     (96)     9846     (96)     9863     (97)     9864     (97)     9865     (97)     9866     (97)     9867     (97)     9892     (98)
                   9893     (98)     9897     (98)     9900     (98)     9902     (98)     9903     (98)
.ENDM              1375     (5)      1418     (5)      2264     (7)      2265     (7)      2471     (8)      2518     (9)      2633     (11)     2700     (12)
                   462      (2)      488      (2)      516      (2)      547      (2)      578      (2)      608      (2)      654      (2)      705      (2)
                   728      (2)      752      (2)      765      (2)      783      (2)      816      (2)      841      (2)      849      (2)      856      (3)
                   863      (4)
.ENDR              1374     (5)      1385     (5)      1393     (5)      1526     (6)      1527     (6)      1528     (6)      1529     (6)      1530     (6)
                   1531     (6)      1532     (6)      1533     (6)      1534     (6)      1535     (6)      1536     (6)      1537     (6)      1538     (6)
                   1539     (6)      1540     (6)      2269     (7)      2270     (7)      2314     (7)      2361     (7)      2362     (7)      2378     (7)
                   2379     (7)      2380     (7)      2381     (7)      2568     (10)     2569     (10)     2570     (10)     2571     (10)     2572     (10)
                   2708     (12)     2710     (12)     2807     (14)     2893     (15)     2942     (16)     3207     (20)     3230     (20)     3252     (20)
                   3253     (20)     3260     (20)     3262     (20)     3267     (20)     3305     (20)     3307     (20)     3313     (20)     3314     (20)
                   3323     (20)     3358     (20)     3369     (20)     3403     (20)     3521     (22)     3606     (23)     3627     (23)     3637     (23)
                   3650     (23)     3661     (23)     3679     (23)     3680     (23)     3811     (24)     3812     (24)     3813     (24)     3817     (24)
                   3856     (24)     3862     (24)     3863     (24)     3894     (25)     4009     (28)     4011     (28)     4040     (29)     4042     (29)
                   4122     (32)     4129     (32)     4137     (32)     4157     (33)     4209     (34)     4210     (34)     4217     (34)     4218     (34)
                   4219     (34)     4231     (34)     4241     (34)     4251     (34)     4254     (34)     4265     (34)     4272     (34)     4309     (35)
                   4311     (35)     4313     (35)     4334     (35)     4616     (39)     4652     (39)     4672     (39)     4685     (39)     4954     (41)
                   4957     (41)     4983     (41)     4987     (41)     4994     (41)     5004     (41)     5480     (47)     5481     (47)     5575     (47)
                   5589     (47)     5591     (47)     5601     (47)     5617     (47)     5627     (47)     5639     (47)     5645     (47)     5827     (49)
                   5849     (49)     5857     (49)     6171     (53)     6428     (56)     6494     (56)     6503     (56)     7036     (63)     7050     (63)
                   7131     (64)     7297     (65)     7320     (65)     7426     (66)     7490     (66)     7491     (66)     7507     (66)     7523     (66)
                   7529     (66)     7629     (67)     7680     (67)     7681     (67)     7682     (67)     7686     (67)     7687     (67)     7693     (67)
                   7726     (67)     7735     (67)     7736     (67)     7737     (67)     7738     (67)     7739     (67)     7749     (67)     7750     (67)
                   7756     (67)     7760     (67)     7761     (67)     7771     (67)     7780     (67)     7781     (67)     7790     (67)     7791     (67)
                   7924     (70)     7933     (70)     8069     (71)     8247     (76)     8260     (76)     8261     (76)     8262     (76)     8267     (76)
                   8346     (76)     8442     (79)     8443     (79)     8452     (79)     8455     (79)     8460     (79)     8497     (80)     8557     (81)
                   8671     (83)     8676     (83)     8724     (84)     8729     (84)     8782     (85)     8783     (85)     8845     (86)     8916     (87)
                   8917     (87)     8930     (87)     8934     (87)     8935     (87)     8939     (87)     8945     (87)     8946     (87)     8947     (87)
                   8951     (87)     8952     (87)     8956     (87)     8957     (87)     8958     (87)     8963     (87)     8973     (87)     8983     (87)
                   8993     (87)     9011     (87)     9042     (87)     9043     (87)     9049     (87)     9076     (87)     9081     (87)     9090     (87)
                   9091     (87)     9099     (87)     9139     (88)     9140     (88)     9150     (88)     9224     (89)     9249     (89)     9259     (89)
                   9272     (89)     9287     (89)     9337     (89)     9396     (90)     9431     (90)     9519     (92)     9628     (94)     9629     (94)
                   9633     (94)     9652     (94)     9655     (94)     9656     (94)     9665     (94)     9706     (95)     9726     (95)     9727     (95)
                   9728     (95)     9835     (96)     9836     (96)     9843     (96)     9845     (96)     9864     (97)     9867     (97)     9892     (98)
                   9893     (98)     9900     (98)     9902     (98)
.IDENT             3        (1)
.IF                1374     (5)      1375     (5)      1384     (5)      1385     (5)      1386     (5)      1387     (5)      1388     (5)      1393     (5)
                   1395     (5)      1396     (5)      1397     (5)      1400     (5)      1402     (5)      1404     (5)      1407     (5)      1408     (5)
                   1429     (5)      1468     (6)      1526     (6)      1527     (6)      1528     (6)      1529     (6)      1530     (6)      1531     (6)
                   1532     (6)      1533     (6)      1534     (6)      1535     (6)      1536     (6)      1537     (6)      1538     (6)      1539     (6)
                   1540     (6)      2264     (7)      2266     (7)      2269     (7)      2270     (7)      2277     (7)      2280     (7)      2288     (7)
                   2294     (7)      2314     (7)      2320     (7)      2361     (7)      2362     (7)      2378     (7)      2379     (7)      2380     (7)
                   2381     (7)      2383     (7)      2467     (8)      2469     (8)      2471     (8)      2518     (9)      2568     (10)     2569     (10)
                   2570     (10)     2571     (10)     2572     (10)     2628     (11)     2630     (11)     2634     (11)     2695     (12)     2702     (12)
                   2703     (12)     2708     (12)     2710     (12)     2753     (13)     2800     (14)     2805     (14)     2807     (14)     2817     (14)
                   2820     (14)     2821     (14)     2822     (14)     2825     (14)     2826     (14)     2827     (14)     2833     (14)     2834     (14)
                   2835     (14)     284i     (14)     2848     (14)     2892     (15)     2893     (15)     2897     (15)     2899     (15)     2941     (16)
                   2942     (16)     2946     (16)     2948     (16)     2951     (16)     3043     (18)     3050     (18)     3055     (18)     3058     (18)
                   3063     (18)     3103     (19)     3109     (19)     3114     (19)     3117     (19)     3122     (19)     3207     (20)     3212     (20)
```

```
                3213   (20)   3216   (20)   3217   (20)   3223   (20)   3224   (20)   3225   (20)   3228   (20)   3229   (20)
                3230   (20)   3231   (20)   3236   (20)   3238   (20)   3242   (20)   3243   (20)   3247   (20)   3248   (20)
                3251   (20)   3252   (20)   3253   (20)   3257   (20)   3260   (20)   3262   (20)   3263   (20)   3267   (20)
                3269   (20)   3270   (20)   3271   (20)   3272   (20)   3278   (20)   3282   (20)   3284   (20)   3286   (20)
                3287   (20)   3291   (20)   3292   (20)   3296   (20)   3297   (20)   3298   (20)   3305   (20)   3307   (20)
                3308   (20)   3313   (20)   3314   (20)   3317   (20)   3318   (20)   3323   (20)   3324   (20)   3325   (20)
                3326   (20)   3327   (20)   3333   (20)   3337   (20)   3339   (20)   3341   (20)   3342   (20)   3348   (20)
                3349   (20)   3350   (20)   3353   (20)   3354   (20)   3358   (20)   3359   (20)   3361   (20)   3363   (20)
                3364   (20)   3366   (20)   3367   (20)   3369   (20)   3370   (20)   3373   (20)   3374   (20)   3375   (20)
                3380   (20)   3384   (20)   3394   (20)   3400   (20)   3403   (20)   3406   (20)   3412   (20)   3413   (20)
                3414   (20)   3415   (20)   3417   (20)   3419   (20)   3420   (20)   3422   (20)   3423   (20)   3425   (20)
                3462   (21)   3466   (21)   3467   (21)   3473   (21)   3475   (21)   3476   (21)   3478   (21)   3479   (21)
                3481   (21)   3482   (21)   3485   (21)   3521   (22)   3524   (22)   3525   (22)   3605   (23)   3606   (23)
                3607   (23)   3608   (23)   3611   (23)   3612   (23)   3618   (23)   3626   (23)   3627   (23)   3629   (23)
                3631   (23)   3632   (23)   3633   (23)   3634   (23)   3636   (23)   3637   (23)   3641   (23)   3642   (23)
                3643   (23)   3644   (23)   3645   (23)   3647   (23)   3648   (23)   3649   (23)   3650   (23)   3654   (23)
                3655   (23)   3661   (23)   3678   (23)   3679   (23)   3680   (23)   3683   (23)   3684   (23)   3685   (23)
                3688   (23)   3690   (23)   3691   (23)   3693   (23)   3694   (23)   3699   (23)   3706   (23)   3707   (23)
                3709   (23)   3710   (23)   3717   (23)   3718   (23)   3720   (23)   3722   (23)   3723   (23)   3801   (24)
                3811   (24)   3812   (24)   3813   (24)   3817   (24)   3822   (24)   3824   (24)   3829   (24)   3833   (24)
                3836   (24)   3856   (24)   3862   (24)   3863   (24)   3894   (25)   3967   (27)   3971   (27)   3985   (27)
                4004   (28)   4005   (28)   4006   (28)   4007   (28)   4009   (28)   4011   (28)   4012   (28)   4013   (28)
                4014   (28)   4015   (28)   4016   (28)   4035   (29)   4036   (29)   4037   (29)   4038   (29)   4040   (29)
                4042   (29)   4043   (29)   4044   (29)   4045   (29)   4046   (29)   4047   (29)   4099   (31)   4122   (32)
                4125   (32)   4126   (32)   4127   (32)   4129   (32)   4130   (32)   4131   (32)   4132   (32)   4133   (32)
                4134   (32)   4137   (32)   4155   (33)   4156   (33)   4157   (33)   4158   (33)   4208   (34)   4209   (34)
                4210   (34)   4212   (34)   4217   (34)   4218   (34)   4219   (34)   4231   (34)   4239   (34)   4241   (34)
                4243   (34)   4246   (34)   4247   (34)   4248   (34)   4249   (34)   4250   (34)   4251   (34)   4253   (34)
                4254   (34)   4265   (34)   4266   (34)   4269   (34)   4272   (34)   4275   (34)   4281   (34)   4309   (35)
                4311   (35)   4312   (35)   4313   (35)   4319   (35)   4321   (35)   4334   (35)   4337   (35)   4339   (35)
                4344   (35)   4349   (35)   4355   (35)   4356   (35)   4375   (35)   4377   (35)   4432   (36)   4445   (36)
                4446   (36)   4449   (36)   4451   (36)   4461   (36)   4462   (36)   4469   (36)   4514   (37)   4518   (37)
                4523   (37)   4526   (37)   4568   (38)   4570   (38)   4571   (38)   4573   (38)   4616   (39)   4618   (39)
                4619   (39)   4620   (39)   4621   (39)   4622   (39)   4623   (39)   4624   (39)   4628   (39)   4633   (39)
                4649   (39)   4652   (39)   4653   (39)   4655   (39)   4656   (39)   4658   (39)   4659   (39)   4660   (39)
                4666   (39)   4672   (39)   4673   (39)   4675   (39)   4676   (39)   4678   (39)   4685   (39)   4686   (39)
                4689   (39)   4690   (39)   4691   (39)   4692   (39)   4696   (39)   4775   (40)   4777   (40)   4782   (40)
                4784   (40)   4838   (40)   4842   (40)   4843   (40)   4845   (40)   4847   (40)   4851   (40)   4858   (40)
                4859   (40)   4861   (40)   4869   (40)   4879   (40)   4880   (40)   4882   (40)   4948   (41)   4954   (41)
                4957   (41)   4983   (41)   4987   (41)   4989   (41)   4990   (41)   4991   (41)   4992   (41)   4994   (41)
                4995   (41)   4996   (41)   4999   (41)   5002   (41)   5003   (41)   5004   (41)   5005   (41)   5006   (41)
                5152   (45)   5154   (45)   5169   (45)   5170   (45)   5172   (45)   5173   (45)   5174   (45)   5183   (45)
                5185   (45)   5196   (45)   5205   (45)   5210   (45)   5211   (45)   5212   (45)   5214   (45)   5215   (45)
                5217   (45)   5224   (45)   5226   (45)   5243   (45)   5246   (45)   5247   (45)   5248   (45)   5250   (45)
                5251   (45)   5277   (45)   5283   (45)   5284   (45)   5285   (45)   5288   (45)   5289   (45)   5350   (46)
                5352   (46)   5375   (46)   5378   (46)   5379   (46)   5380   (46)   5383   (46)   5384   (46)   5393   (46)
                5395   (46)   5400   (46)   5402   (46)   5412   (46)   5413   (46)   5415   (46)   5416   (46)   5417   (46)
                5478   (47)   5480   (47)   5481   (47)   5484   (47)   5487   (47)   5489   (47)   5521   (47)   5524   (47)
                5525   (47)   5532   (47)   5535   (47)   5536   (47)   5575   (47)   5576   (47)   5582   (47)   5585   (47)
                5586   (47)   5589   (47)   5591   (47)   5592   (47)   5601   (47)   5602   (47)   5617   (47)   5618   (47)
                5627   (47)   5628   (47)   5639   (47)   5645   (47)   5716   (48)   5800   (49)   5801   (49)   5804   (49)
                5810   (49)   5814   (49)   5825   (49)   5826   (49)   5827   (49)   5828   (49)   5832   (49)   5833   (49)
                5836   (49)   5837   (49)   5838   (49)   5839   (49)   5849   (49)   5851   (49)   5852   (49)   5853   (49)
                5854   (49)   5855   (49)   5856   (49)   5857   (49)   5858   (49)   5912   (50)   5913   (50)   5920   (50)
                5923   (50)   5928   (50)   5930   (50)   6021   (51)   6041   (51)   6171   (53)   6173   (53)   6174   (53)
```

