Burroughs Corporation
ELECTRODATA M & E DIVISION
PASADENA CALIFORNIA

CHANGE SHEET

TITLE B5000 PROCESSOR MAINTENANCE TEST ROUTINES - GENERAL TEST SPECIFICATION

A- 11187283

REV. A

REV. LTR.	CONT. DOC. & NO.	DFT-MAN	CHKR.	APP'D BY & DATE	PAGE	LINE OR ¶	REVISION DESCRIPTION
A	ER 1635A		[illegible] 5.4.65	[illegible] 6-7-65			Released

TABLE OF CONTENTS

TABLE OF CONTENTS

I INTRODUCTION

1.1 General

One of the functions of this specification is to describe the philosophy and the features of the General Control Routine (GCR). This routine is stored on the B5500 Processor MTR magnetic tape following the four Basic Tests and preceding the data for over 15,000 test cases. These cases cover normal Processor 1 logic plus the P1-P2 interface. The execution of these test cases is controlled by GCR. GCR and its related test cases together form what is called the general test (as opposed to the basic tests). The general test cases are individually documented in an Error Reference Manual (ERM).

Another function of this specification is to describe the use of the general test-ERM combination as a processor hardware maintenance tool.

1.2 Related Documents

B5500 Processor MTR System Specification	11187267
B5500 Processor MTR Basic Test Specification	11187275

It is assumed that the reader is familiar with the contents of the above documents.

B5500 Processor MTR, GCR Program Listing	11187713
B5500 Processor MTR, Error Reference Manual	11187739
B5500 Processor MTR Test Tape	11187614

II GENERAL DESCRIPTION

2.1 Objectives

2.1.1 Detection of Hardware Failures: Perform a comprehensive checkout of the B5500 Central Processor.

2.1.2 Aid in Failure Diagnosis: In case of failure, supply appropriate information to the user to facilitate the rapid restoration of the processor hardware to its error-free state.

2.2 Scope, Important Assumptions

2.2.1 All normal Processor 1 functions, including the control functions over Processor 2 (if such is present), are tested.

Functions (in this context) are considered normal if they can be executed by a properly working processor when all manual switches on the display panel, except the operator (conditional halt) switch and the timer interrupt inhibit switch, are turned to their normal position (down). These two are assumed to be turned up (ON).

2.2.2 The design of the processor is correct and is correctly described by the equations in the Processor Logic Book.

2.2.3 The logic failures caught are made up of two kinds. One is limited to gating errors caused by open diodes and shorted AND diodes. The other failure type includes all failures which display effects similar to those observed when some failure of the first type occurred. Examples: Open wire, faulty driver, flip-flop stuck in one of the two states. Testing is done also for most wire shorts, slow gates, and too slow or too fast set or reset actions of certain flip-flops.

2.2.4 Multiple component failures may occur.

2.2.5 Intermittent failures may occur.

The following assumptions refer to the necessary system make-up and point out what portions of it are to be operational to run GCR as intended.

2.2.6 The number of memory modules may vary, but their designations are contiguous. Memory accesses to and from all available modules work properly. Modules 0 and 1 are present and can be loaded with GCR and with the test case data from magnetic tape. This implies that an operational I/O channel (Model II) is available. The proper working of most Central Control functions is essential. The availability of a supervisory printer is desirable but not essential (see 2.4.2).

2.2.7 Certain logic areas already checked out up to a certain point within a test run do not fail during the rest of the run. In general, these areas may be considered tested (for the purposes of GCR) if the four basic tests have been run from tape without error. The list of the operators and some other basic functions verified by the basic tests appears in the specification of the basic tests.

2.3 GCR Testing Philosophy

2.3.1 System configuration (availability of second processor, availability of memory modules beyond 0 and 1) is interrogated programmatically. Test cases incompatible with the observed configuration are automatically skipped.

2.3.2 Program Control Under Failure Conditions: The General Control Routine does not rely on error branches corresponding to individual component failures. Instead, it performs a comparison on the resulting conditions. A general test case, therefore, contains only the input flip-flop states and the expected correct output flip-flop states. All input flip-flop states which can be set up for testing, and all output flip-flop states which are stored away for comparison, are specified by a test case. If specifically called for, comparison on the contents of certain memory cell(s) is also performed. Any discrepancy from the expected value routes GCR to a common error-handling path. This approach, although not helpful in error diagnosis, provides almost complete program control even under multiple-failure conditions. There is only a small number of test cases where the program may hang up or blow up. Dealing with them is explained in Section 2.4.2.

2.3.3 Logic Slicing: Programmatic single-pulsing is the essential characteristic of the testing approach in GCR. Whenever possible, a test case exercises only as little logic of the processor as is called for from one clock pulse to the next one, within the operator specified. Furthermore, if not otherwise specified, all but one of the actions possible to inhibit at that clock time, are inhibited.

2.3.4 Special MTR Operators (Summary): The Initiate For Test (IFT, 5111 octal) operator is an extension of the Initiate operator. It can set up from the stack a few more of the processor flip-flops than the normal Initiate operator can. These extra flip-flops are J01F through J08F, the contents of Y and Z, Q01 through Q09, NCSF, MROF, MWOF, and a special MTR flip-flop, CCCF (Clock Count Control FF). Since some of these flip-flops control the actions within IFT itself, the desired test values are held in a temporary (TM) register until they can be distributed.

2.3.4 Con't.)

If CCCF is initiated to 1, it may clear TROF on the first pulse of a newly initiated operator. However, should a memory access have been started by the first pulse of the newly initiated operator, the processor waits for the completion of the access. Another MTR operator, Store For Test, is then automatically forced into the T register. The after-test conditions of the J register and NCSF are saved, again in the TM register.

If CCCF is initiated to 0, normal execution of the operator being tested takes place, from whatever J value it was entered. Normal program sequence prevails, possibly involving many operators, until either a Store For Test operator is reached in the program string or the test case forces a Store For Interrupt (in normal state).

The Store For Test (SFT, 3411 octal) operator is the extension of the Store For Interrupt operator, and the inverse of the IFT operator. SFT cannot, however, store away the values of MROF, MWOF, and CCCF, since these were not saved in TM. If SFT was transferred from P to T (CCCF=0), TM is zero.

2.3.5 Test Case Types: There are essentially two test case types. A simple test case is entered with CCCF=1, slicing out minute action(s) of the many sequences in the operator being tested. The entry point may be at any J value (0 through 15). Exit from the test is forced even if the entry occurred at a J value not appearing on the flow of the operator involved or if the operator itself is illegal.

A special test case is entered with CCCF=0, slicing into the operator being tested at some particular action. The entry point can be at any J value, provided a path exists to an exit condition of the operator. When SECL becomes true, the next operator in sequence is fetched and executed. The use of special test cases is restricted to the checkout of those logic areas which are not directly available in the control words of the MTR operators. Special cases sometimes involve sizable strings of operators. A string is always terminated by an SFT or an SFI operator.

2.3.6 Interrupt Handling: The presence of a processor-dependent interrupt or the absence of any interrupt is not tested unless an interrupt check is requested by a test case. If verification of the presence of an interrupt is called for, the construction of the corresponding interrupt address is also verified. Sometimes a test case specifies interrupt testing only, with no control word comparison. This is an exceptional condition.

2.3.6 Con't.

In some cases, the recognition of the correct interrupt priority is tested.

In some other cases, retaining of the second interrupt (of lower priority) is tested. These cases force GCR to ignore the first interrupt.

2.3.7 Exceptional Test Case Conditions: Some of the test cases require that NCSF be ON. As long as the test case is simple, the presence of an interrupt does not create a problem, because SFT has priority over SFI. In a special case, however, SFI may be forced (normal processor action) before a coded SFT would be reached. In fact, SFI may be the expected correct exit from the test case. Since SFT and SFI do not create identical control words[1], comparisons with SFT-type control words are inhibited. Comparison of one or more of the SFI-created control words may be called out instead. Since after-test comparisons on the contents of up to twelve memory cells may be specified by any test case, comparison of the desired control words is a simple matter. Often, L being higher than expected is the indication that an expected interrupt did not occur and the operator string was terminated by a Store For Test operator.

Suppressing the comparison of after-test control words needs not be restricted to normal state special test cases.

2.4 GCR Implementation and Use

2.4.1 General Description: The general Control Routine is loaded into core memory from the MTR tape, under the control of the MTR tape loader routine. Allocation of memory among GCR, the tape loader, and all intermediate working areas, is such as not to interfere with those memory areas used during the correct execution of all possible test cases. Upon completion of a test run on a single processor, an attempt to reload all processor MTR's (beginning with Basic Test #1) is made. On a two-processor system, manual intervention (the reversal of processor designations) is necessary before each programmatic reload.

GCR starts out with the initialization of the run. This includes the interrogation for the presence of the second processor. Memory module interrogation starts with module 0 and goes up. Interrogation is terminated at the first missing module. Changes in the system configuration at any later time within this pass are not recognized by the program. The results of the interrogation are summarized in the header message printed on the supervisory printer.

---

[1]Description of the INCW constructed by SFT is given in the Appendix. Other control words are identical, but SFI does not normally create all six words.

2.4.1 Con't.

Example:

BEGIN MTR, PROCESSOR 1 & 2, MEMORY MODULES USED 0123

Test data from tape are read next. Each tape record contains data for up to 34 test cases. Processing of them is sequential. The configuration of the tape read buffer is explained in the Appendix. Each test case has an identification word and a definition word. Identifications range from *01001.0 to *19017.0 (ascending but not necessarily consecutive). The relationship between the identification numbers and the corresponding operators tested is charted in the Appendix.

Based on the contents of its definition word, a test case is checked to determine if the current processor and memory configuration requires its skipping. The make-up of definition words is described in the Appendix.

The data of a test case accepted for execution are transferred from the tape buffer to the PRT of GCR. The expected output control words are transferred into a tank area also, followed by the contents of after-test memory words, if such are specified. Description of the tank area is given in the Appendix.

Initialization of the program to handle a particular test case is carried out according to definition word specifications. The code number of the operator to be tested ("operatorX") is put into the proper word and syllable position. A series of program switches are pre-set to properly handle the setting up and the evaluation of the test case.

The input control words are transferred into the test stack specified by the test case. Memory settings, if any, are performed. All interrupts are cleared. If so specified by the test case, a dummy I/O finished interrupt is subsequently forced. The INCW of the test case is put on the top of the GCR working stack.

Execution of the Initiate For Test operator is the next step. After the contents of IRCW, ICW, ILCW, B, and A have been distributed from the test stack, control is transferred to operator X for a limited number of pulse times. Whether through the occurrence of a SFT or an SFI operator, after-test control words are constructed and stored

2.4.1 Con't.

in the test stack (in the reverse order). The resulting INCW is stored in R+8, and control is transferred back to GCR to evaluate the results.

A test case may fail if at least one of its results is incorrect. The interrupt address, any of the output control words, or any of the output memory words, may differ from the predicted values(s).

Evaluation begins with the interrogation of interrupts. Even if no interrupt has occurred, GCR switches back to its own working stack to avoid using any cells above the test stack.

The occurrence of a specific interrupt (or the lack of any) is remembered through the storing of the corresponding interrupt address. If the first interrupt were to be ignored, interrupt interrogation would again take place.

