PROCESSOR DIAGNOSTICS
EXTENDED OPERATING SYSTEM
3A PROCESSOR

CONTENTS	PAGE
1. GENERAL	2
2. DIAGNOSTIC PHILOSOPHY	3
A. Sequential Diagnostic Execution	3
B. Diagnostic Methods	4
C. CU Diagnostic Table-Driven Structure	4
D. Diagnostic Macro Language	5
3. DIAGNOSTIC IMPLEMENTATION	5
A. TLM Implementation	5
B. Requesting Diagnostics	5
C. Paging Diagnostics Into Memory	7
4. DIAGNOSTIC SYSTEM FUNCTIONAL OVERVIEW	7
A. General	7
B. CU Diagnostic Entry and Control	8
C. DMA, PCH, and CTI Diagnostic Entry and Control	8
5. DIAGNOSTIC TESTS	9
A. CU Diagnostic Tests	9
B. DMA Diagnostic Tests	14
C. PCH Diagnostic Tests	15
D. CTI Diagnostic Tests	16
6. GLOSSARY	16
Figures	
1. Table-Driven Element Interrelation	17
2. Table Entry Without Loops	17
3. Table Entry With Loops	18
4. Message Processing for Diagnostic Requests	
5. CU Diagnostic Entry and Execution	19
6. DMA, PCH, and CTI Diagnostic Entry and Execution	19
7. Control Unit Intercommunication	20
8. Detailed 3A Central Control Block Diagram	21
9. Subfield Matrix	23
10. Functional Diagram of the Main Store Controller	25
Tables	
A. Abbreviations and Acronyms	27
B. 3A Processor Diagnostic Program Listings	29
C. Diagnostic Macros	30
D. CU Diagnostic Test Sequence	33

CONTENTS	PAGE
E. Subfield Combinations	36
F. DMA Diagnostic Test Sequence . .	36
G. PCH Diagnostic Test Sequence . .	37
H. CTI Diagnostic Test Sequence . .	38

1. GENERAL

1.01 This section describes the 3A Processor diagnostics available with the Extended Operating System (EOS). These include the diagnostics for the control unit (CU), the direct memory access (DMA), the parallel channel (PCH), ➧and the collector diffusion isolation to transistor-transistor logic interface (CTI).➧ In this section, the CU is defined as the 3A central control (3A CC) and associated main store (MAS). Diagnostic tests assist in locating machine faults as well as ensuring operational integrity on a routine basis. ➧Table A contains the definitions of the abbreviations and acronyms used in this section.➧

1.02 ➧This section is reissued for the following reasons:

- To include the CTI diagnostic
- To include a table of 3A Processor diagnostic program listings
- To add tests 79 and 80 to the CU diagnostic tests
- To include the second page of Table D (Table B, Issue 1) which was left out
- To expand Table A (Table F, Issue 1.)➧

Change arrows have been used to denote significant changes.

1.03 The following sections contain background information pertaining to the 3A Processor diagnostics.

SECTION	TITLE
254-300-110	3A Central Control, Description, Common Systems
254-300-130	Input/OutputInterfaces,Description and Theory of Operation, Common Systems
254-340-040	Data Administration, Extended Operating System, 3A Processor
254-340-080	Maintenance Overview, Extended Operating System, 3A Processor
254-340-084	Resident Maintenance, Extended Operating System, 3A Processor
254-340-086	Initialization and Recovery, Extended Operating System, 3A Processor
254-340-090	Peripheral Diagnostics, Extended Operating System, 3A Processor
254-340-106	Glossary and Macros, Extended Operating System, 3A Processor

1.04 The following programs and tables provide EOS facilities used in implementing the 3A Processor diagnostics.

(a) The EOS diagnostic monitor program (AUTOMN), PR-4C626, provides the monitor and control functions for the DMA, PCH, and CTI diagnostics.

(b) The common diagnostic monitor program (CDGNM), PR-1C910, provides the monitor and control functions for the CU diagnostics.

(c) The operating system tables (OSTABS), PR-4C120, provide the various parameters which define the system hardware and software configuration.

(d) The return control to the interrupted routine program (PRCGWG), PR-4C122, is the portion of the dispatcher that returns control to an interrupted task.

(e) The common base level monitor (CBLM), PR-1C950, provides the CU diagnostics access to the multiscan function controller (MSFC).

(f) The common tape paging monitor (CPAGM), PR-1C955, pages the diagnostic programs from the catridge tape into the paging buffer.

(g) The maintenance subroutines program (MAISUB), PR-4C608, provides various maintenance support subroutines.

(h) The EOS maintenance task program (MAINT), PR-4C607, provides task control for the diagnostics.

1.05 ♦Table B contains a list of the program listings that comprise the 3A Processor diagnostics.♦

Diagnostic Functional Applications

1.06 EOS furnishes two types of diagnostics: processor diagnostics and peripheral diagnostics. Similar TTY messages activate both types of diagnostics. However, different diagnostic monitors are used to control the two types of diagnostics:

(a) The common diagnostic monitor (CDGNM) controls the 3A CC and memory diagnostics. These diagnostics always require the off-line CU. This section describes these diagnostics and how they are controlled by the common diagnostic monitor.

(b) The EOS diagnostic monitor (AUTOMN) controls two types of diagnostic programs. One type requires the use of the off-line CU in the same manner as the 3A CC and memory diagnostics. This category consists of the DMA, PCH, and CTI diagnostics. This section describes these diagnostics and how they are controlled by the EOS diagnostic monitor. The other type of diagnostic programs run on the on-line CU and do not require use of the off-line CU. The diagnostics for the common peripheral units (ie, those included as part of EOS facilities) and the application-supplied peripheral units fall into this category. Section 254-340-090 describes the common peripheral unit diagnostics and how they are controlled by the EOS diagnostic monitor.

1.07 The CU, DMA, PCH, and CTI diagnostic tests, together with the Trouble Locating Manual (TLM), provide the craft person a valuable aid in locating circuit faults within the 3A Processor. The tests are a means of using the on-line machine to test the off-line machine. The on-line machine administers the diagnostic tests to the off-line machine. Normally, the off-line machine executes the tests and the on-line machine interprets the results. Functioning in this way, the diagnostics can isolate most subsystem faults to a replaceable circuit pack. In some cases, a fault can be isolated only to a small group of circuit packs. In these cases, the tests can be executed in either the step or repeat mode to aid in further isolating the fault.

2. DIAGNOSTIC PHILOSOPHY

A. Sequential Diagnostic Execution

2.01 The 3A Processor diagnostics use the stop-on-first-failure technique of execution. This is also called the start-small technique. This technique consists of first testing a small amount of circuitry and then sequencing through additional tests. No test uses data from a previous test. However, circuitry that passes a test can be used in further tests in the sequence. If no test fails, the sequence continues until the unit is completely tested. Whenever a test fails, the sequence stops.

2.02 The stop-on-first-failure technique requires the tests to be performed in sequence to avoid erroneous failures. This is because each succeeding test in a sequence depends on the fact that all previous tests have passed. If a test is performed out of sequence, it is impossible to ensure that a failure detected by the test is actually located in the circuitry exercised by the test.

2.03 The 3A Processor diagnostic test sequences can be operated in one of several modes. First, a sequence can be ordered to stop and indicate the first failure detected. Second, a sequence can be ordered to continue after detecting a fault and to indicate each failure detected. In this mode, the first failure detected can generate succeeding failures. When multiple faults exist, the first fault must be cleared before further faults can be isolated. The first mode is used to isolate faults (single or multiple) since only known good circuitry can be used to determine that a fault really exists. Two additional modes of diagnostic execution provide for manual intervention. In one of these modes, a diagnostic request can be repeated by depressing the PF EXECUTE switch on the system status panel. Another mode causes a diagnostic request to run repetitively. In this mode, diagnostic execution can be stopped or started using the PF EXECUTE switch.

B. Diagnostic Methods

2.04 The CU, PCH, DMA, and CTI diagnostics execute with the off-line CU out of service. The off-line CU is used to diagnose all of these units including itself. The on-line CU provides control for the diagnostics and administers the diagnostics to the off-line CU.

2.05 The CU diagnostics are executed using three methods. Each method allows the off-line CU a different degree of responsibility in performing the diagnostic tests. The three diagnostic methods are:

- On-line execution
- On-line/off-line execution
- Off-line execution.

(a) ***On-Line Execution***—The on-line CU diagnoses the off-line CU at the microinstruction level via the maintenance channel (MCH). The output of the off-line microstore is inhibited, and microinstructions are loaded into the off-line microinstruction register by the on-line machine. The sequence of instructions provided by the on-line CU normally will test a particular function of the off-line CU and will cause the results to be returned for checking by the on-line machine.

(b) ***On-Line/Off-Line Execution***—The off-line machine executes short sequences of code under control of the on-line CU. The on-line CU loads a MAS address into the off-line store address register via the MCH and enables the off-line microstore. The off-line CU is then permitted to run a sequence of code, starting at the supplied address and terminating with a ***HALT*** operation. The on-line then checks the off-line via the MCH for proper execution of the code.

(c) ***Off-Line Execution***—The off-line CU is allowed to run code and check the results during the process. The on-line CU waits for the off-line to execute the code and then interrogates control words in the off-line store to determine test results. Hardware check circuits in the off-line CU are prohibited from stopping the machine. Failures are detected as incorrect results for an attempted operation.

2.06 Tests in a diagnostic sequence proceed in order from (a) (on-line execution) to (b) (on-line/off-line execution) to (c) (off-line execution) to ensure that no portion of the CU is relied on as a tool for diagnosis until it has been checked. For example, a CU cannot do bit testing until the data manipulation logic (DML) has been diagnosed.

2.07 The majority of the PCH, DMA, and CTI diagnostics are run on the off-line machine, and the results of the test are returned to the on-line machine for interpretation. However, a few of the DMA diagnostic tests are executed on the on-line machine.

C. CU Diagnostic Table-Driven Structure

2.08 Most CU diagnostics are designed to use a table-driven technique of execution. If the diagnostic is executing in the on-line mode with the on-line machine sending individual microinstructions to the off-line machine, the total diagnostics are table-driven. In the on-line/off-line and off-line modes where the on-line machine instructs the off-line machine to execute code, the on-line portion of the test is table-driven.

2.09 There are three elements in a table-driven diagnostic:

- Data tables
- Task monitor
- Interpretive routines.

Figure 1 depicts the interrelationship of these elements. The actual test is represented by various data tables. These tables define the functions to be performed by the test. Each data table contains a 6-bit op-code that specifies an associated interpretive routine. The task monitor sequences through the table entries. For each table entry, the task monitor examines the op-code and passes control to the designated interpretive routine. The interpretive routine translates the table entry and executes that portion of the diagnostic test specified by the table. When the interpretive routine has finished, it returns control to the task monitor which advances to the next table entry in the test.

The task monitor and the interpretive routines are part of the CDGNM.