```
             6175   (53)   6176   (53)   6183   (53)   6191   (53)   6200   (53)   6205   (53)   6215   (53)   6217   (53)
             6218   (53)   6220   (53)   6268   (54)   6269   (54)   6270   (54)   6271   (54)   6273   (54)   6274   (54)
             6279   (54)   6280   (54)   6281   (54)   6282   (54)   6283   (54)   6284   (54)   6285   (54)   6286   (54)
             6287   (54)   6291   (54)   6292   (54)   6293   (54)   6294   (54)   6295   (54)   6297   (54)   6299   (54)
             6303   (54)   6304   (54)   6305   (54)   6306   (54)   6307   (54)   6309   (54)   6313   (54)   6319   (54)
             6325   (54)   6330   (54)   6331   (54)   6333   (54)   6334   (54)   6343   (54)   6344   (54)   6345   (54)
             6428   (56)   6429   (56)   6438   (56)   6441   (56)   6442   (56)   6447   (56)   6448   (56)   6449   (56)
             6450   (56)   6453   (56)   6454   (56)   6455   (56)   6456   (56)   6460   (56)   6464   (56)   6465   (56)
             6477   (56)   6478   (56)   6479   (56)   6480   (56)   6493   (56)   6494   (56)   6495   (56)   6497   (56)
             6498   (56)   6499   (56)   6502   (56)   6503   (56)   6507   (56)   6552   (57)   6555   (57)   6563   (57)
             6565   (57)   6569   (57)   6570   (57)   6576   (57)   6577   (57)   6580   (57)   6581   (57)   6582   (57)
             6583   (57)   6626   (58)   6639   (58)   6646   (58)   6655   (58)   6661   (58)   6662   (58)   6665   (58)
             6687   (58)   6835   (61)   6838   (61)   6839   (61)   6848   (61)   6851   (61)   6855   (61)   6857   (61)
             6858   (61)   6860   (61)   6861   (61)   6862   (61)   6917   (62)   6919   (62)   6924   (62)   6927   (62)
             6935   (62)   7007   (63)   7009   (63)   7014   (63)   7017   (63)   7028   (63)   7031   (63)   7035   (63)
             7036   (63)   7038   (63)   7039   (63)   7041   (63)   7042   (63)   7043   (63)   7049   (63)   7050   (63)
             7052   (63)   7053   (63)   7055   (63)   7056   (63)   7057   (63)   7065   (63)   7131   (64)   7143   (64)
             7147   (64)   7150   (64)   7152   (64)   7154   (64)   7162   (64)   7164   (64)   7165   (64)   7170   (64)
             7171   (64)   7177   (64)   7197   (64)   7199   (64)   7297   (65)   7311   (65)   7314   (65)   7320   (65)
             7324   (65)   7426   (66)   7448   (66)   7451   (66)   7490   (66)   7491   (66)   7497   (66)   7504   (66)
             7505   (66)   7507   (66)   7523   (66)   7525   (66)   7527   (66)   7529   (66)   7539   (66)   7551   (66)
             7561   (66)   7562   (66)   7575   (66)   7577   (66)   7611   (67)   7622   (67)   7623   (67)   7629   (67)
             7680   (67)   7681   (67)   7682   (67)   7686   (67)   7687   (67)   7693   (67)   7714   (67)   7726   (67)
             7728   (67)   7735   (67)   7736   (67)   7737   (67)   7738   (67)   7739   (67)   7743   (67)   7746   (67)
             7747   (67)   7749   (67)   7750   (67)   7752   (67)   7754   (67)   7756   (67)   7760   (67)   7761   (67)
             7767   (67)   7768   (67)   7769   (67)   7771   (67)   7776   (67)   7777   (67)   7778   (67)   7780   (67)
             7781   (67)   7786   (67)   7787   (67)   7788   (67)   7790   (67)   7791   (67)   7824   (68)   7861   (69)
             7914   (70)   7917   (70)   7924   (70)   7926   (70)   7928   (70)   7929   (70)   7930   (70)   7933   (70)
             7937   (70)   7941   (70)   7949   (70)   7951   (70)   7954   (70)   7956   (70)   7957   (70)   7960   (70)
             7961   (70)   7964   (70)   7965   (70)   7966   (70)   7971   (70)   7974   (70)   7976   (70)   7977   (70)
             7978   (70)   7981   (70)   7982   (70)   7985   (70)   7986   (70)   7987   (70)   7988   (70)   7989   (70)
             7990   (70)   7992   (70)   8032   (71)   8045   (71)   8048   (71)   8050   (71)   8060   (71)   8062   (71)
             8068   (71)   8069   (71)   8074   (71)   8075   (71)   8076   (71)   8077   (71)   8078   (71)   8079   (71)
             8108   (72)   8109   (72)   8110   (72)   8111   (72)   8143   (73)   8144   (73)   8145   (73)   8146   (73)
             8242   (76)   8244   (76)   8247   (76)   8248   (76)   8249   (76)   8254   (76)   8260   (76)   8261   (76)
             8262   (76)   8267   (76)   8271   (76)   8274   (76)   8275   (76)   8281   (76)   8287   (76)   8297   (76)
             8303   (76)   8308   (76)   8310   (76)   8313   (76)   8314   (76)   8316   (76)   8317   (76)   8320   (76)
             8321   (76)   8322   (76)   8325   (76)   8326   (76)   8327   (76)   8334   (76)   8336   (76)   8337   (76)
             8339   (76)   8340   (76)   8342   (76)   8343   (76)   8345   (76)   8346   (76)   8347   (76)   8349   (76)
             8390   (77)   8391   (77)   8420   (78)   8424   (78)   8426   (78)   8442   (79)   8443   (79)   8448   (79)
             8449   (79)   8452   (79)   8454   (79)   8455   (79)   8456   (79)   8457   (79)   8460   (79)   8497   (80)
             8502   (80)   8503   (80)   8504   (80)   8505   (80)   8551   (81)   8557   (81)   8558   (81)   8560   (81)
             8564   (81)   8565   (81)   8566   (81)   8572   (81)   8573   (81)   8574   (81)   8575   (81)   8576   (81)
             8579   (81)   8624   (82)   863    (4)    8630   (82)   8670   (83)   8671   (83)   8675   (83)   8676   (83)
             8679   (83)   868    (5)    8680   (83)   8681   (83)   8682   (83)   8683   (83)   8723   (84)   8724   (84)
             8728   (84)   8729   (84)   8732   (84)   8733   (84)   8734   (84)   8735   (84)   8736   (84)   8781   (85)
             8782   (85)   8783   (85)   8784   (85)   8787   (85)   8790   (85)   8791   (85)   8802   (85)   8805   (85)
             8806   (85)   8845   (86)   8915   (87)   8916   (87)   8917   (87)   8924   (87)   8927   (87)   8928   (87)
             8930   (87)   8934   (87)   8935   (87)   8939   (87)   8945   (87)   8946   (87)   8947   (87)   8951   (87)
             8952   (87)   8956   (87)   8957   (87)   8958   (87)   8963   (87)   8969   (87)   8970   (87)   8971   (87)
             8973   (87)   8979   (87)   8980   (87)   8981   (87)   8983   (87)   8989   (87)   8990   (87)   8991   (87)
             8993   (87)   8998   (87)   8999   (87)   9002   (87)   9005   (87)   9011   (87)   9016   (87)   9020   (87)
             9022   (87)   9024   (87)   9034   (87)   9037   (87)   9040   (87)   9041   (87)   9042   (87)   9043   (87)
             9044   (87)   9045   (87)   9047   (87)   9048   (87)   9049   (87)   9051   (87)   9052   (87)   9053   (87)
             9056   (87)   9061   (87)   9062   (87)   9069   (87)   9070   (87)   9073   (87)   9074   (87)   9075   (87)
```

