IDENTIFICATION

Product Code:	MAINDEC-9A-D7AD-D
Product Name:	PDP-9 Basic Exerciser
Date Created:	May 3, 1968
Maintainer:	Diagnostics Group
Author:	J. W. Richardson

jsm - 369-9239

c. Place the HRI tape in the reader.

d. Press I/O RESET, and then READ-IN.

The program is not self-starting.

4. STARTING PROCEDURE

4.1 Starting Addresses

22 or 10022 if the program is currently in the upper 4K field.

4.1.1 Restarting Addresses - 26 or 10026 if the program is currently in the upper 4K field.

4.2 Operator Action

a. Set the ADDRESS switches to 22.

b. Place all ACS down for normal program operation. See paragraph 5.1 for use of the ACS to inhibit certain portions of the program.

c. Press I/O RESET, and then START.

d. Approximately 3-1/2 feet of leader will be punched. This leader is blank except for one frame which has all channels punched.

e. Place the punched frame directly over the reader drive sprocket, and position the tape between reader and punch for minimum binding.

f. Press CONTINUE.

g. The program will run until an error halt occurs, or manually stopped by the operator.

h. Steps c through f above are similar to the procedure performed with the PDP-9 punch test.

4.2.1 Restarting Procedure -

a. Set the ADDRESS switches to 26.

b. Place all ACS down for normal program operation. See paragraph 5.1 for use of the ACS to inhibit certain portions of the program.

c. If the punch, read and print sequence is not inhibited, make sure there is tape in the reader. The tape does not have to be blank leader when restarting.

d. Press I/O RESET, and then START.

e. The program will run until an error halt occurs, or manually stopped by the operator.

5. OPERATING PROCEDURE

5.1 Operational Switch Settings

Normal program operation is achieved by placing all ACS down before starting from locations 22 or 26.

The operator is provided nine options with which to modify the operation of the program. These may be selected by placing any one or a combination of ACS 0 through 8 up before starting from locations 22 or 26.

To make changes in the ACS settings, the program must be stopped by the operator before the changes are made. The program must then be restarted from location 26 (or 22 if new leader is desired). The program may not recognize the new ACS settings if the above procedure is not followed.

ACS Functions

0	(1)	Run only the punch, read and print sequence plus the real-time clock. Program interrupt will be enabled.
1	(1)	Inhibit the punch, read and print sequence. Program interrupt is disabled. The real-time clock is on. The complete instruction test and memory checkerboard test will be performed.
2	(1)	Loop continuously on the "add random pairs" test. The real-time clock, punch, read and print sequence plus program interrupt will be enabled unless specified otherwise by an ACS.
3	(1)	Loop continuously on the memory checkerboard test. Program relocation will not take place. Program action otherwise is the same as that described for ACS 2.
4	(1)	Inhibit program relocation. Unless otherwise specified, the program will run in a normal way, but will not relocate from its current 4K field location to the opposite field after completing the memory checkerboard test.
5	(1)	Inhibit clock. Unless otherwise specified, program action is normal except that the clock should always be off.
6	(1)	Inhibit the reader and TTY. The punch will run continuously. Tape must be in the reader to prevent the no-tape indicator from being set. Program action is normal unless otherwise specified.
7	(1)	Inhibit the punch and TTY. The reader will run continuously. A loop or fan-fold tape with any data may be used. Program action is normal unless otherwise specified.
8	(1)	Inhibit the punch. The reader will read 52 characters at full speed and then halt. The TTY will then print the 52 characters read. Any tape loop or fan-fold tape may be used. Program action is normal unless otherwise specified.

Any combination of the nine ACS may be used, as long as the operations do not conflict; i.e., if ACS 2 and 3 are both up, the add random pairs test would be looped. Memory checkerboard would not be run unless the program is restarted with ACS 3 alone.

The I/O devices may be controlled with several combinations of ACS 6, 7 and 8. If ACS 7 and 8 are both up the reader will run continuously, as if ACS 7 only were up. If ACS 6 and 7 or 6 and 8 are up, all devices will be inhibited. Program interrupt and the real-time clock will be enabled unless otherwise specified.

5.2 Subroutine Abstracts

The PDP-9 Basic Exerciser may be thought of as three separate programs; i.e., the instruction and memory tests; punch, read and print sequence, and operation of the real-time clock. The instruction and memory tests will be interrupted, at a device rate, by the punch, reader or Teletype. The clock will randomly interrupt any of the above operations at a rate determined by the program. After each clock interrupt, the clock is reinitialized with a new number obtained by a random number generator. The clock interrupts should occur no less than 2 seconds apart, nor more than 9 seconds. The clock interrupts take first priority, followed by the Teletype, reader, and punch.

5.2.1 Instruction and Memory Tests - The instruction test portion of the Basic Exerciser performs tests on all operate group and memory reference instructions. The individual instructions are looped a random number of times before proceeding to the next test. The maximum number of loops made on any one test is 32,767.

The adder is tested using two different methods. The first performs bit by bit tests on the adder using the ADD instruction. Besides checking for correct results after an addition, the link is tested during overflow and no overflow conditions.

The second method, the "Add Random Pairs" test, tests the adder using one pair of random numbers (A and B) and their 1's complement values (-A and -B), and the ADD instruction. These four values are added in various combinations, the results of which are compared against precalculated results. The precalculated results are obtained by adding the two pairs together using the TAD instruction. Four additions are made, the results of which are used in the test. The link is tested after each addition. If it is a 1, a 1 is added to the result to simulate an end-around-carry.

The numbers added and their sums are indicated in the listing using the following symbols:

-B+(-A) = SUMNEG
A+B = SUMPOS
B-A = BMASUM
A-B = AMBSUM

The values of A, -A, B and -B plus their sums are used to test the combinations of ADD's shown below.

ADD	SUM SHOULD EQUAL	
A + B	SUMPOS	
-B + A	AMBSUM	
-B + (-A)	SUMNEG	
B - A	BMASUM	
(A + B) - A	BPOS	(B)
(B - A) -B	ANEG	(-A)
(-A-B) +A	BNEG	(-B)
(A - B) +B	APOS	(A)
777777 + A	APOS	
A + B - A	BPOS	
A + B -A -A	BMASUM	(B - A)
A + B -A -A -B	ANEG	
A + B -A -A -B -B	SUMNEG	(-A -B)
A + B -A -A -B -B + A	BNEG	
A + B -A -A -B -B + A + A	AMBSUM	(-B + A)
A + B -A -A -B -B + A +A +B	APOS	

After completing one pass of the above tests, a second pass is made on the same tests. The second pass makes all "B" constants "A", and all "A" constants "B" before repeating.

Immediately following the second pass, one random number and its 1s complement is obtained and saved in APOS and ANEG, respectively. Bit 0 of APOS is tested for equaling 0 or 1. If the value is 1, the bit remains unchanged, and the respective bit in the complement number is changed to equal a 1. The two numbers are then added together, the sum of which should equal all 0s except for bit 0. If the ADD is successful, the program continues testing all other bit positions in the same manner.