2.10 There are approximately 40 types of table entries, each corresponding to a high level operation. Each table entry consists of a control word and up to three data words. An exception to this is a special form of table entry that allows for a looping feature. A table entry with the looping feature can have up to three data words for each loop. Consequently, there are two basic forms for table entries (Fig. 2 and 3). Either form can be used within a loop. However, the form illustrated in Fig. 3 must be used when it is desired to have different data entries associated with each pass of a loop.

D. Diagnostic Macro Language

2.11 The data tables for table-driven diagnostics are formed using a high level diagnostic language. This language consists entirely of macros. Therefore, the program listing (PR) for a test program written in the diagnostic macro language will bear little resemblance to the PR for a conventional 3A language or Electronic Switching System programming language (EPL) program. Each macro in the diagnostic language generates a data table that represents the high level operation associated with the macro. The 6-bit op-code in the table entry corresponds to this operation and points to the interpretive routine that executes the operation.

2.12 The diagnostic macro language is used primarily by the diagnostic test programmer. However, a knowledge of the basic functions of the macros may be useful in analyzing the CU diagnostic test programs. The diagnostic macros can be divided into the following groups:

- Macros that operate on the on-line CU
- Macros that operate on the off-line CU
- Macros for interrogation of results
- Macros for generation of the TLM
- Macros for controlling the looping feature.

The diagnostic macros and their functions are listed in Table C.

2.13 The macros that control the looping feature are used to establish the boundaries of the loop, ie, the beginning of the loop and the end of the loop. Macros within the loop boundaries may contain multiple subarguments. These subarguments are used to generate looping tables with different data words for each loop pass. The following macros may contain multiple subarguments:

- ADD_OFF
- COMPARE
- FAILTST
- LOAD_MAR
- LOAD_MIRH
- MASK
- MICRO
- NO3_CODE
- SEND
- TTY_DATA.

3. DIAGNOSTIC IMPLEMENTATION

A. TLM Implementation

3.01 When a diagnostic detects a fault, a TTY diagnostic message is generated that contains a trouble number. This trouble number refers to an entry in the TLM that gives details concerning the failure. The failing circuit pack numbers for a given trouble number are listed in order of probability of causing the fault associated with the trouble number.

B. Requesting Diagnostics

3.02 The 3A Processor diagnostics can be initiated automatically or manually via the maintenance TTY using Electronic Switching System (ESS) type commands. The ESS type commands have a very rigid basic format that consists of an action verb, an identification field, a data field, and an execute character (!). This basic format is further defined into two standard TTY input message formats for requesting the processor diagnostics.

3.03 The following general format must be used when requesting CU diagnostics:

DGN:CU;***Act-opt:Test,bbb!***

DGN is the action verb of the input command stream and is required for all diagnostic requests. CU specifies that CU diagnostic requests are to be run. DGN:CU! alone causes all CU diagnostic tests to be run in sequence. If a test fails, the sequence stops and a message is generated. If all tests pass (ATP), an ATP message is output. ***Test*** is a decimal number that can be used to specify a particular test to be run. An ATP or failure message is output upon completion of the test. The following ***Act-opts*** (action options) can be specified:

(a) UCL—The unconditional option prevents termination of a diagnostic test sequence when a failure occurs. Each test that fails will generate a failure message and the test sequence will continue. This option should be used with caution since diagnostic terminations, which assure system integrity when testing a faulty unit, are eliminated.

(b) STEP—This option causes a specified test to run one time and a message to be output upon completion. The same test can be run again by depressing the PF EXECUTE switch on the system status panel. After the initial run, a message is generated only if the test result changes. The status of the results is indicated by the pass/fail lamps on the system status panel. If ***Test*** is not specified, all tests are run sequentially.

(c) RPT—The repeat option causes a specified test to run repetitively until aborted. A message is generated at the end of the first run. Thereafter, a printout is generated only if the test result changes. The status of the results is indicated by the pass/fail lamps on the system status panel. The test cycle can be stopped by depressing the PF EXECUTE switch on the system status panel. The cycle can be restarted by depressing the PF EXECUTE switch again. If ***Test*** is not specified, all CU diagnostic tests will be run repetitively.

(d) ADD—This option makes it possible for growth memory to be diagnosed before it is defined to the system. All of normal store plus recently added store is diagnosed. The additional store is defined by the ***bbb*** field. This field must contain the highest address of store to be diagnosed. The ***bbb*** field should not be used if the ADD option is not selected. The ADD option can be used along with either of the other action options provided they are input separated by a comma. ***Test*** must always be specified in conjunction with the ADD option. Therefore, if the entire CU test sequence is to be run, a zero must be specified for ***Test.***

3.04 The following general format is used when requesting PCH, DMA, or CTI diagnostics.

DGN:***Dev(N);Act-opt:Test!***

Dev must be either PCH, DMA, or CTI. ***N*** is the equipment number which designates the particular device to be tested. The default value for ***N*** is unit number zero. ***Act-opt*** and ***Test*** have the same meanings as already defined for the CU diagnostics in paragraph 3.03. However, exceptions are that ***Test*** must be specified when the RPT (repeat) option is selected and the ADD option is not applicable to PCH, DMA, or CTI diagnostics. Also, AUTOMN provides for two additional action options. LOOPT or LOOPF will loop on a selected diagnostic until halted.

3.05 The system will respond with at least one TTY message after a diagnostic request is entered. The diagnostic monitor in control determines what message is returned. For CU diagnostics which are controlled by the CDGNM, the response will be one of the following messages:

(a) PF

The requested diagnostic test will be run. A pass or fail message will follow.

(b) ?D

Invalid diagnostic test number is specified.

(c) ?I

Invalid action option is specified.

(d) RL

Repeat request later. Another function of equal or higher priority is presently being performed.

(e) NG

No good. The system is not in a valid state for CU diagnostics.

AUTOMN, which controls PCH, DMA channel, and CTI diagnostics, may issue two responses, depending on the validity of the diagnostic request. AUTOMN will always output one of the following messages:

(a) PF

In this case, PF means that the diagnostic request is associated with a valid device.

(b) NG

The device specified is not valid.

(c) RL

Repeat request later.

Even though AUTOMN responds with PF, further processing may find the diagnostic request to be erroneous. When this occurs, AUTOMN will respond with an additional TTY message. This message will be one of the following:

(a) KEYWORD NG

The keyword for the requested device could not be found in the diagnostic transfer vector (DGTAB).

(b) TEST NUMBER NG

The requested test number could not be found in DGTAB.

(c) EQUIPMENT NUMBER NG

The unit number specified for the device to be diagnosed does not match the one assigned to the device by the system.

C. Paging Diagnostics Into Memory

3.06 Almost all 3A Processor diagnostics are stored on the tape cartridge and read into MAS when needed. Most of the CDGNM is also stored on tape except the CDGNM2 and CDGNMPAT portions. The AUTOMN, however, is resident except for CSECT DGNRUN.

3.07 An area in MAS referred to as the paging buffer is reserved for bringing in nonresident programs. The tape cartridge containing the nonresident programs is divided into segments, and each segment is assigned a number. The segment numbers are used as identifiers for paging the programs. Paging is accomplished using the common tape paging monitor (CPAGM).

3.08 When a request for CU diagnostics is entered, the CDGNM is paged and remains for the duration of the test. Each CU test is paged when needed. Therefore, the monitor and the diagnostic being executed coexist in the paging buffer with the monitor occupying the low 1300 locations.

4. DIAGNOSTIC SYSTEM FUNCTIONAL OVERVIEW

A. General

4.01 The 3A Processor diagnostics consist of two groups of diagnostics. The two groups of diagnostics differ in the way they are initiated and in the way they are controlled during execution. One group is the CU diagnostics. The control facilities for these diagnostics include the CDGNM and the multiscan function controller (MSFC) portion of the common base level monitor (CBLM). The other group consists of the DMA, PCH, and CTI diagnostics. These are controlled by AUTOMN.

4.02 The two groups of diagnostics are requested through similar TTY messages (see Part 3). However, EOS uses a different method of entry into each group of diagnostics. CU diagnostics are activated through a function of the EOS maintenance task (MAINT). The DMA, PCH, and CTI diagnostics are activated via event 3.

4.03 Figure 4 illustrates the system level flow in processing input messages for diagnostic requests. This stage of diagnostic initiation is the same for both groups of diagnostics. The data administration package receives all input messages from TTY devices and distributes them to the proper clients. The data administrator passes ESS-type commands to the client interface program (EOSMSG) for parsing and client identification. The client is determined from TTYTBL which contains a description of all commands in the system. ESS commands are then passed to MAINT for distribution to the clients. The client program for CU diagnostic requests is the input message handler (DGNCU) portion of CDGNM; for DMA, PCH, and CTI diagnostic requests, it is event routine 3 (DCT_TTYIN) of AUTOMN.

B. CU Diagnostic Entry and Control

4.04 The process that controls CU diagnostic entry and execution can be divided into two phases. Figure 5 contains a block diagram depicting the system level flow for these two phases. The first phase involves only initiating the diagnostics. When MAINT receives a message containing a request for CU diagnostics, it calls the input message handler in CDGNM (entry point DGNCU). One of the functions of DGNCU is to check the input message for validity. If the request is valid, the multiscan function (MSF) subroutine MSFREQ is called to set up the diagnostic request in the MSF matrix (MSFMTX). DGNCU also sets up control word DGREQ with the type of request. When DGNCU is complete, it returns to MAINT with a return code that corresponds to a 2-character response message to be output on the TTY. This message indicates the disposition of the diagnostic request (see paragraph 3.05).

4.05 The second phase in CU diagnostic initiation is controlled by MAINT. This program is scheduled periodically at a rate determined by the application program. The rate is set through the OSTABS parameter MAIRATE. MAINT begins executing at the high priority level MAIPRIOR. When the high priority level functions have been completed, MAINT dynamically lowers its priority to the medium level MAIPRIM. CU diagnostics execute as a medium priority function of MAINT. Therefore, while CU diagnostics are executing, lower priority levels are blocked. MAINT is designed to execute CU diagnostics from start to finish with no real-time breaks. However, effective real-time breaks will occur if the diagnostics are still executing when the cycle-time defined by MAIRATE has elapsed. When this occurs, MAINT returns to its high priority level, and CU diagnostics are suspended until MAINT returns to its medium priority level.

4.06 One of the functions MAINT performs at priority MAIPRIM is to call MSFC. If a request for CU diagnostics exists in the MSF matrix, MSFC will pass control to the diagnostic test controller in CDGNM (entry point ENTRDG). The test controller examines the control word DGREQ to determine the type of request. DGREQ was set up by the input message handler of CDGNM at the time the diagnostic request was input (see 4.04). The subroutine EXPAGREQ in the paging monitor is called to page in the diagnostic if it is not resident. The test controller then branches to the diagnostic table interpreter in CDGNM to process the diagnostic (entry point INTRPNO3). When the test is complete, the table interpreter will return to the test controller. The test controller now determines the work to be done next, based on such things as the pass/fail results of the previous test and the type of diagnostic request. When the test controller has finished, it returns to CBLM with a return code that indicates if there are more tests to be run, eg, if either the step or repeat mode or the sequence of tests was requested. Normally, when a diagnostic message is to be outputted, it is formed by CDGNM and printed by CBLM. The return code also informs CBLM when a message is to be outputted. After CBLM has completed processing the return from CDGNM, it returns control to MAINT. MAINT checks to determine if the cycle-time defined by MAIRATE has expired. If so, MAINT returns to the MAPRIOR priority level, and any remaining diagnostics are suspended until MAINT returns to the MAIPRIM priority functions. If the cycle-time has not expired and CU diagnostics are still active, MAINT returns to CBLM to continue diagnostic processing.