```
                 9076   (87)    9077   (87)    9078   (87)    9079   (87)    9081   (87)    9082   (87)    9084   (87)    9085   (87)
                 9086   (87)    9087   (87)    9089   (87)    9090   (87)    9091   (87)    9092   (87)    9093   (87)    9095   (87)
                 9096   (87)    9098   (87)    9099   (87)    9138   (88)    9139   (88)    9140   (88)    9144   (88)    9145   (88)
                 9148   (88)    9150   (88)    9151   (88)    9165   (88)    9212   (89)    9213   (89)    9215   (89)    9218   (89)
                 9219   (89)    9220   (89)    9221   (89)    9224   (89)    9226   (89)    9227   (89)    9228   (89)    9229   (89)
                 9235   (89)    9236   (89)    9247   (89)    9249   (89)    9250   (89)    9252   (89)    9253   (89)    9259   (89)
                 9272   (89)    9273   (89)    9281   (89)    9282   (89)    9283   (89)    9285   (89)    9287   (89)    9294   (89)
                 9301   (89)    9302   (89)    9303   (89)    9307   (89)    9308   (89)    9309   (89)    9311   (89)    9312   (89)
                 9313   (89)    9315   (89)    9316   (89)    9319   (89)    9320   (89)    9335   (89)    9336   (89)    9337   (89)
                 9342   (89)    9343   (89)    9344   (89)    9345   (89)    9346   (89)    9349   (89)    9350   (89)    9392   (90)
                 9396   (90)    9399   (90)    9402   (90)    9404   (90)    9405   (90)    9406   (90)    9412   (90)    9413   (90)
                 9416   (90)    9417   (90)    9419   (90)    9420   (90)    9422   (90)    9425   (90)    9429   (90)    9430   (90)
                 9431   (90)    9434   (90)    9437   (90)    9439   (90)    9440   (90)    9441   (90)    9447   (90)    9448   (90)
                 9449   (90)    9450   (90)    9452   (90)    9453   (90)    9455   (90)    9458   (90)    9481   (91)    9482   (91)
                 9488   (91)    9519   (92)    9520   (92)    9521   (92)    9525   (92)    9529   (92)    9532   (92)    9535   (92)
                 9536   (92)    9539   (92)    9584   (93)    9625   (94)    9626   (94)    9627   (94)    9628   (94)    9629   (94)
                 9630   (94)    9631   (94)    9632   (94)    9633   (94)    9634   (94)    9635   (94)    9636   (94)    9641   (94)
                 9644   (94)    9645   (94)    9650   (94)    9652   (94)    9653   (94)    9654   (94)    9655   (94)    9656   (94)
                 9657   (94)    9658   (94)    9660   (94)    9661   (94)    9663   (94)    9664   (94)    9665   (94)    9666   (94)
                 9667   (94)    9706   (95)    9708   (95)    9710   (95)    9714   (95)    9715   (95)    9716   (95)    9719   (95)
                 9720   (95)    9724   (95)    9725   (95)    9726   (95)    9727   (95)    9728   (95)    9729   (95)    9731   (95)
                 9732   (95)    9733   (95)    9734   (95)    9735   (95)    9737   (95)    9738   (95)    9739   (95)    9740   (95)
                 9741   (95)    9743   (95)    9744   (95)    9745   (95)    9746   (95)    9747   (95)    9749   (95)    9750   (95)
                 9751   (95)    9752   (95)    9753   (95)    9754   (95)    9762   (95)    9763   (95)    9766   (95)    9767   (95)
                 9769   (95)    9776   (95)    9777   (95)    9778   (95)    9779   (95)    9787   (95)    9788   (95)    9789   (95)
                 9793   (95)    9796   (95)    9797   (95)    9799   (95)    9801   (95)    9802   (95)    9803   (95)    9805   (95)
                 9806   (95)    9808   (95)    9809   (95)    9810   (95)    9835   (96)    9836   (96)    9840   (96)    9843   (96)
                 9845   (96)    9846   (96)    9863   (97)    9864   (97)    9865   (97)    9866   (97)    9867   (97)    9892   (98)
                 9893   (98)    9897   (98)    9900   (98)    9902   (98)    9903   (98)
.IFF             1367   (5)     1374   (5)     1385   (5)     1393   (5)     1468   (6)     1526   (6)     1527   (6)     1528   (6)
                 1529   (6)     1530   (6)     1531   (6)     1532   (6)     1533   (6)     1534   (6)     1535   (6)     1536   (6)
                 1537   (6)     1538   (6)     1539   (6)     1540   (6)     2264   (7)     2266   (7)     2269   (7)     2270   (7)
                 2314   (7)     2361   (7)     2362   (7)     2378   (7)     2379   (7)     2380   (7)     2381   (7)     2471   (8)
                 2518   (9)     2568   (10)    2569   (10)    2570   (10)    2571   (10)    2572   (10)    2634   (11)    2702   (12)
                 2708   (12)    2710   (12)    2753   (13)    2807   (14)    2893   (15)    2942   (16)    3207   (20)    3230   (20)
                 3252   (20)    3253   (20)    3260   (20)    3262   (20)    3267   (20)    3305   (20)    3307   (20)    3313   (20)
                 3314   (20)    3323   (20)    3354   (20)    3358   (20)    3369   (20)    3403   (20)    3521   (22)    3606   (23)
                 3627   (23)    3637   (23)    3650   (23)    3661   (23)    3679   (23)    3680   (23)    3811   (24)    3812   (24)
                 3813   (24)    3817   (24)    3856   (24)    3862   (24)    3863   (24)    3894   (25)    4009   (28)    4011   (28)
                 4040   (29)    4042   (29)    4122   (32)    4129   (32)    4137   (32)    4157   (33)    4208   (34)    4209   (34)
                 4210   (34)    4212   (34)    4217   (34)    4218   (34)    4219   (34)    4231   (34)    4241   (34)    4251   (34)
                 4254   (34)    4265   (34)    4272   (34)    4275   (34)    4309   (35)    4311   (35)    4313   (35)    4334   (35)
                 4616   (39)    4652   (39)    4672   (39)    4685   (39)    4954   (41)    4957   (41)    4983   (41)    4987   (41)
                 4994   (41)    5004   (41)    5480   (47)    5481   (47)    5575   (47)    5589   (47)    5591   (47)    5601   (47)
                 5617   (47)    5627   (47)    5639   (47)    5645   (47)    5827   (49)    5849   (49)    5857   (49)    5930   (50)
                 6171   (53)    6428   (56)    6494   (56)    6503   (56)    7036   (63)    7050   (63)    7131   (64)    7297   (65)
                 7320   (65)    7426   (66)    7490   (66)    7491   (66)    7507   (66)    7523   (66)    7529   (66)    7561   (66)
                 7562   (66)    7629   (67)    7680   (67)    7681   (67)    7682   (67)    7686   (67)    7687   (67)    7693   (67)
                 7726   (67)    7735   (67)    7736   (67)    7737   (67)    7738   (67)    7739   (67)    7749   (67)    7750   (67)
                 7756   (67)    7760   (67)    7761   (67)    7771   (67)    7780   (67)    7781   (67)    7790   (67)    7791   (67)
                 7924   (70)    7933   (70)    8069   (71)    8247   (76)    8260   (76)    8261   (76)    8262   (76)    8267   (76)
                 8346   (76)    8442   (79)    8443   (79)    8452   (79)    8455   (79)    8460   (79)    8497   (80)    8557   (81)
                 863    (4)     8671   (83)    8676   (83)    8724   (84)    8729   (84)    8782   (85)    8783   (85)    8845   (86)
                 8916   (87)    8917   (87)    8930   (87)    8934   (87)    8935   (87)    8939   (87)    8945   (87)    8946   (87)
                 8947   (87)    8951   (87)    8952   (87)    8956   (87)    8957   (87)    8958   (87)    8963   (87)    8973   (87)
```

```
                      8983       (87)      8993       (87)      9011       (87)      9042       (87)      9043       (87)      9049       (87)      9076       (87)      9081       (87)
                      9090       (87)      9091       (87)      9099       (87)      9138       (88)      9139       (88)      9140       (88)      9150       (88)      9151       (88)
                      9165       (88)      9224       (89)      9249       (89)      9259       (89)      9272       (89)      9287       (89)      9337       (89)      9396       (90)
                      9431       (90)      9519       (92)      9628       (94)      9629       (94)      9633       (94)      9652       (94)      9655       (94)      9656       (94)
                      9665       (94)      9706       (95)      9726       (95)      9727       (95)      9728       (95)      9835       (96)      9836       (96)      9843       (96)
                      9845       (96)      9864       (97)      9867       (97)      9892       (98)      9893       (98)      9900       (98)      9902       (98)
.IRPC                 1374       (5)       1385       (5)       1393       (5)       1526       (6)       1527       (6)       1528       (6)       1529       (6)       1530       (6)
                      1531       (6)       1532       (6)       1533       (6)       1534       (6)       1535       (6)       1536       (6)       1537       (6)       1538       (6)
                      1539       (6)       1540       (6)       2269       (7)       2270       (7)       2314       (7)       2361       (7)       2362       (7)       2378       (7)
                      2379       (7)       2380       (7)       2381       (7)       2568       (10)      2569       (10)      2570       (10)      2571       (10)      2572       (10)
                      2708       (12)      2710       (12)      2807       (14)      2893       (15)      2942       (16)      3207       (20)      3230       (20)      3252       (20)
                      3253       (20)      3260       (20)      3262       (20)      3267       (20)      3305       (20)      3307       (20)      3313       (20)      3314       (20)
                      3323       (20)      3358       (20)      3369       (20)      3403       (20)      3521       (22)      3606       (23)      3627       (23)      3637       (23)
                      3650       (23)      3661       (23)      3679       (23)      3680       (23)      3811       (24)      3812       (24)      3813       (24)      3817       (24)
                      3856       (24)      3862       (24)      3863       (24)      3894       (25)      4009       (28)      4011       (28)      4040       (29)      4042       (29)
                      4122       (32)      4129       (32)      4137       (32)      4157       (33)      4209       (34)      4210       (34)      4217       (34)      4218       (34)
                      4219       (34)      4231       (34)      4241       (34)      4251       (34)      4254       (34)      4265       (34)      4272       (34)      4309       (35)
                      4311       (35)      4313       (35)      4334       (35)      4616       (39)      4652       (39)      4672       (39)      4685       (39)      4954       (41)
                      4957       (41)      4983       (41)      4987       (41)      4994       (41)      5004       (41)      5480       (47)      5481       (47)      5575       (47)
                      5589       (47)      5591       (47)      5601       (47)      5617       (47)      5627       (47)      5639       (47)      5645       (47)      5827       (49)
                      5849       (49)      5857       (49)      6171       (53)      6428       (56)      6494       (56)      6503       (56)      7036       (63)      7050       (63)
                      7131       (64)      7297       (65)      7320       (65)      7426       (66)      7490       (66)      7491       (66)      7507       (66)      7523       (66)
                      7529       (66)      7629       (67)      7680       (67)      7681       (67)      7682       (67)      7686       (67)      7687       (67)      7693       (67)
                      7726       (67)      7735       (67)      7736       (67)      7737       (67)      7738       (67)      7739       (67)      7749       (67)      7750       (67)
                      7756       (67)      7760       (67)      7761       (67)      7771       (67)      7780       (67)      7781       (67)      7790       (67)      7791       (67)
                      7924       (70)      7933       (70)      8069       (71)      8247       (76)      8260       (76)      8261       (76)      8262       (76)      8267       (76)
                      8346       (76)      8442       (79)      8443       (79)      8452       (79)      8455       (79)      8460       (79)      8497       (80)      8557       (81)
                      8671       (83)      8676       (83)      8724       (84)      8729       (84)      8782       (85)      8783       (85)      8845       (86)      8916       (87)
                      8917       (87)      8930       (87)      8934       (87)      8935       (87)      8939       (87)      8945       (87)      8946       (87)      8947       (87)
                      8951       (87)      8952       (87)      8956       (87)      8957       (87)      8958       (87)      8963       (87)      8973       (87)      8983       (87)
                      8993       (87)      9011       (87)      9042       (87)      9043       (87)      9049       (87)      9076       (87)      9081       (87)      9090       (87)
                      9091       (87)      9099       (87)      9139       (88)      9140       (88)      9150       (88)      9224       (89)      9249       (89)      9259       (89)
                      9272       (89)      9287       (89)      9337       (89)      9396       (90)      9431       (90)      9519       (92)      9628       (94)      9629       (94)
                      9633       (94)      9652       (94)      9655       (94)      9656       (94)      9665       (94)      9706       (95)      9726       (95)      9727       (95)
                      9728       (95)      9835       (96)      9836       (96)      9843       (96)      9845       (96)      9864       (97)      9867       (97)      9892       (98)
                      9893       (98)      9900       (98)      9902       (98)
.LIBRARY              424        (2)       425        (2)       426        (2)
.LIST                 1375       (5)       1384       (5)       1386       (5)       1387       (5)       1388       (5)       1395       (5)       1396       (5)       1397       (5)
                      1400       (5)       1402       (5)       1404       (5)       1407       (5)       1408       (5)       1429       (5)       2320       (7)       2469       (8)
                      2630       (11)      2695       (12)      2703       (12)      2800       (14)      2817       (14)      2820       (14)      2825       (14)      2826       (14)
                      2827       (14)      2835       (14)      2841       (14)      2848       (14)      3043       (18)      3055       (18)      3058       (18)      3063       (18)
                      3103       (19)      3114       (19)      3117       (19)      3122       (19)      3213       (20)      3216       (20)      3224       (20)      3225       (20)
                      3229       (20)      3231       (20)      3236       (20)      3238       (20)      3242       (20)      3247       (20)      3248       (20)      3251       (20)
                      3257       (20)      3263       (20)      3271       (20)      3272       (20)      3278       (20)      3282       (20)      3284       (20)      3286       (20)
                      3287       (20)      3291       (20)      3296       (20)      3297       (20)      3298       (20)      3308       (20)      3317       (20)      3318       (20)
                      3326       (20)      3327       (20)      3333       (20)      3337       (20)      3339       (20)      3341       (20)      3342       (20)      3348       (20)
                      3353       (20)      3359       (20)      3367       (20)      3374       (20)      3375       (20)      3380       (20)      3394       (20)      3406       (20)
                      3412       (20)      3413       (20)      3414       (20)      3415       (20)      3423       (20)      3425       (20)      3462       (21)      3466       (21)
                      3467       (21)      3473       (21)      3475       (21)      3478       (21)      3479       (21)      3481       (21)      3482       (21)      3485       (21)
                      3525       (22)      3605       (23)      3607       (23)      3608       (23)      3611       (23)      3612       (23)      3618       (23)      3626       (23)
                      3631       (23)      3633       (23)      3641       (23)      3643       (23)      3644       (23)      3647       (23)      3648       (23)      3654       (23)
                      3655       (23)      3683       (23)      3684       (23)      3690       (23)      3691       (23)      3693       (23)      3694       (23)      3706       (23)
                      3707       (23)      3709       (23)      3710       (23)      3717       (23)      3720       (23)      3722       (23)      3723       (23)      3829       (24)
                      3836       (24)      3967       (27)      3985       (27)      4004       (28)      4005       (28)      4006       (28)      4007       (28)      4012       (28)
                      4013       (28)      4014       (28)      4015       (28)      4016       (28)      4035       (29)      4036       (29)      4037       (29)      4038       (29)
```