Comparison of the obtained interrupt address against an expected address is performed only if such a test was called for. Although one and the same test case may be repeatedly executed, only that interrupt address (correct or incorrect) is remembered which is associated with the first erroneous pass of the test case. Interrupt errors on subsequent passes result in counting up the error tally without comparing control or memory words.

The comparison of control words is performed word by word. Control words found in error are stored over the corresponding expected values in the tank, provided each of them was found during the first erroneous pass through the test case. The first comparison failure encountered during any subsequent pass results in the counting up of the error tally without any further comparison.

The comparison process for output memory words, if such are specified, is identical to the process described for the control words.

If no error (interrupt or comparison) was detected during any pass of the same test case, it is repeated altogether 25 times from the setting up of the test attack with no printed output. The next case is then taken from the tape input buffer.

2.4.1 Con't.

Detection of an error is indicated on the supervisory printer upon the completion of the first erroneous pass through a test case. Example:

*12005.0 I=00 C

The test case identification number is given to reference the test case entry in the Error Reference Manual. I=00 signifies that an interrupt was expected but did not occur. C means that at least one of the control words (or output memory words, if specified) did not correctly compare.

After a failing test case has been executed 25 times, a second line of supervisory printout appears. Example:

ET=0031

This printout indicates that out of 25 trials, this test case failed all 25 times (octal 31 = decimal 25). Should this second line indicate instead that not every trial failed, the failure is considered intermittent. If the first line indicated the presence of comparison failure(s) during the first failing pass (as in the current example), an intermittent failure causes further printing. All output control words and memory words specified by this test case are dumped from the error tank in 16-position octal words, one under the other, even if some of them compared correctly.

Following the second line of print (or the tank dump), the program is stopped by a conditional halt (CHP). The test case identification word is held in the B register. Looping without comparison - or simply continuing - requires manual intervention (see Options, 2.4.3). If the conditional halt is inactive, the program loops on the test case indefinitely, comparing the results on each trial and printing out the error tally after every 25 trials. The error tank is not altered and not dumped.

Consecutive test cases from one tape read to the next are handled in similar fashion. Recognition of the condition that the last Processor 1 test case has been processed initiates one of the following supervisory messages:

2.4.1 Con't.

-END P1; LAST TEST CASE 18253.0; RELOAD

or

-END P1; LAST TEST CASE 18253.0

The first message occurs on a one-processor system, the second one on a two-processor system. Reloading (single processor) is automatic.

On a two-processor system, the test cases created for the checkout of the P1-P2 interface are run next. Upon their completion, the following message is typed:

-END P2; LAST TEST CASE 19017.0 AT P2 CHP, SWAP PROCESSOR DESIGNATIONS--PUSH SINGLE PULSE ON NEW P1

The program then idles until both manual actions have been performed. The completion of the manual actions is acknowledged by the following message:

MTR RESTART FOR NEW P1

Automatic reloading is initiated at this point.

The total execution time of all processor MTR's is less than 15 minutes for one processor with eight memory modules. This assumes cycling on each general test case 25 times.

2.4.2 Exceptional Operating Conditions: The preceding section described those GCR actions which correspond to the most favorable conditions, i.e., all assumptions listed in 2.2 hold. Some of the possible violations were given special consideration in GCR.

Supervisory printer not available: The I/O subroutine loops on the first print attempt, waiting for the unit to become ready. When the supervisory printer is not in local but is simply not available, all messages must be inhibited. Resetting the print switch (see operating instructions) accomplishes this. The loop is left and is not entered again. Note that the only information which is not lost this way is the identification word of a failing test case displayed in the B register at the test case error halt.

Tape Read Problems: The usability of any one of the I/O channels in MTR does not alone guarantee error-free tape read operation.

2.4.2 Con't.

Tape Read Parity is continuously checked. Up to four retries (position backward-read) are provided upon detection of an erroneous read. After the fifth read attempt, the next record is read, and the skipped record is handled as a missing record.

Missing Tape Records are detected through the continuity checking of the records, and are flagged on the supervisory printer. The missing record is expressed in terms of missing test cases.

Example:

*MISSING TEST CASES FROM *09025.0; CONTINUE FROM *09172.0

The skipping of any number of consecutive tape records may be implied by a single message. The flag is printed after the first correct read.

Read Word Count errors are handled as read parity errors.

Unit Not Ready is flagged with the typed message *READY MTR TAPE, followed by a conditional halt and automatic retry. The unit designation is copied from the MTR tape loader PRT after the execution of the basic tests. Therefore, unit designation is not printed.

End-of-File is an error condition in GCR, since the run is terminated upon reaching certain information on tape. The corresponding message is *SPURIOUS END-OF-FILE, followed by the unconditional automatic reload of MTR.

Unit Busy conditions of any type cause the retrying of the I/O operation in question.

Other Result Descriptor Errors are ignored (or handled in one of the already described ways). Consult the program flow chart and/or listing for details.

Program Hang-Up and Blow-Up Conditions: In spite of the care taken in the construction of the test cases, some may, in case of hardware failure, cause the hanging up of the program in the middle of a test case. This type of error detection is permissible, since the register contents and all relevant memory words (see operating instructions) can be examined. Thus, recreation of the test case for single pulsing or immediate troubleshooting (based on the display) may proceed.

2.4.2 Con't.

Any hang-up within GCR itself (not traceable directly to a particular test case) means either that logic areas essential for GCR fail or that a test case failure causes the loss of program control. In this case the path leading to the hang-up condition has to be single-pulsed or single-stepped from the initiation of that test case which immediately precedes the hang-up. The identification of the test case under execution is available from memory (see operating instructions). After looking up the ID, the user is advised to reload the MTR tape and verify the basic logic. It is highly recommended that he repeat the GCR run also, up to the previous hang-up point. This is to find and eliminate all failures possibly developed during the previous run. The user may now prepare for single pulsing. This time he is interested in executing the failing test case only. Normally, a simple way of doing this is the manual setting up of the test case input control words and memory cells as specified in the ERM printout of the test case, followed by pulsing through an Initiate For Test operator (See 2.5). In most cases, preparation for the single-stepping (pulsing) may be done faster and safer if the automatic setting-up machinery of GCR is used instead. A search provision in GCR, useful in getting directly to a specific test case, is explained next.

Keyboard-Initiated Branching to any desired test case is possible from any point, provided GCR is in progress and has passed the printing out of the program header message. The keyboard interrupt first initiates a supervisory printout to indicate where the break-in occurred. Example:

*12345.0 HEADED LAST RECORD, TYPE IN REQUIRED ID

The program then activates the keyboard and idles until the typing in of the desired test case identification is terminated by the depression of the end-of-message key. Depressing it without an input message (same as all zeros) results in a restart from the first test case in the same tape record. This feature may be used for monitoring the progress of GCR. Depressing the error key before termination initiates the above message and reactivates the keyboard. Test case identifications outside of the *01001.0 - *19017.0 range have the same effect. Calling of a non-existent test case causes a restart from the test case with the first higher legitimate identification.

Note that only the last seven characters of the (first) input word are compared during the search.

2.4.2 Con't.

In searching for a test case to be single pulsed, input message termination should be delayed until a conditional halt operator has been manually inserted in GCR just before the initiation of the test case to be single pulsed (see operating instructions). This is necessary to allow the manual setting of the clock mode switch to "single pulse" just before the IFT is executed. It is good practice to ascertain (upon reaching the halt or operator X) that the input conditions in the processor and in the ERM entry for the particular test case are the same.

Program Blow-Up conditions should be handled just like hang-up conditions: through single pulsing (stepping), the source of failure must be tracked down. Difficulty arises when memory is wiped out and thus the identification of the failing test case cannot be looked up. The use of single pulsing must then be delayed until localization of the test case is achieved.

Locating an "explosive" test case requires a separate MTR pass, with a search initiated for the first test case of the tape record containing the "explosive" case. The identification of each test case in the record must then be manually looked up at the (manually inserted) halt before the initiation for test is executed.

NOTE: The first test case of the record can be easily found in memory if a manual backward read is executed after blow-up. Should the tape run away at the time of the program blow-up, the record can still be found by halting after each tape read instead of before each test case (see operating instructions).

2.4.3 Options

Single pulsing of a consistently failing test case is the most direct approach to troubleshooting. Studying the corresponding ERM information before single pulsing is advantageous in a simple test case and indispensable in a special test case. Looping with scoping is the approach for finding the cause of intermittent or frequency-sensitive failures. The two approaches have common GCR provisions. If A02FF is set at the test case error halt, a scoping loop on the test case is forced. There are two conditional halts reached alternately in the loop, one shortly before IFT (LTS, LOD, IFT), the other one is the test case error halt itself. (See operating instructions.) The former one is coded in to facilitate single pulsing through the use of GCR's automatic test case set-up machinery. The second halt is only used for break-out from the scoping loop (manually reset A02FF; to leave the test case, set A01F). Of course, the operator switch has to be OFF during the scoping itself.

2.4.3 Options (Con't.)

NOTE: Although there is no comparison of the results in the scoping loop, it is still lengthy. Furthermore, it may use the operator which is being tested. Finding the proper point for synchronizing requires careful consideration on the part of the user. This is especially true in special test cases.

Although troubleshooting should normally begin upon detecting the first case failure, the user may bypass any failing test case. To do this, he must set A01FF at the test case error halt and depress the single pulse button. A02FF must be OFF.

The running time of a pass through GCR can be varied between wide limits through the varying of the number of times each test case is executed. The repeat tally limit is loaded as 31 octal = 25 decimal (see the operating instructions for its memory location). Values ranging from 0001 to 7777 may be used. Manual modification of the tally value is permitted at any time. The shortest complete pass (RPTL = 0001) takes less than $2\frac{1}{2}$ minutes for one processor.

NOTE: Using the value zero results in the execution of each test case once. Every error is recognized as intermittent.

Other operating options may be thought up by a programming-oriented user. He will find the detailed flow-chart and the program listing (with cross reference printout) useful. Note that the GCR program listing is bound with the basic test documentation.

2.4.4 Operating Instructions:

1) Make sure that all equipment requirements are fulfilled:

   a) Verify that the supervisory printer is in REMOTE. The absence of the supervisory printer is an exceptional condition (2.4.2).

   b) Verify that the designation numbers of all memory modules in REMOTE are continuous. If this condition is not met, the automatic selection of test cases will erroneously cause the execution of unavoidably failing test cases.

2.4.4 Operating Instructions (Con't.)

c) Verify that the operator switch is ON for the second processor, if such is available.

d) Verify the correct working of the basic tests. They must be run from tape (high density), with the operator switch and the inhibit timer switch ON. Inoperative (low order) I/O channel(s) must be in LOCAL. All other switches must be in their normal position. Leave the switch setting unaltered for GCR. GCR loading is automatic only if the basic tests were loaded using the MTR tape loader deck. If the basic tests were loaded manually, manual GCR loading is required. Cell 313 must contain a tape descriptor with the correct unit field. The steps of the manual loading are the same as those for the basic tests. See 2.5.2 in the basic test specifications.

2) Follow supervisory messages. After the header printout, a correct pass through GCR should produce only one message at the end of the Processor 1 test cases and, if the second processor is on line, one at the end of the P1-P2 test cases. Verify that the last P1 test case is *18253.0 and the last P1-P2 test case is *19017.0. If terminated earlier, GCR must be re-run. The same applies if some of the test cases were skipped (see 2.4.2, tape read problems).