C. DMA, PCH, and CTI Diagnostic Entry and Control

4.07 DMA, PCH, and CTI diagnostics run under control of AUTOMN. Portions of AUTOMN are also common to other types of diagnostics. This commonality is described in Section 254-340-090. AUTOMN runs as a low priority task (DGCTSK) under EOS. After initialization, DGCTSK is placed in a "wait" state until activated by either a manual diagnostic request from the maintenance TTY or by an automatic request through an intertask message sent by another EOS task.

4.08 Figure 6 contains a block diagram of the system level flow of DMA, PCH, and CTI diagnostic entry and execution. When MAINT receives a message containing a request for DMA, PCH, or CTI diagnostics, it sends the message to

AUTOMN via event 3. The basic functional sequence of AUTOMN from this point is as follows:

- Determine if the request is valid and return the appropriate 2-character response (see paragraph 3.05)
- Build the diagnostic control buffer (DGN_CB)
- Identify the device for which the diagnostic is requested
- Identify and locate the proper diagnostic program
- Determine if the diagnostic is resident or nonresident
- Initiate the paging function if the diagnostic is nonresident
- Transfer control to the diagnostic
- Interpret returns from the diagnostic and generate output messages.

Each of the above is described in detail in Section 254-340-090.

5. DIAGNOSTIC TESTS

A. CU Diagnostic Tests

5.01 CU diagnostics for the 3A Processor consist of the 80 tests listed in Table D. The total size of these tests is equivalent to approximately 40,000 words of memory. The diagnostics will detect actual memory size and width and bypass any unnecessary tests based on translation parameters. The following list summarizes some of the more important details concerning the CU diagnostics.

(a) CU diagnostics execute at medium priority (see paragraph 4.05). Therefore, lower priority levels are blocked while CU diagnostics are executing.

(b) CU diagnostics can be requested manually via the TTY or automatically by application software.

(c) CU diagnostics destroy the contents of the off-line store. A CU initialization that occurs while CU diagnostics are running will usually cause a high-level initialization.

(d) CU diagnostics are quite reliable in finding CU problems but may not always locate transient memory problems.

(e) DMA, PCH, and CTI diagnostics must be run in conjunction with the CU diagnostics to validate the entire 3A Processor.

5.02 The following is a brief description of the functions performed by each CU diagnostic. A more detailed description of each test is located at the beginning of the test in the associated PR. The PRs are identified in Tables B and D.

5.03 ***Test 1:*** The primary communication link between controllers is the MCH. Access to the other main store is also provided for each CU. The control unit intercommunication links are shown in Fig. 7. Initial tests check the maintenance channel buffer (20 data bits plus 2 parity bits), maintenance channel transmit/receive registers, and maintenance channel control buffer (MCHB) of the on-line machine. Data patterns are sent to the off-line MCH and then returned to the on-line machine for validation. Maintenance channel commands are then executed to ensure proper operation of the maintenance channel link.

5.04 ***Test 2:*** This test attempts to gate data on the off-line gating bus into the on-line machine for checking. The detailed 3A central control block diagram (Fig. 8) shows this gating bus and should be referenced as required for the following test descriptions. All data and parity bits on the off-line gating bus should be zero since the machine is operating under control of the on-line processor. Any deviation from an all-zero pattern indicates a ***stuck high*** bit error.

5.05 ***Test 3:*** Several checks are made to ensure operation of the system clock. Checks are made to determine that all four clock phases are present. An attempt is made to toggle the clock via the MCH with the oscillator blocked. Hardware check circuits, stop clock functions, and automatic clock restart functions are also verified.

5.06 ***Test 4:*** The off-line CU is initialized to a known state before each diagnostic test is performed. This initialization consists primarily of clearing or setting various registers to known values. Test 4 verifies that the attempted off-line processor initialization was successful.

5.07 ***Test 5:*** The TO field crosspoints are selected by a 4-out-of-8 bit decode which provides for a maximum of 70 crosspoint selections. The left four bits are combined to form an n/4 subfield, and the right four bits are combined in a similar manner. The two subfields are derived on the decoder board and checked by the decoder check circuits to ensure that only one active line exists in each subfield. The left and right subfields are logically ***ANDed*** on the board or boards where the crosspoint is required. The combinations available are shown in Table E.

5.08 The multiple combinations (1/4, 2/4, and 3/4) can be represented by matrices. An example of this representation is shown in Fig. 9. If a row subfield lead is shorted to ground (a logically active line is low), multiple crosspoints are ***fired*** when a crosspoint in any other row is selected. Test 5 attempts to prevent this by isolating the stuck row and by sequentially selecting all crosspoints in a given column. A decoder error will exist (two left subfield lines active) on all selections until the crosspoint on the stuck row is selected. Then only one row will show active, identifying the coordinates of the error. Stuck columns are identified by firing crosspoints in a given row. Any error pattern other than one correct crosspoint firing in the sequence indicates problems of a different nature.

5.09 ***Test 6:*** The FROM field crosspoints are derived in the same manner as the TO field crosspoints. Test 6 checks for shorted ground FROM subfield leads, utilizing the same techniques developed for test 5.

5.10 ***Test 7:*** Final FROM crosspoint decoding is generally done on the board or boards where the decoder is used. The subfield signals are inverted and then logically ANDed to select the crosspoint. If the input to an AND gate is open, multiple crosspoints would be fired. The decode error circuitry would not detect this type fault since it is observing the subfield leads. These errors are detected by setting ones in the registers to be tested, including parity high (PH) and parity low (PL). A series of left FROM subfields are fired with the right subfield zero. Then a series of right FROM subfields are fired with the left subfield zero. The TO field is a register not being tested that was initially cleared to all zeros. No gating onto the bus should occur; but if it does, some bit or bits will be set to one if a fault exists.

5.11 ***Test 8:*** Test 8 checks for stuck high inputs to the final TO field decode gates. The same techniques utilized in test 7 are applied to this test.

5.12 ***Test 9:*** The general registers are checked by gating ones and zeros into and out of each register. The results are verified in the on-line machine.

5.13 ***Test 10:*** Special purpose registers are checked for the capability of properly gating all zeros or all ones into or out of each register.

5.14 ***Test 11:*** Contents of the entire microstore are read and verified. Verification is accomplished by calculating a checksum value for each 1024 words and comparing this value with the correct checksum value contained in the Base Level Maintenance Monitor Applications (BLMMA). Also, some 4/8 patterns and parity of the microstore words are checked.

5.15 ***Test 12:*** The function registers (FRs) are duplicated (FR0 and FR1) and used for control operation of the data manipulation logic (DML). Contents of the FR cannot be accessed directly and must be verified by examination of the DML output. The DML and gating into the FR have not yet been diagnosed; therefore, a series of logic functions is initiated to ensure that each bit of the FR can be set and reset.

5.16 ***Test 13:*** The duplicated adders are tested via the maintenance channel for long-add (20 bits), short-add (16 bits), and add-one functions.

5.17 ***Test 14:*** The comparator is checked, using the add operation for generating compare data. The A register (AR) and B register (BR) are also checked for gating capability to the matcher.

5.18 ***Test 15:*** The 16 available Boolean functions, utilizing two variables, are executed and verified in this test.

5.19 ***Test 16:*** The find low zero (FLZ) is a command to locate the least significant zero in a data word. Various data words are set up to verify proper operation of the FLZ test.

5.20 ***Test 17:*** The rotate function is tested by rotating a one through a field of zeros and verifying the location at known points. A zero is then rotated through a field of ones and checked in like manner.

5.21 ***Test 18:*** Various pack and unpack commands are executed in the data manipulation, using crosspoints involving the A- and B-registers. These checks ensure that 20-bit words can be generated using selected bits of smaller words (packing) and 16-bit words generated from selected bits of larger words (unpacking). The integrity of the DML parity generator is also checked. Eight operands are used to verify correct operation.

5.22 ***Test 19:*** Various microcontrol functions are tested at this point. The match circuit, which compares the microaddress register (MAR) and return address register (RAR), is checked. A test is initiated to check the return address update (RU) flip-flop. When RU equals 1, the RAR duplicates the contents of the MAR. When RU equals 0, the RAR is used to save a return address for use upon completion of a subroutine. The program gating bus (PGB) all-zeros checker is tested in conjunction with the data manipulation logic status flip-flop (DS). The 1-out-of-16 decoders on the X- and Y-fields of the instruction buffer (IB) and associated parity generator are tested.

5.23 ***Test 20:*** The second part of the microcontrol test ensures valid operation of the following.

- All-ones detector, which denotes a return from a microprogram subroutine with the presence of all ones in the next address field.

- The gating path from the RAR to the MAR.

- Special crosspoints, which cause outputs of the 3-out-of-6 decoders of the instruction buffer to be ***ORed*** into the low six bits of the TO and FROM field in the microinstruction register.

- The control A (CA) and control B (CB) bits in the microinstruction register (MIR) control several functions. Two of these (fetch and auxiliary control) are tested functionally.

- The parity generator is checked.

5.24 ***Test 21:*** Microcontrol test, Part 3, checks for proper gating between various registers associated with the movement of instructions.

5.25 ***Test 22:*** Microcontrol test, Part 4, checks the circuitry associated with the microinstructions.

5.26 ***Test 23:*** The DS flip-flop is set or cleared by various functions. These functions are initiated, and the DS flip-flop is checked for proper state.

5.27 ***Test 24:*** The adder circuitry for program address (PA) plus 1 and the gating bus from PA plus 1 to store address register (SAR) are tested in all bits and will carry from the next lowest bit.

5.28 ***Tests 25 and 26:*** The store bus controller test ensures that associated registers function properly and that various control signals can be set and cleared. The error return address register (ERAR) and error return address update (ERU) flip-flop used in complement correction are also tested.

5.29 ***Test 27:*** This is the first of a series of tests to validate operation of the off-line MAS. Tests are initiated to ensure that orders can be transmitted and properly executed via the input/output (I/O) subchannel and out-of-service store controller.

5.30 ***Test 28:*** The 4-out-of-8 store maintenance orders are sent to the off-line MAS controller (MASC) to ascertain that they can be transmitted and decoded properly. Figure 10 is a functional diagram of the main store controller.

5.31 ***Tests 29 and 30:*** Tests 29 and 30 contain checks of the MASC multiplexor circuitry. The multiplexing function refers to the memory modes of normal read/write versus the refresh (of volatile memory) cycle. These tests ensure the capability of the memory to operate in the required modes and ensure that associated control and timing signals function properly.

5.32 ***Tests 31 and 32:*** These tests validate correct operation of the MAS multiplexor error detection and check circuitry.