```
4043   (29)    4044   (29)    4045   (29)    4046   (29)    4047   (29)    4125   (32)    4126   (32)    4127   (32)
4130   (32)    4131   (32)    4132   (32)    4133   (32)    4134   (32)    4155   (33)    4158   (33)    417    (2)
4243   (34)    4248   (34)    4253   (34)    4281   (34)    4312   (35)    4355   (35)    4375   (35)    4377   (35)
4445   (36)    4449   (36)    4451   (36)    4461   (36)    4462   (36)    4469   (36)    4518   (37)    4526   (37)
4568   (38)    4571   (38)    4573   (38)    4619   (39)    4621   (39)    4623   (39)    4633   (39)    4649   (39)
4653   (39)    4655   (39)    4658   (39)    4660   (39)    4666   (39)    4676   (39)    4686   (39)    4691   (39)
4692   (39)    4696   (39)    4784   (40)    4842   (40)    4845   (40)    4858   (40)    4861   (40)    4879   (40)
4882   (40)    4990   (41)    4991   (41)    4992   (41)    4995   (41)    4996   (41)    5002   (41)    5003   (41)
5005   (41)    5006   (41)    5154   (45)    5169   (45)    5172   (45)    5173   (45)    5185   (45)    5210   (45)
5211   (45)    5214   (45)    5217   (45)    5226   (45)    5246   (45)    5247   (45)    5250   (45)    5283   (45)
5284   (45)    5288   (45)    5352   (46)    5378   (46)    5379   (46)    5383   (46)    5395   (46)    5412   (46)
5415   (46)    5416   (46)    5478   (47)    5484   (47)    5489   (47)    5524   (47)    5535   (47)    5576   (47)
5585   (47)    5592   (47)    5602   (47)    5618   (47)    5628   (47)    5800   (49)    5801   (49)    5804   (49)
5826   (49)    5833   (49)    5838   (49)    5839   (49)    5851   (49)    5854   (49)    5858   (49)    5920   (50)
5923   (50)    6175   (53)    6176   (53)    6215   (53)    6218   (53)    6268   (54)    6269   (54)    6270   (54)
6271   (54)    6279   (54)    6280   (54)    6281   (54)    6282   (54)    6285   (54)    6291   (54)    6292   (54)
6293   (54)    6295   (54)    6297   (54)    6299   (54)    6303   (54)    6304   (54)    6306   (54)    6307   (54)
6309   (54)    6319   (54)    6325   (54)    6330   (54)    6333   (54)    6344   (54)    6429   (56)    6438   (56)
6441   (56)    6442   (56)    6449   (56)    6450   (56)    6453   (56)    6454   (56)    6455   (56)    6456   (56)
6460   (56)    6464   (56)    6465   (56)    6478   (56)    6480   (56)    6493   (56)    6498   (56)    6499   (56)
6502   (56)    6507   (56)    6552   (57)    6555   (57)    6569   (57)    6577   (57)    6580   (57)    6581   (57)
6583   (57)    6655   (58)    6661   (58)    6687   (58)    6835   (61)    6838   (61)    6839   (61)    6848   (61)
6851   (61)    6855   (61)    6861   (61)    6862   (61)    6924   (62)    6927   (62)    7014   (63)    7017   (63)
7028   (63)    7031   (63)    7035   (63)    7042   (63)    7043   (63)    7049   (63)    7056   (63)    7057   (63)
7143   (64)    7147   (64)    7154   (64)    7164   (64)    7165   (64)    7171   (64)    7177   (64)    7197   (64)
7324   (65)    7743   (67)    7746   (67)    7747   (67)    7767   (67)    7768   (67)    7769   (67)    7776   (67)
7777   (67)    7778   (67)    7786   (67)    7787   (67)    7788   (67)    7861   (69)    7914   (70)    7917   (70)
7929   (70)    7930   (70)    7937   (70)    7941   (70)    7949   (70)    7957   (70)    7960   (70)    7961   (70)
7964   (70)    7977   (70)    7978   (70)    7981   (70)    7982   (70)    7985   (70)    7987   (70)    7989   (70)
7990   (70)    7992   (70)    8032   (71)    8045   (71)    8048   (71)    8050   (71)    8060   (71)    8062   (71)
8068   (71)    8076   (71)    8077   (71)    8108   (72)    8110   (72)    8143   (73)    8145   (73)    8248   (76)
8303   (76)    8308   (76)    8310   (76)    8313   (76)    8314   (76)    8316   (76)    8317   (76)    8320   (76)
8321   (76)    8322   (76)    8325   (76)    8326   (76)    8327   (76)    8334   (76)    8337   (76)    8340   (76)
8343   (76)    8390   (77)    8454   (79)    8456   (79)    8503   (80)    8505   (80)    856    (3)     8560   (81)
8565   (81)    8573   (81)    8575   (81)    863    (4)     8630   (82)    8675   (83)    8679   (83)    8682   (83)
8683   (83)    8728   (84)    8732   (84)    8735   (84)    8736   (84)    8790   (85)    8791   (85)    8805   (85)
8806   (85)    8915   (87)    8924   (87)    8927   (87)    8928   (87)    8969   (87)    8970   (87)    8971   (87)
8979   (87)    8980   (87)    8981   (87)    8989   (87)    8990   (87)    8991   (87)    8998   (87)    9002   (87)
9005   (87)    9016   (87)    9020   (87)    9022   (87)    9024   (87)    9034   (87)    9040   (87)    9041   (87)
9044   (87)    9045   (87)    9047   (87)    9048   (87)    9051   (87)    9052   (87)    9053   (87)    9056   (87)
9061   (87)    9069   (87)    9073   (87)    9074   (87)    9075   (87)    9077   (87)    9079   (87)    9084   (87)
9086   (87)    9092   (87)    9093   (87)    9095   (87)    9096   (87)    9098   (87)    9144   (88)    9145   (88)
9148   (88)    9212   (89)    9213   (89)    9215   (89)    9218   (89)    9219   (89)    9220   (89)    9221   (89)
9228   (89)    9229   (89)    9235   (89)    9252   (89)    9273   (89)    9281   (89)    9282   (89)    9283   (89)
9285   (89)    9301   (89)    9302   (89)    9303   (89)    9307   (89)    9311   (89)    9316   (89)    9319   (89)
9335   (89)    9336   (89)    9342   (89)    9345   (89)    9346   (89)    9349   (89)    9350   (89)    9392   (90)
9405   (90)    9406   (90)    9412   (90)    9413   (90)    9416   (90)    9429   (90)    9430   (90)    9440   (90)
9441   (90)    9447   (90)    9448   (90)    9449   (90)    9481   (91)    9488   (91)    9584   (93)    9625   (94)
9626   (94)    9627   (94)    9630   (94)    9631   (94)    9632   (94)    9634   (94)    9635   (94)    9641   (94)
9644   (94)    9653   (94)    9657   (94)    9658   (94)    9660   (94)    9661   (94)    9663   (94)    9664   (94)
9666   (94)    9667   (94)    9710   (95)    9715   (95)    9720   (95)    9724   (95)    9725   (95)    9732   (95)
9733   (95)    9734   (95)    9738   (95)    9739   (95)    9740   (95)    9744   (95)    9745   (95)    9746   (95)
9750   (95)    9751   (95)    9752   (95)    9753   (95)    9754   (95)    9762   (95)    9763   (95)    9767   (95)
9769   (95)    9776   (95)    9777   (95)    9778   (95)    9779   (95)    9787   (95)    9788   (95)    9802   (95)
9803   (95)    9805   (95)    9806   (95)    9808   (95)    9809   (95)    9840   (96)    9846   (96)    9863   (97)
```