3) MTR reloading from Basic Test #1 is automatic only if the MTR tape loader was used and it has not been destroyed during the whole MTR pass. If the system includes a second processor, manual intervention, as called out on the supervisory printer, must precede MTR reloading, this time on the other processor.

4) Testing may be terminated at any time after a correct pass on each processor has been achieved.

5) A test case failure printout always starts with the identification of the test case. Example:

```
*12345.0 I=65 C
 ET=0031
```

At the subsequent conditional halt, the user is expected to begin troubleshooting. His choices are described in the options (2.4.3). The user is expected to know how to use the Error Reference Manual (see 2.5).

2.4.4 Operating Instructions (Con't.)

6) Hang-up and blow-up conditions are considered exceptional conditions. See 2.4.2 for useful hints.

7) The following is the list of the important program parameters:

LOCATION (OCTAL)	SYLLABLE	NAME	USE
01120		IRT*	Start GCR; S = 100
01260		E13*	If loader is in memory, reload MTR; S = 100
01206	1,2,3	-	Replace NOP with CHP to stop after each tape read
01500	1,2,3	-	Replace NOP with CHP to stop before a test case stack is first set up
01532	3	-	Replace NOP with CHP to stop before IFT of every test case
01532	0	-	Scoping loop halt before IFT
02101	3	-	Test case error halt for manual entry (A01F or A02F); B contains test case ID
00411		-	ID of test case currently being executed
00412		-	Error type indicator (Example: I=00 C)
00413		-	Error tally (Example: ET = 0031)
00414		-	Test case definition word; see Appendix
00415 thru 00435		TANK*	Error tank; see Appendix
00437		PRSW*	Print switch; if cleared, no message is typed
03012		DATA *	Base address of tape read buffer; see Appendix
00542		PTR *	Pointer; points to buffer location of current data
00646		RPTL *	Repeat tally limit (0031); change to alter the running time of GCR.

2.4.4 Operating Instructions (Con't.)

LOCATION (OCTAL)	SYLLABLE	NAME	USE
00001 and 00517		-	ID of test case requested by search
01534		IIR *	Interrogate interrupt; start of test case evaluation
02251		IO *	Start of I/O subroutine
00022		-	Timer interrupt; irrecoverable error, loop on CHP
01533		-	No IFT; irrecoverable error, loop on CHP
02327		-	CHP; Ready MTR tape, depress single pulse button

Parameters common to MTR Tape Loader and GCR:

00313	TRWD	Tape rewind descriptor with unit designation
00330	TRTAL	Test routine tally; 5 in GCR, 0 for reload
10410	START	MTR tape loader entry point

2.5 Use of the Error Reference Manual (ERM)

2.5.1 General: Familiarity with all test and field documents and ingenuity in the use of instruments are still required from the user for efficient troubleshooting. This section describes auxiliary documentation, created chiefly to help find and identify failing logic areas.

Detailed documentation associated with each of the over 15,000 general test cases is available in the Error Reference Manual. ERM is bound into two physical volumes, each containing a header page, a test case sample sheet, and about half of the printed material. The first volume also contains an index of all ERM pages. Provisions to prepare programmatically a delete page index and an add page index have also been made in the ERM maintenance procedures. All pages of the index have the same revision level, that of the latest ERM update run. Each ERM page has the same revision level as the ERM update run introducing the latest change to the particular page.

ERM sheets contain photo-reduced information; up to four computer-produced print pages per ERM sheet. This means that up to four different page numbers may be printed on each ERM sheet, two on the front and two on the back. Page numbering is important only for maintaining ERM. Test cases are referenced by identification number.

The sequence of test case information on the ERM pages follows the execution sequence of the test cases. This corresponds to ascending test case identification numbers.

Unlike in the basic test error documentation, there are no erroneous conditions included in the test case information. Only the expected (correct) output conditions appear. The input conditions necessary to set up a test case are naturally included. Valuable references to various test and field documents also appear along with some miscellaneous information.

2.5.2 Test Case Information Format: There are up to twenty test cases documented on a single ERM sheet, ten on each physical page. There are two page numbers printed on each physical ERM page, one on the top and one in the middle of the page. In the following, the information associated with a single test case is explained. It is essential that the reader look up and study now the sample sheet (after the header page) in one of the ERM volumes. The sample page contains a fictitious test case. The encircled numbers correspond to the numbers listed below.

2.5.2 Con't.

There are two boxes outlined by broken lines. These lines were drawn in for explanation only. Both boxes contain the same information. The left, longer box contains test information grouped more or less as the various values appear in the neons of the processor display panel. The right, compact box contains the same values as they appear packed into control words (octal representation). These represent the only information duplication. Note, however, that the L value appearing in the input IRCW is counted up during the P to T transfer. It is best to rely on the automatic (GCR) initialization before single pulsing, lest the wrong input conditions be manually set in the registers.

To simplify the use of this section for direct referencing, a field by field description of the sample test case information follows.

1. The six-digit test case control number is explained in the Appendix. It is of no concern for the MTR user, since it is used only in test case construction and updating.

2. This date-field is of no interest to the MTR user.

3. The T-register contents appearing here usually refer to the operator being tested, also called as "operator X." There are several exceptions. In some cases, the inhibition of a certain function in "Op X" is tested with the use of another operator. In this case field (7) contains further information about "Op X." Illegal code may be used sometimes to test (usually) inhibit logic. In special test cases, the first operator of a string is very seldom the operator being tested.

4. The mnemonic code is derived from the engineering abbreviations used for operator levels. The fourth character, L, is omitted. This set of mnemonic codes is used in all MTR program listings also. ILLEGAL (or ILL) stands for an undefined code.

5. This number references the operator flow of the operator in T. Note again that this is not necessarily the operator tested. See field (7).

2.5.2 Con't.

6. The number-of-pulses field is -01- for most simple test cases. The values -03- and -06- refer to pulse counts for write and read, respectively, if the system works with 6 usec memory modules. In case of an all-4-usec memory, -02- and -05- are to be thought of by the user. These are the pulse counts necessary to obtain the output register values after all input conditions have been set up by the IFT operator. Additional pulses are necessary to complete the storing of output memory words. This field is left blank for special test cases.

7. The comment field normally contains a reference to the specific function which is primarily tested. When the T-field does not contain the operator tested, its mnemonic code and flow numbers are printed here. In special test cases, a string of mnemonic codes may be given instead, always preceded by the character pair SP (special). If the operator string is too long to be represented in the available remark space, the purpose of the test is summarized in place of the mnemonic codes.

8. The logic book (page and line) reference may sometimes be useful during the troubleshooting process. Note that this reference is not complete by any means. The test case may detect a failure not related to the logic area referenced.

9. It is the (ERM) test case identification number by which a failing test case is referenced on the supervisory printout or in the B-register. The asterisks preceding this field are only to help the user in locating the field. The blank after the second position is a convenience in case the user is interested in how the test cases were grouped by operator (see Appendix). The last digit is an insert digit preceded by a decimal point. The insert position is provided to allow inclusion of further cases in the sequence of tests if necessary because of logic changes.

2.5.2 Con't.

10. This field is blank except in rare instances. Example: When more than one interrupt is required by some test cases, one of them is set up by GCR before the IFT, upon the request of the test case definition word (see Appendix). This interrupt is a dummy I/O finished interrupt (other interrupts, if any, are set up after the test case is initiated). Since the I/O finished interrupt address is determined by the lowest I/O channel in REMOTE, GCR "forces" interrupt address 27 in all four cases to simplify the checking of the output conditions.

11. Even if not all of them apply, all memory module designations between 0 and that of the highest actually used module are printed here as a continuous field (in accordance with the restriction that the designations of memory modules in REMOTE must be contiguous). Test cases specifying higher memory module(s) than available in the system during a particular GCR run are automatically skipped by GCR during that run.

    NOTE: Many test cases may only list MOD 0. Module 1 must still be present to run GCR correctly.

12. This field is rarely used. When it is used, the absence of the module listed is required to correctly execute the test case (invalid address interrupt checking).

13. Before-test (input) conditions of the flip-flops and registers appear immediately above the corresponding expected after-test (output) conditions. Register contents are given in octal, including the contents of J. Flip-flop values are printed in binary. Register names and most flip-flop names are self-explanatory. Note that some registers (E, TM, interrupt, P) and some flip-flops (HLTF, EIHF, PROF, TROF) are completely missing from this field as well as from field (18). The contents of these are not set up from or stored away in control words. Note further that VARF(after) may contain the value of SENL (set E not level). Also, the output values of MROF, MWOF, and CCCF are not listed because they become zero upon the termination of all test cases.

2.5.2 Con't.

14. In most simple test case prints, this composite field is not used. When used, up to three location-information pairs (headed by LOCN and INFO, respectively) may be specified on a single line. Up to four full lines (12 entries) may be used to specify input memory requirements.

    The most common use of this field is by special cases. Control words for a second initiation as well as long strings of operators may be specified. It is a must to jot down the mnemonics in the string (unless already listed in field (7)) and analyze the coding before troubleshooting is begun.

15. The expected after-test (output) location-information pairs are aligned vertically with the input memory specification. The evaluation routine compares the contents of the cells specified under LOCN with the corresponding INFO word. Since there is no separate header line to separate the output conditions from the input conditions, care must be taken to find the beginning of the output specifications. The first output line appears aligned with the output Interrupt Loop Control Word. Sometimes this word may be missing and only an asterisk on the far right is printed (see (19)).

16. The interrupt test field is usually blank. This indicates that checking for the presence or absence of any interrupt is not part of the test case. NO INTERRUPT EXPECTED indicates that the absence of all interrupts must be verified after the execution of the test case.

    INTERRUPT EXPECTED XX indicates that the occurrence of a specific interrupt (with address XX) is checked after the execution of the test case.

17. This message occurs very rarely. If the test on the second interrupt fails (see (16)), then one of the following failures occurred:

    a) If a second interrupt was expected but none was found, one of the interrupts either did not occur or was cleared out with the interrupt of higher priority.

    b) If a second interrupt with a wrong address is found, either the priority of the interrupts was not properly sensed or a third interrupt occurred in place of, or in addition to, the second one.

2.5.2 Con't.

18. Test case control words, output below the input, appear here. Each of them is properly labeled but their composition is not revealed in the printout. This field is rarely used in troubleshooting solid failures. In case of intermittent failures, the user is expected to do his own interpretations of the various fields where mis-matches occur between the expected output control words and those from the error tank dump.

    Note that the before-test (input) control words are held in the test stack as follows: the contents of A is held in S(before)+2 if the test case is to be initiated to word mode with CCCF=1 (it is held in S(before)+1 in any other case). This feature of the MIR hardware has been provided to test the occurrence of stack push-down actions from A or B. The skipped S(before)+1 contains a word different from both A and B. The lack of an expected push-down from A or B (S+1 alone occurred) can then be detected from the wrong contents of S(before)+1 = S(after).

    Note that the storing away of the output control words always begins in cell S(after)+1.

19. Asterisks instead of 16-position output control words are printed only if comparison on the SFT-type control words is to be inhibited. This is the case when SFI rather than SFT is the expected test case terminator. Comparison of (some of) the SFI-produced output control words may be requested by the test case in the form of output memory word specifications. Note that the sample erroneously depicts both the after-test control words and the asterisks.

2.5.3 Final Hints: There are two basic ways a test case may fail. Either some expected actions do not occur or some spurious actions occur (of course, any combination of the two is possible). It is easy to establish what the conditions are through a visual comparison of the erroneous display (or control words) against the print of the expected values.