Example: (Bit 0 altered)

Step	APOS Value	ANEG Value
1	577776	200001
2	577776	600001 (bit 0 altered)
3	Add together. Result should = altered bit.	

```
  101 111 111 111 111 110
+ 110 000 000 000 000 001
-------------------------

  011 111 111 111 111 111
+                       1   (end around carry)
-------------------------
  100 000 000 000 000 000
```

The sum equals the altered bit position.

After completing the adder tests, the remaining memory reference instructions are tested. The last test performed by the exerciser is the memory checkerboard test. This routine writes and reads four different checkerboard patterns similar to the Basic Memory Checkerboard program. The memory test is looped three times before program relocation takes place.

After completing the memory checkerboard test, the program relocates the entire Basic Exerciser to the opposite 4K field. All memory reference instructions and memory locations used for testing are adjusted accordingly. All locations within the program which reference any memory location between 0 and 21 are not adjusted. These locations are used during program interrupts, autoindexing tests, etc., and must not be altered. Program interrupt is disabled while relocation is taking place.

After relocation of the program is completed, the exerciser is automatically restarted at location 70 (or 10067). This location is tagged SEQUEN. The operator is able to determine the location of the program by observing MB bit 5. This bit will glow brightly when the program is in the higher 4K field, as compared to when residing in the lower 4K field.

The Teletype BELL will ring once for each completed pass of the program. One pass is defined as the program performing all tests from each 4K field, and then relocating back to the field in which the program was first initiated.

When operating the Basic Exerciser with program interrupt inhibited (ACS 1 up), the message "COMPLETE" will be printed after five complete passes of the program. This message is printed after ten passes when ACS 4 (inhibit relocation) as well as ACS 1 is up. This feature is included as a means to determine the number of successful passes completed by the program if it is to be run for extended periods of time.

5.2.2 Punch, Read, Print Sequence - The instruction and memory tests will be interrupted, at a device rate, by the punch, reader, or Teletype. The data punched consists of the alphabet characters, followed by numbers 0 through 9, with a space character being punched between each letter or numeral character. The reader will read the tape at punch speed, storing away any punched character. Frames of all 0s are ignored. The punch and read sequence consists of 52 ASCII characters punched, read and stored away in an input buffer (tagged TTBUFA). After reading the 52nd character, the contents of TTBUFA are transferred directly to another 52 - location buffer tagged TTBUFB. This second buffer is provided to enable the operator to stop the program and compare the contents of either buffer A or buffer B with the punched data on tape. Punch and read operation is halted after the 52nd character is read and stored. The contents of TTBUFB are then printed on the Teletype. The punch and read sequence continues immediately after the 52nd character is printed. The data punched and read should appear on the Teletype as the example below.