```
                    9865    (97)    9897    (98)    9903    (98)
.LONG               1767    (6)     1858    (6)     1859    (6)     1860    (6)     1861    (6)     1862    (6)     1863    (6)     2007    (6)
                    2008    (6)     2009    (6)     2010    (6)     2011    (6)     2012    (6)     2013    (6)     2014    (6)     2026    (6)
                    2027    (6)     2028    (6)     2029    (6)     2037    (6)     2038    (6)     2039    (6)     2040    (6)     2041    (6)
                    2042    (6)     2043    (6)     2044    (6)     2056    (6)     2057    (6)     2058    (6)     2059    (6)     2067    (6)
                    2068    (6)     2074    (6)     2075    (6)     2076    (6)     2077    (6)     2085    (6)     2086    (6)     2087    (6)
                    2088    (6)     2096    (6)     2097    (6)     2098    (6)     2099    (6)     2100    (6)     2101    (6)     2102    (6)
                    2103    (6)     2115    (6)     2116    (6)     2122    (6)     2127    (6)     2132    (6)     2133    (6)     2134    (6)
                    2135    (6)     2143    (6)     2144    (6)     2150    (6)     2151    (6)     2152    (6)     2153    (6)     2161    (6)
                    2162    (6)     2163    (6)     2164    (6)
.MACRO              432     (2)     464     (2)     490     (2)     518     (2)     549     (2)     580     (2)     610     (2)     656     (2)
                    707     (2)     730     (2)     754     (2)     767     (2)     786     (2)     818     (2)     843     (2)
.MCALL              856     (3)
.MEXIT              1393    (5)     2269    (7)     2270    (7)     2314    (7)     2361    (7)     2362    (7)     2378    (7)     2379    (7)
                    2380    (7)     2381    (7)     2568    (10)    2569    (10)    2570    (10)    2571    (10)    2572    (10)    2708    (12)
                    2710    (12)    2807    (14)    2893    (15)    2942    (16)    3207    (20)    3230    (20)    3252    (20)    3253    (20)
                    3260    (20)    3262    (20)    3267    (20)    3305    (20)    3307    (20)    3313    (20)    3323    (20)    3358    (20)
                    3369    (20)    3403    (20)    3521    (22)    3606    (23)    3627    (23)    3637    (23)    3650    (23)    3661    (23)
                    3679    (23)    3811    (24)    3812    (24)    3813    (24)    3817    (24)    3856    (24)    3862    (24)    3863    (24)
                    4009    (28)    4011    (28)    4040    (29)    4042    (29)    4122    (32)    4129    (32)    4137    (32)    4157    (33)
                    4209    (34)    4210    (34)    4217    (34)    4218    (34)    4219    (34)    4231    (34)    4241    (34)    4251    (34)
                    4254    (34)    4265    (34)    4272    (34)    4309    (35)    4311    (35)    4313    (35)    4334    (35)    4616    (39)
                    4652    (39)    4672    (39)    4685    (39)    4954    (41)    4957    (41)    4983    (41)    4987    (41)    4994    (41)
                    5004    (41)    5480    (47)    5575    (47)    5589    (47)    5591    (47)    5601    (47)    5617    (47)    5627    (47)
                    5639    (47)    5645    (47)    5827    (49)    5849    (49)    5857    (49)    6171    (53)    6428    (56)    6494    (56)
                    6503    (56)    7036    (63)    7050    (63)    7297    (65)    7320    (65)    7426    (66)    7490    (66)    7491    (66)
                    7507    (66)    7523    (66)    7529    (66)    7629    (67)    7680    (67)    7681    (67)    7682    (67)    7686    (67)
                    7687    (67)    7693    (67)    7726    (67)    7735    (67)    7736    (67)    7737    (67)    7738    (67)    7739    (67)
                    7749    (67)    7750    (67)    7756    (67)    7760    (67)    7771    (67)    7780    (67)    7781    (67)    7790    (67)
                    7791    (67)    7924    (70)    7933    (70)    8069    (71)    8247    (76)    8260    (76)    8261    (76)    8262    (76)
                    8267    (76)    8346    (76)    8442    (79)    8443    (79)    8452    (79)    8455    (79)    8460    (79)    8497    (80)
                    8557    (81)    8671    (83)    8724    (84)    8782    (85)    8783    (85)    8845    (86)    8916    (87)    8917    (87)
                    8930    (87)    8934    (87)    8935    (87)    8939    (87)    8945    (87)    8946    (87)    8947    (87)    8951    (87)
                    8952    (87)    8956    (87)    8957    (87)    8958    (87)    8973    (87)    8983    (87)    8993    (87)    9011    (87)
                    9042    (87)    9043    (87)    9049    (87)    9076    (87)    9081    (87)    9090    (87)    9091    (87)    9099    (87)
                    9139    (88)    9140    (88)    9150    (88)    9224    (89)    9249    (89)    9272    (89)    9337    (89)    9396    (90)
                    9431    (90)    9519    (92)    9628    (94)    9629    (94)    9633    (94)    9652    (94)    9655    (94)    9656    (94)
                    9665    (94)    9706    (95)    9726    (95)    9727    (95)    9728    (95)    9835    (96)    9836    (96)    9843    (96)
                    9845    (96)    9864    (97)    9867    (97)    9892    (98)    9893    (98)    9900    (98)    9902    (98)
.NCHR               1526    (6)     1527    (6)     1528    (6)     1529    (6)     1530    (6)     1531    (6)     1532    (6)     1533    (6)
                    1534    (6)     1535    (6)     1536    (6)     1537    (6)     1538    (6)     1539    (6)     1540    (6)
.NLIST              1375    (5)     1384    (5)     1386    (5)     1387    (5)     1388    (5)     1395    (5)     1396    (5)     1397    (5)
                    1400    (5)     1402    (5)     1404    (5)     1407    (5)     1408    (5)     1429    (5)     2320    (7)     2469    (8)
                    2630    (11)    2695    (12)    2703    (12)    2800    (14)    2817    (14)    2820    (14)    2825    (14)    2826    (14)
                    2827    (14)    2835    (14)    2841    (14)    2848    (14)    3043    (18)    3055    (18)    3058    (18)    3063    (18)
                    3103    (19)    3114    (19)    3117    (19)    3122    (19)    3213    (20)    3216    (20)    3224    (20)    3225    (20)
                    3229    (20)    3231    (20)    3236    (20)    3238    (20)    3242    (20)    3247    (20)    3248    (20)    3251    (20)
                    3257    (20)    3263    (20)    3271    (20)    3272    (20)    3278    (20)    3282    (20)    3284    (20)    3286    (20)
                    3287    (20)    3291    (20)    3296    (20)    3297    (20)    3298    (20)    3308    (20)    3317    (20)    3318    (20)
                    3326    (20)    3327    (20)    3333    (20)    3337    (20)    3339    (20)    3341    (20)    3342    (20)    3348    (20)
                    3353    (20)    3359    (20)    3367    (20)    3374    (20)    3375    (20)    3380    (20)    3394    (20)    3406    (20)
                    3412    (20)    3413    (20)    3414    (20)    3415    (20)    3423    (20)    3425    (20)    3462    (21)    3466    (21)
                    3467    (21)    3473    (21)    3475    (21)    3478    (21)    3479    (21)    3481    (21)    3482    (21)    3485    (21)
                    3525    (22)    3605    (23)    3607    (23)    3608    (23)    3611    (23)    3612    (23)    3618    (23)    3626    (23)
                    3631    (23)    3633    (23)    3641    (23)    3643    (23)    3644    (23)    3647    (23)    3648    (23)    3654    (23)
```

```
                  3655    (23)    3683    (23)    3684    (23)    3690    (23)    3691    (23)    3693    (23)    3694    (23)    3706    (23)
                  3707    (23)    3709    (23)    3710    (23)    3717    (23)    3720    (23)    3722    (23)    3723    (23)    3829    (24)
                  3836    (24)    3967    (27)    3985    (27)    4004    (28)    4005    (28)    4006    (28)    4007    (28)    4012    (28)
                  4013    (28)    4014    (28)    4015    (28)    4016    (28)    4035    (29)    4036    (29)    4037    (29)    4038    (29)
                  4043    (29)    4044    (29)    4045    (29)    4046    (29)    4047    (29)    4125    (32)    4126    (32)    4127    (32)
                  4130    (32)    4131    (32)    4132    (32)    4133    (32)    4134    (32)    4155    (33)    4158    (33)    418     (2)
                  4243    (34)    4248    (34)    4253    (34)    4281    (34)    4312    (35)    4355    (35)    4375    (35)    4377    (35)
                  4445    (36)    4449    (36)    4451    (36)    4461    (36)    4462    (36)    4469    (36)    4518    (37)    4526    (37)
                  4568    (38)    4571    (38)    4573    (38)    4619    (39)    4621    (39)    4623    (39)    4633    (39)    4649    (39)
                  4653    (39)    4655    (39)    4658    (39)    4660    (39)    4666    (39)    4676    (39)    4686    (39)    4691    (39)
                  4692    (39)    4696    (39)    4784    (40)    4842    (40)    4845    (40)    4858    (40)    4861    (40)    4879    (40)
                  4882    (40)    4990    (41)    4991    (41)    4992    (41)    4995    (41)    4996    (41)    5002    (41)    5003    (41)
                  5005    (41)    5006    (41)    5154    (45)    5169    (45)    5172    (45)    5173    (45)    5185    (45)    5210    (45)
                  5211    (45)    5214    (45)    5217    (45)    5226    (45)    5246    (45)    5247    (45)    5250    (45)    5283    (45)
                  5284    (45)    5288    (45)    5352    (46)    5378    (46)    5379    (46)    5383    (46)    5395    (46)    5412    (46)
                  5415    (46)    5416    (46)    5478    (47)    5484    (47)    5489    (47)    5524    (47)    5535    (47)    5576    (47)
                  5585    (47)    5592    (47)    5602    (47)    5618    (47)    5628    (47)    5800    (49)    5801    (49)    5804    (49)
                  5826    (49)    5833    (49)    5838    (49)    5839    (49)    5851    (49)    5854    (49)    5858    (49)    5920    (50)
                  5923    (50)    6175    (53)    6176    (53)    6215    (53)    6218    (53)    6268    (54)    6269    (54)    6270    (54)
                  6271    (54)    6279    (54)    6280    (54)    6281    (54)    6282    (54)    6285    (54)    6291    (54)    6292    (54)
                  6293    (54)    6295    (54)    6297    (54)    6299    (54)    6303    (54)    6304    (54)    6306    (54)    6307    (54)
                  6309    (54)    6319    (54)    6325    (54)    6330    (54)    6333    (54)    6344    (54)    6429    (56)    6438    (56)
                  6441    (56)    6442    (56)    6449    (56)    6450    (56)    6453    (56)    6454    (56)    6455    (56)    6456    (56)
                  6460    (56)    6464    (56)    6465    (56)    6478    (56)    6480    (56)    6493    (56)    6498    (56)    6499    (56)
                  6502    (56)    6507    (56)    6552    (57)    6555    (57)    6569    (57)    6577    (57)    6580    (57)    6581    (57)
                  6583    (57)    6655    (58)    6661    (58)    6687    (58)    6835    (61)    6838    (61)    6839    (61)    6848    (61)
                  6851    (61)    6855    (61)    6861    (61)    6862    (61)    6924    (62)    6927    (62)    7014    (63)    7017    (63)
                  7028    (63)    7031    (63)    7035    (63)    7042    (63)    7043    (63)    7049    (63)    7056    (63)    7057    (63)
                  7143    (64)    7147    (64)    7154    (64)    7164    (64)    7165    (64)    7171    (64)    7177    (64)    7197    (64)
                  7324    (65)    7743    (67)    7746    (67)    7747    (67)    7767    (67)    7768    (67)    7769    (67)    7776    (67)
                  7777    (67)    7778    (67)    7786    (67)    7787    (67)    7788    (67)    7861    (69)    7914    (70)    7917    (70)
                  7929    (70)    7930    (70)    7937    (70)    7941    (70)    7949    (70)    7957    (70)    7960    (70)    7961    (70)
                  7964    (70)    7977    (70)    7978    (70)    7981    (70)    7982    (70)    7985    (70)    7987    (70)    7989    (70)
                  7990    (70)    7992    (70)    8032    (71)    8045    (71)    8048    (71)    8050    (71)    8060    (71)    8062    (71)
                  8068    (71)    8076    (71)    8077    (71)    8108    (72)    8110    (72)    8143    (73)    8145    (73)    8248    (76)
                  8303    (76)    8308    (76)    8310    (76)    8313    (76)    8314    (76)    8316    (76)    8317    (76)    8320    (76)
                  8321    (76)    8322    (76)    8325    (76)    8326    (76)    8327    (76)    8334    (76)    8337    (76)    8340    (76)
                  8343    (76)    8390    (77)    8454    (79)    8456    (79)    8503    (80)    8505    (80)    856     (3)     8560    (81)
                  8565    (81)    8573    (81)    8575    (81)    863     (4)     8630    (82)    8675    (83)    8679    (83)    8682    (83)
                  8683    (83)    8728    (84)    8732    (84)    8735    (84)    8736    (84)    8790    (85)    8791    (85)    8805    (85)
                  8806    (85)    8915    (87)    8924    (87)    8927    (87)    8928    (87)    8969    (87)    8970    (87)    8971    (87)
                  8979    (87)    8980    (87)    8981    (87)    8989    (87)    8990    (87)    8991    (87)    8998    (87)    9002    (87)
                  9005    (87)    9016    (87)    9020    (87)    9022    (87)    9024    (87)    9034    (87)    9040    (87)    9041    (87)
                  9044    (87)    9045    (87)    9047    (87)    9048    (87)    9051    (87)    9052    (87)    9053    (87)    9056    (87)
                  9061    (87)    9069    (87)    9073    (87)    9074    (87)    9075    (87)    9077    (87)    9079    (87)    9084    (87)
                  9086    (87)    9092    (87)    9093    (87)    9095    (87)    9096    (87)    9098    (87)    9144    (88)    9145    (88)
                  9148    (88)    9212    (89)    9213    (89)    9215    (89)    9218    (89)    9219    (89)    9220    (89)    9221    (89)
                  9228    (89)    9229    (89)    9235    (89)    9252    (89)    9273    (89)    9281    (89)    9282    (89)    9283    (89)
                  9285    (89)    9301    (89)    9302    (89)    9303    (89)    9307    (89)    9311    (89)    9316    (89)    9319    (89)
                  9335    (89)    9336    (89)    9342    (89)    9345    (89)    9346    (89)    9349    (89)    9350    (89)    9392    (90)
                  9405    (90)    9406    (90)    9412    (90)    9413    (90)    9416    (90)    9429    (90)    9430    (90)    9440    (90)
                  9441    (90)    9447    (90)    9448    (90)    9449    (90)    9481    (91)    9488    (91)    9584    (93)    9625    (94)
                  9626    (94)    9627    (94)    9630    (94)    9631    (94)    9632    (94)    9634    (94)    9635    (94)    9641    (94)
                  9644    (94)    9653    (94)    9657    (94)    9658    (94)    9660    (94)    9661    (94)    9663    (94)    9664    (94)
                  9666    (94)    9667    (94)    9710    (95)    9715    (95)    9720    (95)    9724    (95)    9725    (95)    9732    (95)
```