The lack of expected actions is an indication that the flow of "operator X" and possibly the logic line reference apply. Finding the source of error may be relatively easy.

2.5.3 Con't.

The occurrence of spurious actions indicates a lack of inhibition(s), probably open AND leg(s).

Special attention is to be given to finding the proper sync for the scoping of intermittent failures (see 2.4.3).

III APPENDIX

3.1 Initiate Test Control Word

Alternate fields are shaded.

(48)	(45)	42	39	36	33	30	27	24	21	18	15	12	9	6	3
(47)	44	41	38	35	32	29	26	23	20	17	14	11	8	5	2
46	43	40	37	34	31	28	25	22	19	16	13	10	7	4	1

Bits 1 thru 15 - S reg., pointing to ILCW.

Bit 16 - - - - - CWMF.

Bits 17 thru 20 - J reg.: J01F thru J08F. Transfers to and from TM 1 thru 4.

Bit 21 - - - - - NCSF. Transfers to and from TM5F.

Bit 22 - - - - - CCCF. Transfers to TM6F (IFT only).

Bits 23 thru 28 - Z reg., Z01F thru ZBF.

Bits 29 thru 34 - Y reg., Y01F thru YBF.

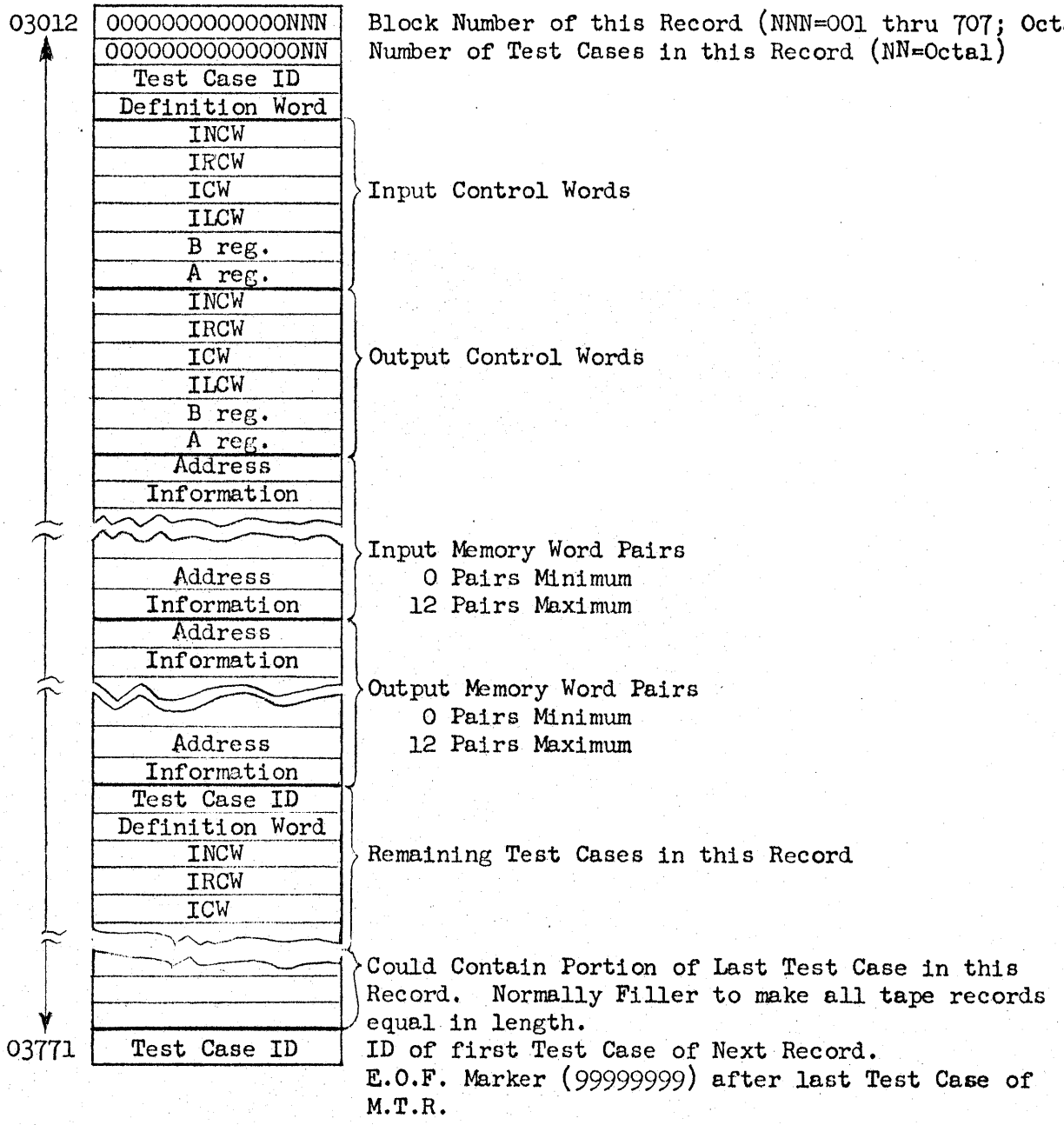
Bits 35 thru 43 - Q reg., Q01F thru Q09F.

Bit 44 - - - - - MWOF. Transfers to TM7F (IFT only).

Bit 46 - - - - - MROF. Transfers to TM8F (IFT only).

Bit 45 = 0
Bit 47 = 1
Bit 48 = 1 } ID Bits for all Control Words

3.2 Tape Input Buffer
(Cells 03012 thru 03771)
(Always 760 Words)



3.3 Grouping of Operators by First Two Digits of Test Case ID

01	02	03	04	05	06	07	08	09	10
SEC	BSD	INI	TRF	AD1	AD2 CCX	DV1	FCX LØE	MU1	TFB
BBC	BSN	LØD	FXS	AD2	BEQ CFX	DV2	FFX LØN	MU2	VFI
BBU	CFE	LTS			BGA CSD	DV3	IPS LØØ	RFB	ZBD
BFC	CFL	PIØ	AD1		BGE CSN	DV4	ISD MDA	RJP	ZBN
BFU	DIA	SFI			BLA CSS	ECM	JBC MDV	RNM	ZFD
BSD	DIB	SFT			BLE DEL	FCX	JBU MSN	RSP	ZFN
	DUP	TRF			BNE DV1		JFC MSØ	SFB	
	EXC						JFU MSP	SSF	BEL
	IFT						LLL MU1	SU1	CFJ
	IIN						LØA	SU2	CJØ
	IND							TFB	CLR
	INI								CRJ
									ENL

11	12	13	14	15	16	17	18	19
ENL	FSX	ØCØ	SDA SNE	TAN TPD	IØR	INTER-	SPECIAL	P1-P2
FAD	FWJ	RDA	SDP SSA	TBZ TSD	ØCS	RUPTS.	CASES	
FAS	ICØ	REB	SEB SSP	TEB TWD	PRE			
FAX	INT	REC	SEQ STA	TEQ TZD	PTØ	See	See	See
FSD	JØL	REJ	SET STD	TGE	RDT	Note 2	Note 3	Note 4
FSS	ØCØ	RPA	SGE STP	TGT CHP	REW			
FSU		RSA	SGT STS	TLE CØM				
FSX		RSD	SLE TAN	TLT DCS				
		RSS	SLT	TND HP2				
		SBD		TNE IØR				
		SBS						
		SDA						

Note 1 - Basic Operators - Groups 01, 02, 03 and 291 of 04.
Word Mode - Groups 04, 05, 06, 07, 08, 09 and 411 of 10 (Except Basic Op.)
Character Mode - Groups 10, 11, 12, 13, 14 and 773 of 15.
Control State Operators - Groups 15 and 16 (Except Basic Op.)

Note 2 - Test all Processor Dependent Interrupts and Interrupt Priority Except for a few which are intermixed with the Operators.

Note 3 - Special Cases. CCCF=0. Test those functions which cannot be tested by a slice test case (CCCF=1).

Note 4 - Tests communication between Processors 1 and 2. (Will run only on a two Processor System.)

Note 5 - All groups have approx. 900 test cases except Groups 16, 17, 18 and 19.

3.4 Grouping of Test Cases by Development Control Numbers

Control Numbers (see field 1 on the ERM Sample Page) were created for the controlling of the test cases during the development phase, when the execution sequence and the corresponding (ascending) test case Identification Numbers had not been assigned. The Control Numbers still remain the means of test case identification within the maintenance of the test cases (see B5500 MTR Support Specification).

The first and second digits of a Test Case Control Number represent the programmer and the logician who developed the test case. The next four digits represent the sequence number consecutively assigned during the development by each logician.

3.5 Test Case Definition Word

		42		36		30	27	24	21			12	9	6	3
	44	41		35		29	26	23	20			11	8	5	2
46	43	40	37	34	31	28	25	22	19			10	7	4	1

▨ Not used (always 0)

Bits 12 thru 1 - T reg. setting of operator being tested (First operator if test uses more than one operator).

Bits 24 thru 19 - If 00 - Check that no interrupt occurs. (None should occur)
If 01 (octal) - Do not check for interrupt.
If not 00 or 01 (octal) - Interrupt should occur.
Check for interrupt specified at this address.

Bits 27 thru 25 - Highest memory module required for this test case (octal, designations range from 0 thru 7)

Bits 31 thru 28 - Number of Output Word Pairs (octal)

Bits 37 thru 34 - Number of Input Word Pairs (octal)

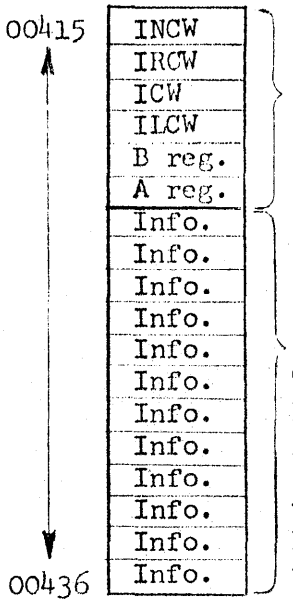
Bits 42 thru 40 - Designation of memory module which must be absent for this test case to be tried (octal)

Bits 44 and 43 - Equal 0 - Compare (Output) Control Words.
Equal 1 - Do not compare Control Words.
Equal 2 - Force dummy I/O finished interrupt for test case.

Bit 46 - Equal 0 - This test case is for P1 only.
Equal 1 - This test case is for P1 and P2.