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9
```

The punch and read sequence generates 72 characters altogether, even though only 52 are punched, read and printed at one time. The alphabet and numbers sets with a space between each character enables a full line to be printed. The spacing also enables the operator to more easily detect a misprinted character. Each group of 72 characters is separated by 8 blank frames. The group which is positioned in the reader is the current line being printed. A carriage return and a line feed is punched at the end of each group. The program will punch 6 extra blank frames between two groups, approximately every fourteenth group, to enable the slack between reader and punch to remain constant.

5.2.3 Use of the Real Time Clock - The instruction and memory tests, and the punch, read and print sequence are both interrupted randomly by the real-time clock. When a clock interrupt occurs, all other operations are halted until the clock interrupt has been serviced. The program allows the clock to continue incrementing for 1/2 second after the interrupt occurs. This is evident during every clock interrupt by observing MB indicators 12 through 17 incrementing. Immediately after the clock has incremented, an additional 1/2 second, it is reset to a new random value. This value is chosen by the program to ensure that the clock interrupts no sooner than 2 seconds, nor later than 9 seconds. The clock is again enabled after being reset to a new value, and the instruction test or read, punch and print sequence is allowed to continue from the point of interrupt.

At times, the Basic Exerciser may appear to be caught in a loop after a clock interrupt occurs. The console indicators will show the clock and PIE as being disabled, and the punch, read and print sequence will be halted for several seconds. The program during this time is attempting to generate a number for the clock which falls within the 2 to 9 second limit. All operations will be resumed as soon as a suitable random value is found.

The operator may disable the clock interrupts by restarting from location 26 with ACS 5 up.

5.2.4 Interrupt Service Routine - Program interrupts by the clock, punch, reader or Teletype are all serviced by a common routine. A common routine for reentering the instruction test is also used.