```
               9733   (95)   9734   (95)   9738   (95)   9739   (95)   9740   (95)   9744   (95)   9745   (95)   9746   (95)
               9750   (95)   9751   (95)   9752   (95)   9753   (95)   9754   (95)   9762   (95)   9763   (95)   9767   (95)
               9769   (95)   9776   (95)   9777   (95)   9778   (95)   9779   (95)   9787   (95)   9788   (95)   9802   (95)
               9803   (95)   9805   (95)   9806   (95)   9808   (95)   9809   (95)   9840   (96)   9846   (96)   9863   (97)
               9865   (97)   9897   (98)   9903   (98)
.SBTTL         1434   (6)    2215   (7)    2402   (8)    2409   (8)    2481   (9)    2530   (10)   2576   (11)   2640   (12)
               2714   (13)   2757   (14)   2853   (15)   2910   (16)   2961   (17)   3011   (18)   3072   (19)   3131   (20)
               3429   (21)   3491   (22)   3540   (23)   3730   (24)   3867   (25)   3899   (26)   3922   (27)   3989   (28)
               4020   (29)   4051   (30)   4084   (31)   4108   (32)   4140   (33)   4164   (34)   4165   (34)   419    (2)
               4286   (35)   4397   (36)   4481   (37)   4540   (38)   4577   (39)   4708   (40)   4709   (40)   4902   (41)
               5021   (42)   5064   (43)   5082   (44)   5112   (45)   5307   (46)   5436   (47)   5665   (48)   5728   (49)
               5729   (49)   5863   (50)   5934   (51)   6080   (52)   6132   (53)   6224   (54)   6349   (55)   6377   (56)
               6515   (57)   6589   (58)   6692   (59)   6735   (60)   6794   (61)   6871   (62)   6956   (63)   7086   (64)
               7205   (65)   7342   (66)   7797   (68)   7835   (69)   7865   (70)   7998   (71)   8083   (72)   8115   (73)
               8150   (74)   8177   (75)   8204   (76)   8368   (77)   8395   (78)   8430   (79)   8468   (80)   8469   (80)
               8511   (81)   852    (3)    8589   (82)   863    (4)    8636   (83)   866    (5)    8689   (84)   8742   (85)
               8812   (86)   8855   (87)   9106   (88)   9171   (89)   9361   (90)   9462   (91)   9493   (92)   9551   (93)
               9603   (94)   9671   (95)   9813   (96)   9849   (97)   9870   (98)
.TITLE         2      (1)
.WORD          1373   (5)    1376   (5)    1377   (5)    1389   (5)    1403   (5)    1410   (5)    1415   (5)    1416   (5)
               1417   (5)    1419   (5)    1421   (5)    1424   (5)    1425   (5)    1430   (5)    1431   (5)    1459   (6)
               1464   (6)    1470   (6)    1471   (6)    1472   (6)    1473   (6)    1474   (6)    1475   (6)    1476   (6)
               1477   (6)    1478   (6)    1479   (6)    1480   (6)    1481   (6)    1482   (6)    1503   (6)    1505   (6)
               1511   (6)    1512   (6)    1513   (6)    1526   (6)    1527   (6)    1528   (6)    1529   (6)    1530   (6)
               1531   (6)    1532   (6)    1533   (6)    1534   (6)    1535   (6)    1536   (6)    1537   (6)    1538   (6)
               1539   (6)    1540   (6)    1552   (6)    1571   (6)    1579   (6)    1580   (6)    1606   (6)    1607   (6)
               1677   (6)    1696   (6)    1697   (6)    1698   (6)    1699   (6)    1700   (6)    1701   (6)    1702   (6)
               1703   (6)    1704   (6)    1705   (6)    1711   (6)    1781   (6)    1796   (6)    1798   (6)    1800   (6)
               1801   (6)    1804   (6)    1805   (6)    1806   (6)    1807   (6)    1808   (6)    1809   (6)    1810   (6)
               1811   (6)    1812   (6)    1813   (6)    1814   (6)    1815   (6)    1816   (6)    1817   (6)    1818   (6)
               1819   (6)    1820   (6)    1821   (6)    1822   (6)    1823   (6)    1824   (6)    1825   (6)    1826   (6)
               1827   (6)    1828   (6)    1829   (6)    1830   (6)    1831   (6)    1832   (6)    1837   (6)    1838   (6)
               1839   (6)    1840   (6)    1841   (6)    1843   (6)    1851   (6)    1852   (6)    1875   (6)    1876   (6)
               1878   (6)    1889   (6)    1890   (6)    1990   (6)    1991   (6)    1992   (6)    1993   (6)    1994   (6)
               1995   (6)    1996   (6)    1997   (6)    1998   (6)    1999   (6)    2000   (6)    2001   (6)    2002   (6)
               2003   (6)    2004   (6)    2264   (7)    2266   (7)    2267   (7)    2269   (7)    2270   (7)    2272   (7)
               2276   (7)    2279   (7)    2282   (7)    2284   (7)    2290   (7)    2296   (7)    2297   (7)    2301   (7)
               2302   (7)    2305   (7)    2310   (7)    2317   (7)    2319   (7)    2321   (7)    2325   (7)    2329   (7)
               2330   (7)    2331   (7)    2332   (7)    2333   (7)    2338   (7)    2340   (7)    2342   (7)    2343   (7)
               2344   (7)    2345   (7)    2349   (7)    2351   (7)    2357   (7)    2361   (7)    2362   (7)    2363   (7)
               2368   (7)    2369   (7)    2370   (7)    2374   (7)    2378   (7)    2379   (7)    2380   (7)    2381   (7)
               2382   (7)    2384   (7)    2386   (7)    2388   (7)    2389   (7)    2390   (7)    2391   (7)    2392   (7)
               2397   (7)    2398   (7)    2399   (7)    2463   (8)    2464   (8)    2465   (8)    2466   (8)    2470   (8)
               2471   (8)    2472   (8)    2473   (8)    2474   (8)    2475   (8)    2476   (8)    2477   (8)    2518   (9)
               2519   (9)    2520   (9)    2522   (9)    2523   (9)    2524   (9)    2525   (9)    2568   (10)   2569   (10)
               2570   (10)   2571   (10)   2572   (10)   2627   (11)   2629   (11)   2631   (11)   2634   (11)   2635   (11)
               2636   (11)   2697   (12)   2699   (12)   2702   (12)   2705   (12)   2707   (12)   2708   (12)   2709   (12)
               2710   (12)   2745   (13)   2747   (13)   2748   (13)   2749   (13)   2751   (13)   2753   (13)   2799   (14)
               2801   (14)   2802   (14)   2803   (14)   2804   (14)   2806   (14)   2837   (14)   2843   (14)   2847   (14)
               2849   (14)   2887   (15)   2889   (15)   2891   (15)   2895   (15)   2901   (15)   2902   (15)   2906   (15)
               2937   (16)   2939   (16)   2940   (16)   2944   (16)   2952   (16)   2953   (16)   2957   (16)   2997   (17)
               2998   (17)   2999   (17)   3001   (17)   3002   (17)   3003   (17)   3005   (17)   3006   (17)   3041   (18)
               3044   (18)   3045   (18)   3046   (18)   3047   (18)   3048   (18)   3051   (18)   3053   (18)   3054   (18)
               3056   (18)   3057   (18)   3062   (18)   3065   (18)   3068   (18)   3101   (19)   3104   (19)   3105   (19)
               3106   (19)   3107   (19)   3108   (19)   3110   (19)   3112   (19)   3113   (19)   3115   (19)   3116   (19)
```