3.6 Error Tank

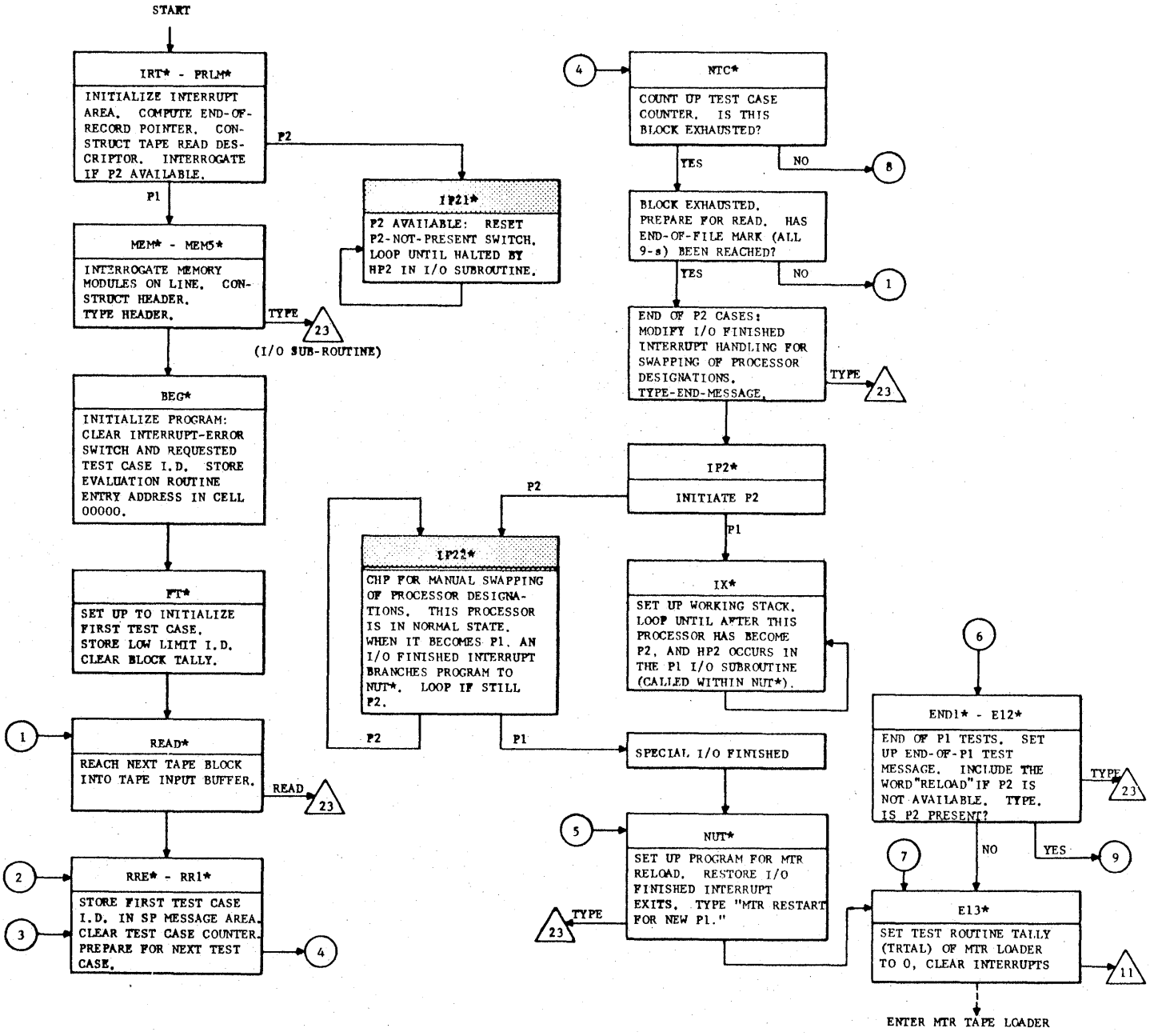
(00415 thru 00436)



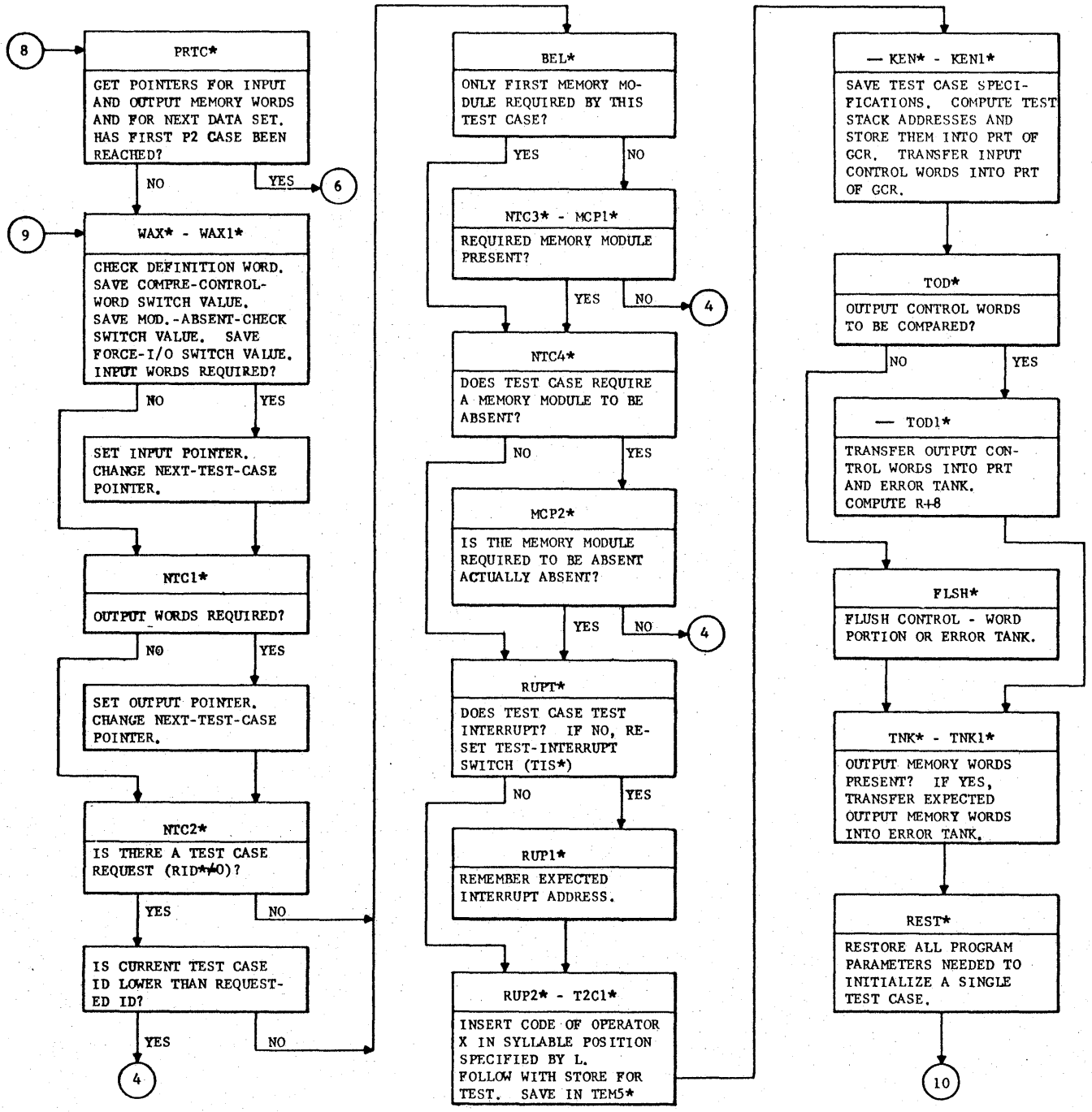
Output Control Words. Only those which failed to compare will differ from the control words on the ERM. If control word comparison is suppressed, this area will be all zeros.

Output Memory Information Words (Min. 0 - Max. 12). The corresponding addresses are to be obtained from ERM. The sequence of tank words corresponds to the left-to-right sequence of LOCN-INFO pairs, up to three per ERM output line (the first such line is aligned with the print of the output ILCW).
If the Test Case which failed has fewer memory words than a previous Test Case, residual information words are also present. They do not pertain to the Test Case which failed.