5.33 ***Test 33:*** Tests 33 through 36 are used to ensure integrity of the MAS bus. Test 33 tests the command portion of the bus. Legitimate 2-out-of-4 codes are sent with no error signals expected. Invalid commands are used to determine that an error signal is generated.

5.34 ***Test 34:*** The address portion of the MAS bus is now checked. Addresses are transmitted over the MAS bus and then retrieved for validation. Check circuits are also validated.

5.35 ***Tests 35 and 36:*** Tests 35 and 36 check the data portion of the MAS bus. Data transmission loop-around tests, as well as error and control signal checks, ensure proper operation.

5.36 ***Tests 37 Through 40:*** These tests check the integrity of MAS control and parity circuits. Data patterns are transferred to initiate the various control and parity functions in the MAS. Results are verified for correct operation.

5.37 ***Tests 41 Through 48:*** The memory cells in a memory module are located on nine memory boards. The signal distribution link between the main store controller and these boards is via a fanout board. Tests 41 through 48 check fanout board signals as follows:

(a) These tests check both the refresh select circuits on the fanout boards and the check circuitry for the refresh select signals. One-eighth of all the memory modules are refreshed at the same time. The refresh select signals are used to indicate to the MASC that the memory modules have been refreshed properly. The check circuitry in the MASC is diagnosed by verifying that none of the refresh signals from the equipped fanout boards are stuck active or inactive.

(b) These tests check the normal select circuits on the fanout boards and the check circuitry for the normal selects. Each mode is selected one at a time. The check circuitry verifies that only one of the modules respond. This is done by the check circuitry in the controller receiving a module select signal back from each selected module in turn. The exercise of the normal select check circuit is completed by applying error conditions and checking for the proper error output.

(c) These tests verify that the address signals can be transmitted from the bit-sliced boards in the MASC to the fanout boards in the memory modules.

(d) The address parity checkers on the fanout boards of all the memory modules are tested.

(e) The tests check that the timing signals are passed from the MASC to the memory plane via the fanout boards. The timing signals are looped back from the fanout board to the controller. The MASC then checks to verify that the right timing signals are transmitted. The timing signals are prevented from being transmitted to the fanout board. This verifies that the check circuitry in the controller can recognize that the timing signals are not being transmitted properly.

5.38 ***Tests 49 Through 51:*** The write-protect function of the MAS is checked by initiating the following tests.

- The write-protect registers are initialized and verified. A check is made to ensure that each bit of the four 16-bit write-protect registers can be written and read independently (only with maintenance store commands).

- Tests are made to verify that a normal write command cannot write into a write-protected area of memory but can write into an unprotected area.

5.39 ***Test 52:*** The on-line bus access to the store controller is verified. This test checks to determine that the active 3A CC can access the off-line MASC via the bus port (Fig. 7). Checks are made to verify that command, address, data, and control signals can be transmitted via this path.

5.40 ***Tests 53 Through 56:*** Outputs and inputs of the store data (SD) bits that were not tested during the store data bus test are now checked for integrity. Control and data bits are tested for a stuck-active condition as well as a stuck-low condition.

5.41 ***Test 57:*** Memory cell integrity is the last diagnostic to be run on the MAS. This test uses the on-line machine to check the first 4K (K=1024) words of the off-line MAS. The first 4K of each store will be tested in the case of two or more stores in the off-line. Various data patterns are written into the first 4K and then read and compared to ensure that data can be stored and retrieved correctly.

5.42 ***Tests 58 Through 64:*** If diagnostic tests are successfully executed through test 57, the off-line CU is presumed to be reasonably capable of executing self-check tests. Tests 58 through 64 allow the off-line machine to test the next 48K of memory in a manner similar to test 57. The off-line diagnostic code (CDGS3A) is placed in the memory previously tested by test 57 and executed from that area of off-line MAS. The corresponding memory locations are tested if the off-line has two or more stores equipped.

5.43 ***Test 65:*** When the 3A CC is in the microinterpret mode, a 16-bit main memory word is loaded into the TO and FROM fields of the MIR. Data transfers (register-to-register) are executed and checked to ensure proper operation of the 3A CC in the microinterpret mode.

5.44 ***Test 66:*** Open loads in the miscellaneous crosspoint subfield can result in multiple crosspoint ***firing*** when only three of four decoder subfields are present. Initial conditions are set up, and a series of crosspoint data is generated. The results are then examined to ensure correct operation.

5.45 ***Test 67:*** Spare.

5.46 ***Test 68:*** The gating bus parity checker is tested using a series of operands containing good parity as well as parity with errors. The off-line machine executes its own code since the parity error checker is inhibited during the stop mode. Parity low (bits 0 through 7) and parity high (bits 8 through 17) are tested. A test is also made to ensure that a stop and switch will not occur if a parity error comes in from an I/O channel.

5.47 ***Test 69:*** The parity generators associated with the X- and Y-fields of the IB are tested here. Known constants are loaded into the X- and Y-fields, and checks are made to determine if an error is set from bad data. The error register should not be set with correct constants.

5.48 ***Test 70:*** The timing register (TI) is a 16-bit counter driven by the output of the 4-bit prescaler. The lower eight bits of the TI are used as the timing counter (TC), while the program time (PT) is comprised of the upper eight bits. Tests are made to ensure that the TI can be loaded externally and that the prescaler and TI increment correctly.

5.49 ***Test 71:*** Each level of interrupt is tested by setting the request bit in the interrupt set register (IS). A test is made to ensure that interrupts can be masked and that interrupts can be processed properly.

5.50 ***Test 72:*** Tests are initiated to ensure that the SAR and address input registers (AI) can be compared for a match or mismatch. The capability of the address mask register (AK) to mask every bit from the SAR is also tested.

5.51 ***Test 73:*** The integrity of the matcher circuit, which compares the data input register (DI), and the data on the store bus are checked. The capability to mask a match on data with the OK register is tested. A test is also made to ensure that data matching can be inhibited when the data match enable (DME) bit is cleared.

5.52 ***Test 74:*** The equipped I/O channels are checked to ensure proper selection. Illegal I/O select orders are generated to determine whether error bits are generated. Tests also determine that one I/O channel does not affect other channels.

5.53 ***Tests 75 and 76:*** A number of system status bits are used to control processor operation (initialization, stop and switch, etc). These controls are functionally tested by verifying that correct operation occurs when each system status bit is set. The off-line sanity test is also run to ensure that it can be executed without firing any errors.

5.54 ***Test 77:*** A test is made to determine that the system status panel (SSP) can be accessed properly. All addressable SSP registers are checked for initialization and data integrity. The read only op-code is verified. The time-out circuitry is checked by setting the timer and waiting for the panel to time out.

5.55 ***Test 78:*** A double store read operation provides the capability for the on-line CU to read the same memory location of the off-line store if an error on its own memory read is detected. Test 78 checks the gating paths, crosspoints, and registers used in the double store read operation. Functional tests are also implemented.

5.56 ➧***Test 79:*** This test performs an off-line sanity test. If the CU has passed all the diagnostics up to this point, this test should not cause any errors.

5.57 ***Test 80:*** This test checks the operation of EOS macro code.➧

B. DMA Diagnostic Tests

5.58 The DMA diagnostic tests are listed in Table F. A DMA communicates with its periphery over a PCH. Therefore, the PCH test sequence (Table G) is included in DMA diagnostics as tests 11-33. Requesting any of the DMA tests (11-33) will actually cause the corresponding PCH test to be executed. However, the tests will exercise only the PCH hardware associated with the DMA channel being tested. The DMA diagnostics assume that the CU is functioning properly. Each of the DMA diagnostics is described briefly in the following paragraphs. The associated PRs contain a more detailed description.

5.59 ***Test 1:*** This test checks the following:

- The interface between the DMA and CU
- The DMA internal bus for a "stuck-at" fault which causes a register or part of a register to be gated permanently onto the bus
- The DMA register sequence.

5.60 ***Test 2:*** This test determines if the DMA can load its registers properly. The program tests the decoder on the CU interface board to determine if an erroneous bit pattern will fire any of its output. Each register is checked for stuck bits. The operation of the address decoder is verified.

5.61 ***Test 3:*** The following parity generator/checkers are tested:

- Present Address
- Final Address
- Next Address
- Device Address.

Also, the DMA memory sequence is tested.

5.62 ***Test 4:*** This test checks the present address and final address comparator to determine if it can recognize when the present address is equal to, less than, or greater than the final address.

5.63 ***Test 5:*** This program tests the counter that is used to increment the present address. The counter is first checked for stuck bits. It is then checked to determine if it can accurately count through a memory sequence.

5.64 ***Test 6:*** The present and next address comparator is checked to see if it is detecting error conditions accurately.

5.65 ***Test 7:*** This test runs on the on-line CU and checks the off-line CU. The CU and DMA share the same store bus. This program tests the address bits and control signals of the store bus. Any stuck address bits will be identified. The capability of the store bus control to access both stores is tested. Also, the capability of the DMA to read a bad parity word from its own store is also tested.

5.66 ***Test 8:*** This test checks the ability of the DMA to access its own store using the following tests.

- P_H and P_L of data words on the store bus are tested.
- All 1s are written into store.
- All 1s are read from store.
- All 0s are written into store.
- All 0s are read from store.

- A word with double parity errors is written into store.
- The store error signals A, B, and C (SERA, SERB, and SERC) are tested.
- Store automatic correction is tested.
- The blind write is tested.

5.67 ***Test 9:*** This is a dummy test reserved for future growth.

5.68 ***Test 10:*** This test checks the signals which enable the main PCH (MPCH) mode. This must be done before PCH diagnostics are called.

5.69 ***Tests 11 through 33:*** Tests the PCH used by the DMA using the PCH diagnostics. These tests correspond to the PCH diagnostics tests 1 through 23.

5.70 ***Test 34:*** This test assumes that the PCH diagnostics ran successfully. It tests the capability of the DMA to communicate with devices while in the regular DMA mode.

5.71 ***Test 35:*** This test checks the priority encoding circuit and the multiplex function of the subparallel channel (SPCH) address.

5.72 ***Test 36:*** This test checks the DMA data transfer sequence using a 3-word transfer in both the read and write directions.

5.73 ***Test 37:*** The DMA is set to the maintenance mode. The mask register is used to generate a request from every device. The dual bus selector (DBS) is put in the DBS mode. Data (4K) is transferred from the DBS to the store and then the reverse is performed. The device error, the LDT and LDTB error, and the data parity checker are tested.

5.74 ***Test 38:*** This test runs on the on-line CU. While all other DMA tests diagnose the off-line DMA, this test diagnoses the capability of the on-line DMA to access the other store. In performing this test, the data bits are checked and the store automatic correction function is tested.

C. PCH Diagnostic Tests

5.75 The PCH diagnostics run on the off-line machine and only the PCHs of the off-line machine are diagnosed. Unless a specific unit is requested, all equipped PCHs of the off-line machine are tested. When a PCH is diagnosed, all of its subunits are tested. The PCH diagnostic sequence is listed in Table G. A brief description of each of these tests follows. A more detailed description is contained in the associated PR.