```
          3121    (19)   3124    (19)   3127    (19)   3207    (20)   3215    (20)   3219    (20)   3226    (20)   3227    (20)
          3230    (20)   3237    (20)   3245    (20)   3250    (20)   3258    (20)   3260    (20)   3262    (20)   3274    (20)
          3278    (20)   3279    (20)   3281    (20)   3285    (20)   3294    (20)   3299    (20)   3305    (20)   3307    (20)
          3308    (20)   3313    (20)   3315    (20)   3316    (20)   3317    (20)   3329    (20)   3333    (20)   3340    (20)
          3352    (20)   3354    (20)   3360    (20)   3368    (20)   3369    (20)   3371    (20)   3376    (20)   3377    (20)
          3379    (20)   3381    (20)   3383    (20)   3385    (20)   3387    (20)   3391    (20)   3392    (20)   3396    (20)
          3402    (20)   3403    (20)   3405    (20)   3416    (20)   3424    (20)   3426    (20)   3463    (21)   3465    (21)
          3469    (21)   3474    (21)   3480    (21)   3484    (21)   3522    (22)   3527    (22)   3529    (22)   3531    (22)
          3535    (22)   3536    (22)   3604    (23)   3606    (23)   3609    (23)   3613    (23)   3636    (23)   3637    (23)
          3646    (23)   3649    (23)   3650    (23)   3657    (23)   3661    (23)   3665    (23)   3667    (23)   3673    (23)
          3677    (23)   3682    (23)   3687    (23)   3695    (23)   3701    (23)   3702    (23)   3724    (23)   3779    (24)
          3781    (24)   3785    (24)   3787    (24)   3791    (24)   3792    (24)   3797    (24)   3798    (24)   3800    (24)
          3803    (24)   3804    (24)   3806    (24)   3807    (24)   3808    (24)   3810    (24)   3811    (24)   3812    (24)
          3813    (24)   3814    (24)   3815    (24)   3816    (24)   3817    (24)   3821    (24)   3826    (24)   3828    (24)
          3830    (24)   3835    (24)   3837    (24)   3841    (24)   3843    (24)   3845    (24)   3850    (24)   3852    (24)
          3856    (24)   3858    (24)   3862    (24)   3863    (24)   3864    (24)   3890    (25)   3893    (25)   3895    (25)
          3915    (26)   3916    (26)   3917    (26)   3918    (26)   3953    (27)   3955    (27)   3959    (27)   3961    (27)
          3962    (27)   3970    (27)   3971    (27)   3975    (27)   3978    (27)   3979    (27)   3983    (27)   3984    (27)
          4009    (28)   4011    (28)   4012    (28)   4040    (29)   4042    (29)   4043    (29)   4068    (30)   4069    (30)
          4073    (30)   4074    (30)   4075    (30)   4078    (30)   4080    (30)   4094    (31)   4095    (31)   4098    (31)
          4099    (31)   4100    (31)   4103    (31)   4104    (31)   4120    (32)   4124    (32)   4128    (32)   4129    (32)
          4136    (32)   4137    (32)   4157    (33)   4161    (33)   4201    (34)   4204    (34)   4206    (34)   4207    (34)
          4208    (34)   4209    (34)   4210    (34)   4212    (34)   4213    (34)   4215    (34)   4216    (34)   4217    (34)
          4218    (34)   4219    (34)   4220    (34)   4226    (34)   4228    (34)   4229    (34)   4230    (34)   4231    (34)
          4232    (34)   4237    (34)   4259    (34)   4268    (34)   4271    (34)   4275    (34)   4280    (34)   4282    (34)
          4310    (35)   4317    (35)   4318    (35)   4322    (35)   4326    (35)   4327    (35)   4329    (35)   4330    (35)
          4335    (35)   4336    (35)   4340    (35)   4346    (35)   4347    (35)   4348    (35)   4351    (35)   4358    (35)
          4359    (35)   4360    (35)   4362    (35)   4363    (35)   4364    (35)   4368    (35)   4369    (35)   4376    (35)
          4378    (35)   4380    (35)   4434    (36)   4436    (36)   4448    (36)   4450    (36)   4452    (36)   4460    (36)
          4463    (36)   4516    (37)   4519    (37)   4525    (37)   4527    (37)   4532    (37)   4569    (38)   4572    (38)
          4630    (39)   4632    (39)   4635    (39)   4637    (39)   4642    (39)   4644    (39)   4662    (39)   4668    (39)
          4679    (39)   4687    (39)   4694    (39)   4695    (39)   4753    (40)   4757    (40)   4758    (40)   4762    (40)
          4763    (40)   4768    (40)   4770    (40)   4773    (40)   4774    (40)   4786    (40)   4791    (40)   4793    (40)
          4795    (40)   4797    (40)   4799    (40)   4801    (40)   4805    (40)   4807    (40)   4808    (40)   4813    (40)
          4815    (40)   4816    (40)   4821    (40)   4823    (40)   4824    (40)   4829    (40)   4831    (40)   4832    (40)
          4840    (40)   4844    (40)   4850    (40)   4854    (40)   4860    (40)   4870    (40)   4872    (40)   4873    (40)
          4881    (40)   4883    (40)   4894    (40)   4896    (40)   4950    (41)   4954    (41)   4955    (41)   4956    (41)
          4957    (41)   4958    (41)   4960    (41)   4961    (41)   4966    (41)   4970    (41)   4972    (41)   4974    (41)
          4975    (41)   4979    (41)   4981    (41)   4982    (41)   4983    (41)   4993    (41)   4994    (41)   4998    (41)
          5001    (41)   5004    (41)   5007    (41)   5056    (42)   5058    (42)   5079    (43)   5098    (44)   5101    (44)
          5102    (44)   5104    (44)   5105    (44)   5106    (44)   5156    (45)   5158    (45)   5160    (45)   5162    (45)
          5164    (45)   5171    (45)   5175    (45)   5176    (45)   5177    (45)   5178    (45)   5187    (45)   5188    (45)
          5189    (45)   5190    (45)   5192    (45)   5198    (45)   5203    (45)   5204    (45)   5207    (45)   5209    (45)
          5213    (45)   5216    (45)   5218    (45)   5228    (45)   5231    (45)   5234    (45)   5237    (45)   5242    (45)
          5245    (45)   5249    (45)   5252    (45)   5253    (45)   5257    (45)   5258    (45)   5264    (45)   5266    (45)
          5270    (45)   5272    (45)   5273    (45)   5279    (45)   5286    (45)   5287    (45)   5290    (45)   5291    (45)
          5354    (46)   5356    (46)   5358    (46)   5360    (46)   5365    (46)   5367    (46)   5368    (46)   5370    (46)
          5371    (46)   5377    (46)   5381    (46)   5382    (46)   5385    (46)   5386    (46)   5387    (46)   5397    (46)
          5398    (46)   5399    (46)   5403    (46)   5405    (46)   5406    (46)   5414    (46)   5418    (46)   5419    (46)
          5420    (46)   5480    (47)   5482    (47)   5483    (47)   5491    (47)   5493    (47)   5495    (47)   5497    (47)
          5499    (47)   5501    (47)   5503    (47)   5505    (47)   5507    (47)   5509    (47)   5511    (47)   5513    (47)
          5515    (47)   5516    (47)   5523    (47)   5526    (47)   5527    (47)   5534    (47)   5538    (47)   5539    (47)
          5545    (47)   5551    (47)   5557    (47)   5563    (47)   5569    (47)   5575    (47)   5576    (47)   5577    (47)
          5584    (47)   5587    (47)   5589    (47)   5591    (47)   5592    (47)   5593    (47)   5595    (47)   5601    (47)
          5602    (47)   5603    (47)   5605    (47)   5606    (47)   5608    (47)   5609    (47)   5611    (47)   5617    (47)
```