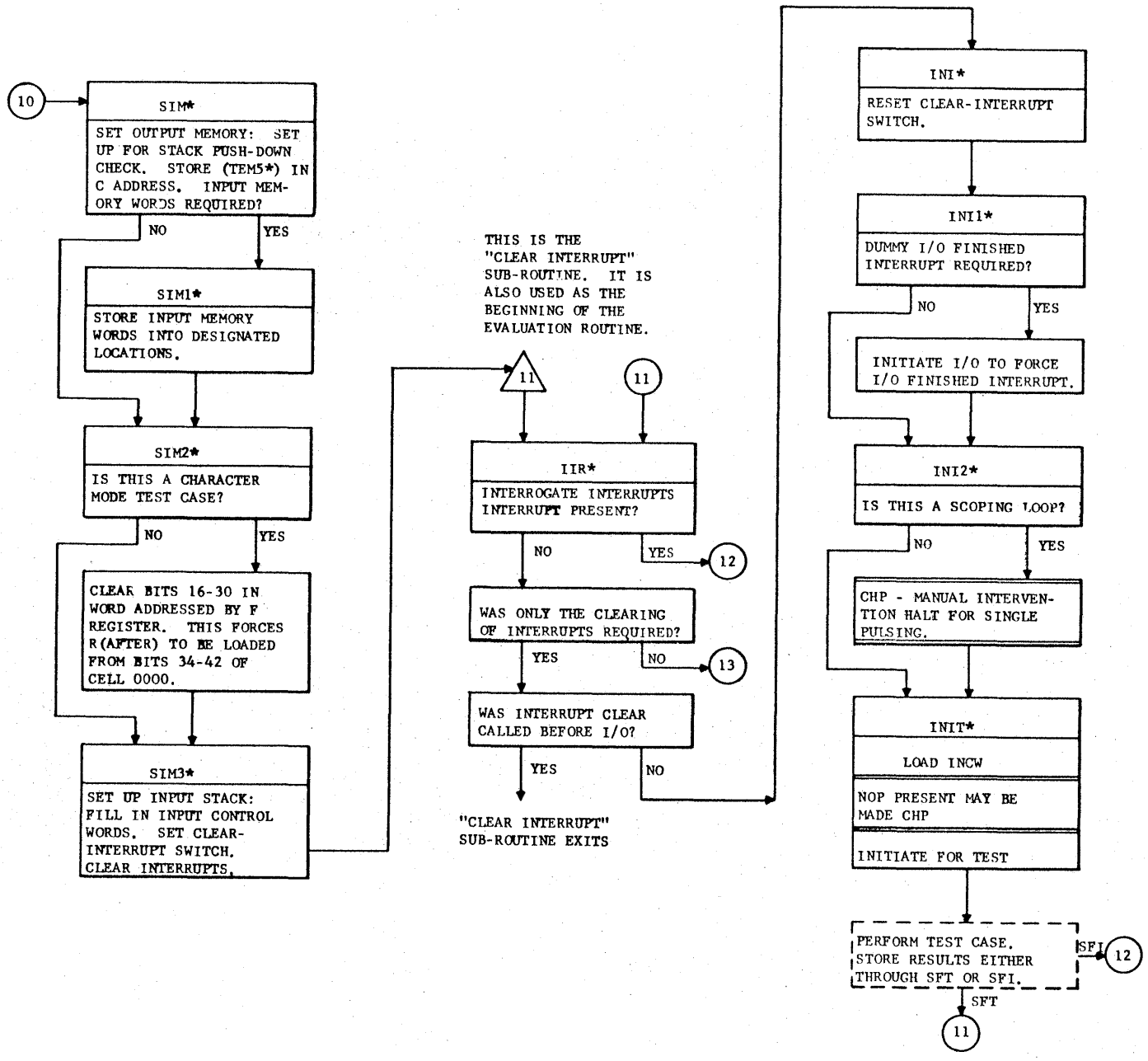
3.7 General Control Routine Flow Chart



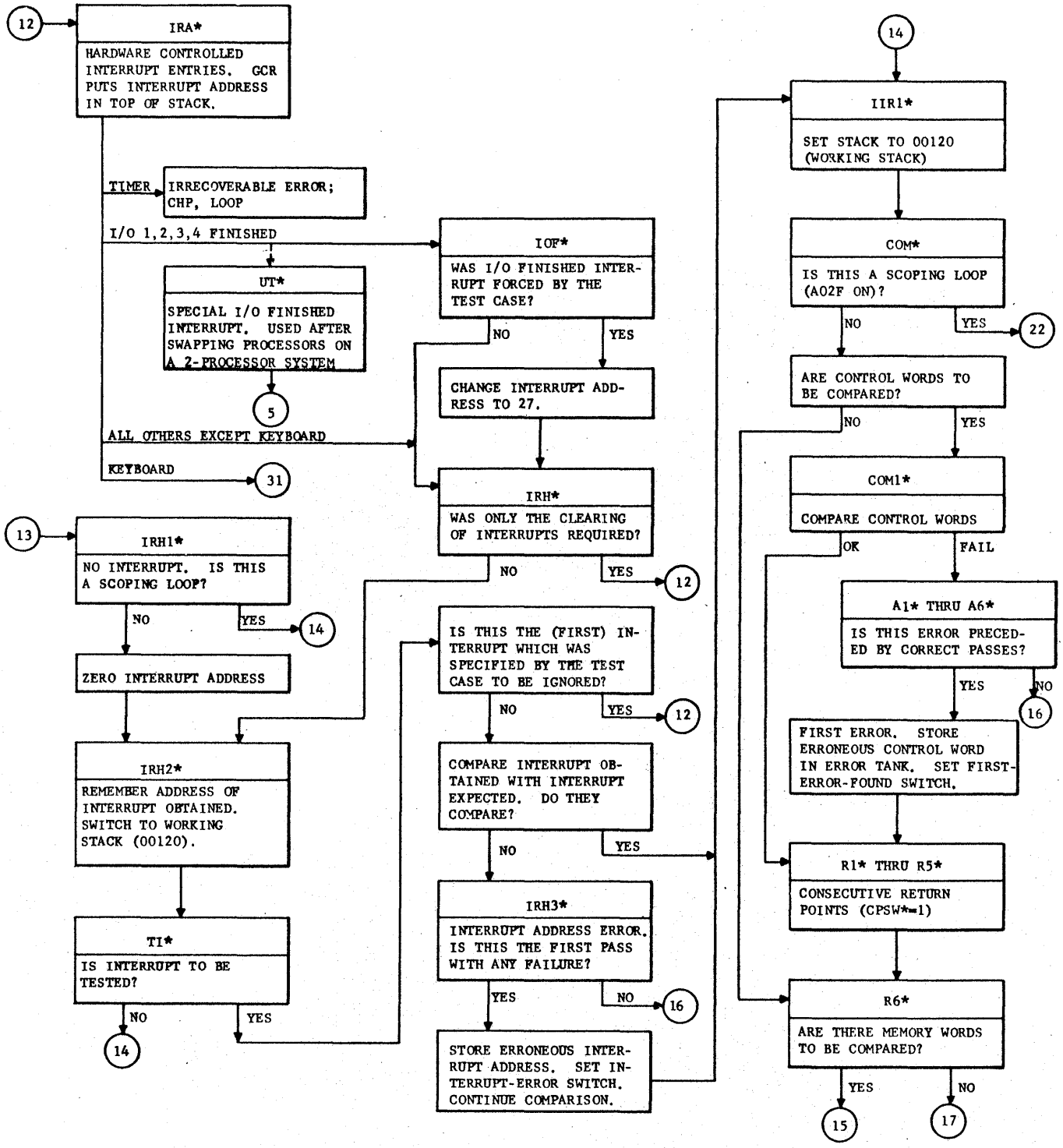
3.7 Con't.


8
PRTC*
GET POINTERS FOR INPUT AND OUTPUT MEMORY WORDS AND FOR NEXT DATA SET. HAS FIRST P2 CASE BEEN REACHED?
NO
YES
6
9
WAX* - WAX1*
CHECK DEFINITION WORD. SAVE COMPRE-CONTROL-WORD SWITCH VALUE. SAVE MOD.-ABSENT-CHECK SWITCH VALUE. SAVE FORCE-I/O SWITCH VALUE. INPUT WORDS REQUIRED?
NO
YES
SET INPUT POINTER. CHANGE NEXT-TEST-CASE POINTER.
NTC1*
OUTPUT WORDS REQUIRED?
NO
YES
SET OUTPUT POINTER. CHANGE NEXT-TEST-CASE POINTER.
NTC2*
IS THERE A TEST CASE REQUEST (RID*≠0)?
YES
NO
IS CURRENT TEST CASE ID LOWER THAN REQUESTED ID?
YES
NO
4
BEL*
ONLY FIRST MEMORY MODULE REQUIRED BY THIS TEST CASE?
YES
NO
NTC3* - MCP1*
REQUIRED MEMORY MODULE PRESENT?
YES
NO
4
NTC4*
DOES TEST CASE REQUIRE A MEMORY MODULE TO BE ABSENT?
NO
YES
MCP2*
IS THE MEMORY MODULE REQUIRED TO BE ABSENT ACTUALLY ABSENT?
YES
NO
4
RUPT*
DOES TEST CASE TEST INTERRUPT? IF NO, RESET TEST-INTERRUPT SWITCH (TIS*)
NO
YES
RUP1*
REMEMBER EXPECTED INTERRUPT ADDRESS.
RUP2* - T2C1*
INSERT CODE OF OPERATOR X IN SYLLABLE POSITION SPECIFIED BY L. FOLLOW WITH STORE FOR TEST. SAVE IN TEM5*
— KEN* - KEN1*
SAVE TEST CASE SPECIFICATIONS. COMPUTE TEST STACK ADDRESSES AND STORE THEM INTO PRT OF GCR. TRANSFER INPUT CONTROL WORDS INTO PRT OF GCR.
TOD*
OUTPUT CONTROL WORDS TO BE COMPARED?
NO
YES
— TOD1*
TRANSFER OUTPUT CONTROL WORDS INTO PRT AND ERROR TANK. COMPUTE R+8
FLSH*
FLUSH CONTROL - WORD PORTION OR ERROR TANK.
TNK* - TNK1*
OUTPUT MEMORY WORDS PRESENT? IF YES, TRANSFER EXPECTED OUTPUT MEMORY WORDS INTO ERROR TANK.
REST*
RESTORE ALL PROGRAM PARAMETERS NEEDED TO INITIALIZE A SINGLE TEST CASE.
10

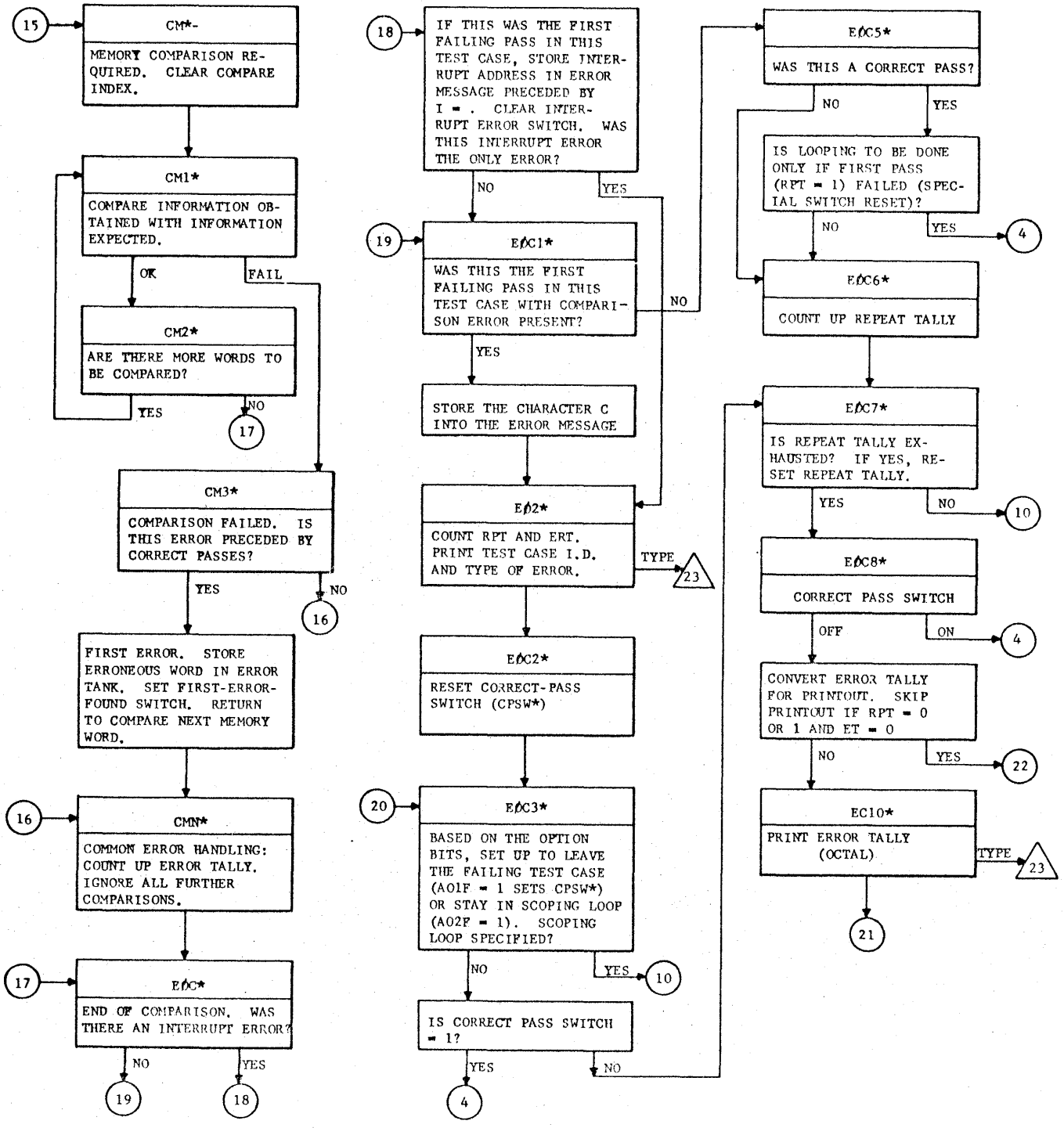
3.7 Con't.