5.76 ***Tests 1 Through 8:*** These programs test the flags identified in Table G for a stuck condition.

5.77 ***Tests 9 and 10:*** These tests exercise the low byte and high byte, respectively, of the information (INF) signal loop-around register of each SPCH in each PCH.

5.78 ***Tests 11 and 12:*** These tests check the INF parity low bit and parity high bit, respectively, for each SPCH in each PCH. The bits are checked for a stuck condition.

5.79 ***Tests 13 and 14:*** These tests exercise the low byte and high byte, respectively, of the AC bus driver and receiver of each SPCH in each PCH.

5.80 ***Tests 15 and 16:*** These tests check the parity low and parity high bits, respectively, of the AC bus driver and receiver of each SPCH in each PCH. Each bit is checked for a stuck condition.

5.81 ***Test 17:*** This test exercises the multiple command check circuit of each PCH. Each circuit is tested with the following commands:

- Data receive (DR)
- Sense status (SST)
- Data present (DP)
- Command present (CP)
- Acknowledge interrupt (ACKI).

5.82 ***Test 18:*** This test exercises the multiple SPCH check circuit in each PCH.

5.83 ***Test 19:*** This test exercises the AC bus from each SPCH in each PCH. The program will detect a failure on any of the following leads:

- SST
- CD
- SYNC
- ER
- INF.

5.84 ***Test 20:*** This test exercises the INF leads of the AC bus from each SPCH in each PCH. The test is performed using the loop-around registers.

5.85 ***Test 21:*** This test exercises the ACKI command lead of the AC parallel bus from each SPCH in each PCH. Each lead is checked for a stuck condition.

5.86 ***Test 22:*** This test exercises the DBS INIT lead of the AC parallel bus from each SPCH in each PCH. Each lead is checked for a stuck condition.

5.87 ***Test 23:*** This test exercises the interrupt lead of the AC parallel bus from each SPCH in each PCH. Each lead is checked to see if it will report an interrupt from the DBS.

D. ➧CTI Diagnostic Tests

5.88 The CTI diagnostic tests are listed in Table H. The CTI provides logic-level conversion for the processor frame. The CTI diagnostic assumes that the CU diagnostics have all completed successfully. The tests of the CTI diagnostics are discussed in the following paragraphs.

5.89 ***Test 1:*** This test verifies the operation of bits 15 through 10 of general register 9 (R9). R9 is used for I/O communication.

5.90 ***Test 2:*** This test checks the remaining bits (0 through 9) of R9.

5.91 ***Test 3:*** The interface between general register 10 (R10) and 11 (R11) to the channels is tested.

5.92 ***Test 4:*** This test checks the I/O unit response line.⬅

6. GLOSSARY

6.01 The following basic terms are defined in the context of their use in this section.

Application—A set of functional system programs which uses the services of the EOS.

Duplex Bus Selector (DBS)—An electronic switch which transfers control of a peripheral device from one processor to the other.

Macro—A precoded sequence of instructions that is assigned a name to serve as a label. This label, along with any parameters, can be used as an instruction in a source language. On each call, the macro instruction is replaced by its equivalent instruction sequence.

Maintenance Channel (MCH)—The serial channel over which the two CUs in a duplex system communicate. The channel is used for control and diagnostic purposes.

Microinstruction—A low level instruction used in microinstruction sequences that are permanently stored in a read-only memory (ROM). The microinstruction sequences are used to implement the 3A CC instruction set and basic control functions.

Off-line—A condition in which a processor is operating correctly but is not called on to perform its primary function.

On-line—A condition in which a processor is performing its primary function without detectable error conditions.

OSTABS—Operating system prototype tables used by the application to define system resources, configuration, parameters, etc.

Own—Refers to a unit in a duplex system which is on the same side.

Other—Refers to a unit in a duplex system which is on a different side.

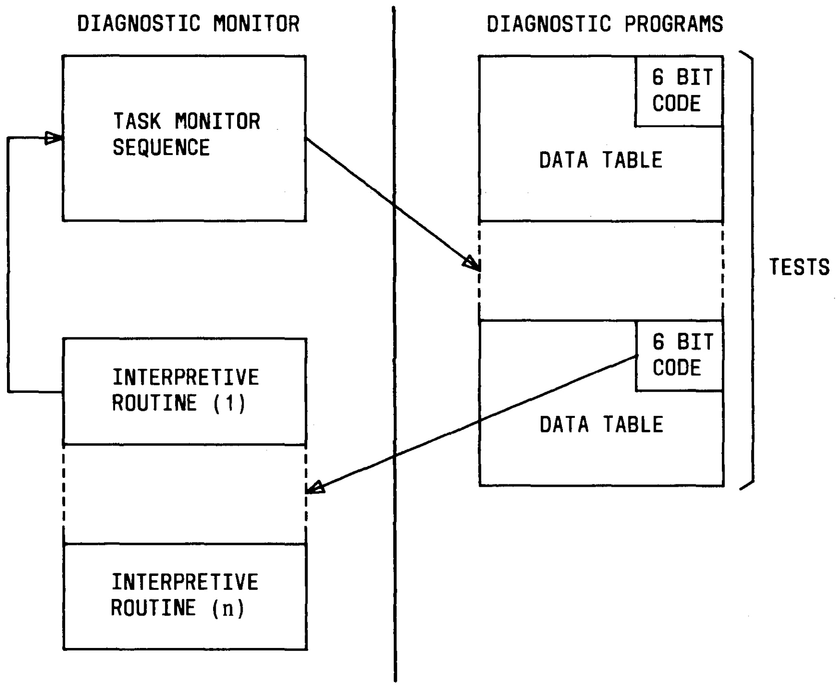


Fig. 1—Table Driven Element Interrelation

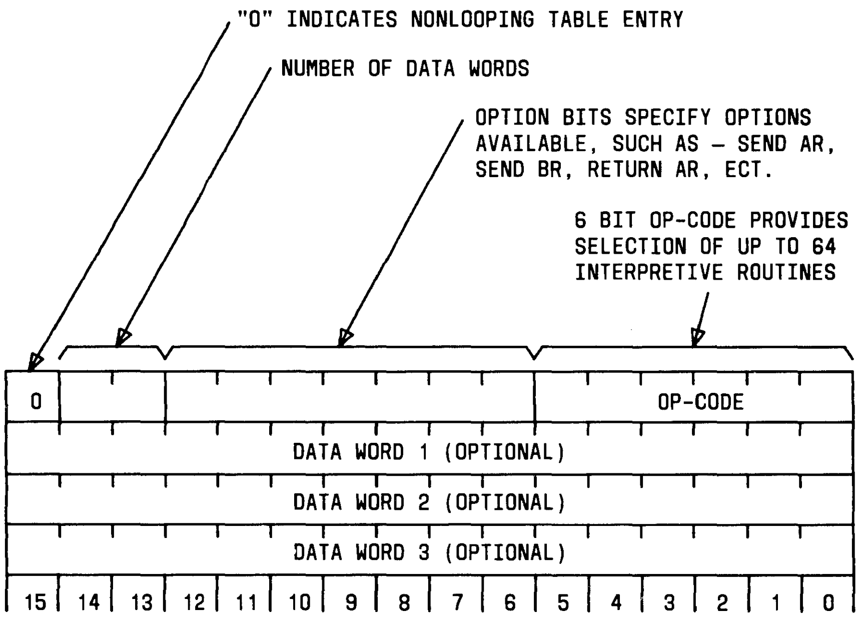

"0" INDICATES NONLOOPING TABLE ENTRY
NUMBER OF DATA WORDS
OPTION BITS SPECIFY OPTIONS AVAILABLE, SUCH AS – SEND AR, SEND BR, RETURN AR, ECT.
6 BIT OP-CODE PROVIDES SELECTION OF UP TO 64 INTERPRETIVE ROUTINES
0
OP-CODE
DATA WORD 1 (OPTIONAL)
DATA WORD 2 (OPTIONAL)
DATA WORD 3 (OPTIONAL)
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Fig. 2—Table Entry Without Loops