Locations 0 through 6 are used to save the contents of the AC and PC immediately after an interrupt occurs. The contents of the AC are stored in the location tagged SAVAC. The contents of the link and PC are stored in the location tagged RJMP. The program then enters a routine which determines which of the four devices interrupted the program. This routine is tagged SRVINT. SRVINT will test for device flags in the following order: clock, Teletype, no-tape flags, reader, punch. The first device flag found to be set indicates the device which must be reinitiated by the program.

Immediately after selecting the proper device, a routine is entered which will restore the contents of the link and AC at the time of the program interrupt. The routine is tagged RTNIT. RTNIT first restores the AC (from SAVAC); restores the link (by testing bit 0 of RJMP); enables program interrupt; and then returns to the instruction test by a JMP indirect on the contents of RJMP.

The operator may disable program interrupts by restarting from location 26 with ACS 1 up.

5.3 Program and Operator Action

See Sections 4.2 and 5.2.

6. ERRORS

6.1 Error Halts and Description

Reference the program listing for all error halts.

All error halts are tagged EXXX, and are commented to aid debugging. Each test is self-contained, and may be looped. See Section 6.2.1 for looping instructions.

Unless a solution is obvious from following the listing, the proper MAINDEC diagnostic for the device in error should be run. This should be necessary mainly when errors are caused by one of the I/O devices. The diagnostics for the I/O devices are listed below:

Device	Program
Real-Time Clock	Instruction Test Part 1 2
Program Interrupt	Instruction Test Part 1 2
Punch	Punch Test (9A-D2DB)
Reader	Reader Test (9A-D2CB)
Teletype	TTY Test (9A-D2BB)

Incorrect operation of the real-time clock will appear as clock interrupts occurring sooner than 2 seconds apart, or greater than 9 seconds, or possibly no clock interrupts will occur. Also, after a clock interrupt, the clock should not increment further for any longer than approximately 1/2 second.

Printing of incorrect data may be caused by the data being incorrectly punched, read, or printed. Storage registers, and their locations in the program, which the punch, read, print sequence use are listed below.