10
SIM*
SET OUTPUT MEMORY: SET UP FOR STACK PUSH-DOWN CHECK. STORE (TEM5*) IN C ADDRESS. INPUT MEMORY WORDS REQUIRED?
NO
YES
SIM1*
STORE INPUT MEMORY WORDS INTO DESIGNATED LOCATIONS.
SIM2*
IS THIS A CHARACTER MODE TEST CASE?
NO
YES
CLEAR BITS 16-30 IN WORD ADDRESSED BY F REGISTER. THIS FORCES R(AFTER) TO BE LOADED FROM BITS 34-42 OF CELL 0000.
SIM3*
SET UP INPUT STACK: FILL IN INPUT CONTROL WORDS. SET CLEAR-INTERRUPT SWITCH. CLEAR INTERRUPTS.
THIS IS THE "CLEAR INTERRUPT" SUB-ROUTINE. IT IS ALSO USED AS THE BEGINNING OF THE EVALUATION ROUTINE.
11
11
IIR*
INTERROGATE INTERRUPTS INTERRUPT PRESENT?
NO
YES
12
WAS ONLY THE CLEARING OF INTERRUPTS REQUIRED?
YES
NO
13
WAS INTERRUPT CLEAR CALLED BEFORE I/O?
YES
NO
"CLEAR INTERRUPT" SUB-ROUTINE EXITS
INI*
RESET CLEAR-INTERRUPT SWITCH.
INI1*
DUMMY I/O FINISHED INTERRUPT REQUIRED?
NO
YES
INITIATE I/O TO FORCE I/O FINISHED INTERRUPT.
INI2*
IS THIS A SCOPING LOOP?
NO
YES
CHP - MANUAL INTERVENTION HALT FOR SINGLE PULSING.
INIT*
LOAD INCW
NOP PRESENT MAY BE MADE CHP
INITIATE FOR TEST
PERFORM TEST CASE. STORE RESULTS EITHER THROUGH SFT OR SFI.
SFI
12
SFT
11

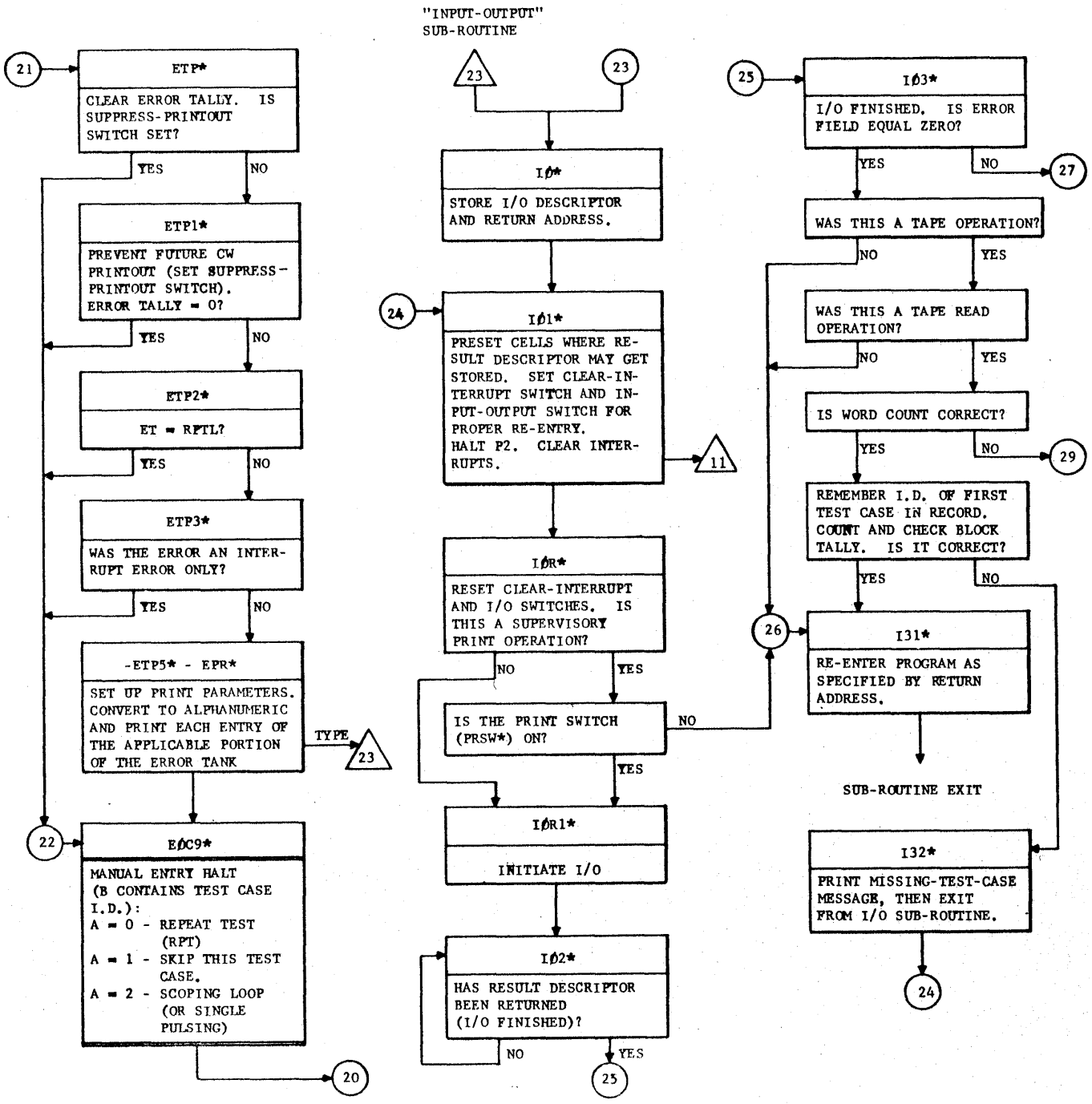
3.7 Con't.


12
IRA*
HARDWARE CONTROLLED INTERRUPT ENTRIES. GCR PUTS INTERRUPT ADDRESS IN TOP OF STACK.
TIMER
IRRECOVERABLE ERROR; CHP, LOOP
I/O 1,2,3,4 FINISHED
UT*
SPECIAL I/O FINISHED INTERRUPT. USED AFTER SWAPPING PROCESSORS ON A 2-PROCESSOR SYSTEM
5
ALL OTHERS EXCEPT KEYBOARD
KEYBOARD
31
IOF*
WAS I/O FINISHED INTERRUPT FORCED BY THE TEST CASE?
NO
YES
CHANGE INTERRUPT ADDRESS TO 27.
IRH*
WAS ONLY THE CLEARING OF INTERRUPTS REQUIRED?
NO
YES
12
13
IRH1*
NO INTERRUPT. IS THIS A SCOPING LOOP?
NO
YES
14
ZERO INTERRUPT ADDRESS
IRH2*
REMEMBER ADDRESS OF INTERRUPT OBTAINED. SWITCH TO WORKING STACK (00120).
TI*
IS INTERRUPT TO BE TESTED?
NO
YES
14
IS THIS THE (FIRST) INTERRUPT WHICH WAS SPECIFIED BY THE TEST CASE TO BE IGNORED?
NO
YES
12
COMPARE INTERRUPT OBTAINED WITH INTERRUPT EXPECTED. DO THEY COMPARE?
NO
YES
IRH3*
INTERRUPT ADDRESS ERROR. IS THIS THE FIRST PASS WITH ANY FAILURE?
YES
NO
16
STORE ERRONEOUS INTERRUPT ADDRESS. SET INTERRUPT-ERROR SWITCH. CONTINUE COMPARISON.
14
IIR1*
SET STACK TO 00120 (WORKING STACK)
COM*
IS THIS A SCOPING LOOP (A02F ON)?
NO
YES
22
ARE CONTROL WORDS TO BE COMPARED?
NO
YES
COM1*
COMPARE CONTROL WORDS
OK
FAIL
A1* THRU A6*
IS THIS ERROR PRECEDED BY CORRECT PASSES?
YES
NO
16
FIRST ERROR. STORE ERRONEOUS CONTROL WORD IN ERROR TANK. SET FIRST-ERROR-FOUND SWITCH.
R1* THRU R5*
CONSECUTIVE RETURN POINTS (CPSW*=1)
R6*
ARE THERE MEMORY WORDS TO BE COMPARED?
YES
NO
15
17

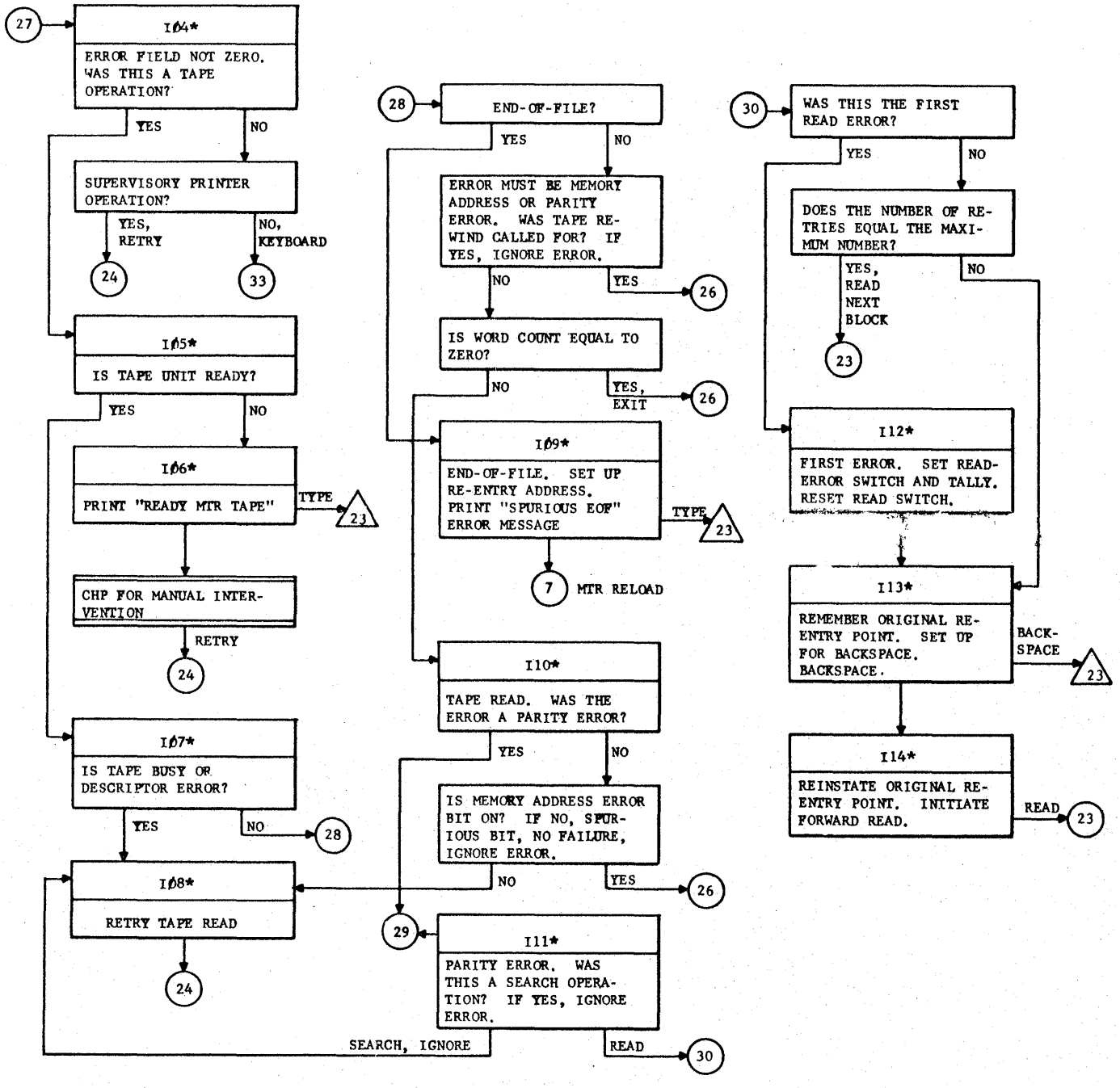
3.7 Con't.