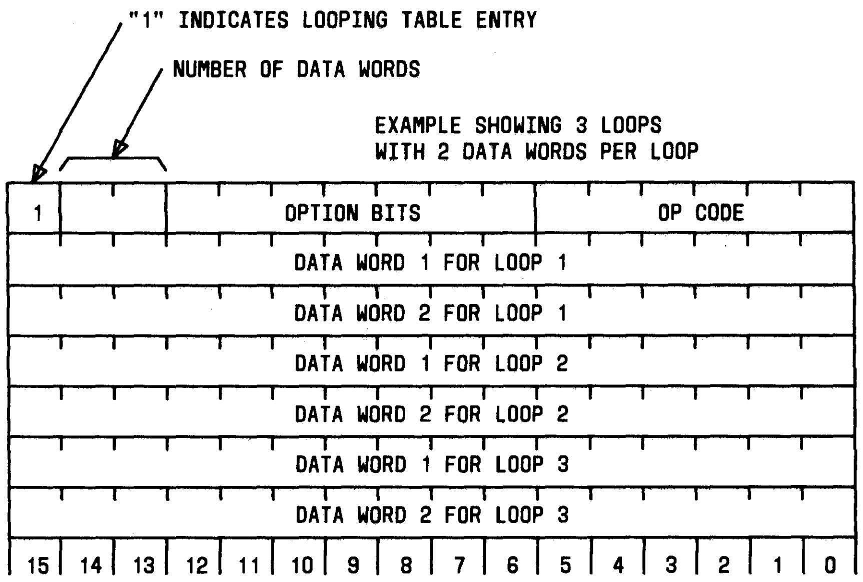


Fig. 3—Table Entry With Loops

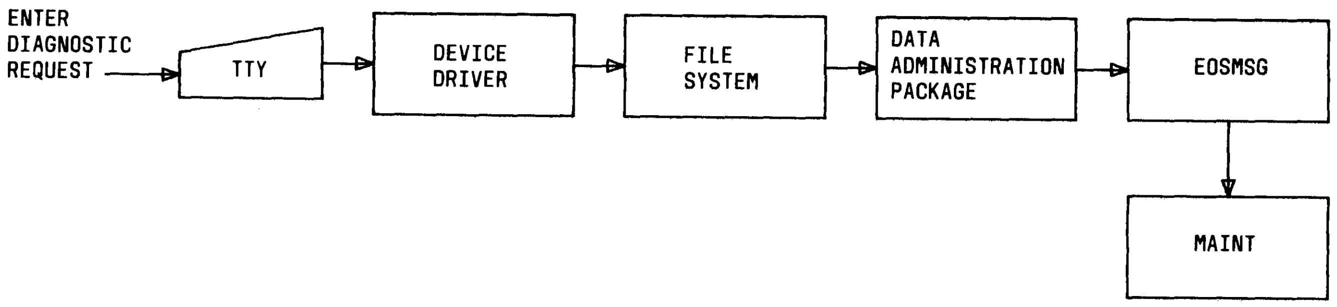


Fig. 4—Message Processing for Diagnostic Requests

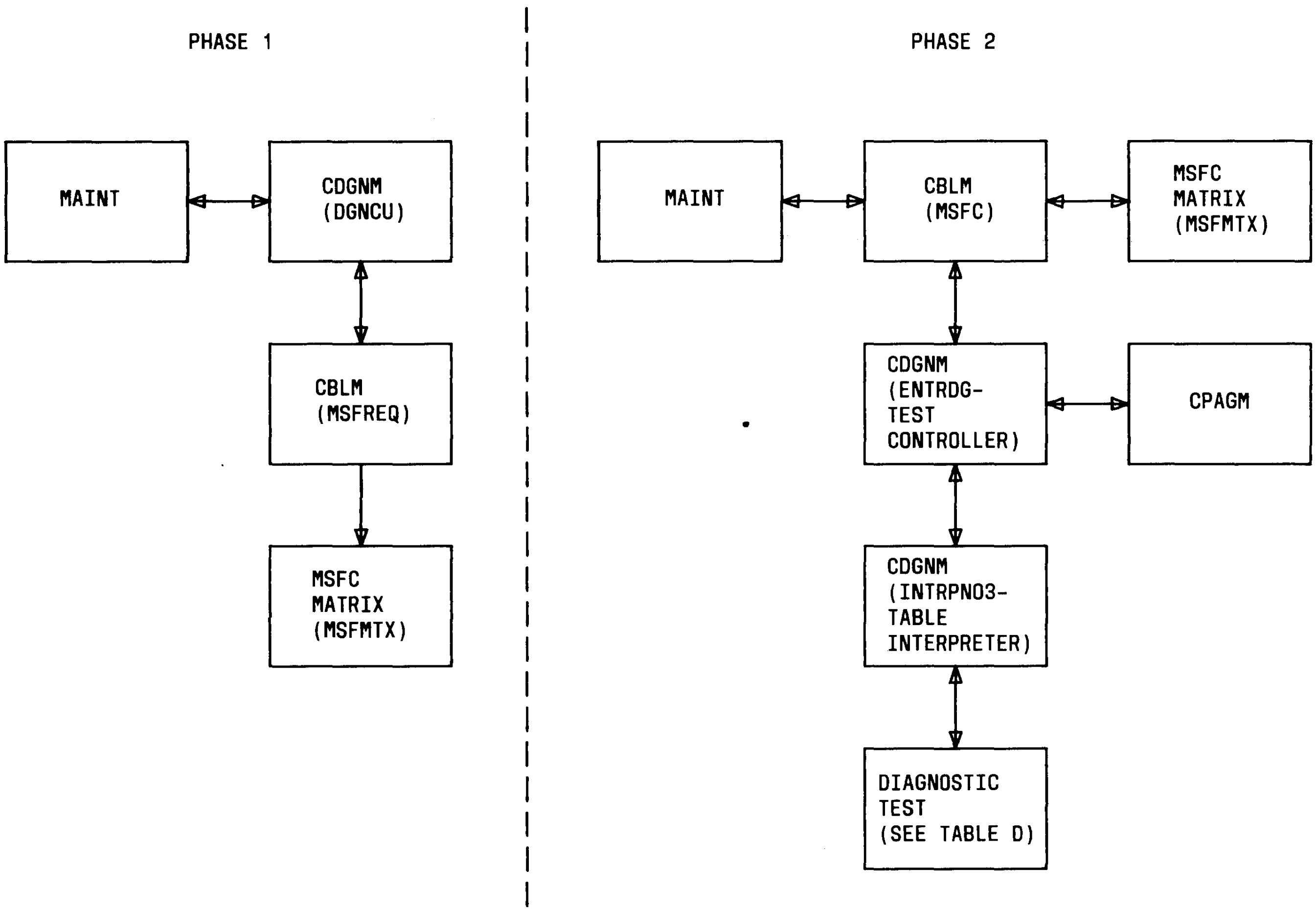


Fig. 5—CU Diagnostic Entry and Execution

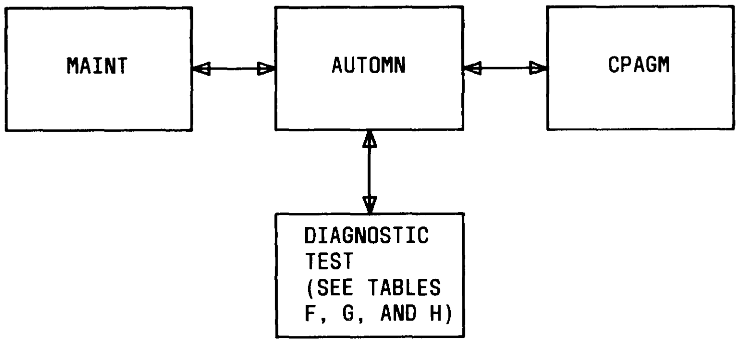


Fig. 6—DMA and PCH Diagnostic Entry and Execution

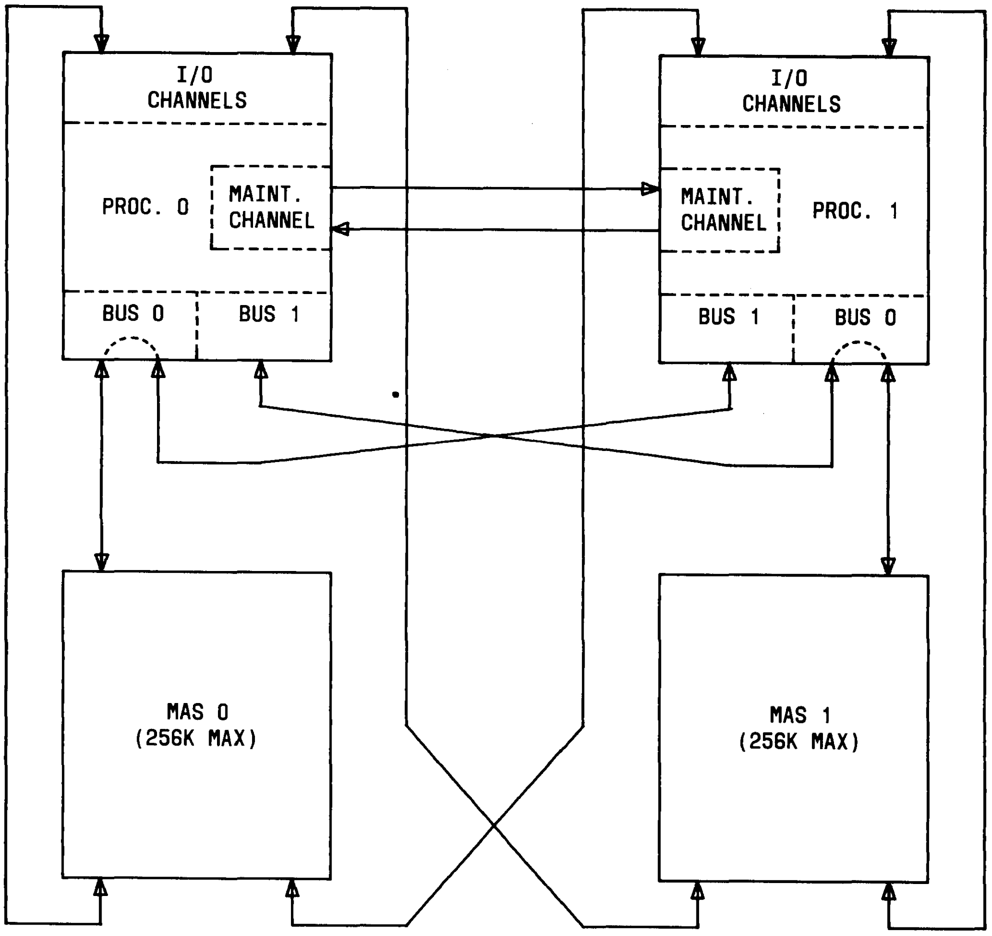
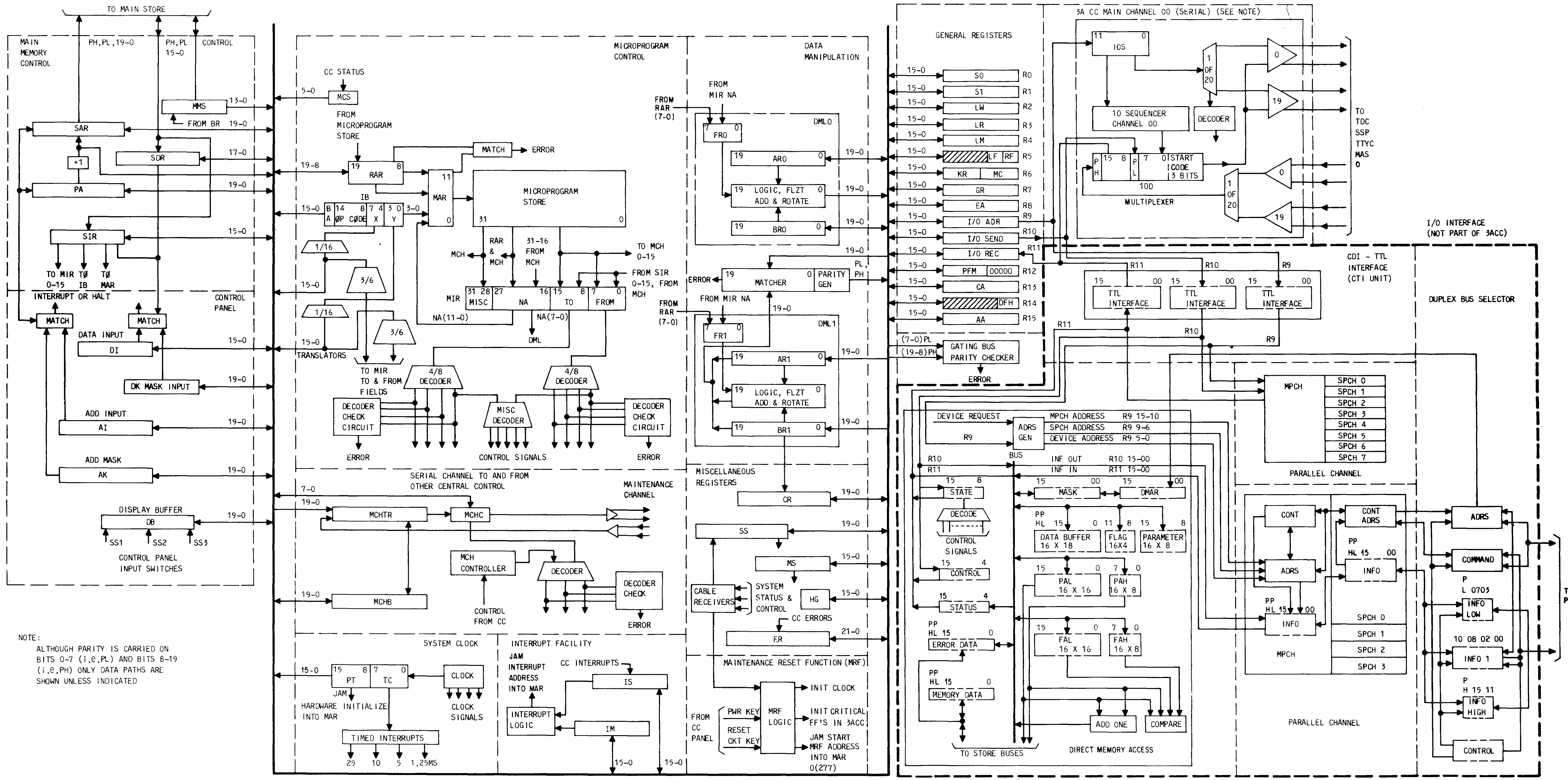