Tag	Function
SAVAC (7642)	Saves contents of AC after a program interrupt.
RJMP (7643)	Saves contents of PC and link after a program interrupt.
WORK (7630)	Bit 1 if set indicates TTY is in use.
GOPNCH (6763)	Contains contents of PC at exit from punch routine.
SETCLK (6604)	Routine which sets a random value in clock register 7 when program interrupt is disabled.
CLKSET (6621)	Same function as SETCLK, but is used only after a clock interrupt.
TTOUT (7325)	Location pointer for TTBUFB when printing.
TTIN (7326)	Location pointer for TTBUFA when reading.

Tag	Function
TTBUFA (7360 to 7443)	Storage buffer for characters read.
TTBUFB (7444 to 7530)	Storage for characters to be printed. Contents should equal TTBUFA.
STORE (7335)	Contains character punched.
SETTY (7037)	Routine which is entered after 52 characters have been punched and read. Sets up TTBUFB before printing.
GENRAN (6102)	Random number generator used when PI is disabled.
RANGEN (6133)	Same as GENRAN, but used only after an interrupt.

When data is incorrectly printed, stop the program during print-out. This will enable both TTBUFA and TTBUFB to remain unchanged. TTBUFA will be changed as soon as reading begins.

The data punched is in ASCII mode, and one printed line is indicated on the paper tape by 8 blank frames separating each line. The punched data starts with character A (301) and ends with a line feed (212). A space (240) is punched between all alphabet and number characters.

The line of punched characters in the reader is the line currently being printed. The operator may inspect the tape for an incorrect character punched. If it appears correctly on the tape, it may have been read or printed incorrectly. The characters read are stored in a 52-word buffer beginning at location 7360 (tagged TTBUFA). The characters being printed are stored in a 52-word buffer beginning at location 7444 (tagged TTBUFB). If the program was stopped during printing, these two buffers should contain exactly the same information. The first character read or printed is stored in the first location of either buffer. One character is stored per location. If the data was read incorrectly, the contents of TTBUFA will not equal the last 52 characters on the tape. If the data on tape, and in TTBUFA and TTBUFB are equal, the teleprinter may be at fault.