15
CM*-
MEMORY COMPARISON REQUIRED. CLEAR COMPARE INDEX.
CM1*
COMPARE INFORMATION OBTAINED WITH INFORMATION EXPECTED.
OK
FAIL
CM2*
ARE THERE MORE WORDS TO BE COMPARED?
YES
NO
17
CM3*
COMPARISON FAILED. IS THIS ERROR PRECEDED BY CORRECT PASSES?
YES
NO
16
FIRST ERROR. STORE ERRONEOUS WORD IN ERROR TANK. SET FIRST-ERROR-FOUND SWITCH. RETURN TO COMPARE NEXT MEMORY WORD.
16
CMN*
COMMON ERROR HANDLING: COUNT UP ERROR TALLY. IGNORE ALL FURTHER COMPARISONS.
17
EØC*
END OF COMPARISON. WAS THERE AN INTERRUPT ERROR?
NO
YES
19
18
18
IF THIS WAS THE FIRST FAILING PASS IN THIS TEST CASE, STORE INTERRUPT ADDRESS IN ERROR MESSAGE PRECEDED BY I = . CLEAR INTERRUPT ERROR SWITCH. WAS THIS INTERRUPT ERROR THE ONLY ERROR?
NO
YES
19
EØC1*
WAS THIS THE FIRST FAILING PASS IN THIS TEST CASE WITH COMPARISON ERROR PRESENT?
NO
YES
STORE THE CHARACTER C INTO THE ERROR MESSAGE
EØ2*
COUNT RPT AND ERT. PRINT TEST CASE I.D. AND TYPE OF ERROR.
TYPE
23
EØC2*
RESET CORRECT-PASS SWITCH (CPSW*)
20
EØC3*
BASED ON THE OPTION BITS, SET UP TO LEAVE THE FAILING TEST CASE (A01F = 1 SETS CPSW*) OR STAY IN SCOPING LOOP (A02F = 1). SCOPING LOOP SPECIFIED?
NO
YES
10
IS CORRECT PASS SWITCH = 1?
YES
NO
4
EØC5*
WAS THIS A CORRECT PASS?
NO
YES
IS LOOPING TO BE DONE ONLY IF FIRST PASS (RPT = 1) FAILED (SPECIAL SWITCH RESET)?
NO
YES
4
EØC6*
COUNT UP REPEAT TALLY
EØC7*
IS REPEAT TALLY EXHAUSTED? IF YES, RESET REPEAT TALLY.
YES
NO
10
EØC8*
CORRECT PASS SWITCH
OFF
ON
4
CONVERT ERROR TALLY FOR PRINTOUT. SKIP PRINTOUT IF RPT = 0 OR 1 AND ET = 0
NO
YES
22
EC10*
PRINT ERROR TALLY (OCTAL)
TYPE
23
21

3.7 Con't.

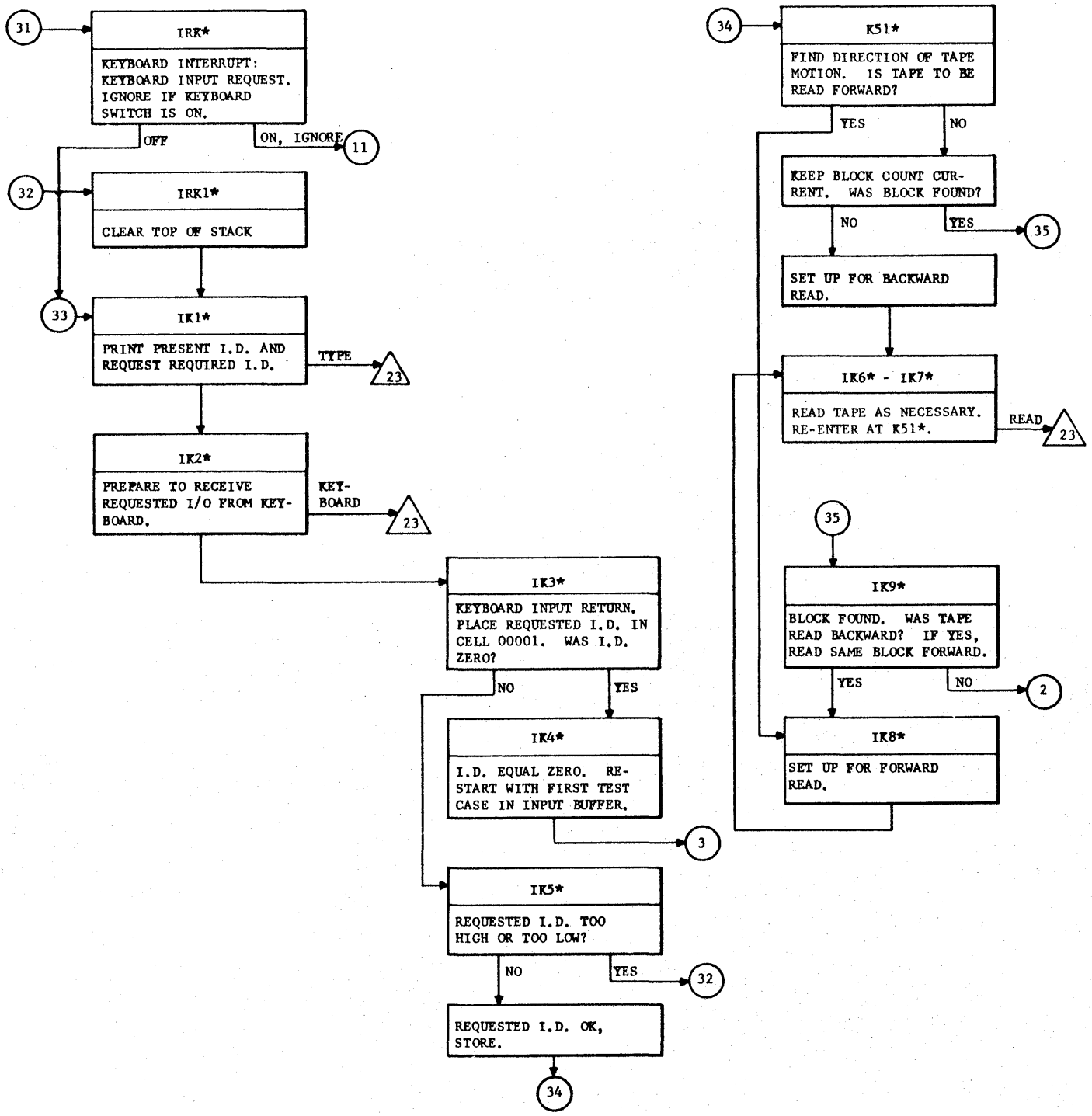

"INPUT-OUTPUT" SUB-ROUTINE
21
ETP*
CLEAR ERROR TALLY. IS SUPPRESS-PRINTOUT SWITCH SET?
YES
NO
ETP1*
PREVENT FUTURE CW PRINTOUT (SET SUPPRESS-PRINTOUT SWITCH). ERROR TALLY = 0?
YES
NO
ETP2*
ET = RPTL?
YES
NO
ETP3*
WAS THE ERROR AN INTERRUPT ERROR ONLY?
YES
NO
-ETP5* - EPR*
SET UP PRINT PARAMETERS. CONVERT TO ALPHANUMERIC AND PRINT EACH ENTRY OF THE APPLICABLE PORTION OF THE ERROR TANK
TYPE
23
22
EØC9*
MANUAL ENTRY HALT
(B CONTAINS TEST CASE I.D.):
A = 0 - REPEAT TEST (RPT)
A = 1 - SKIP THIS TEST CASE.
A = 2 - SCOPING LOOP (OR SINGLE PULSING)
20
23
23
IØ*
STORE I/O DESCRIPTOR AND RETURN ADDRESS.
24
IØ1*
PRESET CELLS WHERE RESULT DESCRIPTOR MAY GET STORED. SET CLEAR-INTERRUPT SWITCH AND INPUT-OUTPUT SWITCH FOR PROPER RE-ENTRY. HALT P2. CLEAR INTERRUPTS.
11
IØR*
RESET CLEAR-INTERRUPT AND I/O SWITCHES. IS THIS A SUPERVISORY PRINT OPERATION?
NO
YES
IS THE PRINT SWITCH (PRSW*) ON?
NO
YES
IØR1*
INITIATE I/O
IØ2*
HAS RESULT DESCRIPTOR BEEN RETURNED (I/O FINISHED)?
NO
YES
25
25
IØ3*
I/O FINISHED. IS ERROR FIELD EQUAL ZERO?
YES
NO
27
WAS THIS A TAPE OPERATION?
NO
YES
WAS THIS A TAPE READ OPERATION?
NO
YES
IS WORD COUNT CORRECT?
YES
NO
29
REMEMBER I.D. OF FIRST TEST CASE IN RECORD. COUNT AND CHECK BLOCK TALLY. IS IT CORRECT?
YES
NO
26
I31*
RE-ENTER PROGRAM AS SPECIFIED BY RETURN ADDRESS.
SUB-ROUTINE EXIT
I32*
PRINT MISSING-TEST-CASE MESSAGE, THEN EXIT FROM I/O SUB-ROUTINE.
24

3.7 Con't.


27
I04*
ERROR FIELD NOT ZERO. WAS THIS A TAPE OPERATION?
YES
NO
SUPERVISORY PRINTER OPERATION?
YES, RETRY
NO, KEYBOARD
24
33
I05*
IS TAPE UNIT READY?
YES
NO
I06*
PRINT "READY MTR TAPE"
TYPE
23
CHP FOR MANUAL INTERVENTION
RETRY
24
I07*
IS TAPE BUSY OR DESCRIPTOR ERROR?
YES
NO
28
I08*
RETRY TAPE READ
24
28
END-OF-FILE?
YES
NO
ERROR MUST BE MEMORY ADDRESS OR PARITY ERROR. WAS TAPE REWIND CALLED FOR? IF YES, IGNORE ERROR.
NO
YES
26
IS WORD COUNT EQUAL TO ZERO?
NO
YES, EXIT
26
I09*
END-OF-FILE. SET UP RE-ENTRY ADDRESS. PRINT "SPURIOUS EOF" ERROR MESSAGE
TYPE
23
7
MTR RELOAD
I10*
TAPE READ. WAS THE ERROR A PARITY ERROR?
YES
NO
IS MEMORY ADDRESS ERROR BIT ON? IF NO, SPURIOUS BIT, NO FAILURE, IGNORE ERROR.
NO
YES
26
29
I11*
PARITY ERROR. WAS THIS A SEARCH OPERATION? IF YES, IGNORE ERROR.
SEARCH, IGNORE
READ
30
30
WAS THIS THE FIRST READ ERROR?
YES
NO
DOES THE NUMBER OF RETRIES EQUAL THE MAXIMUM NUMBER?
YES, READ NEXT BLOCK
NO
23
I12*
FIRST ERROR. SET READ-ERROR SWITCH AND TALLY. RESET READ SWITCH.
I13*
REMEMBER ORIGINAL RE-ENTRY POINT. SET UP FOR BACKSPACE. BACKSPACE.
BACK-SPACE
23
I14*
REINSTATE ORIGINAL RE-ENTRY POINT. INITIATE FORWARD READ.
READ
23

3.7 Con't.

KEYBOARD INTERRUPT



IV. TEST ROUTINE CHECKOUT RECORD

DATE	PRGMR	SYSTEM	REMARKS
4-20-65	A.P.T.	Two-Processor B5500, Memory Modules 0 through 6	
5-1-65	A.P.T.	One-Processor B5500, Memory Modules 0 through 7	