Fig. 7—Control Unit Intercommunication



NOTE:
ALTHOUGH PARITY IS CARRIED ON BITS 0-7 (i.e.PL) AND BITS 8-19 (i.e.PH) ONLY DATA PATHS ARE SHOWN UNLESS INDICATED

Fig. 8—Detailed 3A Central Control Block Diagram

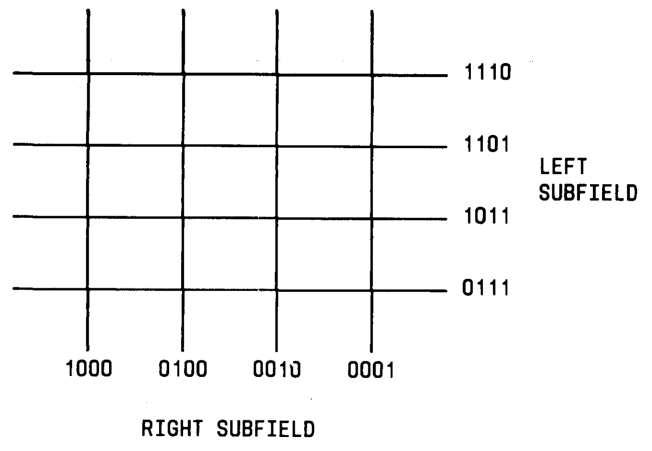


Fig. 9—Subfield Matrix

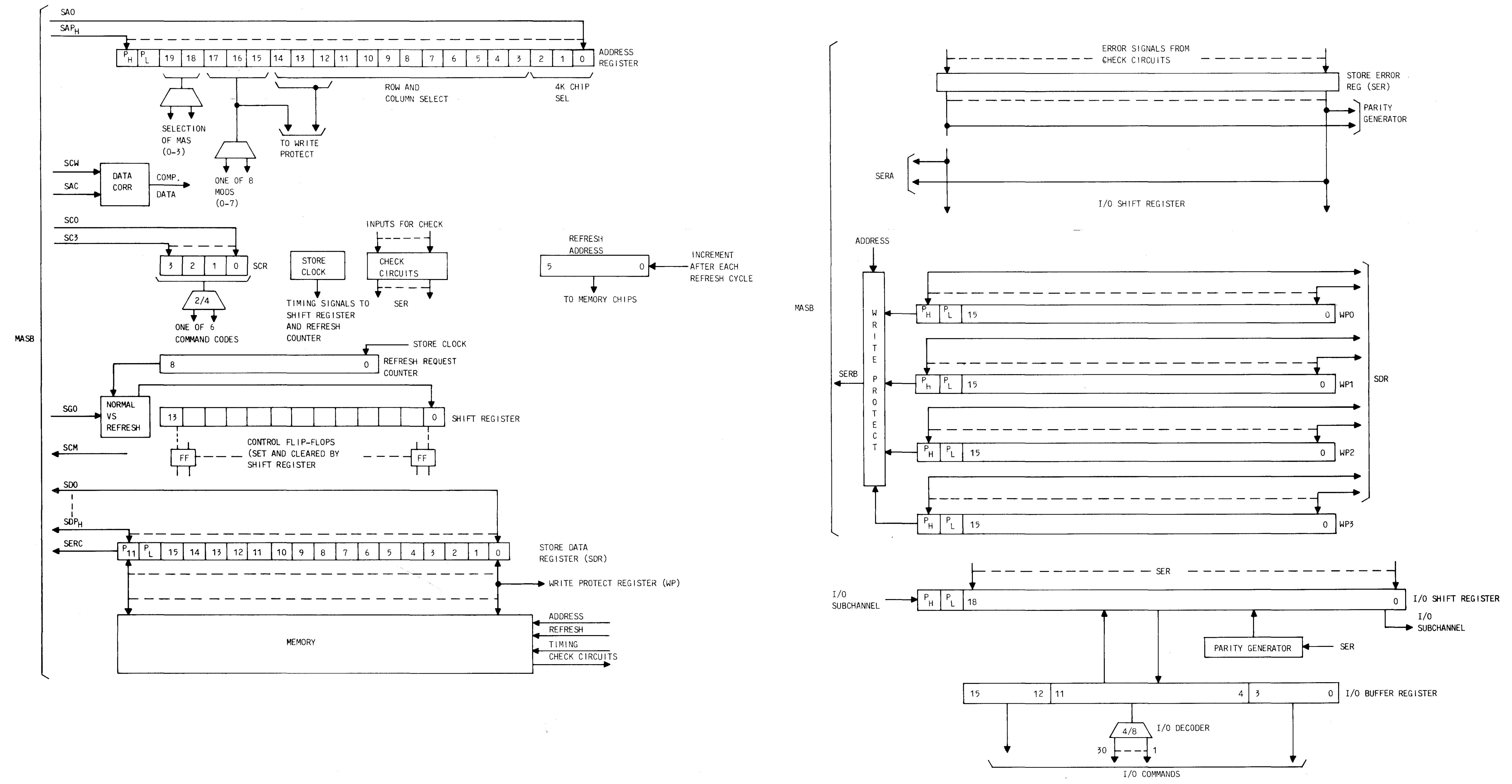


Fig. 10—Functional Diagram of the Main Store Controller

♦TABLE A♦

ABBREVIATIONS AND ACRONYMS

TERM	MEANING
ACKI	Acknowledge Interrupt command/lead
AFO	Address Sent Flag
AI	Address Input Register
AK	Address Mask Register
AR	A Register
ATP	All Tests Pass
BR	B Register
CA	Control Bit A
CB	Control Bit B
CC	Central Control
CFO	Command Sent Flag
CP	Command Present command
CTI	Collector Diffusion Isolation-to-Transistor-Transistor Logic Interface
CU	Control Unit
DBS	Duplex Bus Selector
DI	Data Input Register
DMA	Direct Memory Access
DME	Data Match Enable
DML	Data Manipulation Logic
DP	Data Present command
DR	Data Ready Flip-Flop or Data Receive command
DS	Data Status
EOS	Extended Operating System
EPL	ESS Programming Language
ER	Error Register
ERAR	Error Return Address Register
ERU	Error Return Address Update
ESS	Electronic Switching System
FLZ	Find Low Zero
FR	Function Register
IB	Instruction Buffer
IFH	High Byte Information Sent bit
IFL	Low Byte Information Sent bit
INF	Information Signal Loop-around Register

♦TABLE A (Contd)♦

ABBREVIATIONS AND ACRONYMS

TERM	MEANING
INIT	Initialization lead
I/O	Input/Output
IS	Interrupt Set Register
MAR	Microaddress Register
MAS	Main Store
MASC	Main Store Controller
MCH	Maintenance Channel
MCHB	MCH Control Buffer
MIR	Microinstruction Register
MOD	Memory Module
MPCH	Main Parallel Channel
MSF	Multi-Scan Function
PA	Program Address Register
PCH	Parallel Channel
PCHENA	PCH Enable
PGB	Program Gating Bus
PH	Parity high bit
PL	Parity low bit
PR	Program Listing
PT	Program Timer
R0	General Register 0
R9	General Register 9
R10	General Register 10
R11	General Register 11
RAR	Return Address Register
ROM	Read Only Memory
RU	Return Address Update Flip-Flop
SAR	Store Address Register
SD	Store Data bits
SERA	Store Error A
SERB	Store Error B
SERC	Store Error C
SPCH	Subparallel Channel
SSP	System Status Panel
SST	Sense Status command/lead
SYNC	Synchronization lead
TC	Timing Counter
TI	Timing Register
TLM	Trouble Locating Manual

♦TABLE B♦

3A PROCESSOR DIAGNOSTIC PROGRAM LISTINGS

PROGRAM	NUMBER	NAME
CDGDSR	PR-1C911	Double Store Read Test
CDGFN	PR-1C917	Data Manipulation Logic (DML) Tests
CDGMCH	PR-1C912	MCH, Gating Bus, Clock And Register Initialization Tests
CDGMC1	PR-1C919	Micro Control Tests
CDGMI	PR-1C930	Micro Interpret, Multiple Crosspoint, And Parity Check Tests
CDGMIC	PR-1C916	Microstore Content Test
CDGMLT	PR-1C914	Multiple Crosspoint Tests
CDGMSQ	PR-1C921	Store Bus Controller Tests
CDGNM	PR-1C910	Common Diagnostic Monitor
CDGNOF	PR-1C933	Off-Line Diagnostic Code
CDGNTI	PR-1C931	Timing Counter, Interrupt, Address And Data Matcher, And I/O Tests
CDGREG	PR-1C915	Register Gating Tests
CDGSBS	PR-1C924	Store Bus Communication Test
CDGSCP	PR-1C934	Store Control And Parity Test
CDGSD	PR-1C929	Memory Element Test
CDGSDF	PR-1C935	Store Controller Data Register Test
CDGSFA	PR-1C925	Store Fanout Boards Test, Part A
CDGSFB	PR-1C926	Store Fanout Boards Test, Part B
CDGSIO	PR-1C922	Store I/O Access Test
CDGSMX	PR-1C923	Store Multiplex Circuit Test
CDGSON	PR-1C928	Interprocessor Store Bus Test
CDGSWP	PR-1C927	Store Write Protect Test
CDGS3A	PR-1C918	Off-Line Store Diagnostic Code
CDGTO	PR-1C913	To And From Crosspoint Tests
CDSPA1	PR-1C920	DS And PA+1 Adder Tests
CEOSIN	PR-4C710	Extended Instruction Macro Code Diagnostic
CSTATS	PR-1C932	System Status Bits, Switching, And System Status Panel Tests
DGDMA	PR-4C702	DMA Diagnostic (Part 1)
DGLIU	PR-4C703	DMA Diagnostic (Part 2)
DGNCTI	PR-4C704	CTI Diagnostic
DGNPCH	PR-4C705	Parallel Channel Diagnostic

TABLE C

DIAGNOSTIC MACROS

MACRO	FUNCTION
	On-Line Control Macros
BEGIN_DIAG	This macro is used at the beginning of a diagnostic program. It brings in various libraries, sets up linkage to the diagnostic monitor, and handles titles.
DGTST	Begins a diagnostic test.
TESTSEG	Begins a diagnostic test segment, which is a subsection of a diagnostic test.
CALLPROS	Allows the on-line CU to continue processing while the off-line CU runs the diagnostic.
TIMETST	Causes the on-line CU to loop for the allotted time in microseconds while the off-line CU runs diagnostic code.
ABORT	Sets or clears an interrupt abort switch. If the switch is set every time a COMPARE command is encountered, a check is made to see if an on-line interrupt occurred. If an interrupt occurred, the test is aborted and restarted to avoid false error indications in time-critical tests.
NO3_CODE	Exits the table-driven portion of a diagnostic test and transfers to a specified 3A language routine. This macro enables functions to be programmed that are not provided by macros.
RTN_TBL	Returns to the table-driven portion of a diagnostic test at a specified location.
RTN_TBLX	Returns to the table-driven portion of a diagnostic test at a specified location indexed by R0.
PSOFFTST	Issued in the off-line CU when a diagnostic runs to completion and passes.
FLOFFTST	Issue in the off-line CU when a diagnostic runs to completion and fails.
	Off-Line Control Macros
MICRO	Transmits a microinstruction over the maintenance channel and causes the instruction to execute in the off-line CU.
SEND	Sends a 22-bit constant (20 data bits and 2 parity bits) via the maintenance channel and loads the data into the specified register of the off-line CU.
RETRN	Returns the specified off-line register to the on-line maintenance channel buffer.
ZERO_ER	Zeros the off-line error register.
ZERO_MS	Zeros the off-line maintenance state register.