6.2 Error Recovery

Press CONTINUE to receive further error halts or to continue testing, as indicated by the listing.

Recovery from error halts in the Add Random Pairs test is accomplished by pressing CONTINUE one or more times, depending on the type of error encountered. Pressing CONTINUE after a halt due to an incorrect sum will result in a second halt. The AC will equal the incorrect sum at the first halt, and the sum used for comparison at the second halt. If the error halt is the result of a LINK error, the next test in sequence will be executed.

Recovery from memory checkerboard errors is accomplished by pressing CONTINUE four times. The memory test is restarted after each error halt.

The contents of the AC after each halt will equal the information below.

C (PC)	C (AC)
6351	The address at which the memory error occurred.
6353	What the location should equal (000000 or 777777).
6355	The data as read.
6362	The pattern control word used.

Bit suppression may be accomplished by placing the corresponding ACS up after the halt at location 6354. Place all other ACS down, press CONTINUE, place all ACS down again, and then restore ACS 0 through 5 if needed. The bits selected will not be tested again during the memory test. The selected bits must be reselected after each error halt.

The memory checkerboard test may be continuously looped by restarting from location 26 with ACS 3 up. Program relocation will not take place.

6.2.1 Looping on Individual Tests - Looping on individual tests, except for the interrupt routines, Add Random Pairs test and Memory test, is accomplished by placing a JMP instruction in the first location of the next test in sequence. The address in the JMP instruction should equal the first location of the test to be looped. Restart the program at location 26, if program interrupt is to be enabled. Restart at the first location of the test to be looped, if interrupts are not wanted.

The complete series of tests for any one instruction may be looped by placing a NOP in the location which contains ISZ WORK3. This instruction appears at the end of each series of tests for each instruction. Restart at location 26, or at the beginning of the test to be looped.

6.2.2 Looping on Add Random Pairs - The complete series of tests may be looped by restarting from location 26 with ACS 2 up.