```
                5618    (47)    5619    (47)    5621    (47)    5627    (47)    5628    (47)    5629    (47)    5638    (47)    5640    (47)
                5650    (47)    5651    (47)    5705    (48)    5707    (48)    5712    (48)    5784    (49)    5786    (49)    5790    (49)
                5792    (49)    5797    (49)    5799    (49)    5803    (49)    5806    (49)    5808    (49)    5812    (49)    5813    (49)
                5816    (49)    5817    (49)    5818    (49)    5819    (49)    5823    (49)    5824    (49)    5830    (49)    5835    (49)
                5840    (49)    5844    (49)    5905    (50)    5906    (50)    5910    (50)    5911    (50)    5918    (50)    5919    (50)
                5922    (50)    5927    (50)    5929    (50)    5930    (50)    5996    (51)    5998    (51)    6003    (51)    6007    (51)
                6011    (51)    6013    (51)    6017    (51)    6019    (51)    6020    (51)    6022    (51)    6024    (51)    6025    (51)
                6026    (51)    6029    (51)    6030    (51)    6034    (51)    6036    (51)    6040    (51)    6042    (51)    6044    (51)
                6048    (51)    6050    (51)    6051    (51)    6052    (51)    6053    (51)    6056    (51)    6057    (51)    6061    (51)
                6064    (51)    6068    (51)    6069    (51)    6070    (51)    6071    (51)    6074    (51)    6075    (51)    6076    (51)
                6111    (52)    6116    (52)    6118    (52)    6120    (52)    6122    (52)    6123    (52)    6124    (52)    6128    (52)
                6169    (53)    6170    (53)    6178    (53)    6182    (53)    6184    (53)    6186    (53)    6190    (53)    6192    (53)
                6194    (53)    6195    (53)    6201    (53)    6206    (53)    6210    (53)    6211    (53)    6216    (53)    6219    (53)
                6263    (54)    6265    (54)    6266    (54)    6267    (54)    6277    (54)    6298    (54)    6315    (54)    6320    (54)
                6321    (54)    6326    (54)    6332    (54)    6335    (54)    6340    (54)    6341    (54)    6372    (55)    6373    (55)
                6421    (56)    6422    (56)    6427    (56)    6434    (56)    6436    (56)    6440    (56)    6452    (56)    6461    (56)
                6463    (56)    6470    (56)    6472    (56)    6482    (56)    6483    (56)    6501    (56)    6508    (56)    6544    (57)
                6545    (57)    6546    (57)    6547    (57)    6551    (57)    6554    (57)    6562    (57)    6564    (57)    6573    (57)
                6574    (57)    6579    (57)    6585    (57)    6624    (58)    6627    (58)    6629    (58)    6631    (58)    6633    (58)
                6635    (58)    6641    (58)    6642    (58)    6648    (58)    6649    (58)    6653    (58)    6654    (58)    6656    (58)
                6657    (58)    6664    (58)    6667    (58)    6671    (58)    6673    (58)    6674    (58)    6678    (58)    6680    (58)
                6684    (58)    6686    (58)    6688    (58)    6720    (59)    6724    (59)    6726    (59)    6730    (59)    6731    (59)
                6771    (60)    6773    (60)    6774    (60)    6775    (60)    6779    (60)    6781    (60)    6782    (60)    6784    (60)
                6785    (60)    6789    (60)    6790    (60)    6824    (61)    6825    (61)    6829    (61)    6830    (61)    6834    (61)
                6837    (61)    6843    (61)    6844    (61)    6845    (61)    6846    (61)    6847    (61)    6850    (61)    6856    (61)
                6864    (61)    6909    (62)    6910    (62)    6911    (62)    6915    (62)    6916    (62)    6918    (62)    6920    (62)
                6921    (62)    6922    (62)    6923    (62)    6926    (62)    6928    (62)    6929    (62)    6933    (62)    6934    (62)
                6936    (62)    6937    (62)    6938    (62)    6939    (62)    6943    (62)    6948    (62)    6951    (62)    6952    (62)
                6999    (63)    7000    (63)    7001    (63)    7005    (63)    7006    (63)    7008    (63)    7010    (63)    7011    (63)
                7012    (63)    7013    (63)    7016    (63)    7018    (63)    7019    (63)    7023    (63)    7024    (63)    7025    (63)
                7026    (63)    7027    (63)    7030    (63)    7037    (63)    7045    (63)    7051    (63)    7059    (63)    7063    (63)
                7064    (63)    7066    (63)    7067    (63)    7068    (63)    7069    (63)    7073    (63)    7078    (63)    7081    (63)
                7082    (63)    7125    (64)    7127    (64)    7132    (64)    7133    (64)    7134    (64)    7135    (64)    7139    (64)
                7140    (64)    7142    (64)    7148    (64)    7149    (64)    7151    (64)    7153    (64)    7155    (64)    7161    (64)
                7166    (64)    7175    (64)    7176    (64)    7179    (64)    7183    (64)    7186    (64)    7187    (64)    7192    (64)
                7196    (64)    7198    (64)    7201    (64)    7273    (65)    7275    (65)    7279    (65)    7281    (65)    7285    (65)
                7287    (65)    7291    (65)    7293    (65)    7298    (65)    7300    (65)    7304    (65)    7306    (65)    7310    (65)
                7313    (65)    7316    (65)    7321    (65)    7323    (65)    7325    (65)    7327    (65)    7331    (65)    7332    (65)
                7333    (65)    7335    (65)    7424    (66)    7425    (66)    7427    (66)    7428    (66)    7431    (66)    7432    (66)
                7434    (66)    7438    (66)    7440    (66)    7444    (66)    7446    (66)    7447    (66)    7450    (66)    7453    (66)
                7454    (66)    7456    (66)    7462    (66)    7463    (66)    7465    (66)    7466    (66)    7467    (66)    7473    (66)
                7474    (66)    7476    (66)    7477    (66)    7478    (66)    7480    (66)    7484    (66)    7486    (66)    7490    (66)
                7491    (66)    7492    (66)    7496    (66)    7498    (66)    7500    (66)    7501    (66)    7503    (66)    7506    (66)
                7507    (66)    7511    (66)    7513    (66)    7514    (66)    7517    (66)    7519    (66)    7523    (66)    7524    (66)
                7528    (66)    7529    (66)    7530    (66)    7535    (66)    7537    (66)    7538    (66)    7541    (66)    7542    (66)
                7544    (66)    7550    (66)    7552    (66)    7554    (66)    7555    (66)    7561    (66)    7562    (66)    7563    (66)
                7565    (66)    7569    (66)    7571    (66)    7572    (66)    7574    (66)    7575    (66)    7576    (66)    7577    (66)
                7578    (66)    7582    (66)    7585    (66)    7586    (66)    7587    (66)    7593    (67)    7595    (67)    7596    (67)
                7598    (67)    7603    (67)    7604    (67)    7606    (67)    7607    (67)    7609    (67)    7610    (67)    7614    (67)
                7615    (67)    7617    (67)    7621    (67)    7626    (67)    7628    (67)    7630    (67)    7634    (67)    7635    (67)
                7639    (67)    7641    (67)    7642    (67)    7643    (67)    7647    (67)    7648    (67)    7649    (67)    7651    (67)
                7652    (67)    7655    (67)    7657    (67)    7661    (67)    7663    (67)    7667    (67)    7669    (67)    7673    (67)
                7675    (67)    7680    (67)    7681    (67)    7682    (67)    7683    (67)    7685    (67)    7686    (67)    7687    (67)
                7688    (67)    7692    (67)    7693    (67)    7694    (67)    7695    (67)    7697    (67)    7700    (67)    7701    (67)
                7705    (67)    7707    (67)    7709    (67)    7710    (67)    7715    (67)    7717    (67)    7726    (67)    7727    (67)
                7728    (67)    7729    (67)    7734    (67)    7735    (67)    7736    (67)    7737    (67)    7738    (67)    7739    (67)
                7740    (67)    7742    (67)    7745    (67)    7748    (67)    7749    (67)    7750    (67)    7751    (67)    7755    (67)
                7756    (67)    7760    (67)    7762    (67)    7763    (67)    7765    (67)    7766    (67)    7770    (67)    7771    (67)
                7774    (67)    7775    (67)    7779    (67)    7780    (67)    7781    (67)    7784    (67)    7785    (67)    7789    (67)
                7790    (67)    7791    (67)    7792    (67)    7793    (67)    7823    (68)    7825    (68)    7827    (68)    7828    (68)
                7829    (68)    7830    (68)    7831    (68)    7858    (69)    7860    (69)    7906    (70)    7907    (70)    7908    (70)
                7909    (70)    7913    (70)    7916    (70)    7921    (70)    7923    (70)    7932    (70)    7938    (70)    7940    (70)
                7950    (70)    7953    (70)    7958    (70)    7959    (70)    7963    (70)    7970    (70)    7973    (70)    7979    (70)
                7980    (70)    7984    (70)    7991    (70)    7993    (70)    7994    (70)    8030    (71)    8031    (71)    8033    (71)
                8035    (71)    8039    (71)    8040    (71)    8041    (71)    8043    (71)    8044    (71)    8047    (71)    8051    (71)
                8057    (71)    8059    (71)    8063    (71)    8106    (72)    8107    (72)    8138    (73)    8139    (73)    8140    (73)
                8142    (73)    8171    (74)    8173    (74)    8198    (75)    8200    (75)    8241    (76)    8246    (76)    8251    (76)
                8252    (76)    8253    (76)    8256    (76)    8260    (76)    8261    (76)    8262    (76)    8263    (76)    8270    (76)
                8273    (76)    8277    (76)    8283    (76)    8288    (76)    8289    (76)    8290    (76)    8291    (76)    8292    (76)
                8293    (76)    8294    (76)    8296    (76)    8299    (76)    8304    (76)    8306    (76)    8307    (76)    8311    (76)
                8312    (76)    8318    (76)    8319    (76)    8324    (76)    8329    (76)    8335    (76)    8338    (76)    8341    (76)
                8344    (76)    8350    (76)    8352    (76)    8356    (76)    8357    (76)    8359    (76)    8362    (76)    8363    (76)
                8387    (77)    8389    (77)    8391    (77)    8416    (78)    8417    (78)    8419    (78)    8420    (78)    8421    (78)
                8423    (78)    8424    (78)    8425    (78)    8426    (78)    8439    (79)    8441    (79)    8442    (79)    8444    (79)
                8446    (79)    8451    (79)    8452    (79)    8453    (79)    8455    (79)    8457    (79)    8460    (79)    8462    (79)
                8464    (79)    8507    (80)    8553    (81)    8578    (81)    8581    (81)    8629    (82)    8631    (82)    8669    (83)
                8677    (83)    8678    (83)    8685    (83)    8722    (84)    8730    (84)    8731    (84)    8738    (84)    8786    (85)
                8789    (85)    8793    (85)    8804    (85)    8808    (85)    8809    (85)    8835    (86)    8837    (86)    8839    (86)
                8841    (86)    8845    (86)    8849    (86)    8851    (86)    8916    (87)    8917    (87)    8921    (87)    8923    (87)
                8926    (87)    8929    (87)    8930    (87)    8934    (87)    8935    (87)    8939    (87)    8940    (87)    8942    (87)
                8944    (87)    8945    (87)    8946    (87)    8947    (87)    8951    (87)    8952    (87)    8953    (87)    8955    (87)
                8956    (87)    8957    (87)    8959    (87)    8964    (87)    8965    (87)    8967    (87)    8968    (87)    8972    (87)
                8974    (87)    8977    (87)    8978    (87)    8982    (87)    8984    (87)    8987    (87)    8988    (87)    8992    (87)
                8994    (87)    9001    (87)    9004    (87)    9006    (87)    9007    (87)    9012    (87)    9017    (87)    9019    (87)
                9023    (87)    9028    (87)    9030    (87)    9035    (87)    9036    (87)    9039    (87)    9042    (87)    9043    (87)
                9046    (87)    9050    (87)    9055    (87)    9064    (87)    9072    (87)    9076    (87)    9089    (87)    9090    (87)
                9091    (87)    9097    (87)    9099    (87)    9138    (88)    9139    (88)    9140    (88)    9141    (88)    9143    (88)
                9147    (88)    9149    (88)    9150    (88)    9151    (88)    9152    (88)    9154    (88)    9155    (88)    9157    (88)
                9159    (88)    9165    (88)    9210    (89)    9211    (89)    9214    (89)    9216    (89)    9217    (89)    9223    (89)
                9231    (89)    9239    (89)    9244    (89)    9246    (89)    9248    (89)    9261    (89)    9263    (89)    9274    (89)
                9286    (89)    9289    (89)    9291    (89)    9293    (89)    9317    (89)    9324    (89)    9326    (89)    9328    (89)
                9330    (89)    9331    (89)    9348    (89)    9352    (89)    9353    (89)    9355    (89)    9356    (89)    9401    (90)
                9408    (90)    9423    (90)    9436    (90)    9443    (90)    9456    (90)    9485    (91)    9487    (91)    9523    (92)
                9524    (92)    9528    (92)    9537    (92)    9538    (92)    9540    (92)    9542    (92)    9571    (93)    9573    (93)
                9574    (93)    9576    (93)    9577    (93)    9579    (93)    9583    (93)    9585    (93)    9587    (93)    9588    (93)
                9589    (93)    9591    (93)    9595    (93)    9598    (93)    9628    (94)    9629    (94)    9633    (94)    9638    (94)
                9640    (94)    9642    (94)    9643    (94)    9654    (94)    9655    (94)    9656    (94)    9662    (94)    9665    (94)
                9770    (95)    9772    (95)    9781    (95)    9804    (95)    9810    (95)    9834    (96)    9841    (96)    9843    (96)
                9845    (96)    9846    (96)    9864    (97)    9866    (97)    9867    (97)    9891    (98)    9898    (98)    9900    (98)
                9902    (98)    9903    (98)


                                          +----------------------------+
                                          ! Performance indicators !
                                          +----------------------------+

Phase                     Page faults    CPU Time         Elapsed Time
-----                     -----------    --------         ------------
Initialization                     38    00:00:00.22      00:00:03.36
Command processing                866    00:00:00.83      00:00:02.37
Pass 1                           4745    00:08:31.90      00:13:44.38
```

```
Symbol table sort                  10      00:00:01.15     00:00:01.79
Pass 2                           1311      00:00:29.52     00:00:46.90
Symbol table output                 1      00:00:00.37     00:00:00.43
Psect synopsis output               3      00:00:00.02     00:00:00.06
Cross-reference output           1553      00:01:27.47     00:01:48.91
Assembler run totals             8529      00:10:31.48     00:16:28.23

The working set limit was 2048 pages.
688857 bytes (1346 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 636 non-local and 245 local symbols.
9914 source lines were read in Pass 1, producing 87 object records in Pass 2.
135 pages of virtual memory were used to define 99 macros.

                                                 +----------------------------+
                                                 ! Macro library statistics !
                                                 +----------------------------+

Macro library name                               Macros defined
------------------                               --------------
DISK$UCODPACK00:[VAX750]RDMMAC.MLB;10                         6
DISK$UCODPACK00:[VAX750]COMETMAC.MLB;1                        1
DISK$UCODPACK00:[VAX750]8085MAC.MLB;32                       77
SYS$COMMON:[SYSLIB]STARLET.MLB;2                              0
TOTALS (all libraries)                                       84

1105 GETS were required to define 84 macros.

There were no errors, warnings or information messages.

MAC/LIS=ECKAA.LIS/OBJ=ECKAA.OBJ/CROSS_REFERENCE=(ALL) ECKAA
```