TABLE C (Contd)

DIAGNOSTIC MACROS

MACRO	FUNCTION
LOAD_MIRH	Loads the high 16 bits of the off-line MIR with the specified constant.
LOAD_MAR	Clears the STOP bit in the system status register, loads the MAR with the specified 12-bit constant, and resets the FREEZ flip-flop, holding the constant in the MAR.
ZERO_PT	Zeroes the off-line program timer.
DIS_A	Executes the first step in a 2-part sequence of disabling the off-line I/O channels.
DIS_B	Executes the second step in a 2-part sequence of disabling the off-line I/O channels.
STRT_OFF	Clears the FREEZ flip-flop and thus allows microcontrol to sequence.
STOP_OFF	Sets the STOP bit in the system status register. This causes a special address to be jammed into the MAR with the corresponding location containing all zeros.
STRT_CLK	Starts the off-line clock.
STOP_CLK	Stops the off-line clock. The clock will remain stopped for several milliseconds, after which a hardware backup circuit will restart it.
TOG_CLK	Toggles the off-line clock one-half place. The clock must be stopped to use this macro.
RESET_CLK	Initializes the off-line clock to phase P3.
ADD_OFF	Starts the off-line CU executing at the specified address in main store.
ADD_MICRO	Starts the off-line CU executing microcode at the specified address.
EXECUTE	Executes a 3A language instruction in the off-line CU.
READ_MICRO	Returns the low 16 bits or high 16 bits (as specified) of the off-line microstore output to the on-line MCHB.
	Interrogatory Macros
COMPARE	Compares the contents of the on-line MCHB with the specified constant.
MASK	Zeros the bits in the on-line MCHB that are zero in the specified pattern.
TTY_DATA	Prints a data word in the diagnostic failure message if a failure occurs.
FAILTST	Stops processing and prints a message when a failure occurs.

TABLE C (Contd)

DIAGNOSTIC MACROS

MACRO	FUNCTION
FAILBIT	Stops processing and prints a message when a failure occurs. Generally used when diagnosing bit-sliced hardware.
PASSTST	Returns control to the diagnostic monitor when a test passes.
	TLM Generation Macros
BEGIN_TLM	Begins the TLM portion of the listing.
TLM_TTL	Generates a trouble number in the TLM.
RESET	Resets the sequential part of the trouble number and increments the test number portion of the trouble number.
	Looping Feature
LP_BEGIN	Establishes the loop starting point (or upper boundary).
LP_END	Establishes the end of the loop (or lower boundary).

TABLE D

CU DIAGNOSTIC TEST SEQUENCE

TEST PROGRAM	TEST NUMBER	DESCRIPTION
CDGMCH (PR-1C912)	1	Maintenance Channel
	2	Gating Bus
	3	System Clock
	4	Verification of Initialization by Monitor
CDGTO (PR-1C913)	5	TO Decoder
	6	FROM Decoder
CDGMLT (PR-1C914)	7	Test for Multiple Firing FROM Crosspoints
	8	Test for Multiple Firing TO Crosspoints
CDGREG (PR-1C915)	9	General Registers
	10	Special Registers
CDGMIC (PR-1C916)	11	Microstore
CDGFN (PR-1C917)	12	Function Register
	13	Add Function
	14	Matcher for Duplicate DML
	15	Boolean Logic Functions
	16	Find Low Zero Function
	17	Rotate Function
	18	Pack and Unpack Gating Operation and DML Parity Generator
CDGMC1 (PR-1C919)	19	Microcontrol Part 1
	20	Microcontrol Part 2
	21	Microcontrol Part 3
	22	Microcontrol Part 4
CDSPA1 (PR-1C920)	23	DS Flip-Flop
	24	Address Increment Adder
CDGMSQ (PR-1C921)	25	Store Bus Controller Part 1
	26	Store Bus Controller Part 2
CDGSIO (PR-1C922)	27	Unlock and Test On-Line CC I/O Access to Off-Line Store
	28	Store Controllers I/O Order Decoder

TABLE D (Contd)

CU DIAGNOSTIC TEST SEQUENCE

TEST PROGRAM	TEST NUMBER	DESCRIPTION
CDGSMX (PR-1C923)	29	One-Half of Store Controller Multiplex Circuits
	30	Other Half of Store Controller Multiplex Circuits
	31	One-Half of Store Controller Check Circuits
	32	Other Half of Store Controller Check Circuits
CDGSBS (PR-1C924)	33	Store Command Portion of Store Bus
	34	Store Address Portion of Store Bus
	35	Store Data Portion of Store Bus
	36	Store Data Portion of Store Bus
CDGSCP (PR- 1C934)	37	Bus Control Circuitry
	38	Data Parity Control Circuitry
	39	Data Parity Control Circuitry
	40	Data Parity Check Circuits
CDCSFA (PR-1C923)	41	Refresh Select Signals to Fanout Boards
	42	Normal Select Signals to Fanout Boards
	43	Store Address Signals Through Fanout Boards
	44	Store Address Signals Through Fanout Boards
CDGSFB (PR-1C926)	45	Fanout Board Address Parity Checkers
	46	Fanout Board Address Parity Checkers
	47	Store Timing Signals Through Fanout Boards
	48	Store Timing Signals Through Fanout Boards
CDGSWP (PR-1C927)	49	Write Protect Reads and Writes
	50	Write Protect Reads and Writes
	51	Write Protect Check Circuits
CDGSON (PR-1C928)	52	Store Bus From On-Line CC to Off-Line CC

TABLE D (Contd)

CU DIAGNOSTIC TEST SEQUENCE

TEST PROGRAM	TEST NUMBER	DESCRIPTION
CDGSDF (PR-1C935)	53	Data Register To/From Memory Modules
	54	Data Register To/From Memory Modules
	55	Data Register To/From Memory Modules
	56	Data Register To/From Memory Modules
CDGSD (PR-1C929)	57	First 4K of Memory
	58	Second 4K of Memory
	59	Third 4K of Memory
	60	Fourth 4K of Memory
	61	Fifth 4K of Memory
	62	Sixth 4K of Memory
	63	Seventh 4K of Memory
	64	Eighth 4K of Memory
CDGMI (PR-1C930)	65	Microinterpret Operation
	66	Test For Multiple Firing Miscellaneous Crosspoints
	67	Spare
	68	Bus Parity Check
	69	IB Register X- and Y- Field Parity Generator and Parity Checker
CDGNTI (PR-1C931)	70	Program Timer, Timing Counter
	71	Interrupts
	72	Panel Address Matcher
	73	Panel Data Matcher
	74	I/O Channels
CSTATS (PR-1C932)	75	Status Bits, Initialization
	76	Status Bits, Time-Outs
	77	System Status Panel
CDGDSR (PR-1C911)	78	Double Store Read
	79	Sanity
CEOSIN (PR-1C710)	80	Extended Instruction Macro Code

TABLE E

SUBFIELD COMBINATIONS

RIGHT SUBFIELD	LEFT SUBFIELD	COMBINATION TOTAL
0/4 (1)	4/4 (1)	1
1/4 (4)	3/4 (4)	16
2/4 (6)	2/4 (6)	36
3/4 (4)	1/4 (4)	16
4/4 (1)	0/4 (1)	1

TABLE F

DMA DIAGNOSTIC TEST SEQUENCE

TEST PROGRAM	TEST NUMBER	DESCRIPTION
DGDMA (PR-4C702)	1	Interface to CU, Internal Bus and Register Sequence
	2	Register Loading
	3	Parity Generator/Checkers
	4	Address Comparator (Present and Final)
	5	Present Address Counter
	6	Address Comparator (Present and Next)
DGLIU (PR-4C703)	7	DMA Store Bus
	8	DMA Store Access
	9	Dummy Test
	10	MPCH Mode Enabling Signals
DGNPCH (PR-4C705)	11-33	Corresponds to Tests 1-23 of PCH Diagnostics (See Table G)
DGLIU (PR-4C703)	34	DMA Device Communication Ability While in Regular DMA Mode
	35	Priority Encoding Circuit
	36	DMA Data Transfer Sequence
	37	Maintenance Mode, DBS Mode, and Data Parity Checker
	38	On-Line DMA Capability to Access Off-Line Store

TABLE G

PCH DIAGNOSTIC TEST SEQUENCE

TEST PROGRAM	TEST NUMBER	DESCRIPTION
DGNPCH (PR-4C705)	1	Low-Byte Maintenance Flag
	2	High-Byte Maintenance Flag
	3	Low-Byte Information (IFL) Sent Flag
	4	High-Byte Information (IFH) Sent Flag
	5	PCH Enable (ENA) Flag
	6	PCHENA Flag
	7	Address Sent Flag (AFO)
	8	Command Sent Flag (CFO)
	9	INF Signal Loop-Around Register (Low Byte)
	10	INF Signal Loop-Around Register (High Byte)
	11	INF Parity Low Bit
	12	INF Parity High Bit
	13	AC Bus Driver and Receiver (Low Byte)
	14	AC Bus Driver and Receiver (High Byte)
	15	AC Bus Driver and Receiver Parity Low (PL) Bit
	16	AC Bus Driver and Receiver Parity High (PH) Bit
	17	Multiple Command Check Circuit
	18	Multiple SPCH Check Circuit
	19	AC Bus of SPCH
	20	INF Lead From AC Bus
	21	ACKI Command Lead From Parallel Bus
	22	DBS Init Lead From Parallel Bus
	23	Interrupt Lead From Parallel Bus

♦TABLE H♦

CTI DIAGNOSTIC TEST SEQUENCE

TEST PROGRAM	TEST NUMBER	DESCRIPTION
DGNCTI (PR-4C704)	1	R9 interface bits 15-10
	2	R9 interface bits 0-9
	3	R10 to channel and channel to R11
	4	I/O unit response line