The individual tests may be looped by changing the LAW instruction, appearing after each test, to a JMP. For example, to loop on (A-B) + B = A (tagged AMBPBT), change the LAW AMBPBT instruction to JMP AMBPBT. Restart from location 26, or AMBPBT.

6.2.3 Looping on Memory Checkerboard - Place ACS 3 up, and restart from location 26.

6.2.4 Error Print-outs - The program continually tests for reader or punch no tape indicators being set. When either indicator is set the message "R NO TAPE", or "P NO TAPE" will be printed. The program continues on in sequence after either print-out.

7. OPERATING RESTRICTIONS

All MAINDEC diagnostics which apply to a basic PDP-9 configuration should be run before attempting to run this program.

8. MISCELLANEOUS

8.1 Execution Time

Approximately 1-1/2 minutes are required to execute all tests for one 4K field.

9. PROGRAM DESCRIPTION

The Basic Exerciser performs tests on all operate and memory reference instructions plus core memory.

During normal operation (all ACS down) the program exercises the real-time clock, punch, reader, Teletype and program interrupt while the instruction test portion is running. The MB indicators will change according to the portion of the instruction test currently being executed. When a clock interrupt occurs the MB bits 12 to 17 will increment for 1/2 second (due to clock counts), after which the MB indicators will return to their original state upon re-entering the instruction test.

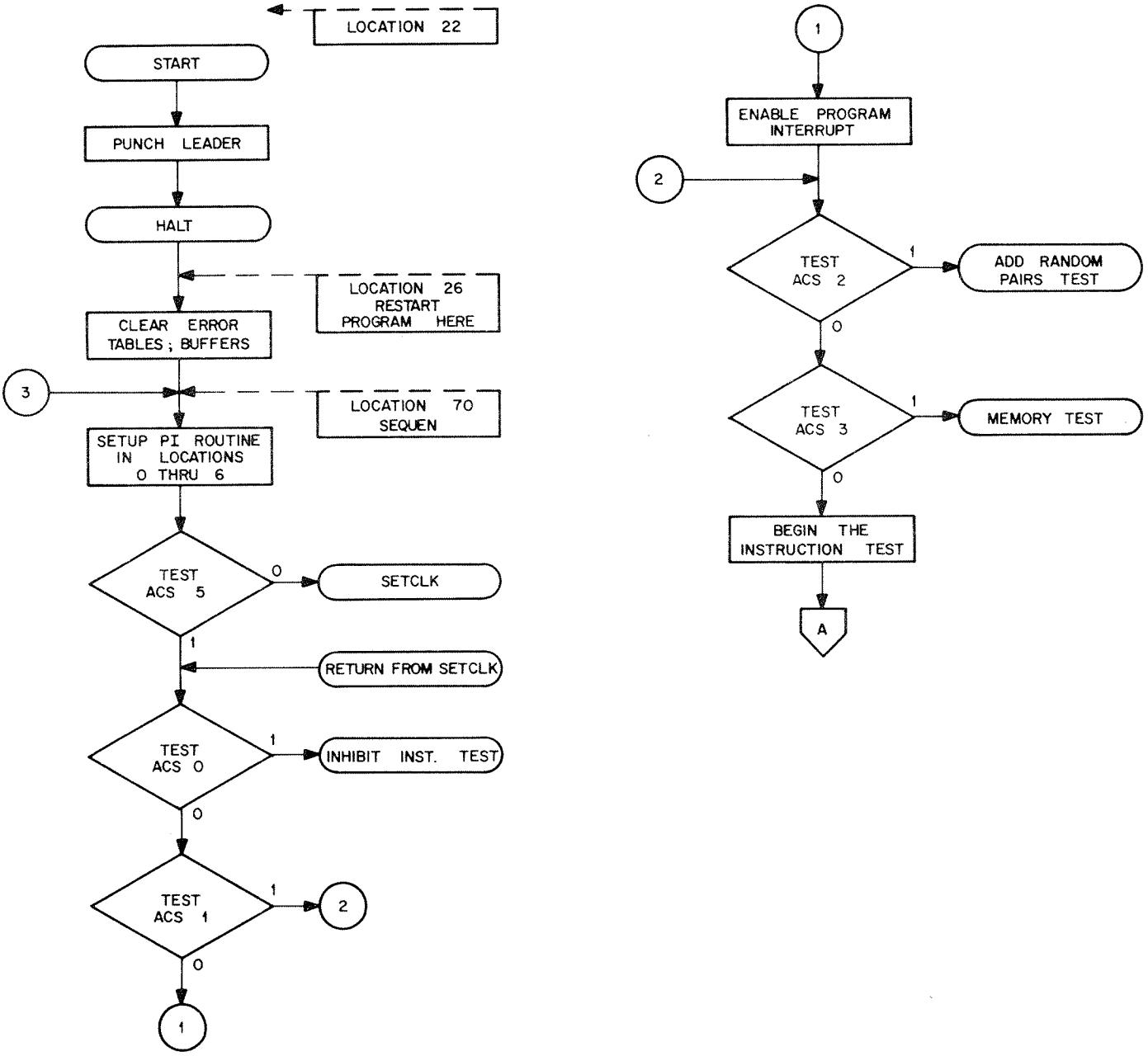
When the punch, read and print sequence is operated with the instruction test inhibited (ACS 0 up) the MB will indicate a constant 600132 (JMP 132). On the listing this is written as JMP at location 132. At location 131 the program interrupt is enabled, and all device interrupts will cause an interrupt immediately after the execution of the JMP instruction. The interrupts are handled in the same manner as if the instruction test portion were operating; the difference being that the interrupt service routine (RTNIT) always returns to location 132 after reinitiating the device which caused the interrupt, instead of returning to the instruction test portion.

The instruction test portion of the program consists of MAINDEC 9A-D01A and D02A (Instruction Test, Parts 1 and 2) condensed onto one tape. Refer to the program listing or flow chart for the testing sequence.

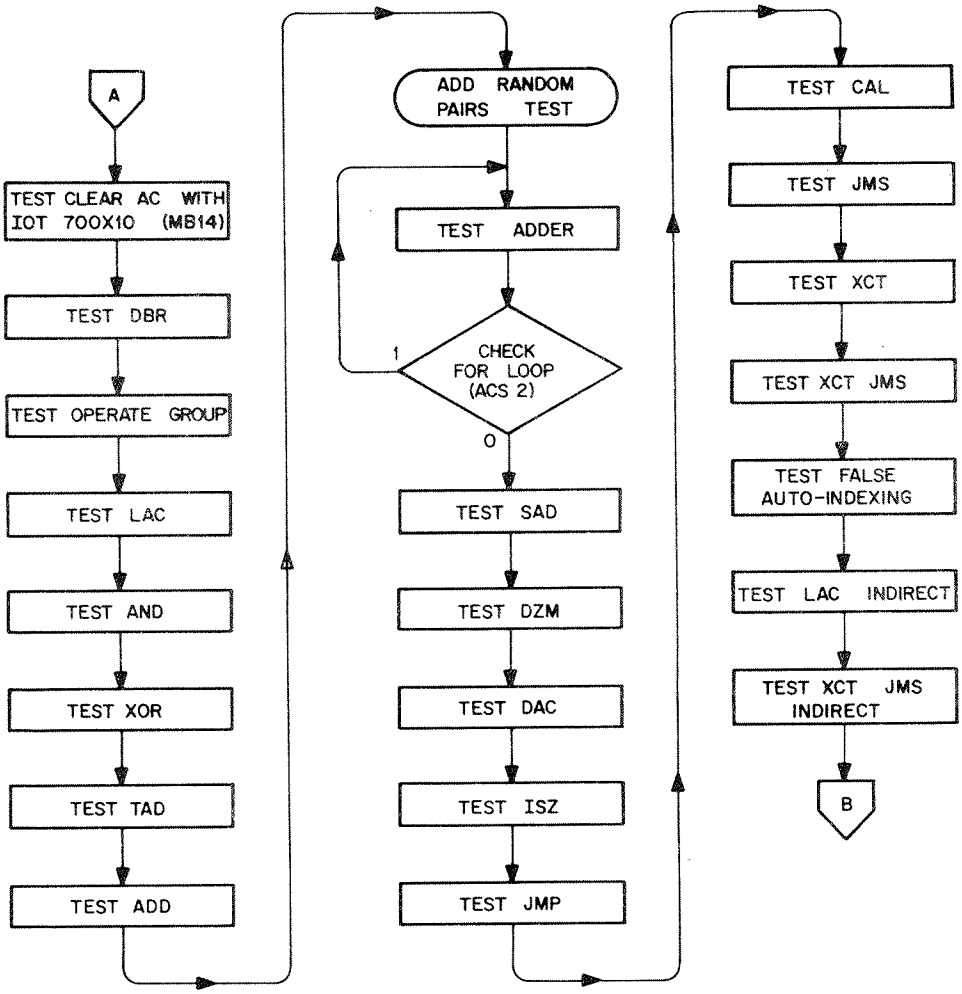
The memory test is executed after the instruction test portion is completed. This test is similar to MAINDEC 9A-D1AB Basic Memory Checkerboard Test. This test is performed three times, after which a routine is entered which relocates the Basic Exerciser to the opposite 4K field in core memory (unless ACS 4 is up). The Exerciser is automatically restarted after relocation is completed.

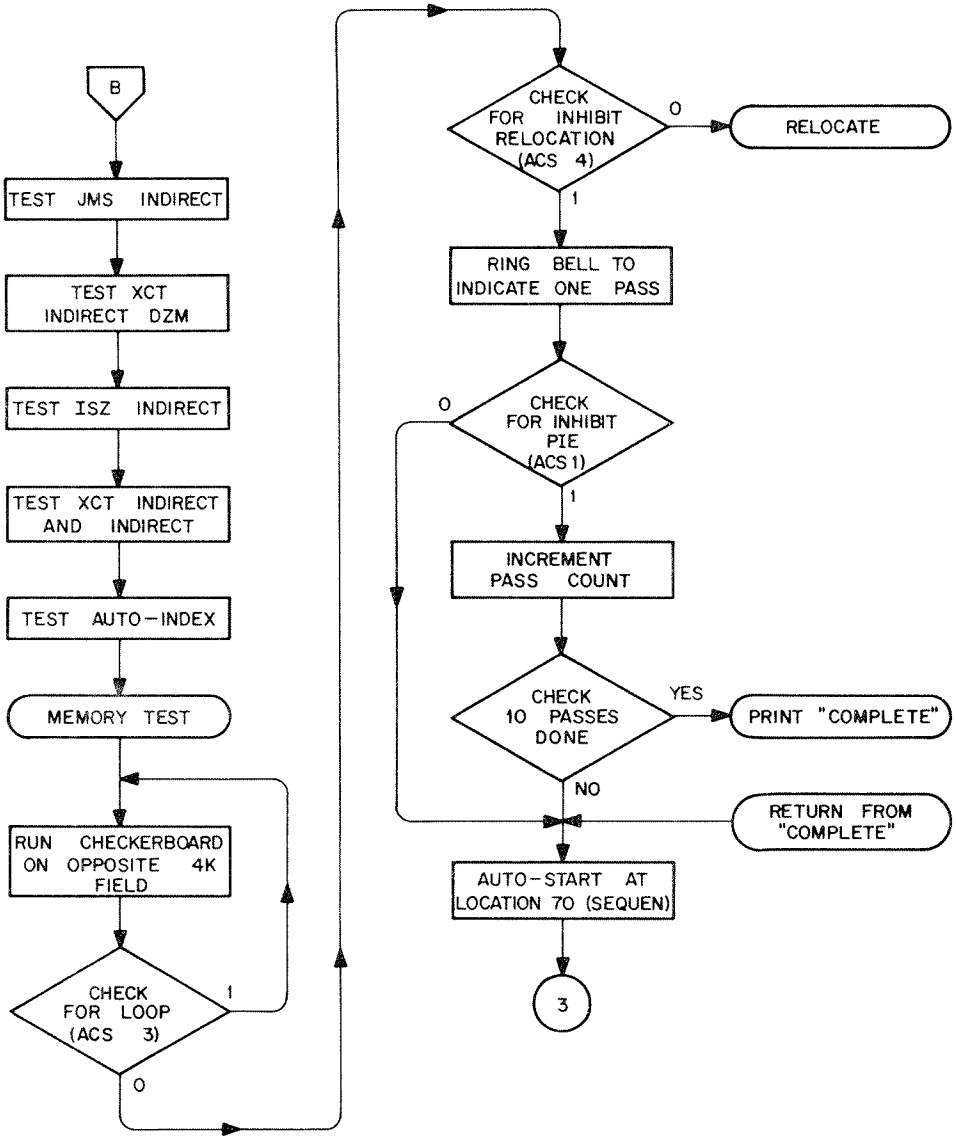
If no errors occur, the Basic Exerciser will run until stopped by the operator.

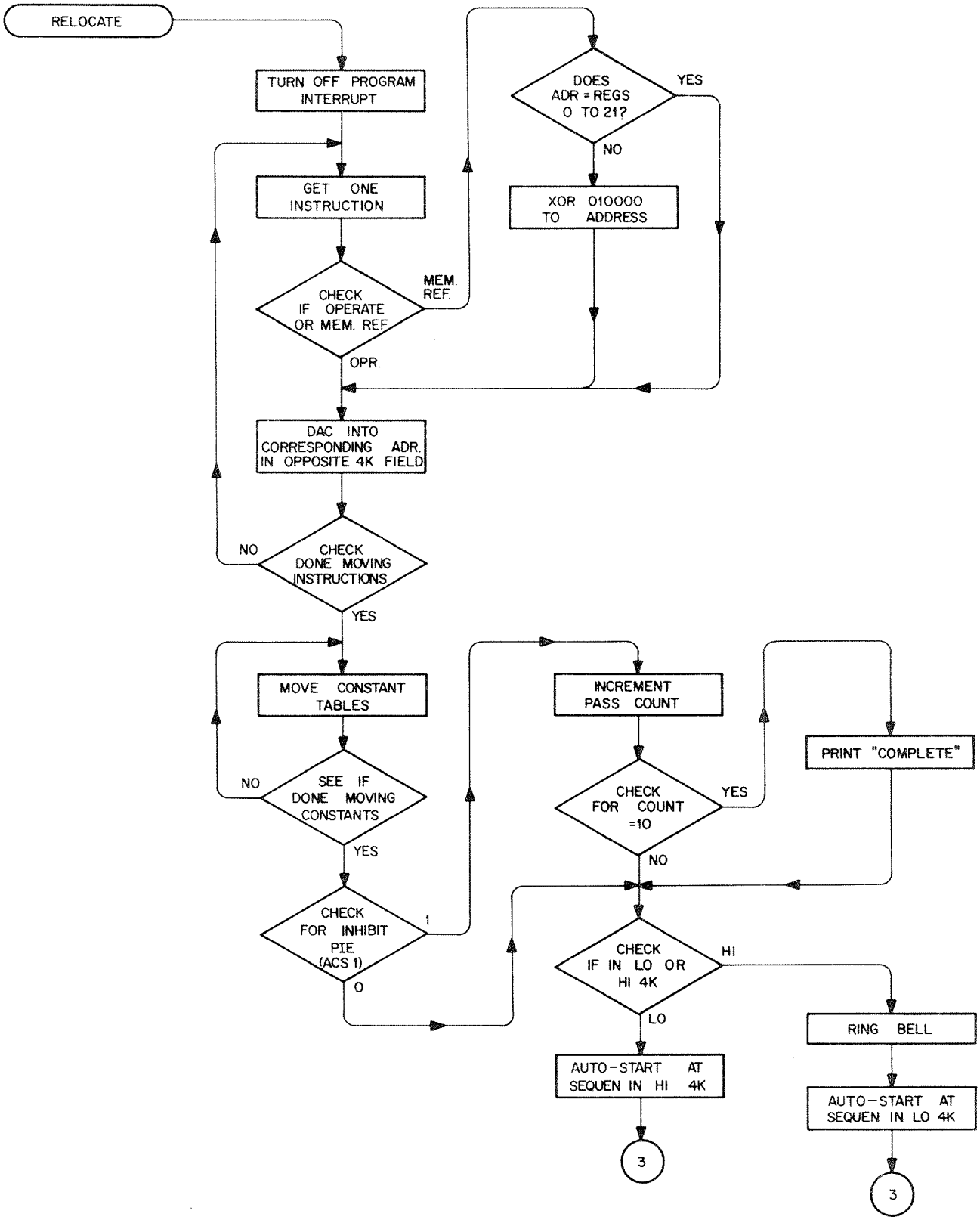
10. FLOW CHARTS



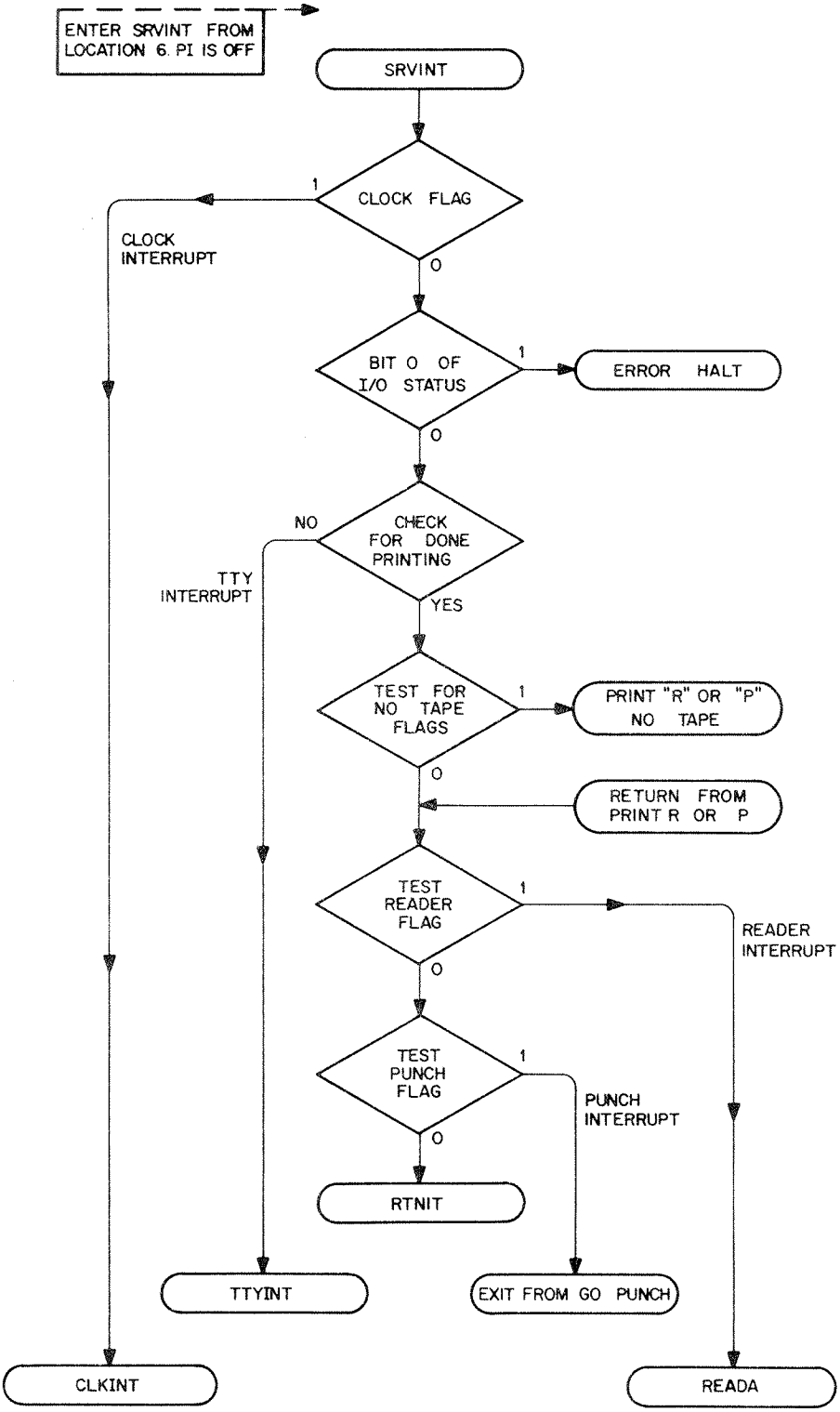
Generalized Flow of PDP-9 Basic Exerciser


A
TEST CLEAR AC WITH IOT 700X10 (MB14)
TEST DBR
TEST OPERATE GROUP
TEST LAC
TEST AND
TEST XOR
TEST TAD
TEST ADD
ADD RANDOM PAIRS TEST
TEST ADDER
CHECK FOR LOOP (ACS 2)
1
0
TEST SAD
TEST DZM
TEST DAC
TEST ISZ
TEST JMP
TEST CAL
TEST JMS
TEST XCT
TEST XCT JMS
TEST FALSE AUTO-INDEXING
TEST LAC INDIRECT
TEST XCT JMS INDIRECT
B

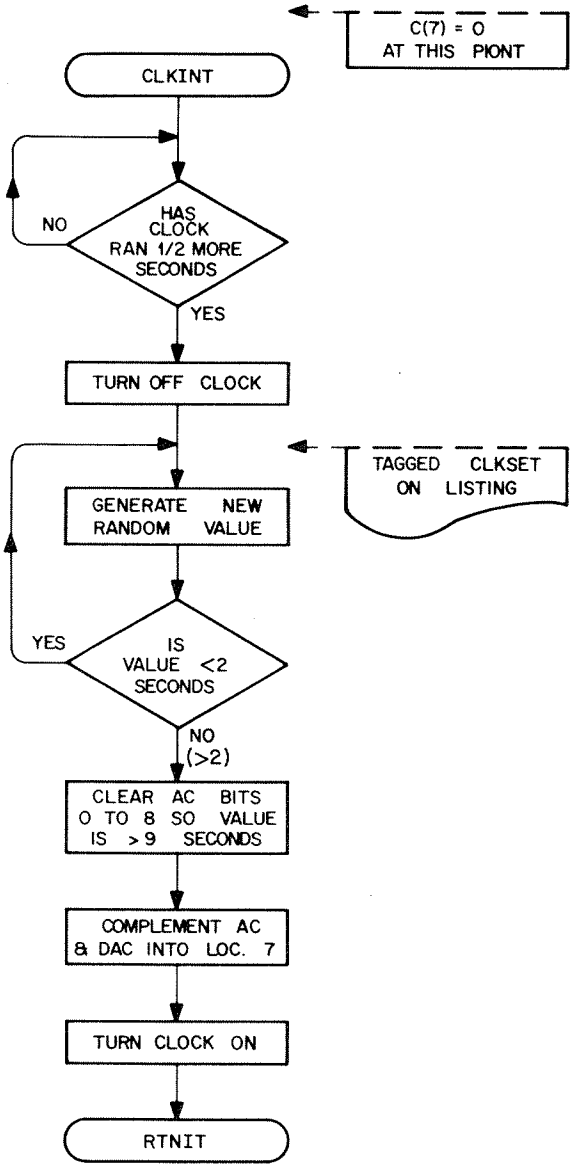

B
TEST JMS INDIRECT
TEST XCT INDIRECT DZM
TEST ISZ INDIRECT
TEST XCT INDIRECT AND INDIRECT
TEST AUTO-INDEX
MEMORY TEST
RUN CHECKERBOARD ON OPPOSITE 4K FIELD
CHECK FOR LOOP (ACS 3)
1
0
CHECK FOR INHIBIT RELOCATION (ACS 4)
0
RELOCATE
1
RING BELL TO INDICATE ONE PASS
CHECK FOR INHIBIT PIE (ACS 1)
0
1
INCREMENT PASS COUNT
CHECK 10 PASSES DONE
YES
PRINT "COMPLETE"
NO
RETURN FROM "COMPLETE"
AUTO-START AT LOCATION 70 (SEQUEN)
3



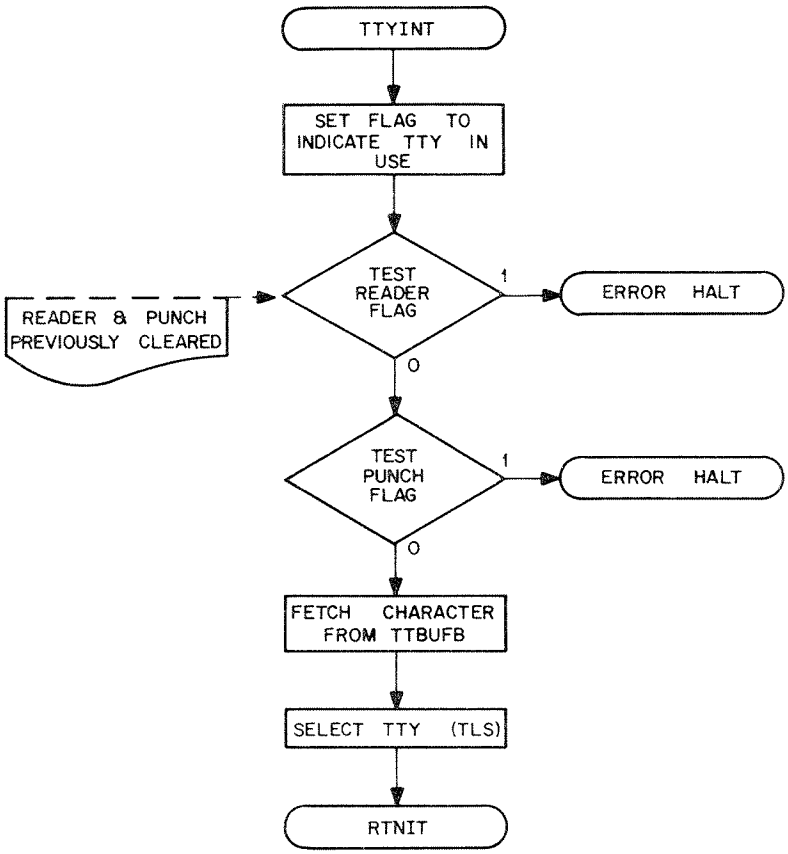
Program Relocation Routine



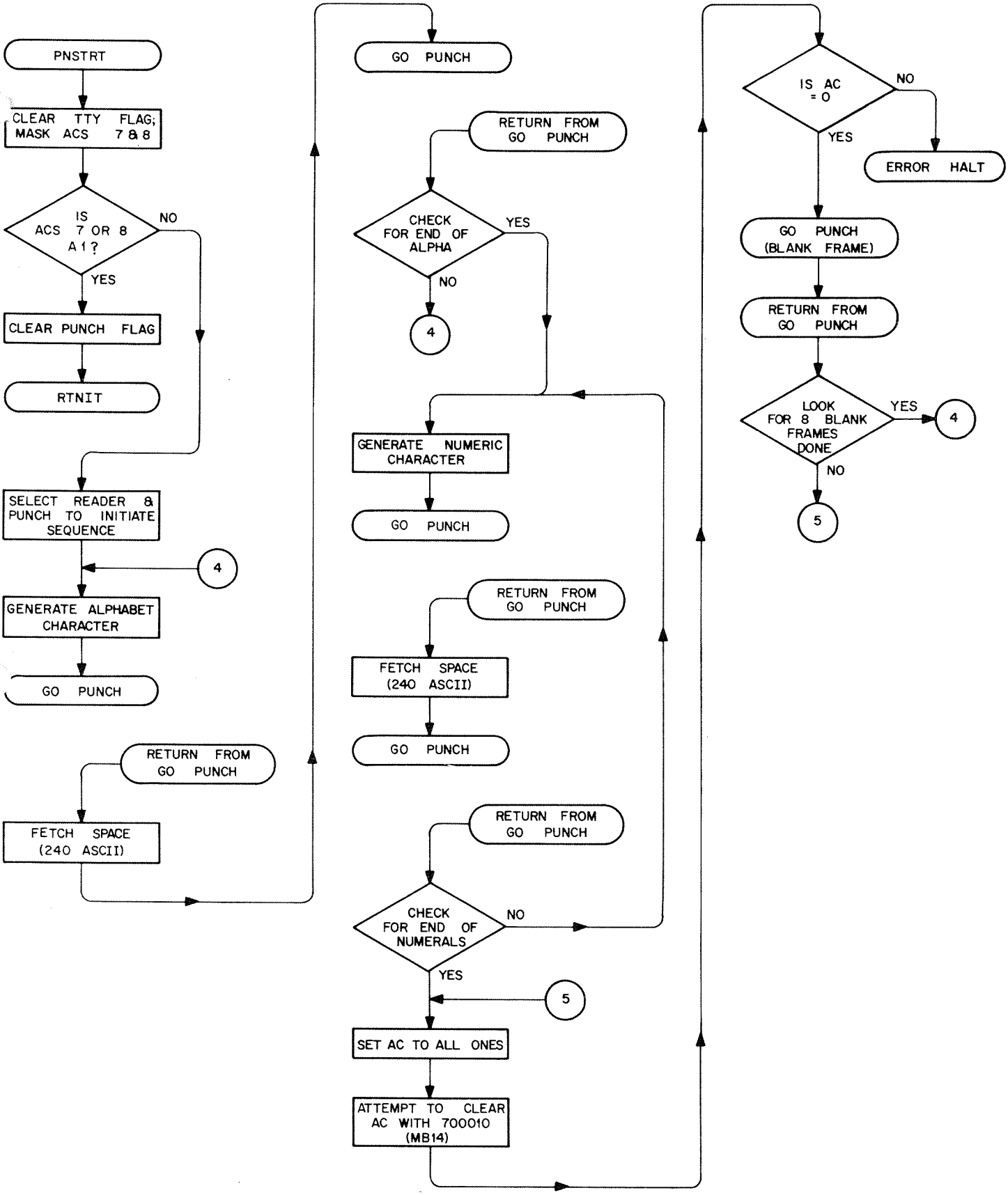
Interrupt Service Routine



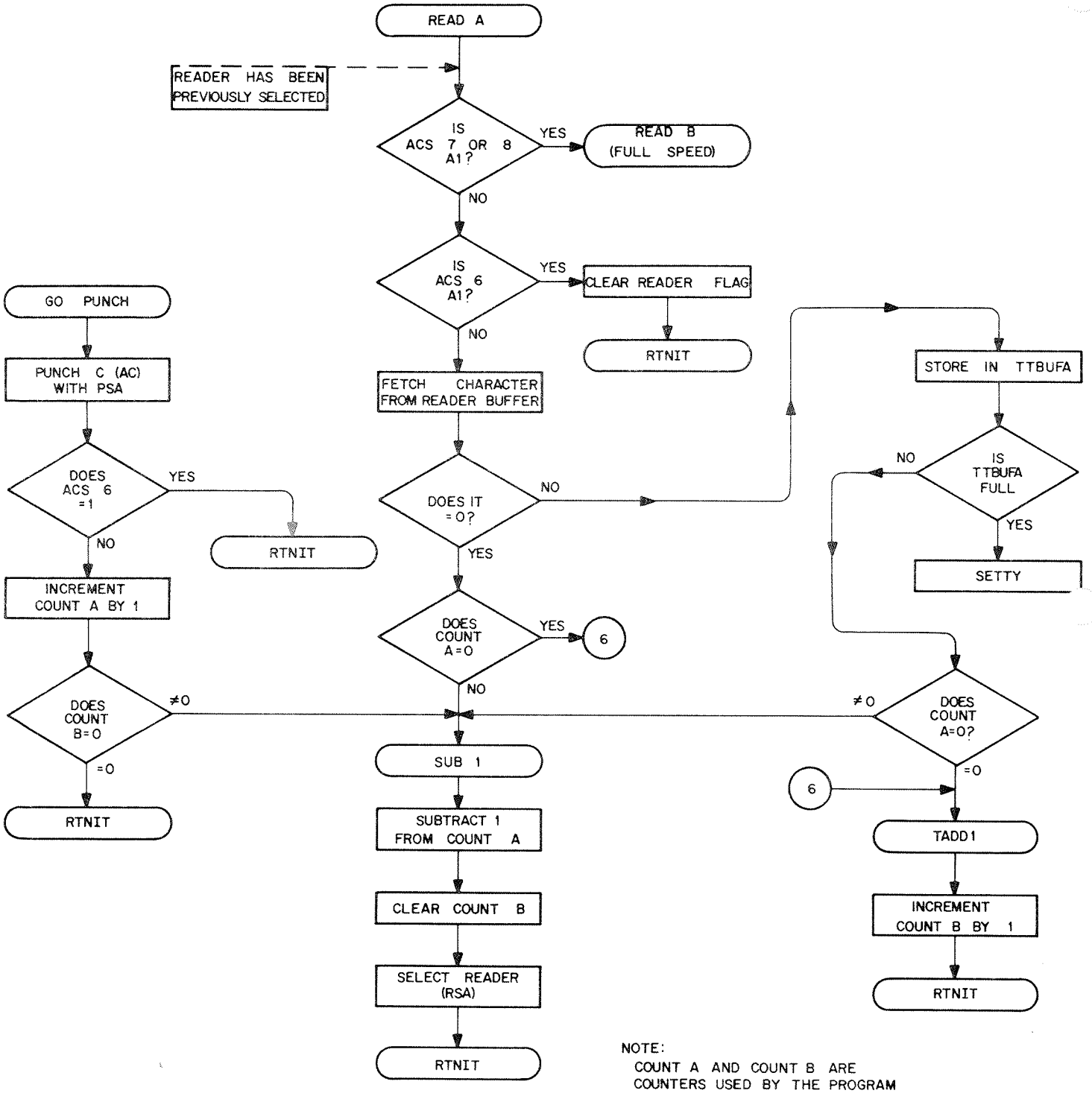
Service Clock Interrupt



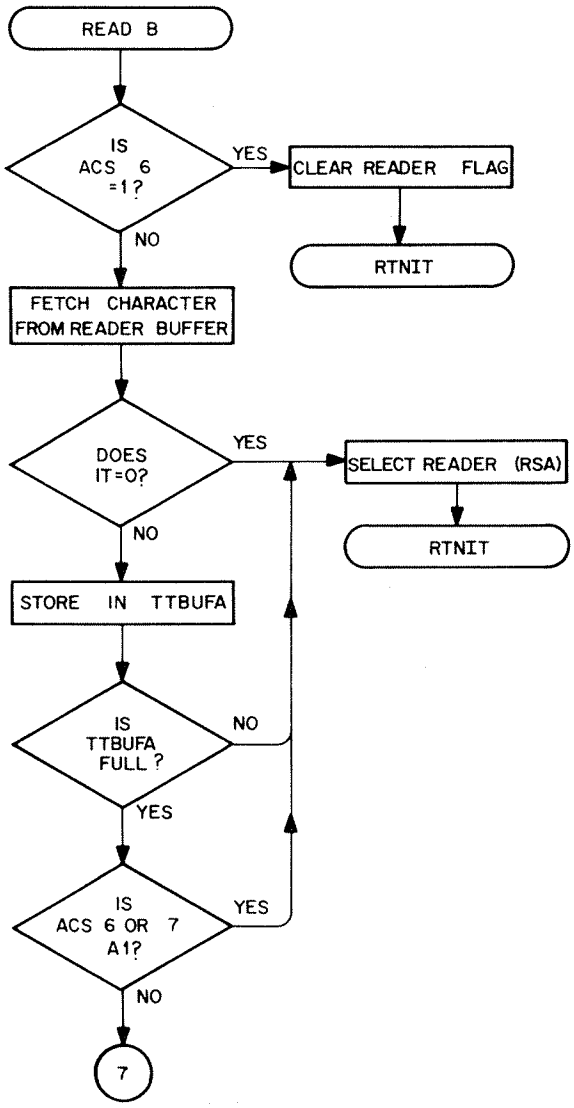
Service TTY Interrupt



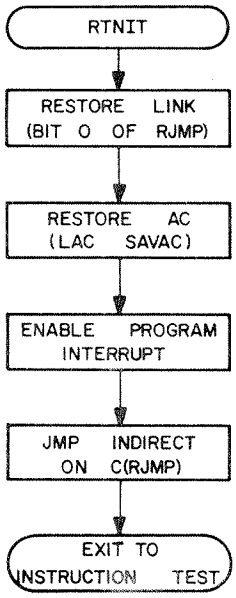
Punch Routine

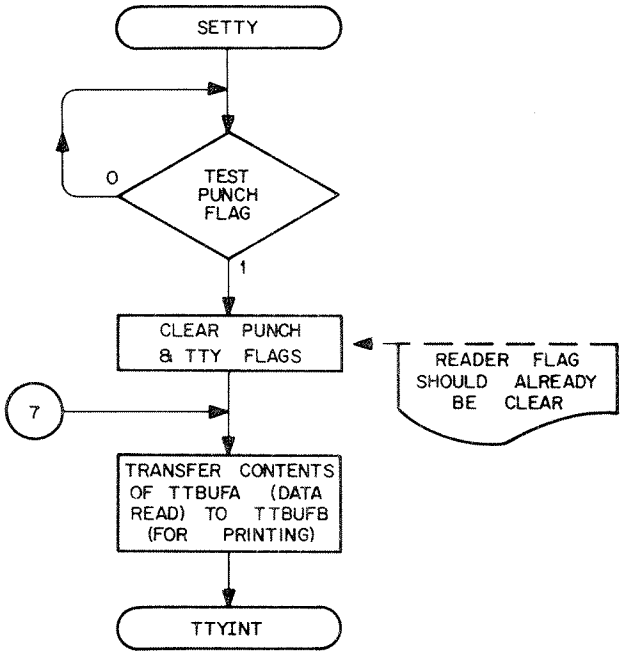


NOTE:
COUNT A AND COUNT B ARE COUNTERS USED BY THE PROGRAM TO ENABLE THE READER TO OPERATE AT PUNCH SPEED.



Read full speed ACS 7 A 1


RTNIT
RESTORE LINK (BIT 0 OF RJMP)
RESTORE AC (LAC SAVAC)
ENABLE PROGRAM INTERRUPT
JMP INDIRECT ON C(RJMP)
EXIT TO INSTRUCTION TEST


SETTY
TEST PUNCH FLAG
0
1
CLEAR PUNCH & TTY FLAGS
READER FLAG SHOULD ALREADY BE CLEAR
7
TRANSFER CONTENTS OF TTBUFA (DATA READ) TO TTBUFB (FOR PRINTING)
TTYINT